Available online at www.sciencedirect.com
ScienceDirect
Computers & Fluids 36 (2007) 313–324
computers & fluids
www.elsevier.com/locate/compfluid

An immersed boundary technique for simulating complex flows with rigid boundary

Shen-Wei Su [a], Ming-Chih Lai [b], Chao-An Lin [a,*]

[a] Department of Power Mechanical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan
[b] Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan

Received 7 March 2005; received in revised form 8 July 2005; accepted 29 September 2005
Available online 19 January 2006

Abstract

A new immersed boundary (IB) technique for the simulation of flow interacting with solid boundary is presented. The present formulation employs a mixture of Eulerian and Lagrangian variables, where the solid boundary is represented by discrete Lagrangian markers embedding in and exerting forces to the Eulerian fluid domain. The interactions between the Lagrangian markers and the fluid variables are linked by a simple discretized delta function. The numerical integration is based on a second-order fractional step method under the staggered grid spatial framework. Based on the direct momentum forcing on the Eulerian grids, a new force formulation on the Lagrangian marker is proposed, which ensures the satisfaction of the no-slip boundary condition on the immersed boundary in the intermediate time step. This forcing procedure involves solving a banded linear system of equations whose unknowns consist of the boundary forces on the Lagrangian markers; thus, the order of the unknowns is one-dimensional lower than the fluid variables. Numerical experiments show that the stability limit is not altered by the proposed force formulation, though the second-order accuracy of the adopted numerical scheme is degraded to 1.5 order. Four different test problems are simulated using the present technique (rotating ring flow, lid-driven cavity and flows over a stationary cylinder and an in-line oscillating cylinder), and the results are compared with previous experimental and numerical results. The numerical evidences show the accuracy and the capability of the proposed method for solving complex geometry flow problems both with stationary and moving boundaries.

1. Introduction

The fluid–solid interaction problems are frequently encountered in many engineering applications. When the immersed object is complex and moving, the problem poses two difficulties; namely, the accuracy of the fluid computation and the treatment of complex boundary. This accuracy of fluid calculations can be improved by employing high resolution schemes [1–3] or using adaptive or moving mesh techniques such as in [4–7] to resolve the small scale structures. The other difficulty arises from the handling of the interaction between the fluid and the immersed body which is the major issue of this work. Traditionally, the body-fitted or unstructured grid methods are adopted to simulate the flow with complex rigid boundary. However, the computational cost and memory requirements of these methods are generally high. Also, the modeling of complex time evolving moving boundary flows requiring transient re-meshing strategies increases further the computational overhead of these methods.

Another simple alternative to tackle the complex fluid–structure problem is to use the Cartesian grid instead of body-fitted grid. Since the boundary now does not conform with the Cartesian grid, the main difficulties are to model the complex immersed boundary accurately and at the same time maintain the solution efficiency discretized on the Cartesian grids. One approach, the so-called Cartesian grid method [8], tracks the boundary by identifying the control volumes (cut cell) in the Cartesian grid that are cut by the immersed boundary and determines the intersection of the boundary with the faces of these cut cells. More

* Corresponding author.
E-mail address: calin@pme.nthu.edu.tw (C.-A. Lin).

0045-7930/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2005.09.004

specifically, a control volume containing the boundary is reshaped by discarding the region that lies within the solid and this results in the trapezoidal shape control volume. Due to the irregular shapes of the cut cells, complex interpolating procedures to approximate the fluxes must be introduced and this unavoidably affects the computational efficiency of the Cartesian solvers.

Instead of using the cut cell approach, one approach also within the Cartesian grid framework, the immersed interface method (IIM) has attracted attention recently for the simulation of fluid flows in irregular regions. This method was originally developed by LeVeque and Li [9] for solving elliptic equations with discontinuous coefficients and singular sources along an arbitrary interface. A derived jump conditions are employed near the boundary, so that the sharp solutions can be achieved. The implementation of the IIM has been applied within the vorticity and stream function formulation [10,11]. Also, based on the work of Li and Lai [12] for the Navier–Stokes equations with singular forces, the immersed interface method using the Navier Stokes equation with primitive variables [13,14] has also been explored. A recent review on the immersed interface method can be found in [15] by Li.

The complex geometry within the Cartesian grid can be simulated by generating external force field to mimic the immersed boundary. The immersed boundary (IB) method proposed by Peskin [16,17], has been applied successfully to blood–valve interaction and other biological problems. The proposed immersed boundary formulation employs a mixture of Eulerian and Lagrangian variables, where the immersed boundary is represented by a set of discrete Lagrangian markers embedding in the Eulerian fluid domain. Those markers can be treated as force generators to the fluid, and meanwhile move along with the fluid. The interaction between the Lagrangian markers and the fluid variables defined on the fixed Eulerian grid is linked by a well-chosen discretized delta function. Although the IB method is developed to handle mostly the fluid problem with elastic structures, it has also been used to simulate the flow with rigid boundaries or structures. Lai and Peskin [18] proposed a new formally second-order accurate immersed boundary method to simulate flows past a rigid cylinder. The rigid boundary is simply mimicked by a stiff spring connecting the boundary points to their target positions and the boundary force is generated by the deviation of the markers from their target positions. Since this feedback force is computed and distributed into the fluid at the beginning of each time step, the problem becomes very stiff which has the consequence of the small time step. Besides, the force generates oscillations along the surface of the boundary.

Also adopting the mixture of Eulerian and Lagrangian variables, Goldstein et al. [19,20] proposed the virtual boundary method to simulate the flow with solid boundary within the spectral method framework. The solid boundary is treated as a force generator where the force field is calculated by a feedback method based on the difference between the predicted velocity and the actual velocity of the boundary. However, in order to prevent the generation of the spurious oscillations in simulating the start-up flow around a cylinder, the magnitude of the CFL number has been kept below $O(10^{-2})$. The feedback force of Goldstein et al. was also adopted by Saiki and Biringen [21] using the finite difference formulation. The distribution of the boundary force to the grid points was achieved by the area-weighted average function. This same function is used to interpolate the fluid velocity to the boundary points. A very good agreement has been found between their numerical computations and experimental results for the simulation of flow around a cylinder. However, the time step restriction of the method is similar to the scheme proposed by Goldstein et al. [19] and an excessive number of surface points $\mathcal{O}(1000)$ were used to model the immersed boundary even for two-dimensional flows.

In [23], Silva et al. calculated the momentum forcing along the immersed boundary using the pressure and velocity derivatives interpolated by a second-order Lagrange polynomial approximation. Once the momentum forcing at the boundary points has been calculated, they are distributed over the Eulerian grid by the discrete delta function. The whole numerical scheme is very similar to the immersed boundary method of Lai and Peskin except the evaluation of the forcing term. While Silva et al.'s approach is ideally simple, the calculations of momentum forcing at the boundary points are quite complex.

Without adopting the Lagrangian markers, Mohd-Yusof [24] proposed a direct forcing method within the spectral framework, where direct momentum forcing is applied to a set of points adjacent to the surface and interior to the body, which is equivalent to the direct imposition of the velocity boundary conditions on the Eulerian grids to ensure the no-slip condition at the immersed boundary. Therefore, information regarding the locations of the Eulerian grids either external or internal to the immersed boundary must be determined. The major advantage of the method is that since the force field is directly computed from the momentum equations, so that the time step can be larger than the previous methods. Fadlun et al. [25] further extended the Mohd-Yusof approach to a finite-difference formulation on a stagger gird system, where direct forcing or velocity boundary condition is applied along the first Eulerian grid external to the immersed boundary. Although this velocity boundary condition is obtained by linear interpolation procedure, the choice of interpolation direction is arbitrary.

Based on the concept of direct forcing, Kim et al. [26] introduced both the momentum forcing and mass source/sink to properly represent the immersed body. The momentum forcing and the mass source/sink are applied only on the body surface or inside the body so that the no-slip boundary condition on the immersed boundary and the continuity for the cell containing the boundary are both satisfied. Since the immersed boundary in general does not coincide with the grid points, an interpolation scheme for computing the momentum forcing must be employed.

Tseng and Ferziger [27] extended the idea of Fadlun et al. via a ghost cell approach. The immersed boundary is represented by piecewise linear segments and the ghost cells are defined to lie just inside the body but adjacent to computational grids of the fluid domain. The values of fluid variables at those ghost cells are obtained by extrapolation using a local quadratic scheme which involves the neighboring flow nodes and the associated velocity boundary condition.

It can be summarized that the major drawback of the existing Eulerian Lagrangian based IB techniques to simulate solid boundary flows is the restriction of small CFL number due to the spring or feedback force formulations employed, despite the simple interpolation procedure used to link the Lagrangian marker and the Eulerian grids. On the other hand, though the existing direct forcing approaches can compute flows using a larger time step, the determinations of the forcing locations and their magnitudes may not be straightforward on the Eulerian grids, especially for time evolving moving boundary flows. Combining the merits of the above two approaches, a new immersed boundary (IB) technique for the simulation of flows interacting with solid boundary is proposed.

The present formulation employs a mixture of Eulerian and Lagrangian variables, where the solid boundary is represented by discrete Lagrangian markers embedding in and exerting forces to the Eulerian fluid domain. Based on the direct momentum forcing on the Eulerian grids, a new force formulation on the Lagrangian marker is proposed, which ensures the satisfaction of the no-slip boundary condition on the immersed boundary. The interactions between the Lagrangian markers and the fluid variables on the fixed Eulerian grid are linked by a simple discrete delta function. The boundary forces are first computed on the Lagrangian markers and then distributed to the Eulerian grids using discrete delta function. As will be shown in the next section that the present formulation does not need to employ any modifications due to the jump conditions such as in the immersed interface method, so the present implementation is simpler but can be potentially less accurate.

Four different test problems are simulated using the present technique (rotating ring flow, lid-driven cavity and flows over a stationary cylinder and an in-line oscillating cylinder). These cases are used to examine the numerical accuracy of the method and its capability to model complex flows including the time evolving moving boundary. The results are compared with previous experimental and numerical results to assess the accuracy and the capability of the proposed method for solving complex geometry flow problems.

2. The methodology of immersed boundary technique

2.1. Mathematical formulation

Here, consider a problem of a viscous incompressible fluid in a two-dimensional square domain $\Omega = [0, L] \times [0, L]$ containing an immersed massless boundary in the form of a simple closed curve Γ, as shown in Fig. 1. The lowercase and uppercase letters are used to represent the variables defined on the Cartesian grid and the Lagrangian markers, respectively. The immersed boundary is tracked by the parametric form, $\mathbf{X}(s)$, $0 \leqslant s \leqslant L_b$, where s is the parameter of the reference configuration of the boundary. (For simplicity, the boundary configuration is assumed to be fixed in time.) As mentioned before, the influence of the immersed boundary to the fluid is represented by forces that exert to the fluid so that the fluid will move by the prescribed velocity $\mathbf{U}(\mathbf{X}(s), t)$ (no-slip condition) at the immersed boundary. Thus, the governing equations of this fluid–structure interaction system are

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{u}\mathbf{u}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\mathbf{f}(\mathbf{x}, t) = \int_0^{L_b} \mathbf{F}(\mathbf{X}(s), t)\delta(\mathbf{x} - \mathbf{X}(s))\,\mathrm{d}s, \tag{3}$$

$$\mathbf{U}(\mathbf{X}(s), t) = \int_\Omega \mathbf{u}(\mathbf{x}, t)\delta(\mathbf{x} - \mathbf{X}(s))\,\mathrm{d}\mathbf{x}. \tag{4}$$

Here, $\mathbf{x} = (x, y)$, $\mathbf{U}(\mathbf{x}, t)$ is the fluid velocity, $p(\mathbf{x}, t)$ is the fluid pressure, and the number Re is the non-dimensional Reynolds number. Note that, the Eulerian force field $\mathbf{f}(\mathbf{x}, t)$ is defined on the whole fluid domain Ω while the Lagrangian force $\mathbf{F}(\mathbf{X}(s), t)$ is on the immersed boundary Γ.

Eqs. (1) and (2) are the familiar Navier–Stokes equations of a viscous incompressible fluid. Eqs. (3) and (4) represent the interaction between the immersed boundary and the fluid. In particular, Eq. (3) describes that the force acting on the fluid $\mathbf{f}$ is due to the boundary force $\mathbf{F}$ alone, and Eq. (4) represents the fluid moves with the same prescribed velocity of the immersed boundary. One can easily see that the present formulation employs a mixture of Eulerian ($\mathbf{x}$) and Lagrangian ($\mathbf{X}$) variables which are linked by the

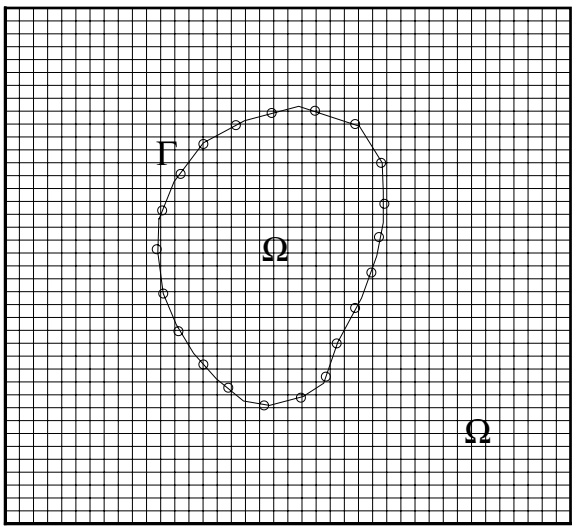

Fig. 1. Flow domain (Ω) with an immersed boundary (Γ).

two-dimensional Dirac delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)$. To close the system, the initial and the physical boundary ($\partial\Omega$) conditions of velocity should be given as well.

The main difficulty of the above mathematical formulation is that the forcing term $\mathbf{f}$ is not known a priori (since the boundary force $\mathbf{F}$ is unknown).The force field $\mathbf{F}$, however, can be determined by enforcing the no-slip boundary condition of the immersed boundary.

2.2. Numerical scheme

The present numerical scheme is a mixed Eulerian–Lagrangian finite difference method for simulating the complex fluid flows interacting with an immersed boundary. Therefore, two distinct discretized grids are introduced; the regular lattice points to cover the whole fluid domain, and the marker points to discretize the immersed boundary. Throughout this work, the spatial discretization makes use of the staggered marker-and-cell (MAC) mesh introduced by Harlow and Welsh [28]. Thus, the fluid variables are defined at different locations of the lattice grids. For instance, the pressure is defined on the grid points labeled as $\mathbf{x} = (x_i, y_j) = ((i-1/2)h, (j-1/2)h)$ for $i,j = 1,2,\ldots,N$, the velocity components u and v are defined at $(x_{i-1/2}, y_j) = ((i-1)h, (j-1/2)h)$ and $(x_i, y_{j-1/2}) = ((i-1/2)h, (j-1)h)$, respectively, where the spacing $h = \Delta x = \Delta y = L/N$. On the other hand, the immersed boundary is discretized by the M discrete Lagrangian markers $\mathbf{X}_k = (X_k, Y_k)$, where the marker spacing is $\Delta s = L_b/M$.

In general, the markers $\mathbf{X}_k$ do not coincide with the Eulerian grid points $\mathbf{x}$, so a discrete delta function linkage between those two grids must be employed. Here, the following discrete version of Dirac delta function is used:

$$\delta_h(\mathbf{x} - \mathbf{X}_k) = d_h(x_i - X_k) d_h(y_j - Y_k), \tag{5}$$

where d_h is the hat function [22] defined by

$$d_h(r) = \begin{cases} (1 - |r|/h)/h, & \text{for } |r| \le h \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The adopted delta function is equivalent to the bilinear interpolation scheme, which is consistent with the interpolation procedure employed in the Navier–Stokes solver.

Based on the adopted grid arrangement, the governing equations (1)–(4) can be discretized directly. The integrals in Eqs. (3) and (4) are approximated by their Riemann sums and the Dirac delta function δ is replaced by its discrete version δ_h in 5. On the other hand, Eqs. (1) and (2) are integrated using the fractional-step method proposed by Choi and Moin [29], and the non-linear convection term and the diffusion term are treated by the Adams–Bashforth and the Crank–Nicholson methods, respectively. All the spatial derivatives are approximated by the second-order central difference method. At the beginning of the time step, the solutions $\mathbf{u}^{n-1}$, $\mathbf{u}^n$ must be given in order to march to $\mathbf{u}^{n+1}$.

The time advancement and spatial discretization can be done in the following steps:

$$\frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} = -\frac{3}{2}\nabla_h(\mathbf{uu})^n + \frac{1}{2}\nabla_h(\mathbf{uu})^{n-1} - \nabla_h p^n + \frac{1}{2Re}\nabla_h^2(\mathbf{u}^n + \tilde{\mathbf{u}}), \tag{7}$$

$$\mathbf{f}^*(\mathbf{x}) = \sum_{k=1}^{M} \mathbf{F}^*(\mathbf{X}_k)\delta_h(\mathbf{x} - \mathbf{X}_k)\Delta s, \quad \text{for all } \mathbf{x} = (x_i, y_j) \tag{8}$$

$$\frac{\mathbf{u}^* - \tilde{\mathbf{u}}}{\Delta t} = \mathbf{f}^* \tag{9}$$

$$\mathbf{U}^*(\mathbf{X}_k) = \sum_{\mathbf{x}} \mathbf{u}^*(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}_k)h^2, \quad k = 1,2,\ldots,M. \tag{10}$$

$$\frac{\mathbf{u}^{**} - \mathbf{u}^*}{\Delta t} = \nabla_h p^n, \tag{11}$$

$$\nabla_h^2 p^{n+1} = \frac{\nabla_h \cdot \mathbf{u}^{**}}{\Delta t}, \tag{12}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{**}}{\Delta t} = -\nabla_h p^{n+1}, \tag{13}$$

where $\tilde{\mathbf{u}}$, $\mathbf{u}^*$, and $\mathbf{u}^{**}$ are the intermediate velocity components between the time step n and $n+1$. The discrete operators ∇_h and ∇_h^2 represent the regular centered difference approximations for the gradient and Laplace operators.

Eq. (8) describes that the Eulerian forces are obtained from distributing the boundary forces at those Lagrangian markers to their neighboring Cartesian grids. Eq. (9) represents the direct momentum forcing to modify the velocities on the Eulerian grids using the forces obtained from Eq. (8). Eq. (10) says that the velocity at the markers can be interpolated from the neighboring grids and should equal to the prescribed boundary velocity. Note that, the summation $\sum_{\mathbf{x}}$ in (10) is only over four neighboring grid points of $\mathbf{X}_k$, since the support of the discrete delta function is as wide as the mesh width h. Also, the above approximation in (10) is nothing but the bi-linear interpolation of the velocity. The continuity condition is satisfied by solving the pressure Poisson equation (12).

It should be noted the discrete momentum forcing in the present algorithm seems to be calculated by the formula $\mathbf{f}^* - \Delta t\nabla_h^2\mathbf{f}^*/(2Re)$ rather than $\mathbf{f}^*$. This is due to the implicit treatment of the intermediate velocity in the prediction step (7). This confusion can be avoided by simply using the explicit Adams–Bashforth discretization for the diffusion term but it will lead to a restrictive time step constraint when Reynolds number is small. Despite the above momentum forcing discrepancy, the physical quantities such as the drag and lift coefficients (as shown in the numerical results) will not be affected. This is because, in the present formulation, the lift and drag are calculated by the discrete sum of momentum forces over the whole fluid grid points and the discrete Laplacian term makes no contribution in the summation. This can be derived easily as follows. (Note that, $\mathbf{f}^*$ is non-zero only in the neighborhood of the immersed boundary.)

$$\sum_i \sum_j \nabla_h^2 \mathbf{f}^* = \sum_j \sum_i \left(\frac{f^*_{i+1,j} - f^*_{i,j}}{h^2} - \frac{f^*_{i,j} - f^*_{i-1,j}}{h^2} \right) + \sum_i \sum_j \left(\frac{f^*_{i,j+1} - f^*_{i,j}}{h^2} - \frac{f^*_{i,j} - f^*_{i,j-1}}{h^2} \right) = 0.$$

The above is verified numerically by computing flows over a cylinder employing the Adams–Bashforth and Crank–Nicholson schemes, respectively, and identical lift and drag coefficients are obtained.

In the present numerical setting, the support of $\mathbf{f}^*$ is only in the neighborhood of the immersed boundary Γ. Also, the force field is computed in the intermediate step, so that the solutions of coupling linear systems can be avoided. The most time-consuming part of the above scheme is to solve three Helmholtz-type equations (two for velocity in (7) and one for the pressure in (12)) which can be solved by the efficient Bi-CGSTAB method [30].

The remaining question is how to find the boundary force field $\mathbf{F}^*$ properly in Eq. (8), so that the resulting velocity can be adjusted by the presence of the forces. More precisely, the present forcing field will bring the intermediate velocity $\mathbf{u}^*$ to the desired velocity $\mathbf{U}^*$ at the immersed boundary. It is important to mention that the present scheme (7)–(13) is very similar to the direct momentum forcing of Mohd-Yusof used in [24,25] except in Eq. (8). Since the present formulation employs a mixture of Eulerian and Lagrangian variables, the boundary forces are first computed on the Lagrangian markers and then are distributed to the Eulerian grids using discrete delta function, which depends on the distances between the Eulerian grids and the Lagrangian markers. Therefore, there is no need to determine the relative locations of the Eulerian grid either internal or external to the immersed boundary, and this further simplifies the computational procedure.

2.3. The boundary force derivation

Here, description as how to find the marker forces $\mathbf{F}^*$ properly is introduced, so that the intermediate velocity $\mathbf{u}^*$ at Lagrangian markers will satisfy the desired boundary values $\mathbf{U}^*$. Firstly, the velocity $\tilde{\mathbf{u}}$ calculated in Eq. (7) is interpolated to the markers to obtain the marker velocity $\widetilde{\mathbf{U}}(\mathbf{X}_k)$, as

$$\widetilde{\mathbf{U}}(\mathbf{X}_k) = \sum_{\mathbf{x}} \tilde{\mathbf{u}}(\mathbf{x}) \delta_h(\mathbf{x} - \mathbf{X}_k) h^2. \tag{14}$$

If the above interpolation procedure is applied to Eq. (9) (i.e. direct momentum forcing on Eulerian grids) directly, then

$$\sum_{\mathbf{x}} \mathbf{f}^*(\mathbf{x}) \delta_h(\mathbf{x} - \mathbf{X}_k) h^2 = \frac{\mathbf{U}^*(\mathbf{X}_k) - \widetilde{\mathbf{U}}(\mathbf{X}_k)}{\Delta t}, \tag{15}$$

where $\mathbf{U}^*(\mathbf{X}_k)$ is the interpolating velocity of $\mathbf{u}^*$ at the marker $\mathbf{X}_k$. By simply letting the marker velocity $\mathbf{U}^*(\mathbf{X}_k) = \mathbf{U}^{n+1}(\mathbf{X}_k)$, the force field $\mathbf{f}^*$ will bring the velocity $\mathbf{u}^*$ in the intermediate step of the Navier–Stokes solver to the prescribed values at the immersed boundary.

Note that, it is the intermediate velocity $\mathbf{U}^*$ rather than the final velocity $\mathbf{U}^{n+1}$ satisfying the prescribed boundary velocity, so that the prediction step (Eq. 7) can be decoupled from the projection step (Eqs. (12) and (13)) completely in the Navier–Stokes integration. It is possible to let the final velocity satisfy the prescribed boundary condition by doing a few iterations at each time step, but this will complicate the present implementation.

Combining Eqs. (8) and (15), the Lagrangian boundary force can be obtained as

$$\sum_{\mathbf{x}} \underbrace{\sum_{j=1}^{M} \mathbf{F}^*(\mathbf{X}_j) \delta_h(\mathbf{x} - \mathbf{X}_j) \Delta s}_{\mathbf{f}^*(\mathbf{x})} \delta_h(\mathbf{x} - \mathbf{X}_k) h^2 = \frac{\mathbf{U}^{n+1}(\mathbf{X}_k) - \widetilde{\mathbf{U}}(\mathbf{X}_k)}{\Delta t}. \tag{16}$$

Therefore

$$\sum_{j=1}^{M} \left(\sum_{\mathbf{x}} \delta_h(\mathbf{x} - \mathbf{X}_j) \delta_h(\mathbf{x} - \mathbf{X}_k) \Delta s h^2 \right) \mathbf{F}^*(\mathbf{X}_j) = \frac{\mathbf{U}^{n+1}(\mathbf{X}_k) - \widetilde{\mathbf{U}}(\mathbf{X}_k)}{\Delta t}. \tag{17}$$

There are M equations for the M Lagrangian marker forces $\mathbf{F}^*(\mathbf{X}_k), k = 1, 2, \ldots, M$. Since the support of the discrete delta function is the same as the grid spacing h, the above coefficient matrix in (17) is banded. (For a closed boundary, it is a periodic banded matrix.) The matrix can become explicit if $\delta_h(\mathbf{x} - \mathbf{X}_j) = 0$, when $j \neq k$. This only happens in regions where the Eulerian grids influenced by Lagrangian marker force $\mathbf{F}^*(\mathbf{X}_k)$ are not affected by Lagrangian marker forces $\mathbf{F}^*(\mathbf{X}_j)$, where $j = 1, M$ and $j \neq k$. For example, the Lagrangian markers may collocate with the Eulerian grids or reside on the edges of the Eulerian grids. While the latter may seem plausible, the restriction is that the grid points connected to the edge should not be affected by other marker forces, which may not be trivial and straightforward.

For simplicity, the markers are generally uniformly distributed along the immersed boundary. As will be shown later, the optimum marker spacing (Δs) scales with the grid spacing h. Therefore, the number of markers (M) required is one dimensional lower than the fluid variables and is the same order of magnitude to the Eulerian grid points surrounding the immersed boundary either on the fluid or the solid sides. Also, for two dimensional flows, the banded matrix is tria- or penta-diagonal. Therefore, the cost of solving Eq. 17 is $\mathcal{O}(M)$, which is negligibly small compared to the cost of solving the momentum and the pressure poisson equations. It should also be noted that a singular matrix may occur if the markers are too close to each other. Numerical experiments show that if the ratio of the marker spacing to the grid width $\Delta s/h$ is greater than 0.5, no singular matrix will be found, and the optimum marker spacing (Δs) scales with the grid spacing h. This further excludes the

possibility of the singular matrix and reduces the sizes of the banded force matrix.

2.4. The full solution procedure

A whole numerical procedure for each time step of the present method is summarized as follows.

(1) Solve Eq. (7) in order to obtain the intermediate velocity component $\tilde{\mathbf{u}}(\mathbf{x})$.
(2) Calculate the Lagrangian boundary force $\mathbf{F}^*(\mathbf{X}_k)$ for all markers using Eq. (17) and the prescribed marker velocity $\mathbf{U}^{n+1}(\mathbf{X}_k)$.
(3) Distribute the Lagrangian force $\mathbf{F}^*(\mathbf{X}_k)$ to the Eulerian grid $\mathbf{f}^*(\mathbf{x})$ using Eq. (8).
(4) Correct the intermediate velocity $\mathbf{u}^*(\mathbf{x})$ using the newly obtained force $\mathbf{f}^*(\mathbf{x})$ through Eq. (9).
(5) Obtain the intermediate velocity $\mathbf{u}^{**}(\mathbf{x})$ using Eq. (11).
(6) Compute the pressure by solving the pressure Poisson equation (12).
(7) Update the fluid velocity $\mathbf{u}^{n+1}(\mathbf{x})$ by Eq. (13).

3. Numerical results

3.1. Flow induced by a rotating ring

Here, the flow considered is a ring rotating at a constant angular velocity locating at the center of a square domain $-1 \leqslant x, y \leqslant 1$. The radius of the ring is 0.3, and the angular velocity Ω of the ring is 2, where the corresponding Reynolds number is 18. Since the square domain boundary is stationary, the flow is induced by the rotating ring. The velocity boundary conditions are imposed along the ring using the immersed boundary method. Since the immersed boundary does not coincide with the Cartesian grids, the forcing interpolation procedure discussed in the previous section is required to obtain the velocity correction near the immersed boundary. At the steady state, the flow inside the ring would undergo a solid body rotation, if the flow remains laminar. Therefore, this can provide a means of examining the numerical accuracy of the present scheme.

Simulations are conducted using four sets of grids, i.e. 40×40, 80×80, 160×160 and 320×320. The grid spacing ratio $\Delta s/h$ is 0.85, and the maximum CFL number is approximately 0.5. Fig. 2 shows the velocity vector plots of the flow, where the rotating ring is marked with the bold line. The solid body rotation can be clearly observed inside the rotating ring. The L_1 and L_2 error norms are shown in Fig. 3. The spatial accuracy of the present scheme is around 1.5 order, though at higher grid density the order of accuracy does increase. Since this case is a discontinuous problem using the simple bi-linear interpolation procedure, the order of accuracy achieved is rather encouraging.

The time of the immersed boundary calculation relative to the overall cost is also examined here using the rotating ring flow as the test bed. The immersed boundary method includes four parts, namely the distribution of the markers, the calculation of the interpolation function, the determination of the maker forces and the distributions of marker forces to the Eulerian grids. The measurements based on the 320×320 grid indicates that the immersed boundary method occupies slightly less than 1% of the overall computational cost, and this is rather small. Most of the computational cost is on the solution of the pressure Poisson equation, which is about 90% of the overall time. For stationary cylinder, the distribution of the markers and the calculation of the interpolation function are calculated only once, and this further reduces the cost of the IBM operation.

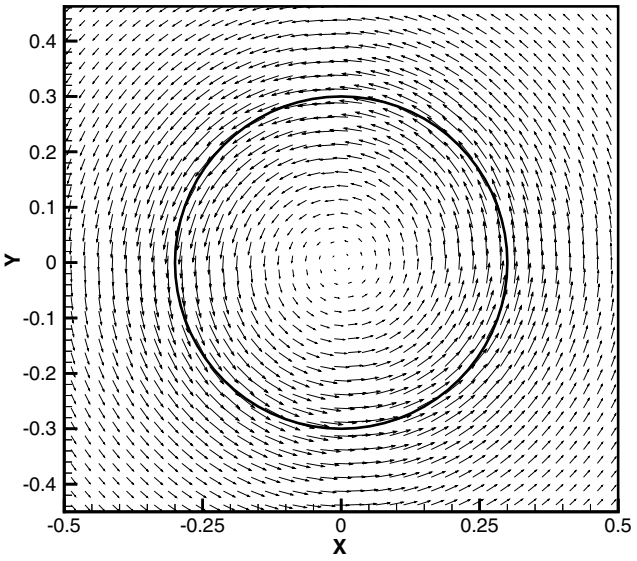

Fig. 2. Velocity vector of flow induced by a rotating ring within a square domain.

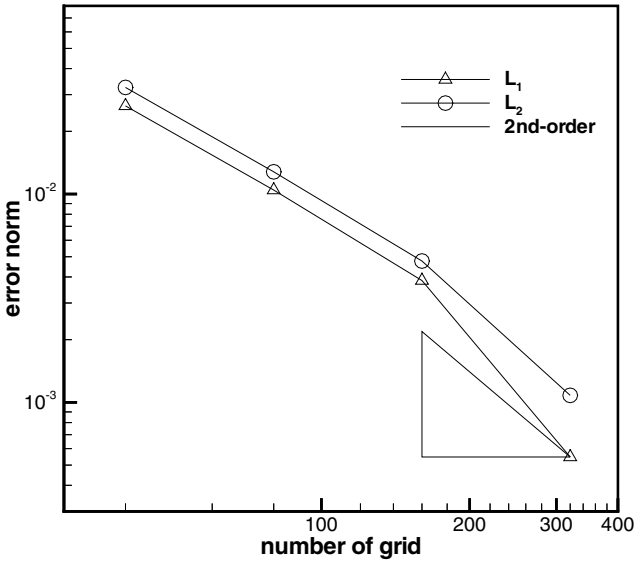

Fig. 3. L_1 and L_2 error norms of the rotating ring flow.

3.2. Lid-driven cavity

The lid-driven cavity flow is widely used to verify the accuracy of numerical method too. Fig. 4 shows the geometric layout of the lid-driven cavity within a square domain $[-1,1]\times[-1,1]$, where the cavity ($|x|+|y|=1/\sqrt{2}$) is obliqued at 45° with respect to the Eulerian grid. Thus, the width of the cavity is 1 and the lid locates along the edge $x+y=1/\sqrt{2}$ with constant driven velocity one. This inclined lid-driven cavity is mimicked by the present immersed boundary method, where the respective velocity distributions are assigned at Lagrangian markers. No-slip conditions are prescribed along the computational boundary of the Eulerian grids.

Three different uniform grids ($N\times N, N=40,80,160$) are used in the simulations with the associated number of Lagrangian markers ($M=80,160,320$), respectively. The grid spacing ratio of marker points and the lattice points $\Delta s/h$ is 1. The Reynolds number is chosen as $Re=100$ and the maximum CFL number is 0.4.

Fig. 5 shows the steady velocity component U inside the cavity but along the line $y=x$. The coordinate $Y=0$ represents the intersection of the lower left edge and $y=x$. Fig. 6 shows the steady velocity component V inside the cavity but along the line $y=-x$. The coordinate $X=0$ represents the intersection of the upper left edge and $y=-x$. One can see that present numerical results converge to the result obtained by Ghia et al. [31] quite well.

To demonstrate the capability of the present scheme to mimic flows with multiple objects, the simulation is applied to flow within the lid-driven cavity with five circular obstacles. The size of the computational domain is the same as the previous one, and the Reynolds number is 100 based on the lid-driven velocity and the height of the cavity.

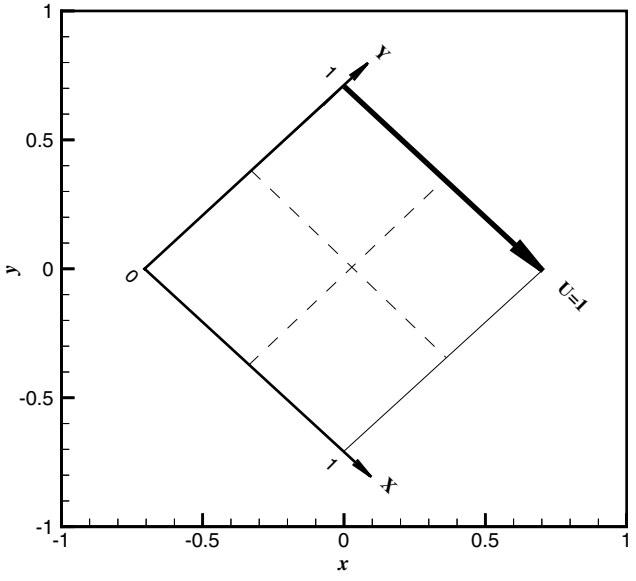

Fig. 4. The configuration of the lid-driven cavity in the computational domain.

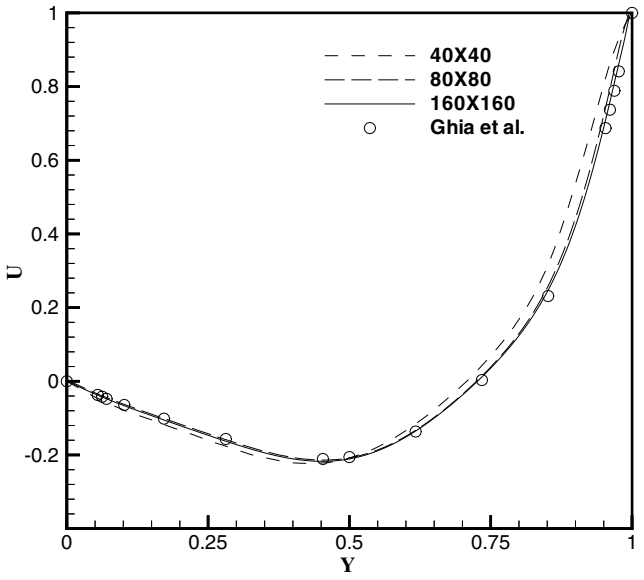

Fig. 5. The steady velocity component U along the line ($y=x$) through the center of cavity with different Eulerian grid sizes.

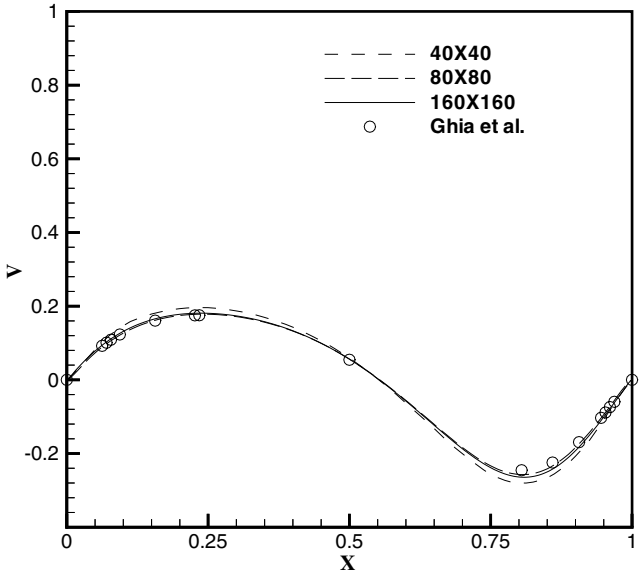

Fig. 6. The velocity component V along the central line ($y=-x$) through the center of cavity with different Eulerian grid sizes.

The grid spacing ratio $\Delta s/h$ is 0.85 within a 160×160 grid. Fig. 7 shows the vector plot of the simulation, where the original lid-driven cavity flow structures have been modified by the presence of the obstacles.

3.3. The flow past a cylinder

The flow past a stationary circular cylinder is a typical problem and has been widely investigated experimentally [32,33] and numerically [8,18,23,26,34–36]. For Reynolds number below 47, the flow structure remains symmetric with stationary recirculating vortices behind the cylinder.

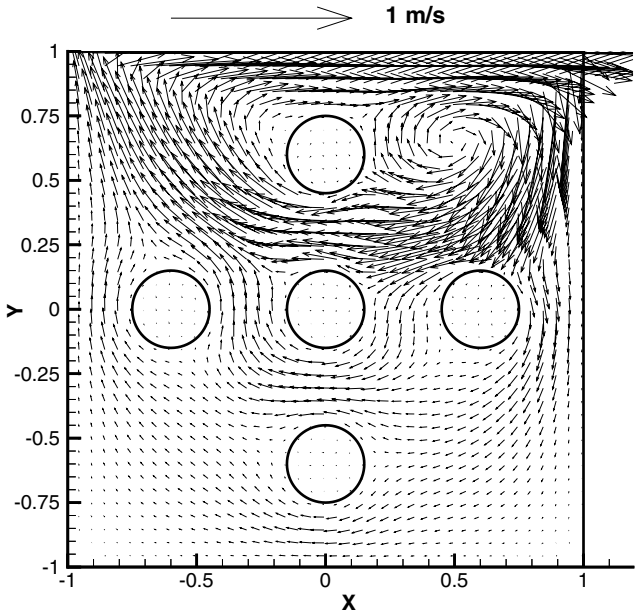

Fig. 7. The velocity vector plot of lid-driven cavity with multi-obstacles.

As the Reynolds number is elevated, the symmetry breaks down and the vortex starts to shed up and down alternatively. This shedding frequency and the intensity of the vortex also increase in tandem with the elevated level of the Reynolds number.

In this test, flows with a broad range of Reynolds number, $Re = 20$, 40, 80, 100 and 150 are examined. The geometric set up in the computational domain and the associate physical boundary conditions are shown in Fig. 8. Based on the diameter D of the cylinder, the range of the computational domain chosen is $(-13.4D \leqslant x \leqslant 16.5D, -8.35D \leqslant y \leqslant 8.35D)$. A non-uniform grid (250×160) is adopted to discretize the computational domain, within which a uniform grid 60×60 is employed in the region $-D \leqslant x, y \leqslant D$. Here, the maximum CFL number employed is 0.37. In the simulation, the boundary of the stationary cylinder is modeled by the present immersed boundary methods, where the Lagrangian markers are assigned the no-slip boundary conditions.

There are three different quantities that are often used to make a comparison of the performance of numerical methods; namely, the drag and lift coefficients, and the Strouhal

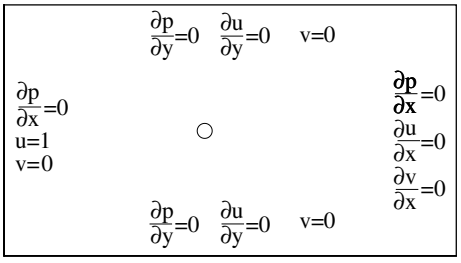

Fig. 8. The boundary condition and computational domain of flow over cylinder.

number. The definitions and computing of these quantities are briefly described below.

The drag coefficient is defined as

$$C_{\mathrm{D}} = \frac{F_{\mathrm{D}}}{U_{\infty}^2 D/2} \tag{18}$$

where F_{D} is the drag force. As in [18], the drag force can be obtained easily from

$$F_{\mathrm{D}} = -\int_{\Omega} f_1(\mathbf{x})\,\mathrm{d}\mathbf{x} = -\sum_{\mathbf{x}} f_1(\mathbf{x})h^2 \tag{19}$$

where $f_1(\mathbf{x})$ is the x component of the Eulerian force $\mathbf{f}(\mathbf{x})$.

Similarly, the lift coefficient is defined as

$$C_{\mathrm{L}} = \frac{F_{\mathrm{L}}}{U_{\infty}^2 D/2} \tag{20}$$

where F_{L} is the lift force and it can be obtained by the Eulerian force as

$$F_{\mathrm{L}} = -\int_{\Omega} f_2(\mathbf{x})\,\mathrm{d}\mathbf{x} = -\sum_{\mathbf{x}} f_2(\mathbf{x})h^2 \tag{21}$$

where $f_2(\mathbf{x})$ is the y component of the Eulerian force $\mathbf{f}(\mathbf{x})$.

The other interesting quantity called the Strouhal number (St) is the dimensionless frequency of vortex shedding. When the flow field becomes unstable, the originally stationary vortices behind the cylinder start moving downstream and shedding alternatively with frequency f_q. The Strouhal number is defined as $St = f_q D/U_{\infty}$.

The influences of $\Delta s/h$ on the predicted lift and drag coefficients and Strouhal number are shown in Table 1, where the Reynolds number is 100. It can be seen that the differences of C_{D} and St resulting from the variations of $\Delta s/h$ ratios are within 3%. It is also observed that $\Delta s/h = 0.94$ and 0.78 produce the similar results. Therefore, in subsequent computations for flows past a cylinder, the $\Delta s/h$ is chosen as 0.94.

Table 2 shows the comparison of drag coefficient with previous numerical predictions [8,18,23,26,34,35] and experimental measurements [33] at different Reynolds numbers. In [34,35], the flows were assumed to be steady and symmetric, therefore only the results of $Re = 20$ and 40 are shown here. The present drag predictions at lower Reynolds numbers are slightly higher than others, but at higher Reynolds numbers they are comparable with other results. Table 3 shows the comparison of lift coefficient for $Re = 100$. Here, C_{L} is defined as the maximum amplitude

Table 1
Influences of the marker spacing to grid width on the lift and drag coefficients and Strouhal number for the case of $Re = 100$

$\Delta s/h$	0.94	0.78	0.67	0.58	0.52
C_{L}	0.34	0.34	0.30	0.31	0.32
C_{D}	1.40	1.41	1.38	1.42	1.42
St	0.168	0.168	0.168	0.165	0.165

Table 2
The comparison of drag coefficients for different Reynolds numbers

Re	Present	Lai and Peskin [18]	Kim et al. [26]	Silva et al. [23]	Ye et al. [8]	Fornberg [34]	Ingham and Tang [35]	Tritton (exp) [33]
20	2.20	–	–	2.04	2.03	2.000	1.995	2.22
40	1.63	–	1.51	1.54	1.52	1.498	–	1.48
80	1.43	–	–	1.40	1.37	–	–	1.29
100	1.40	1.44	1.33	1.39	–	–	–	–
150	1.39	1.44	–	1.37	–	–	–	–

Table 3
The comparison of maximum lift coefficient at $Re = 100$

$Re = 100$	Present	Lai and Peskin [18]	Kim et al. [26]
C_L	0.34	0.33	0.32

Table 4
The comparison of Strouhal number for different Reynolds numbers

Re	Present	Lai and Peskin [18]	Silva et al. [23]	Ye et al. [8]	Williamson (exp) [32]
80	0.153	–	0.15	0.15	0.15
100	0.168	0.165	0.16	–	0.166
150	0.187	0.184	0.18	–	0.183

of the lift coefficient. Table 4 shows the comparison of Strouhal number with the previous numerical and experimental results for different Reynolds numbers. One can see that we obtained the close results with those methods as mentioned in the introduction.

For the flow of $Re = 40$, the drag coefficient remains constant and the lift is zero. The vorticity contour lines of the flow is shown in Fig. 9. It has confirmed the experimental observation that at this Reynolds number, there are two vortices attached behind the cylinder and the lift force is constantly zero. However, if the Reynolds number becomes larger, the symmetry breaks down and the two vortices shed alternatively. Fig. 10(a)–(b) shows the evolution of the drag and lift coefficients, and the vorticity contour lines for $Re = 100$.

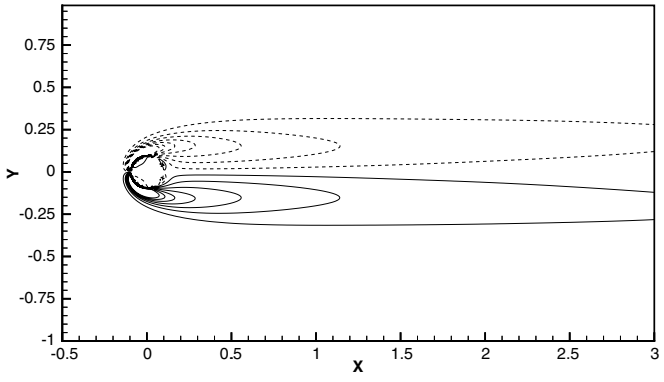

Fig. 9. The instantaneous vorticity contours near the cylinder; dotted and solid lines denote negative and positive contours.

4. The flow past an in-line oscillating cylinder

In order to explore the capability of the present technique in computing moving boundary flows, simulations are further applied to the in-line oscillating cylinder in uniform flow at Reynolds number 100. The present flow layout and numerical details are exactly the same as the flow past a stationary cylinder in the previous section, except the cylinder is now oscillating parallel to the free stream at a frequency $f_c = 2f_q$, i.e. two times the vortex shedding frequency (f_q) of the fixed cylinder flow. The motion of the cylinder is prescribed by setting the horizontal velocities on the Lagrangian markers to $U(X_k) = 0.14D\cos(2\pi f_c t)$, where the amplitude of the oscillation is 0.14 of the cylinder diameter D. This flow has been examined numerically by Hurlbut et al. [37] using the non-inertia coordinate transformation technique, and the results are used to examine the accuracy of the present predictions using the immersed boundary method.

According to Tanida et al. [38] and Griffin, Ramberg [39] and Hurlbut et al. [37], the in-line oscillation of the cylinder at a range of frequencies near twice the Strouhal shedding frequency for the stationary cylinder causes the synchronization, i.e. the phase-locking of the vortex shedding with the cylinder motion. Both the drag and lift coefficients increase. Table 5 shows the comparisons of the present predictions with the numerical results by Hurlbut et al. [37]. It can be seen that the present results are compatible with their results. Further examination of the vortex patterns is referred to the instantaneous vorticity contours over two oscillating periods of the cylinder as shown in Fig. 11.

5. Conclusion

In this paper, a new immersed boundary technique is proposed for the simulation of two-dimensional viscous incompressible flow interacting with complex solid boundary. A mixture of Eulerian and Lagrangian variables is adopted, where the solid boundary is represented by discrete Lagrangian markers embedding in and exerting forces to the Eulerian fluid domain. The interactions between the Lagrangian markers and the fluid variables are linked by a simple discretized delta function. The numerical integration is based on a second-order fractional step method under the staggered grid spatial framework. Based on the direct momentum forcing on the Eulerian grids, a new

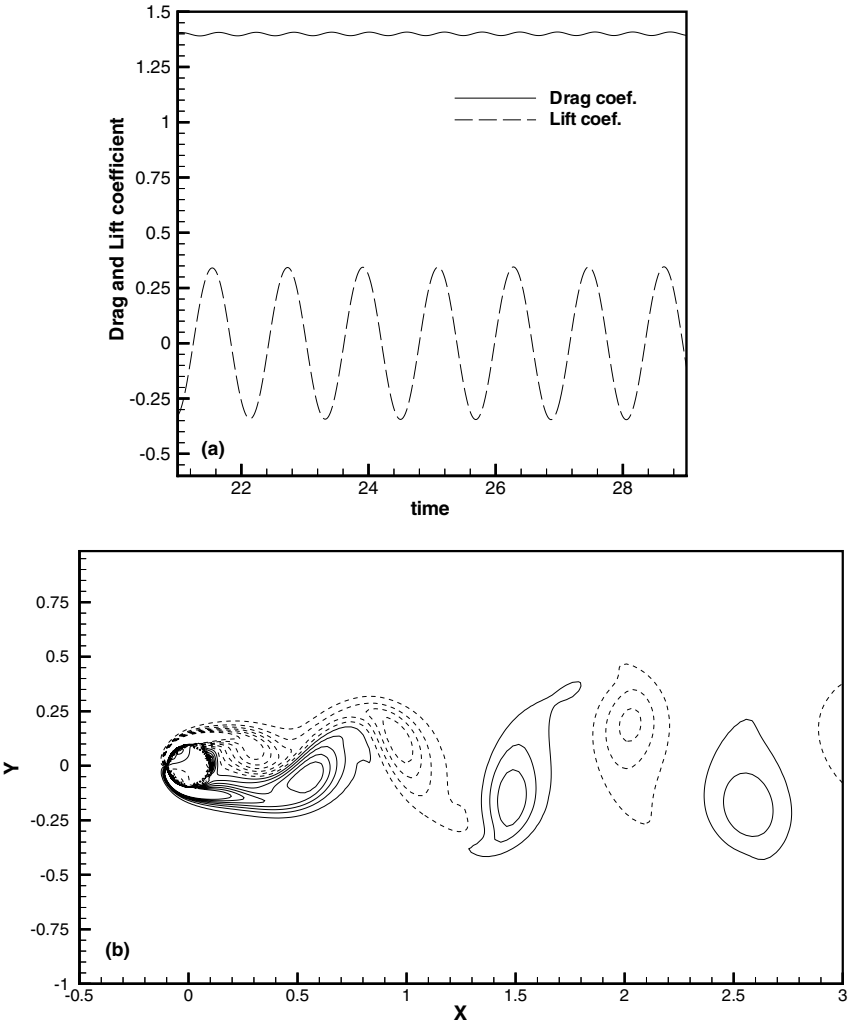

Fig. 10. (a) The time evolution of drag and lift coefficients of $Re = 100$ and (b) the instantaneous vorticity contours near the cylinder; dotted and solid lines denote negative and positive contours.

Table 5
The comparisons of lift and drag coefficients of in-line oscillating cylinder at Reynolds number 100

	Present		Hurlbut et al. [37]	
f_o/f_q	0	2	0	2
C_D	1.4	1.70	1.41	1.68
$C_{\text{L-max}}$	0.34	0.97	0.31	0.95

force formulation on the Lagrangian marker is proposed, where the boundary forces are first computed on the Lagrangian markers and then distributed to the Eulerian grids using discrete delta function to ensure the satisfaction of the no-slip boundary condition on the immersed boundary in the intermediate step. This forcing procedure involves solving a banded linear system of equations whose unknowns consist of the boundary force on the Lagrangian markers; thus, the order of the unknowns is one-dimensional lower than the fluid variables. Four different test problems are simulated using the present technique (rotating ring flow, lid-driven cavity and flows over a stationary cylinder and an in-line oscillating cylinder). Numerical experiments show that the stability limit is not altered by the proposed force formulation, though the second-order accuracy of the adopted numerical scheme is degraded to 1.5 order. Influences of the marker spacing on the solutions are also examined numerically, and the optimal value is found to scale with the grid width. The numerical evidences show the accuracy and the capability of the proposed method for solving complex geometry flow problems both with stationary and moving boundary.

Acknowledgements

This research work was supported by the National Science Council of Taiwan under grant NSC-93-2212-E-007-036 and the computational facilities were provided by the National Centre for High-Performance Computing of Taiwan which the authors gratefully acknowledge.

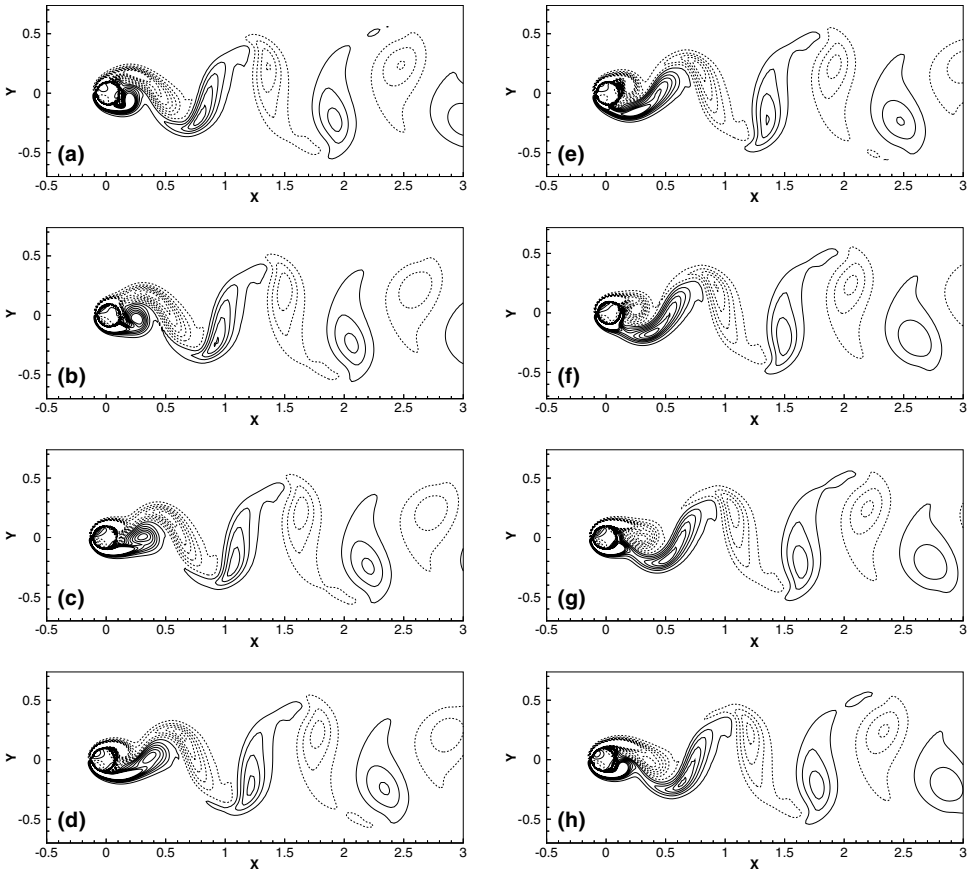

Fig. 11. The instantaneous vorticity contours near the oscillating cylinder—$Re = 100$; dotted and solid lines denote negative and positive contours: (a) $t = T/4$, (b) $t = T/2$, (c) $t = 3T/4$, (d) $t = T$, (e) $t = 5T/4$, (f) $t = 3T/2$, (g) $t = 7T/4$ and (h) $t = 2T$ (T is the oscillation period of the cylinder).

References

[1] Osher S, Shu CW. High-order essentially non-oscillatory schemes for Hamilton-Jacobi equations. SIAM J Numer Anal 1991;28:907.

[2] Jiang GS, Shu CW. Efficient implementation of weighted ENO schemes. J Comp Phys 1996;126:202.

[3] Lin CH, Lin CA. Simple high-order bounded convection scheme to model discontinuities. AIAA J 1997;35:563.

[4] Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. J Comp Phys 1982;46:342.

[5] Dvinsky AS. Adaptive grid generation from Harmonic maps on Riemannian-Manifolds. J Comp Phys 1991;95:450.

[6] Li R, Tang T, Zhang PW. Moving mesh methods in multiple dimensions based on harmonic maps. J Comp Phys 2001;170:562.

[7] Di Y, Li R, Tang T, Zhang PW. Moving mesh finite element methods for the incompressible Navier–Stokes equations. SIAM J Sci Comput 2005;26:1036.

[8] Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. J Comp Phys 1999;156:209.

[9] LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J Numer Anal 1994;31:1019.

[10] Calhoun D. A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. J Comp Phys 2002;176:231.

[11] Li Z, Wang C. A fast finite difference method for solving Navier–Stokes equations of irregular domains. Commun Math Sci 2003;1:180.

[12] Li Z, Lai M-C. The immersed interface method for the Navier–Stokes equations with singular forces. J Comp Phys 2001;171:822.

[13] Le DV, Khoo BC, Peraire J. An immersed interface method for the incompressible Navier–Stokes equations in irregular domains. Proceedings of the third MIT conference on computational fluid and solid mechanics, 710. Elsevier Science; 2005.

[14] Xu S, Wang ZJ. An immersed interface method for simulating the interaction of a fluid with moving boundaries. preprint; 2005.

[15] Li Z. An overview of the immersed interface method and its applications. Taiwanese J Math 2003;7:1.

[16] Peskin CS. Flow patterns around heart valves: a numerical method. J Comp Phys 1972;10:252.

[17] Peskin CS. The immersed boundary method. Acta Numer 2002;1.

[18] Lai M-C, Peskin CS. An immersed boundary method with formal second order accuracy and reduced numerical viscosity. J Comp Phys 2000;160:705.

[19] Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow with an external force field. J Comp Phys 1993;105:354.

[20] Goldstein D, Hadler R, Sirovich L. Direct numerical simulation of turbulent flow over a modeled riblet covered surface. J Fluid Mech 1995;302:333.

[21] Saiki EM, Biringen S. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. J Comp Phys 1996;123:450.

[22] Beyer RP, LeVeque RL. Analysis of a one-dimensional model for the immersed boundary method. SIAM J Numer Anal 1992;29:332.

[23] Lima E Silva ALF, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. J Comp Phys 2003;189:351.

[24] Mohd-Yusof J. Combined immersed boundary/B-Spline method for simulations of flows in complex geometries in complex geometries CTR annual research briefs. NASA Ames/Stanford University; 1997.

[25] Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary methods for three dimensional complex flow simulations. J Comp Phys 2000;161:30.
[26] Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. J Comp Phys 2001;171:132.
[27] Tseng Y-H, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. J Comp Phys 2003;192:593.
[28] Harlow FH, Welsh JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. Phys Fluids 1965;8:2181–9.
[29] Choi H, Moin P. Effects of the computational time step on numerical solutions of turbulent flow. J Comp Phys 1993;113:1.
[30] Van den Vorst HA, Sonneveld P. CGSTAB, a more smoothly converging variant of CGS, Technical Report 90-50, Delft University of Technology; 1990.
[31] Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes Equations and a multi-grid method. J Comp Phys 1982;48:387.
[32] Williamson CHK. Vortex dynamics in the cylinder wake. Ann Ver Fluid Mech 1996;28:477.
[33] Triton DJ. Experiments on the flow past a circular cylinder at low Reynolds number. J Fluid Mech 1959;6:547.
[34] Fornberg B. A numerical study of steady viscous flow past a circular cylinder. J Fluid Mech 1980;98:819.
[35] Ingham DB, Tang T. A numerical investigation into the steady flow past a rotating circular cylinder at low and intermediate Reynolds numbers. J Comp Phys 1990;87:91.
[36] Tang T, Ingham DB. On steady flow past a rotating circular cylinder at Reynolds numbers 60 and 100. Comput Fluids 1991;19:217.
[37] Hurlbut SE, Spaulding ML, White FM. Numerical solution for laminar two dimensional flow about a cylinder oscillating in a uniform stream. Trans ASME, J Fluids Eng 1982;104:214.
[38] Tanida Y, Okajima A, Watanabe Y. Stability of a circular cylinder oscillating in uniform flow or in a wake. J Fluid Mech 1973;61:769.
[39] Griffin OM, Ramberg SE. Vortex shedding from a cylinder vibrating in line with an incident uniform flow. J Fluid Mech 1976;75:257.