Journal of Computational Physics 233 (2013) 295–314

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations

Jinmo Lee [a], Donghyun You [a,b,*]

[a] Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[b] Department of Mechanical Engineering, Pohang University of Science and Technology, 77 Cheongam-Ro, Nam-Gu, Pohang, Gyungbuk 790-784, South Korea

ARTICLE INFO

Article history:
Received 7 November 2011
Received in revised form 18 August 2012
Accepted 28 August 2012
Available online 18 September 2012

Keywords:
Immersed boundary method
Ghost-cell method
Mass source and sink
Moving body problems
Spurious force oscillations

ABSTRACT

A fully-implicit ghost-cell immersed boundary method for simulations of flow over complex moving bodies on a Cartesian grid is presented. The present immersed boundary method is highly capable of controlling the generation of spurious force oscillations on the surface of a moving body, thereby producing an accurate and stable solution. Spurious force oscillations on the surface of an immersed moving body are reduced by alleviating spatial and temporal discontinuities in the pressure and velocity fields across non-grid conforming immersed boundaries. A sharp-interface ghost-cell immersed-boundary method is coupled with a mass source and sink algorithm to improve the conservation of mass across non-grid conforming immersed boundaries. To facilitate the control for the temporal discontinuity in the flow field due to a motion of an immersed body, a fully-implicit time-integration scheme is employed. A novel backward time-integration scheme is developed to effectively treat multiple layers of fresh cells generated by a motion of an immersed body at a high CFL number condition. The present backward time-integration scheme allows to impose more accurate and stable velocity vectors on fresh cells than those interpolated. The effectiveness of the present fully-implicit ghost-cell immersed boundary method coupled with a mass source and sink algorithm for reducing spurious force oscillations during simulations of moving body problems is demonstrated in a number of test cases.

1. Introduction

In recent years, immersed boundary methods have received special attention for simulations of fluid flow in complex configurations or over moving bodies [1–16,18–20]. Compared to methods using body-conforming grids, immersed boundary methods usually utilize a simple Cartesian grid and impose necessary boundary conditions on non-grid conforming surfaces of immersed bodies by assigning proper body forces. The use of an immersed boundary method significantly reduces the computational cost and time by avoiding repeated generation of body-fitted grids during moving-body simulations.

Immersed boundary methods are categorized into two major groups [21]. Methods in the first group utilize continuous body forces [2–8], which are added to the non-discretized Navier–Stokes equations. The continuous forcing immersed boundary methods have been popular for simulations of the interaction between fluid flow and elastic immersed structures. However, it is reported that representation of sharp boundaries and rigid objects are difficult for immersed boundary methods based on continuous forcing because of the restriction imposed by numerical stability [21]. In order to maintain the

[*] Corresponding author at: Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. Tel.: +1 412 268 6808; fax: +1 412 268 3348.
E-mail address: dhyou@cmu.edu (D. You).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.08.044

numerical stability, the continuous forcing methods model interfaces between the fluid and objects with diffuse boundaries and allow a relatively small time-step size for integration of the governing equations.

Methods in the second group correspond to discrete forcing immersed boundary methods and impose discrete body forces on the computational cells close to the surfaces of immersed bodies to realize necessary boundary conditions [1,9–11,13–16,18–20]. Immersed boundary methods in this category allow to use a larger time-step size than that allowed by continuous forcing immersed boundary methods and, with a proper treatment of discrete forcing, permit sharp representation of immersed boundaries [18,20].

However, discrete forcing immersed boundary methods are generally known to suffer from the generation of spurious force or pressure oscillations on the surface of a moving immersed body [1,15,16,20]. Uhlmann [15] attempted to reduce spurious force oscillations by combining a direct forcing method of Fadlun et al. [10] and discrete delta functions of Peskin [2]. To control the spurious force oscillations during a moving body simulation, Yang and Balaras [16] proposed an elaborate extrapolation technique, which assigns the velocity and pressure to grid cells where solid becomes fluid due to a body motion. Later, Yang and Balaras' extrapolation procedure was simplified by Yang and Stern [17]. Recently, Lee et al. [1] identified two major sources of spurious force oscillations through a numerical experiment. One source is related to the spatial discontinuity in the pressure field across the immersed boundary where grid cells, which were previously inside an immersed body, become fluid cells due to a body motion. The other source of spurious force oscillations is associated with the temporal discontinuity in the velocity field where fluid cells turn into solid cells with a body motion. Lee et al. also found that spurious force oscillations tend to be reduced by decreasing the grid spacing and increasing the time-step size. Seo and Mittal [20] identified the primary cause of spurious force oscillations as the non-conservation of fluid mass near the immersed boundary, which is a result of the violation of geometric conservation across the immersed boundary. Seo and Mittal reported that a combination of a sharp-interface immersed boundary method and a cut cell method significantly improves the geometric conservation, thereby reducing spurious force oscillations.

Motivated by the recent understanding of sources of spurious force oscillations, in the present study, a new immersed boundary method which provides highly enhanced capability for controlling the generation of spurious force oscillations in moving-body simulations, is developed. The present method is based on a ghost-cell method [18] coupled with a mass source/sink algorithm [11]. A mass source/sink method is used to better conserve the fluid mass in the vicinity of immersed boundaries, thereby reducing one of the sources of spurious force oscillations [1,20].

Unlike the original ghost-cell method [18], in the present method, the governing equations are integrated in time using a fully implicit method. A fully-implicit time integration of the governing equations allows to use a broad range of ratios of the time-step size to the grid spacing, thereby providing enhanced control of spurious force oscillations. However, a special treatment for multiple layers of fresh cells, due to the use of a large time-step size, is required. To effectively and accurately deal with multiple layers of fresh cells, a novel fresh-cell treatment algorithm based on a backward time-integration scheme is also developed. As will be shown in the following sections, the present fully-implicit ghost-cell immersed boundary method coupled with a mass source/sink algorithm is found to effectively suppress spurious force oscillations during simulations of flow over moving bodies.

In the following section, the present numerical methods including a fully-implicit time-integration method, a ghost-cell immersed boundary method with mass sources/sinks, and a backward time-integration method for assigning velocity vectors to fresh cells, are described in detail. In Section 3, results from test simulations for demonstrating the spatial and temporal accuracy of the method and the effectiveness of the new methodology for reducing spurious force oscillations, are presented. Finally, concluding remarks are followed in Section 4.

2. Numerical methods

2.1. Fully-implicit time-integration method

The magnitude of spurious force oscillations on immersed boundaries is reported to be proportional to $\Delta x^2/\Delta t$, where Δx and Δt are the grid spacing and the time-step size, respectively [1]. Using explicit or semi-implicit time-integration schemes limits the permissible range of the time-step size to the grid spacing or the Courant–Friedrichs–Lewy (CFL) number due to a stability requirement. In the present study, to extend the operational range of the time-step size to the grid spacing, thereby enhancing the capability for controlling spurious force oscillations, the governing equations are integrated by using a fully-implicit time-integration scheme.

The numerical method is based on an incompressible Navier–Stokes solver developed by You et al. [22,23], with significant enhancements to treat complex moving geometries. The incompressible Navier–Stokes equations to be solved are as follows:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} u_i u_j = -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial}{\partial x_j} \frac{\partial u_i}{\partial x_j}, \tag{1}$$

$$\frac{\partial u_i}{\partial x_i} - q = 0, \tag{2}$$

where $i, j = 1, 2$, and 3, and u_i is the velocity component in the i-direction. All the coordinate variables, velocity components, and pressure are non-dimensionalized by a length scale L, a reference velocity U_∞, and ρU_∞^2, respectively. The time is normalized by L/U_∞. The Reynolds number is defined as $Re = U_\infty L/\nu$, where ν is the kinematic viscosity of the fluid. q in the continuity equation (2) is a mass source/sink term introduced to better conserve the fluid mass in the vicinity of immersed boundaries [11]. The pressure and velocity are arranged and discretized on a collocated grid as shown in Fig. 1.

A fractional-step method is used for time integration. A second-order Crank–Nicolson scheme is used for both the convective and diffusion terms. The discretized momentum equations for the intermediate velocity can be written as follows:

$$\frac{u_i^* - u_i^n}{\Delta t} + \frac{1}{2}[N_i^* + N_i^n] = -\frac{\delta p^n}{\delta x_i} + \frac{1}{2Re}(D_i^* + D_i^n), \tag{3}$$

where $N_i = \frac{\delta(U_j u_i)}{\delta x_j}$ and $D_i = \frac{\delta}{\delta x_j}\left(\frac{\delta u_i}{\delta x_j}\right)$ represent the convective and diffusive terms, respectively. $\frac{\delta}{\delta x_j}$ denotes a spatial discretization operator in the j-direction. An iterative method based on a line successive-over-relaxation scheme coupled with a Gauss–Seidel method as a smoother, is employed to solve the discretized non-linear momentum equations for the intermediate velocity field as follows:

$$u_i^{*,k+1} + \frac{\Delta t}{2} N_i^{*,k+1/2} - \frac{\Delta t}{2Re} D_i^{*,k+1} = u_i^n - \frac{\Delta t}{2} N_i^n - \Delta t \frac{\delta p^n}{\delta x_i} + \frac{\Delta t}{2Re} D_i^n, \tag{4}$$

where k is the iteration index and $N_i^{*,k+1/2} = \frac{\delta(U_j^{*,k} u_i^{*,k+1})}{\delta x_j}$.

Eq. (4) is iterated until it satisfies a convergence criterion. The face-center velocity U_j^* is computed as follows:

$$\begin{aligned}
\tilde{u}_i &= u_i^* + \Delta t \left(\frac{\delta p^n}{\delta x_i}\right)_{cc}, \\
\widetilde{U_1} &= \gamma_w \tilde{u}_{1P} + (1 - \gamma_w)\tilde{u}_{1W}, \\
\widetilde{U_2} &= \gamma_s \tilde{u}_{2P} + (1 - \gamma_s)\tilde{u}_{2S}, \\
\widetilde{U_3} &= \gamma_b \tilde{u}_{3P} + (1 - \gamma_b)\tilde{u}_{3B}, \\
U_j^* &= \widetilde{U}_j - \Delta t \left(\frac{\delta p^n}{\delta x_j}\right)_{fc},
\end{aligned} \tag{5}$$

where γ_w, γ_s, and γ_b are the linear interpolation weights for the face velocity components in the west, south, and back directions, respectively. Subscripts cc and fc represent the cell and face centers, respectively (see Fig. 1 for the definitions). The cell face pressure is calculated using the same linear interpolation technique used for calculating the cell face velocity. Then, the pressure gradient at a cell center is computed using the interpolated cell face pressures while the pressure gradient at a cell face is computed using the cell center pressures.

The velocity u_i^{n+1} is computed by the pressure-correction as follows:

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\frac{\delta p'}{\delta x_i}, \tag{6}$$

where $p' = p^{n+1} - p^n$. Eq. (6) and the divergence free condition for u_i^{n+1} result in a Poisson equation:

$$\frac{\delta}{\delta x_i}\left(\frac{\delta p'}{\delta x_i}\right) = \frac{1}{\Delta t}\left(\frac{\delta U_i^*}{\delta x_i} - q\right), \tag{7}$$

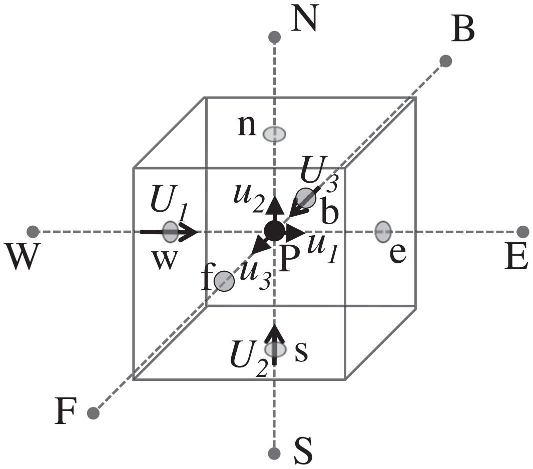


Fig. 1. Illustration of a collocated-grid arrangement of primitive variables.

where q is a mass source/sink term, of which inclusion is discussed in Section 2.2. The Poisson equation is solved using an algebraic multigrid method. Finally, the velocity and pressure are updated as follows:

$$\begin{aligned} p^{n+1} &= p^n + p', \\ u_i^{n+1} &= u_i^* - \Delta t\left(\frac{\delta p'}{\delta x_i}\right)_{cc}, \\ U_i^{n+1} &= U_i^* - \Delta t\left(\frac{\delta p'}{\delta x_i}\right)_{fc}. \end{aligned} \tag{8}$$

2.2. A ghost-cell immersed-boundary method with mass sources/sinks

The present discrete momentum forcing is based on a ghost-cell method proposed by Mittal et al. [18]. Complex geometries are represented with triangular surface elements. The inside and outside of a submerged object are identified by a dot product of a vector normal to the surface element and a position vector which extends to the center of a cell from the closest surface element. If the dot product produces a positive (negative) value, the node is identified as the outside (inside) of a submerged body.

Computational cells identified as the outside and inside of a submerged body are defined as fluid and solid cells, respectively, while cells located at the interface between fluid and solid cells are defined as ghost cells as illustrated in Fig. 2.

To impose necessary boundary conditions on the immersed boundary, proper velocity vectors are assigned to the ghost cells. A velocity vector on a ghost cell is interpolated with the values available at the associated image point (IP) and boundary intercept (BI) (see Fig. 2 for the definitions of IP and BI). Firstly, a velocity vector $\mathbf{v}_{IP}$ at an image point is computed by using velocity vectors at surrounding eight nodes in a three-dimensional configuration as follows:

$$\begin{aligned} \mathbf{v}_i(x_1,x_2,x_3)_i =& C_1(x_1x_2x_3)_i + C_2(x_1x_2)_i + C_3(x_2x_3)_i + C_4(x_1x_3)_i + C_5(x_1)_i + C_6(x_2)_i + C_7(x_3)_i + C_8, \\ & \text{for } i = 1,2,3,\ldots,8, \end{aligned} \tag{9}$$

where x_1, x_2, and x_3 are the Cartesian coordinates of the node. $()_i$ denotes a value evaluated at i-th node.

The eight unknown coefficients C_i can be determined by solving the following algebraic equation:

$$\{C\} = [V]^{-1}\{\mathbf{v}\}, \tag{10}$$

where

$$\begin{aligned} \{C\}^T &= \{C_1, C_2, \ldots, C_8\}, \\ \{\mathbf{v}\}^T &= \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_8\}, \\ [V] &= \begin{bmatrix} (x_1x_2x_3)_1 & (x_1x_2)_1 & (x_1x_3)_1 & (x_2x_3)_1 & (x_1)_1 & (x_2)_1 & (x_3)_1 & 1 \\ (x_1x_2x_3)_2 & (x_1x_2)_2 & (x_1x_3)_2 & (x_2x_3)_2 & (x_1)_2 & (x_2)_2 & (x_3)_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & & & & \\ (x_1x_2x_3)_8 & (x_1x_2)_8 & (x_1x_3)_8 & (x_2x_3)_8 & (x_1)_8 & (x_2)_8 & (x_3)_8 & 1 \end{bmatrix}. \end{aligned}$$

A velocity $\mathbf{v}_{IP}$ at an image point can be expressed using the computed C_i as follows:

$$\mathbf{v}_{IP} = \{C\}^T\{\beta\}, \tag{11}$$

where

$$\{\beta\}^T = \{(x_1x_2x_3)_{IP}, (x_1x_2)_{IP}, (x_1x_3)_{IP}, (x_2x_3)_{IP}, (x_1)_{IP}, (x_2)_{IP}, (x_3)_{IP}, 1\}.$$

Finally a velocity vector at a ghost cell, $\mathbf{v}_{GC}$, is calculated using velocity vectors at the image point and boundary intercept with a linear interpolation technique:

$$\mathbf{v}_{BI} = \frac{1}{2}(\mathbf{v}_{IP} + \mathbf{v}_{GC}) \tag{12}$$

or

$$\mathbf{v}_{GC} = 2\mathbf{v}_{BI} - \mathbf{v}_{IP}. \tag{13}$$

The above described interpolation technique is second-order accurate as shown in Mittal et al. [18].

The present method updates ghost-cell values during the iteration of the non-linear momentum equations for the intermediate velocity field, thereby better satisfying the necessary boundary conditions and the conservation of the fluid mass near the immersed boundary. In the present method, the conservation of the fluid mass in the vicinity of immersed boundaries is even strongly enforced by employing a mass source and sink algorithm.

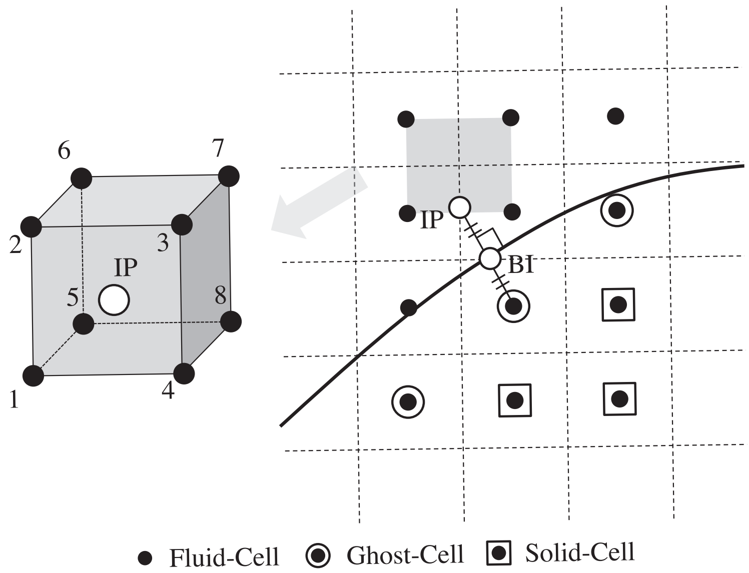


Fig. 2. Definitions of fluid, solid, and ghost cells in the present immersed boundary method. IP and BI indicate the image point and the boundary intercept, respectively.

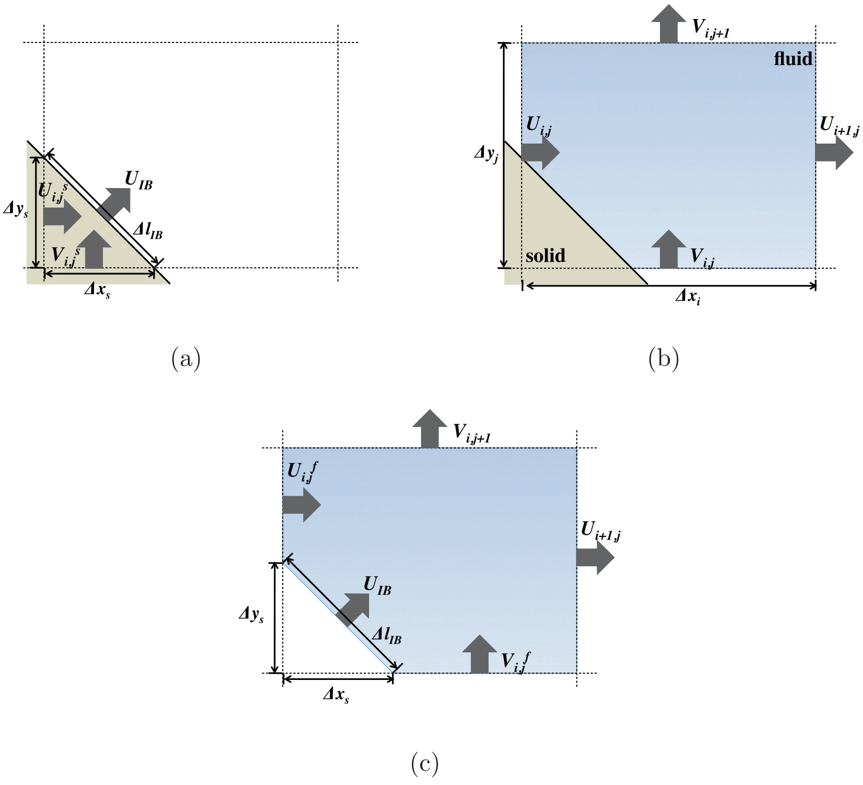


Fig. 3. Illustration of a mass source-sink algorithm for a moving immersed boundary.

The present mass source and sink method is based on the method proposed by Kim et al. [11] and is extended for moving body problems. Computational cells containing both fluid and solid fractions are marked as source/sink cells. To calculate necessary mass sources and sinks, the continuity equation for source/sink cells is split into fluid and solid parts as follows:

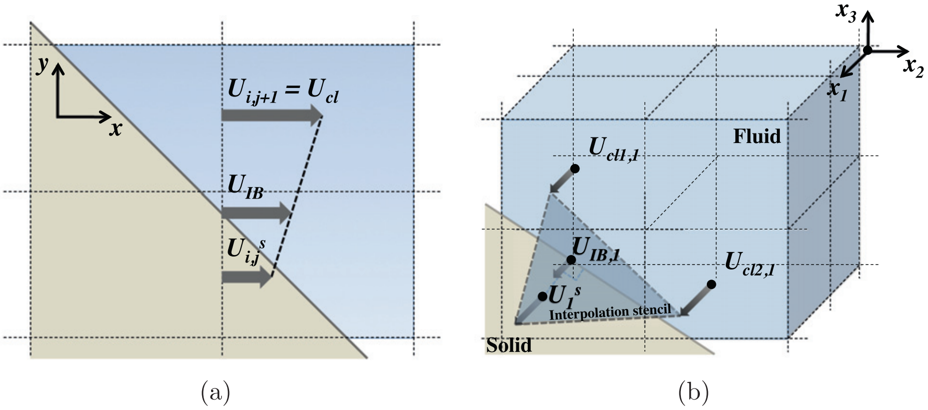


Fig. 4. Illustration of the interpolation method for determine the face velocity at the center of a solid faction in (a) two-dimensional and (b) three-dimensional cases, respectively.

$$\int_{\partial\Omega} \mathbf{v}\cdot\mathbf{n}\mathrm{d}\partial\Omega = \int_{\partial\Omega_f} \mathbf{v}\cdot\mathbf{n}\mathrm{d}\partial\Omega + \int_{\partial\Omega_s} \mathbf{v}\cdot\mathbf{n}\mathrm{d}\partial\Omega, \tag{14}$$

where $\partial\Omega$ and $\mathbf{n}$ are the surface of a mass source/sink cell and the surface normal vector, respectively. Subscripts f and s denote the fluid and solid parts, respectively.

Mass conservation for the fluid in a mass source/sink cell requires $\int_{\partial\Omega_f} \mathbf{v}\cdot\mathbf{n}\partial\Omega = 0$, thereby reducing Eq. (14) as follows:

$$\int_{\partial\Omega} \mathbf{v}\cdot\mathbf{n}\mathrm{d}\partial\Omega - \int_{\partial\Omega_s} \mathbf{v}\cdot\mathbf{n}\mathrm{d}\partial\Omega = 0. \tag{15}$$

Eq. (15) can be recast in a differential form:

$$\frac{\partial u_i}{\partial x_i} - q = 0, \tag{16}$$

where Ω is the volume of the mass source/sink cell and $q = \frac{\mathrm{d}}{\mathrm{d}\Omega}\int_{\partial\Omega_s} \mathbf{v}\cdot\mathbf{n}\mathrm{d}\partial\Omega$.

As shown in Fig. 3(a), in a two-dimensional configuration, the mass source/sink term per unit span can be expressed in a discrete form as follows:

$$q_{i,j} = \frac{1}{\Delta x_i \Delta y_j}\left(-U_{i,j}^s \Delta y_s - V_{i,j}^s \Delta x_s + U_{IB}\Delta l_{IB}\right), \tag{17}$$

where Δx_s and Δy_s are solid fractions of cell faces. Face velocities $U_{i,j}^s$, $V_{i,j}^s$ are located at the middle of solid fractions. U_{IB} is the velocity of the surface of a moving body and Δl_{IB} is the surface area of the immersed body in the cell per unit span. The regular control volume (Fig. 3(b)) for the computation of mass fluxes is modified by excluding the solid fraction from the control volume while the contribution of the immersed body is replaced with a mass source/sink term as shown in Fig. 3(c).

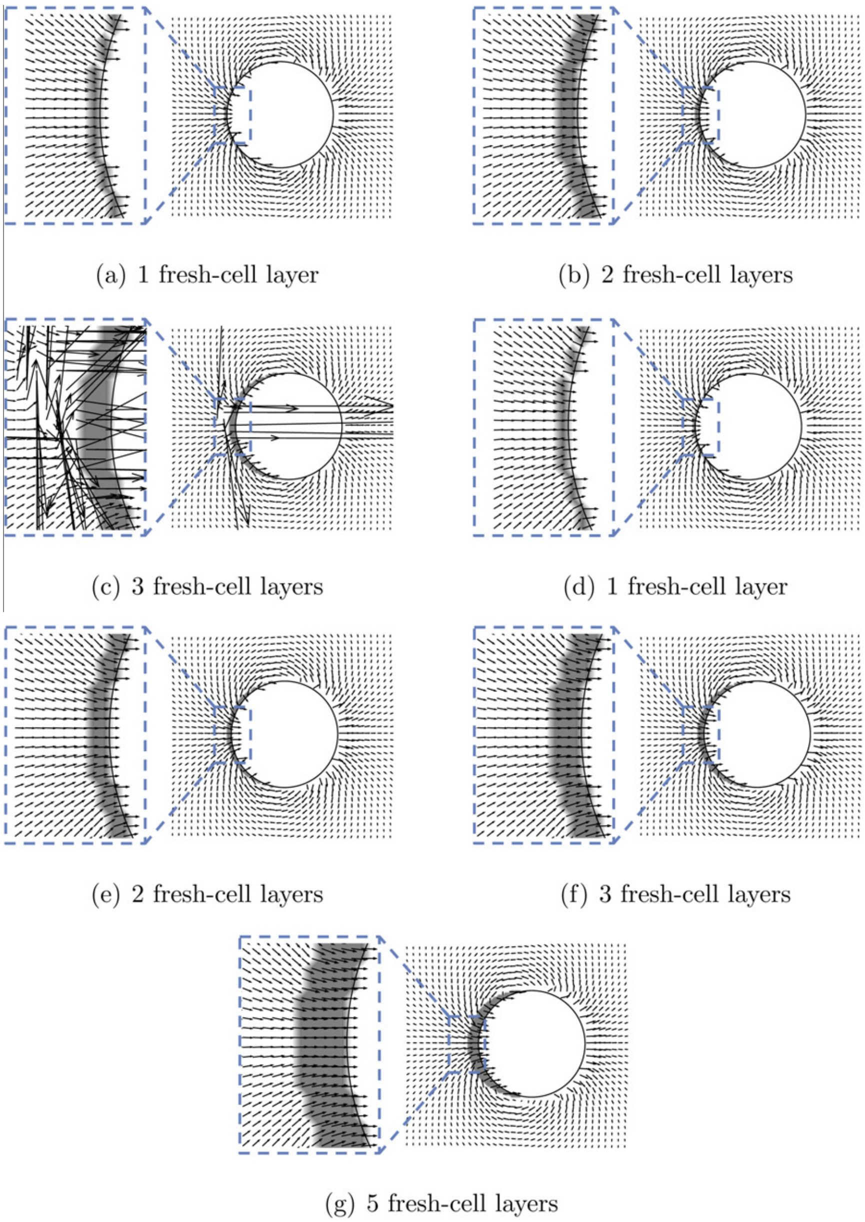
The mass source/sink term can be generalized to a three-dimensional configuration as follows:

$$q_{i,j,k} = \frac{1}{\Delta\Omega}\left(\sum \mathbf{U}_{i,j,k}^s \cdot \mathbf{n}_s \Delta\partial\Omega_s + \sum \mathbf{U}_{IB}\cdot\mathbf{n}_{IB}\Delta\partial\Omega_{IB}\right), \tag{18}$$

where $\Delta\Omega$, $\Delta\partial\Omega_s$, and $\Delta\partial\Omega_{IB}$ are the volume of the cell, solid fractions of cell faces, and the surface area of the immersed body in the cell, respectively. $\mathbf{n}_s$ and $\mathbf{n}_{IB}$ are surface outward normal vectors on the faces of solid fractions and immersed boundary, respectively. $\Delta\partial\Omega_s$ and $\Delta\partial\Omega_{IB}$ are calculated using polygons constructed by the intersection vertices on the cell edges and vertices inside the solid fraction.

The face velocity at the center of a solid fraction, $\mathbf{U}_{i,j,k}^s$, is interpolated from the target boundary condition and neighboring fluid cell values. For example, as shown in Fig. 4(a), in a two-dimensional configuration, the face velocity at the center of a solid fraction, $\mathbf{U}_{i,j}^s$, is determined by a linear interpolation of $\mathbf{U}_{IB}$ and the velocity at the closest cell face $\mathbf{U}_{cl}$. Similarly, in a three-dimensional configuration, $\mathbf{U}_{i,j,k}^s$ is interpolated from a boundary condition at the closest immersed boundary point $\mathbf{U}_{IB}$ and velocity vectors at two closest cell centers $\mathbf{U}_{cl,1}$ and $\mathbf{U}_{cl,2}$ as shown in Fig. 4(b). For stationary body problems, the present interpolation method for calculating a face velocity is close to a method of Huang and Sung [12]. $U_{i,j}^f$ and $V_{i,j}^f$ are determined using Eq. (5) and cell center velocities, which are obtained by solving governing equations with mass sources/sinks.

As will be discussed in Section 3, the inclusion of mass sources and sinks effectively reduce spurious force oscillations on the surface of a moving body.



(a) 1 fresh-cell layer

(b) 2 fresh-cell layers

(c) 3 fresh-cell layers

(d) 1 fresh-cell layer

(e) 2 fresh-cell layers

(f) 3 fresh-cell layers

(g) 5 fresh-cell layers

Fig. 5. Velocity vectors around a forward-moving (rightward-moving) cylinder. Shaded areas represent fresh-cell layers generated by the rightward-motion of the cylinder. (a), (b), and (c) are solutions obtained with the fresh-cell treatment based on a bilinear interpolation scheme for 1, 2, and 3 fresh-cell layers, respectively. (d), (e), (f), and (g) are solutions obtained with the present backward time-integration treatment of 1, 2, 3, and 5 fresh-cell layers, respectively.

2.3. Backward time-integration method for fresh cells

For simulations of flow over a moving body using an immersed boundary method, it is necessary to assign appropriate flow field to fresh cells, which were previously inside a moving body but become fluid cells in the current time-step due to a body motion. Immersed boundary methods which are based on an explicit or a semi-implicit time-integration scheme usually allow a small CFL number and therefore, produce a small number of fresh-cell layers behind a moving body. Velocity vectors on fresh cells are usually interpolated from velocity vectors at neighbor nodes. Interpolation-based treatments of fresh cells have been reported to be successful in many applications [1,18,16,20] where the number of fresh-cell layers is relatively small. However, it is found that, in the present study, an interpolation-based treatment of many layers of fresh cells, which are generated by the use of a high CFL number, can cause numerical instability.

Fig. 5 shows velocity vectors in flow over a forward-moving (rightward-moving) circular cylinder obtained using a method based on a bilinear interpolation technique (Fig. 5(a)–(c)). Non-dimensional time-step sizes in the range of 0.02–0.1 are

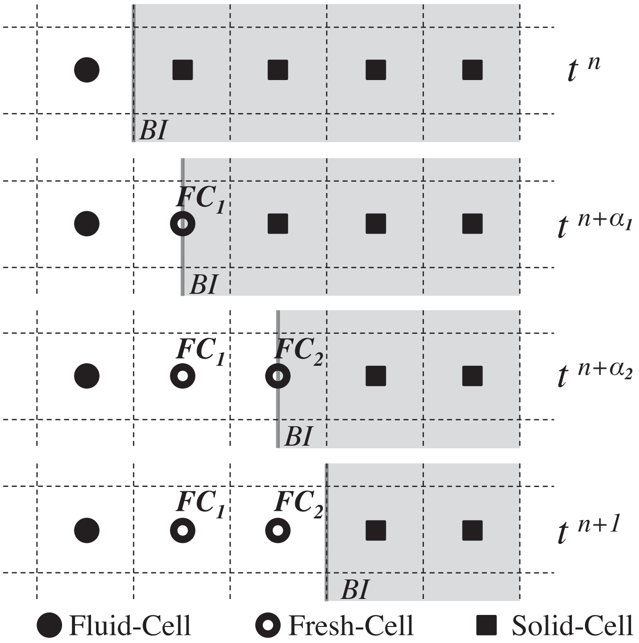


Fig. 6. Illustration of a backward time-integration scheme for assigning velocity vectors to fresh cells. FC_1 and FC_2 represent fresh cells generated at time $t^{n+\alpha_1}$ and $t^{n+\alpha_2}$, respectively. *BI* denotes the boundary intercept.

considered to investigate the adequacy of the interpolation treatment of fresh cells in multiple layers. The bilinear interpolation scheme utilizes velocity vectors available at neighboring fluid cells and the boundary intercept, and is found to reasonably well update one or two layers of fresh cells as shown in Fig. 5(a) and (b).

As found by Lee et al. [1] and Seo and Mittal [20], the magnitude of spurious force oscillations is reduced as increasing the time step size for given grid spacings. The present fully-implicit immersed boundary method allows the use of a large ratio of the time-step size to the grid spacing or the CFL number, thereby providing a higher capability for control of the production of spurious force oscillations. However, as shown in Fig. 5(c), it is found that the flow field becomes unstable when the interpolation-based method is employed for the case with a high CFL number, where three or more layers of fresh cells are generated.

In order to overcome the difficulty, a novel fresh-cell treatment scheme which is based on a backward time integration of the discretized momentum equations on fresh cells, is developed. Fig. 6 illustrates the present method in a one-dimensional configuration. Due to a rightward movement of an immersed boundary from t^n to t^{n+1}, two fresh cells (FC_1 and FC_2) are generated. At $t^{n+\alpha_1}$ and $t^{n+\alpha_2}$ where $0 \leqslant \alpha_1, \alpha_2 \leqslant 1$, the immersed boundary crosses the nodes of fresh cells FC_1 and FC_2, respectively. Therefore, at $t^{n+\alpha_1}$ and $t^{n+\alpha_2}$, velocity vectors on the surface of the immersed boundary are the same with velocity vectors at FC_1 and FC_2, respectively. In general, the fluid velocity on a fresh cell can be defined as $u_{i,FC}^{n+\alpha_k} = u_{i,IB}^{n+\alpha_k}$, where k is the index for the fresh cell, $u_{i,FC}^{n+\alpha_k}$ is the velocity on the k-th fresh cell, and $u_{i,IB}^{n+\alpha_k}$ is the velocity on the immersed boundary at $t^{n+\alpha_k}$.

Assuming a linear velocity profile on the surface of an immersed boundary, the pressure gradient at time step $t^{n+\alpha_k}$ can be calculated as follows:

$$\frac{\delta p^{n+\alpha_k}}{\delta x_i} = -\frac{du_{i,IB}^{n+\alpha_k}}{dt}. \tag{19}$$

Only for fresh cells, the discretized momentum equations are integrated using a backward time-integration scheme and velocity vectors and pressure gradients determined on fresh cells at time step $t^{n+\alpha_k}$, to assign velocity vectors to fresh cells at t^{n+1}:

$$\frac{u_i^{n+1} - u_{i,IB}^{n+\alpha_k}}{\Delta t^{\alpha_k}} + N_i^{n+1} = -\frac{\delta p^{n+\alpha_k}}{\delta x_i} + \frac{1}{Re} D_i^{n+1} \quad \text{for fresh cells only}, \tag{20}$$

where $\Delta t^{\alpha_k} = t^{n+1} - t^{n+\alpha_k}$.

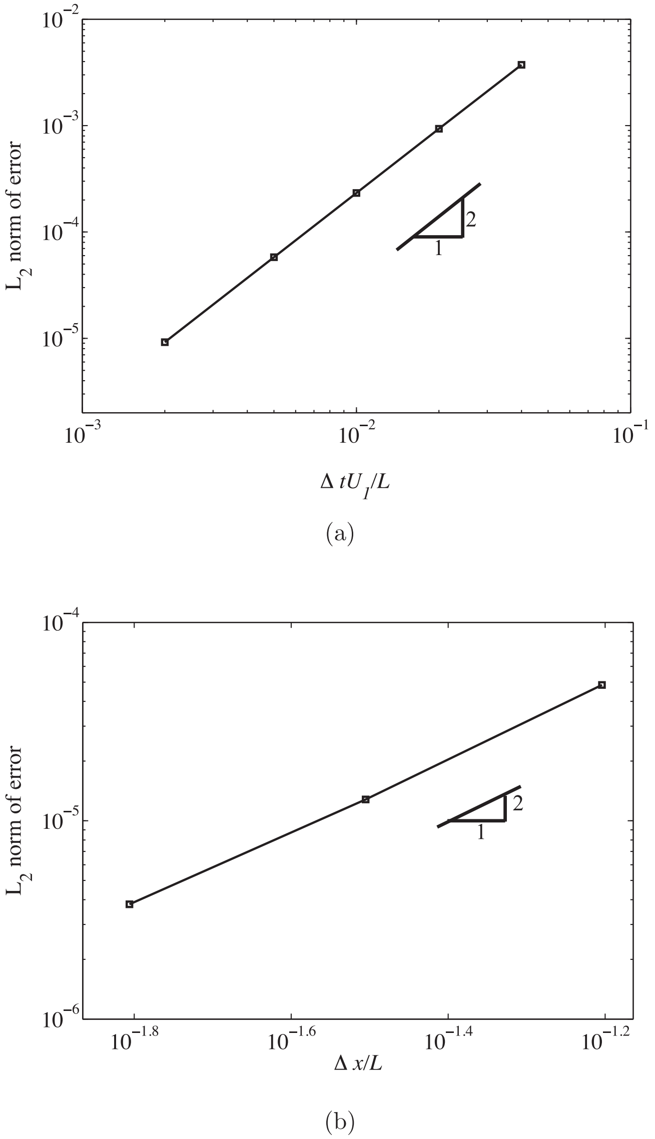


Fig. 7. L_2 norm errors produced by the present method in simulations of decaying vortices as a function of (a) the time-step size and (b) the grid spacing.

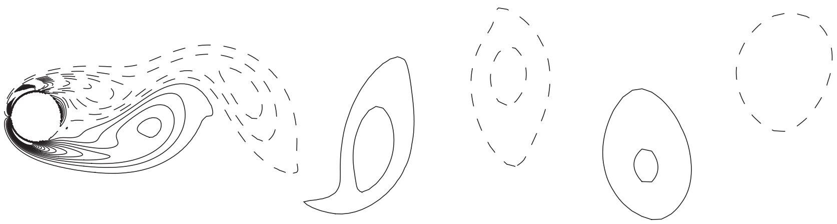


Fig. 8. Contour plot of the instantaneous spanwise vorticity over a stationary circular cylinder at $Re_D = 100$. 30 contour levels in the range of $-10 \sim 10\ U_\infty/d$ are shown.

The present backward time-integration method is found to be capable of assigning velocity vectors to more than three layers of fresh cells without numerical instability as shown in Fig. 5(d)-(g).

Table 1
Drag and lift coefficients and the Strouhal number for flow over a circular cylinder at Reynolds numbers of 100 and 160.

Re	100			160		
	C_L	C_D	St	C_L	C_D	St
Present	0.332	1.36	0.165	0.568	1.39	0.188
You et al. [24]	0.330	1.36	0.165	0.560	1.38	0.187

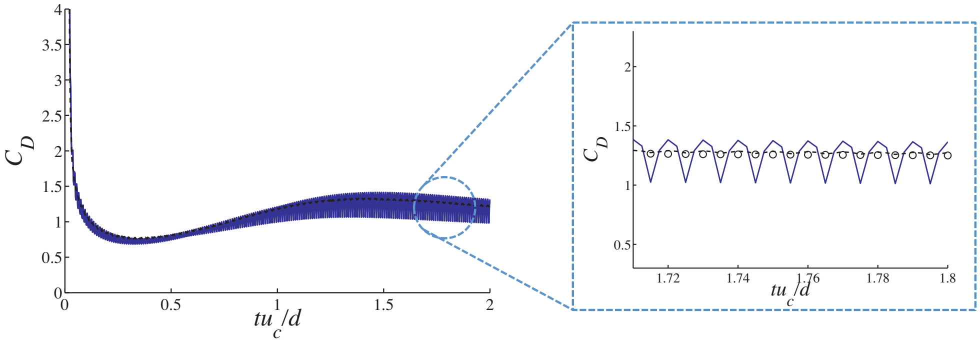


Fig. 9. Variation of the drag coefficient for a circular cylinder as a function of time. – – –, ghost-cell method with mass sources/sinks; —, ghost-cell method without mass sources/sinks; °, Koumoutsacos and Leonard [26].

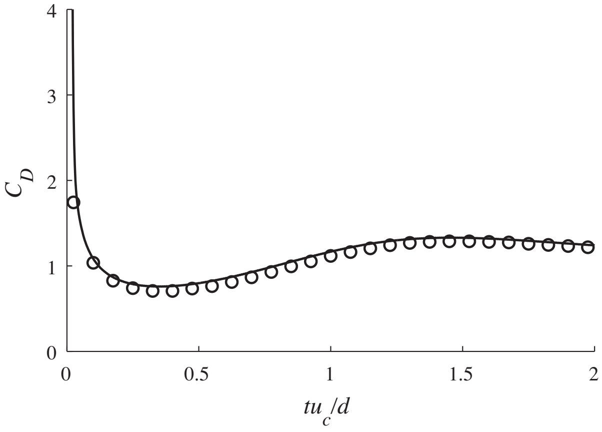


Fig. 10. Drag coefficients for a suddenly moving circular cylinder as a function of time at $Re = 550$ predicted by the present method on a 2188×418 mesh with the maximum CFL number of 4 (—) and by Koumoutsacos and Leonard [26] (°).

3. Verification and validation

3.1. Decaying vortices

To verify the spatial and temporal accuracy of the present numerical method, two-dimensional decaying vortices is simulated. The flow field consists of an array of vortices and the analytical solution for those vortices is as follows:

$$\begin{aligned} u &= -\cos(\pi x)\sin(\pi y)e^{-2\pi^2 t/Re}, \\ v &= \sin(\pi x)\cos(\pi y)e^{-2\pi^2 t/Re}, \\ p &= -\frac{1}{4}[\cos(2\pi x) + \sin(2\pi y)]e^{-4\pi^2 t/Re}. \end{aligned} \tag{21}$$

The Reynolds number of this flow is 100 based on the initial maximum velocity U_1 and the size of a vortex L. The computational domain size is $2L \times 2L$. Periodic boundary conditions are imposed on both x and y directions, while the initial condition is set by the analytical solution at $t = 0L/U_1$. The time-step size is fixed at $\Delta t = 0.001L/U_1$ for spatial convergence tests

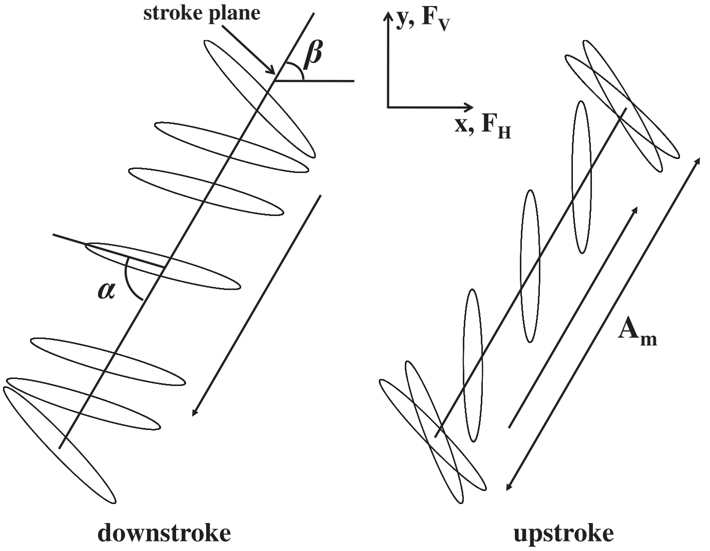


Fig. 11. Flapping kinematics of an elliptic wing. F_V and F_H are the vertical and horizontal forces, respectively. α is the angle between the chord axis and the stroke axis, β is the angle between the stroke axis and the horizontal axis, and A_m is the stroke amplitude.

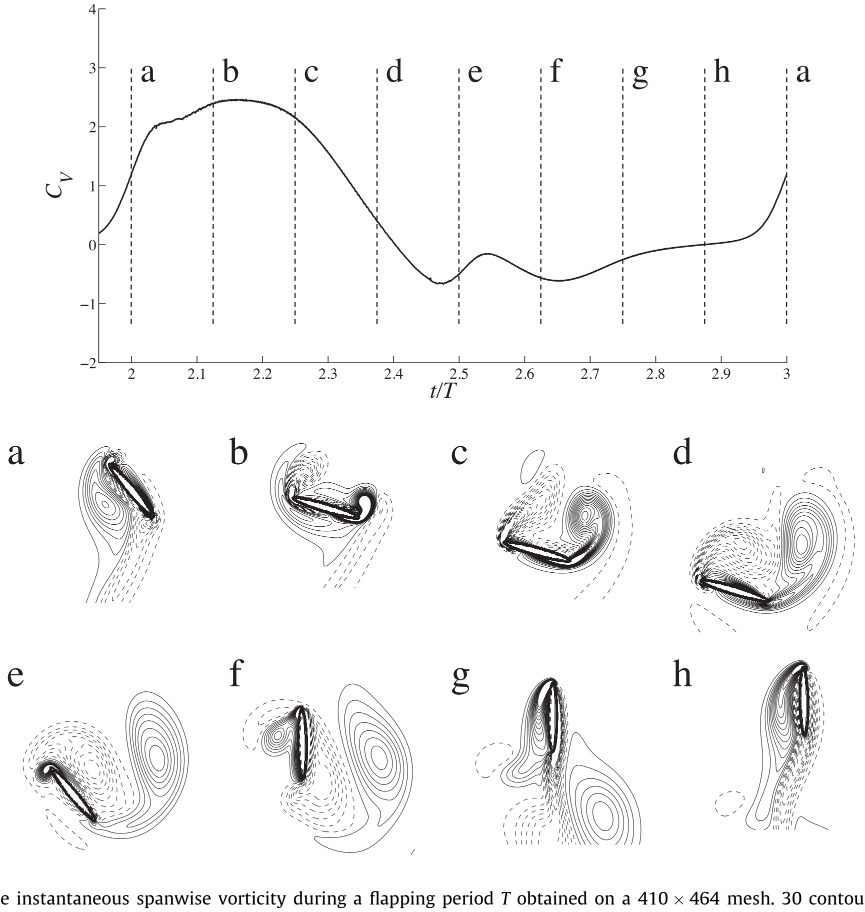


Fig. 12. Contours of the instantaneous spanwise vorticity during a flapping period T obtained on a 410 × 464 mesh. 30 contour levels in the range of $-10 \sim 10\ u_m/c$ are shown.

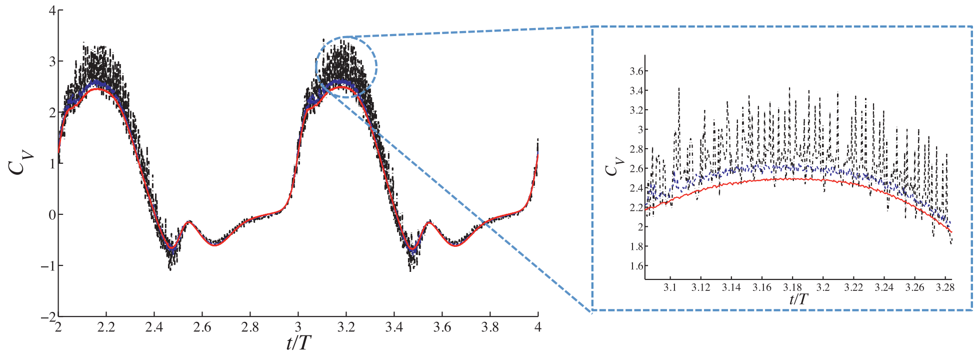


Fig. 13. Comparison of the vertical force coefficient for a flapping wing obtained on different meshes. ⋯, 110 × 128 mesh with $\Delta x_{min} = 0.04c$; − − −, 220 × 256 mesh with $\Delta x_{min} = 0.02c$; and —, 410 × 464 mesh with $\Delta x_{min} = 0.01c$. The time-step size is fixed at $\Delta t = 0.004\ c/u_m$.

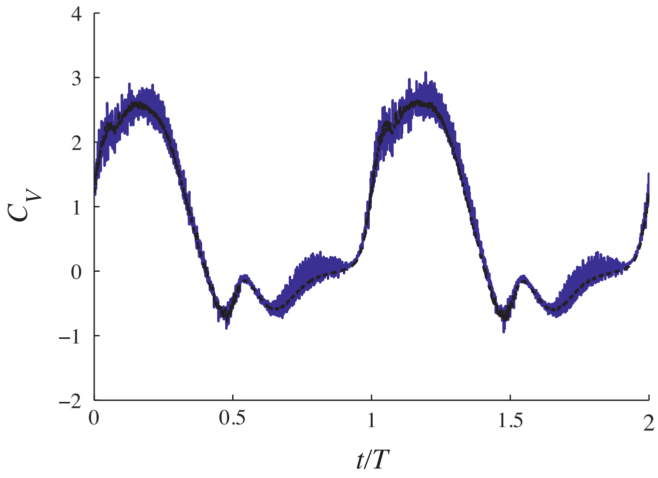


Fig. 14. Vertical force coefficient for a flapping wing obtained by the present method with (black-colored —) and without mass sources/sinks (blue-colored —). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

while the grid spacings in both x and y directions are fixed at $\Delta x = \Delta y = 0.008L$ for a temporal convergence study. The CFL number for a combination of the given time-step sizes and grid-spacings varies in the range of 0.064–5.1. The CFL number is defined as $\mid u\Delta t/\Delta x \mid + \mid v\Delta t/\Delta y \mid$ and $\mid u\Delta t/\Delta x \mid + \mid v\Delta t/\Delta y \mid + \mid w\Delta t/\Delta z \mid$ for two- and three-dimensional cases, respectively. As shown in Fig. 7, the present fully-implicit method is second-order accurate in both time (Fig. 7(a)) and space (Fig. 7(b)).

3.2. Vortex shedding behind a circular cylinder

The accuracy and effectiveness of the present fully-implicit ghost-cell immersed-boundary method for simulations of flow over a stationary immersed body, is evaluated in the simulation of flow over a stationary circular cylinder. The present method shows that the drag and lift forces are not contaminated by spurious force oscillations in stationary body problems including the present case.

The Reynolds numbers are 100 and 160 based on the inflow freestream velocity U_∞ and the cylinder diameter d. The computational domain size is $40d \times 40d$ and is discretized on a grid consisting of 450 × 450 points. The grid lines are clustered around the cylinder with the minimum grid spacing of $0.01d$. The time-step size is fixed at $0.005d/U_\infty$.

The gross feature of flow over a circular cylinder is well represented in Fig. 8, which shows contour lines of the instantaneous spanwise vorticity. As summarized in Table 1, the drag and lift coefficients and the Strouhal number predicted by the present method are found to agree well with results obtained using a finite-difference method employing a curvilinear body-fitted mesh [24]. In the present method, the pressure force and the skin friction are computed using the pressure and velocity gradients which are obtained using the ghost-cell interpolation method detailed in Section 2.2.

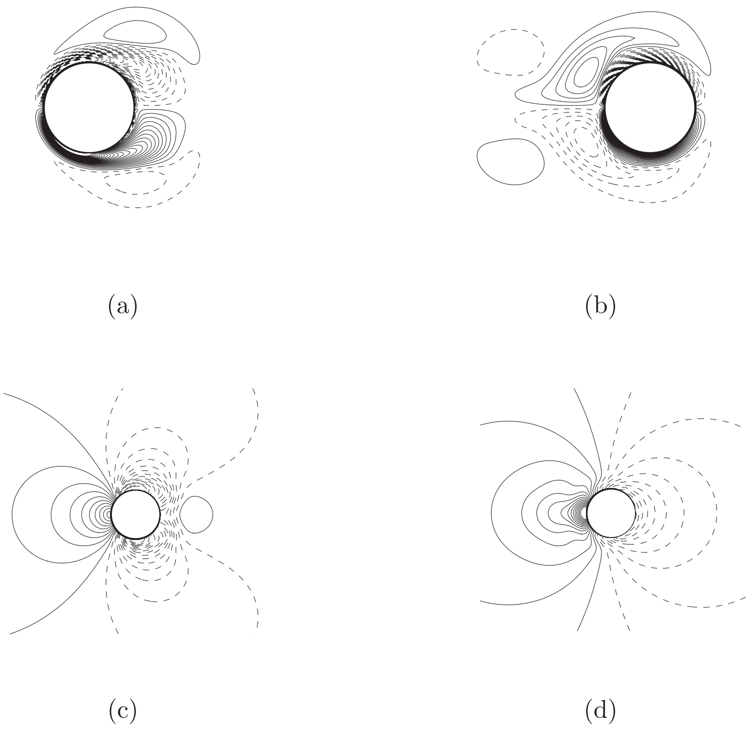


Fig. 15. Contours of the instantaneous spanwise vorticity ((a) and (b)) and pressure ((c) and (d)) over an oscillating cylinder. (a) and (c) are at the phase angle of 0° and (b) and (d) are at the phase angle of 288°. 30 contour levels in the range of $-20 \sim 20\ u_m/d$ for the instantaneous spanwise vorticity and 50 contour levels in the range of $-1 \sim 1\ \rho u_m^2$ for the pressure are shown.

3.3. Flow over a suddenly-moving circular cylinder

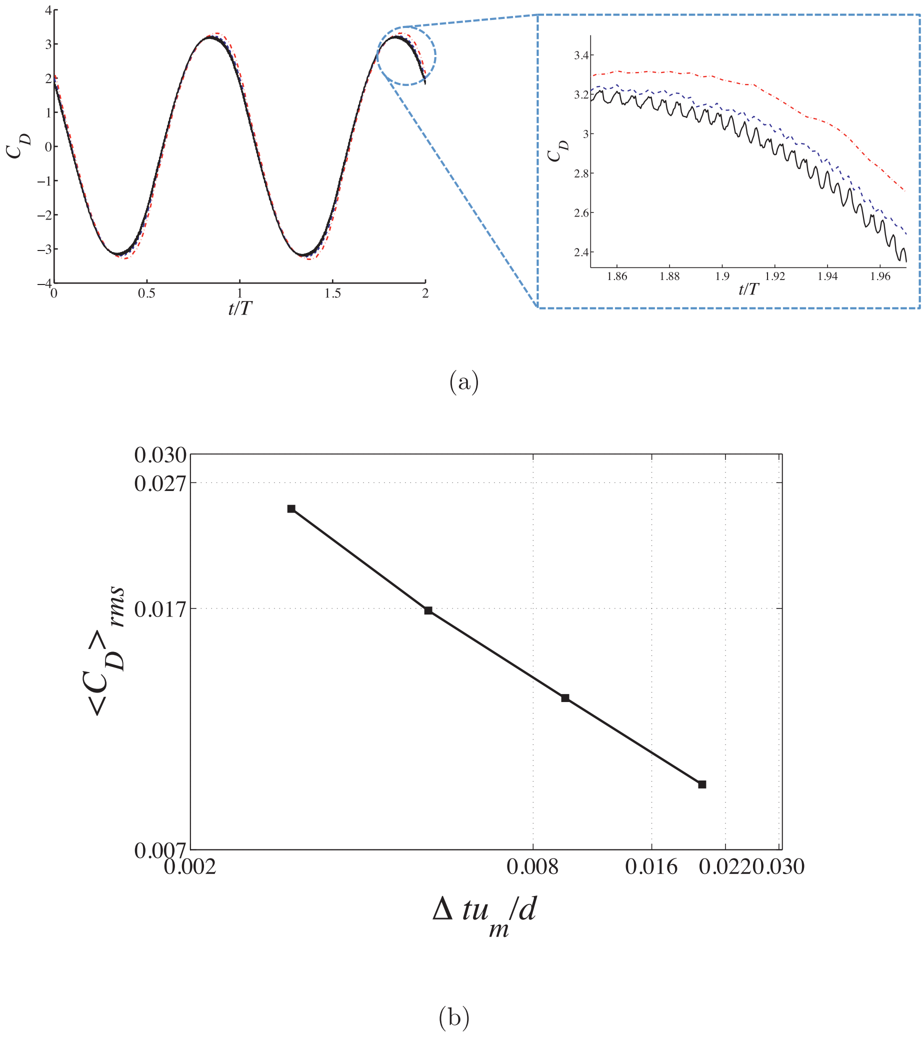
Flow over a suddenly moving circular cylinder at the Reynolds number of 550 is simulated to demonstrate the effectiveness of the present fully-implicit ghost-cell method with mass source and sink for controlling the generation of spurious force oscillations at a relatively high Reynolds number. The same configuration was considered experimentally and computationally by Bouard and Coutanceau [25] and Koumoutsacos and Leonard [26], respectively.

The computational domain size is $50d$, where d is the diameter of the cylinder, in both streamwise and vertical directions. To investigate the effect of the mass source and sink treatment on the reduction of spurious force oscillations, the same flow is simulated with and without the mass sources/sinks on a grid consisting of 468 × 320 points with the minimum grid spacing of $0.01d$, in both streamwise and vertical directions. Fig. 9 shows temporal histories of the drag coefficient predicted by the present method with (—) and without (—) mass sources and sinks. The magnitude of spurious force oscillations is quantified by the root-mean-square of the difference between the corrected drag coefficient, which obtained by the method of Lee et al. [1], and the non-modified actual drag coefficient ($\langle C_D \rangle_{rms}$). On the same mesh and with the same time-step size, the present method with mass sources and sinks is found to produce $\langle C_D \rangle_{rms}$ of 6.44×10^{-3}, while the same method generates $\langle C_D \rangle_{rms}$ of 1.02×10^{-1} without mass sources and sinks. Furthermore, as shown in Fig. 9, the drag coefficient, which is predicted with the mass source and sink treatment, is found to agree well with the data of Koumoutsacos and Leonard [26].

The magnitude of the spurious force oscillations is found to be further reduced by refining the mesh while increasing the maximum CFL number. On a 2188 × 418 mesh with the minimum grid spacing of $0.0015d \times 0.005d$ in the streamwise and vertical directions, respectively, and with the maximum CFL number of about 4, $\langle C_D \rangle_{rms}$ is found to be reduced to 7.67×10^{-4}. Furthermore, as shown in Fig. 10, the temporal variation of the drag coefficient for the moving circular cylinder obtained using the present method at a high CFL number agrees well with the result predicted by Koumoutsacos et al. [26] using a high-resolution vortex method.

3.4. Flow over a flapping wing

The capability of the present numerical method for simulations of flow over an arbitrary moving body is also evaluated in simulations of flow over a two-dimensional flapping wing. The same flow configuration was also considered by many other researchers [1,27,28]. The wing cross-section corresponds to an ellipse with a thickness-to-chord ratio of 0.125. The flapping kinematics of the wing is illustrated in Fig. 11.



(a)

(b)

Fig. 16. (a) Comparison of temporal variations of the drag force coefficient obtained with time-step sizes of $0.02d/u_m$ (·–); $0.01d/u_m$ (– – –); and $0.0025d/u_m$ (—), and (b) root-mean-square fluctuations of the drag coefficient as a function of time-step size. The grid size is fixed at 310 × 160 with the minimum spacing of $0.02d$.

The coordinates of the wing center (x_c, y_c) are prescribed as a function of time:

$$x_c(t) = 0.5A_m \cos(2\pi f t)\cos\beta,$$
$$y_c(t) = 0.5A_m \cos(2\pi f t)\sin\beta, \tag{22}$$

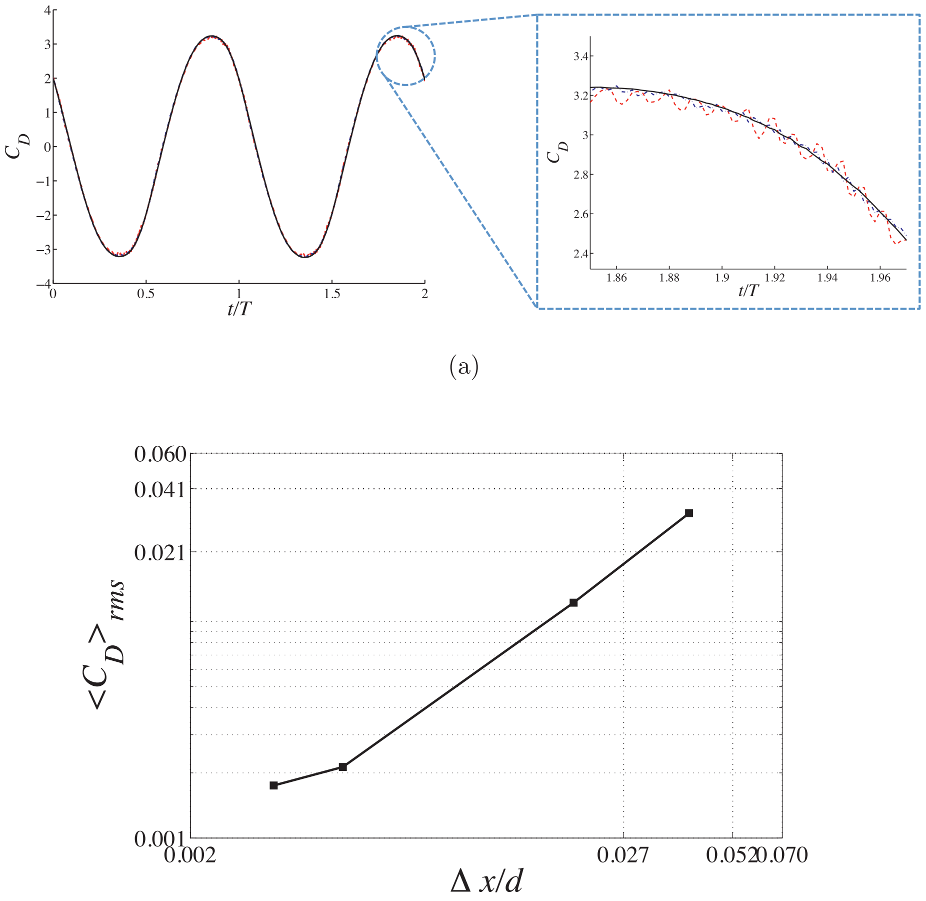
where A_m is the stroke amplitude, β is the angle between the stroke plane and the horizontal axis, and α is the angle between the chord axis and the stroke plane.

The angle of wing rotation is also a function of time and is prescribed as follows:

$$\alpha(t) = \begin{cases} \frac{5}{24}\pi \tanh(-2t) + \frac{5}{8}\pi & \text{for } -\frac{T}{4} \leqslant t < \frac{T}{4}, \\ \frac{5}{24}\pi \tanh[2(t-\frac{T}{2})] + \frac{5}{8}\pi & \text{otherwise.} \end{cases} \tag{23}$$

The Reynolds number is 150 based on the maximum velocity u_m of the wing center and the chord length c. The stroke amplitude A_m is $2.5c$ and the angle of the stoke plane β is 60°. For a computational domain of $20c \times 20c$, three different grid sizes of $110 \times 128, 220 \times 256$, and 410×464 are employed. Grid lines are clustered around the wing trajectory and the smallest grid spacings are $0.04c, 0.02c$, and $0.01c$ for the three different grids, respectively. The computational time-step is fixed at $\Delta t = 0.004c/u_m$.

Fig. 12 shows contours of the instantaneous spanwise vorticity during a flapping cycle and corresponding variation of the vertical force coefficient as a function of time. The force coefficient predicted by the present method is found to agree well with the result obtained by Lee et al. [1] on a 780 × 864 mesh. The present fully-implicit ghost-cell method with mass



(a)

(b)

Fig. 17. (a) Comparison of temporal variations of the drag force coefficient obtained on grids of 155 × 80 with the minimum spacings of 0.04d (– – –); 310 × 160 with the minimum spacing of 0.02d (·–); and 1860 × 960 with the minimum spacing of 0.0033d (—), and (b) root-mean-square fluctuations of the drag coefficient as a function of grid spacing. The time-step size is fixed at 0.005d/u_m.

sources and sinks is found to effectively reduce the magnitude of spurious force oscillations by decreasing the grid spacing while fixing the time-step size (i.e., increasing the CFL number) as shown in Fig. 13.

Fig. 14 shows a comparison of temporal variations of the vertical force coefficient obtained using the present numerical method with and without mass sources and sinks on a 220 × 256 mesh. Similarly to the result in the simulation of flow over a suddenly-moving circular cylinder, the magnitude of spurious force oscillations in the vertical force coefficient is found to be significantly reduced with the use of the mass source and sink method. Quantitatively, the root-mean-square of the spurious force oscillations in the vertical force coefficient is found to be 2.33×10^{-2} and 1.08×10^{-1}, for simulations with and without mass sources and sinks, respectively.

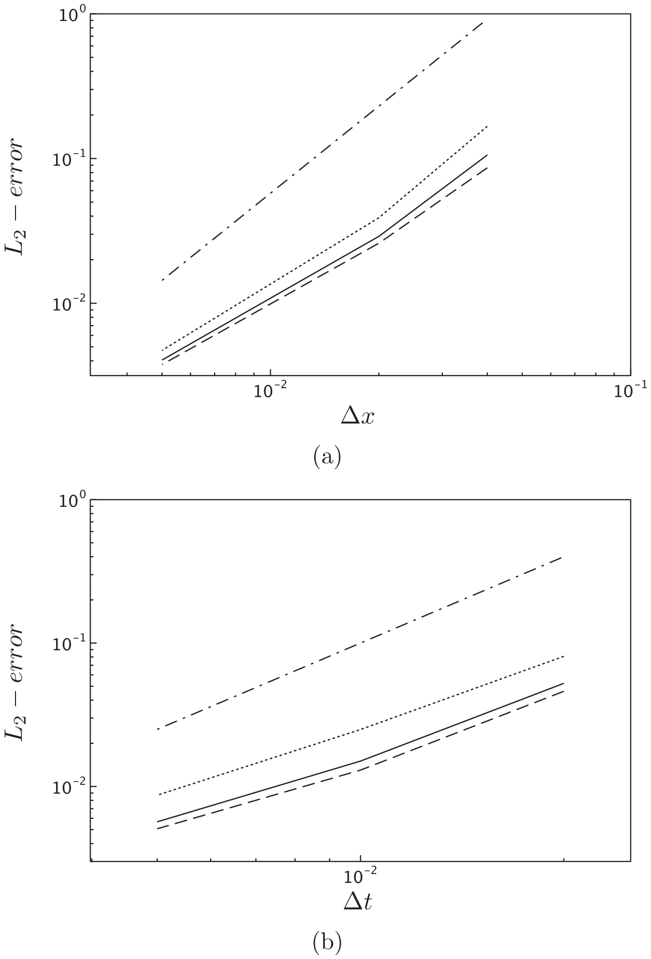
3.5. Flow over an oscillating circular cylinder

The capability of the present method for controlling spurious force oscillations by adjusting the time-step size and the grid spacing is demonstrated in simulations of flow over a periodically oscillating circular cylinder. In the same flow configuration, Dütsch et al. [29] conducted a simulation using a body-fitted-grid method.

The Reynolds number is 100, based on the cylinder diameter d and the maximum velocity of the oscillation u_m. The computational domain size is $50d \times 30d$. A periodic motion of a circular cylinder is prescribed with the Keulegan–Carpenter number of 5 as follows:

$$x_c(t) = A_m \sin(2\pi f t), \tag{24}$$

where x_c is the coordinate of the cylinder center in the streamwise direction. A_m and f are 0.7958d and 0.2u_m/d, respectively.



(a)

(b)

Fig. 18. L_2 norm errors produced by the present method in simulations of flow over an oscillating circular cylinder as a function of (a) the grid spacing and (b) the time-step size. —, streamwise velocity; — — —, vertical velocity; . . ., pressure; ·—, second-order slope.

Fig. 15 shows the instantaneous vorticity and pressure contours around an oscillating cylinder at phase angles of 0° and 288°, which also qualitatively agree well with those reported by Dütsch et al. [29].

To investigate the variation of the magnitude of spurious force oscillations as a function of time-step size, four different time-step sizes of $0.0025d/u_m, 0.005d/u_m, 0.01d/u_m$, and $0.02d/u_m$ are employed, while the grid size is fixed at 310 × 160. As shown in Fig. 16(a), spurious oscillations in the drag coefficient are found to be effectively reduced by increasing the time-step size, which is consistent with the findings of Lee et al. [1] and Seo and Mittal [20]. On the mesh resolution considered in Fig. 16(a) (310 × 160), marginal differences in the phase and amplitude of the temporal variation of the drag coefficient predicted using different time-step sizes. The magnitude of spurious force oscillations is found to be rapidly reduced by increasing the time-step size as shown in Fig. 16(b).

To investigate the variation of the magnitude of spurious force oscillations as a function of grid-spacing, four different grids of $155 \times 80, 310 \times 160, 1240 \times 640$, and 1860×960 with the minimum grid spacings of $0.04d, 0.02d, 0.005d$, and $0.0033d$, respectively, are employed, while the time-step size is fixed at $0.01d/u_m$. As shown in Fig. 17(a), unlike in the case with a varying time-step size on a fixed grid, differences in the phase and amplitude of the temporal variation of the drag coefficient are nearly negligible among different simulations with different CFL numbers (i.e., on varying grid resolution with a fixed time-step size). The simulation on a 1860 × 960 mesh with the time-step size of $0.01d/u_m$ results in the maximum CFL number of about 5 and generates maximum 4 fresh-cell layers. As shown in Fig. 17(b), rapid reduction of spurious force oscillations is achieved by decreasing the grid spacing even with increasing CFL number.

The spatial and temporal orders of accuracy of the present method for moving body simulations have been assessed as similarly done by Seo and Mittal [20]. In the present study, the solution obtained on a 1860 × 960 mesh with the time-step size of $10^{-3}d/u_m$ serves as a reference solution. The spatial convergence is measured on $155 \times 80, 310 \times 160$, and 1240×640 meshes with the fixed time-step size of $10^{-3}d/u_m$, while the temporal order of accuracy is measured on the 1860 × 960 mesh using different time-step sizes of 0.02, 0.01, and 0.005 d/u_m. As shown in Fig. 18(a) and (b), in the interval of relatively large

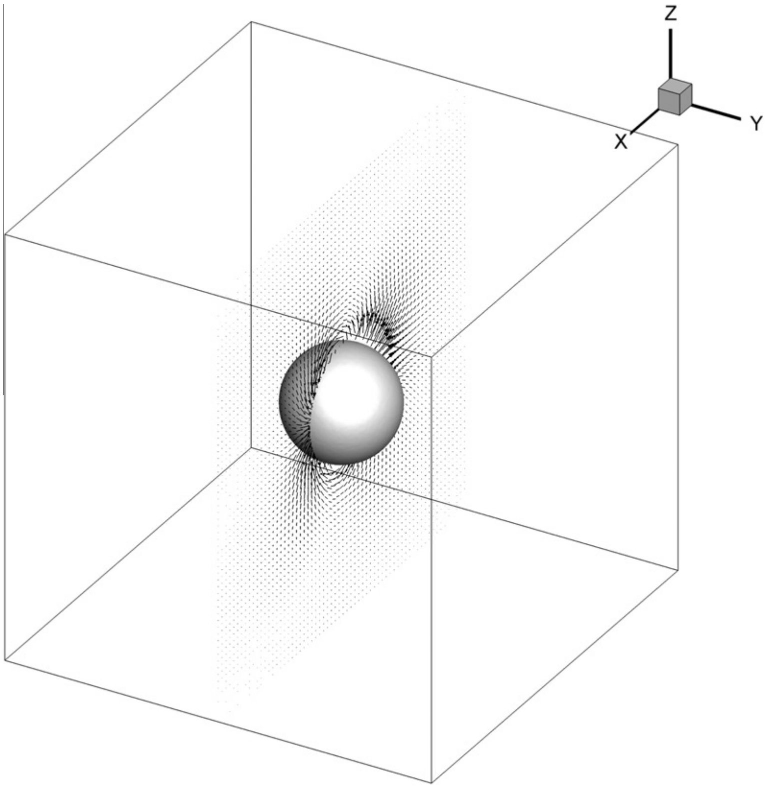


Fig. 19. Instantaneous velocity vectors over an oscillating sphere.

grid spacings or time-step sizes, nearly second-order convergence for velocity components and pressure is observed in both space and time, while the convergence orders are a little deteriorated in the interval of smaller grid spacings and time-step sizes. This behavior is considered to be reasonable considering that the reference solution is not an exact solution.

3.6. Flow over an oscillating sphere

The capability of the present method for simulations of flow over a three-dimensional moving body is assessed in simulations of flow over an oscillating sphere, which is shown in Fig. 19. The Reynolds number is 78.54 based on the sphere diameter d and the maximum speed of the sphere u_m. The Strouhal number of the oscillating sphere is 1.2732. A periodic motion of the sphere is prescribed as follows:

$$x_c(t) = A_m \sin(2\pi f t), \tag{25}$$

where x_c is the coordinate of the sphere center in the streamwise direction. A_m and f are 0.125d and $1.2732u_m/d$, respectively.

For a computational domain of $4d \times 4d$, three different grid sizes of $64 \times 64 \times 64, 110 \times 112 \times 112$, and $124 \times 142 \times 142$ with the minimum grid spacings of $0.064d, 0.01d$, and $0.0075d$, respectively, are employed at a fixed time-step size of $2.094 \times 10^{-2} d/u_m$.

The present result agrees well with the result of Seo and Mittal [20] both qualitatively and quantitatively. Fig. 20 shows contours of the instantaneous pressure around an oscillating sphere at the middle span at three different phase angles. The temporal variation of the pressure drag coefficient predicted by the present method also matches well with the result of Seo and Mittal as shown in Fig. 21.

To evaluate the effectiveness of the mass source and sink treatment, the present immersed boundary method is employed with and without mass sources and sinks on a $64 \times 64 \times 64$ mesh. Like in two-dimensional cases, it is found that the mass source and sink treatment is also effective in three-dimensional simulations in suppressing the generation of spurious force oscillations as shown in Fig. 22.

The CFL number effect on the simulation result is investigated by varying grid resolution while keeping the time-step size. On the $124 \times 142 \times 142$ mesh, the maximum CFL number is about 3.5 and maximum 3 fresh-cell layers are generated. As shown in Fig. 23, the present fully-implicit immersed boundary method coupled with a mass source and sink method and the novel fresh-cell treatment is found to predict the pressure drag coefficient, which is not contaminated by spurious

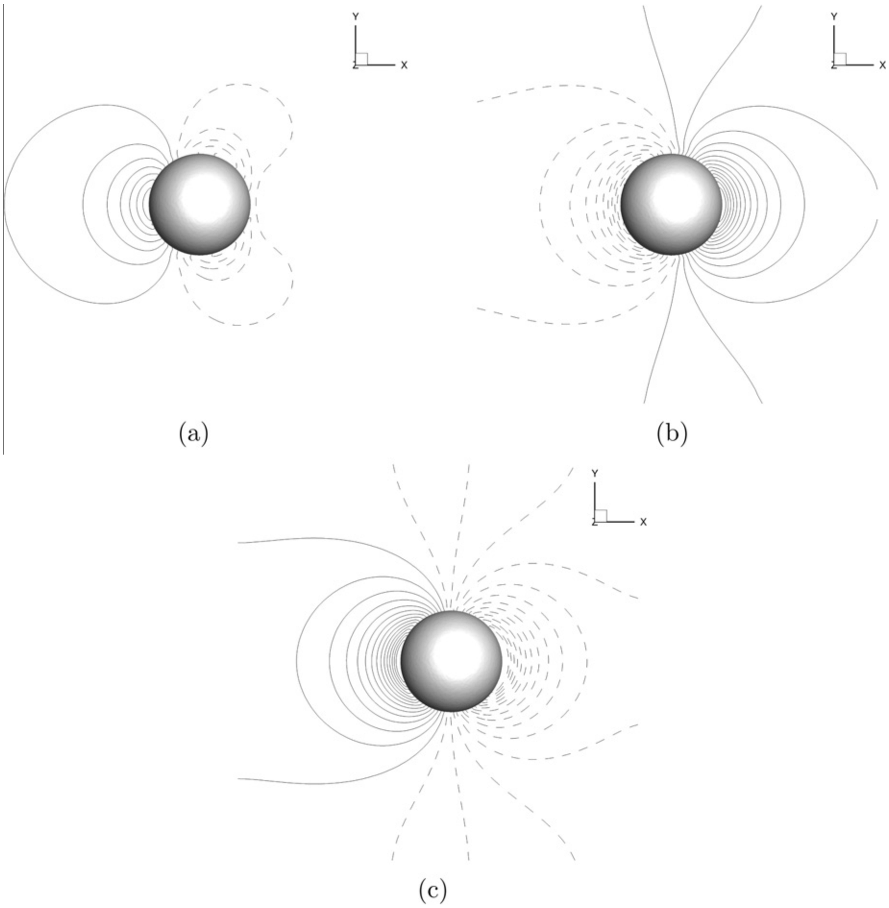


Fig. 20. Contours of the instantaneous pressure over an oscillating sphere. (a), (b), and (c) are at phase angles of 0°, 144°, and 288°, respectively. 30 contour levels in the range of $-2 \sim 2\ \rho u_m^2$ are shown.

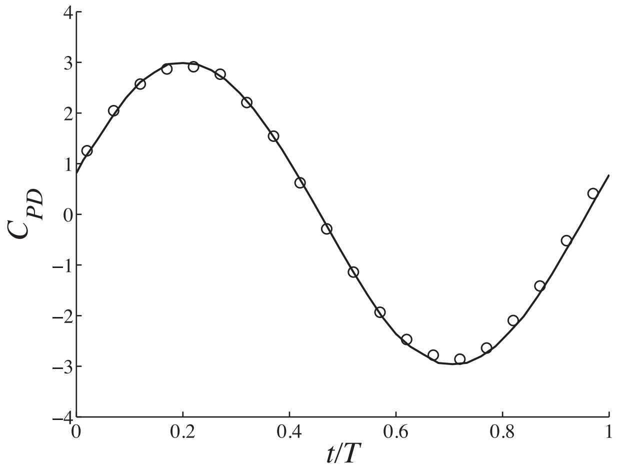


Fig. 21. The pressure drag coefficients for an oscillating sphere at $Re = 78.54$ obtained by the present method on a $124 \times 142 \times 142$ (—) and by Seo and Mittal [20] (◦).

force oscillations, on all three different grids considered. Furthermore, differences among the predicted solutions on different CFL numbers are found to be nearly negligible in terns of phase and amplitude.

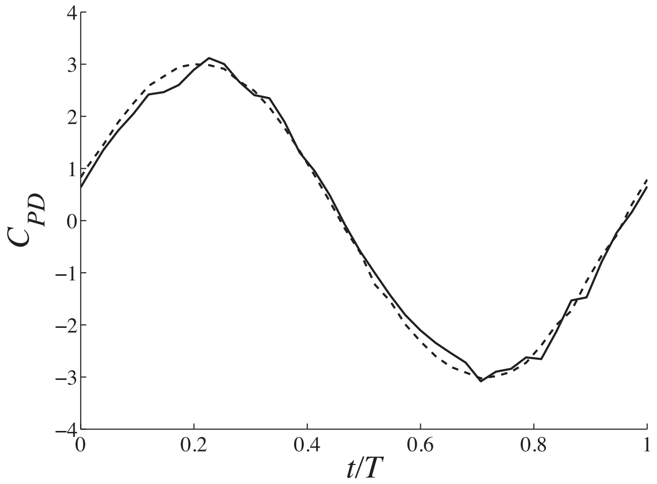


Fig. 22. Temporal variations of the pressure drag coefficients obtained on a grid of 64 × 64 × 64 using the present method with mass sources/sinks – – – and without mass sources/sinks —.

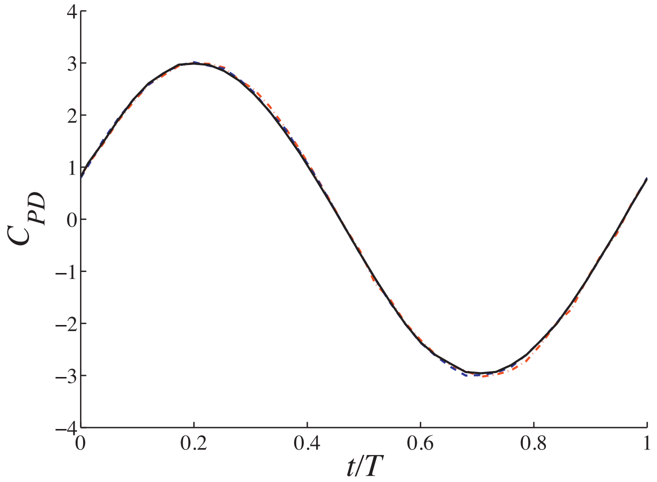


Fig. 23. Temporal variations of the pressure drag coefficients obtained on grids of 64 × 64 × 64 with the minimum spacings of $0.064d$ (·–); 110 × 112 × 112 with the minimum spacing of $0.01d$ (– – –); and 124 × 142 × 142 with the minimum spacing of $0.0075d$ (—).

4. Concluding remarks

A novel computational methodology for simulations of flow over moving bodies on a Cartesian grid has been developed. The new method is based on a fully-implicit time integration of the discretized incompressible Navier–Stokes equations coupled with a ghost-cell immersed boundary method [18] and a mass source and sink method [11]. The present immersed boundary method allows to realize sharp interfaces between the fluid and bodies, and most importantly, provides a capability for controlling the production of spurious force oscillations during moving-body simulations.

In the present immersed boundary method, the control of spurious force oscillations is enabled by two mechanisms. Firstly, the use of a mass source and sink algorithm [11] allows to better conserve the fluid mass in the vicinity of immersed boundaries without any significant complications, thereby reducing the source of spurious force oscillations. Secondly, fully-implicit time integration of the discretized governing equations allows to use a broad range of ratios of the time-step size to the grid spacing. This feature offers an enhanced capability for controlling spurious force oscillations by adjusting the time-step size on given grid spacings or adjusting grid spacings at a fixed time-step size, considering that the magnitude of spurious force oscillations on immersed boundaries is proportional to the grid spacing (or the square of the grid spacing) and is inversely proportional to the time-step size.

Although a fully-implicit time-integration method allows to use a high CFL number, it requires a special treatment for multiple layers of fresh cells which are generated due to a motion of immersed body under a high CFL number condition. To effectively and accurately deal with multiple layers of fresh cells, a novel fresh-cell treatment algorithm based on a backward time-integration scheme has been developed. The present backward time-integration method is found to be capable of assigning velocity vectors to many layers of fresh cells without numerical instability.

As demonstrated in a number of test cases, the present fully-implicit ghost-cell immersed boundary method coupled with a mass source/sink algorithm is found to effectively suppress spurious force oscillations during simulations of flow over moving bodies.

Acknowledgments

The authors acknowledge the support of the Office of Naval Research under Grant No. N000141110652, with Dr. Ki-Han Kim as the program manager.

References

[1] J. Lee, J. Kim, H. Choi, K-S. Yang, Sources of spurious force oscillations from an immersed boundary method for moving-body problems, Journal of Computational Physics 230 (2011) 2677–2695.
[2] C.S. Peskin, Flow patterns around heart valves: a numerical method, Journal of Computational Physics 10 (1972) 252–271.
[3] C.S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (1977) 220–252.
[4] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, Journal of Computational Physics 105 (1993) 354–366.
[5] C.S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 1–39.
[6] M. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, Journal of Computational Physics 160 (2000) 705–719.
[7] Z. Li, M. Lai, The immersed interface method for the Navier-Stokes equations with singular forces, Journal of Computational Physics 171 (2001) 822–842.
[8] K. Taira, T. Colonius, The immersed boundary method: a projection approach, Journal of Computational Physics 10 (2007) 252–271.
[9] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, Annual Research Briefs, Center for Turbulence Research, NASA Ames and Stanford University, 1997, pp. 317–327.
[10] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three dimensional complex flow simulations, Journal of Computational Physics 161 (2000) 35–60.
[11] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, Journal of Computational Physics 171 (2001) 132–150.
[12] W.-X. Huang, H.J. Sung, Improvement of mass source/sink for an immersed boundary method, International Journal for Numerical Methods in Fluids 53 (2007) 1659–1671.
[13] Y. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, Journal of Computational Physics 192 (2003) 593–623.
[14] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, Computers and Fluids 33 (2004) 375–404.
[15] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, Journal of Computational Physics 209 (2005) 448–476.
[16] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, Journal of Computational Physics 215 (2006) 12–40.
[17] J. Yang, F. Stern, A simple and efficient direct forcing immersed boundary framework for fluid–structure interactions, Journal of Computational Physics 231 (2012) 5029–5061.
[18] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, Journal of Computational Physics 227 (2008) 4825–4852.
[19] S. Kang, G. Iaccarino, P. Moin, Accurate immersed-boundary reconstructions for viscous flow simulations, AIAA Journal 47 (2009) 1750–1760.
[20] J.H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, Journal of Computational Physics 230 (2011) 7347–7363.
[21] R. Mittal, G. Iaccarino, Immersed boundary method, Annual Review of Fluid Mechanics 37 (2005) 239–261.
[22] D. You, R. Mittal, M. Wang, P. Moin, Computational methodology for large-eddy simulation of tip-clearance flows, AIAA Journal 42 (2004) 271–279.
[23] D. You, M. Wang, R. Mittal, A methodology for high performance computation of fully inhomogeneous turbulent flows, International Journal for Numerical Methods in Fluids 53 (2007) 947–968.
[24] D. You, H. Choi, M.-R. Choi, S.-H. Kang, Control of flow-induced noise behind a circular cylinder using splitter plates, AIAA Journal 36 (1998) 1961–1967.
[25] R. Bouard, M. Coutanceau, The early stage of development of the wake behind an impulsively started cylinder for $40 \leqslant Re \leqslant 10^4$, Journal of Fluid Mechanics 101 (1980) 583–607.
[26] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, Journal of Fluid Mechanics 296 (1995) 1–38.
[27] Z.J. Wang, Two dimensional mechanism for insect hovering, Physical Review Letters 85 (2000) 2216–2219.
[28] D. Kim, H. Choi, Two-dimensional mechanism of hovering flight by single flapping wing, Journal of Mechanical Science and Technology 21 (1) (2007) 2677–2695.
[29] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, Journal of Fluid Mechanics 360 (1998) 249–271.