Journal of Computational Physics **171,** 132–150 (2001)
doi:10.1006/jcph.2001.6778, available online at http://www.idealibrary.com on IDEAL®

An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries

Jungwoo Kim,[*,†] Dongjoo Kim,[*,†] and Haecheon Choi[*,†]

[*]*School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Korea; and* [†]*National CRI Center for Turbulence and Flow Control Research, Institute of Advanced Machinery and Design, Seoul National University, Seoul 151-742, Korea*
E-mail: choi@socrates.snu.ac.kr

Received July 17, 2000; revised February 14, 2001

A new immersed-boundary method for simulating flows over or inside complex geometries is developed by introducing a mass source/sink as well as a momentum forcing. The present method is based on a finite-volume approach on a staggered mesh together with a fractional-step method. Both momentum forcing and mass source are applied on the body surface or inside the body to satisfy the no-slip boundary condition on the immersed boundary and also to satisfy the continuity for the cell containing the immersed boundary. In the immersed-boundary method, the choice of an accurate interpolation scheme satisfying the no-slip condition on the immersed boundary is important because the grid lines generally do not coincide with the immersed boundary. Therefore, a stable second-order interpolation scheme for evaluating the momentum forcing on the body surface or inside the body is proposed. Three different flow problems (decaying vortices and flows over a cylinder and a sphere) are simulated using the immersed-boundary method proposed in this study and the results agree very well with previous numerical and experimental results, verifying the accuracy of the present method. © 2001 Academic Press

Key Words: immersed-boundary method; momentum forcing; mass source or sink; finite-volume method.

1. INTRODUCTION

The ability to handle complex geometries has been one of the main issues in computational fluid dynamics because most engineering problems have complex geometries. So far, two different approaches to simulating complex flow have been taken: the unstructured grid method and the immersed-boundary method. In the latter method, a body in the flow field is considered a kind of momentum forcing in the Navier–Stokes equations rather than a real body, and therefore flow over a complex geometry can be easily handled with orthogonal (Cartesian or cylindrical) grids which generally do not coincide with the body surface. The

0021-9991/01 $35.00

main advantages of the immersed-boundary method are memory and CPU savings and easy grid generation compared to the unstructured grid method. Even moving-boundary problems can be handled with the immersed-boundary method without regenerating grids in time, unlike the unstructured grid method.

An immersed-boundary method using momentum forcing was presented by Peskin [1], who simulated the blood flow in heart valves. Goldstein *et al.* [2] and Saiki and Biringen [3] employed feedback forcing to represent a solid body, but feedback forcing induced spurious oscillations and restricted the computational time step associated with numerical stability. For example, Goldstein *et al.* [2] used a very small time step equivalent to a CFL (Courant–Friedrichs–Lewy) number of $O(10^{-3}–10^{-2})$ when they simulated the start-up flow around a circular cylinder. Recently, Mohd-Yusof [4] suggested a different approach to evaluating the momentum forcing in a spectral method, and his method does not require a smaller computational time step, which is an important advantage of this method over previous methods.

Fadlun *et al.* [5] applied the approach of Mohd-Yusof [4] to a finite-difference method on a staggered grid and showed that the discrete-time forcing suggested by Mohd-Yusof [4] is more efficient than feedback forcing for three-dimensional flow problems. However, there is a difference between the approaches of Fadlun *et al.* [5] and Mohd-Yusof [4]. In Fadlun *et al.* [5], the velocity at the first grid point external to the body is obtained by *linearly* interpolating the velocity at the second grid point (which is obtained by directly solving the Navier–Stokes equations) and the velocity at the body surface, which conceptually corresponds to applying the momentum forcing inside the flow field. On the other hand, momentum forcing is applied only on the body surface or inside the body in Mohd-Yusof [4]. The interpolation direction (i.e., the direction to the second grid point) taken by Fadlun *et al.* [5] is either the streamwise or the transverse direction, but the choice of interpolation direction is arbitrary. When a bilinear (2-D) interpolation is used to avoid the arbitrariness, however, the use of the approximate factorization scheme combined with a tridiagonal matrix solver may require numerical iteration, which increases the CPU time, especially for three-dimensional flow.

Recently, Ye *et al.* [6] proposed a different approach on a nonstaggered grid called a Cartesian grid method, which does not use the concept of momentum forcing. In this method, a control volume near the immersed boundary is re-formed into a body-fitted trapezoidal shape by discarding the solid part of the cell and adding the neighboring cells. To accurately discretize the governing equations at the cell containing the immersed boundary, they also presented a new interpolation procedure that preserves the second-order spatial accuracy. However, because the stencil of the trapezoidal boundary cell is different from that of the regular cell, an iteration technique is applied to solve the discretized momentum equations at each computational time step.

The objective of the present study is to develop a new immersed-boundary method that introduces both the momentum forcing and mass source/sink to properly represent the immersed body. The present method is based on a finite-volume approach on a staggered grid together with a fractional-step method. The discrete-time momentum forcing as in Mohd-Yusof [4] and the mass source/sink are applied only on the body surface or inside the body. A second-order-accurate interpolation scheme for evaluating the momentum forcing is proposed in this study, which is numerically stable irrespective of the relative position between the grid and the immersed-boundary. It will be also shown that introduction of the mass source/sink is necessary to obtain physical solutions and enhance accuracy. To verify the accuracy of the present method, three different flow problems (decaying vortices and flows over a circular cylinder and a sphere) are simulated and the results are shown in this paper.

2. NUMERICAL METHOD

2.1. *Governing Equations and Time Integration*

In the present study, the discrete-time momentum forcing, f_i, as in Mohd-Yusof [4] is applied to satisfy the no-slip condition on the immersed boundary. The momentum forcing is applied only on the immersed boundary or inside the body. The forcing points are located in a staggered fashion like the velocity components defined on a staggered grid. When the forcing point coincides with the immersed boundary, we apply momentum forcing so that the velocity is zero at that point (see U_1 and U_2 in Fig. 1a and V_3 and V_4 in Fig. 1b). On the other hand, when the forcing point exists inside the body, we apply momentum forcing so that the velocity (V_1 or U_3) is the opposite of that (v_2 or u_4) outside the body for both the wall-normal and tangential velocity components, respectively, as shown in Figs. 1a and 1b. However, because $U_1 = U_2 = 0$ (no-slip) and V_1 and v_2 come into the cell (as shown in Fig. 1a), the cell containing the immersed boundary does not satisfy the mass conservation. Therefore, we introduce a mass source/sink, q, for the cell containing the immersed boundary to satisfy the mass conservation. The mass source/sink is applied to the cell center on the immersed boundary or inside the body.

The governing equations for unsteady incompressible viscous flow are

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i, \tag{1}$$

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

where x_i's are the Cartesian coordinates, u_i's are the corresponding velocity components, p is the pressure, f_i's are the momentum forcing components defined at the cell faces on the

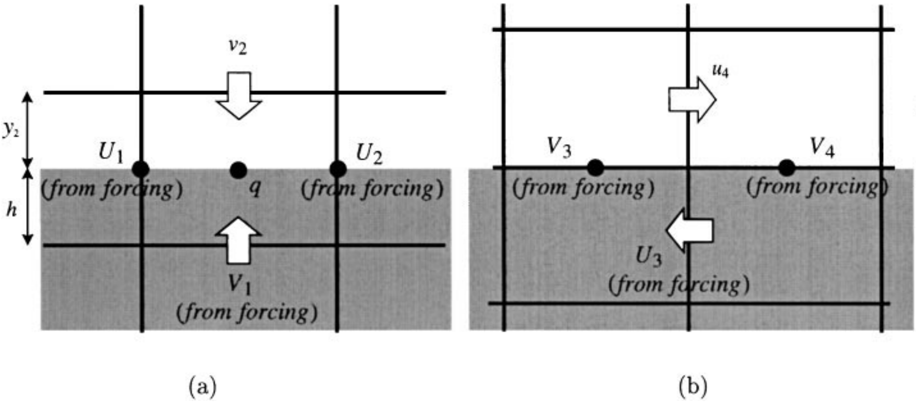


FIG. 1. Velocity vectors near the wall on a staggered mesh: (a) wall-normal velocity; (b) tangential velocity. The shaded area denotes the body. In this figure, we consider a very simple situation for clarity such that $y_2 = h$. More general situations are considered in the following figures.

immersed boundary or inside the body, and q is the mass source/sink defined at the cell center on the immersed boundary or inside the body. All the variables are nondimensionalized by the characteristic velocity and length scales, and Re is the Reynolds number.

The time-integration method used to solve Eqs. (1) and (2) is based on a fractional-step method where a pseudo-pressure is used to correct the velocity field so that the continuity equation is satisfied at each computational time step. In this study, we use a second-order semi-implicit time advancement scheme (a third-order Runge–Kutta method (RK3) for the convection terms and a second-order Crank–Nicolson method for the diffusion terms)

$$\frac{\hat{u}_i^k - u_i^{k-1}}{\Delta t} = \alpha_k L\left(\hat{u}_i^k\right) + \alpha_k L\left(u_i^{k-1}\right) - 2\alpha_k \frac{\partial p^{k-1}}{\partial x_i} - \gamma_k N\left(u_i^{k-1}\right) - \rho_k N\left(u_i^{k-2}\right) + f_i^k, \tag{3}$$

$$\frac{\partial^2 \phi^k}{\partial x_i \partial x_i} = \frac{1}{2\alpha_k \Delta t}\left(\frac{\partial \hat{u}_i^k}{\partial x_i} - q^k\right), \tag{4}$$

$$u_i^k = \hat{u}_i^k - 2\alpha_k \Delta t \frac{\partial \phi^k}{\partial x_i}, \tag{5}$$

$$p^k = p^{k-1} + \phi^k - \frac{\alpha_k \Delta t}{\text{Re}} \frac{\partial^2 \phi^k}{\partial x_j \partial x_j}, \tag{6}$$

where

$$L(u_i) = \frac{1}{\text{Re}} \frac{\partial^2 u_i}{\partial x_j \partial x_j},$$

$$N(u_i) = \frac{\partial u_i u_j}{\partial x_j},$$

$\hat{u}_i$ is the intermediate velocity, and ϕ is the pseudo-pressure. Also, Δt and k are the computational time step and substep's index, respectively, and α_k, γ_k, and ρ_k are the coefficients of RK3 ($\alpha_1 = 4/15$, $\gamma_1 = 8/15$, $\rho_1 = 0$; $\alpha_2 = 1/15$, $\gamma_2 = 5/12$, $\rho_2 = -17/60$; $\alpha_3 = 1/6$, $\gamma_3 = 3/4$, $\rho_3 = -5/12$).

In the present study, the immersed-boundary method is applied to a cylindrical coordinate system as well as a Cartesian one. In the cylindrical coordinate system, the time-integration method is based on the method of Akselvoll and Moin [7] to enhance computational efficiency.

2.2. *Momentum Forcing and Interpolation for the Velocity*

To obtain $\hat{u}_i^k$ from Eq. (3), the momentum forcing f_i^k must be determined in advance such that u_i^k satisfies the no-slip condition on the immersed boundary. Because $u_i^k = \hat{u}_i^k - 2\alpha_k \Delta t \frac{\partial \phi^k}{\partial x_i} = \hat{u}_i^k + \vartheta(\Delta t^2)$, f_i^k is determined such that $\hat{u}_i^k$ satisfies the no-slip condition instead of u_i^k, which does not affect the overall second-order temporal accuracy. (The appropriateness of having the boundary condition accurate to $\vartheta(\Delta t^2)$ for the fractional-step method was shown in Kim and Moin [8].)

When Eq. (1) is provisionally discretized explicitly in time (RK3 for the convection terms and forward Euler method for the diffusion terms) to derive the momentum forcing value, we have

$$\frac{\tilde{u}_i^k - u_i^{k-1}}{\Delta t} = 2\alpha_k L\left(u_i^{k-1}\right) - 2\alpha_k \frac{\partial p^{k-1}}{\partial x_i} - \gamma_k N\left(u_i^{k-1}\right) - \rho_k N\left(u_i^{k-2}\right) + f_i^k. \qquad (7)$$

Rearranging Eq. (7) results in the following equation for f_i^k at a forcing point,

$$f_i^k = \frac{U_i^k - u_i^{k-1}}{\Delta t} - 2\alpha_k L\left(u_i^{k-1}\right) + 2\alpha_k \frac{\partial p^{k-1}}{\partial x_i} + \gamma_k N\left(u_i^{k-1}\right) + \rho_k N\left(u_i^{k-2}\right), \qquad (8)$$

where U_i^k is the velocity that we want to obtain at a forcing point by applying momentum forcing. In the following, $\tilde{u}_i$ or $\tilde{u}_i^k (\neq \hat{u}_i^k$ in Eq. (3)) indicates the velocity at a grid point nearby the forcing point updated from Eq. (7) with $f_i^k = 0$ to determine U_i^k using the linear or bilinear interpolation (see below).

In case of the no-slip wall, U_i^k is zero when the forcing point coincides with the immersed boundary. However, in general the forcing point exists not on the immersed boundary but inside the body, and thus an interpolation procedure for the velocity U_i^k is required. In the present study, second-order linear and bilinear interpolations are used, and Figs. 2–4 show the schematic diagrams for the interpolation scheme used in this study.

Figures 2a and 2b show examples of the use of bilinear and linear interpolations, respectively. In Fig. 2a, P_1 is the point that we want to satisfy the no-slip boundary condition, and U_1, $\tilde{u}_2$, $\tilde{u}_3$, and $\tilde{u}_4$ are four different velocity components surrounding P_1. In the present study, P_1 is defined as the cross-sectional point between the immersed boundary and the wall-normal line passing through the point where U_1 is defined. When all the surrounding velocity components except U_1 exist outside the body as shown in Fig. 2a, the unknown U_1 is obtained from a bilinear interpolation. However, when the bilinear interpolation is not proper because the unknown U_1 is coupled with another unknown U_2 (see Fig. 2b), a linear interpolation is used to obtain U_1 from $\tilde{u}_4$ and the no-slip velocity at P_2, where P_2 is the cross-sectional point between the immersed boundary and the line connecting U_1 and $\tilde{u}_4$.

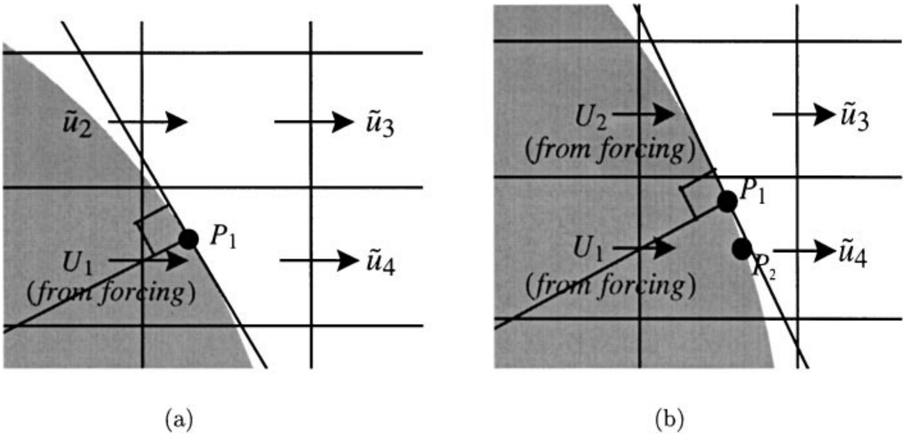
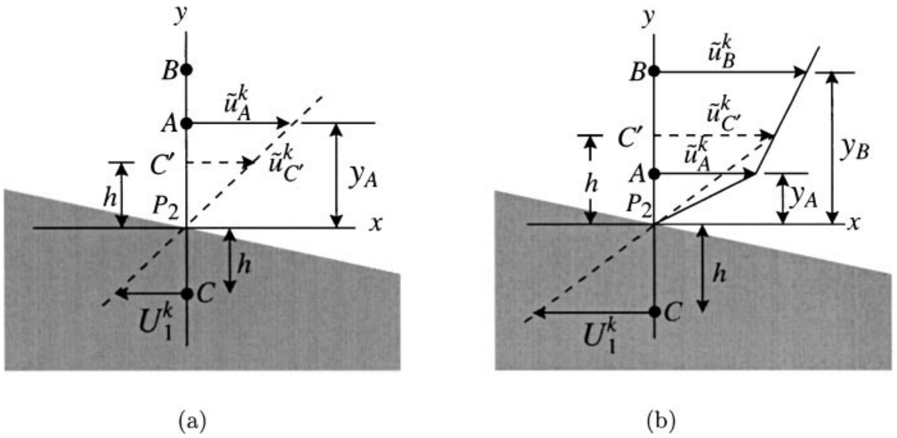


FIG. 2. Schematic diagram for the interpolation scheme: (a) bilinear interpolation; (b) linear interpolation. The shaded area denotes the body.



(a)

(b)

FIG. 3. Linear interpolation scheme for the x-component velocity in case of the no-slip wall: (a) $0 < h \leq y_A$; (b) $h > y_A$. Here, C is the forcing point, and $\overline{C'P_2} = \overline{P_2C}$. A and B are, respectively, the first and second velocity points (center of the cell face) outside the body surface. The shaded area denotes the body.

We now explain the interpolation scheme more specifically in Figs. 3 and 4. First, in Fig. 2b, we consider the following second-order linear interpolation (Fig. 3),

$$U_1^k = -\tilde{u}_{C'}^k, \tag{9}$$

where $\overline{C'P_2} = \overline{P_2C}$. For $0 < h \leq y_A$ (Fig. 3a), $\tilde{u}_{c'}^k$ is obtained from a linear interpolation between $\tilde{u}_A^k$ and the no-slip condition at the point P_2, whereas for $y_A < h < y_B$ (Fig. 3b), $\tilde{u}_C^k$ is obtained from $\tilde{u}_A^k$ and $\tilde{u}_B^k$. That is,

$$U_1^k = \begin{cases} -\frac{h}{y_A}\tilde{u}_A^k & \text{for } 0 < h \leq y_A \\ -\frac{(y_B - h)\tilde{u}_A^k + (h - y_A)\tilde{u}_B^k}{y_B - y_A} & \text{for } y_A < h < y_B. \end{cases} \tag{10}$$

The same scheme is applied to the y-component velocity.

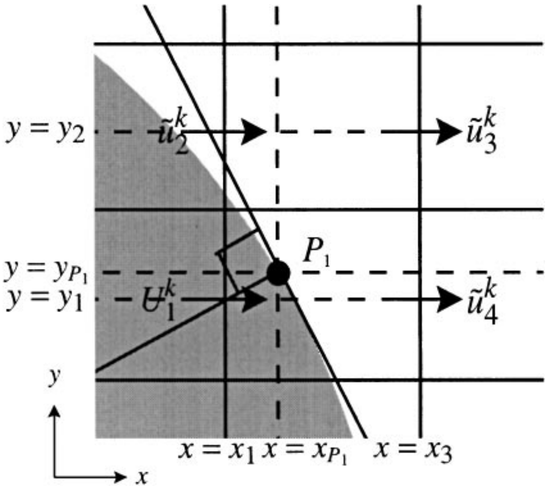


FIG. 4. Bilinear interpolation scheme in case of the no-slip wall. The shaded area denotes the body.

As an alternative, one may consider a simple linear interpolation such as $U_1^k = -\frac{h}{y_A}\tilde{u}_A^k$ even for $y_A < h < y_B$. However, in this case, h/y_A may have a very large value according to the grid distribution in complex geometries and one may fail to get a stable solution due to a nonphysically large value of U_1^k. (For example, our computation using a simple linear interpolation, $U_1^k = -\frac{h}{y_A}\tilde{u}_A^k$, indeed diverged with $h/y_A = 100$ for the decaying-vortex problem in Section 3.1.)

Next, in the case of Fig. 2a, the second-order bilinear interpolation used in this study is (see Fig. 4)

$$U_1^k = -\left[\alpha(1-\beta)\tilde{u}_2^k + (1-\alpha)(1-\beta)\tilde{u}_3^k + (1-\alpha)\beta\tilde{u}_4^k\right]\big/\alpha\beta, \tag{11}$$

where $\alpha = (x_3 - x_{P_1})/(x_3 - x_1)$, $\beta = (y_2 - y_{P_1})/(y_2 - y_1)$, and the no-slip condition at the point P_1 is used. A similar scheme is also applied to the y-component velocity.

With this interpolation scheme (Eqs. (10) and (11), respectively, for cases of Figs. 2b and 2a), we have not encountered any unstable solutions for the flow problems investigated in this study. The interpolation scheme described above can be easily extended to three dimensions as shown in Fig. 5. In Fig. 5a, U_1 is obtained from seven neighboring velocities, whereas U_1 is obtained from three neighboring velocities in Fig. 5b. On the other hand, in Fig. 5c, U_1 is obtained by the same interpolation method as shown in the two-dimensional case, Eq. (10). The formula for each case can be easily derived and thus is not given in this paper.

2.3. *Mass Source and Continuity Equation*

The procedure of obtaining the mass source q^k in Eq. (4) is explained in this section. Consider the two-dimensional cell shown in Fig. 6, where u_1^k and v_1^k are the velocity components inside the body and u_2^k and v_2^k are those outside the body. For the triangular cell containing only fluid ($\triangle P_B P_C P_D$), the continuity reads

$$u_2^k \Delta y + v_2^k \Delta x = 0. \tag{12}$$

Meanwhile, for the rectangular cell containing both the body and the fluid ($\square P_A P_B P_C P_D$), the continuity equation becomes

$$u_2^k \Delta y + v_2^k \Delta x = u_1^k \Delta y + v_1^k \Delta x + q^k \Delta x \Delta y. \tag{13}$$

From Eqs. (12) and (13), the mass source q^k is obtained as

$$q^k = -\frac{u_1^k}{\Delta x} - \frac{v_1^k}{\Delta y}. \tag{14}$$

Note that u_1^k and v_1^k are unknown until Eqs. (4) and (5) are solved and thus we use $\hat{u}_1^k$ and $\hat{v}_1^k$ instead of u_1^k and v_1^k without losing the overall temporal accuracy because $u_i^k = \hat{u}_i^k + \vartheta(\Delta t^2)$. Therefore, the mass source q^k is defined as

$$q^k = -\frac{\hat{u}_1^k}{\Delta x} - \frac{\hat{v}_1^k}{\Delta y} = \frac{1}{\Delta x \Delta y}\left(-\hat{u}_1^k \Delta y - \hat{v}_1^k \Delta x\right). \tag{15}$$

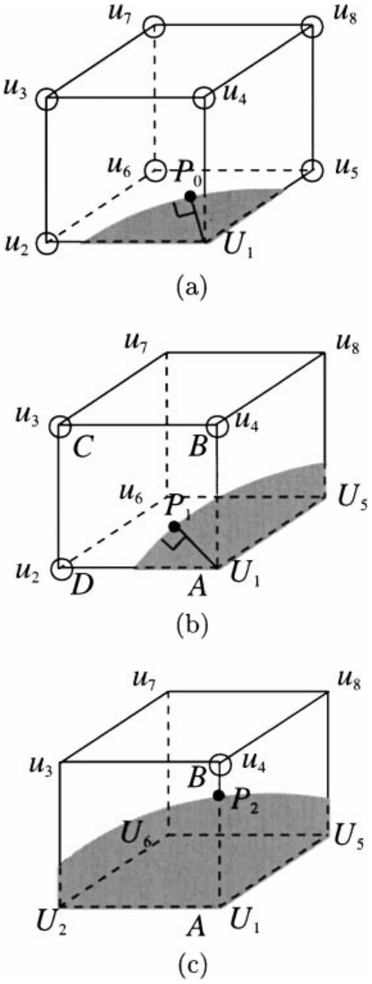


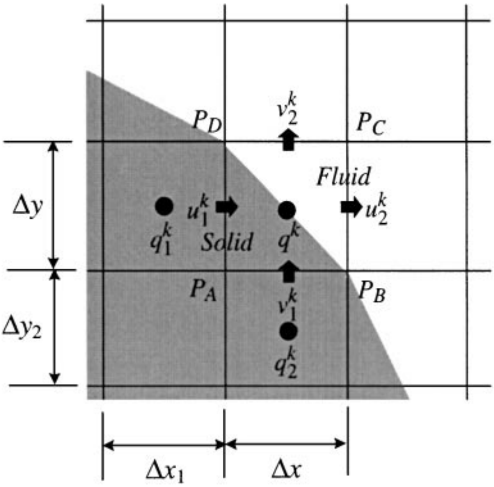
FIG. 5. Schematic diagram for the interpolation scheme in three dimension: (a) three-dimensional extension of bilinear interpolation (P_0 locates inside the cube); (b) bilinear interpolation (P_1 locates on $\Box ABCD$); (c) linear interpolation (P_2 locates on $\overline{AB}$). Here, U_i and u_i locate inside and outside the body, respectively, and ○ represents the neighboring velocity used in obtaining U_1.

Note that $\hat{u}_1^k$ and $\hat{v}_1^k$ in Eq. (15) are obtained from Eq. (3) with nonzero f_i^k. (The method of obtaining f_i^k is already presented in Section 2.2.)

Generally, the mass source q^k for a three-dimensional cell can be expressed as

$$q^k = \frac{1}{\Delta V}\sum_i \omega \hat{\mathbf{u}}^k \cdot \mathbf{n}\Delta S_i, \tag{16}$$

where ΔV is the cell volume, ΔS_i is the area of each cell face, and $\mathbf{n}$ is the unit normal vector outward at each cell face. Here, ω is defined as 1 for the cell face of $f_i^k \neq 0$ and zero elsewhere. Note that $q^k \Delta V$ corresponds to the sum of the volume flux only through the cell faces with nonzero forcings, and two neighboring cells sharing a common cell face have the same value of volume flux but with opposite signs (see Fig. 6). Therefore, the total



$$q^k = \frac{1}{\Delta x \Delta y}\left(-u_1^k \Delta y - v_1^k \Delta x\right)$$

$$q_1^k = \frac{1}{\Delta x_1 \Delta y}\left(u_1^k \Delta y + \cdots\right)$$

$$q_2^k = \frac{1}{\Delta x \Delta y_2}\left(v_1^k \Delta x + \cdots\right)$$

FIG. 6. Mass conservation for a cell containing the immersed boundary.

sum of the volume fluxes due to the mass sources over the computation domain is zero and the global mass conservation is satisfied. Note that in the correction step (Eq. (5)), u_i^k's are updated with the pseudo-pressure obtained from Eq. (4). Near the immersed boundary, the pseudo-pressures with and without q^k are fundamentally different from each other (as is evident from the example shown in Fig. 1a), and thus the velocities u_i^k near the boundary with and without q^k will also be different from each other.

In Fig. 6, we have considered the case where the grid points, P_D and P_B, fall on the immersed boundary, which is not common in general cases. However, the analysis shown above is still valid even in general cases because the volume flux from $\overline{P_D P_C}$, for example, is considered as $v_2^k \Delta x$ in the framework of the immersed-boundary method even if the point P_D exists inside the immersed body. One should try a method similar to that of Ye *et al.* [6] or an unstructured grid method to exactly satisfy the mass and momentum fluxes near the body surface. However, that kind of method inevitably introduces an iterative process to solve the momentum equation at each computational time step, which may increase the CPU time especially for three-dimensional flow. In Sections 3.1 and 3.2, we show that the present approach of introducing q clearly enhances the quality of the solution near the immersed boundary.

3. NUMERICAL EXAMPLES

3.1. *Decaying Vortices*

The temporal and spatial accuracy of the present immersed boundary method is verified by simulating the two-dimensional unsteady flow such as

$$u(x, y, t) = -\cos \pi x \sin \pi y e^{-2\pi^2 t/\mathrm{Re}}, \tag{17}$$

$$v(x, y, t) = \sin \pi x \cos \pi y \, e^{-2\pi^2 t/\mathrm{Re}}, \tag{18}$$

$$p(x, y, t) = -\frac{1}{4}(\cos 2\pi x + \cos 2\pi y)\, e^{-4\pi^2 t/\mathrm{Re}}. \tag{19}$$

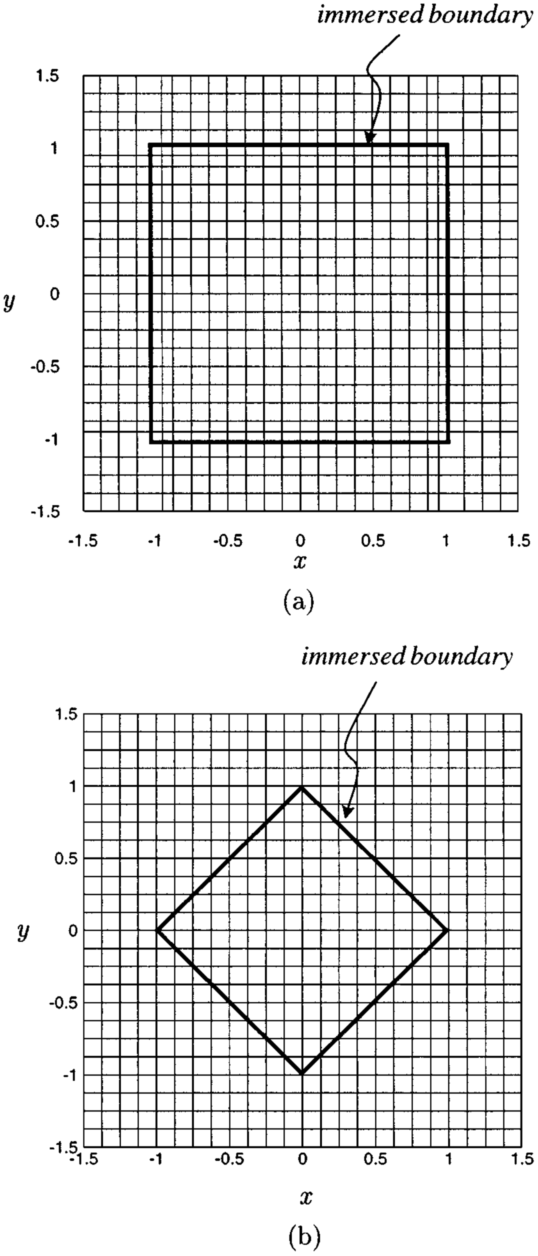


FIG. 7. Grid system, computational domain, and immersed boundary for the decaying vortices: (a) an immersed boundary parallel to the grid lines; (b) an immersed boundary at an oblique angle of 45° with respect to the grid lines.

Figure 7 shows the grid system, computational domain ($-1.5 \leq x, y \leq 1.5$), and immersed boundary (thick lines). The immersed boundary (thick lines at $x = \pm 1$, $y = \pm 1$, respectively) in Fig. 7a is parallel to the grid lines where the linear interpolation (Eq. (10)) is used. On the other hand, the immersed boundary (thick lines at $|x| + |y| = 1$) in Fig. 7b is oblique at 45° with respect to the grid lines where the bilinear interpolation (Eq. (11)) is used. The momentum forcing is applied at the points outside the immersed boundary and we are concerned with the solution inside. Note that near the immersed boundary in Fig. 7a, the grid size is intentionally changed so that the tangential velocity to the immersed boundary is interpolated as in Fig. 3a and the normal velocity as in Fig. 3b. Meanwhile, in Fig. 7b, the grid size remains uniform over the computational domain. For both cases of Figs. 7a and 7b, four different sizes of Cartesian grids are used and the corresponding numbers of grid points in each direction are 25 (17), 49 (33), 97 (65), and 193 (129), respectively. Here, the numbers of grid points in parentheses denote those located in $-1.0 \leq x, y \leq 1.0$. Computations are performed by varying the mesh size but keeping the maximum CFL number at $t = 0$ constant. The initial velocity condition at $t = 0$ and the velocities at the boundaries of the computational domain in time are provided from the exact solution. For all the computations, the Reynolds number ($\mathrm{Re} = UL/\mu$) is prescribed to be 10, where U is the initial maximum velocity and L is the size of a vortex.

Figure 8 shows the variation of the maximum error in u inside the immersed boundary with the mesh refinement. The errors are obtained for three cases: with both the momentum forcing and the mass source, with the momentum forcing only, and without the momentum forcing (i.e., without the immersed-boundary method), respectively. It is shown that the errors with the mass source are much smaller than those without the mass source and are comparable to those obtained without using the immersed-boundary method. From this simulation, it is verified that the linear and bilinear interpolations proposed in the present study are second-order accurate in space, and introducing the mass source enhances the quality of the solution.

3.2. *Flow over a Circular Cylinder*

Flow over a circular cylinder is simulated to verify that the present immersed-boundary method predicts flow phenomena such as the separation and vortex shedding very well. The size of the computational domain is $70d \times 100d$, where d denotes the diameter of the cylinder. Computations are performed at two different Reynolds numbers (40 and 100) based on the freestream velocity u_∞ and diameter with the maximum CFL number of 1.5. The numbers of grid points are 331×193 in the streamwise (x) and transverse (y) directions, respectively, and 30 grid points in each direction are uniformly distributed within the cylinder diameter. Grids are distributed in the surface-normal direction above the cylinder surface such that about six to eight grid points are located inside the boundary layer at $60^\circ \leq \theta \leq 90^\circ$ ($\mathrm{Re} = 100$), where θ is the angle from the stagnation point. Here, the boundary layer thickness is estimated as the position where the surface-parallel velocity is maximum at each θ. A Dirichlet boundary condition ($u/u_\infty = 1$, $v = 0$) is used at the inflow and farfield boundaries, and a convective boundary condition ($\partial u_i/\partial t + c\partial u_i/\partial x = 0$) is used for the outflow boundary where c is the space-averaged streamwise velocity at the exit.

Table I shows that the present drag and lift coefficients agree well with the computational result obtained from a body-fitted mesh (Park *et al.* [9]), where C_D is the drag coefficient (time-averaged value in case of $\mathrm{Re} = 100$) and C_L' is the amplitude of lift-coefficient fluctuations (maximum deviation from the time-averaged value) at $\mathrm{Re} = 100$. In this study, the drag and lift forces are obtained by directly integrating all the momentum forcing values applied inside the body. A useful discussion on how to evaluate the force on a solid body in conjunction with the immersed-boundary method can be found in Lai and Peskin [10]. The Strouhal number ($\mathrm{St} = fd/u_\infty$) is also in excellent agreement with those of Park *et al.* [9] and Williamson [11], where f is the shedding frequency.

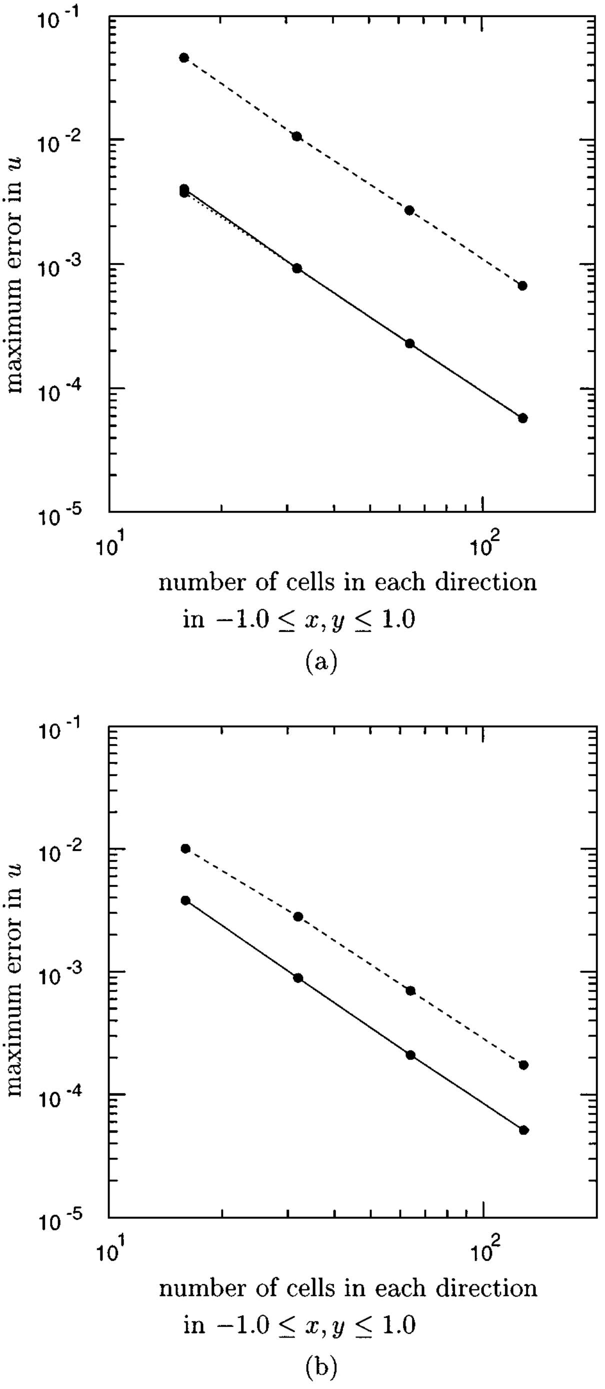


FIG. 8. Maximum error in u at $t = 0.3$: (a) case of Fig. 7a (linear interpolation); (b) case of Fig. 7b (bilinear interpolation). ——, momentum forcing with the mass source; — — —, momentum forcing without the mass source; · · · ·, without the immersed-boundary method.

TABLE I
Simulation Results for Flow over a Circular Cylinder

	Re	C_D	C_L'	St
Present	40	1.51		
	100	1.33	0.32	0.165
Park *et al.* [9]	40	1.51		
	100	1.33	0.33	0.165
Williamson [11]	100			0.164

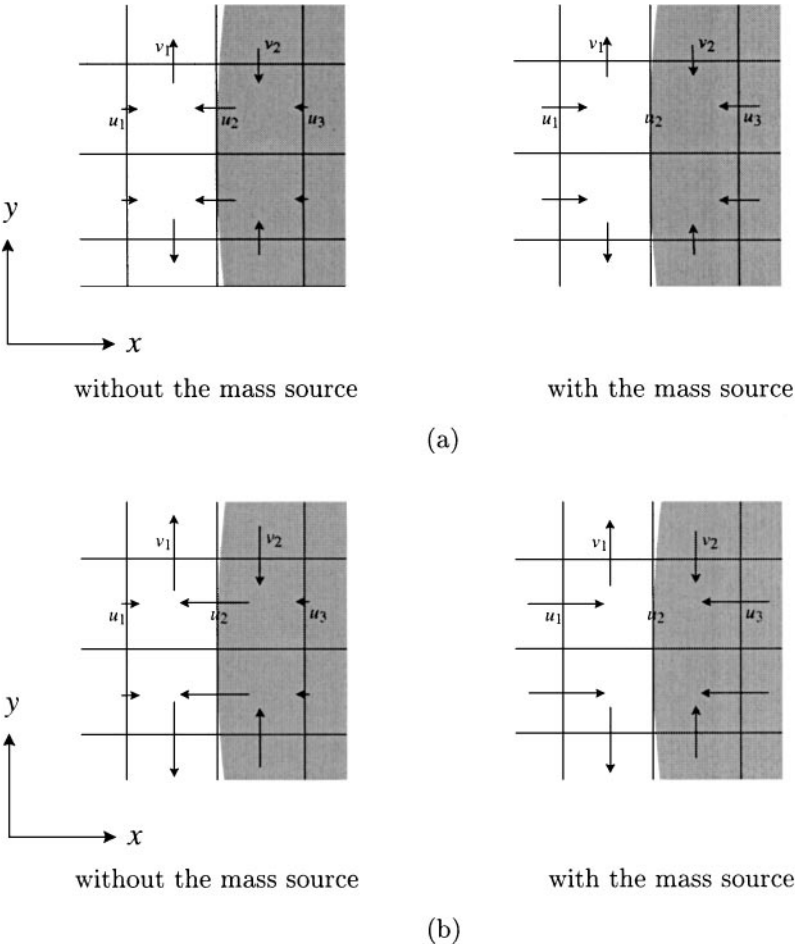


FIG. 9. Velocity vectors near the stagnation point of a circular cylinder. (a) Re = 40: without the mass source ($u_1 = 0.0066$, $u_2 = -0.0240$, $u_3 = -0.0030$, $v_1 = 0.0271$, $v_2 = -0.021$); with the mass source ($u_1 = 0.0274$, $u_2 = 3.84 \times 10^{-8}$, $u_3 = -0.0238$, $v_1 = 0.0242$, $v_2 = -0.0189$); (b) Re = 100: without the mass source ($u_1 = 0.0110$, $u_2 = -0.0415$, $u_3 = -0.0054$, $v_1 = 0.0464$, $v_2 = -0.0361$); with the mass source ($u_1 = 0.0465$, $u_2 = -6.368 \times -10^{-9}$, $u_3 = -0.0408$, $v_1 = 0.0414$, $v_2 = -0.0322$). The velocities at Re = 100 are the time-averaged values.

Figure 9 shows the velocity vectors near the stagnation point of a circular cylinder for two different cases: with and without the mass source, respectively. The arrows at the cell faces denote the velocity vectors. It is clear for both Re = 40 and 100 that without the mass source the wall-normal velocities near the stagnation point (u_2) have relatively large negative (and thus nonphysical) values, whereas with the mass source they are nearly zero. Furthermore, without the mass source, the negative value of u_2 becomes larger as the Reynolds number increases, indicating that introducing the mass source is more necessary for obtaining realistic solutions at higher Reynolds number. Also, without the mass source, $C_D = 1.28$ and $C_L' = 0.28$ at Re = 100, which are smaller than those with the mass source (see Table I).

Figure 10 shows the pressure coefficient and spanwise vorticity variations along the cylinder surface at Re = 40 and 100, together with the numerical results of Park *et al.* [9].

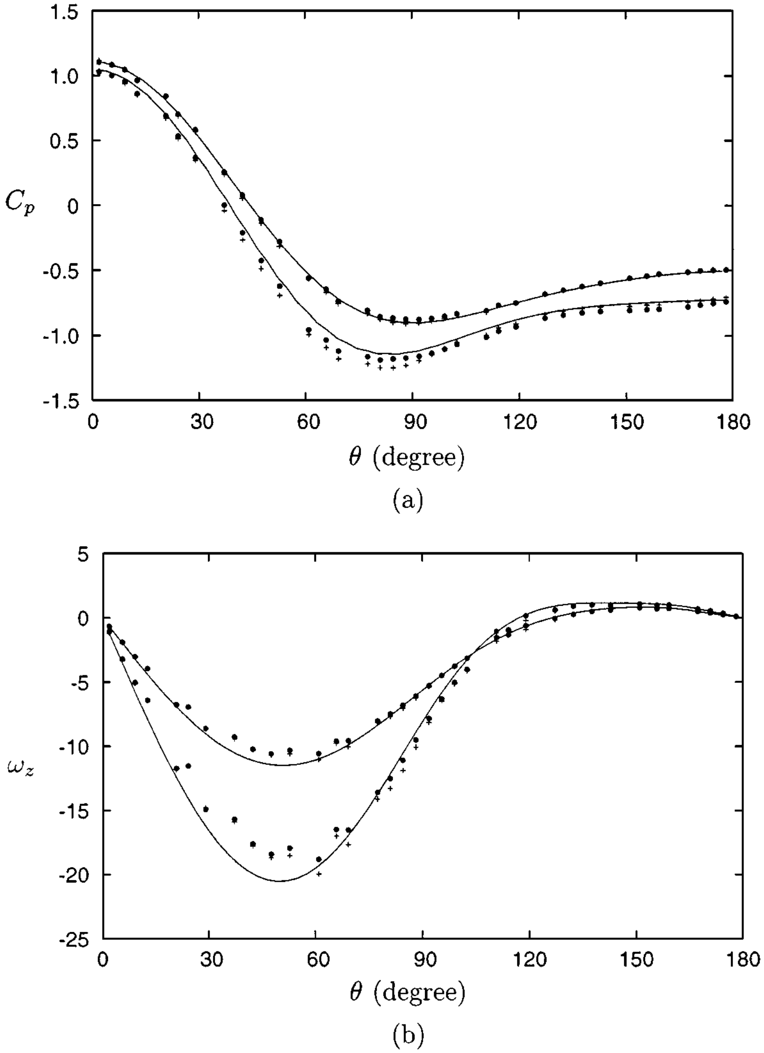


FIG. 10. Flow variables along the cylinder surface at Re = 40 and 100: (a) wall pressure coefficient; (b) wall spanwise vorticity. •, with the mass source (present study); +, without the mass source (present study); ——, Park *et al.* [9]. The C_p and ω_z at Re = 100 are the time-averaged values.

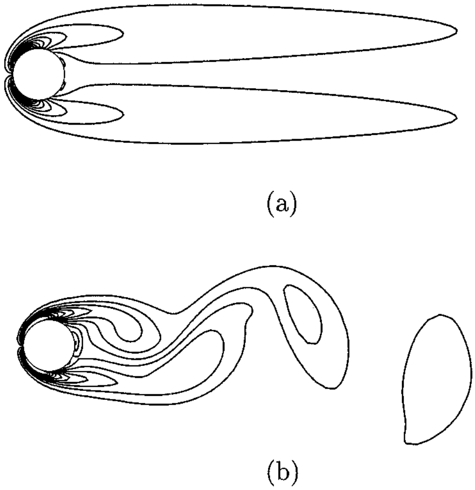


FIG. 11. Spanwise vorticity contours near a circular cylinder: (a) Re = 40; (b) Re = 100. For Re = 100, instantaneous vorticity contours are drawn.

Here, a proper interpolation scheme for measuring the pressure and vorticity at the cylinder surface is required because the grid lines generally do not coincide with the immersed boundary. In the present study, the pressure at the surface is obtained from the nearest cell center pressure outside the body by assuming that the wall-normal derivative of the pressure is zero at the surface. The wall vorticity is obtained from the velocity tangential to the surface at the nearest cell center outside the body. This interpolation is similar to a conventional approach used in a body-fitted staggered mesh. It is shown in Fig. 10 that the present C_p with the mass source agrees well with that of Park *et al.* [9], and the difference between the results with and without the mass source becomes large with increasing Reynolds number. Also, the present immersed-boundary method reasonably captures the vorticity at the cylinder surface. We have confirmed that the magnitude of mass source is also large at $60^\circ \leq \theta \leq 90^\circ$, where the difference between the results with and without the mass source is large.

Figure 11 shows the spanwise vorticity contours for Re = 40 and 100. The flow is symmetric about the streamwise axis at Re = 40, whereas vortex shedding takes place at Re = 100, indicating that the vorticity field is well captured by the present immersed-boundary method.

3.3. *Flow over a Sphere*

As an example of three-dimensional flow, laminar flow over a sphere is simulated at three different Reynolds numbers (Re = 100, 250, and 300), each of which represents one of three

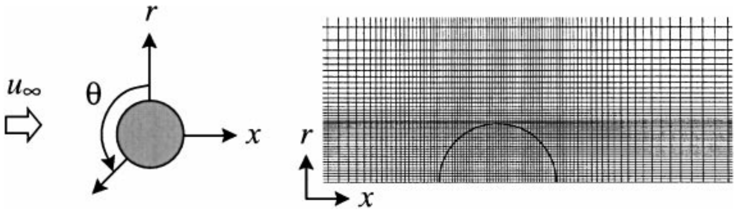
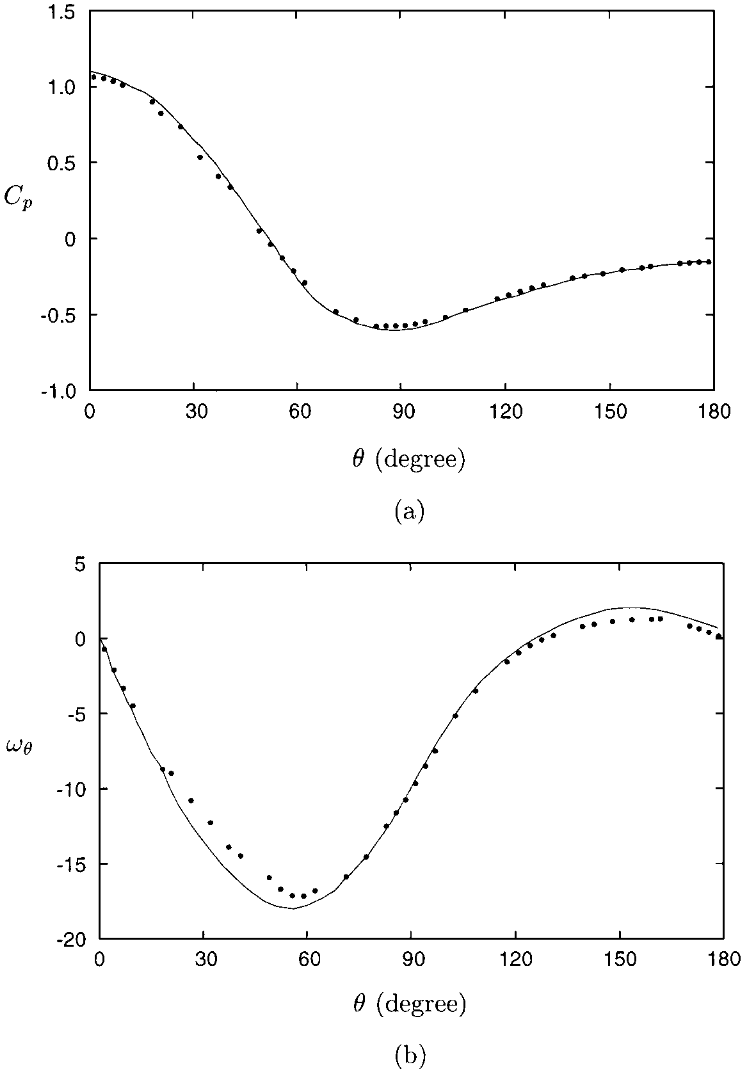


FIG. 12. Coordinate and grid systems near a sphere for Re = 100 and 250.

laminar flow regimes: steady axisymmetric flow (Re $\leq$ 200), steady nonaxisymmetric flow (210 $\leq$ Re $\leq$ 270), and unsteady flow (Re $\geq$ 280). The Reynolds number is based on the uniform inlet velocity u_∞ and the diameter of the sphere d. In this study, we use the cylindrical coordinate system rather than the Cartesian one, because the former requires fewer grid points than the latter by fitting the grids to the sphere surface in the planes perpendicular to the streamwise direction. The coordinate and grid systems near a sphere are shown in Fig. 12. The computational domain used is $-15d \leq x \leq 15d$, $0 \leq r \leq 15d$, and $0 \leq \theta \leq 2\pi$. The numbers of grid points are $145(x) \times 61(r) \times 40(\theta)$ and $289(x) \times 161(r) \times 40(\theta)$, respectively, for steady and unsteady flows.

Table II shows the results of simulations together with the previous numerical results of Fornberg [12], Johnson and Patel [13], and Constantinescu and Squires [14] in which



(a)

(b)

FIG. 13. Flow variables along the sphere surface at Re = 100: (a) wall pressure coefficient; (b) wall azimuthal vorticity. ●, Present study; ——, Johnson [16].

TABLE II
Simulation Results for Flow over a Sphere

	Re	C_D	C_L	St
Present	100	1.087		
	250	0.701	0.059	
	300	0.657	0.067	0.134
Fornberg [12]	100	1.085		
Johnson and Patel [13]	250	0.70	0.062	
	300	0.656	0.069	0.137
Constantinescu and Squires [14]	250	0.70	0.062	
	300	0.655	0.065	0.136

body-fitted grids were used. The drag coefficient C_D at Re = 100 agrees very well with that of Fornberg [12]. The drag and lift (C_L) coefficients at Re = 250 and 300 are also in excellent agreement with those of Johnson and Patel [13] and Constantinescu and Squires [14], where C_D and C_L are the time-averaged values for unsteady flow (Re = 300). The nonzero lift coefficient at Re = 250 indicates that the flow loses the axisymmetry but maintains the steadiness. The present Strouhal number at Re = 300 is in reasonable agreement with those of Johnson and Patel [13] and Constantinescu and Squires [14], but all the numerical results (present study, [13, 14]) are smaller than the experimental result (St = 0.15–0.16) of Sakamoto and Haniu [15].

We have also simulated flow over a sphere at Re = 100 and 250 in the Cartesian coordinate system instead of the cylindrical one. In the Cartesian coordinate, the sphere is considered a complex geometry in three dimensions, and thus a three-dimensional extension of the interpolation scheme explained in Section 2.2 is used to describe the sphere surface. The simulation result shows that the drag and lift coefficients ($C_D = 1.088$ at Re = 100; $C_D = 0.706$, $C_L = 0.061$ at Re = 250) agree well with those in the cylindrical coordinate. However, the number of grid points required is much larger ($145(x) \times 113(y) \times 113(z)$)

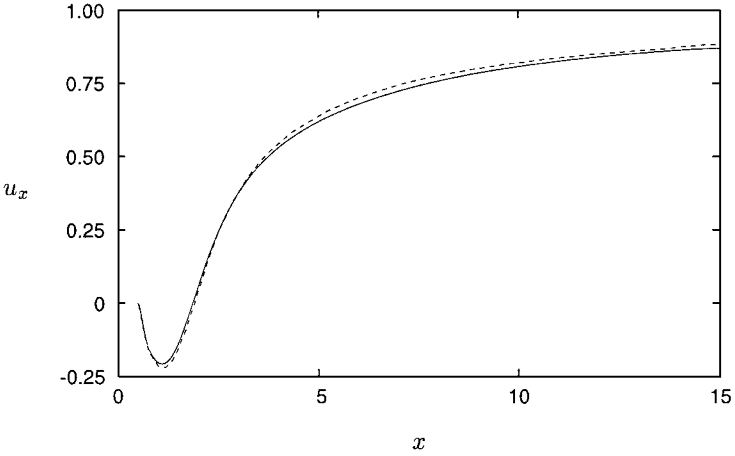


FIG. 14. Time-averaged streamwise velocity along the x-axis at Re = 300:——, present study;– – –, Johnson and Patel [13].

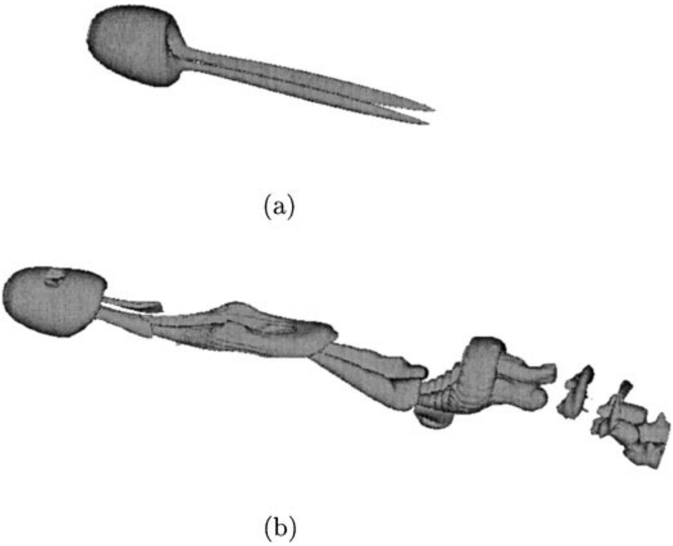


FIG. 15. Vortical structures at (a) Re = 250 and (b) Re = 300. For Re = 300, an instantaneous field is drawn.

than that in the cylindrical coordinate $(145(x) \times 61(r) \times 40(\theta))$, as expected. Therefore, all the simulation results for flow over a sphere presented in this section are those performed in the cylindrical coordinate system.

Figure 13 shows the pressure coefficient and the azimuthal vorticity variations along the sphere surface at Re = 100, where the interpolation method used for evaluating flow variables at the surface is the same as that for the flow over a cylinder. Comparison with the results using a body-fitted mesh (Johnson [16]) indicates that the present immersed-boundary method represents the spherical surface very well. Figure 14 shows the time-averaged streamwise velocity along the x-axis at Re = 300 together with that of Johnson and Patel [13]. The streamwise velocity and the size of the recirculation region $(1.36d)$ agree very well with those of Johnson and Patel [13].

Figure 15 shows three-dimensional vortical structures at Re = 250 and 300. Here, the vortical surfaces are identified using the method of Jeong and Hussain [17]. At Re = 250, the flow is steady nonaxisymmetric, whereas at Re = 300 the flow is unsteady and the vortices are asymmetrically shed. This behavior is nearly the same as that shown in Johnson and Patel [13], indicating that the present method accurately captures the three-dimensional vorticity field.

4. SUMMARY

In this study, a new immersed-boundary method is presented by introducing the mass source/sink as well as the momentum forcing. The present method is based on a finite-volume approach on a staggered mesh together with a fractional-step method. The momentum forcing and the mass source/sink are applied on the body surface or inside the body to satisfy the no-slip boundary condition on the immersed boundary and the continuity for the cell containing the immersed boundary, respectively.

A new second-order linear or bilinear interpolation scheme is developed to satisfy the no-slip velocity on the immersed boundary, which is numerically stable regardless of the relative position between the grid and the immersed boundary.

Three different flow problems (decaying vortices and flows over a cylinder and a sphere) are simulated with the immersed-boundary method proposed in this study. It is shown that without introducing the mass source/sink near the immersed boundary a nonphysical solution is obtained, especially near the stagnation point of flow, and the numerical solution deteriorates more as the Reynolds number increases. The simulation results with both the momentum forcing and mass source/sink agree very well with the previous numerical and experimental results, indicating the validity and accuracy of the present immersed-boundary method.

ACKNOWLEDGMENTS

This work was supported by Creative Research Initiatives of the Korean Ministry of Science and Technology and BK21 of the Korean Ministry of Education. The computations were performed on the CRAY YMP C90 at the KORDIC supercomputer center.

REFERENCES

1. C. S. Peskin, The fluid dynamics of heart valves: Experimental, theoretical, and computational methods, *Annu. Rev. Fluid Mech.* **14**, 235 (1982).
2. D. Goldstein, R. Handler, and L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* **105**, 354 (1993).
3. E. M. Saiki and S. Biringen, Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method, *J. Comput. Phys.* **123**, 450 (1996).
4. 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), p. 317.
5. E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* **161**, 35 (2000).
6. T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* **156**, 209 (1999).
7. K. Akselvoll and P. Moin, *Large Eddy Simulation of Turbulent Confined Coannular Jets and Turbulent Flow over a Backward Facing Step*, Report No. TF-63 (Thermoscience Division, Department of Mechnical Engineering, Stanford University, 1995).
8. J. Kim and P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, *J. Comput. Phys.* **59**, 308 (1985).
9. J. Park, K. Kwon, and H. Choi, Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160, *KSME Int. J.* **12**, 1200 (1998).
10. M.-C. Lai and C. S. Peskin, An immersed-boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* **160**, 705 (2000).
11. C. H. K. Williamson, Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers, *J. Fluid Mech.* **206**, 579 (1989).
12. B. Fornberg, Steady viscous flow past a sphere at high Reynolds numbers, *J. Fluid Mech.* **190**, 471 (1988).
13. T. A. Johnson and V. C. Patel, Flow past a sphere up to a Reynolds number of 300, *J. Fluid Mech.* **378**, 19 (1999).
14. G. S. Constantinescu and K. D. Squires, *LES and DES Investigations of Turbulent Flow over a Sphere*, AIAA Paper 2000–0540 (AIAA Press, Washington, DC, 2000).
15. H. Sakamoto and H. Haniu, A study on vortex shedding from spheres in a uniform flow, *J. Fluid Eng.* **112**, 386 (1990).
16. T. A. Johnson, Numerical and Experimental Investigation of Flow past a Sphere up to a Reynolds Number of 300, Ph.D. dissertation (University of Iowa, 1996).
17. J. Jeong and F. Hussain, On the identification of a vortex, *J. Fluid Mech.* **285**, 69 (1995).