INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **53**:1659–1671
Published online 25 October 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1367

Improvement of mass source/sink for an immersed boundary method

Wei-Xi Huang and Hyung Jin Sung*, †

Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-ku, Taejon 305-701, Korea

SUMMARY

An improved immersed boundary method using a mass source/sink as well as momentum forcing is developed for simulating flows over or inside complex geometries. The present method is based on the Navier–Stokes solver adopting the fractional step method and a staggered Cartesian grid system. A more accurate formulation of the mass source/sink is derived by considering mass conservation of the virtual cells in the fluid crossed by the immersed boundary. Two flow problems (the decaying vortex problem and uniform flow past a circular cylinder) are used to validate the proposed formulation. The results indicate that the accuracy near the immersed boundary is improved by introducing the accurate mass source/sink. Copyright © 2006 John Wiley & Sons, Ltd.

Received 5 April 2006; Revised 13 July 2006; Accepted 6 August 2006

KEY WORDS: immersed boundary method; mass source/sink

1. INTRODUCTION

Recently the immersed boundary (IB) method using momentum forcing and a Cartesian grid Navier–Stokes (N–S) solver has received much attention due to its ability to simulate viscous flows over or inside complex geometries. Depending on how the momentum forcing is applied, the IB method can be classified into two categories [1]: the continuous forcing approach [2] and the discrete forcing approach [3–7]. Compared with the continuous forcing approach, the discrete forcing approach has the advantage of sharp representation of the IB and requires less stability constraint.

In the discrete forcing approach, the values at neighbouring points are interpolated to obtain the desired boundary conditions at the IB, and the momentum forcing is obtained directly from the

*Correspondence to: Hyung Jin Sung, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-ku, Taejon 305-701, Korea.
†E-mail: hjsung@kaist.ac.kr

discretized equation of motion. Many interpolation schemes for the momentum forcing have been developed to simulate an IB of arbitrary shape [3–5]. However, these schemes do not properly account for mass conservation of the cells crossed by the IB. Although mass conservation can be achieved by the N–S solver for all computational cells regardless of the presence of the IB, it is not preserved for the virtual cells in the fluid obtained by cutting the cell with the IB and discarding the solid part (simply referred to as 'virtual cell' in this paper). Thus in practice the non-physical solution inside the solid region influences the physical solution in the fluid region. This problem can be avoided by using the Cartesian grid method (also known as the cut-cell method) [6], in which the solid part of the cell crossed by the IB is discarded and the fluid part either forms a new cell or is absorbed by a neighbouring fluid cell. However, this reshaping procedure introduces substantial complexities in terms of the modifications that must be made to the N–S solver. In addition, the velocity field must be solved by iteration at each time step, which substantially increases the computation time. Kang *et al*. [7] considered strict and approximate mass conservation of the virtual cells, but no obvious improvement can be found in their results when mass conservation of the virtual cells was taken into account.

Another approach of accounting for mass conservation of the cells crossed by the IB, proposed by Kim *et al*. [4], is to introduce a mass source/sink into the continuity equation. In practice, the only modification of the original N–S solver that was required was the addition of a source term to the right-hand side of the pressure Poisson equation. This greatly simplified the manipulation and increased the computation time only slightly. Importantly, using this approach the velocity field can be solved without iteration at each time step. When the mass source/sink term was employed, Kim *et al*. [4] found that the quality of the solution was improved and non-physical solution was corrected. However, they formulated the mass source/sink term in an approximate manner, i.e. the grid points fall on the IB. In the present study, we shall show that this approximation may degrade the quality of the solution. Thus a more accurate mass source/sink term is formulated by introducing face-centred velocities of the virtual cells. Bottino [8] also used a mass source/sink which was added to the continuity equation for establishing the pressure drop across the opening of the pipette. However, his approach was based on the continuous forcing approach applied to the immersed elastic boundary [9, 10].

In the next section, we introduce the general numerical procedures, including the N–S solver and the interpolation schemes. Details of the formulation of the mass source/sink term are described in Section 3 and some numerical examples are shown in Section 4. Finally, the conclusion is drawn in Section 5.

2. NUMERICAL METHODS

The present IB method is based on the N–S solver adopting the fractional step method and a staggered Cartesian grid system [11]. The non-dimensional governing equations for an unsteady incompressible flow are:

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

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

DOI: 10.1002/fld

where $\mathbf{u}$ is the velocity vector, p is the pressure, $\mathbf{f}$ is momentum forcing, q is the mass source/sink and Re is the Reynolds number. A fully implicit time advancement is employed where the Crank–Nicholson scheme is used for both the diffusion and convection terms. The governing equations are discretized as follows:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}+N\mathbf{u}^{n+1}=-Gp^{n+1/2}+\frac{1}{2Re}(L\mathbf{u}^{n+1}+L\mathbf{u}^n)+\mathbf{f}^{n+1/2} \tag{3}$$

$$D\mathbf{u}^{n+1}-q^{n+1/2}=0 \tag{4}$$

where N, G, L and D are the linearized discrete convective operator, the discrete gradient operator, the discrete Laplacian operator and the discrete divergence operator, respectively. Here n denotes the nth time step and Δt denotes the time increment. Velocity and pressure are then decoupled by a block decomposition in conjunction with the method of approximate factorization. Furthermore, since the intermediate velocity components are coupled due to the implicit Crank–Nicholson representation for the convective term, additional decoupling for the intermediate velocity components is achieved by the approximate factorization with only the nth time step velocity. The decoupled momentum equations are solved without iteration and the temporal second-order accuracy is preserved. The pressure Poisson equation is solved either by a direct method using FFT or a multigrid method. Details regarding the present N–S solver can be found in Reference [11].

If the momentum forcings are calculated implicitly [3], a large sparse matrix is introduced for a complicate interpolation scheme which gives a significant increase of computing cost. Instead, we use a prediction step by the forward Euler explicit scheme

$$\frac{\mathbf{u}^{\#}-\mathbf{u}^n}{\Delta t}+N\mathbf{u}^n=-Gp^{n-1/2}+\frac{1}{Re}L\mathbf{u}^n \tag{5}$$

where $\mathbf{u}^{\#}$ is the predicted velocity. Then the momentum forcing is simply expressed as

$$\mathbf{f}^{n+1/2}=\frac{\mathbf{U}^{\#}-\mathbf{u}^{\#}}{\Delta t} \tag{6}$$

where $\mathbf{U}^{\#}$ denotes the velocity which we want to obtain at the forcing point in order to satisfy the desired no-slip condition at the IB by interpolation from the neighbouring points. The above equation indicates that the momentum forcing is proportional to the difference between the desired velocity and the predicted velocity. So it can be regarded as a special case of the feedback forcing [12, 13], while the momentum forcing here is calculated at discrete points instead of distributing into the neighbouring points by the discretized Dirac delta function.

The desired velocity $\mathbf{U}^{\#}$ at the forcing point is obtained by interpolation using the predicted velocity field $\mathbf{u}^{\#}$ [4]. Assume that the forcing points are located inside the IB. If two neighbouring points of the forcing point (X_0) are located in the fluid region, as shown in Figure 1(a), three interpolations are chosen:

(1) linear interpolation:

$$\phi=a_0+a_1x+a_2y \tag{7}$$

(2) bilinear interpolation:

$$\phi=a_0+a_1x+a_2y+a_3xy \tag{8}$$

DOI: 10.1002/fld

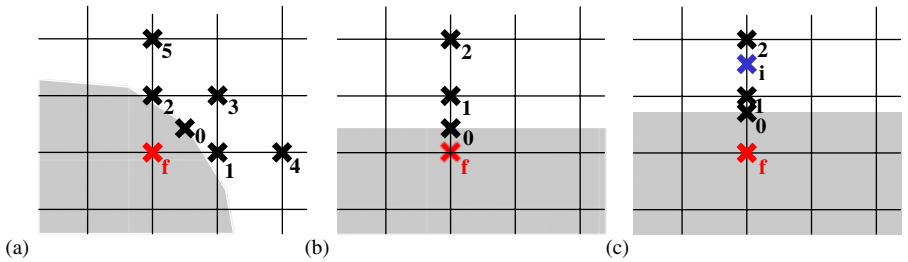

Figure 1. Schematic diagram for the interpolation schemes: (a) two neighbouring points are located in the fluid region; (b) only one neighbouring point is located in the fluid region and the boundary point is close to the forcing point; and (c) only one neighbouring point is located in the fluid region and the boundary point is close to the neighbouring fluid point.

(3) quadratic interpolation:

$$\phi = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \tag{9}$$

where ϕ is an arbitrary physical quantity. Three node values (X_0, X_1 and X_2) are used to determine the coefficients (a_0, a_1 and a_2) of the linear interpolation

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{bmatrix}^{-1} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \end{bmatrix} \tag{10}$$

while four node values (X_0, X_1, X_2 and X_3) are needed for the bilinear interpolation and six node values (X_0, X_1, X_2, X_3, X_4 and X_5) are needed for the quadratic interpolation. Note that the coefficients of large absolute value are produced which may lead to numerical instability when X_0 is near or on the line connecting X_1 and X_2 for the linear interpolation, and when X_0 is near X_1 or X_2 for the quadratic interpolation. The bilinear interpolation (Equation (8)) is most robust for arbitrary position of X_0, and therefore is used for Figure 1(a).

If there is only one neighbouring point located in the fluid region, as shown in Figure 1(b), the linear interpolation is used as

$$\phi = a_0 + a_1 y \tag{11}$$

while the quadratic interpolation is used as

$$\phi = a_0 + a_1 y + a_2 y^2 \tag{12}$$

However, when X_0 is close to X_1, as shown in Figure 1(c), both the linear and quadratic interpolations produce large coefficients. An efficient remedy for this case is to introduce the image point X_i of the forcing point X_f [4, 5, 14]. First the value at the image point is interpolated from its neighbouring points and then the value at the forcing point is calculated according to the relation,

$$\phi_f = 2\phi_0 - \phi_i \tag{13}$$

DOI: 10.1002/fld

Since the above equation is spatially second order, use of the quadratic interpolation does not improve the spatial accuracy [5]. For simplicity, the linear interpolation is used for Figure 1(b) and (c), and the image point is introduced only for Figure 1(c).

3. FORMULATION OF THE MASS SOURCE/SINK TERM

In the present study, we focus on the derivation of a more accurate mass source/sink term. Consider a two-dimensional flow near the stagnation point. The velocities in the fluid region and those at the forcing points obtained from interpolation are displayed in Figure 2(a). For simplicity, the superscripts of the velocity components are neglected. Without special description, they denote the intermediate velocity components in this section. Before pressure correction, mass conservation is not preserved for the computational cell. Without the mass source/sink term, after solving the pressure field and correcting the velocity field, v_2 changes its direction to preserve the mass conservation (see the dashed arrow for v_2), while u_1, u_2 and v_1 keep their directions due to the applied momentum forcings. Note in this diagram that v_1 is obtained not only from the boundary value and v_2, but also from another neighbouring point of v_2, due to the use of the image point. This non-physical solution was also presented in Reference [4], and was corrected by introducing the mass source/sink term

$$q = \frac{1}{\Delta V} \sum_i \omega \mathbf{u} \cdot \mathbf{n} \Delta S_i \tag{14}$$

where ΔV is the cell volume, ΔS_i is the area of each cell face, $\mathbf{n}$ is the unit normal vector outward at each cell face, and ω is defined as 1 for forcing points of $f_i \neq 0$ and 0 elsewhere. In the present case, as shown in Figure 2(a), Equation (14) becomes

$$q = \frac{u_2}{\Delta x} - \frac{u_1}{\Delta x} - \frac{v_1}{\Delta y} \tag{15}$$

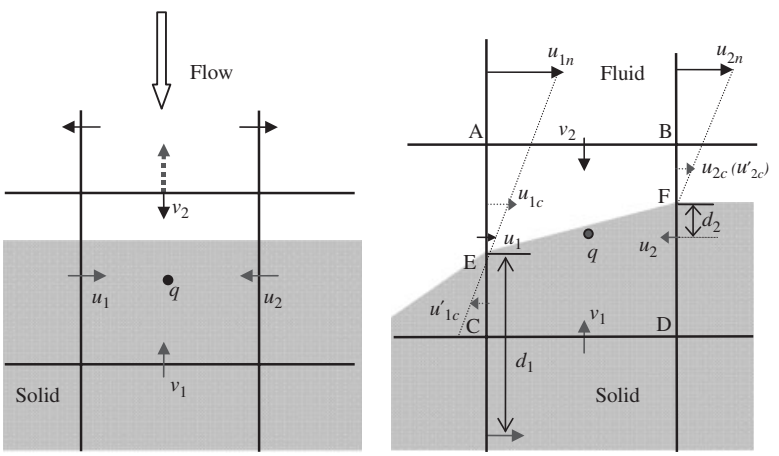

Figure 2. Schematic diagram of the mass source/sink term: (a) flow near the stagnation point; and (b) the general case.

DOI: 10.1002/fld

The discretized continuity equation is represented by

$$u_2\Delta y - u_1\Delta y + v_2\Delta x - v_1\Delta x = q\Delta x\Delta y \tag{16}$$

However, due to the approximation that the grid points fall on the IB when calculating the mass source/sink term, Equations (15) and (16) obviously yield $v_2 = 0$, regardless of the true value of v_2. Thus, a more accurate expression for the mass source/sink term is needed.

Let us consider the general case shown in Figure 2(b). Besides the velocity components defined at the centre of the cell face, additional face-centred velocities u_{1c} and u_{2c} for the virtual cell ($\square AEFB$) are also displayed. The volume flux for the virtual cell ($\square AEFB$) is expressed as

$$v_2\Delta x + u_{2c}\beta_2\Delta y - u_{1c}(1+\beta_1)\Delta y = 0 \tag{17}$$

where

$$\beta_i = 1/2 - d_i/\Delta y \quad (i = 1, 2) \tag{18}$$

represents the dimensionless distance and d_i is the distance from the forcing point to the boundary point intersected by the grid line. Note that $\beta_1 < 0$ and $\beta_2 > 0$ according to Figure 2(b). The values of u_{1c} and u_{2c} are obtained by linear interpolation of the velocities at neighbouring points

$$u_{1c} = \left(1 + \frac{\beta_1}{2}\right) u_1 - \frac{\beta_1}{2} u_{1n} \tag{19}$$

$$u_{2c} = \frac{\beta_2}{1 + 2\beta_2} u_{2n} \tag{20}$$

where u_{1n} and u_{2n} denote the velocities of the neighbouring points (in the fluid) of u_1 and u_2, respectively. For simplicity, Equations (17)–(20) are given only for a uniform grid. However, it is straightforward to extend the formalism to the case of a non-uniform grid.

From Equations (16)–(20), as shown in Figure 2(b), q is expressed as

$$q = \frac{u_2}{\Delta x} - \frac{v_1}{\Delta y} + \frac{\beta_1}{\Delta x} u'_{1c} - \frac{\beta_2}{\Delta x} u'_{2c} \tag{21}$$

where u'_{1c} and u'_{2c} denote the velocities at the middle of EC and BF, respectively,

$$u'_{1c} = \left(\frac{3}{2} + \frac{\beta_1}{2}\right) u_1 - \left(\frac{1}{2} + \frac{\beta_1}{2}\right) u_{1n} \tag{22}$$

$$u'_{2c} = \frac{\beta_2}{1 + 2\beta_2} u_{2n} \tag{23}$$

Note that the first two terms on the right-hand side of Equation (21) are the same as in the method of Kim *et al*. [4], whereas the other two terms ($\beta_1 u'_{1c}/\Delta x - \beta_2 u'_{2c}/\Delta x$) are added in the present study. These two additional terms denote the volume flux through small segments of the cell faces crossed by the IB (EC and BF, respectively). Finally, the following general expression for q can be formulated

$$q = \frac{1}{\Delta V} \sum_i (\omega \mathbf{u} \cdot \mathbf{n} - \beta_i \mathbf{u}'_c \cdot \mathbf{n}) \Delta S_i \tag{24}$$

DOI: 10.1002/fld

where

$$\mathbf{u}_c' = \begin{cases} \dfrac{\beta_i}{1+2\beta_i}\mathbf{u}_n & \text{if } \beta_i > 0 \\ \left(\dfrac{3}{2}+\dfrac{\beta_i}{2}\right)\mathbf{u} - \left(\dfrac{1}{2}+\dfrac{\beta_i}{2}\right)\mathbf{u}_n & \text{if } \beta_i < 0 \end{cases} \tag{25}$$

In Equation (24), $\Delta V, \Delta S_i, \mathbf{u}, \mathbf{n}$ and ω are defined as in Equation (14), β_i is defined in Equation (18) and $\mathbf{u}_c'$ is obtained from Equation (25), where $\mathbf{u}_n$ denotes the velocity at the neighbouring point of $\mathbf{u}$ in the fluid region. This formula is valid for a virtual cell of arbitrary shape. The expression $(\omega\mathbf{u}\cdot\mathbf{n} - \beta_i\mathbf{u}_c'\cdot\mathbf{n})\Delta S_i$ in Equation (24) corresponds to the volume flux through the solid parts of the cell faces, such as EC and FD, where the velocity at the centre of the segment is obtained from linear interpolation. Because each pair of neighbouring cells share a cell face, the volume flux through the solid parts of the cell face will have the same magnitude but opposite sign for the two neighbouring cells. Therefore, the integration of the volume flux due to the mass source/sink over the whole computational domain is equal to zero and the global mass conservation is satisfied. We found that most of the time consuming part in the Navier–Stokes solver is the pressure Poisson equation. The cost for calculating the mass source/sink term is almost negligible.

4. NUMERICAL EXAMPLES

4.1. Decaying vortex problem

To verify the spatial accuracy of the present method, the decaying vortex problem is chosen because it is an unsteady problem with an analytical solution:

$$\begin{aligned} u(x, y, t) &= -\cos\pi x \sin\pi y \, \mathrm{e}^{-2\pi^2 t/Re} \\ v(x, y, t) &= \sin\pi x \cos\pi y \, \mathrm{e}^{-2\pi^2 t/Re} \\ p(x, y, t) &= -\tfrac{1}{4}(\cos 2\pi x + \cos 2\pi y)\, \mathrm{e}^{-4\pi^2 t/Re} \end{aligned} \tag{26}$$

Figure 3(a) shows the contour of u at $t=0$ obtained from the above equation. The computational domain is $-1.5 \leqslant x, y \leqslant 1.5$ and the IB is located at $x = \pm 1$ and $y = \pm 1$ (see Figure 3(b)). The grid is kept uniform inside the IB and is stretched slightly outside it. Six different grid numbers are tested. In all cases, the grid size inside the IB is intentionally set to give a dimensionless distance of $\beta = 0.33$ (see Equation (18)), facilitating comparison. The Reynolds number based on the maximum velocity and vortex size is set to 30, and the initial and boundary conditions are given by the analytical solution (i.e. Equation (26)). The flow at $t=0$ is integrated in time using a time step of $\Delta t = 0.0001$ and the simulated flow field at $t=0.3$ is compared with the analytical solution.

Simulations of three cases are considered: without the mass source/sink term, with the mass source/sink term of Kim *et al.* [4] (referred to as the KKC method hereafter), and with the present mass source/sink term. Figure 4 shows the maximum error of u within the IB for the three cases at each grid size. We can see that due to the interpolation scheme adopted here for the momentum forcing, the accuracy is second order for all the three cases. When the present mass source/sink

DOI: 10.1002/fld

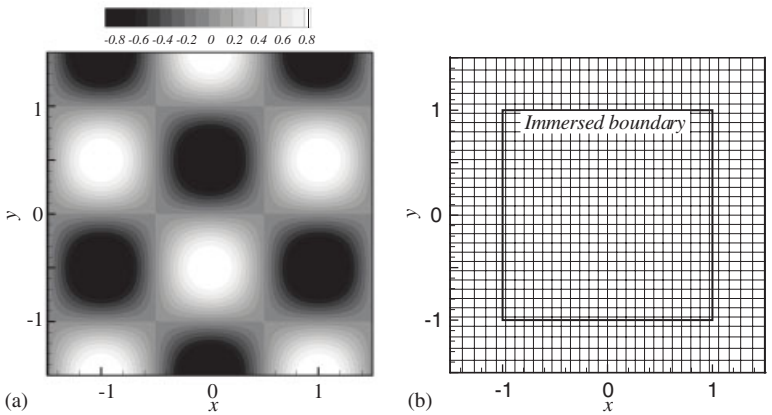

Figure 3. (a) u contour of the analytical solution at $t = 0$ for the decaying vortex problem; and (b) grid system and IB configuration for simulation of the decaying vortex problem.

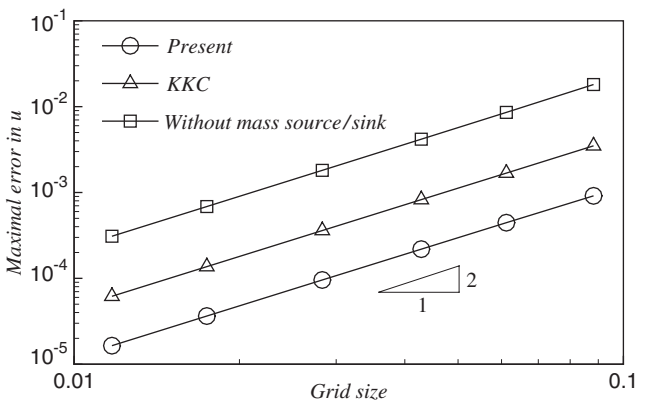

Figure 4. Maximum errors of u within the immersed boundary at $t = 0$.

term is used, however, the errors are much smaller than those obtained using the KKC method, which are in turn much smaller than those obtained when no mass source/sink term is used. This shows that introducing the mass source/sink term improves the quality of the solution significantly and the present method is more accurate than the KKC method.

4.2. Uniform flow past a circular cylinder

As our second example, we consider the flow past a circular cylinder immersed in an unbounded uniform flow, a benchmark problem for numerical validation. A computational domain of $50D \times 30D$ is discretized using two sets of grids systems $257(40) \times 151(40)$ and $513(100) \times 451(100)$ (referred to as $G1$ and $G2$, respectively), where the grid number within the cylinder diameter is given in the parentheses. The grid is uniformly distributed inside the cylinder and stretched outside the

DOI: 10.1002/fld

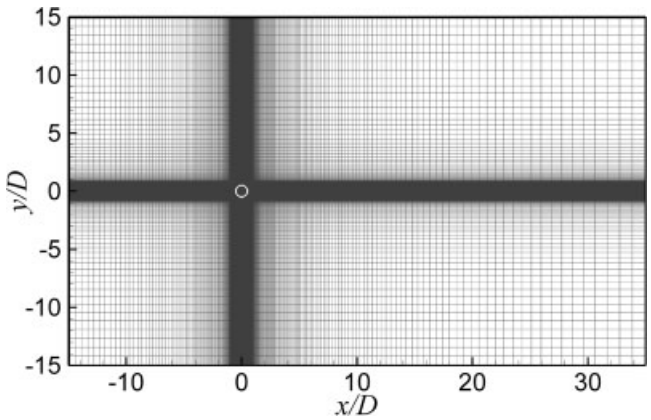

Figure 5. Computational domain and grid system for simulation of uniform flow past a circular cylinder. Grid number is 257×151 in this figure and the cylinder centre is located at the origin.

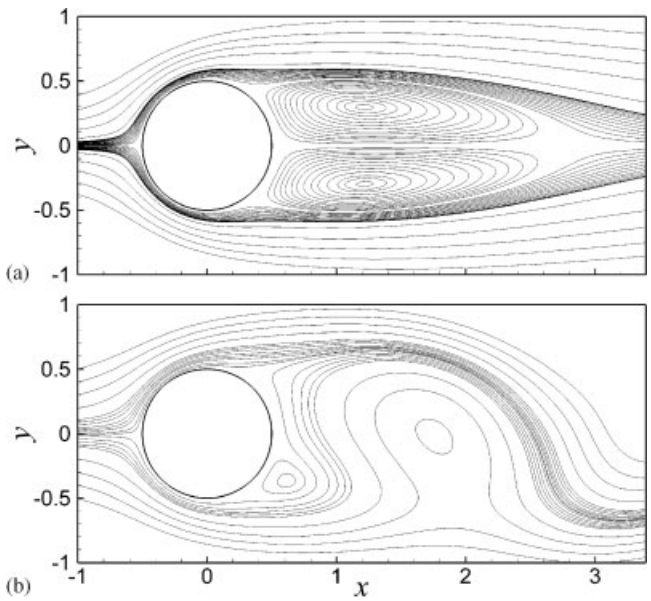

Figure 6. Instantaneous streamlines of uniform flow past a circular cylinder: (a) $Re = 40$; and (b) $Re = 100$.

cylinder, as shown in Figure 5. The flow is coming from left to right of the computational domain. A Dirichlet boundary condition ($u = U_\infty$, $v = 0$) is used at the inflow and far-field boundaries, and a convective boundary condition is used at the outflow.

Simulations are carried out at Reynolds numbers of $Re = 40$ and 100, where $Re = U_\infty D/\nu$ with U_∞ the free-stream velocity and D the cylinder diameter. Figure 6 shows the instantaneous streamlines in the near wake of the cylinder using the grid system $G2$. We can see that a pair of

DOI: 10.1002/fld

Table I. Comparison of recirculation length, drag coefficient, lift coefficient and Strouhal number with those obtained in previous studies.

	$Re=40$		$Re=100$		
	L_w/D	C_D	C_D	C'_L	St
$G1$	2.28	1.57	1.37	0.33	0.167
$G2$	2.27	1.56	1.36	0.33	0.167
Kim *et al.* [4]	—	1.51	1.33	0.32	0.165
Tseng and Ferziger [5]	2.21	1.53	1.42	—	0.164
Ye *et al.* [6]	2.27	1.52	—	—	—
Liu *et al.* [15]	—	—	1.35	0.34	0.165
Linnick and Fasel [16]	2.28	1.54	1.34	0.33	0.166

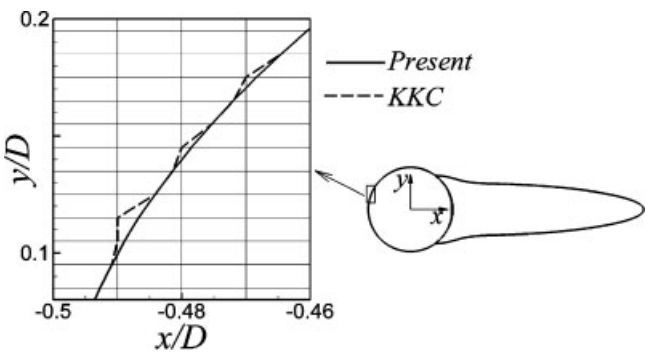

Figure 7. Isoline of $u=0$ along the cylinder surface.

symmetric vortices are attached behind the cylinder at $Re=40$ and the vortex is shedding from the cylinder at $Re=100$. Table I presents a comparison of the present results with previous studies. Excellent agreement is found among the scatter of the data for both the coarse and fine grid systems.

Figure 7 shows the isoline of $u=0$ at $Re=40$ using the grid system $G2$ to check the no-slip condition along the IB. In Figure 7, the result of the KKC method shows deviations at various grid points near the IB, as would be expected for a step-wise approximation, which is corrected by using the present mass source/sink term.

The pressure coefficient along the cylinder surface at $Re=40$ and 100 is shown in Figure 8 for both the coarse grid system $G1$ (Figure 8(a)) and the fine grid system $G2$ (Figure 8(b)). The results obtained with the body-fitted grid [17] are also shown for comparison. The pressure at the cylinder surface is obtained from the pressure at the cell centre (in the fluid region) nearest to the IB by assuming that the wall-normal derivation of pressure at the surface is zero. In Figure 8(a), the results of the KKC method agree with those obtained with body-fitted grid in general and show a little deviation at some points. However, in Figure 8(b), the results of the KKC method exhibit obvious oscillations along the surface, especially near the stagnation point and the separation point, which can be seen more clearly in the inset. These oscillations are due to the inaccurate mass source/sink term. These errors are corrected by the present method. We can see that the present results are smooth and agree well with those obtained with the body-fitted grid.

DOI: 10.1002/fld

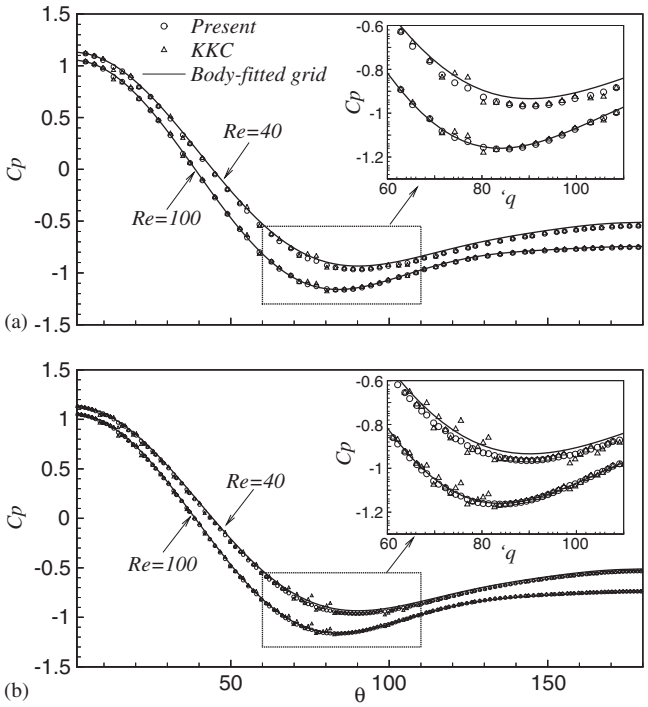

Figure 8. Pressure coefficient along the cylinder surface obtained using: (a) the grid system $G1$; and (b) the grid system $G2$.

In order to better understand the advantage of the present method over the KKC method, the mass source/sink obtained by both methods is examined. Figure 9 shows the mass source/sink near the IB at $Re = 40$ for both grid systems $G1$ and $G2$. Using the KKC method, see Figures 9(a) and (c), the resulted mass source/sink is distributed across the IB at various points and becomes more serious with the fine grid system. The mass source/sink outside the IB is non-physical and will affect the flow field. It is considered to be the reason of pressure oscillation near the IB. This situation is improved by the present method, see Figures 9(b) and (d). We can see the mass source is restrained well inside of the IB, especially with the fine grid system. Thus it is confirmed that the present improvement of the mass source/sink have an effective influence on the numerical solution, especially for the fine grid system.

5. CONCLUSIONS

In the present study, we developed an improved IB method using a mass source/sink as well as momentum forcing for simulating flows over or inside complex geometries. The proposed IB method is based on the Navier–Stokes solver adopting the fractional step method and a staggered Cartesian grid system. The significance of mass conservation of the virtual cell on the quality of the solution near the IB is discussed, and a more accurate formulation of the mass source/sink term

DOI: 10.1002/fld

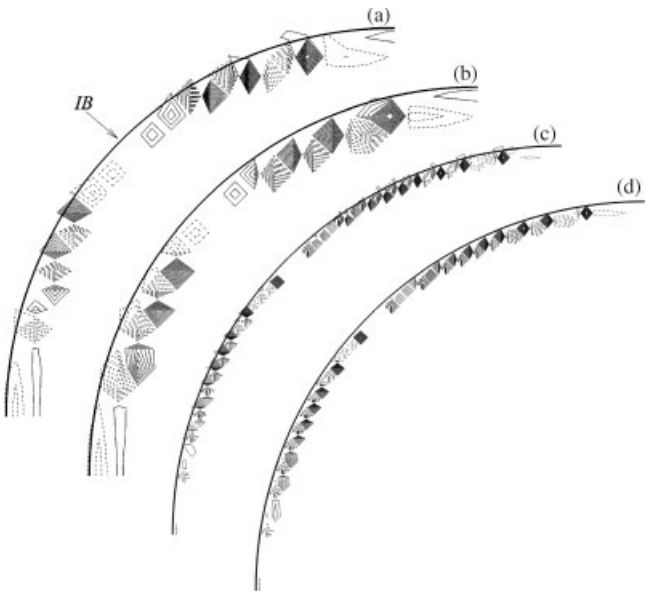

Figure 9. Contour of the mass source/sink near the IB within $0 \leqslant \theta \leqslant 90°$ at $Re = 40$ obtained by (a, c) the KKC method and (b, d) the present method with (a, b) the grid system $G1$ and (c, d) the grid system $G2$. Solid lines represent positive values ranged from 1 to 12 with increment 1 and dashed lines represent negative values ranged from -1 to -12 with increment -1.

is derived by considering mass conservation of the virtual cells. By applying the proposed method to two flow problems (the decaying vortex problem and uniform flow past a circular cylinder), the results indicate that the present method reduces the error of the numerical solution significantly; specifically, the boundary conditions at the IB are better enforced and the oscillations of the pressure field near the IB observed using the previous method are eliminated.

REFERENCES

1. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
2. Lai MC, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics* 2000; **160**:705–719.
3. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics* 2000; **161**:35–60.
4. Kim J, Kim D, Choi H. An immersed boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics* 2001; **171**:132–150.
5. Tseng YH, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
6. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**:209–240.
7. Kang S, Iccarino G, Moin P. Accurate and efficient immersed-boundary interpolations for viscous flows. *Annual Research Briefs*, *Center for Turbulence Research*, NASA Ames/Stanford University, 2004; 377–390.
8. Bottino DC. Modelling viscoelastic networks and cell deformation in the context of the immersed boundary method. *Journal of Computational Physics* 1998; **147**:86–133.
9. Peskin CS, McQueen DM. A three-dimensional computational method for blood flow in the heart. I. Immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics* 1989; **87**:372–405.

DOI: 10.1002/fld

10. Peskin CS, McQueen DM. A general method for the computational simulation of biological systems interacting with fluids. *SEB Symposium on Biological Fluid Dynamics*, July 1994.
11. Kim K, Baek SJ, Sung HJ. An implicit velocity decoupling procedure for incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2002; **38**:125–138.
12. Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics* 1993; **105**:354–366.
13. Saiki EM, Biringen S. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computational Physics* 1996; **123**:450–465.
14. Majumdar S, Iaccarino G, Durbin P. RANS solvers with adaptive structured boundary non-conforming grids. *Annual Research Briefs*, Center for Turbulence Research, NASA Ames/Stanford University, 2001; 353–366.
15. Liu C, Zheng X, Sung CH. Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational Physics* 1998; **139**:35–57.
16. Linnick MN, Fasel HF. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics* 2005; **204**:157–192.
17. Baek SJ, Sung HJ. Numerical simulation of the flow behind a rotary oscillating circular cylinder. *Physics of Fluids* 1998; **10**(4):869–876.

DOI: 10.1002/fld