INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **60**:1378–1401
Published online 6 November 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1942

Computation of incompressible flows with immersed bodies by a simple ghost cell method

Dartzi Pan*,† and Tzung-Tza Shen

Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan, Taiwan, ROC

SUMMARY

The incompressible Navier–Stokes equations are solved by an implicit pressure correction method on Cartesian meshes with local refinement. A simple and stable ghost cell method is developed to treat the boundary condition for the immersed bodies in the flow field. Multigrid methods are developed for both velocity and pressure correction to enhance the stability and convergence of the solution process. It is shown that the spatial accuracy of the method is second order in L_2 norm for both velocity and pressure. Various steady and unsteady flows over a 2D circular cylinder and a 3D sphere are computed to validate the present method. The capability of the present method to treat a moving body is also demonstrated. Copyright © 2008 John Wiley & Sons, Ltd.

Received 25 May 2008; Revised 2 September 2008; Accepted 3 September 2008

KEY WORDS: ghost cell method; immersed boundary; Cartesian mesh; local refinement; incompressible flow; multigrid method

1. INTRODUCTION

Recently numerical methods for solving incompressible flows on fixed Cartesian meshes have gained much popularity for their relative ease in treating complex immersed bodies [1–12]. In these methods the body surface is not aligned with any grid line or surface, and various treatments are developed to simulate the effects of the boundary condition required on the body surface. The immersed boundary method [1] introduces a momentum-forcing term to the Navier–Stokes equations using Dirac delta function. This momentum-forcing term represents the surface forces exerted on the fluid by the immersed elastic or solid bodies. In its numerical implementation, the

*Correspondence to: Dartzi Pan, Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan, Taiwan, ROC.
†E-mail: dpan@mail.ncku.edu.tw

Contract/grant sponsor: National Science Council of Republic of China (Taiwan); contract/grant numbers: NSC95-2221-E006-254, NSC96-2221-E006-179-MY2

Dirac delta function is replaced by a smoother numerical delta function to distribute the momentum forcing over several grid cells across the immersed boundary. In virtual boundary method [2, 3], the momentum-forcing term is obtained using a feedback control loop to asymptotically enforce the no-slip boundary condition at the immersed boundary surface. Both the immersed boundary method and the virtual boundary method have been successfully implemented in various flow problems. However, both methods suffer from numerical stiffness problem either due to the modification of constitutive law to account for solid bodies or due to the necessary tuning of the control gain for the feedback control logic.

The direct forcing immersed boundary method [4, 5] constructs the momentum-forcing term in the framework of discretized equations such that the appropriate boundary condition is enforced at the immersed boundary in the discretized system. This approach is practically equivalent to performing a numerical reconstruction of the flow field near the immersed boundary such that the no-slip boundary condition is implicitly satisfied at the immersed boundary. The variable value of the cell receiving the direct forcing is determined by the chosen reconstruction scheme without actually solving the flow equations. Thus, the differences in the choice of the reconstruction scheme and the grid cells receiving the forcing distinguish various direct forcing methods. The forcing is normally applied to grid cells or nodes external to the immersed body surface [4, 5]. It can also be applied to grid cells or nodes inside the immersed body surface [6]. The latter approach is sometimes called the ghost cell method, in which the variable value of the ghost cell inside the body is obtained by an extrapolation from the flow solutions external to the body.

The fictitious domain method [7] is a finite element method that enforces the fluid–body interface boundary condition in a weak form using Lagrange multipliers, which can be viewed as a term similar to the momentum forcing added to the Navier–Stokes equations in the immersed boundary method. In the Cartesian cut-cell method [8], the flow variables for the cut cells are solved based on the actual shape of the merged cut cells. Thus, this method requires extensive work to account for various shapes of the merged cut cells. Another way [12] to handle cut cells is to use one-sided unequal-spaced finite difference operators for those irregular stencil nodes in the vicinity of the body surface. In volume of body (VOB) method [9, 10], the domain inside the solid body is viewed as being occupied by the same fluid as outside with a prescribed divergence-free velocity field. In this view a fluid–body interface is similar to a fluid–fluid interface commonly encountered in the volume of fluid (VOF) method for the two-fluid flow problems. Thus, a VOB function analogous to the VOF function can be used to identify and track the presence of the immersed body. In a sense, the VOB approach is equivalent to applying the direct forcing to all cells inside the immersed body. Similarly, it is also feasible to view the solid body as being made of a material distinct from the surrounding fluid and being tracked by a color function [11].

In this paper the concept of ghost cell is adopted to treat the immersed boundary on Cartesian meshes. To enhance the accuracy around the immersed boundary, the local refinement capability on Cartesian meshes is also developed in this work. As mentioned in Reference [6], both linear and higher-order reconstruction schemes for ghost cells can yield overall second-order accurate solutions. However, high-order reconstruction schemes require more stencil points and are more likely to introduce spurious oscillations. Thus, at least for low and moderate Reynolds number flows, a stable first-order reconstruction is preferred in this work as long as the flow solution can be spatially second-order accurate. There are two common problems involved in a general first-order reconstruction model. First, there is always an uncertainty of choosing the most appropriate set of stencil points. In Reference [6], one boundary node and two fluid nodes nearest to the ghost node are chosen to be the stencil for a linear reconstruction. This choice seems reasonable, but it may lead to

DOI: 10.1002/fld

the second problem that, when a chosen fluid node is too close to the immersed boundary, large and negative weighting coefficient may result in the reconstruction matrix. In this case the extrapolated value for the ghost node, though algebraically correct, may introduce numerical instability to the flow solution. In Reference [6], when this troubled situation is identified, either the boundary line segment is modified to include the troubled fluid node as a new boundary node or a different linear extrapolation scheme based on the image point of the ghost node symmetric to the immersed boundary is used. All these remedies require local modifications of the reconstruction stencil.

In this work, the concept of image point is adopted and modified to construct a simple and stable linear reconstruction scheme. There is no ambiguity in choosing the stencil point, and the extrapolation to the ghost cell will not introduce numerical instability. It will be shown that the present scheme is second-order accurate in space. As validation examples, various steady and unsteady flows over a 2D circular cylinder and a 3D sphere are computed and compared with the published data. To demonstrate the capability of the present method to treat moving bodies, the 2D flow over an impulsively started moving cylinder is computed and discussed.

2. IMPLICIT PRESSURE CORRECTION METHOD

The integral form of incompressible Navier–Stokes equations can be written as

$$\begin{aligned} &\oint_{\mathrm{CS}} \vec{v}\cdot \mathrm{d}\vec{S}=0 \\ \frac{\partial}{\partial t}\int_{\mathrm{CV}} \vec{v}\,\mathrm{d}V+\oint_{\mathrm{CS}} \vec{v}\vec{v}\cdot\mathrm{d}\vec{S}-\oint_{\mathrm{CS}} \frac{1}{Re}\vec{\nabla}\vec{v}\cdot\mathrm{d}\vec{S}+\oint_{\mathrm{CS}} &P\,\mathrm{d}\vec{S}=0 \end{aligned} \tag{1}$$

where $\vec{v}$ and P are Cartesian velocity and pressure, Re is Reynolds number, CV is the control volume under consideration, CS is the boundary surface of CV, $\mathrm{d}\vec{S}$ is a differential surface area vector of CS pointing outward. All variables in Equation (1) are properly normalized by suitable reference values. Applying the divergence theorem, adapting the backward time differencing scheme and keeping the pressure fixed at the current time level n the discretized momentum equation for a finite volume cell is

$$\left(\frac{c_1\vec{v}^*-c_2\vec{v}^n+c_3\vec{v}^{n-1}}{\Delta t}+\vec{R}^*_{\mathrm{conv}}-\vec{R}^*_{\mathrm{vis}}+\vec{R}^n_P+\vec{F}_{\mathrm{dir}}\right)\Delta V=0 \tag{2}$$

where the variables $\vec{v}$ and P are stored at the cell center; Δt is the time increment; ΔV is the cell volume; the superscript '*' indicates an intermediate state between time level n and $n+1$; $\vec{R}_{\mathrm{conv}}$, $\vec{R}_P$ and $\vec{R}_{\mathrm{vis}}$ correspond to the surface integral of convection flux, pressure flux and viscous flux, respectively; $\vec{F}_{\mathrm{dir}}$ is a direct forcing term added to model the presence of immersed bodies. Note that $\vec{F}_{\mathrm{dir}}$ is null for flows without immersed body. The constants are $c_1=1.5, c_2=2$ and $c_3=0.5$ for the second-order accurate backward differencing scheme, and $c_1=1, c_2=1$ and $c_3=0$ for the first-order Euler implicit scheme. The intermediate velocity $\vec{v}^*$ generally does not satisfy

DOI: 10.1002/fld

the divergence-free condition. The velocity $\vec{v}^*$ and pressure P^n are corrected by the following correction step:

$$\begin{aligned}\vec{v}^{n+1} &= \vec{v}^* - \Delta t \vec{\nabla}\phi^n \\ P^{n+1} &= P^n + \phi^n\end{aligned} \tag{3}$$

where ϕ^n is the pressure correction. By requiring that $\vec{v}^{n+1}$ be divergence-free, we obtain the Poisson equation for the pressure correction:

$$\nabla^2 \phi^n = \frac{\vec{\nabla}\cdot\vec{v}^*}{\Delta t} \tag{4}$$

Equations (2), (3) and (4) constitute the implicit fractional step pressure correction method used in this work.

3. FINITE VOLUME DISCRETIZATION

A MUSCL-type finite volume method developed by Pan [9, 10] is used to discretize the momentum equations on a cell-centered unstructured Cartesian grid system with local refinement. The flow state at a cell vertex j is obtained by weighting the surrounding cell-center values as:

$$Q_{\text{vert},j} = \frac{\sum_{i\in k(j)} \omega_i Q_i}{\sum_{i\in k(j)} \omega_i}, \quad \omega_i = \frac{1}{|\vec{r}_{i,j}|} \tag{5}$$

where $Q_{\text{vert},j}$ indicates the variable value of vertex j; $k(j)$ is the set of cell centers surrounding vertex j; $|\vec{r}_{i,j}|$ is the distance from a cell center i to the vertex j. The variable gradient in a particular cell is obtained by a simple average of the central differences of the vertex values. The left and right states of a cell face f are linearly reconstructed from the neighboring center values using the estimated cell gradient as

$$Q_f^{L,R} = Q_{\text{neig}}^{L,R} + \Delta\vec{r}\cdot\vec{\nabla} Q_{\text{neig}}^{L,R} \tag{6}$$

where $Q_f^{L,R}$ indicates respectively, the left (L) and right (R) face values at the cell face f; $Q_{\text{neig}}^{L,R}$ indicates the left and right neighboring center of the cell face f; $\vec{\nabla} Q_{\text{neig}}^{L,R}$ indicates the cell gradient of the left and right neighbor of the cell face f; $\Delta\vec{r}$ is the distance from the neighboring cell center to the center of the cell face f. The mean variable value Q_f^M at the cell face f is the simple average of the left and the right value of the cell face f:

$$Q_f^M = 0.5(Q_f^L + Q_f^R) \tag{7}$$

The convective flux $\vec{R}_{\text{conv}}$ is computed as

$$\vec{R}_{\text{conv}} = \frac{1}{\Delta V}\sum_{\text{CS}} \begin{bmatrix} u \\ v \end{bmatrix}_{L/R} \vec{v}_f^M \cdot \hat{n}\Delta S \tag{8}$$

where u and v are Cartesian velocity components; $\vec{v}_f^M$ is the mean velocity at the center of cell face f; ΔS is the cell-face area; $\hat{n}$ is the outward unit surface normal; and the summation operator is done over all surfaces of the cell. The subscript 'L/R' represents the velocity upwinding at the cell face f as

$$(\bullet)_{L/R}=\begin{cases}(\bullet)^L & \text{if } \vec{v}_f^M\cdot\hat{n}>0\\ (\bullet)^R & \text{if } \vec{v}_f^M\cdot\hat{n}\leqslant 0\end{cases} \tag{9}$$

where the superscripts 'L' and 'R' indicate respectively, the left and right face values of cell face f. The pressure flux is computed as

$$\vec{R}_P=\frac{1}{\Delta V}\sum_{\text{CS}}P_f^M\Delta S\begin{bmatrix}n_x\\ n_y\end{bmatrix} \tag{10}$$

where n_x and n_y are the components of $\hat{n}$ in x and y directions, respectively; P_f^M is the mean pressure at the center of cell face f. The viscous flux $\vec{R}_{\text{vis}}$ is computed as

$$\vec{R}_{\text{vis}}=\frac{1}{Re\Delta V}\sum_{\text{CS}}\begin{bmatrix}\vec{\nabla}u\\ \vec{\nabla}v\end{bmatrix}\cdot\hat{n}\Delta S \tag{11}$$

The velocity gradient at and normal to a cell face f is obtained by a central difference of the two neighboring center values of the face f. On a regular Cartesian grid, Equations (5)–(11) are spatially second-order accurate flux calculations.

To compute the divergence of velocity, a face-normal velocity U_f is defined independently for each cell face, which is different from the cell-center velocity. Specifically, for a cell face f with unit normal $\hat{n}$, the intermediate state of the face-normal velocity is defined as:

$$U_f^*=\vec{v}_f^M\cdot\hat{n}-\text{cdisp}\frac{\Delta t}{\Delta\ell}(P_f^R-P_f^L)\hat{e}\cdot\hat{n} \tag{12}$$

where $\Delta\ell$ is the distance between the right and the left neighboring cell centers; $\hat{e}$ is the Cartesian unit vector normal to the face; cdisp is an input constant for the pressure dissipation. In this work, cdisp$\leqslant$ 0.3 is used. The divergence of the intermediate velocity for a particular cell is computed as

$$\vec{\nabla}\cdot\vec{v}^*=\frac{1}{\Delta V}\sum_{\text{CS}}(U_f^*\Delta S)=\frac{1}{\Delta V}\sum_{\text{CS}}\vec{v}_f^M\cdot\hat{n}\Delta S-\frac{\text{cdisp}\Delta t}{\Delta V}\sum_{\text{CS}}\frac{P_f^R-P_f^L}{\Delta\ell}\hat{e}\cdot\hat{n}\Delta S \tag{13}$$

The last term on the right-hand side of Equation (13) constitutes a background dissipation term based on the pressure field. It has similar effects as the widely used momentum interpolation method suggested by Rhie and Chow [13]. On a regular Cartesian grid it can be shown that this dissipation is proportional to

$$\left(a\Delta x^2\frac{\partial^4 P}{\partial x^4}+b\Delta y^2\frac{\partial^4 P}{\partial y^4}\right)$$

where a and b are some constants. Note that in Equation (12) the dissipation term is written in a simple form suitable for unstructured meshes.

Similar to Equation (3), the intermediate face-normal velocity is corrected by

$$U_f^{n+1} = U_f^* - \Delta t (\vec{\nabla}\phi^n)_f \cdot \hat{n} \tag{14}$$

where the gradient of pressure correction is computed at the cell-face center using the neighboring center values. By demanding that U_f^{n+1} be divergence-free, a discrete Poisson equation for the pressure correction is constructed as

$$\frac{1}{\Delta V}\sum_{\mathrm{CS}} (\vec{\nabla}\phi^n)_f \cdot \hat{n}\Delta S - \frac{1}{\Delta t \Delta V}\sum_{\mathrm{CS}} U_f^* \Delta S = 0 \tag{15}$$

Equation (15) is a compact discretization of Equation (4). It is used to compute the pressure correction ϕ^n at the cell centers, which in turn determines the cell-center pressure and velocity at the new time level $n+1$ in Equation (3). When the velocity at the domain boundary is known and fixed, the correction of velocity at the domain boundary should be zero. This is equivalent to a Neumann boundary condition for Equation (15), requiring that the normal gradient of ϕ^n at the domain boundary be zero.

4. A SIMPLE GHOST CELL METHOD

The immersed surface boundary is specified by a set of linear line segments whose nodes lie on the body surface, as shown in Figure 1. It is assumed that the position and the velocity $\vec{v}_{\mathrm{B}}$ of the immersed body at any time are known from some appropriate governing equations or boundary conditions. The body velocity $\vec{v}_{\mathrm{B}}$ is assumed to be divergence-free, and the pressure field inside the body obeys the same governing equation as the pressure field outside. Under this assumption, Equation (4) can be applied to the entire computational domain including the body interior.

In Figure 1, all cell centers external to the immersed body are categorized as fluid cells. All cell centers inside the body are body cells. Those body cells neighboring to at least one fluid cell are defined as ghost cells. It is clear that the immersed boundary passes through the space between the ghost cells and their neighboring fluid cells. The variable values at ghost cells are to be determined in such a way that the proper boundary condition is satisfied implicitly at the immersed boundary surface. Once the ghost cell values are determined, they are specified to the particular ghost cell center by the direct forcing method.

In this work, a simple and stable linear reconstruction model is developed to determine the ghost cell values. For each ghost cell center, its projection point on the boundary surface, as shown in Figure 1, is the point where the no-slip condition is to be enforced implicitly. The velocity at the projection point $\vec{v}_{\mathrm{proj}}$ is assumed known because it is on the body surface. Along the surface normal direction passing through the ghost cell center and its projection point, we define the image point of the ghost cell center to be the point at a distance δ away from the body surface, as shown in Figure 1. The choice of δ is artificial, and in this work we take $\delta = \sqrt{2}\Delta x_{\min}$ in 2D problems where $\Delta x_{\min}$ is the minimum Cartesian cell length in the vicinity of the boundary surface. This choice of δ, which is the diagonal length of the Cartesian cell, ensures that the image point is enclosed by four fluid cell centers forming a square region close to but untouched by the boundary surface. The velocity at the image point can be obtained easily by a bilinear interpolation using

DOI: 10.1002/fld

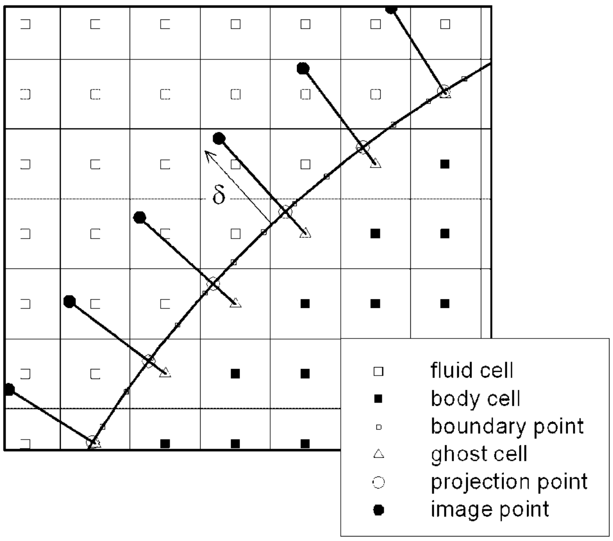

Figure 1. Various cell and node types, including fluid cell, body cell, ghost cell, projection point and image point, δ: distance from projection point to image point.

the velocity of the four enclosing fluid centers. The velocity gradient along the surface normal passing through the projection point can then be estimated by

$$\frac{\partial \vec{v}}{\partial n} = \frac{\vec{v}_{\text{image}} - \vec{v}_{\text{proj}}}{\delta} \tag{16}$$

where $\vec{v}_{\text{image}}$ is the velocity at the image point, $\hat{n}$ is the outward surface unit normal at the projection point. The velocity at the ghost cell center is then extrapolated by

$$\vec{v}_{\text{ghost}} = \vec{v}_{\text{proj}} - \frac{\partial \vec{v}}{\partial n} |\vec{r}_{\text{ghost}} - \vec{r}_{\text{proj}}| \tag{17}$$

where $\vec{r}_{\text{ghost}}$ and $\vec{r}_{\text{proj}}$ are the position vectors of the ghost cell center and the projection point, respectively. The ghost cell velocity $\vec{v}_{\text{ghost}}$ determined by Equations (16) and (17) implies that the no-slip wall condition is satisfied at the projection point.

The linear reconstruction model described above is quite simple, and can be extended to 3D problems in a straightforward manner. The choice of δ in 3D is $\delta = \sqrt{3}\Delta x_{\min}$, and the tri-linear interpolation scheme can be used to obtain $\vec{v}_{\text{image}}$ from the surrounding fluid cell centers. The simplicity of the model is attributed to the choice of a constant δ for all image points. If we choose a shorter δ, the number of fluid centers enclosing the image point may be less than four in 2D or less than eight in 3D, and the stencil for the reconstruction of $\vec{v}_{\text{image}}$ will vary. On the other hand, if we choose a longer δ, the estimated velocity gradient will not be a good representation of the true velocity gradient in the vicinity of the boundary surface. Furthermore, the current choice of δ ensures that $\delta > |\vec{r}_{\text{ghost}} - \vec{r}_{\text{proj}}|$ in Equations (16) and (17), and thus avoids the unstable situation when some stencil point of the reconstruction model gets too close to the boundary surface.

In the case of Neumann boundary condition with a given normal gradient $\partial q/\partial n$ for variable q, for example, the variable value at the ghost cell center, q_{ghost} can be extrapolated by

$$q_{\text{ghost}} = q_{\text{image}} - \frac{\partial q}{\partial n} |\vec{r}_{\text{ghost}} - \vec{r}_{\text{image}}| \tag{18}$$

where q_{image} and $\vec{r}_{\text{image}}$ are the variable value and the position vector of the image point, respectively. Equation (18) and some alternative formulations for Neumann boundary condition will be the subject of another paper.

Remember that in this work the variable values at cell vertices are used to compute the variable gradients in each Cartesian cell. Thus, for consistency, the vertices of ghost cells that are covered by the body surface, or ghost vertices, should also be treated by the same reconstruction described above. That is, the variable values at ghost vertices are reconstructed using their corresponding projection points and image points and Equations (16) and (17).

For body cells with known velocity $\vec{v}_{\text{B}}^{n+1}$, the direct momentum forcing $\vec{F}_{\text{dir}}$ in Equation (2) is

$$\vec{F}_{\text{dir}} = -\left(\frac{c_1 \vec{v}_{\text{B}}^{n+1} - c_2 \vec{v}^{n} + c_3 \vec{v}^{n-1}}{\Delta t} + \vec{R}_{\text{conv}}^{*} - \vec{R}_{\text{vis}}^{*} + \vec{R}_{P}^{n} \right) \tag{19}$$

resulting in the equation for body cells as

$$\frac{c_1(\vec{v}^{*} - \vec{v}_{\text{B}}^{n+1})}{\Delta t} \Delta V = 0 \tag{20}$$

For ghost cells, $\vec{v}_{\text{B}}^{n+1}$ is replaced by $\vec{v}_{\text{ghost}}$ in Equations (19) and (20). Note that Equation (20) is used to replace Equation (2) in numerical implementation without actually computing $\vec{F}_{\text{dir}}$. Because $\vec{v}_{\text{B}}^{n+1}$ is assumed known and divergence-free, the velocity correction step or Equation (3), will not affect for body cell velocity. As for ghost cells, the direct forcing is applied at every time step, thus the no-slip boundary condition on the immersed boundary is always satisfied implicitly during the computation of fluxes.

To compute the aerodynamic forces acting on the immersed body, the surface integral of pressure and viscous stress is needed:

$$\vec{f}_{\text{Body}} = \sum_{\text{surface}} \left(-P\Delta\vec{S} + \frac{1}{Re} \vec{\nabla}\vec{v} \cdot \Delta\vec{S} \right) \tag{21}$$

where the summation is performed on all boundary surface segments of the body. The pressure at the segment center is obtained by a bilinear interpolation based on the vertex values of the cell containing the segment center. The velocity gradient, on the other hand, is assumed constant in each cell. The velocity gradient at a surface segment center is then estimated by differencing the vertex values of the cell containing the segment center.

5. IMPLICIT TIME INTEGRATION

Applying Newton's method to Equation (2) and writing it in delta-law form, the implicit time integration equation can be written as

$$\begin{aligned}\left(\frac{c_1\Delta V}{\Delta t}+\Delta V\frac{\partial(\vec{R}^s_{\text{conv}}-\vec{R}^s_{\text{vis}})}{\partial\vec{v}^s}\right)(\vec{v}^{s+1}-\vec{v}^s)\\ =-\left(\frac{c_1\vec{v}^s-c_2\vec{v}^n+c_3\vec{v}^{n-1}}{\Delta t}+\vec{R}^s_{\text{conv}}-\vec{R}^s_{\text{vis}}+\vec{R}^n_P+\vec{F}_{\text{dir}}\right)\Delta V\\ =-\{\text{RHS}(\vec{v}^s)-\mathit{Source}\}=\text{Res}^s\end{aligned} \tag{22}$$

where the superscript 's' is the index for sub-iteration; Res^s is the residual vector of the unsteady Navier–Stokes equations; $-\text{RHS}(\vec{v}^s)$ contains all operators in the residual involving $\vec{v}^s$; *Source* represents all other terms in the residual involving the known states of P^n, $\vec{v}^n$, $\vec{v}^{n-1}$ and $\vec{v}^{n+1}_{\text{B}}$. Equation (22) can be further represented by

$$[\text{LHS}](\vec{v}^{s+1}-\vec{v}^s)=-\{\text{RHS}(\vec{v}^s)-\mathit{Source}\}=\text{Res}^s \tag{23}$$

where LHS is the Jacobian matrix of $\text{RHS}(\vec{v}^s)$. When the sub-iteration in s converges, the solution of Equation (22) is $\vec{v}^*=\vec{v}^{s+1}$, satisfying the time-accurate Equation (2). Note that the term *Source* is constant during the sub-iteration process. In operator LHS, simplifications are obtained by using first-order upwind scheme for the convection flux and the 3-point compact difference operator for the viscous flux. With these simplifications, the stencil of LHS extends only to the neighboring cell centers while the right-hand side operators are kept second-order accurate in space.

Splitting LHS into the sum of a diagonal part D, a lower triangular part L and an upper triangular part U, a two-step approximate LU factorization method (ALU) is used to invert Equation (23) as

$$\begin{aligned}\omega U\Delta\vec{v}^{s-1}+(D+\omega L)\Delta\vec{v}^{s^*}&=\omega(2-\omega)\text{Res}^s\\ \omega L\Delta\vec{v}^{s^*}+(D+\omega U)\Delta\vec{v}^{s}&=\omega(2-\omega)\text{Res}^s\\ \vec{v}^{s+1}&=\vec{v}^s+\Delta\vec{v}^s\end{aligned} \tag{24}$$

where the superscript 's^*' indicates the intermediate stage in sub-iteration s; ω is a relaxation parameter, and here we take $0.9\leqslant\omega\leqslant 1$. The initial conditions are $\vec{v}^s=\vec{v}^n$ and $\Delta\vec{v}^{s-1}=0$ for $s=1$. For further simplifications in D, L and U, the inviscid flux Jacobians are replaced by their eigenvalue matrices such that the solution process of Equation (24) requires only scalar multiplications and divisions.

6. LOCAL REFINEMENT AND MULTIGRID METHOD

To enhance accuracy and to reduce mesh size, a local refinement technique on Cartesian mesh is developed to refine the grid spacing around the immersed boundary surface. An unstructured Cartesian grid with local refinement is generated from an initial cell (parent cell) that covers the

DOI: 10.1002/fld

entire flow field. This initial cell is recursively subdivided into four identical child cells (eight child cells in 3D) until sufficient spatial resolution around the body surface is reached. Between any two neighboring cells, the difference in the level of refinement is restricted to one. The quadtree (octree in 3D) structure among the parent cells and their child cells provides a natural sequence of grid coarsening from fine to coarse grids.

To accelerate the convergence, a V-cycle multigrid method is also developed on the unstructured Cartesian meshes. For simplicity, we describe only a two-level multigrid cycle for solving Equation (23) on the fine grid. The equation to be solved on the coarse grid is

$$\begin{aligned} -\{\mathrm{RHS}^{\mathrm{C}}(Q^{\mathrm{C}})-\mathit{Source}^{\mathrm{C}}\}&=0 \\ \mathit{Source}^{\mathrm{C}}&=[\mathrm{RHS}^{\mathrm{C}}(J_{\mathrm{F}}^{\mathrm{C}}Q^{\mathrm{F}})+I_{\mathrm{F}}^{\mathrm{C}}\mathrm{Res}^{\mathrm{F}}] \end{aligned} \tag{25}$$

where Q is the unknown flow variable of the problem, which in this case is the velocity vector $\vec{v}$; the super and the subscripts ‘F’ and ‘C’ represent the fine grid and the coarse grid variable or operator, respectively. The injection operator $J_{\mathrm{F}}^{\mathrm{C}}$ for variable Q is defined as

$$J_{\mathrm{F}}^{\mathrm{C}}Q^{\mathrm{F}}=\frac{\sum_{\mathrm{Child}}Q^{\mathrm{F}}\Delta V^{\mathrm{F}}}{\sum_{\mathrm{Child}}\Delta V^{\mathrm{F}}} \tag{26}$$

where the summation is done over all child cells of a coarse cell. This injection operator conserves $Q\Delta V$ in the process. The injection operator $I_{\mathrm{F}}^{\mathrm{C}}$ for the residual vector Res is defined as

$$I_{\mathrm{F}}^{\mathrm{C}}\mathrm{Res}^{\mathrm{F}}=\sum_{\mathrm{Child}}\mathrm{Res}^{\mathrm{F}} \tag{27}$$

which conserves the surface integral of flux function. The operator $\mathrm{RHS}^{\mathrm{C}}$ in Equation (25) on the coarse grid is theoretically equivalent to the operator RHS in Equation (23) on the fine grid. However, simplification is achieved by using only first-order upwind difference for the convection fluxes on coarse grids of all levels. Note that the constant term *source* in Equation (23) on the fine grid is not present in the coarse grid equation. Equation (25) has exactly the same form as the right-hand side of Equation (23). Applying again the Newton’s method to Equation (25), an implicit integration equation similar to Equation (23) can be constructed on the coarse grid, and then solved for Q^{C} by the same approximate factorization method as described by Equation (24).

When Equation (25) has been solved for Q^{C}, the prolongation operator $J_{\mathrm{C}}^{\mathrm{F}}$ transfers the estimated correction from the coarse grid back to the finer grid by

$$Q^{\mathrm{F},s+1}=Q^{\mathrm{F}}+J_{\mathrm{C}}^{\mathrm{F}}(Q^{\mathrm{C}}-J_{\mathrm{F}}^{\mathrm{C}}Q^{\mathrm{F}}) \tag{28}$$

where $J_{\mathrm{F}}^{\mathrm{C}}Q^{\mathrm{F}}$ is the initial variable vector being injected from the fine grid. In this work a simple prolongation that assumes a constant distribution in the coarse cell is used. Specifically, a coarse cell transfers the calculated correction evenly to all its child (finer) cells. On the terminal grid, the body cells including the ghost cells are excluded from the prolongation procedure.

For the solution of pressure Poisson equation of Equation (15) on locally refined Cartesian meshes, a V-cycle multigrid method similar to the one described above is also developed. Applying the Newton’s method to Equation (15), an implicit relaxation method can be written as

$$\nabla^2(\phi^{s+1}-\phi^s)\Delta V=-\left\{\sum_{\mathrm{CS}}(\vec{\nabla}\phi^s)_f\cdot\hat{n}\Delta S-\frac{1}{\Delta t}\sum_{\mathrm{CS}}U_f^*\Delta S\right\}=\mathrm{Res}_\phi^s \tag{29}$$

DOI: 10.1002/fld

where the superscript 's' is the index for sub-iteration. When the sub-iteration in s converges, the solution is $\phi^n = \phi^{s+1}$, which satisfies the Poisson equation of Equation (15). The left-hand side operator is a compact Laplacian operator whose stencil extends only to the neighboring cell centers. Equation (29) can be further represented by

$$[\mathrm{LHS}_\phi](\phi^{s+1} - \phi^s) = -\{\mathrm{RHS}(\phi^s) - Source_\phi\} = \mathrm{Res}_\phi^s \tag{30}$$

where the term $Source_\phi$ is related to the divergence of velocity in each cell. This equation is in the same form as Equation (23). Hence, similar to the method described in the previous sections from Equation (24)–(28), an approximate factorization method and a V-cycle multigrid method are developed to solve Equation (30) on the whole computational domain, including body cells. The initial conditions are $\phi^s = 0$ and $\Delta\phi^{s-1/2} = 0$ for $s = 1$.

7. NUMERICAL EXAMPLES

7.1. Lid-driven cavity with a centered cylinder

To verify the order of spatial accuracy of the current scheme, a lid-driven cavity flow with an immersed cylinder at the center is computed. The upper wall of the cavity moves toward right at a uniform speed of one. The Reynolds number based on the cavity length, upper wall speed and the fluid viscosity is 1000. The cylinder diameter is 0.25. The boundary condition on the cylinder wall is handled by the current ghost cell method. A sequence of regular Cartesian grids, with mesh size ranging from 32 × 32 to 512 × 512, is used in the test.

Comparing with the no-slip boundary condition, the convergence of the computed $\vec{v}_{\mathrm{proj}}$ at the projection point on the body surface is tested first. Since the cylinder is stationary, the velocity $\vec{v}_{\mathrm{proj}}$ on the body surface should be zero. Here the velocity $\vec{v}_{\mathrm{proj}}$ is obtained by a bilinear interpolation using the ghost centers and the fluid centers enclosing the projection point. Table I shows the average interpolated velocity components and the L_2 norms of the error at all projection points. It can be seen that the order of accuracy of the $\vec{v}_{\mathrm{proj}}$ obtained by the current ghost cell method is about 1.8.

Next, the converged flow solution on the 512 × 512 mesh is used as the reference solution in the convergence test. The flow solutions at cell vertices inside a square of length 0.75 enclosing the cylinder, as indicated by the dashed square in Figure 2, are compared with the reference solution. In Figure 3, the L_2 and L_∞ norms of the error of the velocity field on various grids are plotted.

Table I. Lid-driven cavity flow at $Re = 1000$ with a centered cylinder of diameter 0.25, data of interpolated velocity at projection points on cylinder surface.

Grid size	No. of ghost cells and ghost vertices	Mean u_{proj}	Mean v_{proj}	$\|u_{\mathrm{proj}}\|_2$	$\|v_{\mathrm{proj}}\|_2$
32 × 32	20, 28	−2.816E−03	−7.794E−04	4.673E−03	4.491E−03
64 × 64	44, 60	−9.255E−04	−2.030E−04	1.513E−03	1.390E−03
128 × 128	88, 124	−2.844E−04	−7.034E−05	4.494E−04	4.248E−04
256 × 256	180, 268	−7.418E−05	−1.903E−05	1.200E−04	1.133E−04
512 × 512	360, 564	−1.877E−05	−5.166E−06	2.922E−05	2.766E−05
Average order p of Δx^p		1.81	1.81	1.83	1.84

DOI: 10.1002/fld

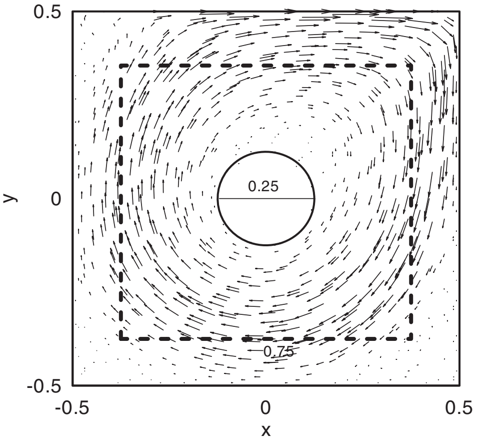

Figure 2. Velocity vectors of lid-driven cavity flow, $Re=1000$, the flow solution inside the dashed square is used for comparison.

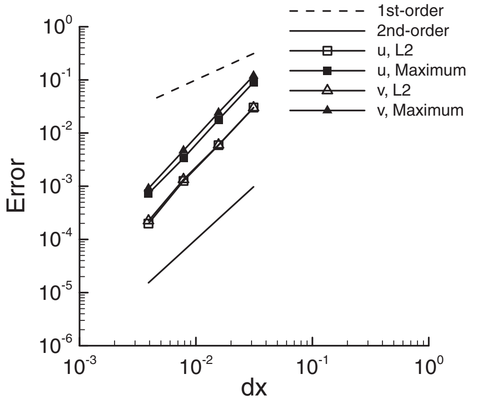

Figure 3. Norms of errors of velocity components compared with the results on the finest grid, unfilled symbols: L_2 norms, solid symbols: L_∞ norms.

Two curves showing first-order and second-order convergence are also displayed for convenience. It is obvious that the spatial accuracy of the current ghost cell method is second order in both norms for both velocity components. Figure 4 shows the L_2 and L_∞ norms of the error of pressure on various grids. The L_2 norm of pressure error is clearly second-order convergent. However, the maximum norm of pressure error is less than second order. This test has basically verified that the current ghost cell method is second-order accurate in L_2 and L_∞ norm of velocity, and in the L_2 norm of pressure for the problem considered here.

7.2. Flows over a 2D circular cylinder

The steady and unsteady flows over a circular cylinder of unit diameter at $Re=20, 40, 100, 200$ and 300 are computed on a Cartesian grid shown in Figure 5. The outer boundaries are 30 diameters

DOI: 10.1002/fld

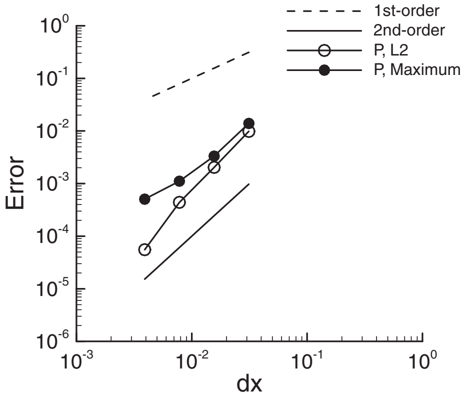

Figure 4. Norms of error of pressure compared with the results on the finest grid, unfilled symbol: L_2 norm, filled symbol: L_∞ norm.

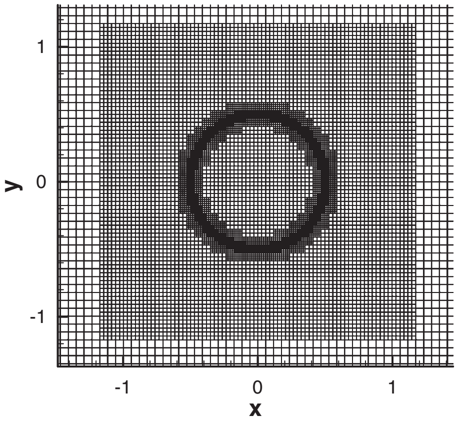

Figure 5. Locally refined Cartesian mesh for flow over a 2D stationary cylinder.

away from the cylinder, either parallel or perpendicular to the free stream. The grid is locally refined around the cylinder to have 384 ghost cells and 620 ghost vertices, while the total number of terminal cells is only 31 504. This high grid resolution around the cylinder is not really necessary for flows with a Reynolds number under 300. This grid is used here only to demonstrate the capability of local refinement, because it is almost impossible to obtain such a high grid resolution on a regular Cartesian grid of the same size over the same computational domain. The free-stream condition is set to the inflow boundary and the two side boundaries. The downstream boundary follows the upwind differenced equation of $(\partial\vec{v}/\partial t)+U_n(\partial\vec{v}/\partial x)=0$, where U_n is the normal outflow velocity at the boundary.

For the steady cases of $Re=20$ and 40, the Euler implicit method is used with a time step $\Delta t=0.5$. This amounts to a maximum CFL number around 20 in the steady state, and the maximum CFL may exceed 80 in transient. For such high CFL numbers, the implicit multigrid method

DOI: 10.1002/fld

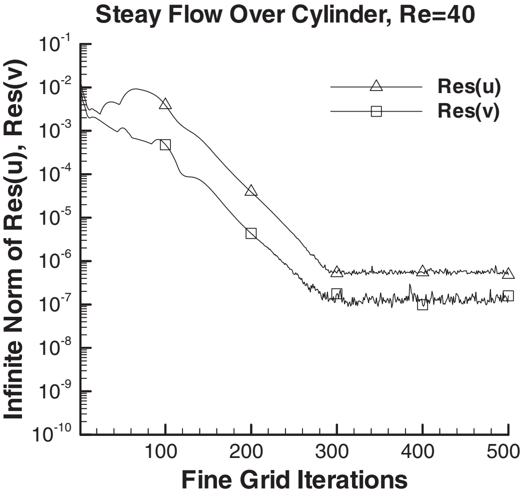

Figure 6. Convergence history for the computation of flow over a 2D stationary cylinder, $Re=40$.

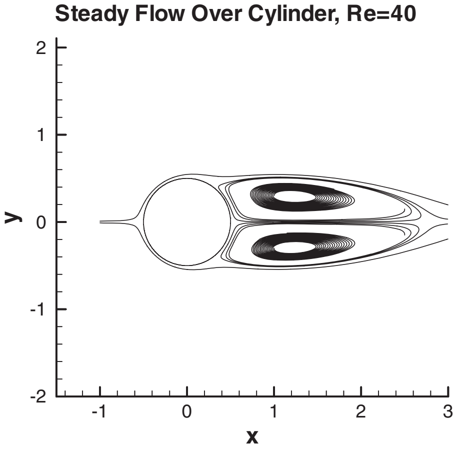

Figure 7. Streamlines showing separation bubble behind a 2D cylinder, $Re=40$.

developed in this work is considered essential for stability. For convenience, one multigrid V-cycle going through the coarse grids is counted as equivalent to one ALU relaxation step on the finest grid. In Figure 6 for the $Re=40$ case, the infinity norm of the steady-state residual for fluid cells or the Res^s in Equation (22) without the term involving Δt dropped 4 orders of magnitude in less than 500 steps. The computed streamlines for the $Re=40$ case are plotted in Figure 7. For convenience, the cylinder surface is also displayed in the figure. A separation bubble behind the cylinder is clear. The streamlines around the cylinder are smooth, indicating that the ghost cell treatment has captured the immersed boundary on the refined Cartesian grid. Figure 8 shows the computed pressure contours. Note that the pressure inside the cylinder adjusts itself automatically

DOI: 10.1002/fld

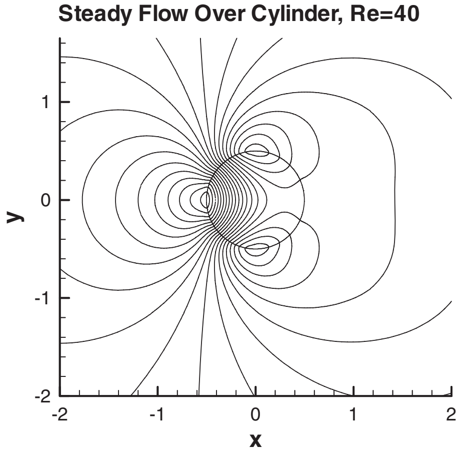

Figure 8. Pressure contours for the flow over a 2D stationary cylinder, $Re=40$.

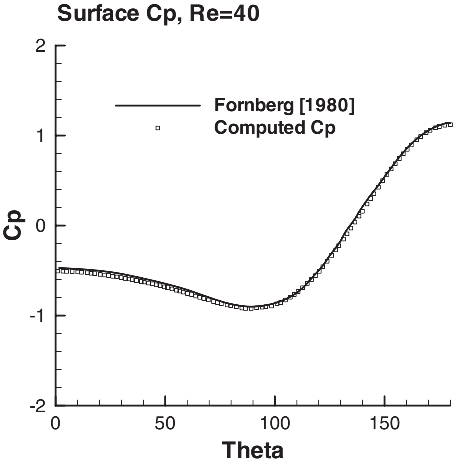

Figure 9. Pressure coefficient along the 2D cylinder surface, $Re=40$.

to the pressure field outside. The pressure contours intersect the cylinder wall in a nearly orthogonal manner. The pressure coefficient on the immersed boundary segment center is interpolated using the surrounding cell-vertex values and plotted in Figure 9. The data from Fornberg [14] are also included for comparison. The two results generally agree with each other very well, but the current computation predicted a slightly lower surface pressure distribution on the leeward surface. As for the case with $Re=20$, the convergence behavior, the velocity and pressure distributions are similar to the above results. Table II lists the computed lift coefficient (Cl), drag coefficient (Cd), the separation bubble length (Lw) normalized by the diameter (d) for both cases and some published data by others. The comparisons are generally good, including the comparison with data obtained by the ghost cell method of Tseng and Ferziger [6].

Table II. Simulation results for flows over a 2D stationary circular cylinder.

Methods	*Re*	Cd	*Lw/d*	Cl	*St*
Present	20	2.032	0.915		
Fornberg [14]	20	2.00	0.91		
Ye *et al.* [8]	20	2.03	0.92		
Present	40	1.517	2.258		
Fornberg [14]	40	1.50	2.24		
Ye *et al.* [8]	40	1.52	2.27		
Tseng and Ferziger [6]	40	1.53	2.21		
Present	100	1.332±0.010		±0.325	0.164
Liu *et al.* [15]	100	1.350±0.012		±0.339	0.165
Kim *et al.* [5]	100	1.33		±0.32	0.165
Tseng and Ferziger [6]	100	1.42		±0.4	0.164
Present	200	1.313±0.043		±0.611	0.194
Liu *et al.* [15]	200	1.31±0.049		±0.69	0.192
Kiris and Kwak [16]	200	1.27±0.04		±0.67	0.184
Present	300	1.315±0.072		±0.784	0.208
Marella *et al.* [12]	300	1.28			0.22
Ye *et al.* [8]	300	1.38			0.21

For the unsteady cases of $Re=100, 200$ and 300, the second-order accurate backward difference scheme is used for time integration with a time step $\Delta t=0.1$. The time step is chosen such that the expected vortex shedding cycle takes about 50 steps to complete The instantaneous streamlines at certain instants in the periodic vortex shedding process for $Re=200$ are plotted in Figure 10. The vorticity contours are shown in Figure 11. The unsteady vortex shedding behind the cylinder is clearly seen. Table II also lists the computed lift coefficient (Cl), drag coefficient (Cd) and the Strouhal number (St) of unsteady vortex shedding. The comparisons between the present work and the work of others are generally good. Note that in the $Re=100$ case, the averaged Cd and the amplitude of Cl obtained by Tseng and Ferziger [6] are both higher than the present results. In Reference [6], only 72 ghost cells are used around the cylinder.

7.3. Flows over a 3D sphere

To validate 3D computations, steady and unsteady flows over a sphere are also computed on a locally refined Cartesian mesh. The sphere has a diameter of unit length. The outer boundaries of the mesh are 15 diameters away from the sphere center, either parallel or perpendicular to the free stream in x direction. The free-stream condition is set to the upstream inflow boundary face and the four side faces of the mesh. The downstream boundary face follows the upwind differenced equation of $(\partial\vec{v}/\partial t)+U_n(\partial\vec{v}/\partial x)=0$, where U_n is the normal outflow velocity at the boundary. For steady-state computations, the Euler implicit time integration is used. For unsteady computation, the second-order accurate backward time integration is used.

The mesh is refined around the sphere surface such that the finest cell size is 7.32×10^{-3} while the largest cell size is 0.9375. There are totally 1 176 624 terminal cells in the mesh with 48 216 ghost cells and 100 604 ghost vertices. Without adaptive local refinement, the number of cells in a regular Cartesian mesh with a cell size of 7.32×10^{-3} would exceed 6.5×10^{10}. This clearly demonstrated the advantage of local refinement. On such a highly refined mesh, the implicit time

DOI: 10.1002/fld

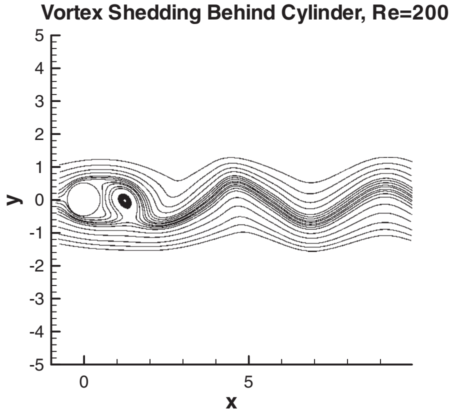

Figure 10. Streamlines showing vortex shedding behind a 2D stationary cylinder, $Re=200$.

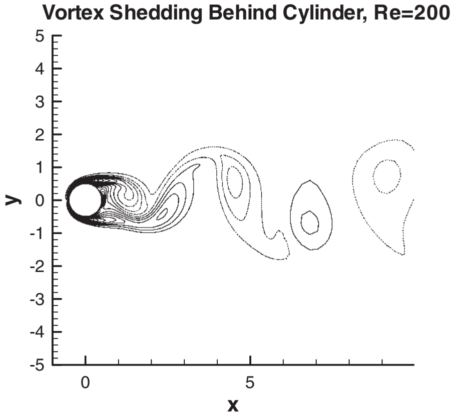

Figure 11. Vorticity contours for the flow over a 2D stationary cylinder, $Re=200$, solid lines: positive vorticity, dotted lines: negative vorticity.

integration method and the multigrid method developed in this work are essential to ensure stability and convergence of the solution process.

It is known that the airflow over a sphere is steady and axi-symmetric about the free-stream direction when Re is less than 210. Figure 12 shows for the $Re=100$ case the symmetric patterns of the streamline on the plane of $z=0$ with a separation bubble in the wake region. Figure 13 shows the pressure coefficient distribution along the sphere wall from front stagnation ($\theta=180°$) to the rear stagnation ($\theta=0°$) on the plane of $z=0$. The current result compares well with the data of Fornberg [17]. Table III lists the computed drag coefficients (in x direction) and the separation bubble length in the wake for different Reynolds numbers, along with some published data. The comparisons are generally good.

When the Reynolds number is 250, the flow is still steady, but no longer axi-symmetric. Instead, the flow becomes asymmetric in the y direction, while still remaining symmetric about the $z=0$

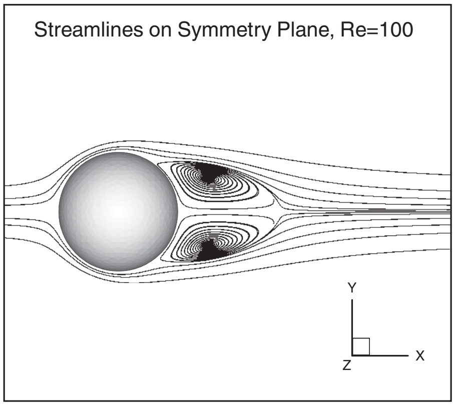

Figure 12. Streamlines for the flow over a stationary 3D sphere on symmetry plane $z=0$, $Re=100$.

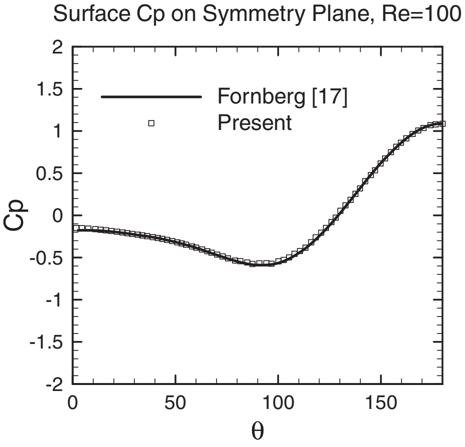

Figure 13. Pressure coefficient on 3D sphere surface cut by $z=0$ plane, $Re=100$.

plane. Figure 14 shows the streamlines on the $z=0$ plane, which exhibit the asymmetric pattern in y direction. In this case, the sphere experiences a drag force in x direction and a lift force in y direction. When the Reynolds number is 300, the asymmetry in y direction is further developed into unsteady vortex shedding, as shown by the drag and lift coefficients history experienced by the sphere in Figure 15. In this figure, the lateral force coefficient in the z direction is in the order of 5×10^{-5}, while lift is about one-tenth of the drag. The computed drag and lift coefficients and the Strouhal number of the vortex shedding are listed in Table III, which shows good comparisons with the results from others.

7.4. Impulsively started 2D moving cylinder

To test the capability of the present method to handle moving bodies, the case of an impulsively started moving cylinder was computed and compared with the experimental data by Bouard and

Table III. Simulation results for flows over a 3D stationary sphere.

Methods	*Re*	Cd	*Lw*/*D*	Cl	*St*
Present	100	1.063	0.852		
Fornberg [17]	100	1.085	0.874		
Johnson and Patel [18]	100	1.08	0.86		
Kim *et al.* [5]	100	1.087			
Marella *et al.* [12]	100	1.06	0.88		
Present	150	0.867	1.17		
Johnson and Patel [18]	150	0.90	1.20		
Marella *et al.* [12]	150	0.85	1.19		
Present	200	0.752	1.405		
Fornberg [17]	200	0.768	1.436		
Johnson and Patel [18]	200	0.771	1.439		
Present	250	0.696		0.0611	
Kim *et al.* [5]	250	0.701		0.059	
Present	300	0.638±0.002		0.069±0.013	0.133
Johnson and Patel [18]	300	0.656		0.069	0.137
Kim *et al.* [5]	300	0.657		0.067	0.134
Marella *et al.* [12]	300	0.621			0.133

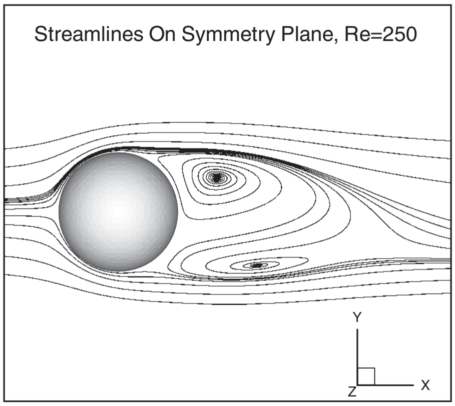

Figure 14. Streamlines for the flow over a stationary 3D sphere on symmetry plane $z=0$, $Re=250$.

Coutanceau [19]. The 2D cylinder was initially at rest and suddenly moved to the left at a constant speed U_{cyl}. The Reynolds number based on the cylinder diameter D and cylinder speed U_{cyl} is 550. The mesh is uniformly refined along the path of the cylinder such that the cylinder surface intersects about 312 cells. During the cylinder movement, some ghost cells near the rear end of the cylinder at time level n will change to fluid cells at time level $n+1$. For these fresh fluid cells, the body velocity $\vec{v}_{\text{B}}^{n}$ at time level n, not the ghost cell velocity $\vec{v}_{\text{ghost}}^{n}$ obtained by Equations (16) and (17), is used as the starting condition for time integration.

The time integration is performed for 600 steps at a constant time increment. At the end of computation, the cylinder has moved three diameters away from its initial position. Figure 16 shows the time-dependent development of velocity along the symmetry axis in the wake. In this

DOI: 10.1002/fld

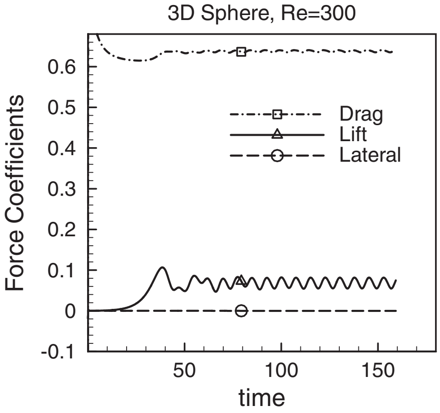

Figure 15. Force coefficients for flow over a 3D stationary sphere, $Re=300$, square: drag coefficient, triangle: lift coefficient, circle: lateral force coefficient.

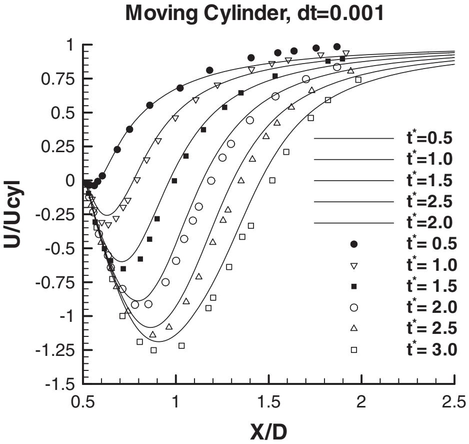

Figure 16. Flow over an impulsively started 2D cylinder, velocity u along symmetry axis in the wake, $U_{\mathrm{cyl}}D/\nu=550, \Delta t=0.001$ symbols: experimental data of Bouard and Coutanceau [19] at different times $t^*=tU_{\mathrm{cyl}}/D$ with an increment of 0.5, lines: computed.

figure the velocity u was measured relative to the moving cylinder and normalized by the cylinder speed U_{cyl}. The distance x/D along the symmetry axis was referenced relative to the cylinder center. The symbols in Figure 16 are measured manually from the experimental points in Bouard and Coutanceau [19], while the lines are from our computation. Different symbols correspond to different times $t^*=tU_{\mathrm{cyl}}/D$ with an increment of 0.5, starting from $t^*=0.5$. The agreement between computation and experiment is generally satisfactory. Figure 17 shows the computed instantaneous streamlines at $t^*=3.0$. The formation of a symmetric separation bubble in the wake and a pair of isolated secondary vortex located between the shoulder and the rear end stagnation

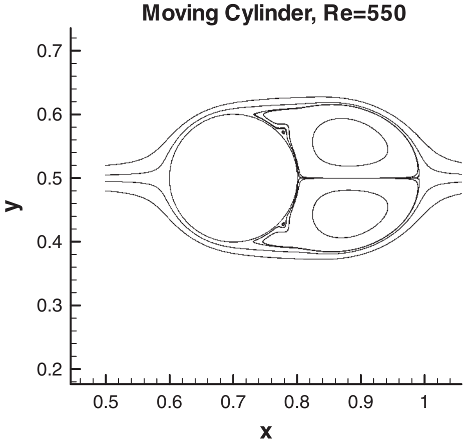

Figure 17. Flow over an impulsively started 2D cylinder, $U_{\mathrm{cyl}}D/\nu = 550, \Delta t = 0.001, t^* = tU_{\mathrm{cyl}}/D$, instantaneous streamlines at $t^* = 3.0$.

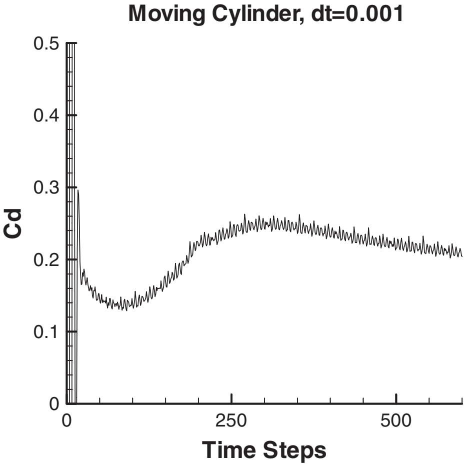

Figure 18. Drag coefficient history, flow over an impulsively started 2D cylinder, $U_{\mathrm{cyl}}D/\nu = 550, t^* = tU_{\mathrm{cyl}}/D, \Delta t = 0.001$.

can be clearly seen. This is in excellent agreement with the experimental observation of Bouard and Coutanceau [19].

The examination of velocity profiles behind the moving cylinder has basically verified that the present ghost cell method can be used to compute flows over a body in forced motion. However, when one examines the drag coefficient history in Figure 18, oscillations are observed. The sudden start of the cylinder causes the huge oscillation in drag coefficient in the initial transient for about 25 time steps. However, because the cylinder is in a preset forced motion, the observed oscillation in drag history is merely a flow response to the cylinder motion, and it does not interact with

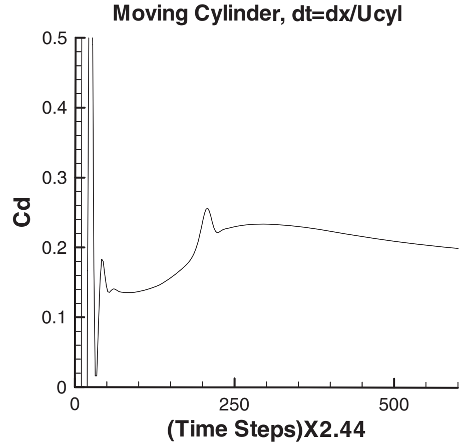

Figure 19. Drag coefficient history, flow over an impulsively started 2D cylinder, $U_{\mathrm{cyl}}D/\nu=550, t^*=tU_{\mathrm{cyl}}/D, \Delta t=\Delta x_{\min}/U_{\mathrm{cyl}}$.

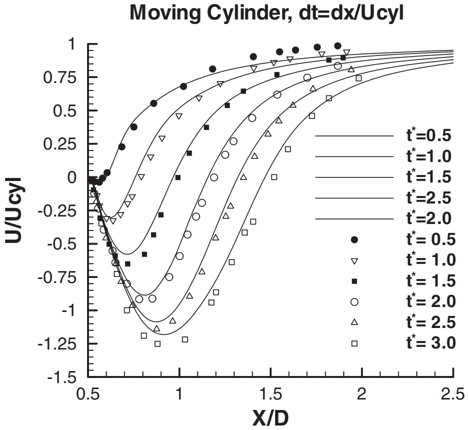

Figure 20. Flow over an impulsively started 2D cylinder, velocity u along symmetry axis in the wake, $U_{\mathrm{cyl}}D/\nu=550, \Delta t=\Delta x_{\min}/U_{\mathrm{cyl}}$, symbols: experimental data of Bouard and Coutanceau [19] at different times $t^*=tU_{\mathrm{cyl}}/D$ with an increment of 0.5, lines: computed.

the cylinder motion. After the initial transient, the drag coefficient then oscillates with smaller amplitude around certain varying mean value. This small oscillation is due to the fact that as the cylinder moves, ghost cells change their position relative to the cylinder surface at every time step, and the grid distribution around the cylinder surface also changes at every time step. Further examinations show that it is the pressure force in Equation (21) that exhibits the oscillatory behavior. The viscous force calculation shows smooth varying history. To eliminate such oscillation, the time step is reset to $\Delta t=\Delta x_{\min}/U_{\mathrm{cyl}}$ such that the cylinder is moved exactly one cell length $\Delta x_{\min}$ at every time step. Under this new time step, the ghost cells remain fixed relative to the cylinder

DOI: 10.1002/fld

surface during the cylinder motion. Figure 19 shows the drag history computed using this new time step. The initial oscillation is still observed, but the latter development of the drag coefficient becomes smooth varying. The velocity profile development in Figure 20 is similar to the results in Figure 16, even though the new time step is about 2.44 times of the original time step used in Figure 16.

Although the new time step in the above test can be set to eliminate the oscillation caused by the change of ghost cell position, this treatment is feasible only when the Cartesian grid around the body is uniform and when the body motion is simple. For a general body movement on an arbitrarily refined Cartesian grid, the ghost cell position relative to the body surface generally will change during the body movement, and the remedy for removing the oscillation of pressure force calculation requires further investigations.

8. CONCLUSIONS

A simple and stable ghost cell method is developed for the computation of incompressible flows with arbitrary immersed bodies. The ghost cells are body cells neighboring at least one fluid cell, whose values are to be determined such that the no-slip boundary condition is implicitly satisfied on the body surface. The ghost cell values are obtained by a linear reconstruction of the flow field in surface normal direction using the projection point and the image point of the ghost cell. The image point is located at a fixed distance δ away from the body surface, such that the resulting linear reconstruction is simple and stable. An implicit pressure correction upwind finite volume method is used to discretize the Navier–Stokes equations on locally refined Cartesian meshes. Multigrid methods are developed to solve the discretized equations for both velocity and pressure correction.

It is shown that the order of accuracy of the computed $\vec{v}_{\text{proj}}$ on body surface is about 1.8 for the lid-driven cavity flow considered. The method is second-order accurate in L_2 and L_∞ norms of velocity, and in the L_2 norm of pressure. Various steady and unsteady flows over a 2D cylinder and a 3D sphere are computed and compared with the published data to validate the present method. The 2D flow over a moving cylinder is also computed to test the capability of the present method to handle moving bodies. Because the distribution of ghost cells relative to the body surface may vary as the body moves, the pressure force calculation exhibits oscillations in the history of aerodynamic forces. For a linear body movement on a uniform grid, this oscillation can be eliminated by a carefully chosen time step. For more complex body movement, however, further studies are needed to find the remedy for the oscillation in the force history.

ACKNOWLEDGEMENTS

This work is partially supported by the National Science Council of Republic of China (Taiwan) through project NSC95-2221-E006-254 and NSC96-2221-E006-179-MY2. The support is highly appreciated.

REFERENCES

1. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.
2. 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.

3. 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.
4. 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.
5. 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.
6. Tseng Y-H, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
7. Frank PTB. A fictitious domain/mortar element method for fluid–structure interaction. *International Journal for Numerical Methods in Fluids* 2001; **35**:743–761.
8. 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.
9. Pan D. An immersed boundary method for incompressible flows using volume of body function. *International Journal for Numerical Methods in Fluids* 2006; **50**:733–750.
10. Pan D. An immersed boundary method on unstructured Cartesian meshes for incompressible flows with heat transfer. *Numerical Heat Transfer*, *Part B* 2006; **49**:277–297.
11. Xiao F. A computational model for suspended large rigid bodies in 3D unsteady viscous flows. *Journal of Computational Physics* 1999; **155**:348–379.
12. Marella S, Krishnan S, Liu H, Hdaykumar HS. Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations. *Journal of Computational Physics* 2005; **210**:1–31.
13. Rhie CM, Chow WL. A numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 1982; **21**:1525–1532.
14. Fornberg B. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics* 1980; **98**:819–855.
15. Liu C, Zheng X, Sung CH. Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational Physics* 1998; **139**:35–57.
16. Kiris C, Kwak D. Numerical solution of incompressible Navier–Stokes equations using a fractional-step approach. *Computers and Fluids* 2001; **30**:829–851.
17. Fornberg B. Steady viscous flow past a sphere at high Reynolds numbers. *Journal of Fluid Mechanics* 1988; **192**:471–489.
18. Johnson TA, Patel VC. Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics* 1999; **378**:19–70.
19. Bouard R, Coutanceau M. The early stage of development of the wake behind an impulsively started cylinder for $40<Re<10^4$. *Journal of Fluid Mechanics* 1980; **101**:583–607.

DOI: 10.1002/fld