

IMPLICIT VIRTUAL BOUNDARY METHOD FOR MOVING BOUNDARY PROBLEMS ON NON-STAGGERED CARTESIAN PATCH GRIDS

S. L. Lee* G. S. Cyue K. W. Chen

*Department of Power Mechanical Engineering
National Tsing Hua University
Hsinchu, Taiwan*

ABSTRACT

A simple numerical method is proposed in the paper for fluid flow around a moving boundary of irregular shape. The unsteady term is discretized with the implicit scheme such that large time step is allowed. All of the computations are performed on non-staggered Cartesian grid system. Fine Cartesian patch grid system covering the moving object is employed to resolve the solution around the solid body. A closed curve defined by connecting the solid grid points adjacent to the solid-liquid interface is referred to as virtual boundary. The narrow irregular strip of solid between the virtual boundary and the actual solid-fluid interface is called pseudo-fluid. The general fluid region consisting of both fluid and pseudo-fluid is a regular domain that can be efficiently solved with conventional numerical method. In this connection, external force is imposed at each fluid grid point adjacent to the solid-fluid interface to compensate for the numerical error arising from the assumption of pseudo-fluid. The solution procedure is iterated until the required external force converges. Accuracy of the new numerical method is validated through three test problems. The numerical method then is used to investigate the flow induced by the flapping wings of a tethered dragonfly in literature. The corresponding CFL numbers of the four examples are infinity, 20, 100, and 3.29.

Keywords: Implicit virtual boundary method, Moving boundary problem, Moving grid patch, Dragonfly flight.

1. INTRODUCTION

There are a few numerical methods for fluid flow around moving objects of irregular shape encountered in many applications in industry and in nature. One of them is to solve the problem on a boundary-fitted curvilinear grid system. In this branch of numerical methods, the grid system should be re-meshed at each time step and thus large CPU time is generally needed [1]. To improve the computational efficiency, the method of moving overset grids [2-4] generates a local non-inertial boundary-fitted grid system for each moving object that is allowed to move on the fixed global Cartesian grid system. However, the solution interpolation and data transfer among the grid systems are very complex and time consuming. Besides, it poses additional numerical errors.

Another important approach known as immersed-boundary method for fluid flow around moving object was introduced by Peskin [5]. The problem is solved on fixed Cartesian grid system. The main concept of this particular numerical method is to impose an external force field at the fluid grid points adjacent to the solid

boundary in the same manner as would the solid boundary. This branch of numerical methods was studied extensively for biological flow, multi-phase flow and problems with elastic boundary [6-12]. One of the major drawbacks of the immersed-boundary method is the smoothing of the external force field that could lead to smearing of interface information. The immersed-boundary method was further improved by Mohd-Yusof [13] and Fadlum *et al.* [14]. They proposed a direct forcing approach such that the desired velocity distribution on the immersed boundary is satisfied explicitly. Various formulations for the direct forcing approach have been developed subsequently [15-25].

In the use of the immersed-boundary method with direct forcing, the numerical procedure in the time coordinate is performed explicitly. Thus, the time step should be sufficiently small due to the CFL restriction. The interface cannot move by more than one grid mesh in each time step as remarked by Yang and Balaras [20]. To maintain the same CFL number, the time step should be reduced to one fourth of the origin size when the grid mesh is halved [26]. Unfortunately, small grid mesh is generally needed to properly describe the shape of the

* Corresponding author (sllee@pme.nthu.edu.tw)

moving object on Cartesian grids. This implies that the time step would be very small in the use of the immersed-boundary method.

In the present study, the implicit virtual boundary method is proposed for moving boundary problems. The implicit scheme is employed to discretize the unsteady term such that large time step is allowed [26]. The external force imposed at the fluid grid points adjacent to the solid-fluid interface is determined numerically. The solution procedure is iterated until the desired velocity distribution on the solid-fluid interface is achieved. All of the computations are performed on a non-staggered Cartesian grid system. Performance of the new numerical method will be examined through four test problems.

2. BRIEF REVIEW OF THE NAPPLE ALGORITHM

The dimensionless governing equations for a two-dimensional incompressible viscous flow are expressible as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (1)$$

$$Re_c \frac{\partial u}{\partial t} + Re_c u \frac{\partial u}{\partial x} + Re_c v \frac{\partial u}{\partial y} = -\frac{\partial \hat{p}}{\partial x} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + s_x \quad (2)$$

$$Re_c \frac{\partial v}{\partial t} + Re_c u \frac{\partial v}{\partial x} + Re_c v \frac{\partial v}{\partial y} = -\frac{\partial \hat{p}}{\partial y} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + s_y \quad (3)$$

where the coordinates (x, y) , velocity (u, v) , and time t have been normalized with the reference quantities, L, U_c , and L/U_c , respectively. The body force (s_x, s_y) is normalized with $\mu U_c/L^2$ if any. The dimensionless pressure is defined by $p = (P - P_{level})/\rho U_c^2$ and $\hat{p} = Re_c p$, where the reference Reynolds number is $Re_c = \rho U_c L/\mu$. The reference quantities (L and U_c) are to be defined for each individual application.

Let the partial differential Eqs. (1)-(3) be discretized on a uniform Cartesian grid system (x_i, y_j) , i.e.

$$x_i = x_1 + (i-1)\Delta x, \quad y_j = y_1 + (j-1)\Delta y \quad (4)$$

with $i = 1, 2, 3, \dots, m$, and $j = 1, 2, 3, \dots, n$. For convenience of numerical procedure, (x_i, y_j) is numbered as the k -th grid point with k being $k = (i-1)n + j$, $k = 1, 2, 3, \dots, mn$ (see Fig. 1). Based on the weighting function scheme [27], the resulting algebraic equation for the momentum Eq. (2) at the k -th grid point can be written as

$$(a_W)_k u_{k-n} + (a_S)_k u_{k-1} + (a_P)_k u_k + (a_N)_k u_{k+1} + (a_E)_k u_{k+n} = (a_u)_k + \left(\frac{\partial \hat{p}}{\partial x} \right)_k \quad (5)$$

where the weighting factors are defined by

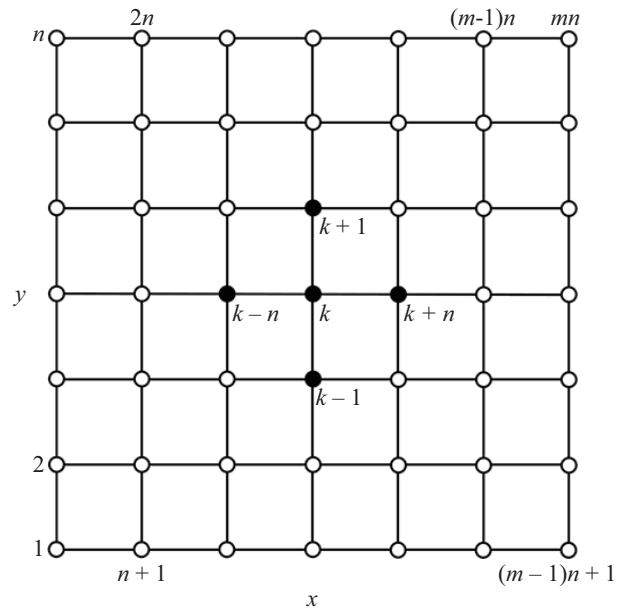


Fig. 1 Numbering of grid points on Cartesian grid system.

$$(a_W)_k = \frac{w_f((z_w)_k)}{(\Delta x)^2}, \quad (a_E)_k = \frac{w_f(-(z_e)_k)}{(\Delta x)^2},$$

$$(a_S)_k = \frac{w_f((z_s)_k)}{(\Delta y)^2}, \quad (a_N)_k = \frac{w_f(-(z_n)_k)}{(\Delta y)^2},$$

$$(a_P)_k = -(a_W)_k - (a_S)_k - (a_N)_k - (a_E)_k - \frac{Re_c}{\Delta t}$$

$$(a_u)_k = -\left((s_x)_k + \frac{Re_c u_0}{\Delta t} \right), \quad w_f(z) = \frac{z}{1 - \exp(-z)} \quad (6)$$

The subscript k represents the physical quantity at the k -th grid point. The grid Peclet numbers are

$$(z_w)_k = Re_c \bar{u}_{i-1} \Delta x, \quad (z_e)_k = Re_c \bar{u}_i \Delta x$$

$$(z_s)_k = Re_c \bar{v}_{j-1} \Delta y, \quad (z_n)_k = Re_c \bar{v}_j \Delta y \quad (7)$$

where \bar{u}_i and \bar{v}_j denote the mean velocity components in the intervals $x_i \leq x \leq x_{i+1}$ and $y_j \leq y \leq y_{j+1}$, respectively. To conserve CPU time, the weighting function $w_f(z)$ is computed with the power-law approximation [27, 28]. As suggested by Lee [26], the unsteady term has been discretized with the implicit scheme

$$\frac{\partial u}{\partial t} = \frac{u - u_0}{\Delta t} \quad (8)$$

in formulation (6) to achieve the best numerical stability, where u_0 represents the velocity at the previous time level ($t_0 = t - \Delta t$). The CFL number $\lambda = \Delta t/(\Delta x)^2$ is the key factor for numerical stability in time marching scheme [26]. The well-known CFL restriction is $\lambda < 0.5$ when an explicit scheme is used.

Equation (5) can be recast into the simple form

$$u_k = \hat{u}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial x} \right)_k \quad (9)$$

where

$$\hat{u}_k = ((a_u)_k - (a_W)_k u_{k-n} - (a_S)_k u_{k-1} - (a_N)_k u_{k+1} - (a_E)_k u_{k+n}) / (a_P)_k$$

$$\gamma_k = -\frac{1}{(a_P)_k} \quad (10)$$

Similarly, discretization of Eq. (3) for the velocity component v yields

$$(a_W)_k v_{k-n} + (a_S)_k v_{k-1} + (a_P)_k v_k + (a_N)_k v_{k+1} + (a_E)_k v_{k+n}$$

$$= (a_v)_k + \left(\frac{\partial \hat{p}}{\partial y} \right)_k \quad (11)$$

and thus

$$v_k = \hat{v}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial y} \right)_k \quad (12)$$

Next, substitute the velocity expression (9) and (12) into the discretized continuity equation

$$\frac{u_{k+n} - u_{k-n}}{2 \Delta x} + \frac{v_{k+1} - v_{k-1}}{2 \Delta y} = 0 \quad (13)$$

and then make the assumption

$$\frac{\gamma_{k+n} \left(\frac{\partial \hat{p}}{\partial x} \right)_{k+n} - \gamma_{k-n} \left(\frac{\partial \hat{p}}{\partial x} \right)_{k-n}}{2 \Delta x} \approx \frac{\partial}{\partial x} \left(\gamma \frac{\partial \hat{p}}{\partial x} \right) \quad (14)$$

$$\frac{\gamma_{k+1} \left(\frac{\partial \hat{p}}{\partial y} \right)_{k+1} - \gamma_{k-1} \left(\frac{\partial \hat{p}}{\partial y} \right)_{k-1}}{2 \Delta y} \approx \frac{\partial}{\partial y} \left(\gamma \frac{\partial \hat{p}}{\partial y} \right) \quad (15)$$

to yield the pressure-linked equation

$$\frac{\partial}{\partial x} \left(\gamma \frac{\partial \hat{p}}{\partial x} \right) + \frac{\partial}{\partial y} \left(\gamma \frac{\partial \hat{p}}{\partial y} \right) = \hat{\varepsilon} \quad (16)$$

$$\hat{\varepsilon}_k = \frac{\hat{u}_{k+n} - \hat{u}_{k-n}}{2 \Delta x} + \frac{\hat{v}_{k+1} - \hat{v}_{k-1}}{2 \Delta y} \quad (17)$$

Use of the approximation (14) and (15) is to prevent the checkerboard pressure error. Finally, discretize Eq. (16) with the harmonic mean scheme [28] to obtain the algebraic equation

$$(b_W)_k \hat{p}_{k-n} + (b_S)_k \hat{p}_{k-1} + (b_P)_k \hat{p}_k + (b_N)_k \hat{p}_{k+1} + (b_E)_k \hat{p}_{k+n} = (b_R)_k \quad (18)$$

$$(b_W)_k = \frac{(\gamma_w)_k}{(\Delta x)^2}, \quad (b_E)_k = \frac{(\gamma_e)_k}{(\Delta x)^2},$$

$$(b_S)_k = \frac{(\gamma_s)_k}{(\Delta y)^2}, \quad (b_N)_k = \frac{(\gamma_n)_k}{(\Delta y)^2},$$

$$(b_P)_k = -(b_W)_k - (b_S)_k - (b_N)_k - (b_E)_k, \quad (b_R)_k = \hat{\varepsilon}_k \quad (19)$$

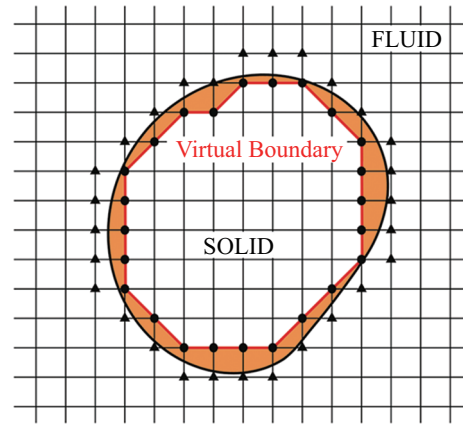


Fig. 2 Virtual points (black dots), virtual boundary, pseudo-fluid, and forcing points (black triangles).

where the “pressure conductivity” γ is evaluated from

$$(\gamma_w)_k = \frac{2}{(\gamma_k)^{-1} + (\gamma_{k-n})^{-1}} = \frac{-2}{(a_P)_k + (a_P)_{k-n}} \quad (20)$$

$$(\gamma_e)_k = \frac{2}{(\gamma_k)^{-1} + (\gamma_{k+n})^{-1}} = \frac{-2}{(a_P)_k + (a_P)_{k+n}} \quad (21)$$

$$(\gamma_s)_k = \frac{2}{(\gamma_k)^{-1} + (\gamma_{k-1})^{-1}} = \frac{-2}{(a_P)_k + (a_P)_{k-1}} \quad (22)$$

$$(\gamma_n)_k = \frac{2}{(\gamma_k)^{-1} + (\gamma_{k+1})^{-1}} = \frac{-2}{(a_P)_k + (a_P)_{k+1}} \quad (23)$$

To achieve a good numerical stability, the algebraic Eqs. (5), (11), and (18) are solved with the strongly implicit solver [29]. The numerical procedure described in this section is called the NAPPLE algorithm [30]. It has been proved to solve fluid flow and heat transfer problems very efficiently in the past decades.

3. FLOW AROUND OBJECT OF IRREGULAR SHAPE

3.1 The Implicit Virtual Boundary Method

Consider the fluid flow around an immersed body of irregular shape as illustrated in Fig. 2. A grid point in the solid region having at least one neighbor grid point in the fluid region is called *virtual point* as that marked with black dot in Fig. 2. A closed curve formed by connecting the adjacent virtual points is referred to as the *virtual boundary* in the present study. The narrow irregular strip of solid between the virtual boundary and the actual solid-fluid interface is called *pseudo-fluid*. Similarly, fluid grid point marked with black triangle in Fig. 2 is called *forcing point*. They have at least one neighbor grid point in the solid region. Obviously, the general fluid region consisting of both fluid and pseudo-fluid is a regular domain that can be efficiently

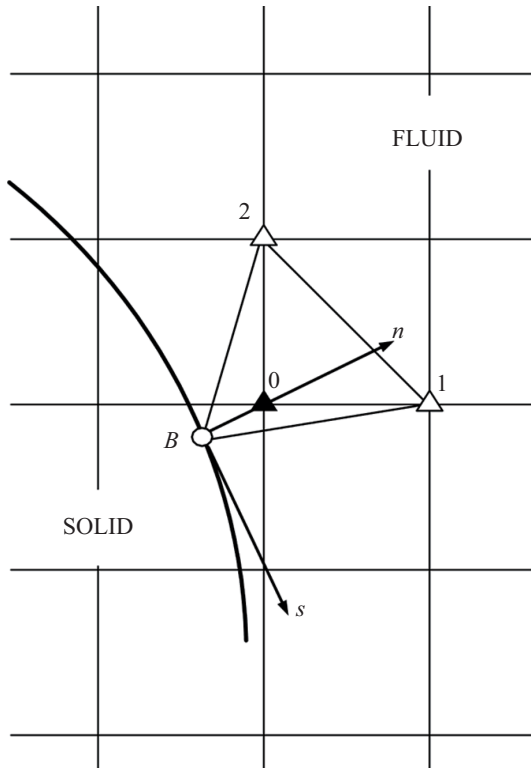


Fig. 3 Bi-linear interpolation stencil on a schematic solid-fluid interface.

solved with the numerical method described in section 2, if the velocity expressions (9) and (12) are employed at the virtual point with the particular values of $\hat{u}_k = u_{vp}$, $\hat{v}_k = v_{vp}$, and $(a_p)_k = \infty$, where (u_{vp}, v_{vp}) is the velocity of the solid body at the same grid point. There is no need to impose pressure boundary condition at the virtual boundary. However, some numerical treatment is required to evaluate the pressure gradients at the forcing points as well as to correct the numerical error arising from the assumption of pseudo-fluid.

As suggested by Yang and Balaras [20], the velocity and pressure at the forcing points are interpolated from the updated solution of the fluid region and the known boundary condition at the solid-fluid interface. Figure 3 shows a schematic solid-fluid interface with a tangent-normal coordinate system (s, n) at point $B(x_B, y_B)$. The directional cosines of the s -axis are l and m in the global coordinate system (x, y) , i.e.

$$\begin{Bmatrix} s \\ n \end{Bmatrix} = \begin{bmatrix} l & m \\ -m & l \end{bmatrix} \begin{Bmatrix} x - x_B \\ y - y_B \end{Bmatrix} \quad (24)$$

The line $\overline{0B}$ (the n -axis) is normal to the solid-fluid interface at point B . The grid point marked with 0 is the forcing point. The velocity and pressure inside the triangle formed by point B and the two fluid grid points marked with 1 and 2 are assumed the bi-linear form

$$u(x, y) = c_{11} + c_{12}x + c_{13}y \quad (25)$$

$$v(x, y) = c_{21} + c_{22}x + c_{23}y \quad (26)$$

$$\hat{p}(x, y) = c_{31} + c_{32}x + c_{33}y \quad (27)$$

In the conventional formulation [20] one has to impose the constraint that the stencil of the bi-linear interpolation does not involve other forcing points. Fortunately, there is no such a constraint in the present formulation due to the use of the implicit scheme for the time coordinate (8). This greatly simplifies the programming of the numerical algorithm.

After substituting the velocity components u_1, u_2, u_B at the three points $(x_1, y_1), (x_2, y_2), (x_B, y_B)$ into Eq. (25), one has

$$\begin{aligned} c_{11} &= u_1 - c_{12}x_1 - c_{13}y_1 \\ c_{12} &= a_1u_1 + a_2u_2 + a_3u_B \\ c_{13} &= b_1u_1 + b_2u_2 + b_3u_B \end{aligned} \quad (28)$$

where

$$\begin{aligned} a_1 &= (y_2 - y_B)/S, & a_2 &= (y_B - y_1)/S, & a_3 &= (y_1 - y_2)/S \\ b_1 &= (x_B - x_2)/S, & b_2 &= (x_1 - x_B)/S, & b_3 &= (x_2 - x_1)/S \\ S &= (x_2 - x_1)(y_B - y_1) - (x_B - x_1)(y_2 - y_1) \end{aligned} \quad (29)$$

Coefficients of Eq. (26) are determined similarly, i.e.

$$\begin{aligned} c_{21} &= v_1 - c_{22}x_1 - c_{23}y_1 \\ c_{22} &= a_1v_1 + a_2v_2 + a_3v_B \\ c_{23} &= b_1v_1 + b_2v_2 + b_3v_B \end{aligned} \quad (30)$$

Generally speaking, the pressure at the boundary point B is not known in practical applications. Nevertheless, the normal pressure gradient $(\partial\hat{p}/\partial n)_B$ on the solid-fluid interface can be estimated from the momentum equations with the bi-linear approximation (25)-(27), i.e.

$$\begin{aligned} \left(\frac{\partial\hat{p}}{\partial x}\right)_B &= -Re_c \left(\frac{\partial u_B}{\partial t} + c_{12}u_B + c_{13}v_B \right) + s_x \\ \left(\frac{\partial\hat{p}}{\partial y}\right)_B &= -Re_c \left(\frac{\partial v_B}{\partial t} + c_{22}u_B + c_{23}v_B \right) + s_y \\ \left(\frac{\partial\hat{p}}{\partial n}\right)_B &= -m \left(\frac{\partial\hat{p}}{\partial x}\right)_B + l \left(\frac{\partial\hat{p}}{\partial y}\right)_B \end{aligned} \quad (31)$$

The coefficients of Eq. (27) turn out to be

$$\begin{aligned} c_{31} &= \hat{p}_1 - c_{32}x_1 - c_{33}y_1 \\ c_{32} &= \frac{l(\hat{p}_2 - \hat{p}_1) - (\partial\hat{p}/\partial n)_B(y_2 - y_1)}{l(x_2 - x_1) + m(y_2 - y_1)} \\ c_{33} &= \frac{(\partial\hat{p}/\partial n)_B(x_2 - x_1) + m(\hat{p}_2 - \hat{p}_1)}{l(x_2 - x_1) + m(y_2 - y_1)} \end{aligned} \quad (32)$$

After the interpolation coefficients are all available, the velocity and pressure at the forcing point (x_0, y_0) are determined by

$$u_{fp} = c_{11} + c_{12}x_0 + c_{13}y_0 \quad (33)$$

$$v_{fp} = c_{21} + c_{22}x_0 + c_{23}y_0 \quad (34)$$

$$\hat{p}_{fp} = c_{31} + c_{32}x_0 + c_{33}y_0 \quad (35)$$

Similarly, the pressure gradients at the forcing points are determined from

$$\left(\frac{\partial \hat{p}}{\partial x}\right)_{fp} = c_{32}, \quad \left(\frac{\partial \hat{p}}{\partial y}\right)_{fp} = c_{33} \quad (36)$$

before the momentum Eqs. (5) and (11) are solved, because they cannot be obtained by the conventional central difference scheme.

Once the pressure gradients at the forcing points are available, the flow solution in the fluid region is updated by solving Eqs. (5), (11), and (18). To compensate for the numerical error arising from the assumption of pseudo-fluid, an external force (f_x, f_y) is applied at the forcing point to yield

$$u_k = \hat{u}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial x}\right)_k + \gamma_k (f_x)_k$$

$$v_k = \hat{v}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial y}\right)_k + \gamma_k (f_y)_k \quad (37)$$

when the momentum Eqs. (2) and (3) are discretized at the forcing point. Suppose that the velocity (\tilde{u}, \tilde{v}) has been obtained based on a guessed external force (\tilde{f}_x, \tilde{f}_y) at the forcing point, i.e.

$$\tilde{u}_k = \hat{u}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial x}\right)_k + \gamma_k (\tilde{f}_x)_k$$

$$\tilde{v}_k = \hat{v}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial y}\right)_k + \gamma_k (\tilde{f}_y)_k \quad (38)$$

For the next iteration, one makes a correction ($\Delta f_x, \Delta f_y$) to the external force to eliminate the discrepancy between the resulting velocity and that from the interpolation procedure (33) and (34), i.e.

$$(u_k)_{fp} = \hat{u}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial x}\right)_k + \gamma_k ((\tilde{f}_x)_k + (\Delta f_x)_k)$$

$$(v_k)_{fp} = \hat{v}_k - \gamma_k \left(\frac{\partial \hat{p}}{\partial y}\right)_k + \gamma_k ((\tilde{f}_y)_k + (\Delta f_y)_k) \quad (39)$$

This leads to

$$(\Delta f_x)_k = -(a_p)_k ((u_k)_{fp} - \tilde{u}_k),$$

$$(\Delta f_y)_k = -(a_p)_k ((v_k)_{fp} - \tilde{v}_k) \quad (40)$$

by subtracting Eq. (38) from Eq. (39). Similarly, one adds a forcing term $(f_p)_k$ to the right-hand side of the pressure-linked Eq. (18) and then corrects the forcing term with

$$(\Delta f_p)_k = (b_p)_k ((\hat{p}_k)_{fp} - \tilde{p}_k) \quad (41)$$

where $(\hat{p}_k)_{fp}$ is the desired pressure (35) and \tilde{p}_k is the pressure solution of the previous iteration.

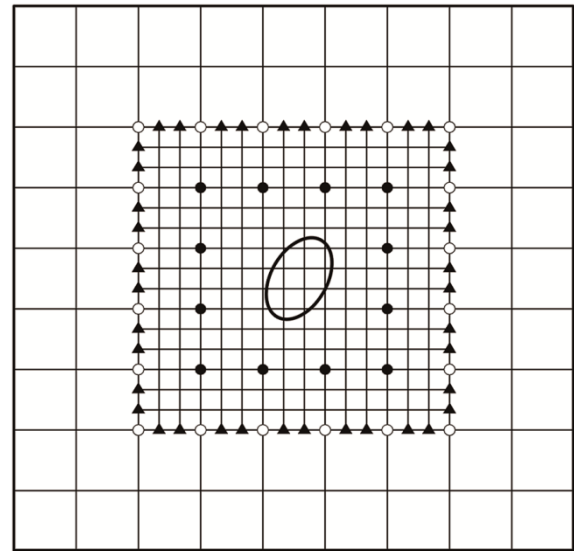


Fig. 4 A fine grid system patched on a global grid system of coarser grid mesh.

3.2 Patch Grids and Numerical Implementation

To conserve CPU time and memory storage, Peng *et al.* [31] proposed a local grid refinement method (nested Cartesian grid method). Solutions of the global grid domain and the nested fine grid domain are solved through the use of some "ghost cells." However, the values at the ghost cells should be interpolated by some means from the solutions of the global and the nested grid domains. In the present study, Cartesian patch grids are proposed without recourse to the ghost cells.

Figure 4 depicts schematically a local fine grid system ($\Delta x \times \Delta y$) patched on a global grid system of coarser grid mesh ($3\Delta x \times 3\Delta y$ for instance). The global grid system has a "hole" defined by the grid points marked with black dot. For convenience of interpretation, the hole surface is referred to as the inner boundary of the global grid system. Similarly, the patch grid system possesses an outer boundary represented by the grid points marked with white dot and black triangle. Between the two grid systems, there is an overlap strip that has a width just equaling to one grid mesh of the global grid system. Practically, the object of irregular shape under study should be placed inside the patch grid system with an adequate margin from the inner boundary of the global grid system.

After the flow field on the patch grid system is updated, the velocity and pressure at the particular grid points marked with black dot are transferred to the corresponding grid points on the inner boundary of the global grid system. Unlike in the nested Cartesian grid method proposed by Peng *et al.* [31], no interpolation procedure is needed in the present study due to the fact that the black dots are the common grid points of the global and the patched grid systems. Based on these Dirichlet boundary conditions on the inner boundary, the flow field in the global grid system is solved with a conventional numerical method such as that described in section 2. The resulting velocity and pressure at the

grid points marked with white dot then is transferred to the outer boundary of the patch grid system. The other grid points on the outer boundary of the patch grid system (marked with black triangle) are determined with the method of linear interpolation. After this is done, the flow field inside the patch grid system is updated with the implicit virtual boundary method. The numerical procedure should be iterated until the solution converges within a prescribed tolerance. Finally, the pressure and shear stress at point $B(x_B, y_B)$ on the solid-fluid interface are evaluated from

$$p = \frac{c_{31} + c_{32}x_B + c_{33}y_B}{Re_c}$$

$$\tau = \frac{l^2 c_{13} + lm(c_{23} - c_{12}) - m^2 c_{22}}{Re_c} \quad (42)$$

where the shear stress τ has been normalized with ρU_c^2 .

4. RESULTS AND DISCUSSION

Accuracy of the implicit virtual boundary method described in the previous section is examined through three examples, namely, uniform flow past a stationary cylinder, in-line oscillating cylinder in rest fluid, and flow past a vertical flat plate of zero thickness. The present numerical method then is employed to investigate the flow induced by the flapping wings of a tethered dragonfly. Both truncated computational domains and grid sizes of the grid systems employed in the four test problems were determined after a series of grid tests. Although only 2D examples are demonstrated herein, the present numerical method applies to 3D problems as well.

4.1 Uniform Flow past a Stationary Cylinder

In this example, the implicit virtual boundary method is employed to simulate the flow past a stationary cylinder. The reference quantities are assigned as $L = D$ and $U_c = U_\infty$, where D is the cylinder diameter and U_∞ is the free stream velocity. For convenience, the free stream pressure is assigned as the pressure level $p_\infty = 0$. The velocity is zero on the cylinder surface due to the no-slip condition. The normal pressure gradient on the cylinder surface is zero too, because of the assumption of bilinear velocity variation (the viscosity effect has been neglected) in the vicinity of the cylinder surface. Computation is performed in the finite domain of $-8 \leq x \leq 24$ and $-8 \leq y \leq 8$ with the cylinder center being located at $(x, y) = (0, 0)$. The free stream condition $(u, v, p) = (1, 0, 0)$ is imposed on all of the boundaries of the computational domain except for $\partial u / \partial x = 0$ at the downstream boundary ($x = 24$). This same flow configuration has been studied by Tseng and Ferziger [16] with the ghost-cell immersed boundary method.

In the present computation, a patch grid system is applied in the region of $-1.2 \leq x \leq 1.2$ and $-1.2 \leq y \leq 1.2$

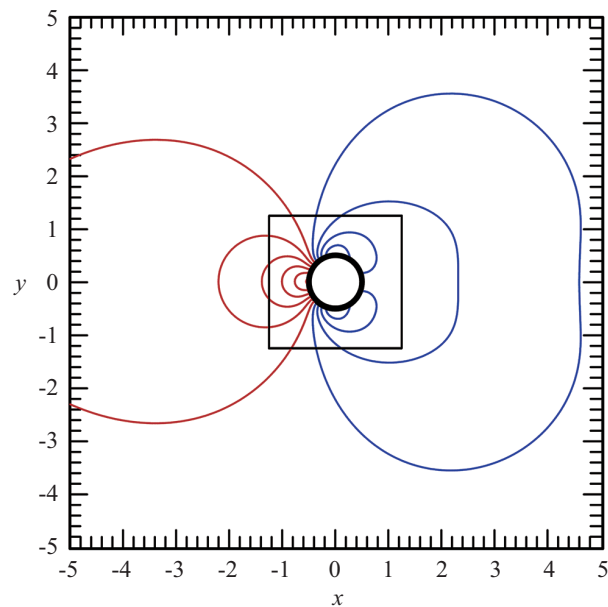


Fig. 5 The resulting isobars with increment of $\Delta p = 0.1$ for $Re = 40$.

with the grid mesh $\Delta x = \Delta y = 0.005$, while the global grid system has the grid mesh $5\Delta x \times 5\Delta y$. Steady state solution is obtained for the Reynolds number of $Re = 20$ and $Re = 40$. The resulting isobars for $Re = 40$ are shown in Fig. 5 with increment of $\Delta p = 0.1$. For convenience of observation, the isobars in red stand for positive pressure while blue for negative pressure. The isobars pass smoothly across the overlap strip of the two grid systems. They are smooth in the vicinity of the cylinder surface too. The corresponding pressure and shear stress distributions along the cylinder surface are revealed in Figs. 6 and 7, respectively. The numerical result essentially has no change when the problem is solved on a single fine grid system of $\Delta x = \Delta y = 0.005$. This evidences the correctness of the patch grid formulation. The existing numerical results from the boundary-fitted grids [32] and the ghost-cell immersed boundary method [16] for $Re = 40$ are plotted also in Figs. 6 and 7 for comparison. From Fig. 6, good agreements are observed among the three numerical results. Similarly, Fig. 7 shows good agreements on the shear stress among the three numerical methods.

For both cases of $Re = 20$ and $Re = 40$, there is a stable recirculation zone behind the cylinder that contains a pair of symmetric vortices. To examine the accuracy of the predicted wake structure, the present numerical results for the length of the recirculation zone l/D , distance from the cylinder to the centers of the vortices a/D , the gap between the centers of the vortices b/D , the separation angle θ_s , and the drag coefficient C_D are compared with that from the existing experimental and numerical studies [33-36] in Table 1. Good agreements among the results are seen. The present numerical method seems to predict a drag coefficient that is slightly smaller than the existing numerical results [34-36]. This will be discussed later.

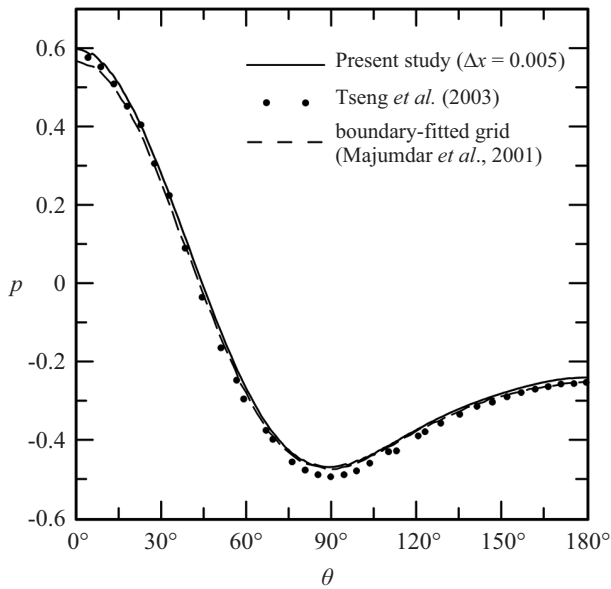


Fig. 6 The resulting pressure distributions along the cylinder surface for $Re = 40$.

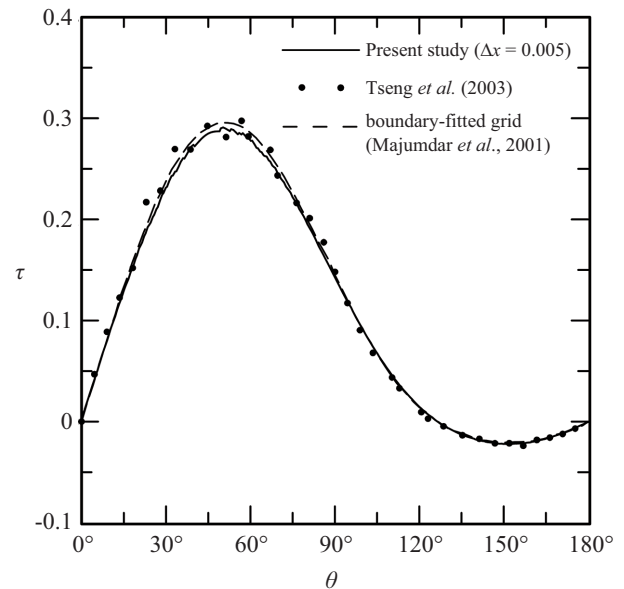


Fig. 7 The resulting shear stress distributions along the cylinder surface for $Re = 40$.

Table 1 Results of wake dimension and drag coefficient for uniform flow past a cylinder

case	references	method	l/D	a/D	b/D	θ_s	C_D
Re = 20	Coutanceau and Bouard [33]	experiment	0.93	0.33	0.46	135.0°	--
	Dennis and Chang [34]	numerical	0.94	--	--	136.3°	2.05
	Linnick and Fasel [35]	numerical	0.93	0.36	0.43	136.5°	2.06
	Taira and Colonius [36]	numerical	0.94	0.37	0.43	136.7°	2.06
	Present study	numerical	0.95	0.35	0.43	136.6°	2.01
Re = 40	Coutanceau and Bouard [33]	experiment	2.13	0.76	0.59	126.2°	--
	Dennis and Chang [34]	numerical	2.35	--	--	126.2°	1.52
	Linnick and Fasel [35]	numerical	2.28	0.72	0.60	126.4°	1.54
	Taira and Colonius [36]	numerical	2.30	0.73	0.60	126.3°	1.54
	Present study	numerical	2.27	0.70	0.61	127.3°	1.48

4.2 In-Line Oscillating Cylinder in Rest Fluid

Flow around an in-line oscillating cylinder in rest fluid has been well-documented in the literature. In this test case, the reference length and reference velocity are assigned as $L = D$ and $U_c = fD$, respectively, where D is the cylinder diameter and f is the frequency of the oscillating cylinder. The ambient pressure is set as the pressure level $p_\infty = 0$. Computation is performed in the finite domain of $-25 \leq x \leq 25$ and $-15 \leq y \leq 15$ with the cylinder center being located at

$$x_c(t) = \frac{KC}{2\pi} \cos(2\pi t), \quad y_c(t) = 0 \quad (43)$$

for the Reynolds number $Re_c = Re = \rho f D^2 / \mu = 20$ and the Keulegan-Carpenter number $KC = 5$. The KC number represents the maximum velocity of the oscillating cylinder. The amplitude of the motion is $5/(2\pi) =$

0.7958. Hence, a patch grid system is employed in the region of $-3 \leq x \leq 3$ and $-2 \leq y \leq 2$ that covers the entire oscillating cylinder with a proper margin. The grid mesh of the patch grid system is $\Delta x = \Delta y = 0.01$, while that of the global grid system is $5\Delta x \times 5\Delta y$. Static fluid condition $(u, v, p) = (0, 0, 0)$ is set on the outer boundary of the global grids.

This test case deals with moving boundary. In most previous studies, the time coordinate is discretized with some explicit scheme such that the solid-fluid interface cannot move by more than one grid mesh in each time step due to the CFL restriction [20]. In the present study, the unsteady term is discretized with the implicit scheme (8), and thus large time step is allowed as mentioned earlier. For a retreating solid body as illustrated in Fig. 8, the grid points that the solid-fluid interface sweeps through within a time step are referred to as the "ghost" points (see the white nodes) when the time elapses from $t_0 = t - \Delta t$ to t . Suppose that the moving

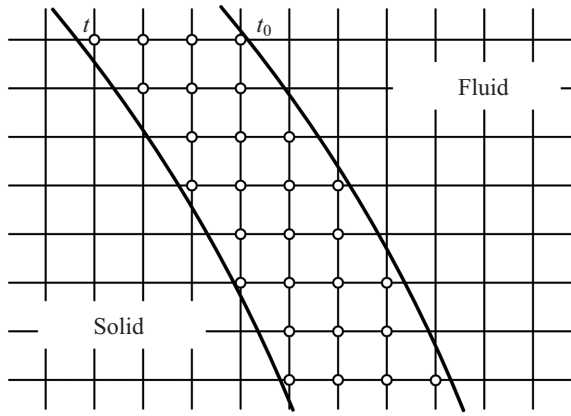


Fig. 8 The ghost points (marked with white dots) appearing from a retreating solid body.

boundary arrive at a ghost point at the time $t_0 + \beta\Delta t$ with $0 < \beta < 1$. At this particular moment, the fluid velocity at this ghost point should be the same as that of the solid-fluid interface (u_B, v_B). Hence, the unsteady term is discretized with

$$\frac{\partial u}{\partial t} = \frac{u - u_B}{(1 - \beta)\Delta t} \quad (44)$$

instead of Eq. (8) at the ghost point. A similar implicit scheme was proposed by Lee and You [37]. In their formulation, however, the pressure gradient at the ghost cell needs to estimate for the time $t_0 + \beta\Delta t$ to refresh the ghost cell at time t . Such a tedious numerical procedure is not required in the present study.

Computation is undertaken with $\Delta t = 0.002$. The corresponding CFL number is $\lambda = 20$. The resulting isobars are depicted in Fig. 9 with increment $\Delta p = 3$ at five representative times in a half period of the oscillation ($0 \leq t < 0.5$) after the fluid flow has reached the quasi-steady state. Again, smooth isobars are observed in the vicinity of the cylinder surface. Figure 9 indicates that positive pressure (in red) occurs on the left-side surface of the cylinder at $t = 0$ due to the deceleration of the cylinder when it moved rightwards in the fourth quarter of the previous cycle ($-0.25 < t < 0$). This causes also negative pressure (in blue) on the right-side surface of the cylinder. The negative pressure is strengthened when the cylinder accelerates leftwards in the first quarter of the cycle ($0 < t \leq 0.25$). In the course of the stroke a detached positive pressure bubble is found to nucleate behind the moving cylinder. In the second quarter of the cycle ($0.25 \leq t < 0.5$), an attached positive pressure bubble appears on the right-side surface of the cylinder due to the deceleration of the cylinder. The two positive pressure bubbles eventually merge together at $t = 0.38$, while the positive pressure on the left-side surface of the cylinder disappears completely. The isobars at the end of the stroke ($t = 0.5$) is the same as that at $t = 0$.

Figure 10 shows the velocity profiles along the y -axis ($x = 0$) at three representative times. The velocity pro-

file in the region of $|y| \leq \sqrt{1 - x_c^2}$ is uniform because it is inside the cylinder. The present numerical result is compared with the embedded-boundary formulation [20] and the experiment data from Dust *et al.* [38] in Fig. 10. Good agreement is observed.

4.3 Flow past a Vertical Flat Plate of Zero Thickness

Consider an impulsively started vertical flat plate of zero thickness immersed in quiescent fluid. The velocity of the flat plate is U_∞ after the onset of the motion. The plate length L and velocity U_∞ are assigned as, respectively, the reference length and reference velocity for the problem. A moving coordinate system (x, y) is defined such that the flat plate is always located at $x = \Delta x/2$ and $-0.5 \leq y \leq 0.5$, where Δx is the grid mesh.

In this test problem, the flat plate is of zero thickness. Care must be exercised in the treatment of the virtual points because there is no pseudo-fluid region. Suppose that the flat plate is located in between grid point k and $k + n$. When the governing equation is discretized at the forcing point k , the virtual point $k + n$ would be in the fluid region behind the flat plate. In the present study, the fluid on the other side of the flat plate, together with the flat plate itself, is regarded as a rigid solid. The velocity at the virtual point $k + n$ is evaluated with the theory of solid-body motion. After discretization of the governing equation at the forcing point k , the term $(a_E)_k u_{k+n}$ involving the velocity of the virtual point is moved to the right-hand-side of the equation to yield

$$\begin{aligned} & (a_W)_k u_{k-n} + (a_S)_k u_{k-1} + (a_P)_k u_k + (a_N)_k u_{k+1} \\ & = (a_U)_k - (a_E)_k u_{k+n} + \left(\frac{\partial \hat{p}}{\partial x} \right)_k - (f_x)_k \end{aligned} \quad (45)$$

This strategy is valid for all problems in spite of the thickness of the solid.

The domain of the test problem is truncated to the region of $-13 \leq x \leq 17$ and $-4 \leq y \leq 4$. A patch grid system covering $-3 \leq x \leq 5$ and $-2 \leq y \leq 2$ is employed. The grid mesh of the patch grid system is $\Delta x = \Delta y = 0.01$, while that of the global grid system is $5\Delta x \times 5\Delta y$. The free stream condition $(u, v, p) = (1, 0, 0)$ is imposed on all of the boundaries of the computational domain except for $\partial u / \partial x = 0$ at the downstream boundary as in the example of uniform flow past a cylinder in section 4.1. The time step is $\Delta t = 0.01$. The corresponding CFL number is as large as $\lambda = 100$ in the patch grid system. Numerical result is obtained for the Reynolds number of $Re = 126$. The free stream pressure is assigned as the pressure level $p_\infty = 0$. Once the velocity solution is available at each time step, the stream function is evaluated from

$$\begin{aligned} \psi(x, 0, t) &= -\int_{0.5\Delta x}^x v(x, 0, t) dx \quad \text{for } -13 \leq x \leq 17 \\ \psi(x, y, t) &= \psi(x, 0, t) + \int_0^y u(x, y, t) dy \quad \text{for } -4 \leq y \leq 4 \end{aligned} \quad (46)$$

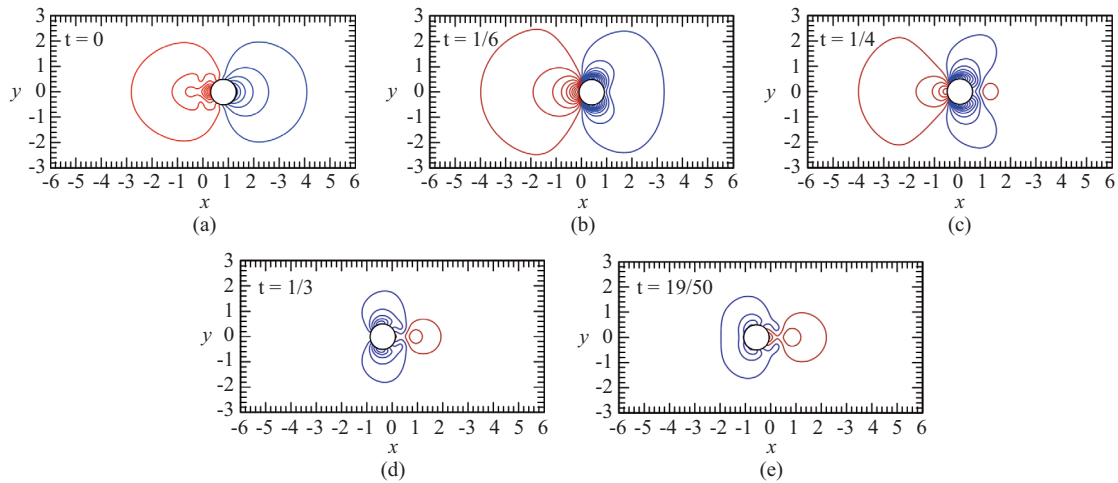


Fig. 9 The resulting isobars with increment $\Delta p = 3$ at (a) $t = 0$ (b) $t = 1/6$ (c) $t = 1/4$ (d) $t = 1/3$ (e) $t = 0.38$.

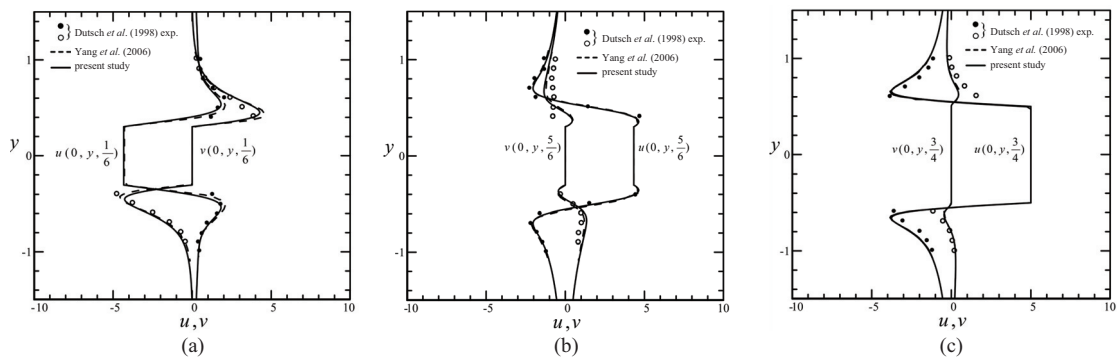


Fig. 10 The velocity profiles along the y -axis at (a) $t = 1/6$ (b) $t = 3/4$ (c) $t = 5/6$.

Figure 11 reveals the resulting streamlines at four representative times. The photographs for this same flow configuration [39] are also shown in Fig. 11 for comparison. Good agreement between the present numerical result and the experimental photographs is observable. This evidences that the implicit virtual boundary method applies to problems dealing with solid of zero thickness as well.

4.4 Flow around the Flapping Wings of a Tethered Dragonfly

Wang and coworkers [40, 41] filmed the wing flapping of a tethered dragonfly at 1500 frames per second. The mean flapping frequency was $f = 33.4$ Hz. Their results included 19 frames from the film sequence for the chord positions of the fore and hind wings at $2/3$ wing-span during one wingbeat (see Fig. 12). This dragonfly was found to beat its forewing and hindwing at different phase, with the hindwing about 22° ahead of the forewing. In the present study, the flow field induced by the flapping wings is computed with the implicit virtual boundary method. The chord length $L = 0.0103m$ and the wing flapping period $t_c = f^{-1}$ are assigned as the reference length and time, respectively. The correspond-

ing reference velocity and Reynolds number are $U_c = fL = 0.344m/s$ and $Re = \rho fL^2/\mu = 275$.

Computation is performed in the range of $-2 \leq x \leq 5.8$ and $-2 \leq y \leq 7$. A moving patch grid system is employed in the region of $0 \leq (x - x_k) \leq 3$ and $0 \leq (y - y_k) \leq 3$, where (x_k, y_k) is the k -th grid point in the global grid system. The reference point (x_k, y_k) is assigned for each time step such that the patch grids always cover both forewing and hindwing with an adequate margin. The computation domain as well as two representative patch grids for frames 1 and 12 is illustrated in Fig. 13. Uniform square grid meshes of $\Delta x = \Delta y = 0.02$ and $\Delta x = \Delta y = 0.06$ are employed in the patch grid system and the global grid system, respectively. The ambient pressure is set as the pressure level $p_\infty = 0$. Static fluid condition $(u, v, p) = (0, 0, 0)$ is assigned on the outer boundary of the global grids. The wings are assumed rigid and of zero thickness. The chord positions of the wings are interpolated in the time coordinate with the method of cubic spline. The time step for the computation is $\Delta t = 1/760$ that corresponds to a CFL number of $\lambda = 3.29$ for the patch grid system. In some particular regions the wings sweep through more than 2 grid points during a single time step. The numerical procedure starts from a flow field of zero velocity with the wings being at frame

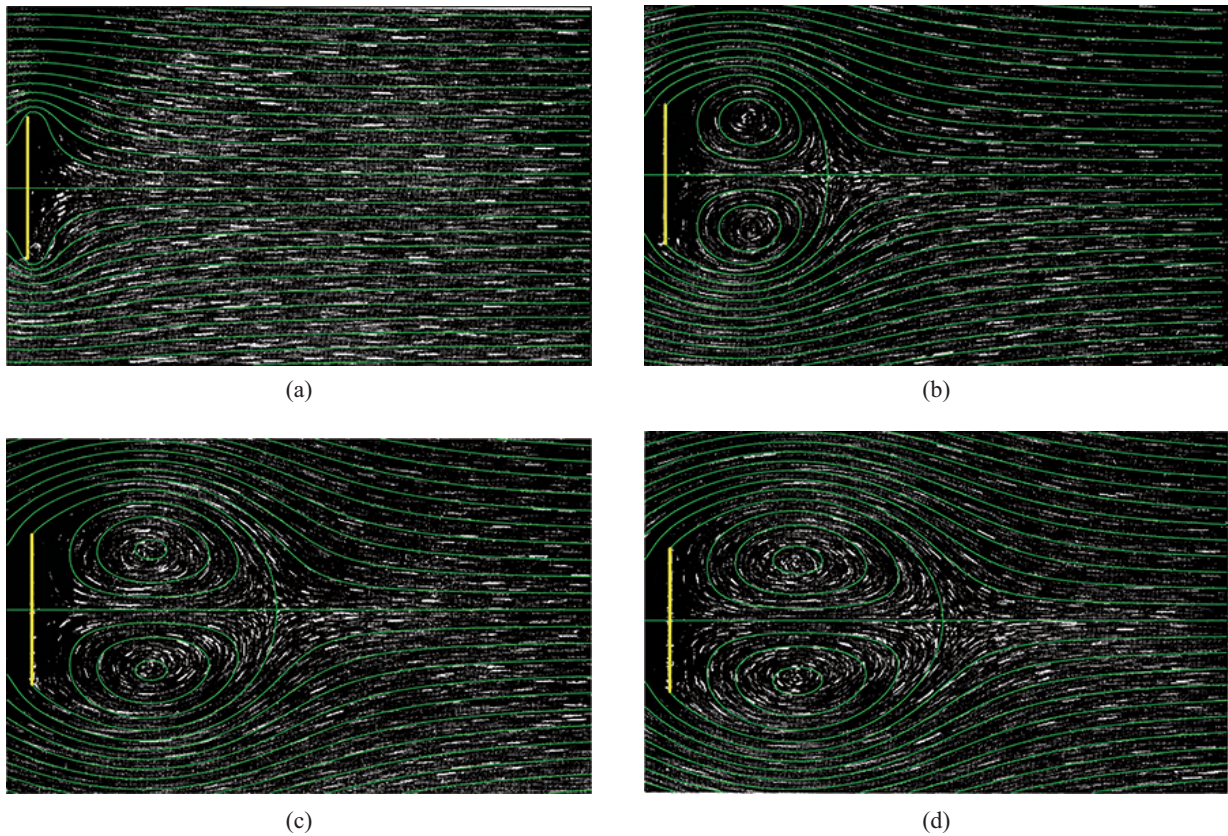


Fig. 11 The resulting streamlines at (a) $t = 0.04$ (b) $t = 1.45$ (c) $t = 2.56$ (d) $t = 3.36$.

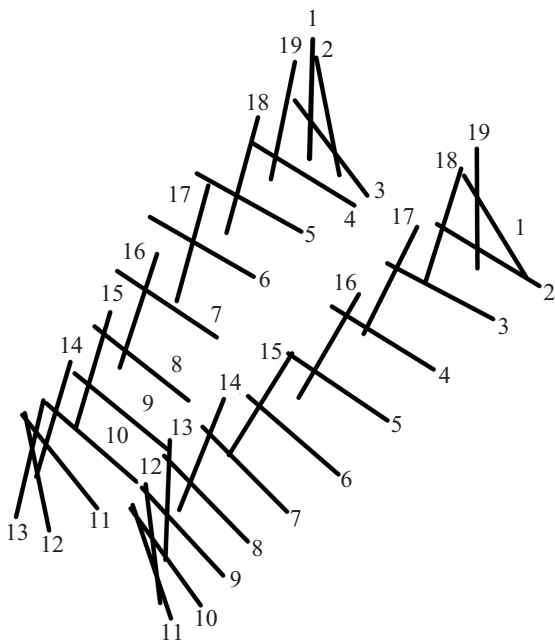


Fig. 12 The chord positions of the fore and hind wings during one wingbeat.

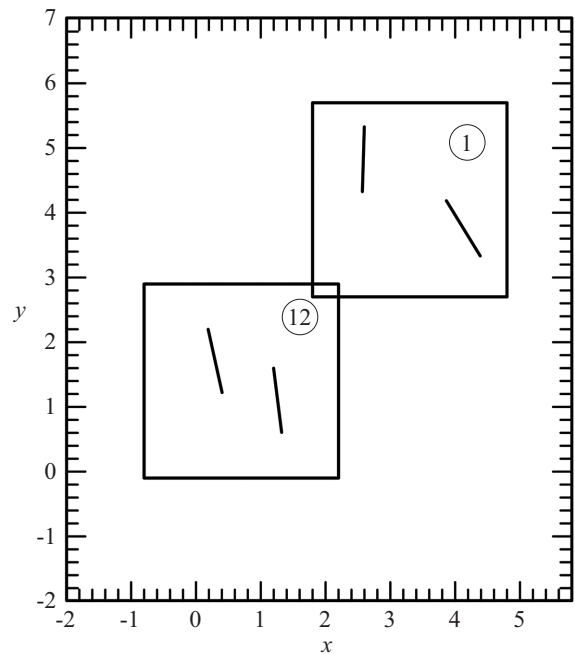


Fig. 13 The computation domain and two representative patch grids.

1. Quasi-steady state solution is reached within 4 wingbeats. The computation is carried out on an Intel Core i5-4200M CPU @ 2.50GHz. It takes 0.04 s for one solution iteration.

Figure 14 shows the numerical results of isobars and velocities at the 19 frames under quasi-steady state. The increment of the isobars is $\Delta p = 10$. Again, the isobars in red and in blue denote positive and negative

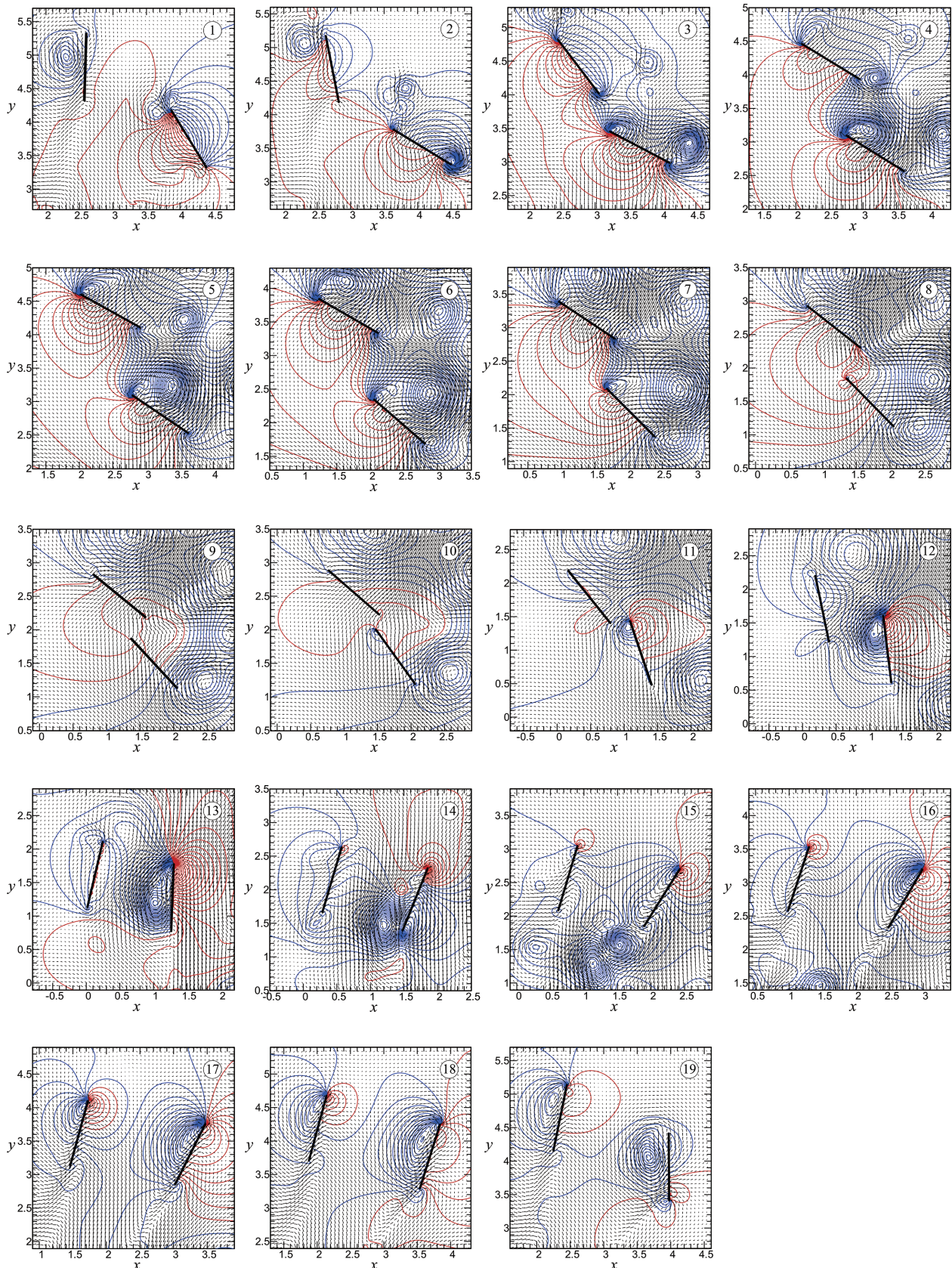


Fig. 14 The resulting isobars ($\Delta p = 10$) and velocity vectors at the 19 frames.

pressures, respectively. The net surface force acting on the forewing and hindwing during a complete cycle of wingbeat is revealed in Fig. 15, where the frame number i corresponds to the dimensionless time $t = (i - 1)/19$.

Frame 20 is identical to frame 1. It is noted from Fig. 12 that the attack angle of the flapping wings is close to 90° at the beginning of the downstroke (see frame 3 or $t = 0.105$ for instance). Under this situation,

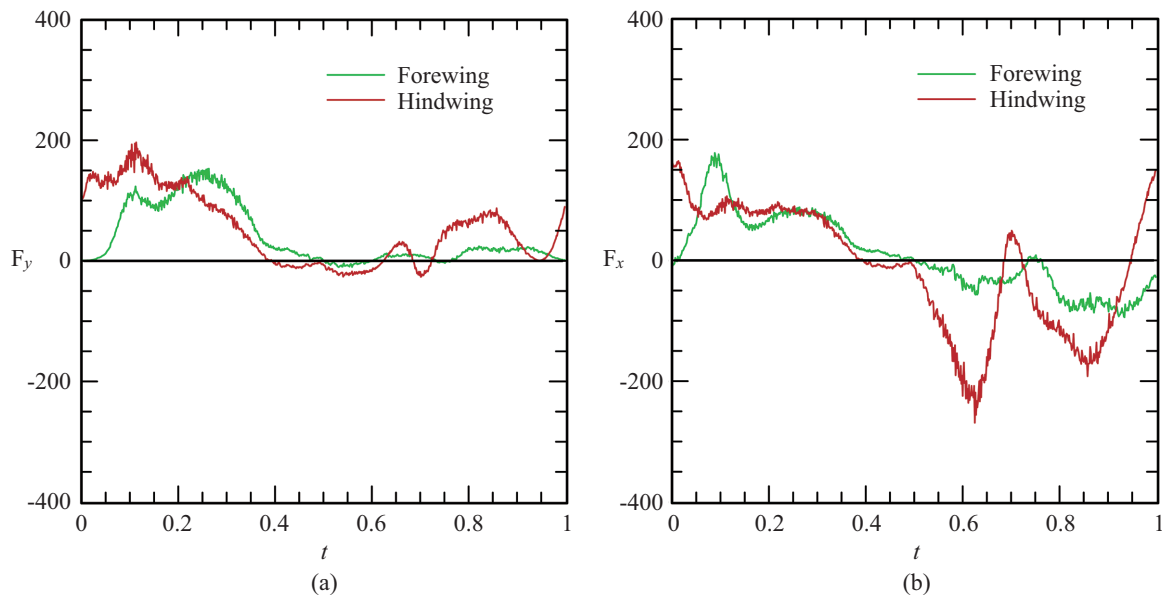


Fig. 15 The net force acting on the forewing and hindwing during a complete cycle of wingbeat (a) vertical component (b) horizontal component.

the vertical force F_y (the vertical components of the resulting force) of both forewing and hindwing reaches the maximum value (see Fig. 15(a)). The vertical force is weakened later at frame 8 or $t = 0.368$ due to the "leakage" of the positive pressure from the downside of the forewing to the upside of the hindwing. Subsequently in the time interval $0.4 \leq t \leq 0.6$ (see frames 9-13), the vertical force is essentially zero when the wings are turning around for the upstroke. From frames 14-15, one sees two large low pressure vortices shedding from the hindwing. In the upstroke the attack angles of the wings are small. Surprisingly, the wings produce upward vertical force even during the upstroke. Moreover, the horizontal force F_x (see Fig. 15(b)) is very strong in the entire wingbeat except for a short time near the end of the downstroke ($0.4 \leq t \leq 0.5$). The back-and-forth horizontal force, together with the upward vertical force, might account for the fact that the dragonfly is trying to escape from being tethered, rather than being hovering. The hindwing seems to offer more force (both vertical and horizontal forces) than the forewing.

4.5 Solution Accuracy and Numerical Stability

Theoretically, the weighting function scheme (6) is a locally analytical formulation. In the conventional NAPPLE algorithm [30], however, the accuracy in space is only of the second-order. This is attributed to the use of the central difference scheme when the continuity equation is discretized in Eq. (13). In the present implicit virtual boundary method, the accuracy would further reduce to the first-order in the vicinity of the solid-fluid interface due to the assumption of the bi-linear function (25)-(27). The major drawback is that the viscosity effect has to be neglected from Eq. (31), and therefore the drag coefficient is slightly under-predicted as revealed in Table 1 when the solid-fluid interface is stationary. Nevertheless, use of the bi-linear function (25)-(27) provides a great numerical stability

during the computation. Numerical stability could be no less important than the order of truncation error that a high-order scheme offers especially when the solid-fluid interface is dynamically moving. On an accelerating boundary, the viscosity effect might not be so significant (see Eq. (31)).

In 1910, Richardson [42] proposed an explicit three-time-level scheme for solving transient heat conduction. The unsteady term was discretized with the standard central difference method such that the scheme is second-order accurate in the time coordinate. Unfortunately, the Richardson method [42] proves to be unconditionally unstable and cannot be used to discretize the unsteady term. This evidences that the accuracy of a numerical scheme in time coordinate cannot be assessed by the order of truncation error alone. Numerical stability should be taken into account. For the case of flow past a stationary cylinder, the CFL number employed in the present computation is $\lambda = \infty$, i.e. the steady-state solution is solved directly. By contrast, Taira and Colonius [36] found that their numerical method is stable up to only $\lambda = 0.46$ for this same case. Similarly, for the case of in-line oscillating cylinder in rest fluid, the CFL number employed in the present computation is $\lambda = 20$. Hence, only 250 time steps are needed in a half period of oscillation as compared to more than 10,000 time steps (due to the CFL restriction $\lambda < 0.5$) required by many previous studies that employed an explicit time-marching scheme at the same spatial resolution. A detailed study on the numerical stability and CFL number can be found in Ref. [26].

5. CONCLUSIONS

The implicit virtual boundary method is proposed in the paper for fluid flow around a moving boundary of irregular shape. Performance of this new numerical

method is examined through four test problems. Based on the numerical results, the following conclusions are drawn.

- (a) The algorithm of the implicit virtual boundary method is very simple, and thus the computer programming is easy.
- (b) The external force imposed at the forcing point is determined numerically. Its value converges stably due to the use of the implicit scheme.
- (c) Use of the path grid system conserves both CPU time and memory storage significantly.
- (d) The tethered dragonfly in section 4.4 is trying to escape from being tethered, rather than being hovering.

ACKNOWLEDGMENTS

The authors wish to express their appreciation to the Ministry of Science and Technology Taiwan for the financial support of this work through the contract MOST 102-2221-E-007-070. Cyue acknowledges the scholarship sponsored by CoreTech System (Moldex3D).

REFERENCES

1. Chen, Y. C., Liao, Y. J., Tseng, S. C. and Giacomini, A. J., "Core Deflection in Plastics Injection Molding: Direct Measurement, Flow Visualization and 3D Simulation," *Polymer-Plastics Technology and Engineering*, **50**, pp. 863-872 (2011).
2. Sun, M. and Lan, S. L., "A Computational Study of the Aerodynamics Forces and Power Requirements of Dragonfly (*Aeschna juncea*) Hovering," *Journal of Experimental Biology*, **207**, pp. 1887-1901 (2004).
3. Wang, J. K. and Sun, M., "A Computational Study of the Aerodynamics and Forewing-Hindwing Interaction of a Model Dragonfly in Forward Flight," *Journal of Experimental Biology*, **208**, pp. 3785-3804 (2005).
4. Chen, H. C., "CFD Simulation of Directional Short-Crested Waves on a Jack-Up Structure," *International Journal of Offshore and Polar Engineering*, **23**, pp. 38-45 (2013).
5. Peskin, C. S., "Flow Pattern around Heart Valves: a Numerical Method," *Journal of Computational Physics*, **10**, pp. 252-271 (1972).
6. Peskin, C. S., "The Immersed Boundary Method," *Acta Numerica*, pp. 459-571 (2002).
7. Mittal, R. and Iaccarino, G., "Immersed Boundary Methods," *Annual Review of Fluid Mechanics*, **37**, pp. 239-261 (2005).
8. Goldstein, D., Handler, R. and Sirovich, L., "Modeling a No-Slip Flow with External Force Field," *Journal Computational Physics*, **105**, pp. 354-366 (1993).
9. Goldstein, D., Handler, R. and Sirovich, L., "Direct Numerical Simulation of Turbulent Flow over a Modeled Riblet Covered Surface," *Journal of Fluid Mechanics*, **302**, pp. 333-376 (1995).
10. Saiki, E. M. and Biringen, S., "Numerical Simulation of a Cylinder in Uniform Flow: Application of a Virtual Boundary Method," *Journal Computational Physics*, **123**, pp. 450-465 (1996).
11. Lai, M. C. and Peskin, C. S., "An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity," *Journal Computational Physics*, **160**, pp. 705-719 (2000).
12. Lima, E., Silva, A. L. F., Silveira-Neto, A. and Damasceno, J. J. R., "Numerical Simulation of Two-Dimensional Flows over a Circular Cylinder Using the Immersed Boundary Method," *Journal Computational Physics*, **189**, pp. 351-370 (2003).
13. Mohd-Yusof, J., "Combined Immersed-Boundary/B-Spline Methods for Simulations of Flow in Complex Geometries," *Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford University*, pp. 317-327 (1997).
14. Fadlun, E. A., Verzicco, R., Orlandi, P. and Mohd-Yusof, J., "Combined Immersed-Boundary Methods for Three Dimensional Complex Flow Simulations," *Journal of Computational Physics*, **161**, pp. 35-60 (2000).
15. Kim, J., Kim, D. and Choi, H., "An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries," *Journal of Computational Physics*, **171**, pp. 132-150 (2001).
16. Tseng, Y. H. and Ferziger, J. H., "A Ghost-Cell Immersed Boundary Method for Flow in Complex Geometry," *Journal of Computational Physics*, **192**, pp. 593-623 (2003).
17. Balaras, E., "Modeling Complex Boundaries Using an External Force Field on Fixed Cartesian Grids in Large-Eddy Simulations," *Computers and Fluids*, **33**, pp. 375-404 (2004).
18. Gilmanov, A. and Sotiropoulos, F., "A Hybrid Cartesian/Immersed Boundary Method for Simulating Flows with 3D, Geometrically Complex, Moving Bodies," *Journal of Computational Physics*, **207**, pp. 457-492 (2005).
19. Kim, D. and Choi, H., "Immersed Boundary Method for Flow around an Arbitrarily Moving Body," *Journal of Computational Physics*, **212**, pp. 662-680 (2006).
20. Yang, J. and Balaras, E., "An Embedded-Boundary Formulation for Large-Eddy Simulation of Turbulent Flows Interacting with Moving Boundaries," *Journal of Computational Physics*, **215**, pp. 12-24 (2006).
21. Su, S. W., Lai, M. C. and Lin, C. A., "A Simple Immersed Boundary Technique for Simulating Complex Flows with Rigid Boundary," *Computers and*

- Fluids*, **36**, pp. 313-324 (2007).
22. Zhang, N. and Zheng, Z. C., "An Improved Direct-Forcing Immersed-Boundary Method for Finite Difference Applications," *Journal of Computational Physics*, **221**, pp. 250-268 (2007).
 23. Choi, J. I., Oberoi, R. C., Edwards, J. R. and Rosati, J. A., "An Immersed Boundary Method for Complex Incompressible Flows," *Journal of Computational Physics*, **224**, pp. 757-784 (2007).
 24. Ghias, R., Mittal, R. and Dong, H., "A Sharp Interface Immersed Boundary Method for Compressible Viscous Flows," *Journal of Computational Physics*, **225**, pp. 528-553 (2007).
 25. Liao, C. C., Chang, Y. W., Lin, C. A. and McDonough, J. M., "Simulating Flows with Moving Rigid Boundary Using Immersed-Boundary Method," *Computers and Fluids*, **39**, pp. 152-167 (2010).
 26. Lee, S. L., "A New Numerical Formulation for Parabolic Differential Equations under the Consideration of Large Time Steps," *International Journal for Numerical Methods in Engineering*, **26**, pp. 1541-1549 (1988).
 27. Lee, S. L., "Weighting Function Scheme and Its Application on Multidimensional Conservation Equations," *International Journal of Heat and Mass Transfer*, **32**, pp. 2065-2073 (1989).
 28. Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington (1980).
 29. Lee, S. L., "A Strongly-Implicit Solver for Two-Dimensional Elliptic Differential Equations," *Numerical Heat Transfer B*, **16**, pp. 161-178 (1989).
 30. Lee, S. L. and Tzong, R. Y., "Artificial Pressure for Pressure-Linked Equation," *International Journal of Heat and Mass Transfer*, **35**, pp. 2705-2716 (1992).
 31. Peng, Y. F., Mittal, R., Sau, A. and Hwang, R. R., "Nested Cartesian Grid Method in Incompressible Viscous Fluid Flow," *Journal of Computational Physics*, **229**, pp. 7072-7101 (2010).
 32. Majumdar, S., Iaccarino, G. and Durbin, P., "RNS Solvers with Adaptive Structured Boundary Non-Conforming Grids," *Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford University*, pp. 353-366 (2001).
 33. Coutanceau, M. and Bouard, R., "Experimental Determination of the Main Features of the Viscous Flow in the Wake of a Circular Cylinder in Uniform Translation. Part 1. Steady Flow," *Journal of Fluid Mechanics*, **79**, pp. 231-256 (1977).
 34. Dennis, S. C. R. and Chang, G., "Numerical Solutions for Steady Flow Past a Circular Cylinder at Reynolds Number up to 100," *Journal of Fluid Mechanics*, **42**, pp. 471-489 (1970).
 35. Linnick, M. N. and Fasel, H. F., "A High-Order Immersed Interface Method for Simulating Unsteady Incompressible Flows on Irregular Domains," *Journal of Computational Physics*, **204**, pp. 157-192 (2005).
 36. Taira, K. and Colonius, T., "The Immersed Boundary Method: A Projection Approach," *Journal of Computational Physics*, **225**, pp. 2118-2137 (2007).
 37. Lee, J. and You, D., "An Implicit Ghost-Cell Immersed Boundary Method for Simulations of Moving Body Problems with Control of Spurious Force Oscillations," *Journal of Computational Physics*, **233**, pp. 295-314 (2013).
 38. Dutsch, H., Dorst, F., Becker, S. and Lienhart, H., "Low-Reynolds-Number Flow around an Oscillating Circular Cylinder at Low Keulegan-Carpenter Number," *Journal of Fluid Mechanics*, **360**, pp. 249-271 (1998).
 39. Taneda, S. and Honji, H., "Unsteady Flow past a Flat Plate Normal to the Direction of Motion," *Journal of the Physical Society of Japan*, **30**, pp. 262-272 (1971).
 40. Wang, Z. J., "Dissecting Insect Flight," *Annual Review of Fluid Mechanics*, **37**, pp. 183-210 (2005).
 41. Wang, Z. J. and Russell, D., "Effect of Forewing and Hindwing Interactions on Aerodynamic Forces and Power in Hovering Dragonfly Flight," *Physical Review Letters*, **99**, 148101 (2007).
 42. Richardson, L. F., "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam," *Philosophical Transactions of the Royal Society A*, **210**, pp. 307-357 (1910).

(Manuscript received October 19, 2016, accepted for publication April 14, 2017.)