INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:1195–1213
Published online 19 June 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1237

A new modification of the immersed-boundary method for simulating flows with complex moving boundaries

Jian Deng[‡], Xue-Ming Shao and An-Lu Ren[*,†]

Department of Mechanics, Institute of Fluid Engineering, Zhejiang University, Hangzhou 310027, People's Republic of China

SUMMARY

In this paper, a new immersed-boundary method for simulating flows over complex immersed, moving boundaries is presented. The flow is computed on a fixed Cartesian mesh and the solid boundaries are allowed to move freely through the mesh. The present method is based on a finite-difference approach on a staggered mesh together with a fractional-step method. It must be noted that the immersed boundary is generally not coincident with the position of the solution variables on the grid, therefore, an appropriate strategy is needed to construct a relationship between the curved boundary and the grid points nearby. Furthermore, a momentum forcing is added on the body boundaries and also inside the body to satisfy the no-slip boundary condition. The immersed boundary is represented by a series of interfacial markers, and the markers are also used as Lagrangian forcing points. A linear interpolation is then used to scale the Lagrangian forcing from the interfacial markers to the corresponding grid points nearby. This treatment of the immersed-boundary is used to simulate several problems, which have been validated with previous experimental results in the open literature, verifying the accuracy of the present method. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: immersed-boundary method; momentum forcing; fractional-step method; finite-difference method

1. INTRODUCTION

How to handle complex geometries has been one of the main challenges in computational fluid dynamics because most engineering problems have complex geometries. So far, two traditional techniques have been developed: using the unstructured grid for finite volume method and the coordinate transformations and mapping techniques for finite difference method. The use of

[*] Correspondence to: A.-L. Ren, Department of Mechanics, Institute of Fluid Engineering, Zhejiang University, Hangzhou 310027, People's Republic of China.
[†] E-mail: renanlu@zju.edu.cn
[‡] E-mail: zjudengjian@zju.edu.cn

Contract/grant sponsor: National Natural Science Foundation of China; contract/grant numbers: 10272094, 10472104

Received 4 October 2005
Revised 28 February 2006
Accepted 19 March 2006

coordinate transformations and mapping techniques is possible but requires a highly accurate way of calculating the transformation Jacobians, and even for simple geometries, generating a good-quality body-conformal grid can be an iterative process requiring significant devotion of the time. As the geometry becomes more complicated, the task of generating an acceptable grid becomes increasingly difficult. The unstructured grid approach is inherently better suited for complex geometries, but here, grid quality can deteriorate with increasing complexity in the geometry. As an alternative solution, the immersed boundary method (IBM) was then developed, and the main advantages of this method are memory and CPU savings and easy grid generation. Grid complexity and quality are not significantly affected by the complexity of the geometry. Even moving-boundary problems can be handled with the immersed-boundary method without regenerating grids in time, unlike the unstructured grid and body-fitted grid methods.

Peskin [1] developed a method that represents a body within a flow field via a forcing term added to the governing equations. If carefully selecting the points where the forcing term is applied, the forcing term can simulate the effect of the body on the flow, allowing the modelling of an arbitrary-shaped boundary within a Cartesian computational box without the necessity of mapping. Peskin [2–5] successfully implemented this immersed boundary technique to model moving boundaries in heart valve simulations. The main advantage of this scheme lies in its ability to model the material properties of the body and movement of the boundaries. However, numerical stiffness of most moving boundary problems restricts the explicit definition of the forcing term in Peskin's method to small time steps. Goldstein *et al.* [6] developed a virtual boundary method (VBM) that employed a forcing term governed by a feedback loop. However, the feedback forcing induced spurious oscillations, restricted the computational time step associated with numerical stability. For instance, Goldstein [6] used a very small time step equivalent to a CFL (Courant–Friedrichs–Lewy) number of $O(10^{-3}\text{–}10^{-2})$ when they simulated the start-up flow around a circular cylinder. As reviewed by Mittal *et al.* [7], this category of methods is termed 'continuous forcing approach', in which the forcing is incorporated into the continuous equations before discretization. The continuous forcing approach is very attractive for simulating the flow with elastic boundaries, and has a sound physical basis. However, using this approach for flows with rigid bodies poses some challenges associated with the fact that the forcing terms are generally not well behaved in the rigid limit.

Recently, Mohd-Yusof [8] suggested a different approach to evaluating the momentum forcing in a spectral method, and his method does not require a small computational time step, which is an important advantage over previous methods. Fadlun *et al.* [9] applied the approach of Mohd-Yusof [8] to a finite-difference method on a staggered grid and showed that the discrete-time forcing suggested by Mohd-Yusof [8] is more efficient than feedback forcing for three-dimensional flows. The discrete form of the forcing can be written as

$$f_{i,j} = \frac{u_\Psi - u_{i,j}^n}{\Delta t} - \text{RHS} \tag{1}$$

where Δt is the time step, and RHS contains the convective, viscous and pressure terms of the governing equations. According to Fadlun *et al.* [9], the velocity at the first grid point external to the body is obtained by linearly interpolating the velocity at the second grid point, which is obtained by directly solving the Navier–Stokes equations, and the velocity at the body surface. The method is straightforward, second-order accurate, and works well for bodies that are largely aligned with the grid lines. For geometrically complex immersed boundaries, however, the choice of the reconstruction direction may not be unique, because often more than one grid line passing through a near-boundary node may intersect the boundary. A multi-dimensional scheme aimed at removing this arbitrary element in the implementation of hybrid formulations was proposed by Kim *et al.* [10]. This scheme uses a bilinear reconstruction procedure, which is reduced to one-dimension when there are no available points in the vicinity of the boundary to support the two-dimensional stencil. Fadlun *et al.* [9] and Kim *et al.* [10] treat immersed boundaries with reconstruction procedure that reevaluate the velocities on grid points (Eulerian points) near the boundaries, and the added momentum forcing is calculated on these grid points. These reconstruction treatments have also been applied by Tseng *et al.* [11], Gilmanov *et al.* [12] and Balaras [13]. These methods fall into a second category, which is termed 'discrete forcing approach'. The forcing procedure is intimately connected to the details of the discretization approach and practical implementation is not as straightforward as the continuous forcing approach. However, discrete forcing enables a sharp representation of the IB, and this is desirable, especially at higher Reynolds numbers.

The discrete forcing approach is more accurate and efficient, and the key problem for these studies is the interpolation approach described in Reference [9]. Other revised interpolation schemes have also been developed [14]. The immersed-boundary method is in general accomplished by modifying the computational stencil near the immersed boundary to impose the boundary condition on the IB, and to maintain a 'sharp' interface, where the local accuracy of the solution assumes greater importance, and the spreading of the effect of the IB introduced by the smooth force distribution function that often used in the continuous forcing approach is less desirable. The trend for the development of IBM is more complicated interpolation schemes and more grid points involved for the calculation of the forcing term, with the result that the modification of the computational stencil near the IB becomes more and more complicated.

If the IB is simple and has a straightforward mathematical expression, then the interpolation procedure is relatively easy, we can find the grid points for interpolation directly. But if the immersed boundary, especially the three-dimensional immersed surface of the body, is expressed by a set of discrete points, then the reconstruction procedure near the IB using traditional method is difficult and the choice of the reconstruction direction is difficult to determine. This is also the reason for that the using of the IBM is limited. We will develop a new version of IBM, which has a robust ability to deal with arbitrary and complex configurations, and at the same time can keep the overall accuracy of the scheme. To verify the accuracy of the present method, we will simulate four different flow problems.

2. NUMERICAL METHODS

2.1. Governing equations

Viscous and incompressible flows in a Cartesian square domain containing immersed boundaries can be modelled by the following Navier–Stokes equations, in which $\mathbf{F}(i,j)$ is the Eulerian forcing on the grid points, and this forcing is not equal to zero only over the immersed boundary

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \boldsymbol{\nabla})\mathbf{V} = -\boldsymbol{\nabla} P + \frac{1}{Re}\nabla^2 \mathbf{V} + \mathbf{F} \tag{2}$$

$$\boldsymbol{\nabla} \cdot \mathbf{V} = 0 \tag{3}$$

2.2. *Immersed boundary treatment*

A general treatment to mimic the immersed boundary (IB) is to replace the IB with a set of discrete control points $\mathbf{x}_k$ (Lagrangian points). The Lagrangian forcing $\mathbf{f}(\mathbf{x}_k)$ are calculated over these control points, and are spread to the nearby Cartesian grid points by a certain distribution function (in general a discrete delta function), so that it yields non-zero forcing term $\mathbf{F}(i,j)$ at grid points near the interface. After the spreading process, the internal boundaries have been well handled and the fluid equation can be solved by a general finite difference method.

The calculation method for the Lagrangian forcing is developed from the Physical Virtual Model (PVM) originally proposed by Silva *et al.* [15]. The main idea we follow PVM is that the added forcing has been calculated explicitly, and is broken into different terms, then each term has a clearly physical meaning. Furthermore, PVM is based on the calculation of the forcing field over a sequence of pre-assigned Lagrangian points, which represent the immersed boundary. Since only a small part of these points are identified to have a direct relationship with the nearby grid points that have been used, we have no need to calculate the forcing on all control points.

2.3. *Added force field*

The Lagrangian forcing is calculated based only on the momentum equation. All the Navier–Stokes terms are calculated over the Lagrangian points. Therefore the forcing $\mathbf{f}(\mathbf{x}_k)$ should be expressed by

$$\mathbf{f}(\mathbf{x}_k) = \mathbf{f}_{\mathrm{a}}(\mathbf{x}_k) + \mathbf{f}_{\mathrm{i}}(\mathbf{x}_k) + \mathbf{f}_{\mathrm{v}}(\mathbf{x}_k) + \mathbf{f}_{\mathrm{p}}(\mathbf{x}_k) \tag{4}$$

The different terms that compose Equation (4) are here taken apart as acceleration forcing $\mathbf{f}_{\mathrm{a}}(\mathbf{x}_k)$, inertial forcing $\mathbf{f}_{\mathrm{i}}(\mathbf{x}_k)$, viscous forcing $\mathbf{f}_{\mathrm{v}}(\mathbf{x}_k)$ and pressure forcing $\mathbf{f}_{\mathrm{p}}(\mathbf{x}_k)$. These forcing components are represented by

$$\mathbf{f}_{\mathrm{a}}(\mathbf{x}_k) = \frac{\partial \mathbf{V}}{\partial t}(\mathbf{x}_k) \tag{5}$$

$$\mathbf{f}_{\mathrm{i}}(\mathbf{x}_k) = (\mathbf{V}\cdot\boldsymbol{\nabla})\mathbf{V}(\mathbf{x}_k) \tag{6}$$

$$\mathbf{f}_{\mathrm{v}}(\mathbf{x}_k) = -\frac{1}{Re}\nabla^2\mathbf{V}(\mathbf{x}_k) \tag{7}$$

$$\mathbf{f}_{\mathrm{p}}(\mathbf{x}_k) = \boldsymbol{\nabla}P(\mathbf{x}_k) \tag{8}$$

The velocity and pressure spatial derivatives in terms described by Equations (5)–(8) must be evaluated over the Lagrangian point $\mathbf{x}_k$ on the immersed boundary and the surrounding virtual points 1–4 as shown in Figure 1, in which the distances between the points $\mathbf{x}_k$ and 1, $\mathbf{x}_k$ and 2, $\mathbf{x}_k$ and 3, $\mathbf{x}_k$ and 4 are h (the mesh size). One of the possible ways to obtain the velocity and pressure values on these five points is to interpolate from the surrounding grid points that enclose the Lagrangian point or virtual points in a grid box. In our procedure, a standard bilinear interpolation is used. When calculating the acceleration force component denoted by Equation (5), we should notice that the fluid velocity at next time step must

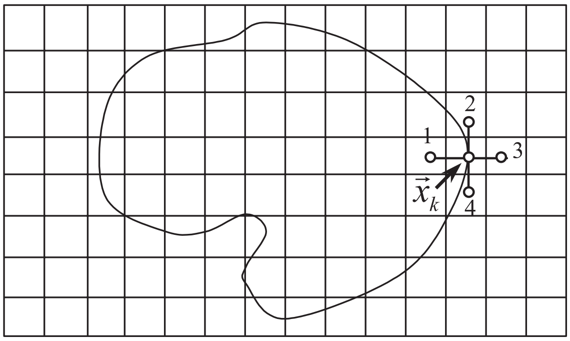

Figure 1. Illustration of a Lagrangian point with surrounding virtual points.

be equal to the no-slip boundary's velocity. Furthermore, in three-dimensional problems, six virtual points should be created instead.

2.4. Interface description

An important step in a proper distribution of the forcing function is the description of the topology of the boundary. We use a method similar to that proposed by Udaykumar *et al.* [16, 17], in which the fluid/solid interface is tracked as a discontinuity. This method has been also adopted by Balaras [13] recently. This interface tracking methodology has been suggested to be very robust and has been applied to a variety of problems in the context of a Cartesian grid method. In the following paragraphs, new strategies, based on this methodology, will be introduced to facilitate the present implementation of IBM, especially for three-dimensional cases.

An immersed solid boundary, Ψ, of arbitrary shape (open or closed) is represented by a series of interfacial markers defined by the global coordinates $\mathbf{X}(k)$ or the local coordinates $\mathbf{X}_b(k)$, whose origin point is located at the geometrical centre of the immersed body, as shown in Figure 2. The markers are also used as the Lagrangian forcing points, as have been mentioned above in Section 2.2. The markers are compactly spaced with spacing much smaller than the local grid size. The sequence of the markers is defined in a way that as the observer moves toward increasing the serial number k, the solid is always to the left. Referring to Figure 2, the global coordinates for each marker point is $\mathbf{X}(k) = x(k) + i \cdot y(k)$, where i denotes the square root of -1. Based on this definition, the normal from any point on the interface to the fluid can be computed in a straightforward manner through the two neighbours $(k-1)$ and $(k+1)$. The normal vector is deduced as

$$\begin{aligned} \mathbf{T}_k &= x(k+1) - x(k-1) + i \cdot (y(k+1) - y(k-1)) \\ \mathbf{N}_k &= \mathbf{T}_k \cdot (-i) = y(k+1) - y(k-1) + i \cdot (x(k-1) - x(k+1)) \end{aligned} \tag{9}$$

where $\mathbf{T}_k$ denotes the vector at the tangent on the interface, and the normal vector can be obtained by rotating the tangent vector clockwise. In three-dimensional problems, the sequence of the markers is defined by two sets of serial number k and l, with the vector $\mathbf{l} \times \mathbf{k}$ pointing

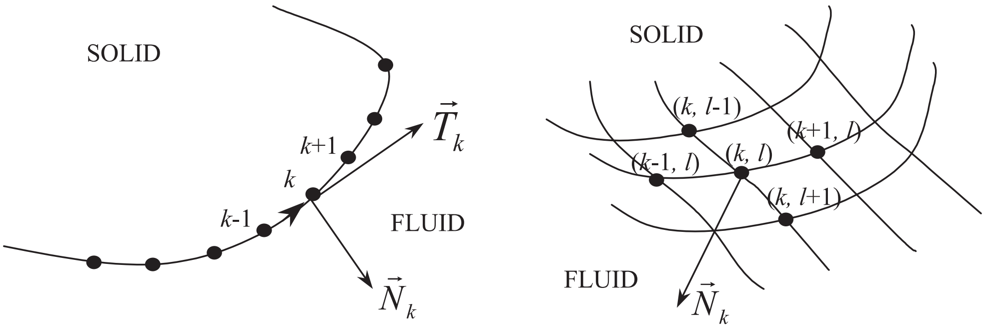

Figure 2. Description of the boundary topology with interface markers, left: 2D problem; right: 3D problem.

to the fluid phase (Figure 2). The normal vector is deduced as

$$\mathbf{N}_k = \mathbf{l}\times\mathbf{k}$$

$$= \begin{vmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \\ x(k,l+1)-x(k,l-1) & y(k,l+1)-y(k,l-1) & z(k,l+1)-z(k,l-1) \\ x(k+1,l)-x(k-1,l) & y(k+1,l)-y(k-1,l) & z(k+1,l)-z(k-1,l) \end{vmatrix} \tag{10}$$

where $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are the base vectors in Cartesian coordinates, in three base directions, respectively.

2.5. Identification of forcing points

Given the above interface description, for each grid point, (i,j), near the interface, we can identify the closest marker point, (k), through a search process. In order to know which of these grid points belong to the fluid and which to the solid phase, we create a vector $\boldsymbol{\lambda}$ from the maker point to a corresponding grid point. To associate this vector with the normal vector $\mathbf{N}_k$ at the marker point, the scalar product $\delta=\boldsymbol{\lambda}\cdot\mathbf{N}_k$ then can be evaluated. If $\delta > 0$, then the point belongs to the fluid phase, whereas if $\delta \leqslant 0$ it belongs to the solid phase. If the point belongs to the fluid phase and the distance between the grid point and the corresponding marker point is less than h, the grid point is tagged with a $+1$ flag. Whereas, if the point belongs to the solid phase, the grid point is tagged with a -1 flag. All other points have the default, zero flag. There, $h=\sqrt{(\mathrm{d}x)^2+(\mathrm{d}y)^2+(\mathrm{d}z)^2}$, denotes the diagonal length of the grid box. Then

For the grid point tagged with a $+1$ flag, a linear interpolation $\mathbf{F}(i,j)=(1-\mathrm{d}s/h)\mathbf{f}(x_k)$ is used to scale the Lagrangian forcing on a related marker to this grid point, where $\mathrm{d}s$ is the distance between the grid point with the related marker point, as shown in Figure 3. This distribution function looks very simple, however, this treatment is proved to be accurate and robust in dealing with complex configurations with moving boundaries. The added forcing

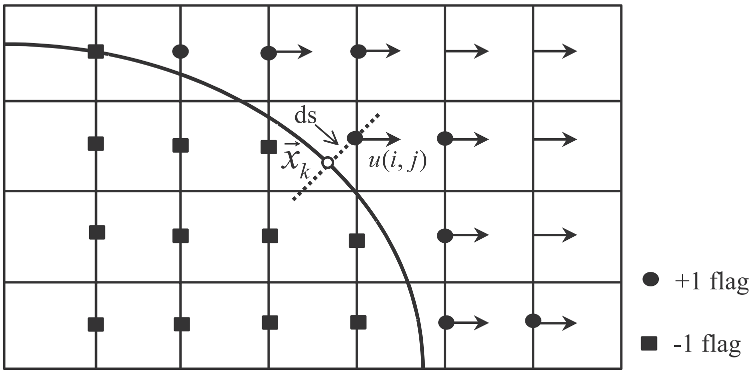

Figure 3. Distribution of the forcing from the immersed boundary to the grid points.

is also calculated inside the body, and this treatment brings the solution to the convergence more quickly.

2.6. The Cartesian grid solver

For the solution of Equations (2) and (3) a second-order projection method is employed. The diffusion terms are advanced by the Crank–Nicolson scheme and the remaining terms are advanced explicitly using an Adams–Bashforth scheme. In the framework of a two-step time-splitting method, first, a provisional value of the velocity field, which is not divergence free, can be obtained

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \left[\frac{3}{2}H(\mathbf{u}^n) - \frac{1}{2}H(\mathbf{u}^{n-1})\right] = -Gp^{n+1} + \frac{1}{2Re}L(\mathbf{u}^n + \mathbf{u}^{n+1}) + \mathbf{F}^{n+1} \quad \text{in } \Omega \tag{11}$$

$$\mathbf{u}^* = \mathbf{b}^{n+1} \quad \text{on } \Gamma \tag{12}$$

where Ω is the computational domain and Γ is the boundary of Ω. $\mathbf{b}^{n+1}$ are the values of the velocities on Γ and $\mathbf{b}^{n+1}$ is given. H is the discrete advection operator, G the discrete gradient, L the Laplace operator. $\mathbf{F}^{n+1}$ is a forcing term that must be determined in advance in a way that $\mathbf{u}^{n+1}$ satisfies the desired boundary conditions on an arbitrary boundary immersed in Ω. The provisional velocity field $\mathbf{u}^*$ can be split into a solenoid field $\mathbf{u}^{n+1}$, which is the final unknown, and the gradient of a scalar

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi \tag{13}$$

$$p^{n+1} = p^n + \phi - \frac{\Delta t}{2Re}\nabla^2 \phi \tag{14}$$

and

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad \text{in } \Omega \tag{15}$$

$$\mathbf{n} \cdot \mathbf{u}^{n+1} = \mathbf{n} \cdot \mathbf{b}^{n+1} \quad \text{on } \Gamma \tag{16}$$

The projection is realized by solving the following Poisson equation:

$$\nabla^2 \phi = \frac{\nabla \mathbf{u}^*}{\Delta t} \quad \text{in } \Omega \tag{17}$$

Boundary conditions for this equation can be derived from the component of Equation (13) normal to the boundary. For the case of a Dirichlet boundary, a consistent boundary condition for the pressure correction ϕ can be obtained introducing (12) and (16) into (13)

$$\nabla \phi \cdot \mathbf{n} = 0 \quad \text{on } \Gamma \tag{18}$$

The Poisson equation (17) for the pressure correction is solved using pre-conditioned conjugate gradient method [18]. All spatial derivatives are approximated with second-order central differences on a staggered grid.

3. NUMERICAL TESTS

3.1. A circular cylinder immersed in a lid-driven cavity

In the original approach by Fadlun *et al.* [9], the velocity components at the first grid point off the IB are determined using a linear interpolation formula rather than the discretized Navier–Stokes equation. The method is equivalent to assuming a one-dimensional linear velocity profile near the boundary. Other interpolation methods have also been developed, for example, revised linear interpolation method (RLIM), quadratic interpolation method (QIM), and quadratic + momentum interpolation method (QMIM), as concluded in Reference [14]. Although the effect of the local pressure gradient has been accounted for by using a few additional corrections, the pressure field is still discontinued, therefore, the local error can be accumulated in the pressure field near the IB. This drawback is especially not suited for moving boundary problems. In this paper, we actually assume a one-dimensional linear added forcing profile near the boundary, and the pressure near the IB is coupled in by explicitly calculating the added forcing.

In order to investigate the accuracy of different IB treatments, we present an analysis of the variation of solution error with grid refinement. This study is performed for the case where a circular cylinder is immersed in a lid-driven cavity. The cylinder of diameter $D = 0.5$ units is placed at the centre of the cavity of dimension 1.0 units. The flow for this immersed boundary problem is simulated using our general solver integrating with the revised linear interpolation method (RLIM) [14], and the new treatment as well, with the Reynolds number (with respect to the dimension of cavity) $Re = 10$. The flow field can be visualized from Figure 4(a), and the y-component of added forcing in shown in Figure 4(b). The following sequence of grid sizes is employed in performing the error analysis: 34×34, 68×68, 130×130, and 258×258 grid points. In the absence of an exact analytical solution to this flow problem, the results on the 258×258 mesh is taken to be the 'exact' solution in order to obtain the error distribution for each of the coarser meshes. Both the L_1 and L_2 norm of the global error and the maximum error are evaluated. The expressions for L_1 and L_2 norm are

$$\varepsilon_1 = \frac{1}{N_x N_y} \sum_{j=1}^{N_x N_y} |\Phi_j^{\text{numerical}} - \Phi_j^{\text{exact}}| \quad \text{and} \quad \varepsilon_2 = \left(\frac{1}{N_x N_y} \sum_{j=1}^{N_x N_y} (\Phi_j^{\text{numerical}} - \Phi_j^{\text{exact}})^2 \right)^{1/2} \tag{19}$$

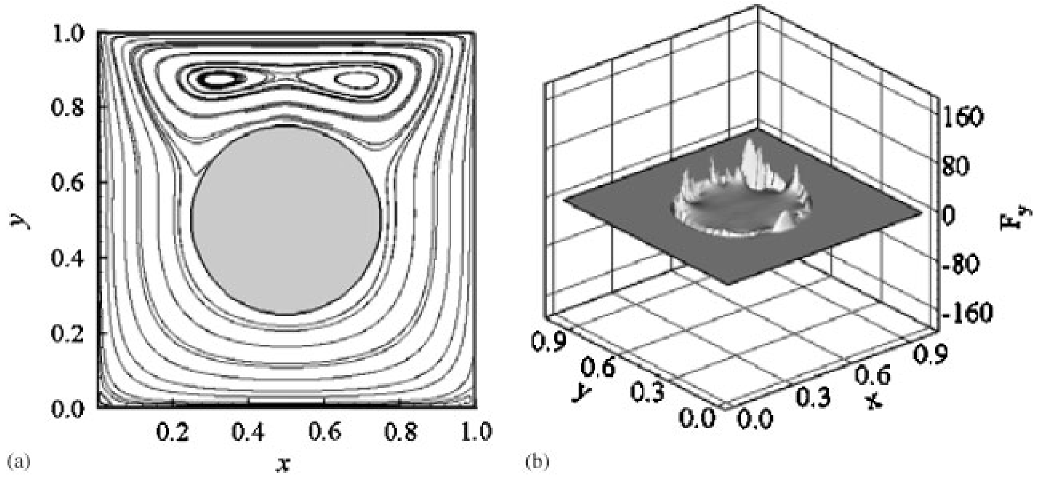

Figure 4. (a) Streamlines for a cylinder in a lid-driven cavity computed on the 130×130 mesh; and (b) forcing field of F_y for 130×130 mesh.

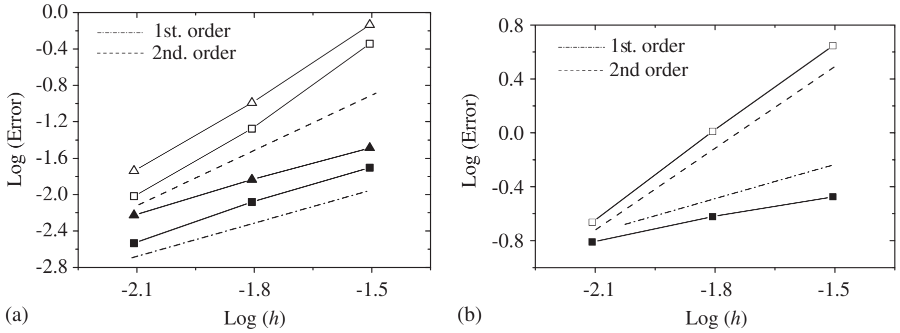

Figure 5. Norm error and maximum error in the u-component of velocities: (a) L_1 norm of the global error: –■–, RLIM, –□–, Present method; L_2 norm of the global error: –▲–, RLIM, –△–, Present method; and (b) maximum error: –■–, RLIM, –□–, Present method.

We show a log–log plot of the errors in velocity component u versus mesh size h in Figure 5, and the errors in pressure in Figure 6, respectively. Also shown are two lines with a slope of -2 and a slope of -1 that correspond to second-order accurate convergence and first-order accurate convergence, respectively. The plot clearly shows that the IBM based on the RLIM shows first-order accuracy in space, whereas the IBM based on the present treatments shows second-order accuracy both in velocity field and pressure field. This test therefore proves that the current approach results in a solver which is second-order accurate. However, we note that exact second-order-accuracy is not expected in this test primarily because the errors are not computed based on an exact solution. The results suggest that even a simple distribution of the Lagrange forcing over the IB can lead to a second-order spatial accuracy.

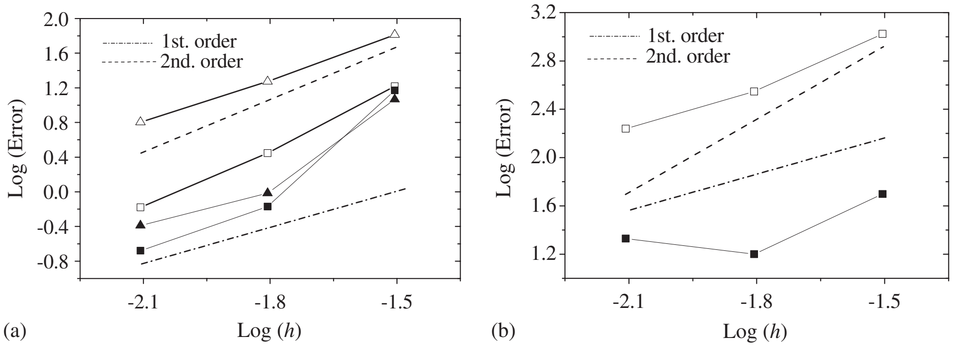

Figure 6. Norm error and maximum error in the pressure: (a) L_1 norm of the global error: –■–, RLIM, –□–, Present method; L_2 norm of the global error: –▲–, RLIM, –△–, Present method; and (b) maximum error: –■–, RLIM, –□–, Present method.

3.2. Uniform flow around a circular cylinder

Next, we simulate a uniform flow around a circular cylinder, to demonstrate the accuracy of this method to predict the hydrodynamics forces on an obstacle immersed in a current. Once the added force have been calculated, the drag and lift coefficient can be calculated directly by an integral of the Eulerian force in the domain [15]. It has been known that the drag and lift forces on an immersed body in a stream, arise from two sources: the shear stress and the pressure distribution along the body, so we can also calculate the drag and lift force from the velocity and pressure interpolated from the flow field.

A rectangular domain is used. The boundary conditions are imposed in such a way that the flow is from the left toward the right of the domain. The circular cylinder is placed inside the domain, with its centre $8D$ away from the inlet, $15D$ from the outlet, $8D$ from the top boundary, and $8D$ from the bottom boundary (D denotes the cylinder diameter). A non-reflect boundary condition is applied on the outlet. A uniform constant velocity is specified at the domain entrance, as well as at the top boundary and bottom boundary. The uniform mesh size is 0.05. Several cases of Reynolds number are considered. The Reynolds number in this flow is defined based on the uniform inflow velocity and the cylinder diameter. A time step $\Delta t = 0.01$ is used for all the cases. For Reynolds number below $Re = 47$ the wake formed behind the cylinder attains a steady symmetric state, which is in agreement with the well-established result by the linear stability theory. The cylinder wake instabilities rises for $Re = 47$.

The length of the wake bubble S/D is here defined as the distance between two stagnation points downstream of the cylinder, as illustrated in Figure 7, as an illustration at $Re = 30$. For different values of Reynolds number the length of wake bubble is presented in Table I. Comparisons of the present results to those of other authors (Teneda [19] and Takami [20]) demonstrate good agreement. The comparison of the drag coefficient, obtained in the present work, with other numerical and experimental results, is presented in Table II. Very good agreement has also been obtained. Furthermore, we show the pressure coefficient on the cylinder surface as a function of the angle θ in Figure 8. The results fit well with that obtained by Silva *et al.* [15], which is also included in this graph.

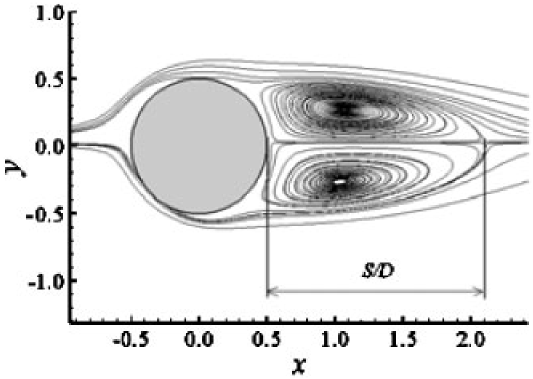

Figure 7. Wake streamlines at $Re = 30$.

Table I. Length of the wake bubble for different Reynolds number.

Reynolds number	20	25	30	40	42	
S/D	Teneda [19]	—	1.150	1.490	2.200	2.350
	Takami [20]	0.935	—	1.612	2.325	—
	Present work	0.962	1.285	1.605	2.220	2.347

Table II. Comparison of mean drag coefficient (C_d) with those of other authors.

Reynolds number	10	20	40	50	80	100	300
Triton [21]	—	2.22	1.48	—	1.29	—	—
Dennis *et al.* [22]	—	2.05	1.52	—	—	1.06	—
Fornberg [23]	—	2.00	1.50	—	—	—	—
Park *et al.* [24]	2.78	2.01	1.51	—	1.35	1.33	—
Silva *et al.* [15]	2.81	2.04	1.54	1.46	1.40	1.39	1.27
Present work	2.98	2.06	1.52	1.42	1.32	1.30	1.26

3.3. Flow around a transversely oscillating cylinder

In this section, we will examine the flow around a transversely oscillating cylinder. The computational domain and grid used for the current simulations are the same with the last case. A uniform free stream velocity is prescribed on the inflow (left), top, and bottom boundaries. The non-reflect boundary condition is employed at the exit (right) boundary. The cylinder sinusoidally oscillates such that the location of its centre (x_c, y_c) is given by $x_c(t) = x_0$, $y_c(t) = y_0 + A\sin(2\pi f_c t)$. A is the amplitude and f_c is the frequency of the oscillation, and all the variables have been non-dimensionalized. The cylinder is immersed and oscillates through the fixed, uniform, Cartesian mesh.

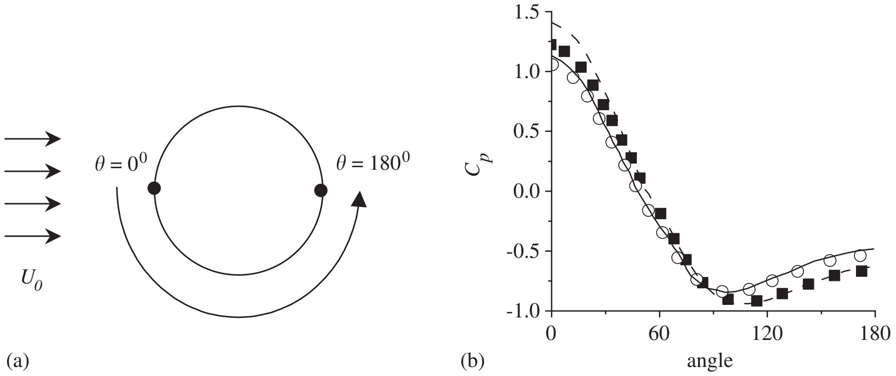

Figure 8. Pressure coefficient distribution, between the stagnation point ($\theta = 0$) and the base point ($\theta = 180$): (a) schematic illustration of the angle θ; and (b) solid line, $Re = 40$, dashed line, $Re = 20$ (present study); symbol ■, $Re = 40$, ○, $Re = 20$ [15].

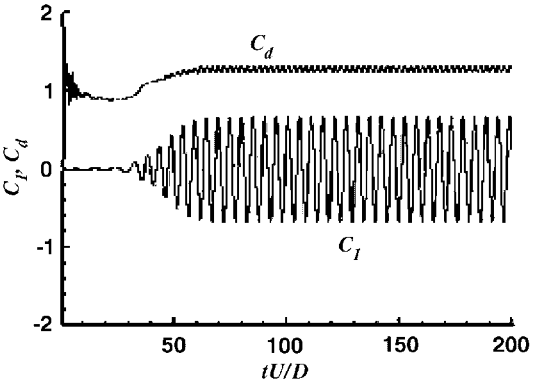

Figure 9. Lift and drag coefficients time history at $Re = 200$.

In all simulations presented in this case, $Re = 200$. Lift and drag coefficients are shown in Figure 9, for the case of a fixed circular cylinder at the Reynolds number equal to 200. A good agreement is obtained in comparison with simulations using other methods, e.g. discrete vortex method as in Reference [25]. The average drag coefficient is approximately 1.28, slightly higher than Bearman's result, where a value of 1.23 was found. Fourier analysis of the fluctuating lift yields a Strouhal number ($f_s D/U$) of about 0.195, which matches well with the experiments by Williamson [26], who obtained a Strouhal number of 0.197, and also the numerical result of 0.196 from Reference [25].

It has been known that Vortex shedding 'lock-on' is a classical phenomenon that is observed in the wake of an oscillating cylinder and refers to the situation where the frequency of vortex shedding in the wake synchronizes with or locks on to the frequency of forced perturbation. Inside the 'lock-in' region, for small amplitudes of oscillation ($A/D < 0.6$), flow visualizations

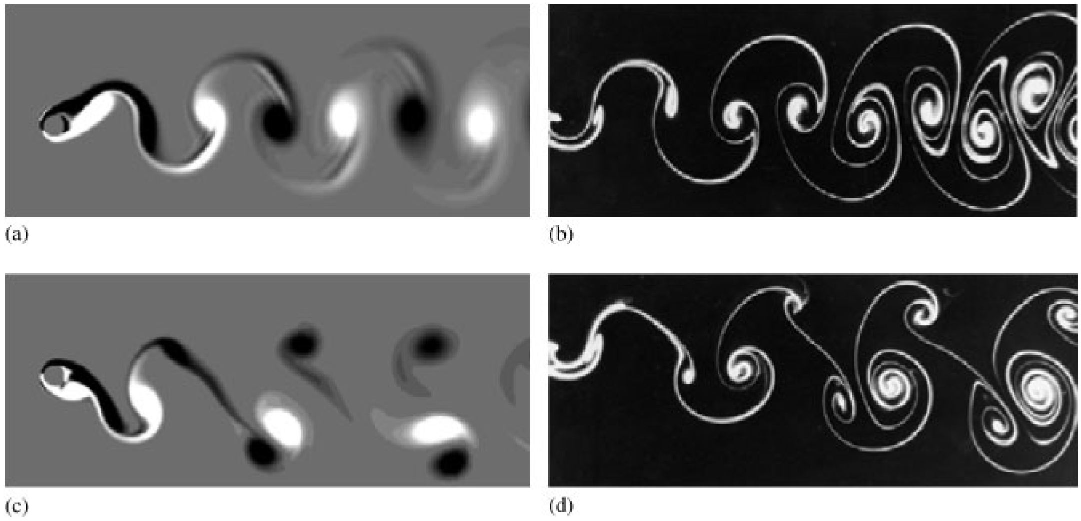

Figure 10. Wake structure for $f_c/f_s = 0.8$: (a) $A/D = 0.55$ present numerical results; (b) $A/D = 0.55$ from experiment [27]; (c) $A/D = 0.65$ present numerical results; and (d) $A/D = 0.65$ from experiment [27].

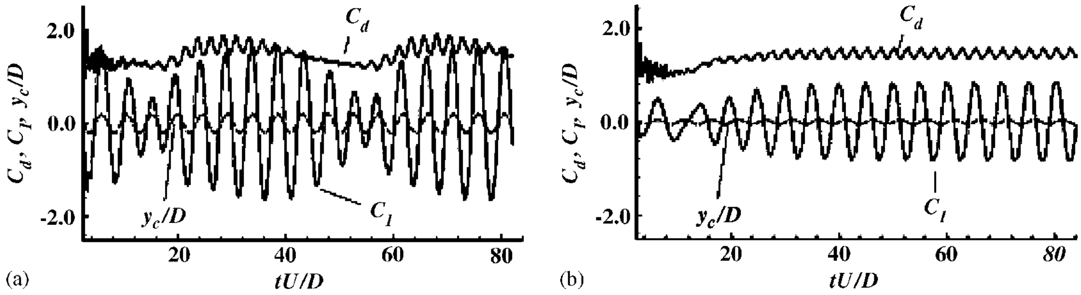

Figure 11. C_d, C_l and y_c/D time history for: (a) $A/D = 0.2$, $f_c/f_s = 1.1$; and (b) $A/D = 0.05$, $f_c/f_s = 1.025$.

show two vortices of opposite circulation shed per cycle as in a conventional Karman vortex street. This kind of synchronization has been called ‘primary lock-in’. For large amplitudes of oscillation, A/D greater than about 0.6, the wake presents a different mode of shedding, where two vortices of opposite circulation are shed on one side of the wake and a single vortex is shed on the other side. In the current study, two simulations have been carried out at fixed forced frequency $f_c/f_s = 0.8$, $A/D = 0.55$ and 0.65. The contours of vorticity are shown in Figure 10, according well with the results by Bearman *et al.* [25]. The wake mode in Figure 10(b) is generally called ‘P + S’ mode. Furthermore, time history of the drag and lift coefficients as well as the cylinder displacement is shown in Figure 11. The results for a frequency of oscillation with $f_c/f_s = 1.1$ and $A/D = 0.2$ are shown in Figure 11(a), in which the beating behaviour is observed. In Figure 11(b), with $f_c/f_s = 1.025$ and $A/D = 0.05$, the phase shift between the lift oscillation and cylinder displacement is about 12°, which is also found out by Bearman *et al.* [25].

3.4. Three-dimensional simulation of a swimming fish

Several theories have been developed to explain fish swimming performance. Many previous researchers investigated the flexible two-dimensional and three-dimensional plates using linearized potential flow theory, instead of real fishes. Although these works provide insight into the basic swimming propulsive mechanics, the details of the three-dimensional flow and the dynamics of the shed vortices are still not well understood. Since the complex configuration of the body and the undulation of the tail fin, direct experimental dynamics measurements of the three-dimensional structure in the wake of the swimming fish are difficult to obtain, and the direct numerical simulations are also difficult to carry out.

Anderson [28] used experimental digital particle image velocimetry (DPIV) to visualize the wake behind a swimming giant danio (*Danio malabaricus*) and identified the active manipulation of shed wake vorticity to create a reverse Karman vortex street. Wolfgang [29] developed a computational tool for investigating the swimming characteristics of a three-dimensional flexible body. With the exception of the wake, the fluid is assumed to be inviscid and irrotational, as well as incompressible, allowing for the existence of a velocity potential. However, this kind of numerical methods is a simplification of the real flow. In this paper, we will rehandle this problem by simulating the full N–S equations.

The computational body geometry is chosen that includes the main portion of the body and the caudal fin (Figure 12), with the smaller dorsal and anal fins left out of account. The main body sections are assumed to be elliptical with a major-to-minor axis ratio of $AR = 2.2$, where the major axis corresponds to the height of the body. A curve-fitting technique if used to determine the profile shape of the body with $L = 1.0$, and is given by

$$z(x) = p(x) \pm 0.1525 \tanh(6x + 1.8) \quad \text{for} \ -0.3 \leqslant x \leqslant 0.1 \tag{20}$$

$$z(x) = p(x) \pm [0.075 - 0.076 \tanh(6.3x - 3.08)] \quad \text{for} \ 0.1 \leqslant x \leqslant 0.7 \tag{21}$$

$$p(x) = 0.195 \tanh[-(0.3 + x)/0.15] + 0.195 \tag{22}$$

where $z(x)$ is the vertical coordinate along the length of the body, $p(x)$ is the vertical coordinate of the mean line along the length of the body. The caudal fin is assumed to have chordwise sections of NACA 0012 shape, and the caudal fin leading-edge and trailing-edge profiles for the semi-span are also determined through a curve-fitting technique. The profiles are given simply by

$$x(z)_{\mathrm{LE}} = 39.543z^3 - 3.685z^2 + 0.636z \tag{23}$$

$$x(z)_{\mathrm{TE}} = -40.74z^3 + 9.666z^2 - 0.15z + 0.1075 \tag{24}$$

where $x(z)_{\mathrm{LE}}$ and $x(z)_{\mathrm{TE}}$ are the longitudinal coordinate of the leading and trailing edges, respectively, along the span of the fin, and where $0 \leqslant z \leqslant 0.15$. The leading edge of the tail

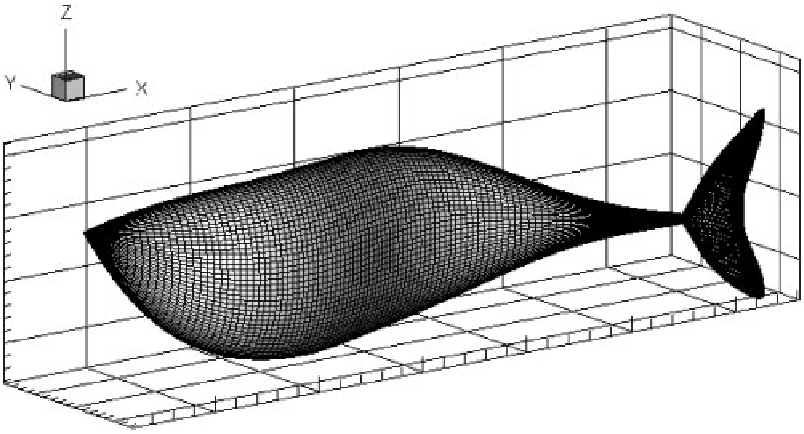

Figure 12. The surface makers chosen for the main body of a giant danio and the dorsal fin.

at midspan intersects the posterior end of the body's contraction region, or the region of the body which narrows in cross-sectional area anterior to the caudal peduncle. The description of the motion is curve-fitted to be purely sinusoidal and to consist of a smooth amplitude-modulated travelling wave along the body length with constant phase speed for a constant swimming speed. The imposed transverse motion $y(x,t)$, with x measure from the nose, has the form

$$y(x,t) = a(x)\sin\left(2\pi\left(\frac{1}{\lambda}x - f\cdot t\right)\right) \tag{25}$$

where $a(x)$ is the amplitude envelope, given the form of a quadratic function

$$a(x) = c1x + c2x^2 \tag{26}$$

where $c1$ and $c2$ are adjustable coefficient, as described by Barrett *et al.* [30], and are chosen to achieve a specific shape of the amplitude envelope $a(x)$ along the length of the body and a specific value of the double amplitude of motion A at the tail. The wake Strouhal number is given by

$$St = fA/U \tag{27}$$

where f is the oscillation and A is the total mean lateral excursion of the tail fin. Our computational parameters are employed to investigate the straight-line swimming of the giant danio, and two typical sets of parameters are chosen for simulations, specifically: the Reynolds number $Re = 600$; swimming speed $U = 1.0\,\mathrm{l\,s}^{-1}$; backbone wavelength $\lambda = 1.1\,\mathrm{l}$; tail-beat frequency $f = 0.8\,\mathrm{Hz}$ and $f = 2.5\,\mathrm{Hz}$; tail-tip double amplitude $A = 0.16\,\mathrm{l}$ and $A = 0.4\,\mathrm{l}$; Strouhal number $St = 0.128$ and 1.0.

Figure 13 illustrates the thrust jet region and the near-body flow dynamics by showing the pressure contours at midbody depth. The present results show a qualitative agreement with the results by Wolfgang *et al.* [29], and not be quantitative, because different numerical methods

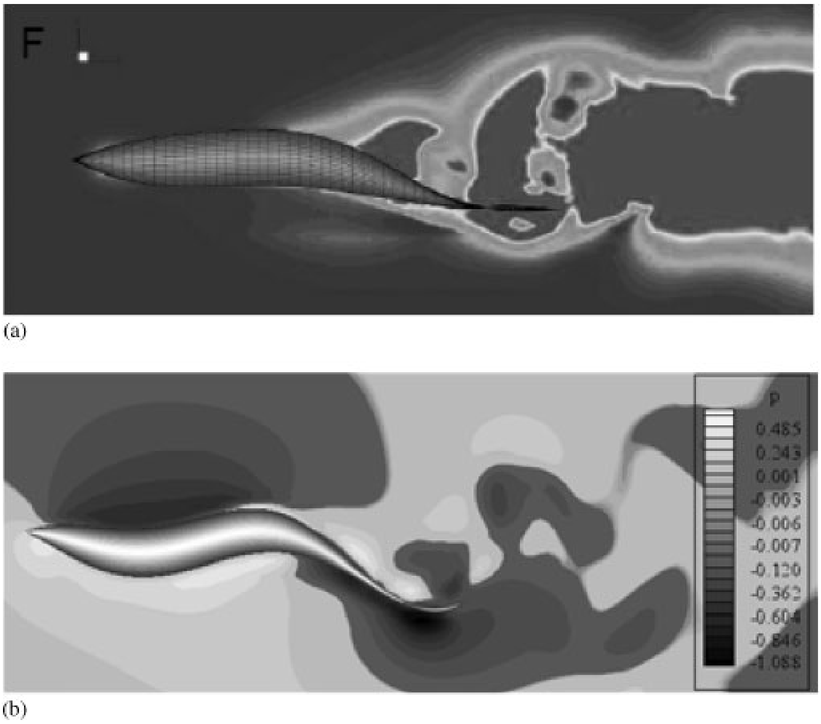

Figure 13. Pressure coefficient contours at midplane depth ($z = 0.5H$, where H is fish body maximum total depth): (a) the black regions in the wake are low pressure, and the corresponding $St = 0.45$ [29]; and (b) the white regions are high pressure, and the black regions are low pressure, and the corresponding $St = 1.0$ (in present work).

are employed, and also due to the different beat amplitudes that we have chosen. In this paper, we simulate the full N–S equations, including the viscous terms. A high-pressure region is observed near the front part of the fish body, and a corresponding low-pressure region in the opposite side of the body. The vortical patterns behind the swimming fish are shown in Figure 14. These two pictures are similar to the results reviewed by Triantafyllou *et al.* [32] behind a rectangular foil with $AR = 3$. For low Strouhal number, the patterns resemble irregular rings, or 'pancakes', and the rings are joined one by one behind the fish body, and this is similar to the flow wake behind a sphere [33]. When considering the flow around a sphere, we also call the rings hairpins vortices. For high Strouhal number, as shown in Figure 14(b), the vortical patterns evolve in a significantly different way: Two distinct branches appear, drifting away from each other. The interconnections are not simple, but distinct ring-like structures form, which in a vertical planar cut, near the centre-plane, would, appear as four major vortices per cycle. In the far wake of the fish body, the vortices become very weak, leading to the disappearance of the ring-like structures, the flow field is characterized by four stable fourfold-threads vortical structures in the far downstream wake. This indicates that the energy generated by the oscillation of the fish body, especially by the caudal fin, will be dissipated in the far wake of the fish body.

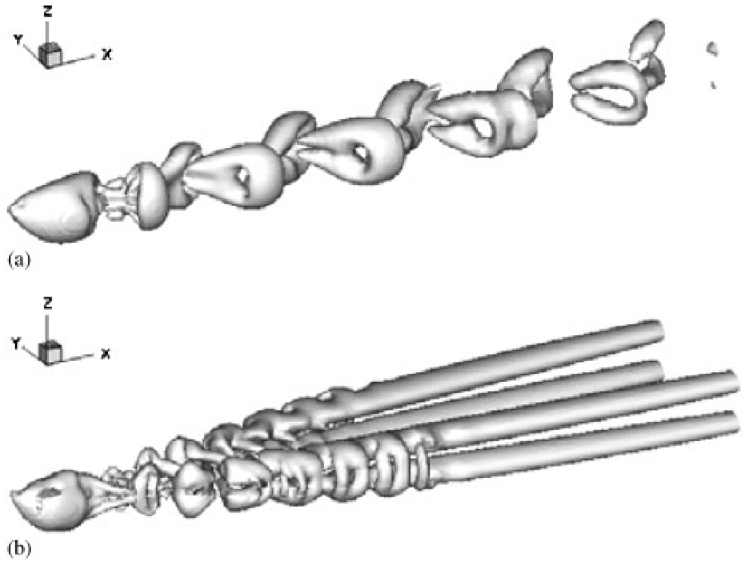

Figure 14. Vortical patterns in the wake of a swimming fish at different Strouhal numbers, and the patterns in the flow are iso-surfaces of the λ_2 eigenvalue [31]: (a) $St = 0.128$; and (b) $St = 1.0$.

4. CONCLUSIONS

A new modification of the immersed-boundary method is developed which allows us to simulate unsteady, viscous incompressible flows with complex immersed and moving boundaries. The underlying method is based on a staggered arrangement of variables. A second-order central difference scheme is used for spatial differencing. Furthermore, the solution is advanced in using a two-step fractional-step scheme. The immersed boundary is represented by a series of interfacial markers. The Lagrangian forcings are calculated on these markers, and then scaled to the grid points nearby through a linear distribution.

The present solver is used to simulate three problems. First, a circular cylinder immersed in a lid-driven cavity is tested to confirm the second-order accuracy of this solver. Second, simulations of flow past a circular cylinder in a uniform stream are performed with the Reynolds number ranging from 20 to 300. Key results such as mean drag coefficient, length of recirculation zone, obtained from our simulations agree well with established experimental and numerical results. Third, the flow past a transversely oscillating cylinder is investigated. We find that the solver is able to simulate the moving boundary problems conveniently without any modification of the codes for fixed boundary. At last, we carry out a numerical study of the flow around a swimming fish. The main advantage of the current approach is that flows with extremely complex internal boundaries can be simulated with relative ease on simple Cartesian meshes.

ACKNOWLEDGEMENTS

We thank the Center for Engineering and Scientific Computation of Zhejiang University for providing the SGI Onyx3900 machine for most of the computational tasks. This work is a part of the Vortex Induced

Vibration (VIV) research project, which is supported by the National Natural Science Foundation of China (grant no: 10272094). This work is also a part of the Particles Suspension Flow research project, which is supported by the National Natural Science Foundation of China (Grant no: 10472104).

REFERENCES

1. Peskin CS. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics* 1972; **10**:252.
2. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**:220.
3. Peskin CS. The fluid dynamics of heart valves: experimental, theoretical, and computational methods. *Annual Review of Fluid Mechanics* 1982; **14**:235.
4. Peskin CS, Printz BF. Improved volume conservation in the computation of flows with immersed elastic boundaries. *Journal of Computational Physics* 1993; **105**:33.
5. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; 479–517.
6. Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics* 1993; **105**:354.
7. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
8. 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 and Stanford University, 1997; 317.
9. 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**:5.
10. 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.
11. Tseng YH, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
12. Gilmanov A, Sotiropoulos F, Balaras E. A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids. *Journal of Computational Physics* 2003; **191**:660–669.
13. Balaras E. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Computers and Fluids* 2004; **33**:375–404.
14. Kang S, Iaccarino G, Moin P. Accurate and efficient immersed-boundary interpolations for viscous flows. *Center for Turbulence Research Annual Research Briefs* 2004; 31–43.
15. Lima ALF, Silva E, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *Journal of Computational Physics* 2003; **189**:351–370.
16. Udaykumar HS, Shyy W, Rao MM. Elafint: a mixed Eulerian Lagrangian method for fluid flows with complex and moving boundaries. *International Journal for Numerical Methods in Fluids* 1996; **22**:691–712.
17. Udaykumar HS, Kan H-C, Shyy W, Tran-Son-Tay R. Multiphase dynamics in arbitrary geometries on fixed Cartesian grids. *Journal of Computational Physics* 1997; **137**:366–405.
18. Sheen SC, Wu JL. Solution of the pressure correction equation by the preconditioned conjugate gradient method. *Numerical Heat Transfer*, *Part B* 1997; **32**:215–230.
19. Teneda S. Experimental investigation of the wakes behind cylinders and plates at low Reynolds numbers. *Journal of the Physical Society of Japan* 1956; **11**(3):302–307.
20. Takami H, Keller HB. Steady two-dimensional viscous flow of an incompressible fluid past a circular cylinder. *Physics of Fluids* 1969; **12**(12):II-51–II-56.
21. Triton DJ. Experiments on the flow past a circular cylinder at low Reynolds number. *Journal of Fluid Mechanics* 1959; **6**:547.
22. Dennis SCR, Chang G. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *Journal of Fluid Mechanics* 1970; **42**:471.
23. Fornberg B. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics* 1980; **98**:819–855.
24. Park J, Kwon K, Choi H. Numerical solutions of flow past a circular cylinder at Reynolds number up to 160. *KSME International Journal* 1998; **12**:1200.
25. Meneghini JR, Bearman PW. Numerical simulation of high amplitude oscillatory flow about a circular cylinder. *Journal of Fluids and Structures* 1995; **9**:435–455.
26. Williamson CHK. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics* 1996; **28**:477.
27. Williamson CHK, Govardhan R. Vortex-induced vibrations. *Annual Review of Fluid Mechanics* 2004; **36**:413–455.
28. Anderson JM. Vorticity control for efficient propulsion. *Doctoral Thesis*, Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution, 1996.
29. Wolfgang MJ, Anderson JM, Grosenbaugh MA, Yue DKP, Triantafyllou MS. Near-body flow dynamics in swimming fish. *The Journal of Experimental Biology* 1999; **202**:2303–2327.

30. Barrett D, Triantafyllou M, Yue D, Grosenbaugh M, Wolfgang M. Drag reduction in fish-like locomotion. *Journal of Fluid Mechanics* 1999; **392**:183–212.
31. Jeong J, Hussein F. On the identification of a vortex. *Journal of Fluid Mechanics* 1995; **285**:69–94.
32. Triantafyllou M, Techet A, Hover F. Review of experimental work in biomimetic foils. *IEEE Journal of Oceanic Engineering* 2004; **29**(3):585–594.
33. Zou JF, Ren AL, Deng J. Study on flow past two spheres in tandem arrangement using a local mesh refinement virtual boundary method. *International Journal for Numerical Methods in Fluids* 2005; **49**(5):465–488.