Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **4**, No. 3, pp. 311-324

DOI: 10.4208/aamm.10-m1171
June 2012

Simulation of Incompressible Viscous Flows by Local DFD-Immersed Boundary Method

Y. L. Wu[1], C. Shu[1,*]and H. Ding[2]

[1] *Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 117576*
[2] *Department of Modern Mechanics, University of Science and Technology of China, Hefei, Anhui 230026, China*

Received 28 April 2011; Accepted (in revised version) 26 September 2011

Available online 30 April 2012

Abstract. A local domain-free discretization-immersed boundary method (DFD-IBM) is presented in this paper to solve incompressible Navier-Stokes equations in the primitive variable form. Like the conventional immersed boundary method (IBM), the local DFD-IBM solves the governing equations in the whole domain including exterior and interior of the immersed object. The effect of immersed boundary to the surrounding fluids is through the evaluation of velocity at interior and exterior dependent points. To be specific, the velocity at interior dependent points is computed by approximate forms of solution and the velocity at exterior dependent points is set to the wall velocity. As compared to the conventional IBM, the present approach accurately implements the non-slip boundary condition. As a result, there is no flow penetration, which is often appeared in the conventional IBM results. The present approach is validated by its application to simulate incompressible viscous flows around a circular cylinder. The obtained numerical results agree very well with the data in the literature.

AMS subject classifications: 76 Fluid mechanics; 35 Partial differential equations

Key words: Local domain free discretization (local DFD), immersed boundary method (IBM), incompressible flow, flow past a circular cylinder.

1 Introduction

Immersed Boundary Method (IBM) has been becoming more and more popular in the numerical simulation of incompressible viscous flows since it was firstly proposed by Peskin in 1970s for the study of blood flow in the heart valve [1]. The original IBM

*Corresponding author.
URL: http://serve.me.nus.edu.sg/shuchang/
Email: mpeshuc@nus.edu.sg (C. Shu)

uses a fixed Eulerian mesh for the flow field, and a set of Lagrangian points to represent the boundary of objects immersed in the fluid. The basic idea of IBM is that the physical boundary is treated as deformable with high stiffness. A small distortion of the boundary will yield a force which tends to restore the boundary into its original shape. The balances of such forces are distributed into the Eulerian mesh points and the Navier-Stokes (N-S) equations with a body force are solved on the whole domain, including exterior and interior of the object. After the work of Peskin [1], numerous research works have been made to improve IBM. Goldstein et al. [2] proposed a model named virtual boundary method which permits simulations with complex geometries. Lai and Peskin [3] presented a so-called second-order accurate IBM with adoption of a well-chosen Dirac delta function. Linnick and Fasel [4] used the fourth-order compact finite-difference schemes for approximation of spatial derivatives in the IBM application. Lima E Silva et al. [5] proposed a version named physical virtual model, which is based on the conservation laws, and simulated a channel flow and the flow around a circular cylinder. Feng and Michaelides [6] firstly presented the immersed boundary-lattice Boltzmann method (IB-LBM) to simulate the motion of rigid particles in the flow field, where the solution of flow field is obtained by the newly-developed lattice Boltzmann method (LBM) [7]. Later, Niu et al. [8] proposed the momentum exchange-based IB-LBM for simulation of several incompressible flows, and Peng et al. [9] developed the multi-block IB-LBM for simulation of flows around a circular cylinder and an airfoil.

The major advantage of IBM is its simplicity and easy implementation because it decouples the solution of governing equation with the implementation of boundary conditions at immersed boundaries. This means that the governing equation can always be solved on a regular domain without considering the presence of solid boundary immersed in the flow field. The effect of solid boundary to surrounding fluids is through the introduction of body force in the governing equation. On the other hand, it should be indicated that the conventional IBM suffers two major drawbacks. One is the flow penetration to the immersed boundary. This is because in the IBM, the non-slip boundary condition is not enforced, and is only approximately satisfied at the converged state. As a consequence, some streamlines may penetrate the immersed boundary. The flow penetration means some mass exchange through the boundary, which will cause the momentum exchange and lead to numerical error of force calculation. Another drawback is the low order accuracy nearby the immersed boundary due to the use of Dirac delta function interpolation. In the IBM, the distribution of restoring force at the boundary (Lagrangian) nodes to the Eulerian mesh points and interpolation of flow velocity at the Eulerian mesh points to the boundary nodes are through the Dirac delta function, which only has the first order of accuracy. To remove the drawback of flow penetration, Shu et al. [10] did some analysis and found that unsatisfying of non-slip boundary condition in IBM is in fact due to pre-calculated restoring force. Using the fractional step technique, they concluded that, adding a body force in the momentum equations is equivalent to make a correction in the velocity field. To enforce the non-slip boundary condition, the velocity correction (restoring force) should be considered as unknown, which is determined by enforcing the non-slip boundary condition. In the work of Shu et al. [10], the flow penetration is avoided. However, the numerical accuracy near the immersed boundary is still the first-order. In addition, it needs to find and store the intersection points between the immersed boundary and mesh lines. This may bring some inconveniences in the numerical computation. Recently, Wu and Shu [11] further incorporated the idea of enforcing non-slip boundary condition into the conventional IBM and presented the boundary condition-enforced IBM. The drawback of flow penetration is also removed, but the first order delta function interpolation is still used. It seems that under the current framework of IBM, the drawback of low order accuracy due to the use of delta function is difficult to be overcome. There is a demanding to develop more accurate and efficient solver to remove this drawback.

On the other hand, it was found that the local domain-free discretization (DFD) method can enforce the non-slip boundary condition with the second or higher order of accuracy [12]. The DFD method was firstly proposed by Shu and Fan [13], Shu and Wu [14]. Its basic idea is inspired from the analytical method. Consider a partial differential equation (PDE) on an irregular domain. The PDE is discretized at all mesh points inside the solution domain (defined as *interior points*), but the spatial discretization at an interior mesh point may involve some points outside the solution domain (defined as *exterior points*). The functional values at those exterior points can be evaluated by approximate forms of solution at the interior point, either globally or locally, leading to two versions of DFD methods, i.e., global DFD and local DFD [12]. If the approximate form of solution is obtained by using all the information along a mesh line, it is named as the global DFD approach. If the approximate form of solution is obtained locally by using a low order polynomial, it is termed as the local DFD approach. In general, DFD method can be applied to any kind of mesh configuration. When the local DFD method is applied on the Cartesian mesh, it can be considered as a kind of Cartesian mesh solvers. Usually, for the application of local DFD method to problems with complex geometry or moving boundaries [15], the mesh points take one of the following three statuses: interior points where the governing equations are solved; exterior dependent points adjacent to the boundary where the functional values are determined by approximate forms of the solution; exterior independent points where we do not need to do anything. It should be indicated that the process of computing functional values at exterior dependent points by approximate forms of solution is actually extrapolation, which would bring larger numerical errors than the interpolation.

From the above descriptions, we can see clearly that the conventional IBM and local DFD method have different features. To summarize, IBM is easy to apply, which solves governing equations in the whole domain, but it may not accurately satisfy the non-slip boundary condition and the numerical treatment near the immersed boundary only has the first order of accuracy. In contrast, the local DFD method can accurately implement the non-slip boundary condition, but it only solves governing equations at interior points and the functional values at exterior dependent points are cal-

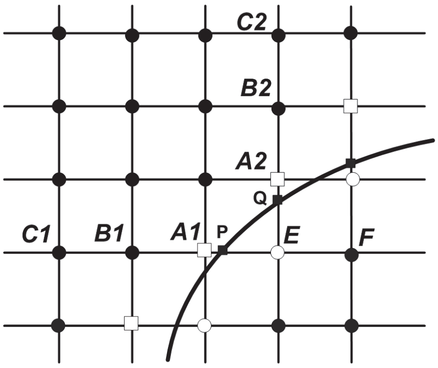


Figure 1: Classification of points in Local DFD-IBM.

culated by extrapolation. In this work, we will incorporate the idea of IBM (solving governing equations in the whole domain) into the local DFD method, and the approach is named as local DFD-IBM. Like the conventional IBM, two sets of nodes are also used in the local DFD-IBM. One set of nodes are the Eulerian points defined on the fixed Cartesian mesh (mesh nodes in Fig. 1). The other set of nodes are the Lagrangian points to represent the immersed boundary (represented by small solid squares distributed along the boundary in Fig. 1). Different from the local DFD method [12], in the local DFD-IBM, the Eulerian mesh points are classified into three categories: discretization points (including points inside and outside the immersed object that are not adjacent to the immersed boundary, represented by black solid circles in Fig. 1) where the governing equations are solved; interior dependent points just adjacent to the immersed boundary (represented by empty squares in Fig. 1) where approximate forms of solution are used to evaluate the functional values; exterior dependent points just adjacent to the immersed boundary (represented by empty circles in Fig. 1) where the velocity is set to the boundary velocity. It is indicated that the approximate forms of solution used for interior dependent points serve as a bridge to link information between Eulerian and Lagrangian nodes with accurate implementation of non-slip boundary condition, which has the same role as Dirac delta function used in the conventional IBM. The process is in fact interpolation. Clearly, the present approach combines the advantages of conventional IBM and local DFD method. It is validated by its application to simulate incompressible viscous flows around a circular cylinder. Numerical experiments show that the present results agree very well with available data in the literature.

2 Solution of incompressible N-S equations by local DFD-IBM

In this section, we will describe how to apply the local domain-free discretization-immersed boundary method (DFD-IBM) to solve incompressible N-S equations. The primitive variable form of N-S equations is taken as the governing equation for numerical simulation, which can be written as

$$\text{Momentum equation:} \quad \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}} \Delta \mathbf{u}, \tag{2.1a}$$

$$\text{Continuity equation:} \quad \nabla \cdot \mathbf{u} = 0. \tag{2.1b}$$

Re denotes the Reynolds number, defined as

$$\mathrm{Re} = \frac{UL}{\nu},$$

where L is the reference length, U the reference velocity and ν the kinematic viscosity. Eqs. (2.1a) and (2.1b) can be solved by pressure correction method [16]. Using Euler explicit scheme, Eq. (2.1a) can be discretized as

$$\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\Delta t} = -H(\mathbf{u}^n) - Gp^n + \frac{1}{\mathrm{Re}} L(\mathbf{u}^n), \tag{2.2}$$

where H denotes the discrete advection operator, G is the discrete gradient operator, L the discrete Laplacian operator. Superscripts n and $n+1$ denote the time levels.

The basic idea of local DFD-IBM has been described in the introduction. Like the conventional IBM, Eq. (2.2) is solved in the whole domain. However, at interior dependent points adjacent to the immersed boundary (represented by Lagrangian nodes), we need to use approximate forms of the solution to evaluate the velocity, and at the exterior dependent points adjacent to the immersed boundary, we need to set their velocity to the wall velocity. Note that the local DFD method only gives discretization strategy, that is, functional values at dependent points to the boundary are evaluated by approximate forms of the solution. It still needs conventional numerical schemes to discretize the spatial derivatives. In this work, the second order finite difference schemes are used to approximate spatial derivatives in discrete operators H, G and L.

To show the process of evaluating the velocity at dependent interior and exterior points, let us consider a solid boundary immersed in a fixed Cartesian mesh as shown in Fig. 1, where the symbol ″□″ represents the interior dependent points just adjacent to the immersed boundary and ″○″ represents the exterior dependent points just adjacent to the immersed boundary, ″●″ represents for all other Eulerian nodes in the local DFD-IBM method. The symbol ″■″ denotes the Lagrangian nodes, i.e., the intersection points of mesh lines and the immersed boundary. Obviously, $A1$ and $A2$ are interior dependent points and E is the exterior dependent point. The velocity at $A1$ and $A2$ can be evaluated by approximate forms of the solution which involve the solution at interior mesh points and the velocity at immersed boundary. The evaluation process is actually interpolation. Take point $A1$ as an example. Since numerical discretization in the x direction at the exterior dependent point E needs the information at $A1$, the velocity at $A1$ can be evaluated by a second order polynomial along the x direction, which involves three points P, $B1$, $C1$ as shown in Fig. 1. Here, $B1$ and $C1$ are the interior mesh points, and point P is the intersection point of the horizontal mesh line with the immersed boundary, where the velocity of immersed boundary is assigned, i.e.,

$$u_P = U_b, \quad v_P = V_b. \tag{2.3}$$

U_b, V_b are velocity components of the immersed boundary. For stationary boundary, they are zero. The interpolation form for calculation of velocity at $A1$ can be written as

$$\begin{aligned} \hat{u}^n_{A1} = & \frac{(x_{A1}-x_{B1})(x_{A1}-x_{C1})}{(x_P-x_{B1})(x_P-x_{C1})}u^n_P + \frac{(x_{A1}-x_{C1})(x_{A1}-x_P)}{(x_{B1}-x_{C1})(x_{B1}-x_P)}u^n_{B1} \\ & + \frac{(x_{A1}-x_P)(x_{A1}-x_{B1})}{(x_{C1}-x_P)(x_{C1}-x_{B1})}u^n_{C1}, \end{aligned} \tag{2.4a}$$

$$\begin{aligned} \hat{v}^n_{A1} = & \frac{(x_{A1}-x_{B1})(x_{A1}-x_{C1})}{(x_P-x_{B1})(x_P-x_{C1})}v^n_P + \frac{(x_{A1}-x_{C1})(x_{A1}-x_P)}{(x_{B1}-x_{C1})(x_{B1}-x_P)}v^n_{B1} \\ & + \frac{(x_{A1}-x_P)(x_{A1}-x_{B1})}{(x_{C1}-x_P)(x_{C1}-x_{B1})}v^n_{C1}. \end{aligned} \tag{2.4b}$$

Similarly, the velocity at point $A2$ can be updated by the following interpolation form

$$\begin{aligned} \hat{u}^n_{A2} = & \frac{(y_{A2}-y_{B2})(y_{A2}-y_{C2})}{(y_Q-y_{B2})(y_Q-y_{C2})}u^n_Q + \frac{(y_{A2}-y_{C2})(y_{A2}-y_Q)}{(y_{B2}-y_{C2})(y_{B2}-y_Q)}u^n_{B2} \\ & + \frac{(y_{A2}-y_Q)(y_{A2}-y_{B2})}{(y_{C2}-y_Q)(y_{C2}-y_{B2})}u^n_{C2}, \end{aligned} \tag{2.5a}$$

$$\begin{aligned} \hat{v}^n_{A2} = & \frac{(y_{A2}-y_{B2})(y_{A2}-y_{C2})}{(y_Q-y_{B2})(y_Q-y_{C2})}v^n_Q + \frac{(y_{A2}-y_{C2})(y_{A2}-y_Q)}{(y_{B2}-y_{C2})(y_{B2}-y_Q)}v^n_{B2} \\ & + \frac{(y_{A2}-y_Q)(y_{A2}-y_{B2})}{(y_{C2}-y_Q)(y_{C2}-y_{B2})}v^n_{C2}, \end{aligned} \tag{2.5b}$$

which involves three points Q, $B2$, $C2$ in the vertical (y) direction as shown in Fig. 1. Here $B2$, $C2$ are the interior mesh points, and Q is the intersection point of a vertical mesh line with the immersed boundary, where the immersed boundary velocity is assigned, i.e.,

$$u_Q = U_b, \quad v_Q = V_b. \tag{2.6}$$

Again, U_b, V_b are velocity components of the solid boundary. Note that in Eqs. (2.3) and (2.6), the non-slip boundary condition is directly implemented through the velocity at points P and Q. The approximate forms of solution (2.4) and (2.5) have the second-order of accuracy, and they are actually interpolation forms to compute velocity at $A1$ and $A2$.

Point E is the exterior dependent node. Since E is inside the solid body (outside flow domain), in this work, its velocity is simply assigned to the wall velocity, i.e.:

$$\hat{u}^n_E = U_b, \quad \hat{v}^n_E = V_b. \tag{2.7}$$

On the other hand, the solution of Eq. (2.2) can also be obtained by the following two steps

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -H(\mathbf{u}^n) + \frac{1}{\mathrm{Re}} L(\mathbf{u}^n), \tag{2.8a}$$

$$\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^*}{\Delta t} = -Gp^n. \tag{2.8b}$$

Here $\mathbf{u}^*$ is the intermediate velocity. Since $\tilde{\mathbf{u}}^{n+1}$ has been calculated by Eq. (2.2), we can use Eq. (2.8b) to compute the intermediate velocity $\mathbf{u}^*$. Note that in this process, the boundary effect to the surrounding fluids has been considered by Eqs. (2.4), (2.5) and (2.7). With $\mathbf{u}^*$, the velocity at time level t^{n+1} can be corrected by

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -Gp^{n+1}, \tag{2.9}$$

where p^{n+1} is the pressure at time level t^{n+1}. Apart from the momentum equation, the continuity equation (2.1b) should also be satisfied at time level t^{n+1}, that is,

$$D\mathbf{u}^{n+1} = 0, \tag{2.10}$$

where D is the divergence operator. Taking divergence of Eq. (2.9) and using Eq. (2.10), we have

$$Lp^{n+1} = \frac{1}{\Delta t}(D\mathbf{u}^*). \tag{2.11}$$

Eq. (2.11) is Poisson equation for pressure p^{n+1}. Once the pressure p^{n+1} is calculated by Eq. (2.11), the velocity $\mathbf{u}^{n+1}$ can be given by

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t G(p^{n+1}). \tag{2.12}$$

Obviously, in the above process, the physical boundary condition is implemented. As a result, there will be no flow penetration to the solid boundary. This will be confirmed by the test examples shown in the next section.

To sum up, the present solver has the following features:

(1) In the present solver, there is no need to calculate the restoration force F on the Lagrangian points, and thus no need to distribute the restoring force F to the Eulerian nodes. The boundary effect is considered in the approximate form of solution to update the velocity at interior and exterior dependent points.

(2) The boundary condition is implemented accurately. As a consequence, the flow penetration is avoided.

(3) Since the pressure on all the Eulerian nodes is obtained by solving the pressure Poisson equation, the treatment of Neumann boundary condition for pressure as did in other Cartesian mesh solvers, is avoided.

(4) Like the original local DFD method, an important step of the local DFD-IBM method is to identify which mesh node is the interior dependent node, and which mesh node is the exterior dependent node. In other words, we should know the status of the mesh nodes. With our recently developed "odd/even parity method" [12], this judgment can be done very quickly. For rigid body problem, this judgment can be done only once before the iteration. Then the status of the mesh nodes can be stored in the rest of the computation to save the running time.

3 Application of local DFD-IBM to simulate flows over a stationary circular cylinder

The incompressible, viscous flow around a circular cylinder is a classical problem in fluid mechanics. This kind of problems has been studied extensively and there are numerous theoretical, experimental and numerical results available in the literature. It is well known that for this problem, different Reynolds number gives different kind of flow behaviors. It is generally agreed that in two dimensions, the vortex shedding begins at a critical Reynolds number around 49. For the case of Reynolds numbers less than the critical value ($\text{Re}_{\text{critical}} = 49$), the flow is an essentially steady one. Above this critical Reynolds number, the introduced perturbation will trigger the vortex shedding process to form a Von Karman vortex street. The flow will show an unsteady feature. It serves as a good sample problem for validating a new numerical method in solving unsteady two-dimensional Navier-Stokes equations. In this paper, the local DFD-IBM is used to solve this problem. In the present study, we performed numerical simulation at a series of Reynolds numbers ranging from 10 to 200 with various flow patterns in the steady and unsteady state. The Reynolds number Re is defined as

$$\text{Re} = \frac{U_\infty D}{\nu},$$

where D is the cylinder diameter, and ν is the kinematic viscosity.

The problem configuration is shown in Fig. 2, which is an incompressible, viscous fluid flow at a constant velocity U_∞ past a stationary cylinder of diameter D.

The two-dimensional Navier-Stokes equations (2.1a) and (2.1b) are taken as the governing equations for this problem. The boundary conditions of the problem are: At the inlet, a free stream velocity profile is specified, that is,

$$\begin{cases} u = 1, \\ v = 0. \end{cases} \tag{3.1}$$

On the cylinder surface, non-slip boundary condition is implemented by

$$\begin{cases} u = 0, \\ v = 0. \end{cases} \tag{3.2}$$

On the far field boundary except downstream of the cylinder, undisturbed velocity field is used, that is,

$$\begin{cases} u = 1, \\ v = 0. \end{cases} \tag{3.3}$$

At the far field downstream boundary of the cylinder, the natural boundary condition is applied by

$$\begin{cases} \dfrac{\partial u}{\partial x} = 0, \\ \dfrac{\partial v}{\partial x} = 0. \end{cases} \tag{3.4}$$

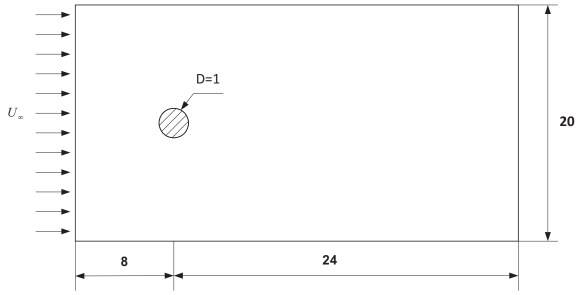


Figure 2: Computational domain for simulation of flow around a circular cylinder.

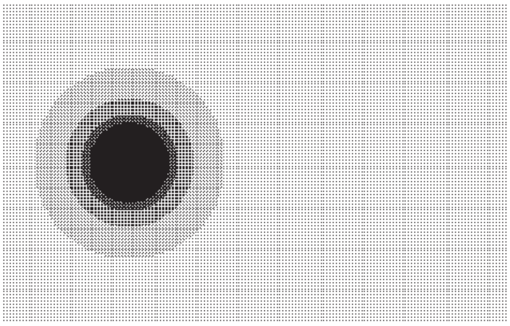


Figure 3: Local refined mesh for simulation of flow past a circular cylinder (45363 nodes).

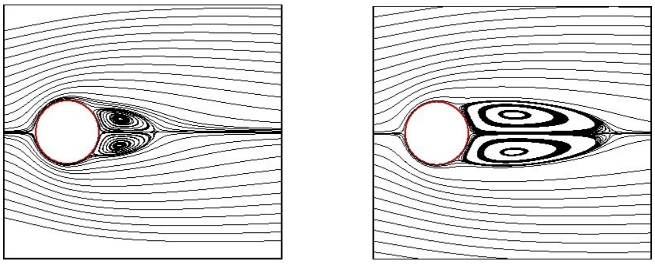
At beginning of numerical simulation, the unsymmetrical initial flow field is used to act as the initial perturbation, which is given as

$$\begin{cases} u|_{t=0} = 1.0 - \dfrac{0.05 * x}{\sqrt{x^2+y^2}}, \\ v|_{t=0} = \dfrac{0.05 * y}{\sqrt{x^2+y^2}}. \end{cases} \tag{3.5}$$

In the simulation, an initial Cartesian mesh of 160 × 101 is used. The initial uniform mesh is obviously not fine enough when Re becomes larger and larger. This is because the boundary layer will be thinner and thinner as Re increases. Therefore, in this work, we adopted the stencil mesh refinement algorithm proposed by Ding & Shu [17] to refine the mesh around the cylinder with 8 refinement levels. The final mesh is shown in Fig. 3, which has 45363 nodes. The mesh spacing near the cylinder is about 0.0125D.

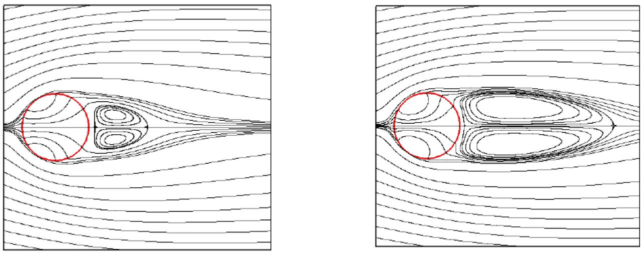
3.1 Simulation of steady flow over a stationary circular cylinder

The local DFD-IBM is firstly applied to simulate the steady flow over a stationary circular cylinder with Re = 20 and 40. Fig. 4 illustrates the streamlines when flow reaches its final steady state. For both cases, a pair of vortices develops behind the cylinder and is perfectly aligned. This is consistent with previous observation. To demonstrate that the present approach has no flow penetration to the boundary of immersed object, the streamlines obtained by conventional IBM [18] are shown in Fig. 5 for comparison.



(a) Re = 20 (b) Re = 40

Figure 4: Streamlines for steady flow with Re = 20 and 40 (Present method).



(a) Re = 20 (b) Re = 40

Figure 5: Streamlines for steady flow with Re = 20 and 40 from conventional immersed boundary method [18]: Flow penetration is very clear in this figure.

Fig. 5 shows very clearly that some streamlines pass through the solid body, which is not true in physics. Comparison between Fig. 4 and Fig. 5 demonstrates that the local DFD-IBM has the advantage to satisfy the non-slip boundary condition accurately.

Some quantitative parameters for the recirculation region, such as the length of the recirculation region, L_s, from the rearmost point of the cylinder to the end of the wake, separation angle θ_s and drag coefficient C_D, as well as the results from other researchers, are listed in Table 1. It was found that both the geometrical and dynamical parameters agree well with the results of previous studies for all two Reynolds numbers studied.

Table 1: Comparison of geometrical and dynamical parameters with previous studies for Re = 20 and 40.

		L_s/a	θ_s	C_D
Re = 20	Dennis and Chang [19]	1.88	43.7°	2.05
	He and Doolen [20]	1.84	43.0°	2.15
	Calhoun [21]	1.82	45.5°	2.19
	Tuann and Olson [22]	1.80	44.1°	2.25
	Ding et al [23]	1.88	43.8°	2.14
	Present	1.80	43.8°	2.22
Re = 40	Dennis and Chang [19]	4.69	53.8°	1.522
	Calhoun [21]	4.36	54.2°	1.62
	Tuann and Olson [22]	4.20	54.8°	1.675
	Ding et al. [23]	4.64	52.8°	1.58
	Present	4.45	53.7°	1.567

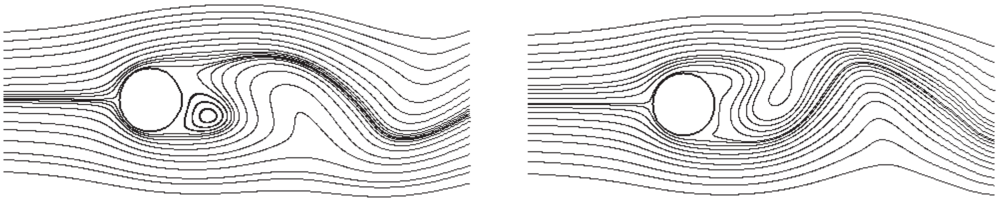
3.2 Simulation of unsteady flow over a stationary circular cylinder

Simulation of the vortex shedding process behind a circular cylinder is a standard test case for validating new numerical approaches to simulate unsteady flows. In this paper, comparisons are made between the present study and other numerical work for Reynolds numbers of Re $=$ 100 and 200. It is generally agreed that in two dimensions, the vortex shedding begins when Reynolds number is above 49. Fig. 6 and Fig. 7 show the instantaneous streamlines and vorticity contours near the wake at Re $=$ 100 and 200. It is found that for Re $=$ 100, 200, the flow field eventually settles into a periodic oscillatory pattern. This confirms the other experimental and numerical findings.

For the flow past blunt bodies, the drag and lift coefficients at the surface of body are two important parameters. The time-evolution of these two characteristic parameters illustrates the variation of the flow field. The drag and lift coefficients are defined by

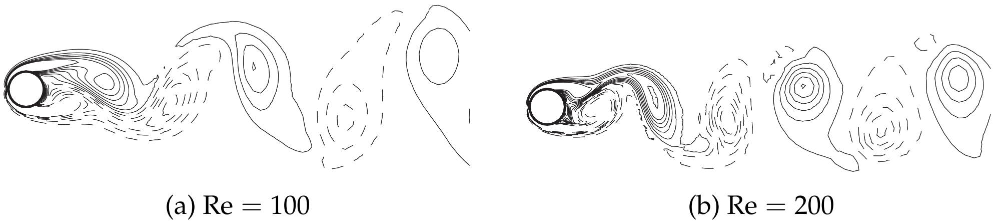
$$C_D = \frac{\mathbf{F}_D}{\rho U^2 D/2}, \qquad C_L = \frac{\mathbf{F}_L}{\rho U^2 D/2}, \tag{3.6}$$

where $\mathbf{F}_D$ is the drag force and $\mathbf{F}_L$ is the lift force acting on the circular cylinder. In this study, the drag force and lift force are obtained by integrating the local pressure and stress distributions along the cylinder wall. Figs. 8 and 9 show the final periodic state of these two parameters for Re $=$ 100 and Re $=$ 200. It can be observed in Figs. 8 and 9 that lift and drag coefficients show obvious periodic oscillations for both Re $=$ 100 and 200 cases. This implies the periodic variation of flow field. From Figs. 8 and 9, it can also be found that the lift coefficient oscillates with larger amplitude than the drag coefficient, and the drag coefficient varies twice as fast as the lift coefficient. These phenomena are consistent with those observed by other researchers. The reason lies



(a) Re $=$ 100 (b) Re $=$ 200

Figure 6: Near wake structures of instantaneous streamlines for Re $=$ 100, 200.



(a) Re $=$ 100 (b) Re $=$ 200

Figure 7: Near wake structures of instantaneous vorticity contours for Re $=$ 100, 200.

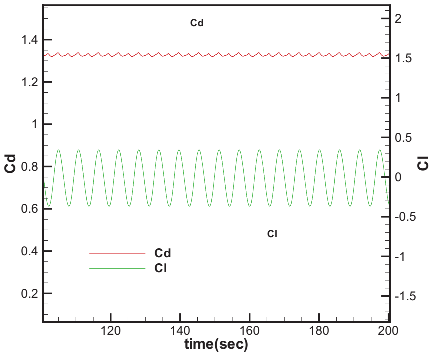


Figure 8: The time-evolution of Lift and Drag coefficients for Re = 100.

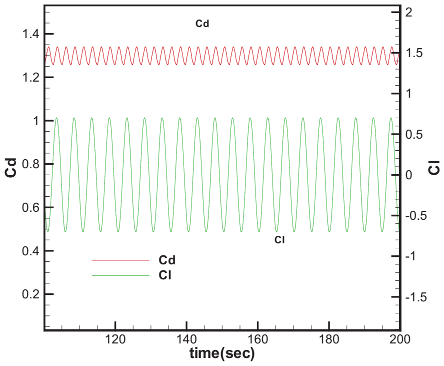


Figure 9: The time-evolution of Lift and Drag coefficients for Re = 200.

in the fact that the drag coefficient is affected by vortex shedding process from both sides of the cylinder.

The average drag coefficient and lift coefficient and Strouhal number ($St = fD/U$, where f is the shedding frequency) are calculated and listed in Table 2. The vortex shedding frequency is obtained by measuring the final period of the lift coefficient. In Table 2, the present results are quantitatively compared with numerical results from other researchers for Re = 100 and 200, respectively. It can be observed that good agreement has been achieved.

Table 2: Comparison of dynamical parameters with previous studies for Re = 100 and 200.

		C_D	C_L	St
Re = 100	Braza et al. [24]	1.364 ± 0.015	± 0.25	0.160
	Liu et al. [25]	1.350 ± 0.012	± 0.339	0.164
	Ding et al. [26]	1.325 ± 0.008	± 0.28	0.164
	Present results	1.330 ± 0.010	± 0.341	0.166
Re = 200	Braza et al. [24]	1.40 ± 0.05	± 0.75	0.200
	Liu et al. [25]	1.31 ± 0.049	± 0.69	0.192
	Ding et al. [26]	1.327 ± 0.045	± 0.60	0.196
	Present results	1.30 ± 0.041	± 0.69	0.200

4 Conclusions

The major advantage of conventional IBM is its simplicity and easy implementation. It solves the governing equations in the whole domain and decouples the solution of governing equations with the implementation of boundary conditions at immersed boundaries. However, it suffers two major drawbacks. One is the flow penetration due to less accurate satisfaction of non-slip boundary conditions. The other is low order accuracy near the immersed boundary due to the use of Dirac delta function. As shown in this paper, these two drawbacks can be removed in the proposed local DFD-IBM, which incorporates the idea of IBM (solving governing equations in the whole domain) into local DFD method. As the non-slip boundary condition is accurately implemented in the local DFD-IBM, the flow penetration often appeared in the IBM results is avoided. This is confirmed by its application to simulate flows around a circular cylinder. The present numerical results are also in good agreement with other experimental and numerical data in the literature. It seems that the present approach has a potential to become an effective solver for simulation of incompressible viscous flows.

References

[1] C. S. PESKIN, *Numerical analysis of blood flow in the heart*, J. Comput. Phys., 25 (1977), pp. 220–252.

[2] D. GOLDSTEIN, R. HADLER AND L. SIROVICH, *Modeling a no-slip flow boundary with an external force field*, J. Comput. Phys., 105 (1993), pp. 354–366.

[3] M. LAI AND C. S. PESKIN, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, J. Comput. Phys., 160 (2000), pp. 705–719.

[4] M. N. LINNICK AND H. F. FASEL, *A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains*, J. Comput. Phys., 204 (2005), pp. 157–192.

[5] E. LIMA, A. L. F. SILVA, A. SILVERIRA-NETO AND J. J. R. DAMASCENO, *Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method*, J. Comput. Phys., 189 (2003), pp. 351–370.

[6] Z. G. FENG AND E. E. MICHAELIDES, *Proteus: a direct forcing method in the simulations of particulate flow*, J. Comput. Phys., 202 (2005), pp. 20–51.

[7] S. CHEN AND G. D. DOOLEN, *Lattice Boltzmann method for fluid flows*, Ann. Rev. Fluid. Mech., 30 (1996), pp. 329–364.

[8] X. D. NIU, C. SHU, Y. T. CHEW AND Y. PENG, *A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows*, Phys. Lett. A., 354 (2006), pp. 173–182.

[9] Y. PENG, C. SHU, Y. T. CHEW, X. D. NIU AND X. Y. LU, *Application of multi-block approach in the immersed boundary-lattice Boltzmann method for viscous fluid flows*, J. Comput. Phys., 218 (2006), pp. 460–478.

[10] C. SHU, N. Y. LIU AND Y. T. CHEW, *A novel immersed boundary velocity correction-lattice Boltzmann method and its application to simulate flow past a circular cylinder*, J. Comput. Phys., 226 (2007), pp. 1607–1622.

[11] J. Wu and C. Shu, *Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications*, J. Comput. Phys., 228 (2009), pp. 1963–1979.

[12] C. Shu and W. L. Wu, *Adaptive mesh refinement-enhanced local DFD method and its application to solve Navier-Stokes equations*, Int. J. Numer. Meth. Fluids., 51 (2006), pp. 897–912.

[13] C. Shu and L. F. Fan, *A new discretization method and its application to solve incompressible Navier-Stokes equation*, Comput. Mech., 27 (2001), pp. 292–301.

[14] C. Shu and Y. L. Wu, *Domain-free discretization method for doubly connected domain and its application to simulate natural convection in eccentric annuli*, Comput. Methods. Appl. Mech. Eng., 191 (2002), pp. 1827–1841.

[15] Y. L. Wu and C. Shu, *Application of local DFD method to simulate unsteady flows around an oscillating circular cylinder*, Int. J. Numer. Meth. Fluids., 58 (11) (2008), pp. 1223–1236.

[16] J. Kim and P. Moin, *Application of fractional-step method to incompressible Navier-Stokes equations*, J. Comput. Phys., 59 (1985), pp. 308–323.

[17] H. Ding and C. Shu, *A stencil adaptive algorithm for finite difference solution of incompressible viscous flows*, J. Comput. Phys., 214 (2006), pp. 397–420.

[18] J. Wu and C. Shu, *Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications*, J. Comput. Phys., 228 (2009), pp. 1963–1979.

[19] S. C. R. Dennis and G. Z. Chang, *Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100*, J. Fluid. Mech., 42 (1970), pp. 471–489.

[20] X. Y. He and G. D. Doolen, *Lattice Boltzmann method on a curvilinear coordinate system: vortex shedding behind a circular cylinder*, Phys. Rev., 56 (1997), pp. 434–440.

[21] D. Calhoun, *A Cartesian grid method for solving the two-dimensional stream function-vorticity equatins in irregular regions*, J. Comput. Phys., 176 (2002), pp. 231–275.

[22] S. Y. Tuann and M. D. Olson, *Numerical studies of the flow around a circular cylinder by a finite element method*, Comput. Fluid., 6 (1978), pp. 219–240.

[23] H. Ding, C. Shu and Q. D. Cai, *Applications of stencil-adaptive finite difference method to incompressible viscous flows with curved boundary*, Comput. Fluids., 36 (2007), pp. 786–793.

[24] M. Braza, P. Chassaing and H. Ha Minh, *Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder*, J. Fluid. Mech., 165 (1986), pp. 79–130.

[25] C. Liu, X. Zheng and C. H. Sung, *Preconditioned multigrid metrhods for unsteady incompressible flows*, J. Comput. Phys., 139 (1998), pp. 39–57.

[26] H. Ding, C. Shu, K. S. Yeo and D. Xu, *Simulation of incompressible viscous flows past circular cylinder by hybrid FD scheme and meshless least square-based finite difference method*, Comput. Methods. Appl. Mech. Eng., 193 (2004), pp. 727–744.