Abstract
The immersed boundary (IB) method is a novel strategy to treat the boundary condition of a solid immersed in a fluid. In the original IB method, the forces exerted by the solid on the fluid are spread to grid points in the vicinity of the solid boundary in order to account for the effect of the solid. Then the Navier-Stokes equations with additional body forces are solved on a Cartesian mesh. This treatment can get reasonable velocity distribution near the fluid-solid interface. The primary advantage of the IB method is that the grid generation is greatly simplified and the mesh movement/regeneration is avoided.

Since the initial idea of the IB method, many additional features have been developed to enhance the capability and to improve the performance of the method. In this paper, we will introduce our recent developments for the IB solvers: the IB method based on the lattice Boltzmann method, the sharp-interface IB method based on the finite difference method, and extensions to other physical equations. A variety of applications of the IB solvers will be demonstrated. The applications include insect flight, fish swimming, red blood cells, fluid-structure interaction during phonation, heat transfer and electrodynamics.

Introduction
Numerical methods based on fixed grids have attracted growing interest in recent years due to their advantages in handling complex/moving boundaries. The immersed boundary (IB) method, which is the most notable among them, has gained popularity for a wide range of applications in recent years. The IB method, first developed by Peskin [8], is a novel strategy to treat the boundary condition of a solid immersed in a fluid. In the original version of the IB method, a continuous force is distributed as a body-force term onto the volumetric mesh in the vicinity of the boundary in order to account for the effect of the boundary. The Navier-Stokes equations with additional body forces are then discretized on a fixed Cartesian grid. Later, several families of the IB method have been developed, examples are the direct forcing approach based on local flow reconstruction in [7,21,24] and the projection approach by Taira & Colonius [12]. The underlying ideas of these works are very different depending on the specific implementations. Nevertheless, all of these versions of the method are able to treat the irregular and time-varying boundaries using a fixed, single-block Cartesian grid. Therefore, they share the merits of simple grid generation, efficient computation on the structured grid, and easy partition based on domain decomposition.

Immersed boundary–lattice Boltzmann method
In this method, the fluid flow is solved by the lattice Boltzmann (LB) method. In the present LB method, the kinematics of the fluid is governed by the discrete LB equation of a single relaxation time [2,17],

\[ g_i(x + \mathbf{e}_i \Delta t, t + \Delta t) - g_i(x, t) = -\frac{1}{\tau} [g_i(x, t) - g_i^eq(x, t)] + \Delta t G_i, \quad (1) \]

where \( g_i(x, t) \) is the distribution function for particles with velocity \( \mathbf{e}_i \) at position \( x \) and time \( t \), \( \Delta t \) is the size of the time step, \( g_i^{eq}(x, t) \) is the equilibrium distribution function, \( \tau \) represents the non-dimensional relaxation time, and \( G_i \) is the term representing the body force effect on the distribution function. In Eq. (1), \( G_i \) and \( G_i \) are calculated according to

\[ g_i^{eq} = \omega_0 \rho \left[ 1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} \right] \mathbf{u} : \left( \mathbf{e}_i \mathbf{e}_i - c_s^2 I \right), \quad (2) \]

\[ G_i = \left( 1 - \frac{1}{2\tau} \right) \omega_0 \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} \mathbf{e}_i + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} \mathbf{f}, \quad (3) \]

where \( \omega_0 \) are the weighing factors, \( \mathbf{u} \) is the velocity of the fluid, \( c_s \) is the speed of sound defined by \( c_s = \Delta t / \sqrt{3 \Delta x} \), and \( \mathbf{f} \) is the body force acting on the fluid. The relaxation time \( \tau \) is related to the kinematic viscosity \( \nu \) in the Navier–Stokes equations:

\[ \nu = (\nu - 0.5) c_s^2 \Delta t. \quad (4) \]

In the 2D nine-speed (D2Q9) model [11], as shown in figure 1(a), the nine particle velocities are given by

\[ \mathbf{e}_0 = (0, 0), \]

\[ \mathbf{e}_i = \left( \frac{\cos \left( \frac{\pi (i-1)}{2} \right)}{2}, \frac{\sin \left( \frac{\pi (i-1)}{2} \right)}{2} \right) \frac{\Delta x}{2}, \quad i = 1-4, \]

\[ \mathbf{e}_i = \left( \frac{\cos \left( \frac{\pi (i - 9/2)}{2} \right)}{2}, \frac{\sin \left( \frac{\pi (i - 9/2)}{2} \right)}{2} \right) \sqrt{2} \Delta x, \quad i = 5-8, \]

where \( \Delta x \) is the lattice spacing. The weight factors are given by \( \omega_0 = 4/9, \omega_i = 1/9 \) for \( i = 1 \) to \( 4 \) and \( \omega_i = 1/36 \) for \( i = 5 \) to \( 8 \). In the 3D nineteen-speed (D3Q19) model [11] shown in figure 1(b), the particle velocities are defined by

\[ \mathbf{e}_0 = (0, 0, 0), \]

\[ \mathbf{e}_i = \left[ \left( \pm 1, 0, \pm 1 \right), \left( 0, \pm 1, \pm 1 \right) \right] \frac{\Delta x}{\Delta t}, \quad i = 1-6, \]

\[ \mathbf{e}_i = \left[ \left( \pm 1, \pm 1, 0 \right), \left( \pm 1, 0, \pm 1 \right), \left( 0, \pm 1, \pm 1 \right) \right] \frac{\Delta x}{\Delta t}, \quad i = 7-18. \]

The weight factors of D3Q19 model take the values \( \omega_0 = 1/3, \omega_i = 1/18 \) for \( i = 1 \) to \( 6 \) and \( \omega_i = 1/36 \) for \( i = 7 \) to \( 18 \). The values of \( \mathbf{e}_i \) ensure that within one time step, a fluid particle moves to one of the neighboring nodes as shown in figure 1, or stays at its current location.

Once the particle density distribution is known, the fluid density, velocity and pressure are then computed from

\[ p = \sum_i g_i, \quad \mathbf{u} = \frac{\sum_i \mathbf{e}_i g_i + 4 \Delta t \Delta \mathbf{u}}{\rho}, \quad \rho = \rho c_s^2. \quad (5) \]
where \( T \) and \( B \) representing 9 possible velocity directions in the D2Q9 lattice model.

\[
E_p = \frac{1}{2} \sum_{i,j=1}^{n_d} |\tau_{ij}(T_{ij} - \delta_{ij})|^2 + \gamma_j (B_{ij})^2 dA,
\]

\[
E_k = \frac{1}{2} m_p \int \frac{\partial^2 \mathbf{X}}{\partial t^2} dA,
\]

where \( m_p \) is the density of the plate, \( n_d \) is the dimension of the plate, \( T_{ij} = (\partial \mathbf{X}/\partial s_i, \partial \mathbf{X}/\partial s_j)^{1/2} \) is the stretching and shearing effects, \( B_{ij} = (\partial^2 \mathbf{X}/\partial s_i \partial s_j, \partial^2 \mathbf{X}/\partial s_i \partial s_j)^{1/2} \) is the bending and twisting coefficients, and summation convention is not applied on both \( i \) and \( j \). For 1D filament, \( n_d = 1 \) and the integral is taken along the filament, while for 2D plate, \( n_d = 2 \) and the integral is taken over the whole plate. The potential energy of external force (hydrodynamic force in the present work) is expressed by

\[
E_h = \int F_f \cdot (\mathbf{X} - \mathbf{X}^0) dA,
\]

where \( \mathbf{X}^0 \) is the initial position of the structure. Then the total potential energy of the plate can be defined by \( \Pi = E_k - E_p + E_h \). By using the Hamilton’s principle together with the variational derivative of the total potential energy, the governing equation for the plate is obtained,

\[
m_p \frac{\partial^2 \mathbf{X}}{\partial t^2} + \sum_{i,j=1}^{n_d} \left[ -\frac{\partial}{\partial s_i} \left( \tau_{ij} \frac{\partial \mathbf{X}}{\partial s_j} \right) + \frac{\partial^2}{\partial s_i \partial s_j} \left( B_{ij} \frac{\partial^2 \mathbf{X}}{\partial s_i \partial s_j} \right) \right] = F_f,
\]

where \( \tau_{ij} = \psi_{ij}(1 - \delta_{ij}/T_{ij}) \). Note that Eq. (9) is equivalent to that in Ref. [17,25] for 1D filament and Ref. [5,16] for 2D plate if the stretching and shearing effects are small.

Explicitly including the inertial force of the structure in the IB method when calculating the hydrodynamic stress on the solid surface may easily destabilize the simulation. To address this issue, we have incorporated a penalty IB method in the LB method [17]. In this method, the plate itself is assumed to be massless, but a ghost plate of density \( m_p \) is attached to the physical plate through virtual springs of stiffness \( K_s \) (see figure 2). The ghost plate only affects the dynamics of the physical plate but is not seen by the flow solver directly. Thus, the density of the fluid is a constant \( \rho \), and the Lagrangian force is modified to incorporate the ghost plate,

\[
\mathbf{F} = \mathbf{F}_k + \mathbf{F}_e, \quad \mathbf{F}_k = K_s(\mathbf{X} - \mathbf{X}_0), \quad m_p \frac{\partial^2 \mathbf{X}}{\partial t^2} = -\mathbf{F}_k,
\]

where \( \mathbf{F}_k \) is the spring force, \( \mathbf{F}_e \) is the elastic force given by

\[
\mathbf{F}_e = \sum_{i,j=1}^{n_d} \left[ \sigma_{ij} \frac{\partial \mathbf{X}}{\partial s_i} \right] - \frac{\partial^2}{\partial s_i \partial s_j} \left( \gamma_j \frac{\partial^2 \mathbf{X}}{\partial s_i \partial s_j} \right) \cdot K_s \text{ is the stiffness of virtual springs, and } \mathbf{Y} \text{ is the position vector of the point on the ghost plate connecting to point } \mathbf{X} \text{ on the physical plate. Essentially, the effect of the inertia of the ghost plate is cushioned through the virtual springs. This method has been used to simulate hydrodynamic interaction between the flexible plates/capsules and the incompressible viscous flow [13,17–19].}

The Lagrangian velocity, position of the physical plate, and body force can be discretized as follows,

\[
\mathbf{U}^{n+1} = \sum_x \mathbf{u}_x^{n+1}(x,t) \delta_h(x - \mathbf{X}^n) \Delta V, \quad \mathbf{X}^{n+1} = \mathbf{X}^n + \mathbf{U}^{n+1} \Delta t, \quad \mathbf{F}^{n+1} = \sum_s \mathbf{F}^{n+1}_s(s,t) \delta_h(s - \mathbf{x}) \Delta A.
\]

In Eq. (12), \( \Delta A = \Delta x \) and \( \Delta s \Delta t \) for the 1D and 2D plates, respectively, while in Eq. (11), \( \Delta V = \Delta x \Delta y \) and \( \Delta V = \Delta x \Delta y \Delta z \) for the 2D and 3D simulations, respectively. The notations \( \Sigma_x \) and \( \Sigma_s \) mean the sum over all the discrete points of \( x \) and the sum over all the discrete points of \( s \), respectively. \( \delta_h \) is a smooth approximation of Dirac’s delta function [9,10]. This version of IB-LB method has been used in fish swimming, flag flapping and red blood cell [3,13,17–19,22,23], as shown in figure 3.

**Sharp-interface immersed boundary–finite element method**

In this section, we introduce the sharp-interface IB–finite element (FE) method for 3D FSI involving large deformations. The IB method for fluid solver was previously developed by...
Figure 4: 2D schematics illustrating the sharp-interface immersed boundary method: (a) interpolation stencil, and (b) extrapolation stencil [3].

Mittal et al. [7] and later improved by Lao et al. [6] and Tian et al. [14, 15]. This method retains the sharp-interface representation of the fluid–solid interface and employs local flow reconstruction to facilitate the finite difference discretization near the immersed boundary. The 2D schematics of this method is shown in figure 4. When the second-order central difference scheme is applied to discretize the Navier–Stokes equations in the fluid region near the fluid–solid interface, incomplete schemes could be encountered. Specifically, on the node \((i, j)\) in figure 4(a), the finite difference stencil will involve nodes \((i-1, j)\) and \((i, j-1)\) that are located inside the solid body. Here we introduce two methods to update the variables on node \((i, j)\).

In the first method, the variables on \((i, j)\) are interpolated by using the interpolation stencil shown in figure 4(a). A body intercept (BI) point can be found by projecting the \((i, j)\) onto the boundary along the surface normal. The variable, \(\Phi\), in the local area around \((i, j)\) is approximated by \(\Phi = a_1x + a_2y + a_3\) where \(a_1, a_2, a_3\) and \(a_4\) can be determined by using the values on BI, together with \((i+1,j), (i, j+1), (i+1, j+1)\). Then the value on \((i, j)\) can be obtained by \(\Phi_{i,j} = \sum B_m \Phi_m\) where \(\Phi_m\) is one of the 4 data points. In the second method, we first apply the extrapolation, and then use the finite difference method. As shown in figure 4(b), to calculate the value on \((i, j)\), the values on \((i-1, j)\) and \((i, j-1)\) are first extrapolated, and those on \((i, j)\) are then calculated by using the finite difference method. Take node \((i-1, j)\) as an example, the BI point can be determined by the same way as the interpolation. The image point (IP) can be found by taking the symmetrical point about the boundary. The value on IP can be determined by using the shaded stencil, i.e. the values at previous time step on BI, \((i, j), (i, j+1), (i-1, j+1)\). Then \(\Phi_{IP} = 2\Phi_{BD} - \Phi_{BI}\). Therefore, \(\Phi_{IP} + 1\) can be updated by the finite difference method. Iteration is required in the cases where the points used to interpolate the unknown values are in the solid region or immediately next to the interface. In the practice, the numerical oscillations in the moving boundary problems associated with the sudden change of the stencils can be effectively reduced by applying the hybrid scheme of these two methods [6, 14, 15].

The FE formulation in the structure solver is derived from the standard virtual work method. Let the displacement in a volume element be represented by \(u(X, Y, Z) = \sum h_k(X, Y, Z)u_k = [H'][u]\), where \(h_k(X, Y, Z)\) is the shape function associated with the \(k\)th node in the element and \(u_k\) is the displacement at this node. Other variables can be expanded in a similar manner.

Using the virtual work of the inertial load, body force \([b]\), and surface traction \([f]\) along with the expansion of the variables, the assembled equation system for the entire body can be writ-

Figure 5: Applications of the sharp-interface IB–FE method: (a) vortical structures around a hoverfly and (b) modelling of vocal-fold vibration [15, 20].

\[
\begin{bmatrix}
[M] & [C]
\end{bmatrix} \{u\} = \{P\} - \{F\},
\]

where \([M]\) is the mass matrix, \([C]\) is the mass-damping matrix, \([P]\) is the force vector from the external load, and \([F]\) is the body stress vector. These assembled terms can be found in Refs. [14, 15]. For general 3D bodies, hexahedral (or brick-type) quadratic 20-node elements [1] are used in the FE formulation. The FE formulation of the thin-walled structures includes the three-node plate elements and two-node frame elements, where each node has six global degrees of freedom, including three displacement components, \(u, v, w\), and three angles of rotation, \(\Phi\). The discrete equations can be written in the same forms as in the general structure form, except that \(\{u\}\) represents the generalized displacement vector with \(\Phi\) included and \([\sigma]\) represents the generalized stress with moments included. The large-displacement and small-strain deformation in the structure solver is handled using the corotational scheme. The time stepping is achieved using a case of Newmark scheme [4].

The fluid–structure coupling is done by iterating the two solvers by exchanging the boundary information until convergence is reached. The residuals as measured by the maximum errors of the displacement, the velocity, and the traction at the solid surface are used to determine whether final convergence is reached. To ensure the numerical stability of this staggered iteration, the velocity of the solid surface are updated in the flow solver in a gradual fashion according to \(v_{k+1} = \alpha v_k + (1 - \alpha)v_{k+1}\), where \(v_k\) is the predicted velocity by the structure solver and \(\alpha\) is the relaxation factor between 0 and 1.

This method has been used in insect flight, and FSI during phonation, as shown in figure 5.

Immersed boundary method for heat transfer and electro-dynamics

Heat transfer can be described by \(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = k \nabla^2 T\). To account for the effect of the boundary, a heat source is applied in the convection-diffusion equation, \(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = k \nabla^2 T + q\), where \(q\) is the heat source which is obtained by spreading Lagrangian heat source \(Q\) to the points near the boundary. For the Dirichlet boundary condition, \(Q\) at \(n + 1\) step is calculated by

\[
Q^{n+1} = (T_B - T^n)/\Delta t + \mathbf{u} \cdot \nabla T^n - k \nabla^2 T^n,
\]

where \(T_B\) is the boundary temperature. For the Neumann boundary condition, \(Q\) is determined by

\[
Q^{n+1} = 2(Q_B + \frac{\partial T^n}{\partial n}).
\]
In the electrodynamics applications, the Poisson equation, $V^2 \Phi = -q$, is generally used to describe the electric potential field $\Phi$. This equation can be rewritten as $\partial \Phi / \partial t = \kappa_0 (V^2 \Phi + q)$ where $t$ is the pseudo time. Then similar treatment as in heat transfer can be used.

Conclusions
The IB method based on LB method and FE method has been briefly introduced. The applications in insect flight, fish swimming, red blood cells, FSI during phonation, heat transfer and convection-diffusion processes.

Acknowledgements
This research was partly supported by the Australian Research Council’s Discovery Project Funding Scheme (project number DP130103850).

References


