IMMERSED BOUNDARY METHODS

Rajat Mittal
*Department of Mechanical and Aerospace Engineering, George Washington University,
Washington, D.C. 20052; email: mittal@gwu.edu*

Gianluca Iaccarino
*Center for Turbulence Research, Mechanical Engineering Department,
Stanford University, Stanford, California 94305; email: jops@ctr.stanford.edu*

1. INTRODUCTION

The term "immersed boundary method" was first used in reference to a method developed by Peskin (1972) to simulate cardiac mechanics and associated blood flow. The distinguishing feature of this method was that the entire simulation was carried out on a Cartesian grid, which did not conform to the geometry of the heart, and a novel procedure was formulated for imposing the effect of the immersed boundary (IB) on the flow. Since Peskin introduced this method, numerous modifications and refinements have been proposed and a number of variants of this approach now exist. In addition, there is another class of methods, usually referred to as "Cartesian grid methods," which were originally developed for simulating inviscid flows with complex embedded solid boundaries on Cartesian grids (Berger & Aftosmis 1998, Clarke et al. 1986, Zeeuw & Powell 1991). These methods have been extended to simulate unsteady viscous flows (Udaykumar et al. 1996, Ye et al. 1999) and thus have capabilities similar to those of IB methods. In this review, we use the term immersed boundary (IB) method to encompass all such methods that simulate viscous flows with immersed (or embedded) boundaries on grids that do not conform to the shape of these boundaries. Furthermore, this review focuses mainly on IB methods for flows with immersed solid boundaries. Application of these and related methods to problems with liquid-liquid and liquid-gas boundaries was covered in previous reviews by Anderson et al. (1998) and Scardovelli & Zaleski (1999).

Consider the simulation of flow past a solid body shown in Figure 1*a*. The conventional approach to this would employ structured or unstructured grids that conform to the body. Generating these grids proceeds in two sequential steps. First, a surface grid covering the boundaries Γ_b is generated. This is then used as a boundary condition to generate a grid in the volume Ω_f occupied by the fluid. If a finite-difference method is employed on a structured grid, then the differential form of the governing equations is transformed to a curvilinear coordinate system aligned with the grid lines (Ferziger & Peric 1996). Because the grid conforms to the surface of the body, the transformed equations can then be discretized in the
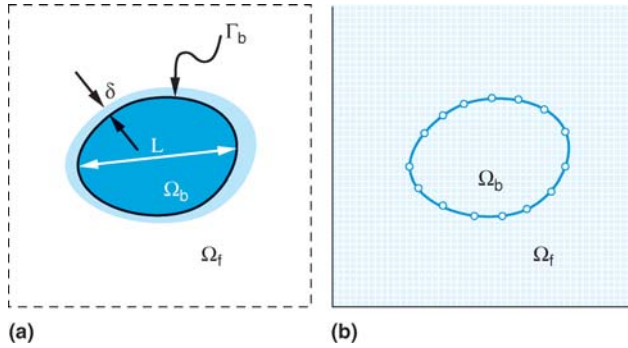
Figure 1 (*a*) Schematic showing a generic body past which flow is to be simulated. The body occupies the volume Ω_b with boundary Γ_b. The body has a characteristic length scale L, and a boundary layer of thickness δ develops over the body. (*b*) Schematic of body immersed in a Cartesian grid on which the governing equations are discretized.

computational domain with relative ease. If a finite-volume technique is employed, then the integral form of the governing equations is discretized and the geometrical information regarding the grid is incorporated directly into the discretization. If an unstructured grid is employed, then either a finite-volume or a finite-element methodology can be used. Both approaches incorporate the local cell geometry into the discretization and do not resort to grid transformations.

Now consider employing a nonbody conformal Cartesian grid for this simulation, as shown in Figure 1*b*. In this approach the IB would still be represented through some means such as a surface grid, but the Cartesian volume grid would be generated with no regard to this surface grid. Thus, the solid boundary would cut through this Cartesian volume grid. Because the grid does not conform to the solid boundary, incorporating the boundary conditions would require modifying the equations in the vicinity of the boundary. Precisely what these modifications are is the subject of a detailed discussion in subsequent sections. However, assuming that such a procedure is available, the governing equations would then be discretized using a finite-difference, finite-volume, or a finite-element technique without resorting to coordinate transformation or complex discretization operators.

At this point it is useful to compare the advantages and disadvantages of these two approaches. Clearly, imposing the boundary conditions is not straightforward in IB methods and, furthermore, the ramifications of the boundary treatment on the accuracy and conservation properties of the numerical scheme are not obvious. In addition, alignment between the grid lines and the body surface in body-conformal grids allows better control of the grid resolution in the vicinity of the body and this has implications for the increase in grid size with increasing Reynolds number. For example, consider the case of the two-dimensional (2D) body of characteristic length L in Figure 1*a* with a boundary layer of thickness δ wherein it is required

to provide an average grid spacing of Δ_n and Δ_t in the directions normal and tangential to the body surface. It is now easy to show that for moderately high Reynolds numbers for which $\delta \ll L$, the size of a body-conformal grid scales as $(L/\Delta_t)(\delta/\Delta_n)$, whereas that of a Cartesian grid scales as (L^2/Δ_n^2). Assuming further that $\Delta_n \propto \delta$ and $\Delta_t \propto L$, we find that the ratio of sizes for a Cartesian to a body-conformal grid will scale as $(L/\delta)^2$. For a laminar boundary layer, $(L/\delta) \propto Re^{0.5}$ (Schlichting & Gersten 1996), which implies that the grid-size ratio will scale with $(Re)^{1.0}$ for 2D bodies. For a three-dimensional (3D) body, it can similarly be shown that the grid-size ratio would scale with $(Re)^{1.5}$. Thus, as the Reynolds number increases, the size of a Cartesian grid increases faster than a corresponding body-conformal grid. However, note that this faster increase in grid size does not necessarily imply a corresponding increase in the computational cost because a substantial fraction of the grid points can be inside the solid body where the fluid flow equations need not be solved. This fraction is proportional to the ratio of the volume of the solid to the volume of the bounding box for the body and therefore depends on the shape and orientation of the body. Furthermore, in comparison with structured curvilinear body-formal grids, using a Cartesian grid can significantly reduce the per-grid-point operation count due to the absence of additional terms associated with grid transformations. When compared with unstructured grid methods, the Cartesian grid–based IB method retains the advantage of being amenable to powerful line-iterative techniques and geometric multigrid methods, which can also lead to a lower per-grid-point operation count.

The primary advantage of the IB method is associated with the fact that the task of grid generation is greatly simplified. Generating body-conformal structured or unstructured grid is usually very cumbersome. The objective is to construct a grid that provides adequate local resolution with the minimum number of total grid points. For anything but the simplest geometries, these conflicting requirements can lead to a deterioration in grid quality, which can negatively impact the accuracy and the convergence properties of the solver (Ferziger & Peric 1996). Thus, even for simple geometries, generating a good-quality body-conformal grid can be an iterative process requiring significant input from the person generating the grid. As the geometry becomes more complicated, the task of generating an acceptable grid becomes increasingly difficult. Within the structured grid approach, complex geometries are often handled by decomposing the volume into subdomains and generating a separate grid in each subdomain (Quarteroni & Valli 1999). Apart from the complexity that is introduced into the solution algorithm due to the presence of multiple subdomains, grid smoothness can deteriorate at the interface between subdomains. The unstructured grid approach is inherently better suited for complex geometries, but here, too, grid quality can deteriorate with increasing complexity in the geometry. In contrast, for a simulation carried out on a nonbody conformal Cartesian grid, grid complexity and quality are not significantly affected by the complexity of the geometry.

The advantage of the Cartesian grid–based IB method also becomes eminently clear for flows with moving boundaries. Simulating such flows on body-conformal

grids requires generating a new grid at each time step as well as a procedure to project the solution onto this new grid (Tezduyar 2001). Both of these steps can negatively impact the simplicity, accuracy, robustness, and computational cost of the solution procedure, especially in cases involving large motions. In contrast, including body motion in IB methods is relatively simple due to the use of a stationary, nondeforming Cartesian grid. Thus, despite the significant progress made in simulating flow with moving boundaries on body-conformal grids (Baum et al. 1998, Ramamurti & Sandberg 2001, Tezduyar 2001), due to its inherent simplicity, the IB method represents an extremely attractive alternative for such flows.

2. IMPOSITION OF BOUNDARY CONDITION ON IMMERSED BOUNDARIES

Imposition of boundary conditions on the IB is the key factor in developing an IB algorithm. It is also what distinguishes one IB method from another. Consider the simulation of incompressible flow past the body in Figure 1b, which is governed by the following equations:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho}\nabla p - \frac{\mu}{\rho}\nabla^2 \vec{u} = 0 \quad \text{and} \tag{1}$$

$$\nabla \cdot \vec{u} = 0 \quad \text{in} \quad \Omega_f$$

$$\text{with} \quad \vec{u} = \vec{u}_\Gamma \quad \text{on} \quad \Gamma_b, \tag{2}$$

where $\vec{u}$ is the the fluid velocity, p is the pressure, and ρ and μ the density and viscosity, respectively. The solid body occupies the domain Ω_b, with boundary denoted by Γ_b, and the surrounding fluid domain denoted by Ω_f. The outer boundary of the flow domain is disregarded for the purposes of this discussion. For ease of discussion, the coupled system of momentum and continuity equations can be notionally written as

$$\mathcal{L}(\underline{U}) = 0 \quad \text{in} \quad \Omega_f \tag{3}$$

$$\text{with} \quad \underline{U} = \underline{U}_\Gamma \quad \text{on} \quad \Gamma_b, \tag{4}$$

where $\underline{U} = (\vec{u}, p)$ and $\mathcal{L}$ is the operator representing the Navier-Stokes equations as in Equation 1. It should be noted that in the context of the imcompressible Navier-Stokes equations, pressure is determined by the continuity constraint and, consequently, the continuity equation is considered an implicit equation for pressure. A number of numerical integration schemes such as fractional-step (Chorin 1968) and SIMPLE (Patankar 1980) also explicitly derive and solve a Poisson equation for pressure, which, depending on the particular implementation, also requires appropriate boundary conditions.

Conventional methods proceed by developing a discretization of Equation 3 on a body-conformal grid where the boundary condition (Equation 4) on the

immersed boundary Γ_b is enforced directly. In an IB method, Equation 1 would be discretized on a nonbody conformal Cartesian grid and the boundary condition would be imposed indirectly through modification of Equation 3. In general, the modification takes the form of a source term (or forcing function) in the governing equations that reproduces the effect of boundary. Introduction of a forcing function, the precise nature of which will be discussed in the following sections, into the governing equations can be implemented in two different ways and this leads to a fundamental dichotomy in IB methods. In the first implementation, the forcing function, denoted here by $\underline{f}_b$, is included into the continuous governing Equation 3 leading to the equation $\mathcal{L}(\underline{U}) = \underline{f}_b$, which then applies to the entire domain $(\Omega_f + \Omega_b)$. Note that $\underline{f}_b = (\vec{f}_m, f_p)$ where $\vec{f}_m$ and f_p are the forcing functions applied to the momentum and pressure, respectively. This equation is subsequently discretized on a Cartesian grid, leading to the following system of discrete equations:

$$[L]\{\underline{U}\} = \{\underline{f}_b\}, \tag{5}$$

and this system of equations is solved in the entire domain.

In the second approach, the governing equations are first discretized on a Cartesian grid without regard to the immersed boundary, resulting in the set of discretized equations $[L]\{\underline{U}\} = 0$. Following this, the discretization in the cells near the IB is adjusted to account for its presence, resulting in a modified system of equations $[L']\{\underline{U}\} = \{\underline{r}\}$, which are then solved on the Cartesian grid. In this equation, $[L']$ is the modified discrete operator and $\{\underline{r}\}$ represents known terms associated with the boundary conditions on the immersed surface. The above system of equations can be rewritten as

$$[L]\{\underline{U}\} = \{\underline{f}'_b\}, \tag{6}$$

where $\{\underline{f}'_b\} = \{\underline{r}\} + [L]\{\underline{U}\} - [L']\{\underline{U}\}$. Comparing Equations 5 and 6 clearly shows the connection between the two approaches. In the first approach, which we term "continuous forcing approach," the forcing is incorporated into the continuous equations before discretization, whereas in the second approach, which can be termed the "discrete forcing approach," the forcing is introduced after the equations are discretized. An attractive feature of the continuous forcing approach is that it is formulated independent of the underlying spatial discretization. On the other hand, the discrete forcing approach very much depends on the discretization method. However, this allows direct control over the numerical accuracy, stability, and discrete conservation properties of the solver. In the following sections we describe methods that fall into each of these categories.

3. CONTINUOUS FORCING APPROACH

Among existing methods in this category, elastic and rigid boundaries require different treatments and we discuss these separately in this section.

3.1. Flows with Elastic Boundaries

The original IB method by Peskin (1972, 1981) was developed for the coupled simulation of blood flow and muscle contraction in a beating heart and is generally suitable for flows with immersed elastic boundaries. In this method the fluid flow is governed by the incompressible Navier-Stokes equations and these are solved on a stationary Cartesian grid. The IB is represented by a set of elastic fibers and the location of these fibers is tracked in a Lagrangian fashion by a collection of massless points that move with the local fluid velocity. Thus, the coordinate $\vec{X}_k$ of the kth Lagrangian point is governed by the equation

$$\frac{\partial \vec{X}_k}{\partial t} = \vec{u}(\vec{X}_k, t). \tag{7}$$

The stress (denoted by $\vec{F}$) and deformation of these elastic fibers is related by a constitutive law such as the Hooke's law. The effect of the IB on the surrounding fluid is essentially captured by transmitting the fiber stress to the fluid through a localized forcing term in the momentum equations, which is given by

$$\vec{f}_m(\vec{x}, t) = \sum_k \vec{F}_k(t)\delta(|\vec{x} - \vec{X}_k|), \tag{8}$$

where δ is the Dirac delta function. Because the location of the fibers does not generally coincide with the nodal points of the Cartesian grid, the forcing is distributed over a band of cells around each Lagrangian point (see Figure 2a) and this distributed force is imposed on the momentum equations of the surrounding nodes. Thus, the sharp delta function is essentially replaced by a smoother distribution function, denoted here by d, which is suitable for use on a discrete mesh. Due to
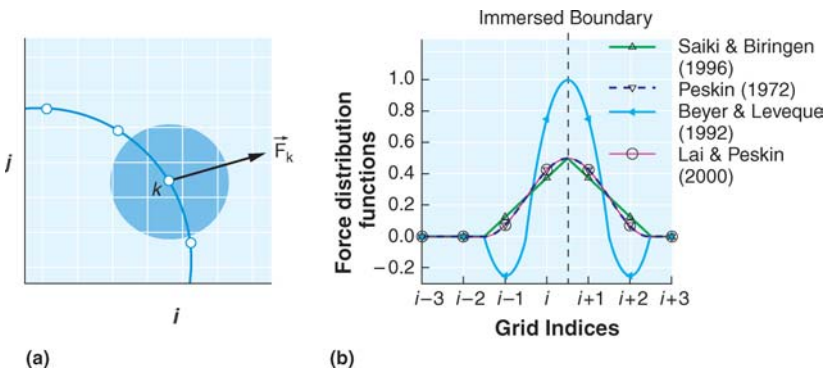
Figure 2 (a) Transfer of forcing $\vec{F}_k$ from Lagrangian boundary point $(\vec{X}_k)$ to surrounding fluid nodes. Shaded region signifies the extent of the force disctribution. (b) Distribution functions employed in various studies.

the fibers, the forcing at any grid point $x_{i,j}$ is then given by

$$\vec{f}_m(\vec{x}_{i,j}, t) = \sum_k \vec{F}_k(t) d(|\vec{x}_{i,j} - \vec{X}_k|). \tag{9}$$

The fiber velocity in Equation 7 is also obtained through the use of the same smooth function. The choice of the distribution function d is a key ingredient in this method. Several different distribution functions were employed in the past (Beyer & Leveque 1992, Lai & Peskin 2000, Peskin 1972, Saiki & Biringen 1996) and Figure 2b shows four such functions. For a simple one-dimensional model problem, Beyer & Leveque (1992) showed that it is possible to design a distribution function that preserves the accuracy of the spatial scheme. Methods in this category have been successfully used for a wide variety of problems including cardiac mechanics (Peskin 1981), cochlear dynamics (Beyer 1992), aquatic animal locomotion (Fauci & McDonald 1994), bubble dynamics (Unverdi & Tryggvason 1992), and flow past flexible filaments (Zhu & Peskin 2003).

3.2. Flows with Rigid Boundaries

The previous method is naturally well suited for elastic bodies but its application to flows with rigid bodies poses problems because the constitutive laws used for elastic boundaries are not generally well posed in the rigid limit. This problem could be circumvented by considering the body to be elastic but extremely stiff. A second approach is to consider the structure attached to an equilibrium location (Beyer & Leveque 1992, Lai & Peskin 2000) by a spring with a restoring force $\vec{F}$ given by

$$\vec{F}_k(t) = -\kappa \left(\vec{X}_k - \vec{X}_k^e(t) \right) \tag{10}$$

where κ is a positive spring constant and $\vec{X}_k^e$ is the equilibrium location of the kth Lagrangian point. Accurately imposing the boundary condition on the rigid IB requires large values of κ. However, this results in a stiff system of equations that is subject to severe stability constraints (Lai & Peskin 2000, Stockie & Wetton 1998). On the other hand, lower values of κ can lead to spurious elastic effects such as excessive deviation from the equilibrium location as noted in the low Reynolds cylinder wake simulations of Lai & Peskin (2000).

The above approach can be viewed as a specific version of the model developed by Goldstein et al. (1993) to simulate the flow around rigid bodies. In this model, the effect of the rigid body on the surrounding flow is modeled through a forcing term

$$\vec{F}(t) = \alpha \int_0^t \vec{u}(\tau) d\tau + \beta \vec{u}(t), \tag{11}$$

where the coefficients α and β are selected to best enforce the boundary condition at the immersed solid boundary. The original intent behind Equation 11 was to provide feedback control of the velocity near the surface (Goldstein et al. 1993), but from a physical point of view, it can also represent a damped oscillator

(Iaccarino & Verzicco 2003). This method has been used to simulate start-up flow past a circular cylinder at a moderate Reynolds number and a low Reynolds number turbulent flow in a channel with streamwise grooves (Goldstein et al. 1993). In general, results are promising at low Reynolds numbers but accurately enforcing the boundary conditions requires large values of α and β, which can lead to stability problems, especially for highly unsteady flows.

Another method in this class is the one developed by Angot et al. (1999) and Khadra et al. (2000). In this method, the entire flow is assumed to occur in a porous medium and is therefore governed by the Navier-Stokes/Brinkman equations (Brinkmann 1947). These equations contain an extra force term with respect to the classical Navier-Stokes equations of the form $\vec{F} = (\mu/K)\vec{u}$. Here K is the permeability of the medium and is defined as infinity or zero for fluid and solid regions, respectively. The force therefore activates only within the solid, driving the velocity field to zero. In practice, K is large (small) in fluid (solid) regions and this, along with the smoothing of the variation of K at the fluid-solid interface, leads to an error in the imposition of the correct velocity on the solid surface. The similarity between this and the previous forcing approach is evident because it is essentially equivalent to choosing $\alpha \equiv 0$ and $\beta \equiv \mu/K$. As such, this method is also subject to stiffness problems associated with large variations in the values of K. This method has been used to simulate flow past a circular cylinder for Reynolds numbers up to 200 and for flow over a backward facing step (Khadra et al. 2000). In direct contrast to this approach, in which the fluid is considered as a solid with infinite porosity, the approach of Glowinski et al. (1994) treats the solid as a fluid subject to a rigidity constraint, which can also be interpreted as a forcing term in the governing equations (Patankar 2001).

3.3. General Considerations

The continuous forcing approach is very attractive for flows with immersed elastic boundaries. For such flows, the method has a sound physical basis and is simple to implement. Consequently, many of the applications of these methods are found in biological (Beyer & Leveque 1992, Fauci & McDonald 1994, Peskin 1981) and multiphase flows (Unverdi & Tryggvason 1992) where elastic boundaries abound. However, using this approach for flows with rigid bodies poses some challenges associated with the fact that the forcing terms used are generally not well behaved in the rigid limit. This problem is essentially tackled by employing simplified models that attempt to mimic the effect of the solid boundary on the flow. The parameters introduced in these models, however, have implications for numerical accuracy as well as stability. The smoothing of the forcing function inherent in these approaches also leads to an inability to provide a sharp representation of the IB and this can be especially undesirable for high Reynolds number flows. Finally, all of these methods require the solution of the governing equations inside the immersed body. As noted earlier, with increasing Reynolds numbers, the proportion of grid points inside the IB also increases, and the requirement of solving the equations inside the solid can be a burdensome overhead.

4. DISCRETE FORCING APPROACH

In this section, methods are categorized into those that are formulated so as to impose the boundary condition on the immersed boundary through indirect means, and those that directly impose the boundary conditions on the IB.

4.1. Indirect BC Imposition

For a simple, analytically integrable, one-dimensional model problem, it is possible to formally derive a forcing term that enforces a specific condition on a boundary inside the computational domain (Beyer & Leveque 1992). The same is not usually feasible for the Navier-Stokes equations because the equations cannot be integrated analytically to determine the forcing function. Consequently, all the approaches in the previous section employ what are essentially simplified models of the required forcing. To avoid this issue, Mohd-Yosuf (1997) and Verzicco et al. (2000) developed a method that extracts the forcing directly from the numerical solution for which an *a priori* estimate can be determined. Starting from the discretized Navier-Stokes equations without any modification due to the presence of the IB, and using the same notation introduced in section 2, the system $[L]\{\underline{U}^*\} = 0$ is solved at each time step where $\{\underline{U}^*\}$ represents a prediction of the velocity field. The forcing $\{\underline{f}'_b\}$ in Equation 6 is then defined as

$$\{\underline{f}'_b\} \approx \{\underline{r}\} + [L]\{\underline{U}^*\} - [L']\{\underline{U}^*\} = \{\underline{r}\} - [L']\{\underline{U}^*\}, \tag{12}$$

where $\{\underline{r}\} = \{\underline{U}_\Gamma\}\delta(|\vec{X}_k - \vec{x}_{i,j}|)$ and $[L'] = [L] + ([I] - [L])\delta(|\vec{X}_k - \vec{x}_{i,j}|)$, $[I]$ being the identity matrix. As before, the Dirac delta function is replaced by a smooth distribution function d and the Equation 6 for this method then becomes:

$$[L]\{\underline{U}\} = \{\underline{U}_\Gamma - \underline{U}^*\}d(|\vec{X}_k - \vec{x}_{i,j}|) + [L]\{\underline{U}^*\}d(|\vec{X}_k - \vec{x}_{i,j}|) \tag{13}$$

and this formally represents the enforcement of the boundary condition at location $\vec{X}_k$ on the immersed surface.

The major advantage of the discrete forcing concept is the absence of user-specified parameters in the forcing and the elimination of associated stability constraints. However, the forcing still extends into the fluid region due to the use of a distribution function and the details of the implementation depend strongly on the numerical algorithm used to discretize the governing equations. This technique has been applied to several problems including turbulent flow inside an internal combustion engine (Verzicco et al. 2000), and flow past 2D (Balaras 2004) and 3D bluff bodies (Verzicco et al. 2002) and in a cylindrical stirred tank (Verzicco 2003).

4.2. Direct BC Imposition

Although the application of IB methods to low and moderate Reynolds number flows has been successful, their extension to higher Reynolds numbers is

challenging due to the need to accurately resolve the boundary layers on (immersed) surfaces not aligned with the grid lines. In such cases 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 is less desirable. For this reason, other approaches can be considered where the immersed boundary is retained as a "sharp" interface with no spreading and where greater emphasis is put on the local accuracy near the IB. This can usually be accomplished by modifying the computational stencil near the immersed boundary to directly impose the boundary condition on the IB. Here we describe two methods that fit into this category.

4.2.1. GHOST-CELL FINITE-DIFFERENCE APPROACH   The boundary condition on the IB is enforced here through the use of "ghost cells." Ghost cells are defined as cells in the solid that have at least one neighbor in the fluid. For instance, cell "G" in Figure 2 is a ghost cell. For each ghost cell, an interpolation scheme that implicitly incorporates the boundary condition on the IB is then devised. A number of options are available for constructing the interpolation scheme (Majumdar et al. 2001). One simple option is bilinear (trilinear in 3D) interpolation where a generic flow variable ϕ can be expressed with reference to Figure 2 as

$$\phi = C_1 x_1 x_2 + C_2 x_1 + C_3 x_2 + C_4. \tag{14}$$

The four coefficients in the above equation can be evaluated in terms of the values of ϕ at fluid nodes F_1, F_2, and F_3, and at the boundary point B_2, which is the normal intercept from the ghost node to the IB. Boundary point B_1, which is midway between points P_1 and P_2, can also be used instead of B_2. Note that P_1 and P_2 are the intercepts with the y- and x-lines passing through the ghost point, respectively. A less accurate, linear interpolation scheme (i.e., $C_1 \equiv 0$ in Equation 14) would not employ the fluid node F_3 and therefore would retain a discrete form, which is well suited for line-solution techniques (Ferziger & Peric 1996).

Applying a linear reconstruction is acceptable for laminar flows or for high Reynolds number flows when the first grid point is located in the viscous sublayer (Iaccarino & Verzicco 2003). At high Reynolds numbers when the resolution is marginal, linear reconstruction could lead to erroneous predictions. For such cases higher-order interpolation can be used. For instance, one could employ an interpolant that is linear in the tangential direction and quadratic in the normal direction (Majumdar et al. 2001), such as

$$\phi = C_1 n^2 + C_2 nt + C_3 n + C_4 t + C_5, \tag{15}$$

where n and t are local coordinates normal and tangent, respectively, to the IB. The five coefficients can be determined by using the four fluid points values F_1 to F_4 and the boundary condition at point B_2 where the selection of point F_4 depends on the surface normal. Alternatively, the points F_1 to F_3 and the two boundary points P_1 and P_2 could be used without losing generality. Other interpolation schemes can also be employed (Ghias et al. 2004).

Irrespective of the particular interpolation scheme used, the value of the variable at the ghost-cell node, ϕ_G, can be expressed as

$$\sum \omega_i \phi_i = \phi_G, \qquad (16)$$

where the summation extends over all the points in the stencil, including one or more boundary points, and ω_i are known geometry dependent coefficients. The above equation represents the modified discrete Equation 6 for the ghost cell and this can now be solved simultaneously with the discretized Navier-Stokes equations for fluid nodes. This method has been used for simulating a wide variety of flows including compressible flow past a circular cylinder and an airfoil (Ghias et al. 2004) at Reynolds numbers up to $O(10^5)$, aquatic propulsion (Mittal et al. 2004), flow through a rib-roughened serpentine passage (Iaccarino et al. 2003) and turbulent flow past a road vehicle (Kalitzin et al. 2003).

4.2.2. CUT-CELL FINITE-VOLUME APPROACH   None of the IB methods discussed so far are designed to satisfy, the underlying conservation laws for the cells in the vicinity of the IB. Strict global and local conservation of mass and momentum can only be guaranteed by resorting to a finite-volume approach and this is the primary motivation for the cut-cell methodology. This methodology was first introduced in the context of Cartesian grid methods for inviscid flow computations (Clarke et al. 1986) and was later applied to simulation of viscous flows (Udaykumar et al. 1996, 2001, 2002; Ye et al. 1999). Figure 3a shows a schematic of a Cartesian grid with an IB that demarcates a solid from a fluid. In this method, cells in the Cartesian grid that are cut by the IB are identified, and the intersection of the boundary with the sides of these cut cells is determined. Next, cells cut by the IB, whose cell center lies in the fluid, are reshaped by discarding the portion of these cells that lies in the solid. Pieces of cut cells whose centers lie in the solid are absorbed by neighboring cells. This results in the formation of control volumes, which are trapezoidal in shape (Ye et al. 1999), as shown in Figure 3a.
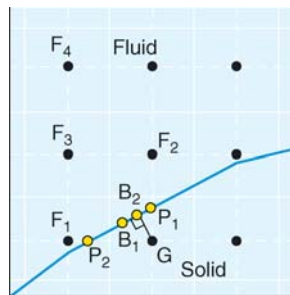
Figure 3   Representation of the points in the vicinity of an immersed boundary used in the ghost-cell approach. F_i are fluid points, G is the ghost point, and B_i and P_i are locations where the boundary condition can be enforced.

Finite-volume discretization of the Navier-Stokes equations requires the estimation of mass, convective and diffusive flux integrals, and pressure gradients on the faces of each cell and the issue is to evaluate these on the cell faces of the trapezoidal cells. The approach proposed by Ye et al. (1999) is to express a given flow variable ϕ in terms of a two-dimensional polynomial interpolating function in an appropriate region and evaluate the fluxes f based on this interpolating function. For instance, to approximate the flux on the southwest face, f_{sw}, ϕ (in the shaded trapezoidal region shown in Figure 3b) is expressed in terms of a function that is linear in x_1 and quadratic in x_2:

$$\phi = C_1 x_1 x_2^2 + C_2 x_2^2 + C_3 x_1 x_2 + C_4 x_1 + C_5 x_2 + C_6, \tag{17}$$

where C_1 to C_6 are six unknown coefficients that can be expressed in terms of values of ϕ at the six stencil points shown in Figure 3b and an expression similar to Equation 16 is developed for f_{sw}. Equation 17 represents the most compact function that allows at least a second-order accurate evaluation of ϕ or its derivative at the sw location. A similar approach can be employed to evaluate the flux on the east-face f_e as well as the interface flux f_i. This approach results in a discretization scheme that is globally as well as locally second-order accurate and also satisfies conservation of mass and momentum exactly irrespective of the grid resolution.

This method has been used to simulate various flows with stationary and moving boundaries including flow-induced vibrations (Mittal et al. 2003), flapping foils (Mittal et al. 2002b), objects in free fall through a fluid (Mittal et al. 2004), and diaphragm-driven synthetic jets (Utturkar & Mittal 2002). Extending this approach to three dimensions, however, is nontrivial because the cut-cell procedure leads to complex polyhedral cells, and discretization of the full Navier-Stokes equations on such cells is complicated. Extension to three dimensions would likely be based on "cell-trimming" procedures (Berger & Aftosmis 1998) that generate body-fitted grids from a Cartesian grid.

4.3. General Considerations

The methods presented in this section and other related methods not discussed here (Fedkiw & Liu 2000, Leveque & Li 1994) introduce the boundary condition directly into the discrete equations. The forcing procedure is therefore 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. Furthermore, the discrete forcing approach does not introduce any extra stability constraints in the representation of solid bodies. Finally, this approach decouples the equations for the fluid nodes from those for the nodes in the solid, thereby obviating the solutions of the governing equations for the solid grid nodes. This is highly desirable for high Reynolds number flows. As we discuss in the following section, one disadvantage of the discrete forcing approach is that inclusion of boundary motion can be more difficult. Finally, methods in

this category also usually require imposition of a pressure boundary condition on the immersed boundary (see, for instance, Udaykumar et al. 2001), whereas no pressure boundary condition is needed for methods that employ continuous forcing.

## 5. FLOWS WITH MOVING BOUNDARIES

In the context of flows with moving boundaries, most of the methods described here can be viewed as Eulerian-Lagrangian, wherein the Eulerian form of the governing equations (as in Equation 7) is solved on a stationary grid and moving boundaries are tracked in a Lagrangian fashion (as in Equation 7). The use of a stationary, non-deforming grid and the associated retention of the Eulerian form of the governing equations greatly simplify the incorporation of moving boundaries into IB methods. In contrast, Lagrangian methods have to deal with moving/deforming grids (Tezduyar 2001) as well as discretized equations that incorporate time derivatives of cell volumes and other grid-related quantities.

Further distinctions among these methods can be made based on the technique used to track the IB as well as the approach used to represent its effect on the underlying Eulerian flow-field variables. For instance, in Peskin's IB method (1981), the boundary is tracked as a distinct and sharp Lagrangian entity while it is treated as diffuse in accounting for its effect on the fluid phase. In contrast, for methods such as cut cell and ghost cell, the IB is tracked as a sharp, Lagrangian entity and also treated as such when incorporating its effect on the fluid phase. Using this taxonomy, IB methods can also be contrasted with so-called Eulerian methods such as Volume-of-Fluid (Anderson et al. 1998), which retain the diffuse nature of the interface both in tracking as well as representing its effect on the flow field.

For sharp-interface methods such as cut-cell and ghost-cell, one additional issue has to be dealt with in order to enable boundary motion. As shown in Figure 4, as the IB moves across the fixed Cartesian grid, "freshly-cleared" cells, i.e., cells in
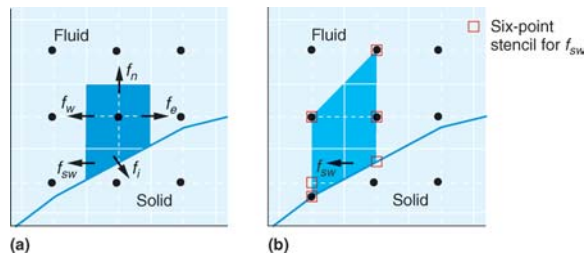
Figure 4 Schematics showing the key features of the cut-cell methodology. (*a*) Trapezoidal finite volume formed near the immersed boundary for which f denotes the face-flux of a generic variable. (*b*) Region of interpolation and stencil employed for approximating the flux f_{sw} on the southwest face of the trapezoidal finite volume.
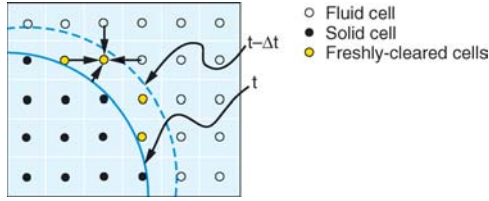
Figure 5   Schematic showing the creation of "freshly-cleared" cells on a fixed Carte-
sian grid due to boundary motion from time step $(t - \Delta t)$ to t. Schematic indicates how
the flow variables at one such cell could be obtained by interpolating from neighboring
nodes and from the immersed boundary.

the fluid which were inside the solid at the previous time step, are encountered. In
effect, for cases involving boundary motion, the spatial discontinuity associated
with the sharp IB leads to a temporal discontinuity for cells near the boundary.
Straightforward temporal discretization of the momentum equation for these cells
is not possible since flow variables in these cells do not have a valid time-history.
One approach to handle this issue is to merge these cells with adjacent fluid cells
(Udaykumar et al. 1999) for the first time step after a cell emerges from the body.
This approach is essentially similar to what is done in body-conformal Lagrangian
methods and does not affect the spatial accuracy of the method (Udaykumar et al.
1999). Another approach is to determine the flow velocity in this cell for one time
step by interpolating from neighboring cells (Udaykumar et al. 2001). The issue
of freshly-cleared cells is not encountered in IB methods that employ continuous
forcing since the spreading of the effect of the IB over a few grid cells on both
sides of the boundary, provides a smooth transition between the fluid and solid
phases, and removes the temporal discontinuity for cells emerging into the fluid.
Thus, inclusion of boundary motion is quite straightforward in these methods.

6. APPLICATIONS

Applications included here cover a broad spectrum of flows and methodologies
and are intended to highlight the extensive capabilities of these methods.

6.1. Flow Past Flapping Filaments

Simulations of flow past two flexible, flapping filaments (threads) in a flowing soap
film were performed using Peskin's original IB method (Zhu & Peskin 2003). The
force density $\vec{F}$ contributed by the fibers is computed from the elastic potential
energy associated with the stretching and bending of the fibers. The Reynolds
number based on the thread length (L_t) and terminal velocity of the soap film is 200
and the simulations employ a 512×256 Cartesian grid. The initial configuration
chosen for the two filaments is a pair of in-phase, parallel sine waves separated by
a distance equal to $0.3\,L_t$ and have amplitudes equal to $0.25\,L_t$.
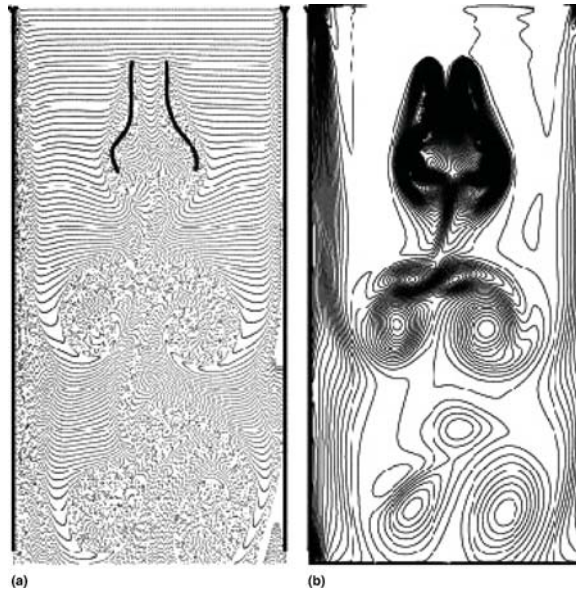
Figure 6   Clapping motions of two filaments in a flowing soap film simulated by
Zhu (2003) using an immersed boundary method. (*a*) Instantaneous snapshot of fluid
markers. (*b*) Spanwise vorticity contours at this time instant.

Figure 6 shows computed results from this simulation. The flow of the soap
film is from top to bottom and this film is driven by gravity and falls against
air resistance. Figure 6*a* shows the instantaneous motion of fluid tracers that are
introduced intermittently along the top boundary. In Figure 6*b*, contours of vorticity
are plotted, clearly showing the complex vortex shedding from this filament pair.
Even though the two filaments start in phase with each other, they spontaneously
develop a 180° difference in oscillation after about one cycle, and maintain this
phase difference thereafter. Thus, after an initial transient, the filaments settle into
a stable flapping state, which consists of a clapping motion that is symmetrical
with respect to the flow midline. This clapping is self-sustained and periodic in
time and these results are in general agreement with experiments conducted at
higher Reynolds numbers (Zhang et al. 2000).

6.2. Flow Past a Pick-Up Truck

As discussed above, applying the IB method to high Reynolds numbers external
flows is challenging because resolution of thin boundary layers present on the IB
requires extremely large computational grids. In Kalitzin et al. (2003), a 3D, ghost-
cell-based IB Reynolds-Averaged Navier-Stokes (IBRANS) solver was employed
for simulating the flow past a road vehicle, in this case a pickup truck, at a realistic
Reynolds number of 3×10^5 based on the vehicle length.
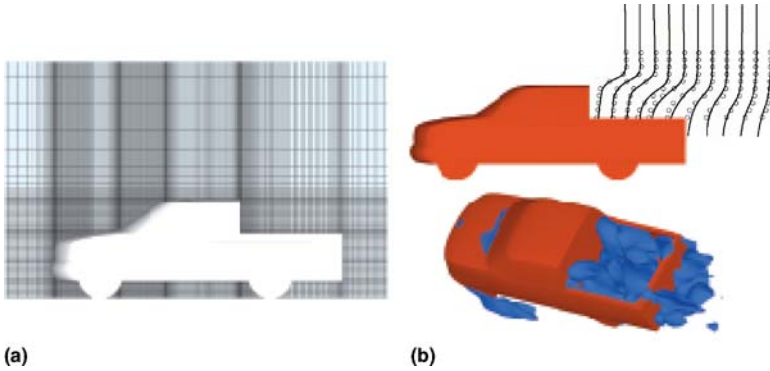
Figure 7   Simulation of the flow around a pickup truck using a ghost-cell method (Kalitzin et al. 2003). (*a*) Side view of the geometry and the nonuniform Cartesian grid in the symmetry plane. (*b*) The top plot shows comparison of computed (*line*) and measured (*symbols*) velocity profiles behind the cabin and the bottom figure shows vortical structures in the wake of the truck.

Figure 7*a* shows the vehicle geometry with a cross-section of a typical nonuniform Cartesian grid. The total number of grid cells for these simulations ranged from 2 to 30 million, with an average of 30% of the cells located inside the body of the truck. The simulation on the largest grid required about six CPU hours on 16 processors of an SGI Origin-2000. The computational grid and the additional information required to define the forcing terms was generated in less than 20 minutes starting from a CAD representation of the geometry. The flow is extremely complicated, exhibiting a large recirculation region downstream of the cabin and inside the truck bed, as Figure 7*b* shows. A comparison between computed velocity profiles and PIV data is shown in Figure 7*b* and this, along with other extensive comparisons with experiments (Kalitzin et al. 2003), indicates that the IBRANS technique successfully predicts the key features of the flow.

6.3. Flow Past a Sphere

Yun et al. (2003) conducted large-eddy simulations (LESs) of flow past a sphere at the Reynolds numbers of 3700 and 10^4, based on the freestream velocity and sphere diameter (d). The IB method was implemented in a cylindrical coordinate system, and momentum forcing and mass sources/sinks were introduced inside the IB to satisfy the no-slip condition on the sphere surface and continuity for the cell containing the IB, respectively (Kim et al. 2001). A hybrid spatial discretization scheme was used wherein a third-order compact upwind scheme was employed before separation to avoid dispersion errors and the second-order central difference scheme was applied to the wake region together with a dynamic subgrid-scale model. The computational domain was $-15 \leq x/d \leq 15$, $0 \leq r/d \leq 15$ and $0 \leq \theta < 2\pi$, where x, r, and θ are the streamwise, radial, and azimuthal

(a)   (b)
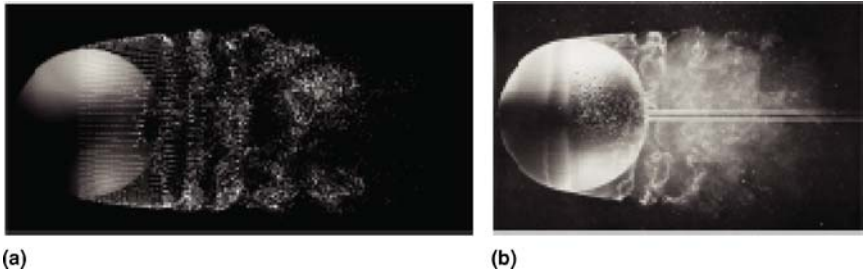
Figure 8 Flow structures in the near wake behind a sphere: (*a*) $Re = 10^4$ (IB simulation); (*b*) $Re = 1.5 \times 10^4$ (experiment).

directions, respectively. The numbers of grid points used were $577(x) \times 141(r) \times 40(\theta)$.

Figure 8 shows computed vortical structures visualized using particle tracers for $Re = 10^4$, together with an experimental flow visualization at $Re = 1.5 \times 10^4$ (Werlé 1980). Vortex rings are observed forming immediately behind the sphere and the computed wake structure for $Re = 10^4$ (Figure 8*a*) is very similar to that observed in the experiment (Figure 8*b*). Simulation results are summarized in Table 1, where the computed time-averaged drag coefficient ($\bar{C}_d$), base pressure coefficient ($\bar{C}_{p_b}$), and Strouhal number (St) are presented together with previous experimental and numerical data. In general, the results from the IB simulations are in good agreement with these other studies, thereby further validating the fidelity of the IB simulations.

6.4. Flutter and Tumble of Bodies in Free Fall

Cut-cell method-based simulations have also been employed to examine the dynamics of plates falling freely in a fluid under the influence of gravity (Mittal et al. 2003, 2004). The motion of such plates is governed by three parameters: the

TABLE 1 Flow parameters for turbulent flows over a sphere

|	Re	$\overline{C}_d$	$\overline{C}_{p_b}$	St
LES with an IB method	3700	0.355	−0.194	0.208
(Yun et al. 2003)	10^4	0.393	−0.274	0.167
Experiment	3700			0.22
(Kim & Durbin 1988)	4200		−0.23	
	10^4			0.16
Experiment	3700			0.21
(Sakamoto & Haniu 1990)	10^4			0.18
Detached eddy simulation	10^4	0.393	−0.275	0.195
(Constantinescu & Squires 2004)				
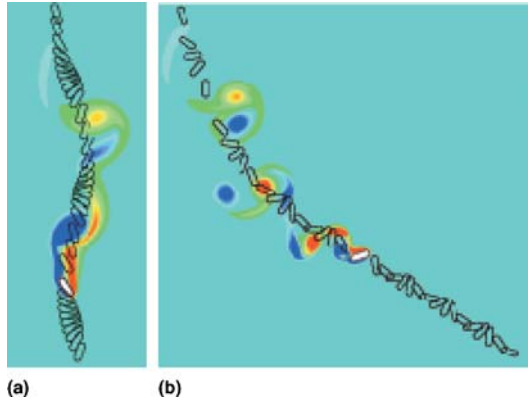
Figure 9 Simulation of plate in free fall carried out by using a cut-cell method (Mittal et al. 2004). These plots show a sequence of plate positions over time along with spanwise vorticity contours corresponding to one time instance. Simulations predict (*a*) flutter as well as (*b*) tumble-type motion, depending on the particular choice of parameters.

plate aspect ratio, the terminal velocity–based Reynolds numbers, and the nondimensional moment of inertia. In particular, depending on the particular values of these parameters, the plate can either undergo a flutter (side-to-side) or a "tumble" (end-over-end rotation) motion as it floats down (Lugt 1983), and the effect of plate-aspect ratio and Reynolds numbers on this flutter-to-tumble transition was examined (Mittal et al. 2004). Figure 9*a,b* shows the tracks of a plate exhibiting flutter and tumble motion respectively. These simulations have been carried out on a 982×982 Cartesian grid. Also shown in these plots are contours of spanwise vorticity at one time instant, which clearly show the presence of Karman-type vortex shedding, which, the simulations indicate, plays a key role in driving the flutter and tumble motion. The power of the Cartesian grid based IB method is aptly demonstrated by Figure 10, which shows results from a simulation that contains five blocks falling freely in a fluid (Mittal et al. 2002a). This simulation employed a uniform 900×1200 Cartesian grid and took approximately 50 CPU hours on a single-processor 733 MHz Alpha workstation. Such a flow is certainly amenable to conventional, body-conformal unstructured grid methods (see, for example, Tezduyar 2001), but the presence of multiple moving bodies would require a high level of sophistication in the grid generation and numerical solution algorithm. In contrast, this increase in complexity poses no particular difficulty for the IB method.

7. CONCLUSIONS

The last decade has seen a tremendous rise in the popularity of IB methods. The primary factor driving this is the relative ease with which this methodology allows researchers to develop computational models of flows with extremely complex

(a)   (b)   (c)
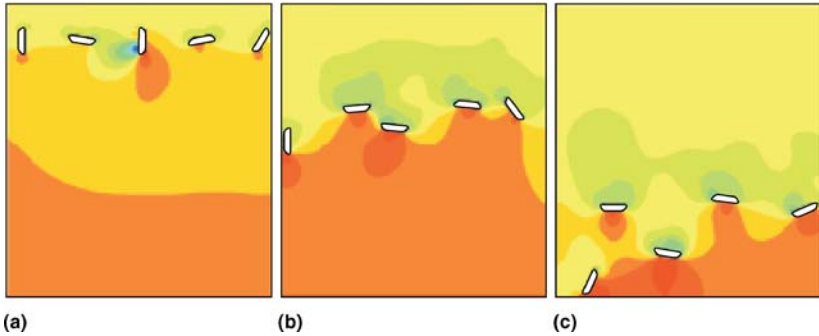
Figure 10   Sequence of plots showing results from simulation of five trapezoidal blocks in free fall carried out on a stationary Cartesian grid using a cut-cell method (Mittal et al. 2003). Plots show three time instances in the simulation and instantaneous pressure contours.

geometries and/or moving boundaries. Our own experience is that a rudimentary IB capability can be incorporated into a Navier-Stokes solver in a matter of a few weeks! In addition, by eliminating the need for complex grids, these methods also significantly reduce the time and effort required to set up and initiate a simulation. A number of variants of these methods currently exist and not all are covered in this
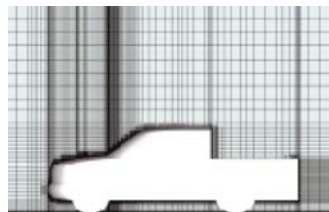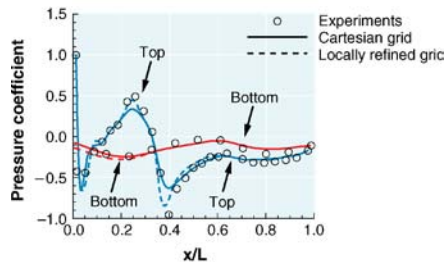
Figure 11   Simulation of the flow around a pickup truck using a ghost-cell method on a locally refined grid by Iaccarino et al. (2004). (*a*) Grid-employed (*b*) surface pressure distribution on the symmetry plane.

review. Instead, we have attempted to systematically categorize selected methods and provide a framework that a CFD researcher can use to critically assess any IB method.

Current research in these methods is focused toward improving their accuracy and efficiency. Using adaptive grid refinement with IB methods is promising (Roma et al. 1999), especially for high Reynolds number flows (Iaccarino et al. 2004). An example of the application of this technique is shown in Figure 11 for the flow around the truck presented in section 6.2. This simulation employs a smaller computational grid with about 3 million cells shown in Figure 11*a*, and the results are compared to those obtained on a 30 million point grid without local refinement in Figure 11*b*. Predicting the pressure distribution on the vehicle symmetry plane is considerably improved in the regions where pressure peaks are observed experimentally through the use of adaptive grids. However, using local refinement increases the complexity of the algorithm and also begins to blur the line between these and unstructured grid methods.

The few applications highlighted here do not even begin to scratch the surface of the many different flows that have already been simulated using these methods. The largest number of applications of these methods are currently found in biological and multiphase flows. In addition to these, IB methods will see increased application in complex turbulent flows, fluid-structure interaction, and multimaterial (Tran & Udaykumar 2004) and multiphysics simulations.

ACKNOWLEDGMENTS

The *Annual Review of Fluid Mechanics* is online at http://fluid.annualreviews.org

LITERATURE CITED

Anderson DM, McFadden GB, Wheeler AA. 1998. Diffuse-interface methods in fluid mechanics. *Annu. Rev. Fluid Mech.* 30:139–65

Angot P, Bruneau CH, Frabrie P. 1999. A penalization method to take into account obstacles in viscous flows. *Numer. Math.* 81:497–520

Balaras E. 2004. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Comput. Fluids* 33:375–404

Baum JD, Luo H, Lonher R. 1998. The numerical simulation of strongly unsteady flows with hundreds of moving bodies. *AIAA Pap. 1998-0788*

Berger MJ, Aftosmis MJ. 1998. Aspects (and aspect ratios) of Cartesian mesh methods. *Proc. 16th Int. Conf. Numer. Methods Fluid Dyn.*

Beyer RP. 1992. A computational model of the cochlea using the immersed boundary method. *J. Comp. Phys.* 98:145–62

Beyer RP, Leveque RJ. 1992. Analysis of a one-dimensional model for the immersed boundary method. *SIAM J. Numer. Anal.* 29:332–64

Brinkmann HC. 1947. A calculation of the viscous force exerted by a flowing fluid on a swarm of particles. *Appl. Sci. Res.* 1:27–34

Chorin AJ. 1968. Numerical solution of the Navier-Stokes equations. *Math. Comput.* 22:745

Clarke D, Salas M, Hassan H. 1986. Euler calculations for multi-element airfoils using Cartesian grids. *AIAA J.* 24:1128–35

Constantinescu G, Squires K. 2004. Numerical investigations of flow over a sphere in the subcritical and supercritical regimes. *Phys. Fluid* 16:1449–66

Fauci LJ, McDonald A. 1994. Sperm motility in the presence of boundaries. *Bull. Math. Biol.* 57:679–99

Fedkiw R, Liu XD. 2002. The ghost fluid method for viscous flows. *Innovative Methods for Numerical Solutions of Partial Differential Equations*, ed. M Hafez, JJ Chattot, pp. 111–143. Singapore: World Sci.

Ferziger JH, Peric M. 1996. *Computational Methods in Fluid Dynamics*. New York: Springer-Verlag

Ghias R, Mittal R, Lund TS. 2004. A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries. *AIAA Pap. 2004-0080*

Glowinski R, Pan TW, Periaux J. 1994. A fictitious domain method for external incompressible viscous flows modeled by Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.* 112:133–48

Goldstein D, Handler R, Sirovich L. 1993. Modeling a no-slip flow boundary with an external force field. *J. Comput. Phys.* 105:354–66

Iaccarino G, Kalitzin G, Elkins CJ. 2003. Numerical and experimental investigation of the turbulent flow in ribbed serpentine passage. *Annu. Res. Briefs, Cent. Turbul. Res.*, pp. 379–88

Iaccarino G, Kalitzin G, Moin P, Khalighi B. 2004. Local grid refinement for an immersed boundary RANS solver. *AIAA Pap. 2004-0586*

Iaccarino G, Verzicco R. 2003. Immersed boundary technique for turbulent flow simulations. *Appl. Mech. Rev.* 56:331–47

Kalitzin G, Iaccarino G, Khalighi B. 2003. Towards an immersed boundary solver for RANS simulations. *AIAA Pap. 2003-0770*

Khadra K, Angot P, Parneix S, Caltagirone JP. 2000. Fictitious domain approach for numerical modeling of Navier-Stokes equations. *Int. J. Numer. Methods Fluids* 34:651–84

Kim HJ, Durbin PA. 1988. Observations of the frequencies in a sphere wake and of drag increase of acoustic excitation. *Phys. Fluid* 31:3260–65

Kim J, Kim D, Choi H. 2001. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Phys.* 171:132–50

Lai MC, Peskin CS. 2000. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *J. Comput. Phys.* 160:705–19

Leveque RJ, Li Z. 1994. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* 31:1001–25

Lugt HJ. 1983. Autorotation *Annu. Rev. Fluid Mech.* 15:123–47

Majumdar S, Iaccarino G, Durbin PA. 2001. RANS solver with adaptive structured boundary non-conforming grids. *Annu. Res. Briefs 2001, Cent. Turbul. Res.* 353–64

Mittal R, Bonilla C, Udaykumar HS. 2003. Cartesian grid methods for simulating flows with moving boundaries. In *Computational Methods and Experimental Measurements-XI*, ed. CA Brebbia, GM Carlomagno, P Anagnostopoulous

Mittal R, Seshadri V, Sarma S, Udaykumar HS. 2002a. Computational modeling of fluidic micro-handling processes. In *Tech. Proc. 5th Int. Conf. Model. Simul. Microsyst., San Juan, Puerto Rico*, pp. 388–91

Mittal R, Seshadri V, Udaykumar HS. 2004. Flutter, tumble and vortex induced autorotation. *Theor. Comput. Fluid Dyn.* 17(3):165–70

Mittal R, Utturkar Y, Udaykumar HS. 2002b.

Computational modeling and analysis of biomimetic flight mechanisms. *AIAA Pap. 2002-0865*

Mohd-Yosuf J. 1997. Combined immersed boundary/B-spline methods for simulation of flow in complex geometries. *Annu. Res. Briefs, Cent. Turbul. Res.,* pp. 317–28

Patankar NA. 2001. A formulation for fast computation of rigid particulate flows. *Annu. Res. Briefs, Cent. Turbul. Res.*, pp. 197–208

Patankar SV. 1980. *Numerical Heat Transfer and Fluid Flow.* New York: McGraw Hill

Peskin CS. 1972. *Flow patterns around heart valves: a digital computer method for solving the equations of motion.* PhD thesis. Physiol., Albert Einstein Coll. Med., Univ. Microfilms. 378:72–30

Peskin CS. 1981. The fluid dynamics of heart valves: experimental, theoretical and computational methods. *Annu. Rev. Fluid Mech.* 14:235–59

Quarteroni A, Valli A. 1999. *Domain Decomposition Methods for Partial Differential Equations.* Oxford: Oxford Univ. Press

Ramamurti R, Sandberg WC. 2001. Simulation of flow about flapping airfoils using a finite element incompressible flow solver. *AIAA J.* 39:253–352

Roma AM, Peskin CS, Berger MJ. 1999. An adaptive version of the immersed boundary method. *J. Comput. Phys.* 153:509–34

Saiki EM, Biringen S. 1996. Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *J. Comput. Phys.* 123:450–65

Sakamoto H, Haniu H. 1990. A study on vortex shedding from spheres in a uniform flow. *Trans ASME J. Fluid Eng.* 112:386–92

Scardovelli R, Zaleski S. 1999. Direct numerical simulation of free-surface and interfacial flows. *Annu. Rev. Fluid. Mech.* 31:567–603

Schlichting H, Gersten K. 1996. *Boundary Layer Theory.* Berlin: Springer-Verlag

Stockie JM, Wetton BR. 1998. Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. *J. Comput. Phys.* 154:41–64

Tezduyar TE. 2001. Finite element methods for flow problems with moving boundaries and interfaces. *Arch. Comput. Methods Eng.* 8:83–130

Tran L, Udaykumar HS. 2004. A particle-levelset based sharp interface Cartesian grid method for impact, penetration and void collapse. *J. Comp. Phys.* 193:469–510

Udaykumar HS, Mittal R, Rampunggoon P. 2002. Interface tracking finite volume method for complex solid-fluid interactions on fixed meshes. *Commun. Numer. Methods Eng.* 18:89–97

Udaykumar HS, Mittal R, Rampunggoon P, Khanna A. 2001. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *J. Comput. Phys.* 174:345–80

Udaykumar HS, Mittal R, Shyy W. 1999. Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids. *J. Comput. Phys.* 153:534–74

Udaykumar HS, Shyy W, Rao MM. 1996. Elafint: A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries. *Int. J. Numer. Methods Fluids* 22:691–705

Unverdi S, Tryggvason G. 1992. A front-tracking method for viscous, incompressible, multifluid flows. *J. Comput. Phys.* 100:25–42

Utturkar Y, Mittal R. 2002. Sensitivity of synthetic jets to the design of the jet cavity. *AIAA Pap. 2002-0124*

Verzicco R, Fatica M, Iaccarino G, Moin P, Khalighi B. 2002. Large eddy simulation of a road-vehicle with drag reduction devices. *AIAA J.* 40:2447–55

Verzicco R, Mohd-Yusof J, Orlandi P, Haworth D. 2000. LES in complex geometries using boundary body forces. *AIAA J.* 38:427–33

Werlé H. 1980. ONERA photograph. In *An Album of Fluid Motion*, assembled by M. Van Dyke, p. 34. Stanford, CA: Parabolic Press

Ye T, Mittal R, Udaykumar HS, Shyy W. 1999. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *J. Comput. Phys.* 156:209–40

Yun G, Choi H, Kim D. 2003. Turbulent flow past a sphere at RE = 3700 and 10000. *Phys. Fluid* 15:S6

Zeeuw D, Powell K. 1991. An adaptively-refined Cartesian mesh solver for the Euler equations. *AIAA Pap. 1991-1542*

Zhang J, Childress S, Libchaber A, Shelley M. 2000. Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind. *Nature* 408:835–45

Zhu L, Peskin C. 2003. Interaction of two filaments in a flowing soap film. *Phys. Fluids* 15:128–36

Ⓡ *Annual Review of Fluid Mechanics*
Volume 37, 2005

Contents