

Origin and evolution of immersed boundary methods in computational fluid dynamics

Rajat Mittal^{*} and Jung Hee Seo[†]

Mechanical Engineering, Johns Hopkins University, 3400 N. Charles St. Baltimore, Maryland 21218, United States



(Received 2 July 2023; accepted 31 August 2023; published 6 October 2023)

This article presents the evolutionary history of immersed boundary methods (IBMs), tracing their origins to the very beginning of computational fluid dynamics in the late 1950s all the way to the present day. The article highlights the advancements in this simulation methodology over the last 50 years and explores the interplay between IBMs and body-conformal grid methods during this time. Drawing upon the authors' combined experience of more than 40 years in this arena, the perspective offered is personal and subjective. By employing a critical and comparative approach through the chronological lens, we hope that this article empowers the reader to understand both the capabilities and limitations of these methods, and to pursue advancements that fill the key gaps and break new ground.

DOI: [10.1103/PhysRevFluids.8.100501](https://doi.org/10.1103/PhysRevFluids.8.100501)

I. ORIGINS

The ground-breaking doctoral dissertation of Charles Peskin, “Flow Patterns around Heart Valves: A Digital Computer Method for Solving the Equations of Motion,” published almost exactly 50 years ago [1] introduced the method that would later become known as the “immersed boundary method” (IBM). However, the roots of this method can be traced back to the late 1950s, when digital computers started to be adopted by various national laboratories in the U.S. and concerted efforts to compute solutions for a wide range of flow and heat transfer problems were initiated. Indeed, within a few years of these computers arriving at these labs, simulations of 2D, time-dependent, incompressible flows also began to appear. The first simulation of the von Kármán vortex street in the wake of a normal flat plate [2] and a square cylinder [3] was certainly a watershed moment in the history of computational fluid dynamics (CFD).

However, all of these early simulations were on Cartesian grids and could not address curved geometries and boundaries. The first glimpse of a curved interface appears in 1965 in the “marker-and-cell” (MAC) method by Harlow and Welsh [4] at the Los Alamos National Laboratory (LANL), which was designed to simulate the evolution of free surfaces on a Cartesian grid. In this method, Lagrangian markers that identified the liquid phase were advected in the flow on a stationary Cartesian grid. These markers were used to identify “surface cells” at the liquid-gas interface, and an ambient pressure boundary condition was specified for these cells to solve for the sloshing of the liquid on a Cartesian grid.

Not too much later, Viecelli, also at LANL, modified the MAC method for curved *solid* boundaries by imposing a normal pressure gradient condition that enforced vanishing velocity of the marker particles normal to the curved solid boundary, i.e., the no-penetration condition, and demonstrated sloshing flow inside a circular container using this method [5]. Viecelli extended his

^{*}mittal@jhu.edu

[†]jhseo@jhu.edu

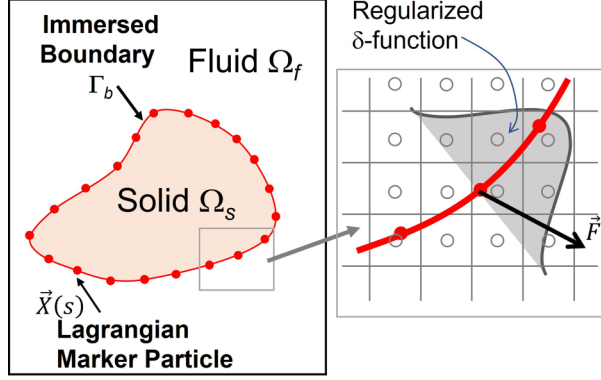


FIG. 1. Schematic of a viscous incompressible flow inside a rectangular 2D domain Ω_f containing an immersed elastic boundary Γ_b .

method to flows with *moving* immersed boundaries [6], although it seems that the method continued to be limited to free-slip boundaries.

II. PESKIN'S METHOD

Peskin's research was centered around simulating two-way coupled fluid-structure interaction in heart valves, and given that Viecelli's method [6] was specifically designed for simulating flow around immersed boundaries with *prescribed* motion, Peskin concluded in his thesis that it was not suitable for his intended application.

The method that Peskin ended up developing for his unique application departed from contemporary approaches in a number of important ways. First and most significantly, instead of the standard approach of incorporating the boundary conditions as a constraint on the discrete governing equations, Peskin imposed the boundary conditions on the immersed boundary through the stresses that they induced on the flow via a body force in the momentum equation, *viz.*,

$$\frac{\partial \rho \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla} \rho \vec{u} = -\vec{\nabla} p + \mu \nabla^2 \vec{u} + \vec{f}_b, \quad (1)$$

where ρ and μ are the fluid density and dynamic viscosity \vec{u} and p are the fluid velocity and pressure, respectively, and \vec{f}_b is the body force that is used to apply the velocity boundary conditions on the immersed boundary (see Fig. 1).

Second, following the Euler-Lagrange formulations of the MAC and Viecelli's methods, he formed the body surface from a set of Lagrangian points [the position of each point defined by a position vector $\vec{X}(s, t)$] which move with the local velocity, $\partial \vec{X} / \partial t = \vec{u}(\vec{X}, t)$. Departing from Viecelli's method, Peskin connected these Lagrangian points with massless elastic fibers which were deformed and stretched by the flow, generating a fiber tension, T , that was determined via a general Hooke's law of the form $T = \epsilon(|\partial \vec{X} / \partial s|; s, t)$ where ϵ is the local strain in the fiber.

Third, the effect of the fiber on the flow was imposed by transmitting the tension force density along the fiber, given by $\vec{F}(s, t) = \partial(T \hat{\tau}) / \partial s$ where $\hat{\tau}$ is the unit vector tangent to the fiber, to the fluid. This transmission of forces from the solid to the fluid is akin to the stress-continuity condition applied at fluid-fluid interfaces. The underlying notion was that a solid wall would be created by increasing the density and stiffness of fibers.

Since the underlying Cartesian grid did not coincide with the Lagrangian marker body points, a method was needed to transfer quantities such as flow velocity and fiber force density between the two sets of points and the fourth innovation of the method was the use of the Dirac- δ to execute this transfer. For instance, the following integral would transfer the fiber force density from the

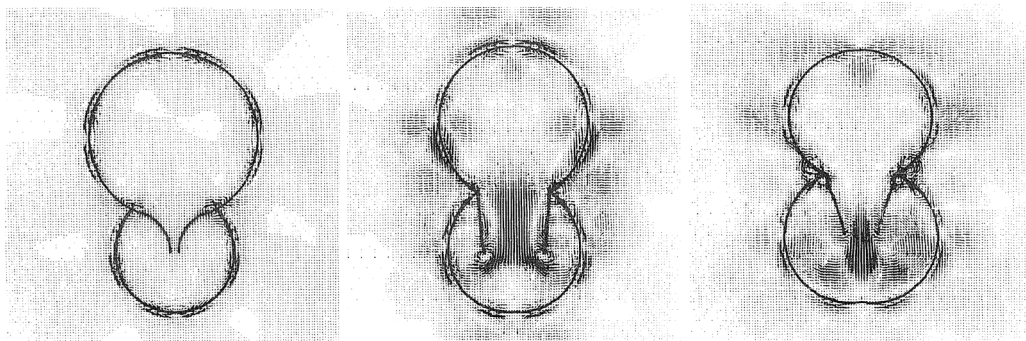


FIG. 2. Figure adapted from Peskin’s thesis [1] showing results from the modeling of flow in the left heart, including flow-induced deformation of the heart valve. This figure is reproduced with permission from Charles Peskin.

Lagrangian marker points to the Eulerian flow field:

$$\vec{f}(\vec{x}, t) = \int_{\Gamma_b} \vec{F}(s, t) \delta(\vec{x} - \vec{X}) ds. \quad (2)$$

However, the Dirac- δ function creates problems for the discrete evaluation of the above integral, and to overcome these issues, Peskin “regularized” the δ function by spreading it across a finite width across the immersed boundary

$$\vec{f}(\vec{x}_{i,j}, t) = \sum_k \vec{F}_k D(|\vec{x}_{i,j} - \vec{X}_k|), \quad (3)$$

where D is the regularized δ function (see Fig. 1).

This fifth step was highly inventive and a key enabler of the method, but it also became the Achilles heel of the method since it led to a “diffuse” interface, i.e., the boundary condition not being applied precisely at the location of the immersed boundary, but instead over a region spanning several grid points across the boundary (see Fig. 1). Furthermore, the degree to which the no-slip, no-penetration boundary condition was satisfied also became dependent on the local grid resolution.

Nevertheless, the method allowed Peskin to produce impressive results for his problem. Figure 2 is plots from his dissertation [1] that show the formation of vortices from the tips of the valve leaflets. It bears emphasizing the degree to which this method represented an advance in the state of the art in computational fluid dynamics at that time. Not only was this one of the first 2D Navier-Stokes simulations of flow with geometrically complicated, no-slip, moving boundaries, it seemed also to be the first simulation that included large-scale fluid-structure interaction.

III. THE NEXT 20 YEARS

A. The rise of body-conformal grid (BCG) methods

Interestingly, despite the versatility of Peskin’s method, its adoption into the wider CFD community was slow. Computational fluid dynamics at that time was driven primarily by disciplines such as aerospace and naval engineering, and meteorology that mostly involved flow over solid bodies and surfaces. It was therefore not clear how Peskin’s method, which incorporated flexible fibers to construct solid bodies, and therefore required specification of stress-strain relationships inside the solid, could be applied to such problems. The early 1970s also saw the emergence of other approaches for dealing with complex geometries in computational fluid dynamics: finite-difference and finite-volume methods on body-conformal structured grids based on coordinate transformations [7–12] and finite-element methods on unstructured grids [13–15]. Flows in many of the above

applications were at high Reynolds numbers and mostly involved stationary solid boundaries, and for such flows, body-fitted grids with the ability to provide high resolution in the boundary layers offered many advantages such as accurate prediction of surface shear and flow separation.

Rapid advances in methods for generating curvilinear grids [16] as well as the formulation of discretization schemes that enabled strict satisfaction of conservation laws on such grids [8,17,18] increased the reach and robustness of these methods. At about the same time, Hirt *et al.* [19,20] introduced the so-called arbitrary Lagrangian-Eulerian (ALE) method, which combined curvilinear grids with grid deformation and movement, and enabled the simulation of flows with moving boundaries. The “overset grid” approach, which employed curvilinear body-conformal grids (BCGs) “patched” or overlaid onto a Cartesian background mesh, appeared in the early 1980s [21,22]. This approach enabled the simulation of flows with moving solid bodies, all accomplished without necessitating grid deformation. Driven by these advances, CFD methods with BCGs became, for at least the next few decades, the mainstay of computational fluid dynamics.

B. Peskin-type methods pick up steam

Peskin’s method continued to be employed mostly for problems in the field of cardiovascular mechanics, and primarily by him and his collaborators and students till the late 1980s [23–28]. Thus, for about 20 years, this powerful method remained hidden in plain sight of computational fluid dynamicists! However, starting in the late 1980s and early 1990s, applications of the method to problems beyond cardiovascular biomechanics began to appear [29–33]. Second, other methods that employed body forces on Cartesian grids to impose wall boundary conditions on immersed boundaries began to appear as well. One such method was the so-called “mask method” [34] for simulation of flow over solid bodies that appeared around this time. This method essentially convolves the intermediate velocity field with a function (the “mask”) that zeros out the velocity inside the immersed body and can be viewed as the application of a momentum forcing that extends over the volume of the solid body. In fact, the mask method could be considered a progenitor of the “penalization”-type methods that appeared in the late 1990s [35]. An important contribution in the early 1990s to immersed boundary methods was the “feedback forcing method” [36,37], which was formulated to simulate incompressible viscous flow past *solid* bodies.

C. Cartesian grid methods appear on the scene

Another class of methods for simulating flows on body-nonconformal Cartesian grids also appeared in the 1980s. One of the first among these was the method of Clark *et al.* [38] who used a Cartesian grid finite-volume approach to solve the 2D Euler equations for transonic flow over a multi-element airfoil (see Fig. 3). This “cut-cell” method was extended to 3D problems [39] as well as to steady, laminar viscous flows [40]. Berger and Leveque [41] presented a finite-volume based Cartesian grid method for simulating inviscid supersonic flow over grid nonconforming boundaries and also introduced adaptive mesh refinement (AMR) to resolve shocks as well as the flow gradients near the immersed boundaries. This overall approach was adopted by other groups [42,43] and became the basis for the NASA code Cart3D [44,45].

The fundamental difference between these methods, which are collectively referred to as “Cartesian grid methods” (CGMs), and Peskin’s method was that these methods employed the conventional approach of modifying the discretization scheme in the vicinity of the boundary to add the boundary condition as a constraint. Perhaps due to this difference as well as the initial emphasis on inviscid external aerodynamics, it seems that these Cartesian grid methods were developed without recognizing any significant connections to Peskin’s method.

The early history of IBM methods summarized above is by no means comprehensive, and the interested reader is referred to the historical perspective of IB methods by Verzicco [46].

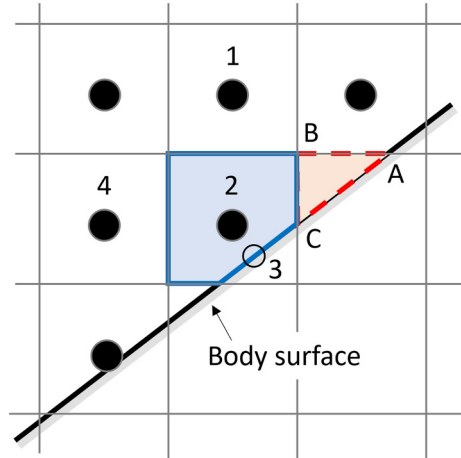


FIG. 3. Schematic of cut-cell method developed by Clark *et al.* [38]. The boundary condition on point 3 was computed by 1D extrapolation using points 1, 2, and 4. Small cells (such as ABC) were incorporated into adjacent cells to avoid instability.

IV. THE 1990S AND BEYOND: LIMITATIONS OF BCG METHODS SPUR ADVANCEMENTS IN IB METHODS

As discussed above, in the roughly 10-year period from 1975 to 1985, significant advances were made in body-conformal grid (BCG) methods for simulating flows. However, as the CFD community moved beyond simple geometries and targeted configurations with increasing geometric complexity such as complete air vehicles, the limitations of the BCG approaches started to become apparent. Many different methods to address these limitations began to appear, including body-conformal multiblock and overset grid methods [21].

Among these developments were methods inspired from Peskin's approach including the "front-tracking" method [47] and the "immersed-interface" method [48], both of which were designed for simulating flows with evolving fluid-fluid interfaces. As mentioned before, the formulation of the "feedback forcing method" of Goldstein *et al.* [36] and Saiki and Biringen [37] was a significant advancement because it enabled simulation of incompressible viscous flow past *solid* bodies. Lai and Peskin [49] also proposed a modification of Peskin's original method for simulating flow past solid bodies. All of these extensions and modifications brought the immersed boundary method out from the niche area of biofluid dynamics into the wider arena of aero- and hydrodynamics. Within this genealogy of methods, Fadlun *et al.* [50] and Verzicco *et al.* [51] presented an approach inspired from the work of Mohd-Yusof [52] that employed a forcing term in the *discretized* Navier-Stokes equations, and in doing so addressed the numerical difficulties associated with the method of [36].

Extensions of the cut-cell method to unsteady viscous flows appeared in the work of Udaykumar [53] and a cut-cell method for unsteady viscous flows with a strictly second-order accurate, "sharp-interface" boundary treatment of the immersed boundary for stationary as well as moving boundaries was presented by Ye *et al.* [54] and Udaykumar *et al.* [55], respectively.

As CFD applications progressed towards complex 3D configurations involving moving and deforming geometries and multiphysics setups, the limitations of BCG methods became apparent. Generating smooth and high-quality BCGs around intricate configurations, which typically consumes the majority of person-hours in complex CFD projects, became even more challenging when dealing with body movement and deformation. ALE and overset grid methods, while offering solutions, introduced additional errors in interpolation and increased computational costs for moving boundary problems. While these challenges could potentially be mitigated with increased computational power, certain problems, such as those featuring extremely complex geometries, large-scale deformation, contact, and fragmentation, remained practically beyond the reach of BCG methods.

For IB methods, grid generation is a trivial step and often requires a few dozen lines of code and a few seconds of computational time. Furthermore, for IB methods, movement of the immersed body directly affects the discretization scheme only on points on or near the immersed boundary, leaving the rest of the grid unaffected. This eases the inclusion of moving boundaries and maintains global discrete operators that are relatively agnostic to the movement of the boundary. Finally, there is virtually no configuration ranging from crystal growth [56], bubble coalescence [47,57], droplet fragmentation [58,59], to contact between solids [60] that IB methods could not tackle.

Recognition of these capabilities among the broader scientific and engineering communities led to an explosion of interest in IBM methods, and the 2000s mark the start of a Cambrian-like period for IBMs with a proliferation of immersed boundary methodologies. Improvements in accuracy, stability, conservation, computational efficiency, versatility, simplicity and parallelizability were just some of the developments that were targeted by the CFD community. Even the finite-element (FEM) CFD community recognized the power of simulating flows on body nonconformal grids and embarked on developing IB methods within the FEM framework [61,62].

We conclude this section by pointing out that the last three decades have seen tremendous advancements in methods for automated generation of unstructured grids over complex geometries [63,64], but the geometry preparation step, particularly the construction of “water-tight” geometries from computer-aided design models, remains a challenge for BCG as well as IBM methods.

V. CLASSIFICATION OF IBMs

In their 2005 review article, Mittal and Iaccarino [65] referred to all finite-difference and finite-volume methods that employed body-nonconformal grids including methods derived from Peskin’s approach as well as the Cartesian grid methods as “immersed boundary methods” and based on the stage in the method where forcing is deployed, classified all IBMs into “continuous forcing” and “discrete forcing” methods.

A. Continuous forcing IBMs

These are IB methods where the forcing term is introduced in the *continuous* form of the governing equations, before the equations are discretized. The original method of Peskin [23] as well as the feedback forcing [36], fictitious domain [36], front tracking [47], and penalization methods [35] all fall in this category.

The primary advantage of these continuous forcing IBMs is their relative simplicity. These IBMs can be implemented in conjunction with a variety of discretization methods such as finite-difference [24], spectral [34,36], and even finite-element methods [61,66]. These methods though also have some shortcomings; the representation of the discontinuous force on a grid with finite spacing leads to a diffuse-interface representation. It can also diminish the formal order-of-accuracy [49] and conservation properties of the method. Furthermore, the prescription of the body force can introduce *ad hoc* parameters that affect the fidelity of the solution, as well as the stiffness and computational efficiency of the solution procedure. Finally, the discretized equations are now solved in the entire domain including the region inside the body, and this represents wasted computational effort especially for high Reynolds number flows [65].

B. Discrete forcing IBMs

In *discrete forcing methods* the forcing term for boundary condition imposition is added to the equations *after* they are discretized. The forcing may be based on “jump conditions” such as in the immersed interface method [32,67,68] or methods such as those in Refs. [50,51], which combine spatial interpolation with a temporally extrapolative approach to estimate the forcing term. This category also includes methods where no explicit forcing term is employed, but instead, the boundary condition are implemented directly as constraints on the discretized equations such as via cut cells [54,69], ghost cells [70–72], or one-sided interpolation schemes [73].

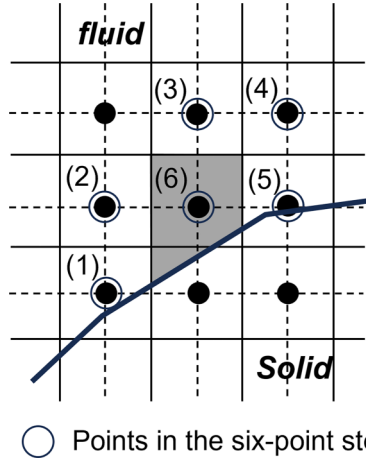


FIG. 4. Figure from Ye *et al.* [54] depicting the cut-cell methodology, including the nodes employed to implement boundary conditions on the immersed boundary to second-order accuracy by using a 2D polynomial interpolation. Reproduced from Ref. [54] with permission from Elsevier.

1. Cut-cell methods and CGMs

All Cartesian grid methods of Refs. [41–43] as well as cut-cell based methods [12,53–55,74] are discrete forcing methods. In cut-cell methods, which are typically based on finite-volume discretizations, the cells intersected by the immersed boundary are “reshaped” to conform to the local boundary (see Fig. 4), and the discretization in these cells is also modified concomitantly. Implementations of cut-cell methods to 3D problems are rare since these method have to contend with cut cells of many different topologies [69,74]. To alleviate this difficulty, Seo and Mittal [75] introduced a “virtual” cut-cell method that was implemented in conjunction with a 3D finite-difference ghost-cell IBM [72] and which provided a higher degree of discrete conservation without the attendant problems of 3D cut cells.

2. Ghost-cell-based methods

Within the category of discrete forcing methods are also methods that employ *ghost cells* (see Fig. 5) to impose the boundary conditions on the immersed boundary. Using a layer of cells immediately outside the computational boundary to impose external boundary conditions is a well-established procedure in many BCG codes [76–78]. This idea has been adopted as a way of imposing boundary conditions on immersed no-slip boundaries in body-nonconformal grid codes [70–72,75,79] as well as for multifluid interfaces [80,81]. These ghost-cell approaches are typically based on finite-difference methods and do not reshape boundary cells. Instead they introduce an auxiliary equation for the ghost cells that enforces the boundary condition on the immersed boundary, and this equation is coupled to the variables in the flow adjacent to the immersed boundary. In the method of Mittal *et al.* [72]. The variable value on the ghost cell (GC) is determined by a second-order interpolation using the boundary condition on the “body-intercept” (BI) point and the flow variable interpolated on the “image-point” (IP).

The ghost-cell method (GCM) can also be extended to higher orders. In the work of Seo and Mittal [82], the ghost-cell value is obtained from an n th order approximating polynomial interpolation by using an arbitrarily large number of stencil points around the immersed boundary (see Fig. 6). This high-order GCM has been employed for the wave propagation [82,83] and compressible flow problems [84].

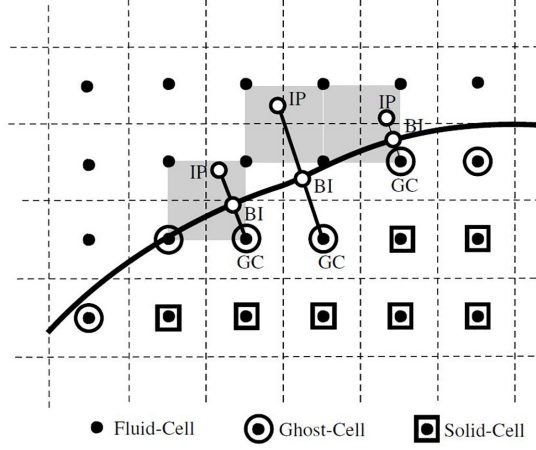


FIG. 5. Schematic of sharp-interface methodology of [72] that implements a second-order boundary condition on the immersed boundary using ghost cells. Reprinted from Ref. [72] with permission from Elsevier.

3. Pros and cons

Since discrete forcing methods do not employ a Dirac- δ function, there is no need for regularization, nor is there the need for any *ad hoc* forcing parameters such as in [36,49]. This avoids the boundary diffusion that results from the regularization of the Dirac- δ function and also obviates the compromise between accuracy and numerical stability that is required in these methods [36]. This also allows these methods to more easily achieve higher than first-order accuracy.

The cut-cell method is the one IBM that allows for strict discrete conservation of mass and momentum [54], but as pointed out earlier, higher-order conservation properties have been embedded into finite-difference based ghost-cell IBMs as well [75]. Cut-cell methods though have their own disadvantages: an increased complexity that is particularly severe for 3D problems [74] as well as the issue of “small cells” [54]. Finally, discrete forcing methods also have to contend with the issue of “fresh cells” for moving boundary problems [75]. Fresh cells are computational cells that emerge from the solid into the fluid due to the movement of the body on the stationary grid and the temporal

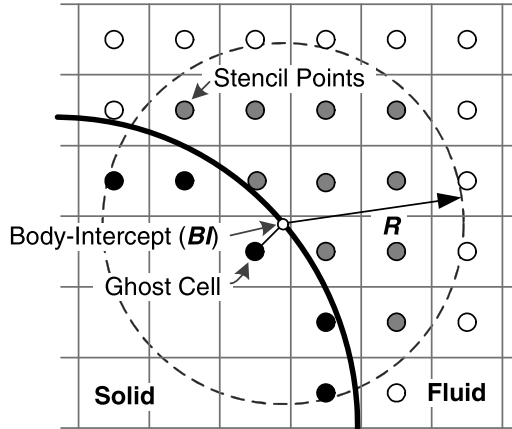


FIG. 6. Schematic of the high-order ghost-cell method of Seo and Mittal [82]. The ghost-cell value is obtained from an n th-order approximating polynomial interpolation by using arbitrary number of stencil points around the immersed boundary in the range of R .

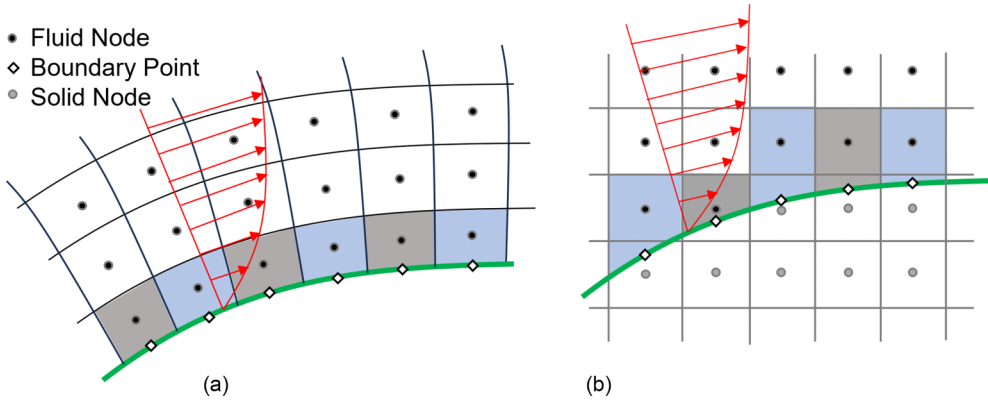


FIG. 7. Schematic of a (a) curvilinear body-fitted grid and (b) Cartesian grid and cut cells around a curved boundary.

discretization of the governing equations for these cells is problematic. Continuous forcing methods experience neither of the above problems due to the diffuse nature of the interface.

VI. SHARP-INTERFACE IB METHODS

The “diffuse” and “sharp” interface terminology in IBM originates from the physics of interfaces between two fluids [85,86], but translating these notions into a precise definition for what constitutes a sharp-interface IBM is challenging. When an IBM method is labeled as a “sharp-interface” method, the implication is that the method applies boundary conditions on the immersed boundary with the same level of accuracy and precision as an equivalent BCG method. This equivalence to BCG methods (see Fig. 7) can be distilled down to the following set of conditions for sharp-interface IBMs:

- (1) The imposition of no-slip, no-penetration boundary conditions is at a set of discrete points located precisely on the immersed boundary and nowhere else within the flow domain.
- (2) The spacing and resolution of these boundary points aligns with the underlying fluid grid.
- (3) The boundary conditions on these boundary points is imposed with an accuracy that is consistent with the underlying numerical scheme employed.
- (4) All grid points and cells in the flow domain impose the “native” governing equation, i.e., the Navier-Stokes equation, and not any other augmented or auxiliary equation.
- (5) The governing equations of flow are *not* solved inside the immersed body.

A. Categorization of IB methods

All continuous forcing methods such as those of Peskin [1] and Goldstein *et al.* [36] as well as the fictitious domain [87] and front-tracking methods [47] are diffuse-interface methods since they do not satisfy any of the conditions enumerated above.

Cut-cell methods [53–55,74] are unequivocally sharp-interface methods since they clearly satisfy the above conditions. In fact, the cut-cell method can be viewed as a body-conformal finite-volume method where the finite-volumes are Cartesian everywhere in the domain except at the boundary, where they assume non-Cartesian shapes [see Fig. 7(b)].

Discrete forcing methods such as the ghost-cell methods of Tseng and Ferziger [70], Ghias *et al.* [71], Mittal *et al.* [72], Seo and Mittal [75], embedded boundary methods of Balaras and others [60,73,88,89], immersed interface based methods [69], and the level-set based Cartesian grid methods of Udaykumar and co-workers [59,90] satisfy the above conditions, and therefore function as sharp-interface methods. In fact, most if not all of these above mentioned methods impose the

velocity boundary conditions on the immersed boundary with *local* second-order accuracy, which is higher than that needed to ensure global second-order accuracy and in doing so, provide high resolution to the boundary layers that develop on the immersed surfaces. The sharp-interface method of Seo and Mittal [75] even incorporates regional discrete conservation via a virtual cut-cell method, which further improves the accuracy of boundary layer flows.

All discrete forcing methods do not however satisfy the five conditions enumerated above. For instance, the mask method [34] spreads the discrete force over a layer of grid cells and is therefore a diffuse-interface method. Another exception is that of the so-called “stair-step” or “staircase” method [91]. In this discrete forcing method, the immersed boundary is represented by a stair-step-shaped surface that conforms to the underlying Cartesian grid. This method does not satisfy the first condition enumerated above and, consequently, is not a sharp-interface method.

The fifth condition seems innocuous but it has important implications for IB methods. This condition is generally associated with the fact that some IB methods (such as all the continuous forcing methods and most penalization methods) do not impose an explicit boundary condition for pressure on the immersed surface and, consequently, have to compute the pressure and the velocity everywhere on the grid including the region inside the body. This can significantly reduce the computational cost of the simulation since the inclusion of a boundary condition for pressure on the immersed surface can significantly amplify the computational effort for solving the pressure Poisson equation.

In the context of the fractional-step method, which is the standard choice for these solvers, the slip and penetration on a surface at the end of the pressure correction step is $O[(\Delta t)^m \partial p / \partial \tau]$ and $O[(\Delta t)^m \partial p / \partial n]$, respectively, where m is the order of the fractional step method and τ and n are the directions tangential and normal to the immersed body. For standard fractional-step schemes, $m = 1$, but second-order implementations [92] also exist. In BCG methods, it is standard to apply $\partial p / \partial n = 0$ on all boundaries since it is consistent with the fractional-step method [93,94], and this results in satisfaction of no-penetration to machine zero at the end of the time step irrespective of the size of the time step Δt as well as the magnitude of the normal pressure gradient. The latter can be large, especially in regions where the flow impacts normal to the body. All the discrete forcing methods described in the third paragraph of this section explicitly impose the von Neumann condition for pressure on the immersed boundary, and this allows them to enforce no penetration to machine zero at the boundary points at the end of the time step. This also decouples the grid points inside the immersed body from those outside, and eliminates the need to solve for the flow variables inside the body.

As pointed out in the previous paragraph, penalization methods, at least in their classic form, do not satisfy the fifth condition. We note, however, that the five conditions for sharpness enumerated earlier in this section may neither be necessary nor sufficient for generating high quality results from IBM simulations. This is because other factors such as the accuracy and dissipative nature of the spatial discretization scheme [95], and the choice of temporal discretization may be entangled with any specific implementation of IBM in a way as to impose additional constraints on its accuracy and fidelity.

B. Predictions from sharp and diffuse interface methods

Udaykumar and co-workers [96,97] conducted a systematic head-to-head comparison of sharp-interface and diffuse-interface methods. They examined the effect of the interface treatment for flow with Reynolds numbers ranging from $O(100)$ to $O(1000)$ and noted that the diffuse-interface method provided better prediction of drag on relatively coarse grids, but the sharp-interface method was more accurate on fine grids. This is not surprising given that on a sufficiently coarse grid, a lower-order method can have a lower absolute truncation errors than a higher-order method. The authors also found that diffuse-interface IBM under-predicted the surface vorticity on all grids and this was a consequence of the regularization of the forcing term.

The generality of the above conclusions, which are necessarily based on specific implementations of interface treatments, remains to be fully established. However, the experience of the current authors with other variants of the sharp-interface method [54,72,75,82] is consistent with the above observations. Accurate prediction of surface vorticity, especially for higher Reynolds number flows where instabilities and transition to turbulence may occur, as well as the general equivalence to BCG methods are, in our view, the primary advantages of sharp-interface methods over their diffuse counterparts.

C. Are sharp interface IB methods \equiv BCG methods?

If sharp-interface IB methods satisfy the five conditions that derive from BCG methods, are there any differences between sharp-interface IB methods and BCG methods that have implications for the numerical accuracy of the flow near the immersed boundary? In our experience, there are some subtle issues that bear pointing out, and we do this with reference to Fig. 7, which shows schematics of a notional BCG as well the second-order cut-cell grid of Ye *et al.* [54] for a curved boundary. We choose the cut-cell method for comparison since as pointed out earlier, this is the IBM that is closest in its implementation to a BCG method.

The first difference to note is that while the shapes and sizes of the cells and finite volumes adjacent to the boundary generally vary smoothly for the BCG, for the cut-cell method, the finite volumes can vary discontinuously for boundary cells in regions where the boundary crosses a layer of Cartesian grid points. This discontinuous variation in local boundary discretization is symptomatic not just of cut-cell methods but of all IB methods. As the boundary layer encounters these cells, it can affect the variation of surface quantities such as surface shear and pressure. Fortunately, this variation in the size of the cell is, limited to between $\pm 50\%$ of a nominal cell, and this boundary cell “unsmoothness” typically occur only for a small fraction of the cells in the entire domain.

Grid unsmoothness starts to manifest its impact primarily when the gradients within the boundary layer lack proper resolution. This issue becomes particularly pronounced in scenarios characterized by complex boundary conditions and is accentuated by under-resolved wall processes, such as reacting flows, multiphase flows, radiation transport, and surfactant models. Addressing these challenges in an immersed boundary method (IBM) might necessitate case-specific adjustments. For instance, while wall models for turbulence can be incorporated within the IBM framework [98,99], this process might not be as straightforward as it is for BCG methods.

One solution for IB methods is to increase the resolution in the boundary region, but this is not always feasible. Another well-known technique to enhance solution quality in cases of marginal resolution is the imposition of discrete conservation. In the context of IBMs, cut cells [54,55,74] or “virtual” cut-cell [75] methods provide discrete conservation and lead to improvements in solution quality.

We observe that BCG methods are not entirely impervious to the influence of grid unsmoothness. Instances of discontinuous variations in the boundary cells can manifest due to inherent grid irregularities. This phenomenon is particularly notable in regions characterized by significant surface curvature, corners, branch cuts, or other intricate topological transitions.

Another difference between BCG methods and sharp-interface IBMs is for problems with moving or deforming boundaries. For such problems, IBM methods encounter the so-called “fresh cell” problem [55,72], where cells that were inside the body at one time step emerge into the fluid at the subsequent time step. For sharp-interface IBMs, these fresh cells do not have a valid time history, and one has to temporarily employ some space-time interpolation [72] or cell merging [55] to advance the equation in time for these cells. For the small number of fresh cells that typically emerge at a given time step, this can reduce the local accuracy of the solution.

BCG methods such as ALE employ deforming meshes coupled with time-evolving control-volume formulations and do not have the fresh-cell problem as long as the movement of the boundary is limited. Once the boundary motion is large enough so as to require a local and even a

global remeshing [100], ALE methods also have to resort to interpolation schemes for the remeshed cells in order to advance the governing equation for these cells. Thus, while the fresh-cell problem in sharp-interface IBM is limited to a few boundary cells, it can occur on a much larger scale for ALE methods.

IBMs also have a subtle advantage over conventional BCG methods that in our view, is not fully appreciated. It is well known that grid skewness (i.e., nonorthogonality in the shape of cells) generates additional dissipation and dispersion errors that have the potential to affect the evolution and advection of vortex structures, waves, and turbulence on the grid [101]. The use of Cartesian grids eliminates this additional source of error in IBM simulations and enhances the ability of IBM methods to more accurately simulate turbulent flows.

VII. GALLERY OF IBM SIMULATIONS

Applications of IBMs span virtually all scientific and engineering domains in fluid mechanics, and it is not possible to summarize this vast literature here. Instead, we present 11 curated cases that demonstrate not only the unique capabilities of IBM methods but also the accuracy and fidelity of these methods. While four examples are from our own work, case studies from other research groups are also included in order to highlight the diversity of IBM approaches and their applications.

A. Simulation of fish swimming in a school

This is a direct numerical simulation of a school of nine fish swimming with prescribed carangiform kinematics [102] against an incoming flow. The Reynolds number based on the tail-beat frequency(f) and the body length(L) is $Re = fL^2/\nu = 5000$. Simulations are performed using our sharp-interface incompressible immersed boundary flow solver (ViCar3D) [72,75] on a fixed nonuniform Cartesian grid [see Fig. 8(a)] with a total of 54 million grid points, and Fig. 8(b) shows the vortex structures generated by the school. Figure 8(c) shows the time-averaged net force in the surge direction generated by each fish normalized by the mean thrust for a single fish, and we note that for the tail-beat phase and interfish distance chosen for this case (which is based on an earlier investigation of two fish by Seo and Mittal [102]), most fish in the school experience an increase in net surge force due to the hydrodynamic interactions with the wakes of upstream fish.

This case is shown primarily to highlight the complexity of the flow configurations and immersed bodies that can be handled by IBMs. Simulation of such a flow that contains multiple moving and deforming bodies and membranes (the caudal fin) in close proximity would represent a nearly insurmountable challenge for any BCG method. Indeed, the vast majority of computational studies of the fluid dynamics of swimming and flying employ IBMs because of this reason although there are some notable exceptions [103–106].

B. Effect of dimple shapes on golf ball drag

It has been known that passive roughness such as dimples are effective in reducing the drag force on bluff bodies. A well-known example of this application is a golf ball, and Smith *et al.* [107] performed direct numerical simulations of flow over a golf ball at a Reynolds number of 1.1×10^5 with a sharp-interface IBM. Beratlis *et al.* [108] investigated the drag reduction on spheres by tessellation using the IBM. Figure 9(a) shows the vortex structure for a ball with tessellated dimples at $Re = 1.1 \times 10^5$. A $1100 \times 1500 \times 3000$ point grid was used in this simulation, and each dimple on the sphere was resolved by $40 \times 50 \times 70$ grid points. The simulation ran on 1500 CPU cores for 2 weeks, and the drag coefficients obtained from the IBM simulations were in good agreement with the experiments [see Fig. 9(b)].

Other examples of IBM based simulations of high Reynolds number bluff body wake flows with validation against experimental data are by Meyer *et al.* ($Re = 3900$ flow past a cylinder) [69] and Xu *et al.* [109] (flow past a sphere at $Re = 3700$ – $10\,000$). Data from compressible flow IBM simulations of bluff-body wakes flows are also highlighted later in this section.

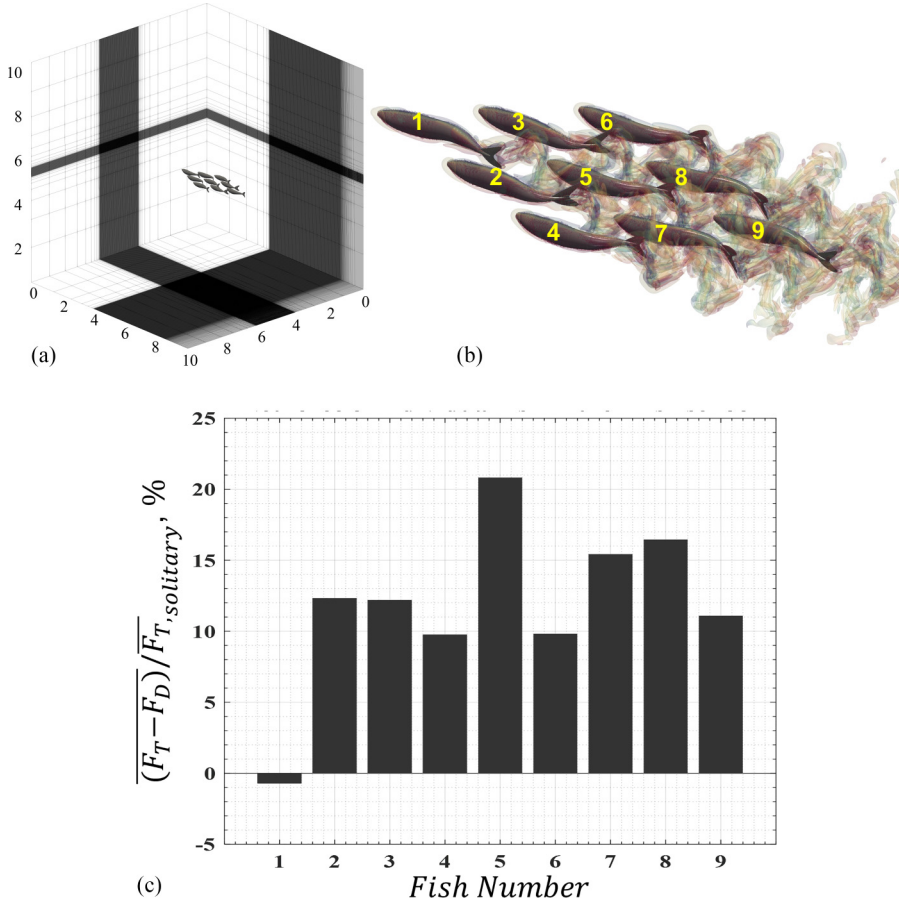


FIG. 8. Direct numerical simulation of a school of 9 fish. (a) Fish school model immersed in the Cartesian grid. (b) 3D vortical structures. (c) Net force in the surge direction normalized by the mean thrust of a single fish. Number indicates individual fish in the school shown in (b). Figure provided by Ji Zhou.

C. Turbulent boundary layer over sharkskin denticles

Boomsma and Sotiropoulos [110] performed direct numerical simulations of flow over sharkskin denticles in turbulent channel flow using a sharp-interface IBM [111]. A representative denticle from *Isurus oxyrinchus* (shortfin Mako) was scanned using micro-CT, and a total of 324 individual denticles were used to cover the channel wall. In the baseline simulation, the Reynolds number based on the bulk flow velocity was $Re = 2800$ ($Re_\tau = 180$), and about 130 million grid points were used to resolve the turbulent flow over the surface covered by the denticles. Figure 10 shows a close-up view of the flow field around the denticles. The denticles were found to increase the total drag on the wall (see Fig. 10), and the change in drag force due to the denticles compared reasonably well to the experimental measurement [112].

IBM-based simulations of transitional and turbulent wall bounded flows with and without complex surface geometries have also been simulated in several other studies; see, for instance, Refs. [113,114].

D. Flight aerodynamics of a hummingbird

Figure 11 shows the simulation of hummingbird hovering performed by Song *et al.* [116]. 3D high-fidelity wing kinematics of hummingbird hovering was reconstructed from high-speed videos

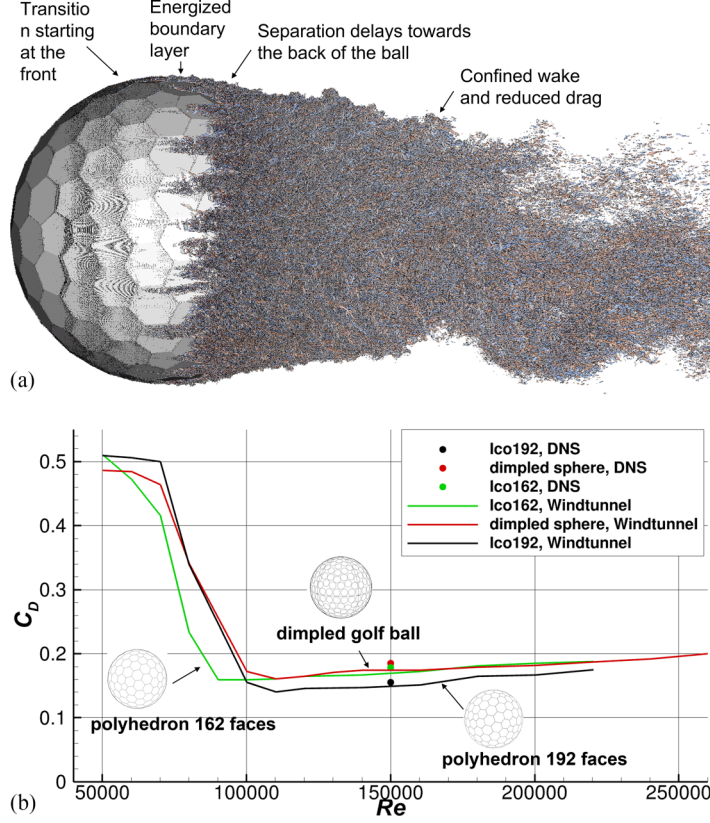


FIG. 9. (a) Isosurface of the Q criterion visualizing vortical structures near the top of the tessellated sphere at $Re = 1.1 \times 10^5$. (b) Drag coefficient vs Re for three different golf balls, with dimples, tessellated with 162 polyhedron faces, and 192 faces. Wind tunnel measurements are shown with solid line, and DNS values are shown with dots. Figures provided courtesy of Elias Balaras.

and was used as input for simulation of the unsteady aerodynamics of a hovering hummingbird. As shown in Fig. 11(a), the 3D model of the hummingbird was immersed in the Cartesian grid, and the flow simulation was done by a sharp-interface IBM [117]. The flow Reynolds number based on the average wing-tip velocity was 3000 and a total of 30 million grid points were used to resolve the flow field. For the validation of the simulation, the phase-averaged circulation around the wing was compared with the experiment data of Warrick *et al.* [115], and Fig. 11(c) shows that the circulation at 50% wingspan matches reasonably well with the experimental data.

Other examples of IBM simulations in the arena of biolocomotion that have been verified against experiments are those for a hovering moth [118] and a pectoral fin of a fish [119].

E. Impact of a sphere with an elastic membrane in fluid

This case highlights the ability of IB methods to model complex fluid-structure interaction (FSI) problems. Verzicco and Querzoli [60] analyzed the impact of a rigid spherical pendula on rubber membranes [Fig. 12(a)] in a fluid at different Reynolds numbers to understand the contact dynamics in deformable bodies in a viscous fluid. They investigated the problem by both laboratory and numerical experiments [Fig. 12(b)] and developed a new contact model to perform the simulations. The simulations were carried out using the sharp-interface IB method of De Tullio and Pascazio [89]. Simulations employed grids with $O(100)$ million grid points and Reynolds numbers extended

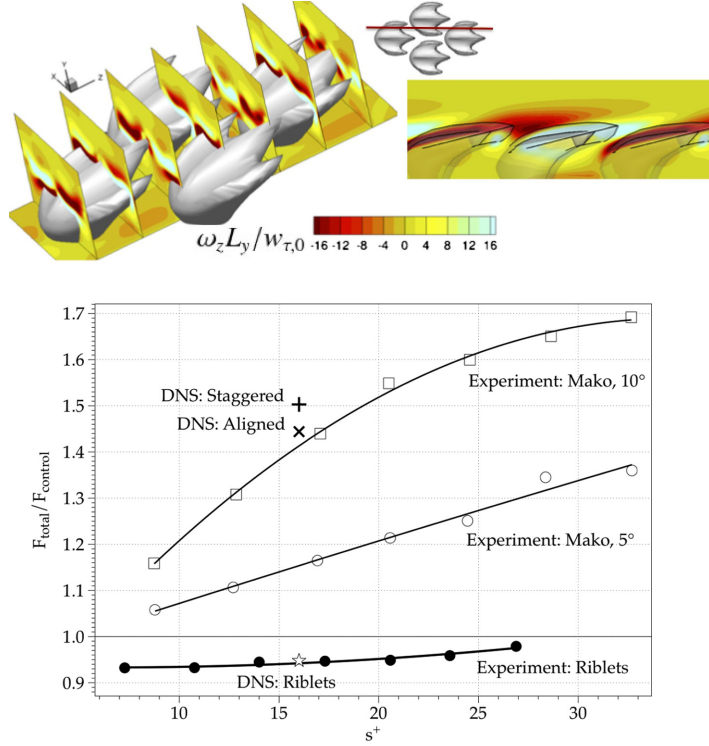


FIG. 10. Direct numerical simulation of a turbulent boundary layer at $Re_\tau = 180$ over a surface with modeled sharkskin denticles using a sharp-interface IBM [110]. Top figure shows views of mean streamwise vorticity for a staggered arrangement of denticles. Inset shows the spanwise location of side view. Bottom figure shows the ratio between total drag with and without sharkskin (and with and without riblets) from the present DNS. Fitted curves, open squares or circles, and filled circles are experimental data. The + and \times mark IBM based DNS data for the staggered and aligned denticle cases, respectively. The open star is the IBM-based DNS result for the riblets. Reprinted from Ref. [110] with the permission of AIP Publishing.

up to 1000. They found that the collision dynamics depended on many parameters, the most important ones being the impact Stokes number and the ratio of the membrane thickness-to-sphere diameter. Extensive comparisons were made with the experimental results [Fig. 12(c)], and the simulations were found to accurately predict the features of the impact on the sphere and the membrane.

Other studies involving IB simulations of FSI problems with verification against experiments can be found in the literature (see, for instance, Refs. [120–122]), and the interested reader is referred to the review article of Griffith and Patankar [123] for a comprehensive discussion of this topic.

F. Generation of wing tones by flying mosquitoes

Multiphysics modeling such as the FSI modeling shown in the previous section, is a particular strength of IB methods. In this subsection, as well as the next, we show two additional examples of multiphysics simulations done using IBMs. Seo *et al.* [124] studied wing tone generation by mosquitoes using the IBM flow simulation coupled with aeroacoustic sound prediction based on the Ffowcs Williams and Hawkins (FW-H) equation [125]. Figure 13(a) shows the instantaneous vortical structure around the flapping mosquito wing at the phase of peak lift during down-stroke. The wing tone was then predicted by the FW-H equation with the time-dependent surface pressure data obtained from the flow simulations. Figure 13(b) shows 3D sound pressure level (SPL)

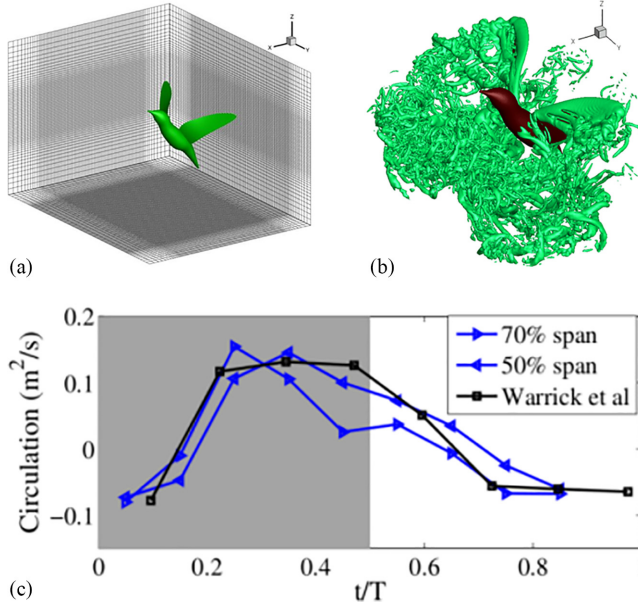


FIG. 11. IBM simulation of hummingbird hovering. (a) 3D kinematic model of hummingbird immersed in the Cartesian grid. (b) 3D vortical structures in the flow. (c) Comparison of circulation around the wing with experiment of Warrick *et al.* [115]. Figures provided courtesy of Haoxiang Luo.

directivity pattern. The predicted wing tone compared well with the measurement of Arthur *et al.* [126].

In order to further validate the computational methods, a joint experimental-computational study has also been performed [127]. In the study, tethered mosquitoes were imaged via a high-speed video camera and the wing tone sounds were recorded by a microphone array. Simulations of the flow and acoustics were carried out with the wing kinematics extracted from the experiments, and Fig. 13(c) shows the wing tone SPL directivity on the coronal plane at 10 cm distance for the fundamental wing-beat frequency. The agreement between the experimental measurements and the predictions from the simulations was found to be excellent, indicating the fidelity of the IBM-based aeroacoustic modeling approach.

G. Heart murmurs generated by turbulent flow in the aorta

This case is specifically for the heart murmurs generated by a valvular constriction [83] in the human aorta, but it generally relates to structural acoustics associated with turbulent wall pressure fluctuations such as in hydroacoustic noise [128].

The *in vitro* model consisted of a stenosed tube embedded inside a cylinder made of a tissue mimicking viscoelastic material [see Fig. 14(a), middle]. The flow through the tube, which was at a $Re_D = 4,000$, transitions to turbulence downstream of the constriction and generates wall pressure fluctuations which subsequently generate acoustic waves that propagate through the surrounding viscoelastic material to the surface. Surface measurements of the acoustic wave for this setup were conducted in an anechoic chamber with a contact microphone (BIOPAC) and an accelerometer (HP).

The configuration was modeled computationally with a sharp-interface incompressible immersed boundary flow solver (ViCar3D) [72] coupled with a high-order, sharp-interface immersed boundary solver for acoustic wave propagation [83] in viscoelastic media. Figure 14(a) shows the turbulent flow structures in the stenosed tube as well as the elastic waves generated in the surrounding

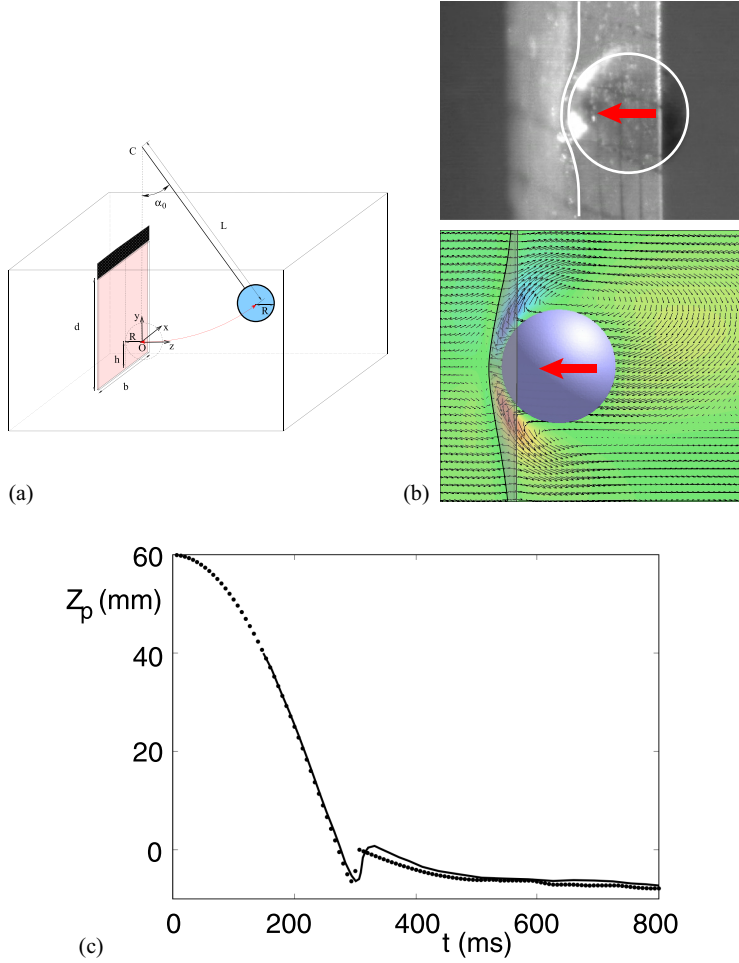


FIG. 12. (a) Schematic of the problem configuration. (b) (Upper) Experimental visualisations of the pendulum approaching the membrane. (Lower) Numerical results for $Re = 1000$ of the vertical velocity component overlaid with velocity vectors in the symmetry y - z plane. The thin solid line shows the computed membrane profile. (c) Comparison of numerical (symbols) and experimental (line) time evolution of the vertical coordinate of the pendulum center for a membrane for a selected case. Figures provided courtesy of Roberto Verzicco.

viscoelastic medium by the wall pressure fluctuation. The comparison of the surface fluctuations between the simulations and experiment [Fig. 14(b)] showed excellent agreement over a wide range of frequencies and spatial locations [83]. This study showcased the ability of the sharp-interface IB approach to predict not only turbulent wall pressure fluctuations but also the multiphysics problem of flow-noise and acoustic wave propagation.

H. Flow past a caudal-fin-inspired pitching panel

Figure 15 shows the simulation of flow past a pitching panel at $Re = 10\,200$ and Strouhal number of 0.27 . The panel was sinusoidally pitched with a peak-to-peak amplitude of 15° . The simulation was carried out by Zhang *et al.* [129] using a sharp-interface immersed boundary method with local grid refinement. The computational domains with different refinement levels and baseline Cartesian mesh are shown in Fig. 15(a). To resolve the flow structures at this high Reynolds number, two layers

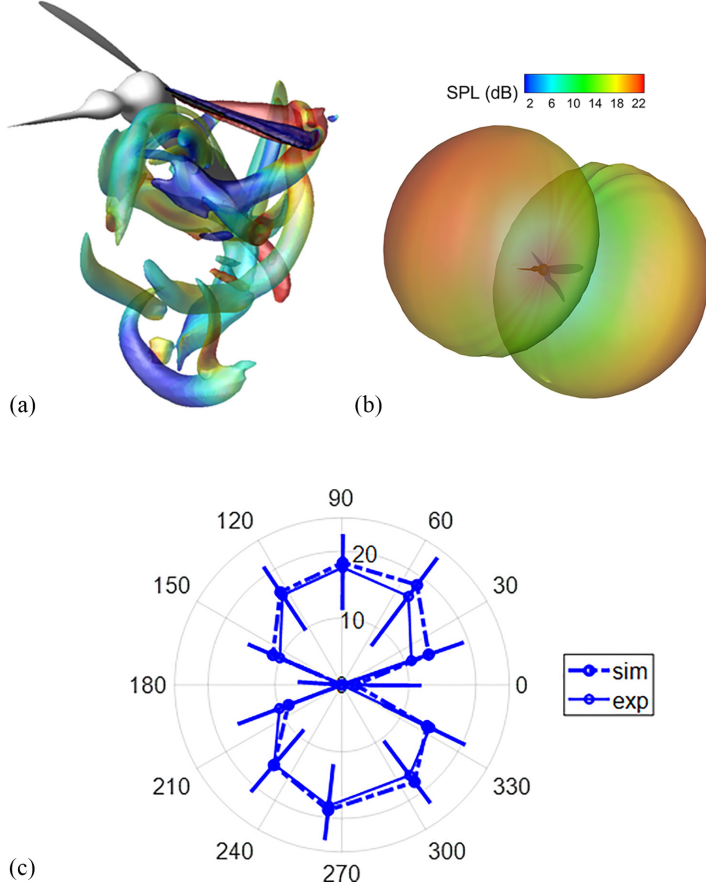


FIG. 13. Prediction of wing tones by a flying mosquito. (a) Vortex structures around flapping wing of mosquito resolved by the IBM simulation. (b) 3D sound pressure level (SPL) directivity pattern for the fundamental wing beat frequency at 10 cm distance. (c) Comparison of SPL directivity on the coronal plane at 10 cm distance. sim: Prediction from the simulation. exp: Experimental measurements by microphone arrays. Error bars denote cycle-to-cycle variation.

of refined mesh blocks were employed, and the finest resolution around the panel was $0.0052C$ (C is the chord length of the panel). The total number of grid points used was around 15.4 million. The comparisons of computed wake structures against the PIV measurements of King *et al.* [130] are shown at two time instances in the pitching cycle in Fig. 15(b), and the simulation results were found to match very well with the experiment.

I. High-speed wake flows

Figures 16 and 17 show the results of a high-order sharp-interface IBM (ViCAS3D [84]) applied to compressible flows. The simulation of flow past a circular cylinder at $Re = 200\,000$ and flow Mach number, $M = 1.7$ is shown in Fig. 16. A sharp-interface IBM with high-order polynomial interpolation [82] was used, and the cylinder diameter was resolved by 400 grid points. Density gradient magnitude contours shown in Fig. 16(a) indicate various flow features typical of a high-speed compressible flow including bow, separation, and oblique shocks. The pressure coefficient (C_p) on the cylinder surface computed from this IBM simulation is compared with the results from other BCG as well as IBM simulations in Fig. 16(b) and found to match reasonably well.

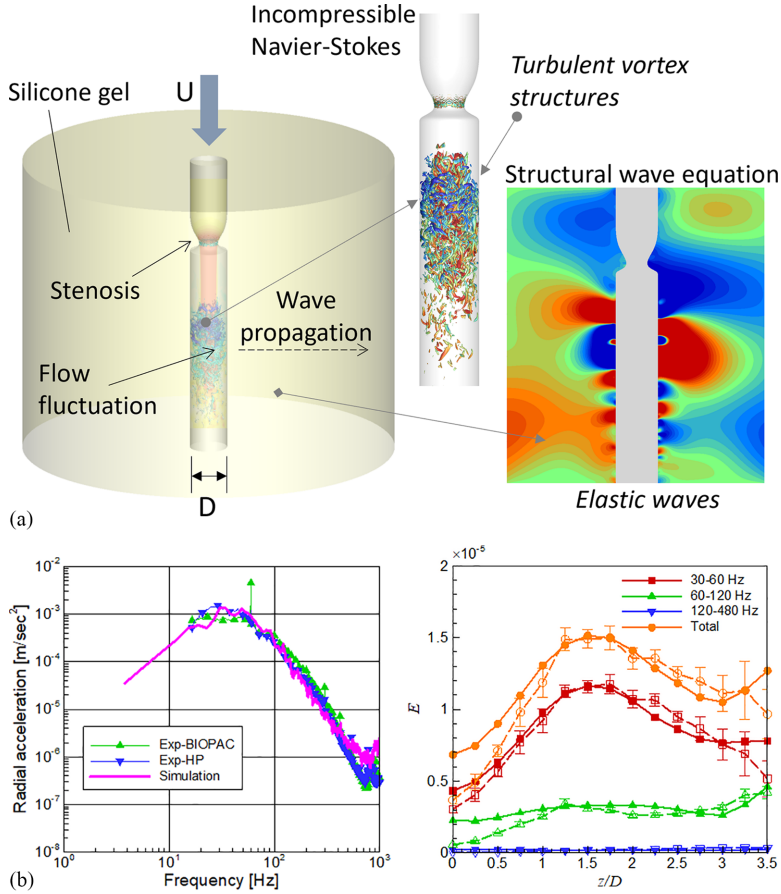


FIG. 14. (a) Modeling of heart murmur generated by a constriction in an artery. (Left) Schematic of the setup turbulent flow in the stenosed tube embedded within a tissue mimicking viscoelastic cylinder. (Middle) Turbulent flow in the stenosed tube. (Right) Elastic waves in the viscoelastic media resolved by the simulation. (b) Comparisons between the simulation results and the experimental measurements. (Left) Frequency spectra of radial accelerations measured on the outer surface of gel material. Exp-BIOPAC: measurement by BIOPAC contact microphone, Exp-HP: measurement by HP accelerometer. (Right) Variation of spectral energy of the signal for three different frequency bands; solid line with filled symbol: simulation; dashed line with hollow symbol and error bar: experimental measurement.

Figure 17 shows the simulation of flow past a sphere at $Re = 50\,000$ and $M = 4.0$. A total of 90 million grid points were used in this simulation. A bow shock in front of the sphere and complex vortex structures in the wake are clearly visible. The drag coefficient obtained from the simulation was 0.98, which is within 6% of the reported value 1.04 [131].

Other compressible flow simulations using IBM that have been verified against experiments or BCG methods are by De Tulio *et al.* [132], Nam and Lien [133], Al-Marouf and Samtaney [134], and Mao *et al.* [135].

J. Shock-void interaction

IB methods have been applied to a variety of compressible multiphase and multimaterial flows. Figure 18 shows the collapse of a void pore in the polymethyl methacrylate (PMMA) medium by the interaction with the shock wave at 2.3 km/s which was simulated by the sharp-interface IBM based

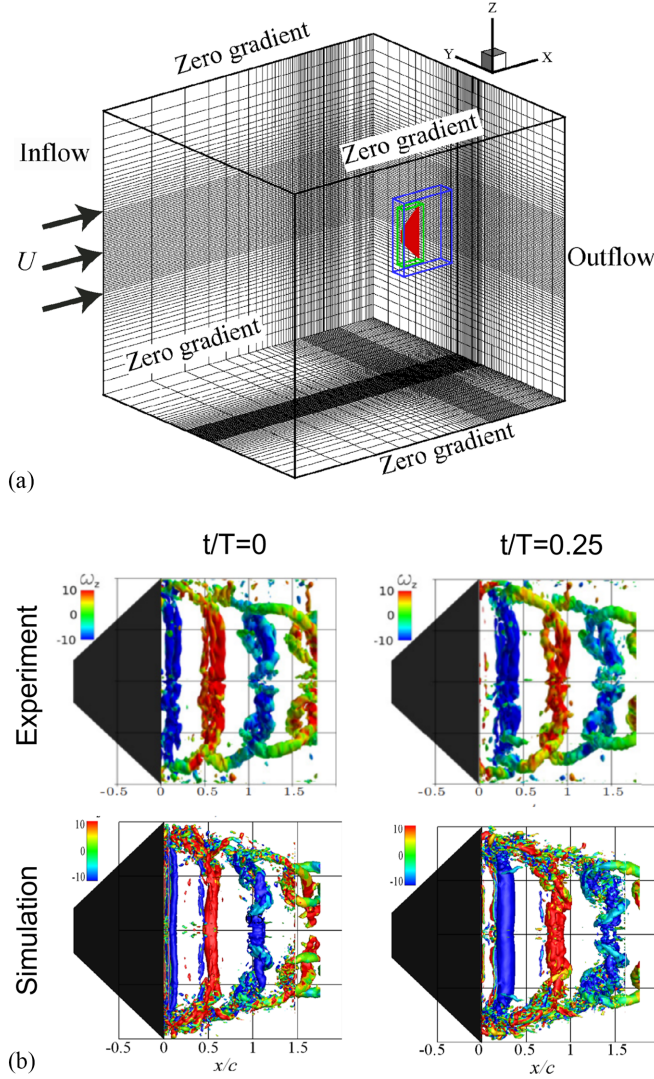


FIG. 15. Simulation of flow past a pitching panel at $Re = 10\,200$ and Strouhal number of 0.27 carried out by Zhang *et al.* using a sharp-interface immersed boundary method [129]. Comparison of the computed wake against the experiments of King *et al.* [130] is shown at two time instances in the pitching cycle. Figures provided courtesy of Haibo Dong.

multiphase flow solver SCIMITAR3D [136,137]. The interface was represented via level sets and resolved by a ghost-fluid method. Figure 18(a) shows a 3D void interface and temperature contours on the cross section, and the evolution of the void interface is found to compare reasonable well with experimental measurements [Fig. 18(b)].

K. Wall-modeled LES of hypersonic ramp flow

The final case presented in this section is of a wall-modeled large-eddy simulation of a hypersonic turbulent boundary layer traversing a compression ramp. The Reynolds and Mach numbers for the flow are 1.89×10^7 and 7.2, respectively. The finite-difference method employed here has a spatial accuracy that ranges between second and third order, and the IB method implemented in the solver

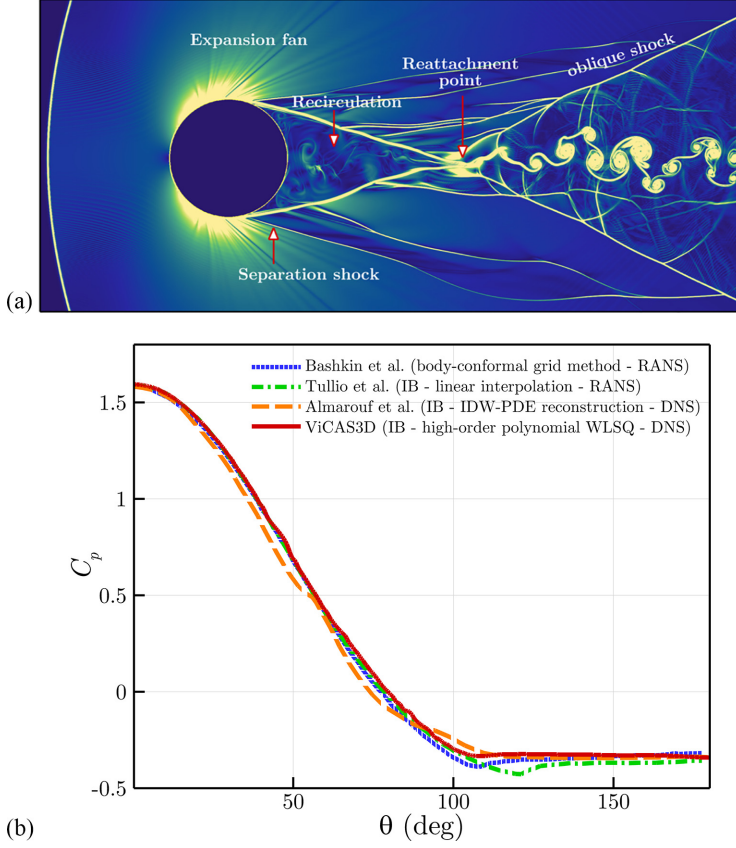


FIG. 16. Flow past a circular cylinder at $Re = 200\,000$ and flow Mach number, $M = 1.7$ using the code described by Turner *et al.* [84] (a) Density gradient magnitude $|\nabla \rho^*|$ produced by a circular cylinder. (b) Comparison of cylinder pressure coefficient C_p with other simulation results. IB: immersed boundary, URANS: unsteady Reynolds averaged Navier-Stokes, DNS: direct numerical simulation. Figure provided by Jacob Turner.

employs ghost-point based interpolations to impose the boundary conditions. As pointed out earlier, sharp-interface methods experience localized grid discontinuities in the boundary cells, and these can be particularly important in scenarios characterized by complex boundary conditions and the implementation of wall models for turbulence can be more difficult than for BCG methods. This case therefore addresses both of these challenges.

The results in the Fig. 19 as well as the other test cases in this study suggest that wall-modeled LES with IBMs can provide reasonable results, but the authors noted the need for particular attention to the numerical accuracy and dissipation characteristics of the numerical treatment near the wall.

VIII. FREQUENTLY ASKED QUESTIONS

In this section we address some of questions that come up frequently regarding the capabilities of IBMs. These questions provide a practical and useful context for comparing IB and BCG methods in CFD.

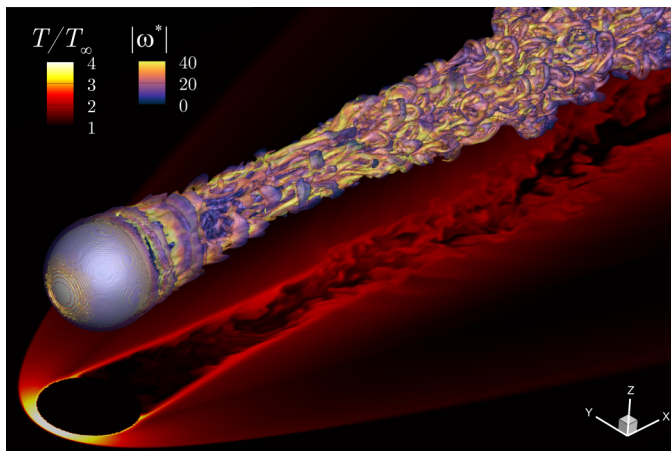


FIG. 17. Flow past a sphere at $Re = 50\,000$ and flow Mach number, $M = 4.0$. Flow field at an early time in the flow development visualized by Q criterion colored by streamwise vorticity with the temperature contours shown in the $z/D = 0$ plane. Figure provided by Jacob Turner.

A. Can IB methods compute boundary layer flows accurately?

The implicit assumption underlying this query is that all IBMs (immersed boundary methods) are diffuse-interface approaches incapable of enforcing boundary conditions with the same level of precision as BCG (boundary-conforming grid) methods. Consequently, they are believed to fall short in accurately capturing boundary layers. Nevertheless, as succinctly summarized earlier, the past two decades have witnessed a predominant shift in the realm of IBMs towards sharp-interface methodologies with second-order (and even higher; see Refs. [82,138] for examples) levels of both local and global accuracy. These advanced techniques deliver a degree of precision and exactitude in boundary layer resolution that mirrors the capabilities of BCG methods (as discussed in Sec. VII). Indeed, beyond the few examples showcased here, there exist several studies where sharp-interface IBMs have been used to accurately predict the evolution of boundary layers on complex surfaces, thereby answering the above question in the affirmative.

B. Can IB methods simulate flows at high Reynolds numbers?

Given that away from the immersed boundary, the discretization of IB methods is similar to that of BCG methods, the question at hand implicitly concerns the ability of IBMs to accurately resolve flow in the vicinity of the immersed body at high Reynolds numbers. As mentioned in the preceding paragraph, sharp-interface methods possess capabilities akin to BCG methods in resolving boundary layers. Therefore, simulating high Reynolds number flows hinges on the capacity to increase the resolution within the boundary layers.

BCG methods generally allow for a finer resolution perpendicular to the wall with a level of precision that is typically not attainable with the Cartesian grids commonly employed in IBMs. As highlighted by Mittal and Iaccarino in their work [65], this discrepancy results in a more rapid increase in overall grid size with Reynolds numbers for IBMs compared to BCG methods. Curvilinear as well as unstructured and overset grid methods, however, have a much higher operation count per grid point than IB methods. This is because they have significantly larger number of terms in the equations, and the condition number of discretized operators can also be larger, resulting in slow convergence. Thus, a larger grid for IB methods does not necessarily imply an equivalently larger CPU time for the simulations. Second, for immersed boundaries with geometrical complexities such as high curvatures, corners, and junctions, BCGs can also have difficulties in imparting high resolution that is limited to only the boundary layer regions while maintaining grid quality.

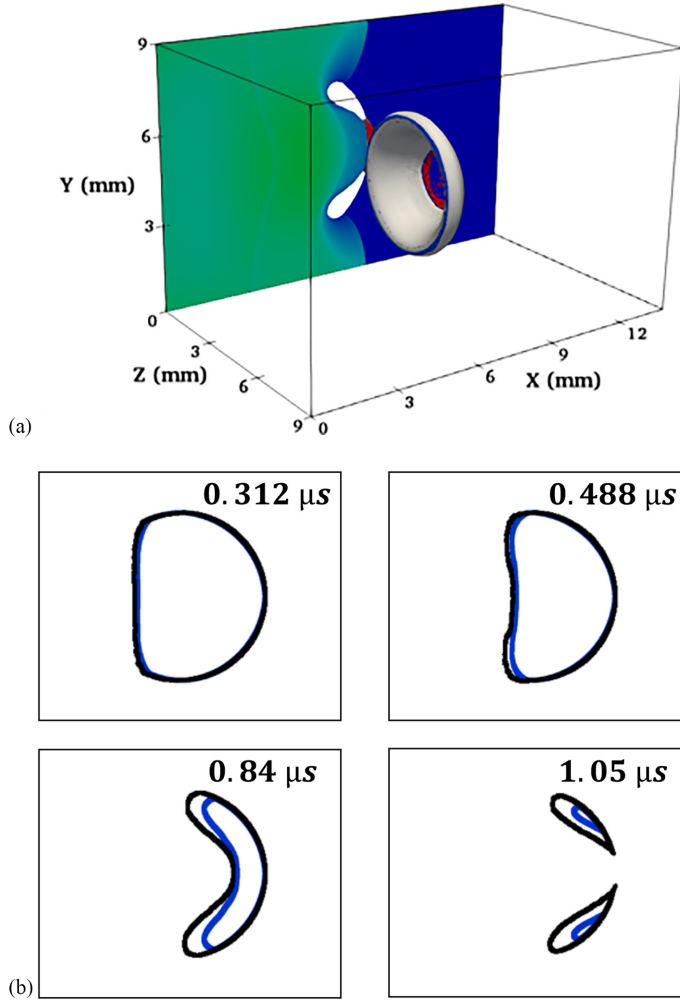


FIG. 18. Simulation of shock-void interaction. (a) Instantaneous void interface and temperature contours on the cross section. (b) Time evolution of void interface compared with experimental measurement. Black: experiment. Blue: simulation. Figures provided courtesy of H. S. Udaykumar.

Finally, the use of local-grid refinement [129,132,136,139,140] allows IB methods to provide high resolution in regions around the immersed body in a more selective manner, but it diminishes the structured nature of the mesh and has implications for the convergence of the sparse systems that result from the discretization. In the previous section we have provided examples and references for many studies for high Reynolds number flows where IB methods have generated verifiable results, including surface-related quantities such as pressure and drag.

The question in the end comes down to IBM's computational efficiency compared to the BCG method. Such comparisons are made complex due to various factors like solver selection, test-case choice, and computing hardware. One such study along these lines was by Capuano *et al.* [141] who compared three solvers for simulating flow over a sphere at $Re = 3700$ Nek5000 (a high-order spectral-element code), OpenFOAM (a general-purpose unstructured finite-volume solver), and their in-house Cartesian IBM solver. The study indicated that Nek5000 and IBM performed similarly in terms of cost effectiveness for global and local flow properties. Conversely, OpenFOAM required significantly more degrees of freedom (and higher cost) to match key features like the

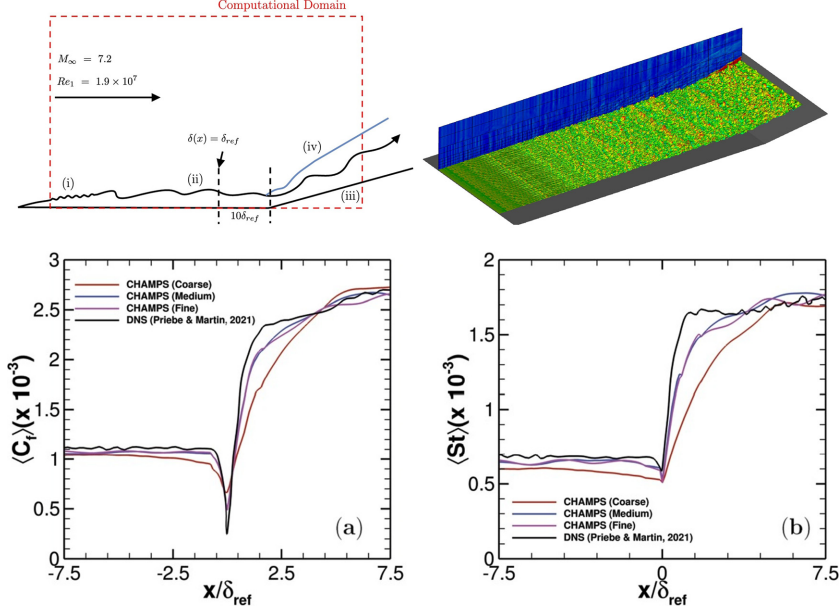


FIG. 19. Results from IBM wall-modeled LES of hypersonic flow over a 8° compression corner by van Noordt *et al.* [99]. Top left: Schematic diagram for the compression corner case. Top right: Instantaneous flow field visualization for the test case showing isosurfaces of temperature colored by the streamwise velocity. Bottom: Comparison of surface quantities Stanton number (left) and skin friction coefficient (right) against DNS. Reproduced with permission from William van Noordt.

downstream recirculation bubble length. At finer resolutions, the three codes closely agreed on most flow metrics. The authors concluded that high-order methods and second-order, energy-conserving IBM approaches are viable for high-fidelity simulations of turbulent flows with separation. Similar comparisons for other complex geometries and moving boundary problems would go a long way in providing informing CFD practitioners about the computational costs of IBM simulations.

C. Can IB methods resolve turbulent flows?

As with the previous question, this question also relates to the ability of IBMs to resolve turbulent flows that occur in the vicinity of immersed boundaries. As outlined in the previous two paragraphs, the numerical accuracy and fidelity of sharp-interface IB methods is similar to standard BCG methods, and thus the question really boils down to the ability of IB methods to provide adequate resolution near the immersed boundary to resolve turbulence.

In order to examine this question, it is useful to consider two relevant resolution requirements for DNS of canonical attached turbulent boundary layers. The first is that the grid resolution requirement at the wall be $(\Delta x^+, \Delta y^+, \Delta z^+) \approx (12, 0.1, 7)$ [142], which implies a grid aspect ratio near the wall of roughly 120:1:70. Second, the wall-normal grid size can increase by a factor of about 50 in the outer regions of the turbulent boundary layer. BCG methods can potentially take advantage of these requirements by employing grids that are highly nonisotropic ($\Delta x; \Delta z \gg \Delta y$) near the wall, and which expand rapidly away from the wall. A state-of-the-art example of this is the CharLES code developed at the Center for Turbulence Research [143]. In IB methods on the other hand, since the underlying Cartesian grid is not necessarily aligned with the boundary, there is no easy way to fully exploit these resolution requirements. While local refinement techniques [129, 132, 136, 139, 140] can be used to take advantage of the second condition, most IB methods, even those that employ local-grid refinement, use isotropic grids, i.e., grids with $\Delta x \approx \Delta y \approx \Delta z$,

near the immersed boundary, and therefore do not exploit the first condition. Thus, for canonical turbulent flows, the size of the grid for IB methods is significantly larger than for a corresponding BCG method.

This difference between IB and BCG method, however, diminishes for geometrically complex boundaries, such as for rough walls [144] or for noncanonical turbulent flows such as those with separation [145,146], since such flows require near-wall grids that are more isotropic. Indeed, several examples exist for such flows being simulated successfully with sharp-interface IB methods (see Ref. [98,114,147]) including those shown here [107,108,110]. In fact, for this class of turbulent wall bounded flows, the ability of IB methods to easily handle geometric complexity gives them, in our view, an edge over BCG methods. Similarly, the IB method generally has an advantage in simulating turbulent flow with moving boundaries for all the reasons outlined in this article.

Incorporating Reynolds-averaged Navier-Stokes (RANS), wall-resolved LES, and wall-modeled LES into Cartesian grid IBMs may introduce certain added intricacies. However, these challenges are certainly surmountable, as evidenced by several successful implementations [98,148–150].

Finally, as pointed out earlier, the lack of grid skewness related errors in IBM could be a significant advantage for these method over BCG methods, since vortices and turbulence structures away from the immersed boundary can be resolved and convected over Cartesian grids with higher fidelity than on nonorthogonal curvilinear grids.

D. Can IB method simulate high-speed flows?

In the context of IB methods, flow compressibility brings in a few additional aspects for consideration. These are the appearance of complex heat-flux boundary conditions at the surface associated with the energy equation, and the need to resolve shocks in the vicinity of the body. As pointed out by Mittal and Bhardwaj [151], the thermal boundary conditions for compressible flows have been incorporated in several different sharp-interface IB solvers. For example, using the ghost-cell-based sharp-interface IBM, Dirichlet as well as von Neumann-type boundary conditions for arbitrary variables can be imposed on the solid surface and interface [82], which enables the incorporation of heat and mass flux boundary conditions necessary for the high-speed compressible flows [84]. There also exist several examples of simulations at high Mach numbers (see Figs. 16–19 and references in Mittal and Bhardwaj [151]) with IB methods that resolve shock-boundary layer interaction. Therefore, while addressing local numerical accuracy near the immersed body is paramount for high-speed compressible flows, this challenge is surmountable. Nonetheless, it's imperative to recognize that these flows often occur at high Reynolds numbers, warranting consideration of the three additional issues delineated earlier in this section.

IX. CLOSING

This perspective piece on immersed boundary methods draws inspiration from the aphorism “Those who do not understand history may be forced to repeat it.” IB methods are increasingly becoming an essential tool for fluid dynamicists who are keen to attack flows problem in all their native complexity. Our hope is that by comprehending the historical trajectory of the advancements in these methods—why, when, and how they were developed—researchers are better prepared to critically evaluate these methods and empowered to innovate in ways that significantly advance the state of the art in these methods.

ACKNOWLEDGMENTS

Support from the NSF (Grants No. CBET-2034983, No. CBET-1738918, No. PHY-1806689, No. DUE-1734744, No. CBET-2019405, and No. CBET-1357819), AFOSR (Grants No. FA9550-21-1-0286 and No. FA9550-23-1-0010), ONR (Grants No. N00014-22-1-2655 and No. N00014-22-1-2770), ARO (Cooperative Agreement No. W911NF2120087), Human Frontier Science Program (RGP0038/2019) and NIH (5R21GM139073-02) is gratefully acknowledged. We thank our fellow

IBM enthusiasts Elias Balaras, Haibo Dong, Haoxiang Luo, H.S. Udaykumar, Fotis Sotiropoulos, and Roberto Verzicco for graciously allowing us to use their data and plots in this article. We thank Ji Zhou and Jacob Turner from Johns Hopkins University for providing plots on fish schooling and compressible wake flows, respectively. Finally, we thank the anonymous reviewer who provided thoughtful comments on our submission.

- [1] C. S. Peskin, Flow patterns around heart valves: A digital computer method for solving the equations of motion, Ph.D. thesis, Albert Einstein College of Medicine of Yeshiva University (1972).
- [2] J. E. Fromm and F. H. Harlow, Numerical solution of the problem of vortex street development, [Phys. Fluids](#) **6**, 975 (1963).
- [3] F. H. Harlow and J. E. Fromm, Dynamics and heat transfer in the von Kármán wake of a rectangular cylinder, [Phys. Fluids](#) **7**, 1147 (1964).
- [4] F. H. Harlow and J. E. Welsh, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, [Phys. Fluids](#) **8**, 2182 (1965).
- [5] J. A. Viegelli, A method for including arbitrary external boundaries in the MAC incompressible fluid computing technique, [J. Comput. Phys.](#) **4**, 543 (1969).
- [6] J. A. Viegelli, A computing method for incompressible flows bounded by moving walls, [J. Comput. Phys.](#) **8**, 119 (1971).
- [7] A. A. Amsden and C. W. Hirt, A simple scheme for generating general curvilinear grids, [J. Comput. Phys.](#) **11**, 348 (1973).
- [8] A. W. Rizzi and M. Inouye, Time-split finite-volume method for three- dimensional blunt-body flow, [AIAA J.](#) **11**, 1478 (1973).
- [9] T. Gal-Chen and R. C. Somerville, On the use of a coordinate transformation for the solution of the Navier-Stokes equations, [J. Comput. Phys.](#) **17**, 209 (1975).
- [10] T. K. Hung and T. D. Brown, An implicit finite-difference method for solving the Navier-Stokes equation using orthogonal curvilinear coordinates, [J. Comput. Phys.](#) **23**, 343 (1977).
- [11] F. C. Thames, J. F. Thompson, C. W. Mastin, and R. L. Walker, Numerical solutions for viscous and potential flow about arbitrary two-dimensional bodies using body-fitted coordinate systems, [J. Comput. Phys.](#) **24**, 245 (1977).
- [12] T. L. Clark, A small-scale dynamic model using a terrain-following coordinate transformation, [J. Comput. Phys.](#) **24**, 186 (1977).
- [13] J. T. Oden, Finite-element analogue of Navier-Stokes equation, [J. Eng. Mech. Division](#) **96**, 529 (1970).
- [14] R. T. S. Cheng, Numerical solution of the Navier-Stokes equations by the finite element method, [Phys. Fluids](#) (1958–1988) **15**, 2098 (1972).
- [15] C. Taylor and P. Hood, A numerical solution of the Navier-Stokes equations using the finite element technique, [Comput. Fluids](#) **1**, 73 (1973).
- [16] J. F. Thompson, F. C. Thames, and C. W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, [J. Comput. Phys.](#) **15**, 299 (1974).
- [17] M. Vinokur, Conservation equations of gasdynamics in curvilinear coordinate systems, [J. Comput. Phys.](#) **14**, 105 (1974).
- [18] B. Launder and D. Spalding, The numerical computation of turbulent flows, [Comput. Methods Appl. Mech. Eng.](#) **3**, 269 (1974).
- [19] C. W. Hirt, An arbitrary Lagrangian-Eulerian computing technique, in *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics*, edited by S. B. Heidelberg (1971), pp. 350–355.
- [20] C. W. Hirt, A. A. Amsden, and J. L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, [J. Comput. Phys.](#) **14**, 227 (1974).

- [21] J. L. Steger, F. C. Dougherty, and J. A. Benek, A chimera grid scheme, in *Advances in Grid Generation: Presented at Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, Texas, June 20–22, 1983*, Vol. 5, edited by K. N. Ghia and U. Ghia (American Society of Mechanical Engineers, 1983).
- [22] J. Benek, P. Buning, and J. Steger, A 3-D chimera grid imbedding technique. AIAA paper 85-1523, in *7th Computational Fluid Dynamics Conference, Cincinnati, OH* (1985).
- [23] C. S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.* **10**, 252 (1972).
- [24] C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
- [25] M. F. McCracken and C. S. Peskin, A vortex method for blood flow through heart valves, *J. Comput. Phys.* **35**, 183 (1980).
- [26] D. M. McQueen and C. S. Peskin, Computer-assisted design of pivoting-disc prosthetic mitral valves, *J. Thorac. Cardiovasc. Surg.* **86**, 126 (1983).
- [27] D. M. McQueen and C. S. Peskin, Computer-assisted design of butterfly bileaflet valves for the mitral position, *Scand. J. Thorac. Cardiovasc. Surg.* **19**, 139 (1985).
- [28] D. M. McQueen and C. S. Peskin, A three-dimensional computational method for blood flow in the heart. II. Contractile fibers, *J. Comput. Phys.* **82**, 289 (1989).
- [29] L. J. Fauci and C. S. Peskin, A computational model of aquatic animal locomotion, *J. Comput. Phys.* **77**, 85 (1988).
- [30] L. J. Fauci, Interaction of oscillating filaments: A computational study, *J. Comput. Phys.* **86**, 294 (1990).
- [31] R. J. LeVeque, C. S. Peskin, and P. D. Lax, Solution of a two-dimensional cochlea model using transform techniques, *SIAM J. Appl. Math.* **45**, 450 (1985).
- [32] R. P. Beyer, A computational model of the cochlea using the immersed boundary method, *J. Comput. Phys.* **98**, 145 (1992).
- [33] R. Dillon, L. Fauci, A. Fogelson, and D. Gaver III, Modeling biofilm processes using the immersed boundary method, *J. Comput. Phys.* **129**, 57 (1996).
- [34] M. Briscolini and P. Santangelo, Development of the mask method for incompressible unsteady flows, *J. Comput. Phys.* **84**, 57 (1989).
- [35] P. Angot, C. H. Bruneau, and P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, *Numer. Math.* **81**, 497 (1999).
- [36] D. Goldstein, R. Handler, and L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* **105**, 354 (1993).
- [37] E. M. Saiki and S. Biringen, Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method, *J. Comput. Phys.* **123**, 450 (1996).
- [38] D. K. Clarke, H. A. Hassan, and M. D. Salas, Euler calculations for multielement airfoils using Cartesian grids, *AIAA J.* **24**, 353 (1986).
- [39] R. Gaffney, Jr. and H. Hassan, Euler calculations for wings using Cartesian grids, in *25th AIAA Aerospace Sciences Meeting* (1987), p. 356.
- [40] P. D. Frymier, H. A. Hassan, and M. D. Salas, Navier-Stokes calculations using Cartesian grids I—Laminar flows, *AIAA J.* **26**, 1181 (1988).
- [41] M. J. Berger and R. J. Leveque, An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries, AIAA paper **1930** (1989).
- [42] D. DeZeeuw and K. G. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, *J. Comput. Phys.* **104**, 56 (1993).
- [43] J. J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, *Comput. Fluids* **23**, 125 (1994).
- [44] J. E. Melton, M. J. Berger, M. J. Aftosmis, and M. D. Wong, 3D applications of a Cartesian grid Euler method, AIAA Paper 95-0853 (1994).
- [45] M. Nemec, M. Aftosmis, and T. Pulliam, Cad-based aerodynamic design of complex configurations using a Cartesian method, in *42nd AIAA Aerospace Sciences Meeting and Exhibit* (2004), pp. 2004–113.
- [46] R. Verzicco, Immersed boundary methods: Historical perspective and future outlook, *Annu. Rev. Fluid Mech.* **55**, 129 (2023).

- [47] S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).
- [48] R. J. Leveque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* **31**, 1019 (1994).
- [49] M. C. Lai and C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* **160**, 705 (2000).
- [50] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* **161**, 35 (2000).
- [51] R. Verzicco, J. Mohd-Yusof, P. Orlandi, and D. Haworth, Les in complex geometries using boundary body forces, *Center for Turbulence Research Proceedings of the Summer Program*, NASA Ames-Stanford University, (1998), p. 171.
- [52] J. Mohd-Yusof, Interaction of massive particles with turbulence, Ph.D. thesis, Cornell University (1996).
- [53] H. S. Udaykumar, W. Shyy, and M. M. Rao, Elafint: A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries, *Int. J. Numer. Methods Fluids* **22**, 691 (1996).
- [54] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* **156**, 209 (1999).
- [55] H. S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, *J. Comput. Phys.* **174**, 345 (2001).
- [56] H. Udaykumar, R. Mittal, and W. Shyy, Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids, *J. Comput. Phys.* **153**, 535 (1999).
- [57] C. Wang, X. Wang, and L. Zhang, Connectivity-free front tracking method for multiphase flows with free surfaces, *J. Comput. Phys.* **241**, 58 (2013).
- [58] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, A front-tracking method for the computations of multiphase flow, *J. Comput. Phys.* **169**, 708 (2001).
- [59] H. Liu, S. Krishnan, S. Marella, and H. Udaykumar, Sharp interface Cartesian grid method II: A technique for simulating droplet interactions with surfaces of arbitrary shape, *J. Comput. Phys.* **210**, 32 (2005).
- [60] R. Verzicco and G. Querzoli, On the collision of a rigid sphere with a deformable membrane in a viscous fluid, *J. Fluid Mech.* **914**, A19 (2021).
- [61] D. Boffi and L. Gastaldi, A finite element approach for the immersed boundary method, *Comput. Struct.* **81**, 491 (2003).
- [62] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu, Immersed finite element method, *Comput. Methods Appl. Mech. Eng.* **193**, 2051 (2004).
- [63] R. Löhner, Automatic unstructured grid generators, *Finite Elements Anal. Design* **25**, 111 (1997).
- [64] T. J. Baker, Mesh generation: Art or science? *Prog. Aerospace Sci.* **41**, 29 (2005).
- [65] R. Mittal and G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* **37**, 239 (2005).
- [66] D. Boffi, L. Gastaldi, L. Heltai, and C. S. Peskin, On the hyper-elastic formulation of the immersed boundary method, *Comput. Methods Appl. Mech. Eng.* **197**, 2210 (2008).
- [67] Z. Li and M. C. Lai, The immersed interface method for the Navier-Stokes equations with singular forces, *J. Comput. Phys.* **171**, 822 (2001).
- [68] S. Xu and Z. J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, *J. Comput. Phys.* **216**, 454 (2006).
- [69] M. Meyer, A. Devesa, S. Hickel, X. Y. Hu, and N. A. Adams, A conservative immersed interface method for large-eddy simulation of incompressible flows, *J. Comput. Phys.* **229**, 6300 (2010).
- [70] Y. H. Tseng and J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* **192**, 593 (2003).
- [71] R. Ghias, R. Mittal, and H. Dong, A sharp interface immersed boundary method for compressible viscous flows, *J. Comput. Phys.* **225**, 528 (2007).
- [72] R. Mittal, H. Dong, M. Bozkurtas, F. M. Najjar, A. Vargas, and A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* **227**, 4825 (2008).

- [73] J. Yang and E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, *J. Comput. Phys.* **215**, 12 (2006).
- [74] M. Meinke, L. Schneiders, C. Günther, and W. Schröder, A cut-cell method for sharp moving boundaries in Cartesian grids, *Comput. Fluids* **85**, 135 (2013).
- [75] J. H. Seo and R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, *J. Comput. Phys.* **230**, 7347 (2011).
- [76] R. J. LeVeque, Boundary conditions and ghost cells, in *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics (Cambridge University Press, 2002), p. 129.
- [77] A. Gross and H. Fasel, Characteristic ghost cell boundary condition, *AIAA J.* **45**, 302 (2007).
- [78] E. Motheau, A. Almgren, and J. B. Bell, Navier–Stokes characteristic boundary conditions using ghost cells, *AIAA J.* **55**, 3399 (2017).
- [79] R. Ghias, R. Mittal, and T. Lund, A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries, in *42nd AIAA Aerospace Sciences Meeting and Exhibit* (2004), p. 80.
- [80] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.* **152**, 457 (1999).
- [81] R. P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, *J. Comput. Phys.* **175**, 200 (2002).
- [82] J. H. Seo and R. Mittal, A high-order immersed boundary method for acoustic wave scattering and low-mach number flow-induced sound in complex geometries, *J. Comput. Phys.* **230**, 1000 (2011).
- [83] J. H. Seo, H. Bakhshaei, G. Garreau, C. Zhu, A. Andreou, W. R. Thompson, and R. Mittal, A method for the computational modeling of the physics of heart murmurs, *J. Comput. Phys.* **336**, 546 (2017).
- [84] J. Turner, J. H. Seo, and R. Mittal, Analysis of the flow physics of transonic flutter using energy maps, in *AIAA SCITECH 2023 Forum* (2023), p. 0083.
- [85] D. M. Anderson, G. B. McFadden, and A. A. Wheeler, Diffuse-interface methods in fluid mechanics, *Annu. Rev. Fluid Mech.* **30**, 139 (1998).
- [86] R. Scardovelli and S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* **31**, 567 (1999).
- [87] R. Glowinski, T. W. Pan, and J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.* **112**, 133 (1994).
- [88] M. Vanella and E. Balaras, Short note: A moving-least-squares reconstruction for embedded-boundary formulations, *J. Comput. Phys.* **228**, 6617 (2009).
- [89] M. D. de Tullio and G. Pascasio, A moving-least-squares immersed boundary method for simulating the fluid–structure interaction of elastic bodies with arbitrary thickness, *J. Comput. Phys.* **325**, 201 (2016).
- [90] S. Marella, S. Krishnan, H. Liu, and H. Udaykumar, Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations, *J. Comput. Phys.* **210**, 1 (2005).
- [91] M. P. Kirkpatrick, S. W. Armfield, and J. H. Kent, A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid, *J. Comput. Phys.* **184**, 1 (2003).
- [92] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Stat. Comput.* **7**, 870 (1986).
- [93] J. Kim and P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.* **59**, 308 (1985).
- [94] H. Choi, P. Moin, and J. Kim, Turbulent drag reduction: Studies of feedback control and flow over riblets, Report No. TF-55, Thermosciences Div., Dept. of Mech. Engr., Stanford University (1992).
- [95] R. Mittal and P. Moin, Suitability of upwind-biased finite difference schemes for large-eddy simulation of turbulent flows, *AIAA J.* **35**, 1415 (1997).
- [96] F. Mohaghegh and H. S. Udaykumar, Comparison of sharp and smoothed interface methods for simulation of particulate flows I: Fluid structure interaction for moderate Reynolds numbers, *Comput. Fluids* **140**, 39 (2016).
- [97] F. Mohaghegh and H. S. Udaykumar, Comparison of sharp and smoothed interface methods for simulation of particulate flows II: Inertial and added mass effects, *Comput. Fluids* **143**, 103 (2017).

- [98] X. I. Yang, J. Sadique, R. Mittal, and C. Meneveau, Exponential roughness layer and analytical model for turbulent boundary layer flow over rectangular-prism roughness elements, *J. Fluid Mech.* **789**, 127 (2016).
- [99] W. van Noordt, S. Ganju, and C. Brehm, An immersed boundary method for wall-modeled large-eddy simulation of turbulent high-Mach-number flows, *J. Comput. Phys.* **470**, 111583 (2022).
- [100] F. Duarte, R. Gormaz, and S. Natesan, Arbitrary Lagrangian–Eulerian method for Navier–Stokes equations with moving boundaries, *Comput. Methods Appl. Mech. Eng.* **193**, 4819 (2004).
- [101] D. You, R. Mittal, M. Wang, and P. Moin, Analysis of stability and accuracy of finite-difference schemes on a skewed mesh, *J. Comput. Phys.* **213**, 184 (2006).
- [102] J.-H. Seo and R. Mittal, Improved swimming performance in schooling fish via leading-edge vortex enhancement, *Bioinspir. Biomim.* **17**, 066020 (2022).
- [103] H. Liu, Integrated modeling of insect flight: From morphology, kinematics to aerodynamics, *J. Comput. Phys.* **228**, 439 (2009).
- [104] G. Li, D. Kolomenskiy, H. Liu, B. Thiria, and R. Godoy-Diana, On the energetics and stability of a minimal fish school, *PLoS ONE* **14**, e0215265 (2019).
- [105] G. Li, U. K. Müller, J. L. van Leeuwen, and H. Liu, Body dynamics and hydrodynamics of swimming fish larvae: A computational study, *J. Exp. Biol.* **215**, 4015 (2012).
- [106] G. Li, D. Kolomenskiy, H. Liu, R. Godoy-Diana, and B. Thiria, Intermittent versus continuous swimming: An optimization tale, *Phys. Rev. Fluids* **8**, 013101 (2023).
- [107] C. Smith, N. Beratlis, E. Balaras, K. Squires, and M. Tsunoda, Numerical investigation of the flow over a golf ball in the subcritical and supercritical regimes, *Int. J. Heat Fluid Flow* **31**, 262 (2010).
- [108] N. Beratlis, K. Squires, and E. Balaras, An efficient drag reduction of bluff bodies by tessellation, in *12th International Symposium on Turbulence and Shear Flow Phenomena, TSFP 2022* (2022).
- [109] L. Xu, F.-B. Tian, J. Young, and J. C. Lai, A novel geometry-adaptive Cartesian grid based immersed boundary–lattice Boltzmann method for fluid–structure interactions at moderate and high Reynolds numbers, *J. Comput. Phys.* **375**, 22 (2018).
- [110] A. Boomsma and F. Sotiropoulos, Direct numerical simulation of sharkskin denticles in turbulent channel flow, *Phys. Fluids* **28**, 035106 (2016).
- [111] L. Ge and F. Sotiropoulos, A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries, *J. Comput. Phys.* **225**, 1782 (2007).
- [112] D. Bechert and W. Reif, On the drag reduction of the shark skin, in *23rd Aerospace Sciences Meeting* (1985), p. 546.
- [113] A. Scotti, M. G. Rubinat, and E. Balaras, Les of pulsating turbulent flows over smooth and wavy boundaries, in *Turbulence and Interactions: Proceedings of the TI 2009 Conference* (Springer, 2010), pp. 25–36.
- [114] M. Aghaei-Jouybari, J.-H. Seo, J. Yuan, R. Mittal, and C. Meneveau, Contributions to pressure drag in rough-wall turbulent flows: Insights from force partitioning, *Phys. Rev. Fluids* **7**, 084602 (2022).
- [115] D. R. Warrick, B. W. Tobalske, and D. R. Powers, Lift production in the hovering hummingbird, *Proc. R. Soc. B: Biol. Sci.* **276**, 3747 (2009).
- [116] J. Song, H. Luo, and T. L. Hedrick, Three-dimensional flow and lift characteristics of a hovering ruby-throated hummingbird, *J. R. Soc. Interface* **11**, 20140541 (2014).
- [117] H. Luo, H. Dai, P. J. F. de Sousa, and B. Yin, On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries, *Comput. Fluids* **56**, 61 (2012).
- [118] L. Zheng, T. L. Hedrick, and R. Mittal, A multi-fidelity modelling approach for evaluation and optimization of wing stroke aerodynamics in flapping flight, *J. Fluid Mech.* **721**, 118 (2013).
- [119] H. Dong, M. Bozkurtas, R. Mittal, P. Madden, and G. Lauder, Computational modelling and analysis of the hydrodynamics of a highly deformable fish pectoral fin, *J. Fluid Mech.* **645**, 345 (2010).
- [120] F.-B. Tian, H. Dai, H. Luo, J. F. Doyle, and B. Rousseau, Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems, *J. Comput. Phys.* **258**, 451 (2014).
- [121] W. Jiang, J. H. Rasmussen, Q. Xue, M. Ding, X. Zheng, and C. P. Elemans, High-fidelity continuum modeling predicts avian voiced sound production, *Proc. Natl. Acad. Sci. USA* **117**, 4718 (2020).

- [122] W. Jiang, C. Farbos de Luzan, X. Wang, L. Oren, S. M. Khosla, Q. Xue, and X. Zheng, Computational modeling of voice production using excised canine larynx, *J. Biomech. Eng.* **144**, 021003 (2022).
- [123] B. E. Griffith and N. A. Patankar, Immersed methods for fluid–structure interaction, *Annu. Rev. Fluid Mech.* **52**, 421 (2020).
- [124] J.-H. Seo, T. L. Hedrick, and R. Mittal, Mechanism and scaling of wing tone generation in mosquitoes, *Bioinspr. Biomim.* **15**, 016008 (2019).
- [125] J. E. Ffowcs Williams and D. L. Hawkings, Sound generation by turbulence and surfaces in arbitrary motion, *Philos. Trans. R. Soc.* **264**, 321 (1969).
- [126] B. J. Arthur, K. S. Emr, R. A. Wytenbach, and R. R. Hoy, Mosquito (*Aedes aegypti*) flight tones: Frequency, harmonicity, spherical spreading, and phase relationships, *J. Acoust. Soc. Am.* **135**, 933 (2014).
- [127] L. Feugère, J.-H. Seo, U. Ismail, G. Gibson, and R. Mittal, 3D audio-visual recordings of free-flying mosquito wings for aeroacoustic simulation, in *Forum Acusticum 2023, Politecnico di Torino, Italy, 11–15, September* (2023).
- [128] J. Abshagen, I. Schäfer, C. Will, and G. Pfister, Coherent flow noise beneath a flat plate in a water tunnel experiment, *J. Sound Vib.* **340**, 211 (2015).
- [129] W. Zhang, Y. Pan, J. Wang, V. Di Santo, G. V. Lauder, and H. Dong, An efficient tree-topological local mesh refinement on Cartesian grids for multiple moving objects in incompressible flow, *J. Comput. Phys.* **479**, 111983 (2023).
- [130] J. T. King, R. Kumar, and M. A. Green, Experimental observations of the three-dimensional wake structures and dynamics generated by a rigid, bioinspired pitching panel, *Phys. Rev. Fluids* **3**, 034701 (2018).
- [131] M. V. Krumis, A review of sphere drag coefficients applicable to atmospheric density sensing, Naval Ordnance Laboratory, **NASA-CR-126742** (1972).
- [132] M. D. de Tulio, P. De Palma, G. Laccarino, G. Pascasio, and M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, *J. Comput. Phys.* **225**, 2098 (2007).
- [133] J. W. Nam and F. S. Lien, Assessment of ghost-cell based cut-cell method for large-eddy simulations of compressible flow at high Reynolds number, *Int. J. Heat Fluid Flow* **53**, 1 (2015).
- [134] M. Al-Marouf and R. Samtaney, A versatile embedded boundary adaptive mesh method for compressible flow in complex geometry, *J. Comput. Phys.* **337**, 339 (2017).
- [135] C. Mao, T. Jin, and K. Luo, Investigation of supersonic turbulent flows over a sphere by fully resolved direct numerical simulation, *Phys. Fluids* **31**, 056102 (2019).
- [136] S. Sambasivan, A. Kapahi, and H. Udaykumar, Simulation of high speed impact, penetration and fragmentation problems on locally refined Cartesian grids, *J. Comput. Phys.* **235**, 334 (2013).
- [137] A. Kapahi, S. Sambasivan, and H. Udaykumar, A three-dimensional sharp interface Cartesian grid method for solving high speed multi-material impact, penetration and fragmentation problems, *J. Comput. Phys.* **241**, 308 (2013).
- [138] M. N. Linnick and A. H. F. Fasel, High-order immersed interface method for simulating unsteady incompressible flows on irregular domains, *J. Comput. Phys.* **204**, 157 (2005).
- [139] Y.-F. Peng, R. Mittal, A. Sau, and R. R. Hwang, Nested Cartesian grid method in incompressible viscous fluid flow, *J. Comput. Phys.* **229**, 7072 (2010).
- [140] M. Vanella, P. Rabenold, and E. Balaras, A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid–structure interaction problems, *J. Comput. Phys.* **229**, 6427 (2010).
- [141] F. Capuano, N. Beratlis, F. Zhang, Y. Peet, K. Squires, and E. Balaras, Cost vs accuracy: DNS of turbulent flow over a sphere using structured immersed-boundary, unstructured finite-volume, and spectral-element methods, *Eur. J. Mech. B: Fluids* **102**, 91 (2023).
- [142] J. Kim, P. Moin, and R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, *J. Fluid Mech.* **177**, 133 (1987).
- [143] O. Lehmkuhl, G. Park, S. Bose, and P. Moin, *Large-eddy simulation of practical aeronautical flows at stall conditions*, in *Proceedings of 2018 Center for Turbulence Research Center Summer Program*, pp. 87–96.
- [144] J. Jiménez, Turbulent flows over rough walls, *Annu. Rev. Fluid Mech.* **36**, 173 (2004).

- [145] H.-J. Kaltenbach, M. Fatica, R. Mittal, T. Lund, and P. Moin, Study of flow in a planar asymmetric diffuser using large-eddy simulation, [J. Fluid Mech.](#) **390**, 151 (1999).
- [146] W. Wu, C. Meneveau, and R. Mittal, Spatio-temporal dynamics of turbulent separation bubbles, [J. Fluid Mech.](#) **883**, A45 (2020).
- [147] J. Sadique, X. I. Yang, C. Meneveau, and R. Mittal, Aerodynamic properties of rough surfaces with high aspect-ratio roughness elements: Effect of aspect ratio and arrangements, [Boundary-Layer Meteorol.](#) **163**, 203 (2017).
- [148] S. Ghosh, J.-I. Choi, and J. Edwards, RANS and hybrid LES/RANS simulation of the effects of micro vortex generators using immersed boundary methods, in *38th Fluid Dynamics Conference and Exhibit* (2008), p. 3728.
- [149] F. Cadieux, J. Sadique, X. I. Yang, C. Meneveau, and R. Mittal, Wall-modeled large eddy simulation of laminar and turbulent separation bubble flows, in *46th AIAA Fluid Dynamics Conference* (2016), p. 3189.
- [150] M. Tyagi and S. Acharya, Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method, [Int. J. Numer. Methods Fluids](#) **48**, 691 (2005).
- [151] R. Mittal and R. Bhardwaj, Immersed boundary methods for thermofluids problems, [Annu. Rev. Heat Transfer](#) **24**, 33 (2021).