1
Introduction

1.1 Motivation

Computational science is a relatively new, interdisciplinary approach to the study of natural and technological complex systems that combines elements from different scientific fields and relies on computer-based models.

The quote "The computer is incredibly fast, accurate, and stupid. Man is incredibly slow, inaccurate, and brilliant. The marriage of the two is a force beyond calculation" has been attributed, with several variations, to many different scientists although Garland (1982) reports a specific event in which Leo Cherne wrote it in 1977: Regardless the original author, on account of the five-decade unrestrained growing power of computers, it sounds like a prophecy that has now come true.

In fact, alongside theory and laboratory experiments, numerical simulations have emerged as an essential tool for advancing science and technology with computer models that are routinely used for the analysis of complex systems.

Numerical computations are essential for the design and optimization of not only everyday products like cars, aircraft, buildings and bridges, but also innovative biomedical and microelectronic devices. Computational science is at the core of meteorology, oceanography, climatology and even of epidemiology (Smolinski et al., 2003) or dynamics of crowds and populations (Curtis et al., 2016).

The common feature of all these disciplines is the use of complex computational models to predict the system dynamics. These models can be fully exploited only by implementing dedicated numerical methods on powerful computers.

The main benefits of computation over other approaches are that system dynamics can be predicted in advance, hazardous scenarios can be safely explored, performances can be assessed and improved without employing hardware mod-

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

els, and interactions among various systems can be reproduced and studied in a controlled environment.

One of the branches of computational science that has greatly benefited from the availability of powerful computers is computational fluid dynamics (CFD). Traditionally, it has been the most demanding in terms of computational power.

The great scientist J. von Neumann was prophetic in this respect when in 1940 he wrote: "Our present analytical methods seem unsuitable for the solution of important problems arising in connection with non-linear partial differential equations ... The truth of this statement is particularly striking in the field of fluid dynamics" (Goldstine, 1980). In fact, the governing equations of fluid dynamics are quadratically nonlinear and strongly coupled, making analytical solutions unattainable in realistic conditions. This explains why using simplified models or empirical correlations limits the overall reliability of the results. By contrast, in sectors such as aeronautical or automotive engineering, where efforts have been made to develop and apply high-fidelity CFD models, the progress has been very rapid and this has prompted other branches of science to pursue a similar approach.

A growing number of engineers and scientists nowadays rely on high-fidelity CFD tools for the design and analysis of new technological solutions or complex systems. However, in spite of its widespread use, there are still difficulties that prevent CFD from being used as a standard tool. Among many, accurate predictions of highly turbulent flows, parametric analyses of complex systems and design optimization are examples of critical applications.

As an example, we can consider the flow around an object, such as the hull in Figure 1.1, having a complex geometry. As we will detail in Chapter 2, owing to the nonlinear nature of the governing relations, the only way to obtain a solution is to integrate the relations by a numerical technique, which requires a discretization procedure aimed at transforming the original set of differential equations into a discretized algebraic system. This implies that the entire domain of interest is tessellated by small elements, the so-called computational mesh, whose "distribution" and "quality" will determine the reliability of the numerical solution.

Usually starting from some geometrical description of the boundaries (bodies and external domain), a surface grid is first constructed to be used as a boundary condition for the generation of a volume mesh covering the whole fluid domain. Building the initial surface grid can be extremely difficult and time-consuming for realistic, complex boundaries, but the problem is further exacerbated when there are multiple moving elements (see Figure 1.1).

Although different techniques (structured or unstructured grids, block or overlapping meshes, see Thompson et al. (1999) for more details) are available

for mesh generation, their common objective is to produce a *regular* discretization upon which the quality of the flow solution heavily relies. Regularity in this context is intended in general terms, for example, cell size and shape, element orthogonality, smooth variation of the spacing, etc.

In general, generating a high-quality grid, fitted to a complex three-dimensional object (as in Figure 1.1), can become difficult enough that only highly trained and specialized researchers accomplish it. Furthermore, the procedure can be so time-consuming that it can easily exceed the time needed to obtain the flow solution. Usually, the resulting meshes are typically not guaranteed to be orthogonal, and this requires more complex solution algorithms with significant overhead in the per-cell operation count and, often, with high computational cost and limited precision (Chung, 2002).

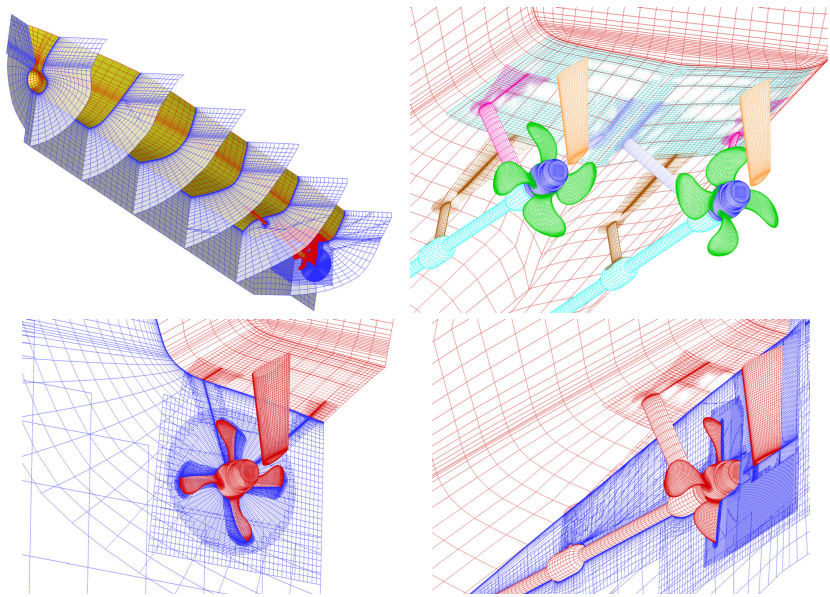

Figure 1.1 Example of body-fitted overset grids around a hull with appendices and impellers. Courtesy of Professor A. Di Mascio, reproduced with permission.

In some applications, the boundaries are not even stationary and when there are multiple parts whose relative position varies, simply changing the reference frame is not enough to compensate for the motion. This entails additional complexity since solution methods that rely on body-conformal discretizations require complex algorithms such as sliding surfaces, overlapping grids and deformation or regeneration of the mesh. Obviously, all these procedures need

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

interpolations of the solution, thus resulting in a degradation of accuracy and conservation properties, in addition to the considerable overhead of computational time.

In this context, it is clear that a numerical procedure capable of handling complex geometric configurations without resorting to body-fitted meshes would contribute significantly to promoting the application of CFD to complex problems: The **immersed boundary method** (IBM) has emerged in the past two decades as a suitable and versatile candidate.

1.2 Background

The key idea of the immersed boundary (IB) approach is to replace the boundaries by a time–space varying distribution of force densities mimicking the effect of the body on the flow. The main advantage of this approach is that the forces can be prescribed on simple and regular meshes, thus retaining all the ease and efficiency of the numerical methods developed for these discretizations.

It will be shown that there is a wide variety of possible forcing terms and an even wider assortment of implementations, thus yielding a multitude of different IBMs, each one with advantages and drawbacks. The aim of this book is to guide the reader through the basics of the methodology and to the implementation of some of the main techniques; the latest developments are left to the scientific papers whose number has grown continuously since the beginning of the 2000s.

The immersed boundary technique allows the simulation, using simple discretizations, of flows in complex geometries by forcing the governing equations along surfaces corresponding to the physical location of the boundaries; the computation can then be performed on a much simpler mesh with fewer computational resources. The essence of the method is the decoupling of the geometry of the fluid domain and the grid for the flow solution. In more detail, if the flow around object $\mathbf{B}$ in Figure 1.2 has to be computed in the fluid domain Γ, a standard CFD approach would require the generation of a grid in the region bounded by $\partial\mathbf{B}$ and $\partial\Gamma$; a possible standard configuration is that of Figure 1.2a. In IBM, in contrast, an underlying grid is laid in the inner region of $\partial\Gamma$ regardless of the shape of the object $\mathbf{B}$, which is *immersed* into the computational domain. The effect of the body on the flow is enforced by prescribing a suitable distribution of fictitious terms (forces, velocities or generalized constraints) such that the correct boundary conditions on $\partial\mathbf{B}$ are obtained for the flow.

Of course, in the IB context $\partial\mathbf{B}$ is not a coordinate line (see Figure 1.3), so the computational cells must be classified according to their position relative to the

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

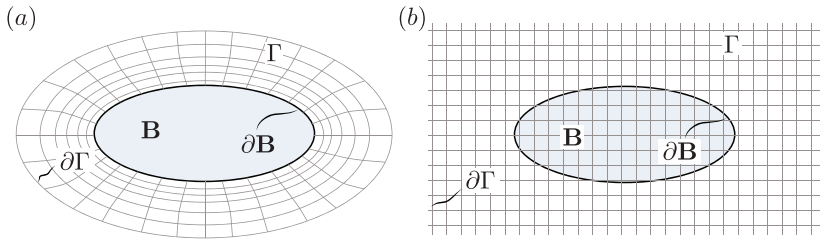

Figure 1.2 Sketch of the domain discretization with a body-conformal mesh (a) and with the immersed boundary method (b).

body ("internal," "external" and "interface"); and while ad hoc strategies can be devised for simple objects, for general complex geometries the task can become computationally intensive. Another difficulty is that the spatial position of the discrete unknowns generally does not coincide with the body surface where the former are known through the boundary conditions. This introduces the additional challenge of forcing the equations at some nodes to impose a solution at a different position, which involves appropriate interpolation techniques.

These problems become more severe when the object is moving or, even worse, deforming with respect to the grid, since the steps of cell tagging and interpolation/reconstruction at the IB must be repeated at every time step. In addition, when a body moves across the grid, there are cell nodes whose state changes in one time step from "internal" to "interface" and they miss all the necessary flow history to compute its evolution.

Each of the above difficulties has been tackled by many different research groups worldwide and many ingenious numerical algorithms have been made available through scientific publications. If we recall that, in fact, the body is not physically present in the domain except for some layers of forced computational nodes, we could simplify the essence of the technique by stating that immersed boundary methods consist of the "art" of imposing boundary conditions in the middle of the computational domain; though at first glance the principle might seem hopeless, it is in fact a very effective strategy and showing the advantages and weaknesses of these methods is the main aim of this book.

1.3 Literature and History

The idea of using non-body-conformal meshes in computational fluid dynamics is definitely not new and it has been pursued by many researchers at least since the 1950s. The *father* of immersed boundaries is unanimously considered to

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

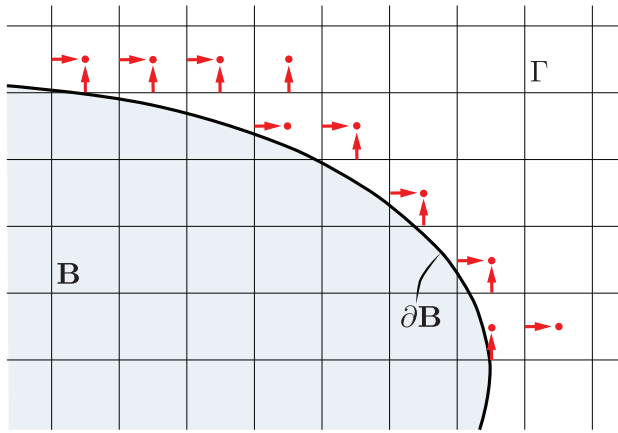

Figure 1.3 Detail of the intersection between an immersed boundary and the grid; the red arrows and bullets indicate possible positions of the discretized unknowns (on a staggered mesh) with respect to the boundary.

be Charles Peskin who, in his seminal paper (Peskin, 1972a), showed the huge potential of the method by simulating the two-dimensional flow through the natural mitral heart valve. The basic concepts, however, had already been around for a while and, to the authors' knowledge, the first published IB example was given by Gentry et al. (1966) for unsteady compressible flows. In this paper, the authors note that the governing "equations for those cells adjacent to the bounding surface of a rigid body are modified to insure that there is no flow of mass or energy across the boundary." The backbone of the proposed numerical scheme, however, was the "fluid-in-cell" method (Rich, 1963) in turn relying on the work of Evans and Harlow (1957): Already in these reports some techniques for computing compressible flows around obstacles without using body-fitted meshes were described in detail and numerical calculations on two-dimensional Cartesian meshes with up to a few thousand nodes were shown. We report here, as a historical note, some sketches taken from those reports (Figure 1.4) where it can be appreciated how the same concepts as those in Figure 1.3 had already been put forward in the middle of the 1950s. As evident from Figure 1.4c, not only boundaries aligned with the Cartesian mesh were considered, but also surfaces that could intersect the coordinate lines with arbitrary position and orientation.

Incompressible flows were tackled using a similar approach (see Welch et al., 1965) and in the paper by Harlow and Whelch (1971) the authors presented the marker and cell (MAC) method in which two-dimensional flows

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

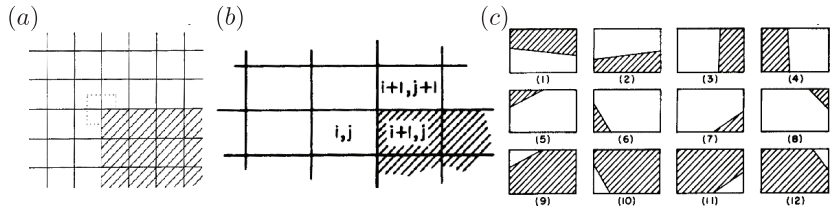

Figure 1.4 Schematics of boundary–mesh intersection extracted from: (a) Evans and Harlow (1957), p. 26; (b) Rich (1963), p. 36; (c) Rich (1963), p. 38.

with free surface and obstacles were computed. In the simulations, the Eulerian (Cartesian) grid was not fitted to the obstacles and the nodes of the mesh were tagged as fluid or solid to impose the appropriate velocity boundary conditions through finite differences. Although the main idea of an immersed boundary was already there, the authors stated that "walls are restricted in orientation so that they lie along the boundaries of the Eulerian cells. Relaxation of this requirement could be accomplished only at the expense of a considerable increase in complication."

The above limitation was removed a few years later by Viecelli (1969), who extended the MAC method to include boundaries of arbitrary shape. The basic idea consisted of treating the fluid–boundary interface as a free surface and imposing there pressure boundary conditions so that particles could move only tangentially to the surface. This procedure led to an iteration between pressure and velocity fields until flow incompressibility and boundary impermeability were both satisfied. The method, referred to as ABMAC (arbitrary boundary MAC), was also generalized in a follow-up paper (Viecelli, 1971) for handling moving walls and, in this case, velocity boundary conditions, in addition to pressure, were imposed at the interface. This technique allowed the treatment of walls moving with a prescribed law or as a consequence of the forces exerted by the fluid on the surface. Early attempts to solve partial differential equations on non-body-conformal meshes can also be found in the scientific literature of countries in the former Soviet bloc. The diffusion equation $\nabla \cdot (k \nabla \Phi) = f$, with k a position-dependent diffusivity of the scalar field Φ, was solved by Saulev (1963) on a nonfitted mesh by introducing a modified diffusivity $k_c = k(1 + 1/\epsilon)$ with $\epsilon \to \infty$ in the fluid domain and $\epsilon \to 0$ inside the body. Rigorous proof was provided that the resulting solution Φ_c approached the correct one Φ in the limiting behavior. Kuznetsov (2001) later showed that approach to be equivalent to introducing a Lagrangian multiplier $\lambda = \Phi/\epsilon$ in the region inside the body and letting $\epsilon \to 0$: This is essentially the distributed

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

Lagrange multiplier fictitious domain method proposed by Dihn et al. (1992) thirty years later.

Yanenko (1967) formulated a method, similar to that of Gentry et al. (1966), for solving fluid dynamics equations on non-body-conformal meshes referred to as the "unmatched or incoherent grids" (Figure 1.5b); after explaining how to use a fractional-step method to solve flow problems on body-fitted grids as in Figure 1.5a, he explicitly states that "for incoherent grids additional interpolations are needed in order to impose boundary conditions at the walls."

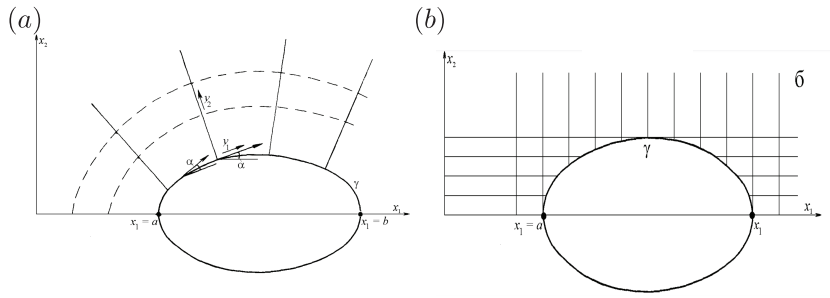

Figure 1.5 Representation of "matched" (a) and "unmatched" (b) grids for the flow around a curvilinear object as represented by Yanenko (1967) on p. 130 and p. 137 of his book.

At the beginning of the 1970s, Peskin (1972a) reported simulations of the blood flow in the heart system assuming low Reynolds number and two-dimensional dynamics. The approach was innovative since the incompressible Navier–Stokes equations were solved on a Cartesian grid but the boundaries were discretized by Lagrangian markers connected by elastic elements *immersed* in the flow: Fluid and boundaries exerted forces on each other thus determining the flow and the structure dynamics. The Lagrangian markers moved with the local fluid velocity and their location was tracked in space; the position of the markers and their forcing on the fluid were then transferred to the Eulerian mesh, where the flow solution was obtained. The method was extended in Peskin (1972b) to mimic the active contraction of the heart muscular fibers, relying on a clever set of links among the Lagrangian markers; using a mesh discretized by 64^2 nodes (and low Reynolds number), the basic left-heart flow features were reproduced numerically.

Benefiting also from a continuous growth of available computational power, the procedure was further improved by Peskin and McQueen (1989) and by McQueen and Peskin (1989), who were able to simulate the three-dimensional blood flow in the heart on a 128^3 cubic lattice using realistic contractile fibers.

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

Although the method was originally developed to simulate flows interacting with elastic boundaries, in principle it could also be used for rigid objects once the links connecting the Lagrangian markers were made stiff enough to prevent the structure from deforming. Unfortunately, this strategy is inefficient as it leads to stiffer equations that require significant computational resources for integration; we will come back to this point in the later chapters when various IB implementations will be illustrated in detail.

A different IB approach for flows around solid, rigid objects was proposed by Basdevant and Sadourny (1983) for the incompressible Navier–Stokes equations in vorticity–streamfunction formulation and later extended by Briscolini and Santangelo (1989) to primitive variables (velocity and pressure). These authors note that the procedure "treats the no-slip condition of the Navier–Stokes equations as a forcing term acting, at each time step, on the physical boundary ∂B; the forcing depends on the velocity field." The numerical scheme was referred to as the "mask method" since it consisted essentially of deleting the field inside the body by a mask, and smoothly connecting it to the external field.

They computed the unsteady two-dimensional flow around circular and square cylinders at Reynolds numbers up to 1000, obtaining a good agreement with the results from the literature.

Rather than altering the velocity field inside the body, Goldstein et al. (1993) exploited the body force term of the Navier–Stokes equations; "forces were chosen to lie along a desired surface and to have a magnitude and a direction opposing the local flow such that the flow was brought to rest on an element of the surface." This resulted in an easy implementation of the method that was flexible enough to allow the simulation of the three-dimensional plane- and ribbed-turbulent channel flow.

The main drawback of the method of Goldstein et al. (1993) is that it contains free parameters requiring a problem-dependent tuning; in particular, for unsteady flows this forcing introduces a time step limitation which reduces the efficiency and the applicability of the technique (see the following chapters for a detailed description).

Soon after, Saiki and Biringen (1996) found that the body forces proposed in Goldstein et al. (1993) were more efficient when used in combination with finite-difference discretizations than with spectral methods. The reason was that the immersed boundaries forcing terms, localized at the fluid–boundary interface, tend to generate unphysical oscillations when described by nonlocal spectral operators (Gibbs phenomenon).

Consistent results were obtained by Elghaoui and Pasquetti (1996) that applied a spectral embedding method to a convection diffusion equation. They used Fourier expansions for the latter and a boundary integral equation to

enforce the boundary conditions on a hexagonal line in a two-dimensional plane. However, because of the discontinuities introduced by the boundaries, the convergence rate of the method was not fully satisfactory and the authors concluded that the accuracy of the solution in the vicinity of the boundaries needed to be improved.

Cortez (2000) and Cortez and Minion (2000) proposed a variant of the Peskin approach in which the momentum equation was forced through a series of regularized vortex dipoles (referred to as vortex blobs) that were shown to be more accurate than standard forcings. The numerical examples, however, were limited to two-dimensional flows at relatively low Reynolds numbers.

Despite the increasing number of papers dealing with or applying immersed boundary techniques, until the late 1990s, the scenario was essentially dominated by the method developed by Peskin (1972a); the review by Peskin (2002) is an exhaustive reference for the state of the art at the time.

An important breakthrough was obtained by Mohd-Yusof (1997) who derived an alternative formulation of the forcing based on the time-discrete version of the Navier–Stokes equations that neither requires user-defined parameters nor affects the stability of the time integration; these are highly desirable features in order to make the approach flow independent and the simulation of complex three-dimensional flows feasible. Mohd-Yusof combined the new method with B-splines to compute the laminar flow over a three-dimensional ribbed channel, showing substantial improvements with respect to the earlier formulations.

A forcing based on the idea of Mohd-Yusof (1997) was developed by Fadlun et al. (2000) using finite differences and extended in Verzicco et al. (2000) to large-eddy simulation; in both cases, second-order accuracy was preserved in the IB context. The applications included the flow around simple and complicated geometries for a wide range of Reynolds numbers.

The review papers by Iaccarino and Verzicco (2003) and Mittal and Iaccarino (2005) provide additional references and a comprehensive discussion of almost all methods developed until that period, and the advantages and drawbacks of the various techniques are illustrated through specific numerical examples and applications. Since then, however, the number of IB papers has literally exploded and accounting for all possible developments and applications has become practically impossible. In Figure 1.6, we report the number of papers per year published in the period 1970–2020 and containing the words "immersed boundary" in the title; it can be seen that while initially these publications were only sporadic, since 2000 they have constantly grown at a rate that has been sustained. It is also worthwhile noting that Figure 1.6 does not account for all the literature on the subject since expressions like "virtual boundary," "immersed surface," "boundary body forces," "mask method" and similar have not been included in the search.

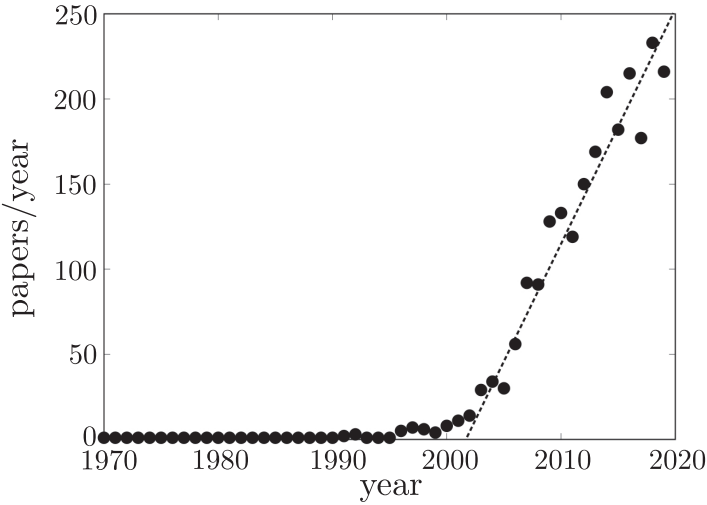

Figure 1.6 Number of published papers per year in the period 1970–2020 and containing in the title the words "immersed boundary" or "immersed boundaries" in all fields. The fit in the range 2000–2020 indicates a growth with a constant acceleration of 10 papers/year^2 since the year 2000. The dashed line is the fit papers=5*(year-2000)^2. Source: Scopus, December 2022.

Even if we are conscious that a comprehensive literature review covering the past two decades is not possible, we wish to mention those contributions that have fostered innovative research in IBMs.

The Physalis method (Prosperetti and Oğuz, 2001; Zhang and Prosperetti, 2005) is somehow unique in the field of IB since it can simulate flows with spherical particles using Cartesian grids, although it cannot be easily generalized to different complex geometries. In fact, the main idea is that, owing to the no-slip condition, in the reference frame of each sphere the near-wall velocity is so small that the Stokes equations are an excellent approximation to the full Navier–Stokes system, and this is true regardless of the large-scale Reynolds number of the external flow. Since the Stokes flow around a sphere is known analytically, any information can be transferred to the surrounding Eulerian points without approximations, and the forces can also be computed accurately. However, the analytical solution for the Stokes problem is not known for a generic complex geometry and the method loses all advantages.

In the context of incompressible flows, Kim et al. (2001) observed that using IB forces in combination with a fractional-step integration scheme (Kim and Moin, 1985) could yield a velocity field that was locally non-solenoidal. They solved the problem by also forcing the continuity equation by source/sink

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press

terms at the cells crossed by the immersed boundary, thus eliminating the mass conservation problem. The proposed strategy was easy to implement and was quickly adopted by many research groups.

Tseng and Ferziger (2003) introduced ghost cells at the immersed interface to add flexibility to the imposition of boundary conditions; it was shown that while retaining the second-order accuracy, it was possible to impose Dirichlet, Neumann and Robin velocity boundary conditions.

All the previously described implementations of IBMs were based on flow solvers that relied on finite differences, finite volumes or spectral spatial discretizations. In Feng and Michaelides (2004) a lattice–Boltzmann method was instead coupled with the IB forcing of Peskin (1972a) in order to combine the most advantageous features of the two techniques. The resulting method could accurately reproduce the sedimentation of two circular particles in a quiescent fluid and the same phenomenon for more than 500 particles: All the cases were simulated in two dimensions and for low values of the Reynolds number.

Although the IB community had always considered the progeny of Peskin's methods (Peskin, 2002) and those derived by Mohd-Yusof (1997) as alternatives, Uhlmann (2005) showed that it was possible to combine the two to obtain a powerful scheme that allowed a smooth transfer between Eulerian and Lagrangian meshes (thus giving regular load distributions) while at the same time avoiding strong restrictions on the time step. The resulting method was efficient and accurate enough to allow the simulation of $O(1000)$ fully resolved spherical particles settling in a fluid at a particle Reynolds number of a few hundred, which was a landmark achievement in the field.

One of the applications in which IBMs really show all their potential is when multiple bodies are in relative motion, since this can be accounted for without deforming or regenerating the grid. Gilmanov and Sotiropoulos (2005) and Yang and Balaras (2006), however, noted that when a body moves, some computational nodes that at one time step are inside the body can become fluid points at the next step and they do not have a valid "fluid history" to compute all the terms of the Navier–Stokes equations. The problem was solved by a field extension procedure in which all the missing information was obtained by extrapolation from the closest fluid points.

Using finite-element approximations, Glowinski et al. (1998) resorted to distributed Lagrange multipliers to impose boundary conditions, within the fictitious domain method, to rigid bodies moving along a prescribed trajectory in an incompressible viscous fluid. Taira and Colonius (2007) observed that pressure also acts like a Lagrange multiplier to ensure incompressibility, and so the distributed IB forcing terms can be handled in a unified way in a projection method like the fractional step of Kim and Moin (1985). The method was shown

to predict correctly some two-dimensional canonical flows without resorting to the additional forcing of the continuity equation of Kim et al. (2001) or to iterations between momentum forcing and pressure correction as suggested by Fadlun et al. (2000).

A versatile and powerful tool was developed by Ge and Sotiropoulos (2007) that combined a curvilinear-mesh body-fitted flow solver with a sharp interface IBM. This idea is very interesting since the curvilinear grid can be used to resolve the near-wall regions of a complex domain, while IBs can deal with small geometrical details or moving/deforming objects. The method was, in fact, used to simulate the flow through an aortic root with a mechanical heart valve in which the leaflet motion was determined by the hydrodynamic loads. The problem was studied further in a subsequent paper of Borazjani et al. (2008) that provided a theoretical framework for the application of IBMs to fluid–structure interaction problems.

A lot of effort in developing IBMs is made in reconstructing the solution at the immersed interface and in transferring information from the Eulerian mesh to the boundary element (or its Lagrangian counterpart) and vice versa. Starting from the ideas of Uhlmann (2005), Vanella and Balaras (2009) developed a moving least squares procedure to build the transfer function between the grid and the boundary. Among the advantages, there is the possibility to impose a variety of different boundary conditions and to compute analytically the solution derivatives at the wall through the basis functions. This method has been widely adopted in IB applications owing to its second-order accuracy and the smooth distribution of surface forces, which is mandatory when simulating deformable boundaries.

Although the vast majority of IBMs have been developed for and applied to incompressible flows, there have been attempts to employ the technique to compressible flows also. For example, de Tullio et al. (2007) combined IBMs with a local grid refinement procedure to simulate supersonic flows around spheres and transonic flows around airfoils, showing the full second-order accuracy.

Before concluding this literature review, we wish to point out that the techniques we have described can all be gathered under the generic name of immersed boundary methods, which are the main object of the present book. There are other procedures, however, that rely on a similar philosophy but differ in technical aspects. For the sake of brevity, we will not discuss such approaches in detail, although we now mention the main references in order to provide the interested reader with additional material.

A class of methods, referred to as "penalty methods" or "fictitious domain methods" or "domain embedding methods," considers the immersed body as a

porous medium and solves the Navier–Stokes–Brinkman equations; these are the Navier–Stokes equations with the addition of a volumetric loss term, called Darcy drag, which accounts for the action of the porous medium on the flow. When the permeability of the material tends to zero, it behaves as a solid and the flow around an arbitrary shaped object can be simulated by prescribing this velocity penalization term in space. A detailed description of the method can be found in Heinrich and Vionnet (1995), Angot et al. (1999) and Khandra et al. (2000) and references therein. Penalty methods could also be considered, from a mathematical viewpoint, as a particular case of the IB forcing proposed by Goldstein et al. (1993) and Saiki and Biringen (1996), although the physical interpretation of the results is different. This point will be further discussed in the later chapters when we deal with the details of IB implementations.

A different technique, called the "Cartesian grid method," also relies on non-body-fitted meshes to describe the flow around complex geometries. In this case, however, instead of adding forcing terms to the governing equations, the grid cells are modified at the body interface according to their intersections with the underlying grid. Given the large number of possible intersections, this generates a wide variety of "cut interface cells," which must be handled in a separate way depending on their topology. Finite-volume methods have been successfully used for the calculation of complex three-dimensional flows, although most of the developments concerned the Euler equations (Epstein et al., 1992; Pember et al., 1995; Almgren et al., 1997). Similarly to IBMs, in this case also, the transfer of information from the embedded surface to the underlying Cartesian grid needs special treatments to satisfy the conservation laws at the irregularly shaped boundary cells (Cartesian mesh cells cut by the interface). Various cell-cut algorithms have been developed by Forrer and Jeltsch (1998) and Aftosmis et al. (2000), while accuracy assessments have been carried out by Coirier (1994) for the compressible Navier–Stokes equations and by Almgren et al. (1997) for the Poisson equations; full second-order accuracy was obtained considering the cells not intersected by the boundaries.

Finally, an entire class of numerical techniques has been developed with the objective of handling moving interfaces (of various physical natures) on a fixed grid. Two of these are the volume-of-fluid (VOF) and the level-set methods (Hirt and Nichols, 1981; Brackbill et al., 1992; Sussman and Puckett, 2000). The main application of these techniques is the tracking of interfaces between immiscible fluids (Popinet, 2003; Tryggvason et al., 2011) or the propagation of flames in combustion (Pitsch and Duchamp de Lageneste, 2002). Nevertheless, the boundary between a fluid and a solid can also be thought of as an interface and so the aforementioned techniques can be used in a context similar to the IBMs (Lorstadt et al., 2004; Cottet and Maitre, 2004).

1.4 Objectives and Plan

This book aims to provide the reader with the main concepts of the immersed boundary methods with a particular focus on their application to direct numerical simulation of complex flows. In Chapters 2–9 the required tools will be explained and discussed: governing equations, forcing methods, boundary conditions and reconstruction, moving/deforming boundaries and fluid–structure interaction. Some basic concepts of computational geometry will also be given in order to efficiently handle complex three-dimensional geometries in the IB context.

The final chapter (Chapter 10) of this book describes some numerical examples of IBMs applied to flows of increasing complexity and suggestions for further developments are also provided.

Some of the applications can also be viewed as a tutorial to get familiar with the use of the IB flow solver available with this book.

Finally, it is worthwhile pointing out that, although we have tried to give a comprehensive overview of the subject, the reader must be aware that the field is evolving so rapidly that up-to-date material can only be found in the current issues of archival journals.

https://doi.org/10.1017/9781009128438.002 Published online by Cambridge University Press