# REVIEW ARTICLE

# Numerical Methods for Fluid-Structure Interaction — A Review

Gene Hou[1], Jin Wang[2,*] and Anita Layton[3]

[1] *Department of Mechanical and Aerospace Engineering, Old Dominion University, Norfolk, VA 23529, USA.*
[2] *Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, USA.*
[3] *Department of Mathematics, Duke University, Durham, NC 27708, USA.*

**Abstract.** The interactions between incompressible fluid flows and immersed structures are nonlinear multi-physics phenomena that have applications to a wide range of scientific and engineering disciplines. In this article, we review representative numerical methods based on conforming and non-conforming meshes that are currently available for computing fluid-structure interaction problems, with an emphasis on some of the recent developments in the field. A goal is to categorize the selected methods and assess their accuracy and efficiency. We discuss challenges faced by researchers in this field, and we emphasize the importance of interdisciplinary effort for advancing the study in fluid-structure interactions.

# Contents

*Corresponding author. Email addresses:* `ghou@odu.edu` (G. Hou), `j3wang@odu.edu` (J. Wang), `alayton@math.duke.edu` (A. Layton)

# 1 Introduction

In fluid-structure interaction (**FSI**) problems, one or more solid structures interact with an internal or surrounding fluid flow. FSI problems play prominent roles in many scientific and engineering fields, yet a comprehensive study of such problems remains a challenge due to their strong nonlinearity and multidisciplinary nature (Chakrabarti 2005, Dowell and Hall 2001, Morand and Ohayon 1995). For most FSI problems, analytical solutions to the model equations are impossible to obtain, whereas laboratory experiments are limited in scope; thus to investigate the fundamental physics involved in the complex interaction between fluids and solids, numerical simulations may be employed.

With recent advances of computer technology, simulations of scientific and engineering systems have become increasingly sophisticated and complicated. For example, the speed requirement of a planing boat hull has advanced to such a degree and with such a speed that has outpaced the availability of testing data and existing design equations (Weymouth et al. 2006, 2008). To fill the technological gap, an efficient numerical algorithm can be used to investigate in detail the interaction between water waves and the motion of the boat. Such an investigation is typically multidisciplinary. In this example, the performance of the boat is a result of the interaction between water hydrodynamics and structural dynamics. Other FSI applications include, but are not limited to, sedimentation (Mucha et al. 2004, Tornberg and Shelley 2004, Wang and Layton 2009), particle assembly (Liu et al. 2006), aerodynamics (Haase 2001, Zhang, Jiang and Ye 2007), turbulence (Kaligzin and Iaccarino 2003, Yang and Balaras 2006), complex flows in irregular domains (Fadlun et al. 2000, Udaykumar et al. 1996, 2001), electro-hydrodynamics (Hoburg and Melcher 1976), magneto-hydrodynamic flows (Grigoriadis et al. 2009), biofluid and bio-mechanics (such as cell aggregation and deformation, blood-heart interaction, inner ear fluid dynamics, jellyfish swimming, sperm motility, cilliary beating, etc.).

The numerical procedures to solve these FSI problems may be broadly classified into two approaches: the *monolithic approach* and the *partitioned approach*. It is understood that the distinction between the monolithic and partitioned approaches may be viewed differently by researchers from different fields. In this paper, we intend to define these two approaches from the engineering application point of view. Fig. 1 illustrates the solution procedures of the monolithic and partitioned approaches.

The monolithic approach (Hubner et al. 2004, Michler et al. 2004, Ryzhakov et al. 2010) treats the fluid and structure dynamics in the same mathematical framework to form a single system equation for the entire problem, which is solved simultaneously by a unified algorithm. The interfacial conditions are implicit in the solution procedure. This approach can potentially achieve better accuracy for a multidisciplinary problem, but it may require substantially more resources and expertise to develop and maintain such a specialized code. In contrast, the partitioned approach treats the fluid and the structure as two computational fields which can be solved separately with their respective mesh discretization and numerical algorithm. The interfacial conditions are used explicitly to communicate information between the fluid and structure solutions. A motivation of
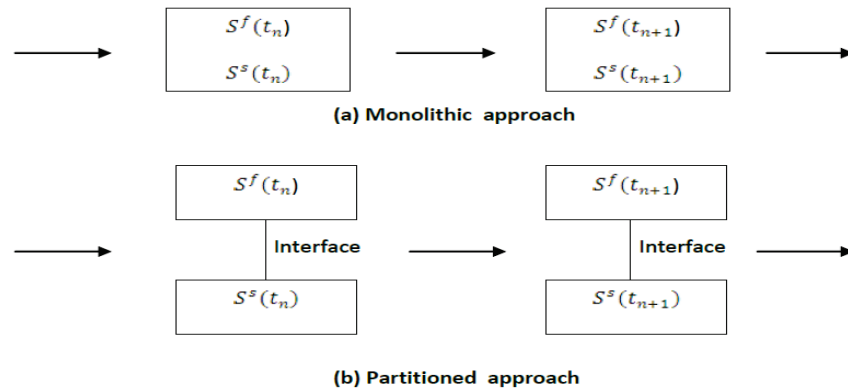
Figure 1: Schematic of the monolithic approach (a) and the partitioned approach (b) for fluid-structure interactions, where $S^f$ and $S^s$ denote the fluid and structure solutions, respectively.

the later approach is to integrate available disciplinary (i.e., fluidic and structural) algorithms and reduce the code development time by taking advantage of the "legacy" codes or numerical algorithms that have been validated and used for solving many complicated fluid or structural problems. As a result, a successful partitioned method can solve a FSI problem with sophisticated fluid and structural physics. The challenge of this approach is, however, to coordinate the disciplinary algorithms to achieve accurate and efficient fluid-structure interaction solution with minimal code modification. Particularly, the interface location that divides the fluid and the structure domains is not known a priori and usually changed in time; thus, the partitioned approach requires the tracking of the new interface location and its related quantities, which can be cumbersome and error-prone.

Another general classification of the FSI solution procedures is based upon the treatment of meshes: the *conforming mesh methods* and *non-conforming mesh methods*. The conforming mesh methods consider the interface conditions as physical boundary conditions, which treat the interface location as part of the solution, and requires meshes that conform to the interface. Owing to the movement and/or deformation of the solid structure, re-meshing (or mesh-updating) is needed as the solution is advanced. On the other hand, the non-conforming mesh methods treat the boundary location and the related interface conditions as constraints imposed on the model equations so that non-conforming meshes can be employed. As a result, the fluid and solid equations can be conveniently solved independently from each other with their respective grids, and re-meshing is not necessary. The distinction between these two types of meshes can be observed in Fig. 2, where a solid body (a sphere) is moving in a fluid domain. Most of the partitioned approach-based numerical works reviewed in this article are the conforming mesh methods (see Section 3), whereas the immersed methods that perhaps represent most of the recent developments in FSI methods are the non-conforming mesh methods (see Section 4).
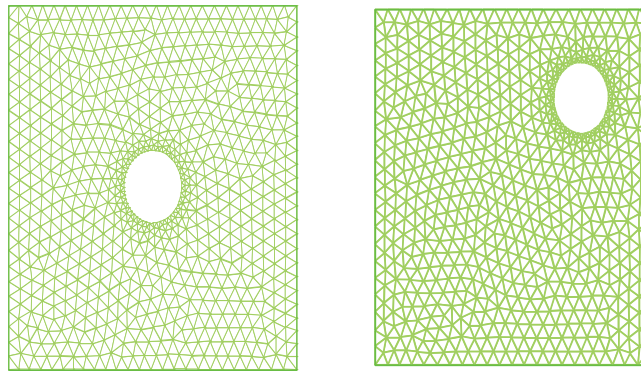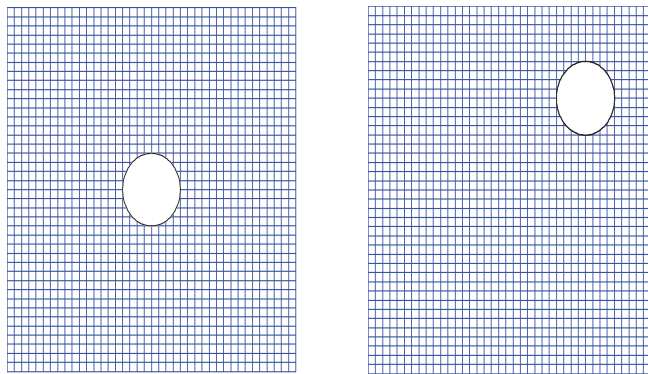
(a) Conforming mesh. Left: $t=t_1$; Right: $t=t_2$.



(a) Non-conforming mesh. Left: $t=t_1$; Right: $t=t_2$.

Figure 2: Examples of conforming mesh (a) and non-conforming mesh (b).

There have been several books and reviews related to the numerical study of fluid-structure interactions. Morand and Ohayon (1995) presented a number of numerical methods in modeling the linear vibrations of elastic structures coupled with internal fluids, with applications focused on sloshing, hydroelasticity and structural acoustics. Dowell and Hall (2001) provided an in-depth discussion of nonlinear dynamical modeling of FSI problems, largely drawn from applications in aerospace engineering, with an emphasis on the construction of reduced-order models (ROM) based on rigorous fluid dynamical theory. Related computational challenges were also discussed in this work. Chakrabarti (2005) represented a collection of several numerical works in modeling FSI problems in the context of ocean engineering. Mittal and Iaccarino (2005) extensively reviewed FSI computational techniques based on the immersed boundary formulation, originally proposed by Peskin (1977). Shyy et al. (2007) described a variety of computational methods for general moving boundary problems in fluid dynamics which also cover FSI applications. Particularly, quite a few numerical techniques in the framework of the finite-volume approach were carefully discussed and demonstrated by various ap-

plications. In addition, Lefrançois and Boufflet (2010) presented several numerical FSI models, based on a simple example of a gas enclosed in a chamber with a moving piston, and conducted detailed analysis for the pros and cons of each model.

In the current review article, we intend to review numerical methods for FSI problems with incompressible flows from a broader context of scientific and engineering disciplines, and discuss the importance of interdisciplinary collaboration in advancing the study in this field. Particularly, this article will review the solution procedures of the partitioned approach-based conforming mesh methods and the immersed method-based non-conforming mesh methods. It is a goal of this article to identify the key features of the methods reviewed here that may be integrated to form an efficient and accurate algorithm to meet the computational challenges of FSI problems.

This paper first outlines the basic FSI problem formulation in Section 2. The partitioned approach-based conforming-mesh methods are reviewed in Section 3. The review of the non-conforming mesh methods is given in Section 4. Discussion and remarks are made in Section 5 to conclude the paper.

## 2  FSI problem formulation

We consider a computational domain, denoted by $\Omega$, with an external boundary $\Gamma$. The domain includes the structural domain, $\bar{\Omega}_s$, and the fluid domain, $\bar{\Omega}_f$; i.e., $\Omega = \bar{\Omega}_s \cup \bar{\Omega}_f$. The fluid-structure interface is defined by $\Gamma_s = \bar{\Omega}_s \cap \bar{\Omega}_f$. See Fig. 3 for illustration of the domains. For notational simplicity, we adopt the tensor notation below.
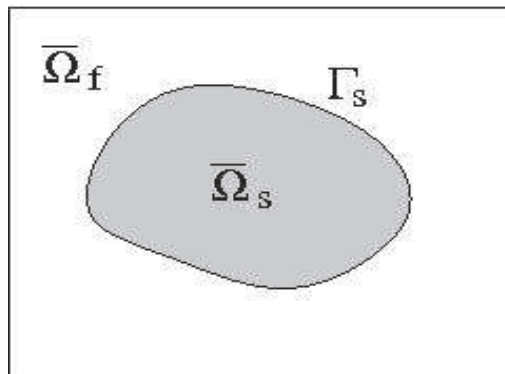


Figure 3: Schematic of the fluid and solid domains in a FSI problem.

The equations of motion for the fluid and structure may be expressed in the same index form, as a result of the D'Alembert's principle:

$$\rho \dot{v}_i - \sigma_{ij,j} + f_i = 0, \tag{2.1}$$

where $f_i$ is the body force, such as gravity. Specifically, in the structural domain, the

equation is written as

$$\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s = 0, \qquad \text{in } \bar{\Omega}_s, \tag{2.2}$$

where the superscript, $s$, denotes the quantity associated with the structure. Note that the velocity, $v_i^s$, is the material (or total) time derivative of the displacement field $u_i^s$, i.e., $v_i^s = \dot{u}_i^s$. Eq. (2.2) is usually given in the Lagrangian description. The first two terms in Eq. (2.2) are associated with inertia and internal stresses, respectively. For example, for linear elastic materials, the structural stress follows the linear Hooke's law; i.e.,

$$\sigma_{ij}^s = \lambda \delta_{ij} \varepsilon_{ll} + 2G\varepsilon_{ij},$$

where the structural stress $\sigma_{ij}^s$ is a function of the strains, $\varepsilon_{ij}$, and the Lame constants $\lambda$ and $G$, which are defined by

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right),$$
$$G = \frac{E}{2(1+\nu)},$$
$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)},$$

where $E$ and $\nu$ are the Young's modulus and the Poisson's ratio, respectively. In the fluid domain, the equation is given by

$$\rho^f \dot{v}_i^f - \sigma_{ij,j}^f + f_i^f = 0, \qquad \text{in } \bar{\Omega}_f, \tag{2.3}$$

which is usually represented by the Eulerian description. Thus, in the inertia term, one has

$$\dot{v}_i^f = \frac{dv_i^f}{dt} = \frac{\partial v_i^f}{\partial t} + v_j^f v_{i,j}^f.$$

Assuming that the incompressible Newtonian fluid model is used here, the fluid stress, $\sigma_{ij}^f$, is then given by

$$\sigma_{ij}^f = -p\delta_{ij} + \tau_{ij},$$

where

$$\tau_{ij} = 2\mu(e_{ij} - \delta_{ij}e_{kk}/3), \qquad e_{ij} = (v_{j,i}^f + v_{i,j}^f).$$

Note that $p$ is the static pressure which may be viewed as the necessary force to enforce the incompressibility condition, $v_{i,i}^f = 0$.

To maintain the no-slip condition along the fluid-structure interface $\Gamma_s$, the following Dirichlet and Neumann conditions can be imposed,

$$v_i^s = v_i^f, \qquad \text{on } \Gamma_s, \tag{2.4}$$

$$\sigma_{ij}^s n_i = \sigma_{ij}^f n_i, \qquad \text{on } \Gamma_s. \tag{2.5}$$

Eq. (2.5) is in fact the differentiation of the displacement condition that both fields share the same interface,

$$x_i^s = x_i^f, \qquad \text{on } \Gamma_s. \tag{2.6}$$

For an interface profile that is smooth in time and space, some FSI methods consider Eq. (2.6) as the Dirichlet constraint, instead of Eq. (2.4).

As mentioned before, FSI numerical techniques can be categorized into two classes; i.e., methods with conforming and non-conforming meshes. These in turn depend upon the procedure used to enforce the transmission conditions, Eqs. (2.4)-(2.6). The conforming-mesh methods track the motion of the interface and enforce Eqs. (2.5) and (2.6) on the interface explicitly, thus requiring mesh update. The conforming-mesh method provides a convenient framework to incorporate the partitioned approach. The non-conforming mesh methods, most notably, the immersed boundary method (Peskin 1977, 2002), enforce the Dirichlet condition, Eq. (2.4) instead. The non-conforming mesh methods can be derived from the theorem of Lagrange multipliers (Haug 1992), where the Lagrange multipliers in most cases appear as source (or, forcing) terms in the fluid equation. Thus, in these methods, computation of the Lagrange multipliers is essential and directly affects the accuracy of the fluid and solid solutions. These two classes of FSI methods are discussed below in detail in Sections 3 and 4, respectively.

## 3   Conforming-mesh methods

The FSI methods with conforming meshes usually involve three fields that describe respectively the fluid dynamics, structural dynamics and mesh movement. The emphasis of these methods is on the coordination of data transfer and consistency between the existing fluid and structural codes. Most FSI methods use the generalized Gauss-Seidel (GGS) approach (Newman et al. 1999) for the coupled analysis, in which the fluidic and structural computation will be performed in a sequential manner to achieve a multidisciplinary solution. In other words, one may first solve the fluid field at a given time instance with an assumed interface location. The resulting fluid pressure and stress are then applied to the structure as external forces. The structural computation is then conducted to update the position of the structural surface. New fluid mesh is then created to accommodate the new interface location. An iterative process may be required to ensure that the interfacial conditions of both the displacement and the force are satisfied at the given time instance before marching to the next time instance. The challenges one might encounter when computing by means of an iterative coupled procedure are to maintain proper data transfer between the disciplines and to reach the converged solution efficiently.

Below we review techniques for interface data transfer and mesh movement and we address the accuracy, stability and efficiency of the methods.

## 3.1   Interface data transfer

Generally speaking, the fluid dynamic module in the conforming-mesh methods tends to pay attention to the physics details around the fluid-structure interface. The fluid dynamics mesh model usually faithfully represents the structure geometry including details such as tunnel, struts and hard chine. On the other hand, the structural analysis module concerns mainly the force bearing members. Therefore, the refinement of the structural mesh will be placed around the high stress areas which most likely will not be on the fluid-structure interface. As a result, depending upon the degree of fidelity used in the fluid or structure computation, their associated meshes on the interface contain mismatches and even gaps. This incongruence can cause numerical difficulties in dealing with fluid dynamic load transfer and elastic deformation update. Remedies that have been proposed may be collectively categorized into two approaches, which we refer to as the point match method and the artificial interface structure method.

The first step in the point match method is to identify and match a fluid mesh point to a structural mesh point on the fluid-structure interface. The fluid or structural mesh points selected can be at a vertex, the center or the Gauss point of a mesh element. The connection relation between the matched points may be established by determining the shortest distances between the points (Brown 1997) or based upon the normal projection (Onishi et al. 1998).

The displacement of the structural mesh point can be transferred to the fluid surface mesh point through a rigid element that connects the matched points (Brown 1997, Onishi 1998, Cebral and Lohner 1997a,b; Farhat et al. 1998). Once the displacements at the selected fluid surface mesh points are known, the displacement vectors at the rest of the fluid surface mesh points can be obtained through local or global interpolation (Raveh 1998, Brown 1997, Farhat et al. 1998, Tsong et al. 1996).

The aerodynamic pressure load on the fluid surface mesh can also be transferred to the structure surface mesh based upon the connection relation between a pair of the identified match points. This process is usually completed with help of the consistency of the virtual work. That is, the work done by the structural load applied to the structural surface mesh is the same as that of the fluid dynamic load applied to the fluid surface mesh. However, this procedure does not guarantee the conservative aspect of the load transferring method. For example, the resultant fluid dynamic loads may not necessarily be equal to the resultant structural loads. Cebral and Lohner (1997a) thus developed a conservative load projection method to transfer fluid dynamic load. Although their new method preserves the magnitude of the loads, it may not guarantee the consistency between the fluid and structural solutions.

Samareh (1996, 1998, 1999a,b) developed a special connection method that takes the shape design representation into consideration. In his works, a non-uniform rational B-spline (NURBS) representation is first constructed to model the wing of an aircraft. The structural displacements on the structure surface mesh points are not transferred to the matched fluid surface mesh points directly; instead, it is projected onto the NURBS

model. A new NURBS is then constructed to represent the deformed geometry with which the new fluid dynamic surface mesh may be established. The load transfer can then be accomplished in a similar manner.

Although Samareh's method does not appear to be consistent or conservative, it offers a distinct advantage from the design point of view. Since the geometry of the NURBS surface is regulated by the control points, the coordinates of those points become a natural choice in shape design variables. Furthermore, since the NURBS representation is a linear function of the coordinates of the control points, the shape derivatives of the NURBS surface are readily available. Thus, the shape derivatives of the load transfer and deformation tracking necessary in coupled FSI sensitivity analysis can be obtained without difficulty (Samareh 1999c).

The second group of remedial procedures (Appa 1989, Guruswamy 1994, Kapania et al. 1996), sometimes called the mortar method, introduces an artificial structure to cover the interface between the structural model and the fluid dynamic model. An example of the mortar method presented by Hou and Satyanarayana (2000) is given hereafter for reference.

Let the three-dimensional unstructured computational mesh be represented by $X_f$, known as the fluid dynamic mesh. This mesh encompasses the fluid domain surrounding the structure, $\bar{\Omega}_f$. The fluid dynamic pressure acting on the fluid-structure interface has to be converted to forces acting at the structural surface mesh points, so that the structural module can be solved. The mesh on the fluid dynamic surface is referred to as the fluid dynamic surface mesh. The mesh on the structural surface for structural analysis is called the structural surface mesh, $X_s^s$. The equivalent forces acting on the structural surface mesh must be approximated from the pressure and stresses acting on fluid dynamic surface meshes. This is due to the fact that two different mesh sizes are usually employed on the respective sides of the fluid-structure interface. That is, generally a fine mesh has been used for fluid dynamic simulation, while a coarse mesh has been used for structural simulation. The mortar method introduces an artificial thin shell structure that covers the fluid-structure interface as a vehicle for transferring the load and the displacement data between the fluid and the structural domains. Detailed discussion of such load transfer and deformation tracking are given respectively in the next two subsections.

**Structural load approximation**

The artificial shell structure in the mortar method covers the interface with the grids that include the fluid dynamic as well as the structural surface meshes, $X_f^s$ and $X_s^s$. To build the mesh of the artificial shell structure, one can start with the fluid dynamic surface mesh. One then updates the mesh by dividing the surface mesh elements so as to create new mesh that embraces the structural mesh points. As a result, the artificial shell structure will include the structural nodes as part of its mesh. To find the corresponding nodal forces applied to the structural surface, one can conduct static analysis of the artificial shell structure in which all of the fluid dynamics nodes will be subjected to fluid forces, $f_f^s$, while all of the structural nodes will be completely constrained; i.e., $U_s^s = 0$.

The process can be described by the following linear matrix equation

$$
\begin{bmatrix} K_{ff}^s & K_{fs}^s \\ K_{sf}^s & K_{ss}^s \end{bmatrix} \left\{ \begin{array}{c} U_f^s \\ U_s^s \end{array} \right\} = \left\{ \begin{array}{c} f_f^s \\ f_s^s \end{array} \right\},
\tag{3.1}
$$

where $U_s^s = 0$ and where the $K^s$ matrix is the stiffness matrix associated with the artificial shell structure. The subscript $f$ indicates that the associated quantity is related to the fluid dynamic surface node and s to the structural surface mesh node. The reaction force vector, $f_s^s$, that is applied at the constrained structural nodes is given by

$$
f_s^s = K_{sf}^s U_f^s,
$$

where the displacement, $U_f^s$, at the fluid dynamic surface nodes is obtained by solving

$$
K_{ff}^s U_f^s = f_f^s.
$$

The fluid dynamic load that will be applied to the structural nodes for structural analysis is then simply obtained as $r_s = -f_s^s$.

**Deformation tracking**

Updating the fluid dynamic mesh from the structural displacements is an important step in conforming-mesh methods. The mortar method can help accomplish this in two steps. The first step is to transform structural surface displacements to fluid dynamic surface displacements. The second step is to transfer the fluid dynamic surface displacements to the interior grid points of the fluid dynamic mesh. In the first step, the corresponding fluid dynamic surface deformation, $U_f^s$, is obtained by solving the following system pertaining to the artificial shell structure described by Eq. (3.1):

$$
\begin{bmatrix} K_{ff}^s & K_{fs}^s \\ K_{sf}^s & K_{ss}^s \end{bmatrix} \left\{ \begin{array}{c} U_f^s \\ U_s^s \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0 \end{array} \right\}.
\tag{3.2}
$$

However, at this time, no external forces is involved, but the boundary displacement, $U_s^s$, is known. The first row of the above equation gives a solution that yields the displacements on the fluid dynamic surface nodes, $U_f^s$:

$$
K_{ff}^s U_f^s = -K_{fs}^s U_s^s.
$$

Or simply put in a linear transformation form as

$$
U_f^s = \bar{K}_{fs} U_s^s,
\tag{3.3}
$$

where the transformation matrix, $\bar{K}_{fs}$, is defined by

$$
\bar{K}_{fs} = -\left( K_{ff}^s \right)^{-1} K_{fs}^s.
$$

Note that the known structural displacement vector, $U_s^s$, is used in Eq. (3.3) as the prescribed boundary movement at the structural surface mesh points of the artificial shell structure. It is possible to apply this type of boundary movement to the artificial shell structure because both the fluid dynamic and the structural nodes are the subsets of the surface mesh.

The second step in deformation tracking is to transfer fluid dynamic surface displacements, $U_f^s$, to the fluid dynamic interior grid points. This can be accomplished by following the same strategy adopted for the fluid dynamic surface mesh. In this strategy, an artificial elastic media covers the entire fluid dynamic domain is introduced. The mesh of this artificial elastic media is identical to that in the fluid dynamic domain. The movement of the interior fluid dynamic mesh can be obtained by solving the following matrix equation, which is similar to Eq. (3.2),

$$
\begin{bmatrix} K_{ff}^a & K_{fs}^a \\ K_{sf}^a & K_{ss}^a \end{bmatrix} \begin{Bmatrix} U_f^a \\ U_s^a \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}. \tag{3.4}
$$

However, $K^a$ in Eq. (3.4) is the stiffness matrix of the artificial elastic media, which is different from $K^s$ in Eq. (3.2). The movement of the interior fluid mesh points, $U_f^a$, is related to fluid dynamics surface mesh movement, $U_s^a$, as

$$
K_{ff}^a U_f^a = -K_{fs}^a U_s^a, \tag{3.5}
$$

where the movement of the fluid dynamics surface mesh has been solved by Eq. (3.3) in the first step; i.e., $U_s^a = U_f^s$. The resulting set of linear systems, Eq. (3.5), may be solved for the displacements of each interior node using several Jacobi iterations without explicitly forming $K^a$ so as to save the computer memory. The positions of the interior nodes are then updated using the determined displacements of the fluid dynamic interior mesh points, $U_f^a$. This iterative method does not require a large amount of memory, but does require an initial guess for the solution. Eq. (3.5) can also be put into a simpler form as

$$
U^a = \bar{K}^a U_s^a, \tag{3.6}
$$

where $U^a$ is the mesh for the entire fluid domain and the transformation matrix, $\bar{K}^a$, is defined by

$$
\bar{K}^a = \begin{bmatrix} -\left(K_{ff}^a\right)^{-1} K_{fs}^a \\ I \end{bmatrix}.
$$

The load transfer method introduced above can be proven to be consistent and conservative. The conservativeness of the process can be proven by illustrating that the summation of forces and moments on fluid dynamic surface is equal to that on the structural surface. The consistency of the transfer process is demonstrated by showing that the work done by the structural surface load applied to the structure is equal to the work done by the fluid dynamic load.

One popular FSI method in this class is the Arbitrary Lagrangian-Eulerian (ALE) technique (Souli and Benson 2010) which incorporates the moving mesh explicitly into the fluid dynamics equation. It allows arbitrary motion of grid points with respect to their frame of reference by taking into account the convection of the material points. The material derivative in this case is expressed as

$$\frac{d\boldsymbol{v}_f}{dt} = \frac{\partial \boldsymbol{v}_f}{\partial t} + (\boldsymbol{v}_f - \boldsymbol{U}) \bullet \nabla \boldsymbol{v}_f.$$

The movement of the fluid mesh, $\boldsymbol{U}$, can be set as $U^a$ of Eq. (3.6).

## 3.2  Accuracy and efficiency

The sub-iterations between the fluid and structure solutions are important to the numerical performance of the method. In fact, Wood et al. (2010) showed that the FSI solution based on sequential computation of fluid and structural dynamics becomes unstable, if there are no sub-iteration steps between fluid and structural computations. One additional sub-iteration can reduce two order magnitude of numerical error. And with more sub-iterations, better convergence can be achieved without a substantial increase in computational time. The particular example they studied for numerical performance is the flow-induced oscillation of a flexible cantilever. The authors used the three-step second-order backward difference algorithm to approximate the first-order time derivative in the fluid solver and used the one-step Newmark predictor-corrector algorithm to solve the nonlinear structural dynamic problem. The meshes range from 23,334 to 46,164 three-dimensional fluid nodes and 567 to 850 two-dimensional structural nodes. The example is run with Dell PowerEdge SC1420 Severer with two Intel Xeon processors. Ten sub-iterations take about 4 minutes of wall clock time and the problem itself takes about 4 days of wall clock time for the simulation. It is interesting to note that in their study, 80% of computational time is for the fluid, 10% for the structure, and 10

Many researchers have developed methods to improve the treatment of interface conditions, in an attempt to attain better accuracy, stability and efficiency for the three-field FSI methods. Some suggested ways to estimate the interface location before starting the new FSI iteration or even replace the standard Dirichlet and Neumann interface conditions by a general Robin Transmission (Badia et al. 2008). With better prediction of the interface locations, Farhat et al. (2006) built a FSI method with second-order accuracy in time. Zhang et al. (2007) also developed a second-order FSI method in which the computational fluid dynamics (**CFD**) code is treated as a black box. Vierendeels et al. (2008) constructed reduced-order models to improve computational efficiency. These three methods are described below, along with the work on Robin Transmission conditions by Badia et al. (2008).

**Second-order accuracy without sub-iterations**

Farhat et al. (2006) proposed two algorithms with second-order temporal accuracy. The algorithms are derived based upon second-order solution procedures for solving the fluid and the structure fields. A three-point backward difference method is used for solving the fluid field. The first solution procedure proposed by the authors is described below.

Step 1: Predict the interface velocity based upon the result of a second-order accurate structural dynamic calculation:

$$u_{\Gamma_s}^{n+1,p} = u_{\Gamma_s}^n + \Delta t v_{\Gamma_s}^n + \frac{1}{2}\Delta t \left( v_{\Gamma_s}^n - v_{\Gamma_s}^{n-1} \right).$$

Step 2. Update the interface location and generate new fluid domain mesh. The new position of fluid dynamic surface mesh is obtained by

$$x_{\Gamma_s}^{n+1,p} = x_{\Gamma_s}^{n,p} + \bar{K}_{fs}\Delta u_{\Gamma_s}^{n+1,p},$$

where $\Delta u_{\Gamma}^{n+1,p} = u_{\Gamma}^{n+1,p} - u_{\Gamma}^n$ and $\bar{K}_{fs}$ is the transformation matrix defined by Eq. (3.3). Similarly, the new fluid domain mesh can be updated by the following equation,

$$x_f^{n+1,p} = x_f^{n,p} + K^a \Delta u_{\Gamma_s}^{n+1,p}.$$

$K^a$ is given by

$$K^a = \frac{\bar{K}^{a,n} + \bar{K}^{a,n+1,p}}{2},$$

where the terms in the numerator are associated with the transformation matrix defined by Eq. (3.6) as

$$\bar{K}^{a,n} = \begin{pmatrix} -\left( K_{ff}^a(x^n) \right)^{-1} K_{fs}^a(x^n) \\ I \end{pmatrix}$$

in which the stiffness matrices are associated with Eq. (3.4).

Step 3: Solve the fluid equation with a second-order accurate algorithm for the updated velocity and pressure, $v_f^{n+1}$ and $p^{n+1}$, based upon the updated mesh, $x_f^{n+1,p}$.

Step 4: Find the equivalent force resulting from the fluid pressure, $p^{n+1}$, acting on the wet surface (i.e., the fluid-structure interface) described by $x_{\Gamma_s}^{n+1,p}$ and use it to find the corrected structural response, $u_s^{n+1}$.

In contrast, the second algorithm Farhat et al. proposed is a half-a-time-step algorithm. The details are presented below.

Step 1: Predict the interface velocity based upon the result of a second-order accurate structural dynamic calculation:

$$u_{\Gamma_s}^{n+\frac{1}{2},p} = u_{\Gamma_s}^n + \frac{\Delta t}{2} v_{\Gamma_s}^n + \frac{1}{8}\Delta t \left( v_{\Gamma_s}^n - v_{\Gamma_s}^{n-1} \right).$$

Step 2. Update the fluid dynamic surface mesh and generate new fluid domain mesh accordingly. The updated fluid dynamic surface location is obtained by

$$x_f^{n+\frac{1}{2},p} = x_f^{n-\frac{1}{2},p} + K_a^n \Delta u_{\Gamma_s}^{n+\frac{1}{2},p}, \tag{3.7}$$

where $\Delta u_{\Gamma_s}^{n+1/2,p} = u_{\Gamma_s}^{n+1/2,p} - u_{\Gamma_s}^{n-1/2}$. To simplify the algorithm, the authors suggested approximating Eq. (3.7) by

$$x_f^{n+\frac{1}{2},p} = x_f^{n-\frac{1}{2},p} + K_a^{n-\frac{1}{2}} \Delta u_{\Gamma_s}^{n+\frac{1}{2},p}.$$

Step 3: Solve the fluid equation from $t^{n-1/2}$ to $t^{n+1/2}$ for the updated velocity and pressure, $v_f^{n+1/2}$ and $p^{n+1/2}$, based upon the updated mesh, $x_f^{n+1/2,p}$.

Step 4: Find the equivalent force resulting from the fluid pressure, $p^{n+1/2}$, acting on the wet surface described by $x_{\Gamma_s}^{n+1/2,p}$ and use it to solve the structural dynamic problem at $t^{n+1/2}$ for the new structural response, $u_s^{n+1/2}$. The solution at $t^{n+1}$ is obtained by

$$u_s^{n+1} = 2u_s^{n+\frac{1}{2}} - u_s^n.$$

Both algorithms achieve the second-order temporal accuracy without additional sub-iterations.

Zhang et al. (2007) used an aeroelastic flutter problem as a vehicle to investigate the accuracy, stability and efficiency of two proposed algorithms. These algorithms are centered on the structural dynamic equation in which the pressure force is supplied by the external blackbox CFD code.

The first algorithm used the standard fourth-order Runge-Kutta method to solve the structural dynamic equation. The discretized equation requires the values of the fluid pressure at the current time and at other intermediate time steps such as $p(t+\Delta t/2)$. The latter were then approximated by a second-order backward extrapolation in time; e.g.,

$$p\left(t + \frac{\Delta t}{2}\right) \approx \frac{1}{8}(3p(t-2\Delta t) - 10p(t-\Delta t) + 15p(t)).$$

As a result, the discretized structural dynamic equation can be solved with the fluid pressure calculated at the last three time steps. Once new structural solution is found, the interface boundary is updated. The CFD code is called to generate the new pressure load which is saved for solving the structural equation at the next time step. Only one CFD procedure is called at each time step.

The second algorithm proposed by Zhang et al. used the multi-step, implicit second-order Adams scheme to solve the structure dynamic equation, in which the predictor is an explicit second-order Adams scheme. The aerodynamic force at time $n+1$ in the corrector can be approximated by a second-order relation,

$$p(t+\Delta t) = 2p(t) - p(t-\Delta t)$$

to result in a solution with second-order accuracy, or by a fourth-order relation,

$$p(t+\Delta t) = 4p(t) - 6p(t-\Delta t) + 4p(t-2\Delta t) - p(t-3\Delta t)$$

to result in a solution with fourth-order accuracy. Again, the algorithm requires one CFD call at each time step. Their numerical results based on the flutter analysis showed that both algorithms are superior to the conventional nested method in which the fluid and structure equations are solved alternately.

**Reduced-order modeling and interface location prediction**

Vierendeels et al. (2008) introduced a reduced-order modeling (ROM) method to solve the heart valve dynamics problem. The heart valve is modeled by a chain of rigid linkages, joined by hinges along with torsional compliance. The set of implicit equations for the discretized FSI problem can be symbolically represented by

$$\begin{cases} G\left(x^{n+1}, p^{n+1}\right) = 0, \\ p^{n+1} = F\left(x^{n+1}\right), \end{cases} \tag{3.8}$$

where the first equation is the discretized structural dynamic equation and the second equation is the fluid solver. At the time step $t^{n+1}$, the input of the structural equation is the fluid pressure, $p^{n+1}$, while the output is the interface position, $x^{n+1}$. Conversely, the input of the fluid equation is $x^{n+1}$ and the output is $p^{n+1}$. The structural dynamic equation may be simplified as a nonlinear equation of $x^{n+1}$ as

$$G\left(x^{n+1}, p^{n+1}\right) = G\left(x^{n+1}, F\left(x^{n+1}\right)\right) = 0.$$

The interface variables are the interface condition and pressure. An iterative process can be set up to solve for $x^{n+1}$:

$$0 \approx G\left(x^{n+1,k}, p^{n+1,k}\right)$$
$$= G\left(x^{n+1,k-1}, p^{n+1,k-1}\right) + \left.\frac{\partial G}{\partial x}\right|_{x^{n+1,k-1}, p^{n+1,k-1}} \times \Delta x + \left.\frac{\partial G}{\partial p}\right|_{x^{n+1,k-1}, p^{n+1,k-1}} \times \Delta p, \tag{3.9}$$

where $k$ is the iteration index and $\Delta p \approx p(x^{n+1,k-1} + \Delta x) - p(x^{n+1,k-1})$. Now, let the changes in the interface location be represented by a linear combination of $(k-1)$ "displacement modes", $\{\varphi_i\}$, which are the changes in the interface location in the $i$-th iteration; i.e., $\varphi_i = x^{n+1,i} - x^{n+1,0}$. With a collection of $k$ displacement vectors, one can then approximate any possible changes in $x$ as $\Delta x \approx \sum_{i=1}^{k-1} a_i \varphi_i = \Phi^{k-1} a$ in which $a$ may be found through a best fit as $a = (\Phi^{k-1^T} \Phi^{k-1})^{-1} \Phi^{k-1^T} \Delta x$. The change in $p$ due to the change in $x$ can be approximated at the beginning of each iteration by

$$\Delta p \approx \sum_i^{k-1} (\Delta p_i \bullet a_i) = \sum_i^{k-1} \left( \left[ p\left(x^{n+1,0} + \varphi_i\right) - p\left(x^{n+1,0}\right) \right] \bullet a_i \right).$$

Let $\Psi^{k-1} \equiv [\Delta p_1 \ \Delta p_2 \ \cdots \ \Delta p_{k-1}]$ Specifically, the pressure change, $\Delta p_i$, in each column of $\Psi$ is solved by an additional CFD run at the beginning of the iterations as

$$\Delta p_i = p\left(x^{n+1,0} + \varphi_i\right) - p\left(x^{n+1,0}\right) = F\left(x^{n+1,0} + \varphi_i\right) - F\left(x^{n+1,0}\right). \tag{3.10}$$

Then, $\Delta p = \Psi^{k-1} a$, or

$$\Delta p = \Psi^{k-1} a = \Psi^{k-1} (\Phi^{k-1^T} \Phi^{k-1})^{-1} \Phi^{k-1^T} \Delta x.$$

Therefore, Eq. (3.9) becomes

$$
0 \approx G\left(x^{n+1,k}, p^{n+1,k}\right)
$$
$$
= G\left(x^{n+1,k-1}, p^{n+1,k-1}\right) + \left(\left.\frac{\partial G}{\partial x}\right|_{x^{n+1,k-1}, p^{n+1,k-1}} + A^{n+1,k-1}\right) \Delta x, \tag{3.11}
$$

where

$$
A^{n+1,k-1} = \left.\frac{\partial G}{\partial p}\right|_{x^{n+1,k-1}, p^{n+1,k-1}} \Psi^{k-1}\left(\Phi^{k-1^T}\Phi^{k-1}\right)^{-1}\Phi^{k-1}.
$$

Eq. (3.11) can be solved for $\Delta x$ so as to update $x^{n+1}$ and the associated terms,

$$
\left.\frac{\partial G}{\partial x}\right|_{x^{n+1,k-1}, p^{n+1,k-1}} + A^{n+1,k-1}, \tag{3.12}
$$

which can be a time-consuming task because updating $A^{n+1,k-1}$ involves solving the fluid equations. The authors suggested solving Eq. (3.11) in two steps. Eq. (3.11) is used only to update $x^{n+1}$. Specifically, Eq. (3.11) becomes

$$
0 \approx G\left(x^{n+1,k,s}, \tilde{p}^{n+1,k,s}\right)
$$
$$
= G\left(x^{n+1,k-1,s-1}, p^{n+1,k-1}\right) + \left(\left.\frac{\partial G}{\partial x}\right|_{x^{n+1,k-1,s}, p^{n+1,k-1}} + A^{n+1,k-1}\right) \Delta x, \tag{3.13}
$$

where

$$
\tilde{p}^{n+1,k,s} = p^{n+1,k-1} + A^{n+1,k-1}\Delta x.
$$

Note that in the above iteration, $A^{n+1,k-1}$ remains unchanged during the interactions. Therefore, once $x^{n+1,k}$ is converged in the iteration of Eq. (3.13), one solves the CFD equation, $p^{n+1,k} = F(x^{n+1,k})$, to find the corrected pressure. A new $\varphi_k$ and a new $\Delta p_k$ are then found;

$$
\varphi_k = x^{n+1,k} - x^{n+1,k} \quad \text{and} \quad \Delta p_k = p^{n+1,k} - p^{n+1,k-1},
$$

which are used to expand $A^{n+1,k}$. With the newly expanded $A^{n+1,k}$, the iteration of Eq. (3.13) is continued until the condition, $|x^{n+1,k} - x^{n+1,k-1}| \leqslant \varepsilon$ for a prescribed small $\varepsilon$, is satisfied. The FSI solution of Eq. (3.8) at time $t_{n+1}$ is then regarded as having converged. The authors suggested starting the iteration with one displacement mode. The heart valve problem studied by the authors required only 3 to 4 displacement modes to reach convergence at every time step.

   In the companion paper (Degroote et al. 2008), the authors indicated that the conventional block Gauss-Seidel method may not converge for FSI problems with strong coupling such as the case when the flexibility of the structure increases (Causin et al. 2005). The problem they studied is the unsteady blood flow in a flexible tube. In the first part of the paper when solving a 1D example, the authors derived a linear relation based upon

the linear structure equation with which the update of the interface location is related to that of the pressure. In this way, the interface location is updated while solving the fluid equation for flow velocity and pressure. This extra relation helps the FSI iteration method converge for a very flexible structure.

For more complex problems, the authors suggested using the ROM as an efficient reanalysis method to support sub-iterations and maintain the stability of the iterative algorithm. They constructed ROMs for both the fluid and the structure based upon the procedure introduced above, in which the computational fluid dynamics and the structural analysis codes are used as black boxes. In this study, the authors incorporated ROMs in their FSI solution procedure after two sub-iterations. In the first sub-iteration, a multi-step predictor is introduced to estimate the interface location used in the fluid solution,

$$x_{\Gamma_s,1}^{n+1} = \frac{5}{2}x_{\Gamma_s}^n - 2x_{\Gamma_s}^{n-1} + \frac{1}{2}x_{\Gamma_s}^{n-2}. \tag{3.14}$$

In the second iteration, after the interface location, $x_{\Gamma_s,1}^{n+1}$, is found by solving the structural equation, the following relation is used to correct the interface location for the new value of the fluid solution,

$$x_{\Gamma_s,i+1}^{n+1} = (1 - \omega_i)x_{\Gamma_s,i}^{n+1} + \omega_i x_{\Gamma_s,i+1}^{n+1}, \tag{3.15}$$

where $\omega_i = 1 - \mu_i^{n+1}$. The Aitken factor $\mu_i^{n+1}$ (Irons and Tuck 1969) is obtained by the following relation

$$\mu_i = \mu_{i-1} - (\mu_{i-1} - 1) \times \frac{(\Delta x_{\Gamma_s,i} - \Delta x_{\Gamma_s,i+1})^T \Delta x_{\Gamma_s,i+1}}{(\Delta x_{\Gamma_s,i} - \Delta x_{\Gamma_s,i+1})^2}, \tag{3.16}$$

where $\Delta x_{\Gamma_s,i} = x_{\Gamma_s,i} - d_{\Gamma_s,i-1}$. These two estimated interface locations can be conveniently used as the displacement modes to construct the ROM.

Instead of Eq. (3.14), Wall et al. (2008) proposed the following equation to more accurately estimate the new location of the fluid-structure interface before starting the conventional nested fluid-structure iteration. The new location of the fluid-structure interface is first estimated by

$$x_{\Gamma_s}^{n+1} = x_{\Gamma_s}^n + \Delta t \left( \frac{3}{2}u_{\Gamma_s}^n - \frac{1}{2}u_{\Gamma_s}^n \right).$$

This equation is in fact the same as Eq. (3.14), if the velocity terms are replaced by a forward difference of the displacement. With this new domain boundary, the artificial elastic media equation is solved to determine the new interior mesh, upon which the Navier-Stokes equation is solved. The pressure forces on the fluid-structure interface are then incorporated into the structural equation of motion to determine the structural response. The new position of the fluid-structure interface is then updated through the Aitken under-relaxation factor described by Eqs. (3.15) and (3.16).

**Modification of interface conditions: Robin transmission conditions**

The FSI problems require the fluid and the structure fields at the common interface to share not only the same interface location, but also the same velocity due to the no-lip condition and the common normal stress. The velocity condition is a Dirichlet condition, a time-integration of which should yield the condition for the same interface location. The stress condition, on the other hand, is a Neumann condition. To advance the solution, the Dirichlet condition is usually imposed onto the fluid field and the Neumann condition onto the structure field. That is, once the interface location is updated, the fluid field is solved, subject to the condition that the flow velocity along the interface boundary should be the same as the known velocity of the structure. On the other hand, the structure field is solved subject to the applied load which is statically equivalent to the fluid pressure applied to the interface.

Badia et al. (2008) followed the conventional block Gauss-Seidel scheme to partition the FSI problem. The modification they introduced in the process is a replacement of the individual Dirichlet or Neumann conditions at the interface by a Robin transmission condition. The Robin transmission condition is a weighted, linear combination of the Dirichlet and Neumann conditions; i.e., the velocity and the stress conditions. The weighted coefficients are carefully selected to maintain the stability of the proposed partitioned method. The authors used a simplified fluid-structure model to estimate the weighted coefficients for the problem studied in the paper, which is an approximation of a blood-vessel system. Among the many transmission conditions investigated by the authors, the Robin-Neumann algorithm achieved the best convergence property in that it is always convergent and insensitive to the added mass effect. This Robin-Neumann algorithm imposes the Robin transmission condition on the fluid field and the Neumann transmission condition on the structure field.

# 4   FSI computation using immersed methods

Most of the non-conforming mesh methods are based upon the framework of the immersed methods, which are a class of FSI methods that add force-equivalent terms to fluid equations to represent the fluid-structure interaction and to avoid mesh update in the numerical procedure. The immersed structure can be either a boundary (e.g., a curve in 2D and a surface in 3D) or a body with finite area (in 2D) or volume (in 3D), either rigid or flexible. Below we derive two classes of immersed methods, using the Lagrange multiplier approach: the *immersed boundary method* and the *immersed domain method*. Other types of immersed methods will be reviewed thereafter.

The immersed boundary method was originally developed by Peskin (1977) for studying blood flow through a beating heart, and has since been extensively studied and applied to a wide variety of FSI problems (e.g., Beyer 1992, Blake 1999, Dillon et al. 1995, Fadlun et al. 2000, Fauci and McDonald 1995, Griffith 2005, Huang and Sung 2009, Kim and Choi 2006; Kim, Kim and Choi 2001, Kim and Peskin 2007; Le, Khoo and Lim 2008,
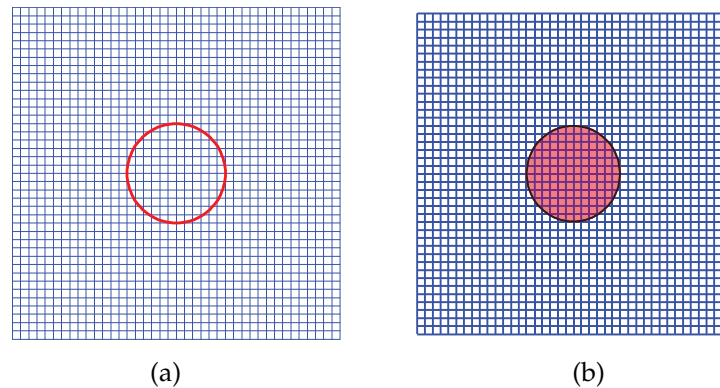
Figure 4: Examples of an immersed boundary (a) and an immersed domain (b).

Stockie and Green 1998, Wang and Layton 2009). This method solves the fluid equations with an additional term, the FSI force, which represents the effects of the immersed boundary acting on the fluid motion. The FSI force is computed explicitly from the structural configuration, which is then used to compute the fluid velocity. The no-slip condition is imposed on the immersed boundary, the location of which is updated by the structural velocity. Essentially, the background fluid equations are solved in the entire domain with a fixed Eulerian mesh, and the moving boundary is tracked separately. The need for mesh update is completely eliminated. For detailed analysis and various applications of this method, the reader is referred to the excellent reviews by Peskin (2002) and Mittal and Iaccarino (2005). ANSYS (1970-2011), one of the most popular computational mechanics and engineering software, incorporated the immersed boundary method for its FSI module in 2009.

In principle, the immersed boundary method deals with structures that do not occupy volumes, e.g., a fiber or a closed curve in 2D space (see Fig. 4a) and a membrane in 3D space. An immersed body that occupies volume (see Fig. 4b) can be approximated by a network of connected fibers, each of which can be treated as an immersed boundary. The disadvantage of this approach is that the realistic structural response to the fluid motion may not be accurately modeled. To more accurately represent the interaction between a fluid and a bulk structure described by detailed constitutive laws, the immersed domain method was introduced. In the immersed domain method, an artificial fluid is introduced to cover the structural domain; thus, fluid domain is extended to the entire computational domain. In the artificial fluid domain, the no-slip condition implies the matching of the position and velocity between the immersed structure and the local fluid. To enforce this no-slip condition, the FSI force is imposed not only on the fluid-structure interfaces but also to every grid point in the artificial fluid domain. The fluid equation is then solved to yield the velocity field of the entire domain. Thus, the structural displacement and velocity are, at this stage, known. They can then be substituted into the suitable structural constitutive law to update the FSI force, which in turn can be used by the fluid

equation to find the new velocity of the fluid points. Representative examples of the immersed domain method include the immersed finite element method developed by Liu et al. (2006), Wang and Liu (2004) and Zhang et al. (2004, 2007), and the immersed continuum method developed by Wang (2006, 2007, 2010).

## 4.1 Basic formulation

We consider the equations that describe the motion of the fluid in $\bar{\Omega}_f$ and the structure in $\bar{\Omega}_s$, given by

$$\rho^f \dot{v}_i^f - \sigma_{ij,j}^f + f_i^f = 0, \qquad \text{in } \bar{\Omega}_f, \tag{4.1}$$

$$\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s = 0, \qquad \text{in } \bar{\Omega}_s, \tag{4.2}$$

where $f_i^f$ and $f_i^s$ are external body forces (e.g., gravity) acting on the fluid and structure, respectively. For a FSI problem, the displacements should be the same along the interface:

$$u_i^f = u_i^s, \qquad \text{on } \Gamma_s, \tag{4.3}$$

which may be viewed as a point-wise constraint applied to the interface, $\Gamma_s$. The no-slip condition imposed on the interface between these two domains is the result of the time differentiation of Eq. (4.3):

$$\dot{u}_i^s = \dot{u}_i^f, \qquad \text{on } \Gamma_s, \tag{4.4}$$

$$\ddot{u}_i^s = \ddot{u}_i^f, \qquad \text{on } \Gamma_s. \tag{4.5}$$

Or, in terms of velocities,

$$v_i^s = v_i^f, \qquad \text{on } \Gamma_s, \tag{4.6}$$

$$\dot{v}_i^s = \dot{v}_i^f, \qquad \text{on } \Gamma_s. \tag{4.7}$$

For simplicity, the superscript f that indicates quantity associated with the fluid field will be dropped from the notation. Based upon the principle of virtual work and the theorem of Lagrange multipliers, Eqs. (4.1)-(4.3) may be combined into a single weak form as

$$0 = \int_{\bar{\Omega}_s} \left( \rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s \right) \delta u_i^s dv + \int_{\bar{\Omega}_f} \left( \rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv + \int_{\Gamma_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv, \tag{4.8}$$

where $\bar{\lambda}_i$ is the associated Lagrange multiplier defined over $\Gamma_s$, representing the force generated from the fluid-structure interaction. Note that the location of the interface boundary, $\Gamma_s$, is part of the unknown, and its position is determined by the interaction between the fluid and the structure.

## Immersed boundary method

In the original formulation of the immersed boundary method invented by Peskin (1977, 2002), the structure is represented by an immersed boundary which does not occupy a finite volume. Thus we have $\bar{\Omega}_s = \Gamma_s$ and the fluid domain becomes the entire computational domain: $\Omega = \Omega_f$. Consequently, Eq. (4.8) becomes

$$0 = \int_{\Gamma_s} \left( \rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s + \bar{\lambda}_i \right) \delta u_i^s dv + \int_{\Omega} \left( \rho \dot{v}_i - \sigma_{ij,j} + f_i - \bar{\lambda}_i L(\Gamma_s) \right) \delta u_i dv, \qquad (4.9)$$

where the delta function, $L(\Gamma_s)$, is defined as

$$L(\Gamma_s) = \begin{cases} 1, & \text{if } x \in \Gamma_s, \\ 0, & \text{if } x \notin \Gamma_s. \end{cases}$$

Eq. (4.9) thus yields two independent equations

$$\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s + \bar{\lambda}_i = 0, \qquad \text{on } \Gamma_s, \qquad (4.10)$$

$$\rho \dot{v}_i - \sigma_{ij,j} + f_i - \bar{\lambda}_i L(\Gamma_s) = 0, \qquad \text{in } \Omega. \qquad (4.11)$$

In the immersed boundary method, the fluid-structure interaction force (i.e., the Lagrange multiplier $\bar{\lambda}_i$) is computed explicitly using Eq. (4.10). The computed force is then imposed on to Eq. (4.11), which is solved to yield fluid motion. In a numerical implementation, the discontinuous function $L(\Gamma_s)$ can be replaced by a continuous discrete delta function, which typically has compact support over a band of grid points neighboring $\Gamma_s$. See Peskin (2002) for a detailed discussion and common choices of discrete delta functions. The use of a discrete delta function can be also regarded as an interpolation of the FSI force from the immersed boundary (the structural domain) to the fluid domain. As a result, the sharp interface is numerically represented by a thin layer of finite depth.

Once the fluid velocity is solved, the velocity of the structure is determined by applying the no-slip condition (4.6). The same discrete delta function is applied to interpolate the velocity from the fluid domain to the boundary. The location of the boundary $\Gamma_s$ is then updated by using the structural velocity, and then used in the next cycle of computation.

## Immersed domain method

The immersed domain method is an extension of the immersed boundary method that simulates motion of an immersed structure which occupies a finite volume. In this case, the constraint described in Eq. (4.3) is extended to the entire structural domain. Thus, Eq. (4.3) becomes

$$u_i^s = u_i, \qquad \text{in } \bar{\Omega}_s, \qquad (4.12)$$

which leads to the modification of the last term in Eq. (4.8) as

$$0 = \int_{\bar{\Omega}_s} \left( \rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s \right) \delta u_i dv + \int_{\bar{\Omega}_f} \left( \rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv + \int_{\bar{\Omega}_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv. \qquad (4.13)$$

To expand the fluid domain to the entire computational domain $\Omega$, the structural domain $\bar{\Omega}_s$ is filled with an artificial fluid. Meanwhile, the virtual work done by the expanded fluid,

$$\int_{\bar{\Omega}_s} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i\right) \delta u_i dv$$

is added to, and subtracted from, the original virtual work equation (4.13). Thus, the modified weak form becomes

$$0 = \int_{\bar{\Omega}_s} \left(\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s\right) \delta u_i^s dv - \int_{\bar{\Omega}_s} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i\right) \delta u_i dv$$

$$+ \left[\int_{\bar{\Omega}_s} (\rho \dot{v}_i - \sigma_{ij,j} + f_i) \delta u_i dv + \int_{\bar{\Omega}_f} (\rho \dot{v}_i - \sigma_{ij,j} + f_i) \delta u_i dv\right] + \int_{\bar{\Omega}_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv$$

$$= \int_{\bar{\Omega}_s} \left[\left(\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s\right) \delta u_i^s - \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i\right) \delta u_i\right] dv$$

$$+ \int_{\Omega = \bar{\Omega}_s \cup \bar{\Omega}_f} (\rho \dot{v}_i - \sigma_{ij,j} + f_i) \delta u_i dv + \int_{\bar{\Omega}_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv. \tag{4.14}$$

Note that Eq. (4.12) implies $\delta u_i^s = \delta u_i$ in $\bar{\Omega}_s$. We thus obtain

$$0 = \int_{\bar{\Omega}_s} \left[\left(\rho^s \dot{v}_i^s - \sigma_{ij,j}^s - \rho \dot{v}_i + \sigma_{ij,j} + f_i^s - f_i + \bar{\lambda}_i\right) \delta u_i^s\right] dv$$

$$+ \int_{\Omega = \bar{\Omega}_s \cup \bar{\Omega}_f} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i - \bar{\lambda}_i L(\bar{\Omega}_s)\right) \delta u_i dv, \tag{4.15}$$

where the function $L(\bar{\Omega}_s)$ is defined as

$$L(\bar{\Omega}_s) = \begin{cases} 1, & \text{if } x \in \bar{\Omega}_s, \\ 0, & \text{if } x \notin \bar{\Omega}_s. \end{cases}$$

Eq. (4.15) yields two independent equations:

$$\rho^s \dot{v}_i^s - \sigma_{ij,j}^s - \rho \dot{v}_i + \sigma_{ij,j} + \bar{\lambda}_i + f_i^s - f_i = 0, \qquad \text{in } \bar{\Omega}_s, \tag{4.16}$$

$$\rho \dot{v}_i - \sigma_{ij,j} + f_i - \bar{\lambda}_i L(\Omega_s) = 0, \qquad \text{in } \Omega. \tag{4.17}$$

Based on Eq. (4.12), the structural velocity is the same as the fluid velocity in $\bar{\Omega}_s$. Therefore, Eq. (4.16) yields

$$\bar{\lambda}_i = (\rho - \rho^s) \dot{v}_i + (\sigma_{ij,j}^s - \sigma_{ij,j}) + (f_i - f_i^s), \qquad \text{in } \bar{\Omega}_s. \tag{4.18}$$

Eqs. (4.17) and (4.18) are the basic formulations for the immersed domain method. Zhang and Gay (2007) derived the governing equations in their immersed finite element method which are similar to those presented here. See Liu, Kim and Tang (2006) for more discussion on the mathematical foundation of this numerical technique. In the original immersed finite element method, Eq. (4.18) is first evaluated to find the FSI force, $\bar{\lambda}_i$. The known force is then imposed on Eq. (4.17) to solve the fluid motion in the entire domain, where a discrete delta function based on the Reproducing Kernel Particle Method, or RKPM (Wang and Liu 2004, Zhang and Gay 2007), is applied to interpolate the force from the structural domain to the fluid domain. The computed fluid velocity is then interpolated back to the structural domain, based on the no-slip condition, and the structural configuration is then updated using the structural velocity. To improve the accuracy at the fluid-structure interface, Lee, Chang and Choi et al. (2008) replaced the discrete delta functions by the Directly Coupled Euler-Lagrange Method (DCELM), in their simulation of rigid body motion using the immersed finite element method.

We note that an important assumption in the immersed domain method is that the structure is incompressible (or nearly incompressible), since the immersed structure has to abide by the same velocity constraint as that of the surrounding incompressible fluid. For many FSI problems, either the volumetric strain of the structure is very low, or the volume of the structure is significantly smaller that of the fluid, thus the incompressibility condition can be approximately satisfied. This assumption, however, may not be valid in situations such as the acoustic FSI problems, where both the fluid and structure have to be modeled as compressible materials. The current review article does not cover FSI simulation with compressible flows. Interested readers may refer to the works of Bathe, Nitikitpaiboon and Wang (1995), Howes (1998), Mönkölä (2010), Ross (2006), Wang, Zhang and Liu (2009), and the references therein.

## 4.2   Other immersed methods

Since Peskin's pioneering work on the immersed boundary method, many related numerical techniques have been developed. In addition to the immersed domain method mentioned above, notable examples include the *immersed interface method* (Layton 2009, Leveque and Li 1997, Li 2003, Li and Ito 2006, Li and Lai 2001; Tan, Lim and Khoo 2009, Xu and Wang 2008), the *direct forcing method* (Fadlun et al. 2000, Guy and Hartenstine 2010, Luo et al. 2007, Mohd-Yusof 1999, Mark and van Wachem 2008, Shen and Chan 2008), and the *distributed Lagrange multiplier method* (Glowinski et al. 1999, 2000, 2001; Patankar 2001, Yu 2005, Yu and Shao 2007). Below we briefly review these methods and some of their variants. We will pay more attention to the distributed Lagrange multiplier method as this method has gained considerable popularity in FSI simulation, especially in dealing with immersed rigid structures.

The original immersed boundary method is first order in space, and tends to suffer from leakage problem (Leveque and Li 1997, Peskin and Printz 1993), although mass conservation can be improved via the use of divergence-free finite-difference operators (Pe-

skin and Printz 1993). In part to achieve better volume conservation, Li and co-workers developed the immersed interface method (Leveque and Li 1997, Li and Lai 2001). Similar to the immersed boundary method, the FSI force is computed explicitly from the structural configuration in the immersed interface method. The boundary force is then used to derive the jump conditions in the pressure and the normal derivatives of the velocity. The immersed interface method produces second-order approximations by incorporating the known jumps in the solutions or its derivatives into the finite difference scheme, rather than using discrete delta functions as in the immersed boundary method.

However, not only is the immersed interface method limited to structures without volumes and cannot handle embedded bulk structures, the derivation of the jump conditions also requires that the immersed structures be a closed surface (in 3D) or a closed curve (in 2D). Various applications of this method have been studied (e.g., Jayathilake, Khoo and Tan 2010; Layton 2009, Li and Lai 2001; Tan, Lim and Khoo 2009); careful mathematical analysis of its accuracy has also been conducted (e.g., Beale and Layton 2006, 2009; Tornberg and Engquist 2004). For comprehensive reviews on the immersed interface method, we refer to Li (2003) and Li and Ito (2006).

The direct forcing method was developed by Mohd-Yusof (1999) to simulate fluid motion with immersed structures. By simply imposing the no-slip condition on the fluid momentum equations at the interface, this method directly evaluates the FSI force from the fluid equations with the incorporation of the known structural interfacial velocity through interpolation. The computed force, with nonzero values only near the interface, is then used to solve the fluid equations in the entire fluid domain. One advantage of this method is that it avoids the numerical stiffness usually encountered in various penalty forcing techniques (see, e.g., Goldstein et al. 1993). Guy and Hartenstine (2010) carefully analyzed the accuracy of this method. The direct forcing method can also be implemented in an implicit manner (e.g., Luo et al. 2007, Mark and van Wachem 2008), in which the FSI force and the fluidic and structural velocities are solve simultaneously through a large coupled system. In the work of Mark and van Wachem (2008), two numerical methods are proposed to overcome the difficulty of locally preserving the mass of the flow. One of them, the mirroring immersed boundary method, is the preferable choice in terms of numerical stability and efficiency. The method identifies an interior point and an exterior point near the immersed boundary which is assumed to be a closed surface, and linear interpolation is used to find the velocity of the interior point. The known velocities of the interior points are directly substituted into the fluid momentum equations to solve the exterior fluid field. The authors reported second-order spatial accuracy in the simulation of a unit sphere immersed in a fluid.

The distributed Lagrange multiplier method can be further classified into two approaches, depending upon whether the constraint condition is incorporated into the solution procedure *before* or *after* the time discretization of the FSI equations.

Once the governing equations of a FSI problem are discretized in both space and time, the results are a set of algebraic equations of the velocity and pressure, subjected to the velocity constraints that may include the divergence-free condition in the fluid

field and the Dirichlet condition along the fluid-structure interface. The constraints can be incorporated into the field equations to form an augmented matrix equation which involves the Lagrange multipliers as unknowns. Several authors have used this method to solve the FSI problems. Wachs (2007) looked for a steady state solution of a viscoplastic flow through an eccentric annular cross-section. Besides the divergence-free condition, a pointwise velocity constraint is introduced at any interior point of the immersed inner cylinder which is covered by the artificial fluid. This velocity constrained equation is viewed as the saddle-point problem which can be solved by an iterative algorithm. The key step of the method is to solve the following matrix equation

$$
\begin{bmatrix} A & M^T \\ M & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{u} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{Bmatrix}, \tag{4.19}
$$

where $A$ is a $N \times N$ symmetric positive definite matrix, $M$ is a $N \times L$ matrix, and $M\boldsymbol{u} = \boldsymbol{g}$ is the result of the velocity constraint. For better computational efficiency, it is beneficial to solve Eq. (4.19) in the following equivalent form:

$$
\begin{bmatrix} A + rM^T M & M^T \\ M & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{u} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} + rM^T \boldsymbol{g} \\ \boldsymbol{g} \end{Bmatrix}, \tag{4.20}
$$

where $r$ is the penalty coefficient, a positive scalar parameter. The above matrix system facilitates an iterative procedure based on a Uzawa/conjugate gradient algorithm. Taira and Colonius (2007) treated the FSI force as the Lagrange multipliers resulting from the no-slip constraint on the fluid-structure interface. The coupled fluid-structure equation can then be constructed based upon the Karush-Kuhn-Tucker necessary conditions (Nocedal and Wright 1999), which include the Lagrange multipliers as unknowns. The resultant discretized equation at a given time instance can be formulated in a matrix equation similar to Eq. (4.19). The authors solved the problem by means of the fractional-step/projection method.

In the second group of the distributed Lagrange multiplier approaches, the field equations are first discretized in space to form a set of ordinary equations of velocity, which are subjected to velocity constraints. Fractional time-stepping methods are frequently used in this situation to gradually adjust the field solution to satisfy the constraints. Most of these methods are first-order accurate. The works done by Glowinski et al. (1999, 2000, 2001), Patankar (2001) and Yu (2005, 2007) are the representatives of this group of techniques. These are also frequently referred to as the *fictitious domain method* in the literature.

The work of Glowinski et al. (1999, 2000, 2001) solves FSI problems with many rigid bodies moving in an incompressible flow. The repelling force between the rigid bodies due to collision is added to the equation describing the rigid body motion. The equations of the fluid domain and the rigid bodies are first presented in the weak form defined in the respective, disjoined fluid and structural domains. In this method, the fluid domain is extended to cover the rigid domain where the fluid velocity is required to be the same

as the rigid body velocity, a condition that is enforced by means of distributed Lagrange multipliers. The extension of the artificial fluid domain allows the fluid domain to span the entire computational domain, but the motion of the artificial fluid must be accounted for by the rigid body equation of motion. In the derivation, the pressure is defined in the $L^2$ space and the Lagrange multipliers are in the $H^1$ space. The later can also be discretized by the collocation method. The mesh size in the background fluid domain, $h_f$, must be similar to that in the particle domain, $h_s$; i.e., $h_f = kh_s$, for some constant $0.5 \leq k \leq 1$. The solution is advanced using a first-order accurate Marchuk-Yanenko fractional time-stepping scheme. In the first step, the Navier-Stoke equation is solved for the intermediate fluid velocity and pressure, which includes the incompressibility condition. In a second step, the Navier-Stoke equation is solved for an updated intermediate fluid velocity, which includes part of the extra stress tensor term. Then the center position of each rigid body is estimated via a solution of the rigid body motion equation, which contains the repelling force term given by a nonlinear equation of the position. After the locations of the rigid bodies is updated, the fluid velocity, the rigid structure velocity and the Lagrange multipliers are solved in the final equation that includes the rest of the fluid stress tensor term, but not the pressure and the repelling force term. Finally, the interface location is updated. Glowinski et al. (2000) used a similar four-step Marchuk-Yanenko fractional time-stepping scheme.

Patankar (2001) proposed to use sub-iterations to improve the numerical stability of an explicit time-stepping method. In his formulation, the fluid domain is extended to cover the domain of the particle, but the Dirichlet interface condition is imposed only on the fluid-particle interface. The particle is first modeled as an elastic body. The rigidity constraint is then introduced to force the strain rate in the particle to be zero. The Dirichlet interface condition, imposed only on the fluid-particle interface, is not included in the weak form of the system. In the numerical procedure, the source term is added to the fluid momentum equation of the entire domain. This source term is the divergence of the strain rate in terms of the Lagrange multipliers associated with the rigidity constraint. Based upon the fractional step method, the computed fluid velocity is first projected to a divergence-free field and then to the rigid body motion of the particles. The difference between the rigid body velocity and the divergence-free velocity yields a relation to update the source term.

Yu (2005) applied the distributed Lagrange multiplier method to solve the motion of a nonlinear elastic body immersed in a fluid domain. For small strains, the elastic body is approximately incompressible, when the incompressibility condition, $\nabla \bullet \boldsymbol{u}_f = 0$, is imposed onto the fictitious domain; thus structural displacement is not substantially affected. Continuous bilinear shape functions are used to interpolate the Lagrange multiplier in each finite element. However, the associated integration over the element is done with the trapezoidal rule in order to maintain numerical stability, because when commonly used finite element integration schemes such as the $2 \times 2$ Gaussian integration and piece-wise constant interpolation for Lagrange multipliers are used, the resultant method is unstable. In this study, the mesh size in the solid is twice of that in the fluid

domain. The mesh of the Lagrange multipliers can be even coarser. (In the work of Yu and Shao (2007), the authors investigated, in more details, the interpolation method as well as the distribution function method to project the quantities between the Eulerian mesh and the Lagrange mesh.) For time integration, a fractional step scheme is used to form a stable, first-order accurate, explicit method. Given the Lagrange multiplier $\lambda^n$ the fluid equation is first solved to yield an intermediate flow variable, $\boldsymbol{u}_f^*(\boldsymbol{x}^n)$. The structure equation is then solved to given the structural boundary location, $\boldsymbol{x}^{n+1}$. The fluid stress term that is present in the structure equation is computed from $\boldsymbol{u}_f^*(\boldsymbol{x}^n)$. Finally, $\boldsymbol{u}_f^{n+1}$ and $\lambda^{n+1}$ are computed by solving the following equations

$$\frac{\boldsymbol{u}_f^{n+1}-\boldsymbol{u}_f^*}{\Delta t}=\lambda^{n+1}-\lambda^n, \qquad \text{in } \Omega_s, \tag{4.21}$$

$$\boldsymbol{u}_f^{n+1}=\frac{\boldsymbol{x}^{n+1}-\boldsymbol{x}^n}{\Delta t}, \qquad \text{in } \Omega_s, \tag{4.22}$$

which can be viewed as the result of a saddle point problem. Eqs. (4.21) and (4.22) form an augmented Lagrange equation and can be solved iteratively by the conjugate gradient method. Through two carefully conducted numerical tests (the flow-driven oscillation and the self-sustained oscillation of a flexible plate), the paper confirms the suggestion of Glowinski et al. (1999) that the ratio of the structural mesh size and the fluid mesh size should be between one and two.

## 4.3 Mesh size and accuracy

The mesh size is an important factor in determining the stability and accuracy of the immersed methods. The choice of mesh size becomes more crucial for FSI problems involving complex interface geometry and flow physics, in which smaller mesh size or better approximation of variables around the interface are required, particularly in the case of high Reynolds and Mach numbers.

Based upon a finite element error analysis, Glowinski et al. (1999) indicated that the fluid mesh size, $h_f$, should be smaller than the structure mesh size, $h_s$, in order to maintain efficiency, while it is the other way around to maintain accuracy. To reach a compromise, they recommended that $h_f$ and $h_s$ should be on the same order. Zhao et al. (2008) studied the FSI problems associated with biological systems in which the immersed flexible body is made of neo-Hookean materials. The many examples studied in their work include the deformation of an elastic wall driven by fluid flow, an oscillating disk immersed in a fluid, the swimming of a two-dimensional jellyfish, a thin leaflet in an oscillating channel flow, and a deformable lid-driven cavity. Their numerical results suggested that the existence of the fluid-structure interface deteriorates solution accuracy. Particularly, the accuracy of the proposed FSI method achieved an accuracy between second and third order in terms of $h_s$, but only between first and second order in $h_f$. The localization of major errors near the interface was also observed in their study. Based on

these observations, the authors suggested using the adaptive mesh refinement (AMR) for better accuracy.

Several studies have incorporated the local refinement techniques into the immersed methods to yield better accuracy. Kaligzin and Iaccarino (2003) combined the immersed boundary formulation with AMR to simulate 3D high Reynolds number flows. The generation of AMR grids was carried out by first creating a fine underlying grid and then coarsening it in the region away from the immersed boundary. Simulation results of a flat plate boundary layer with Reynolds numbers as high as $Re = 10^6$ were presented in their work. Ramiere et al. (2007) mentioned that the accuracy of their method was only first order in terms of the mesh. To improve solution accuracy, they proposed a multi-level adaptive procedure to refine the mesh locally around the interface. de Tullion et al. (2007) solved several test problems of flow past rigid bodies using the immersed boundary method. A locally refined mesh was used near the interface and in the high flow-gradient region. Tai et al. (2005, 2007) introduced a densed overlapping mesh around the interface to obtain better estimate of the friction and pressure distributions on the rigid body surface. Three meshes were used: the stationary fluid mesh, the sub-domain with the overlapping mesh which is dense and wrapped around the structure, and the rigid nodes distributed within the rigid structure. A loosely coupled iterative procedure between the fluid and the structure is used in solving the FSI problem. The fluid equation with the immersed object is solved first. The fluid solution is then computed on the overlapping domain using the same Navier-Stokes solver, but with a moving grid. The "boundary" velocity of the overlapping fluid domain is determined by projecting the fluid velocity obtained from the underlying fluid domain to the neighboring overlapping mesh. The structural equations are then solved on the overlapping domain to yield detailed stress and pressure distribution on the object, which is used to find the resultant force that generates the motion of the object. A triangular mesh (not the Cartesian mesh) is used in this study. An algorithm was introduced to determine whether a fluid point is inside the overlapping or the rigid domain, or neither. The rigid objects considered in the study include a stationary cylinder, an oscillating cylinder and the bileaflets in a mechanical heart valve.

Better approximations of the fluid velocity or the FSI force on the fluid-structure interface have also been explored in several studies. Luo et al. (2007) used a nonlinear weighted average method to approximate the boundary layer velocity in the vicinity of the interface. Various techniques using ghost cells (e.g., Ghias et al. 2004, Iaccarino and Verzicco 2003) have been introduced to better approximate the interface velocity in the direct forcing method. Developing better distribution function to model interface conditions has also received attention. For example, Weymouth (2008) noted that the no-slip condition on the fluid-body interface, $\boldsymbol{u} = \boldsymbol{U}$, implies $\boldsymbol{u}_t = \boldsymbol{U}_t$. Thus, the author combined the no-slip condition with the fluid equation

$$\boldsymbol{u}_t = -\frac{1}{\rho}\nabla p + \boldsymbol{r}$$

to form a single equation,

$$\boldsymbol{u}_t = (1-\delta)\left(-\frac{1}{\rho}\nabla p + \boldsymbol{r}\right) + \delta \boldsymbol{U}_t,$$

where $\delta$ is a replacement of the Dirac delta function; $\delta(\boldsymbol{x}) = 1$ if $\boldsymbol{x}$ is in the domain of the body, and $\delta(\boldsymbol{x}) = 0$ otherwise.

Zhao et al. (2008) observed that transferring the equivalent force terms from the solid Lagrangian mesh to the underlying fluid Eulerian mesh is the critical step to achieving better solution accuracy. They proposed and tested three approaches to transfer the forces. These methods include the distributed force method, which uses the distribution function with a narrow support; the discrete momentum equation for the interface jump conditions that minimizes the truncation error; and the finite element Galerkin projection method. Among these, the distributed force method and the finite element method are relatively easy to implement. All three methods devised by the authors can maintain a sharp interface and conserve the momentum across the interface.

## 4.4   Stable time integration

Another challenge that researchers of immersed methods frequently face is that boundary forces may impose a severe restriction on time-step size in order to maintain numerical stability (Fauci and Folgelson 1993, LeVeque and Li 1997, Peskin 2002, Stockie and Wetton 1999). The numerical stability of an immersed method can be much improved if the boundary forces are treated implicitly to advance the boundary in time. Although much effort has been invested in developing implicit and semi-implicit versions of the immersed boundary method and related methods, e.g. (Tu and Peskin 1992, Mayo and Peskin 1993, Fauci and Folgelson 1993, LeVeque and Li 1997, LeVeque and Long 2003, Mori and Peskin 2008, Newren et al. 2008, Hou and Shi 2008, Hou and Shi 2008b, Ceniceros et al. 2009), it remains a challenge to develop a immersed method that is computationally efficient even for stiff boundary forces.

Owing to the coupling among fluid motion, boundary configuration, and the boundary force, the implicit or semi-implicit formulation of the immersed boundary-type methods typically requires the solution of a large system of coupled nonlinear equations via iterations, and the convergence of those iterations can be a concern. Perhaps owing to that difficulty, the majority (though by no means all) of the implementations of the immersed boundary and immersed interface methods are explicit ones. Nonetheless, there have been a number of recent studies on the development and analysis of implicit or semi-implicit formulation of immersed methods. Newren and co-workers (2008) showed that a lagged-operators semi-implicit discretization scheme, originally introduced by Peskin (1977), is unconditionally stable in its first- or second-order Crank-Nicolson form when inertia is neglected and the interfacial force is linear and self-adjoint. A variation of this scheme was studied by Mori and Peskin (2008), who proposed a fully implicit method in which the system that requires iterative solves has the same structure as the linearized

semi-implicit discretization at each iterate. Krylov subspace methods were used to solve the linear system. Also recently, Ceniceros and co-workers (2009) proposed cost-effective computational strategies for solving the linear systems arising from that semi-implicit discretization. In addition, the immersed continuum method, proposed by Wang (2006, 2007, 2010), uses fully implicit time integration and employs a matrix-free combination of Newton-Raphson and GMRES iterative solvers.

Alternatively, the stability of a method can be improved if an approximation to an implicit step for the most singular part of the velocity is used to modify an explicit method. This is the essence of the small-scale decomposition approach proposed by Hou, Lowengrub, and Shelley (1994). This approach has the advantage of not requiring the iterative solution of systems of equations. It has been applied to Stokes flow (Kopinski 2001, Hou and Shi 2008, Sohn et al. 2010), and used as a preconditioner by Veerapanei et al. (2009). Hou and Shi (2008, 2008b) developed a version of the immersed boundary method for both Stokes flow and Navier-Stokes flow using this approach with arclength-tangent angle coordinates for the interface. The small-scale decomposition approach was also used by Layton and Beale (2010), who developed a partially implicit method for Stokes flows that does not require computations in the arclength-tangent angle coordinates as in previous studies.

## 5   Discussion

The last few decades have seen a tremendous number of numerical methods developed for the simulation of FSI. The primary driving force for these developments is the demand from a wide range of scientific and engineering disciplines, where FSI problems are playing increasingly important roles. Meanwhile, the fast improvement of computational powers has made large-scale FSI simulations possible and has facilitated many realistic applications of these numerical techniques. Indeed, the numerical study of FSI has evolved into a distinct scientific field, which continues to grow and to attract enormous effort from scientists and engineers. Owing to the multidisciplinary nature of FSI problems, in this review, we have emphasized the numerical procedures used by various methods to treat the interface conditions between fluids and structures.

The first class of methods we reviewed is based on the partitioned approach which requires conforming mesh. The partitioned approach allows the fluid dynamics and structural mechanics that are involved in the FSI problems to be solved separately by their respective algorithms and codes. Since it provides flexibility in spatial meshing, the methods using the partitioned approach can conveniently catch the detailed physics along the fluid-structure interface. However, the difficulty in data handling along the fluid-structure interface and the lack of temporal convergence study hinder the use of the partitioned approach for broad FSI applications. The second class of methods we focused on are the immersed methods which use non-conforming mesh. The immersed methods, in recent years, have become increasingly popular in FSI simulations owing

to their simplicity, efficiency and flexibility, as well as their capability to handle complex flows and large structural deformations, compared to the partitioned methods with conforming meshes. Many immersed methods reviewed here are supported by rigorous temporal convergence study. A major disadvantage of these methods, however, is their lack of resolution near the interface (an exception is the immersed interface method, see below). Typically, the immersed methods smear out sharp interfaces to a thickness in the order of mesh width. This may impose a significant limitation on the immersed methods' applicability to high Reynolds number flows. Local grid refinement techniques (e.g., Kaligzin and Iaccarino 2003, Ramiere et al. 2007, Roma et al. 1999) provide a promising improvement for the immersed methods, but at the cost of increasing the complexity of the algorithm. To some extent, local grid refinement blurs the distinction between the immersed methods and those with moving meshes. The immersed interface method is one of the few immersed methods that can achieve second-order spatial accuracy and preserve sharp fluid-structure interface. A recent development of this method is the velocity decomposition approach proposed by Beale and Layton (2009) which significantly simplifies the correction terms otherwise needed to ensure second-order accuracy in Navier-Stokes flow simulation, and makes the implementation of the immersed interface method more efficient. Another progress in this method is the augmented immersed interface method (Li et al. 2007, 2010; Tan et al. 2009) which, through the introduction of appropriate augmented variables, allows the decoupling of the jump conditions so that the immersed interface formulation can be applied to flows with discontinuous viscosity. The GMRES iterative method is applied to solve the resulting system for the augmented variables which are only defined on the interface. Still, these methods are limited to structures which do not occupy volume space and which are closed curves or surfaces. The development of second or higher-order immersed method to accurately compute FSI problems with embedded bulk structures remains an open question.

To conduct a comprehensive review of the literature in this fast growing field is a daunting task, and we have realized what we presented here merely scratches the surface of the vast number of FSI methods. Nevertheless, we have attempted to assess the two classes of methods in this article, the conforming mesh and non-conforming mesh methods, for their suitability and applicability in FSI simulation. The strengths and deficiencies of these methods revealed here may help the researchers in the field to broaden their focus.

Some important works we have omitted in this review include the particle finite element method (Idelsohn et al. 2006, 2008), the ghost-cell method (Tseng and Ferziger 2003, Iaccarino and Verzicco 2003), the cut-cell method (Udaykumar et al. 1996, 1999, 2001; Ye et al. 1999), the blob projection method (Cortez and Minion 2000), the extended finite element method (Dolbow et al. 2001, Moës et al. 1999), the lattice Boltzmann method (Lallemand and Luo 2003, Owen et al. 2010), the meshfree method (Belytschko et al. 1996; Zhang, Wagner and Liu 2003), and (many) others. Interested readers may refer to those publications and the references therein.

Furthermore, the article only reviews the numerical methods and applications that

consider the interaction between immersed structures and *one fluid* (i.e., single-phase flow). An exciting area of research would be the simulation of fluid-structure interaction in multiphase flows. Notable examples of related applications include high-speed boats cruising on water, wind turbines floating in oceans, and energy buoys interacting with waves. A deeper understanding of the FSI mechanism in these applications would enable more efficient and robust design of marine crafts and energy devices that can sustain strong wave impacts, and enhance the technological development in related industry. Such FSI problems pose significant challenges to current numerical methods, as both the fluid-fluid interface and fluid-solid interface have to be accurately computed to faithfully represent the important physics involved. To our knowledge, few works have been conducted along this direction, and even fewer have considered the realistic interactions involving all the important aspects: fluid motion, structure movement and deformation, and multiphase free-surface flow. Weymouth et al. (2006, 2008) combined the volume-of-fluid technique (Hirt and Nichols 1981) and the immersed boundary formulation to simulate ship hydrodynamics with overall first-order accuracy. Shen and Chan (2008) similarly applied the combined volume-of-fluid and immersed boundary approach in several 2D case studies, including wave propagation over a submerged structure and wave generation by a moving bed. Paik (2010) incorporated the level set method (Osher and Sethian 1988) into CFDShip-Iowa, a computational ship code developed at the University of Iowa based on Reynolds-averaged Navier-Stokes (RANS) models. Most recently, Sanders et al. (2010) conducted preliminary numerical study on the rigid-body motion in 2D incompressible two-phase flows, also by incorporating the level set formulation for the free surface representation. The authors reported an order slightly above 1 for the computed terminal velocity in a simplified test where a buoyant rigid disk interacts with a single-phase channel flow. Further development of more accurate and versatile numerical methods for FSI problems with multiphase flows will benefit from interdisciplinary effort.

## Acknowledgments

### References

[1] ANSYS web page: www.ansys.com.
[2] Appa, K., Finite-Surface Spline, Journal of Aircraft, Vol. 26, No.5, 1989, pp. 495-496.
[3] Badia, S., Nobile, F., and Vergara, C., Fluid-structure partitioned procedures based on robin transmission conditions, Journal of Computational Physics, Vol. 227, 2008, pp. 7027-7051.

[4]  Bathe, K. J., Nitikitpaiboon, C. and Wang, X., A mixed displacement-based finite element formulation for acoustic fluid-structure interaction, Computers & Structures, Vol. 56, 1995, pp. 225-237.

[5]  Beale, J. T. and Layton, A. T., On the accuracy of finite difference methods for elliptic problems with interfaces, Communications in Applied Mathematics and Computational Sciences, Vol. 1, 2006, pp. 91-119.

[6]  Beale, J. T. and Layton, A. T., A velocity decomposition approach for moving interfaces in viscous fluids, Journal of Computational Physics, Vol. 228, 2009, pp. 3358-3367.

[7]  Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. and Krysl, P., Meshless methods: An overview and recent developments, Computer Methods in Applied Mechanics and Engineering, Vol. 139, 1996, pp. 3-47.

[8]  Berthelsen, P. A. and Faltinsen, O. M., A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries, Journal of Computational Physics, Vol. 227, 2008, pp. 4354-4397.

[9]  Beyer, R. P., A computational model of the cochlea using the immersed boundary methods, Journal of Computational Physics, Vol. 98, 1992, pp. 145-162.

[10]  Blake, J., Fluid mechanics of cilliary propulsion, Computational Modeling in Biological Fluid Dynamics (Eds. Fauci and Gueron), Springer-Verlag: NY, 1999.

[11]  Brown, S. A., Displacement Extrapolations for CFD+CSM Aeroelastic Analysis, AIAA-97-1090, presented at the 38th AIAA/ASME/ASCE/AHAIASC, Structures, Structural Dynamics and Materials, 1997, pp. 291-300.

[12]  Byun, C. and Guruswamy, G. P., Static Aeroelasticity Computations for Flexible 6th Wing-Body-Control Configurations, AIAA-96-4059, presented at the AlAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 4-6, 1996, pp. 744-754.

[13]  Causin, P., Gerbeau, J. F. and Nobile, F., Added-mass effect in the design of partitioned algorithms for fluid-structure problems, Computer Methods in Applied Mechanics and Engineering, Vol. 194, 2005, pp. 4506-4527.

[14]  Cebral, J. R. and Lohner, R, Conservative load projection and tracking for fluid-structure problems, AIAA Journal, Vol. 35, No.4, 1997, pp. 687-691.

[15]  Cebral, J. R and Lohner, R, Fluid-Structure Coupling: Extensions and Improvements, AIAA-97-0858, presented at the 35th Aerospace Sciences Meeting & Exhibit, Reno, NV, January 6-9, 1997.

[16]  Ceniceros, H. D., Fisher, J. E., and Roma, A. M., Efficient solutions to robust, semi-implicit discretizations of the immersed boundary method, Journal of Computational Physics, Vol. 228, 2009, pp. 7137-7158.

[17]  Chakrabarti, S. K. (Ed.), Numerical Models in Fluid Structure Interaction, Advances in Fluid Mechanics, Vol. 42, WIT Press, 2005.

[18]  Cortez, R. and Minion, M. L., The blob projection method for immersed boundary problems, Journal of Computational Physics, Vol. 161, 2000, pp. 428-453.

[19]  Degroote, J., Bruggeman, P., Haelterman, R. and Vierendeels, J., Stability of a coupling technique for partitioned solvers in FSI applications, Computers and Structures, Vol. 86, 2008, pp. 2224-2234.

[20]  de Tullion, M. D., De Palma, P., Iaccarino, G., Pascazio, G., and Napolitano, M., An immersed boundary method for compressible flows using local grid refinement, Journal of Computational Physics, Vol. 225, 2007, pp. 2098-2117.

[21]  Dillon, R., Fauci, L. and Gaver III, D., A microscale model of bacterial swimming, chemo-

taxis, and substrate transport, Journal of Theoretical Biology, Vol. 177, 1995, pp. 325-340.

[22] Dolbow, J., Moës, N. and Belytschko, T., An extended finite element method for modeling crack growth with frictional contact, Computer Methods in Applied Mechanics and Engineering, Vol. 190, 2001, pp. 6825-6846.

[23] Dowell, E. H. and Hall, K. C., Modeling of fluid-structure interaction, Annual Review of Fluid Mechanics, Vol. 33, 2001, pp. 445-490.

[24] Fadlun, E. A., Verzicco, R., Orlandi, P. and Mohd-Yusof, J., Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics, Vol. 161, 2000, pp. 35-60.

[25] Fauci, L. J. and Folgelson, A. L., Truncated newton methods and the modeling of complex immersed elastic structures, Communications in Pure and Applied Mathematics, Vol. 66, 1993, pp. 787-818.

[26] Fauci, L. and McDonald, A., Sperm mobility in the presence of boundaries, Bulletin of Mathematical Biology, Vol. 57, 1995, 679-699.

[27] Farhat, C., van der Zee, K. G. and Geuzaine, P., Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, Computer Methods in Applied Mechanics and Engineering, Vol. 195, 2006, pp. 1973-2001.

[28] Farhat, C., Lesoinne, M. and LeTallec, P, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity, Computer Methods in Applied Mechanics and Engineering, Vol. 157, 1998, pp. 95-114.

[29] Ghias, R., Mittal, R. and Lund, T. S., A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries, AIAA paper 2004-0080.

[30] Glowinski, R., Pan, T.-W., Hesla, T. I., and Joseph, D. D., A distributed Lagrange multiplier/fictitious domain method for particulate flows, International Journal of Multiphase Flow, Vol. 25, 1999, pp. 755-794.

[31] Glowinski, R., Pan, T.-W., Hesla, T. I., Joseph, D. D., and Periaux, J., A distributed lagrange multiplier/fictitious domain method for the simulation of flows around moving rigid bodies: application to particulate flow, Computer Methods in Applied Mechanics and Engineering, Vol. 184, 2000, pp. 241-268.

[32] Glowinski, R., Pan, T.-W., Hesla, T. I., Joseph, D. D., and Periaux, J., A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow, Journal of Computational Physics, Vol. 169, 2001, pp. 363-427.

[33] Goldstein, D., Handler, R. and Sirovich, L., Modeling a no-slip flow boundary with an external force field, Journal of Computational Physics, Vol. 105, 1993, pp. 354-366.

[34] Griffith, B. E., Simulating the blood-muscle-valve mechanics of the heart by an adaptive and parallel version of the immersed boundary method, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, 2005.

[35] Griffith, B. E. and Peskin, C. S., On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems, Journal of Computational Physics, Vol. 208, 2005, pp. 75-105.

[36] Grigoriadis, D. G. E., Kassinos, S. C. and Votyakov, E. V., Immersed boundary method for the MHD flows of liquid metals, Journal of Computational Physics, Vol. 228, 2009, pp. 903-920.

[37] Guruswamy, P. G. and Byun, C., Direct coupling of euler flow equations with plate finite element structures, AlAA Journal, Vol. 33, No.2, 1994, pp. 375-377.

[38] Guy, R. D. and Hartenstine, D. A., On the accuracy of direct forcing immersed boundary methods with projection methods, Journal of Computational Physics, Vol. 229, 2010, pp. 2479-2496.

[39] Haase, W., Unsteady Aerodynamics Including Fluid/Structure Interaction, Air and Space Europe, Vol. 3, 2001, pp. 83-86.

[40] Haug, E. J., Intermediate Dynamics, Prentice Hall, 1992.

[41] Hirt, C. W. and Nichols, B. D., Volume of fluid (VOF) method for dynamics of free boundaries, Journal of Computational Physics, Vol. 39, 1981, pp. 201-225.

[42] Hoburg, J. F. and Melcher, J. R., Internal electrohydrodynamic instability and mixing of fluids with orthogonal field and conductivity gradients, Journal of Fluid Mechanics, Vol. 73, 1976, pp. 333-351.

[43] Hou, G., and Satyanarayana, A., Analytical Sensitivity Analysis of a Static Aeroelastic Wing, Proceedings of 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, Sept. 2000; Also AIAA Paper 2000-4824.

[44] Hou, T. Y., Lowengrub, J. S., and Shelley, M. J., Removing the stiffness from interfacial flows with surface tension., Journal of Computational Physics, Vol. 114, 1994, pp. 312-338.

[45] Hou, T. Y. and Shi, Z., An efficient semi-implicit immersed boundary method for the Navier-Stokes equations, Journal of Computational Physics, Vol. 227, 2008, pp. 8968-8991.

[46] Hou, T. Y. and Shi, Z., Removing the stiffness of elastic force from the immersed boundary method for the 2D Stokes equations, Journal of Computational Physics, Vol. 227, 2008, pp. 9138-9169.

[47] Howe, M. S., Acoustics of Fluid-Structure Interactions, Cambridge University Press, 1998.

[48] Huang, W. X. and Sung, H. J., An immersed boundary method for fluid-flexible structure interaction, Computer Methods in Applied Mechanics and Engineering, Vol. 198, 2009, pp. 2650-2661.

[49] Hubner, B., Walhorn, E. and Dinkler, D., A monolithic approach to fluid-structure interaction using space-time finite elements, Computer Methods in Applied Mechanics and Engineering, Vol. 193, 2004, pp. 2087-2104.

[50] Iaccarino, G. and Verzicco, R., Immersed boundary technique for turbulent flow simulations, Applied Mechanics Review, Vol. 56, 2003, pp. 331-347.

[51] Idelsohn, S. R., Onate, E., Del Pin, F. and Calvo, N., Fluid-structure interaction using the particle finite element method, Computer Methods in Applied Mechanics and Engineering, Vol. 195, 2006, pp. 2100-2123.

[52] Idelsohn, S. R., Marti, J., Limache, A. and Onate, E., Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid-structure interaction problems via the PFEM, Computer Methods in Applied Mechanics and Engineering Vol. 197, 2008, pp. 1762-1776.

[53] Irons, B. and Tuck, R. C., A Version of the Aitken accelerator for computer implementation, International Journal for Numerical Methods in Engineering, Vol. 1, 1969, pp. 275-277.

[54] Jayathilake, P. G., Khoo, B. C. and Tan, Z., Effect of membrane permeability on capsule substrate adhesion: Computation using immersed interface method, Chemical Engineering Science, Vol. 65, 2010, pp. 3567-3578.

[55] Kaligzin, G. and Iaccarino, G., Toward immersed boundary simulation of high Reynolds number flows, Annual Research Briefs, Center for Turbulence Research, Stanford University, 2003, pp. 369-378.

[56] Kapania, R. K., Bhardwaj, M. K., Reichenbach, E. and Guruswamy, G. P., Aeroelastic Analysis of Modern Complex Wings, AIAA-96-38728, presented at the 6th

AlAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimiza-
tion, Bellevue, WA, September 4-6, 1996, pp. 258-265.

[57] Kim, D. and Choi, H., Immersed boundary method for flow around an arbitrarily moving
body, Journal of Computational Physics, Vol. 212, 2006, pp. 662-680.

[58] Kim, J., Kim, D. and Choi, H., An immersed-boundary finite-volume method for simula-
tions of flow in complex geometries, Journal of Computational Physics, Vol. 171, 2001, pp.
132-150.

[59] Kim, Y. and Peskin, C. S., Penalty immersed boundary method for an elastic boundary with
mass, Physics of Fluids, Vol. 19, 053103, 2007, pp. 1-18.

[60] Kropinski, M. C. A., An efficient numerical method for studying interfacial motion in two-
dimensional creeping flows, Journal of Computational Physics, Vol. 171, 2001, pp. 479-508.

[61] Lallemand, P. and Luo, L.-S., Lattice Boltzmann method for moving boundaries, Journal of
Computational Physics, Vol. 184, 2003, pp. 406-421.

[62] Layton, A. T., Using integral equations and the immersed interface method to solve im-
mersed boundary problems with stiff forces, Computer and Fluids, Vol. 38, 2009, pp. 266-
272.

[63] Layton, A. T., and Beale, J. T., A partially implicit hybrid method for computing interface
motion in Stokes flow, Discrete and Continuous Dynamical Systems B, 2010, submitted.

[64] Le, D. V., Khoo, B. C. and Lim, K. M., An implicit-forcing immersed boundary method for
simulating viscous flows in irregular domains, Computer Methods in Applied Mechanics
and Engineering, Vol. 197, 2008, pp. 2119-2130.

[65] Lee, T. R., Chang, Y. S., Choi, J. B., Kim, D. W., Liu, W. K. and Kim, Y. J., Immersed finite el-
ement method for rigid body motions in the incompressible Navier-Stokes flow, Computer
Methods in Applied Mechanics and Engineering, Vol. 197, 2008, pp. 2305-2316.

[66] Lefrançois, E. and Boufflet, J.-P., An Introduction to Fluid-Structure Interaction: Applica-
tion to the Piston Problem, SIAM Review, Vol. 52, 2010, pp. 747-767.

[67] Leveque, R. J. and Li, Z., Immersed interface method for Stokes flow with elastic bound-
aries or surface tension, SIAM Journal on Scientific Computing, Vol. 18, 1997, 709-735.

[68] Lee, L. and LeVeque, R. J., An immersed interface method for the incompressible Navier-
Stokes equations, SIAM Journal on Scientific Computing, Vol. 25, 2003, pp. 832-856.

[69] Li, Z., An overview of the immersed interface method and its applications, Taiwanese Jour-
nal of Mathematics, Vol. 7(1), 2003, pp. 1-49.

[70] Li, Z. and Ito, K., The Immersed Interface Method: Numerical Solutions of PDEs Involving
Interfaces and Irregular Domains, Society for Industrial and Applied Mathematic, 2006.

[71] Li, Z., Ito, K. and Lai, M.-C., An augmented approach for Stokes equations with a discon-
tinuous viscosity and singular forces, Computers & Fluids, Vol. 36, 2007, pp. 622-635.

[72] Li, Z. and Lai, M. C., The immersed interface method for the Navier-Stokes equations with
singular forces, Journal of Computational Physics, Vol. 171, 2001, pp. 822-842.

[73] Li, Z., Lai, M.-C., He, G. and Zhao, H., An augmented method for free boundary problems
with moving contact lines, Computers & Fluids, Vol. 39, 2010, pp. 1033-1040.

[74] Liu, W. K., Kim, D. W. and Tang, S., Mathematical foundations of the immersed finite
element method, Computational Mechanics, Vol. 39, 2006, pp. 211-222.

[75] Liu, W. K., Liu, Y., Farrell, D., Zhang, L., Wang, X., Fukui, Y., Patankar, N., Zhang, Y.,
Bajaj, C., Chen, X. and Hsu, H., Immersed finite element method and its applications to
biological systems, Computer Methods in Applied Mechanics and Engineering, Vol. 195,
2006, pp. 1722-1749.

[76] Luo, K., Wang, Z. and Fan, J., A modified immersed boundary method for simulation of

fluid-particle interactions, Computer Methods in Applied Mechanics and Engineering, Vol. 197, 2007, pp. 36-46.

[77] Mark, A. and van Wachem, B. G. M., Derivation and validation of a novel implicit second-order accurate immersed boundary method, Journal of Computational Physics, Vol. 227, 2008, pp. 6660-6680.

[78] Mayo A. and Peskin, C. S., An implicit numerical method for fluid dynamics problems with immersed elastic boundaries, in Fluid Dynamics in Biology (Ed. Cheer, A. Y. and von Dam, C. P.), pp. 261-278, Providence, RI, 1993, AMS.

[79] Mönkölä, S., Time-harmonic solution for acousto-elastic interaction with controllability and spectral elements, Journal of Computational and Applied Mathematics, Vol. 234, 2010, pp. 1904-1911.

[80] Michler, C., Hulshoff, S. J., van Brummelen, E. H. and de Borst, R., A monolithic approach to fluid-structure interaction, Computers & Fluids, Vol. 33, 2004, pp. 839-848.

[81] Mittal, R. and Iaccarino, G., Immersed boundary methods, Annual Review of Fluid Mechanics, Vol. 37, 2005, pp. 239-261.

[82] Mittal, R., Dong, H., Bozkurttas, M and Najjar, F. M., A versatile sharp interface immersed boundary method for incompressible flow with complex boundaries, Journal of Computational Physics, Vol. 227, 2008, pp. 4825-4852.

[83] Moës, N., Dolbow, J. and Belytschko, T., A finite element method for crack growth without remeshing, International Journal for Numerical Methods in Engineering, Vol. 46, 1999, pp. 131-150.

[84] Mohd-Yusof, J., Combined immersed boundary/B-spline methods for simulations of flow in complex geometries, Annual Research Briefs, Center for Turbulence Research, Stanford University, 1999, pp. 317-327.

[85] Morand, H. J.-P. and Ohayon, R., Fluid-Structure Interaction: Applied Numerical Methods, Wiley, 1995.

[86] Mori, Y. and Peskin, C. S., Implicit second order immersed boundary methods with boundary mass., Computer Methods in Applied Mechanics and Engineering, Vol. 197, 2008, pp. 2049-2067.

[87] Mucha, P. J., Tee, S. Y., Weitz, D. A., Shraiman, B. I. and Brenner, M. P., A model for velocity fluctuations in sedimentation, Journal of Fluid Mechanics, Vol. 501, 2004, pp. 71-104.

[88] Newren, E., Fogelson A., Guy R., and Kirby, M., A comparison of implicit solvers for the immersed boundary equations, Computer Methods in Applied Mechanics and Engineering, Vol. 197, 2008, pp. 2290-2304.

[89] Newman III, J. c., Newman, P. A., Taylor III, A. C. and Hou, G. J.-W., Efficient non-linear static aeroelastic wing analysis, International Journal of Computers and Fluids, Vol. 28, January 1999, pp. 615-628.

[90] Nocedal, J. and Wright, S. J., Numerical Optimization, Springer, 1999.

[91] Onishi, R, Kimura, T, Guo, Z. and Iwamiya, T, Coupled Aero-Structural Model: Approach and Application to High Aspect-Ratio Wing-Box Structures, AIAA-98-4837, presented at the 7th AIAAlUSAFINASA/ISSMO Symposium on Multidisci-plinary Analysis and Optimization, St. Louis, MO, September 2-4, 1998, pp. 1004-1010.

[92] Osher, S. and Sethian, J. A., Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics, Vol. 79, 1988, pp. 12-49.

[93] Owen, D. R. J., Leonardi, C. R. and Feng, Y. T., An efficient framework for fluid-structure interaction using the lattice Boltzmann method and immersed moving bound-

aries, International Journal for Numerical Methods in Engineering, 2010, in press (DOI: 10.1002/nme.2985).

[94] Paik, K. J., Simulation of fluid-structure interaction for surface ships with linear/nonlinear deformations, Ph.D. Thesis, The University of Iowa, 2010.

[95] Patankar, N. A., A Formulation for Fast Computations of Rigid Particulate Flows, Center for Turbulence Research, Annual Research Briefs, 2001, pp. 1-19.

[96] Peskin, C. S., Numerical analysis of blood flow in the heart, Journal of Computational Physics, Vol. 25, 1977, pp. 220-252.

[97] Peskin, C. S., The immersed boundary method, Acta Numerica, Vol. 11, 2002, pp. 479-517.

[98] Peskin, C. S. and Printz, B. F., Improved volume conservation in the computation of flows with immersed elastic boundaries, Journal of Computational Physics, Vol. 105, 1993, pp. 33-46.

[99] Ramiere, I., Angot, P. and Belliard, M., A general ficitious domain method with immersed jumps and multilevel nested structured meshes, Journal of Computational Physics, Vol. 225, 2007, pp. 1347-1387.

[100] Raveh, D. E. and Karpel, M., Structural Optimization of Flight Vehicles with Non-linear Aerodynamic Loads, AIAA-98-4832, presented at the 7th AIAA/USAF/NASA/ ISSMO Symposium ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, September 2-4, 1998, pp. 967-977.

[101] Roma, A. M., Peskin, C. S. and Berger, M. J., An adaptive version of the immersed boundary method, Journal of Computational Physics, Vol. 153, 1999, pp. 509-534.

[102] Ross, M. R., Coupling and simulation of acoustic fluid-structure interaction systems using localized Lagrange multipliers, Ph.D. Thesis, University of Colorado at Boulder, 2006.

[103] Ryzhakov, P. B., Rossi, R., Idelsohn, S. R. and Oñate, E., A monolithic Lagrangian approach for fluid-structure interaction problems, Computational Mechanics, Vol. 46, 2010, pp. 883-899.

[104] Samareh, J. A., Use of CAD Geometry in MDO, AIAA-96-3991, presented at the 6th AIAAlUSAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 4-6, 1996.

[105] Samareh, J. A., Aeroelastic Deflection of NURBS Geometry, presented at the 6th International Conference on Numerical Grid Generation in Computational Field Simulation, University of Greenwhich, Avery Hill Campus, London, UK, July 6-9, 1998, pp. 727-736.

[106] Samareh, J. A., A Novel Shape Parameterization Approach, NASA Contract No. TM-1999-209116, NASA Langley Research Center, Hampton, VA 23681-2199, May 1999.

[107] Samareh, J. A., A Survey of Shape Parameterization Techniques, presented at the AIANCEASIICASEINASA-LaRC International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA, June 22-25, 1999, pp. 333-343.

[108] Samareh, J. A., Status and future of geometry modeling and grid generation for design and optimization, Journal of Aircraft, Vol. 36, No. I, 1999, pp. 97-104.

[109] Sanders, J., Dolbow, J. E., Mucha, P. J. and Laursen, T. A., A new method for simulating rigid body motion in incompressible two-phase flow, International Journal for Numerical Methods in Fluids, 2010, in press (DOI: 10.1002/fld.2385).

[110] Shen, L. and Chan, E-S, Numerical simulation of fluid-structure interaction using a combined volume of fluid and immersed boundary method, Ocean Engineering, Vol. 35, 2008, pp. 939-952.

[111] Shyy, W., Udaykumar, H. S., Rao, M. M. and Smith, R. W., Computational Fluid Dynamics with Moving Boundaries, Dover Publications, 2007.

[112] Sohn, J. S., Tseng, Y.-H., Li, S., Voigt, A., and Lowengrub, J. S., Dynamics of multicomponent vesicles in a viscous fluid, Journal of Computational Physics, Vol. 229, 2010, pp. 119-144.

[113] Souli, M. and Benson, D. J. (Ed.), Arbitrary Lagrangian Eulerian and Fluid-Structure Interaction: Numerical Simulation, Wiley-ISTE, 2010.

[114] Stockie, J. M., and Green, S. I., Simulating the motion of flexible pulp fibres using the immersed boundary method, Journal of Computational Physics, Vol. 147, 1998, pp. 147-165.

[115] Stockie, J. M. and Wetton, B. R., Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes, Journal of Computational Physics, Vol. 154, 1999, pp. 41-64.

[116] Tai, C. H., Liew, K. M. and Zhao, Y., Numerical simulation of 3D fluid-structure interaction flow using an immersed object method with overlapping grids, Computers and Structures, Vol. 85, 2007, pp. 749-762.

[117] Tai, C. H., Zhao, Y. and Liew, K. M., Parallel computation of unsteady incompressible viscous flows around moving rigid bodies using an immersed object method with overlapping grids, Journal of Computational Physics, Vol. 207, 2005, pp. 151-172.

[118] Taira, K. and Colonius, T., The immersed boundary method: A project approach, Journal of Computational Physics, Vol. 225, 2007, pp. 2118-2137.

[119] Tan, Z., Le, D. V., Lim, K. M. and Khoo, B. C., An immersed interface method for the incompressible Navier-Stokes equations with discontinuous viscosity across the interface, SIAM Journal on Scientific Computing, Vol. 31, 2009, pp. 1798-1819.

[120] Tan, Z., Lim, K. M. and Khoo, B. C., An immersed interface method for Stokes flows with fixed/moving interfaces and rigid boundaries, Journal of Computational Physics, Vol. 228, 2009, pp. 6855-6881.

[121] Tornberg, A.-K. and Engquist, B., Numerical approximations of singular source terms in differential equations, Journal of Computational Physics, Vol. 200, 2004, pp. 462-488.

[122] Tornberg, A.-K. and Shelley, M. J., Simulating the dynamics and interactions of flexible fibers in Stokes flow, Journal of Computational Physics, Vol. 196, 2004, pp. 8-40.

[123] Tseng, Y.-H. and Ferziger, J. H., A ghost-cell immersed boundary method for flow in complex geometry, Journal of Computational Physics, Vol. 192, 2003, pp. 593-623.

[124] Tu, C. and Peskin, C. S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods, SIAM Journal on Scientific and Statistical Computing, Vol. 13, 1992, pp. 1361-1376.

[125] Tzong, G., Chen, H. H., Chang, K. C., Wu, T and Cebeci, T, A General Method for Calculating Aero-Structure Interaction on Aircraft Configurations, AIAA-96-38704, presented at the 6th AIAAlUSAFINASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, September 4-6, 1996, pp. 14-24.

[126] Udaykumar, H. S., Mittal, R. and Shyy, W., Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids, Journal of Computational Physics, Vol. 153, 1999, pp. 534-574.

[127] Udaykumar, H. S., Mittal, R., Rampunggoon, P., Khanna, A., A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, Journal of Computational Physics, Vol. 174, 2001, pp. 345-380.

[128] Udaykumar, H. S., Shyy, W. and Rao, M. M., ELAFINT: A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries, International Journal for Numerical Methods in Fluids, Vol. 22, 1996, pp. 691-705.

[129] Veerapaneni, S. K., Gueyffier, D., Zorin, D., and Biros, G., A boundary integral method

for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, Journal of Computational Physics, Vol. 228, 2009, pp. 2334-2353.

[130] Vierendeels, J., Dumont, K. and Verdonck, P. R., A partitioned strongly coupled fluid-structure interaction method to model heart valve dynamics, Journal of Computational and Applied Mathematics, Vol. 215, 2008, pp. 602-609.

[131] Wachs, A., Numerical simulation of steady bingham flow through an eccentric annular cross-section by distributed Lagrange multiplier/fictitious domain and augmented Lagrangian methods, Journal of Non-Newtonian Fluid Mechanics, Vol. 142, 2007, pp. 183-198.

[132] Wall, W. A., Gerstenberger, A., Meayer, U. and Küttler, U., Advanced approaches for fluid-shell interaction, Proceedings of the 6th International Conference on Computation of Shell and Spatial Strucutres IASS-IACM, USA, 2008, pp. CD/ F-2-C.

[133] Wang, J. and Layton, A., Numerical simulations of fiber sedimentation in Navier-Stokes flows, Communications in Computational Physics, Vol. 5, No. 1, 2009, pp. 61-83.

[134] Wang, X. S., From immersed boundary method to immersed continuum method, International Journal for Multiscale Computational Engineering, Vol. 4, No. 1, 2006, pp. 127-145.

[135] Wang, X. S., An iterative matrix-free method in implicit immersed boundary/continuum method, Computers and Structures, Vol. 85, 2007, pp. 739-748.

[136] Wang, X. S., Immersed boundary/continuum methods, in Computational Modeling in Biomechanics, De S. et al. (Eds.), Springer, 2010, pp. 3-48.

[137] Wang, X. S. and Liu, W. K., Extended immersed boundary method using FEM and RKPM, Computer Methods in Applied Mechanics and Engineering, Vol. 193, 2004, pp. 1305-1321.

[138] Wang, X. S., Zhang, L. T. and Liu, W. K., On computational issues of immersed finite element methods, Journal of Computational Physics, Vol. 228, 2009, pp. 2535-2551.

[139] Weymouth, G., Physics and Learning Based Computational Models for Breaking Bow Waves Based on New Boundary Immersion Approaches, Ph.D. Dissertation, MIT, 2008.

[140] Weymouth, G., Dommermuth, D. G., Hendrickson, K. and Yue, D. K.-P., Advances in Cartesian-grid Methods for Computational Ship Hydrodynamics, 26th Symposium on Naval Hydrodynamics, Rome, Italy, 17-22 September 2006.

[141] Wood, C., Gil, A. J., Hassan, O. and Bonet, J., Partitioned block-gauss-seidel coupling for dynamic fluid-structure interaction, Computers and Structures, Vol. 88, 2010, pp. 1367-1382.

[142] Xu, S. and Wang, Z. J., A 3D immersed interface method for fluid-solid interaction, Computer Methods in Applied Mechanics and Engineering, Vol. 197, 2008, pp. 2068-2086.

[143] Yang, J. and Balaras, E., An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, Journal of Computational Physics, Vol. 215, 2006, pp. 12-40.

[144] Ye, T., Mittal, R., Udaykumar, H. S. and Shyy, W., An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, Journal of Computational Physics, Vol. 156, 1999, pp. 209-240.

[145] Yu, Z., A DLM/FD method for fluid/flexible-body interactions, Journal of Computational Physics, Vol. 207, 2005, pp. 1-27.

[146] Yu, Z. and Shao, X., A direct-forcing fictitious domain method for particulate flows, Journal of Computational Physics, Vol. 227, 2007, pp. 292-314.

[147] Zhang, L. T., Gerstenberger, A., Wang, X. and Liu, W. K., Immersed finite element method, Computer Methods in Applied Mechanics and Engineering, Vol. 193, 2004, pp. 2051-2067.

[148] Zhang, L. T. and Gay, M., Immersed finite element method for fluid-structure interactions, Journal of Fluids and Structures, Vol. 23, No. 6, 2007, pp. 839-857.

[149] Zhang, L. T., Wagner, G. J. and Liu, W. K., Modeling and simulation of fluid structure interaction by meshfree and FEM, Communications in Numerical Methods in Engineering, Vol. 19, 2003, pp. 615-621.

[150] Zhang, W., Jiang, Y. and Ye, Z., Two better loosely coupled solution algorithms of CFD based aeroelastic simulation, Engineering Applications of Computational Fluid Mechanics, Vol. 1, No. 4, 2007, pp. 253-262.

[151] Zhao, H., Freund, J. B. and Moser, R. D., A fixed-mesh method for incompressible flow-structure systems with finite solid deformation, Journal of Computational Physics, Vol. 227, 2008, pp. 3114-3140.