Commun. Comput. Phys.
doi: 10.4208/cicp.080815.090516a

Vol. **21**, No. 1, pp. 182-210
January 2017

Computing Fluid-Structure Interaction by the Partitioned Approach with Direct Forcing

Asim Timalsina[1], Gene Hou[2] and Jin Wang[3,*]

[1] *Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529, USA.*
[2] *Department of Mechanical and Aerospace Engineering, Old Dominion University, Norfolk, VA 23529, USA.*
[3] *Department of Mathematics, University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA.*

Received 8 August 2015; Accepted (in revised version) 9 May 2016

Abstract. In this paper, we propose a new partitioned approach to compute fluid-structure interaction (FSI) by extending the original direct-forcing technique and integrating it with the immersed boundary method. The fluid and structural equations are calculated separately via their respective disciplinary algorithms, with the fluid motion solved by the immersed boundary method on a uniform Cartesian mesh and the structural motion solved by a finite element method, and their solution data only communicate at the fluid-structure interface. This computational framework is capable of handling FSI problems with sophisticated structures described by detailed constitutive laws. The proposed methods are thoroughly tested through numerical simulations involving viscous fluid flow interacting with rigid, elastic solid, and elastic thin-walled structures.

AMS subject classifications: 65Z05, 74F10

Key words: Fluid-structure interaction, immersed boundary method, partitioned approach.

1 Introduction

The interactions between viscous fluid flows and immersed solid structures are nonlinear multi-physics phenomena with application to a wide range of scientific and engineering disciplines [4, 10]. The study of fluid-structure interaction (FSI) is an emerging field that has been fast growing in recent decades. Owing to the difficulties in analysis and limitations in experiments for these strongly nonlinear problems, FSI research and development largely rely on computational methods.

*Corresponding author. *Email addresses:* atimalsi@odu.edu (A. Timalsina), ghou@odu.edu (G. Hou), Jin-Wang02@utc.edu (J. Wang)

Current numerical methods for FSI may be broadly classified in two categories: the *monolithic* approach and the *partitioned* approach. The first approach [13,23] uses a unified system to represent the entire FSI problem and employs a single algorithm to solve the fluid and structure dynamics simultaneously. The monolithic approach can potentially achieve very high accuracy in FSI simulation, but it typically demands large computational effort, and it may require substantial resources and expertise to develop and implement such an algorithm. In contrast, the partitioned approach [26,30] treats the fluid and the structure as two systems which can be computed separately with their respective solvers. Thus, fluid and structural dynamics may be solved with different mesh discretizations and numerical algorithms, and the solution data communicate at the fluid-structure interface. A significant advantage of this approach is the capability to integrate available disciplinary algorithms with respect to the fluid and structural dynamics, thus reducing the effort and time in FSI code development. A challenge, however, is to effectively coordinate the disciplinary solvers to achieve an accurate and efficient FSI solution procedure.

Another way to categorize FSI methods is based on the types of meshes employed, and there are two major classes: the *conforming mesh* methods and the *non-conforming mesh* methods. The Lagrangian methods [4, 6] and the Arbitrary Lagrangian-Eulerian methods [25] are typical examples of the conforming mesh approach, where a mesh-updating procedure is generally required at each time step corresponding to the movement/deformation of the immersed structures. On the other hand, non-conforming mesh methods employ fixed (normally Cartesian) grids which eliminate the need of re-meshing and lead to reduced algorithmic complexity and improved computational efficiency, an advantage compared to the conforming mesh methods.

The best known non-conforming mesh method for FSI is probably the immersed boundary method invented by Peskin [20]. This numerical technique solves the fluid equations with an additional forcing term which represents the effects of the immersed structure acting on the fluid motion. Essentially, the fluid equations are solved in the entire domain with a fixed Eulerian mesh, and the immersed structure is represented as a moving boundary tracked on a separate manner. Due to its efficiency, flexibility and robustness, the immersed boundary method has become increasingly popular in FSI study and many progresses, in both the methodology and application, have been made in recent years (see, e.g., [11,15,17,24,31,32]). One major limitation associated with this approach, however, is that it primarily deals with structures that do not occupy volumes; e.g., a fiber in 2D space and a membrane in 3D space. Although immersed bodies with a finite volume can be approximated by a network of connected fibers, each of which can be treated as an immersed boundary, such an approximation may not accurately model the realistic structural response to the fluid motion. Some other variations of the method, such as the immersed finite element method [32], are able to handle bulk solid structures described by material constitutive laws, yet their applications to more sophisticated structural settings are still to be seen. Additionally, in the presence of rigid or nearly rigid structures, methods of the immersed boundary type generally result in highly stiff

systems which are challenging to solve numerically.

The direct-forcing method developed by Mohd-Yusof [18] overcomes the numerical stiffness encountered by the immersed boundary method and several other penalty forcing techniques (see, e.g., [8]). The method implements the no-slip condition on the fluid momentum equations at the interface, so that the FSI force can be directly evaluated with the incorporation of the known structural interfacial velocity. Some recent work related to the direct-forcing method includes, among others, [9, 29, 34]. We note that most (if not all) direct-forcing techniques published so far are concerned with FSI problems involving rigid solid structures.

The main objective of the present paper is to combine the immersed boundary and the direct-forcing methods into a new, partitioned computational framework that enables accurate and efficient FSI computation and that allows realistic material representation of sophisticated structures. Within this framework, Cartesian grids will be used for the majority of the computation, and the fluid and structure equations will be computed separately by their respective disciplinary solvers, typically with finite differences for the fluid domain and finite elements for the structure domain. In particular, a variety of structural geometries and configurations can be investigated and well-developed structural algorithms with detailed constitutive laws can be easily incorporated. The fluid and structural solutions are connected at the fluid-structure interface through a procedure based on an extension of the direct-forcing approach. These algorithms are thoroughly tested by a number of numerical simulations involving FSI problems with different material types such as rigid, elastic solid, and elastic thin-walled structures.

The remainder of this paper is organized as follows. We first present the mathematical formulation and numerical treatment of the fluid and structural systems, respectively, in Sections 2 and 3. Then we describe in detail the integration of the fluid and structural solutions; i.e., the FSI coupling procedure, in Section 4. We conduct careful numerical simulations in Section 5 to demonstrate our computational methods. Finally, we conclude the paper with some discussion in Section 6.

2 Fluid system

We first describe the dynamics of the fluid part in our FSI problem. Below we present the governing equations of the fluid motion and the numerical treatment. We refer to [22] and references therein for a large body of work on computational fluid dynamics, particularly for solving Navier-Stokes equations. We also list in Appendix A the nomenclature used in this paper.

2.1 Equations of motion

We consider a two-dimensional (2D) fluid domain, Ω^f, with an immersed solid domain, Ω^s (see Fig. 1). Let Γ^s denote the fluid-structure interface. Let $[0,T]$ denote the time domain. The flow is assumed to be viscous and incompressible, and the motion is described

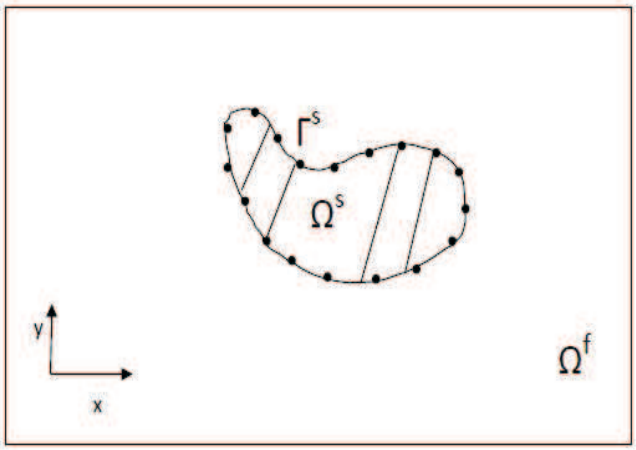


Figure 1: Fluid-structure domain.

by the standard Navier-Stokes equations

$$\rho^f\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)+\nabla p-\mu\nabla^2\mathbf{u}=\mathbf{f} \quad \text{in} \quad \Omega^f\times[0,T], \tag{2.1a}$$

$$\nabla\cdot\mathbf{u}=\mathbf{0} \quad \text{in} \quad \Omega^f\times[0,T], \tag{2.1b}$$

$$\mathbf{u}|_{t=0}=\mathbf{u_0} \quad \text{in} \quad \Omega^f, \tag{2.1c}$$

where $\mathbf{u}$ and p denote the fluid velocity and pressure, ρ^f and μ denote the fluid density and dynamic viscosity, respectively, and $\mathbf{f}$ is the external force. At the fluid-structure interface, we have the no-slip condition

$$\mathbf{u}=\mathbf{U}^s \quad \text{on} \quad \Gamma^s. \tag{2.1d}$$

In addition, we assume periodic boundary conditions on both the x and y directions of the fluid domain. Below we briefly describe our numerical procedure for solving the fluid equations.

2.2 Numerical calculation

For ease of notations, we introduce the central difference operators on the Eulerian grids. Let $\mathbf{e}_1$, $\mathbf{e}_2$ be the standard basis of $\mathbb{R}^2$. For any function $\varphi(\mathbf{x})$, $\mathbf{x}\in\Omega^f$, we define

$$(D_\beta\varphi)(\mathbf{x})=\frac{\varphi(\mathbf{x}+h\mathbf{e}_\beta)-\varphi(\mathbf{x}-h\mathbf{e}_\beta)}{2h}, \qquad \beta=1,2; \tag{2.2}$$

$$(L\varphi)(\mathbf{x})=\sum_{\beta=1}^{2}\frac{\varphi(\mathbf{x}+h\mathbf{e}_\beta)+\varphi(\mathbf{x}-h\mathbf{e}_\beta)-2\varphi(\mathbf{x})}{h^2}, \tag{2.3}$$

where $h=\Delta x=\Delta y$. Let $\mathbf{D}=(D_1,D_2)$. The differential operators in the Navier-Stokes equations are then discretized as follows:

$$\nabla\cdot\varphi\approx\mathbf{D}\cdot\varphi, \tag{2.4a}$$

$$\nabla\varphi\approx\mathbf{D}\varphi, \tag{2.4b}$$

$$\nabla^2\phi\approx L\phi. \tag{2.4c}$$

Since $\nabla\cdot\mathbf{u}=\mathbf{0}$, we have,

$$\mathbf{u}\cdot\nabla\varphi=\nabla\cdot(\mathbf{u}\varphi). \tag{2.4d}$$

To take advantage of (2.4d), a skew symmetric difference operator is employed to approximate the convective term $\mathbf{u}\cdot\nabla\mathbf{u}$:

$$(S(\mathbf{u})\mathbf{u})=\frac{1}{2}\mathbf{u}\cdot\nabla\mathbf{u}+\frac{1}{2}\nabla\cdot(\mathbf{u}\mathbf{u})\sim\frac{1}{2}\mathbf{u}\cdot\mathbf{D}\mathbf{u}+\frac{1}{2}\mathbf{D}\cdot(\mathbf{u}\mathbf{u}). \tag{2.4e}$$

As a result, equations (1a)(1b) can be spatially discretized as follows:

$$\rho^f\left(\frac{\partial\mathbf{u}}{\partial t}+S(\mathbf{u})\mathbf{u}\right)+\mathbf{D}p-\mu L\mathbf{u}=\mathbf{f}, \tag{2.5a}$$

$$\mathbf{D}\cdot\mathbf{u}=\mathbf{0}. \tag{2.5b}$$

For temporal discretization, implicit methods can be applied to Eqs. (2.5a) and (2.5b) to ensure time-stepping stability. However, to treat the nonlinear terms, an iterative procedure is usually necessary. To reduce computational effort, fractional-step methods are widely used to calculate the incompressible Navier-Stokes equations. Here we adopt the fractional-step method presented in [20].

Assume that the solution is known at time t_n. At the preliminary (or, predictor) step, we solve for the intermediate variables $\mathbf{u}^{n+\frac{1}{2}}$ and $p^{n+\frac{1}{2}}$ by

$$\rho^f\left(\frac{\mathbf{u}^{n+\frac{1}{2}}-\mathbf{u}^n}{\Delta t/2}+S(\mathbf{u}^n)\mathbf{u}^n\right)+\mathbf{D}p^{n+\frac{1}{2}}-\mu L\mathbf{u}^{n+\frac{1}{2}}=\mathbf{f}^{n+\frac{1}{2}}, \tag{2.6a}$$

$$\mathbf{D}\cdot\mathbf{u}^{n+\frac{1}{2}}=\mathbf{0}. \tag{2.6b}$$

Then we update the solution at the corrector step; i.e., solve for $\mathbf{u}^{n+1}$ and $\mathbf{p}^{n+1}$, using

$$\rho^f\left(\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}+S(\mathbf{u}^{n+\frac{1}{2}})\mathbf{u}^{n+\frac{1}{2}}\right)+\mathbf{D}p^{n+1}-\mu L\left(\frac{\mathbf{u}^n+\mathbf{u}^{n+1}}{2}\right)=\mathbf{f}^{n+\frac{1}{2}}, \tag{2.6c}$$

$$\mathbf{D}\cdot\mathbf{u}^{n+1}=\mathbf{0}. \tag{2.6d}$$

To solve the system of equations (2.6), a Fast Fourier Transform is implemented [21]. In both the steps above, we need to solve linear systems of the form

$$\left(1-\frac{\Delta t}{2}\frac{\mu}{\rho^f}L\right)\mathbf{u}+\frac{\Delta t}{2\rho^f}\mathbf{D}p=\mathbf{w}, \tag{2.7a}$$

$$\mathbf{D}\cdot\mathbf{u}=0, \tag{2.7b}$$

where $\mathbf{w}$ refers to all the known terms, and $\mathbf{u}$ and p refer to the unknown velocity and pressure terms. The Discrete Fourier Transform of the above equations are given by (with $\widehat{\ }$ representing the Fourier space):

$$\left(1-\frac{\Delta t}{2}\frac{\mu}{\rho^f}\widehat{L}\right)\widehat{\mathbf{u}}+\frac{\Delta t}{2\rho^f}\widehat{\mathbf{D}}\widehat{p}=\widehat{\mathbf{w}}, \tag{2.8a}$$

$$\widehat{\mathbf{D}}\cdot\widehat{\mathbf{u}}=0. \tag{2.8b}$$

Once solved, $\widehat{\mathbf{u}}$ and $\widehat{p}$ are transferred back to the physical space to obtain the solution $\mathbf{u}$ and p at the given time level.

3 Structure system

We now present the structural part of the FSI problem. Although the computational framework proposed in this paper can be applied to solid structures with various constitutive laws (e.g., linear elastic, hyperelastic, and viscoelastic), we will, for illustration, focus our attention on structures described by a linear elastic model. In what follows, we will describe the equations of motion and numerical treatment of the structural dynamics. As a primary emphasis of our work is to use disciplinary algorithms and codes to solve fluid and structural equations separately, we will employ the standard notations in solid mechanics when presenting the structure system below. We refer to [14] for a detailed explanation of equations and notations in solid mechanics.

3.1 Equations of motion

We consider a linear isotropic elastic beam. Since the strains on the elastic beam are small, we use the linear strain model to describe the deformations. In such an elastic domain Ω^s, the strain-displacement equation, the stress-strain relation and the dynamic equations of equilibrium state are given as follows:

$$\boldsymbol{\varepsilon}=\boldsymbol{\partial}\mathbf{W}, \tag{3.1a}$$

$$\boldsymbol{\sigma}=\mathbf{D_e}\boldsymbol{\varepsilon}, \tag{3.1b}$$

$$\boldsymbol{\partial}^T\boldsymbol{\sigma}+\bar{\mathbf{b}}=\rho^s\ddot{\mathbf{W}}, \tag{3.1c}$$

where $\boldsymbol{\varepsilon}$ is the strain, $\boldsymbol{\sigma}$ is the stress, $\mathbf{D_e}$ is the material stiffness matrix, ρ^s is the density of the structure, $\bar{\mathbf{b}}$ is the body force, $\mathbf{W}$ is the displacement, and $\dot{\mathbf{W}}$ and $\ddot{\mathbf{W}}$, respectively, denote the first and second time derivatives of $\mathbf{W}$. Specifically,

$$\boldsymbol{\sigma}=\begin{pmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{pmatrix}=\begin{pmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{12}\end{pmatrix},\quad \boldsymbol{\epsilon}=\begin{pmatrix}\epsilon_x\\ \epsilon_y\\ \gamma_{xy}\end{pmatrix}=\begin{pmatrix}\epsilon_{11}\\ \epsilon_{22}\\ 2\epsilon_{12}\end{pmatrix},\quad \mathbf{W}=\begin{pmatrix}W_1\\ W_2\end{pmatrix},\quad \bar{\mathbf{b}}=\begin{pmatrix}\bar{b}_1\\ \bar{b}_2\end{pmatrix},$$

$$\boldsymbol{\partial}=\begin{pmatrix}\frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y}\end{pmatrix}, \quad \text{and} \quad \mathbf{D_e}=\frac{E}{1-\nu^2}\begin{pmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{pmatrix},$$

where E and ν are Young's modulus and Poisson's ratio, respectively.

We write the boundary of the structure as $\Gamma^s=\Gamma^d\cup\Gamma^t$, where Γ^d refers to the fixed part of the boundary and Γ^t refers to the movable part. The boundary conditions are

$$\mathbf{W}=\mathbf{W}_0 \quad \text{on} \quad \Gamma^d, \tag{3.1d}$$

$$\mathbf{n}\sigma=\mathbf{T}_0 \quad \text{on} \quad \Gamma^t, \tag{3.1e}$$

where

$$\mathbf{n}=\begin{pmatrix}n_1 & 0 & n_2\\ 0 & n_2 & n_1\end{pmatrix}$$

and (n_1,n_2) is the outward unit normal to the boundary. The initial displacement and velocity are given by

$$\mathbf{W}(\mathbf{x},\mathbf{0})=\mathbf{g}_1, \tag{3.1f}$$

$$\dot{\mathbf{W}}(\mathbf{x},0)=\mathbf{g}_2. \tag{3.1g}$$

These equations are then solved using the finite element method (see Appendix B for the detailed formulation). The following matrix equation is eventually obtained:

$$M\ddot{\underline{\mathbf{W}}}+K\underline{\mathbf{W}}-\mathbf{f}_t-\mathbf{f}_b=0. \tag{3.2}$$

The dynamic response $\underline{\mathbf{W}}$, the matrices M and K, and the force terms $\mathbf{f}_t$ and $\mathbf{f}_b$ are defined in Appendix B.

3.2 Numerical calculation

We rewrite the dynamic equation (3.2) in the form as

$$M\ddot{\underline{\mathbf{W}}}+K\underline{\mathbf{W}}-\mathbf{f}_b=\mathbf{F}(t), \tag{3.3}$$

where $\mathbf{F}(t)$ is emphasized as the force load acting on the boundary of the structure. Eq. (3.3) is solved based upon the modal synthesis approach in which the displacement vector is approximated by modal superposition. That is, the dynamic response $\underline{\mathbf{W}}$ is expanded as a linear combination of eigenvectors with weighted coefficients $\mathbf{a}(t)$; i.e.

$$\underline{\mathbf{W}}=\boldsymbol{\psi}\mathbf{a}(t). \tag{3.4}$$

The columns of the matrix $\boldsymbol{\psi}$ are the eigenvectors of the eigenvalue problem pertaining to the equation of motion,

$$K\boldsymbol{\psi}=\Lambda M\boldsymbol{\psi}, \tag{3.5}$$

where the eigenvectors $\boldsymbol{\psi}$ satisfy the orthonormal conditions $\boldsymbol{\psi}^T K\boldsymbol{\psi}=\Lambda$ and $\boldsymbol{\psi}^T M\boldsymbol{\psi}=\mathbf{I}$, and Λ is the diagonal matrix consisting of the eigenvalues of the problem.

Substituting (3.4) in (3.3), we obtain

$$M\boldsymbol{\psi}\ddot{\mathbf{a}}(t)+K\boldsymbol{\psi}\mathbf{a}(t)-\mathbf{f}_b=\mathbf{F}(t). \tag{3.6}$$

Multiplying the above equation by $\boldsymbol{\psi}^T$, we obtain

$$\boldsymbol{\psi}^T M\boldsymbol{\psi}\ddot{\mathbf{a}}(t)+\boldsymbol{\psi}^T K\boldsymbol{\psi}\mathbf{a}(t)-\boldsymbol{\psi}^T\mathbf{f}_b=\boldsymbol{\psi}^T\mathbf{F}(t). \tag{3.7}$$

The dynamic equation can then be decoupled into a set of ordinary differential equations as

$$\ddot{\mathbf{a}}(t)+\Lambda\mathbf{a}(t)-\boldsymbol{\psi}^T\mathbf{f}_b=\boldsymbol{\psi}^T\mathbf{F}(t). \tag{3.8}$$

In this study, a one-step implicit Newmark method [1] is used to solve equation (3.8). It can be summarized as follows:

Assume that $\mathbf{a}_{n-1}$, $\dot{\mathbf{a}}_{n-1}$, $\ddot{\mathbf{a}}_{n-1}$ at time t_{n-1} are known. The Newmark method extrapolates $\mathbf{a}_n$ and $\dot{\mathbf{a}}_n$, at time $t_n=t_{n-1}+\Delta t$ in terms of $\ddot{\mathbf{a}}_n$ as

$$\dot{\mathbf{a}}_n=\dot{\mathbf{a}}_{n-1}+\Delta t[(1-\alpha)\ddot{\mathbf{a}}_{n-1}+\alpha\ddot{\mathbf{a}}_n], \tag{3.9a}$$

$$\mathbf{a}_n=\mathbf{a}_{n-1}+\Delta t\dot{\mathbf{a}}_{n-1}+\Delta t^2\left[\left(\frac{1}{2}-\beta\right)\ddot{\mathbf{a}}_{n-1}+\beta\ddot{\mathbf{a}}_n\right]. \tag{3.9b}$$

The above approximation corresponds to the linear acceleration method if $\alpha=1/2$ and $\beta=1/6$, and the constant average acceleration method if $\alpha=1/2$ and $\beta=1/4$. We use the latter one which is unconditionally stable.

At time t_n, equation (3.8) becomes,

$$\ddot{\mathbf{a}}_n+\Lambda\mathbf{a}_n-\boldsymbol{\psi}^T\mathbf{f}_{b,n}=\boldsymbol{\psi}^T\mathbf{F}_n. \tag{3.10}$$

Substituting equations (3.9) into equation (3.10), we obtain

$$\ddot{\mathbf{a}}_n=(I+\Lambda\beta\Delta t^2)^{-1}\left[\boldsymbol{\psi}^T(\mathbf{F}_n+\mathbf{f}_{b,n})-\Lambda\left(\mathbf{a}_{n-1}+\Delta t\dot{\mathbf{a}}_{n-1}+\Delta t^2\left(\frac{1}{2}-\beta\right)\ddot{\mathbf{a}}_{n-1}\right)\right]. \tag{3.11}$$

Once $\ddot{\mathbf{a}}_n$ is found, equations(3.9) can be used to find $\mathbf{a}_n$ and $\dot{\mathbf{a}}_n$. The displacement and velocity at the finite element nodes are then computed by

$$\underline{\mathbf{W}}_n=\boldsymbol{\psi}\mathbf{a}_n, \tag{3.12a}$$

$$\dot{\underline{\mathbf{W}}}_n=\boldsymbol{\psi}\dot{\mathbf{a}}_n. \tag{3.12b}$$

The stresses at each node can be calculated based on Eqs. (3.12) and (B.3) (see Appendices B and C for details). The magnitude of the stresses is often used to measure the strength of the structure, and is an important factor for structure design consideration. The displacement $\underline{\mathbf{W}}_n$ at the boundary reveals the deformed position of the structure. The velocity $\dot{\underline{\mathbf{W}}}_n$ at the boundary; i.e., $\mathbf{U}^s$, is then passed to the fluid code.

4 Fluid-structure interaction

As discussed above, our computational framework allows that the fluid and structural motions are solved separately by their disciplinary methods, and their solutions now need to be connected at the interface where the fluid-structure interaction takes place. Below we describe the details of our numerical technique to link the fluid and structural dynamics using the direct-forcing approach.

4.1 Calculation of FSI forces

The original direct-forcing technique by Mohd-Yusof was implemented in Cartesian grids where the force was calculated directly from the Navier-Stokes equations [18]. Alternatively, the force can be evaluated at the Lagrangian grid points, as suggested by Uhlmann [29]. We adopt the latter approach in the current study. Let $\mathbf{U}^n$ and P^n denote, respectively, the fluid velocity and pressure evaluated on the Lagrangian grids. Our forcing term $\mathbf{F}^{n+1}$ takes the form

$$\mathbf{F}^{n+1}=\begin{cases}\rho^f\left[\dfrac{\mathbf{U^s}-\mathbf{U}^n}{\Delta t}+((\mathbf{U}\cdot\nabla)\mathbf{U})^n\right]+\nabla P^n-\mu\nabla^2\mathbf{U}^n & \text{on } \Gamma^s,\\ \mathbf{0} & \text{elsewhere.}\end{cases} \tag{4.1}$$

The force $\mathbf{F}$ is passed to the structure code acting as the force load on the right-hand side of Eq. (3.3). Once Eq. (3.3) is solved, the structural displacement and velocity fields, particularly the interfacial velocity U^s, are updated. Meanwhile, the force $\mathbf{F}$ is interpolated to the Eulerian grid so as to solve the Navier-Stokes equations and update the fluid motion. The interpolation is done through a discrete delta function, described below. Once the interpolation is completed, the forcing term $\mathbf{f}$ on the right-hand side of the fluid momentum equations is obtained.

4.2 Interpolation functions

A 2D discrete delta function, denoted δ_h, is typically given by the product of two 1D discrete delta functions,

$$\delta_h=\frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right),$$

where $h=\Delta x=\Delta y$. Here we use the function ϕ introduced by Peskin [20]:

$$\phi(x)=\frac{1}{8}\begin{cases}3-2|x|+\sqrt{1+4|x|-4x^2} & :|x|\le 1,\\ 5-2|x|-\sqrt{-7+12|x|-4x^2} & :1\le|x|\le 2,\\ 0 & :|x|\ge 2.\end{cases}$$

The distribution of forces from Lagrange to Euler grids and the interpolation of velocities from Euler to Lagrange grids are then represented as follows [29]:

$$\mathbf{f}(\mathbf{x}_{ij}) = \sum_l \mathbf{F}(\mathbf{X}_l) \cdot \delta_h(\mathbf{x}_{ij} - \mathbf{X}_l) \Delta V_l, \tag{4.2a}$$

$$\mathbf{U}(\mathbf{X}_l) = \sum_{i,j} \mathbf{u}(\mathbf{x}_{ij}) \cdot \delta_h(\mathbf{x}_{ij} - \mathbf{X}_l) h^2, \tag{4.2b}$$

where ΔV_l is the discrete volume at each Lagrangian point. Interpolation/distribution of other quantities can be done in a similar way.

4.3 Fluid-structure coupling

In a typical computational procedure for fluid-structure interaction based on the immersed boundary approach, the fluid and structure parts are solved sequentially (see, e.g., [20, 24]). Such an algorithm generally contains the following basic steps:

- Calculate the FSI forces at the fluid-structure interface.
- Distribute the forces from the interface to the background Eulerian grids.
- Solve the Navier-Stokes equations on the Eulerian grids with the forcing term.
- Interpolate the fluid velocity from the background Eulerian grids to the interfacial Lagrangian grids.
- Advance the structural positions (particularly at the interface) with the updated velocity.

Our numerical approach will integrate the immersed boundary method and the direct-forcing technique into a new, partitioned framework, based on which well-developed disciplinary methods can be employed to solve the fluid and structural motions and their solution data only communicate at the fluid-solid interface. This approach will extend the immersed boundary method to the computation of more realistic solid structures that occupy finite volumes, and will extend the direct-forcing technique toward dealing with more general, elastic materials described by detailed constitutive laws. Key steps of our FSI algorithm are summarized below, where the emphasis is put on the fluid-structure coupling. We assume that at $t = t_{n+1}$, the fluid velocity u^n and pressure p^n (defined on the Eulerian grids) and the structural boundary position $\Gamma^{s,n}$ are known. We again let $\mathbf{U}^n$ and P^n denote the fluid velocity and pressure evaluated on the Lagrangian grids at the time t_n.

1. Use the known information on u^n, p^n and $\Gamma^{s,n}$ to compute $\mathbf{U}^n$, $((\mathbf{U}\cdot\nabla)\mathbf{U})^n$, $\nabla^2\mathbf{U}^n$, and ∇P^n on the structural boundary through interpolation from the Eulerian grids to the Lagrangian grids.

2. Calculate the forces $\mathbf{F}^{n+1}$ on the Lagrangian grids using the direct-forcing approach (see Eq. (4.1)).
3. Distribute the forces from the interface to the fluid domain to obtain $\mathbf{f}^{n+1}$.
4. Compute the Navier-Stokes equation (e.g., by the fractional-step method; see Eq. (2.6)). Also calculate $\mathbf{u}^{n+1}$, $((\mathbf{u}\cdot\nabla)\mathbf{u})^{n+1}$, $\nabla^2\mathbf{u}^{n+1}$, and ∇p^{n+1} for the next time step.
5. Pass the force $-\mathbf{F}^{n+1}$ as the load to the structural solver.
6. Solve the structural system at $t=t_{n+1}$; particularly, obtain the boundary displacement $\mathbf{W}^{s,n+1}$ and velocity $\mathbf{U}^{s,n+1}$.
7. Update the fluid-structure interface with the updated structural displacement obtained from the structure code; e.g., $\Gamma^{s,n+1}=\Gamma^{s,n}+\mathbf{W}^{s,n+1}$.
8. Pass the updated fluid velocity $\mathbf{u}^{n+1}$ and pressure p^{n+1}, interface location $\Gamma^{s,n+1}$ and structural velocity $\mathbf{U}^{s,n+1}$ to the next step of computation.

5 Examples

We present three examples to demonstrate our numerical algorithms. The first one is the canonical problem of viscous flow past a rigid circular cylinder. We will compare our numerical findings with published results. The second example is a more complex problem with viscous flow past an elastic and deformable solid beam. We will pay special attention to the movement and deformation of the solid structure under the impact of fluid flow. The third example, computationally most challenging, is concerned with viscous flow past an elastic and deformable hallow beam, where complex interaction occurs between the fluid flow and the thin-walled structure.

5.1 Flow past a rigid cylinder

We first verify our method for the well-known problem of 2D viscous flow past a rigid cylinder. In this case, though, the structure solver is not needed; Γ^s remains unchanged and $\mathbf{U}^s=0$. Under the standard setting, we compare the drag coefficient and the length of the eddies to those in the literature. The results are presented in Table 1. The drag coefficient in 2D is calculated using the formula,

$$C_D=\frac{2F_x}{\rho u^2 D}, \tag{5.1}$$

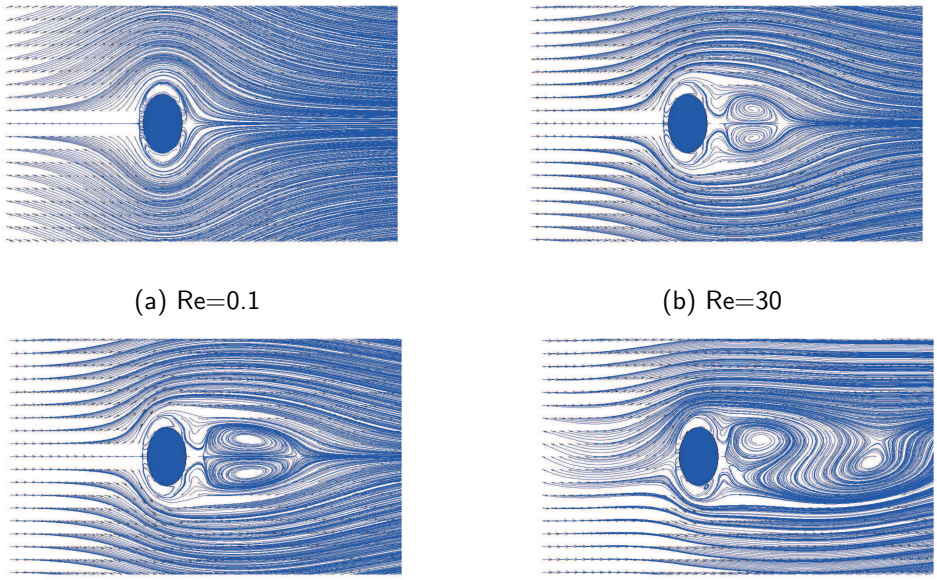
where F_x is the x component of the fluid force acting on the cylinder, and D is the diameter of the cylinder. The Reynolds number Re is given by

$$\mathrm{Re}=\frac{\rho^f u D}{\mu}.$$

Table 1: Comparison of the wake lengths and drag coefficients.

Re=20	L	C_D
Fornberg [7]	0.91	2
Dennis and Chang [3]	0.94	2.05
Donna Calhoun [2]	0.94	2.05
Taira and Colonius [28]	0.97	2.07
Present	0.9	2

Re=40	L	C_D
Fornberg [7]	2.24	1.50
Dennis and Chang [3]	2.35	1.52
Donna Calhoun [2]	2.18	1.62
Taira and Colonius [28]	2.33	1.55
Present	2.36	1.54



(a) Re=0.1 (b) Re=30

(c) Re=40 (d) Re=100

Figure 2: Streamline visualization with different Reynolds numbers for the flow past a rigid cylinder.

We observe good agreement between our numerical measurement and published results. Meanwhile, Fig. 2 shows the streamline plots of the flow with different Reynolds numbers. We clearly observe three distinct types of flow behaviors: when Re is very low, the flow is symmetric and steady, and no wakes form in the tail region; when Re is in an intermediate range (typically between 5 and 40), a pair of vortices in wakes are observed and the length of the eddies increases with Re; when Re grows even larger (e.g., 100), the motion becomes asymmetric and unsteady, and the wakes evolve into vortex streets.

5.2 Flow past an elastic solid beam

In this problem, a 30×3 (in^2) elastic beam is immersed in a viscous fluid. The bottom of the beam is fixed, whereas other parts of the beam are movable and deformable. The parameters characterizing material properties of the beam are listed in Table 2. We con-

Table 2: Material parameters for the elastic beam.

Young's modulus	Poisson's ratio	density	thickness
$3.0E+09$ psi	0.3	284 lb/in^3	0.125 in

Table 3: Eigenvalues and frequencies of the ten modes of the solid beam.

Mode	Eigenvalue, λ	Frequency
1	1.19E+03	5.5
2	4.30E+04	33
3	2.90E+05	85.8
4	2.98E+05	86.9
5	9.85E+05	158
6	2.29E+06	241
7	2.61E+06	257
8	4.34E+06	332
9	7.21E+06	427
10	7.21E+06	427

sider ten mode shapes of the elastic beam to form the eigenvector matrix $\boldsymbol{\psi}$ in Eq. (3.4); these ten modes are visualized in Fig. 3. The corresponding eigenvalues and frequencies of each mode are listed in Table 3. The dynamic responses of the beam is subject to the transient force $\mathbf{F}(t)$ induced by the surrounding fluid.

The beam is discretized into 720 triangular elements with 427 nodes. Starting from the lower left corner, we index the structure boundary from 1 to 127 (61 points on each of the left and right sides and 5 on the top); see Fig. 4. These 127 nodes are used to communicate data between the structure code and the fluid code. Initially, the structure is at rest, and the fluid motion is set as parallel flow with a constant velocity $\mathbf{u_0}$. The fluid solver calculates $\mathbf{F}$ using the Direct Forcing Approach and passes $\mathbf{F}$ to the structure code; the structural motion is computed using the Newmark method [1]. It then returns the structural velocity $\mathbf{U}^s$ at the boundary nodes(127 in total) to the fluid code.

Through the interaction with the fluid flow, the beam oscillates on the horizontal direction (left and right). Note again that the bottom of the beam is fixed. In Fig. 5, we highlight the vibration and deformation of the beam over time. The computational domain here is set as $[0,50]\times[0,50]$ which, we have found, is large enough so that the fluid boundary does not play a role in the structural motion. The other parameters for the simulation are $\mu=0.1$, $\mathbf{u}_0=50$, $\rho^f=1.5$, $\Delta q/h=0.64$, and $\Delta t=0.001$. As can be naturally expected, the beam has a larger displacement in the direction of the flow and a smaller displacement in the opposite way.

Fig. 6 illustrates the case of larger deformations with a more flexible body, where the Young's modulus E is decreased by 100 times; i.e., $E=3\times10^7$ psi, to make the beam softer and more deformable.

In Fig. 7, we visualize the flow field and the structural stresses at different times.

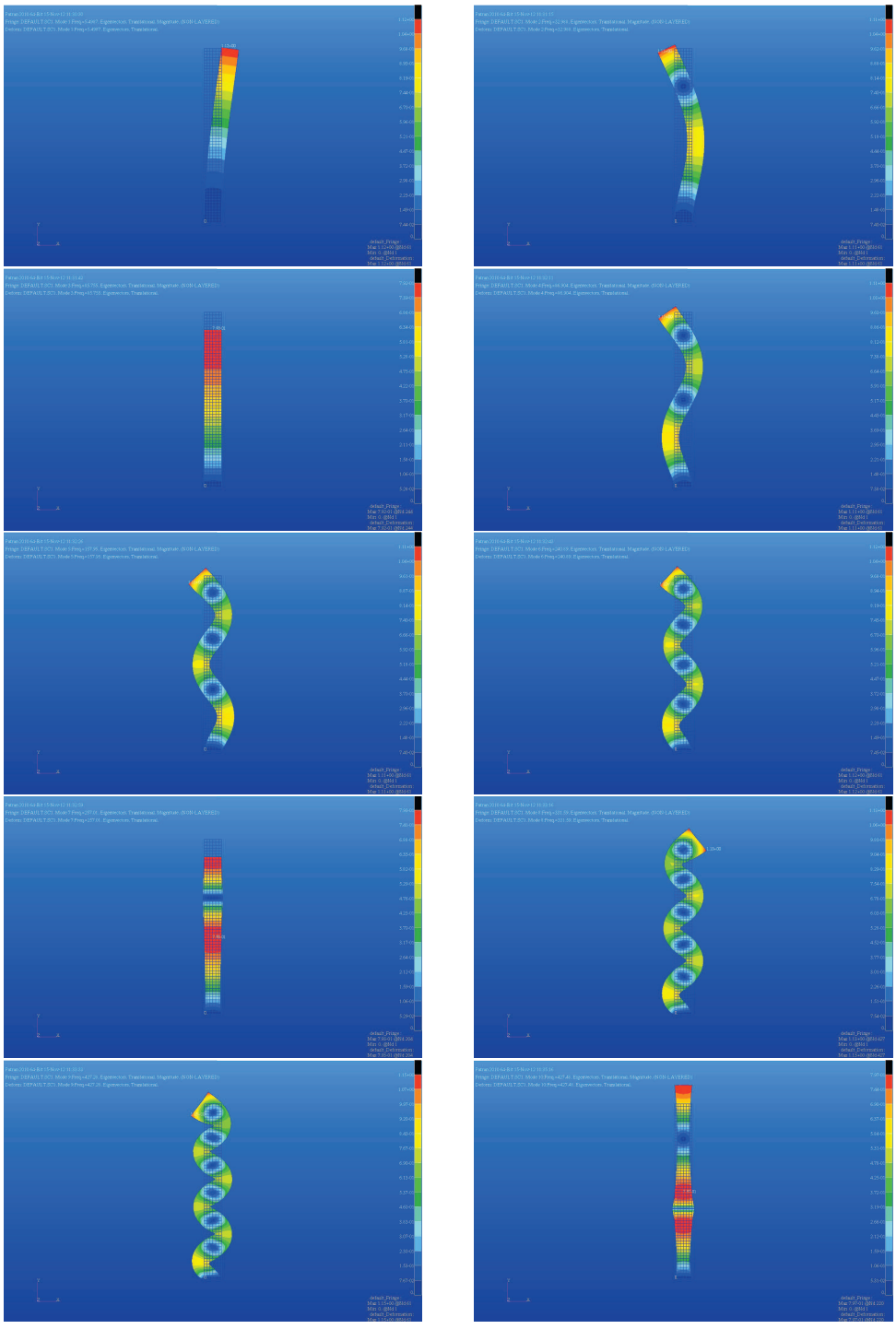


Figure 3: Ten mode shapes of the elastic solid beam.

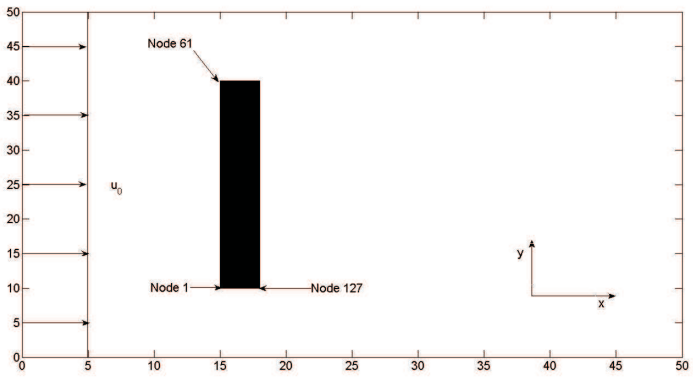


Figure 4: Fluid-structure interaction diagram of an elastic beam.

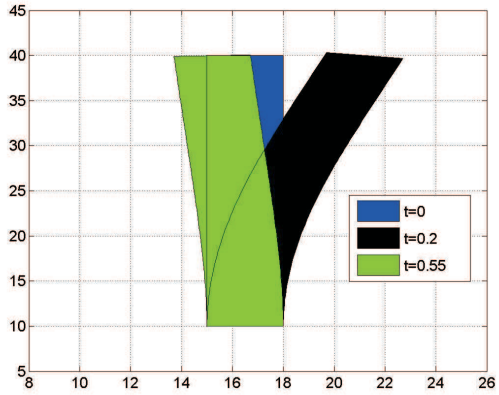


Figure 5: Movement of the solid beam.

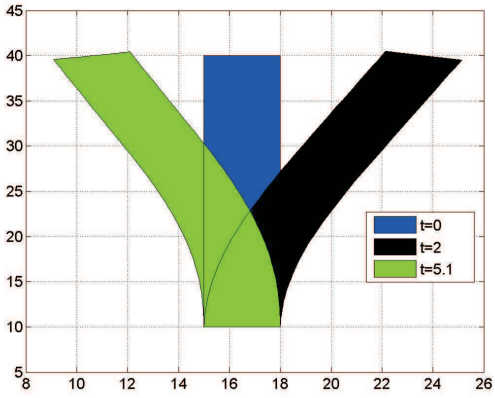
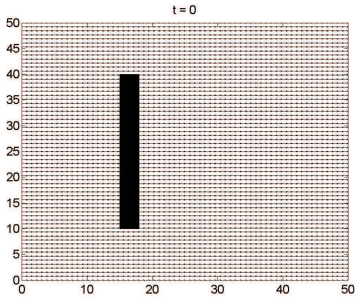
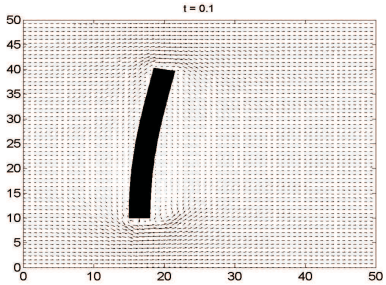


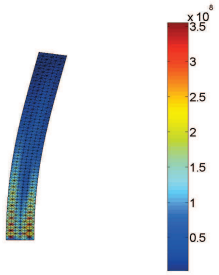
Figure 6: Larger displacement with a more elastic body.



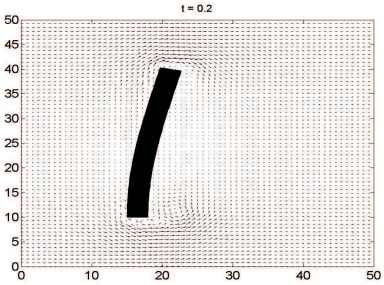
(a) flow field at $t=0$



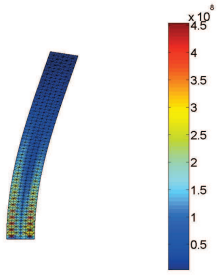
(b) flow field at $t=0.1$



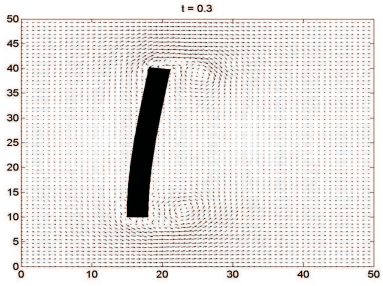
(c) stress plot at $t=0.1$



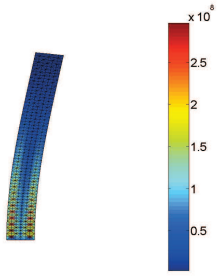
(d) flow field at $t=0.2$



(e) stress plot at $t=0.2$



(f) flow field at $t=0.3$



(g) stress plot at $t=0.3$

Figure 7: The solid beam with fluid flow and its stress plots.

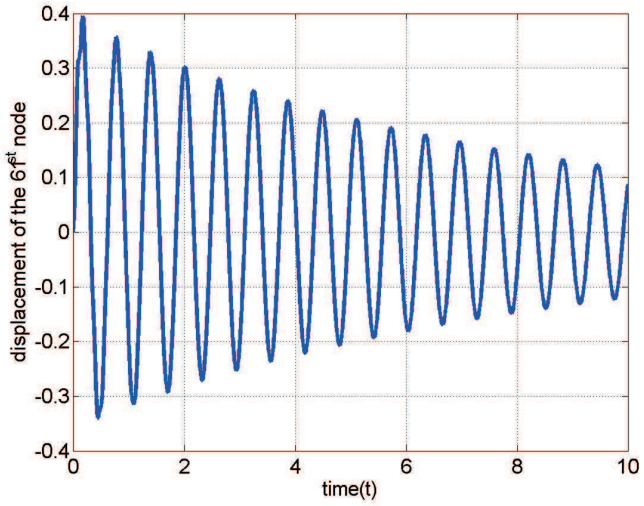


Figure 8: Tip displacement of the solid beam in the x-direction.

The structural stresses are obtained based upon Eq. (C.2) described in Appendix C. We observe that a vortex is formed near the fixed end of the beam shortly after the flow starts, and another vortex is created near the other end later as the beam starts coming back to the original position. In order to examine the dynamics more carefully, we pick a particular tip of the beam at the free end; i.e., node 61 for the upper left corner of the beam, and look into its detailed motion. Fig. 8 shows the displacement of this node over a long period of time, where we clearly observe a regular oscillation with decaying amplitude over time. Moreover, the oscillation in the positive direction (i.e., the direction of the fluid flow) has larger amplitude than that in the negative (i.e., opposite) direction. The fluid parameters used in this simulation are $\mu = 0.1$, $\mathbf{u}_0 = 10$, and $\rho^f = 0.5$. Fig. 9 compares the displacement of the 61^{st} node over one cycle for different choices of Δt, and we observe close agreement of these curves.

As a means to verify our methods and results, we have also conducted the numerical simulation using a different approach proposed by Zhang et al. [34]. This method calculates the FSI force $\mathbf{F}$ based on a predictor-corrector procedure using an intermediate fluid velocity. The method originally employs stream functions, and we have generated a variant using the velocity-pressure formulation which seems more suitable for our simulation and comparison.

For distinction, we refer to the algorithm originated from [34] as Method 2, and the algorithm described in Section 4.3 as Method 1. We have applied these two methods in our numerical simulation of the elastic solid beam problem, and their results match each other very well. In particular, Fig. 10 shows the displacement of the 61^{st} node using the two methods with the same parameters. We observe excellent agreement between the two numerical solutions.

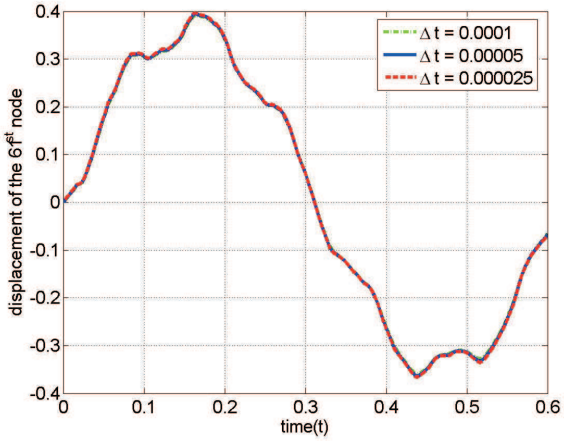


Figure 9: Tip displacement of the solid beam with different Δt.

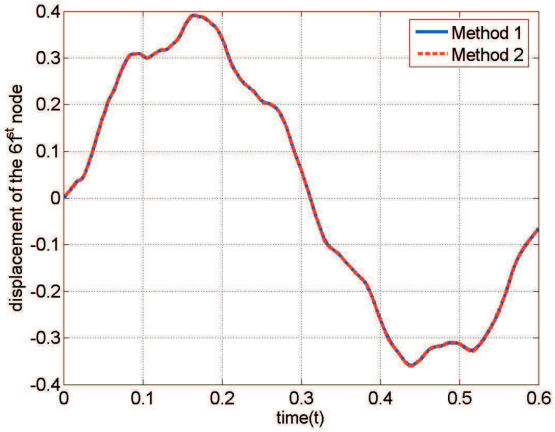


Figure 10: Comparison of the two numerical methods for the solid beam.

5.3 Flow past an elastic hollow beam

In this example, we investigate the more challenging FSI problem of an immersed elastic hollow beam interacting with a viscous fluid flow. We consider the same set-up as shown in Fig. 4, except that the solid beam is now replaced by a hollow beam which has an outside dimension of 30×3 (in^2) and a 28×1 (in^2) rectangular hole in the middle. The material properties of the beam remain the same as those in the previous example and the key parameters are listed in Table 2. We again consider ten mode shapes of the hollow beam (see Fig. 11) to form $\boldsymbol{\psi}$ in Eq. (3.4). The corresponding eigenvalues and frequencies of each mode are listed in Table 4. The inside part of the beam is hollow and the beam is closed on all sides, so that the fluid cannot pass inside. The entire computational domain

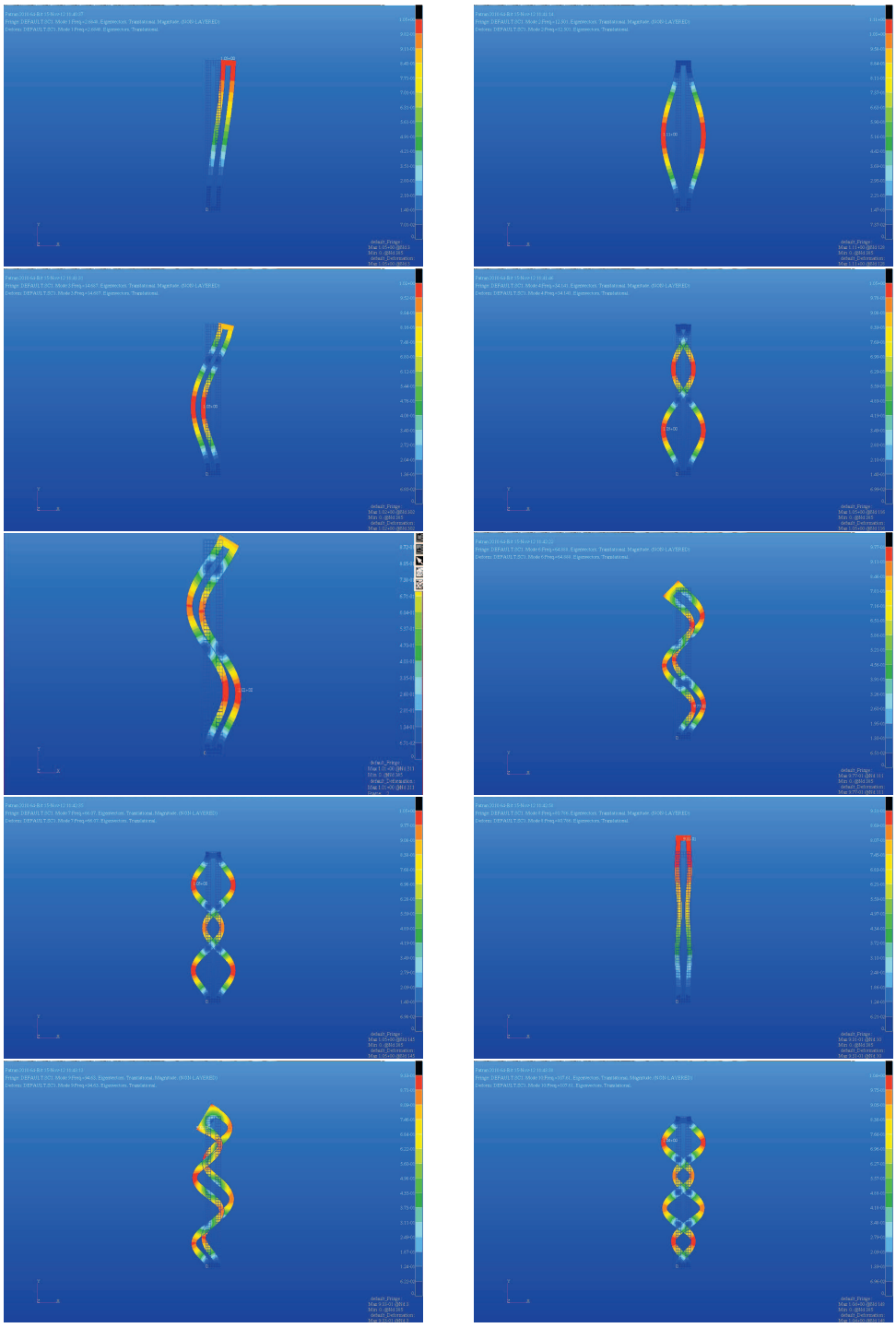


Figure 11: Ten Mode Shapes of the hollow beam.

Table 4: Eigenvalues and frequencies of the ten modes of the hollow beam.

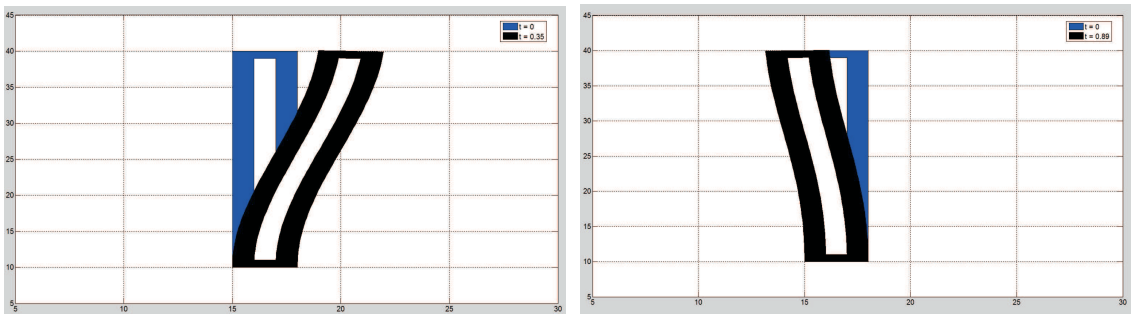
Mode	Eigenvalue, λ	Frequency
1	2.85E+02	2.68
2	6.17E+03	12.50
3	8.52E+03	14.69
4	4.60E+04	34.14
5	4.98E+04	35.50
6	1.66E+05	64.89
7	1.72E+05	66.07
8	2.58E+05	80.79
9	3.54E+05	94.63
10	4.57E+05	107.61

remains as $[0,50]\times[0,50]$. The hollow beam is discretized into 496 triangular elements with 372 nodes. We again use the 61^{st} node in reference to the top left corner of the hollow beam.

We have conducted similar simulation tasks as those in the previous example, and results show that our methods are able to accurately track the fluid - hollow beam interaction and resolve the detailed movement and deformation of the structure. In particular, Fig. 12 illustrates the motion of the beam at different times. Similar to the solid case, the hollow beam exhibits a larger displacement in the direction of the flow and a smaller displacement in the opposite direction.

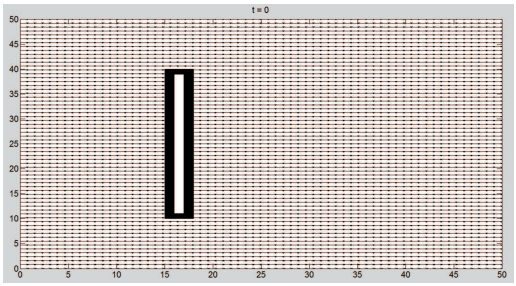
Fig. 13 shows the movement of the structure and the fluid together, and the structural stresses at different times. We observe similar pattern of vortex formation as in Example 5.2.

Fig. 14 shows the horizontal displacement of the 61^{st} node of the hollow beam over a long period of time, with a regular oscillation of decaying amplitude. The material and numerical parameters are the same as those in the previous example. Compared to

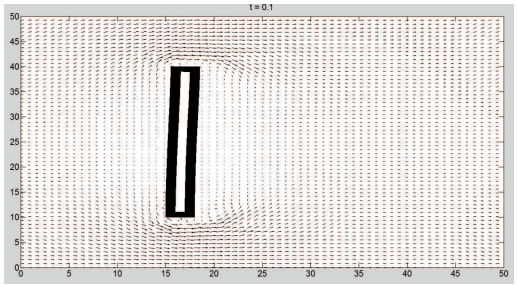


(a) (b)

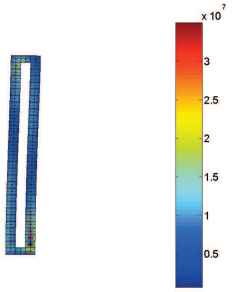
Figure 12: Movement of the hollow beam.



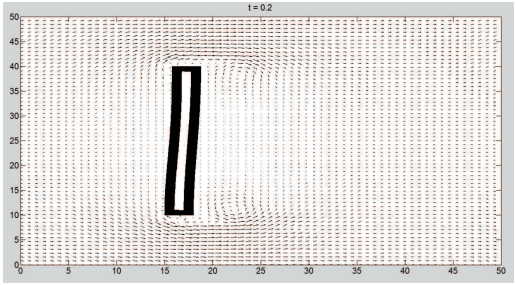
(a) $t=0$



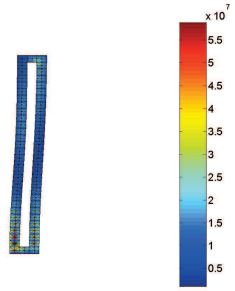
(b) $t=0.1$



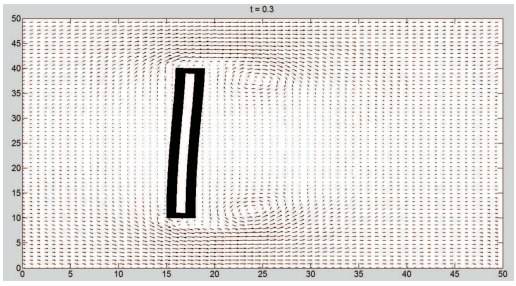
(c) $t=0.1$



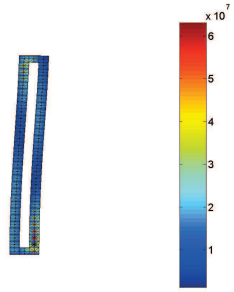
(d) $t=0.2$



(e) $t=0.2$



(f) $t=0.3$



(g) $t=0.3$

Figure 13: The hollow beam with fluid flow and its stress plots.

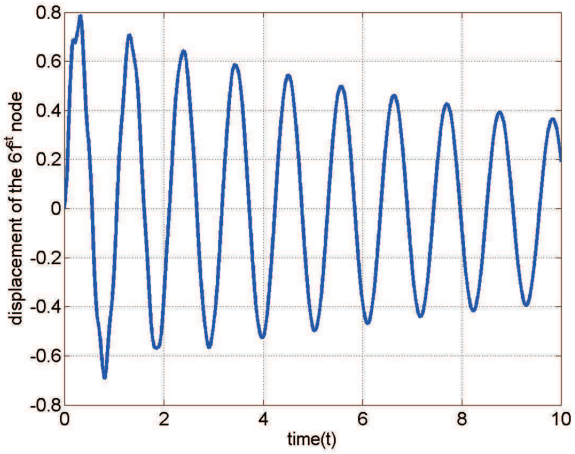


Figure 14: Tip displacement of the hollow beam in the x-direction.

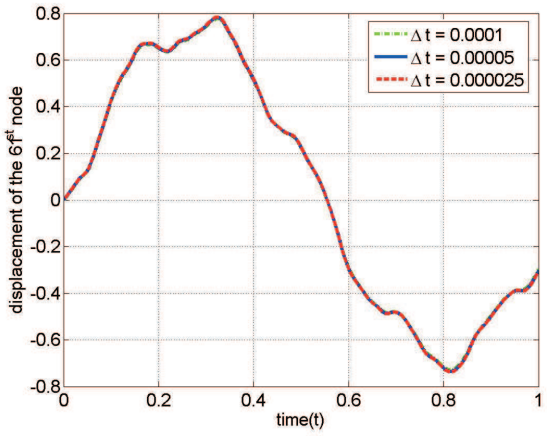


Figure 15: Tip displacement of the hollow beam for different Δt.

the solid case (see Fig. 8), the tip of the hollow beam exhibits larger displacements, in both positive and negative directions, as well as larger periods (or, smaller frequencies). Fig. 15 compares the tip displacement of the 61^{st} node over one cycle for different Δt, where we see again the three curves closely match each other. Fig. 16 shows the results of the displacement of the 61^{st} node using the two numerical approaches described before; again very good agreement is observed for these two results.

Finally, we compare the displacement of the 61st node between the hollow beam and the solid beam using the same material and numerical parameters, and the result is displayed in Fig. 17. As is clearly shown, the solid beam vibrates faster than the hollow beam, whereas the deformation of the hollow beam is larger than that of the solid beam.

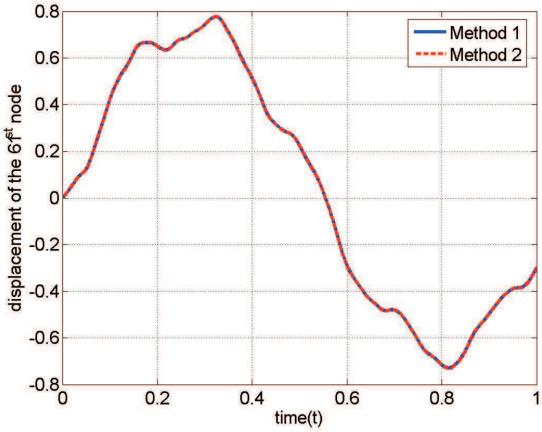


Figure 16: Comparison of the two numerical methods for the hollow beam.

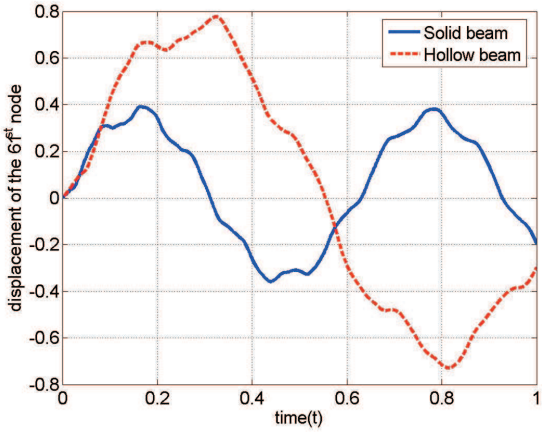


Figure 17: Comparison of the tip displacement for the solid and hollow beams with the same material and numerical parameters.

6 Conclusion

We have presented a new framework to compute fluid-structure interactions by integrating the immersed boundary techniques and the direct-forcing method. Through a partitioned approach, our algorithms allow well-developed disciplinary methods and codes to solve the respective fluid and structural equations, whose solutions are connected at the fluid-structure interface by communicating forces and velocities. Unlike most of the existing FSI methods which are restricted to relatively simple structural settings, our framework makes it possible to explore highly challenging FSI problems that involve complex fluid motion and sophisticated structural shapes and configurations. We have

demonstrated our methods through careful numerical simulations that involve viscous fluid flow interacting with rigid, elastic solid, and elastic thin-walled, structures. Results clearly show that our numerical approach is capable of handling various structural types and resolving the detailed movement and deformation of the structures when interacting with fluid flow. Our work generalizes the original immersed boundary method to accommodate structures that occupy finite volumes and that are represented by realistic material constitutive laws. Meanwhile, our approach extends the conventional direct-forcing technique to the simulation of FSI problems involving elastic structures. To our knowledge, these represent a first-of-its-kind advance in FSI computational study.

The advantages of the partitioned approach lie in its flexibility and portability in FSI simulation. Each of the fluid and structure systems can be handled independently with different solvers and meshes, and details of the solution procedure for one system can be easily modified without affecting the other party. These features make the foundation to employ disciplinary solvers that have been well developed, rigorously validated, and constantly updated, in the FSI computation for attacking the potentially most sophisticated problems in the field. The typical challenge in the implementation of the partitioned approach, however, is the coupling of the fluid and structural solvers for an accurate representation of the interaction at the fluid-structure interface. In this work, we have applied numerical techniques originated from the immersed boundary and the direct-forcing methods to link the fluid and structural solutions for an accurate, stable and efficient FSI computational procedure.

FSI problems are inherently interdisciplinary, and the present paper strongly emphasizes interdisciplinary collaboration in advancing the study in this fast-growing scientific field. Our work provides a platform that can naturally assemble researchers in the fluid and structure dynamics fields to collaborate on FSI simulation and analysis. Future development of this work would involve exploration of more accurate and efficient techniques for coupling the fluid and structural solutions, and extension of the methodology from 2D space to 3D space.

Acknowledgments

This work was partially supported by the National Science Foundation under Grant Numbers 1216936 and 1319078, and by the Tennessee Higher Education Commission through a CEACSE grant. The authors are grateful to the two referees for their valuable comments to improve this paper.

Appendix

A Nomenclature

We list here all the symbols used throughout this paper.

Symbol	Description
Ω^f	fluid domain
Ω^s	structure domain
$\mathbf{u}$	fluid velocity
$\mathbf{U}^s$	structure velocity
p	pressure
μ	dynamic viscosity
ρ^f	fluid density
ρ^s	structural density
C_D	drag coefficient
T	final time
h	Eulerian grid size($\Delta x = \Delta y$)
Δq	Lagrangian grid size
$\mathbf{W}$	structural displacement
Γ^s	fluid-structure interface
$\mathbf{D_e}$	elastic material stiffness matrix
σ	stress components
ϵ	strain components
$\bar{\mathbf{b}}$	body forces
E	Young's modulus
ν	Poisson's ratio

B Finite element approximation

The equations given in (3.1) are solved using the finite element method. The weak form is given by

$$\int_{\Omega^s} \sigma^T \delta\varepsilon \mathrm{d}v = \int_{\Gamma^t} \mathbf{T}_0^T \delta\mathbf{W} \mathrm{d}s + \int_{\Omega^s} \bar{\mathbf{b}}^T \delta\mathbf{W} \mathrm{d}v - \int_{\Omega^s} \rho^s \ddot{\mathbf{W}}^T \delta\mathbf{W} \mathrm{d}v, \tag{B.1}$$

where $\delta\mathbf{W}$ and $\delta\varepsilon$ are arbitrary virtual displacement and strain fields respectively. The finite element approximation estimates the displacement vector by a linear combination of shape functions as

$$\mathbf{W}(\mathbf{x},t) = \mathbf{N}(\mathbf{x})\underline{\mathbf{W}}(t) \quad \text{and} \tag{B.2a}$$

$$\ddot{\mathbf{W}}(\mathbf{x},t) = \mathbf{N}(\mathbf{x})\underline{\ddot{\mathbf{W}}}(t). \tag{B.2b}$$

Consequently, one has

$$\varepsilon = \partial\mathbf{W} = (\partial\mathbf{N})\underline{\mathbf{W}} = \mathbf{B}(\mathbf{x})\underline{\mathbf{W}}, \tag{B.3a}$$

where $\mathbf{B} = \partial\mathbf{N}$ is the strain displacement matrix. And

$$\sigma = \mathbf{D_e}\varepsilon = \mathbf{D_e}\mathbf{B}(\mathbf{x})\underline{\mathbf{W}}. \tag{B.3b}$$

Therefore, the virtual displacement and the virtual strain field are given by

$$\delta\mathbf{W} = \mathbf{N}\delta\underline{\mathbf{W}}, \tag{B.4a}$$

$$\delta\boldsymbol{\varepsilon} = \mathbf{B}\delta\underline{\mathbf{W}}. \tag{B.4b}$$

Substituting Eqs. (B.3) and (B.4) in (B.1), we obtain the discretized weak form,

$$\begin{aligned}
0 =& \int_{\Omega^s} \rho^s \ddot{\mathbf{W}}^T \delta\mathbf{W} \mathrm{d}v + \int_{\Omega^s} \boldsymbol{\sigma}^T \delta\boldsymbol{\varepsilon} \mathrm{d}v - \int_{\Gamma^t} \mathbf{T}_0^T \delta\mathbf{W} \mathrm{d}s - \int_{\Omega^s} \bar{\mathbf{b}}^T \delta\mathbf{W} \mathrm{d}v \\
=& \ddot{\underline{\mathbf{W}}}^T \int_{\Omega^s} \mathbf{N}^T\mathbf{N}\delta\underline{\mathbf{W}} \mathrm{d}v + \underline{\mathbf{W}}^T \int_{\Omega^s} \mathbf{B}^T\mathbf{D_e}\mathbf{B}\delta\underline{\mathbf{W}} \mathrm{d}v - \int_{\Gamma^t} \mathbf{T}_0^T\mathbf{N}\delta\underline{\mathbf{W}} \mathrm{d}s - \int_{\Omega^s} \bar{\mathbf{b}}^T\mathbf{N}\delta\underline{\mathbf{W}} \mathrm{d}v \\
=& (\ddot{\underline{\mathbf{W}}}^T M^T + \underline{\mathbf{W}}^T K^T - \mathbf{f}_t^T - \mathbf{f}_b^T)\delta\underline{\mathbf{W}} \\
=& \delta\underline{\mathbf{W}}^T (M\ddot{\underline{\mathbf{W}}} + K\underline{\mathbf{W}} - \mathbf{f}_t - \mathbf{f}_b).
\end{aligned} \tag{B.5}$$

The derivation above is done by defining its integrals in terms of the mass matrix, the stiffness matrix, the surface force term and the body force term, M, K, $\mathbf{f}_t$ and $\mathbf{f}_b$, respectively, as

$$\begin{aligned}
&\int_{\Omega^s} \mathbf{N}^T\mathbf{N} \mathrm{d}v = M^T, \\
&\int_{\Omega^s} \mathbf{B}^T\mathbf{D_e}\mathbf{B} \mathrm{d}v = K^T, \\
&\int_{\Gamma^t} \mathbf{T}_0^T\mathbf{N} \mathrm{d}s = \mathbf{f}_t^T, \\
&\int_{V} \bar{\mathbf{b}}^T\mathbf{N} \mathrm{d}v = \mathbf{f}_b^T.
\end{aligned}$$

Eq. (B.5) results in a matrix equation for the motion:

$$M\ddot{\underline{\mathbf{W}}} + K\underline{\mathbf{W}} - \mathbf{f}_t - \mathbf{f}_b = 0 \tag{B.6}$$

with consistent boundary and initial conditions.

C Results of dynamic stresses

The values of the von Mises stresses are often required as an output of an analysis to measure the capability of the structure failure to sustain the given load in engineering design. This section briefs the procedure to calculate the required stresses in the framework presented in Section 3.2.

Once the time history of the elastic displacement is obtained, we can then proceed to compute the dynamic stresses. This can be done using the known mode shapes of the given structure $\boldsymbol{\psi}$ and the modal coordinates $\mathbf{a}$ from the solution of Eq. (3.9b). The modal superposition can be applied here to find the total stresses, $\boldsymbol{\sigma}^T = \begin{pmatrix} \sigma_x & \sigma_y & \tau_{xy} \end{pmatrix}$. The

combination of Eqs. (3.4) and (B.2a) describes the displacement field in the structure in terms of modal coordinates as

$$\mathbf{W}(\mathbf{x},t)=\mathbf{N}(\mathbf{x})\boldsymbol{\psi}\mathbf{a}(t).$$

The state of stresses at any point, $\mathbf{x}$, in the structure is then obtained as, according to Eqs. (3.1a) and (3.1b),

$$\begin{aligned}\boldsymbol{\sigma}(\mathbf{x},t)&=\mathbf{D_e}\boldsymbol{\varepsilon}(\mathbf{x})\\&=\mathbf{D_e}\boldsymbol{\partial}\mathbf{W}(\mathbf{x})\\&=\mathbf{D_e}(\boldsymbol{\partial}\mathbf{N}(\mathbf{x}))\boldsymbol{\psi}\mathbf{a}(t).\end{aligned}$$

Since the material properties $\mathbf{D_e}$, the differential operator $\boldsymbol{\partial}$, the finite element interpolation function and the mode shapes are known before starting the FSI solution process, the relationship between the stresses $\boldsymbol{\sigma}$ and the modal coordinates can be defined as,

$$\begin{aligned}\boldsymbol{\sigma}(\mathbf{x},t)&=[\mathbf{D_e}(\boldsymbol{\partial}\mathbf{N}(\mathbf{x}))\boldsymbol{\psi}]\mathbf{a}(t)\\&=\mathbf{B}(\mathbf{x})\mathbf{a}(t).\end{aligned}\tag{C.2}$$

Once the normal stresses σ_x, σ_y and the shear stress τ_{xy} are known, one can then use the following equation to find the von Mises stress

$$\sigma_{v,c}=\sqrt{\sigma_x^2+\sigma_y^2-\sigma_x\sigma_y+3\tau_{xy}^2}.$$

Collection of the von Mises stresses at the centers of all elements can be used to find the von Mises stress distribution over the entire finite element model.

References

[1] K.J. Bathe. *Finite Element Procedures*, pp. 780-782. Prentice-Hall, Englewood CliVs, New Jersey, 1996.

[2] D. Calhoun. A cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *Journal of Computational Physics*, 176(2):231–275, 2002.

[3] S.C.R. Dennis and G.Z. Chang. Numerical solutions for steady flow past a circular cylinder at reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(03):471–489, 1970.

[4] E.H. Dowell and K.C. Hall. Modeling of fluid-structure interaction. *Annual Review of Fluid Mechanics*, 33:445–490, 2001.

[5] E.A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60, 2000.

[6] C. Farhat, M. Lesoinne, and P. LeTallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.

[7] B. Fornberg. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics*, 98(04):819–855, 1980.

[8] D. Goldstein, R. Handler, and L. Sirovich. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105:354–366, 1993.
[9] R.D. Guy and D.A. Hartenstine. On the accuracy of direct forcing immersed boundary methods with projection methods. *Journal of Computational Physics*, 229:2479–2496, 2010.
[10] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction - A review. *Communication in Computational Physics*, 12(2):337–377, 2012.
[11] W.X. Huang, S.J. Shin, and H.J. Sung. Simulation of flexible filaments in a uniform flow by the immersed boundary method. *Journal of Computational Physics*, 226(2):2206–2228, 2007.
[12] W.X. Huang and H.J. Sung. An immersed boundary method for fluid–flexible structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 198(33):2650–2661, 2009.
[13] B. Hubner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193:2087–2104, 2004.
[14] M. Jirásek. Basic concepts and equations of solid mechanics. *Revue Européenne de Génie Civil*, 11(7-8):879–892, 2007.
[15] Z. Li. An overview of the immersed interface method and its applications. *Taiwanese Journal of Mathematics*, 7:1–49, 2003.
[16] H. Luo, H. Dai, P. Ferreira de Sousa, and B. Yin. On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries. *Computers & Fluids*, 56:61–76, 2012.
[17] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.
[18] J. Mohd-Yusof. Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries. *Annual Research Briefs. NASA Ames Research Center, Stanford University Center of Turbulence Research: Stanford*, pp. 317–327, 1997.
[19] D.Z. Noor, M.J. Chern, and T.L. Horng. An immersed boundary method to solve fluid–solid interaction problems. *Computational Mechanics*, 44(4):447–453, 2009.
[20] C.S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479-517, 2002.
[21] C.S. Peskin. The immersed boundary method in a simple special case, September 2007. Lecture notes. Retrieved on April, 14, 2013.
[22] R.H. Pletcher, J.C. Tannehill, and D.A. Anderson. *Computational Fluid Mechanics and Heat Transfer, Third Edition*. CRC Press, 2011.
[23] P.B. Ryzhakov, R. Rossi, S.R. Idelsohn, and E. Onate. A monolithic Lagrangian approach for fluid-structure interaction problems. *Computational Mechanics*, 46:883–899, 2010.
[24] F. Sotiropoulos and X. Yang. Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, 2013.
[25] M. Souli and D.J. Benson (Eds.). *Arbitrary Lagrangian Eulerian and Fluid-Structure Interaction: Numerical Simulation*. Wiley-ISTE, 2010.
[26] J. Steindorf and H.G. Matthies. Numerical efficiency of different partitioned methods for fluid-structure interaction. *ZAMM - Journal of Applied Mathematics and Mechanics*, 80(S2):557–558, 2000.
[27] S.W. Su, M.C. Lai, and C.A. Lin. An immersed boundary technique for simulating complex flows with rigid boundary. *Computers & Fluids*, 36(2):313–324, 2007.
[28] K. Taira and T. Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.
[29] M. Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005.

[30] J. Vierendeelsa, K. Dumontb, and P.R. Verdonckb. A partitioned strongly coupled fluid-structure interaction method to model heart valve dynamics. *Journal of Computational and Applied Mathematics*, 215:602–609, 2008.
[31] J. Wang and A. Layton. Numerical simulations of fiber sedimentation in Navier-Stokes flows. *Communications in Computational Physics*, 5(1):61–83, 2009.
[32] L.T. Zhang and M. Gay. Immersed finite element method for fluid-structure interactions. *Journal of Fluids and Structures*, 23:839–857, 2007.
[33] N. Zhang and Z.C. Zheng. An improved direct-forcing immersed-boundary method for finite difference applications. *Journal of Computational Physics*, 221(1):250–268, 2007.
[34] X. Zhang, X. Zhu, and G. He. An improved direct-forcing immersed boundary method for fluid-structure interaction simulations. In *ASME 2013 Fluids Engineering Division Summer Meeting*, V01AT08A005. American Society of Mechanical Engineers, 2013.