A strongly-coupled immersed-boundary formulation for thin deforming surfaces, with application to elastic beams

Andres Goza[a,*], Tim Colonius[a]

[a]*Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA*

Abstract

We present a strongly-coupled immersed-boundary method for flow-structure interaction problems involving thin deforming bodies. The method is stable for arbitrary choices of solid-to-fluid mass ratios and for large body motions. As with many strongly-coupled immersed-boundary methods, our method requires the solution of a nonlinear algebraic system at each time step. The system is solved through iteration, where the iterates are obtained by linearizing the system and performing a block LU factorization. This restricts all iterations to small-dimensional subsystems that scale with the number of discretization points on the immersed surface, rather than on the entire flow domain. Moreover, the iteration procedure we propose does not involve heuristic regularization parameters, and has converged in a small number of iterations for all problems we have considered. We derive our method for general deforming surfaces, and verify the method with two-dimensional test problems of geometrically nonlinear beams undergoing large amplitude flapping behavior.

Keywords: flow structure interaction, projection method, strong coupling, immersed-boundary

1. Introduction

The immersed-boundary (IB) method is commonly used for flow-structure interaction (FSI) problems because of its ability to handle the fluid and immersed body separately. This flexibility comes at the cost of complicating the implementation of the no-slip boundary condition. This interface constraint nonlinearly couples the fluid and solid, and is therefore difficult to impose efficiently and robustly. We restrict our attention here to strongly-coupled IB methods, which strictly enforce the constraint. Weakly coupled IB methods do not impose the constraint, and are unstable for small solid-to-fluid mass ratios and large body motions [1–3].

Due to the nonlinear nature of the constraint, most strongly-coupled methods must solve a large nonlinear system of equations at each time step. The block Gauss-Seidel procedure is one approach to solving this nonlinear system. It is attractive for its ease of implementation, but requires relaxation to converge for a wide range of mass ratios. Employing relaxation requires a heuristically chosen parameter, and can involve dozens of iterations to converge for small mass ratios [4], though Wang and Eldredge [5] improved this convergence behavior using information about the system's added mass. Alternatively, the nonlinear system can be solved with a Newton-Raphson method. This removes the need of free parameters, and typically requires a small number of iterations irrespective of the mass ratio. However, this approach often involves computing several matrix-vector products per time step, each involving large Jacobian matrices [6–8].

[*] Corresponding author
Email address: ajgoza@gmail.com (Andres Goza)

In the context of rigid body FSI problems, some strongly-coupled methods evaluate the constraint equation at the previous time step, which allows for the resulting equations of motion to be linear [9, 10]. Thus, these methods do not require iterations, though the temporal accuracy is reduced to first order due to the time lag introduced in the constraint. It is difficult to avoid iteration in deforming body problems, since the solid equations have a nonlinear stiffness term that is frequently treated implicitly when discretized in time.

We present a strongly-coupled IB method for thin deforming surfaces that iteratively solves the nonlinear algebraic equations using a linearization of the system, as is done in the Newton-Raphson approach. Therefore, our method does not require free relaxation parameters, and has exhibited fast convergence behavior for all problems we have considered so far. A distinction between our method and a standard Newton-Raphson implementation is that we avoid large Jacobian matrices by performing a block LU factorization of the linearized system. This reduces all iterations to subsystems whose dimensions scale with the number of discretization points on the immersed surface, rather than on the entire flow domain.

Our method treats the fluid with the two-dimensional (2D) discrete streamfunction formulation of Colonius and Taira [11], and the solid with a finite element formulation that applies to various solid materials undergoing large deformations and rotations. We verify our method on 2D test problems of flow past deforming beams. The test problems involve large body motions and a wide range of mass ratios, and for all cases our proposed iteration process required a small number of iterations to converge. To supplement the 2D method presented in the main text, we derive in appendix A an analogous method that treats the fluid with primitive variables. This method has a similar iteration procedure to the proposed 2D formulation, and can be applied in both 2D and 3D.

2. Governing equations

We consider a fluid domain Ω and an immersed body Γ. We let $\mathbf{x}$ denote the Eulerian coordinate representing a position in space, and define $\boldsymbol{\chi}(s,t)$ as the Lagrangian coordinate attached to the body Γ (s is a variable that parametrizes the surface). The dimensionless governing equations are written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u} + \int_\Gamma \mathbf{f}(\boldsymbol{\chi}(s,t))\delta(\boldsymbol{\chi}(s,t)-\mathbf{x})ds \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{2}$$

$$\frac{\rho_s}{\rho_f}\frac{\partial^2\boldsymbol{\chi}}{\partial t^2} = \frac{1}{\rho_f U_\infty^2}\nabla\cdot\boldsymbol{\sigma} + \mathbf{g}(\boldsymbol{\chi}) - \mathbf{f}(\boldsymbol{\chi}) \tag{3}$$

$$\int_\Omega \mathbf{u}(\mathbf{x})\delta(\mathbf{x}-\boldsymbol{\chi}(s,t))d\mathbf{x} = \frac{\partial\boldsymbol{\chi}(s,t)}{\partial t} \tag{4}$$

In the above, $\mathbf{x}$, $\boldsymbol{\chi}$, and s were nondimensionalized by a characteristic length scale, L; $\mathbf{u}$ was nondimensionalized by a characteristic velocity scale, U_∞; t was nondimensionalized by L/U_∞; and p and $\mathbf{f}$ were nondimensionalized by $\rho_f U_\infty^2$, where ρ_f is the fluid density (ρ_s is the solid density). The Reynolds number in (1) is defined as $\mathrm{Re} = U_\infty L/\nu$, where ν is the kinematic viscosity of the fluid. The term $\mathbf{g}$ represents a body force per unit volume (e.g. gravity) applied to the immersed body, and was nondimensionalized by $\rho_f U_\infty^2/L$. The momentum equations (3) were divided by $\rho_f U_\infty^2/L$ to arrive at the form seen above.

In (3), the time derivative is understood to be a Lagrangian derivative, and the stress tensor used is the Cauchy stress, which is related to the second Piola-Kirchoff stress in the undeformed configuration, $\boldsymbol{\sigma}^K$, by

$$\sigma_{mn} = \frac{1}{J}\frac{\partial x_m}{\partial \chi_i^0}\frac{\partial x_n}{\partial \chi_j^0}\sigma_{ij}^K \tag{5}$$

where $\boldsymbol{\chi}^0$ is the position of the body in its undeformed configuration, and $J = \det(\partial x_i/\partial\chi_j^0)$. The second Piola-Kirchoff stress is related to the strains within the solid via

$$\sigma_{ij}^K = D_{ijkl}E_{kl} \tag{6}$$

where $\mathbf{D}$ depends on Young's Modulus, E, the bulk shear modulus, G, and Poisson's ratio, ν_s; and $\mathbf{E}$ is the strain tensor given by

$$E_{ij} = \frac{1}{2}\left(\frac{\partial \chi_i}{\partial \chi_j^0} + \frac{\partial \chi_j}{\partial \chi_i^0} + \frac{\partial \chi_m}{\partial \chi_i^0}\frac{\partial \chi_m}{\partial \chi_j^0}\right) \tag{7}$$

(summation implied on repeated indices). Again, all variables that comprise $\mathbf{E}$ were nondimensionalized using the characteristic length L.

The boundary condition on the interface Γ is explicitly written as a constraint in (4). Its role is analogous to that of the continuity equation in computing the pressure: it is used to solve for the singular source term $\mathbf{f}(\boldsymbol{\chi})$ that enforces the boundary condition on Γ. Note that $\mathbf{f}(\boldsymbol{\chi})$ represents the surface stress imposed on the fluid by the immersed body. This can be seen by multiplying the momentum equations (1) by $\delta(\mathbf{x} - \boldsymbol{\chi}(s,t))$ and integrating over the domain Ω to get

$$\mathbf{f}(\boldsymbol{\chi}(s,t)) = \left[\left(\frac{\partial}{\partial t} - \frac{1}{Re}\nabla^2\right)\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p\right]\Bigg|_{\mathbf{x}=\boldsymbol{\chi}(s,t)} \qquad \forall\ \boldsymbol{\chi} \in \partial\Gamma \tag{8}$$

The negative of this term represents the stress imposed on the immersed body by the fluid, including added mass effects. The immersed-boundary method places fictitious fluid inside of bodies that contributes to the surface stress in (8). We restrict our attention to thin bodies, such as shells or membranes, where the fictitious fluid may be neglected in (8). Treating thick bodies would require the ability to remove the stress contribution from the fictitious fluid in (8) to obtain the stress from the physical fluid. This was done for rigid body FSI problems by Lācis *et al.* [10], but is difficult to extend to the deforming body case.

3. Numerical method

In this section, we discretize the equations of motion in space to arrive at the coupled semi-discrete equations of motion: the fluid equations are discretized using the 2D discrete streamfunction formulation of Colonius and Taira [11], and the solid equations are discretized using a finite element formulation. We then discretize in time and introduce an efficient iteration procedure for solving the resulting nonlinear algebraic equations.

3.1. Semi-discrete equations

We begin by spatially discretizing the equations of motion for the fluid on a uniform Cartesian grid as

$$\dot{u} + \mathcal{N}(u) = -Gp + Lu + H(\chi)\tilde{f} \tag{9}$$

$$Du = 0 \tag{10}$$

where the overdot denotes differentiation with respect to time, u, p, χ and $\tilde{f}$ denote the spatially discrete velocity, pressure, body position, and surface stresses; $\mathcal{N}(u)$ is a discretization of the nonlinear term; G, L, and D are discretizations of the gradient, Laplacian, and divergence operators, respectively; and $H(\chi)$ is a discretization of the term involving the delta function in (1). The convective term is discretized in standard convective form with a central difference approximation of the derivatives, L is built using the common 5 point finite difference stencil, and G and D are constructed with finite difference schemes such that $G = -D^T$. For computational efficiency, H is built to satisfy $H\tilde{f} = -E^T f$, where f is equal to $\tilde{f}$ to within a scaling factor and the matrix E is a discretization of the term involving the delta function in (4). See reference [12] for more details.

Following Colonius and Taira [11], we avoid the incompressibility constraint by using a discrete curl operator that lies in the null space of the discrete divergence operator D. That is, $DC \equiv 0$, which implies that $C^T G = -(DC)^T \equiv 0$. The discrete curl operator engenders the use of a discrete streamfunction that is related to the discrete velocity field by $u = Cs$. Using this and premultiplying (9) and (10) by C^T then gives the final form of the semi-discrete fluid equations that we consider:

$$C^T C\dot{s} + \mathcal{N}(Cs) = C^T LCs - C^T E^T(\chi)f \tag{11}$$

The equations for the solid are discretized in space using a finite element procedure: the body is broken up into isoparametric finite elements with an associated set of compatible shape functions [13]. A variable $(\cdot)$ belonging to the surface may be expressed with these basis functions as

$$(\cdot)(\boldsymbol{\chi}) = \sum_{j=1}^{N_{node}} (\cdot)_j b_j(\boldsymbol{\chi}) \tag{12}$$

where $(\cdot)_j$ is the nodal value of $(\cdot)$ and b_j is the shape function corresponding to node j. Using this expansion of each variable in (3), we arrive at a system of ordinary differential equations in time by multiplying (3) by the various shape functions and integrating over the volume of the immersed body. Letting B be a matrix containing the various shape functions b_j, we write the spatially discretized form of (3) as

$$M\ddot{\chi} + R(\chi) = Q(g + W(\chi)f) \tag{13}$$

where

$$M = \frac{\rho_s}{\rho_f}\sum_{j=1}^{N_{el}} \int_{\Gamma_j^0} B^T B d\boldsymbol{\chi}^0 \tag{14}$$

$$R(\chi) = \frac{1}{\rho_f U_\infty^2}\sum_{j=1}^{N_{el}} \int_{\Gamma_j^0} B_E^T \sigma^K d\boldsymbol{\chi}^0 \tag{15}$$

$$Q = \sum_{j=1}^{N_{el}} \int_{\Gamma_j^0} B^T B d\boldsymbol{\chi}^0 = \frac{\rho_f}{\rho_s} M \tag{16}$$

In the above, Γ_j^0 denotes element j of Γ in its undeformed configuration, B_E is a matrix containing the derivatives of the shape functions with respect to the nodal positions, and σ^K contains nodal values of its continuous analog. Note that σ^K is arranged as a vector so that $R(\chi)$ is also a vector. The nonlinearity of $R(\chi)$ is due to the dependence of B_E and σ_K on χ.

It is worth mentioning that the surface stresses f obtained by many immersed-boundary methods contain spurious oscillations [14–16]. In the context of the present immersed-boundary method, these oscillations are due to the fact that the surface stresses are obtained from an ill-posed integral equation. We recently observed that for an appropriate choice of delta function, a weighting matrix, W, may be used to obtain accurate surface stresses [14]. Thus, W is included in (13) to apply the physically correct surface stresses on the immersed surface. The specific form of W is described in reference [14], but we note here that Wf is a discretization of

$$\frac{\int_\Omega \int_\Gamma \mathbf{f}(\boldsymbol{\chi}(s',t))\delta_h(\mathbf{x} - \boldsymbol{\chi}(s',t))\delta_h(\mathbf{x} - \boldsymbol{\chi}(s,t))ds'd\mathbf{x}}{\int_\Gamma \delta_h(\mathbf{x} - \boldsymbol{\chi}(s',t))ds'} \tag{17}$$

where δ_h is a continuous delta function whose support depends on the grid spacing h (see, for example, the review of Peskin [17] for more information on continous delta functions and how to construct them). The expression in (17) is a spatially first order approximation to $\mathbf{f}(\boldsymbol{\chi})$, so the presence of W is a first order modification of (3). In the present work, we use the delta function of Roma [18], which we found to provide a good combination of computational efficiency and accuracy of the surface stress.

Equation (13) and the corresponding definitions of M, R, and Q are valid for a variety of solid materials undergoing large deformations, displacements, and rotations. In this work we restrict our attention to beams, for which we employ a corotational formulation [19]. In this formulation, arbitrarily large displacements and rotations are accommodated by attaching a local coordinate frame to each beam element. The strains are assumed to be small in this frame, and the corresponding internal stresses $R(\chi)$ are well known (see, *e.g.*, reference [19]). Materials other than beams would require changes in the choice of elements, shape functions, and model for σ^K. However, these changes would not affect the structure of (13) or the ensuing time discretization procedure.

Using these discretizations of the fluid and solid equations, the discrete form of the constraint (4) is written in terms of the streamfunction as

$$E(\chi)Cs - \zeta = 0 \tag{18}$$

Defining $\zeta := \dot{\chi}$, the fully coupled FSI equations may be written as a first order system of differential-algebraic equations given by

$$C^T C\dot{s} = -\mathcal{N}(Cs) + C^T LCs - C^T E^T(\chi)f \tag{19}$$

$$M\dot{\zeta} = -R(\chi) + Q(g + W(\chi)f) \tag{20}$$

$$\dot{\chi} = \zeta \tag{21}$$

$$E(\chi)Cs - \zeta = 0 \tag{22}$$

Note that (9)–(10) and (13) are valid in both 2D and 3D; the restriction to 2D is due only to the discrete streamfunction formulation (11). We derive in appendix A an alternative method that treats the fluid using (9)–(10). This method has an analogous iteration procedure to the one proposed in section 3.2 and is readily extendible to 3D.

3.2. Time discretization and efficient factorization procedure

We discretize (19) using an Adams-Bashforth scheme for the nonlinear term and a Crank-Nicholson method for the diffusive term. Equations (20)–(21) are discretized using an implicit Newmark scheme. The constraint equation (22) is evaluated at the current time step. That is, the method is strongly-coupled, which is necessary for the method to be stable for a wide range of mass ratios and in the presence of large body displacements [1–3]. We discretize the equations of motion in the aforementioned way to illustrate the iteration procedure on a commonly used scheme. However, the proposed iteration approach can readily be extended for a variety of time stepping schemes.

Discretizating (19)–(22) as described in the previous paragraph leads to a system of nonlinear algebraic equations given by

$$C^T ACs_{n+1} + C^T E^T_{n+1} f_{n+1} = r^f_n \tag{23}$$

$$\frac{4}{\Delta t^2} M\zeta_{n+1} + (R(\chi_{n+1}) - QW_{n+1}f_{n+1}) = r^\zeta_n \tag{24}$$

$$\frac{2}{\Delta t}\chi_{n+1} - \zeta_{n+1} = r^\chi_n \tag{25}$$

$$E_{n+1}Cs_{n+1} - \zeta_{n+1} = 0 \tag{26}$$

where $A = \frac{1}{\Delta t}I - \frac{1}{2}L$, $r^f_n = (\frac{1}{\Delta t}C^T C + \frac{1}{2}C^T LC)s_n + \frac{3}{2}C^T\mathcal{N}(Cs_n) - \frac{1}{2}C^T\mathcal{N}(Cs_{n-1})$, $r^\zeta_n = M(\frac{4}{\Delta t^2}\chi_n + \frac{4}{\Delta t}\zeta_n + \dot{\zeta}_n) + Qg$, and $r^\chi_n = \zeta_n + \frac{2}{\Delta t}\chi_n$. Note that the operators E, E^T, and W are given subscripts to indicate their dependence on χ.

We now describe an iteration procedure for using a guess for the solution at iteration (k) to compute a new guess at iteration $(k+1)$ (We use the solution at time step n as the guess for $(k) = 0$). To do this, we write $\chi^{(k+1)}_{n+1} = \chi^{(k)}_{n+1} + \Delta\chi$, $\zeta^{(k+1)}_{n+1} = \zeta^{(k)}_{n+1} + \Delta\zeta$, where the increments $\Delta\chi$, $\Delta\zeta$ are assumed to be small. Substituting this decomposition into (23)–(26) and retaining first order terms in the increments and Δt gives the linear system[1]

$$\begin{bmatrix} C^TAC & 0 & 0 & C^TE_{n+1}^{(k)T} \\ 0 & 0 & \frac{4}{\Delta t^2}M + K^{(k)} & -QW_{n+1}^{(k)} \\ 0 & -I & \frac{2}{\Delta t}I & 0 \\ E_{n+1}^{(k)}C & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} s_{n+1} \\ \Delta\zeta \\ \Delta\chi \\ f_{n+1}^{(k+1)} \end{bmatrix} = \begin{bmatrix} r_n^f + O(\Delta t) \\ r_n^\zeta - \frac{4}{\Delta t^2}M\chi_{n+1}^{(k)} - R(\chi_{n+1}^{(k)}) + O(\Delta t) \\ r_n^\chi - \frac{2}{\Delta t}\chi_{n+1}^{(k)} + \zeta_{n+1}^{(k)} \\ \zeta_{n+1}^{(k)} + O(\Delta t) \end{bmatrix} := \begin{bmatrix} r_n^f \\ r^{\zeta\,(k)} \\ r^{\chi\,(k)} \\ r^{c\,(k)} \end{bmatrix} \tag{27}$$

where $K^{(k)} = dR/d\chi|_{\chi=\chi_{n+1}^{(k)}}$. For beams this stiffness matrix has well known analytical expressions [13, 19].

The linear system (27) may be factored using a block LU decomposition. Defining $\hat{K}^{(k)} := \frac{4}{\Delta t^2}M + K^{(k)}$ and $B_{n+1}^{(k)} := E_{n+1}^{(k)}C(C^TAC)^{-1}C^TE_{n+1}^{(k)T}$, the factored equations are

$$s^* = (C^TAC)^{-1}r_n^f \tag{28}$$

$$\begin{bmatrix} B_{n+1}^{(k)} & I \\ -\frac{2}{\Delta t}QW_{n+1}^{(k)} & \hat{K}^{(k)} \end{bmatrix} \begin{bmatrix} f_{n+1}^{(k+1)} \\ \Delta\zeta \end{bmatrix} = \begin{bmatrix} E_{n+1}^{(k)}Cs^* - r^{c\,(k)} \\ \frac{2}{\Delta t}r^{\zeta\,(k)} - r^{\chi\,(k)} \end{bmatrix} \tag{29}$$

$$\Delta\chi = \frac{\Delta t}{2}(\Delta\zeta + r^{\chi\,(k)}) \tag{30}$$

$$s_{n+1} = s^* - (C^TAC)^{-1}C^TE_{n+1}^Tf_{n+1} \tag{31}$$

Note that (28) does not depend on information at time step $n+1$, and (29)–(30) do not require knowledge of s_{n+1}. Thus, (28) can be solved once and for all at the beginning of each time step, and s_{n+1} only needs to be computed once, after (29)–(30) have been iterated to convergence. This has the benefit that all iterations are restricted to (29)–(30), which have dimensions on the order of the number of body points, rather than the total number of points in the flow domain. We do not write a superscript on s_{n+1} since it does not need to be iterated on. Wang *et al.* [5] also restricted iterations to small dimensional subsystems like (29)–(30). In their case, the solid equations were replaced by the rigid body equations of motion, and a block Gauss-Seidel procedure with added mass relaxation was used to solve their analogous nonlinear system.

A Poisson-like problem $(C^TAC)^{-1}$ must be solved in (28), (31), and in each matrix-vector multiply with $B_{n+1}^{(k)}$. Solving the Poisson-like problem may be done efficiently using fast Fourier transforms, but requires operations of the order of the number of points on the flow domain. Thus, depending on the dimensions of the system, the computation and storage of $(\hat{K}^{(k)})^{-1}$ may be small compared with solving the Poisson-like problem. In this case one may perform block Gauss-elimination to reformulate (29) as

$$\left(B_{n+1}^{(k)} + \frac{2}{\Delta t}(\hat{K}^{(k)})^{-1}QW_{n+1}^{(k)}\right)f_{n+1}^{(k+1)} = E_{n+1}^{(k)}Cs^* - r^{c\,(k)} - \frac{2}{\Delta t}(\hat{K}^{(k)})^{-1}r^{\zeta\,(k)} + r^{\chi\,(k)} \tag{32}$$

$$\Delta\zeta = \frac{2}{\Delta t}(\hat{K}^{(k)})^{-1}(r^{\zeta\,(k)} + QW_{n+1}^{(k)}f_{n+1}^{(k)}) - r^{\chi\,(k)} \tag{33}$$

This allows for (32) and (33) to be solved sequentially. Moreover, while (32) must still be solve iteratively because of the embedded Poisson-like problem in $B_{n+1}^{(k)}$, the conditioning of the system is improved because the heterogeneous blocks of (29) are no longer present.

[1] Derivative terms that arise in the expansion of $E_{n+1}^{(k+1)}$, $E_{n+1}^{(k+1)T}$, and $W_{n+1}^{(k+1)}$ may be neglected to within $O(\Delta t)$. For example, the fourth block equation of (27) including this extra term is $E_{n+1}^{(k)}Cs_{n+1} + (DE_{n+1}^{(k)}\Delta\chi)Cs_{n+1} - \Delta\zeta = \zeta_{n+1}^{(k)}$. The DE term is of order $O(\Delta t)$ compared with the $\Delta\zeta$ term, since by the third block equation $\Delta\chi = O(\Delta\zeta\Delta t)$.

We have tested both (29) and (32)–(33), and found the formulation (32)–(33) to be more efficient for the test problems considered. We solve (32) using the BiCGSTAB method, which typically requires 2–8 iterations to converge using $f_{n+1}^{(k)}$ as the initial guess. Moreover, the number of iterations required by BiCGSTAB often decreases as k increases, since $f_{n+1}^{(k)}$ becomes an increasingly good guess for $f_{n+1}^{(k+1)}$.

It is also worth noting that the updates for $\chi_{n+1}^{(k)}$, $\zeta_{n+1}^{(k)}$, and $f_{n+1}^{(k)}$ do not depend on a heuristic relaxation parameter, as is often required when applying the block Gauss-Seidel iterative procedure to FSI solvers. Moreover, the proposed iteration procedure has required a small number of iterations to converge for all problems we have considered so far.

4. Verification on flapping beam problems

We verify our method for several test problems of 2D flow past deforming beams. The dynamics of this system are governed by the Reynolds number, dimensionless mass ratio, dimensionless bending stiffness, and Froude number. These are given, respectively, as

$$\mathrm{Re} = \frac{U_\infty L}{\nu},\ \mathrm{M}_\rho = \frac{\rho_s h}{\rho_f L},\ \mathrm{K_B} = \frac{EI}{\rho_f U_\infty^2 L^3},\ \mathrm{Fr} = \frac{U_\infty}{\sqrt{gL}} \tag{34}$$

where h is the thickness of the beam, EI is the bending stiffness of the beam, ν is the kinematic viscosity of the fluid, and g is the gravitational constant.

We consider problems with the beam pinned at the leading edge (standard configuration) and with the beam clamped at the trailing edge (inverted configuration). Schematics of the different problem setups are shown in figure 1.

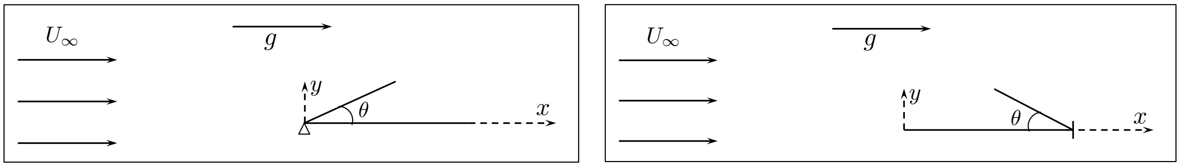

Figure 1: Flow moves from left to right past a beam in the standard configuration (left) or the inverted configuration (right).

Each configuration presents different challenges: beams in the standard configuration are typically associated with smaller mass ratios, whereas those in the inverted configuration often undergo larger motions and have greater dimensionless bending stiffnesses.

The flow equations are solved using a multidomain approach: the finest grid surrounds the body and grids of increasing coarseness are used as distance from the body increases (see reference [11] for details). For all cases, the immersed-boundary spacing is set to be twice that of the flow grid spacing on the finest sub-domain. We found that to give well-conditioned matrices $B_{n+1}^{(k)}$ without making the body porous (Kallemov *et al.* [20] also found this ratio to be preferable). We used a convergence criteria of $||\Delta\chi||_\infty / ||\chi_{n+1}^{(k+1)}||_\infty \leq 1 \times 10^{-7}$ when iterating between (30), (32), and (33).

4.1. Flow past a beam in the standard configuration

We first consider flow past a deforming beam in the standard configuration with $\mathrm{Re} = 1000$, $\mathrm{M}_\rho = 0.075$, $\mathrm{K_B} = 0.0001$, and $\mathrm{Fr} = 0$ (*i.e.*, no gravitational body force). This choice of parameters leads to limit cycle flapping, which we triggered in our simulations by initializing the beam at $\theta = 1°$ (see figure 1 for the definition of θ). After an initial transient, the trailing edge of the beam has oscillatory transverse displacement of fixed amplitude and frequency. Table 1 shows this amplitude and frequency as computed by various authors. Our finest domain was of size $[-0.2, 1.8] \times [-0.3, 0.3]$, and the entire flow domain size was $[-15.20, 16.80] \times [-4.78, 4.78]$. The grid spacing on the finest subdomain was $h = 0.0025$, and the time step was $\Delta t = 0.0006$. Using a grid spacing of $h = 0.003$ changed our results in table 1 by less than one percent. Five iterations of (30), (32), and (33) were required for the first time step, and a maximum of three iterations were required for all remaining time steps.

	Amplitude	Frequency (St)
Connell *et al.* [21]	±0.096	0.93
Gurugubelli *et al.* [22]	±0.098	0.95
Present	±0.097	0.94

Table 1: Amplitude and frequency associated with the transverse displacement of the beam's trailing edge; obtained for Re = 1000, $\mathrm{M}_\rho = 0.075$, $\mathrm{K_B} = 0.0001$, and Fr = 0.

Limit cycle flapping also occurs for Re = 200, $\mathrm{M}_\rho = 1.5$, $\mathrm{K_B} = 0.0015$, and Fr = 1.4. We show in table 2 the associated amplitude and frequency of the transverse trailing edge displacement. For comparison with the literature, we initialized the beam at $\theta = 18°$. Our finest sub-domain for this problem was of size $[-0.2, 1.8] \times [-0.9, 0.9]$, and the total domain size was $[-15.20, 16.80] \times [-7.95, 7.95]$. The grid spacing on the finest domain was $h = 0.00625$, and the time step was $\Delta t = 0.0001$. Using $h = 0.005$ changed our reported results from table 2 by less than one percent. Three iterations of (30), (32), and (33) were required for the first time step, and a maximum of two iterations were required for all remaining time steps.

	Amplitude	Frequency (St)
Huang *et al.* [23]	±0.35	0.30
Wang *et al.* [5]	±0.35	0.31
Lee *et al.* [24]	±0.38	0.31
Present	±0.38	0.32

Table 2: Amplitude and frequency associated with the transverse displacement of the beam's trailing edge; obtained for Re = 200, $\mathrm{M}_\rho = 1.5$, $\mathrm{K_B} = 0.0015$, and Fr = 1.4.

Figure 2 gives a time history of the trailing edge transverse displacement for the parameters corresponding to table 2 (results from references [23, 24] are included for comparison). To emphasize the robustness of the method for a range of mass ratios, figure 2 also shows the trailing edge displacement for $\mathrm{M}_\rho = 0.0001$ and $\mathrm{M}_\rho = 100$. To our knowledge, the $\mathrm{M}_\rho = 0.0001$ and $\mathrm{M}_\rho = 100$ cases have not been simulated before. Moreover, we were not able to simulate the $\mathrm{M}_\rho = 0.0001$ case when solving (23)–(26) with the Gauss-Seidel method, even when using extensive relaxation. Using (30), (32), and (33), a maximum of three iterations per time step were required after the first twenty time steps. During the first twenty time steps, up to fifteen iterations were required due to the impact of the impulsive start on a beam with such small inertia.

In figure 3, we show vorticity contours at different time instances for $\mathrm{M}_\rho = 0.0001$, $\mathrm{M}_\rho = 1.5$, and $\mathrm{M}_\rho = 100$. When $\mathrm{M}_\rho = 0.0001$, the impulsive start pushes the beam down quickly towards the $\theta = 0°$ position. At $t \approx 0.5$, the vortical structure created during the impulsive start reaches the trailing edge of the beam. Due to the beam's small inertia, this substantially affects the trailing edge displacement (see insert in figure 2). The vortical structure then advects away from the body, and the fluid wake becomes symmetric as the beam fully reaches its $\theta = 0°$ position. This symmetric wake is a well known feature of flow past thin rigid bodies at low Reynolds numbers, and the small inertia of the beam does not allow for any minor beam deformations to break the flow symmetry. Thus, the beam stays in this neutral position for all remaining time. When $\mathrm{M}_\rho = 1.5$, the limit cycle flapping of the beam is associated with a periodic vortex street [5, 23, 24]. As the beam becomes increasingly massive, the flapping amplitude increases. This is associated with a thicker, more irregular wake profile and more chaotic flapping behavior. Figure 2 shows that the $\mathrm{M}_\rho = 100$ case

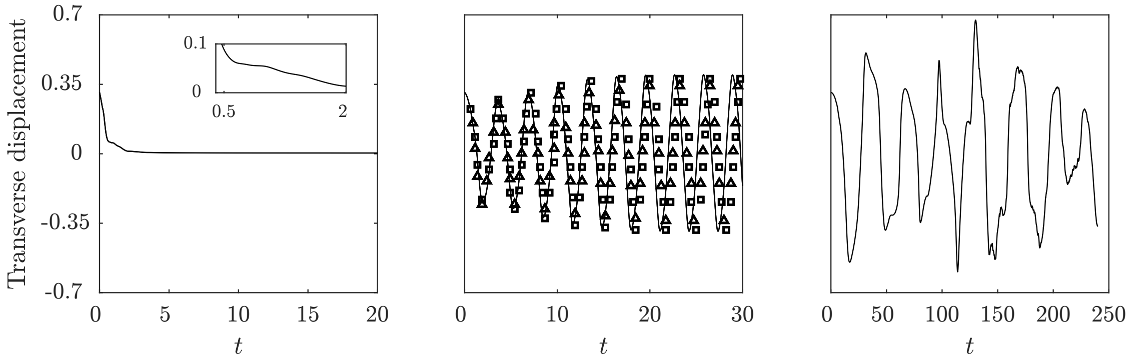

Figure 2: Transverse displacement of the beam's trailing edge for Re= 200, $K_B = 0.0015$, and Fr = 1.4. Left panel: $M_\rho = 0.0001$, middle panel: $M_\rho = 1.5$, right panel: $M_\rho = 100$. Present: —, Lee *et al.* [24]: □, Huang *et al.* [23]: △. The insert on the left panel is a zoom-in of the $M_\rho = 0.0001$ case for $t \in [0.5, 2]$. Note the different horizontal axis values on each panels.

does not enter into limit cycle flapping after 250 convective time units and several periods of flapping. The destabilizing nature of increasing the mass ratio was also noted by Connell *et al.* [21].

4.2. Flow past a beam in the inverted configuration

For a beam initially placed at $\theta = 0°$, there are three possible regimes: fixed-point stable, static divergence, or unstable flapping [22] (see figure 4 for an illustration).

In table 3, we summarize the regimes we obtained by varying K_B for Re = 200, $M_\rho = 0.1$, and Fr = 0. Our finest sub-domain for this problem was of size $[-0.2, 1.8] \times [-0.5, 0.5]$, and the total domain size was $[-15.20, 16.80] \times [-16, 16]$. The grid spacing on the finest domain was $h = 0.002$, and the time step was $\Delta t = 0.00018$. Using $h = 0.0018$ change our reported results by less than one percent. Three iterations of (30), (32), and (33) were required for the first time step, and a maximum of two iterations were required for all remaining time steps. The table also shows the regimes computed by Gurugubelli *et al.* [22] for the same parameter ranges.

	K_B										
	0.41	0.42	0.43	0.44	0.45	0.46	0.47	0.48	0.49	0.50	0.51
Gurugubelli *et al.* [22]	UF	UF	UF	UF	UF	SD	SD	SD	SD	SD	FPS
Present	UF	UF	UF	SD	SD	SD	SD	SD	SD	SD	FPS

Table 3: Flapping regimes obtained for different K_B values. UF = unstable flapping, SD = static divergence, FPS = fixed point stable. The other parameters were Re = 200, $M_\rho = 0.1$, and Fr = 0.

We next consider the case when Re = 1000, $M_\rho = 0.1$, $K_B = 0.4$, and Fr = 0. For this set of parameters, the beam enters large amplitude limit cycle flapping. Table 4 shows the amplitude and frequency of the leading edge transverse displacement computed in our work and in reference [22]. Our finest sub-domain for this problem was of size $[-0.2, 1.8] \times [-1, 1]$, and the total domain size was $[-15.20, 16.80] \times [-14.35, 14.35]$. The grid spacing on the finest domain was $h = 0.0033$, and the time step was $\Delta t = 0.0008$. Using $h = 0.0025$ did not change our reported results.

Figure 5 shows vorticity contours at four different times during a flapping cycle. The figure shows features consistent with what was observed by Gurugubelli *et al* [22]. The fluid stresses associated with the leading edge vortex deform the beam (leftmost figure). The beam begins to flap back towards the centerline once

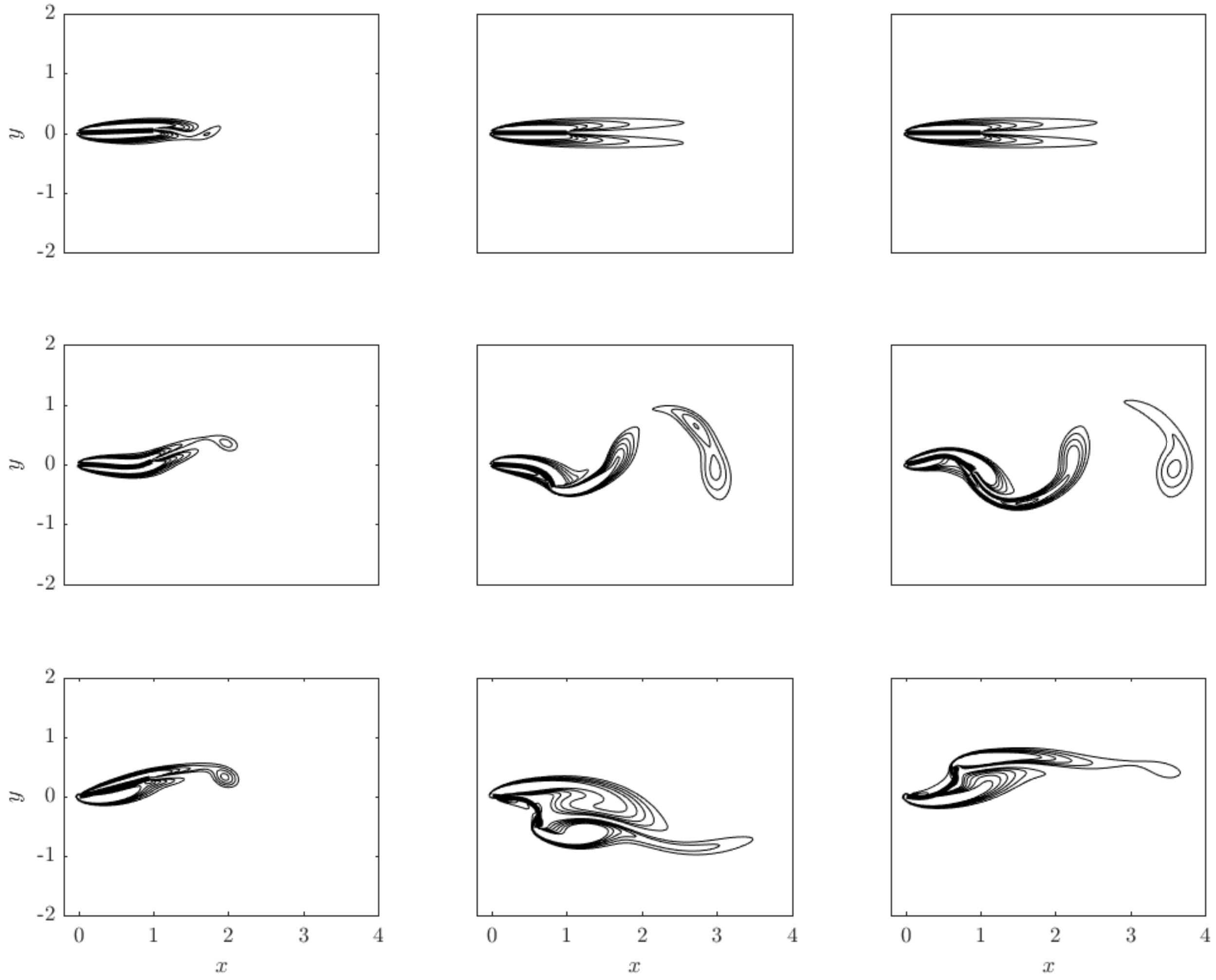

Figure 3: Vorticity contours of flow past a deforming beam for $\mathrm{M}_\rho = 0.0001$ (top row), $\mathrm{M}_\rho = 1.5$ (middle row), and $\mathrm{M}_\rho = 100$ (bottom row). Left column: $t = 1.2$; middle column: $t = 18$; right column: $t = 31.2$. Contours are in 15 evenly spaced increments from -5 to 5. The other parameters were Re = 200, $\mathrm{M}_\rho = 1.5$, and Fr = 1.4.

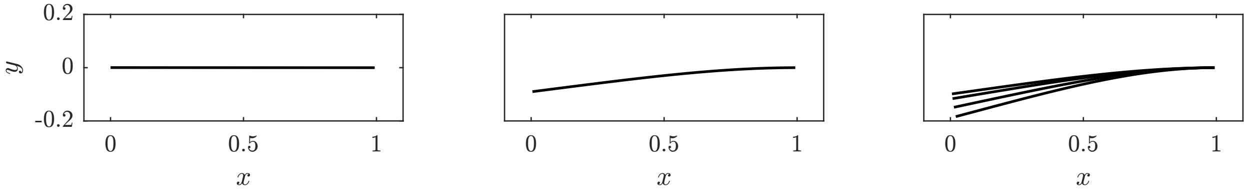

Figure 4: Snapshots of the beam at various times for the fixed-point stable (left), static divergence (middle), and unstable flapping (right) regimes. Only one curve is visible for the fixed-point stable and static divergence cases, since the beam does not deflect about these equilibrium positions.

the internal beam stresses counteract the imposed fluid stresses and beam inertia, and the leading edge vortex detaches (second leftmost figure). The inertia of the beam causes it to continue to move towards the centerline while the detached vortex grows in size (second rightmost figure). The beam then flaps upward,

	Amplitude	Frequency (St)
Gurugubelli *et al.* [22]	±0.83	0.204
Present	±0.82	0.198

Table 4: Amplitude and frequency associated with the transverse displacement of the beam's leading edge; obtained for Re = 1000, M_ρ = 0.5, K_B = 0.4, and Fr = 0.

and a vortex grows at the trailing edge (rightmost figure). After the snapshot in the rightmost figure, the detached leading and trailing edge vortices advect downstream and form a vortex pair of opposite sign to the one seen in the beam's near-wake in figure 5. The upward motion of the beam creates an attached leading edge vortex, which starts an analogous process to the one just described. The result of this upward-downward beam motion is limit cycle flapping that repeats indefinitely.

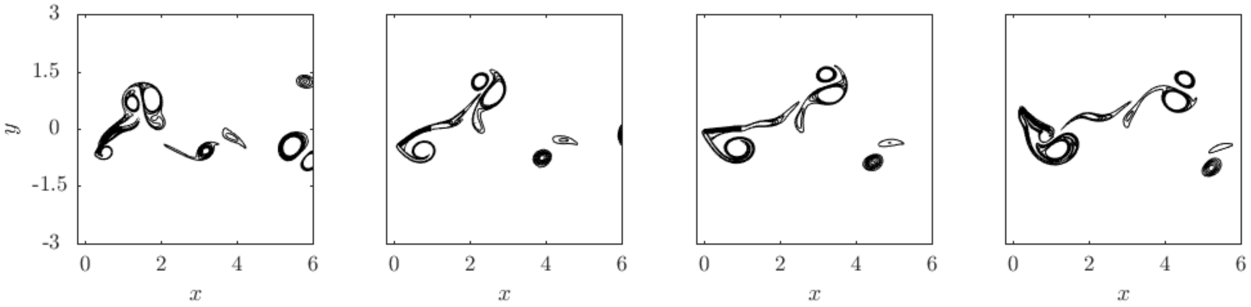

Figure 5: Vorticity contours during a flapping cycle for Re = 1000, M_ρ = 0.5, K_B = 0.4, Fr = 0. Contours are in 15 evenly spaced increments from −5 to 5.

5. Conclusions

We presented an immersed-boundary method for fully coupled flow-structure interaction problems involving thin deforming surfaces. The method is strongly-coupled, and is therefore stable for wide ranges of solid-to-fluid mass ratios and large body motions. As with many strongly-coupled methods, our method requires the solution of a nonlinear system of equations at each time step. This system is solved by iteration, using a linearization of the nonlinearly coupled equations and a block LU factorization of the linearized system to reduce iterations to small-dimensional subsystems of equations. The iteration process does not involve heuristic relaxation parameters. We derived the method for general deforming surfaces, and verified it for 2D flow past deforming beams. The test problems involved a wide range of mass ratios and large body motions, and the method required a small number of iterations to converge for all cases considered. It is straightforward to extend the method to fully coupled problems involving rigid bodies.

6. Acknowledgments

This research was partially supported by a grant from the Jet Propulsion Laboratory (Grant No. 1492185). Many of the simulations were performed using the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. The first author gratefully acknowledges funding from the National Science Foundation Graduate Research Fellowship Program (Grant No. DGE-1144469). We thank Professor Jeff Eldredge for his helpful comments on the manuscript.

Appendix A An efficient iteration procedure in primitive variables

Using primitive variables for the fluid, the differential-algebraic semidiscrete FSI system is

$$\dot{u} + \mathcal{N}(u) = -Gp + Lu - E^T(\chi)f \tag{A.1}$$

$$M\dot{\zeta} = -R(\chi) + Q(g + W(\chi)f) \tag{A.2}$$

$$\dot{\chi} = \zeta \tag{A.3}$$

$$Du = 0 \tag{A.4}$$

$$E(\chi)u - \zeta = 0 \tag{A.5}$$

Using the same time discretization schemes as in section 3 and introducing the decomposition $\chi_{n+1}^{(k+1)} = \chi_{n+1}^{(k)} + \Delta\chi$, $\zeta_{n+1}^{(k+1)} = \zeta_{n+1}^{(k)} + \Delta\zeta$, we have the following system to within first order in the increments and Δt:

$$\begin{bmatrix} A & G & 0 & 0 & E_{n+1}^{(k)\,T} \\ G^T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{K}^{(k)} & -QW_{n+1}^{(k)} \\ 0 & 0 & -I & \frac{2}{\Delta t}I & 0 \\ E_{n+1}^{(k)} & 0 & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ p_{n+1} \\ \Delta\zeta \\ \Delta\chi \\ f_{n+1}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \tilde{r}_n^f \\ \tilde{r}_n^p \\ r^{\zeta\,(k)} \\ r^{\chi\,(k)} \\ r^{c\,(k)} \end{bmatrix} \tag{A.6}$$

where the right hand side terms are known and analogous to those in section 3. Performing a block-LU decomposition of (A.6) gives the sequence of equations

$$u^* = A^{-1}\tilde{r}_n^f - A^{-1}G(G^TA^{-1}G)^{-1}(G^TA^{-1}\tilde{r}_n^f - \tilde{r}^p) \tag{A.7}$$

$$p^* = (G^TA^{-1}G)^{-1}(G^TA^{-1}\tilde{r}_n^f - \tilde{r}_n^p) \tag{A.8}$$

$$\begin{bmatrix} \tilde{B}_{n+1}^{(k)} & I \\ -\frac{2}{\Delta t}QW_{n+1}^{(k)} & \hat{K}^{(k)} \end{bmatrix} \begin{bmatrix} f_{n+1}^{(k+1)} \\ \Delta\zeta \end{bmatrix} = \begin{bmatrix} E_{n+1}^{(k)}u^* - r^{c\,(k)} \\ \frac{2}{\Delta t}r^{\zeta\,(k)} + r^{\chi\,(k)} \end{bmatrix} \tag{A.9}$$

$$\Delta\chi = \frac{2}{\Delta t}(\Delta\zeta + r^{\chi\,(k)}) \tag{A.10}$$

$$p_{n+1} = p^* - (G^TAG)^{-1}G^TA^{-1}E_{n+1}^{(k)\,T}f_{n+1} \tag{A.11}$$

$$u_{n+1} = u^* - A^{-1}(I - G(G^TA^{-1}G)^{-1}G^TA^{-1})E_{n+1}^{(k)\,T}f_{n+1} \tag{A.12}$$

where $\tilde{B}_{n+1}^{(k)} = E_{n+1}^{(k)}A^{-1}(I - G(G^TA^{-1}G)^{-1}G^TA^{-1})E_{n+1}^{(k)\,T}$.

As in section 3, all iterations are restricted to (A.9)–(A.10), which have dimensions on the order of the number of body points; (A.7)–(A.8) may be computed once and for all at the start of a time step, and (A.11)–(A.12) need only be solved after all iterations are completed to convergence.

Again, when $(\hat{K}^{(k)})^{-1}$ can be computed and stored, the system (A.9) may be reformulated as the sequence of equations given by

$$\left(\tilde{B}_{n+1}^{(k)} + \frac{2}{\Delta t}(\hat{K}^{(k)})^{-1}QW_{n+1}^{(k)}\right)f_{n+1}^{(k+1)} = E_{n+1}^{(k)}u^* - r^{c\,(k)} - \frac{2}{\Delta t}(\hat{K}^{(k)})^{-1}r^{\zeta\,(k)} + r^{\chi\,(k)} \tag{A.13}$$

$$\Delta U = \frac{2}{\Delta t}(\hat{K}^{(k)})^{-1}(r^{\zeta\,(k)} + QW_{n+1}^{(k)}f_{n+1}^{(k)}) - r^{\chi\,(k)} \tag{A.14}$$

References

References

[1] P. Causin, J.-F. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid–structure problems, Computer Methods in Applied Mechanics and Engineering 194 (42) (2005) 4506–4527.

[2] C. Förster, W. A. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, Computer Methods in Applied Mechanics and Engineering 196 (7) (2007) 1278–1293.

[3] I. Borazjani, L. Ge, F. Sotiropoulos, Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3d rigid bodies, Journal of Computational Physics 227 (16) (2008) 7587–7620.

[4] F.-B. Tian, H. Dai, H. Luo, J. F. Doyle, B. Rousseau, Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems, Journal of Computational Physics 258 (2014) 451–469.

[5] C. Wang, J. D. Eldredge, Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method, Journal of Computational Physics 295 (2015) 87–113.

[6] J. Degroote, K.-J. Bathe, J. Vierendeels, Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction, Computers & Structures 87 (11) (2009) 793–801.

[7] Y. Mori, C. S. Peskin, Implicit second-order immersed boundary methods with boundary mass, Computer Methods in Applied Mechanics and Engineering 197 (25) (2008) 2049–2067.

[8] G. Hou, J. Wang, A. Layton, Numerical methods for fluid-structure interaction—a review, Communications in Computational Physics 12 (02) (2012) 337–377.

[9] J. Yang, F. Stern, Sharp interface immersed-boundary/level-set method for wave–body interactions, Journal of Computational Physics 228 (17) (2009) 6590–6616.

[10] U. Lācis, K. Taira, S. Bagheri, A stable fluid–structure-interaction solver for low-density rigid bodies using the immersed boundary projection method, Journal of Computational Physics 305 (2016) 300–318.

[11] T. Colonius, K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, Computer Methods in Applied Mechanics and Engineering 197 (25) (2008) 2131–2146.

[12] K. Taira, T. Colonius, The immersed boundary method: a projection approach, Journal of Computational Physics 225 (2) (2007) 2118–2137.

[13] K. J. Bathe, Finite element procedures, Prentice-Hall: Englewood Cliffs, NJ, 1996.

[14] A. Goza, S. Liska, B. Morley, T. Colonius, Accurate computation of surface stresses and forces with immersed boundary methods, Journal of Computational Physics 321 (2016) 860–873.

[15] J. H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, Journal of Computational Physics 230 (19) (2011) 7347–7363.

[16] X. Yang, X. Zhang, Z. Li, G.-W. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, Journal of Computational Physics 228 (20) (2009) 7821–7836.

[17] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[18] A. M. Roma, C. S. Peskin, M. J. Berger, An adaptive version of the immersed boundary method, Journal of Computational Physics 153 (2) (1999) 509–534.

[19] M. Criesfield, Non-linear finite element analysis of solids and structures, vol. 1, Wiley, New York, 1991.

[20] B. Kallemov, A. Bhalla, B. E. Griffith, A. Donev, An immersed boundary method for rigid bodies, Communications in Applied Mathematics and Computational Science 11 (1) (2016) 79 – 141.

[21] B. S. Connell, D. K. Yue, Flapping dynamics of a flag in a uniform stream, Journal of Fluid Mechanics 581 (2007) 33–67.

[22] P. Gurugubelli, R. Jaiman, Self-induced flapping dynamics of a flexible inverted foil in a uniform flow, Journal of Fluid Mechanics 781 (2015) 657–694.

[23] W.-X. Huang, S. J. Shin, H. J. Sung, Simulation of flexible filaments in a uniform flow by the immersed boundary method, Journal of Computational Physics 226 (2) (2007) 2206–2228.

[24] I. Lee, H. Choi, A discrete-forcing immersed boundary method for the fluid–structure interaction of an elastic slender body, Journal of Computational Physics 280 (2015) 529–546.