



ELSEVIER

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp


A strongly-coupled immersed-boundary formulation for thin elastic structures

 Andres Goza^{*}, Tim Colonius


Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA

ARTICLE INFO

Article history:

Received 12 September 2016

Received in revised form 7 February 2017

Accepted 8 February 2017

Available online 14 February 2017

Keywords:

Flow–structure interaction

Projection method

Strong coupling

Immersed-boundary

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 flags undergoing large amplitude flapping behavior.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The immersed-boundary (IB) method treats the fluid and immersed body with separate grids, which obviates the need for computationally expensive tasks like re-meshing. For this reason, IB methods are often used in flow–structure interaction (FSI) problems with bodies undergoing large deformations and rotations. Yet, efficiently handling the interface constraint, which nonlinearly couples the fluid and structure, remains a challenge. 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.

E-mail address: ajgoza@gmail.com (A. 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 structures undergoing large deformations and rotations. We verify our method on 2D test problems of flow past deforming flags. 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

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

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

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

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

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 $\text{Re} = U_{\infty} L / \nu$, 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–Kirchhoff stress in the undeformed configuration, $\boldsymbol{\sigma}^K$, by

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

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–Kirchhoff stress is related to the strains within the solid via

$$\sigma_{ij}^K = D_{ijkl} E_{kl} \quad (6)$$

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

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

(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 $\mathbf{f}(\chi)$ that enforces the boundary condition on Γ . Note that $\mathbf{f}(\chi)$ represents the surface stress imposed on the fluid by the immersed body. This can be seen by multiplying the momentum equations (1) by $\delta(\mathbf{x} - \chi(s, t))$ and integrating over the domain Ω to get

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

The negative of this term represents the stress imposed on the immersed body by the fluid (including added mass effects), and is therefore the source term in (3). The immersed-boundary method places fictitious fluid inside of bodies that produces unphysical contributions to the surface stress in (8). For rigid body FSI problems (e.g., freely falling cylinders), the spurious stress contribution can be explicitly removed to obtain the correct stress [10], but this procedure is difficult to extend to deforming bodies. We therefore restrict our attention to thin bodies, such as shells or membranes, where these unphysical contributions are negligible.

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 staggered 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). Spatial derivatives are approximated with the standard second-order finite difference operators. For computational efficiency, the spatial operators are formulated such that $D = -G^T$ and $H = -E^T$ (to within a scaling factor), where E is a discretization of the operator 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, s , 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:

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

where f is used instead of \tilde{f} in (11) to account for the scaling factor used in writing H as E^T (see reference [12] for more details).

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

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

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

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

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

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

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

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

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 continuous delta functions and how to construct them). The expression in (17) is a spatially first order approximation to $\mathbf{f}(\chi)$, 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 structures undergoing large deformations, displacements, and rotations. In this work we restrict our attention to flags, which we model as thin elastic beams using a corotational formulation [19]. In this formulation, arbitrarily large displacements and rotations are accommodated by attaching a local coordinate frame to each flag 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]). Structures other than flags would require changes in the choice of elements, shape functions, and model for σ^K . However, these changes would not affect the nature 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

$$E(\chi)Cs - \zeta = 0 \quad (18)$$

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

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

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

$$\dot{\chi} = \zeta \quad (21)$$

$$E(\chi)Cs - \zeta = 0 \quad (22)$$

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–Nicolson 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.

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

$$C^T ACs_{n+1} + C^T E_{n+1}^T f_{n+1} = r_n^f \tag{23}$$

$$\frac{4}{\Delta t^2} M \zeta_{n+1} + (R(\chi_{n+1}) - Q W_{n+1} f_{n+1}) = r_n^\zeta \tag{24}$$

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

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

where $A = \frac{1}{\Delta t} I - \frac{1}{2} L$, $r_n^f = (\frac{1}{\Delta t} C^T C + \frac{1}{2} C^T L C) s_n + \frac{3}{2} C^T \mathcal{N}(C s_n) - \frac{1}{2} C^T \mathcal{N}(C s_{n-1})$, $r_n^\zeta = M(\frac{4}{\Delta t^2} \chi_n + \frac{4}{\Delta t} \zeta_n + \dot{\zeta}_n) + Q g$, and $r_n^\chi = \zeta_n + \frac{2}{\Delta t} \chi_n$. 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 $\chi_{n+1}^{(k+1)} = \chi_{n+1}^{(k)} + \Delta\chi$, $\zeta_{n+1}^{(k+1)} = \zeta_{n+1}^{(k)} + \Delta\zeta$, 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 system¹

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

where $K^{(k)} = dR/d\chi|_{\chi=\chi_{n+1}^{(k)}}$. For flags 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 $B_{n+1}^{(k)} := E_{n+1}^{(k)} C (C^T AC)^{-1} C^T E_{n+1}^{(k)T}$, the factored equations are

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

$$\begin{bmatrix} B_{n+1}^{(k)} & I \\ -\frac{2}{\Delta t} Q W_{n+1}^{(k)} & \hat{K}^{(k)} \end{bmatrix} \begin{bmatrix} f_{n+1}^{(k+1)} \\ \Delta\zeta \end{bmatrix} = \begin{bmatrix} E_{n+1}^{(k)} C s^* - r_n^c \\ \frac{2}{\Delta t} r_n^\zeta - r_n^\chi \end{bmatrix} \tag{29}$$

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

$$s_{n+1} = s^* - (C^T AC)^{-1} C^T E_{n+1}^T f_{n+1} \tag{31}$$

The LU-factorized equations (28)–(31) are analogous to the classical fractional-step procedure used for the Navier–Stokes equations. In fact, Perot [20] showed that the fractional-step procedure for the Navier–Stokes equations can be derived by a block-LU factorization of the associated semi-discrete system. Taira and Colonius [12] used a block-LU factorization to derive an analogous fractional step procedure for an immersed boundary method applied to rigid bodies undergoing prescribed kinematics, and Lācis et al. [10] extended this analysis to allow for rigid-body flow–structure interaction. Our method applies this block-LU fractional-step philosophy to FSI problems involving deforming bodies.

The fractional-step procedure (28)–(31) has a physical interpretation: first, a trial streamfunction that does not account for the updated body position and velocity boundary conditions is computed through (28). Then the surface stress and the associated body position and velocity are computed through iteration in (29)–(30). Finally, in (31) the updated surface stress is used to remove the part of the streamfunction that does not satisfy the updated boundary conditions.

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

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 $E_{n+1}^{(k)} C s_{n+1} + (DE_{n+1}^{(k)} \Delta\chi) C s_{n+1} - \Delta\zeta = \zeta_{n+1}^{(k)}$. The DE term is of order $O(\Delta t)$ compared with the $\Delta\zeta$ term, since by the third block equation $\Delta\chi = O(\Delta\zeta \Delta t)$.

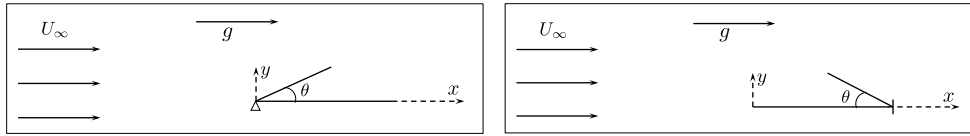


Fig. 1. Flow moves from left to right past a flag in the standard configuration (left) or the inverted configuration (right).

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 $(C^T AC)^{-1}$ 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 this requires a number of operations that scales with the number of points on the flow domain. Thus, the computation and storage of $(\hat{K}^{(k)})^{-1}$ may be small compared with solving the Poisson-like problem. In this case one may analytically perform a block-Gaussian elimination of (29) to get

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

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

One may therefore equivalently solve (28)–(31) with (29) replaced by (32)–(33), which has the benefit of avoiding the inhomogeneous block structure of (29). We use this formulation for the results shown below, as we found it 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 flag problems

We verify our method for several test problems of 2D flow past thin deforming flags. 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

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

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

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

Each configuration presents different challenges: flags 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. [21] 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 flag in the standard configuration

We first consider flow past a deforming flag in the standard configuration with $\text{Re} = 1000$, $M_\rho = 0.075$, $K_B = 0.0001$, and $\text{Fr} = 0$ (i.e., no gravitational body force). This choice of parameters leads to limit cycle flapping, which we triggered in our simulations by initializing the flag at $\theta = 1^\circ$ (see Fig. 1 for the definition of θ). We show the transverse tip displacement and lift coefficient in Fig. 2. After an initial transient, the trailing edge of the flag has oscillatory transverse displacement of fixed amplitude and frequency. The lift and tip displacement share the same frequency but are out of phase. As the flag nears its peak amplitude, the lift force opposes it enough to overcome the flag’s inertia, which initiates flag motion in the

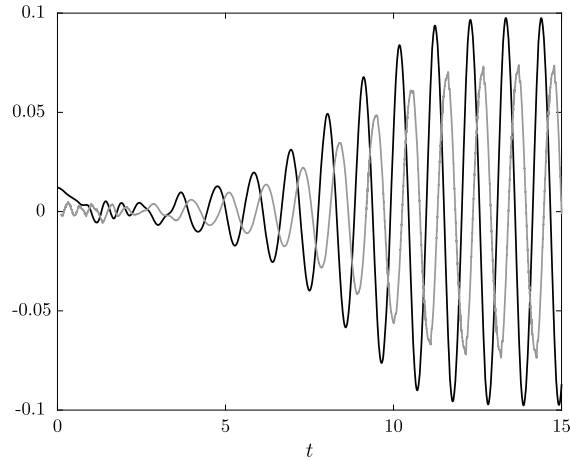


Fig. 2. Transverse displacement of the flag's trailing edge (black) and coefficient of lift (gray) for $Re = 1000$, $K_B = 0.0001$, and $Fr = 0$.

Table 1

Amplitude and frequency associated with the transverse displacement of the flag's trailing edge; obtained for $Re = 1000$, $M_\rho = 0.075$, $K_B = 0.0001$, and $Fr = 0$.

	Amplitude	Frequency (St)
Connell et al. [22]	± 0.096	0.93
Gurugubelli et al. [23]	± 0.098	0.95
Present	± 0.097	0.94

Table 2

Amplitude and frequency associated with the transverse displacement of the flag's trailing edge; obtained for $Re = 200$, $M_\rho = 1.5$, $K_B = 0.0015$, and $Fr = 1.4$.

	Amplitude	Frequency (St)
Huang et al. [24]	± 0.35	0.30
Wang et al. [5]	± 0.35	0.31
Lee et al. [25]	± 0.38	0.31
Present	± 0.38	0.32

other direction. This interplay of fluid and flag forces modulates and sustains the limit cycle flapping behavior seen for this set of parameters. Our computed amplitude and frequency of the flag tip displacement in the limit cycle regime agree well with published values for these quantities (see Table 1).

The finest domain for this simulation 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. There were a total of 960,000 points in the flow domain, and it took 30.83 hours to run this simulation on a single CPU core. 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.

Limit cycle flapping also occurs for $Re = 200$, $M_\rho = 1.5$, $K_B = 0.0015$, 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 flag at $\theta = 18^\circ$. 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. There were a total of 460,800 points in the flow domain, and it took 10.42 hours to run this simulation on a single CPU core. 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.

Fig. 3 gives a time history of the trailing edge transverse displacement for the parameters corresponding to Table 2 (results from references [24,25] are included for comparison). To emphasize the robustness of the method for a range of mass ratios, Fig. 3 also shows the trailing edge displacement for $M_\rho = 0.0001$ and $M_\rho = 100$. To our knowledge, the $M_\rho = 0.0001$ and $M_\rho = 100$ cases have not been simulated before. Moreover, we were not able to simulate the $M_\rho = 0.0001$ 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 flag with such small inertia.

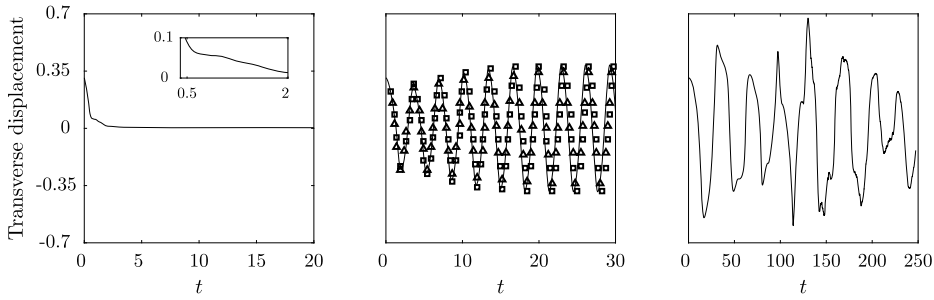


Fig. 3. Transverse displacement of the flag's trailing edge for $Re = 200$, $K_B = 0.0015$, and $Fr = 1.4$. Left panel: $M_\rho = 0.0001$, middle panel: $M_\rho = 1.5$, right panel: $M_\rho = 100$. Present: —, Lee et al. [25]: □, Huang et al. [24]: △. The insert on the left panel is a zoom-in of the $M_\rho = 0.0001$ case for $t \in [0.5, 2]$. Note the different horizontal axis values on each panels.

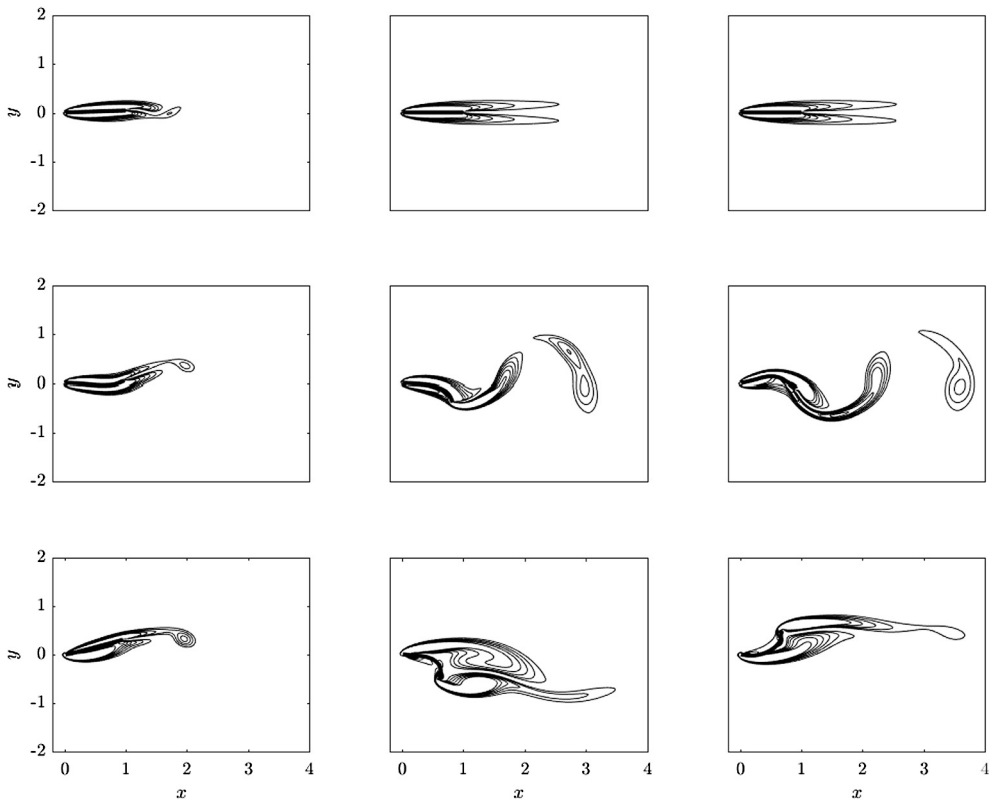


Fig. 4. Vorticity contours of flow past a deforming flag for $M_\rho = 0.0001$ (top row), $M_\rho = 1.5$ (middle row), and $M_\rho = 100$ (bottom row). Left column: $t = 1.2$; middle column: $t = 18$; right column: $t = 31.2$. Contours are in 15 evenly spaced increments from -5 to 5 . The other parameters were $Re = 200$, $M_\rho = 1.5$, and $Fr = 1.4$.

In Fig. 4, we show vorticity contours at different time instances for $M_\rho = 0.0001$, $M_\rho = 1.5$, and $M_\rho = 100$. When $M_\rho = 0.0001$, the impulsive start pushes the flag down quickly towards the $\theta = 0^\circ$ position. At $t \approx 0.5$, the vortical structure created during the impulsive start reaches the trailing edge of the flag. Due to the flag's small inertia, this substantially affects the trailing edge displacement (see insert in Fig. 3). The vortical structure then advects away from the body, and the fluid wake becomes symmetric as the flag fully reaches its $\theta = 0^\circ$ 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 flag does not allow for any minor flag deformations to break the flow symmetry. Thus, the flag stays in this neutral position for all remaining time. When $M_\rho = 1.5$, the limit cycle flapping of the flag is associated with a periodic vortex street [5,24,25]. As the flag becomes increasingly massive, the flapping amplitude increases. This is associated with a thicker, more irregular wake profile and more chaotic flapping behavior. Fig. 3 shows that the $M_\rho = 100$ case 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. [22].

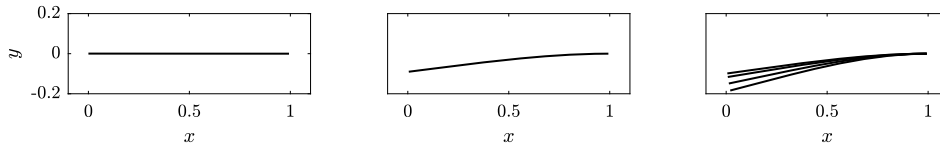


Fig. 5. Snapshots of the flag at various times for the fixed-point stable (left), static divergence (middle), and unstable flapping (right) regimes. Only one curve is visible for the fixed-point stable and static divergence cases, since the flag does not deflect about these equilibrium positions.

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 $Fr = 0$.

	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. [23]	UF	UF	UF	UF	UF	SD	SD	SD	SD	SD	SD	FPS
Present	UF	UF	UF	SD	SD	SD	SD	SD	SD	SD	SD	FPS

Table 4

Amplitude and frequency associated with the transverse displacement of the flag’s leading edge; obtained for $Re = 1000$, $M_\rho = 0.5$, $K_B = 0.4$, and $Fr = 0$.

	Amplitude	Frequency (St)
Gurugubelli et al. [23]	± 0.83	0.204
Present	± 0.82	0.198

4.2. Flow past a flag in the inverted configuration

An inverted flag initially placed at $\theta = 0^\circ$ can enter a wide range of possible dynamical regimes. Three of these are fixed-point stable (FS), static divergence (SD), or unstable biased flapping (UF) [23] (see Fig. 5 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 $h = 0.005$, and the time step was $\Delta t = 0.001$. Using $h = 0.004$ change our reported results by less than one percent. There were a total of 400,000 points in the flow domain, and it took 5.56 hours to run one simulation on a single CPU core. 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. [23] for the same parameter ranges.

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 flag 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 [23]. 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 $\Delta t = 0.0003$. Using $h = 0.0025$ did not change our reported results. There were a total of 1,800,000 points on the flow domain, and it took 106 hours to run the simulation on a single CPU core. 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.

Fig. 6 shows vorticity contours at four different times during a flapping cycle. The figure shows features consistent with what was observed by Gurugubelli et al. [23]. The fluid stresses associated with the leading edge vortex deform the flag in the downward direction (leftmost figure). The flag begins to flap back up towards the centerline once the internal flag stresses counteract the imposed fluid stresses and flag inertia, and the leading edge vortex detaches (second leftmost figure). The inertia of the flag causes it to continue to move towards the centerline while the detached vortex grows in size (second rightmost figure). The flag then flaps upward, 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 flag’s near-wake in Fig. 6. The upward motion of the flag creates an attached leading edge vortex, which starts an analogous process to the one just described. The result of this upward–downward flag motion is limit cycle flapping that repeats indefinitely.

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

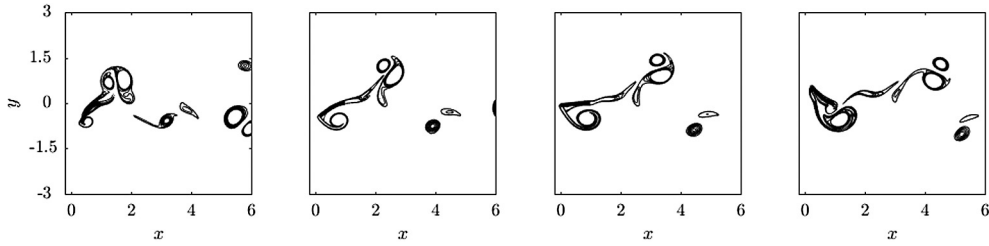


Fig. 6. Vorticity contours during a flapping cycle for $Re = 1000$, $M_\rho = 0.5$, $K_B = 0.4$, $Fr = 0$. Contours are in 15 evenly spaced increments from -5 to 5 .

iteration process does not involve heuristic relaxation parameters. We derived the method for general deforming surfaces, and verified it for 2D flow past deforming flags. 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.

Acknowledgements

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), and thanks Anthony Massari for many helpful discussion about corotational flag formulations. 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 semi-discrete FSI system is

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

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

$$\dot{\chi} = \zeta \quad (\text{A.3})$$

$$Du = 0 \quad (\text{A.4})$$

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

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 :

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

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

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

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

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

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

$$p_{n+1} = p^* - (G^T A G)^{-1}G^T A^{-1}E_{n+1}^{(k)T} f_{n+1} \quad (\text{A.11})$$

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

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

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

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

$$\Delta U = \frac{2}{\Delta t} (\hat{K}^{(k)})^{-1} (r^{\zeta}{}^{(k)} + Q W_{n+1}^{(k)} f_{n+1}^{(k)}) - r^{\chi}{}^{(k)} \quad (\text{A.14})$$

A similar block-LU based immersed boundary method was developed for bodies undergoing prescribed kinematics [12, 26]. The method we propose in (A.7)–(A.12) is different from these in the nonlinear iterations introduced in (A.9)–(A.10). Since these iterations are restricted to small dimensional subsystems, we do not believe they will drastically affect the effectiveness of the method.

References

- [1] P. Causin, J.-F. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid–structure problems, *Comput. Methods Appl. Mech. Eng.* 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, *Comput. Methods Appl. Mech. Eng.* 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, *J. Comput. Phys.* 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, *J. Comput. Phys.* 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, *J. Comput. Phys.* 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, *Comput. Struct.* 87 (11) (2009) 793–801.
- [7] Y. Mori, C.S. Peskin, Implicit second-order immersed boundary methods with boundary mass, *Comput. Methods Appl. Mech. Eng.* 197 (25) (2008) 2049–2067.
- [8] G. Hou, J. Wang, A. Layton, Numerical methods for fluid–structure interaction—a review, *Commun. Comput. Phys.* 12 (02) (2012) 337–377.
- [9] J. Yang, F. Stern, Sharp interface immersed-boundary/level-set method for wave–body interactions, *J. Comput. Phys.* 228 (17) (2009) 6590–6616.
- [10] U. Lăcis, K. Taira, S. Bagheri, A stable fluid–structure-interaction solver for low-density rigid bodies using the immersed boundary projection method, *J. Comput. Phys.* 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, *Comput. Methods Appl. Mech. Eng.* 197 (25) (2008) 2131–2146.
- [12] K. Taira, T. Colonius, The immersed boundary method: a projection approach, *J. Comput. Phys.* 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, *J. Comput. Phys.* 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, *J. Comput. Phys.* 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, *J. Comput. Phys.* 228 (20) (2009) 7821–7836.
- [17] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [18] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, *J. Comput. Phys.* 153 (2) (1999) 509–534.
- [19] M. Criesfield, *Non-linear Finite Element Analysis of Solids and Structures*, vol. 1, Wiley, New York, 1991.
- [20] J.B. Perot, An analysis of the fractional step method, *J. Comput. Phys.* 108 (1) (1993) 51–58.
- [21] B. Kallemov, A. Bhalla, B.E. Griffith, A. Donev, An immersed boundary method for rigid bodies, *Commun. Appl. Math. Comput. Sci.* 11 (1) (2016) 79–141.
- [22] B.S. Connell, D.K. Yue, Flapping dynamics of a flag in a uniform stream, *J. Fluid Mech.* 581 (2007) 33–67.
- [23] P. Gurugubelli, R. Jaiman, Self-induced flapping dynamics of a flexible inverted foil in a uniform flow, *J. Fluid Mech.* 781 (2015) 657–694.
- [24] W.-X. Huang, S.J. Shin, H.J. Sung, Simulation of flexible filaments in a uniform flow by the immersed boundary method, *J. Comput. Phys.* 226 (2) (2007) 2206–2228.
- [25] I. Lee, H. Choi, A discrete-forcing immersed boundary method for the fluid–structure interaction of an elastic slender body, *J. Comput. Phys.* 280 (2015) 529–546.
- [26] S. Liska, T. Colonius, A fast immersed boundary method for external incompressible viscous flows using lattice Green’s functions, *J. Comput. Phys.* 331 (2016) 257–279.