This is the authors' final draft of this article that was published in the Journal of Fluid Mechanics, Vol. 898, A9 (2020). (doi:10.1017/jfm.2020.353)

Reference Map Technique for Incompressible Fluid–Structure Interaction

Chris H. Rycroft[1,2]†, Chen-Hung Wu[1,3], Yue Yu[4], and Ken Kamrin[3]‡

[1]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02139, USA

[2]Mathematics Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA

[3]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[4]Department of Mathematics, Lehigh University, Bethlehem, PA 18015, USA

(Received xx; revised xx; accepted xx)

We present a general simulation approach for fluid–solid interactions based on the fully-Eulerian reference map technique (RMT). The approach permits the modeling of one or more finitely-deformable continuum solid bodies interacting with a fluid and with each other. A key advantage of this approach is its ease of use, as the solid and fluid are discretized on the same fixed grid, which greatly simplifies the coupling between the phases. We use the method to study a number of illustrative examples involving an incompressible Navier–Stokes fluid interacting with multiple neo-Hookean solids. Our method has several useful features including the ability to model solids with sharp corners and the ability to model actuated solids. The latter permits the simulation of active media such as swimmers, which we demonstrate. The method is validated favorably in the flag-flapping geometry, for which a number of experimental, numerical, and analytical studies have been performed. We extend the flapping analysis beyond the thin-flag limit, revealing an additional destabilization mechanism to induce flapping.

1. Introduction

Fluid–structure interaction (FSI) problems highlight a natural dichotomy in the simulation approaches for solids and fluids, where fluid problems tend to be solved using Eulerian-frame methods (Chorin 1967; Hirt *et al.* 1974; Versteeg & Malalasekera 1995; Tannehill *et al.* 1997) and solids with Lagrangian approaches (Zienkiewicz & Taylor 1967; Sulsky *et al.* 1994; Hoover 2006; Belytschko *et al.* 2013). An FSI simulation method must therefore bridge the gap between these two perspectives. For example, one set of FSI approaches treats both fluid and solid phases in a Lagrangian frame, with a finite-element representation in the solid and an adaptive Lagrangian mesh in the fluid (Rugonyi & Bathe 2001; Bathe 2007; Froehle & Persson 2015), or with both phases treated with a mesh-free approach (Rabczuk *et al.* 2010). An alternative methodology is to treat the fluid on a fixed Eulerian mesh and the solid with Lagrangian points, such as the family of immersed boundary methods developed by Peskin (1972*a*,*b*, 1977, 2002), which have been extensively used to simulate membranes (Griffith *et al.* 2009; Fai *et al.* 2013), and obstacles (Coquerelle & Cottet 2008; Gazzola *et al.* 2011; Engels *et al.* 2015) via Brinkman volume penalization.

A fully Eulerian method whereby fluid and solid are both computed on a fixed grid

---

† Email address for correspondence: chr@seas.harvard.edu
‡ Email address for correspondence: kkamrin@mit.edu

has its advantages. Computation time benefits arise from both phases being treated on a single fixed background grid. The handling of multiple objects interacting or of topological changes to objects can be done with level set fields (Osher & Sethian 1988; Sethian 1999; Osher & Fedkiw 2003) rather than requiring complex on-the-fly Lagrangian remeshing. In addition, certain common conditions such as incompressibility are easier to implement in an Eulerian form. Lastly, fixed-grid approaches are well suited to numerical analysis, such as a von Neumann stability analysis (LeVeque 2007).

The key challenge for a fully-Eulerian FSI method is to develop an Eulerian description of the solid. In a small strain limit, this can be achieved by writing the equations of linear elasticity in rate form, referred to as hypoelasticity (Truesdell 1955), which has formed the basis of several numerical techniques (Udaykumar *et al.* 2003; Rycroft & Gibou 2012; Rycroft *et al.* 2015). However, here our aim is to develop a large-deformation description of the solid, the more general approach in solid mechanics (Gurtin *et al.* 2010; Belytschko *et al.* 2013). This has attracted interest over the past three decades, with a variety of different approaches being explored in the literature, such as the conservative first-order method of Plohr & Sharp (1988), the deformation gradient-based method of Liu & Walkington (2001), and the initial point set (IPS) method of Dunne (2006). See Sec. 2.2 for further references and a comprehensive discussion.

In recent years, we have contributed to this field by developing an Eulerian-frame solid simulation approach called the reference map technique (RMT) (Kamrin 2008; Kamrin & Nave 2009; Kamrin *et al.* 2012; Valkov *et al.* 2015), which is based on tracking the reference map field—*i.e.* where material started from—as the primary simulation variable on the Eulerian grid. The reference map field allows the finite deformation of the solid to be computed, from which the material stress is calculated according to a prescribed nonlinear constitutive law. This approach has shown the ability to simulate basic FSI and separately cover a span of desirable features. However, a single implementation covering all needed features for robust physical simulation—*e.g.* (i) numerical stability, (ii) good convergence properties for fluid and solid phases, and (iii) desirable physical traits such as the ability to model incompressible materials, objects with sharp corners, and activated media—has been lacking and non-trivial to produce. In this paper we present such a method and provide a variety of physical simulations using it, which extend our understanding of certain FSI problems.

To represent incompressible solids and fluids we have reformulated the numerical discretization using the projection method framework of Chorin (1967, 1968). In this method, to integrate the velocity field forward by a time step, an intermediate velocity field is computed where the incompressibility constraint is temporarily relaxed. After this, a Poisson problem is solved for the pressure, which is used to project the velocity to be divergence-free. The method has been extensively developed since Chorin's original work (Brown *et al.* 2001). Here, we consider a modern second-order implementation developed by Almgren, Bell, and coworkers (Bell *et al.* 1989; Puckett *et al.* 1997; Almgren *et al.* 1998). The implementation uses the discretization of Colella (1990) for accurately calculating advective terms. It employs the approximate projection approach of Almgren *et al.* (1996) based on a finite-element discretization. The implementation was subsequently extended to two-phase flows using a level set method (Sussman *et al.* 1999), which was used to simulate an inkjet printer nozzle (Yu *et al.* 2003, 2007). In our approach we deliberately keep the fluid component of the simulation to match this existing implementation, to emphasize that the reference map technique does not require any special treatment of the fluid. However, we show that the discretization techniques can be generalized to simulate solids via the RMT, and we find that the advective discretization is especially well suited to simulating the reference map update equations in a fashion more accurate than the approach of Valkov *et al.* (2015).

The projection method removes the Courant–Friedrichs–Lewy (CFL) condition (Courant *et al.* 1967) associated with pressure waves. This makes it possible to simulate a wide variety of problems in an intermediate Reynolds number regime (and potentially for high Reynolds problems should an adaptive background grid be used). Following Valkov *et al.* (2015), the level set field representing interface(s) is not explicitly updated, but is tied to where the boundary should be in the reference map field. However, here we switch to a regression-based extrapolation method, which is more stable, simpler, and allows shapes with corners to be considered. Two physically-motivated examples of a flexible rotor and a paddle are presented, and a variety of convergence and performance tests are provided in Appendices B, C, D, & E. Our numerical tests show that the method is first-order accurate for a typical fluid–solid simulation, with the interface being the largest source of error. With respect to the L^2 norm, second-order accuracy is achieved for a fluid-only discretization, and for a solid-only discretization with specific choices for numerical damping.

With the properties of the method established, we consider the flag flapping stability problem, which has been studied extensively (Zhang *et al.* 2000; Watanabe *et al.* 2002; Zhu & Peskin 2002; Connell & Yue 2007). We can quantitatively reproduce the phase plot of stability for a thin flag (Connell & Yue 2007) with very good accuracy for Reynolds numbers below 1000, and reasonably good accuracy for Reynolds numbers above 1000. Our method also makes it possible to simulate flags with substantial thickness, which show a different instability mechanism due to vortex shedding from the tip. The transition between the thin and thick flag behaviors is captured and studied with our method. We also augment the approach to allow internal actuation of the solid bodies. With this addition, the method is well suited to biolocomotion problems and we show an example of this by modeling a jellyfish-like swimmer. Another advantage of the method is the ability to perform many-body contact problems quickly but in a fashion that accurately balances momentum. We demonstrate this approach with two examples of many objects of various sizes and densities settling under gravity. All computer code to generate the results in this paper is released as open source software (Appendix F).

2. Theory

2.1. *Overview of the reference map technique*

We begin by considering the solid material, which we model using the large-deformation hyperelastic framework (Lubliner 2008; Gurtin *et al.* 2010). As shown in Fig. 1(a), we introduce an undeformed reference configuration for the solid at time $t = 0$ with coordinate system $\mathbf{X}$. We then consider a time-dependent map $\boldsymbol{\chi}(\mathbf{X}, t)$ from the undeformed configuration to the deformed state in the physical frame at time t. The deformation gradient tensor is defined as

$$\mathbf{F} = \frac{\partial \boldsymbol{\chi}}{\partial \mathbf{X}} \tag{2.1}$$

and represents how an infinitesimal element of the solid has been deformed and rotated. From here, a constitutive law

$$\boldsymbol{\sigma}_s = \mathbf{f}(\mathbf{F}, \boldsymbol{\zeta}) \tag{2.2}$$

can be used to calculate the Cauchy stress $\boldsymbol{\sigma}_s$ in the physical frame, where $\boldsymbol{\zeta}$ represents any internal state variables such as plastic deformation. The material velocity $\mathbf{v}(\mathbf{x}, t)$ then

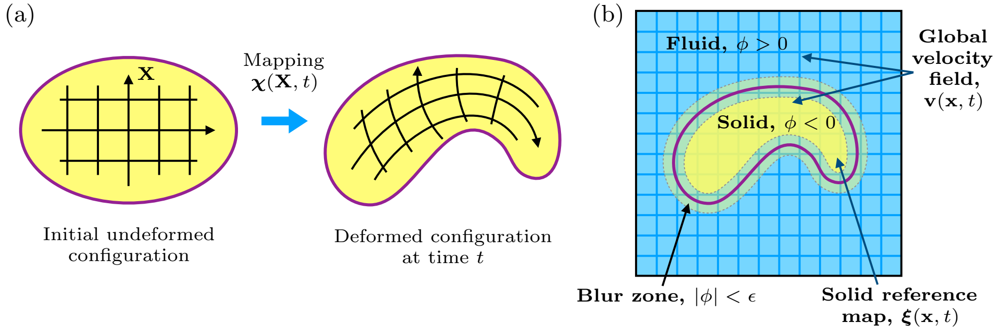

FIGURE 1. (a) Overview of the hyperelastic framework, whereby an initially undeformed solid with reference coordinate system $\mathbf{X}$ undergoes a time-dependent mapping $\boldsymbol{\chi}(\mathbf{X},t)$ to its current configuration at time t. (b) Overview of the reference map technique for simulating fluid–structure interaction on a fixed background grid. The sign of the level set function $\phi(\mathbf{x},t)$ demarcates the solid and fluid phases. The blur zone, used to transition from solid to fluid stress, is defined as the region where $|\phi| < \epsilon$.

satisfies

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v}\right) = \nabla\cdot\boldsymbol{\sigma} \tag{2.3}$$

where $\boldsymbol{\sigma} = \boldsymbol{\sigma}_s$ in this case, $\rho = \rho_s/(\det\mathbf{F})$, and ρ_s is the solid density in the undeformed configuration.

The most commonly used approach to simulate hyperelastic solids is to introduce a deforming mesh on the solid, and then solve for the nodal displacements, from which Eq. (2.1) can be used to compute the stress (Belytschko *et al.* 2013). However, here we take an alternative approach of introducing the *reference map* field in the physical frame $\boldsymbol{\xi}(\mathbf{x},t)$ that represents the inverse mapping of $\boldsymbol{\chi}$. The field is initialized as $\boldsymbol{\xi}(\mathbf{x},0) = \mathbf{x}$, and satisfies the advection equation

$$\frac{\partial\boldsymbol{\xi}}{\partial t} + (\mathbf{v}\cdot\nabla)\boldsymbol{\xi} = \mathbf{0}. \tag{2.4}$$

The deformation gradient tensor is computed from the reference map field according to

$$\mathbf{F} = \left(\frac{\partial\boldsymbol{\xi}}{\partial\mathbf{x}}\right)^{-1}, \tag{2.5}$$

from which the Cauchy stress is evaluated. Equations (2.2), (2.3), (2.4), & (2.5) then form a minimal system of equations for finite-strain hyperelasticity in an Eulerian frame. The reference map $\boldsymbol{\xi}(\mathbf{x},t)$ and velocity $\mathbf{v}(\mathbf{x},t)$ can be represented on a fixed grid. At each timestep Eqs. (2.5) & (2.2) can be used to evaluate the Cauchy stress, after which Eqs. (2.3) & (2.4) can be integrated forward in time. So far, this prescription is general, and could be solved using a variety of discretization approaches such as a finite-difference method, finite-volume method, or a discontinuous Galerkin method.

2.2. *Other related approaches*

The reference map is a standard definition in solid mechanics (Gurtin *et al.* 2010), and it has been used in problems of inverse design (Govindjee & Mihalic 1996; Fachinotti *et al.* 2008), but it is not widely employed as a primary simulation variable in the physical frame. Fixed-grid approaches by Plohr & Sharp (1988), Trangenstein & Colella (1991), and Liu & Walkington (2001) have been developed that use the deformation gradient tensor $\mathbf{F}$ as a primary simulation variable. Takagi and coworkers have developed a related approach based on using the left Cauchy–Green deformation tensor $\mathbf{B} = \mathbf{F}\mathbf{F}^{\mathsf{T}}$ as the primary simulation variable since this quantity features in many constitutive laws (Takagi *et al.* 2011; Sugiyama *et al.* 2011; Ii *et al.* 2012). Methods based on $\mathbf{F}$ and $\mathbf{B}$ have the advantage of requiring a single spatial derivative to compute acceleration, whereas our approach requires two, in Eqs. (2.5) & (2.3). However, the reference map contains additional information for locating the boundary and has fewer components to store. Using $\mathbf{F}$ requires that a gauge constraint is imposed to ensure that it remains a valid gradient of a material mapping.

Cottet *et al.* (2008) and Maitre *et al.* (2009) independently developed simulation approaches using the reference map, and stated its potential to be used as part of a general fluid–structure simulation. Further work by Milcent & Maitre (2016) simulated an immersed interface with full membrane elasticity. Bellotti & Theillard (2019) coupled the reference map to the level-set method for improved tracking of interfaces in two-phase flow simulation.

The IPS method of Dunne (2006) is closely related to our approach. The IPS method is based on using finite elements to track the solid displacements $\mathbf{u}$ as the primary simulation variable. However, the reference map field (referred to by Dunne as the *set of initial positions*) emerges as part of the computational procedure, via the relationship $\mathbf{u} = \mathbf{x} - \boldsymbol{\xi}$. The IPS method has been developed further (Dunne *et al.* 2010; Richter 2013; Wick 2013), and has the advantage that the Eulerian field $\mathbf{u}$ can be used to track the deformation of the fluid–structure interface, in a similar manner to the approach we discuss in Sec. 3.2.

2.3. *Incompressible fluid–structure interaction*

In this paper we employ the reference map technique to simulate incompressible fluid–structure interactions. We shall use the terms $\boldsymbol{\tau}$, $\boldsymbol{\tau}_s$, and $\boldsymbol{\tau}_f$ to refer only to the deviatoric part of the stress, as the pressure field is now deformation independent and separately calculated. We make use of a globally defined velocity field $\mathbf{v}(\mathbf{x}, t)$ that satisfies the incompressibility constraint

$$\nabla \cdot \mathbf{v} = 0. \tag{2.6}$$

We consider a solid immersed within the fluid, and introduce a level set function $\phi(\mathbf{x}, t)$ (Sethian 1999; Osher & Fedkiw 2003) that is the signed distance to the solid–fluid interface with the convention that $\phi < 0$ in the solid and $\phi > 0$ in the fluid. The reference map $\boldsymbol{\xi}(\mathbf{x}, t)$ is defined within the solid region only.

Let the fluid have density ρ_f and dynamic viscosity μ_f. The fluid stress deviator at any gridpoint is given by

$$\boldsymbol{\tau}_f = \mu_f(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathsf{T}}). \tag{2.7}$$

Kinematic viscosity is defined as $\nu_f = \mu_f/\rho_f$. The deviatoric stress is then defined as a smooth transition between the fluid and solid stresses,

$$\boldsymbol{\tau} = \boldsymbol{\tau}_s + H_\epsilon(\phi)(\boldsymbol{\tau}_f - \boldsymbol{\tau}_s), \tag{2.8}$$

where

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi \leqslant -\epsilon, \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\frac{\pi\phi}{\epsilon}) & \text{if } |\phi| < \epsilon, \\ 1 & \text{if } \phi \geqslant \epsilon, \end{cases} \tag{2.9}$$

is a smoothed Heaviside function with a transition region of width 2ϵ. Here we use a twice-differentiable form for H_ϵ that has been employed in previous studies (Sussman

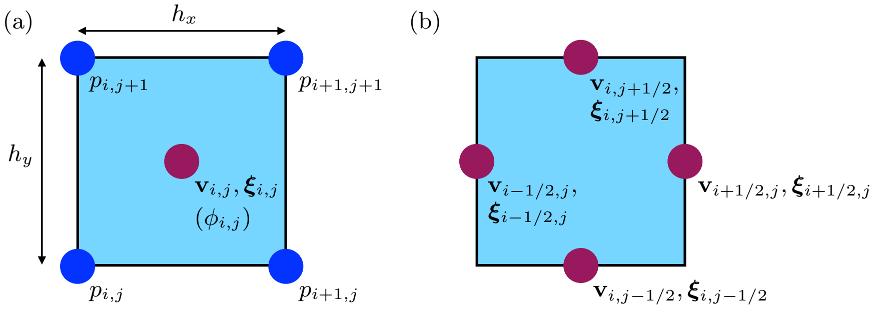

FIGURE 2. (a) Arrangement of the fields within a simulation grid cell. The reference map $\boldsymbol{\xi}_{i,j}$, velocity $\mathbf{v}_{i,j}$, and level set field $\phi_{i,j}$ are held at the cell center, while the pressure is held at the cell corners. The level set field is bracketed to emphasize that it is not time-evolved independently, but is instead derived from the reference map. (b) Arrangement of the edge velocities and reference maps that are computed at the half-timestep to evaluate the advective terms.

et al. 1994, 1999; Yu *et al.* 2003, 2007). For more details on the choice of ϵ and the precise form of H_ϵ, see Appendix D. In order to calculate $\boldsymbol{\tau}$ it is necessary to smoothly extend $\boldsymbol{\xi}$ in the region $0 < \phi < \epsilon$, which is done using extrapolation methods that will be described in the following section. The density is also defined as a smooth transition between the solid and fluid, as

$$\rho = \rho_s + H_\epsilon(\phi)(\rho_f - \rho_s). \tag{2.10}$$

Formally, our limit of interest is when $\epsilon \to 0$, when there is a sharp interface between the fluid and solid. In our numerical method, we choose ϵ to scale proportionally with the grid spacing, and thus we approach this limit as the simulation resolution is increased.

3. Numerical Method

The numerical procedure is based on the projection method of Chorin (1967, 1968) for solving the incompressible Navier–Stokes equations. Specifically, we consider a modern second-order method described by Almgren *et al.* (1998) that is especially effective at dealing with advection, and incorporates a number of algorithmic advancements from Chorin's original treatment.

The simulation domain is a rectangle of size W by H, and is divided into an $M \times N$ grid of rectangular cells of size h_x by h_y. Following the work of Colella (1990), the velocity, the reference map, and the level set are held at cell centers and are indexed as $\mathbf{v}_{i,j}$, $\boldsymbol{\xi}_{i,j}$, and $\phi_{i,j}$, respectively, for $i = 0, \ldots, M-1$ and $j = 0, \ldots, N-1$ (Fig. 2(a)). The components of the velocity field are written as $\mathbf{v} = (u, v)$. Pressures are held at cell corners and are indexed as $p_{i,j}$ for $i = 0, \ldots, M$ and $j = 0, \ldots, N$. In addition, the grid is padded by two layers of cells in each direction whose values are populated to enforce different boundary conditions.

Superscripts are used to denote timesteps. To advance the simulation forward from timestep n to $n+1$ with interval Δt, the following procedure is used. The reference map field is updated using

$$\frac{\boldsymbol{\xi}^{n+1} - \boldsymbol{\xi}^n}{\Delta t} = -\left[(\mathbf{v} \cdot \nabla)\boldsymbol{\xi}\right]^{n+1/2} \tag{3.1}$$

and an intermediate velocity $\mathbf{v}^*$ is computed using

$$\frac{\mathbf{v}^* - \mathbf{v}^n}{\Delta t} = -\left[(\mathbf{v} \cdot \nabla)\mathbf{v}\right]^{n+1/2} + \frac{1}{\rho(\phi^{n+1/2})} \nabla \cdot \left[\boldsymbol{\tau}(\boldsymbol{\xi}^{n+1/2}, \mathbf{v}^n)\right]. \tag{3.2}$$

Here, the advective derivatives $[(\mathbf{v}\cdot\nabla)\boldsymbol{\xi}]^{n+1/2}$ and $[(\mathbf{v}\cdot\nabla)\mathbf{v}]^{n+1/2}$ are evaluated at the middle of the timestep using a second-order explicit Godunov scheme, described in Sec. 3.1. Once the advective derivatives are evaluated, Eq. (3.1) allows $\boldsymbol{\xi}^{n+1}$ to be computed. This allows the time-centered reference map to be defined as $\boldsymbol{\xi}^{n+1/2} = (\boldsymbol{\xi}^{n} + \boldsymbol{\xi}^{n+1})/2$ after which $\mathbf{v}^*$ is computed using Eq. (3.2). From here, the Poisson problem for pressure is evaluated using

$$\nabla\cdot\mathbf{v}^* = \nabla\cdot\left(\frac{\Delta t}{\rho(\phi^{n+1/2})}\nabla p^{n+1}\right). \tag{3.3}$$

Following Almgren *et al.* (1996) and Puckett *et al.* (1997), Eq. (3.3) is solved using a finite-element formulation, described in Sec. 3.4. After this, the velocity is projected to be divergence-free using

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{\Delta t}{\rho(\phi^{n+1/2})}\nabla p^{n+1} \tag{3.4}$$

where the gradient of p^{n+1} is evaluated using a second-order centered difference formula.

3.1. *Advective terms*

To evaluate the advective terms appearing in Eqs. (3.1) and (3.2), a second-order explicit Godunov scheme is used. The same scheme is applied to both the velocity $\mathbf{v}$ and reference map $\boldsymbol{\xi}$. Throughout this section, we denote a to be a generic scalar component of either of these two fields. We also refer the reader to recent work by Jain & Mani (2017), which introduces an alternative numerical treatment for reference map advection.

3.1.1. *Godunov upwinding*

To begin, the gradients of the reference map and velocity at each cell center are evaluated using the fourth-order monotonicity-limited scheme of Colella (1985) described in Appendix A.1. Once the gradients are calculated at the center of each cell, edge-centered velocities and reference maps are created at $t+\Delta t/2$ using Taylor expansions to each of the four edges, which are indexed using half-integers as shown in Fig. 2(b). As an example, an extrapolation of the reference map to a vertical edge from the left (with superscript L) is given by

$$\begin{aligned}\boldsymbol{\xi}^{L,n+1/2}_{i+1/2,j} &= \boldsymbol{\xi}^{n}_{i,j} + \frac{\Delta t}{2}(\partial_t\boldsymbol{\xi})^n_{i,j} + \frac{h_x}{2}(\partial_x\boldsymbol{\xi})^n_{i,j} \\ &= \boldsymbol{\xi}_{i,j} + \frac{1}{2}\left(h_x - u^n_{i,j}\Delta t\right)\boldsymbol{\xi}^n_{x,i,j} - \frac{\Delta t}{2}(\widetilde{v\boldsymbol{\xi}_y})^n_{i,j},\end{aligned} \tag{3.5}$$

where Eq. (2.4) has been substituted for the $\boldsymbol{\xi}_t$ derivative. The extrapolation of the velocity from the left to this edge is

$$\begin{aligned}\mathbf{v}^{L,n+1/2}_{i+1/2,j} &= \mathbf{v}^{n}_{i,j} + \frac{\Delta t}{2}\mathbf{v}^n_{t,i,j} + \frac{h_x}{2}\mathbf{v}^n_{x,i,j} \\ &= \mathbf{v}^n_{i,j} + \frac{1}{2}\left(h_x - u^n_{i,j}\Delta t\right)\mathbf{v}^n_{x,i,j} - \frac{\Delta t}{2}(\widetilde{v\mathbf{v}_y})^n_{i,j} - \frac{\Delta t}{2}\mathbf{a}^n_{i,j},\end{aligned} \tag{3.6}$$

where Eq. (2.3) has been substituted to replace the $\mathbf{v}^n_{t,i,j}$ term, and $\mathbf{a}^n_{i,j}$ is defined according to

$$\mathbf{a}^n_{i,j} = \left[-\frac{1}{\rho(\phi)}\nabla p + \frac{1}{\rho(\phi)}\nabla\cdot\boldsymbol{\tau}\right]^n_{i,j}. \tag{3.7}$$

Equivalent procedures are used to compute extrapolations from the right, down, and up with superscripts R, D, and U, respectively. To ensure tangential stability the terms with tildes in Eqs. (3.5) & (3.6) are computed differently using the procedure in Appendix A.2.

After this procedure, each edge has velocities and reference maps from the two cells that adjoin it, and a Godunov upwinding procedure is used to select which values to use. On the vertical edge at $(i+{}^1\!/_2, j)$,

$$a^{n+1/2}_{i+1/2,j} = \begin{cases} a^{L,n+1/2}_{i+1/2,j} & \text{if } u^{L,n+1/2}_{i+1/2,j} > 0 \text{ and } u^{L,n+1/2}_{i+1/2,j} + u^{R,n+1/2}_{i+1/2,j} > 0, \\ a^{R,n+1/2}_{i+1/2,j} & \text{if } u^{R,n+1/2}_{i+1/2,j} < 0 \text{ and } u^{L,n+1/2}_{i+1/2,j} + u^{R,n+1/2}_{i+1/2,j} < 0, \\ \mathcal{F}(a^{L,n+1/2}_{i+1/2,j}, a^{R,n+1/2}_{i+1/2,j}) & \text{otherwise.} \end{cases} \tag{3.8}$$

where a is a generic component. Thus if the velocity field points rightward then the components are taken from the left cell, and if the velocity field points leftward then the components are taken from the right cell. The function $\mathcal{F}$ is used when the two velocities are ambiguous. For the horizontal velocity $\mathcal{F}(\beta,\gamma)=0$ (Case A), and for all other components $\mathcal{F}(\beta,\gamma)=(\beta+\gamma)/2$ (Case B). On an edge where a velocity boundary condition is applied (*e.g.* a no-slip condition) the corresponding edge velocity is set to exactly match the condition. In this paper we restrict to cases of localized solid objects that do not extend to the boundary and thus we do not apply special boundary condition treatment for edge reference map fields.

3.1.2. *Marker-and-cell (MAC) projection*

The edge velocities calculated in Sec. 3.1.1 may not be precisely divergence free. We therefore apply an intermediate MAC projection step to ensure that the discrete flux entering any grid cell is exactly zero. Let $\mathbf{v}_\mathrm{e}$ be the edge velocities, and let q be a cell-centered scalar field. We aim to make

$$\mathbf{v}_\mathrm{e} - \frac{1}{\rho}\nabla q \tag{3.9}$$

divergence free. Taking the divergence of Eq. (3.9) yields

$$\nabla \cdot \left(\frac{1}{\rho}\nabla q\right) = \nabla \cdot \mathbf{v}_\mathrm{e}, \tag{3.10}$$

which is discretized as

$$\begin{aligned} \frac{1}{h_x^2}\left(\frac{q_{i+1,j}-q_{i,j}}{\rho_{i+1/2,j}} - \frac{q_{i,j}-q_{i-1,j}}{\rho_{i-1/2,j}}\right) + \frac{1}{h_y^2}\left(\frac{q_{i,j+1}-q_{i,j}}{\rho_{i,j+1/2}} - \frac{q_{i,j}-q_{i,j-1}}{\rho_{i,j-1/2}}\right) \\ = \frac{u_{i+1/2,j}-u_{i-1/2,j}}{h_x} + \frac{v_{i,j+1/2}-v_{i,j-1/2}}{h_y}. \end{aligned} \tag{3.11}$$

Edge-based densities appearing in this equation are computed via linear interpolation from the two adjacent grid cells. At boundaries where a velocity boundary condition is applied, any derivative on the left hand side of Eq. (3.11) is omitted if it contains q values that are out of range. If a pressure condition is applied, then a Dirichlet condition of $q = \Delta t\, p/2$ is applied, where the factor of two arises because the edge velocities are time-centered.

Equation (3.11) results in a large linear system $Aq=b$ where A is a sparse matrix, b is the source term, and q is a vector of the components $q_{i,j}$, which we solve using the geometric multigrid method with a standard V-cycle iteration (Demmel 1997). Since the q field typically varies smoothly in time, the initial guess for the multigrid method is computed as a linear interpolation from the previous two timesteps. Multigrid V-cycles are performed until the root mean squared (RMS) element in the residual vector $r = Aq - b$ reaches a required tolerance T_MAC. We assume that velocities and densities are within several orders of magnitude of unity. Then an appropriate scale for an element of the residual is $r_{\mathrm{s}} = 4(h_x^{-2} + h_y^{-2})\Delta t$, and a tolerance of $T_{\mathrm{MAC}} = 10^4 r_s \epsilon_m$ is used, where ϵ_m is the machine epsilon for double precision floating point arithmetic. Once the tolerance level is reached, one further V-cycle is performed to further improve accuracy. See Appendices C & F for more details on the multigrid code library and performance.

3.1.3. *Evaluation of the derivative*

Once the MAC projection has been performed the time-centered advective terms for the velocity and reference maps are evaluated as

$$[(\mathbf{v}\cdot\nabla)a]_{i,j}^{n+1/2} = \frac{u_{i+1/2,j}^{n+1/2} + u_{i-1/2,j}^{n+1/2}}{2}\,\frac{a_{i+1/2,j}^{n+1/2} - a_{i-1/2,j}^{n+1/2}}{h_x} + \frac{v_{i,j+1/2}^{n+1/2} + v_{i,j-1/2}^{n+1/2}}{2}\,\frac{a_{i,j+1/2}^{n+1/2} - a_{i,j-1/2}^{n+1/2}}{h_y} \tag{3.12}$$

where a is a generic field component.

3.2. *Level set update and reference map extrapolation*

The simulation makes use of a cell-centered level set function $\phi_{i,j}$ for tracking the fluid–solid boundary. The level set routine is stored in a narrow band of points of width $2\phi_W$ surrounding the interface (Sethian 1999; Rycroft & Gibou 2012) that is chosen to be large enough to contain the entire blur zone and perform finite-difference calculations at all points in this region. The level set is used to extrapolate the reference map fields in the narrow band, and to calculate the mixing of stress and density according to Eqs. (2.8) & (2.10), respectively. Unlike typical applications of the level set method, the ϕ field is not explicitly updated, but is instead continually given by the reference map field using the procedure first described by Valkov *et al.* (2015).

3.2.1. *Level set construction*

For a given shape, define a continuous function of the reference map $\phi_0(\boldsymbol{\xi})$ such that $\phi_0 < 0$ for reference map values in the solid, $\phi_0 > 0$ for reference map values outside the solid, and $\phi_0 = 0$ on the interface. During the timestep, the reference map field $\boldsymbol{\xi}^{n+1}$ is computed inside the solid using Eq. (3.1), from which the half-timestep reference map is defined as $\boldsymbol{\xi}^{n+1/2} = (\boldsymbol{\xi}^n + \boldsymbol{\xi}^{n+1})/2$. Both fields are extended into the narrow band fluid region using the extrapolation methods described in Sec. 3.2.2.

To construct the new level set function $\phi^{n+1/2}$, an auxiliary function $\psi^{n+1/2}$ is first computed in the narrow band such that $\psi^{n+1/2} = \phi_0(\boldsymbol{\xi}^{n+1/2})$. The zero contour of $\psi^{n+1/2}$ will lie at the fluid–solid interface, but this function itself may not satisfy the signed-distance property. To recover the signed-distance property, the level set $\phi^{n+1/2}$ is constructed from $\psi^{n+1/2}$ using the reinitialization procedure described by Rycroft & Gibou (2012). This procedure first considers points (i, j) that straddle the interface, so that one of their orthogonal neighbors has a $\psi^{n+1/2}$ value of an opposite sign. Each straddling point is considered. The bicubic interpolant $\psi_{\mathrm{c}}^{n+1/2}$ is computed, and the modified Newton–Raphson approach of Chopp (2001, 2009) is used to find the shortest distance vector $\boldsymbol{\Delta}\mathbf{x}$ from each straddling point to the interface $\psi_{\mathrm{c}}^{n+1/2}(\mathbf{x}) = 0$, after which the level set function is initialized to $\pm|\boldsymbol{\Delta}\mathbf{x}|$. In extremely rare cases the root-finding method can fail, in which case the routine falls back on the first-order method described by Sethian (1999). For further details, see the paper by Rycroft & Gibou (2012).

With the straddling points of $\phi^{n+1/2}$ now initialized, the remaining points are filled in using the second-order fast marching method of Sethian (1999). In the fluid, the positive

$\phi^{n+1/2}$ values are computed in order of increasing value, until reaching a cutoff ϕ_W that defines the width of the narrow band. The same procedure is used to fill in the negative $\phi^{n+1/2}$ values in the solid, until reaching a cutoff $-\phi_W$. After this procedure, the level set function is now a signed-distance function inside the narrow band. Note that these routines work reliably even if the function $\psi_{n+1/2}$ has a loss of regularity as some points: since the entire $\phi^{n+1/2}$ field is directly constructed, there is no possibility for instabilities to grow over time, as can happen in update procedures based on partial differential equations (PDEs). Identical methods are used to construct ϕ^{n+1} from $\boldsymbol{\xi}^{n+1}$.

3.2.2. *Extrapolation*

During the construction of the level set function, a list of fluid gridpoints sorted in order of increasing value, $0 < \phi_1 < \phi_2 < \ldots$ is constructed, which is used for extrapolating the reference map $\boldsymbol{\xi}$ from the solid into the fluid narrow band. Previous approaches to do this have employed PDE-based methods by defining a normal vector $\mathbf{n} = \nabla\phi$ and extrapolating outwards from the object in the direction of $\mathbf{n}$ (Aslam 2004; Rycroft & Gibou 2012). While these methods are well suited to mathematical analysis, they require considerable bookkeeping for performing the finite difference calculations of ϕ and $\boldsymbol{\xi}$ due to the fields only existing at certain grid locations. In previous work we have found this to be a source of difficulty (Valkov *et al.* 2015).

In the current work, we make use of the following alternative extrapolation procedure. Consider the points in increasing order of ϕ value. For a particular point (i, j) at physical location $\mathbf{x}_{i,j}$:

(i) Set $r = 3$.

(ii) Use least-squares regression to fit a linear map $\boldsymbol{\xi}_{\text{lm}}(\mathbf{x}) = \mathbf{A}x + \mathbf{B}y + \mathbf{C}$ using all available reference map values at (i', j') such that $|i - i'| \leqslant r, |j - j'| \leqslant r$. Weight each value in the regression according to $\phi_{i,j} - \phi_{i',j'}$.

(iii) If the linear map is degenerate then increment r and return to Step 2. Otherwise, continue.

(iv) Set $\boldsymbol{\xi}_{i,j} = \boldsymbol{\xi}_{\text{lm}}(\mathbf{x}_{i,j})$.

This procedure is simpler than the PDE-based methods since it does not require extensive bookkeeping. Since the method uses all available values in a neighborhood, this repeated averaging results in substantial blurring if the extrapolation is continued far away from the interface. However, here, only values near the interface are required, and the averaging is beneficial, serving to damp out high-frequency modes that could be the source of instability. In Step 3, degeneracies occur only when the available points are colinear, in which case there is insufficient information to determine the linear map. In this case, Step 4 causes the algorithm to retry using more neighboring points.

The approach described here makes it possible to simulate objects with sharp corners. The reference map is smoothly defined within the object, and using the above procedure allows it to be smoothly extended into the fluid domain from which solid stresses in the blur zone can be computed. The function $\phi_0(\boldsymbol{\xi})$ that defines the object boundary need not be smooth itself, and can describe a shape with corners.†

3.3. *Computation of stress*

In order to evaluate the stress divergence terms that appear in Eqs. (3.2) & (3.7), the stresses are first computed on the edges of each grid cell. The stress term in Eq. (3.7) is

---

† The IPS method of Dunne (2006) has a similar capability to handle corners, although it is based on a harmonic continuation of the solid velocity, which is used to update the solid displacement field $\mathbf{u}$, from which the boundary can be located by examining $\mathbf{u} - \mathbf{x}$.

computed as

$$\nabla\cdot[\boldsymbol{\tau}(\boldsymbol{\xi}^n)] = \frac{[\boldsymbol{\tau}_x]^n_{i+1/2,j} - [\boldsymbol{\tau}_x]^n_{i-1/2,j}}{h_x} + \frac{[\boldsymbol{\tau}_y]^n_{i,j+1/2} - [\boldsymbol{\tau}_y]^n_{i,j-1/2}}{h_y} \tag{3.13}$$

where $\boldsymbol{\tau}_x = (\tau_{xx}, \tau_{xy})$ and $\boldsymbol{\tau}_y = (\tau_{xy}, \tau_{yy})$ are the components acting on the vertical and horizontal edges, respectively.

3.3.1. *Solid stress*

To begin, the components of the Jacobian are computed using the second-order finite-difference formulae

$$\left(\frac{\partial\boldsymbol{\xi}}{\partial x}\right)_{i-1/2,j} = \frac{\boldsymbol{\xi}_{i,j} - \boldsymbol{\xi}_{i-1,j}}{h_x}, \qquad \left(\frac{\partial\boldsymbol{\xi}}{\partial y}\right)_{i-1/2,j} = \frac{\boldsymbol{\xi}_{i,j+1} + \boldsymbol{\xi}_{i-1,j+1} - \boldsymbol{\xi}_{i,j-1} - \boldsymbol{\xi}_{i-1,j-1}}{4h_y} \tag{3.14}$$

after which the deformation gradient is evaluated as

$$\mathbf{F}_{i-1/2,j} = \left(\left(\frac{\partial\boldsymbol{\xi}}{\partial\mathbf{x}}\right)_{i-1/2,j}\right)^{-1}. \tag{3.15}$$

From here, any constitutive law $\boldsymbol{\tau}_s = \mathbf{f}(\mathbf{F})$ could be used to evaluate the deviatoric stress, $\boldsymbol{\tau}_s$. In the current work, we employ the plane-strain incompressible neo-Hookean law,

$$\boldsymbol{\tau}_s = \mathbf{f}(\mathbf{F}) = G\left(\mathbf{F}\mathbf{F}^{\mathsf{T}} - \tfrac{1}{3}\mathbf{1}(\operatorname{tr}\mathbf{F}\mathbf{F}^{\mathsf{T}} + 1)\right), \tag{3.16}$$

where G is the small-strain shear modulus.

3.3.2. *Fluid stress*

To evaluate the fluid stress, the gradients of the velocity on vertical edges are computed as

$$\left(\frac{\partial\mathbf{v}}{\partial x}\right)_{i-1/2,j} = \frac{\mathbf{v}_{i,j} - \mathbf{v}_{i-1,j}}{h_x}, \tag{3.17}$$

$$\left(\frac{\partial\mathbf{v}}{\partial y}\right)_{i-1/2,j} = \frac{\mathbf{v}_{i,j+1} + \mathbf{v}_{i-1,j+1} - \mathbf{v}_{i,j-1} - \mathbf{v}_{i-1,j-1}}{4h_y}. \tag{3.18}$$

Equivalent stencils are used to compute velocity gradients on horizontal edges, after which the fluid stress is given by

$$\boldsymbol{\tau}_f = \mu_f(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathsf{T}}) \tag{3.19}$$

where μ_f is the viscosity. Equation (3.19) is our standard approach for computing the fluid stress. However, we have also investigated a simplified calculation. Since $\nabla\cdot\mathbf{v} = 0$, it follows that in the bulk of the fluid, the second term in Eq. (3.19) has zero contribution to $\nabla\cdot\boldsymbol{\tau}_f$. Hence an alternative formula is

$$\boldsymbol{\tau}_f^{(\mathrm{simp})} = \mu_f\nabla\mathbf{v}. \tag{3.20}$$

This formula only requires evaluating the simpler stencil in Eq. (3.17). However, Eq. (3.20) is not strictly valid in the blur zone since taking the divergence in Eq. (2.8) results in a non-zero contribution from the second term of Eq. (3.19).

Once all edge stresses are computed, the divergence is computed using

$$[\nabla\cdot\boldsymbol{\tau}]_{i,j} = \frac{\boldsymbol{\tau}_{i+1/2,j} - \boldsymbol{\tau}_{i-1/2,j}}{h_x} + \frac{\boldsymbol{\tau}_{i,j+1/2} - \boldsymbol{\tau}_{i,j-1/2}}{h_y}. \tag{3.21}$$

3.4. *Finite-element projection*

To solve the Poisson problem in Eq. (3.3), we make use of a finite-element formulation. The pressure is comprised of piecewise bilinear elements, and the velocity and density are piecewise constant on the grid cells. For a given pressure element ψ the weak formulation of Eq. (3.3) is

$$-\int_{\Omega} \mathbf{v}^* \cdot \nabla\psi \, dx \, dy + \int_{\Omega} \frac{\Delta t}{\rho(\phi^{n+1/2})} \nabla p^{n+1} \cdot \nabla\psi \, dx \, dy = -\int_{\Gamma_1} \psi \mathbf{v}^{\mathrm{BC}} \cdot \mathbf{n} \, dS \tag{3.22}$$

where Γ_1 is the section of the boundary where inflow and outflow conditions are prescribed. Consider a particular bilinear element function ψ located at a pressure point $p_{i,j}$ in the bulk of the domain. The first term of Eq. (3.22) is

$$\frac{h_y(u^*_{i,j} + u^*_{i,j-1} - u^*_{i-1,j} - u^*_{i-1,j-1})}{2} + \frac{h_x(v^*_{i,j} - v^*_{i,j-1} + v^*_{i-1,j} - v^*_{i-1,j-1})}{2}. \tag{3.23}$$

For the constant-density case when $\Delta t/\rho$ can be taken out as a prefactor, the second term of Eq. (3.22) is

$$\lambda_a p_{i,j} + \lambda_b (p_{i-1,j} + p_{i+1,j}) + \lambda_c (p_{i,j-1} + p_{i,j+1}) + \lambda_d \sum_{\substack{k=\pm 1 \\ l=\pm 1}} p_{i+k,j+l}, \tag{3.24}$$

where

$$\lambda_a = \frac{4(h_x^2 + h_y^2)}{3h_x h_y}, \quad \lambda_b = \frac{-2h_y^2 + h_x^2}{3h_x h_y}, \quad \lambda_c = \frac{-2h_x^2 + h_y^2}{3h_x h_y}, \quad \lambda_d = \frac{-h_x^2 - h_y^2}{6h_x h_y}. \tag{3.25}$$

The expression in Eq. (3.24) can be generalized for the case of non-constant density. Dirichlet conditions on pressure may also be imposed as essential boundary conditions (Johnson 2009). The resultant linear system is solved using the same multigrid library introduced in Sec. 3.1.2 using an error tolerance of $T_{\mathrm{FEM}} = \lambda_a 10^4 \epsilon_m$. In cases where no Dirichlet conditions are used, the pressure field is projected at each step so that it has zero mean.

3.5. *Parameter choices and stability*

Our test cases involve four physical parameters: solid shear modulus G, solid density ρ_s, fluid viscosity μ_f, and fluid density ρ_f. In the solid, the shear wave speed is $c_s = \sqrt{G/\rho_s}$. The CFL condition requires that the simulation timestep be less than or equal to

$$\Delta t_{\mathrm{I}} = \frac{1}{c_s} \min\{h_x, h_y\} = \sqrt{\frac{\rho_s}{G}} \min\{h_x, h_y\}. \tag{3.26}$$

In addition, performing a von Neumann stability analysis shows that the timestep must be less than or equal to

$$\Delta t_{\mathrm{II}} = \frac{\rho_f}{2\mu_f(h_x^{-2} + h_y^{-2})} \tag{3.27}$$

in order to resolve the viscous fluid stress. Inside the solid, we find that simulating stress using only Eq. (3.16) results in an instability—this should be expected since we are effectively solving a hyperbolic system using centered finite differences. To rectify this, we incorporate an extra small artificial viscous stress inside the solid. Based on dimensional considerations, the artificial viscosity should satisfy

$$\mu_e = \kappa_e \rho_s c_s \max\{h_x, h_y\} \tag{3.28}$$

where κ_e is a dimensionless constant. In addition, we also find that artificial viscosity is useful in the fluid–solid transition region. We therefore define the extra viscous stress as

$$\boldsymbol{\tau}_e(\mathbf{x}) = \mu_e(1 - H(\phi(\mathbf{x})))(1 + q\epsilon H'_\epsilon(\phi))\nabla\mathbf{v} \tag{3.29}$$

where q is a dimensionless constant. Based on a variety of tests, we use $q = 1$ and $\kappa_e = 0.4$ throughout the paper. Since the purpose of this extra stress is to stabilize the numerical system, we employ the simpler form of fluid stress given in Eq. 3.20. Since μ_e scales linearly with grid spacing, and the simpler fluid stress only introduces a discrepancy in the blur zone, any errors that are introduced will reduce to zero as the grid is refined. The corresponding timestep restriction is

$$\Delta t_{\mathrm{III}} = \frac{\rho_s}{2\mu_e(1+q)(h_x^{-2} + h_y^{-2})}. \tag{3.30}$$

With these definitions in place, the simulation timestep Δt is chosen to be smaller than the minimum of the three conditions in Eqs. (3.26), (3.27), & (3.30), so that

$$\Delta t = \min\{\alpha_{\mathrm{pad}}\Delta t_{\mathrm{I}}, \alpha_{\mathrm{pad}}\Delta t_{\mathrm{II}}, \beta_{\mathrm{pad}}\Delta t_{\mathrm{III}}\}. \tag{3.31}$$

Here, α_{pad} and β_{pad} are padding factors that are smaller than one. For this paper we use $\alpha_{\mathrm{pad}} = 0.4$ and $\beta_{\mathrm{pad}} = 0.8$, so that the restrictions arising from the two physical stresses (I & II) are applied more stringently than the one for the artificial stress (III). Note that in the limit as $h_x, h_y \to 0$, the artificial viscosity vanishes.

4. Results

Since our purpose is to demonstrate the numerical method as opposed to apply it to a specific problem, we make use of non-dimensionalized quantities for all of the results that we present. To connect the results to reality, the simulation parameters and output can be multiplied by appropriate length, time, and mass scales. Our results also focus on the case of equal grid spacing, $h_x = h_y = h$.

4.1. *A spinning flexible rotor*

The first example demonstrates our method's ability to handle sharp solid corners within a non-trivial FSI setting. It consists of a spinning flexible regular seven-pointed star that is centered on the origin and has vertices at $(L\cos\frac{2\pi k}{7}, L\sin\frac{2\pi k}{7})$ for $k \in \mathbb{Z}$, with length scale $L = 0.62$, density $\rho_s = 3$, and shear modulus $G = 24$. The resolution is 800×800, the simulation domain is $[-1, 1)^2$ and periodic boundary conditions are used. The fluid and rotor are initially stationary. The region $r = |\mathbf{x}| < 0.16$ is used as a pivot. To excite the fluid, the pivot is rotated with an oscillatory motion with angle $\theta(t) = \pi(1 - \cos t)$. This is done by applying an external tether force to the pivot region of

$$\mathbf{f}_{\mathrm{teth}}(\mathbf{x}) = -K_{\mathrm{teth}}H_\epsilon(r_{\mathrm{teth}} - r)(\mathbf{x} - R_{\theta(t)}\boldsymbol{\xi}(\mathbf{x})) \tag{4.1}$$

where $r_{\mathrm{teth}} = 0.16$ and $R_{\theta(t)}$ is a rotation matrix with angle $\theta(t)$. The spring constant is set to $K_{\mathrm{teth}} = 10^{-2}\rho_s\Delta t^2$, which ensures that the natural frequency of the tether satisfies the timestep restriction imposed by the method. The fluid has density $\rho_f = 1$ and viscosity $\mu_f = 10^{-3}$. The RMS angular velocity of the rotor is $\omega_{\mathrm{RMS}} = \pi/2$, and hence a characteristic tip velocity is $\omega_{\mathrm{RMS}}L$. Hence, we define the Reynolds number for this simulation as

$$Re = \frac{\rho_f L(\omega_{\mathrm{RMS}}L)}{\mu_f} \approx 600. \tag{4.2}$$

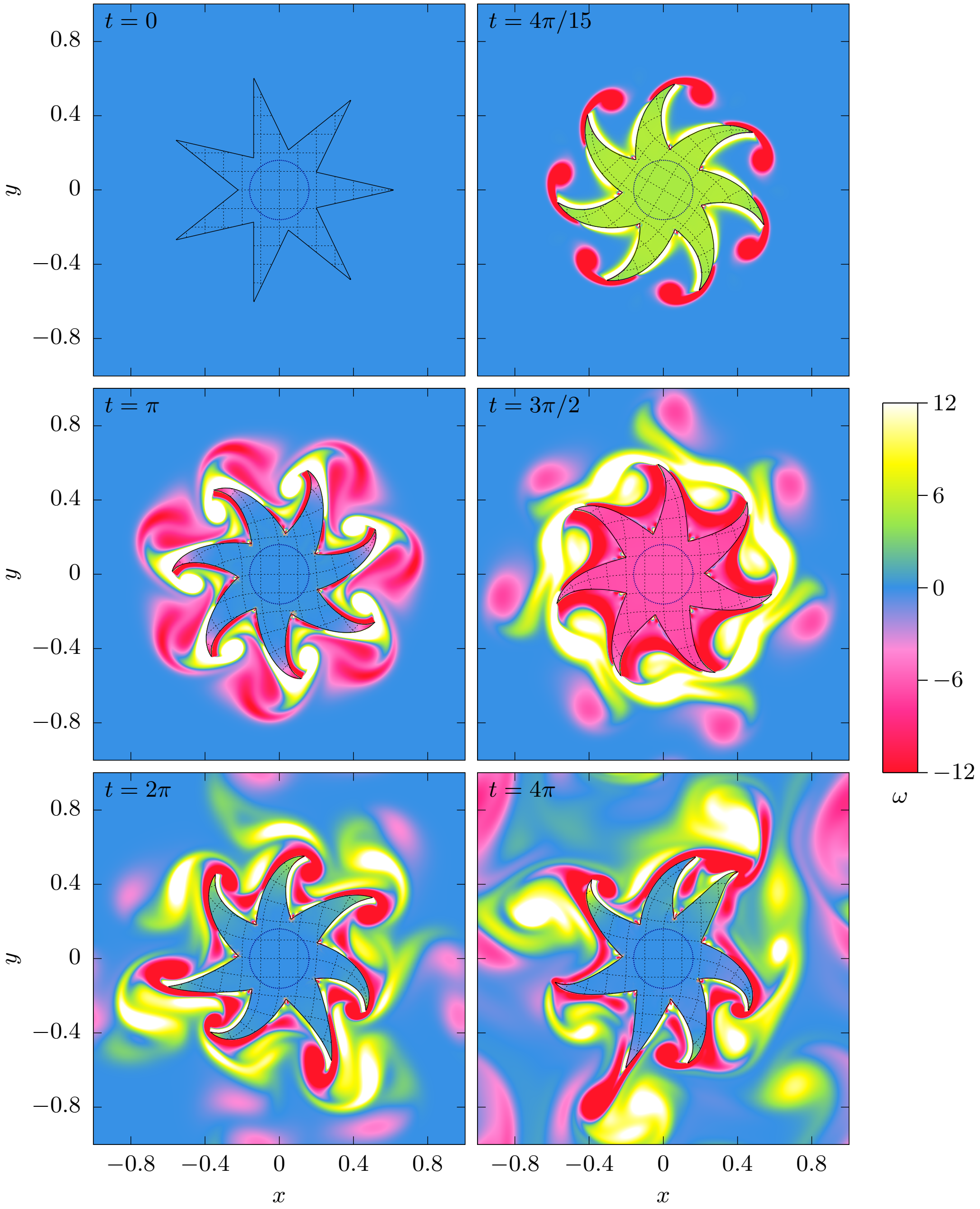

FIGURE 3. Snapshots of vorticity ω in a simulation of a flexible seven-pointed rotor being spun with an oscillatory motion in a fluid. The thick black line marks the fluid–structure interface. The thin dashed lines are contours of the components of the reference map and indicate how the rotor has deformed. The dark blue dotted circle shows the pivot region. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G) = (1, 10^{-3}, 3, 24)$.

The simulation was run from $t = 0$ to $t = 4\pi$ using sixteen threads on a Linux computer with dual 10-core 2.2 GHz Intel Xeon E5-2630 processors. For the given parameters, the timestep of $\Delta t = 1.105 \times 10^{-4}$ was determined by the extra viscous stress in the solid. Simulation output was saved at regular intervals of $\pi/150$. The total wall clock time for the simulation was 6.53 h. A total of 114,000 timesteps were performed, with each taking 206 ms to compute. A substantial fraction of the computation time is spent performing the two linear solves. The MAC projections take on average 13.75 V-cycles and require 43.1 ms per timestep. The finite-element projections take on average 11.91 V-cycles and require 48.1 ms per timestep.

Snapshots of vorticity $\omega = \partial_x v - \partial_y u$ in the simulation are shown in Fig. 3. The vorticity is computed on each grid cell corner, using centered finite differences of the velocities in the four adjoining grid cells. As the star begins to rotate, each point deforms clockwise, and vortices are shed from the points, which are visible at $t = 4\pi/15$. By $t = \pi$, the rotor is stationary, and the points are now deformed anti-clockwise due to the angular deceleration. As time progresses, the disturbance to the fluid becomes larger. By $t = 2\pi$, the rotational symmetry of the fluid flow is lost, due to interactions across the periodic boundaries, which break the seven-fold symmetry. By $t = 4\pi$, after two complete cycles of the oscillatory motion, there are vortices present throughout the fluid. Supplemental Movie 1 shows the complete simulation. To visualize the fluid motion, the movie also shows a number of tracers with trajectories $\mathbf{x}(t)$. The tracers are initialized at random positions in the fluid and are updated using the ordinary differential equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}_{\mathrm{bic}}(\mathbf{x}(t)), \tag{4.3}$$

where $\mathbf{v}_{\mathrm{bic}}$ is the bicubic interpolation of the velocity field, and the time integration is performed using the second-order improved Euler method (Süli & Mayers 2003).

4.2. *Flag flapping*

Besides numerical convergence, as a test of the robustness of our approach and its accuracy across Reynolds numbers, we consider the example of flag flapping, a problem that has been studied extensively from experimental, numerical, and analytical perspectives (Zhang *et al.* 2000; Watanabe *et al.* 2002; Zhu & Peskin 2002; Connell & Yue 2007). Following the problem description and notation of Connell & Yue (2007),† we introduce a thin filament of length L, thickness $h \ll L$, density ρ_s, and Young's Modulus E, clamped at its left endpoint and submerged in fluid of kinematic viscosity ν_f and density ρ_f, flowing rightward with speed V at infinity. Three dimensionless numbers can be introduced to study the dynamical behavior of the filament: the mass ratio $\mu = \rho_s h/\rho_f L$, Reynolds number $Re = VL/\nu_f$, and nondimensional bending rigidity $K_B = EI/(\rho_f V^2 L^3)$. Unlike the previous numerical approaches that consider the filament to be a one-dimensional beam, our method uses a true continuum solid formulation so we can consider cases beyond the thin filament limit, such as a thick flag for which the parameter h does not necessarily satisfy $h \ll L$.

We first seek to determine if our method correctly captures the transition of the filament dynamics from stable to flapping in the limit of a thin filament. We consider a filament with $L = 1$, $h = 0.05$, $K_B = 0.001$, in a fluid of density $\rho_f = 1$. To set K_B, we use the fact that in the linear elastic regime $E = 3G$, and the moment of inertia is $I = h^3/12$. We vary ρ_s and ν_f in order to test a range of μ and Re. The simulation domain is set to be a $[-1, 5] \times [-1, 1)$ rectangle with assigned rightward velocity of speed $V = 1$ on the left boundary, vanishing pressure on the right boundary, and periodic boundary conditions on the top and bottom boundaries. We use a 1824×608 grid to represent the domain. The

---

† For consistency with Connell & Yue (2007) we fully adhere to their notation. However we draw attention to the reader that h (filament thickness) has a different meaning than h (grid spacing) used in all other sections. Furthermore, μ (mass ratio) is distinct from μ_f (dynamic viscosity).

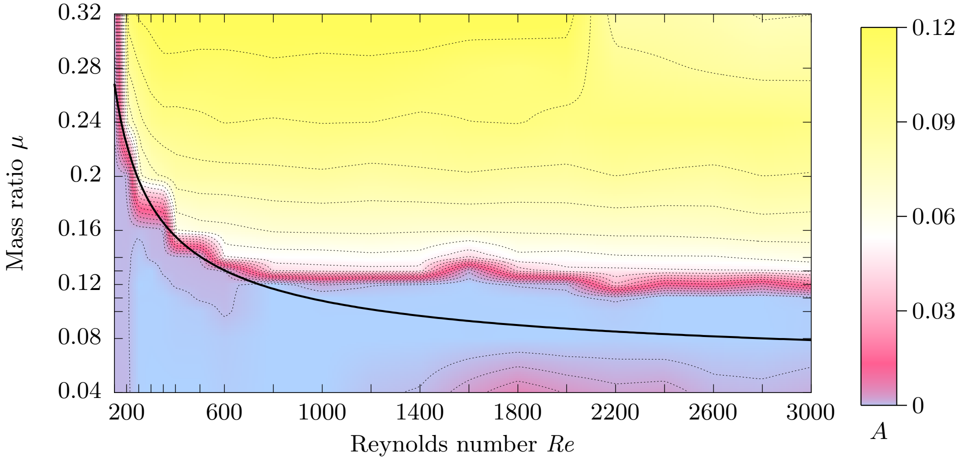

FIGURE 4. Plot showing the steady-state oscillation amplitude A of a thin flag with aspect ratio 20 and bending rigidity $K_B = 0.001$, as a function of the Reynolds number Re and mass ratio μ. The colors shown are based on a bilinear interpolation of a two-dimensional grid of simulations. The axis ticks show the sampled values of Re and μ, with more simulations being performed in parameter ranges of interest. The thin dotted lines are contours at spacings of $({}^{n}/_{50})^2$ for $n \in \mathbb{N}$. The thick solid line is the stable-to-flapping transition formula, Eq. (4.6) of Connell & Yue (2007).

filament is modeled as a rectangle $0 < x < 1$, $-{}^{h}/_{2} < y < {}^{h}/_{2}$ with semicircular end caps. The filament is anchored at $(0, 0)$ using the tethering methodology described in Sec. 4.1, with $\theta(t) = 0$ and $r_{\text{teth}} = h/4$ in this case. We track the filament tip by introducing a tracer $\mathbf{x}(t)$ that starts from $(1, 0)$ and is integrated according to Eq. (4.3). To prevent integration errors building up over time, the position of the tracer is periodically reset to satisfy $\boldsymbol{\xi}_{\text{bic}}(\mathbf{x}(t)) = (1, 0)$ using a Newton–Raphson root-finding method, where $\boldsymbol{\xi}_{\text{bic}}$ is the bicubic interpolant of the reference map field. The results of this section are based upon 556 simulations with different parameters that were run on a variety of Linux and Apple servers at Harvard University and the Lawrence Berkeley National Laboratory. Depending on parameters and computer speed, each simulation took approximately 3–12 days using 4–6 threads. Simulations with smaller Re generally take longer, since resolving the fluid viscosity requires a smaller timestep.

To systematically evaluate the behavior of the filament, we store the perpendicular tip deflection $y(t)$ over the interval $t \in [120, 160]$. Since the typical filament oscillation period is approximately 1.7, the simulations correspond to almost one hundred complete oscillations, and hence the interval $[120, 160]$ is sufficient for the oscillation amplitude to reach a steady state. The normalized Fourier transform is given by

$$\tilde{y}(k) = \frac{1}{40} \int_{120}^{160} e^{ikt} y(t) dt. \tag{4.4}$$

The maximum Fourier amplitude is given by

$$A = \max_{k \in [0.1, 50]} |\tilde{y}(k)|. \tag{4.5}$$

Typically the oscillation frequency is $2\pi/1.7 \approx 4$, and the range of k in Eq. (4.5) is chosen to cover broadly the possible values. If $A \approx 0$ the filament is in the stable (no-flapping) regime and otherwise the filament is flapping, with A serving as a scalar measure of the amplitude of the dominant flapping mode. Since our initial conditions are symmetric, the

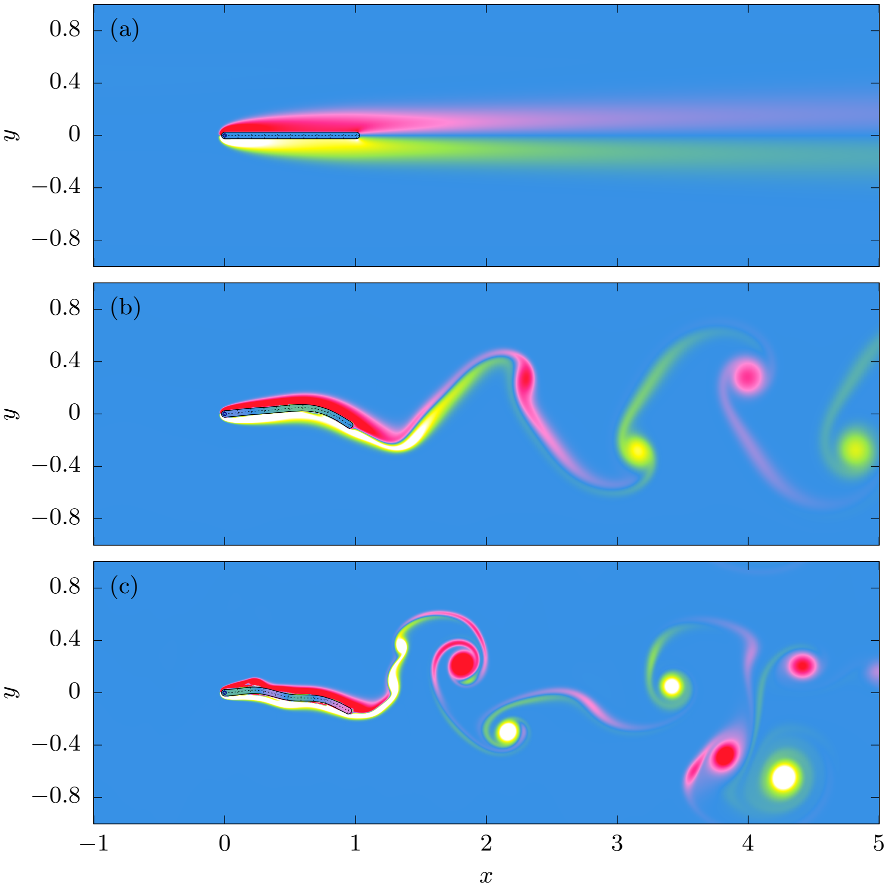

FIGURE 5. Simulations of a thin flexible flag anchored at $(0, 0)$ in a fluid with mean velocity $\mathbf{v} = (1, 0)$, at $t = 160$. The flag has an aspect ratio of 20. Three simulations with different parameters are shown: (a) stable with $(\mu, K_B, Re) = (0.04, 0.001, 400)$, (b) limit-cycle flapping with $(\mu, K_B, Re) = (0.16, 0.001, 1400)$, and (c) chaotic flapping with $(\mu, K_B, Re) = (0.32, 0.001, 3000)$. The thick black lines mark the fluid–structure interfaces. The thin dashed lines are contours of the components of the reference map and indicate how the flags deform. The small dark blue circles show the anchored regions. The colors show vorticity, using the same scale as Fig. 3.

breakage of symmetry occurs due to numerical noise introduced by the multigrid solver, on the scale of the parameters T_{MAC} and T_{FEM} introduced previously. We also investigated explicitly breaking symmetry by applying an initial perturbation to the perpendicular velocity in the filament tip, but the calculations of A were insensitive to this. Connell & Yue (2007) proposed an analytical formula for the stable-to-flapping transition line:

$$\mu = \frac{1.3Re^{-1/2} + K_B 4\pi^2}{1 - 0.65Re^{-1/2} 2\pi - 0.5K_B 8\pi^3}. \tag{4.6}$$

Connell & Yue (2007) validated this equation numerically using a direct fluid–filament coupling procedure, a procedure that itself was validated against experiments (Zhang *et al.* 2000; Watanabe *et al.* 2002). The above formula is obtained without consideration

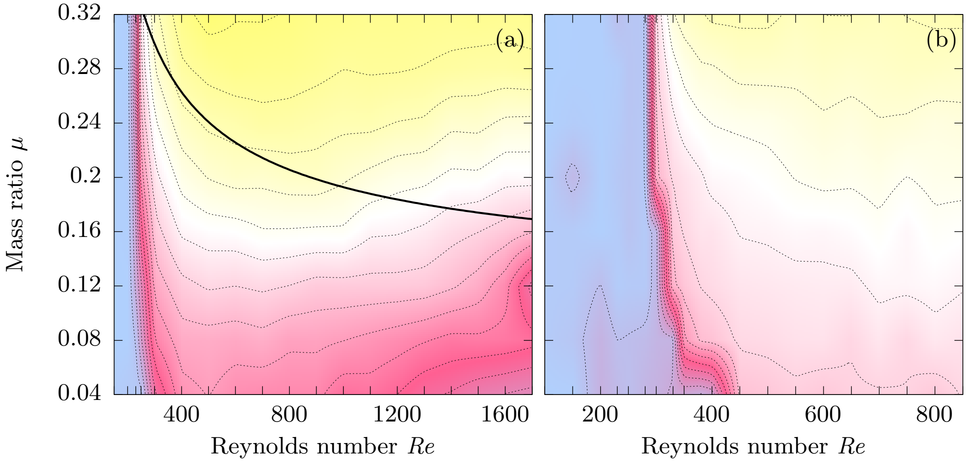

FIGURE 6. Plot showing the steady-state oscillation amplitude A as a function of the Reynolds number Re and mass ratio μ, for (a) flags with $K_B = 0.002$ and aspect ratio 10, and (b) flags with $K_B = 0.004$ and aspect ratio 5. The colors shown are based on a bilinear interpolation of a two-dimensional grid of simulations, using the same scale as Fig. 4. The axis ticks show the sampled values of Re and μ, with more simulations being performed in parameter ranges of interest. The thin dotted lines are contour at spacings of $(n/50)^2$ for $n \in \mathbb{N}$. The thick solid line in (a) is the stable-to-flapping transition formula for thin flags, Eq. (4.6) of Connell & Yue (2007). For (b), the formula is out of range and the entire parameter space is in the stable region.

of certain effects, such as possible variations of tension along the flag and the presence of global lift forces attempting to realign the flag with the flow, although methods to include these phenomena exist in certain limits (Argentina & Mahadevan 2005). In Fig. 4 we show the behavior of A from our numerical simulations together with the analytical phase boundary above. For Reynolds numbers below 1000, there is very good agreement between the locus where A goes non-zero and the analytical curve. When $Re \geqslant 1000$ the transition predicted by the simulation happens at a slightly higher μ than predicted by Eq. (4.6). The most likely explanation for this is that numerical diffusion from the fluid advection effectively increases the fluid viscosity. However, other factors such as the finite domain size, the extensibility of the filament, and the non-zero h may also play a role.

The behavioral switch from stable to flapping is also quite evident in the long-time flow fields, shown in Fig. 5. Small values of μ and K_B result in stable behavior, characterized by a straight filament and fluid flow that is symmetric about the filament axis (Fig. 5(a)). Upon crossing the transition, periodic undulatory filament motions develop with a fluid vortex street shed from the filament (Fig. 5(b)). Increasing Re and μ even further reveals a chaotic filamentary motion, which was also observed by Connell & Yue (2007) (Fig. 5(c), Supplemental Movie 2). The chaotic regime coincides with a drop in A shown in the top right of Fig. 4 because the tip deflection no longer has a clear single dominant oscillatory mode. Because the filament is modeled as a thin continuum body of isotropic elastic media as opposed to an inextensible beam, we observe filament extension in the initial moments of the simulation as the imposed fluid flow applies a net rightward traction.

We explore the importance of aspect ratio by introducing $R = h/L$ as an independent dimensionless group. We observe that as one departs from the $R \ll 1$ regime, adherence to Eq. (4.6) is diminished. In Fig. 6 we show results for $R^{-1} = 10$ and 5. In general, thick flags have a smaller stable domain than would be predicted by the thin-filament

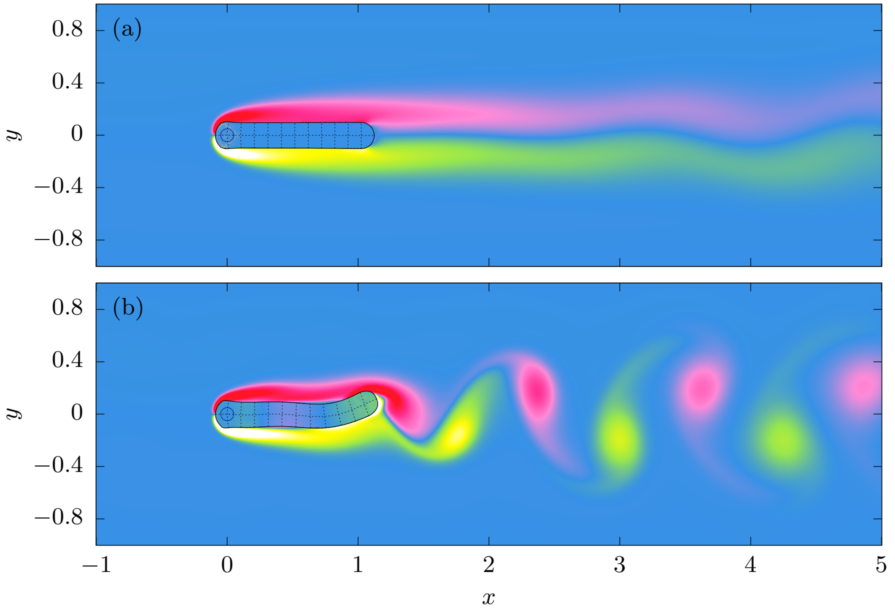

FIGURE 7. Simulations of a thick flexible flag anchored at $(0, 0)$ in a fluid with mean velocity $\mathbf{v} = (1, 0)$, at $t = 160$. The flag has an aspect ratio of 5. Two simulations with different parameters are shown: (a) vortex-shedding with $(\mu, K_B, Re) = (0.04, 0.004, 750)$ and (b) limit-cycle flapping with $(\mu, K_B, Re) = (0.28, 0.004, 750)$. The thick black lines mark the fluid–structure interfaces. The thin dashed lines are contours of the components of the reference map and indicate how the flags deform. The dark blue dotted circles show the anchored regions. The colors show vorticity, using the same scale as Fig. 3.

limit. We can explain this effect at least in part with bluff-body dynamics. When R is non-negligible, the thickness of the flag allows the solid geometry to act as a bluff body over which the fluid is driven to flow. Flow over a fixed cylinder of diameter D undergoes a transition from a laminar flow to a periodic vortex street as DV/ν_f grows beyond ~ 50 (Lienhard 1966). In our case, the flag thickness acts like D, and once a vortex street is induced off the bluff back end of the flag, the oscillatory force it induces necessitates flapping. We reiterate that this physical source of oscillatory forcing emerges only when flags are thick enough to act as a bluff body. Consistent with this expectation, when $Vh/\nu_f = Re \times R > 50$ we see only flapping states for any choice of μ or Re. Figures 7(a) & 7(b) show simulation snapshots of bulky flags with low and high mass ratios, respectively. Simulations of these two cases are shown in Supplemental Movie 3 and Supplemental Movie 4, respectively.

4.3. *Tests for a range of elastic modulus*

To demonstrate that the method works across a wide range of shear moduli, we consider a piston-like geometry where a flexible paddle is pushed through a fluid-filled cavity. The domain is $-1 \leqslant x \leqslant 1$ and $0 \leqslant y \leqslant 5$ and no-slip boundary conditions are used on all sides. The grid size is 160×400, the fluid density is $\rho_f = 1$, the fluid viscosity is $\mu_f = 10^{-3}$, and the solid density is $\rho_s = 2$.

A rectangular flexible paddle of width 1.6 and height 0.4 is initially centered at $(0, 0.4)$.

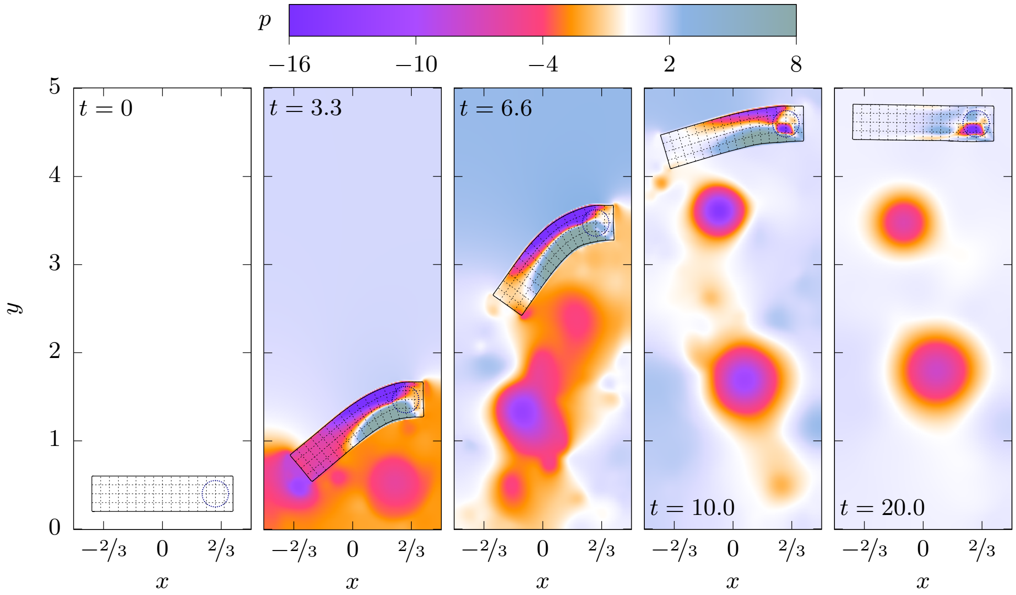

FIGURE 8. Snapshots of pressure p for a simulation where a flexible paddle is pushed through a fluid-filled cavity. The paddle has shear modulus $G = 100$ and is anchored on its right end. The thick black line marks the fluid–structure interface. The thin dashed lines are contours of the components of the reference map and indicate how the paddle deforms. The dark blue dotted circle shows the region of prescribed displacement. Simulation parameters are $(\rho_f, \mu_f, \rho_s) = (1, 10^{-3}, 2)$.

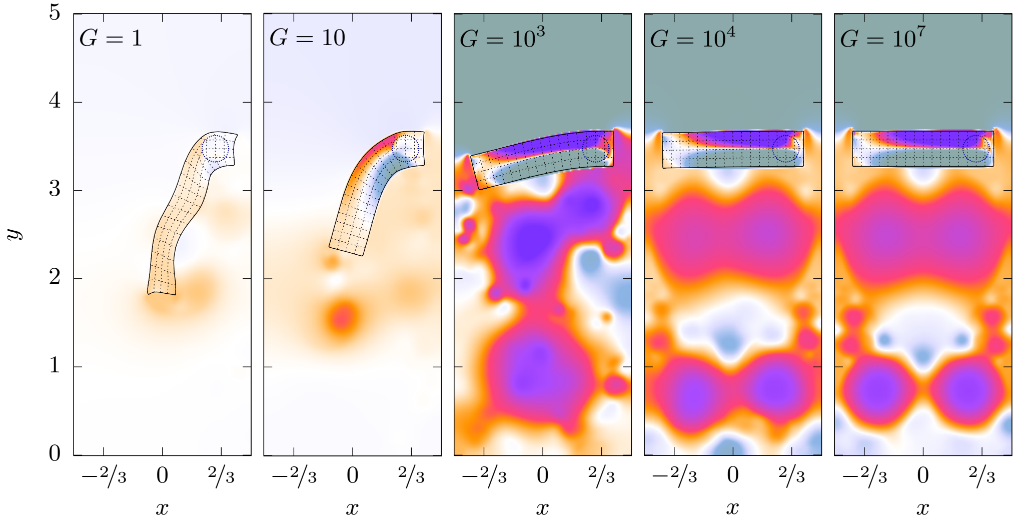

FIGURE 9. Snapshots of pressure p at time $t = 6.6$, in a sequence of simulations where flexible paddles with different shear moduli G are pushed through a fluid-filled cavity. The paddles are anchored on its right end. The thick black lines mark the fluid–structure interfaces. The thin dashed lines are contours of the components of the reference map and indicate how the paddles deform. The dark blue dotted circles show the regions of prescribed displacement. The colors show pressure using the same scale as Fig. 8. Simulation parameters are $(\rho_f, \mu_f, \rho_s) = (1, 10^{-3}, 2)$.

The displacement of the paddle is prescribed in a circular region of radius 0.15 centered on $(0.6, y_p(t))$, using the same tethering procedure as in Eq. (4.1). The simulation is run for a duration of $T = 20$, and the circular region moves vertically according to $y_p(t) = 0.4 + 4.2Y(t/T)$ where

$$Y(\tau) = \begin{cases} 4\tau^2(3-4\tau) & \text{for } t < 1/2, \\ 1 & \text{for } t \geqslant 1/2. \end{cases} \tag{4.7}$$

Hence, for $0 \leqslant t < 10$, the paddle is dragged through the fluid, and for $10 \leqslant t \leqslant 20$ the circular region is stationary and the paddle equilibrates.

Simulations are run using shear moduli from $G = 1$ to $G = 10^7$, following the standard choices for timestep and extra solid viscosity described in the main text. Figure 8 shows a sequence of snapshots of pressure in the simulation with $G = 100$. As the paddle is pushed through the cavity, it is bent downward due to the pressure of the fluid. Vortices are shed from the paddle tip, creating regions of low pressure visible at $t = 10$ and $t = 20$. Figure 9 shows a sequence of snapshots at $t = 6.6$ for the full range of shear moduli. For $G = 1$ and $G = 10$ the paddle deformed so strongly that there is little pressure build-up in the upper part of the domain. However for $G \geqslant 10^3$ the pressure build-up is large, and fluid must push through the thin gaps on either side of the paddle. For $G = 10^7$, the paddle becomes near-rigid, so that the fluid flow becomes almost symmetric even though the paddle's motion is only prescribed on the right side. In these simulations the timestep is set by the limit from the extra solid viscous term (Eq. (3.30)). The total number of timesteps scales according to $\sqrt{G}$ and thus the simulation for $G = 10^7$ takes approximately 3,100 times more computational resources than that for $G = 1$.

4.4. *Solid actuation*

The method also admits a simple approach for simulating actuated solids. This feature allows one to assign time-dependent internal deformations to subregions of a solid, which is useful for modeling active media such as swimmers. Unlike the tethering approach used in Section 4.1, which assigns the full motion of a region by adding an external body force in that region, here what is done is to add extra internal stresses to achieve a desired shape change in a subdomain, without adding net external force. To actuate a particular (Lagrangian) solid region, B_a, one writes the desired actuated deformation gradient $\mathbf{F}_a(\mathbf{X} \in B_a, t)$, which can then be equivalently expressed in Eulerian frame as $\mathbf{F}_a(\mathbf{X} = \boldsymbol{\xi}_a(\mathbf{x} \in b_a, t), t)$ for b_a the image of B_a in the Eulerian frame. At any point $\mathbf{x} \in b_a$, the constitutive relation is adjusted by replacing all references to $\mathbf{F}(\mathbf{x}, t)$ with $\mathbf{F}(\mathbf{x}, t)\mathbf{F}_a(\mathbf{x}, t)^{-1}$. In an isotropic hyperelastic system, for example, this effectively distorts the region's rest configuration to the distortional state given by $\mathbf{F}_a$. If at any moment in time a configuration of the actuated domain differs from the intended actuated configuration, a stress given by $\mathbf{f}(\mathbf{F}(\mathbf{x}, t)\mathbf{F}_a(\mathbf{x}, t)^{-1})$ emerges that moves the system toward the actuated deformation (where $\mathbf{f}$ is defined from Eq. 3.16). One could in principle assign a stiffer response in the actuated domain if a faster conformation is desired, but we have found it to be sufficient to use the same underlying hyperelastic constitutive model in the actuated and passive subregions of the solid. This approach is similar to the multiplicative Kröner–Lee decomposition used in plasticity (Kröner 1960; Lee 1969), where a tensorial state variable $\mathbf{F}_p$ is introduced and the elastic deformation gradient, which produces the stress, is given by $\mathbf{F}\mathbf{F}_p^{-1}$. But unlike $\mathbf{F}_p$, which evolves under a constitutive flow rule, here we assign $\mathbf{F}_a(\mathbf{x}, t)$ directly.

A similar, albeit reduced-dimensional approach has been used to model muscle contraction in swimming lampreys and other narrow-body swimmers. Tytell *et al.* (2010),

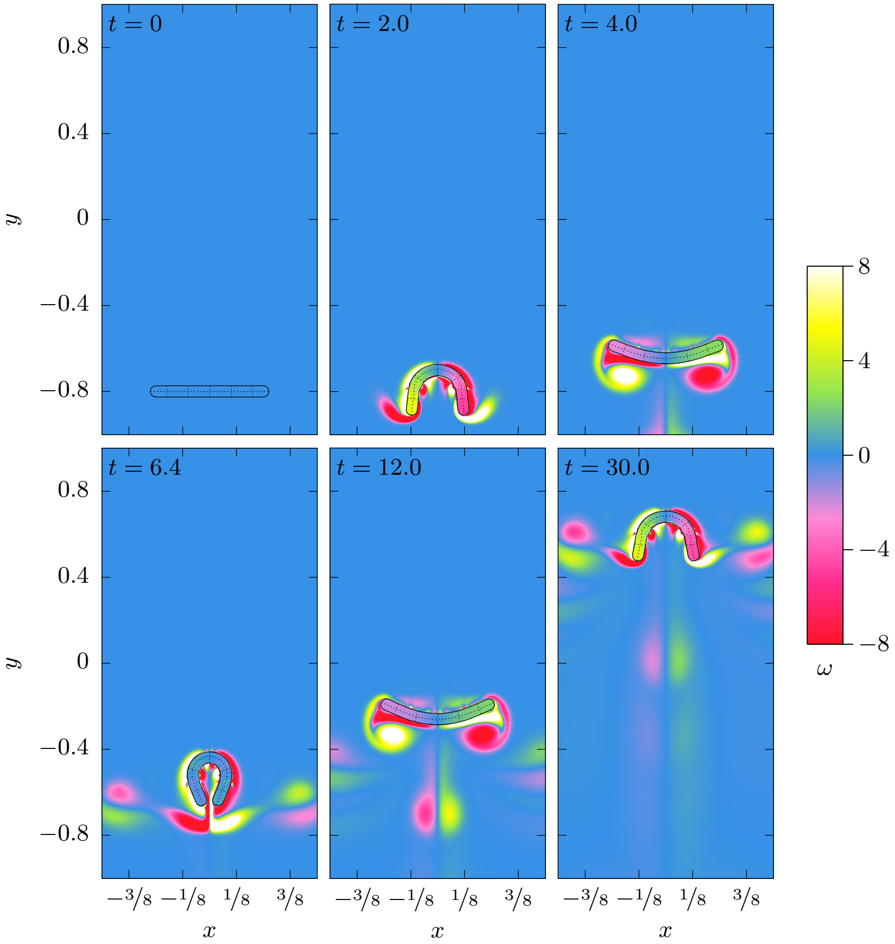

FIGURE 10. Six successive snapshots of the flapping swimmer ($Re \approx 200$), with colors showing vorticity ω. A subregion within the solid body is actuated to bend periodically and the remaining solid is passive. The motion induces the flapping body to swim. The thick black line marks the fluid–structure interface. The thin dashed lines are contours of the components of the reference map and indicate how the swimmer deforms. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G) = (1, 5 \times 10^{-4}, 4, 10)$.

simulated a lamprey swimming in a 2D geometry. The lamprey is modeled with three connected parallel filaments, the outer two of which obey a one-dimensonsal viscoelastic model with an additional user-defined contractile stress (McMillen *et al.* 2008). Actuated bending of the lamprey occurs through asymmetric contraction of the filaments. An alternative approach, as used by Gazzola *et al.* (2015) and Patel *et al.* (2018), involves direct assignation of an external bending moment on the swimmer cross-section, which in a stiff limit equates to directly assigning the swimmer shape through time. The approach we describe above could be seen as a 2D (or potentially 3D) generalization of approaches like these, permitting possibly more through-thickness spatial variation in actuation. The implementation shown here could be made more realistic by including dissipation within the swimmer, neuro-muscular signaling, and contractile-only forcing as was done by Tytell *et al.* (2010), McMillen *et al.* (2008) and Patel *et al.* (2018).

As an example, we consider a flapping swimmer (Fig. 10, Supplemental Movie 5). The swimmer is a rectangle of width $W = 0.5$ and height $H = 0.052$ with circular end caps, initially centered on $(0, -0.8)$, which we choose to be the location of the origin. We choose the actuated domain, B_a, to be a centered subregion within the swimmer, comprising a rectangle of width 0.28 and height 0.042 with circular end caps. The following actuation is applied:

$$\mathbf{F}_a(\mathbf{X}, t) = \begin{pmatrix} e^{-\alpha(\mathbf{X},t)} & 0 \\ 0 & e^{\alpha(\mathbf{X},t)} \end{pmatrix} \tag{4.8}$$

where

$$\alpha(\mathbf{X}, t) = -\lambda X_y H_\epsilon(-d) \sin^8 \omega t = -\lambda \xi_y(\mathbf{x}, t) H_\epsilon(-d) \sin^8 \omega t \tag{4.9}$$

and d is the signed distance from the Eulerian boundary of b_a. By blurring the boundary of the actuated domain under $H_\epsilon(-d)$, it should be noted material positioned up to ϵ away from the true boundary of b_a will receive some actuation stress. The parameters used in the simulation are $\omega = 2\pi/8$, $\epsilon = 2.5h$, and $\lambda = (\log 2.2)/0.021$. Thus the maximum stretch on the top boundary is 2.2. The simulation uses a 1200×1200 grid in $[-1.5, 1.5)^2$ with periodic boundary conditions. The densities, dynamic viscosity, and solid shear modulus are $(\rho_f, \mu_f, \rho_s, G) = (1, 5 \times 10^{-4}, 4, 10)$.

By actuating the flapper in this fashion, the Lagrangian domain B_a, which comprises roughly half the area of the body, is forced to bend periodically in time. The unactuated portion of the swimmer remains passive and flaps as an elastic body in response to be being conjoined to the actuated region. The swimming flapper achieves a Reynolds numbers of $Re = V_{\text{solid}}^{\max} W/\nu_f \approx 200$. Its ability to translate its center of mass by swimming evidences that this example is not near zero Reynolds number; vortex shedding can be seen for each flap.

4.5. *Multi-body contact*

Since the reference map technique does not employ moving meshes, it is particularly well suited to problems involving many objects coming into contact. This capability would be useful for a variety of problems, such as studying colloidal mixtures with soft, deformable particles.

To generalize the method to N objects, we introduce independent reference maps $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \ldots, \boldsymbol{\xi}^{(N)}$ with the "(j)" suffix being used to denote any quantity associated with object j. For the purposes of exposition, we assume each field is defined as a separate globally defined function that is extrapolated separately, although in reality each reference map only need be defined in a local neighborhood of each object. Each reference map is updated using Eq. (3.1). For a given $\boldsymbol{\xi}^{(j)}$, the solid stress $\boldsymbol{\tau}_s^{(j)}$ is computed using the methods of Sec. 3.3.

When two or more objects come together, their blur zones may overlap, and thus it is necessary to generalize the definition of global stress that was given in Eq. (2.8). At a given point, define $\lambda^{(j)} = 1 - H_\epsilon(\phi^{(j)})$ to be the solid fraction of object j. Then the stress is given by

$$\boldsymbol{\tau} = \begin{cases} \boldsymbol{\tau}_f + \sum_i \lambda^{(i)} (\boldsymbol{\tau}_s^{(i)} - \boldsymbol{\tau}_f) & \text{if } \sum_i \lambda^{(i)} \leqslant 1, \\ \dfrac{\sum_i \lambda^{(i)} \boldsymbol{\tau}_s^{(i)}}{\sum_i \lambda^{(i)}} & \text{if } \sum_i \lambda^{(i)} > 1. \end{cases} \tag{4.10}$$

If only one object is present, this definition exactly matches Eq. (2.8). If several objects are present, then they each contribute to the global stress, with the fluid stress filling any unassigned fraction. In rare situations (*e.g.* three objects meeting at a point) the solid fractions may total more than one. In this case, $\boldsymbol{\tau}$ is taken as a weighted average of the solid stresses, and the fluid stress does not contribute at all. The global density field is defined using the same mixing procedure as in Eq. (4.10).

In our tests, we have found that independently updating N reference maps and computing a global stress according to Eq. (4.10) is sufficient to perform multi-body simulations. However, since the simulation employs a single globally-defined velocity field, it becomes problematic when shapes become very close together, since it is hard for them to separate as they move according to the same underlying velocity. Similar behavior has been noted in the literature on the immersed boundary method (Lim & Peskin 2012; Krishnan *et al.* 2017), which also employs a single global velocity field for the movement of structures. To rectify this, we introduce a small contact stress (in addition to the stress of Eq. (4.10)) when the blur zones of two objects overlap, which penalizes the interfaces from becoming too close together. We first define a contact force function of

$$f(x) = \begin{cases} \frac{1}{2}(1-\frac{x}{\epsilon}) & \text{if } x < \epsilon, \\ 0 & \text{if } x \geqslant \epsilon. \end{cases} \tag{4.11}$$

Now, consider the stress calculation at an edge that is within the blur zones of two or more solids. Consider a pair of the solids (i) and (j). Using finite differences, compute a unit normal vector

$$\mathbf{n} = \frac{\nabla(\phi^{(i)} - \phi^{(j)})}{\|\nabla(\phi^{(i)} - \phi^{(j)})\|_2} \tag{4.12}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. The contact stress is defined as

$$\boldsymbol{\tau}_{\text{col}} = -\eta \min\{f(\phi^{(i)}), f(\phi^{(j)})\}(G^{(i)} + G^{(j)}) \left(\mathbf{n}\otimes\mathbf{n} - \tfrac{1}{2}\mathbf{1}\right), \tag{4.13}$$

where $\eta = 4$ is a dimensionless constant, the $G^{(i)}$ are object-dependent shear moduli, and the $\mathbf{1}$ term is included to make the stress trace-free. In the rare case where the edge is within three or more solid blur zones, the calculation is repeated for each pair, and each contribution $\boldsymbol{\tau}_{\text{col}}$ is added to the global stress.

These collision stress terms induce forces that push apart objects when they become close. Formulating the collision interaction as an additional stress is advantageous since it immediately ensures that total momentum of the entire simulation is numerically conserved. The method is not sensitive to the exact functional form of f in Eq. (4.11). An alternative formulation is to directly use the transition function, $f(\alpha) = 1 - H_\epsilon(\alpha)$, but we find that the faster growth of the function in Eq. (4.11) when α becomes smaller than ϵ yields better results in our test cases.

Figure 11 shows snapshots from a multi-body simulation in a non-periodic box $[-1,1]^2$ using a resolution of 1280×1280 with fluid density $\rho_f = 1$ and dynamic viscosity $\mu_f = 10^{-3}$. Forty-two squares with shear modulus $G = 2$ and density $\rho_s = 3$ are inserted at random positions in the box, with side lengths chosen uniformly over the range $[0.1, 0.4]$. Any squares that lie within a distance of 0.1 of another square are rejected, and are chosen again. At $t = 0$, each square is set to initially spin with angular velocity chosen uniformly from the range $[-5, 5]$. A gravitational acceleration of $g = 0.5$ in the negative y direction is applied, so that the squares sediment at the bottom of the box. The full simulation is shown in Supplemental Movie 6.

A benefit of the reference map technique is that it can handle both neutrally buoyant solids, and solids that are lighter than the surrounding fluid, without any modification. To demonstrate this, we consider a second multi-body example with 75 solids made of rectangles of length 0.44, thickness 0.044, and rounded end caps, all of which are initially vertically aligned. The solid densities are randomly chosen uniformly over the range $[0.4, 1.6]$ and a gravitational acceleration of $g = 1$ in the negative y direction is applied. A

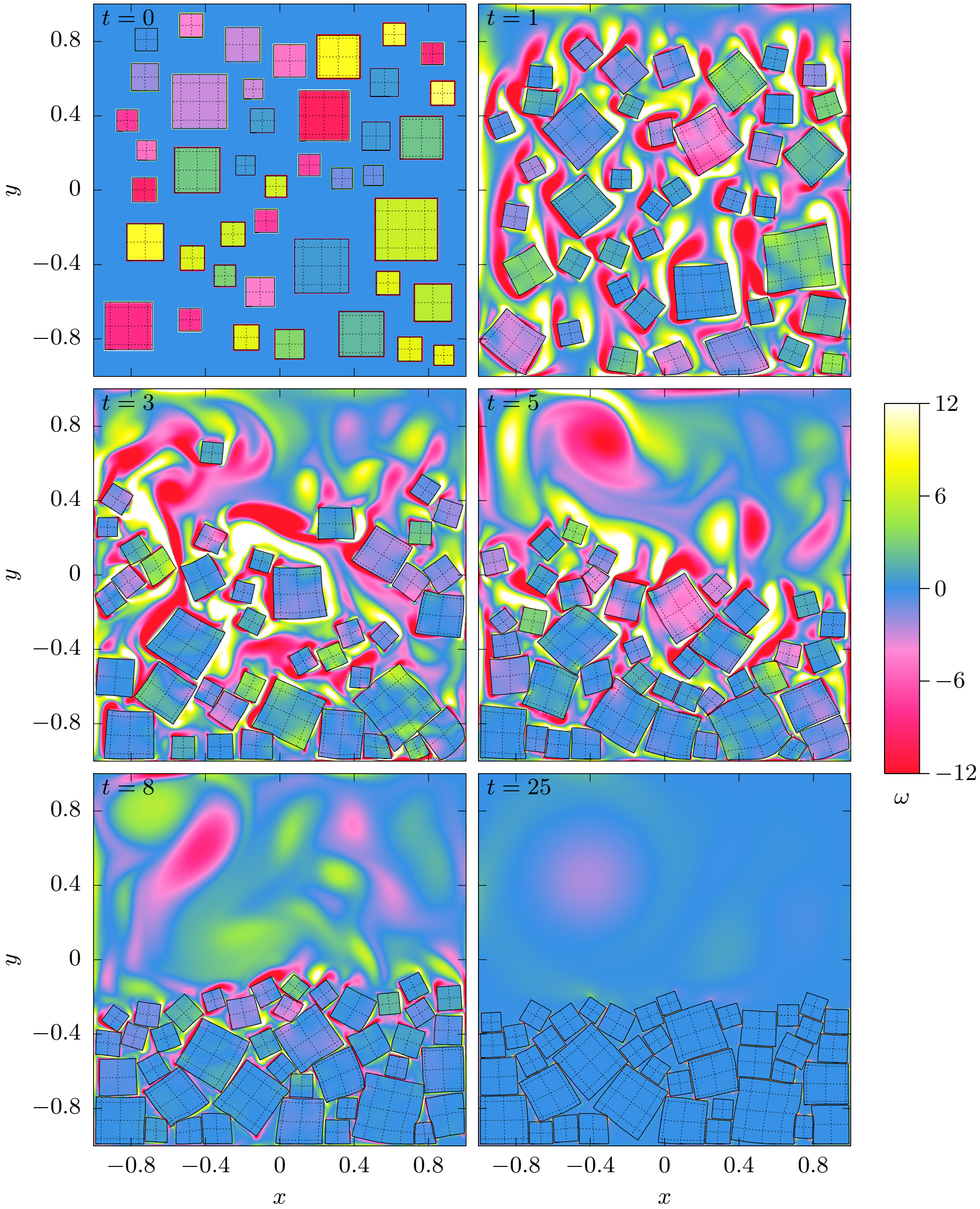

Figure 11. Snapshots of vorticity ω in a simulation of forty-two squares sedimenting in a fluid-filled box. The thick black lines mark the fluid–structure interfaces. The thin dashed lines are contours of the components of the reference map defined in each object and indicate how the squares deform. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G, g) = (1, 10^{-3}, 3, 2, 1.5)$.

simulation grid of 640×1280 is used on the domain $x \in [-0.75, 0.75], y \in [-1.5, 1.5]$, the dynamic viscosity is $\mu_f = 2 \times 10^{-3}$, the shear modulus is $G = 1.5$, and the fluid density is $\rho_f = 1$. Figure 12 snapshots of the density, the deviation of pressure from the background gradient due to gravity on the fluid $p' = p + \rho_f g y$, and the vorticity at five time points.

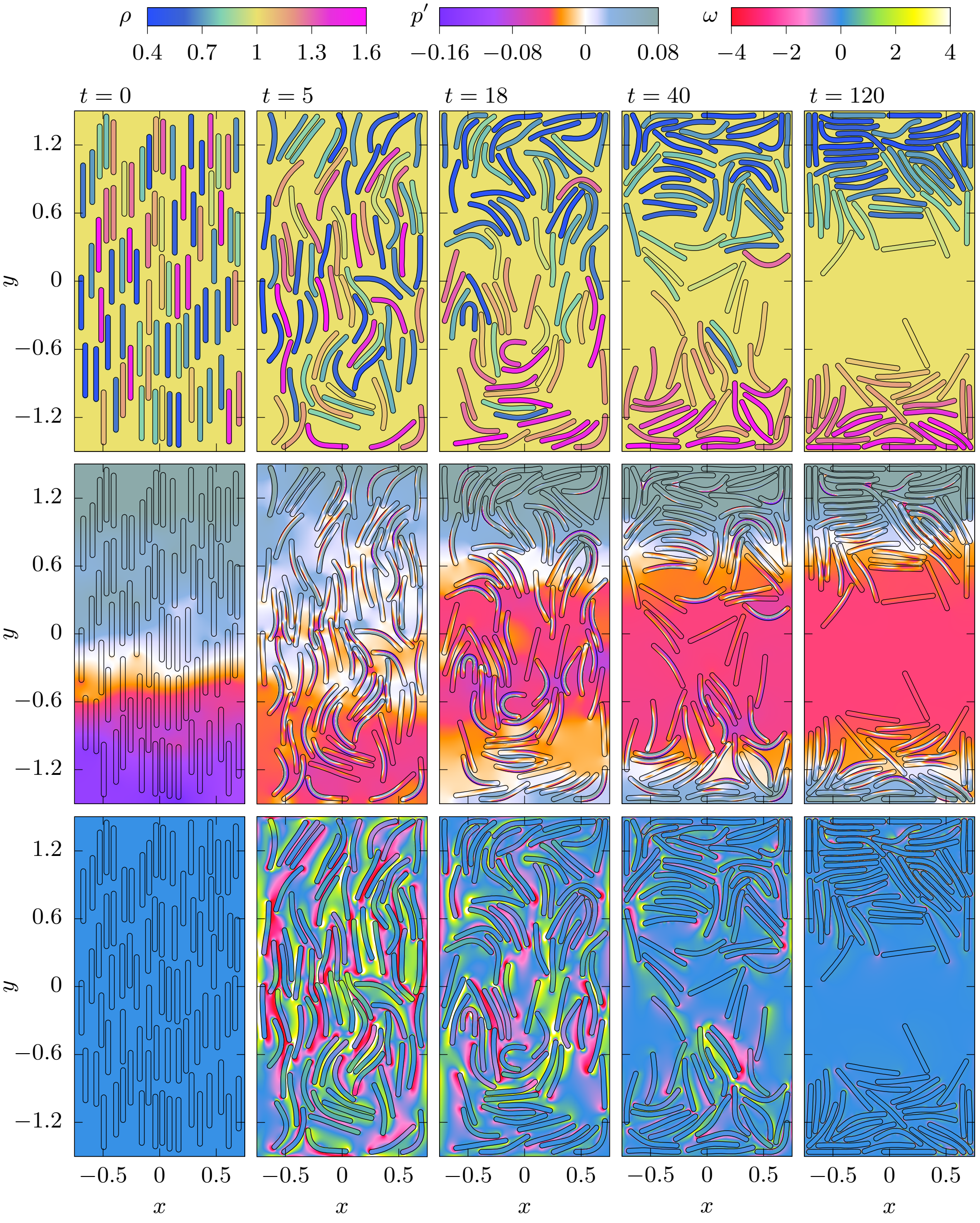

FIGURE 12. Snapshots of density ρ (top row), pressure deviation $p' = p + \rho_f gy$ (middle row), and vorticity ω (bottom row) in a simulation of 75 flexible rods of variable density $\rho_s \in [0.4, 1.6]$ rearranging in a fluid-filled box. The thick black lines mark the fluid–structure interfaces. Simulation parameters are $(\rho_f, \mu_f, G, g) = (1, 2 \times 10^{-3}, 1.5, 1)$. The plot of p' at $t = 0$ is based upon taking a small timestep of $\Delta t = 10^{-6}$ and computing pressure based on the finite-element projection.

In the particular random sample chosen, the average rod density is $\bar{\rho}_s \approx 0.904$. Thus the average density in the simulation is slightly lower than ρ_f and hence there is a small positive gradient in p' in the y direction at $t = 0$. The solids separate into two families, with solids with $\rho_s < \rho_f$ rising to the top of the domain, and solids with $\rho_s > \rho_f$ sinking to the bottom of the domain. While most rods have separated out by $t = 40$, it takes a long time for the separation process to fully complete, since several rods are close to neutrally buoyant and the reduced gravity they experience is small. By $t = 120$, all rods have completely separated into two families although there is still some residual movement visible. The density field, pressure deviation field, and vorticity field of the full simulation are shown in Supplemental Movie 7, Supplemental Movie 8, and Supplemental Movie 9, respectively.

5. Conclusion

Herein, we have presented a robustly accurate, yet straightforward to implement, reference map technique, which has allowed us to study a variety of FSI problems using a single background grid. It augments the multi-phase fluid framework of Almgren, Bell, and coworkers (Bell *et al.* 1989; Puckett *et al.* 1997; Almgren *et al.* 1998) by allowing general finite-deformation solid models to be coupled directly to a fluid. In doing so, it maintains a number of the advantages of working on a fixed Eulerian grid that are enjoyed in fluid simulation methods. The practicality and usefulness of this approach is demonstrated in various tests. It is shown to capture the flapping phase diagram for thin flags and the transition from thin- to thick-flag behaviors, which highlights the role of new mechanisms to initiate flapping. Additional physics, such as actuation of solids, is straightforward to implement with a user-described actuated deformation gradient. This capability is used to model a swimming object with realistic internal driving. The ability to model objects with sharp corners is typically a challenge in Eulerian approaches, but here it can be done by exploiting the reference map field near the edge of the object. We also present an improved contact algorithm, which we use to simulate situations with many soft interacting objects submerged in a fluid.

There are a number of future directions. One of clearest applications is in biomechanics, with the simulation of systems of many interacting, actuated cells. We also foresee modeling solids beyond hyperelasticity, such as plasticity, thermal material models, and growth. These modifications can be done through the inclusion of new state variables in the solid and/or the addition of a heat diffusion equation; there are clear advantages to implementing thermal diffusion in the Eulerian frame. Beyond extensions to three dimensions, there are opportunities to use the approach for dimensionally-reduced models such as membranes and shells by restricting the reference map to a lower dimensional set. Regarding contact modeling, the reference map field could be used to instruct formulations for more advanced contact problems, including friction and self-contact. Lastly, it is a major goal to extend the approach to allow for non-persistent material boundary sets, as occurs in fracture. It may be possible to represent crack surfaces through intersecting level set fields and to couple this capability with physical traction–separation relations to generate new surface material as cracks advance.

Appendix A. Additional numerical details

A.1. *Monotonicity-limited derivative*

The gradients of the reference map and velocity appearing in Eq. 3.5 are computed using the fourth-order monotonicity-limited scheme of Colella (1985). For the derivative of a generic component $a_{i,j}$ in the x direction, finite differences

$$D^c(a)_{i,j} = (a_{i+1,j} - a_{i-1,j})/2, \quad D^+(a)_{i,j} = a_{i+1,j} - a_{i,j}, \quad D^-(a)_{i,j} = a_{i,j} - a_{i-1,j} \tag{A 1}$$

are introduced, from which the limiting slope is defined as

$$\delta_{\mathrm{lim}}(a)_{i,j} = \begin{cases} 2 \times \min(|D^-(a)_{i,j}|, |D^+(a)_{i,j}|) & \text{if } D^-(a)_{i,j} D^+(a)_{i,j} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{A 2}$$

The second-order limited slope is then

$$\delta_f(a)_{i,j} = \min(|D^c(a)_{i,j}|, \delta_{\mathrm{lim}}(a)_{i,j}) \times \mathrm{sign}(D^c(a)_{i,j}) \tag{A 3}$$

from which the fourth-order monotonicity limited derivative is defined as

$$\delta^4(a)_{i,j} = \min\left(\frac{|8D^c(a)_{i,j} - \delta_f(a)_{i+1,j} - \delta_f(a)_{i-1,j}|}{6}, \delta_{\mathrm{lim}}(a)_{i,j}\right) \times \frac{\mathrm{sign}(D^c(a)_{i,j})}{h_x}. \tag{A 4}$$

The y-derivative is evaluated similarly.

A.2. *Tangential derivatives*

To ensure stability, the tangential derivatives appearing in Eqs. (3.5) & (3.6) are computed using

$$(\widetilde{v\boldsymbol{\xi}_y})^n_{i,j} = \frac{\tilde{v}^{\mathrm{adv}}_{i,j-1/2} + \tilde{v}^{\mathrm{adv}}_{i,j+1/2}}{2} \frac{\tilde{\boldsymbol{\xi}}_{i,j+1/2} - \tilde{\boldsymbol{\xi}}_{i,j-1/2}}{h_y}, \tag{A 5}$$

$$(\widetilde{v\mathbf{v}_y})^n_{i,j} = \frac{\tilde{v}^{\mathrm{adv}}_{i,j-1/2} + \tilde{v}^{\mathrm{adv}}_{i,j+1/2}}{2} \frac{\tilde{\mathbf{v}}_{i,j+1/2} - \tilde{\mathbf{v}}_{i,j-1/2}}{h_y}, \tag{A 6}$$

where the terms with tildes are computed using a preliminary Godunov upwinding step where stress, pressure, and tangential derivatives are neglected (Yu *et al.* 2003). Extrapolations to a vertical edge from the left are given by

$$\tilde{\boldsymbol{\xi}}^{L,n+1/2}_{i+1/2,j} = \boldsymbol{\xi}_{i,j} + \frac{1}{2}\left(h_x - u^n_{i,j}\Delta t\right)\boldsymbol{\xi}^n_{x,i,j}, \tag{A 7}$$

$$\tilde{\mathbf{v}}^{L,n+1/2}_{i+1/2,j} = \mathbf{v}^n_{i,j} + \frac{1}{2}\left(h_x - u^n_{i,j}\Delta t\right)\mathbf{v}^n_{x,i,j}, \tag{A 8}$$

and extrapolations to the other edges are given similarly. On each edge, the selection procedure of Eq. (3.8) is used, with Case A used for $\tilde{\mathbf{v}}^{\mathrm{adv}} = (\tilde{u}^{\mathrm{adv}}, \tilde{v}^{\mathrm{adv}})$ and Case B used for $\tilde{\boldsymbol{\xi}}$ and $\tilde{\mathbf{v}}$.

Appendix B. Tests of convergence and accuracy

B.1. *Overview of the test configurations*

To study the accuracy of the numerical method, we performed a convergence test in the periodic domain $[-1,1)^2$ using an initial incompressible velocity field of

$$\mathbf{v}(\mathbf{x},0) = \sum_{k=0}^{5} (-1)^k \mathbf{v}_{\mathrm{vor}}\left(x - \frac{-5+2k}{6}, y - \frac{-5+2k}{6}, 2(k+1)\right) \tag{B 1}$$

where

$$\mathbf{v}_{\mathrm{vor}}(\mathbf{x},\lambda) = (-\sin \pi y, \cos \pi x) \times e^{-\lambda(2-\cos \pi x - \cos \pi y)}. \tag{B 2}$$

This velocity field is designed to have features with a variety of length scales. We simulated up to $t = 0.5$, used a shear modulus of $G = 1$, a fluid density of $\rho_f = 1$, and employed the

Test	State	μ_f	ρ_s	CEV	$\mathbf{v}, L^2$	$\mathbf{v}, L^1$	$\mathbf{v}, L^\infty$	p, L^2
A	Fluid only	10^{-3}	–	No	1.92 *(1.00)*	1.94 *(1.00)*	1.92 *(1.00)*	2.00 *(1.00)*
B	Fluid only	4×10^{-3}	–	No	1.98 *(1.00)*	1.99 *(1.00)*	1.98 *(1.00)*	2.00 *(1.00)*
C	Solid only	10^{-3}	1	No	0.96 *(1.00)*	0.97 *(1.00)*	0.94 *(1.00)*	0.96 *(1.00)*
C’	Solid only	10^{-3}	1	Yes	2.13 *(0.72)*	2.16 *(0.79)*	1.20 *(0.00)*	1.98 *(0.93)*
D	Square	10^{-3}	1	No	1.31 *(1.00)*	1.20 *(0.91)*	0.77 *(0.65)*	0.59 *(0.94)*
E	Circle	10^{-3}	3	No	1.32 *(1.00)*	1.27 *(1.00)*	1.40 *(1.00)*	0.47 *(0.95)*
F	Circle	10^{-3}	1	No	1.29 *(0.99)*	1.28 *(1.00)*	1.55 *(0.05)*	0.61 *(0.88)*
F’	Circle	10^{-3}	1	Yes	1.04 *(0.99)*	1.08 *(0.97)*	0.88 *(1.00)*	0.55 *(0.93)*

TABLE 1. Details of the eight convergence tests that were performed with model problem described in the text. Tests C’ and F’ were performed using constant extra viscosity (CEV) whereby the extra viscosity was held constant at the standard value for the lowest resolution grid, 360 × 360, as opposed to scaling linearly with the grid spacing. The last four columns give the exponents of convergence for velocity $\mathbf{v}$ and pressure p under different L^q norms, based on a linear fit of the three-parameter error model of Eq. (B 9) that incorporates a Richardson extrapolation correction. The proportion of Richardson extrapolation correction is shown in italics in brackets.

standard choices for extra viscosity and timestep selection. Using the same initial velocity field, we ran tests using (i) fluid only, (ii) solid only, (iii) a circle of radius 0.6 centered on $(-0.1, 0)$, and (iv) a square of side length 1.2 centered on $(-0.1, 0)$. We examined the effect of viscosity and the fluid/solid density ratio.

The configurations of eight different tests are shown in Table 1. In our tests, we also considered two different models for the scaling of the extra viscosity. Our primary tests A–F follow Sec. 3.5 and choose it to scale linearly with the grid size. This procedure is consistent with standard numerical schemes; for example, in the second-order Lax–Wendroff method (Lax & Wendroff 1960; LeVeque 2002) the stabilizing diffusive term scales linearly with the grid spacing. However, we also considered alternative tests C’ and F’, whereby the extra viscosity is viewed as a physical dissipation within the solid and is therefore held constant rather than scaling with the grid spacing. It is chosen based on the 360 × 360 grid and then held constant for the higher-resolution grids.

Due to the complexity of the governing equations, it is near-impossible to write down an analytical solution to compare against for any test configuration. We therefore performed reference simulations using a 5040 × 5040 grid. For each test, we then ran a suite of coarser simulations using $N \times N$ grids where $N \in \{2520, 1680, 1260, 1008, 840, 720, 630, 560, 504, 420, 360\}$ to compare against the reference results. Since each N divides evenly into 5040, the grid squares of these coarse simulations align with the reference simulations.

We calculated normalized error measures with respect to L^q norms

$$E_q^p = \left(\frac{1}{A} \int_\Omega |p_{\mathrm{ref}}(\mathbf{x}) - p_{\mathrm{coa}}(\mathbf{x})|^q \,\mathbf{dx} \right)^{1/q}, \quad E_q^{\mathbf{v}} = \left(\frac{1}{A} \int_\Omega \|\mathbf{v}_{\mathrm{ref}}(\mathbf{x}) - \mathbf{v}_{\mathrm{coa}}(\mathbf{x})\|_2^q \,\mathbf{dx} \right)^{1/q}, \tag{B 3}$$

where $A = 4$ is the area of the domain, and the ‘ref’ and ‘coa’ subscripts refer to the reference and coarse simulation fields, respectively. The integral is calculated using a direct sum over the field values in the coarser simulation grid. The pressure field is cell-cornered, and hence each coarse gridpoint exactly coincides with a reference gridpoint. The velocity field is cell-centered, so some coarse gridpoints may not align with a reference gridpoint, in which case the reference value is computed using bilinear interpolation. The errors

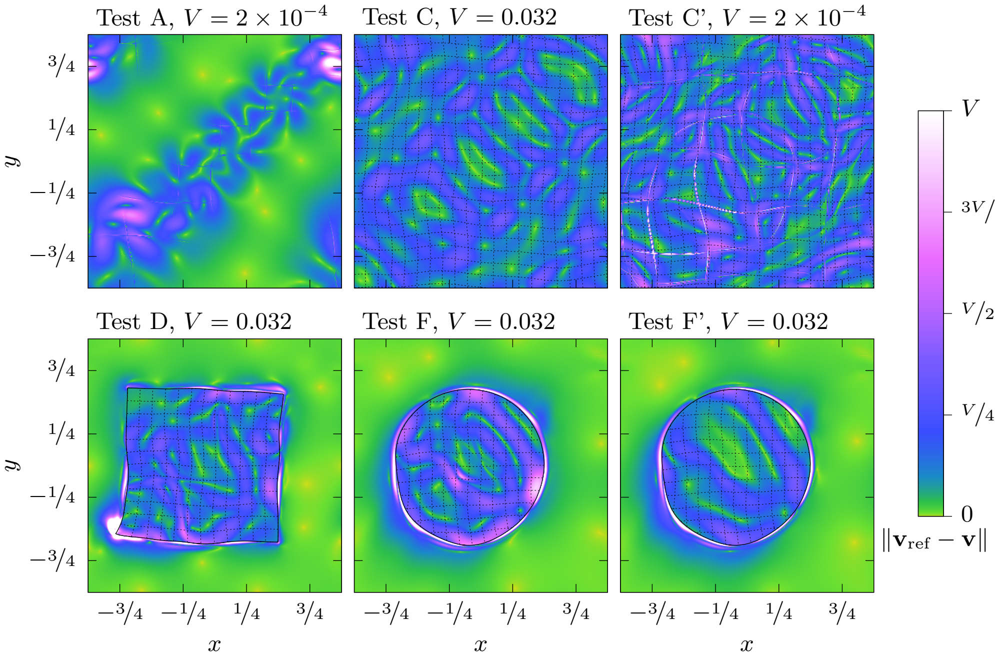

Figure 13. Differences between the velocity fields in the reference simulation (using a 5040×5040 grid) and the coarsest simulation (using a 360×360 grid). Plots are shown at $t = 0.5$ for six of the converegence tests. The colors in each panel are normalized differently by a maximum value V. The thick black lines mark the fluid–structure interfaces. The thin dashed lines are contours of the components of the reference map. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G) = (1, 10^{-3}, 1, 1)$.

associated with this interpolation are $O(h^3)$ and are small compared to the errors to be measured.

Figure 13 shows plots of the difference in velocity fields between the reference simulation and the simulation on the coarsest grid, for six of the convergence tests at $t = 0.5$. The colors in the panels are normalized differently, with differences for tests A and C' being much smaller than those for the other tests that are shown. In the fluid-only test A, the largest errors are present on the diagonal line $x = y$, where the initial vortices are located. Some higher errors are visible on thin curved lines, which is a consequence of the switching between cases in the advection discretization. Test C shows the errors for the solid-only simulation, which are about two orders of magnitude larger than test A. In test C' where the extra viscosity is held fixed, the additional dissipation allows a closer match to be achieved. Tests D, F, and F' show that the largest errors are all near the fluid–solid interface. Unlike models C and C', there is limited difference between models F and F', since the errors near the interface dominate.

B.2. *Calculating convergence rates*

Figure 14 shows convergence plots for the velocity in the L^2, L^1, and L^∞ norms, plus the pressure in the L^2 norm; our discussion focuses on velocity, since the pressure can be viewed as a Lagrange multiplier enforcing the incompressibility constraint. If the method is of order s, and the reference solution is treated as exact, then the errors scale according to

$$E_q^a(h) \approx B h^s \tag{B 4}$$

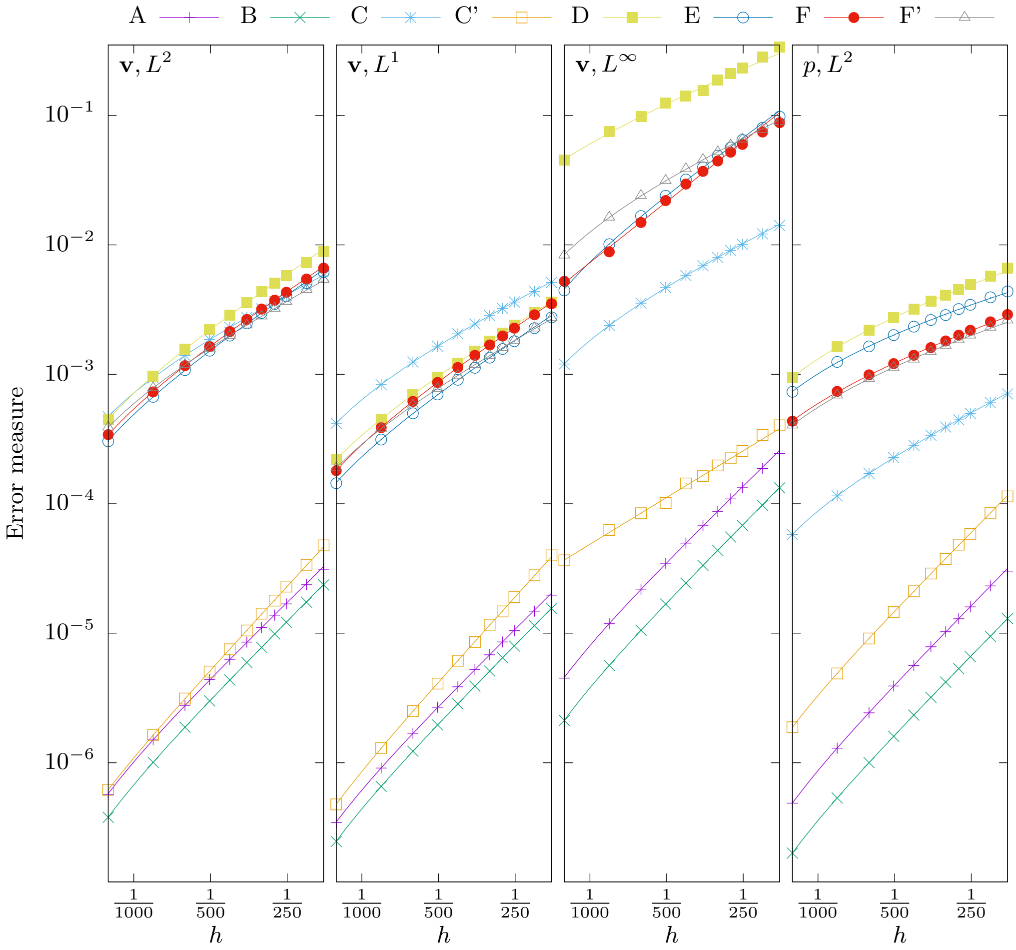

FIGURE 14. Plots showing the accuracy of the solutions for different grid sizes h for the eight convergence tests given in Table 1. Accuracy is computed with respect to reference solutions on a 5040×5040 grid. Four accuracy measures are shown: the velocity in the L^2, L^1, and L^∞ norms, and the pressure in the L^2 norm. The symbols correspond to the computed errors with Eq. (B 3) and the lines correspond to the fitted three-parameter error model using Eq. (B 9).

for some constant B where a is either p or $\mathbf{v}$. However, in reality the reference solution will not be exact. In particular, one could apply Richardson extrapolation (Richardson 1911; Hairer *et al.* 1993; Heath 2002) and propose that the numerical solution has a Taylor series in h, so the leading-order error term is of the form h^s. Specifically, let $f(h) \in C^1(\Omega)$ be a representation of a component (u, v, or p) of the numerical solution computed with grid spacing h so that it agrees with the numerical values at the grid points, and smoothly interpolates between them. Then

$$f(h) = f_0 + f_1 h^s + O(h^{s+1}) \tag{B 5}$$

for some smooth functions $f_0, f_1 \in C^1(\Omega)$. Under this assumption the error scales according to

$$E_q^a(h) = B(h^s - h_{\text{ref}}^s) + O(h^{s+1}). \tag{B 6}$$

However, for the current method, Eq. (B 5) is not precisely true, since there are several steps in the numerical method are not Taylor expandable to higher order. The advective terms in the discretization rely on discrete switching between different cases, which manifests as the lines of higher error in tests A and C' in Fig. 13. When a grid point leaves the solid, the $\boldsymbol{\xi}$ switches from a time-integrated value to an extrapolated value, causing a small, discrete jump in the field, potentially contributing to errors near the boundary.

From Fig. 13 it is apparent that most of the errors in the domain are smooth, and regular. We therefore propose a model whereby the simulation domain is split into Ω_T where the Richardson error model, Eq. (B 6), is applied, and Ω_S where the original error model, Eq. (B 4), is applied. This leads to a three-parameter error model

$$E_q^a(h) = B(h^s - \alpha h_{\text{ref}}^s) + O(h^{s+1}), \tag{B 7}$$

where $\alpha \in [0,1]$ is the proportion of the Richardson error correction. Taking the logarithm of Eq. (B 7) yields

$$\log E_q^a = C + s \log h + \log \left(1 - \alpha \frac{h_{\text{ref}}^s}{h^s}\right) \tag{B 8}$$

where $C = \log B \in \mathbb{R}$. Define $X = \log h_{\text{ref}}$, and let (x_k, y_k) be pairs $(\log h, \log E_q^a(h))$ for each lower resolution that was considered. Then s, C, and α may be determined by solving the constrained nonlinear least squares problem

$$\phi(C, s, \alpha) = \sum_k \Big(C + s x_k + \log(1 - \alpha e^{s(X - x_k)}) - y_k\Big)^2. \tag{B 9}$$

Equation (B 9) fits all 32 data sets in Fig. 14 accurately, and the Richardson term correctly captures how most data sets curve downwards. The convergence rates, and proportion of Richardson correction are shown in Table 1. The fluid-only tests, A & B, are the most accurate, exhibiting clear second-order convergence across all metrics. Results for the solid-only test C are less accurate with error measures on the scale of 10^{-3}, and a convergence rate of one is seen across all four metrics in Table 1. This is due to the linear scaling of the extra viscosity, which effectively results in changing the physical parameters as the grid spacing changes, approaching the limit of a non-dissipative process as $h \to 0$. In test C' where the extra viscosity is held fixed, and the physical parameters remain the same, a convergence rate of two is achieved in the L^1 and L^2 norms. Thus second-order accuracy of the solid discretization is achieved, but only for the case where damping is a fixed physical parameter. It remains an open question to find a second-order discretization for a perfectly non-dissipative solid.

The remaining tests, D, E, F, and F' all involve fluid–structure interaction. In the L^1 and L^2 norms, the convergence rate is approximately 1.3 for tests D, E, and F, and 1.0 for test F'. As seen in Fig. 13 the largest deviations occur at the fluid–structure interface. Since the blur zone is defined in terms of grid points, its width shrinks at higher resolution. This involves altering the underlying equations over a region of size $O(h)$, and results in a lower rate of convergence. However, since the fluid and solid discretizations are independently second order, is likely that an alternative boundary treatment—perhaps using a sharp interface approach (Gibou & Fedkiw 2005; Francois *et al.* 2006)—could yield improved results. Test E shows that a fluid–solid density ratio has little effect on the convergence rate. Test D shows that the square geometry does not affect the convergence rate in the L^2 and L^1 norms, but results in a lower convergence rate in the L^∞ norm due to localized effects at the corners.

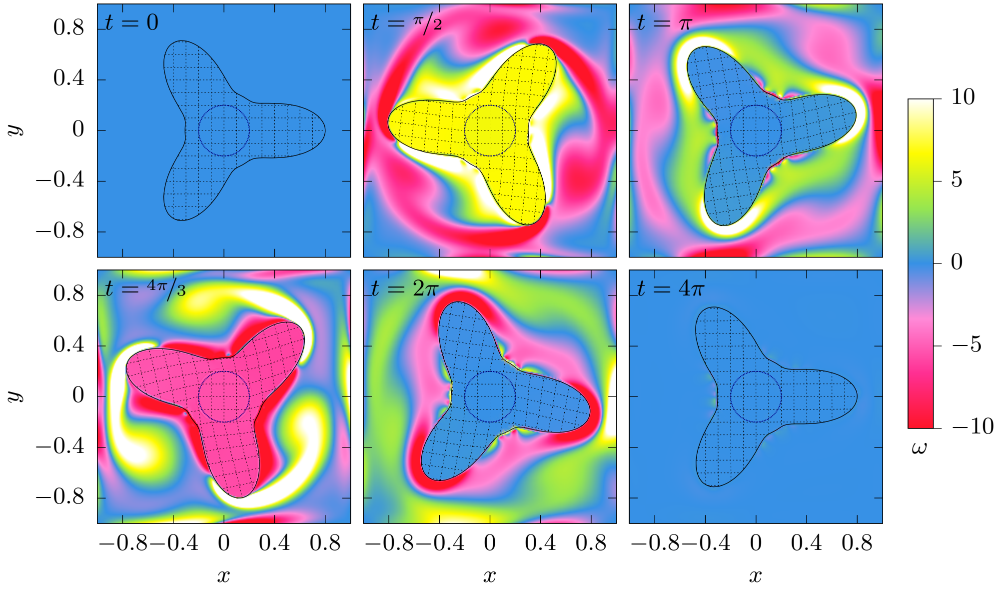

FIGURE 15. Snapshots of vorticity ω in a simulation of three-pronged rotor being spun with an oscillatory motion in a fluid. The thick black line marks the fluid–structure interface. The thin dashed lines are contours of the components of the reference map and indicate how the rotor has deformed. The dark blue dotted circle shows the pivot region. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G) = (1, 10^{-2}, 3, 48)$.

Appendix C. Performance tests over a range of resolutions

The simulations that are shown in the main text make use of high resolution to ensure accurate results. Here, we demonstrate that the method can work over a wide range of resolutions, and we examine its performance. We consider a three-pronged object whose boundary is described in polar coordinates by

$$r(\theta) = R\frac{1 + \alpha_3 \cos 3\theta + \alpha_6 \cos 6\theta}{1 + \alpha_3 + \alpha_6}. \tag{C 1}$$

The simulation domain is $|x| \leqslant 1, |y| \leqslant 1$ and no-slip boundary conditions are used on all sides. An $N \times N$ grid is used. Parameters of $\rho_f = 1$, $\rho_s = 3$, $\mu_f = 10^{-2}$, $G = 48$ are used, and the simulation duration is $T = 4\pi$. The shape is parameterized with $(R, \alpha_3, \alpha_6) = (0.8, 0.5, 0.125)$. The shape is rotated via a pivot centered at the origin of radius 0.2, whose angle is set to

$$\theta(t) = \begin{cases} \pi(1 - \cos t) & \text{if } t < 2\pi, \\ 0 & \text{if } t \geqslant 2\pi, \end{cases} \tag{C 2}$$

following the same pinning method as in Eq. (4.1) with a stiffness constant of $K_{\text{teth}} = 10^{-1}\rho_s \mathit{\Delta} t^{-2}$. The RMS angular velocity for $T \in [0, 2\pi]$ is $\omega_{\text{RMS}} = \pi/2$, Hence, we define the Reynolds number to be

$$Re = \frac{\rho_f R(\omega_{\text{RMS}} R)}{\mu_f} \approx 100. \tag{C 3}$$

Figure 15 shows a snapshot of vorticity for six different time points for an intermediate resolution of $N = 240$. Since the viscosity is higher by a factor of ten from the example in

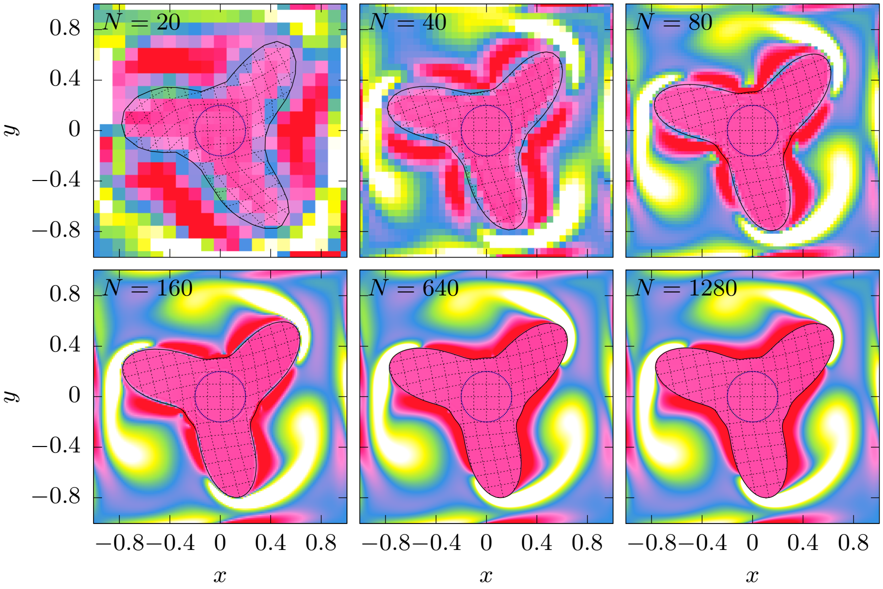

FIGURE 16. Snapshots of vorticity ω in the three-pronged rotor simulation at time $t = 4\pi/3$ using an $N \times N$ computational grid for six different values of N. The thick black line marks the fluid–structure interface. The thin dashed lines are contours of the components of the reference map and indicate how the rotor has deformed. The colors use the same scale as in Fig. 15. The dark blue dotted circle shows the pivot region. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G) = (1, 10^{-2}, 3, 48)$.

Sec. 4.1, fewer vortices are visible. Figure 16 shows snapshots at $T = 4\pi/3$ for a range of resolutions from $N = 20$ to $N = 1280$. At lower resolution, the accuracy of the solid deformation and the fluid flow is reduced, but the flow is qualitatively similar, and the simulation runs successfully.

Figure 17(a) shows a comparison of wall clock times for a Linux computer with dual 10-core 2.2 GHz Intel Xeon E5-2630 processors. The total simulation time varies over a large range, from 8.1 s for $N = 20$ to 67.7 h for $N = 800$ when using a single thread. Using multiple threads creates a substantial speedup, reducing the time for $N = 800$ to 20.9 h for four threads and 9.4 h for sixteen threads; these times correspond to parallel efficiencies of 80.1% and 44.9%, respectively. Some parts of the simulation (*e.g.* the extrapolation routine) are not threaded, and contribute to this loss of efficiency.

Figure 17(b) shows the timestep restrictions due to the liquid viscosity, the shear wave CFL condition, and the extra solid viscosity. For small grid sizes, the timestep is set by the extra solid viscosity, but for $n > 640$ the fluid viscosity provides the strongest restriction. Because of this, the total work scales like N^3 for most of the data in Fig. 17(a), but will eventually transition to N^4 once the fluid viscosity becomes important.

A large fraction of the computation time is spent solving the two linear systems for the marker-and-cell projection (Sec. 3.1.2) and the finite-element projection (Sec. 3.4). These are solved using a C++/OpenMP multigrid library (Appendix F). The library works on any grid size, and uses a hierarchy of grids where the grid dimensions are halved at each successive level. On very small grids, it is inefficient to use threads due to the performance overhead of thread initialization. Because of this, the library self-tunes prior to use to determine the grid level at which to enable threading. Figure 17(c) shows the average

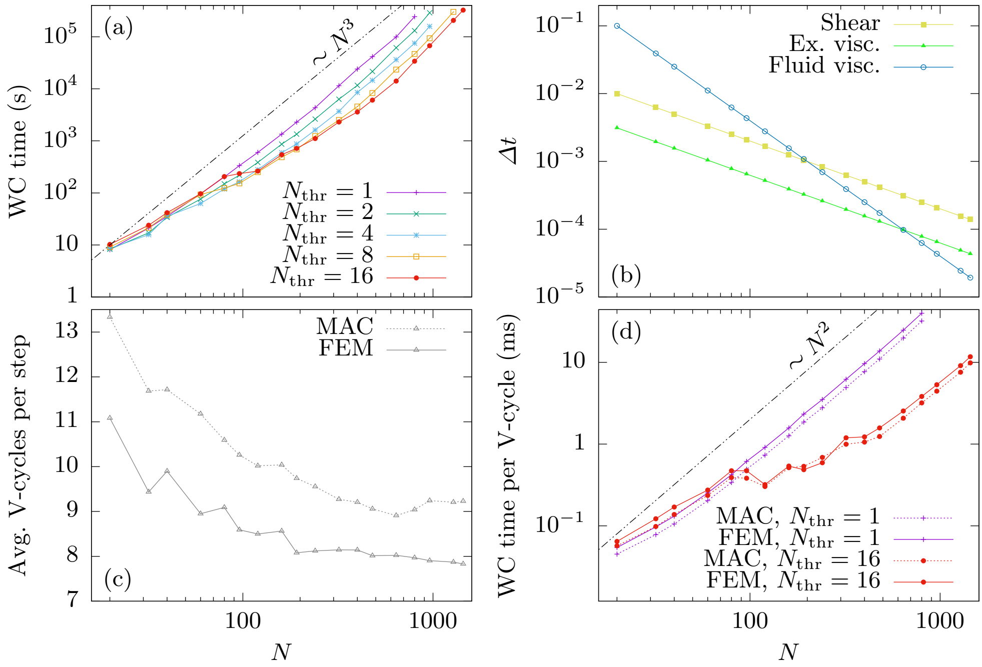

FIGURE 17. Performance of the simulation code for the three-pronged rotor in Appendix C when run on simulations with different $N \times N$ grids. (a) Total wall-clock (WC) time as a function of N for different numbers of threads. (b) The three timestep restrictions based on the shear wave speed, extra viscosity, and fluid viscosity as a function of N. (c) The average number of V-cycles required to solve the marker-and-cell (MAC) linear system and the finite-element method (FEM) linear system as a function of N. (d) The wall-clock time for performing a single V-cycle on different grid sizes, using one and sixteen threads, as a function of N.

number of V-cycles that are performed to reach the required error tolerances, showing that there is limited variation as N changes. Figure 17(d) shows the time required to perform a V-cycle as a function of N. As expected the single-threaded times scale like N^2, proportional to the degrees of freedom. For $N \leqslant 100$ the time for sixteen threads is similar to the single-threaded performance, since the grids are not large enough for threading to be enabled. However, for large grids, threading results in more than a 10× speedup. Across all simulation sizes and thread numbers, the MAC and finite-element linear systems each take between 20%–30% of the total computation time.

Appendix D. Effect of the blur zone

As described in Sec. 2, the numerical method uses a transition region of width 2ϵ to blur between the fluid and solid stresses at an interface. The form of the blurring function is given by Eq. (2.9) based on previous studies (Sussman *et al.* 1994, 1999; Yu *et al.* 2003, 2007), and throughout the paper we use $\epsilon = 2.5h$. Here, we explore the effect of the blur zone width, by running the three-pronged rotor of Appendix C using different values of ϵ.

Figure 18 shows close-ups of the vorticity field for four different values of ϵ at $t = 2\pi$. At this time point, the clockwise motion of the rotor is arrested, which causes momentarily large shear stresses. If $\epsilon = 0$, so that there is a hard transition between fluid and solid stress, then this creates numerical noise in the vorticity field near the interfaces. Since vorticity is based on first derivatives, the variations in the underlying velocity field are

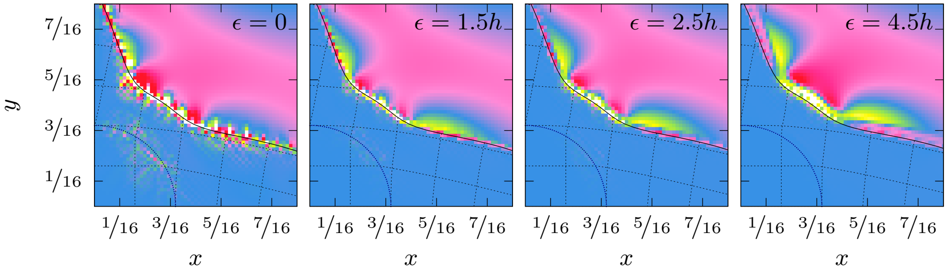

FIGURE 18. Close-ups of vorticity ω the three-pronged rotor simulation at $t = 2\pi$ using a 240×240 grid for varying values of the blur zone half-width, ϵ. The thick black line marks the fluid–structure interface. The thin dashed lines are contours of the components of the reference map and indicate how the rotor has deformed. The colors use the same scale as in Fig. 15. The dark blue dotted quarter-circle shows the pivot region. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G) = (1, 10^{-2}, 3, 48)$.

smaller. In these simulations the tethering force (Eq. (4.1)) is also smoothed using the same value of ϵ, and hence numerical noise is also visible at the edge of the pivot region. Despite the noise, the simulation with $\epsilon = 0$ runs adequately. As ϵ increases the noise at the interface is progressively blurred out. Figure 19(a) shows a plot of the maximum vorticity field in the simulation over time for differing values of blur zone width, highlighting the trade-off between additional noise for small ϵ, and excessive blurring for large ϵ. Our default value of $2.5h$ is an acceptable compromise between the two limits.

The precise form of smoothed delta and Heaviside functions has been studied in the literature, and we also considered the delta function studied by Tornberg & Engquist (2003, 2004) where

$$\delta_\epsilon^{\mathrm{alt}}(\phi) = \max\left\{0, \tfrac{1}{\epsilon} - \tfrac{|\phi|}{\epsilon^2}\right\}. \tag{D 1}$$

Equation (D 1) satisfies several discrete moment conditions (Beyer & LeVeque 1992) and has better convergence properties for some problems. We calculated an associated Heaviside function using integration, and ran a variant of convergence test F from Appendix B. The measured convergence exponents are within ±0.03 of the original test. Hence, for the current method, errors at the boundary are dominated by other processes, such as extrapolation. Designing better smoothing and extrapolation is an interesting avenue for further study.

Appendix E. Volume conservation

Since the reference map field represents an incompressible solid, the determinant of the deformation gradient $J = \det \mathbf{F}$ should remain equal to unity throughout the simulation, but in the numerical scheme this property will only be maintained approximately. This issue is encountered in other incompressible fluid–structure interaction approaches, such as the immersed boundary method where the Lagrangian description of the solid is updated from the fluid velocity and may experience volumetric changes over time (Wang *et al.* 2009; Griffith 2012; Vadala-Roth *et al.* 2020).

To investigate the magnitude of volumetric deviations, we simulate the three-pronged rotor of Appendix C and compute the field $J - 1$. The field is evaluated on all cell corners in the interior of the solid, where the bilinear interpolation of ϕ is negative. Calculating J requires the gradient of the reference map, which is computed at each cell corner using

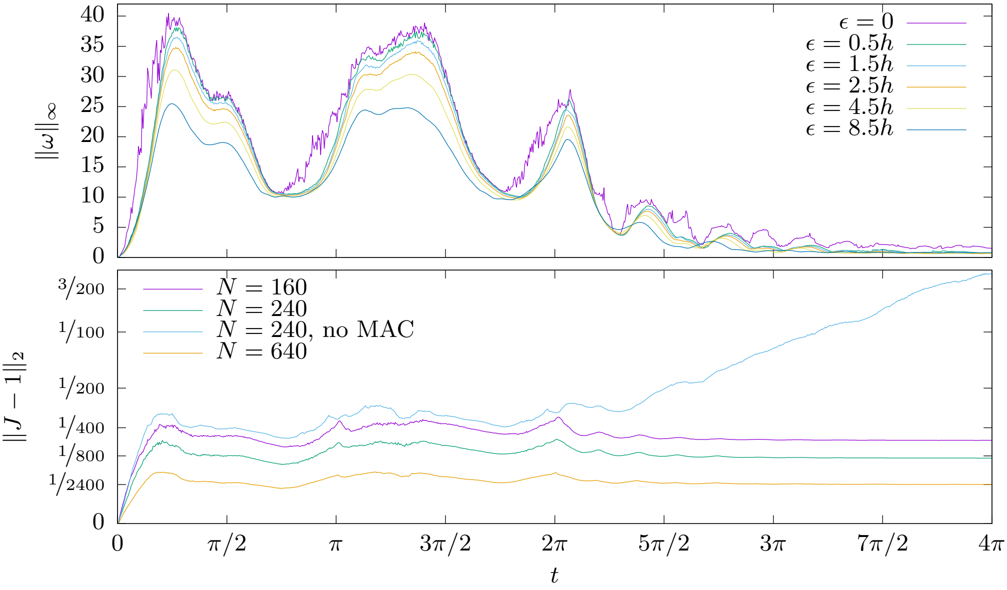

FIGURE 19. (a) Plots of the maximum value of vorticity field as a function of time, in the three-pronged rotor simulations using six different values of the blur zone half-width ϵ. (b) Plots of the difference in the Jacobian J from unity in the solid for the three-pronged rotor simulations for several $N \times N$ grids. A nonlinear square root scale is used on the vertical axis to account for the substantially larger differences in J for the simulation without the MAC projection.

centered finite differences of the four adjoining cell-centered reference map fields, some of which may be based on extrapolation.

Figure 20 shows snapshots of $J - 1$ for several simulation configurations. The top row shows three snapshots on a 240×240 grid. Small volumetric deviations on the order of 10^{-2} are visible and are concentrated near the fluid–solid interface, and at the edge of the pivot region. Figure 19(b) shows a plot of $\|J - 1\|_2$ as a function of time. The volumetric deviations grow rapidly up to $t = \pi/4$ but then remain relatively stable; in particular, for $t > 2\pi$ when the pivot stops rotating, $\|J - 1\|_2$ tends to a constant value. The volumetric deviations have the expected dependence on resolution: as shown in Figs. 19 & 20 the errors are larger for $N = 160$ and smaller for $N = 640$. The values of $\|J - 1\|_2$ at $t = 4\pi$ scale approximately like $h^{1.1}$, consistent with the convergence rates in Table 1. We find that the MAC projection substantially improves the volume conservation of the reference map: switching off the projection results in larger deviations overall, plus a tendency for them to grow once the pivot stops rotating (Fig. 19). This can cause considerable errors near the pivot, as shown in the bottom right panel of Fig. 20.

Appendix F. Computer code

All results in this paper were created using a custom C++ code that uses OpenMP for multithreading. The code is released as an open source software package, IncRMT, via a GitHub repository at `https://github.com/chr1shr/incrmt`. As discussed in Appendix C, a large fraction of the computation time is spent solving the MAC projection and finite-element projection to enforce incompressibility. This is done using the TGMG (Templated Geometric Multigrid) library, which is available on GitHub at `https://github.com/chr1shr/tgmg`.

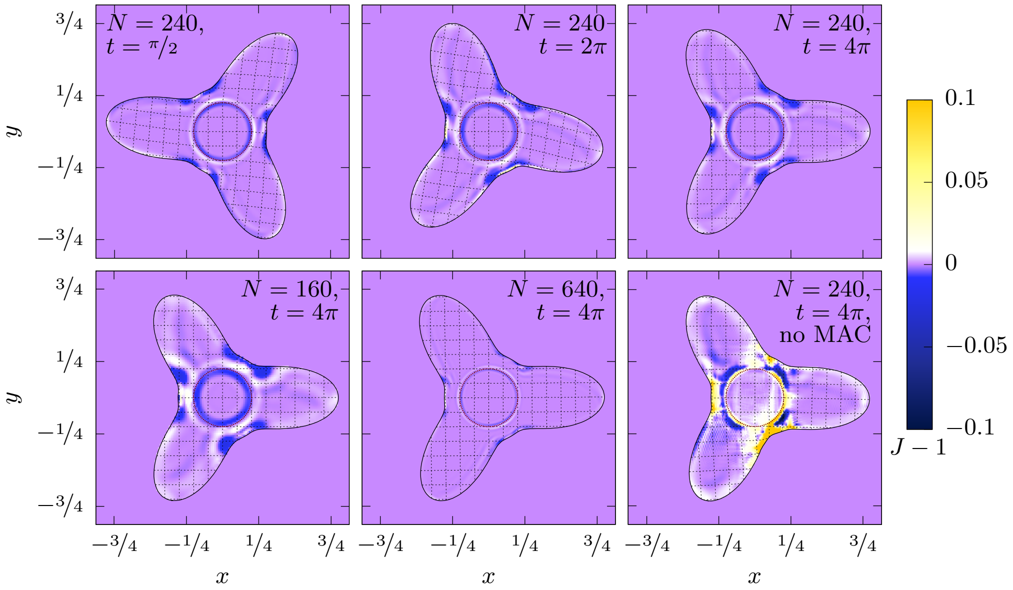

FIGURE 20. Snapshots of volumetric change in the solid, $J - 1$, in several simulations of the three-pronged rotor on $N \times N$ grids. The thick black line marks the fluid–structure interface. The thin dashed lines are contours of the components of the reference map and indicate how the rotor has deformed. The dark red dotted circle shows the pivot region. The bottom right panel is from a simulation where the MAC projection is switched off. Simulation parameters are $(\rho_f, \mu_f, \rho_s, G) = (1, 10^{-2}, 3, 48)$.

Acknowledgments

We thank the three anonymous reviewers for constructive feedback on this work. C. H. Rycroft was partially supported by the Applied Mathematics Program of the U.S. DOE Office of Science Advanced Scientific Computing Research under contract number DE-AC02-05CH11231. Y. Yu acknowledges support from the National Science Foundation under Award No. DMS-1620434 and the Class of 1968 Junior Faculty Fellowship from Lehigh University. K. Kamrin acknowledges support from Army Research Office grants W911NF-16-1-0440, W911NF-18-1-0118, and W911NF-19-1-0431.

Declaration of Interests

The authors report no conflict of interest.

REFERENCES

Almgren, Ann S., Bell, John B., Colella, Phillip, Howell, Louis H. & Welcome, Michael L. 1998 A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. *J. Comput. Phys.* **142** (1), 1–46.

Almgren, Ann S., Bell, John B. & Szymczak, William G. 1996 A numerical method for the incompressible Navier–Stokes equations based on an approximate projection. *SIAM J. Sci. Comput.* **17** (2), 358–369.

Argentina, Médéric & Mahadevan, Lakshminarayanan 2005 Fluid-flow-induced flutter of a flag. *Proc. Natl. Acad. Sci.* **102** (6), 1829–1834.

ASLAM, TARIQ D. 2004 A partial differential equation approach to multidimensional extrapolation. *J. Comput. Phys.* **193** (1), 349–355.

BATHE, KLAUS-JÜRGEN 2007 Proceedings of the fourth MIT conference on computational fluid and solid mechanics. Elsevier Science.

BELL, JOHN B., COLELLA, PHILLIP & GLAZ, HARLAND M. 1989 A second-order projection method for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **85** (2), 257–283.

BELLOTTI, THOMAS & THEILLARD, MAXIME 2019 A coupled level-set and reference map method for interface representation with applications to two-phase flows simulation. *J. Comput. Phys.* **392**, 266–290.

BELYTSCHKO, TED, LIU, WING KAM, MORAN, BRIAN & ELKHODARY, KHALIL 2013 *Nonlinear Finite Elements for Continua and Structures*, second edition edn. Wiley.

BEYER, R. P. & LEVEQUE, RANDALL J. 1992 Analysis of a one-dimensional model for the immersed boundary method. *SIAM Journal on Numerical Analysis* **29** (2), 332–364, arXiv: https://doi.org/10.1137/0729022.

BROWN, DAVID L., CORTEZ, RICARDO & MINION, MICHAEL L. 2001 Accurate projection methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **168** (2), 464–499.

CHOPP, DAVID L. 2001 Some improvements of the fast marching method. *SIAM J. Sci. Comput.* **23** (1), 230–244.

CHOPP, DAVID L. 2009 Another look at velocity extensions in the level set method. *SIAM J. Sci. Comput.* **31** (5), 3255–3273.

CHORIN, ALEXANDRE J. 1967 A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.* **2** (1), 12–26.

CHORIN, ALEXANDRE J. 1968 Numerical solution of the Navier–Stokes equations. *Math. Comput.* **22** (104), 745–762.

COLELLA, PHILLIP 1985 A direct Eulerian MUSCL scheme for gas dynamics. *SIAM J. Sci. Stat. Comput.* **6** (1), 104–107.

COLELLA, PHILLIP 1990 Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.* **87** (1), 171–200.

CONNELL, BENJAMIN S. H. & YUE, DICK K. P. 2007 Flapping dynamics of a flag in a uniform stream. *J. Fluid Mech.* **581**, 33–67.

COQUERELLE, MATHIEU & COTTET, GEORGES-HENRI 2008 A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *J. Comput. Phys.* **227** (21), 9121–9137.

COTTET, GEORGES-HENRI, MAITRE, EMMANUEL & MILCENT, THOMAS 2008 Eulerian formulation and level set models for incompressible fluid–structure interaction. *ESAIM: Mathematical Modelling and Numerical Analysis* **42**, 471–492.

COURANT, RICHARD, FRIEDRICHS, KURT & LEWY, HANS 1967 On the partial difference equations of mathematical physics. *IBM Journal of Research and Development* **11** (2), 215–234.

DEMMEL, JAMES W. 1997 *Applied Numerical Linear Algebra*. SIAM.

DUNNE, THOMAS 2006 An Eulerian approach to fluid–structure interaction and goal-oriented mesh adaptation. *Int. J. Numer. Methods Fluids* **51** (9-10), 1017–1039.

DUNNE, THOMAS, RANNACHER, ROLF & RICHTER, THOMAS 2010 *Numerical simulation of fluid–structure interaction based on monolithic variational formulations*, pp. 1–75. World Scientific.

ENGELS, THOMAS, KOLOMENSKIY, DMITRY, SCHNEIDER, KAI & SESTERHENN, JÖRN 2015 Numerical simulation of fluid–structure interaction with the volume penalization method. *J. Comput. Phys.* **281**, 96–115.

FACHINOTTI, VÍCTOR D., CARDONA, ALBERTO & JETTEUR, PHILIPPE 2008 Finite element modelling of inverse design problems in large deformations anisotropic hyperelasticity. *Int. J. Numer. Meth. Eng.* **74** (6), 894–910.

FAI, THOMAS G., GRIFFITH, BOYCE E., MORI, YOICHIRO & PESKIN, CHARLES S. 2013 Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers I: Numerical method and results. *SIAM J. Sci. Comput.* **35** (5), B1132–B1161.

FRANCOIS, MARIANNE M., CUMMINS, SHAREN J., DENDY, EDWARD D., KOTHE, DOUGLAS B., SICILIAN, JAMES M. & WILLIAMS, MATTHEW W. 2006 A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.* **213** (1), 141–173.

Froehle, Bradley & Persson, Per-Olof 2015 Nonlinear elasticity for mesh deformation with high-order discontinuous galerkin methods for the Navier–Stokes equations on deforming domains. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, pp. 73–85. Springer.

Gazzola, Mattia, Argentina, Médéric & Mahadevan, Lakshminarayanan 2015 Gait and speed selection in slender inertial swimmers. *Proc. Natl. Acad. Sci.* **112** (13), 3874–3879.

Gazzola, Mattia, Chatelain, Philippe, van Rees, Wim M. & Koumoutsakos, Petros 2011 Simulations of single and multiple swimmers with non-divergence free deforming geometries. *J. Comput. Phys.* **230** (19), 7093–7114.

Gibou, Frédéric & Fedkiw, Ronald 2005 A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *J. Comput. Phys.* **202** (2), 577–601.

Govindjee, Sanjay & Mihalic, Paul A. 1996 Computational methods for inverse finite elastostatics. *Comput. Method. Appl. M.* **136** (1–2), 47–57.

Griffith, Boyce E. 2012 On the volume conservation of the immersed boundary method. *Commun. Comput. Phys.* **12** (2), 401–432.

Griffith, Boyce E., Luo, Xiaoyu, McQueen, David M. & Peskin, Charles S. 2009 Simulating the fluid dynamics of natural and prosthetic heat valves using the immersed boundary method. *Int. J. Appl. Mech.* **1** (1), 137–177.

Gurtin, Morton E., Fried, Eliot & Anand, Lallit 2010 *The Mechanics and Thermodynamics of Continua*. Cambridge University Press.

Hairer, E., Nørsett, S. P. & Wanner, G. 1993 *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin.

Heath, Michael T. 2002 *Scientific Computing: An Introductory Survery*. McGraw-Hill.

Hirt, Cyril W., Amsden, Anthony A. & Cook, J. L. 1974 An arbitrary Lagrangian Eulerian computing method for all flow speeds. *J. Comput. Phys.* **14**, 227–253.

Hoover, William G. 2006 *Smooth Particle Applied Mechanics: The State of the Art*. World Scientific.

Ii, Satoshi, Sugiyama, Kazuyasu, Takagi, Shu & Matsumoto, Yoichiro 2012 A computational blood flow analysis in a capillary vessel including multiple red blood cells and platelets. *Journal of Biomechanical Science and Engineering* **7** (1), 72–83.

Jain, Suhas S. & Mani, Ali 2017 An incompressible Eulerian formulation of soft solids in fluids. In *Annual Research Briefs, Center for Turbulence Research*, pp. 347–362.

Johnson, Claes 2009 *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover, New York.

Kamrin, Ken 2008 Stochastic and deterministic models for dense granular flow. PhD thesis, MIT.

Kamrin, Ken & Nave, Jean-Christophe 2009 An Eulerian approach to the simulation of deformable solids: Application to finite-strain elasticity. http://arxiv.org/abs/0901.3799.

Kamrin, Ken, Rycroft, Chris H. & Nave, Jean-Christophe 2012 Reference map technique for finite-strain elasticity and fluid–solid interaction. *J. Mech. Phys. Solids* **60** (11), 1952–1969.

Krishnan, Sreenath, Shaqfeh, Eric S.G. & Iaccarino, Gianluca 2017 Fully resolved viscoelastic particulate simulations using unstructured grids. *J. Comput. Phys.* **338**, 313–338.

Kröner, Ekkehart 1960 Allgemeine kontinuumstheorie der versetzungen und eigenspannungen. *Arch. Ration. Mech. Anal.* **4**, 273–334.

Lax, Peter & Wendroff, Burton 1960 Systems of conservation laws. *Commun. Pure Appl. Math.* **13**, 217–237.

Lee, Erastus H. 1969 Elastic plastic deformation at finite strain. *J. Appl. Mech.* **36**, 1–6.

LeVeque, Randall J. 2002 *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press.

LeVeque, Randall J. 2007 *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM.

LIENHARD, JOHN H. 1966 *Synopsis of lift, drag, and vortex frequency data for rigid circular cylinders*, , vol. 300. Technical Extension Service, Washington State University.

LIM, SOOKKYUNG & PESKIN, CHARLES S. 2012 Fluid-mechanical interaction of flexible bacterial flagella by the immersed boundary method. *Phys. Rev. E* **85**, 036307.

LIU, CHUN & WALKINGTON, NOEL J. 2001 An Eulerian description of fluids containing visco-elastic particles. *Archive for Rational Mechanics and Analysis* **159** (3), 229–252.

LUBLINER, JACOB 2008 *Plasticity Theory*. Dover, New York.

MAITRE, EMMANUEL, MILCENT, THOMAS, COTTET, GEORGES-HENRI, RAOULT, ANNIE & USSON, YVES 2009 Applications of level set methods in computational biophysics. *Mathematical and Computer Modelling* **49** (11–12), 2161–2169, trends in Application of Mathematics to Medicine.

MCMILLEN, TYLER, WILLIAMS, T. & HOLMES, P. 2008 Nonlinear muscles, passive viscoelasticity and body taper conspire to create neuromechanical phase lags in anguilliform swimmers. *PLoS Computational Biology* **4** (8), e1000157.

MILCENT, THOMAS & MAITRE, EMMANUEL 2016 Eulerian model of immersed elastic surfaces with full membrane elasticity. *Commun. Math. Sci.* **14** (3), 857–881.

OSHER, STANLEY & SETHIAN, JAMES A. 1988 Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* **79** (1), 12–49.

OSHER, STANLEY J. & FEDKIW, RONALD P. 2003 *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, New York.

PATEL, NAMRATA K., BHALLA, AMNEET PAL SINGH & PATANKAR, NEELESH A. 2018 A new constraint-based formulation for hydrodynamically resolved computational neuromechanics of swimming animals. *J. Comput. Phys.* **375**, 684–716.

PESKIN, CHARLES S. 1972*a* Flow patterns around heart valves: A digital computer method for solving the equations of motion. PhD thesis, Albert Einstein College of Medicine.

PESKIN, CHARLES S. 1972*b* Flow patterns around heart valves: A numerical method. *J. Comput. Phys.* **10** (2), 252–271.

PESKIN, CHARLES S. 1977 Numerical analysis of blood flow in the heart. *J. Comput. Phys.* **25** (3), 220–252.

PESKIN, CHARLES S. 2002 The immersed boundary method. *Acta Numerica* **11**, 479–517.

PLOHR, BRADLEY J. & SHARP, DAVID H. 1988 A conservative Eulerian formulation of the equations for elastic flow. *Advances in Applied Mathematics* **9** (4), 481–499.

PUCKETT, ELBRIDGE GERRY, ALMGREN, ANN S., BELL, JOHN B., MARCUS, DANIEL L. & RIDER, WILLIAM J. 1997 A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.* **130** (2), 269–282.

RABCZUK, TIMON, GRACIE, ROBERT, SONG, JEONG-HOON & BELYTSCHKO, TED 2010 Immersed particle method for fluid–structure interaction. *Int. J. Numer. Meth. Eng.* **81** (1), 48–71.

RICHARDSON, LEWIS FRY 1911 IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philos. Trans. Royal Soc. A* **210** (459–470), 307–357.

RICHTER, THOMAS 2013 A fully Eulerian formulation for fluid–structure-interaction problems. *J. Comput. Phys.* **233**, 227–240.

RUGONYI, SANDRA & BATHE, KLAUS-JÜRGEN 2001 On finite element analysis of fluid flows fully coupled with structural interactions. *Comput. Model. Eng. Sci.* **2**, 195–212.

RYCROFT, CHRIS H. & GIBOU, FRÉDÉRIC 2012 Simulations of a stretching bar using a plasticity model from the shear transformation zone theory. *J. Comput. Phys.* **231** (5), 2155–2179.

RYCROFT, CHRIS H., SUI, YI & BOUCHBINDER, ERAN 2015 An Eulerian projection method for quasi-static elastoplasticity. *J. Comput. Phys.* **300**, 136–166.

SETHIAN, JAMES A. 1999 *Level Set Methods and Fast Marching Methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science*. Cambridge University Press.

SUGIYAMA, KAZUYASU, II, SATOSHI, TAKEUCHI, SHINTARO, TAKAGI, SHU & MATSUMOTO, YOICHIRO 2011 A full Eulerian finite difference approach for solving fluid–structure coupling problems. *J. Comput. Phys.* **230** (3), 596–627.

SÜLI, ENDRE & MAYERS, DAVID 2003 *An Introduction to Numerical Analysis*. Cambridge University Press.

Sulsky, Deborah, Chen, Zhen & Schreyer, Howard L. 1994 A particle method for history-dependent materials. *Comput. Method. Appl. M.* **118** (1-2), 179–196.

Sussman, Mark, Almgren, Ann S., Bell, John B., Colella, Phillip, Howell, Louis H. & Welcome, Michael L. 1999 An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.* **148** (1), 81–124.

Sussman, Mark, Smereka, Peter & Osher, Stanley 1994 A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114** (1), 146–159.

Takagi, Shu, Sugiyama, Kazuyasu, Ii, Satoshi & Matsumoto, Yoichiro 2011 A review of full Eulerian methods for fluid structure interaction problems. *J. Appl. Mech.* **79** (1), 010911.

Tannehill, John C., Anderson, Dale A. & Pletcher, Richard H. 1997 *Computational Fluid Mechanics and Heat Transfer*. Taylor and Francis.

Tornberg, Anna-Karin & Engquist, Björn 2003 Regularization techniques for numerical approximation of pdes with singularities. *J. Sci. Comput.* **19** (1), 527–552.

Tornberg, Anna-Karin & Engquist, Björn 2004 Numerical approximations of singular source terms in differential equations. *J. Comput. Phys.* **200** (2), 462–488.

Trangenstein, John A. & Colella, Phillip 1991 A higher-order Godunov method for modeling finite deformation in elastic-plastic solids. *Commun. Pure Appl. Math.* **44** (1), 41–100.

Truesdell, Clifford 1955 Hypo-elasticity. *Indiana Univ. Math. J.* **4**, 83–133.

Tytell, Eric D., Hsu, Chia-Yu, Williams, Thelma L., Cohen, Avis H. & Fauci, Lisa J. 2010 Interactions between internal forces, body stiffness, and fluid environment in a neuromechanical model of lamprey swimming. *Proc. Natl. Acad. Sci.* **107** (46), 19832–19837.

Udaykumar, Holavanahalli S., Tran, Linhbao, Belk, Davy M. & Vanden, Kirk J. 2003 An Eulerian method for computation of multimaterial impact with ENO shock-capturing and sharp interfaces. *J. Comput. Phys.* **186** (1), 136–177.

Vadala-Roth, Ben, Acharya, Shashank, Patankar, Neelesh A., Rossi, Simone & Griffith, Boyce E. 2020 Stabilization approaches for the hyperelastic immersed boundary method for problems of large-deformation incompressible elasticity. *Comput. Method. Appl. M.* **365**, 112978.

Valkov, Boris, Rycroft, Chris H. & Kamrin, Ken 2015 Eulerian method for multiphase interactions of soft solid bodies in fluids. *J. Appl. Mech.* **82** (4), 041011.

Versteeg, Henk K. & Malalasekera, Weeratunge 1995 *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Addison Wesley Longman, Ltd.

Wang, X. Sheldon, Zhang, Lucy T. & Liu, Wing Kam 2009 On computational issues of immersed finite element methods. *J. Comput. Phys.* **228** (7), 2535–2551.

Watanabe, Y., Suzuki, Setsuo, Sugihara, Masahiro & Sueoka, Yasuhiro 2002 An experimental study of paper flutter. *Journal of fluids and Structures* **16** (4), 529–542.

Wick, Thomas 2013 Fully Eulerian fluid–structure interaction for time-dependent problems. *Comput. Method. Appl. M.* **255**, 14–26.

Yu, Jiun-Der, Sakai, Shinri & Sethian, James A. 2003 A coupled level set projection method applied to ink jet simulation. *Interfaces and Free Boundaries* **5** (4), 459–482.

Yu, Jiun-Der, Sakai, Shinri & Sethian, James A. 2007 Two-phase viscoelastic jetting. *J. Comput. Phys.* **220** (2), 568–585.

Zhang, Jun, Childress, Stephen, Libchaber, Albert & Shelley, Michael 2000 Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind. *Nature* **408** (6814), 835–839.

Zhu, Luoding & Peskin, Charles S. 2002 Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. *J. Comput. Phys.* **179** (2), 452–468.

Zienkiewicz, Olek C. & Taylor, Robert L. 1967 *The Finite Element Method for Solid and Structural Mechanics*. McGraw Hill.