Boris Valkov
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA 02139
e-mail: bvalkov@alum.mit.edu

Chris H. Rycroft
School of Engineering and Applied Sciences,
Harvard University,
Cambridge, MA 02138;
Department of Mathematics,
Lawrence Berkeley Laboratory,
Berkeley, CA 94720
e-mail: chr@seas.harvard.edu

Ken Kamrin[1]
Department of Mechanical Engineering,
Massachusetts Institute of Technology,
Cambridge, MA 02139
e-mail: kkamrin@mit.edu

Eulerian Method for Multiphase Interactions of Soft Solid Bodies in Fluids

We introduce an Eulerian approach for problems involving one or more soft solids immersed in a fluid, which permits mechanical interactions between all phases. The reference map variable is exploited to simulate finite-deformation constitutive relations in the solid(s) on the same fixed grid as the fluid phase, which greatly simplifies the coupling between phases. Our coupling procedure, a key contribution in the current work, is shown to be computationally faster and more stable than an earlier approach and admits the ability to simulate both fluid–solid and solid–solid interaction between submerged bodies. The interface treatment is demonstrated with multiple examples involving a weakly compressible Navier–Stokes fluid interacting with a neo-Hookean solid, and we verify the method's convergence. The solid contact method, which exploits distance-measures already existing on the grid, is demonstrated with two examples. A new, general routine for cross-interface extrapolation is introduced and used as part of the new interfacial treatment. [DOI: 10.1115/1.4029765]

1 Introduction

The challenges of simulating fluid–structure interaction (FSI) have been approached from many directions. One set of challenges stem from domain discretization: fluid problems on their own are amenable to solution on an Eulerian computational domain [1–5] and solid deformation is natural to compute within a Lagrangian framework [6–9]. To couple solid and fluid phases in one setting, several approaches have been proposed to mitigate this separation in methodology. One approach is to treat the solid with a standard Lagrangian finite-element framework and to apply an Arbitrary-Lagrange Eulerian method in the fluid [10–12], which remeshes the fluid domain in order to remedy issues of excessive mesh deformation. Other approaches include the family of immersed methods [10,11,13], which keep an ambient stationary Eulerian grid throughout, on which fluid flow is solved, as well as a moving collection of interacting material points representing the solid structure. Here, discretized delta functions are used to pass information between the grid and the nodes.

Some advantages of a fully Eulerian method—fluid and solid both computed on an Eulerian grid—can be directly seen. Since all phases are solved by sweeping through a single fixed mesh, there are computation time advantages. Topological changes are easy to manage on a fixed grid using level sets to track interfaces [4,14]. Furthermore, multiscale and multiphysics coupling can also have advantages when done on an Eulerian grid. One specific example, which will be discussed in more detail later, is the case of multiple solids making contact immersed in a fluid. Finding contact between two solid phases on an Eulerian grid can be achieved using gridwise distance functions or simply identifying grid points which become occupied by multiple solid phases during a trial step.

To achieve these goals, one must pose an Eulerian scheme capable of solving finite-deformation solid problems. The recently proposed reference map technique (RMT) is such an Eulerian framework, based on tracking the *reference map* field [15,16]. Other approaches for solid deformation on a fixed Eulerian grid include hypoelastic implementations [17,18], which may succeed for small elastic strains but lack a thermodynamically consistent form as needed for large deformations, and methods that directly evolve the deformation gradient tensor field as the primitive kinematic grid variable [19,20,21]. In a previous paper [16], the RMT demonstrated the capability of accurately solving hyperelastic solid deformation problems on a fixed mesh—including shock propagation problems and problems with varied boundary and initial conditions—up to second-order accuracy in space and time. It also provided the first demonstrations of using the method to solve fully coupled problems of FSI. There, the FSI method hinged on a *sharp-interface representation*, extending on that of the Ghost Fluid Method for fluid–fluid interaction [22]. Sharp methods make a distinct separation between each phase down to the subgrid level. In the current work, our efforts exploit a *blurred interface*, a simpler and computationally faster implementation, involving fewer numerical extrapolations. A blurred interface method uses a thin *transition zone* where one phase converts into the other. As the grid size decreases, the corresponding transition zone decreases, and results approach that of a sharp interface method. Here, we show that the blurred interface approach has advantages over the sharp, most important of which is the ability to represent key mechanical behaviors such as submerged solid–solid contact.

To satisfy subgrid jump conditions, sharp interface methods require a large number of gridwise extrapolations of kinematic and stress fields across the interface. These produce the "ghost values" for each phase, which represent an extension of each field into the region occupied by the other phase(s). The validity of a sharp interface method is limited by the quality of continued and progressive extrapolation. When used in the FSI method of the previous work [16], accrued extrapolation error can have a destabilizing effect—shots of pressure along the interface may erroneously appear when the interface crosses through grid cell boundaries, in response to the sampling of extrapolated values when new points enter the solid domain. Adding significant solid dissipation or surface tension can penalize these artifacts, but this can alter the physicality of the simulation. We have found that

errors of this type often produce routine-ending numerical instabilities, posing a serious implementational issue. By contrast, in fluid–fluid interaction methods, this effect is less important due to the natural viscous damping within all phases.

By switching the interfacial treatment to a blurred method, we present a compromise wherein we move away from the subgrid interfacial representation of a sharp technique in order to achieve a faster, simpler, and more stable method. Existing fluid–fluid blurred techniques [23,24] do not require any extrapolation. In the case of a fluid–solid problem as shown herein, using only one extrapolation—that of the reference map (described in more detail in Sec. 2.1)—we are able to simulate FSI with a hyperelastic solid body on a single fixed grid using a single velocity field for the whole computational domain. In addition, jump conditions do not need to be explicitly applied but are implicitly met inside the transition zone. Moreover, as discussed in Sec. 7, the fact that the method uses one universal velocity field valid for all phases permits a straightforward approach for simulating multibody contact of solids submerged in a fluid, a new capability for multiphase RMT simulation.

This paper proceeds as follows. We first review the fundamental relations of each phase. The RMT methodology is then described, and we present the selected stencils for our finite differences. Next, we define the transition zone and how the fields are appropriately mixed within its band of influence. We then discuss a number of FSI test results and the convergence of the method. An important ingredient in our scheme is a new spatial extrapolation procedure that enforces certain physical and smoothness requirements in the field extrapolation. This is presented and shown to remedy deleterious extrapolation artifacts near the interface that can occur otherwise. We then describe how the routine can be appended with a solid–solid contact subroutine by taking advantage of the signed distance-measures inherent within the existing level set fields used in distinguishing the interfaces. Two examples are provided of submerged solid–solid contact.

2 Theory

2.1 Eulerian-Frame Solid and Fluid Formulations. We first provide a brief review of the Eulerian formulation for solid simulation on which the RMT is based; details can be found in Ref. [16]. The motion function χ is defined as the time-dependent map from points $\mathbf{X}$ in the reference configuration to their current position $\mathbf{x}$ in the deformed configuration; i.e., $\mathbf{x} = \chi(\mathbf{X}, t)$. We define the spatial velocity field, $\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$, the material density, $\rho = \rho(\mathbf{x}, t)$, and Cauchy stress, $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{x}, t)$. In Eulerian-frame, the conservation of mass and momentum in strong form (when deformations remain smooth) are

$$\rho_t = -\mathbf{v} \cdot \nabla \rho - \rho \nabla \cdot \mathbf{v} \tag{1}$$

and

$$\mathbf{v}_t = -(\mathbf{v} \cdot \nabla)\mathbf{v} + \frac{\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}}{\rho} \tag{2}$$

respectively, where $\mathbf{g}$ is the acceleration due to gravity, ∇ is the gradient operator in the deformed space and $\nabla \cdot$ is the spatial divergence operator. The motion function permits us to define the deformation gradient as

$$\mathbf{F}(\mathbf{X}, t) = \frac{\partial \chi(\mathbf{X}, t)}{\partial \mathbf{X}} \tag{3}$$

We define the reference map $\boldsymbol{\xi}(\mathbf{x}, t)$, an Eulerian field, as the inverse motion, so that

$$\mathbf{X} = \boldsymbol{\xi}(\mathbf{x}, t) = \chi^{-1}(\mathbf{x}, t) \tag{4}$$

Because the reference map indicates the original location of a particle, the reference map for a tracer particle never changes, giving us an advective evolution law

$$\boldsymbol{\xi}_t + \mathbf{v} \cdot \nabla \boldsymbol{\xi} = \mathbf{0} \tag{5}$$

In the case where the initial configuration is undeformed (i.e., no prestrain), the initial condition for the reference map is

$$\boldsymbol{\xi}(\mathbf{x}, t = 0) = \mathbf{x} = \mathbf{X} \tag{6}$$

Using the chain rule, it can then be shown that the deformation gradient $\mathbf{F}$ is

$$\mathbf{F}(\boldsymbol{\xi}(\mathbf{x}, t), t) = (\nabla \boldsymbol{\xi}(\mathbf{x}, t))^{-1} \tag{7}$$

and as such we can express the density in terms of the reference map and the original density, ρ_0, by

$$\rho = \frac{\rho_0}{\det \mathbf{F}} = \rho_0 \det \nabla \boldsymbol{\xi} \tag{8}$$

Consider a solid body. Because the deformation gradient tensor is describable in terms of the reference map, we are thus capable of modeling the constitutive response of large-deformation, thermodynamically compatible solid laws in Eulerian frame. For example, given the reference map field corresponding to the deformation of an isotropic hyperelastic material with strain-energy density per reference volume ψ_R, the stress is given by

$$\boldsymbol{\sigma} = 2(\det \mathbf{F})^{-1} \frac{\partial \hat{\psi}_R(\mathbf{B})}{\partial \mathbf{B}} \mathbf{B} \bigg|_{\mathbf{F} = (\nabla \boldsymbol{\xi})^{-1}, \mathbf{B} \equiv (\nabla \boldsymbol{\xi})^{-1}(\nabla \boldsymbol{\xi})^{-T}} \tag{9}$$

where $\mathbf{B}$ is the left Cauchy–Green tensor and the $\hat{}$ indicates a constitutive function. The specific model applied in this work will be a compressible neo-Hookean elastic solid model, where the strain-energy density per reference volume is

$$\psi_R = \frac{G}{2}(J^{-\frac{2}{3}} \text{tr} \mathbf{B} - 3) + \frac{\kappa}{2}(J - 1)^2 \tag{10}$$

and the Cauchy stress is

$$\boldsymbol{\sigma} = GJ^{-5/3}\mathbf{B}' + \kappa(J - 1)\mathbf{1} \tag{11}$$

where $J = \det \mathbf{F}$ and a prime indicates the deviatoric part of a tensor. Equations (2), (5), and (7)–(9) are a closed Eulerian system for computing solid deformations. As described in Ref. [16], this system of equations can be recast in conservative form and implemented numerically in a discrete conservation scheme should nonsmooth solutions be expected.

Considering a fluid, the same equations of mass and momentum balance are valid but the constitutive law for the stress is different and no reference map is needed. In this work, we choose a weakly compressible viscous flow relation for fluid stress

$$\boldsymbol{\sigma} = \eta \frac{\nabla \mathbf{v} + (\nabla \mathbf{v})^T}{2} - \lambda \left(\frac{\rho}{\rho_0} - 1\right)\mathbf{1} \tag{12}$$

where η is the viscosity and λ is the compressibility modulus.

In the continuum limit, Eqs. (1) and (8) give identical results. For later purposes of accurate discretization, it is preferable to implement Eq. (1) when computing fluid density, since Eq. (8) would require keeping a reference map field in a fluid, which would quickly lose accuracy due to the levels of distortion and mixing in the fluid. However, Eq. (8) is preferable in a solid as it ensures consistency between the deformation and the density.

Since we will soon consider coupled fluid–solid problems, it is helpful to distinguish between stresses and densities obtained from the different formulae. The solid stress and density have an "s" superscript, so that

$$\boldsymbol{\sigma}^s = GJ^{-5/3}\mathbf{B}' + \kappa(J-1)\boldsymbol{I}, \qquad \rho^s = \rho_0^s \det \nabla\boldsymbol{\xi} \qquad (13)$$

and the fluid stress and density have an "f" superscript, so that

$$\boldsymbol{\sigma}^f = \eta\frac{\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathrm{T}}}{2} - \lambda\left(\frac{\rho^f}{\rho_0^f} - 1\right)\mathbf{1},$$
$$\rho_t^f = -\mathbf{v}\cdot\nabla\rho^f - \rho^f\nabla\cdot\mathbf{v} \qquad (14)$$

2.2 The Computational Domain. The computations are carried out on a two-dimensional $m \times n$ grid of square cells with side length h, using plane strain conditions. The interface between the fluid and solid phases is described using the level set method [4,25], whereby an auxiliary Eulerian function $\phi(\mathbf{x},t)$ is introduced such that the interface is given by the zero contour, $\phi = 0$. At the start of each simulation step, the level set function is initialized to be the signed distance to the interface, using the convention that $\phi < 0$ inside the solid region and $\phi > 0$ in the fluid region. While there are a number of different numerical approaches for the level set method, we make use of the specific implementation described by Rycroft and Gibou [18]. The implementation contains a procedure for reinitializing the level set function so that it is a signed distance function to the zero contour, using a combination of the modified Newton–Rapshon iteration of Chopp [26] to update values of ϕ adjacent to the interface, and a second-order fast marching method [4] to update values further away.

Because we develop a blurred interface technique, we introduce a transition zone corresponding to the region where $|\phi| < w_{\mathrm{T}}$, where w_{T} is a constant (Fig. 1). As described in detail later, the material response in the transition zone is modeled as a mixture of fluid and solid phases. The region $0 < \phi < w_{\mathrm{T}}$ is more fluid than solid, and the region $w_{\mathrm{T}} < \phi < 0$ is more solid than fluid. For a positive constant w_E such that $w_E > w_{\mathrm{T}}$, we define the *extended solid domain* to be the region $\phi < w_E$, which corresponds to a region with any fraction of solid response together with a thin band of points adjacent to the transition zone. For computational efficiency, our level set method implementation only stores and updates the values of ϕ in a *narrow band* of grid points within a distance $\pm w_E$ of the interface.

Figure 1 shows the discretization of the simulation fields. Based on considerations of the finite-difference stencils presented later, a

staggered field arrangement is used. Some fields are held at cell corners, which are indexed using i, j for $i = 0,\dots, m$ and $j = 0,\dots, n$. Some fields are held at cell centers, which are indexed using $[i, j]$ for $i = 0,\dots, m-1$ and $j = 0,\dots, n-1$. The level set function ϕ, velocity field $\mathbf{v}$, fluid density ρ^f, and combined density ρ are stored globally at cell corners in both the fluid and solid domains. The fluid stress is stored at globally at cell centers. In addition, in the extended solid domain, the reference map field $\boldsymbol{\xi}$ is stored at cell corners, and the solid stress $\boldsymbol{\sigma}^s$ and density ρ^s are stored at cell centers.

3 Outline of the Method

We now describe the progression of the blurred interface method. Starting at a time step n, the algorithm builds the kinematic fields for the next step $n + 1$. The stress fields are calculated at every time step as a midstep calculation. The algorithm below summarizes the major steps of the routine. Details are provided thereafter.


Algorithm 1   Outline of the blurred interface method for FSI.

*Given:* $\mathbf{v}^n$, $\boldsymbol{\xi}^n$, ρ^{fn}, and ϕ^n
Compute: $\mathbf{v}^{n+1}$, $\boldsymbol{\xi}^{n+1}$, $\rho^{f(n+1)}$, and ϕ^{n+1}

(1) Compute the reference map gradient (Eq. (15)) and use it to calculate the solid stress $\boldsymbol{\sigma}^{sn}$ in the extended solid domain (Sec. 3.1).
(2) Compute the velocity gradient and use it with ρ^{fn} to compute the fluid stress $\boldsymbol{\sigma}^{fn}$ in the entire domain (Eq. (12)).
(3) Apply mixing rule to fluid/solid stress and fluid/solid density (Sec. 3.2) to obtain the actual stress field $\boldsymbol{\sigma}^n$ and density ρ^n in the entire domain.
(4) Calculate the update to reference map (Eq. (20)) in the region $\phi^n < 0$.
(5) Move the level set field to ϕ^{n+1} (Sec. 6).
(6) Calculate the fluid density update (Eq. (22)) on the entire domain.
(7) Calculate the update to the velocity field (Eq. (21)).
(8) Apply the previously computed updates to obtain $\boldsymbol{\xi}^{n+1}$, $\mathbf{v}^{n+1}$, and $\rho^{f(n+1)}$.
(9) Set boundary conditions and kinematic constraints.
(10) Extrapolate reference map into the portion of the extended solid domain where $\phi > 0$ to obtain $\boldsymbol{\xi}^{n+1}$ in that region (Sec. 6).

3.1 Finite-Difference Stencils. We employ a tight stencil to calculate the divergence of stress and the (constitutive input) gradients of $\boldsymbol{\xi}$ and $\mathbf{v}$ as in Fig. 2, with more details on our stencil selection to be discussed later. The discretized gradient of the reference map is given by
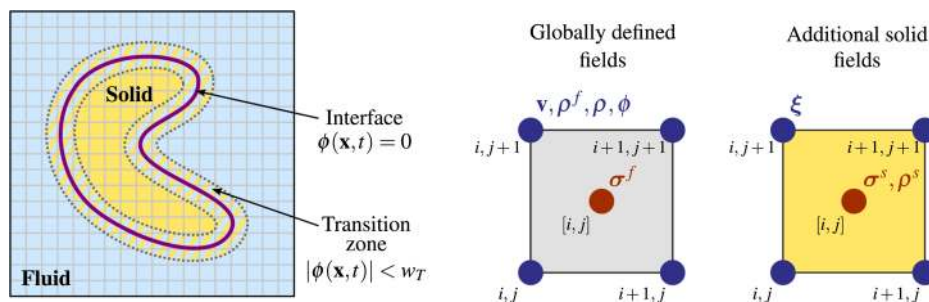
Fig. 1 (Left) Overview of the simulation in which fluid and solid phases are simulated on a fixed Eulerian grid. The phases are determined by using a level set field $\phi(\mathbf{x}, t)$, such that $\phi < 0$ in the solid, $\phi > 0$ in the fluid, and $\phi = 0$ at the interface. In the numerical method, the constitutive response between fluid and solid is blurred across a small transition zone, $|\phi| < w_{\mathrm{T}}$. (Right) Arrangement of the simulation fields on each grid cell. The level set function ϕ, fluid density ρ^f, fluid stress σ^f, combined density ρ, and velocity v are stored globally on all grid cells. In addition, the reference map field ξ, solid stress σ^s, and ρ^s are stored in the solid and slightly beyond (see text). As shown in the figure, some fields are stored at cell corners (blue) indexed by i, j, and some are stored at cell centers (red) indexed by $[i, j]$.
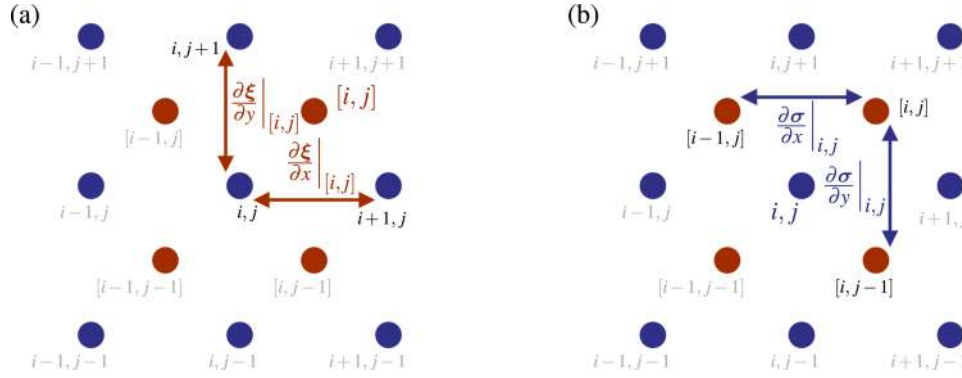
Fig. 2 Stencils for (a) the reference map gradient at [i, j] per Eq. (15) and (b) the divergence of stress at i, j per Eq. (21). The cell-cornered grid is shown as blue circles and the cell-centered grid is shown as red circles. The indexes of grid points that are used in the stencils are shown in black, while those that are not used are shown in gray.

$$\left.\frac{\partial \boldsymbol{\xi}}{\partial x}\right|_{[i,j]}^n = \frac{\boldsymbol{\xi}_{i+1,j}^n - \boldsymbol{\xi}_{i,j}^n}{h}, \qquad \left.\frac{\partial \boldsymbol{\xi}}{\partial y}\right|_{[i,j]}^n = \frac{\boldsymbol{\xi}_{i,j+1}^n - \boldsymbol{\xi}_{i,j}^n}{h} \qquad (15)$$

The discretization of $\nabla \mathbf{v}$ is identical to Eq. (15). The discretization of divergence of stress, $\nabla \cdot \boldsymbol{\sigma}$, uses the following stencils for the stress derivatives:

$$\left.\frac{\partial \boldsymbol{\sigma}}{\partial x}\right|_{i,j}^n = \frac{\boldsymbol{\sigma}_{[i,j]}^n - \boldsymbol{\sigma}_{[i-1,j]}^n}{h}, \qquad \left.\frac{\partial \boldsymbol{\sigma}}{\partial y}\right|_{i,j}^n = \frac{\boldsymbol{\sigma}_{[i,j]}^n - \boldsymbol{\sigma}_{[i,j-1]}^n}{h} \qquad (16)$$

In the extended solid domain, $\boldsymbol{\sigma}^s$ and ρ^s are computed directly from the above calculated reference map gradient as applied to Eq. (13). For example,

$$\mathbf{B}_{[i,j]}^n = (\nabla \boldsymbol{\xi}_{[i,j]}^n)^{-1} (\nabla \boldsymbol{\xi}_{[i,j]}^n)^{-T} \qquad (17)$$

and $J_{[i,j]}^n = (\det \nabla \boldsymbol{\xi}_{[i,j]}^n)^{-1}$ would be applied to create $\boldsymbol{\sigma}_{[i,j]}^{sn}$ and $\rho_{[i,j]}^{sn}$. When put together, the divergence of the solid stress at i, j is calculated from nearby values of $\boldsymbol{\xi}$ at i, j, at $i-1, j$, at $i+1, j$, at $i, j-1$, at $i, j+1$, at $i-1, j+1$, and at $i+1, j-1$.

In computing fluid stress, the nonviscous fluid pressure is calculated as[2]

$$p_{[i,j]}^{fn} = \lambda \left(\frac{\rho_{i,j}^{fn}}{\rho_0^f} - 1 \right) \qquad (18)$$

The fluid stress tensor is then given by

$$\boldsymbol{\sigma}_{[i,j]}^{fn} = \eta \frac{\nabla \mathbf{v}_{[i,j]}^n + (\nabla \mathbf{v})_{[i,j]}^{nT}}{2} - p_{[i,j]}^{fn} \mathbf{1} \qquad (19)$$

where the discretization of $\nabla \mathbf{v}$ uses the same stencil as in Eq. (15).

Now we discuss the stencil for the fields that are updated each step. The reference map is updated using Eq. (5), which is discretized to

$$\boldsymbol{\xi}_{i,j}^{n+1} = \boldsymbol{\xi}_{i,j}^n - \Delta t [(\mathbf{v} \cdot \nabla) \boldsymbol{\xi}]_{i,j}^n \qquad (20)$$

where the advection term $(\mathbf{v} \cdot \nabla) \boldsymbol{\xi}$ is calculated using a second-order, upwinded essentially nonoscillatory (ENO) discretization [18,27] for stability. The update to velocity is given by

---

[2]This definition is such that, under Eq. (16), the stencil for divergence of $p^{fn} \mathbf{1}$ at i, j is given by one-sided differences at i, j.

$$\mathbf{v}_{i,j}^{n+1} = \mathbf{v}_{i,j}^n + \Delta t \left(-[(\mathbf{v} \cdot \nabla) \mathbf{v}]_{i,j}^n + \frac{1}{\rho_{i,j}} \left((\nabla \cdot \boldsymbol{\sigma}^n)_{i,j} + \mathbf{f}_{i,j} + \rho_{i,j} \mathbf{g} \right) \right) \qquad (21)$$

where the divergence of stress is discretized as in Eq. (16) and the advection term uses the ENO discretization. Unless otherwise stated, we choose the gravity $\mathbf{g}$ to be zero in the results that follow. The above equation permits an additional body force $\mathbf{f}_{i,j}$ which we call upon in Sec. 7. A global damping term $\eta_a (\nabla^2 v)_{i,j}$ with artificial viscosity η_a is also permitted as part of $\mathbf{f}_{i,j}$, but we ensure η_a is small in comparison to η so as to minimally affect the physics. A standard second-order, five-point stencil is used in computing this Laplacian. The fluid density is updated using

$$\begin{aligned} \rho_{i,j}^{f(n+1)} &= \rho_{i,j}^{fn} + \Delta t \left(-\left[(\mathbf{v} \cdot \nabla) \rho^f \right]_{i,j}^n - \rho_{i,j}^{fn} (\nabla \cdot \mathbf{v})_{i,j}^n \right), \\ &= \rho_{i,j}^{fn} + \Delta t \left(-\left[(\mathbf{v} \cdot \nabla) \rho^f \right]_{i,j}^n - \rho_{i,j}^{fn} \left(\frac{\mathbf{v}_{i+1,j}^n - \mathbf{v}_{i,j}^n}{h} + \frac{\mathbf{v}_{i,j+1}^n - \mathbf{v}_{i,j}^n}{h} \right) \right) \end{aligned} \qquad (22)$$

where the advection term uses the ENO discretization, and the discretization of the fluid divergence is chosen in a similar manner to Eq. (15).

3.2 Mixing Quantities in the Transition Zone. We construct a transition field centered about the interface, which determines the fraction of solidlike behavior (i.e., density and stress) a point experiences, with the rest being attributed to fluid. This field defines how one phase cross-fades into the other through the mixing rules defined below. We base our transition field on the smoothed Heaviside function

$$H_s(x) = \begin{cases} 0 & \text{if } x \leq -w_T, \\ \frac{1}{2}\left(1 + \frac{x}{w_T} + \frac{1}{\pi}\sin\frac{\pi x}{w_T}\right) & \text{if } |x| < w_T, \\ 1 & \text{if } x \geq w_T \end{cases} \qquad (23)$$

where w_T is the transition zone width that was introduced previously. The choice of this function is not unique, although the above has the benefit of smoothly transitioning from exactly zero to exactly one over a finite interval, and it has been used in other blurred-interface methods [28,29]. We use the level set function as the input to the H_s, since it measures the distance from the interface. We define cell-cornered and cell-centered transition fields as

$$\Omega_{[i,j]} = H_s(\phi_{[i,j]}), \qquad \Omega_{i,j} = H_s(\phi_{i,j}) \qquad (24)$$

respectively, where $\phi_{[i,j]}$ is calculated by averaging the four values on the grid cell corners.

We now have a straightforward way of defining the mixture of quantities everywhere in our computational domain. For instance, the stress components become a mixture of the stress obtained from the solid constitutive law and that of the fluid constitutive law weighted, respectively, by Ω and $1 - \Omega$, so that

$$\boldsymbol{\sigma}_{[i,j]} = \Omega_{[i,j]}\boldsymbol{\sigma}^s_{[i,j]} + (1 - \Omega_{[i,j]})\boldsymbol{\sigma}^f_{[i,j]} \qquad (25)$$

The global density field is defined similarly as

$$\rho_{i,j} = \Omega_{i,j}\rho^f_{i,j} + (1 - \Omega_{[i,j]})\frac{\rho^s_{[i,j]} + \rho^s_{[i+1,j]} + \rho^s_{[i,j+1]} + \rho^s_{[i+1,j+1]}}{4}$$

$$(26)$$

where the above uses the average of four values to interpolate ρ^s onto the cell-cornered grid.

The above mixing procedure models the *no slip* condition between fluid and solid. Other interfacial conditions are possible within this framework, though we have yet to perform extensive tests on them. For example, perfect slip may be achievable by applying a shear traction elimination step after the above mixing is applied, which modifies the stress in the transition zone by smoothly decreasing the shear traction resolved onto the level set contours such that shear traction vanishes at $\phi = 0$.

3.3 Reference Map Extrapolation. In order to define $\boldsymbol{\xi}$ and consequently $\boldsymbol{\sigma}^s$ in the portion of the extended solid domain where $\phi > 0$, great care must be taken. This zone represents a region whose behavior is more fluid than solid, and, thus, is liable to undergo extensive shearing and nonlinear deformation. This is problematic because large nonlinear deformations reduce the accuracy of computation of $\nabla\boldsymbol{\xi}$ necessary to calculate $\boldsymbol{\sigma}^s$, and, moreover, the solid stress can become unboundedly large as deformation builds up in this zone, which negates the purpose of the gradual switch-over to a fluid-dominated stress response in Eq. (25). For these reasons, we avoid advecting the reference map per Eq. (20) in this region and instead obtain values for it through extrapolation from the nearby grid points where $\phi < 0$. This constitutes the only field extrapolation in our current method, and it is recalculated every time step. Note that in the velocity update step, only those values of stress within the transition zone (i.e., very close to the $\phi = 0$ contour) influence the outcome, which supports our usage of near-field extrapolation as an appropriate remedy. Our algorithm for consistent extrapolation is left for a detailed discussion in Sec. 6.

4 Results

The above comprises a general fluid–solid interaction routine for simulating solids with finite-strain constitutive relations coupled to a fluid phase that obeys the compressible Navier–Stokes equations. We now verify the method and its accuracy using several test geometries, many of which (those of Sec. 4.1) under the previous sharp-interface approach [16] exhibited interfacial errors of the type previously described, even with the inclusion of surface tension and significant solid viscosity. Our current blurred-interface routine has also reduced the number of extrapolation steps significantly—20 scalar fields are extrapolated per time step in the sharp-interface approach, but now only the two components of $\boldsymbol{\xi}$ are now extrapolated. Unlike the sharp approach, we do not require a specific routine to ensure interfacial jump conditions as they are implicitly satisfied in the transition zone. These comments are not to suggest that the numerical interfacial errors of the sharp approach are irreparable in that framework; it is possible they could be alleviated in part through integration of our improved extrapolation algorithm of Sec. 6 among other adjustments, but we reserve this topic for future work.

Throughout this paper, we make use of dimensionless simulation units. In all tests shown, unless otherwise stated, the following input material parameters are used. The initial fluid density is $\rho^f_0 = 1.0$, the viscosity is $\eta = 0.12$, and the compressibility is $\lambda = 60.0$. The solid uses $G = 10$ and $\kappa = 50$. The artificial viscosity is $\eta_a = 0.012$. These parameters match our previous work [16], though it should be noted the previous work also included a significant separately added solid viscosity. The predominant Courant–Friedrichs–Lewy (CFL) criterion is due to the fluid viscosity, which is proportional to $h^2\rho^f/2\eta$. The other criteria that need to be considered are the one due to compressibility of the solid $\sim h\sqrt{\rho^s/\kappa}$, the one due to compressibility of the fluid $\sim h\sqrt{\rho^f/\lambda}$, and the convective CFL $\sim h/\max|\mathbf{v}|$. For all of our results, we use a time step of $\Delta t = 0.05h^2\rho^f/\eta$, which satisfies the dominant CFL condition for the fluid viscosity and also satisfies the additional CFL conditions for all of the grids and parameters that are used. Throughout, we use a transition zone width of $w_T = 3h$ and we use $w_E = 6h$. The simulations are written in C++ and use the OpenMP library to multi-thread many of the operations that scan over the entire grid of points.

For all of the simulations presented here, domain boundary conditions are applied by fixing the values of $\mathbf{v}$ and ρ^f on the edge of the cell-cornered grid, where $i = 0$, $j = 0$, $i = m$, or $j = n$. We consider two types of boundary conditions: Dirichlet conditions where the boundary field values are fixed, and free boundary conditions where the boundary field values are linearly extrapolated from the two adjacent layers of interior points. For example, to implement a free boundary condition on an arbitrary field $f_{i,j}$ at the top boundary, we set $f_{i,n} = 2f_{i,n-1} - f_{i,n-2}$ for $i = 0, 1,\ldots, m$. Note that a free condition on the fluid velocity component tangent to a boundary gives the perfect slip condition.

4.1 Incoming Fluid Deforms Anchored Rubber Rod. The first example we consider involves a bulky, highly deformable neo-Hookean rod. The rod is initially vertical and anchored at its top end. It is placed in a horizontal fluid flow, which causes it to deform significantly. The domain covers $-2 \le x \le 2$ and $-2 \le y \le 2$ using a square grid of size 240×240. The bar initially covers the rectangle $-1.4 < x < 0.6$ and $|y| < 1.1$ and has semicircular end caps. The anchored region is a circle with center $(x, y) = (-1, 1.1)$ and radius r_a. During the simulation, the reference map in the anchored region is enforced to be constant and the velocity is enforced to be zero. We have simulated two different anchor sizes $r_a = 0.25$ (Fig. 3) and $r_a = 0.15$ (Fig. 4). The fluid inflow and outflow is controlled by applying a constant horizontal velocity of $(u, v) = (0.24, 0)$ on the left and right sides of the domain window. Perfect slip boundary conditions are used on the top and bottom sides, where $v = 0$, and u and ρ^f are free. In each simulation, the fluid flow deforms the rod inducing large local stretches in the solid, with stretch ratios exceeding two in parts of the bar. In the case of the smaller anchor, the bar is less able to resist the incoming fluid flow and swivels out of the way to a greater extent as expected. The simulations each reach a steady state by $t \approx 20$ where the rod remains in a static, bent configuration with steady fluid flow surrounding it. On an Apple MacBook Pro (Mid 2014) system with a quad-core 2.8 GHz Intel i7 processor, the simulation using $r_a = 0.25$ takes 997 s using 216,090 time steps using a single thread. If two, three, or four threads are used, the simulation takes 773 s, 705 s, and 635 s, respectively. Because this simulation only uses a small grid, the speedup from multi-threading is only modest, since the overhead from creating threads is comparable the computational work done each time step. However, for some of the larger simulation grids considered later, multithreading becomes significantly more advantageous.

We also used this configuration with $r_a = 0.25$ to test the speed of the simulation method against the previous sharp interface method [16]. For a benchmark test simulating over the interval $0 \le t \le 2.5$ with 3,200 time steps using a single thread, the sharp
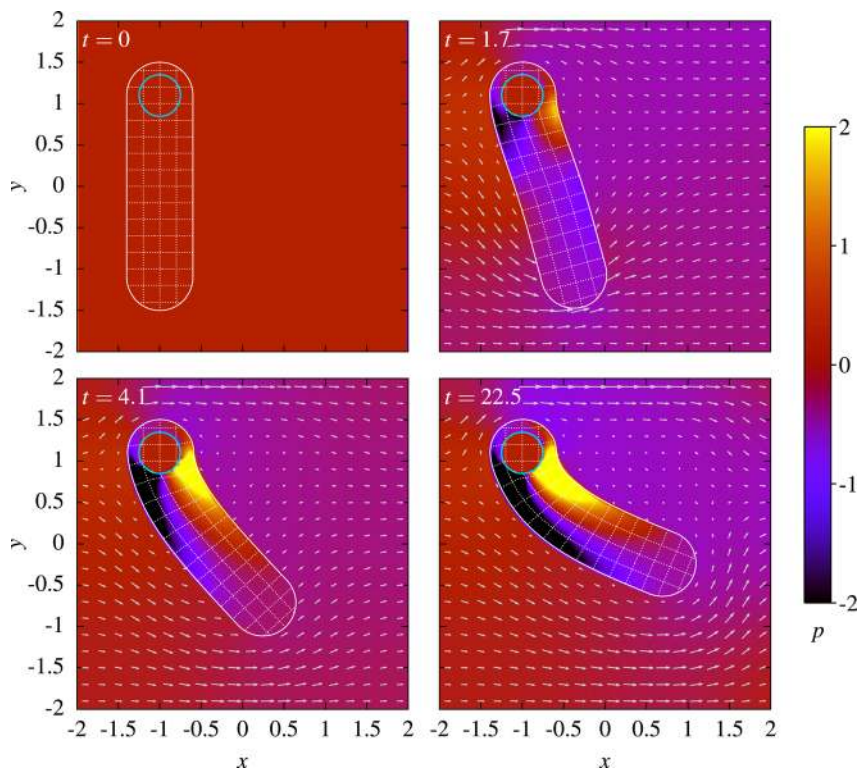
Fig. 3 Four snapshots of the in-plane pressure $p \equiv (\sigma_{xx} + \sigma_{yy})/2$ (colors) and fluid velocity (arrows) for the simulation of an anchored flexible rod deformed to large strain by incoming fluid flow. The solid white line shows the boundary of the rod, given by the zero contour of the level set function $\phi(\mathbf{x}, t)$. The thin dashed white lines are the contours of the reference map $\xi(\mathbf{x}, t)$. The cyan circle with radius of 0.25 centered at $(-1, 1.1)$ shows the anchored region where the reference map is fixed and the velocity is zero.

interface method took 94.01 s, while the new method took 53.97 s, corresponding to a speedup factor of 1.74. These two simulations were not perfectly comparable, since the memory organization and boundary implementation are slightly different. However, the results demonstrate that the present method is significantly faster, due to the simplification of the finite-difference stencils, a reduction in the number of simulation fields required, and fewer boundary extrapolations.
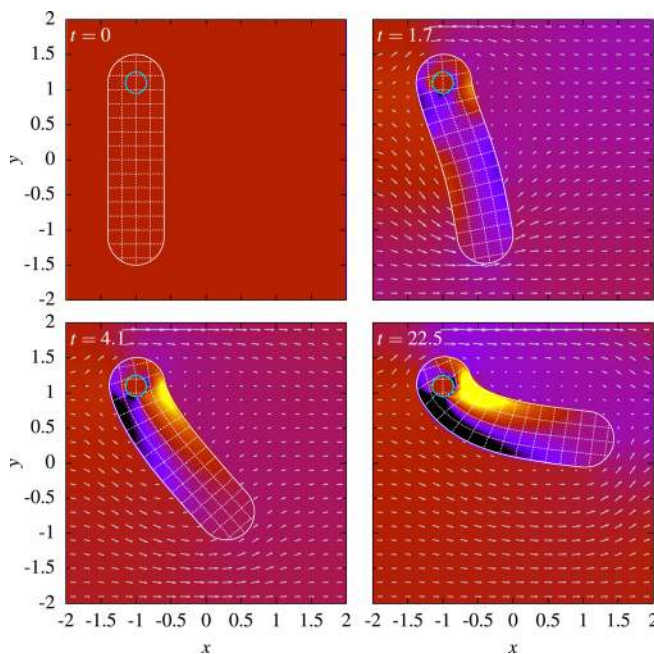
Fig. 4 Four snapshots of the in-plane pressure field (colors) and fluid velocity (arrows) for the simulation of an anchored flexible rod, using a smaller anchored region than in Fig. 3. The color gradient is the same as in Fig. 3. The solid white line shows the boundary of the rod, given by the zero contour of the level set function $\phi(\mathbf{x}, t)$. The thin dashed white lines are the contours of the reference map $\xi(\mathbf{x}, t)$. The cyan circle with radius of 0.15 centered at $(-1, 1.1)$ shows the anchored region where the reference map is fixed and the velocity is zero.
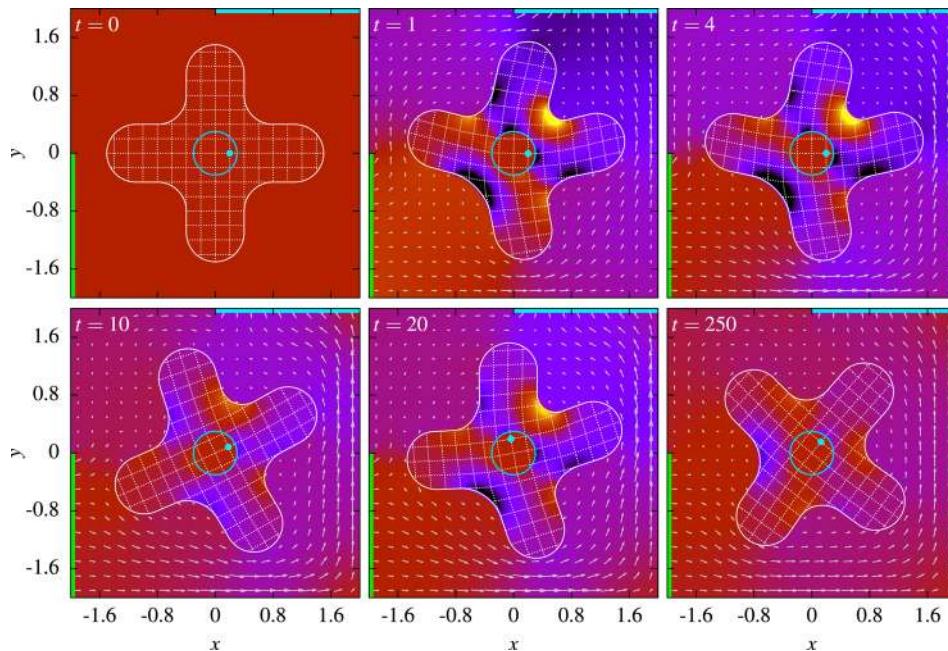
Fig. 5 Six snapshots of the in-plane pressure field (colors) and fluid velocity (arrows) for the simulation of a flexible rotor. The color gradient is the same as in Fig. 3. The green rectangle on the left edge of the domain shows the region where fluid is added, and the cyan rectangle on the upper edge of the domain shows where fluid is removed. The solid white line shows the boundary of the solid, given by the zero contour of the level set function. The thin dashed white lines are the contours of the reference map ξ. The cyan circle with radius of 0.3 centered on the origin shows the pivot, and the small circular dot shows how the pivot has rotated. By $t = 250$, the rotor has undergone roughly 5.5 complete rotations.

4.2 Fluid Deforms and Spins a Four-Blade Rotor. The second test case we consider involves a rotor of neo-Hookean material anchored around a frictionless pivot at its center. It is deformed and ultimately set into steady rotation by incoming fluid. Each blade of the rotor comprised a rectangle of length 1.1 with a semi-circular end cap. The join between each pair of blades is smoothed with a quarter-circle with radius of 0.4. The fluid boundary condition consists of a prescribed fluid inflow of $(u, v) = (0.2, 0)$ on the bottom half of the left edge of the computational domain and an outflow of $(u, v) = (0, 0.2)$ in the top right half edge of the domain; at all other boundaries, perfect slip boundary conditions are used. The rotor and fluid have the same material parameters as Sec. 4. This particular geometry, when implemented under the previous sharp-interface method, displayed many erroneous pressure shots about

the interface, as the particular shape involved has much curvature variation, as can be seen from Fig. 5, this artifact is nonexistent in the current scheme. By $t = 14$, the rotor motion and fluid flow approach a steady repeating cycle, with each cycle corresponding to one quarter-turn of the rotor. By the final snapshot in Fig. 5 at $t = 250$, the rotor has undergone approximately 5.5 complete rotations without any interfacial perturbations. Figure 6 confirms that the rotor enters a steady repeating cycle, by showing the rotation angle θ_p and angular velocity ω_p of the pivot as a function of time. Figure 6(c) shows a plot of θ_p against ω_p, where after an initial transient period as the rotor begins to turn, a steady relationship between ω_p and θ_p develops, with the angular velocity being slightly higher during periods when a blade of the rotor is directly in front of the region of fluid inflow.
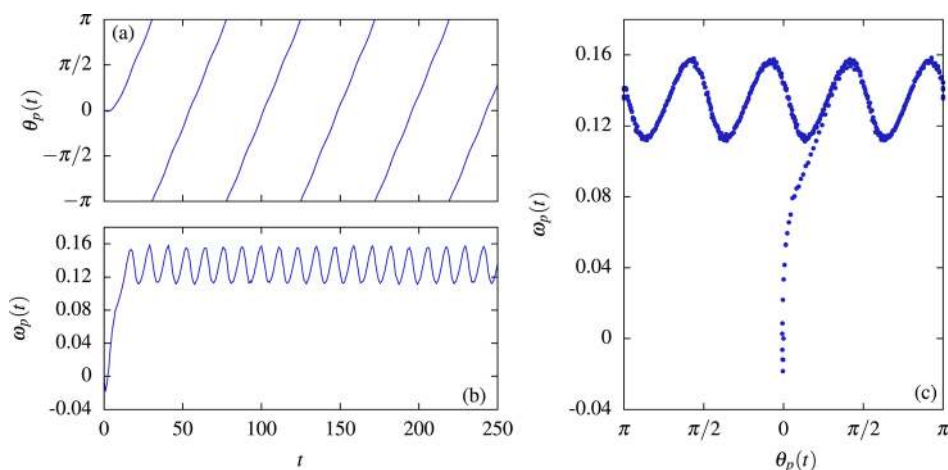
Fig. 6 Plot of (a) rotation angle θ_p and (b) angular velocity ω_p of the pivot as a function of time t for the flexible rotor simulation shown in Fig. 5. (c) Plot of the relationship between θ_p and ω_p.
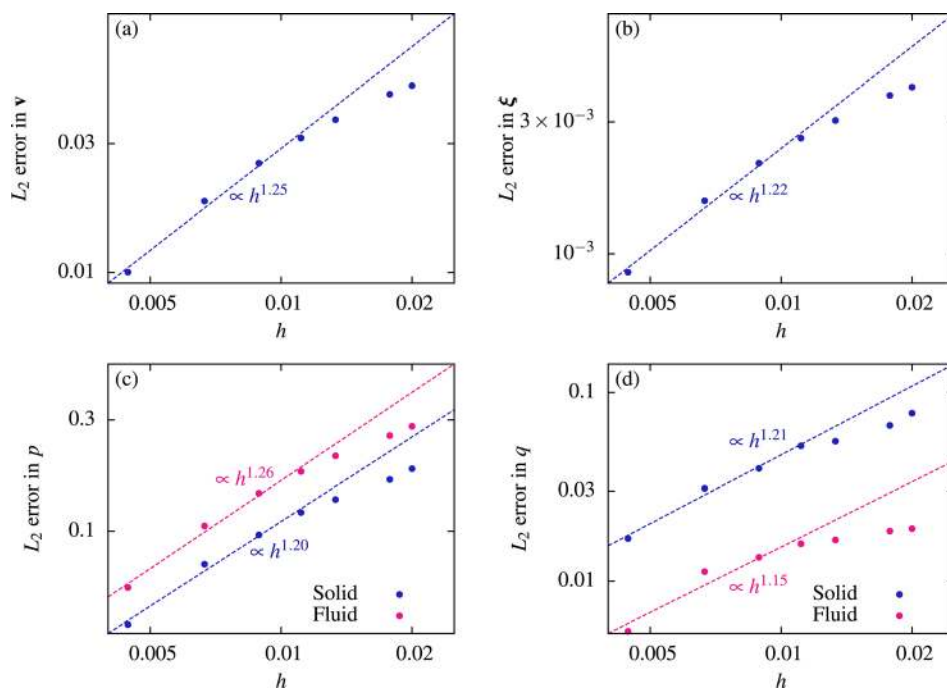
Fig. 7 Log–log plots of the L_2 error for the flexible rotor simulation at $t = 0.4$ as a function of grid spacing h for (*a*) velocity v, (*b*) the reference map ξ, (*c*) *p* as calculated from the mixed Cauchy stress σ, and (*d*) the effective shear stress *q*. The circles show the calculated error for simulations of $n \times n$ resolution where $n = 900, 600, 450, 360, 300, 225, 200$ and are compared against a reference simulation with $n = 1800$. The dashed lines show linear fits only using data where $h < 0.0125$.

5 Convergence

We measure the convergence of our simulation method by comparing solutions of the flexible rotor simulation of Sec. 4.2 as the grid size is reduced, against the solution in a highly refined case. This reference solution plays the role that an exact solution would normally play in such an analysis, but which we do not have given the complexity of the FSI problems that we wish to verify. We have carried out simulations on multiple $n \times n$ grids, for $n = 1800$, 900, 600, 450, 360, 300, 225, and 200. All simulations use $w_T = 3h$, so that the width of transition zone via Eq. (24) is fixed in terms of the number of grid spacings. We choose the rotor simulation as a test case and simulate to $t = 0.4$, which is long enough for there to be some interaction between the fluid and the solid. At that time, we then compare the fields of each simulation against the corresponding solution of the 1800×1800 grid, using the discrete L^2 norm. For an arbitrary cell-cornered field f, this is defined as

$$||f^{\text{numerical}} - f^{\text{exact}}||_2 = \sqrt{\frac{1}{4^2} \sum_{i=0}^{n} \sum_{j=0}^{n} T_i T_j |f_{i,j}^{\text{numerical}} - f_{i,j}^{\text{exact}}|^2 h^2}$$

(27)

where $T_k = 1/2$ if $k = 0$ or $k = n$ and $T_k = 1$ otherwise, so that the sum calculates the trapezoidal rule. By interpreting the $|\cdot|^2$ operator appropriately, Eq. (27) can be applied to scalars, vectors, or tensors. Since the resolution of each smaller grid is an exact multiple of the reference grid, each grid point (i, j) on a smaller grid exactly coincides with a grid point (i', j') on the 1800×1800 grid, and hence we take $f_{i,j}^{\text{exact}} = f_{i',j'}$. For an arbitrary cell-centered field f, the same definition is used but replacing i, j with $[i, j]$. In this case, the smaller grid points may not coincide with the 1800×1800 grid points, in which case we calculate f^{exact} using bilinear interpolation. Figure 7 displays the convergence properties of four key fields. The velocity convergence plot in Fig. 7(*a*) compares the velocity fields in the entire domain to that of the

reference solution. For the convergence plots of ξ in Fig. 7(*b*), the sum in Eq. (27) is only evaluated at grid points where $\phi < 0$ in both the smaller grid and the reference grid. The figure indicates that all fields are converging with at least a linear rate, as we expect for the chosen stencils.

Figures 7(*c*) and 7(*d*) show the convergence of the in-plane pressure p and the in-plane effective shear stress $q = |\sigma_1 - \sigma_3|/2$ where σ_1 and σ_3 are, respectively, the maximum and minimum principal stresses. The norms are split into contributions from solid and fluid phases, determined by the sign of ϕ on the coarse grid. In both phases, the stress converges at a similar rate to the velocity and reference map. For the pressure, the fluid phase has the larger errors, due to the high value of λ. However, for the effective shear stress, the solid phase has larger errors, suggesting that errors in solid shear stress are more significant than errors in fluid viscous stress.

It is interesting to zoom into a region containing the interface and compare simulations with a coarse grid to a more refined one. Figure 8 shows a comparison of the pressure fields at $t = 4$ in the upper right concave part of the rotor with the 200×200 grid and the 600×600 grid. As the grid size decreases, and likewise the transition zone width, the interfacial behavior approaches that of a sharp interface; to wit, the pressure field develops a rapid variation across the interface that approaches a discontinuity. Analytically, the stress component representing tension in the direction tangent to the interface need not be continuous across the interface, which is why the pressure field should adopt this feature in the sharp limit.

6 Improved Extrapolation Procedure

With basic demonstrations in hand, we now return to discuss the details of our reference map extrapolation step. Physically consistent extrapolation of ξ is key to a proper interface representation, as we shall show. This section highlights two new subroutines we have developed for this purpose.

The extrapolation algorithm we use is based on that of Aslam [30], which takes advantage of the regularized level set ϕ already
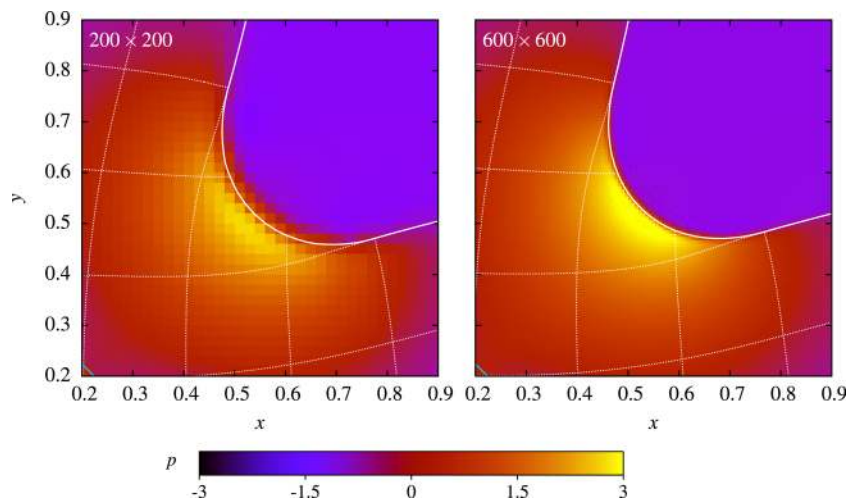
Fig. 8 Comparison of the in-plane pressure field in two simulations of the flexible rotor using different grid resolutions. In the left panel, each square that is visible corresponds to a grid cell of width *h*, colored according to the pressure stored at the cell-centered grid point within the cell. The solid white line shows the boundary of the rod, given by the zero contour of the level set function. The thin dashed white lines are the contours of the reference map ξ. Part of the pivot, shown in cyan, it just visible in the bottom left corner.

stored on the grid. This routine has the benefit of providing linear extrapolation outside an arbitrary-shaped domain, by extrapolating fields in the outward-normal direction to the interface—the outward normal is obtained easily via $\hat{\mathbf{n}} = \nabla\phi$. An immediate requirement of such a routine is that the gridwise field being extrapolated be smooth near the edge of its known domain. Numerical oscillation error in the known domain, hence, invalidates the extrapolation procedure. This motivates our use of one-sided difference stencils in Sec. 3.1, which do not support odd–even oscillation error. Future work will explore higher-order, nonoscillatory stencils to achieve this same aim akin to the usage of staggered grids in computational fluid dynamics.

While its speed and geometric generality are benefits of the Aslam method, we and others [18] have noticed that the hyperbolic nature of the extrapolation routine can cause extrapolated fields to develop striations—loss of smoothness in the direction parallel to the interface—even when the known domain data are smooth. If uncorrected, striations in the extrapolated values of ξ cause oscillations in the corresponding $\boldsymbol{\sigma}^s$, which induce an artificial wrinkling motion that becomes apparent in the $\phi = 0$ level set (see Fig. 9). This phenomenon destabilizes the routine if the wrinkle curvature grows to a level that competes with the grid spacing.

Another important consideration is to ensure agreement between ξ and the level set field ϕ with regard to where the interface lies. We remind that the reference map indicates where material at a current point originated from. If we define ψ by $\psi(\mathbf{x}, t) = \phi(\boldsymbol{\xi}(\mathbf{x}, t), t = 0)$ then a consistency constraint is

$$\psi(\mathbf{x}, t) = 0 \Leftrightarrow \phi(\mathbf{x}, t) = 0 \qquad (28)$$
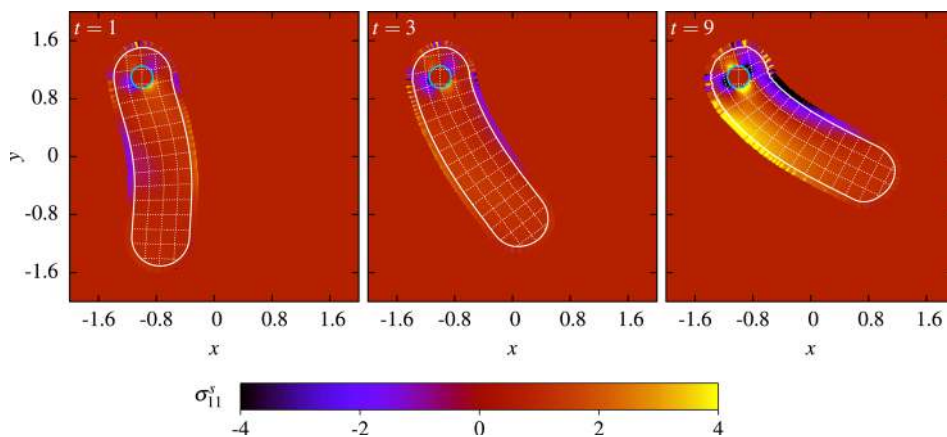
Fig. 9 Three snapshots of the solid stress component σ_{11}^s in the solid and the extrapolated region, for the anchored rod simulation shown in Fig. 4 but where the extrapolation procedure and level set motion routine of previous work [16] are used instead. The solid white line shows the boundary of the rod, given by the zero contour of the level set function $\phi(\mathbf{x}, t)$. The thin dashed white lines are the contours of the reference map $\xi(\mathbf{x}, t)$. The cyan circle with radius of 0.15 centered at $(-1,1.1)$ shows the anchored region where the reference map is fixed and the velocity is zero.
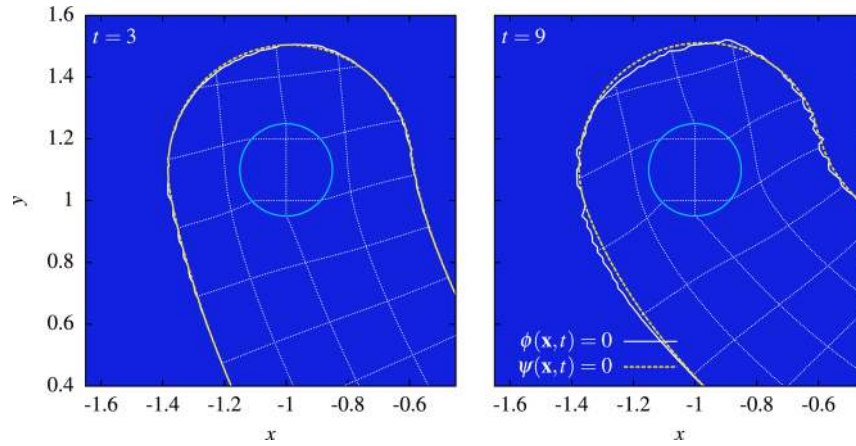
Fig. 10 Zoomed-in plots of the last two panels of Fig. 9, displaying drift between the interface as described by the level set $\phi(\mathbf{x}, t)$ and as described by the reference map, via $\psi(\mathbf{x}, t) = \phi(\xi(\mathbf{x}, t), 0)$. The thin dashed white lines are the contours of the reference map $\xi(\mathbf{x}, t)$. The cyan circle with radius of 0.15 centered at $(-1, 1.1)$ shows the anchored region where the reference map is fixed and the velocity is zero.

We note that satisfaction of the above constraint is equivalent to ensuring that all material points initially within the solid remain within the solid.

Because the values of ξ and ϕ are obtained from different numerical routines, discretization error can cause the above constraint to be violated over time. Recall that ξ is generated on points with $\phi > 0$ solely through linear extrapolation from the adjacent $\phi < 0$ domain, and hence the values of ψ near the zero contour of ϕ are not more than first-order accurate. The comparison between them in Fig. 10 is indicative that a drift between ψ and ϕ grows after the onset of the previously described wrinkling artifact. This in turn causes persistence of the wrinkled shape, because some material that began on the solid side of the $\phi = 0$ interface is falsely assigned as fluid, which permits the wrinkles to sustain themselves over time without the elastic response correcting their shape. This calls for a routine to ensure the *physical requirement* that the zero contours of ϕ and ψ satisfy Eq. (28), i.e., remain "pinned," to ensure that solid stays solid and fluid stays fluid. One could argue at this moment that ϕ is a redundant field, in that ψ could operate just as well on its own at discerning phases. However, we recall that ϕ also gives a measure of distance, as its gradient always has magnitude 1. This feature is exploited in the Aslam extrapolation scheme and is needed for the solid–solid contact algorithm of Sec. 6.

We now describe two midstep subroutines, which are applied after an initial Aslam extrapolation of ξ and successfully resolve the interfacial wrinkling, phase exchange error, and instabilities relating to these two phenomena. The need to appropriately pin the reference map extrapolation field and the level set field was pointed out previously [16] but a general and accurate pinning technique was not employed in that work. Artificial surface tension, projection of the extrapolated reference map field, and solid damping were used to reduce the appearance of these problems in that work.

The first step in our procedure is to remove artificial striations in the extrapolated values, see Algorithm 2. It is a general algorithm that can be applied to reduce fluctuations in an arbitrary extrapolated field f. In the FSI routine, it is applied separately to ξ_x and ξ_y. The fields on the solid side of the interface are never affected; only the extrapolated values within the fluid domain are adjusted. The algorithm consecutively applies a diffusion stencil to improve the extrapolated values. The edge of the solid domain is sampled by the diffusion stencil and hence the resulting extrapolation is both smooth within the fluid domain and continuously extends data from the solid domain.


Algorithm 2   Extrapolation smoothing subroutine.

Given: A field $f_{i,j}$ defined where $\phi_{i,j} < 0$, and an extrapolation of f into the region $w_E > \phi_{i,j} \geq 0$
Compute: A smoother extrapolation of f where $\phi_{i,j} \geq 0$
(1) Define a starting field $f_{i,j}^0 = f_{i,j}$ at all points in the region where $\phi_{i,j} > 0$ and all four orthogonal adjacent neighbors are part of the extrapolated region.
(2) For $k = 0, 1, \ldots, 5$ calculate $f_{i,j}^{k+1} = 0.05(f_{i,j-1}^k + f_{i-1,j}^k + f_{i+1,j}^k + f_{i,j+1}^k - 4f_{i+1,j}^k)$;
(3) Return $f_{i,j}^5$ as the smoothed extrapolation.

Upon completion of Algorithm 2, the second midstep routine starts by updating the level set values $\phi(\mathbf{x}, t)$ in the narrow band to be equal to $\psi(\mathbf{x}, t)$. This ensures that the zero contour of the level set is consistent with the reference map, but the new values of $\phi(\mathbf{x}, t)$ may not satisfy the signed distance function property, $|\nabla \phi| = 1$. We therefore make use of the reinitialization routine described in Ref. [18], to rebuild the narrow-banded level set function and recover this property. Figure 11 displays the solid stress from the reference map and its extrapolation in the same test geometry but using our new two-part extrapolation-pinning routine. The formerly observed striated solid stress, interfacial wrinkles, and drift between fields have been eliminated.

7 Simulation of Contact Between Two Solids Immersed in a Fluid

We now consider a second solid phase and seek to simulate a fully coupled fluid–solid–solid interaction, solids interacting through contact while submerged in a fluid. We describe the procedure for two solid phases, but the procedure could be generalized to more solids. We use superscripts (1) and (2) to denote each solid phase. There are now two level sets, $\phi_{i,j}^{(1)}$ and $\phi_{i,j}^{(2)}$, to describe each solid's boundary and measure distance to it. Each level set has a corresponding transition field, $\Omega^{(1)}$ and $\Omega^{(2)}$, using the same w_T for both. We maintain two reference map fields, $\xi_{i,j}^{(1)}$ and $\xi_{i,j}^{(2)}$, and corresponding solid stresses, $\sigma_{[i,j]}^{s(1)}$ and $\sigma_{[i,j]}^{s(2)}$, within the extended domain of each associated solid. The velocity field, as before, is a single unique field for the whole computational domain, independent of phase. The main procedure, Algorithm 1, requires very little change to represent interacting submerged solids. All steps corresponding to the solid are now simultaneously
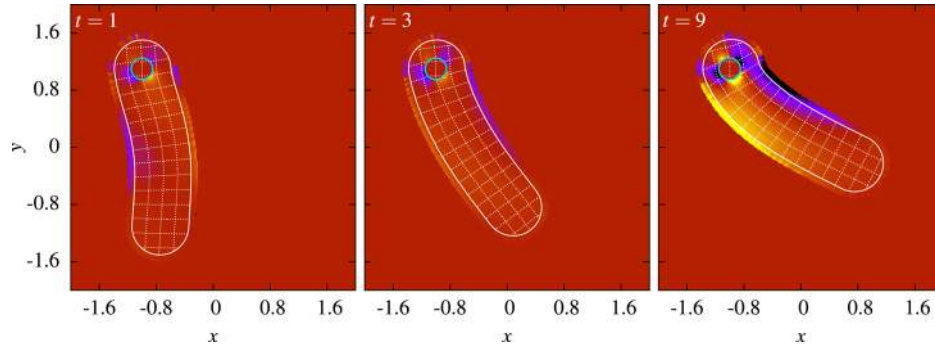
Fig. 11 Three snapshots of the solid stress component σ_{11}^s in the solid and the extrapolated region, for the anchored rod simulation of Fig. 4 that uses the new extrapolation procedure. The color gradient is the same as in Fig. 9. The solid white line shows the boundary of the rod, given by the zero contour of the level set function $\phi(\mathbf{x}, t)$. The thin dashed white lines are the contours of the reference map $\xi(\mathbf{x}, t)$. The cyan circle with radius of 0.15 centered at $(-1, 1.1)$ shows the anchored region where the reference map is fixed and the velocity is zero.

performed on both sets of solid fields. The major changes we must prescribe are to adjust the mixing rule in step 8 to correctly cross-fade between the two solids and fluid and to add an extra routine to correctly set contact conditions. The latter consideration ultimately arises as an adjustment in Step 6.

7.1 Mixing Formulation. The transition function of solid phase (i) indicates at any location the fraction of the material behavior attributable to that phase. Therefore, in the presence of two solid phases and fluid, we arrive at the below three-way mixing protocol

$$\boldsymbol{\sigma}_{[i,j]} = \Omega_{[i,j]}^{(1)} \boldsymbol{\sigma}_{[i,j]}^{s(1)} + \Omega_{[i,j]}^{(2)} \boldsymbol{\sigma}_{[i,j]}^{s(2)} + \left(1 - \Omega_{[i,j]}^{(1)} - \Omega_{[i,j]}^{(2)}\right) \boldsymbol{\sigma}_{[i,j]}^{f} \quad (29)$$

On each cell-centered grid point, the quantity of each solid follows the previous behavior, and the remainder is filled with fluid. The coefficient of the third term remains non-negative so long as the solid phases do not penetrate. The same approach is applied in defining the density.

7.2 Setting Contact Conditions. The contact algorithm focuses on the case of frictionless contact, but it is sufficiently general that other contact conditions could be implemented. In the blurred interface routine, we define contact between two objects whenever there is an overlap between the transition zone of one object and the interface of the other. We define penetration to occur if the zero contours of each level set pass through one another. Hence, a nonpenetrating contact routine permits the zero level sets of the two bodies to be separated by less than half the transition zone width—the "start" of blurred contact—and rapidly penalizes any closer approach of the two solids. When the solids are farther apart than half the transition zone width, the routine should have no effect. As the grid size shrinks, this description approaches the standard definition of nonpenetrating contact between sharp interfaces.

The key idea is to exploit the quantity defined as the difference between the two solid level set functions

$$\phi_{12} = \frac{\phi^{(1)} - \phi^{(2)}}{2} \quad (30)$$

Note that $\phi_{12} = 0$ implies a point in space equidistant between the two solid boundaries, and we refer to the set of all such points as the midsurface between the two bodies. At a point on the boundary of phase 1, the value of $2|\phi_{12}|$ indicates the distance to the nearest point on the boundary of phase 2 and vice versa. This is true regardless of the geometry of the two solids. This property

lets us use ϕ_{12} to construct a short-range separation force that captures our desired contact condition. We define a compactly supported influence function as

$$\delta_s(x) = \begin{cases} \dfrac{1 + \cos\dfrac{\pi x}{w_T}}{2w_T} & \text{if } |x| < w_T, \\ 0 & \text{if } |x| \geq w_T \end{cases} \quad (31)$$

so that $\delta_s(x)$ is the derivative of $H_s(x)$ as defined in Eq. (23). The above is used to define a mutually repulsive force field centered on the midsurface, which acts only on solid points that intersect its influence. Mathematically, this is achieved by defining the separation function

$$\gamma_{i,j} = k_{rep}\delta_s(\phi_{12i,j}) \quad (32)$$

from which the body force field is defined as

$$\mathbf{f}_{i,j} = \begin{cases} \gamma_{i,j}\hat{\mathbf{n}}_{12i,j} & \text{if } \phi_{i,j}^{(1)} < 0 \text{ or } \phi_{i,j}^{(2)} < 0, \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (33)$$

which is used within Eq. (21), where $\hat{\mathbf{n}}_{12i,j} = \pm\nabla\phi_{12i,j}/|\nabla\phi_{12i,j}|$ is the vector normal to the contours of ϕ_{12}, pointing away from the midsurface. The prefactor k_{rep} must be chosen large enough to successfully repel a crossing of the interface-centers, but must be small enough that the stable time-increment of such an explicit contact routine also decreases with k_{rep}. In the following examples, we choose $k_{rep} = 1/20$.

7.3 Computational Results. We first consider two submerged, colliding neo-Hookean disks with radius of 0.7 that are initially centered at $(x, y) = (\pm1.1, 0)$, using a 256×256 simulation grid and simulating over the interval $0 \leq t \leq 25$. Each disk has a circular anchoring region with radius of 0.25 at its center. The left disk's anchoring region is permanently fixed, while the right disk's anchoring region has a time-varying horizontal position

$$x_d(t) = 1.1 - 0.5\left(1 - \cos\frac{2\pi t}{25}\right) \quad (34)$$

and horizontal velocity

$$v_d(t) = -\frac{\pi}{25}\sin\frac{2\pi t}{25} \quad (35)$$
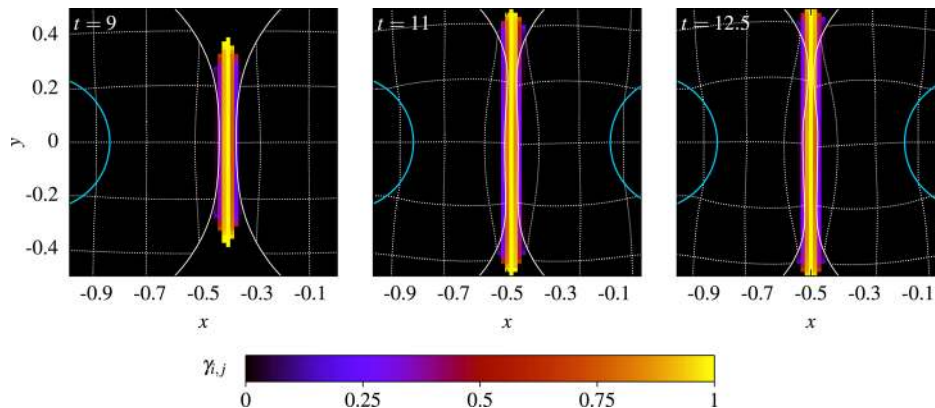
Fig. 12 Three snapshots showing a zoomed-in region of the separation function $\gamma_{i,j}$ in the simulation of two submerged, colliding disks. The function is only defined in the region where the narrow bands for the two level sets overlap and is plotted as zero outside this region. The solid white lines show the boundaries of the two disks, given by the zero contours of the level set functions. The thin dashed white lines are the contours of the reference map $\xi(\mathbf{x}, t)$ defined within the two disks. In each snapshot, the left cyan circle with radius of 0.25 is anchored, and the right cyan circle with radius of 0.25 is moving.

Figure 13 shows six snapshots of pressure during the simulation. By $t = 9$, the pressure starts to build up between the two disks as they come into contact. By the time the moving disk reaches its leftmost position at $t = 12.5$ when $x_d(t) = 0.1$, the pressure has increased further. At $t = 18$, the moving disk has separated from the anchored disk, and a small region of negative pressure is created between the two disks as fluid. At $t = 25$, the moving disk comes to rest at its original position, and returns to its original undeformed circular shape.

Figure 12 displays the $\gamma_{i,j}$ function at three time points and shows how the solids do not feel the interaction until the moment when the narrow δ_s function enters into the solids. Within the

simulation, the function is only defined in the region where the two narrow bands of the level set overlap. The repulsion force succeeds in separating the two interface-centers but maintains contact in the sense defined above. Some small asymmetry is visible, which is expected due to the asymmetry in the stencils described in Fig. 2; this effect diminishes with grid spacing. As is evident in Fig. 13 using this technique, we are now capable of getting very large deformations in the solids through contact, without any problem of penetration or sticking of one solid onto the other.

The second example that we consider is a disk bouncing on an anchored rubber bar, all while submerged in fluid. The disk has radius of 0.4 and is initially centered at $(x, y) = (1, 0)$. The bar
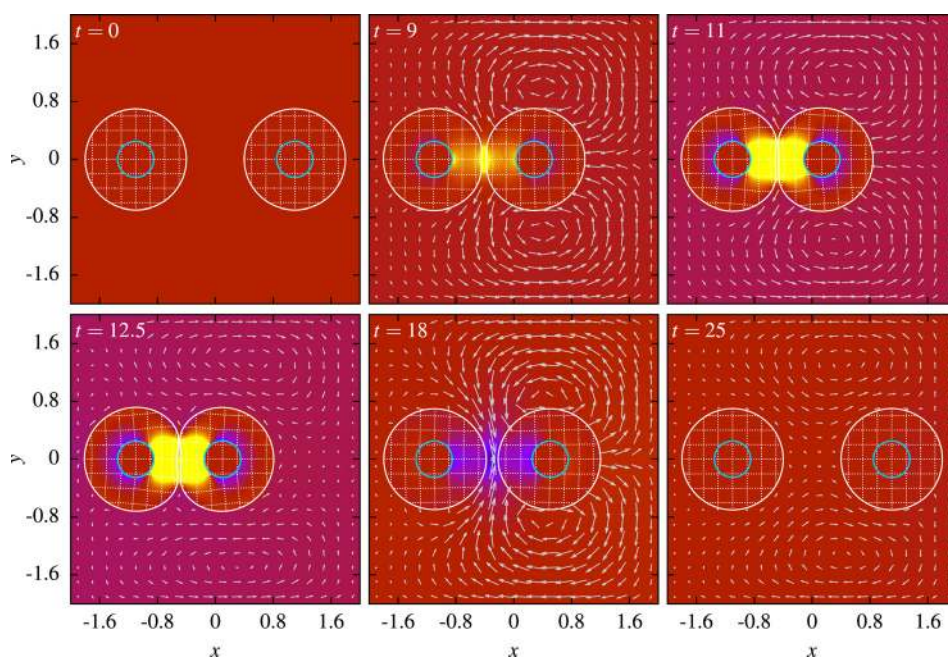
Fig. 13 Six snapshots of the in-plane pressure field (colors) and fluid velocity (arrows) for a simulation where a moving disk comes into contact with an anchored disk. The lengths of the arrows are proportional to $\sqrt{|\mathbf{v}|}$. The colors for the in-plane pressure use the same key as in Fig. 3. The solid white lines show the boundaries of the two disks, given by the zero contours of the level set functions. The thin dashed white lines are the contours of $\xi(\mathbf{x}, t)$ defined within the two disks. In each snapshot, the left cyan circle with radius of 0.25 is anchored, and the right cyan circle with radius of 0.25 is moving.
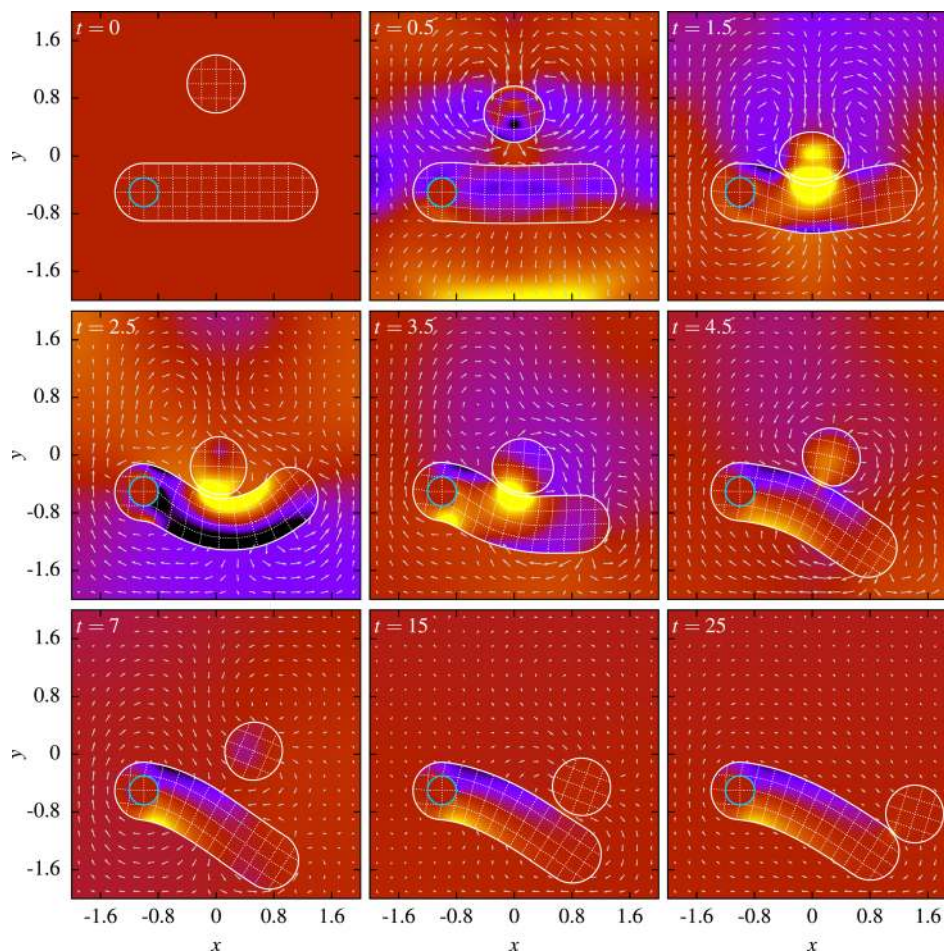
Fig. 14 Nine snapshots of the in-plane pressure field (colors) and fluid velocity (arrows) for the simulation of ball fired into an anchored flexible rod. The length of the arrows are proportional to $\sqrt{|\mathbf{v}|}$, where a nonlinear scaling is used to the large variations in the size of the velocity. The colors for the pressure use the same key as in Fig. 8. The solid white lines shows the boundary of the rod and the circle, given by the zero contour of the level set functions. The thin dashed white lines are the contours of $\xi(\mathbf{x}, t)$ defined within each of the two objects. The cyan circle with radius of 0.2 centered at $(-1, -0.5)$ shows the anchored region of the rod where the reference map is fixed and the velocity is zero.

comprises the rectangle $-1 < x < 1$, $-0.9 < y < -0.1$ along with semi-circular end caps and is anchored in a circle with radius of 0.2 at $(x, y) = (-1, -0.5)$. For this example, the density of the disk is 20, the disk's initial downward velocity is -1, the density of the rod is 6, the gravity is 0.03, and the fluid viscosity is $\eta = 0.04$. All other parameters are unchanged from the previous results. A grid of 600×600 is used and the system is simulated over the range of $0 \leq t \leq 25$.

Figure 14 shows nine snapshots of the in-plane pressure field during this simulation. The disk first reaches the rod at approximately $t = 1.5$, and a large region of positive pressure is visible between the two. By $t = 2.5$, the force exterted by the heavy disk deforms the bar into a U-shape, with a large tension visible on the bottom side of the bar. By $t = 4.5$, the right end of the bar has moved downward, and this motion pushes the disk upward so that it undergoes a bounce. By $t = 7$, the disk is fully separated from the bar. The disk then sinks and comes into contact with the bar at $t = 15$, before slowly sliding down the bar.

This approach allows us to model frictionless nonsticking contact that avoids cross penetration of the solid phases. We could imagine adherence conditions, friction, or other contact laws through new definitions of a possibly evolutionary influence function centered at the midsurface. Other Eulerian contact approaches also exist; we can, for example, measure the amount of overlap a step would cause and then apply a correction based on the

intersecting volume [31] and a minimization problem finding the optimal velocity field which solves the equilibrium condition while minimizing overlap.

8 Conclusion

This work has described a blurred-interface finite-difference method for FSI on a single Eulerian grid. The method is notable for its simplicity and speed. Our explicit algorithm invokes a computation of fluid and solid stress, which are then mixed in accordance with a transition function, as are the fluid and solid density. The solid stress and density are computed in Eulerian-frame with the aid of the reference map field, which is stored and updated throughout. An important pair of subroutines is also presented, to improve the representation of the reference map and level set fields near the interface. Our method is shown to converge and as grid-size decreases we recover results that properly display the signatures of a sharp interface. The framework we create extends to the case of solid–solid contact as well and we have given two examples of submerged contact for the case of nonpenetrating, frictionless, nonsticking behavior. Here, the key idea is to take advantage of the distance-function property of the level set fields about each solid to produce a short-range separation force that acts only within the solid interfacial regions when they are close enough to each other. We note that while our method has the key

feature of enabling simulation of largely deforming soft solids, there is nothing preventing this approach in the case of stiff solids; one can choose the elastic constants at will, as long as the stable time step is selected correspondingly.

There are a number of important directions to consider from this point. From a numerics standpoint, we can go to higher convergence rates by moving to a higher-order set of finite-difference stencils, though we emphasize the importance of removing oscillatory numerical errors, as we discussed in Sec. 6, which can arise in higher-order stencils. It is also important to port our scheme into 3D, which should require very little algorithmic adjustment. We can also implement incompressibility constraints by adapting the projection method [32]. It would be worthwhile to parallelize the scheme beyond multithreading to take advantage of a distributed-memory architecture. While we have focused on Cartesian grids for ease, this is not a necessity, and it would be useful to consider general meshes that enable local refinement.

There are also several next steps to be taken from the applications standpoint. Because our solid formulation is rooted in finite-deformation mechanics, any thermodynamically consistent constitutive law should be within the range of simulation. For example, one could model solid behaviors that depend on evolving internal variables by expressing the evolution laws of said variables in Eulerian-frame and storing/updating those variables on the grid (and using our various extrapolation routines to maintain accuracy near solid boundaries). We are also actively pursuing submodeling capabilities within our approach, which model the domain using two different grids of different refinement levels, and enable us to refine a local zone without the need to decrease the time step in the coarse grid zone. This would enable local refinement near a rough fluid–solid interface and could be applied to multiscale problems such as nuclear fuel-rod fretting, which involve interactions between cooling fluid and roughened solid parts. To extend the solid contact routine to the case of many submerged solid phases could have considerable usage in modeling dense suspensions of soft particles. To be able to model contact-induced finite deformations of particles is crucial when the compliance of particles is too high to admit simple contact-force laws. Biological applications, particular at the cellular level, could offer another arena in which this approach could be of use, as the solid components are highly deformable and fluid permeates the system. The Eulerian form could be advantageous in the implementation of growth models [33], or simultaneous species diffusion, which is also amenable to Eulerian-frame.

Acknowledgment

References

[1] Chorin, A. J., 1967, "A Numerical Method for Solving Incompressible Viscous Flow Problems," J. Comput. Phys., **2**(1), pp. 12–26.

[2] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., 1997, *Computational Fluid Mechanics and Heat Transfer*, Taylor and Francis, Bristol, PA.

[3] Versteeg, H. K., and Malalasekera, W., 1995, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Addison Wesley Longman, Ltd., Boston.

[4] Sethian, J. A., 1999, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science*, Cambridge University Press, Cambridge, UK.

[5] Hirt, C. W., Amsden, A. A., and Cook, J. L., 1974, "An Arbitrary Lagrangian Eulerian Computing Method for All Flow Speeds," J. Comput. Phys., **14**(3), pp. 227–253.

[6] Zienkiewicz, O. C., and Taylor, R. L., 1967, *The Finite Element Method for Solid and Structural Mechanics*, McGraw-Hill, New York.

[7] Sulsky, D., Chen, Z., and Schreyer, H., 1994, "A Particle Method for History-Dependent Materials," Comput. Methods Appl. Mech. Eng., **118**(1–2), pp. 179–196.

[8] Hoover, W. G., 2006, *Smooth Particle Applied Mechanics: The State of the Art*, World Scientific, Singapore.

[9] Belytschko, T., Liu, W. K., and Moran, B., 2000, *Nonlinear Finite Elements for Continua and Structures*, Wiley, New York.

[10] Bathe, K. J., ed., 2007, Fourth MIT Conference on Computational Fluid and Solid Mechanics, Comput. Struct., **85**(11–14), pp. 627–1164.

[11] Wang, X., 2008, *Fluid–Solid Interaction*, Elsevier Science, Amsterdam.

[12] Rugonyi, S., and Bathe, K. J., 2001, "On Finite Element Analysis of Fluid Flows Fully Coupled With Structural Interactions," Comput. Model. Eng. Sci., **2**(2), pp. 195–212.

[13] Peskin, C. S., 2002, "The Immersed Boundary Method," Acta Numer., **11**, pp. 497–517.

[14] Osher, S., and Sethian, J. A., 1988, "Fronts Propagating With Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," J. Comput. Phys., **79**(1), pp. 12–49.

[15] Kamrin, K., and Nave, J.-C., 2009, "An Eulerian Approach to the Simulation of Deformable Solids: Application to Finite-Strain Elasticity," http://arxiv.org/abs/0901.3799

[16] Kamrin, K., Rycroft, C. H., and Nave, J.-C., 2012, "Reference Map Technique for Finite-Strain Elasticity and Fluid–Solid Interaction," J. Mech. Phys. Solids, **60**(11), pp. 1952–1969.

[17] Udaykumar, H. S., Tran, L., Belk, D. M., and Vanden, K. J., 2003, "An Eulerian Method for Computation of Multimaterial Impact With ENO Shock-Capturing and Sharp Interfaces," J. Comput. Phys., **186**(1), pp. 136–177.

[18] Rycroft, C. H., and Gibou, F., 2012, "Simulations of a Stretching Bar Using a Plasticity Model From the Shear Transformation Zone Theory," J. Comput. Phys., **231**(5), pp. 2155–2179.

[19] Plohr, B. J., and Sharp, D. H., 1988, "A Conservative Eulerian Formulation of the Equations for Elastic Flow," Adv. Appl. Math., **9**(4), pp. 481–499.

[20] Trangenstein, J. A., and Colella, P., 1991, "A Higher-Order Godunov Method for Modeling Finite Deformation in Elastic–Plastic Solids," Commun. Pure Appl. Math., **44**(1), pp. 41–100.

[21] Liu, C., and Walkington, N. J., 2001, "An Eulerian Description of Fluids Containing Visco-Elastic Particles," Arch. Ration. Mech. Anal., **159**(3), pp. 229–252.

[22] Fedkiw, R. P., Aslam, T., Merriman, B., and Osher, S., 1999, "A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)," J. Comput. Phys., **152**(2), pp. 457–492.

[23] Chern, I. L., and Shu, Y., 2007, "A Coupling Interface Method for Elliptic Interface Problems," J. Comput. Phys., **225**(2), pp. 2138–2174.

[24] Liu, X. D., Fedkiw, R. P., and Kang, M., 2000, "A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains," J. Comput. Phys., **160**(1), pp. 151–178.

[25] Osher, S. J., and Fedkiw, R. P., 2003, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York.

[26] Chopp, D. L., 2001, "Some Improvements of the Fast Marching Method," SIAM J. Sci. Comput., **23**(1), pp. 230–244.

[27] Shu, C., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," J. Comput. Phys., **77**(2), pp. 439–471.

[28] Yu, J.-D., Sakai, S., and Sethian, J. A., 2003, "A Coupled Level Set Projection Method Applied to Ink Jet Simulation," Interfaces Free Boundaries, **5**(4), pp. 459–482.

[29] Yu, J.-D., Sakai, S., and Sethian, J., 2007, "Two-Phase Viscoelastic Jetting," J. Comput. Phys., **220**(2), pp. 568–585.

[30] Aslam, T. D., 2004, "A Partial Differential Equation Approach to Multidimensional Extrapolation," J. Comput. Phys., **193**(1), pp. 349–355.

[31] Levin, D. I. W., Litven, J., Jones, G. L., Sueda, S., and Pai, D. K., 2011, "Some Improvements of the Fast Marching Method," ACM Trans. Graph., **30**(4), pp. 36:1–36:10.

[32] Chorin, A. J., 1968, "Numerical Solution of the Navier–Stokes Equations," Math. Comput., **22**(104), pp. 745–762.

[33] Garikipati, K., 2009, "The Kinematics of Biological Growth," ASME Appl. Mech. Rev., **62**(3), p. 030801.