



# Reference map technique for finite-strain elasticity and fluid–solid interaction

Ken Kamrin<sup>a,\*</sup>, Chris H. Rycroft<sup>b,c</sup>, Jean-Christophe Nave<sup>d</sup>

<sup>a</sup> Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

<sup>b</sup> Department of Mathematics, University of California, Berkeley, CA 94720, USA

<sup>c</sup> Department of Mathematics, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

<sup>d</sup> Department of Mathematics and Statistics, McGill University, Montreal, QC, Canada H3A2K6

## ARTICLE INFO

### Article history:

Received 24 December 2011

Received in revised form

19 March 2012

Accepted 5 June 2012

Available online 23 June 2012

### Keywords:

Finite-differences

Numerical algorithms

Finite strain

Rubber material

## ABSTRACT

The reference map, defined as the inverse motion function, is utilized in an Eulerian-frame representation of continuum solid mechanics, leading to a simple, explicit finite-difference method for solids undergoing finite deformations. We investigate the accuracy and applicability of the technique for a range of finite-strain elasticity laws under various geometries and loadings. Capacity to model dynamic, static, and quasi-static conditions is shown. Specifications of the approach are demonstrated for handling irregularly shaped and/or moving boundaries, as well as shock solutions. The technique is also integrated within a fluid–solid framework using a level-set to discern phases and using a standard explicit fluid solver for the fluid phases. We employ a sharp-interface method to institute the interfacial conditions, and the resulting scheme is shown to efficiently capture fluid–solid interaction solutions in several examples.

Published by Elsevier Ltd.

## 1. Introduction

A classic dilemma in computational continuum mechanics is the choice of Lagrangian versus Eulerian frame techniques, each having certain benefits depending on the material type and conditions. A key example is the solid/fluid dichotomy: solids are typically simulated using Lagrangian methods with moving material nodes (e.g. finite-element methods, material point methods Zienkiewicz and Taylor, 1967; Sulsky et al., 1994; Hoover, 2006; Belytschko et al., 2000) and fluids using an Eulerian spatial grid (e.g. finite-difference/volume methods, level-set methods Chorin, 1967; Tannehill et al., 1997; Versteeg and Malalasekera, 1995; Sethian, 1999; Hirt et al., 1974). This division is partially rooted in constitutive response—solid stress depends on total deformation, computable from the relative positions of neighboring material nodes, whereas fluid stress depends on the deformation rate, obtainable from the finite-difference of a velocity field on a fixed-space mesh. Moreover, fluid flows often invoke mixing and in-flow/out-flow boundaries, which both point to simulation on an Eulerian grid. On the other hand, solid deformation is more inherently Lagrangian, characterized by smaller total strains and boundary conditions that generally move *with* the deforming boundary surface.

Fluid–structure interaction (FSI) is a prototypical example where the above dichotomy is problematic – the usual paradigm for each phase would necessitate a non-trivial and costly computation at the interface to reinterpret Lagrangian data into the fluid grid and vice versa. Methods that attempt to resolve this include the family of immersed methods,

\* Corresponding author.

E-mail addresses: [kkamrin@mit.edu](mailto:kkamrin@mit.edu) (K. Kamrin), [chr@math.berkeley.edu](mailto:chr@math.berkeley.edu) (C.H. Rycroft), [jcnave@math.mcgill.ca](mailto:jcnave@math.mcgill.ca) (J.-C. Nave).

which maintain an ambient stationary Eulerian grid throughout as well as a moving collection of interacting material points representing the solid structure (Peskin, 1977; Bathe, 2007; Wang, 2008). Another approach is to treat both phases with Lagrangian finite-elements but use an Arbitrary Lagrangian–Eulerian (ALE) method to remap the fluid mesh to prevent excessive distortion (Bathe, 2007; Wang, 2008; Rugonyi and Bathe, 2001). This approach is the more commonly used, and can be coupled with the Volume-of-Fluid method (Hirt and Nichols, 1981) to permit mixed-phase elements.

We propose to address this challenge and more with an explicit finite-difference method called the reference map technique (RMT) for finite-deformation solid laws. The idea was originally proposed by the author in Ken Kamrin (2008) and independently put forward by a Joseph Fourier University applied mathematics group (Cottet et al., 2008; Maitre et al., 2009) at around the same time. In both, it was emphasized for its potential use as part of a simple, fast, and generalizable fluid–solid coupling algorithm performable on a single fixed grid. The method is based on storage of the *reference map* vector field, which permits the construction of needed solid kinematic quantities, and which has been useful in several other contexts (see Section 5).

In this paper, we focus on hyperelasticity and perform a number of investigations to validate the numerical implementation of the basic method. We also describe and numerically validate specifications to handle an assortment of common solid loading conditions. Once accuracy is demonstrated, we propose a sharp-interface method based on a level-set formulation (Sethian, 1999; Osher and Sethian, 1988) to unite the RMT with an explicit fluid algorithm and compute fully coupled fluid/soft-solid interactions. The sharp interface approach we use, which ensures that material properties do not blur across the interface, is key to the stability and success of the method, and is based on an extension of the Ghost Fluid Method (GFM) for fluid/fluid interaction (Fedkiw and Liu, 1998).

To maintain a clear presentation, several avenues of motivation are first provided. The needed calculus for the reference map is described with emphasis on its relationship to the deformation gradient tensor, which leads to a presentation of the basic RMT iteration. Using three examples, we demonstrate the method’s capacity to handle dynamic, static, and quasi-static deformations, under both control-volume boundary conditions as well as Lagrangian displacement boundary conditions. We extend the method to an FSI algorithm and demonstrate its efficacy with three different FSI examples, each involving hyperelastic solids being deformed to large deformation against a flowing fluid. Lastly, we discuss and implement the method in rudimentary conservative form, and show its ability to track the motion of a genuinely non-linear one-dimensional shock.

## 2. Notation and kinematics

In Eulerian frame, the deformation of an isothermal material satisfies the conservation laws of mass and momentum balance,

$$\rho_t + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

$$(\rho \mathbf{v})_t + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} - \mathbf{T}) = \rho \mathbf{g} \quad (2)$$

which can be expressed in strong form when deformations remain smooth:

$$\rho_t = -\mathbf{v} \cdot \nabla \rho - \rho \nabla \cdot \mathbf{v} \quad (3)$$

$$\mathbf{v}_t = -\mathbf{v} \cdot \nabla \mathbf{v} + (\nabla \cdot \mathbf{T} + \rho \mathbf{g}) / \rho. \quad (4)$$

Here, the spatial velocity field is  $\mathbf{v}(\mathbf{x}, t)$  and the Cauchy stress tensor field is  $\mathbf{T}(\mathbf{x}, t)$ . For most of this paper it is appropriate to consider the strong form laws, though the conservative version is necessary in the presence of shocks.

To close the system, we now review the kinematics used to express finite-deformation solid constitutive relations. Suppose, that a body of material is in a reference configuration  $B$  with reference coordinate  $\mathbf{X}$ . The body then undergoes a deformation process such that at time  $t$  later an element of material originally at  $\mathbf{X}$  has been moved to  $\mathbf{x}$ . The motion is defined by  $\mathbf{x} = \boldsymbol{\chi}(\mathbf{X}, t)$ , and the body at  $t$  is in the deformed configuration  $B_t$ . The motion defines the deformation gradient  $\mathbf{F}$ ,

$$\mathbf{F}(\mathbf{X}, t) = \frac{\partial \boldsymbol{\chi}(\mathbf{X}, t)}{\partial \mathbf{X}} \quad \text{or} \quad F_{ij} = \frac{\partial \chi_i(\mathbf{X}, t)}{\partial X_j}. \quad (5)$$

Note that we use  $\nabla$  for gradients in  $\mathbf{x}$ , and always write gradients in  $\mathbf{X}$  in derivative form as above. The evolution of  $\mathbf{F}$  can be connected back to the velocity gradient via

$$\dot{\mathbf{F}} = (\nabla \mathbf{v}) \mathbf{F} \quad (6)$$

where we use  $\dot{\cdot}$  for material time derivatives. Since  $\det \mathbf{F} > 0$  for any physical deformation, the deformation gradient admits a polar decomposition  $\mathbf{F} = \mathbf{R} \mathbf{U}$  where  $\mathbf{R}$  is a rotation, and  $\mathbf{U}$  is a symmetric positive definite tensor obeying  $\mathbf{U}^2 = \mathbf{F}^T \mathbf{F} \equiv \mathbf{C}$  where  $\mathbf{C}$  is the right Cauchy–Green tensor.

## 3. General finite-strain elasticity

To demonstrate the use and simplicity of the method, this paper shall focus on one broad class of materials: large-strain, 3D, purely elastic solids at constant temperature. These materials are well-described, in a thermodynamically

consistent fashion, by the theory of hyperelasticity. Though other elasticity formulations exist (e.g. hypoelasticity and small-strain theory) the next section will recall how these are in fact specific limiting approximations to hyperelasticity. A brief summary is provided next to establish notation and key results (see Gurtin et al., 2010 for details).

As a noncommittal 3D extension of spring mechanics, one first presumes that the Helmholtz free-energy per unit (undeformed) volume  $\psi$  and Cauchy stress  $\mathbf{T}$  both depend only on the total local deformation:

$$\psi = \hat{\psi}(\mathbf{F}), \quad \mathbf{T} = \hat{\mathbf{T}}(\mathbf{F}) \quad (7)$$

where  $\hat{\cdot}$  is used to designate functions. It is also assumed that if no deformation has occurred (i.e.  $\mathbf{F} = \mathbf{1}$ ), then  $\mathbf{T} = \mathbf{0}$ . The form reduces greatly by requiring that the law obey frame-indifference and the second law of thermodynamics, which we express by the dissipation inequality

$$\rho \dot{\psi} - \mathbf{T} : \mathbf{D} \leq 0 \quad (8)$$

for deformation rate  $\mathbf{D} = (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)/2$ . To uphold Eq. (8) under all imposable deformations, it follows that

$$\mathbf{T} = 2J^{-1} \mathbf{F} \frac{\partial \hat{\psi}(\mathbf{C})}{\partial \mathbf{C}} \mathbf{F}^T \quad (9)$$

for  $J = \det \mathbf{F}$ . Likewise,  $\mathbf{C} = \mathbf{0}$  corresponds to a local minimum of  $\psi$ . Eq. (9) gives the (compressible) hyperelastic constitutive law.

#### 4. Other Eulerian approaches

To implement hyperelasticity and related thermodynamic solid models, the deformation gradient must be obtained. In Lagrangian frame, each material point is identified by  $\mathbf{X}$ , so  $\mathbf{F}$  can be computed by numerically differentiating current location against initial. In Eulerian frame the problem is more subtle, since it requires knowledge of past configurations. Previous work on this front is summarized below. We restrict attention to “fully Eulerian” approaches, though this is not to disregard partially Eulerian algorithms that iteratively remap Lagrangian deformation onto a fixed grid (Hirt et al., 1974; Benson, 1995).

A common approach is to state the elasticity relation as a rate-form for  $\mathbf{T}$ , which avoids having to directly store and update  $\mathbf{F}$ . The most well-known of such is the family of *hypoelastic* relations

$$\mathbf{T}^\circ = \mathcal{C} : \mathbf{D} \quad (10)$$

where  $\mathcal{C}$  is a fourth-rank tensor of elastic moduli and  $\mathbf{T}^\circ$  is an objective stress-rate, which is equal to  $\dot{\mathbf{T}}$  plus additional terms (generally functions of  $\mathbf{T}$  and  $\nabla \mathbf{v}$ ) that guarantee frame-indifference of the relation. Objective stress-rates are not unique and many forms for  $\mathbf{T}^\circ$  have been used (discussed in, for example, Meyers et al., 2006).

Expressing  $\dot{\mathbf{T}}$  as  $\mathbf{T}_t + \mathbf{v} \cdot \nabla \mathbf{T}$ , one can discretize Eq. (10) giving an Eulerian numerical scheme where the stress tensor and the velocity vector fields are stored and updated. Hypoelasticity is simple to use and effective in certain simulations (as in Tran and Udaykumar, 2004; Rycroft and Gibou, 2012), however it carries physical drawbacks that make it inappropriate for our work here. In view of future intentions to simulate non-equilibrium or thermalized materials, the hypoelastic framework is problematic as it lacks connection to a thermodynamic potential. As a consequence, certain processes can cause hypoelastic laws to give pathological results, such as a non-zero stress at zero deformation after cyclic straining (Kojic and Bathe, 1987). Although hypoelasticity can approximate isotropic hyperelastic behavior under small-stretch conditions, it fails to do so for finite deformations.

A different Eulerian computational approach is to store  $\mathbf{F}$  on the grid as a primitive variable and evolve it directly each time-step. To update  $\mathbf{F}$ , Eq. (6) must be expressed in Eulerian frame and discretized in space and time. Different Eulerian representations of Eq. (6) have been proposed leading to numerical routines with certain features. Examples include

$$\text{Plohr and Sharp (1989): } (\rho \mathbf{F}^T \hat{\mathbf{x}}_i)_t + \nabla \cdot (\rho \mathbf{F}^T (\hat{\mathbf{x}}_i \otimes \mathbf{v} - \mathbf{1}(\hat{\mathbf{x}}_i \cdot \mathbf{v}))) = \mathbf{0} \quad \{i = 1, 2, 3\} \quad (11)$$

$$\text{Trangenstein and Colella (1991): } (\mathbf{F}^{-T})_t + \nabla(\mathbf{v} \cdot \mathbf{F}^{-T}) = \mathbf{0} \quad (12)$$

$$\text{Liu and Walkington (2001): } \mathbf{F}_t + \mathbf{v} \cdot \nabla \mathbf{F} = (\nabla \mathbf{v}) \mathbf{F}. \quad (13)$$

To ensure  $\mathbf{F}$  remains a gradient quantity during numerical implementation, a gauge constraint must also be imposed during the calculation process. Eqs. (11) and (12) have the benefit of being expressed in conservative form, which extends their applicability to problems that lack a smooth solution. This feature has sparked interest in the approach and a number of recent validations and high-order tests have been conducted (Barton et al. 2010; Miller and Colella, 2002; Gavriluk et al., 2008).

### 5. The reference map technique

The RMT is a different approach, which we shall now describe. Define the reference map  $\xi(\mathbf{x}, t)$  as the inverse of the motion function, i.e.

$$\mathbf{X} = \xi(\mathbf{x}, t). \tag{14}$$

The map could be seen as a vector field in the deformed body that indicates the initial (or reference) location of the material currently occupying the position  $\mathbf{x}$ . Applying the chain rule to Eq. (14) at fixed  $t$ , we find  $d\mathbf{X} = \nabla \xi d\mathbf{x}$ . In view of Eq. (5),

$$\mathbf{F}(\mathbf{X}, t) = (\nabla \xi(\mathbf{x}, t))^{-1}. \tag{15}$$

Eq. (15) provides the underpinnings for the RMT approach. Rather than discretizing  $\chi$  in the reference space, as per Lagrangian solid computation, we discretize the reference map in the deformed space. By Eq. (15), given the reference map on a discrete set of Eulerian points, a consistent approximation for the  $\mathbf{F}$  tensor is found by taking the finite-difference gradient of  $\xi$  and inverting. This provides a straightforward Eulerian-frame calculation for  $\mathbf{F}$ , and consequently a mechanism for simulating thermodynamically compatible solid laws on a fixed grid. For example, hyperelastic stress is simply

$$\mathbf{T} = 2(\det \mathbf{F})^{-1} \mathbf{F} \frac{\partial \hat{\psi}(\mathbf{C})}{\partial \mathbf{C}} \mathbf{F}^T \Big|_{\mathbf{F} = (\nabla \xi)^{-1}, \mathbf{C} = (\nabla \xi)^{-T} (\nabla \xi)^{-1}} \tag{16}$$

which maintains the connection between the stress and the strain-energy potential through  $\xi$ .

We must also write an Eulerian rule for updating  $\xi$ . This is inferred by observing that the reference map never changes for a tracer particle—its reference location is always the same. Hence,  $\dot{\xi} = \mathbf{0}$ . Switching perspective, we obtain the evolution law

$$\xi_t + \mathbf{v} \cdot \nabla \xi = \mathbf{0} \tag{17}$$

or in conservation form,

$$(\rho \xi)_t + \nabla \cdot (\rho \xi \otimes \mathbf{v}) = \mathbf{0}. \tag{18}$$

If the initial configuration is undeformed (no pre-strain/stress) then we initialize  $\xi(\mathbf{x}, t = 0) = \mathbf{x} = \mathbf{X}$ . Otherwise, we may assign compatible pre-strain directly through  $\mathbf{X}_{pre} = \xi(\mathbf{x}, t = 0)$ , as described in Section 6.2.1, and incompatible pre-strain by a non-curl-free initial deformation gradient field  $\mathbf{F}_{pre}(\mathbf{x}, t = 0)$ . In the latter, the deformation gradient used in the constitutive law becomes  $\mathbf{F} = (\nabla \xi)^{-1} \mathbf{F}_{pre}$ . On moving material boundaries,  $\xi$  is obtained from the boundary displacement—that is, if a boundary point originally at  $\mathbf{X}_b$  is prescribed a displacement bringing it to  $\mathbf{x}_b$  at time  $t$ , then  $\xi(\mathbf{x}_b, t) = \mathbf{X}_b$  (see Sections 6.2.1 and 6.2.2). Along Eulerian in-flow/out-flow boundaries we obtain  $\xi$  from velocity conditions and Eq. (17) (see Sections 6.2.1 and 8.2); this equips the method for problems where material may enter or leave the computational domain during deformation.

This approach has certain benefits within the realm of Eulerian solid methods. Since  $\xi$  relates directly to the motion function, no gauge constraints are needed to enforce consistency in the deformation. The ease of implementing displacement boundary conditions is another advantage; methods that store only the  $\mathbf{F}$  tensor generally require global invariants or indirect methods to implement displacement conditions on moving boundaries. In keeping the connection to points in the reference configuration, we point out our method can lose accuracy in the presence of excessive mixing—methods keeping  $\mathbf{F}$  as the primitive variable are less affected by this problem. It might be possible to remedy the issue by enacting an inverse analogy of Arbitrary Lagrangian–Eulerian, whereby after a critical distortion (say, at some time  $t_c$ ), the reference map is reinitialized to  $\xi = \mathbf{x}$  and the deformation gradient  $\mathbf{F}^{t_c}$  is stored, so that  $\mathbf{F}(t > t_c) = (\nabla \xi)^{-1} \mathbf{F}^{t_c}(\xi)$ . However, we also note materials that can mix excessively are commonly of a fluid-like constitutive representation, whose stress can be expressed accurately (regardless of mixing) using the gradient of  $\mathbf{v}$  instead of  $\xi$ ; Section 7 exploits this.

The notion of a map that records initial locations of material has been defined by others in various different contexts. Koopman et al. (2008) use an “original coordinate” function akin to our reference map in defining a pseudo-concentration method for flow fronts. In Pons et al. (2006), the map was used in conjunction with a level-set function for enhanced processing of data along an interface. The inverse motion is also discussed in Belytschko et al. (2000) for use in Arbitrary Lagrangian–Eulerian finite element analysis.

### 6. Implementation

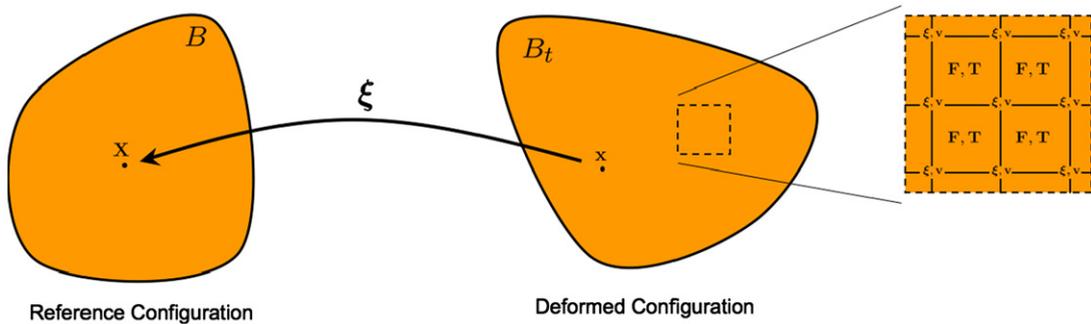
#### 6.1. Basic scheme—strong formulation

In this section we present the basic numerical scheme in its strong formulation, in the absence of boundaries. It is therefore assumed that the dynamics remain smooth in the sense that solutions stay sufficiently regular up to the final time. We define an Euler step,  $(\xi^{n+1}, \mathbf{v}^{n+1}) \equiv Eu(\xi^n, \mathbf{v}^n)$ , of our method as per Table 1, which updates the fields one time-step  $\Delta t$ . The entire solution scheme, corresponding to an Euler step is then embedded in the third order Runge–Kutta TVD

**Table 1**

The “basic” RMT routine;  $Eu(\xi^n, \mathbf{v}^n)$ , strong formulation.  
 (All  $\nabla$  operators discretize appropriately to centered or WENO finite-differences.)

<b>Given:</b> $\mathbf{v}^n$ and $\xi^n$	
<b>Goal:</b> Calculate $\mathbf{v}^{n+1}$ and $\xi^{n+1}$	
Step 1: Construct $\mathbf{F}$	$\mathbf{F} = (\nabla \xi^n)^{-1}$
Step 2: Compute $\rho$	$\rho = \rho_0(\det \mathbf{F})^{-1}$
Step 3: Compute $\mathbf{T}$	$\mathbf{T} = \hat{\mathbf{T}}(\mathbf{F})$
Step 4: Update $\mathbf{v}$	$\frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} + \mathbf{v}^n \cdot \nabla \mathbf{v}^n = \frac{1}{\rho} \nabla \cdot \mathbf{T} + \mathbf{g}$
Step 5: Update $\xi$	$\frac{\xi^{n+1} - \xi^n}{\Delta t} + \mathbf{v}^n \cdot \nabla \xi^n = \mathbf{0}$



**Fig. 1.** Locations of various fields with respect to the grid.

scheme (Shu and Osher, 1988),

$$(\xi^{n+1}, \mathbf{v}^{n+1}) = \frac{1}{3} Eu(\xi^n, \mathbf{v}^n) + \frac{2}{3} Eu \left\{ \frac{3}{4} (\xi^n, \mathbf{v}^n) + \frac{1}{4} Eu[Eu(\xi^n, \mathbf{v}^n)] \right\}. \tag{19}$$

For now we study the scheme in two dimensions of space by utilizing plane-strain conditions. On a two-dimensional grid (see Fig. 1), with grid spacing  $h$ , the velocity  $\mathbf{v}$  and reference map  $\xi$  are located at corner points  $(i, j)$ , while  $\mathbf{F}$  and the stress  $\mathbf{T}$  are located at cell centers,  $(i + \frac{1}{2}, j + \frac{1}{2})$ . Thus, away from any boundary, we can compute  $\partial_x \xi$  by finite-difference at the mid-point of horizontal grid edges, and similarly,  $\partial_y \xi$  on vertical grid edges, i.e.

$$\partial_x \xi_{(i+1/2, j)} = (\xi_{(i+1, j)} - \xi_{(i, j)}) / h. \tag{20}$$

We then obtain  $\nabla \xi$  at cell centers by bilinear interpolation, which is used to compute the deformation gradient tensor  $\mathbf{F}$  at cell centers as per the first step on Table 1. With  $\mathbf{F}$  computed, we now can define stress and density at cell centers using, respectively, the hyperelasticity law and the relation  $\rho = \rho_0(\det \mathbf{F})^{-1}$ . We compute  $\partial_x \mathbf{T}$  at the mid-point of vertical grid edges, and similarly,  $\partial_y \mathbf{T}$  on horizontal grid edges, i.e.

$$\partial_x \mathbf{T}_{(i, j+1/2)} = (\mathbf{T}_{(i+1/2, j+1/2)} - \mathbf{T}_{(i-1/2, j+1/2)}) / h.$$

As a result, bilinear interpolation gives  $\nabla \cdot \mathbf{T}$  at cell corners where  $\mathbf{v}$  is stored.

Finally, in implementing the advection laws Eqs. (4) and (17) (the last two steps on Table 1),  $\nabla \mathbf{v}$  and  $\nabla \xi$  are discretized using a standard WENO scheme (Liu et al., 1994). The mass density in the velocity advection equation is the average value from the four surrounding cell centers.

**Example: Elastic wave in a periodic domain.** To verify the method in smooth dynamic situations, we choose a problem for which an exact elastodynamic solution exists, and then compare the exact results to the numerical. To this end, consider a basic finite-strain elasticity law

$$\mathbf{T} = \kappa(\mathbf{V} - \mathbf{1}) \tag{21}$$

for  $\mathbf{V} \equiv \sqrt{\mathbf{F}\mathbf{F}^T}$  the left stretch tensor. Let the material body be a rectangular slab constrained in plane-strain conditions. The unstressed material density is uniform and has a value  $\rho_0$ . Under these conditions, the following  $\xi$  and  $\mathbf{v}$  fields are an analytical solution for a rightward moving compression wave passing through the slab

$$\hat{\mathbf{x}} \cdot \xi(x,y,t) = x + \frac{1}{2}\text{erf}(x-ct) \tag{22}$$

$$\hat{\mathbf{x}} \cdot \mathbf{v}(x,y,t) = c \left( 1 - \frac{1}{1 + \frac{1}{\sqrt{\pi}}e^{-(x-ct)^2}} \right). \tag{23}$$

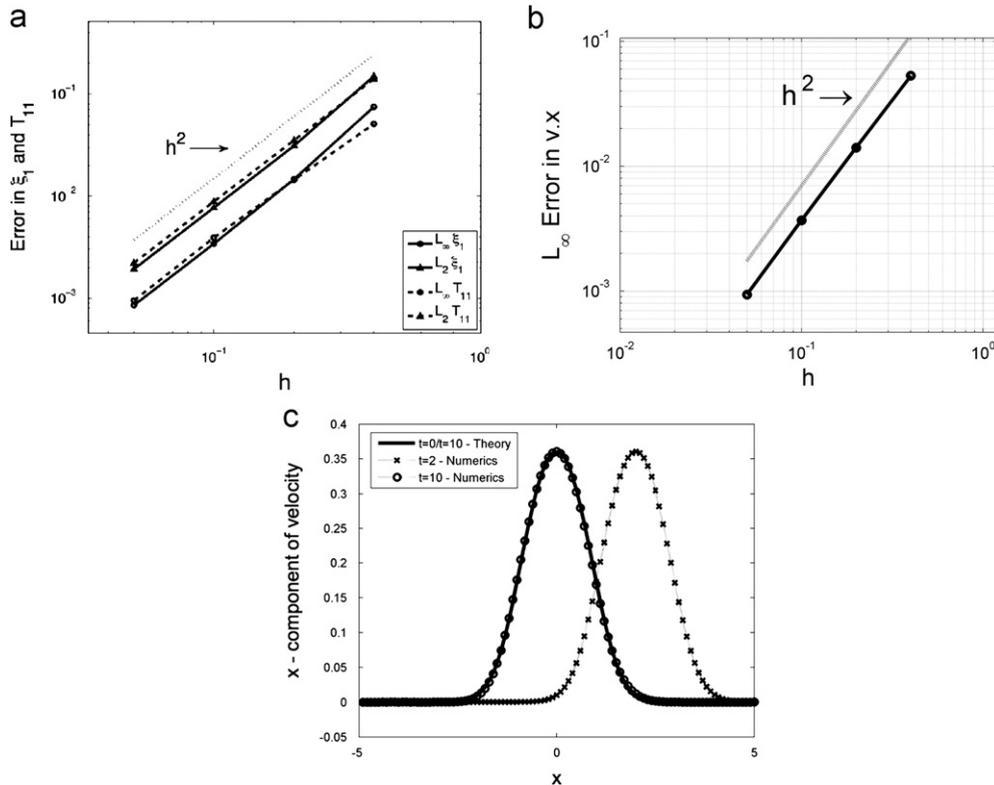
Due to symmetry, the  $\hat{\mathbf{y}}$  and  $\hat{\mathbf{z}}$  components of both fields do not change from their initial, unstressed values. The constant  $c = \sqrt{\kappa/\rho_0}$  is the wave speed. This solution invokes a large-strain deformation with compressive strain as high as  $|\hat{\mathbf{x}} \cdot (\mathbf{V}-\mathbf{1})\hat{\mathbf{x}}| \approx 36\%$  at the center of the pulse. As a result, this example serves as a test of the ability for the present approach to represent dynamic effects with large-strain deformation.

The Euler step is carried out as per the discretization presented above. The stability restriction of this fully explicit scheme is  $\Delta t < \alpha \min(\Delta x, \Delta y) \sqrt{\kappa/\rho_0}$ , for some small constant  $\alpha$ . From this approach we expect second-order global convergence. In order to verify the convergence rate, we set up a two-dimensional doubly-periodic domain  $\Omega = [-5, 5] \times [-5, 5]$ . At  $t=0$ , we use Eqs. (22) and (23), with  $c = \kappa = \rho_0 = 1$ . The travelling wave solution should come back to its original shape and location at  $t=10$ . For a sequence of grids with  $h = \Delta x = \Delta y = \{\frac{2}{40}, \frac{2}{20}, \frac{2}{10}, \frac{2}{5}\}$ , we set  $\Delta t$  using  $\alpha = \frac{1}{10}$ , and compute the  $L_\infty$  error of  $\xi$  as

$$E_\infty^\xi(h) = \sup_{(x,y) \in \Omega} |\hat{\mathbf{x}} \cdot \xi(x,y,10) - \hat{\mathbf{x}} \cdot \xi(x,y,0)|.$$

We report in Fig. 2(a) second-order global convergence in  $\xi$ , as expected. The convergence rate between the two finest grids,  $h = \frac{2}{10}$ , and  $h = \frac{2}{5}$  is computed to be 1.97. Also illustrated is a second-order convergence in the compressive stress as computed using the finite-difference of  $\xi$ .

In a similar manner, we compute and present convergence for  $\hat{\mathbf{x}} \cdot \mathbf{v}$ . In Fig. 2(b) we report the expected second-order global convergence for velocity. The convergence rate between the two finest grids,  $h = \frac{2}{10}$ , and  $h = \frac{2}{5}$  is computed to be 1.99. Finally, Fig. 2(c) shows one-dimensional cross sections for  $\hat{\mathbf{x}} \cdot \mathbf{v}$  at different times. The solid line represents the exact



**Fig. 2.**  $L_\infty$  and  $L_2$  norms of the error in  $\xi_1 = \xi \cdot \hat{\mathbf{x}}$  and  $T_{11} = \hat{\mathbf{x}} \cdot \mathbf{T}\hat{\mathbf{x}}$ , and in (b), the  $L_\infty$  error in  $v_1 = \mathbf{v} \cdot \hat{\mathbf{x}}$ . We observe a second-order global rate of convergence in all cases. (c) The velocity field  $\hat{\mathbf{x}} \cdot \mathbf{v}$  through the cross-section  $y=0$  over one full period  $t = [0, 10]$ . Comparison between analytical and numerical solutions.

solution computed from Eq. (23) at  $t=0$ , which corresponds as well to  $t=10$ , the time for which a wave has come back fully to its original location. We see that the exact solution and numerical solution agree well for  $h = \frac{2}{10}$ , a rather coarse grid. We also plot the numerical solution at  $t=2$  for illustrative purposes.

6.2. Finite body—kinematic boundary conditions

In this section we describe the treatment of kinematic boundary conditions with examples of a static and quasi-static problem, and demonstrate that the method converges to the correct solution, having a second-order rate of convergence.

6.2.1. Static case

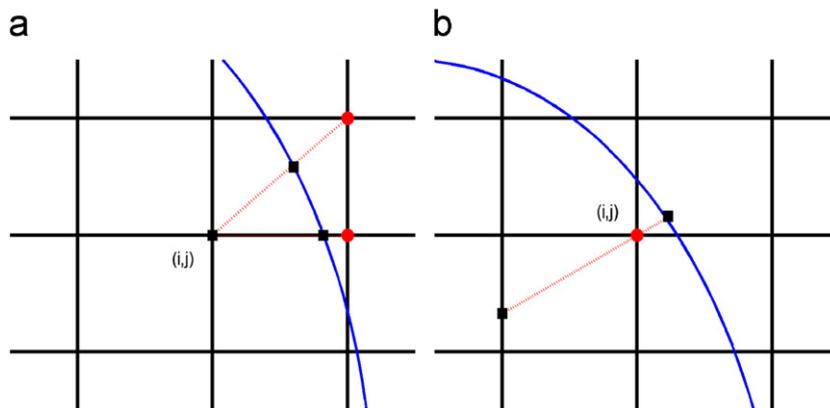
Here, we assume that the reference map on the boundary,  $\xi_B$ , is prescribed and fixed, and therefore the boundary velocity is  $\mathbf{v}_B = \mathbf{0}$ . Since we seek a static solution, a viscous term is added to the stress formula so that elastic waves are damped and the static result emerges from relaxation. We also choose a pre-strained initial  $\xi$  field, which matches  $\xi_B$  on the boundary of the domain and maps bijectively between the reference and the deformed body—by starting with a deformed “guess” for  $\xi(\mathbf{x})$  instead of initializing  $\xi(\mathbf{x}) = \mathbf{x}$  we promote a faster convergence to the true static solution by avoiding the need to model the boundary deformation process. This will be demonstrated best through a simple example later. One way to generate such a guess if no obvious one can be found, is to construct harmonic fields for  $\xi_1$  and  $\xi_2$  that satisfy the prescribed  $\xi_B$  using a boundary integral method.

It is important to note that we can only use the discretization above provided we adapt it near boundaries. To identify grid points outside the boundary and to locate where a boundary lies between grid points, we prescribe a level-set function  $\phi(\mathbf{x})$  whose zero contour is consistent with the deformed boundary. When the boundary crosses any edge between a given point  $(i,j)$ , and one of its eight neighbors,  $\{i-1,i,i+1\} \otimes \{j-1,j,j+1\} \setminus (i,j)$ , we use linear extrapolation to provide a value to the point outside the boundary. The procedure is illustrated in Fig. 3(a). In this example, two points (red circles) required to properly evaluate the stress fall outside the boundary. Along each dotted red line, we use the values given at the location of the black squares to extrapolate linearly to the location of the red circles. Note that the values of  $\xi$  and  $\mathbf{v}$  on the boundary are given from the boundary conditions. Once all values of points falling outside (red circles) the boundary have been updated, the algorithm described in the previous section can be applied without modification to update the fields at the point  $(i,j)$ .

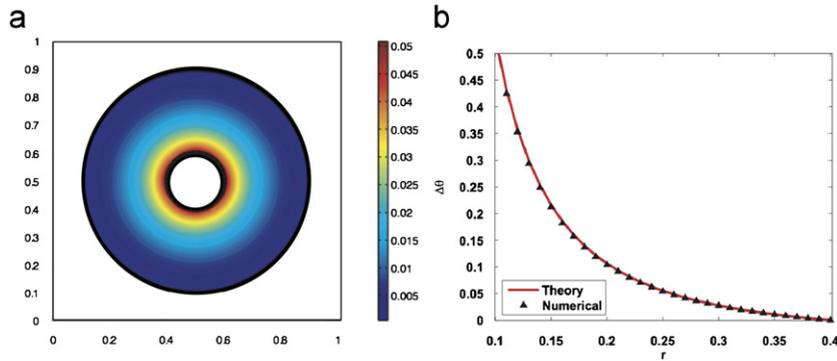
**Example: Circular washer shear.** To conduct a suitable test, we seek a problem having a known exact solution with a non-trivial, inhomogeneous deformation that is non-conforming to the Cartesian grid. Such problems are difficult to find, but one example is the Levinson–Burgess hyperelasticity law applied in a circular washer geometry for which the outer wall is fixed and the inner wall is rotated over a large angle. Admittedly, this particular elasticity law is somewhat esoteric, however it is adequate for the purposes of verifying the accuracy of the numerical method. The Levinson–Burgess free-energy function, after application of Eq. (9), gives the following relation under plane-strain conditions (Haughton, 1993):

$$\mathbf{T} = f_1(I_3)\mathbf{B} + f_2(I_1, I_3)\mathbf{1} \tag{24}$$

where  $\mathbf{B} = \mathbf{F}\mathbf{F}^T$  is the left Cauchy–Green tensor,  $I_1 = \text{tr } \mathbf{B}$ ,  $I_3 = \det \mathbf{B}$  are invariants of the  $\mathbf{B}$  tensor, and  $f_1(I_3) = G(3 + 1/I_3)/(4\sqrt{I_3})$  and  $f_2(I_1, I_3) = G\sqrt{I_3}(\kappa/G - 1/6 + (1 - I_1)/(4I_3^2)) - G/3 - \kappa$ . Under small strains,  $\kappa$  and  $G$  represent the bulk and shear moduli. Throughout, we use  $\kappa = G$ .



**Fig. 3.** (a) Routine for handling irregular boundaries. Values are extrapolated to nearby grid-points beyond the material domain using known information at the boundary. (b) Treatment of grid-points that enter the material domain during a time-step. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)



**Fig. 4.** (a) Scalar displacement field, as predicted by the method, for a static hyperelastic washer sheared by a finite rotation angle on its inner wall. Performed on a  $100 \times 100$  grid. (b) Corresponding angular displacement profile.

Angular displacement boundary conditions  $\Delta\theta_{in} = \pi/6$  and  $\Delta\theta_{out} = 0$  are prescribed to the inner and outer walls of the washer. We choose a simple, pre-strained initial  $\xi$  field for the simulation, corresponding to a purely angular displacement that is linear in the radial coordinate and matches the prescribed boundary displacements.

The analytical solution for the static displacement field, under Levinson–Burgess hyperelasticity, is  $\Delta\theta = A - B/r^2$  and  $\Delta r = 0$  where  $A$  and  $B$  are used to fit the boundary conditions. The graph in Fig. 4(b), which is in consistent though arbitrary length units, shows excellent agreement between our numerical solution (sampled along the central horizontal cross-section) and the analytical despite the fact that the scheme is based on a cartesian mesh while the geometry is radially symmetric. We have observed equally good agreement when the inner wall rotation angle is varied.

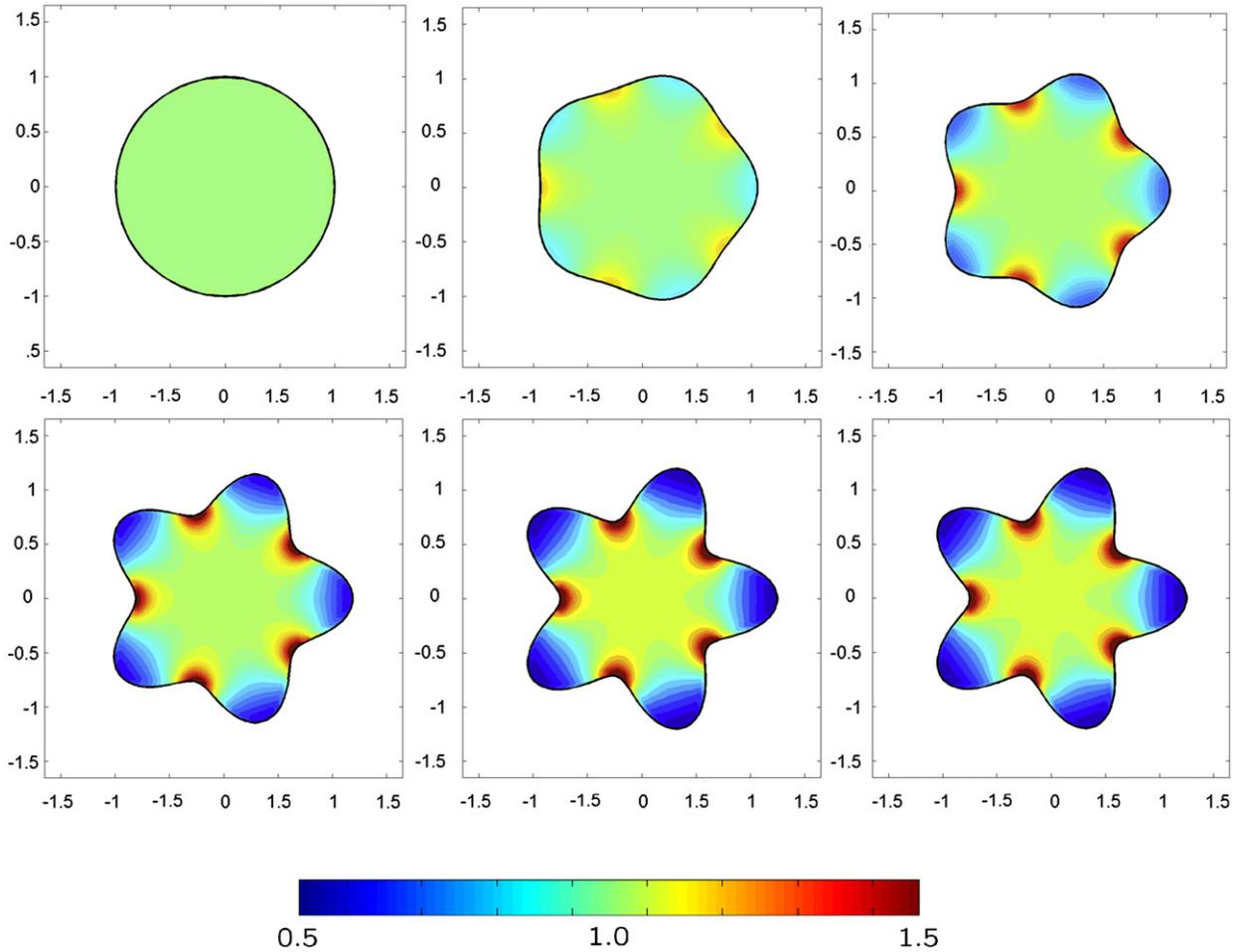
### 6.2.2. Quasi-static case

Here, we assume that the reference map and velocity on the boundary,  $\xi_B$  and  $\mathbf{v}_B$ , are prescribed and time-dependent. We include viscous damping, as before, so that the motion appears as a sequence of static states. This ensures we stay within the dissipative regime in which the strong form of our algorithm is valid.

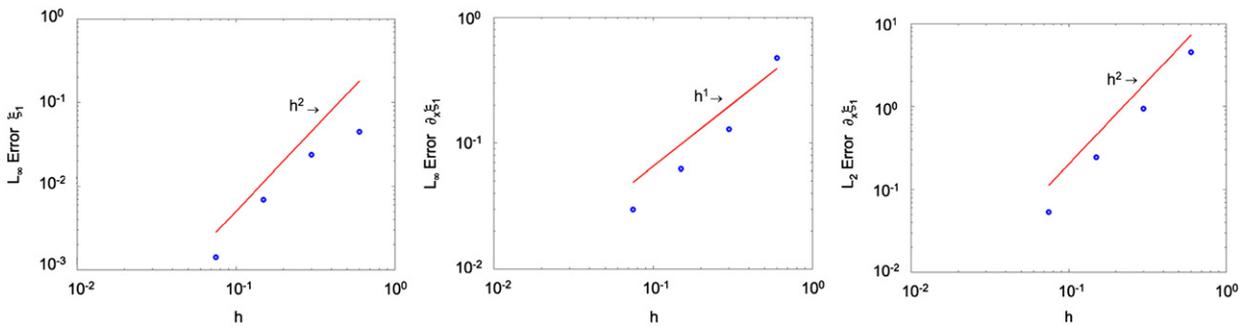
As before we utilize a level-set function  $\phi$  to distinguish the boundary, but now we let it be time-dependent in agreement with the applied boundary motion. One must pay special attention when the boundary crosses a point within one time step  $\Delta t$ . In this case, we provide  $\xi$  and  $\mathbf{v}$  values to the new point (red dot in Fig. 3(b)) by interpolating values from the inside of the domain and the boundary (black squares). The interpolation is linear, along a line normal to the boundary and going through the point in question. This line crosses the boundary and a cell edge (black squares) where the fields can be evaluated. This procedure is also applied to any grid point within the body that is very close to the boundary, and the size of the cut-cell is too small to guarantee numerical stability. As a rule of thumb, we switch to this procedure when the cut-cell is smaller than  $\Delta x/10$ .

A simple benchmark property of the boundary routines we have just described is that all discretization error vanishes in the case of a uniform body moving under constant velocity boundary conditions. This fact can be proven by noting that the exact kinematic fields vary linearly in this case, which removes discretization error from the boundary and finite-difference calculations. We have verified this as well with numerical tests, by simulating the movement of an elastic disk with an initially uniform deformation  $\mathbf{F}^0$ , uniform initial velocity  $\mathbf{v}^0$ , and displacement boundary conditions for  $t > 0$  consistent with a boundary velocity of  $\mathbf{v}^0$ . Regardless of  $\mathbf{v}^0$ ,  $\mathbf{F}^0$ , and the grid-spacing, we have verified that the  $\mathbf{F}$  matrix at all locations and times is within floating-point error of the exact solution  $\mathbf{F}^0$ . Likewise the Cauchy stress is equally accurate.

**Example: Quasi-static flower deformation.** Fig. 5 shows a sequence in time for the molding of a circular object into a five-petal flower shape, using the hyperelasticity law from the prior section. The boundaries are defined by a level set function on the regular  $80 \times 80$  Cartesian grid. We choose this particular example because it has large deformation and symmetry-breaking characteristics with respect to the Cartesian grid. The sequence shows relative density contours as the object is deformed from its initial configuration to the final, static state. Additionally, we have performed a convergence study of the present problem using a sequence of grids from  $10 \times 10$  to  $80 \times 80$  points. For each grid, after the time-dependent boundary deformation has concluded, we hold the final boundary configuration in place to ensure a static solution: Fig. 5 shows that the changes occurring after boundary deformation has ceased are quite miniscule, as desired of a quasi-static process. The error in the  $L_\infty$  norm is then evaluated by comparing the final solution against the same solution on a  $160 \times 160$  grid. Fig. 6 shows the second-order convergence of  $\xi$ , as predicted by our analysis. In order to quantify the accuracy of the Cauchy stress, we also perform a convergence study of  $\nabla\xi$ , noting that  $\mathbf{T}$  and  $\nabla\xi$  must have the same order of accuracy. The tensor  $\nabla\xi$  is computed using standard centered finite-difference of the field  $\xi$ . Despite the second-order convergence of  $\xi$ , we observe a first-order  $L_\infty$  convergence in  $\nabla\xi$ . This is in contrast to the case of the elastic wave in the periodic domain described previously for which second-order was observed for both  $\xi$  and its finite-difference derivative. We expect such a difference here, since the introduction of internal boundaries as discretized above will disrupt the spatial



**Fig. 5.** A sequence of snapshots of contours of  $\rho/\rho_0 = (\det \mathbf{F})^{-1}$  as a hyperelastic circular disk is quasi-statically molded into a five-petal flower over some time  $t_f$ . The first through fifth snapshots are equally spaced during the deformation and the last image is the result after holding the final boundary configuration in place for an additional time  $t_f$ , indicative of a fully-relaxed result.



**Fig. 6.** Convergence studies for the five-petal flower deformation test.

smoothness of the truncation term in the Taylor series analysis causing a local accuracy drop in the finite-difference of  $\xi$  in the direct neighborhood of the boundary. To verify that this is the case, Fig. 6 also presents the convergence in the  $L_2$  norm for  $\nabla \xi$ . We observe a second-order convergence in that norm, clearly indicating that the drop in order observed for  $L_\infty$  is localized at the boundary and of minimal effect within the bulk. It is clear that on one hand, more work is required to provide a boundary discretization that will guarantee second-order convergence in the  $L_\infty$  norm for the derivative of  $\xi$ . But, on the other hand, the present boundary treatment is simple to implement and provides second-order accurate solutions for the kinematic fields and, under the  $L_2$  norm, the stresses.

## 7. Fluid–solid coupling

In the previous sections, the reference map technique has been introduced and compared against analytical results. With the basic approach validated, we now proceed to demonstrate the ability of the method to simulate fluid–structure interaction problems. For simplicity of presentation, we consider two-dimensional plane-strain examples as before, and we make use of non-dimensional simulation units—converting the results to physical units can be carried out by introducing a mass, length, and time scale.

In the examples presented here, a hyperelastic solid is coupled to a weakly compressible, athermal fluid phase with velocity field  $\mathbf{v}_f$  and density  $\rho_f$ . The fluid stress tensor satisfies

$$\mathbf{T}_f = \eta \frac{\nabla \mathbf{v}_f + (\nabla \mathbf{v}_f)^T}{2} - \lambda_f \left( \frac{\rho_f}{\rho_{f0}} - 1 \right) \mathbf{1} \quad (25)$$

where  $\eta$  is a viscosity,  $\lambda_f$  is the fluid compressibility modulus, and  $\rho_{f0}$  is the initial density. The velocity and density obey

$$\rho_f \left( \frac{\partial \mathbf{v}_f}{\partial t} + (\mathbf{v}_f \cdot \nabla) \mathbf{v}_f \right) = \nabla \cdot \mathbf{T}_f \quad (26)$$

$$\frac{\partial \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{v}_f) = 0. \quad (27)$$

We consider a no-slip boundary condition in which  $\mathbf{v}_f = \mathbf{v}_s$  on the fluid–solid interface. The components of stress in the normal direction must match, so that if  $\mathbf{n}$  is the normal vector, then  $\mathbf{T}_s \cdot \mathbf{n} = \mathbf{T}_f \cdot \mathbf{n}$ .

It is key to point out how little the algorithm changes to switch from simulating the solid to the fluid phase under the RMT approach. When in fluid, the routine merely replaces Steps 2 and 3 from Table 1 with the equivalent stress and density formulations for fluid, using a finite-difference of  $\mathbf{v}_f$  rather than  $\xi$  to compute the stress, and evolution of  $\rho_f$  by Eq. (27) rather than  $\rho = \rho_0 (\det \mathbf{F})^{-1}$ .

### 7.1. Methods

The simulations are carried out on an  $m$  by  $n$  rectangular grid of square cells with side length  $h$ . The time-integrated fields of  $\mathbf{v}_s$ ,  $\xi$ ,  $\mathbf{v}_f$ ,  $\rho_f$  are computed on the  $(m+1)$  by  $(n+1)$  staggered grid of cell corners, while the stress tensors  $\mathbf{T}_f$  and  $\mathbf{T}_s$  are computed on cell centers. For time-stepping, a first-order forward Euler scheme is employed. The fluid–solid interface is described by a level set function  $\phi$  held at cell corners, with solid grid points corresponding to  $\phi \leq 0$  and fluid grid points corresponding to  $\phi > 0$ . The level set update is carried out using the hybrid approach described by Rycroft and Gibou (2012) that makes use of the method described by Chopp (2001) to update points straddling the interface, coupled to a second-order fast marching method to determine points that are further away.

As part of the simulation it is also necessary to characterize which phase the cell-centered grid points are within. If all four of a cell's corners are in fluid, then the node is classed as “full fluid”. If all four corners are solid, then the node is classed as “full solid”. Otherwise, a value  $\phi_c$  is computed at the cell center as the average of  $\phi$  at the four corners. If  $\phi_c \leq 0$ , the node is classed as “partial solid” and if  $\phi_c > 0$ , the node is classed as “partial fluid”. The simulation method is not sensitive to the distinction between partial fluid and partial solid, and other prescriptions such as those based on the number of corners that are fluid or solid have achieved similar results.

The level set field can also be used to carry out linear field extrapolation using the equations described by Aslam (2004) and the algorithms described by Rycroft and Gibou (2012); the routines can be applied in both directions, so that fluid fields can be extrapolated to solid grid points and vice versa. In addition, a routine has been written to allow linear extrapolation of the cell-centered stress fields using the corner-centered level set function. The routine can extrapolate a field from full solid nodes to all partial nodes and all full fluid nodes, and vice versa.

Each test problem is initialized by specifying a level set function, and defining the field  $\mathbf{v}_f$  and  $\rho_f$  at fluid grid points, and the fields  $\mathbf{v}_s$  and  $\xi$  at solid grid points. Before starting the simulation the fields  $\mathbf{v}_s$  and  $\xi$  are extrapolated into the fluid phase, and the fields  $\mathbf{v}_f$  and  $\rho_f$  are extrapolated into the solid phase. To take a step forward in time, the following procedure is applied, which is essentially an extension of the Ghost Fluid Method (Fedkiw and Liu, 1998) to the case of a fluid–solid interface:

1. Calculate the update to the reference map  $\xi$  based on advection by  $\mathbf{v}_s$ , but do not apply immediately.
2. Compute the intermediate stresses  $\mathbf{T}_s$  and  $\mathbf{T}_f$  at the full solid and full fluid grid points respectively. Extrapolate both stress tensors to all partial grid points.
3. Fix extrapolated velocity fields to match the no-slip boundary condition, and project the extrapolated  $\xi$  to maintain consistency with the interface  $\phi = 0$ .
4. Fix the extrapolated stresses to ensure proper continuity of the stress.
5. Calculate the update to the velocities  $\mathbf{v}_f$  and  $\mathbf{v}_s$ , density  $\rho_f$  but do not apply immediately.
6. Update the fields  $\mathbf{v}_s$ ,  $\mathbf{v}_f$ ,  $\xi$ , and  $\rho_f$ .
7. Set any values of the fields  $\mathbf{v}_s$ ,  $\mathbf{v}_f$ ,  $\xi$ , and  $\rho_f$  that are fixed according to boundary conditions.

8. Extrapolate the fields  $\mathbf{v}_s$ ,  $\mathbf{v}_f$ ,  $\xi$ , and  $\rho_f$ .
9. Move the level set according to the extrapolated solid velocity.

Additional computational details about steps 3 and 4 are provided in Appendix A.1. Snapshots of the simulation fields are saved at periodic intervals.

The fluid has initial density  $\rho_{f0} = 1$ , viscosity  $\eta = 0.12$ , and compressibility modulus  $\lambda_f = 60$ . The solid phase is modeled as a compressible neo-Hookean elastic solid, with small-strain bulk and shear moduli  $\kappa = 50$  and  $G = 10$  respectively. That is, for  $J = \det \mathbf{F}$  and  $'$  indicating the deviator,

$$\psi = \frac{G}{2} (J^{-2/3} \text{tr}(\mathbf{B}) - 3) + \frac{\kappa}{2} (J - 1)^2, \quad \mathbf{T}_s = G J^{-5/3} \mathbf{B}' + \kappa (J - 1) \mathbf{1}. \tag{28}$$

A damping viscosity of 0.06 is also applied to the solid. The simulation is written in C++, and for each of the test runs described below, a running time on a 3.4 GHz Apple iMac system is reported.

### 7.2. Release of a pre-strained disk within fluid

The first test of the method is carried out in a square domain  $[-2, 2]^2$  with a grid of  $m = n = 129$ . A disk of radius  $\frac{4}{5}$  is initially stretched so that its reference map is given by

$$(\xi_1, \xi_2) = \left( \frac{8x+5}{10} + \frac{y}{2}, y \right).$$

The boundary of the circle is initially given by

$$\phi(x, y) = |\xi| - \frac{4}{5}$$

and the solid and fluid are initially at rest. No slip boundary conditions are applied on all four edges of the domain. Fig. 7 shows a sequence of snapshots of the pressure from  $t = 0$  to  $t = 5$ ; simulating over this time interval took 22.7 s to carry out. Since the body is initially stretched, its pressure is negative. As the simulation progresses, the circle undergoes several oscillations that are slowly damped as energy is lost due to viscosity.

At  $t = 5$  it can be seen that the circle has returned to its rest configuration. Even though the level set only defines the position of the boundary implicitly, the continual projection of the reference map at the boundary ensures that the shape is retained throughout the simulation, and numerical errors do not distort its shape over time. Since the circle was initially

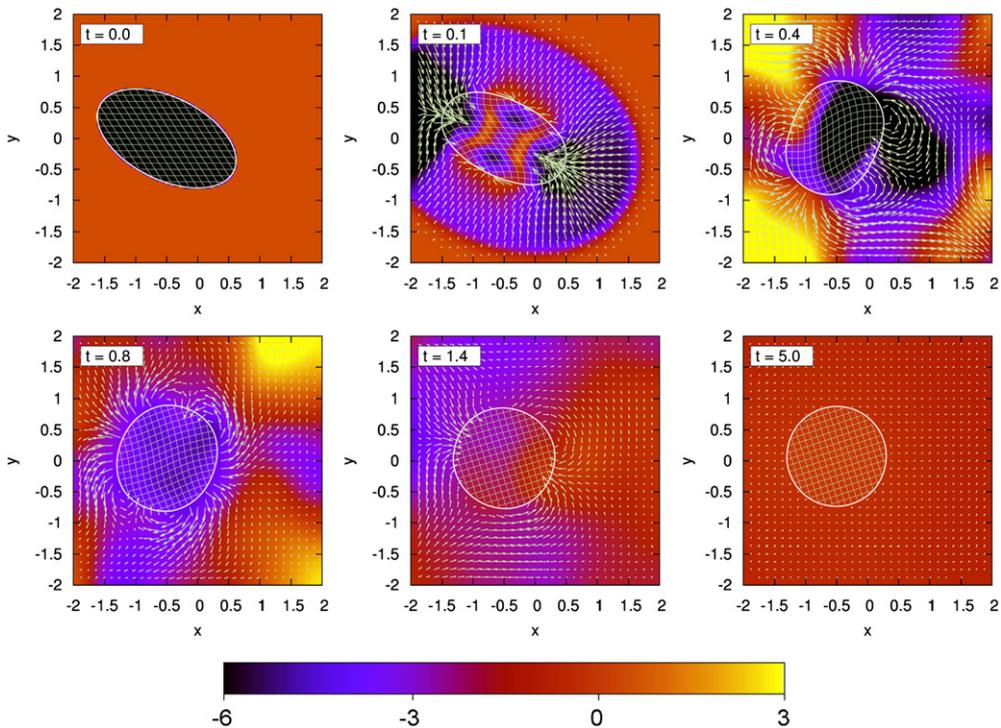
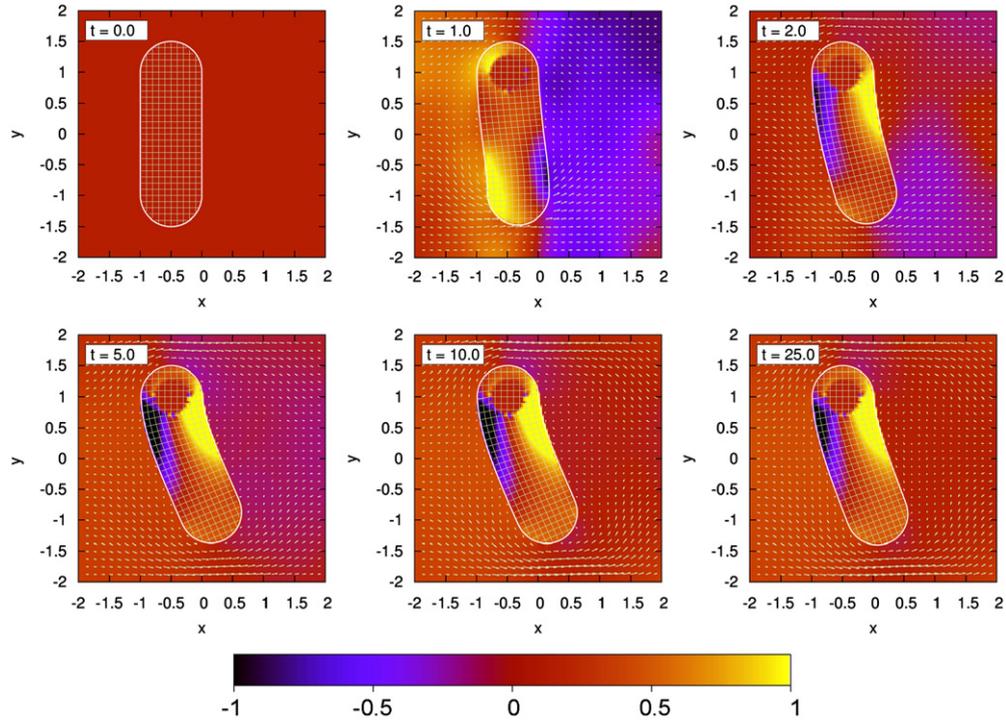


Fig. 7. Snapshots of pressure for the test problem in which an initially stretched disk is released within a fluid. The fine rectangular grid on the circle shows the contours of the  $\xi_1$  and  $\xi_2$  components of the reference map. Arrows show the fluid velocity.



**Fig. 8.** Snapshots of pressure for the test problem in which an anchored flexible rod is deformed by a fluid flow. The fine rectangular grid on the rod shows the contours of the  $\xi_1$  and  $\xi_2$  components of the reference map. Arrows show the fluid velocity.

stretched and has now contracted, the average pressure in the fluid in the final configuration is negative due to conservation of volume.

### 7.3. Body within a flow

In the second test simulation, a rod centered on  $(x,y)=(0.5,0)$ , with rounded ends, is initialized with the level set function

$$\phi(x,y) = \begin{cases} \sqrt{(x+0.5)^2 - (|y|-1)^2} - 0.5 & \text{for } |y| > 1 \\ |x+0.5| - 0.5 & \text{for } |y| \leq 1 \end{cases}$$

The rod is initially at rest so that  $\xi = \mathbf{x}$  and  $\mathbf{v}_f = \mathbf{0}$ . The rod is anchored, so that for the region

$$\sqrt{(x+0.5)^2 + (y-1)^2} < 0.3$$

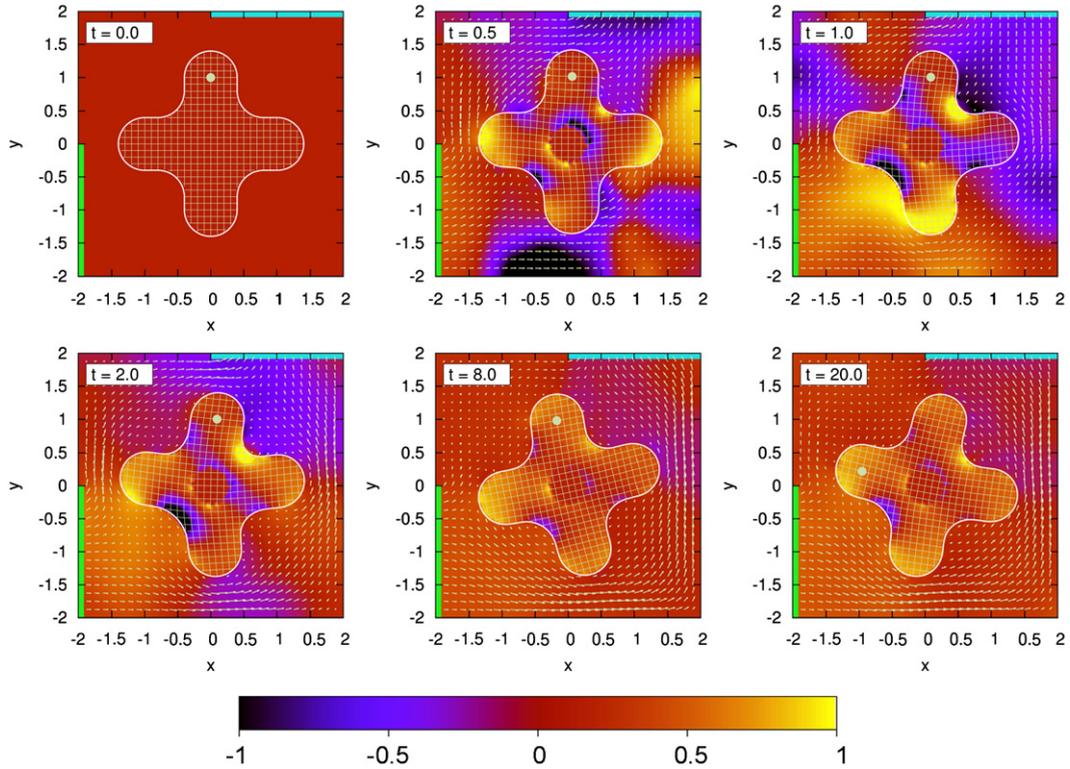
the velocity is constrained to be zero and the reference map is constrained to be equal to  $\mathbf{x}$ . To investigate how the body deforms in response to a fluid flow around it, a horizontal fluid velocity of  $\mathbf{v}_f = (0.12, 0)$  is applied at  $x = \pm 2$ . At  $y = \pm 2$ , perfect slip (shear-traction free) boundary conditions are employed.

Fig. 8 shows six snapshots of the fluid flow around the rod as it deforms. Initially the bottom of the rod is passively transported with the fluid flow as can be seen in the snapshot at  $t = 1$ . Later, the rod begins to bend, and pressures build up at  $(x,y)=(0,0.8)$ . At  $t = 10$  the rod reaches maximum bending, before relaxing slightly into a stationary configuration at  $t = 25.0$ . The simulation took 125 s to carry out.

### 7.4. Fluid spinning a flexible rotor

In the final example, a flexible rotor is introduced, whose shape is a combination of two of the rods from the previous example, but with the internal corners rounded out. Initially the rotor and fluid are at rest, and the reference map is set to that  $\xi = \mathbf{x}$ .

The rotor is placed in flow by creating an inwards horizontal velocity of  $\mathbf{v}_f = (0.2, 0)$  for the boundary  $x = -2, y < 0$  and an outwards vertical velocity of  $\mathbf{v}_f = (0, 0.2)$  for the boundary  $y = 2, x > 0$ . All of the other boundaries have impermeable, slip boundary conditions.



**Fig. 9.** Snapshots of pressure for the test problem in which a flexible rotor begins to rotate due to fluid flow. The fine rectangular grid on the rod shows the contours of the  $\xi_1$  and  $\xi_2$  components of the reference map. The small white circle is a marker on one of the blades; by  $t=20$ , the rotor has rotated almost  $90^\circ$ . Arrows show the fluid velocity, with the green strip highlighting the part of the boundary where fluid is injected and the cyan strip highlighting the part of the boundary where fluid is removed. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

The rotor is anchored in the region  $|\mathbf{x}| < 0.4$  but is allowed to rotate freely. To do this, at each simulation step, the variables  $\omega$  and  $\theta$  are fit by solving the least squares problems

$$\begin{pmatrix} v_{f1} \\ v_{f2} \end{pmatrix} = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

for all gridpoints in the region  $|\mathbf{x}| < 0.5$ . Once these variables have been found, then the above formulae can be used to set  $\mathbf{v}_f$  and  $\xi$  in the anchored region so that they represent a rigidly rotating body. Snapshots of the pressure are shown in Fig. 9. Initially the rotor is pushed diagonally upwards and rightwards, but as the simulation progresses the rotor begins to spin. To simulate from  $t=0$  to  $t=25$  took 146 s to carry out.

### 8. Conservative form

Thus far, we have been dealing with the discretization generalized in Table 1. This scheme is not discretely conservative, and is not intended for situations invoking discontinuous solutions. Here, we present two ways to rewrite the system in a fashion that upholds discrete conservation.

#### 8.1. Approach 1: independent density evolution

Our entire system can be recast in a divergence form of the type

$$\mathbf{U}_t + \nabla \cdot \mathbf{f}(\mathbf{U}, \nabla \mathbf{U}) = \mathbf{s}(\mathbf{U}). \tag{29}$$

Specifically,

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho \xi \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} - \hat{\mathbf{T}}((\nabla \xi)^{-1}) \\ \rho \xi \otimes \mathbf{v} \end{pmatrix} = \begin{pmatrix} 0 \\ \rho \mathbf{g} \\ \mathbf{0} \end{pmatrix}. \tag{30}$$

These equations have the benefit of being relatively straightforward to implement under a finite-volume discretization, which by telescoping of the flux terms ensures discrete conservation. One potential drawback is that the density evolves independently from the deformation gradient. While this is to no avail analytically, discretization error over time can cause the density to differ from a discrete computation of  $\rho_0(\det \nabla \xi)$ .

In past studies, source terms have been used that penalize any such differences as in Miller and Colella (2001). Since  $\xi$  and not  $\mathbf{F}$  is our primitive variable here, it is unclear if a similar fix is available. But on the other hand, without  $\mathbf{F}$  primitive, it could be argued that this consistency condition is less crucial. Even so, we present a second way to rewrite and implement the system conservatively, that guarantees kinematic consistency between the density  $\rho$  and the deformation gradient.

## 8.2. Approach 2: density from deformation

To confront the previous issue, we describe a method that always defines density in terms of the motion, while simultaneously maintaining discrete conservation. The general idea is to take advantage of the fact that  $\xi$  is already an integrated quantity of the deformation, which can be used to relate boundary information directly to the total mass. Let the unstressed reference body have some uniform density  $\rho_0$ . After deformation, in the  $n$ -dimensional continuum limit, the mass within an Eulerian domain  $\Omega$  is equivalent to

$$\text{Mass}_\Omega = \int_\Omega \rho \, dV = \int_\Omega \rho_0 \det(\nabla \xi) \, dV = \rho_0 \int_{\Omega_R} dV_R = \int_{\partial\Omega} \frac{\rho_0}{n} \det(\nabla \xi) (\nabla \xi)^{-1} \xi \cdot d\mathbf{S} \quad (31)$$

where the subscript  $R$  denotes reference space variables. The last integral reflects that knowledge of  $\xi$  along the boundary of a domain is enough to correctly compute the mass contained within.

The integral form of conservation of mass can be rewritten using Eq. (31) in terms of a single boundary integral involving  $\xi$  and its derivatives. To wit:

$$\begin{aligned} 0 &= \frac{\partial}{\partial t} \left( \int_\Omega \rho \, dV \right) + \int_{\partial\Omega} \rho \mathbf{v} \cdot d\mathbf{S} \\ \Leftrightarrow 0 &= \int_{\partial\Omega} (\nabla \xi)^{-1} \rho_0 \det(\nabla \xi) (\xi_t + (\nabla \xi) \mathbf{v}) \cdot d\mathbf{S}. \end{aligned} \quad (32)$$

Observe that the term in big parentheses on the right of Eq. (32) is precisely the expression on the left side of Eq. (17). As a consequence, we deduce that if  $\nabla \xi$  and  $\mathbf{v}$  exist on  $\partial\Omega$  and the usual advection law for  $\xi$  is enforced along the boundary, then the integrand above vanishes and mass conservation is automatically satisfied within  $\Omega$ , regardless of the smoothness properties within the interior of  $\Omega$ .

Following the discretization in the Appendix, which includes a formal discrete conservation proof, we demonstrate the technique using an elastic constitutive law permitting a direct analogy to the 1D isothermal Euler equations, so that analytical shock solutions are readily available for the purposes of testing the method:

$$\hat{T}(V) = a\rho_0(1-V^{-1}) = a^2\rho_0(1-\partial_x \xi). \quad (33)$$

A genuinely nonlinear, discontinuous solution satisfying the Rankine–Hugoniot and entropy conditions is the rightward traveling shock

$$s = v_L - a\sqrt{1+\beta} \quad (34)$$

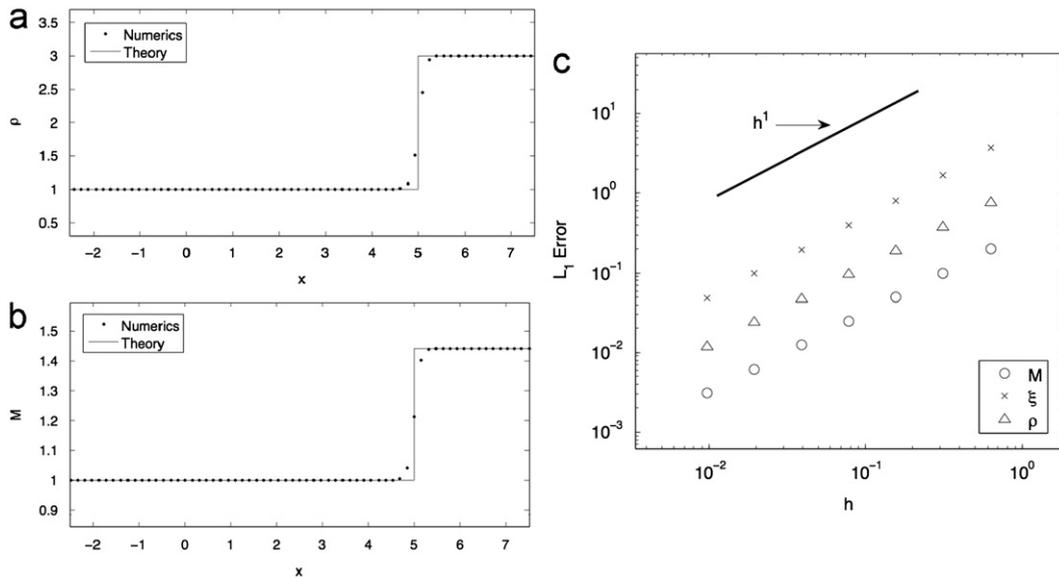
$$\rho(x < st, t) = \rho_L, \quad \rho(x \geq st, t) = \rho_L(\beta+1) = \rho_R \quad (35)$$

$$v(x < st, t) = v_L, \quad v(x \geq st, t) = ((v_L - a\sqrt{1+\beta})\beta + v_L)/(\beta+1) = v_R \quad (36)$$

for some  $0 < \beta < (v_L/a)^2 - 1$  and positive incoming velocity  $v_L$ .

Letting  $v_L = \rho_L = 1$  and  $\beta = 2$ , we presume a shock front initially at the origin of a material with  $a=0.45$  and  $\rho_0 = 1$  within the Eulerian domain  $x \in [-2.5, 7.5]$ . The discontinuous initial compression is instituted through a kinked  $\xi$  field defined by  $\xi(x < 0, 0) = \rho_L x$  and  $\xi(x \geq 0, 0) = \rho_R x$ . The initial momentum density field is  $M(x < 0, 0) = \rho_L v_L$  and  $M(x \geq 0, 0) = \rho_R v_R$ .

These initial conditions are discretized and the numerical method is computed up to time  $t_f = 5/(2s)$  permitting the shock to travel half the length of the domain. A transparent left boundary condition is used throughout, and the routine is tested on grids of size  $h = 10/2^k$  for  $k=4-10$ . Fig. 10 summarizes the findings. As expected from discrete conservation, plots (a) and (b) show that at  $t_f$  the discrete fields  $\rho$  and  $M$  possess a (blurred) shock front well-centered at the analytical position of the moving discontinuity. The plot in (c) illustrates first-order  $L_1$  convergence, which is expected given the order of the scheme and is indicative of the method's ability to capture the shock speed. Due to numerical diffusion about the jump, a standard concession in shock capturing schemes, higher  $L_p$  norms give a lower convergence rate for  $\rho$  and  $M$ ; namely  $1/p$ -order convergence. However,  $\xi$  remains first-order accurate because it is continuous.



**Fig. 10.** Solutions at the instant the shock has travelled half the length of the domain. Numerical vs analytical solution for the (a) mass density and (b) momentum density using a grid of 64 points. (c) Convergence plot of the various quantities illustrating first-order  $L_1$  convergence.

## 9. Conclusion and future work

This work has demonstrated and tested the Reference Map Technique for use in simulating solid deformation under a fully Eulerian, finite-difference framework, and extended the approach to a fast, simple fluid–solid interaction routine generalizable to many situations. The current work has tested the technique under various conditions. For statics, the solution it presents was compared against an analytical solution for the hyperelastic deformation of a washer, showing that the method approaches the correct solution under inhomogeneous deformation. The approach was expanded to quasi-static conditions, and the convergence properties were measured using a large-strain, grid-misaligned five-petal deformation. To test dynamics, the case of smooth motion was validated against an analytical solution, and two different approaches were proposed in the case of discontinuous dynamics. The preferred approach, which involves recovering the density from the deformation, was tested using an introductory 1D example. We have also provided a number of demonstrations of an extended routine to simulate sharp-interface fluid–structure interactions, including three examples where fluids induce large elastic solid deformations and vice versa. The simplicity of the finite-difference framework enables these simulations to run rather quickly.

There are several avenues of future investigation. We have begun to extend the approach to more solid behaviors, including those with added state variables such as rate-dependent/rate-independent hyperplasticity, which utilize a tensorial state variable  $\mathbf{F}^p$  (Lee, 1969) and possible evolving hardening parameters. In the RMT framework, state variables are appended as separate fields on the grid and evolved accordingly during the time step. Inclusion of thermal effects may also be handled this way, i.e. by adding temperature as a grid variable. By inclusion of an Eulerian projection step (Chorin, 1967), it is also possible to institute incompressible solid models. We also note that while we have primarily considered two-dimensional geometries for simplicity, it is conceptually straightforward to apply the method in fully three-dimensional geometries and results on this front are forthcoming. Regarding shock simulations, only a rudimentary first-order analysis has been provided herein, and it would be important to create high-order versions of this technique especially with regard to satisfaction of the entropy condition in more complex cases. A subroutine to institute traction boundary conditions is currently in the validation phase. In it, stress fields are extrapolated beyond the solid body and the normal components are adjusted to ensure boundary interpolation correctly represents the boundary traction. Details will be reported in a future work. As for the FSI routine, it remains to include subroutines for other interaction laws such as perfect slip, and, as general RMT capabilities expand, to test interactions between fluids and solids of various constitutive laws.

## Acknowledgments

K. Kamrin would like to acknowledge partial support from the NSF MSPRF. C. H. Rycroft acknowledges support by the Director, Office of Science, Computational and Technology Research, U.S. Department of Energy under contract number DE-AC02-05CH11231. J.-C. Nave would like to acknowledge partial support by the National Science Foundation under Grant DMS-0813648 and the NSERC Discovery Program.

## Appendix A

### A.1. Fluid/solid interfacial extrapolation

One of the key components of the fluid–solid coupling simulation presented in Section 7 is the extrapolation and subsequent fixing of simulation fields in order to apply the boundary conditions between the fluid and solid phases. The calculations presented in this section can be applied by considering the fields available at a single grid point. It is assumed that by making use of existing extrapolation methods (Aslam, 2004; Rycroft and Gibou, 2012) that linear extrapolations of all fields are available at the grid point – these are labeled with an “ex” superscript. In this appendix, the procedures are described to make use of these fields to construct ghost field values, labeled with a “g” superscript, which are then employed in the finite-difference update.

In step 3, the node-centered fields are considered. At fluid grid points, next to the solid body, the values of  $\xi^g$  are constructed by normally projecting the value of  $\xi^{\text{ex}}$  to be consistent with the level set function. The level set function  $\phi$  at the grid point gives the distance that this grid point is away from the solid boundary, and the value of  $\xi^g$  should be projected to be consistent with this. For example, for the simulation involving the circle with radius 0.8, it should be the case that at a fluid grid point, a distance  $\phi$  away from the boundary, then  $|\xi^g| = 0.8 + \phi$ . The ghost value can therefore be constructed according to

$$\xi^g = \left(\frac{4}{5} + \phi\right) \frac{\xi^{\text{ex}}}{|\xi^{\text{ex}}|}.$$

For the three simulations considered here, featuring objects comprised of arcs and lines, carrying out a normal projection of this type can be easily done. For a more general object, a more complex procedure would have to be employed perhaps based on the methods used in Pons et al. (2006).

To construct the ghost velocity, it is first necessary to construct a normalized tangential vector  $\mathbf{n}_\perp$  at the gridpoint. This can be done by calculating the gradient of the level set field and normalizing it, and then rotating it by  $90^\circ$ . After this, the tangential components of the velocity can be constructed according to

$$v_{s,t} = \mathbf{v}_s \cdot \mathbf{n}_\perp$$

$$v_{f,t} = \mathbf{v}_f \cdot \mathbf{n}_\perp.$$

After this, a ghost solid velocity can be constructed according to

$$\mathbf{v}_s^g = \mathbf{v}_f + \mathbf{n}_\perp (v_{s,t} - v_{f,t})$$

taking the real fluid velocity, and replacing the tangential component with the extrapolated solid velocity. Physically speaking, both velocity components are continuous across a no-slip interface, but the replacement above, consistent in the limit as  $\Delta x \rightarrow 0$ , improves the accuracy of the  $\nabla \mathbf{v}$  calculation (needed for velocity evolution) near the interface by using  $\mathbf{v}_t$  information from the solid rather than differencing over the very large kink in  $\mathbf{v}_t$  at the interface. At the solid grid points near the fluid, the ghost fluid velocity is taken to be the solid velocity, i.e.  $\mathbf{v}_f^g = \mathbf{v}_s$ , unadjusted because the fluid stress must arise from a strain-rate consistent with the no-slip condition.

In step 4, the cell-centered fields are considered. Consider a partial fluid gridpoint, where extrapolated stress tensors  $\mathbf{T}_s^{\text{ex}}$  and  $\mathbf{T}_f^{\text{ex}}$  are available. In a similar manner to above, it is possible to construct a normalized tangential vector  $\mathbf{n}_\perp$ . After that, the tangential–tangential components of the stress tensors can be evaluated as

$$\sigma_{s,tt}^{\text{ex}} = \mathbf{n}_\perp^T \cdot \mathbf{T}_s^{\text{ex}} \mathbf{n}_\perp$$

$$\sigma_{f,tt}^{\text{ex}} = \mathbf{n}_\perp^T \cdot \mathbf{T}_f^{\text{ex}} \mathbf{n}_\perp.$$

In order to allow the tangential–tangential component of the ghost solid stress tensor to be free, but to fix the other components to match across the boundary, the ghost solid stress tensor is constructed as

$$\mathbf{T}_s^g = \mathbf{T}_f^{\text{ex}} + \mathbf{n}_\perp \otimes \mathbf{n}_\perp (\sigma_{s,tt}^{\text{ex}} - \sigma_{f,tt}^{\text{ex}})$$

which replaces the tangential–tangential component of the fluid stress with the value from the solid stress. The same procedure can be applied at partial solid gridpoints in order to construct  $\mathbf{T}_f^g$  by making use of the extrapolated tangential–tangential component of the fluid stress, with the normal–normal and normal–tangential components of the solid stress.

In principle, these procedures offer a very general approach for handling a variety of boundary conditions. Depending on which fields are being set or being allowed to vary freely, different components of the fields can be replaced or fixed in the extrapolated fields.

A.2. Shock numerics

A conservative numerical scheme based on Eqs. (31), (17) and (2) is easiest to illustrate in one dimension. On an Eulerian domain  $\Omega = [x_L, x_R]$ , the total mass and the law of mass conservation reduce to

$$\text{Mass}_\Omega = \rho_0(\zeta(x_R) - \zeta(x_L)) \tag{37}$$

$$0 = \rho_0 \left( \zeta_t + \frac{\partial \zeta}{\partial x} v \right) \Big|_{x_R} - \rho_0 \left( \zeta_t + \frac{\partial \zeta}{\partial x} v \right) \Big|_{x_L}. \tag{38}$$

To demonstrate formal discrete conservation, we suppose the following circumstances as per (LeVeque, 1992). For all  $x \in \Omega$ ,  $\zeta(x,0)$  and  $v(x,0)$  are given analytically, and  $\partial_x \zeta(x,0)$  and  $v(x,0)$  are both constant in a set  $S = [x_L, x_L + d] \cup [x_R - d, x_R]$  for some  $d$ . Let  $\Delta x \ll d$  such that  $x_R - x_L = K\Delta x$ . At  $t=0$  we represent the field variables  $\zeta$  and momentum density  $M \equiv \rho_0 v \partial_x \zeta$  on a grid of points  $x = x_L + i\Delta x$  by assigning  $\zeta_i^0 = \zeta(x_L + i\Delta x, 0)$  and letting  $M_i^0$  be the average of  $M(x,0)$  over the interval  $x_L + [i-1/2, i+1/2]\Delta x$ . To determine  $M_i^j$  and  $\zeta_i^j$  at later times  $t = j\Delta t$ , we can use an explicit finite-difference scheme of the form outlined below:

$$\rho_{i-1/2}^j = \rho_0(\zeta_i^j - \zeta_{i-1}^j) / \Delta x \tag{39}$$

$$T_{i-1/2}^j = \hat{T}(((\zeta_i^j - \zeta_{i-1}^j) / \Delta x)^{-1}) \tag{40}$$

$$M_i^{j+1} = M_i^j - \frac{\Delta t}{\Delta x} (\phi_{i+1/2}^j - \phi_{i-1/2}^j) \tag{41}$$

$$\zeta_i^{j+1} = \zeta_i^j - \Delta t D_i^j M_i^j \tag{42}$$

where the fields  $\phi_{i+1/2}^j$  and  $D_i^j$  are computed from the discrete  $\rho$ ,  $T$ , and  $\zeta$  fields to give numerical approximations for  $(M^2/\rho - T)$  and  $(\partial_x \zeta)/\rho$  at locations  $x_L + (i+1/2)\Delta x$  and  $x_L + i\Delta x$  respectively. We require that the formulae for  $\phi_{i+1/2}^j$  and  $D_i^j$  depend only on data points positioned closer than  $l$  away for some  $l \ll d$ . The formulae must also be consistent such that the approximation error vanishes at a point  $x$  if  $\partial_x \zeta$  and  $M$  are both constant within  $[x-l, x+l]$ . For example, a simple upwind formula for rightward motion would be

$$\phi_{i+1/2}^n = (M_i^n)^2 / \rho_{i-1/2}^n - T_{i-1/2}^n \tag{43}$$

$$D_i^n = ((\zeta_i^n - \zeta_{i-1}^n) / \Delta x) / \rho_{i-1/2}^n = 1 / \rho_0. \tag{44}$$

The constraints above imply a finite domain of dependence. This combined with the fact that  $\partial_x \zeta$  and  $M$  begin constant within  $d$  of the domain's endpoints, means there exists some  $N < (d-l)/\Delta x$  such that for  $n \leq N$ , the numerical boundary terms exactly match their analytical counterparts due to consistency; notably  $\zeta_0^n = \zeta(x_L, n\Delta t)$  and  $\zeta_K^n = \zeta(x_R, n\Delta t)$  by consistency between Eqs. (42) and (17), which in turn guarantees Eq. (38). For similar reasons, the numerical flux  $\phi$  also remains exact at the endpoints up to time  $N\Delta t$ . These results imply

$$\text{Mass}_\Omega(t = N\Delta T) = \int_{x_L}^{x_R} \rho(x, N\Delta t) dx = \rho_0(\zeta(x_R, N\Delta t) - \zeta(x_L, N\Delta t)) = \rho_0(\zeta_K^N - \zeta_0^N) = \sum_{j=1}^K \rho_{j-1/2}^N \Delta x \tag{45}$$

$$\text{Momentum}_\Omega(t = N\Delta T) = \int_{x_L+1/2}^{x_R-1/2} M(x, N\Delta t) dx = \sum_{j=0}^K M_j^N \Delta x. \tag{46}$$

The last result applies a standard telescoping argument to the fluxes in Eq. (41) to establish equivalence to the integral.

Eqs. (45) and (46) mean that the discrete method is conservative as per the formal definition in LeVeque (1992), despite the fact that the density is not updated in a “standard” divergence form. In the canonical case of no mass flux through the system boundaries, then  $\zeta(x_L, t)$  and  $\zeta(x_R, t)$  remain fixed for all time and the computed mass per Eq. (45) is always guaranteed to equal the initial.

The same reasoning applies in higher dimensions as long as the density formula is defined from the discrete  $\zeta$  field in terms of the equivalent volume enclosed in the reference space, as was done here. For example, a density formula in 2D that would conserve mass as in Eq. (45), would be

$$\rho_{i-1/2, j-1/2}^k = \frac{\rho_0}{2\Delta x \Delta y} (|\zeta_{ij}^k - \zeta_{i-1, j}^k| \times |\zeta_{ij}^k - \zeta_{i, j-1}^k| + |\zeta_{i-1, j-1}^k - \zeta_{i-1, j}^k| \times |\zeta_{i-1, j-1}^k - \zeta_{i, j-1}^k|). \tag{47}$$

The term in big parenthesis represents the area of the quadrilateral in the reference space associated to the Eulerian grid square centered at  $(x_{i-1/2}, y_{j-1/2})$ . Under analogous conditions to the one-dimensional case, up to some finite time the sum of the discrete density field times  $\Delta x \Delta y$  gives precisely  $\rho_0 \times \{\text{reference area enclosed}\}$ , equivalent to the analytical total mass.

While our current interest is in athermal materials, we briefly note that the approach here could be augmented to handle temperature shocks by direct inclusion of a separate internal energy field  $E$  obeying some constitutive equation of state.

An explicit, conservative update law for  $E$  would follow from a divergence-form representation of energy conservation implemented in a fashion analogous to Eq. (41) for momentum conservation.

## References

- Aslam, T.D., 2004. A partial differential equation approach to multidimensional extrapolation. *J. Comp. Phys.* 193 (1), 349–355.
- Barton, P.T., Drikakis, D., Romenski, E.I., 2010. An Eulerian finite-volume scheme for large elastoplastic deformations in solids. *Int. J. Numer. Meth. Eng.* 81, 453–484.
- Bathe, K.J. (Ed.), 2007. *Proceedings of the Fourth MIT Conference on Computational Fluid and Solid Mechanics*. Elsevier Science.
- Belytschko, T., Liu, W.K., Moran, B., 2000. *Nonlinear Finite Elements for Continua and Structures*. John Wiley and Sons.
- Benson, D.J., 1995. A multi-material eulerian formulation for the efficient solution of impact and penetration problems. *Comput. Mech.* 15, 558–571.
- Chopp, D.L., 2001. Some improvements of the fast marching method. *SIAM J. Sci. Comput.* 23 (1), 230–244.
- Chorin, A.J., 1967. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.* 2, 12–26.
- Cottet, G.-H., Maitre, E., et Milcent, T., 2008. Eulerian formulation and level set models for incompressible fluid–structure interaction. *ESAIM-Math. Model. Numer. Anal.* 42 (3), 471–492.
- Fedkiw, R., Liu, X.-D., 1998. The ghost fluid method for viscous flows. *Progress in Numerical Solutions of Partial Differential Equations*. Arcachon, France.
- Gavrilyuk, S.L., Favrie, N., Saurel, R., 2008. Modelling wave dynamics of compressible elastic materials. *J. Comput. Phys.* 227, 2941–2969.
- Gurtin, M.E., Fried, E., Anand, L., 2010. *The Mechanics and Thermodynamics of Continua*. Cambridge University Press.
- Houghton, D.M., 1993. Circular shearing of compressible elastic cylinders. *Q. J. Mech. Appl. Math.* 46, 471–486.
- Hirt, C.W., Amsden, A.A., Cook, J.L., 1974. An arbitrary Lagrangian Eulerian computing method for all flow speeds. *J. Comput. Phys.* 14, 227–253.
- Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys.* 39, 201–225.
- Hoover, W.G., 2006. *Smooth Particle Applied Mechanics: The State of the Art*. World Scientific.
- Ken Kamrin, 2008. *Stochastic and Deterministic Models for Dense Granular Flow*. Ph.D. Thesis, MIT.
- Kojic, M., Bathe, K.-J., 1987. Studies of finite element procedures—stress solution of a closed elastic strain path with stretching and shearing using the updated lagrangian jaumann formulation. *Comput. Struct.* 26 (1–2), 175–179.
- Koopman, A.J., Geiselaers, H.J.M., Nilsen, K.E., Koenis, P.T.G., 2008. Numerical flow front tracking for aluminium extrusion of a tube and a comparison with experiments. *Int. J. Mater. Form. Suppl* 1, 423–426.
- Lee, E.H., 1969. Elastic plastic deformation at finite strain. *J. Appl. Mech.* 36, 1–6.
- LeVeque, R.J., 1992. *Numerical Methods for Conservation Laws*. Birkhauser Verlag.
- Liu, C., Walkington, N.J., 2001. An Eulerian description of fluids containing visco-elastic particles. *Arch. Rational Mech. Anal.* 159, 229–252.
- Liu, X.-D., Osher, S., Chan, T., 1994. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* 115, 200–212.
- Maitre, E., Milcent, T., Cottet, G.-H., Raoult, A., et Usson, Y., 2009. Applications of level set methods in computational biophysics. *Math. Comput. Model.* 49 (11–12), 2161–2169.
- Meyers, A., Xiao, H., Bruhns, O.T., 2006. Choice of objective rate in single parameter hypoelastic deformation cycles. *Comput. Struct.*, 1134–1140.
- Miller, G.H., Colella, P., 2001. A high-order Eulerian Godunov method for elastic–plastic flow in solids. *J. Comput. Phys.* 167, 131–176.
- Miller, G.H., Colella, P., 2002. A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing. *J. Comput. Phys.*, 26–82.
- Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* 79, 12–49.
- Peskin, C.S., 1977. Numerical analysis of blood flow in the heart. *J. Comput. Phys.* 25, 220–252.
- Plohr, B.J., Sharp, D.H., 1989. A conservative Eulerian formulation of the equations for elastic flow. *Adv. Appl. Math.* 9, 418.
- Pons, J.-P., Hermosillo, G., Keriven, R., Faugeras, O., 2006. Maintaining the point correspondence in the level set framework. *J. Comput. Phys.* 220 (1), 339–354.
- Rugonyi, S., Bathe, K.J., 2001. On finite element analysis of fluid flows fully coupled with structural interactions. *Comput. Model. Eng. Sci.* 2, 195–212.
- Rycroft, C.H., Gibou, F., 2012. Simulations of a stretching bar using a plasticity model from the shear transformation zone theory. *J. Comput. Phys.* 231, 2155–2179.
- 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.
- Shu, C.-W., Osher, S., 1988. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* 77, 439–471.
- Sulsky, D., Chen, Z., Schreyer, H.L., 1994. A particle method for history-dependent materials. *Comput. Meth. Appl. M.* 118, 179–196.
- Tannehill, J.C., Anderson, D.A., Pletcher, R.H., 1997. *Computational Fluid Mechanics and Heat Transfer*. Taylor and Francis.
- Tran, L.B., Udaykumar, H.S., 2004. A particle-level set-based sharp interface cartesian grid method for impact, penetration, and void collapse. *J. Comput. Phys.* 193, 2004.
- Trangenstein, J.A., Colella, P., 1991. A higher-order Godunov method for modeling finite deformation in elastic–plastic solids. *Comm. Pure Appl. Math.* 44, 41.
- Versteeg, H.K., Malalasekera, W., 1995. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Addison Wesley Longman Ltd.
- Wang, X., 2008. *Fluid–Solid Interaction*. Elsevier Science.
- Zienkiewicz, O.C., Taylor, R.L., 1967. *The Finite Element Method for Solid and Structural Mechanics*. McGraw Hill.