Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 208 (2005) 75–105
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems

Boyce E. Griffith *, Charles S. Peskin

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, United States

Received 1 November 2004; received in revised form 26 January 2005; accepted 6 February 2005
Available online 14 April 2005

Abstract

The immersed boundary method is both a mathematical formulation and a numerical scheme for problems involving the interaction of a viscous incompressible fluid and a (visco-)elastic structure. In [M.-C. Lai, Simulations of the flow past an array of circular cylinders as a test of the immersed boundary method, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, 1998; M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719], Lai and Peskin introduced a formally second order accurate immersed boundary method, but the convergence properties of their algorithm have only been examined computationally for problems with nonsmooth solutions. Consequently, in practice only first order convergence rates have been observed. In the present work, we describe a new formally second order accurate immersed boundary method and demonstrate its performance for a prototypical fluid–structure interaction problem, involving an immersed viscoelastic shell of finite thickness, studied over a broad range of Reynolds numbers. We consider two sets of material properties for the viscoelastic structure, including a case where the material properties of the coupled system are discontinuous at the fluid–structure interface. For both sets of material properties, the true solutions appear to possess sufficient smoothness for the method to converge at a second order rate for fully resolved computations.

MSC: 65M06; 73K15; 73V15; 76D05

Keywords: Immersed boundary method; Fluid–structure interaction; Incompressible flow; Viscoelasticity; Convergence

* Corresponding author. Tel.: +1 212 998 3079/3126; fax: +1 212 995 4121.
E-mail addresses: griffith@cims.nyu.edu (B.E. Griffith), peskin@cims.nyu.edu (C.S. Peskin).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.02.011

1. Introduction

Many problems of interest in biofluid mechanics involve the dynamic interaction of a viscous incompressible fluid and an elastic or viscoelastic structure. One approach to modeling and simulating such interactions is provided by the immersed boundary method [3–9]. In the immersed boundary formulation of such problems, the configuration of the elastic structure is described by Lagrangian variables (i.e., variables indexed by a coordinate system attached to the elastic structure), whereas the momentum, velocity, and incompressibility of the coupled fluid–structure system are described by Eulerian variables (i.e., in reference to fixed physical coordinates). In the continuous equations of motion, these two descriptions are connected by making use of the Dirac delta function, whereas a smoothed approximation to the delta function is used to link the Lagrangian and Eulerian descriptions when the continuous equations are discretely approximated for computer simulation.

A formally second order accurate version of the immersed boundary method was introduced in the Ph.D. thesis of Lai [1,2] – prior to this work, computations performed with the immersed boundary method typically employed a variety of genuinely first order accurate schemes. The second order accuracy of the method of Lai and Peskin is formal in the sense that second order convergence rates are expected only for problems where the true solution is sufficiently smooth. However, the rate of convergence of the immersed boundary method has almost always been assessed in situations where the true solutions do not possess enough regularity for the formal convergence rate to be attained.

In the present work, we introduce a new formally second order accurate version of the immersed boundary method and demonstrate *actual* second order numerical convergence rates for a prototypical fluid–structure interaction problem. In our computational convergence study, we consider the interaction between a viscous incompressible fluid and a viscoelastic shell (i.e., a body which, although thin, is not infinitely thin). The numerical performance of the method is examined over a broad range of Reynolds numbers for shells with two sets of elastic properties. For the first set of elastic properties, the stiffness of the shell tapers to zero at its edges, so that there is a continuous transition in material properties between the fluid and the structure. We also consider the case where the stiffness of the shell is constant, so that there is a sharp discontinuity in the material properties of the coupled system at the fluid–structure interface. At least at low and moderate Reynolds numbers, in each situation the true solution appears to be sufficiently regular for the numerical method to converge at its formal order of accuracy as the computational grids are refined.

To our knowledge this is the first time that convergence rates in excess of first order have been documented using the immersed boundary method, although higher order convergence rates have been observed for related methods [10–13]. Unlike the problem considered in the present work, previous convergence studies for the immersed boundary method have typically considered the case of an infinitely thin elastic membrane (i.e., an elastic boundary or interface). The analytic solutions to such interface problems possess discontinuities in the pressure and in derivatives of the velocity, and the immersed boundary method does not accurately capture these discontinuities. By considering the interaction between a viscoelastic shell and a viscous incompressible fluid, we avoid these difficulties and are able to obtain second order convergence rates.

The numerical scheme we present is essentially a refinement of the formally second order method of Lai and Peskin [2]. Several modifications are made to the method detailed in [2] in an attempt to reduce the occurrence of nonphysical oscillations in the computed dynamics. The simplest of these modifications is our use of a strong stability-preserving Runge–Kutta method [14] for the time integration of the Lagrangian equations of motion (i.e., the equations that specify the evolution of the configuration of the elastic structure).

We more drastically depart from [2] in our treatment of the Eulerian equations of motion, namely the incompressible Navier–Stokes equations. In the present work, the solution of these equations is by a

projection method that makes use of an implicit L-stable discretization of the viscous terms [15,16] and a second order Godunov method for the explicit treatment of the nonlinear advection terms [17–19]. Generally speaking, projection methods [20–22] are a class of fractional step algorithms for incompressible flow problems that update the velocity by first solving the momentum equation over a time interval without imposing the constraint of incompressibility. Doing so yields an ''intermediate'' velocity field that is generally not divergence free. The ''true'' updated velocity is then obtained by solving a Poisson problem to enforce the incompressibility constraint. Mathematically speaking, this process projects the intermediate velocity onto the space of divergence free vector fields.

When an ''exact'' projection method is used, the discrete divergence of the updated velocity is identically zero (to within roundoff error). In the present work, we employ a projection method that is not exact but rather is ''approximate'' in the sense that the discrete divergence of the velocity only *converges* to zero at a second order rate as the computational grid is refined. When such methods are used with the immersed boundary method, we have found that it is beneficial to determine the updated velocity and pressure in terms of the solutions to two *different* approximate projection equations at each timestep. This so-called hybrid approach was originally proposed by Almgren et al. for inviscid flow [23], and our approach is essentially an extension of their algorithm (''version 5'') to the viscous case. A more traditional projection method would employ only a single projection at each timestep. Consequently, when compared to more traditional projection algorithms, hybrid methods require the solution of additional systems of linear equations at each timestep, although this additional expense can be made modest. In the present context, we have found that the use of a more traditional projection method can result in spurious oscillations in the computed pressure. These oscillations, sometimes considered to be characteristic of the immersed boundary method [13], can be dramatically reduced by making use of the hybrid approach we present. Notably, this is an improvement that holds for thick structures (such as shells) as well as the more challenging thin interface case.

The remainder of the paper begins with a presentation of the immersed boundary formulation of the continuous fluid–structure interaction equations in Section 2. Formally second order accurate spatial and temporal discretizations of the continuous equations are then described in Section 3, although some important numerical details regarding our treatment of the nonlinear advective terms are postponed to an appendix. In Section 4, we verify that the scheme attains second order rates for a prototypical fluid–structure interaction problem, using two different sets of elastic material properties and several smoothed delta functions, and in Section 5, we demonstrate in the context of a thin interface problem that the hybrid projection method we employ reduces the magnitude of nonphysical pressure oscillations when compared to a more standard projection method. Conclusions and directions for future work are discussed in Section 6.

2. The continuous equations of motion

Consider a system comprised of a viscoelastic structure immersed in a viscous incompressible fluid. We assume that the fluid has uniform density, ρ, and uniform dynamic viscosity, μ. The structure is taken to be incompressible and neutrally buoyant, and the viscous properties of the structure are assumed to be identical to those of the fluid in which it is immersed. Consequently, the momentum, velocity, and incompressibility of the coupled system can be described via the incompressible Navier–Stokes equations, augmented by an appropriately defined body force. (Even in the more complicated case in which the mass density of the structure differs from that of the fluid, the momentum, velocity, and incompressibility of the coupled system can still be described by the incompressible Navier–Stokes equations; see [9]. The case in which the viscosity of the structure differs from that of the fluid can also presumably be done by a generalization of the methods proposed here, but this has not yet been attempted.)

The immersed boundary formulation of this problem employs an Eulerian description of the velocity and incompressibility of the fluid–structure system and a Lagrangian description of the configuration of the immersed elastic structure. In particular, the velocity of the entire coupled system is described in terms of an Eulerian velocity field, $\mathbf{u}(\mathbf{x}, t)$, where $\mathbf{x} = (x, y)$ are fixed physical (Cartesian) coordinates (i.e., $\mathbf{u}(\mathbf{x}, t)$ refers to the velocity of *whichever* material is physically located at position $\mathbf{x}$ at time t), whereas the configuration of the immersed elastic structure is described in terms of a curvilinear coordinate system. Let (r, s) be material curvilinear coordinates attached to the elastic structure so that fixed values of (r, s) label a material point for all time t, with $\mathbf{X}(r, s, t)$ referring to the Cartesian position of such a material point at time t. The physical domain consists of a region $U \subset \mathbb{R}^2$. For simplicity, we take U to be the unit square and impose periodic boundary conditions. The curvilinear coordinates are restricted to some region of (r, s)-space, here denoted $\Omega \subset \mathbb{R}^2$. The configuration of the elastic structure at time t is denoted by $\mathbf{X}(\cdot, \cdot, t)$, and the curvilinear force density (i.e., the density with respect to (r, s)) generated by the elasticity of the structure is taken to be a known functional of this configuration.

The equations of motion for the system can be written in the following form:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p = \mu\nabla^2\mathbf{u} + \mathbf{f}, \tag{1}$$

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

$$\mathbf{f}(\mathbf{x}, t) = \int_\Omega \mathbf{F}(r, s, t)\delta(\mathbf{x} - \mathbf{X}(r, s, t))\,\mathrm{d}r\,\mathrm{d}s, \tag{3}$$

$$\frac{\partial \mathbf{X}}{\partial t}(r, s, t) = \mathbf{u}(\mathbf{X}(r, s, t), t) = \int_U \mathbf{u}(\mathbf{x}, t)\delta(\mathbf{x} - \mathbf{X}(r, s, t))\,\mathrm{d}\mathbf{x}, \tag{4}$$

$$\mathbf{F}(\cdot, \cdot, t) = \mathscr{F}[\mathbf{X}(\cdot, \cdot, t)]. \tag{5}$$

Eqs. (1) and (2) are the incompressible Navier–Stokes equations written in Eulerian form, where $p(\mathbf{x}, t)$ is the pressure and $\mathbf{f}(\mathbf{x}, t)$ is the (Cartesian) elastic force density. Eq. (5) formalizes the assumption that the curvilinear elastic force density, $\mathbf{F}(\cdot, \cdot, t)$, is a given functional of the structure configuration, $\mathbf{X}(\cdot, \cdot, t)$. A generalization that we do not consider here would be to allow $\mathscr{F}$ to be a time-dependent functional.

Eqs. (3) and (4) describe the interaction between the Lagrangian and Eulerian variables. The two-dimensional Dirac delta function, $\delta(\mathbf{x}) = \delta(x)\delta(y)$, appears in both of these equations. In each case, it acts as a kernel of an integral transform that facilitates conversions between Eulerian and Lagrangian quantities. Eq. (3) converts the curvilinear force density into the Cartesian force density. Note that their numerical values are generally not equal at corresponding points. Nevertheless, the Cartesian and curvilinear elastic force densities are equivalent *as densities*. Recalling the defining property of the Dirac delta function,

$$\int_V \delta(\mathbf{x} - \mathbf{X})\,\mathrm{d}\mathbf{x} = \begin{cases} 1 & \text{if } \mathbf{X} \in V, \\ 0 & \text{otherwise,} \end{cases}$$

where $V \subset U$ is an arbitrary region of physical space, we see that the densities are indeed equivalent via

$$\begin{aligned}\int_V \mathbf{f}(\mathbf{x}, t)\,\mathrm{d}\mathbf{x} &= \int_V \int_\Omega \mathbf{F}(r, s, t)\delta(\mathbf{x} - \mathbf{X}(r, s, t))\,\mathrm{d}r\,\mathrm{d}s\,\mathrm{d}\mathbf{x} = \int_\Omega \mathbf{F}(r, s, t)\left(\int_V \delta(\mathbf{x} - \mathbf{X}(r, s, t))\,\mathrm{d}\mathbf{x}\right)\mathrm{d}r\,\mathrm{d}s \\ &= \int_{\mathbf{X}^{-1}(V,t)} \mathbf{F}(r, s, t)\,\mathrm{d}r\,\mathrm{d}s,\end{aligned}$$

where

$$\mathbf{X}^{-1}(V,t) = \{(r,s)\,|\,\mathbf{X}(r,s,t) \in V\}.$$

The second of the interaction equations, Eq. (4), relates the material velocity of the elastic structure to the Eulerian velocity field for the coupled system. Since $\mathbf{u}(\mathbf{x}, t)$ is the velocity of whichever material is physically located at position $\mathbf{x}$ at time t, for any $(r, s) \in \Omega$,

$$\frac{\partial \mathbf{X}}{\partial t}(r,s,t) = \mathbf{u}(\mathbf{X}(r,s,t),t).$$

We may evaluate the velocity at $\mathbf{X}(r, s, t)$ by making use of the delta function,

$$\mathbf{u}(\mathbf{X}(r,s,t),t) = \int_U \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(r,s,t))\,\mathrm{d}\mathbf{x},$$

so long as $\mathbf{u}$ is continuous. For the coupled system, continuity of the velocity follows from the presence of viscosity in both the fluid and the structure.

Before concluding this section, we mention the particular elastic force density functional that will be used in this work. Suppose that the immersed elastic structure consists of a collection of elastic fibers, where the material coordinates (r, s) have been chosen so that a fixed value of r labels a particular fiber for all time. Let $\boldsymbol{\tau}$ denote the unit tangent vector in the fiber direction,

$$\boldsymbol{\tau} = \frac{\partial \mathbf{X}/\partial s}{|\partial \mathbf{X}/\partial s|}. \tag{6}$$

Since the fibers are elastic, the fiber tension, T, is related to the fiber strain, which is determined by $|\partial \mathbf{X}/\partial s|$. The fiber tension can be expressed by a generalized Hooke's law of the form

$$T = \sigma(|\partial \mathbf{X}/\partial s|; r, s). \tag{7}$$

One can show [5,9] that the corresponding curvilinear elastic force density functional can be put in the form

$$\mathscr{F}[\mathbf{X}(\cdot,\cdot,t)] = \frac{\partial}{\partial s}(T\boldsymbol{\tau}). \tag{8}$$

Since T and $\boldsymbol{\tau}$ are both defined in terms of $\partial \mathbf{X}/\partial s$, $\mathscr{F}$ is a functional that maps the structure configuration to the curvilinear force density, $\mathbf{F}(\cdot, \cdot, t)$.

3. The discrete equations of motion

3.1. Lagrangian and Eulerian discretizations

In the immersed boundary approach to fluid–structure interaction problems, the solution to the continuous equations of motion, (1)–(5), is approximated by discretizing the Eulerian equations on a Cartesian grid and by discretizing the Lagrangian equations on a discrete lattice in the curvilinear coordinate space. In most work using the immersed boundary method, these discretizations are fixed throughout the computation [5,24–26]. However, there has been work on utilizing structured adaptive mesh refinement in the solution of the Eulerian equations [27].

In the present work, the physical domain is taken to be the periodic unit square. It is described using a fixed Cartesian grid with uniform meshwidths $h = \Delta x = \Delta y$. The centers of the Cartesian grid cells are the points $\mathbf{x}_{i,j} = \left((i + \frac{1}{2})h, (j + \frac{1}{2})h\right)$, where $i, j \in \{0, 1, \ldots, N - 1\}$ and $h = 1/N$.

For a quantity $\psi(\mathbf{x}, t)$ defined on the Cartesian grid, we employ the notation $\psi^n_{i,j} \equiv \psi(\mathbf{x}_{i,j}, t_n)$, where t_n is the time of the nth timestep. The timestep size is implicitly defined by $\Delta t_n = t_{n+1} - t_n$, although our convergence studies will employ a fixed uniform timestep Δt. Note that quantities are occasionally defined at "half-timesteps", $t_{n+\frac{1}{2}} = t_n + \frac{1}{2}\Delta t_n$.

The curvilinear coordinate space is discretized on a fixed lattice in (r, s)-space with uniform meshwidths $(\Delta r, \Delta s)$. Unless otherwise noted, from now on the curvilinear coordinate indices (r, s) will refer to the "nodes" of the curvilinear computational lattice, so that $(r, s) = (r_0, s_0) + (m\Delta r, n\Delta s)$ for fixed constants r_0 and s_0 and integer values of m and n.

Although the discretization of the curvilinear coordinate space is fixed, it is important to note that the physical locations of the nodes of the curvilinear mesh, $\mathbf{X}^n(r, s) \equiv \mathbf{X}(r, s, t_n)$, are free to move throughout the physical domain. In particular, the physical positions of the nodes of the curvilinear mesh are in no way required to conform to the Cartesian grid.

3.2. Cartesian grid finite difference operators

We now introduce finite difference approximations to the spatial differential operators appearing in the Eulerian equations of motion, beginning with the discretization of the divergence and gradient operators. Both approximations employ second order accurate centered differences. The divergence of a vector field, $\mathbf{u} = (u, v)$, is approximated by

$$(\mathbf{D} \cdot \mathbf{u})_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + \frac{v_{i,j+1} - v_{i,j-1}}{2h}, \tag{9}$$

and the gradient of a scalar function, ψ, is approximated by

$$(\mathbf{G}\psi)_{i,j} = \left(\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h}, \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2h}\right). \tag{10}$$

The Laplacian of ψ is approximated using standard second order accurate finite differences, denoted by

$$(L\psi)_{i,j} = \frac{\psi_{i+1,j} + \psi_{i-1,j} - 2\psi_{i,j}}{h^2} + \frac{\psi_{i,j+1} + \psi_{i,j-1} - 2\psi_{i,j}}{h^2}. \tag{11}$$

Note that L does *not* equal $\mathbf{D} \cdot \mathbf{G}$.

The discrete vector Laplacian of a vector field is $(L\mathbf{u})_{i,j} = ((Lu)_{i,j}, (Lv)_{i,j})$, where $\mathbf{u} = (u, v)$. It is simply the application of the discrete scalar Laplacian to the individual components of $\mathbf{u}$.

3.3. The discrete approximate projection operator

Like all projection-type methods for incompressible flow, our method for the incompressible Navier–Stokes equations makes use of the Hodge decomposition theorem. This result says that an arbitrary smooth vector field can be uniquely defined as the sum of a divergence free vector field and the gradient of a scalar function. The discrete analog of this decomposition is

$$\mathbf{w} = \mathbf{v} + \mathbf{G}\varphi, \tag{12}$$

where $\mathbf{w}$ is an arbitrary vector field on the Cartesian grid and $\mathbf{v}$ satisfies $(\mathbf{D} \cdot \mathbf{v})_{i,j} \equiv 0$ on the grid. Eq. (12) implicitly defines a projection operator, P, given by

$$\mathbf{v} = P\mathbf{w} = (I - \mathbf{G}(\mathbf{D} \cdot \mathbf{G})^{-1}\mathbf{D}\cdot)\mathbf{w}. \tag{13}$$

Since $(\mathbf{D} \cdot \mathbf{v})_{i,j} \equiv 0$, for any vector field $\mathbf{w}$, $P^2\mathbf{w} = P\mathbf{w}$, so P is a projection.

In practice, the application of the operator defined by Eq. (13) requires the solution of a system of linear equations of the form $\mathbf{D}\cdot\mathbf{G}\varphi = \mathbf{D}\cdot\mathbf{w}$. For a two-dimensional periodic grid with an even number of grid cells in each direction, $\mathbf{D}\cdot\mathbf{G}$ has a nontrivial four-dimensional nullspace. This complicates the solution process when iterative methods (such as multigrid) are employed to solve for φ. Moreover, when P is used in the solution of the incompressible Navier–Stokes equations, this nontrivial nullspace results in the decoupling of pressure field on four sub-grids, leading to a so-called "checkerboard" instability.

To avoid these difficulties, it was originally proposed in [28] that the foregoing exact projection be replaced by a carefully chosen "approximate" projection operator. In the present work, we use a cell centered approximate projection operator of a type first introduced by Lai [29] (see also [30]). This approximate projection operator, $\tilde{P}$, is defined by

$$\tilde{P}\mathbf{w} = (I - \mathbf{G}(L)^{-1}\mathbf{D}\cdot)\mathbf{w}. \tag{14}$$

It is important to note that this operator is *not* a projection, since $L \neq \mathbf{D}\cdot\mathbf{G}$. However, L and $\mathbf{D}\cdot\mathbf{G}$ agree to second order in h so that for smooth $\mathbf{w}$, $\mathbf{D}\cdot\tilde{P}\mathbf{w} = \mathcal{O}(h^2)$. Moreover, $\|P\mathbf{w} - \tilde{P}\mathbf{w}\| \to 0$ as $h \to 0$. On a uniform grid with periodic boundary conditions, it can be demonstrated that $\|\tilde{P}\mathbf{w}\| \leqslant \|\mathbf{w}\|$, so the cell centered approximate projection operator is stable [29]. Another important issue with regards to the stability of the overall method is the question of which quantity is to be (approximately) projected [23]; we address this issue below in Section 3.6.

3.4. A discrete curvilinear force density

The continuous version of the elastic force density functional that we employ in the present work is given by Eqs. (6)–(8). To approximate this functional on the curvilinear computational lattice, we introduce a finite difference approximation to differentiation in the s curvilinear coordinate direction, defined by

$$(D_s\Psi)(r,s) = \frac{\Psi\left(r, s+\frac{1}{2}\Delta s\right) - \Psi\left(r, s-\frac{1}{2}\Delta s\right)}{\Delta s}, \tag{15}$$

where $\Psi(r,s)$ is a function defined on the curvilinear computational lattice.

Given a structure configuration, $\mathbf{X}$, the unit tangent vector, (6), is approximated at "half-integer" multiples of Δs by

$$\boldsymbol{\tau}\left(r, s+\frac{1}{2}\Delta s\right) = \frac{(D_s\mathbf{X})(r, s+\frac{1}{2}\Delta s)}{\left|(D_s\mathbf{X})(r, s+\frac{1}{2}\Delta s)\right|}. \tag{16}$$

Similarly, the fiber tension, (7), is approximated by

$$T\left(r, s+\frac{1}{2}\Delta s\right) = \sigma\left(\left|(D_s\mathbf{X})\left(r, s+\frac{1}{2}\Delta s\right)\right|; r, s+\frac{1}{2}\Delta s\right). \tag{17}$$

Finally, Eqs. (16) and (17) may be used to approximate the curvilinear elastic force density, (8), at integer multiples of Δs by

$$\mathbf{F}(r,s) = (D_s(T\boldsymbol{\tau}))(r,s). \tag{18}$$

Note that the half-integer multiples of Δs that appear in the foregoing are only intermediate values. In the end, the evaluation of (18) at the nodes of the curvilinear computational lattice requires only the values of $\mathbf{X}(r,s)$ at the nodes of the curvilinear computational lattice, i.e., for $(r,s) = (r_0,s_0) + (m\Delta r, n\Delta s)$ for fixed constants r_0 and s_0 and for integer values of m and n.

3.5. Smoothed versions of the Dirac delta function

In its treatment of the interaction equations that connect the Lagrangian and Eulerian frames, the immersed boundary method makes use of a smoothed approximation to the Dirac delta function. In the computational results below, we will employ several such functions, though our choices are by no means exhaustive. In each case, the smoothed delta function, denoted $\delta_h(\mathbf{x})$, is of the form

$$\delta_h(\mathbf{x}) = \frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right), \tag{19}$$

recalling that $\mathbf{x} = (x, y)$ and that $h = \Delta x = \Delta y$. Our particular choices for $\phi(r)$ are displayed in Fig. 1 and are defined presently.

In [9], a collection of axioms (including moment conditions and a quadratic condition) are described that lead to the unique definition of a particular smoothed delta function with finite support. A family of such functions may be generated by imposing additional moment conditions and correspondingly broadening the support. The first member of this family, the so-called four-point delta function, $\delta_{4h}^{\mathrm{IB}}(\mathbf{x})$, satisfies two discrete moment conditions with a support of four meshwidths in each spatial dimension (i.e., a support of a total of 16 grid cells in two dimensions). It is defined in terms of the function $\phi_4^{\mathrm{IB}}(r)$, where

$$\phi_4^{\mathrm{IB}}(r) = \begin{cases} \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right), & 0 \leqslant |r| < 1, \\ \frac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}\right), & 1 \leqslant |r| < 2, \\ 0, & 2 \leqslant |r|. \end{cases} \tag{20}$$

By requiring the smoothed delta function to satisfy two additional discrete moment conditions, a six-point delta function is obtained. This delta function, first employed by Stockie [31], is denoted $\delta_{6h}^{\mathrm{IB}}(\mathbf{x})$ and is defined in terms of

$$\phi_6^{\mathrm{IB}}(r) = \begin{cases} \frac{61}{112} - \frac{11}{42}|r| - \frac{11}{56}|r|^2 + \frac{1}{12}|r|^3 + \frac{\sqrt{3}}{336}(243 + 1584|r| - 748|r|^2 & \\ \quad -1560|r|^3 + 500|r|^4 + 336|r|^5 - 112|r|^6)^{\frac{1}{2}}, & 0 \leqslant |r| < 1, \\ \frac{21}{16} + \frac{7}{12}|r| - \frac{7}{8}|r|^2 + \frac{1}{6}|r|^3 - \frac{3}{2}\phi_6^{\mathrm{IB}}(|r| - 1), & 1 \leqslant |r| < 2, \\ \frac{9}{8} - \frac{23}{12}|r| + \frac{3}{4}|r|^2 - \frac{1}{12}|r|^3 + \frac{1}{2}\phi_6^{\mathrm{IB}}(|r| - 2), & 2 \leqslant |r| < 3, \\ 0, & 3 \leqslant |r|. \end{cases} \tag{21}$$

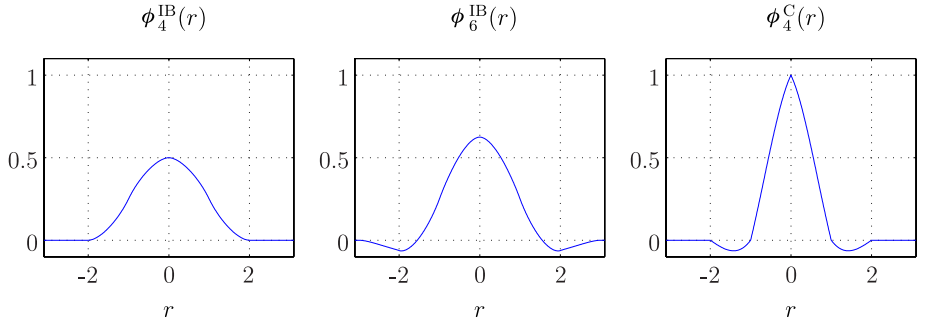

Fig. 1. Three choices among many for $\phi(r)$ when constructing a smoothed approximation to the Dirac delta function using Eq. (19). These functions are defined by Eqs. (20)–(22), respectively.

In [32], Tornberg and Engquist examine the use of smoothed delta functions in the regularization of singular source terms in a variety of settings. One smoothed delta function that provides them with particularly good results is a piecewise cubic function, $\delta_{4h}^{\mathrm{C}}(\mathbf{x})$, defined in terms of

$$\phi_4^{\mathrm{C}}(r) = \begin{cases} 1 - \frac{1}{2}|r| - |r|^2 + \frac{1}{2}|r|^3, & 0 \leqslant |r| < 1, \\ 1 - \frac{11}{6}|r| + |r|^2 - \frac{1}{6}|r|^3, & 1 \leqslant |r| < 2, \\ 0, & 2 \leqslant |r|. \end{cases} \tag{22}$$

This smoothed delta function has a support of four meshwidths in each spatial dimension and satisfies four moment conditions, but it does not satisfy all of the axioms prescribed in [9]. In practice, it is also less costly to compute than the other delta functions considered as it does not require the evaluation of any square roots.

3.6. Timestepping

At the beginning of timestep n, we possess approximations to the values of the state variables at time t_n, namely $\mathbf{u}^n$ and $\mathbf{X}^n$. The pressure (which is not a state variable) must be defined at half-timesteps to obtain a consistent second order accurate method. Thus, at the beginning of each timestep $n > 0$, we also possess an approximation to a "time-lagged" pressure, $p^{n-\frac{1}{2}}$.

To advance the solution forward in time by the increment Δt, we first compute $\mathbf{X}^{(n+1)}(r, s)$, a *preliminary* approximation to the locations of the nodes of the curvilinear mesh at time t_{n+1}. To do so, Eq. (4) is approximated by

$$\mathbf{X}^{(n+1)}(r,s) = \mathbf{X}^n(r,s) + \Delta t \sum_{i,j} \mathbf{u}_{i,j}^n \delta_h\big(\mathbf{x}_{i,j} - \mathbf{X}^n(r,s)\big)h^2. \tag{23}$$

A discrete approximation to $\mathscr{F}[\mathbf{X}(\cdot,\cdot)]$ provides the curvilinear elastic force densities corresponding to structure configurations $\mathbf{X}^n$ and $\mathbf{X}^{(n+1)}$, respectively denoted $\mathbf{F}^n$ and $\mathbf{F}^{(n+1)}$. The equivalent Cartesian elastic force densities are obtained by discretizing (3) and are given by

$$\mathbf{f}_{i,j}^n = \sum_{r,s} \mathbf{F}^n(r,s)\delta_h\big(\mathbf{x}_{i,j} - \mathbf{X}^n(r,s)\big)\Delta r \Delta s, \tag{24}$$

$$\mathbf{f}_{i,j}^{(n+1)} = \sum_{r,s} \mathbf{F}^{(n+1)}(r,s)\delta_h\big(\mathbf{x}_{i,j} - \mathbf{X}^{(n+1)}(r,s)\big)\Delta r \Delta s. \tag{25}$$

A timestep-centered approximation to the Cartesian elastic force density is defined by

$$\mathbf{f}^{n+\frac{1}{2}} \equiv \frac{1}{2}(\mathbf{f}^n + \mathbf{f}^{(n+1)}). \tag{26}$$

We next determine $\mathbf{u}^{n+1}$ and $p^{n+\frac{1}{2}}$ by integrating the incompressible Navier–Stokes equations in time via a second order projection method similar to the method introduced by Bell, Colella, and Glaz [22], a method that in turn is a second order accurate version of Chorin's original projection method [20,21]. Our algorithm extends to the viscous case the hybrid approximate projection method ("version 5") introduced by Almgren et al. for the incompressible Euler equations [23]. In particular, as in [23], we obtain the values of $\mathbf{u}^{n+1}$ and $p^{n+\frac{1}{2}}$ in terms of the solutions to *different* projection equations, as follows.

Given $\mathbf{u}^n$, $\mathbf{f}^{n+\frac{1}{2}}$, and $p^{n-\frac{1}{2}}$, we first obtain the approximation to the updated velocity, $\mathbf{u}^{n+1}$. We do so by discretizing the momentum equation (1) over the time interval Δt without imposing the constraint of incompressibility on $\mathbf{u}^{n+1}$. Instead, the gradient of the time-lagged pressure provides an approximation to the true pressure gradient. The nonlinear advection term is treated explicitly, and a version of the implicit L-stable

method of Twizell et al. [15] introduced by McCorquodale et al. [16] is used to integrate the viscous terms in time. With $\nu \equiv \mu/\rho$, the discretization of (1) is

$$(I - \eta_2 \nu L)(I - \eta_1 \nu L)\mathbf{u}^* = (I + \eta_3 \nu L)\mathbf{u}^n + \Delta t (I + \eta_4 \nu L)\left(-\mathbf{N}^{n+\frac{1}{2}} + \frac{1}{\rho}\left(\mathbf{f}^{n+\frac{1}{2}} - \mathbf{G}p^{n-\frac{1}{2}}\right)\right), \tag{27}$$

where $\mathbf{N}^{n+\frac{1}{2}}$ is the explicit approximation to $[(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+\frac{1}{2}}$ detailed in Appendix A, and

$$\eta_1 = \frac{a - \sqrt{a^2 - 4a + 2}}{2}\Delta t, \quad \eta_2 = \frac{a + \sqrt{a^2 - 4a + 2}}{2}\Delta t,$$

$$\eta_3 = (1 - a)\Delta t, \quad \eta_4 = \left(\frac{1}{2} - a\right)\Delta t,$$

with $a = 2 - \sqrt{2} - \epsilon$, where ϵ is machine precision.

The solution to Eq. (27) yields an "intermediate" velocity field, traditionally denoted $\mathbf{u}^*$, that is generally not discretely divergence free. In formulating a projection method, one may project either the velocity increment (i.e., $\mathbf{u}^{n+1} - \mathbf{u}^*$) or the intermediate velocity itself. Although either choice yields the same value for $\mathbf{u}^{n+1}$ when exact projections are employed, this is not the case when approximate projection operators are used. Several studies have found that a more stable algorithm is obtained by approximately projecting the intermediate velocity [23,30], and we follow this approach. In particular, $\mathbf{u}^{n+1}$ is obtained by making use of the approximate projection operator, $\tilde{P}$, defined by Eq. (14), yielding

$$\mathbf{u}^{n+1} = \tilde{P}\mathbf{u}^*. \tag{28}$$

Although it is possible to determine the value of the updated pressure in terms of the approximate projection of $\mathbf{u}^*$, we have found that it is beneficial to determine $p^{n+\frac{1}{2}}$ by approximately projecting a second intermediate velocity field that is given by a second treatment of the momentum equation. This alternate treatment of (1) is identical to (27) except that it does not include *any* approximation to the pressure gradient, i.e.,

$$(I - \eta_2 \nu L)(I - \eta_1 \nu L)\tilde{\mathbf{u}}^* = (I + \eta_3 \nu L)\mathbf{u}^n + \Delta t (I + \eta_4 \nu L)\left(-\mathbf{N}^{n+\frac{1}{2}} + \frac{1}{\rho}\mathbf{f}^{n+\frac{1}{2}}\right). \tag{29}$$

The solution to this equation, $\tilde{\mathbf{u}}^*$, is the intermediate velocity that we project to obtain $p^{n+\frac{1}{2}}$. We emphasize that $\tilde{\mathbf{u}}^*$ is *only* used to compute $p^{n+\frac{1}{2}}$ and is not used in determining our final approximation to the velocity at time t_{n+1}. The approximate projection of $\tilde{\mathbf{u}}^*$, however, generates an *alternate* approximation to the velocity at time t_{n+1},

$$\tilde{\mathbf{u}}^{n+1} = \tilde{P}\tilde{\mathbf{u}}^* = \tilde{\mathbf{u}}^* - \mathbf{G}\tilde{\varphi},$$

i.e.

$$\tilde{\mathbf{u}}^* = \tilde{\mathbf{u}}^{n+1} + \mathbf{G}\tilde{\varphi}, \tag{30}$$

where $\tilde{\varphi}$ is defined as the solution to a discrete Poisson problem,

$$L\tilde{\varphi} = \mathbf{D}\cdot\tilde{\mathbf{u}}^*.$$

Since $\tilde{P}$ is an approximate projection operator, in general $\tilde{\mathbf{u}}^{n+1} \neq \mathbf{u}^{n+1}$.

The pressure consistent with (29) and (30) is the scalar function $p^{n+\frac{1}{2}}$ that satisfies

$$(I - \eta_2 \nu L)(I - \eta_1 \nu L)\tilde{\mathbf{u}}^{n+1} = (I + \eta_3 \nu L)\mathbf{u}^n + \Delta t (I + \eta_4 \nu L)\left(-\mathbf{N}^{n+\frac{1}{2}} + \frac{1}{\rho}\left(\mathbf{f}^{n+\frac{1}{2}} - \mathbf{G}p^{n+\frac{1}{2}}\right)\right). \tag{31}$$

Following [33], a second order accurate approximation to the updated pressure is determined by substituting (30) back into (29) and comparing the result to (31). Doing so, the discrete pressure gradient is seen to satisfy

$$(I + \eta_4 vL)\mathbf{G}p^{n+\frac{1}{2}} = \frac{\rho}{\Delta t}(I - \eta_2 vL)(I - \eta_1 vL)\mathbf{G}\tilde{\varphi}.$$

Consequently, we obtain $p^{n+\frac{1}{2}}$ via

$$p^{n+\frac{1}{2}} = \frac{\rho}{\Delta t}(I + \eta_4 vL)^{-1}(I - \eta_2 vL)(I - \eta_1 vL)\tilde{\varphi}. \tag{32}$$

Since $\eta_4 = (\sqrt{2} - \frac{3}{2} + \epsilon)\Delta t < 0$, $p^{n+\frac{1}{2}}$ is well defined by (32). Note that $\tilde{\varphi}$ is proportional to a first order accurate approximation to the time centered pressure. Full second order accuracy is obtained by solving a system of linear equations in (32). Although $p^{n+\frac{1}{2}}$ has no influence on the value obtained for $\mathbf{u}^{n+1}$, it is used in the next timestep, when computing $\mathbf{u}^{n+2}$.

Having obtained the values $\mathbf{u}^{n+1}$ and $p^{n+\frac{1}{2}}$, we complete the timestep by computing $\mathbf{X}^{n+1}$ via

$$\mathbf{X}^{n+1}(r,s) = \mathbf{X}^n(r,s) + \frac{\Delta t}{2}\left(\sum_{i,j}\mathbf{u}^n_{i,j}\delta_h\big(\mathbf{x}_{i,j} - \mathbf{X}^n(r,s)\big)h^2 + \sum_{i,j}\mathbf{u}^{n+1}_{i,j}\delta_h(\mathbf{x}_{i,j} - \mathbf{X}^{(n+1)}(r,s))h^2\right). \tag{33}$$

Note that the evolution of the structure configuration via (23) and (33) takes the form of a second order accurate strong stability-preserving Runge–Kutta method [14]. Eq. (33) is an explicit formula for $\mathbf{X}^{n+1}$, since $\mathbf{X}^{(n+1)}$ is already defined; see Eq. (23).

Finally, we must discuss the initial timestep. The initial state of the system is completely determined by the initial values of $\mathbf{u}$ and $\mathbf{X}$. To ensure that the initial velocity at least *approximately* satisfies $(\mathbf{D}\cdot\mathbf{u}^0)_{i,j} \equiv 0$, we first replace $\mathbf{u}^0$ by its approximate projection,

$$\mathbf{u}^0 \leftarrow \tilde{P}\mathbf{u}^0. \tag{34}$$

Next, the pressure must be determined from the values of $\mathbf{u}^0$ and $\mathbf{X}^0$. We obtain the pressure iteratively as follows. First, the pressure is provisionally set to be identically zero. We then perform a preliminary timestep. The computation of this preliminary timestep yields a first approximation to the pressure at time $t = \frac{1}{2}\Delta t_0$. We then iteratively re-compute the initial timestep, always using the most recent approximation to the pressure. After a small number of iterations (we use a total of five), we obtain a sufficiently accurate approximation to the pressure at time $t = \frac{1}{2}\Delta t_0$ to achieve overall second order accuracy.

3.7. Implementation issues

Before concluding this section, there are a few issues we wish to address regarding our software implementation of the foregoing algorithm. The implementation of the present version of the immersed boundary method makes extensive use of the SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) library, a C++ framework for developing parallel scientific applications which provides support for block structured adaptive mesh refinement (AMR) [34,35]. Although our implementation allows us to make use of the parallel and AMR capabilities provided by SAMRAI, we do not do so in the present work since our focus is on the order of accuracy of the immersed boundary method itself and not on further enhancements to the basic method.

In our implementation, SAMRAI provides a mechanism for organizing computations on the Cartesian grid. To manage Lagrangian quantities defined on the curvilinear mesh (i.e., $\mathbf{X}$ and $\mathbf{F}$), we additionally make use of PETSc (the Portable, Extensible Toolkit for Scientific Computation) [36–38]. All numerical quantities defined on the curvilinear mesh are stored in PETSc vectors, allowing for parallel communication of data on the curvilinear mesh. Additionally, by making use of the linear and nonlinear solvers provided

by PETSc, it is straightforward to solve systems of equations on the curvilinear mesh, potentially facilitating the implementation of efficient fully implicit time discretizations for the immersed boundary method.

At each timestep, it is necessary to compute the solution to several discrete Poisson and Helmholtz type systems of linear equations. The Helmholtz problems we encounter are generally well conditioned, and it is possible to solve them efficiently via red–black Gauss–Seidel iterations. The particular solver we use for these problems is provided by SAMRAI. The Poisson problems are more challenging and require a more sophisticated solution method. For these problems, we make use of the structured multigrid solvers provided by the *hypre* project, a library of high performance preconditioners [39,40]. (See [41] for an introduction to multigrid methods that includes a description of red–black Gauss–Seidel.)

4. Computational convergence results for smooth test problems

Typically, the convergence of the immersed boundary method has been studied computationally for problems which do not possess a sufficient degree of smoothness for the method to attain its formal convergence rate. In particular, most previous convergence studies have focused on the case of a viscous incompressible fluid interacting with an infinitely thin elastic membrane (i.e., an elastic interface or boundary). The true solutions to such problems possess discontinuities at the interface in the pressure and in the normal derivative of the velocity, and these discontinuities are not accurately captured by the immersed boundary method. An alternative approach is taken by the immersed interface method [10,13], where such discontinuities are explicitly accounted for by the method in a manner that yields higher order accuracy.

In order to assess the performance of the present scheme in a setting where we can expect convergence rates that correspond to the formal order of accuracy of the method, we consider the interaction between a viscous incompressible fluid and a viscoelastic shell of finite thickness. Since the shell is thin but not infinitely thin, the discontinuities present in the true interface problem do not arise in this situation. The elastic properties of the shell are described in terms of a continuum of anisotropic elastic fibers. In Section 4.1, we specify the stiffness of the fibers so that the fiber tension smoothly tends to zero at the edges of the shell. As long as the structure does not become too distorted, the resulting Cartesian elastic force density, $\mathbf{f}$, will be a continuous function of $\mathbf{x}$. In Section 4.2, the fiber tension is taken to be a constant multiple of $|\partial \mathbf{X}/\partial s|$. In this case, the resulting Cartesian elastic force density is only piecewise continuous due to the sharp discontinuity in material properties that occurs at the fluid–structure interface. For low and moderate Reynolds number flows, we observe second order convergence rates in *both* situations. At higher Reynolds numbers, it appears that under-resolution of the velocity prevents the method from attaining full second order convergence rates, although in all cases empirical convergence rates in excess of first order are observed. Presumably, second order convergence would be observed even for the high Reynolds number cases on sufficiently fine grids.

Before proceeding to the specification of the two sets of elastic properties in Sections 4.1 and 4.2 and the corresponding computational results, we first describe the common aspects for both sets of computations, including the computational meshes used to describe the Cartesian and curvilinear coordinate spaces, the initial conditions, and the choice of timestep.

Recall that the physical domain, U, is specified to be the periodic unit square and is described using a fixed $N \times N$ Cartesian grid with uniform meshwidths $h = \Delta x = \Delta y$, where $h = 1/N$. For this computational study, we likewise take the curvilinear coordinate space to be $\Omega = [0, 1] \times [0, 1]$. We employ a fixed $N_r \times N_s$ computational lattice in the curvilinear coordinate space, where $N_r = \frac{3}{8}N$ and $N_s = \frac{75}{16}N$. With $\Delta r = 1/N_R$ and $\Delta s = 1/N_s$, the "nodes" of the curvilinear mesh are the points

$$(r, s) = (r_0, s_0) + (m\Delta r, n\Delta s) = \left(\frac{\Delta r}{2}, \frac{\Delta s}{2}\right) + (m\Delta r, n\Delta s),$$

where $m \in \{0, 1, \ldots, N_r - 1\}$ and $n \in \{0, 1, \ldots, N_s - 1\}$. Here, the shift by $\frac{\Delta r}{2}$ avoids having fibers at the exact edges of the shell, while the shift by $\frac{\Delta s}{2}$ is for notational consistency only.

As discussed in Sections 2 and 3.4, the elastic force density generated by the structure configuration is defined in terms of a continuum of elastic fibers, and the curvilinear coordinates, (r, s), are chosen so that a fixed value of r labels a particular fiber for all time t. So that each fiber forms a closed loop, Ω is taken to be periodic in the s-coordinate direction. Note that s is *not* equal to arc length along the fibers. For this particular Lagrangian elastic force density functional, no boundary conditions are imposed in the r-coordinate direction. The initial configuration of the viscoelastic body is given by the mapping

$$\mathbf{X}(r, s, 0) = \left(\frac{1}{2}, \frac{1}{2}\right) + \left(\left(\alpha + \gamma\left(r - \frac{1}{2}\right)\right)\cos(2\pi s), \left(\beta + \gamma\left(r - \frac{1}{2}\right)\right)\sin(2\pi s)\right), \tag{35}$$

with $\alpha = 0.2$, $\beta = 0.25$, and $\gamma = 0.0625$. This mapping defines the initial configuration of each fiber to be an ellipse, so that the initial configuration of the entire structure is a "thick" elliptical shell. The value of γ determines the thickness of the shell and is chosen so that the thickness of the initial configuration is approximately four Cartesian meshwidths on the coarsest grid considered.

In all computations, the initial velocity of the system is taken to be $\mathbf{u}(\mathbf{x}, 0) \equiv 0$. After being released at $t = 0$, the shell undergoes damped oscillations and tends toward its resting configuration, a circular shell. The uniform density of the fluid–structure system is taken to be $\rho = 1.0$, and the uniform viscosity, μ, is successively assigned the values 0.05, 0.005, and 0.0005. Using the fiber tensions specified in either Section 4.1 or Section 4.2, the corresponding flows have Reynolds numbers of approximately 10, 100, and 1000.

The computation is halted and convergence is assessed at $t = 0.4$. For each parameter regime considered, the shell will have approximately completed its first oscillation at this point in the computation. In all cases, we employ a uniform timestep that is chosen so that the computed velocity satisfies

$$\Delta t \|\mathbf{u}\|_\infty < 0.1h. \tag{36}$$

This is a more severe restriction than our explicit treatment of the nonlinear advection term requires; however, since we are treating the elastic force density in an explicit manner, the hyperbolic stability restriction is not the only stability constraint that the timestep must satisfy. Although (36) may not be sufficient to ensure stability in the limit as $h \to 0$, it appears to be adequate for the values of h considered here.

For most of the following computations, Eq. (36) is satisfied for $\Delta t = 0.08/N$, and this choice generally appears to result in stable computations for the grid spacings and parameter ranges we consider. The sole exception is for the computations in Section 4.2 that employ the piecewise cubic delta function for the case $\mu = 0.0005$. We found that in under-resolved computations, the piecewise cubic delta function can introduce strong oscillations in the computed velocity near the fluid–structure interface. In this particular case, we found that it is sufficient to reduce the size of the timestep to $\Delta t = 0.04/N$ in order to satisfy (36).

Below, we present empirical convergence rates for u, v, p, and $\mathbf{X}$ in appropriately defined discrete L^p norms for $p = 1, 2$. The discrete L^p norm of a scalar valued function defined on the Cartesian grid, ψ, is given by

$$\|\psi_{i,j}\|_p = \left(\sum_{i,j} |\psi_{i,j}|^p h^2\right)^{1/p}.$$

For a vector valued function defined on the curvilinear mesh, $\mathbf{W}(r, s) = (W_1(r, s), W_2(r, s))$, the discrete L^p norm is likewise

$$\|\mathbf{W}_{i,j}\|_p = \left(\sum_{r,s} \left|W_1^2(r, s) + W_2^2(r, s)\right|^{p/2} \Delta r \Delta s\right)^{1/p}.$$

For a computed quantity, q, let $e_p[q;N]$ denote the discrete L^p norm of the difference in the approximation to q obtained using an $N\times N$ Cartesian grid (and the corresponding curvilinear mesh and uniform timestep) and the approximation to q obtained using a $2N\times 2N$ Cartesian grid (and the corresponding curvilinear mesh and uniform timestep), i.e.,

$$e_p[q;N] = \|q^N - \mathcal{I}^{2N\to N} q^{2N}\|_p, \tag{37}$$

where $\mathcal{I}^{2N\to N}$ denotes interpolation from finer to coarser spatial grids. An empirical estimate for the convergence rate of q in this norm is given by

$$r_p[q;N] = \log_2\left(\frac{e_p[q;N]}{e_p[q;2N]}\right). \tag{38}$$

4.1. Tapered elastic stiffness

In the first set of computations, we set the fiber tension, T, via

$$T = \sigma(|\partial\mathbf{X}/\partial s|;r,s) = (1+\sin(2\pi r - \pi/2))|\partial\mathbf{X}/\partial s|.$$

Recalling Eqs. (6)–(8), the resulting Lagrangian elastic force density is given by

$$\mathbf{F} = \frac{\partial}{\partial s}(T\boldsymbol{\tau}) = (1+\sin(2\pi r - \pi/2))\frac{\partial^2\mathbf{X}}{\partial s^2}.$$

In the absence of sharp corners in the elastic fibers that comprise the shell, this Lagrangian force density smoothly tapers to zero as r approaches 0 or 1, i.e., there is a *continuous* transition in material properties at the fluid–structure interface. As long as the structure does not become too distorted, the resulting Cartesian elastic force density, $\mathbf{f}$, will remain a continuous function of $\mathbf{x}$.

Table 1 summarizes the empirical convergence rates in the discrete L^1 and L^2 norms observed for u, v, p, and $\mathbf{X}$ at time $t = 0.4$. These rates are obtained via (38) with $N = 128$. Fig. 2 includes the values of $e_2[q;N]$ that were used to obtain these rates. The values for $e_1[q;N]$ are similar and not shown. Second order convergence rates are observed for $\mu = 0.05$ and 0.005 in nearly all quantities. For $\mu = 0.0005$, somewhat less

Table 1
Empirical convergence rates for u, v, p, and $\mathbf{X}$ in the discrete L^1 and L^2 norms at time $t = 0.4$

δ_h	q	$\mu = 0.05$		$\mu = 0.005$		$\mu = 0.0005$	
		$r_1[q;128]$	$r_2[q;128]$	$r_1[q;128]$	$r_2[q;128]$	$r_1[q;128]$	$r_2[q;128]$
$\delta_{4h}^{\mathrm{IB}}$	u	2.15	2.16	2.11	2.20	0.77	0.51
	v	2.12	2.15	2.08	2.12	0.47	0.34
	p	2.00	1.89	2.20	1.86	3.76	3.32
	$\mathbf{X}$	2.13	1.98	1.82	1.74	2.04	1.79
$\delta_{6h}^{\mathrm{IB}}$	u	2.10	2.08	2.13	2.14	2.63	2.86
	v	2.21	2.33	2.36	2.45	2.83	3.25
	p	3.37	3.57	2.72	3.01	2.16	2.08
	$\mathbf{X}$	2.60	2.35	2.32	2.03	1.47	1.16
δ_{4h}^{C}	u	2.15	2.13	2.14	2.20	2.19	2.15
	v	2.28	2.45	2.31	2.52	2.25	2.49
	p	3.51	3.83	2.77	2.97	2.38	2.37
	$\mathbf{X}$	2.72	2.49	2.45	2.17	1.66	1.52

In these computations, the stiffness of the elastic fibers comprising the shell tapers to zero at the edges of the structure, so that there is a continuous transition in material properties at the fluid–structure interface. Convergence rates are obtained via Eqs. (37) and (38).

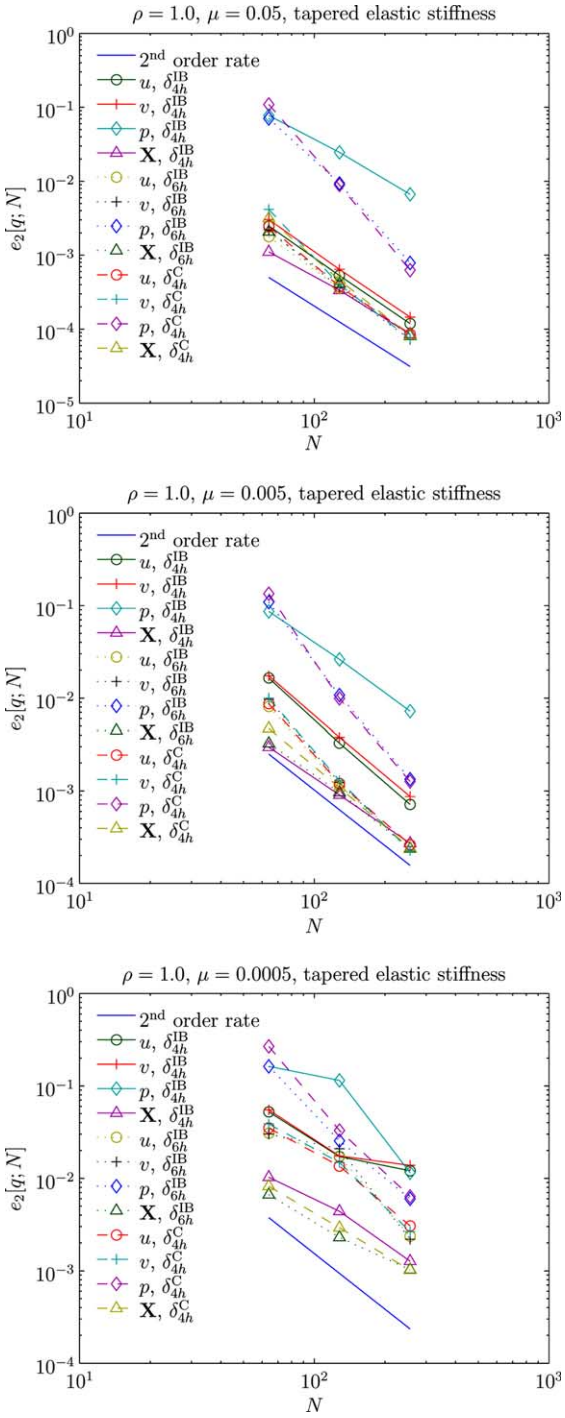

Fig. 2. The discrete L^2 norm of the difference in the values computed for an $N \times N$ Cartesian grid (with the corresponding curvilinear mesh and uniform timestep) and for a $2N \times 2N$ Cartesian grid is plotted at $t = 0.4$ for $N = 64$, 128, and 256, for u, v, p, and $\mathbf{X}$. For these computations, the stiffness of the elastic fibers comprising the shell tapers to zero at the edges of the structure. Second order convergence is generally indicated except for $\mu = 0.0005$. See also Table 1.

than second order accuracy is observed in some variables, especially for $\delta_{4h}^{\mathrm{IB}}$. This is likely due to inadequate resolution of the flow for $N = 128$ in the vicinity of narrow bands of high vorticity that form near the fluid–structure interface.

As shown in Table 1, when we employ delta functions which satisfy four moment conditions (i.e., δ_{4h}^{C} and $\delta_{6h}^{\mathrm{IB}}$) we observe convergence rates for the Eulerian quantities that are in excess of second order. In particular, third order convergence rates are observed for the pressure in several cases. One possible reason for these high rates may be the rapid dampening of oscillations in the computed pressure as N increases. Recall that both δ_{4h}^{C} and $\delta_{6h}^{\mathrm{IB}}$ possess negative "tails" (see Fig. 1). For the coarser grid computations, these negative tails appear to induce oscillations in the computed Eulerian quantities near the fluid–structure interface. As the spatial resolution is increased, these oscillations rapidly die out, possibly resulting in somewhat inflated convergence rates.

Representative results for this particular set of material properties are displayed in the left-hand columns of Figs. 4, 5, and 6, corresponding to $t = 0.08$, 0.2, and 0.32, respectively. These computed values were obtained using the six-point delta function, $\delta_{6h}^{\mathrm{IB}}$, for $\mu = 0.005$ and $N = 512$, although similar results were obtained for the other delta functions.

4.2. Constant elastic stiffness

In the second set of computations, we set the fiber tension, T, via

$$T = |\partial \mathbf{X}/\partial s|,$$

i.e., the stiffness of the fibers does *not* taper to zero at the edge of the shell. Recalling Eqs. (6)–(8), the resulting Lagrangian elastic force density is given by

$$\mathbf{F} = \frac{\partial}{\partial s}(T\boldsymbol{\tau}) = \frac{\partial^2 \mathbf{X}}{\partial s^2}.$$

In this case, the Cartesian elastic force density is only a piecewise continuous function of $\mathbf{x}$ due to a *sharp* transition in material properties at the fluid–structure interface.

Table 2
Empirical convergence rates for u, v, p, and $\mathbf{X}$ in the discrete L^1 and L^2 norms at time $t = 0.4$

δ_h	q	$\mu = 0.05$		$\mu = 0.005$		$\mu = 0.0005$	
		$r_1[q; 128]$	$r_2[q; 128]$	$r_1[q; 128]$	$r_2[q; 128]$	$r_1[q; 128]$	$r_2[q; 128]$
$\delta_{4h}^{\mathrm{IB}}$	u	1.78	1.81	1.75	1.80	1.73	1.84
	v	1.81	1.83	1.73	1.76	1.80	2.00
	p	1.94	1.48	1.79	1.65	1.59	1.51
	$\mathbf{X}$	2.00	1.83	1.71	1.67	1.14	1.05
$\delta_{6h}^{\mathrm{IB}}$	u	1.80	1.80	1.64	1.64	1.94	1.64
	v	1.86	1.86	1.60	1.56	2.02	1.77
	p	2.02	1.55	1.83	1.56	1.92	1.75
	$\mathbf{X}$	2.41	2.05	2.05	1.77	1.51	1.10
δ_{4h}^{C}	u	1.90	1.91	1.50	1.52	1.50	1.17
	v	1.98	2.03	1.48	1.57	1.61	1.41
	p	2.29	1.90	1.77	1.63	1.13	0.88
	$\mathbf{X}$	2.60	2.19	1.91	1.65	1.11	0.87

In these computations, the stiffness of the elastic fibers comprising the shell is constant throughout the structure, so that there is a sharp transition in material properties at the fluid–structure interface. Convergence rates are obtained via Eqs. (37) and (38).

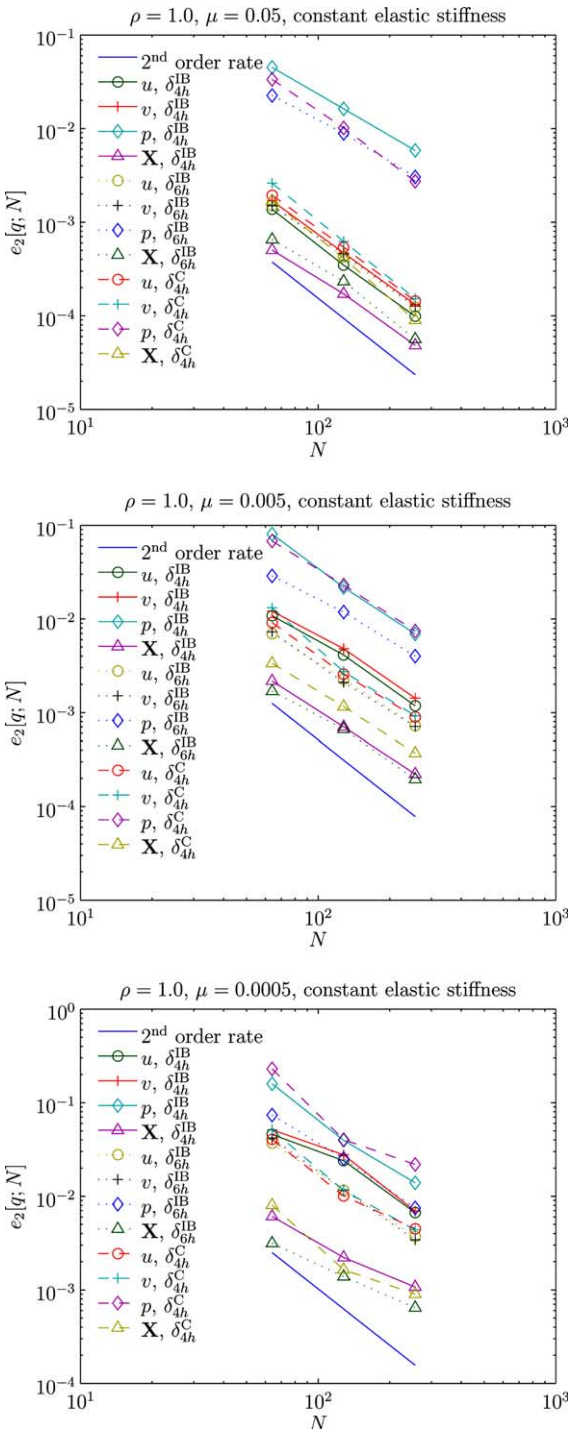

Fig. 3. Same as Fig. 2, except for these computations, the stiffness of the elastic fibers comprising the shell is constant throughout the structure, yielding a sharp transition in material properties at the fluid–structure interface. Second order convergence is indicated for $\mu = 0.05$, although somewhat less than second order rates are observed in the other cases. See also Table 2.

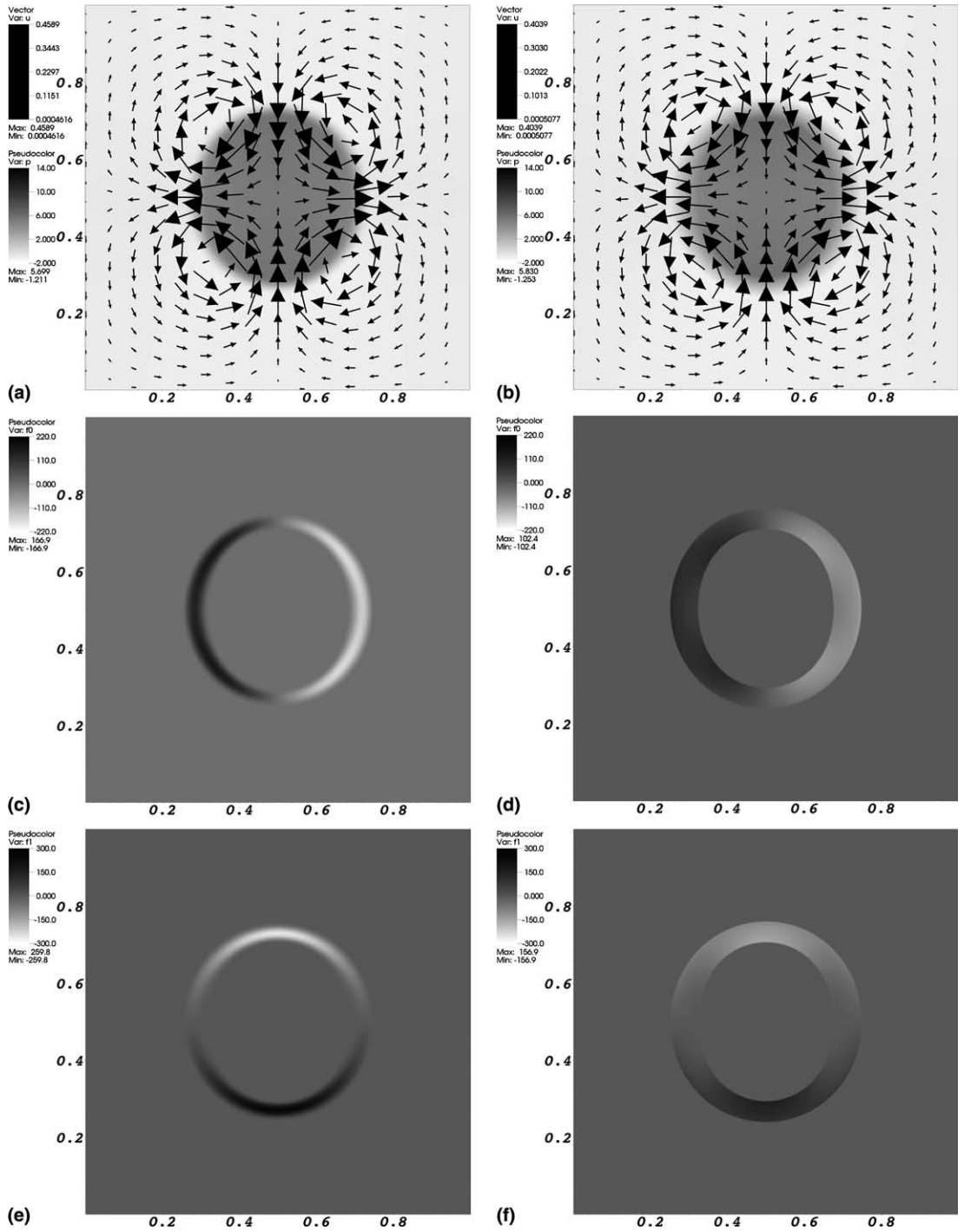

Fig. 4. Computed values of **u**, p, and **f** for a shell with tapered (left-hand column) and constant (right-hand column) elastic stiffnesses, displayed at $t = 0.08$. The velocity and pressure are displayed in the top row, whereas the x- and y-components of **f** are displayed in the middle and bottom row, respectively. For these computations, we use $\delta_{6h}^{\mathrm{IB}}$ with $\rho = 1.0$, $\mu = 0.005$, and $N = 512$.

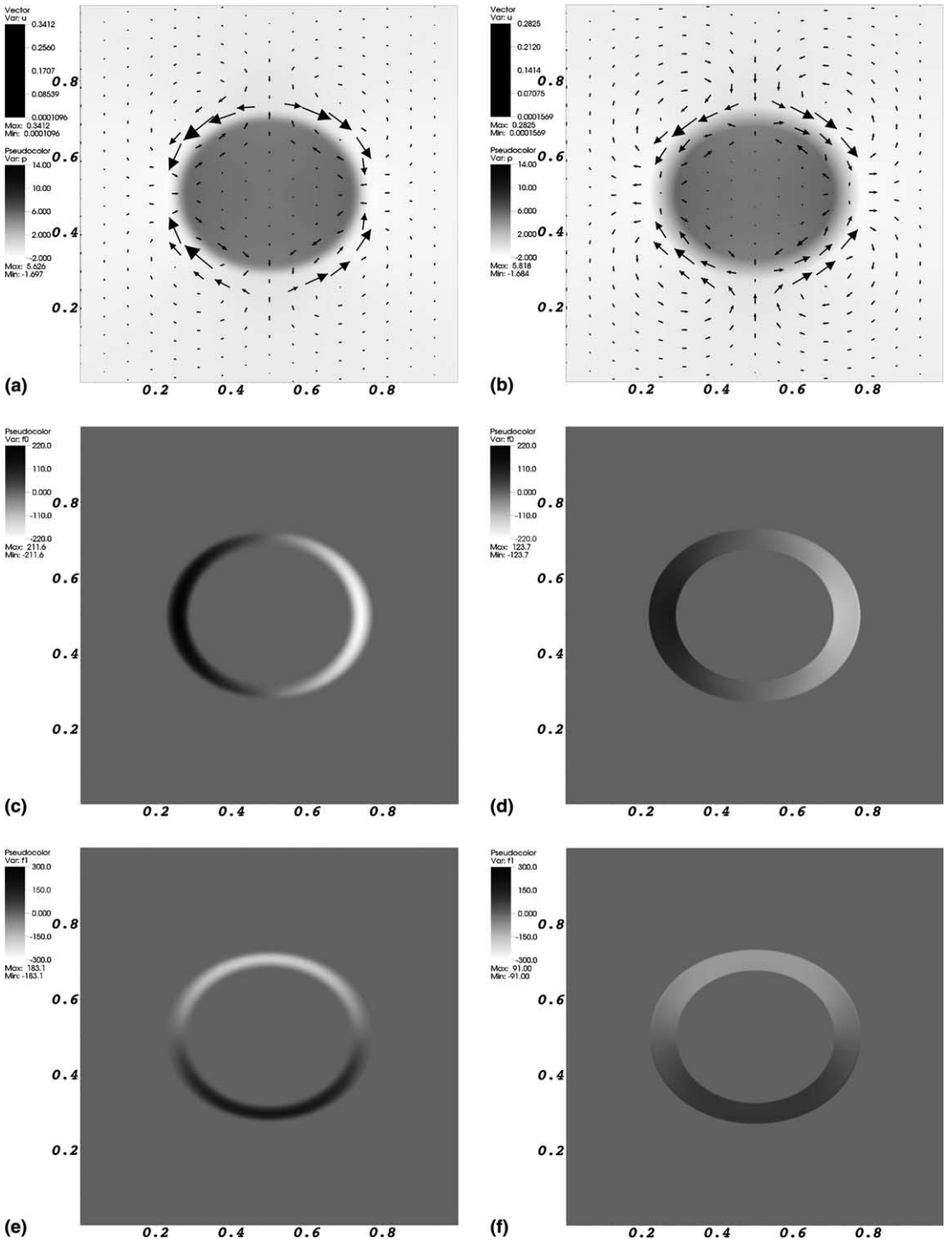

Fig. 5. Data as in Fig. 4, except here displayed at $t = 0.2$.

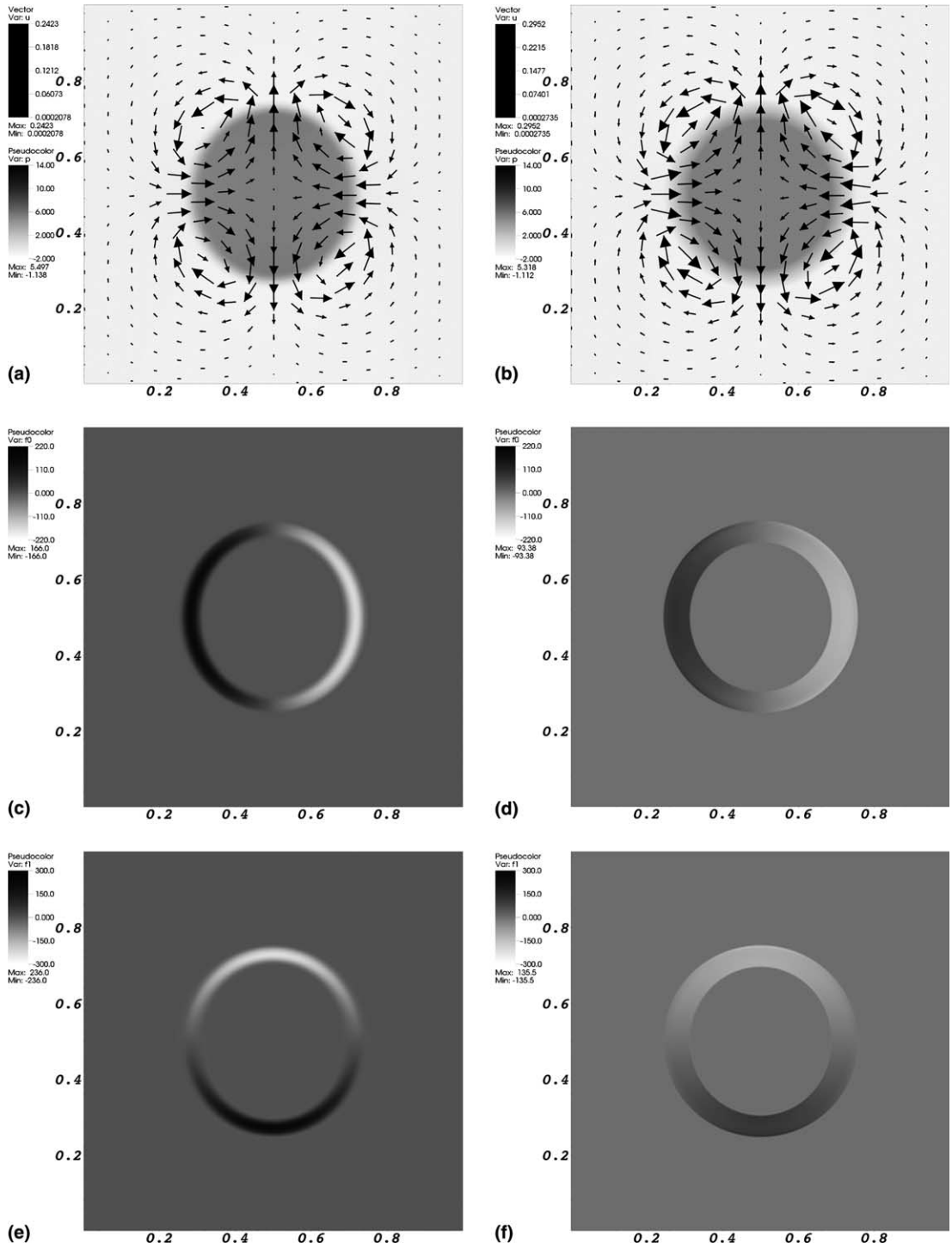

Fig. 6. Data as in Fig. 4, except here displayed at $t = 0.32$.

Note that when the piecewise cubic delta function, δ_{4h}^{C}, is used with $\mu = 0.0005$, we set $\Delta t = 0.04/N$ in order to satisfy the constraint (36). In all other cases, $\Delta t = 0.08/N$.

For this particular choice of material properties, Table 2 summarizes the empirical L^1 and L^2 norm convergence rates observed for u, v, p, and $\mathbf{X}$ at $t = 0.4$. These rates are obtained via (38) with $N = 128$. Fig. 3 includes the values of $e_2[q; N]$ that were used to obtain these rates. The values for $e_1[q; N]$ are similar and not shown. In this case, the empirically observed convergence rates are generally not as high as when the stiffness of the elastic fibers comprising the shell is tapered near the edge of the structure. This is not surprising since, unlike the tapered case, the Cartesian force density in this case is in fact discontinuous. Nonetheless, convergence rates at or near second order are generally indicated for $\mu = 0.05$. For the other values of μ considered, it is possible that second order convergence rates are not observed due to the under-resolution of the velocity in the vicinity of very narrow vorticity layers that form near the fluid–structure interface. Note that these vorticity layers appear well resolved for $N = 256$ and 512 but not for $N = 128$.

Representative results for this choice of elastic properties are displayed in the right-hand columns of Figs. 4, 5, and 6, corresponding to $t = 0.08$, 0.2, and 0.32, respectively. These computed values were obtained using the six-point delta function, $\delta_{6h}^{\mathrm{IB}}$, for $\mu = 0.005$ and $N = 512$, although similar results were obtained for the other delta functions.

5. Hybrid approximate projection methods

Historically, projection methods have generally used the solution to a single projection equation at each timestep to determine both the updated velocity and the updated pressure (see [20–22], among many others). One alternative approach is to define the updated pressure in terms of a projection that is different from that used to obtain the updated velocity. When exact projection operators (i.e., projections that exactly enforce the discrete incompressibility of the updated velocity to machine precision) are used with periodic computational domains, there is generally no reason to employ an additional projection, since the value of the computed velocity, $\mathbf{u}^{n+1}$, is unaffected by the approximation made to the true pressure gradient used to obtain the intermediate velocity, $\mathbf{u}^*$. This is because the approximation to the true pressure gradient used to obtain the intermediate velocity is *exactly* removed from $\mathbf{u}^*$ by the exact projection. Even when physical (i.e., nonperiodic) boundaries are present, the approximation to the true pressure gradient typically influences the velocity primarily near the physical boundary. When approximate projections are used in place of exact ones, the situation is more complicated, since the approximation to the true pressure gradient used to compute $\mathbf{u}^*$ is only *approximately* corrected by the projection.

It is well known that even exact cell centered projection methods can introduce nonphysical oscillations because the operator $\mathbf{D} \cdot \mathbf{G}$ possesses a nontrivial nullspace (resulting in so-called checkerboard modes). Approximate cell centered projection methods can exacerbate this problem by producing computed velocities that are contaminated by components that are nonsolenoidal with respect to the cell centered divergence operator. A common approach to dealing with these difficulties is to stably filter the undesirable components (i.e., the components corresponding to the nonsolenoidal and checkerboard modes) from the computed velocity [29,42,43]. An alternative approach to improving the quality of the computed solution was suggested by Almgren et al. [23], who introduced a hybrid approximate projection method for the incompressible Euler equations. The approximate projection method presented in Section 3.6 is an extension of this method to the viscous case. At each timestep, such hybrid methods determine two different intermediate velocities from two different treatments of the momentum equation. Each intermediate velocity is approximately projected, with the first projection determining the updated velocity and the second yielding the updated pressure. This approach is clearly more computationally demanding than simply determining the velocity and pressure in terms of a single approximate projection.

To justify this additional computational cost and algorithmic complexity, we demonstrate below that nonphysical oscillations can occur in the computed pressure when a more "traditional" projection method is used (i.e., a projection method that obtains the updated velocity and pressure in terms of the solution to a single projection at each timestep). These oscillations, sometimes considered to be a characteristic of the immersed boundary method [13], are virtually eliminated by making use of our hybrid method.

5.1. A more traditional approximate projection method

Before demonstrating the reduction in nonphysical oscillations provided by our hybrid method, we must briefly describe a more traditional second order projection method that we will compare to the hybrid scheme. This method is essentially a version of the projection method of Bell, Colella, and Glaz (BCG) [22], making use of a modification similar to one suggested by Brown et al. [33] to obtain full second order accuracy in the pressure. We refer to the resulting scheme as a "BCG-like" approximate projection method. For this method, the velocity, denoted $\bar{\mathbf{u}}$, and pressure, denoted $\bar{p}$, are both obtained in terms of the solution to the same projection equation. It is important to emphasize that since we are making use of approximate projections, it is generally the case that $\bar{\mathbf{u}} \neq \mathbf{u}$ and $\bar{p} \neq p$, where $\mathbf{u}$ and p are the velocity and pressure obtained via the hybrid approximate projection method of Section 3.6.

In the BCG-like method, the intermediate velocity is determined by

$$(I - \eta_2 vL)(I - \eta_1 vL)\bar{\mathbf{u}}^* = (I + \eta_3 vL)\bar{\mathbf{u}}^n + \Delta t(I + \eta_4 vL)\left(-\mathbf{N}^{n+\frac{1}{2}} + \frac{1}{\rho}\left(\mathbf{f}^{n+\frac{1}{2}} - \mathbf{G}\bar{p}^{n-\frac{1}{2}}\right)\right).$$

Next, $\bar{\mathbf{u}}^{n+1}$ is given by approximately projecting $\bar{\mathbf{u}}^*$, i.e.,

$$\bar{\mathbf{u}}^* = \bar{\mathbf{u}}^{n+1} + \mathbf{G}\bar{\varphi},$$

where

$$L\bar{\varphi} = \mathbf{D} \cdot \bar{\mathbf{u}}^*.$$

The pressure consistent with this treatment of the incompressible Navier–Stokes equations can be obtained in a manner similar to that used above to determine the updated pressure in the hybrid scheme. In this case, the updated pressure is the scalar function $\bar{p}^{n+\frac{1}{2}}$ given by

$$\bar{p}^{n+\frac{1}{2}} = \bar{p}^{n-\frac{1}{2}} + \frac{\rho}{\Delta t}(I + \eta_4 vL)^{-1}(I - \eta_2 vL)(I - \eta_1 vL)\bar{\varphi}.$$

5.2. Reducing nonphysical oscillations via a hybrid projection method

To demonstrate the effectiveness of our hybrid approximate projection method in reducing nonphysical oscillations when compared to a BCG-like projection method, we restrict the curvilinear coordinate space to $\Omega = [\frac{1}{2}, \frac{1}{2}] \times [0, 1]$, so that the structure is a true elastic interface. This is a situation where the immersed boundary method can produce nonphysical oscillations in the computed pressure near the interface [13]. As with the previous computations, the initial configuration is given by (35), although here restricted to $r \equiv \frac{1}{2}$ so that the initial configuration is an ellipse. Following its release at $t = 0$, the membrane undergoes damped oscillations until eventually settling in a circular configuration. We compute the motion of the coupled system up to $t = 0.4$, at which time the elastic structure has completed one full oscillation and is beginning its second. The material properties are as described in Section 4.2, with $\rho = 1.0$, $\mu = 0.005$, and $N = 256$. The smoothed delta function employed for this comparison is $\delta_{4h}^{\mathrm{IB}}$.

The computation is first performed using the BCG-like approximate projection method. When this method is employed, oscillations are readily observed in the computed pressure plotted in Fig. 7. It is clear

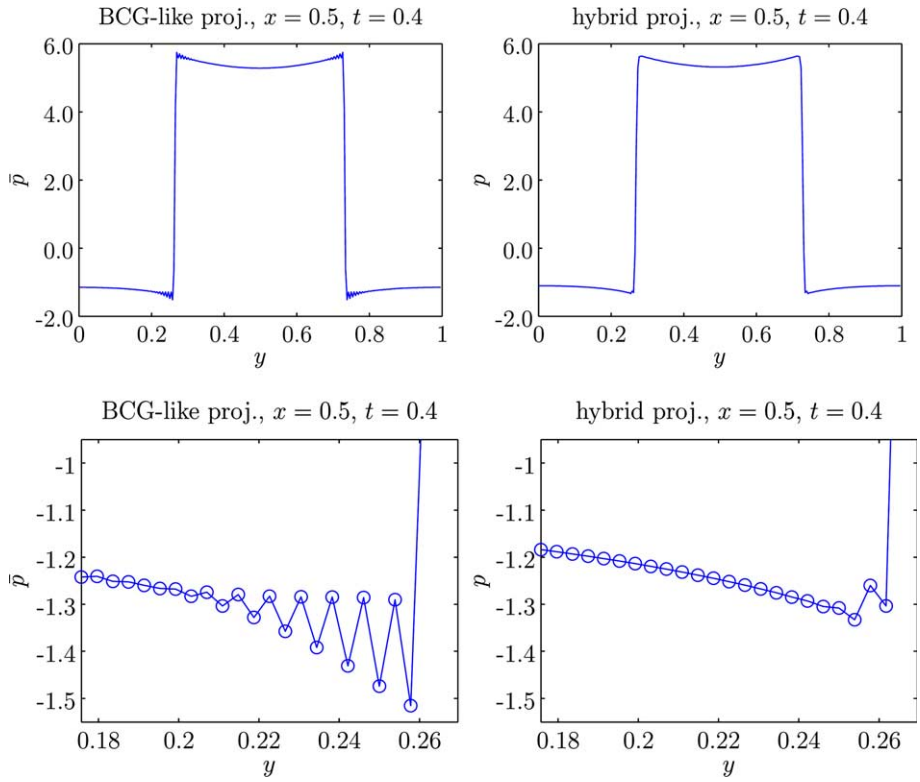

Fig. 7. The pressure at $t = 0.4$ for an elastic *interface* interacting with a viscous incompressible fluid. The pressure plotted in the left-hand column is obtained via a BCG-like projection method. Damped oscillations are evident. In the right-hand column, the hybrid approximate projection method of Section 3.6 is used, virtually eliminating the oscillations in the pressure. Note that the lower plots offer a magnified view of the pressure near $y = 0.25$. For these computations, we use $\delta_{4h}^{\mathrm{IB}}$ with $\rho = 1.0$, $\mu = 0.005$, and $N = 256$.

that these oscillations continue for many grid cells away from the interface before dying out. When we perform the computation again, this time using the hybrid approximate projection method of Section 3.6, the oscillations are virtually eliminated from the computed pressure.

6. Conclusions

In the present work, we have introduced a new formally second order accurate version of the immersed boundary method and examined the performance of the scheme for a prototypical fluid–structure interaction problem with two sets of elastic properties. The new algorithm makes use of several numerical methods intended to reduce the occurrence of nonphysical oscillations in the computed dynamics. In particular, we use a strong stability-preserving Runge–Kutta method for the time integration of the structure configuration, an implicit L-stable discretization of the viscous terms in the momentum equation, and a second order Godunov method for the explicit treatment of the nonlinear terms in the momentum equation. We also employ a new hybrid approximate projection method for the incompressible Navier–Stokes equations – a method that can reduce the occurrence of oscillations in the computed pressure when compared to more traditional projection methods.

By considering fluid–structure interaction problems which possess sufficiently smooth solutions, *actual* second order convergence rates were demonstrated in our numerical tests of the method at low and moderate Reynolds numbers. Even for higher Reynolds numbers flows, empirical convergence rates generally exceeded first order. Unlike previous convergence studies, we did not consider the interaction of a true *interface* and an incompressible fluid. When the immersed boundary method is applied to such problems, second order convergence rates are not observed because of the inability of the method to properly capture discontinuities in the pressure and normal derivative of the velocity across the interface. We avoided these discontinuities by considering the interaction of a viscoelastic shell of finite thickness and an incompressible fluid. Although such problems are in some sense not as difficult as interface problems, they are relevant to many application areas where the immersed boundary method is used. A particular example is the work of Peskin and McQueen, who have used the immersed boundary method to study coupled blood–muscle–valve dynamics in the beating heart [3–5,24,8]. Although elastic surfaces are used in their work to describe the heart valve leaflets, their description of the muscular heart wall is analogous to a viscoelastic shell – albeit one with complex, time-dependent elastic properties.

In our computational convergence study, we considered viscoelastic shells with two types of material properties. In the first case, the elastic properties of the shell were defined so that there is a *continuous* transition in the material properties at the fluid–structure interface. The resulting Cartesian elastic force density, $\mathbf{f}$, is a continuous function of $\mathbf{x}$. We also considered a case where there is a *sharp* discontinuity in the material properties at the fluid–structure interface, resulting in *discontinuous* $\mathbf{f}$. Convergence results were obtained for a wide range of Reynolds numbers using several different smoothed versions of the Dirac delta function. For both sets of material properties, empirical second order convergence rates were generally observed except for the highest Reynolds number cases. In most cases, good performance is obtained for the piecewise cubic delta function, δ_{4h}^{C}, introduced in [32]. This is notable in part because this piecewise cubic function can be computed more efficiently than the other delta functions considered. This delta function should be used with some care, however, as in some cases we have found that it can introduce large oscillations, especially when used for true interface problems. The rather expensive six point delta function, $\delta_{6h}^{\mathrm{IB}}$, produces essentially equivalent numerical results for the problems examined in the present work. It also does not seem to result in the same difficulties as the piecewise cubic function for interface problems.

Though not a phenomenon limited to the immersed boundary method, fine spatial grids appear to be required to adequately resolve the dynamics at higher Reynolds numbers. At least for the problems which we have considered here, we believe this requirement to be localized near thin bands of high vorticity that occur near the fluid–structure interface. Away from these regions, the velocity and pressure are generally slowly varying compared to the resolution of the Cartesian grid. Consequently, we feel that these problems are good candidates for the use of adaptive local mesh refinement. Even though the results presented in this work were for uniform grid computations, our present software actually allows for adaptive block structured refinement in the Cartesian computational grid.

Finally, in order to use the immersed boundary method to model and simulate complex three-dimensional systems, parallel computing is a necessity. Our implementation of the present algorithm allows for distributed memory parallelism. Although we have not made use of this capability in the present work, it will prove necessary when we make use of this software for more complex problems.

Acknowledgments

This work was supported in part by the Department of Energy Computational Science Graduate Fellowship Program of the Office of Scientific Computing and Office of Defense Programs in the United States Department of Energy under contract DE-FG02-97ER25308.

Portions of this work were performed under the auspices of the United States Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

Special thanks go to Richard Hornung, who graciously arranged for the summer visits to Lawrence Livermore National Laboratory that led to the present work, and who has provided (and continues to provide) invaluable assistance in our use of SAMRAI.

Appendix A. An explicit second order Godunov method

Even though the explicit Goudnov method that we employ to treat the nonlinear advection terms appearing in the incompressible Navier–Stokes equations is based on well-established methods introduced by Colella [17] and modified by Minion [18,19] and Martin and Colella [30], it is hard to find a single place where they are brought together in the particular way that we have done in this work. The purpose of this appendix, therefore, is to document what we have done for the convenience of the reader.

A.1. Face centered notation and finite difference operators

Throughout the foregoing discussion, all Eulerian quantities have been described at the *centers* of the cells of the Cartesian grid. To describe the particular Godunov procedure used to approximate the nonlinear advection term in Eqs. (27) and (29), we additionally make use of Eulerian quantities described at the *faces* of the Cartesian grid cells (or really the *edges* of the grid cells in the two-dimensional case that we consider here).

If a quantity $\psi(\mathbf{x}, t)$ is defined on the faces of the Cartesian grid cells, we employ the notation $\psi^n_{i-\frac{1}{2},j} \equiv \psi(\mathbf{x}_{i-\frac{1}{2},j}, t_n)$ to indicate the evaluation of ψ on the x-faces of the grid, i.e., at the points $\mathbf{x}_{i-\frac{1}{2},j} = \left(ih, (j+\frac{1}{2})h\right)$. Similarly, $\psi^n_{i,j-\frac{1}{2}} \equiv \psi(\mathbf{x}_{i,j-\frac{1}{2}}, t_n)$ indicates the evaluation of ψ on the y-faces of the grid, i.e., at the points $\mathbf{x}_{i,j-\frac{1}{2}} = \left((i+\frac{1}{2})h, jh\right)$. Recall that $i, j \in \{0, 1, \ldots, N-1\}$ index the centers of the Cartesian grid cells (see Fig. A.1).

By convention, a vector field defined on the Cartesian grid in terms of those vector components that are normal to the faces of the grid cells is called a MAC vector field [44]. That is to say, if $\mathbf{u}^{\mathrm{MAC}} = (u^{\mathrm{MAC}}, v^{\mathrm{MAC}})$ is a MAC vector field, u^{MAC} is defined at the points $\mathbf{x}_{i-\frac{1}{2},j} = \left(ih, (j+\frac{1}{2})h\right)$, whereas v^{MAC} is defined at the points $\mathbf{x}_{i,j-\frac{1}{2}} = \left((i+\frac{1}{2})h, jh\right)$. In contrast to this *staggered grid* description, the components of the Eulerian vector fields previously encountered, such as $\mathbf{u}$ and $\mathbf{f}$, have been *co-located* at cell centers.

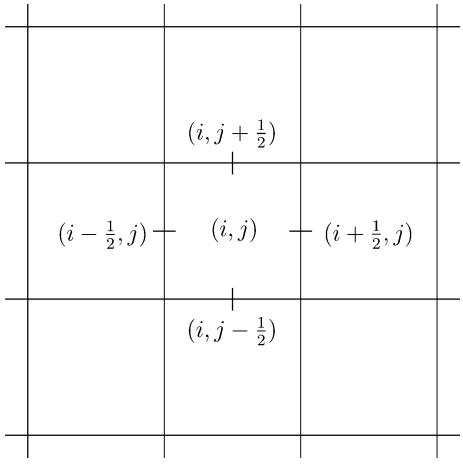

Fig. A.1. Locations of cell centered and face centered quantities about Cartesian grid cell (i, j).

In the present work, when a MAC vector field is defined by interpolating $\mathbf{u}_{i,j} = (u_{i,j}, v_{i,j})$ from cell centers to cell faces, the individual components of $\mathbf{u}^{\text{MAC}}$ are obtained by linear interpolation (averaging). We employ the notation

$$u^{\text{MAC}}_{i+\frac{1}{2},j} = (A^{c\to f}\mathbf{u})_{i+\frac{1}{2},j} = \frac{u_{i,j} + u_{i+1,j}}{2},$$

$$v^{\text{MAC}}_{i,j+\frac{1}{2}} = (A^{c\to f}\mathbf{u})_{i,j+\frac{1}{2}} = \frac{v_{i,j} + v_{i,j+1}}{2},$$

and say in this case that $\mathbf{u}^{\text{MAC}} = A^{c\to f}\,\mathbf{u}$. Notice that only the normal component of $\mathbf{u}^{\text{MAC}}$ is defined at a given cell face.

Similar to their purely cell centered counterparts, the cell centered divergence of a MAC vector field is approximated by centered differences, i.e.,

$$(\mathbf{D}^{f\to c}\cdot\mathbf{u}^{\text{MAC}})_{i,j} = \frac{u^{\text{MAC}}_{i+\frac{1}{2},j} - u^{\text{MAC}}_{i-\frac{1}{2},j}}{h} + \frac{v^{\text{MAC}}_{i,j+\frac{1}{2}} - v^{\text{MAC}}_{i,j-\frac{1}{2}}}{h},$$

whereas the MAC gradient of a cell centered scalar quantity is approximated at cell faces by

$$(\mathbf{G}^{c\to f}\psi)_{i+\frac{1}{2},j} = \frac{\psi_{i+1,j} - \psi_{i,j}}{h}, \quad (\mathbf{G}^{c\to f}\psi)_{i,j+\frac{1}{2}} = \frac{\psi_{i,j+1} - \psi_{i,j}}{h}.$$

It is important to note that for a cell centered scalar quantity,

$$(\mathbf{D}^{f\to c}\cdot\mathbf{G}^{c\to f}\psi)_{i,j} = (L\psi)_{i,j},$$

where L is the cell centered approximation to the Laplace operator introduced in Section 3.2. This correspondence allows us to easily compute the exact projection of a MAC vector field. In particular, the MAC projection of $\mathbf{w}^{\text{MAC}}$ is given by

$$\mathbf{v}^{\text{MAC}} = P^{\text{MAC}}\mathbf{w}^{\text{MAC}} = \left(I - \mathbf{G}^{c\to f}L^{-1}\mathbf{D}^{f\to c}\cdot\right)\mathbf{w}^{\text{MAC}}.$$

This is an *exact* projection, since $(\mathbf{D}^{f\to c}\cdot\mathbf{v}^{\text{MAC}})_{i,j} \equiv 0$. Note that in practice, the application of this exact MAC projection does not require the use of any additional linear solvers beyond those required by the cell centered approximate projection, $\tilde{P}$, described in Section 3.3.

A.2. An auxiliary MAC velocity

Before describing the Godunov procedure used to approximate $[(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+\frac{1}{2}}$ in our treatment of the incompressible Navier–Stokes equations, we describe an auxiliary MAC velocity, denoted $\mathbf{u}^{\text{MAC}}$, that is maintained in addition to the cell centered velocity, $\mathbf{u}$. This MAC velocity was not introduced in the initial discussion of the approximate projection method as it is *only* used in the Godunov extrapolation procedure.

The auxiliary MAC velocity is obtained in the process of computing the cell centered velocity, $\mathbf{u}$. In our approximate projection method, recall that an intermediate cell centered velocity, $\mathbf{u}^*$, is determined by solving Eq. (27). The true cell centered velocity, $\mathbf{u}^{n+1}$, is then obtained in Eq. (28) as the approximate projection of $\mathbf{u}^*$. To compute $\mathbf{u}^{\text{MAC},n+1}$, we first interpolate $\mathbf{u}^*$ from cell centers to cell faces, obtaining

$$\mathbf{u}^{\text{MAC},*} = A^{c\to f}\mathbf{u}^*.$$

$\mathbf{u}^{\text{MAC},n+1}$ is then obtained by computing the MAC projection of $\mathbf{u}^{\text{MAC},*}$. Luckily, this does not require the solution of an additional system of linear equations! To see why this is so, recall that the approximate projection of $\mathbf{u}^*$ requires the solution of a discrete Poisson problem of the form

$$L\varphi = \mathbf{D}\cdot\mathbf{u}^*. \tag{A.1}$$

Since $(\mathbf{D}^{f\to c}\cdot\mathbf{u}^{\mathrm{MAC},*})_{i,j}\equiv(\mathbf{D}\cdot\mathbf{u}^{*})_{i,j}$, (A.1) is the same equation that must be solved in order to project $\mathbf{u}^{\mathrm{MAC},*}$. The solution to (A.1), φ, may simply be re-used to directly evaluate

$$\mathbf{u}^{\mathrm{MAC},n+1}=\mathbf{u}^{\mathrm{MAC},*}-\mathbf{G}^{c\to f}\varphi.$$

The initial value of $\mathbf{u}^{\mathrm{MAC}}$ may be obtained from the initial value of $\mathbf{u}$ analogously. Before projecting the initial velocity in (34), we first compute

$$\mathbf{u}^{\mathrm{MAC},0}=A^{c\to f}\mathbf{u}^{0}.$$

We then approximately project $\mathbf{u}^{0}$ and, as above, re-use φ to exactly project $\mathbf{u}^{\mathrm{MAC},0}$.

A.3. An explicit Godunov extrapolation procedure

In order to simplify the description of the Godunov procedure, we temporarily restrict our attention in this section to the advection–diffusion equation,

$$\frac{\partial q}{\partial t}+(\mathbf{u}\cdot\nabla)q=\nu\nabla^{2}q+\psi, \tag{A.2}$$

where q is a scalar quantity, $\mathbf{u}$ is a specified advection velocity, $\nu\geqslant 0$ is the diffusion coefficient, and ψ is a given source term. Our goal in this section is to describe an explicit second order accurate procedure that uses values defined at time t_n to extrapolate the face centered value of q at time $t_{n+\frac{1}{2}}$. Although described only for the advection–diffusion equation, the following procedure is used *without modification* as part of our approximation to the nonlinear advection term appearing in the incompressible Navier–Stokes equations.

The particular procedure we employ is an explicit Godunov method introduced by Colella [17,30] and modified by Minion [18,19]. The idea is to extrapolate q from cell centers to cell faces by using Taylor expansions for q about the Cartesian cell centers. Notice that this extrapolation process is ambiguous: at every cell face, there are *two* nearest cell centers and hence two Taylor expansions to choose from. The ambiguity is resolved by using the expansion that is about the cell center that lies in the upwind direction – a method motivated by the solution to the Riemann problem for the one-dimensional Burgers' equation.

In order to obtain a stable explicit extrapolation scheme, the timestep must satisfy a condition of the form $\Delta t=\mathcal{O}(h)$. Consequently, the Taylor series for $q(\mathbf{x},t)$, taken about the point $\mathbf{x}_{i,j}$ and evaluated at cell face $\mathbf{x}_{i+\frac{1}{2},j}$ and at time $t=t_{n+\frac{1}{2}}$, is

$$q^{n+\frac{1}{2},\mathrm{L}}_{i+\frac{1}{2},j}=q^{n}_{i,j}+\frac{h}{2}(q_x)^{n}_{i,j}+\frac{\Delta t}{2}(q_t)^{n}_{i,j}+\mathcal{O}(h^{2}).$$

Note that in this expansion, the face centered value of q is obtained in terms of cell centered quantities that lie to the *left* of cell face $\mathbf{x}_{i+\frac{1}{2}}$. The time derivative, q_t, can be eliminated by making use of Eq. (A.2), yielding

$$q^{n+\frac{1}{2},\mathrm{L}}_{i+\frac{1}{2},j}=q^{n}_{i,j}+\left(\frac{h}{2}-\frac{\Delta t}{2}u^{n}_{i,j}\right)(q_x)^{n}_{i,j}-\frac{\Delta t}{2}v^{n}_{i,j}(q_y)^{n}_{i,j}+\frac{\Delta t}{2}\left(\nu(\nabla^{2}q)^{n}_{i,j}+\psi^{n}_{i,j}\right)+\mathcal{O}(h^{2}), \tag{A.3}$$

where $\mathbf{u}=(u,v)$. A similar expansion about $\mathbf{x}_{i,j+1}$, evaluated at face $\mathbf{x}_{i,j+\frac{1}{2}}$, yields the *top* state,

$$q^{n+\frac{1}{2},\mathrm{T}}_{i,j+\frac{1}{2}}=q^{n}_{i,j+1}-\left(\frac{h}{2}+\frac{\Delta t}{2}v^{n}_{i,j+1}\right)(q_y)^{n}_{i,j+1}-\frac{\Delta t}{2}u^{n}_{i,j+1}(q_x)^{n}_{i,j+1}+\frac{\Delta t}{2}\left(\nu(\nabla^{2}q)^{n}_{i,j+1}+\psi^{n}_{i,j+1})\right)+\mathcal{O}(h^{2}). \tag{A.4}$$

Similar expansions define the *right* and *bottom* states, $q^{n+\frac{1}{2},R}_{i+\frac{1}{2},j}$ and $q^{n+\frac{1}{2},B}_{i,j+\frac{1}{2}}$.

Following [18,19], a second order approximation to each of the proceeding Taylor expansions is computed in two steps. The resulting scheme is stable so long as the timestep satisfies a CFL condition of the form $\|\mathbf{u}\|_{\infty}\Delta t\leqslant h$. Note that the stability restriction is independent of the value of $\nu\geqslant 0$.

In the first step of the scheme, (A.3) is approximated by

$$\hat{q}^{n+\frac{1}{2},\mathrm{L}}_{i+\frac{1}{2},j} = q^n_{i,j} + \left(\frac{h}{2} - \frac{\Delta t}{4}\left(u^{\mathrm{MAC},n}_{i+\frac{1}{2},j} + u^{\mathrm{MAC},n}_{i-\frac{1}{2},j}\right)\right)\left(D^0_x q\right)^n_{i,j} + \frac{\Delta t}{2}\left(v(Lq)^n_{i,j} + \psi^n_{i,j}\right),$$

where D^0_x is a fourth order centered difference operator defined by

$$\left(D^0_x q\right)^n_{i,j} = \frac{2}{3}\left(q^n_{i+1,j} - q^n_{i-1,j}\right) - \frac{1}{12}\left(q^n_{i+2,j} - q^n_{i-2,j}\right).$$

Similarly, (A.4) is approximated by

$$\hat{q}^{n+\frac{1}{2},\mathrm{T}}_{i,j+\frac{1}{2}} = q^n_{i,j+1} - \left(\frac{h}{2} + \frac{\Delta t}{4}\left(v^{\mathrm{MAC},n}_{i,j+\frac{3}{2}} + v^{\mathrm{MAC},n}_{i,j+\frac{1}{2}}\right)\right)\left(D^0_y q\right)^n_{i,j+1} + \frac{\Delta t}{2}\left(v(Lq)^n_{i,j+1} + \psi^n_{i,j+1}\right).$$

Analogous approximations define $\hat{q}^{n+\frac{1}{2},\mathrm{R}}_{i+\frac{1}{2},j}$ and $\hat{q}^{n+\frac{1}{2},\mathrm{B}}_{i,j+\frac{1}{2}}$. Note that each of these values includes approximations only to those derivative terms that are *normal* to the cell face where the expansion is being approximated. For now, transverse derivatives are not included.

At each cell face, $\hat{q}^{n+\frac{1}{2}}$ is defined by choosing the upwind state, namely

$$\hat{q}^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \begin{cases} \hat{q}^{n+\frac{1}{2},\mathrm{L}}_{i+\frac{1}{2},j} & \text{if } u^{\mathrm{MAC},n}_{i+\frac{1}{2},j} > 0, \\ \hat{q}^{n+\frac{1}{2},\mathrm{R}}_{i+\frac{1}{2},j} & \text{if } u^{\mathrm{MAC},n}_{i+\frac{1}{2},j} < 0, \\ \frac{1}{2}\left(\hat{q}^{n+\frac{1}{2},\mathrm{L}}_{i+\frac{1}{2},j} + \hat{q}^{n+\frac{1}{2},\mathrm{R}}_{i+\frac{1}{2},j}\right) & \text{if } u^{\mathrm{MAC},n}_{i+\frac{1}{2},j} = 0, \end{cases} \tag{A.5}$$

and

$$\hat{q}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = \begin{cases} \hat{q}^{n+\frac{1}{2},\mathrm{B}}_{i,j+\frac{1}{2}} & \text{if } v^{\mathrm{MAC},n}_{i,j+\frac{1}{2}} > 0, \\ \hat{q}^{n+\frac{1}{2},\mathrm{T}}_{i,j+\frac{1}{2}} & \text{if } v^{\mathrm{MAC},n}_{i,j+\frac{1}{2}} < 0, \\ \frac{1}{2}\left(\hat{q}^{n+\frac{1}{2},\mathrm{B}}_{i,j+\frac{1}{2}} + \hat{q}^{n+\frac{1}{2},\mathrm{T}}_{i,j+\frac{1}{2}}\right) & \text{if } v^{\mathrm{MAC},n}_{i,j+\frac{1}{2}} = 0. \end{cases} \tag{A.6}$$

The second step in the extrapolation procedure introduces approximations to the *transverse* derivative terms. These approximations are obtained by differencing the initial extrapolation, $\hat{q}^{n+\frac{1}{2}}$. In particular,

$$\tilde{q}^{n+\frac{1}{2},\mathrm{L}}_{i+\frac{1}{2},j} = \hat{q}^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - \frac{\Delta t}{4h}\left(v^{\mathrm{MAC},n}_{i,j+\frac{1}{2}} + v^{\mathrm{MAC},n}_{i,j-\frac{1}{2}}\right)\left(\hat{q}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - \hat{q}^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\right),$$

and

$$\tilde{q}^{n+\frac{1}{2},\mathrm{T}}_{i,j+\frac{1}{2}} = \hat{q}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - \frac{\Delta t}{4h}\left(u^{\mathrm{MAC},n}_{i+\frac{1}{2},j+1} + u^{\mathrm{MAC},n}_{i-\frac{1}{2},j+1}\right)\left(\hat{q}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+1} - \hat{q}^{n+\frac{1}{2}}_{i-\frac{1}{2},j+1}\right),$$

Similar formulas yield the remaining values. Finally, on each cell face, the value of $\tilde{q}^{n+\frac{1}{2}}$ is obtained by choosing the upwind state as in (A.5) and (A.6).

A.4. Computing the advection term

In order to compute the explicit approximation to the nonlinear advection term, $[(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+\frac{1}{2}}$, used in the solution of the incompressible Navier–Stokes equations, we employ a timestep centered "advection" velocity, denoted $\mathbf{u}^{\mathrm{ADV}}$. This advection velocity is a discretely divergence free MAC vector field and is obtained in two steps:

The first step in obtaining $\mathbf{u}^{\mathrm{ADV}}$ employs the Godunov scheme detailed in Appendix A.3. This procedure uses $\mathbf{u}^{\mathrm{MAC},n}$ to extrapolate the cell centered velocity, $\mathbf{u}^n_{i,j} = (u^n_{i,j}, v^n_{i,j})$, to cell faces. This is performed

component-wise, i.e., we first employ the Godunov procedure with $u^n_{i,j}$ replacing $q^n_{i,j}$ and employ the corresponding source term,

$$\psi^n_{i,j} = \frac{1}{\rho}\left((f_1)^n_{i,j} - (G_x p)^{n-\frac{1}{2}}_{i,j}\right),$$

where $\mathbf{f} = (f_1, f_2)$ is the discrete Cartesian elastic force density and $\mathbf{G}p = (G_x p, G_y p)$ is the discrete pressure gradient. This yields a timestep centered approximation to u at each cell face in the Cartesian grid. Next, we perform the analogous procedure for v, i.e., we replace $q^n_{i,j}$ with $v^n_{i,j}$ and use the appropriate source term,

$$\psi^n_{i,j} = \frac{1}{\rho}\left((f_2)^n_{i,j} - (G_y p)^{n-\frac{1}{2}}_{i,j}\right),$$

yielding a timestep centered approximation to v at each cell face in the grid. These extrapolated velocities are denoted by $\tilde{u}^{n+\frac{1}{2}}$ and $\tilde{v}^{n+\frac{1}{2}}$.

The second step in obtaining $\mathbf{u}^{\mathrm{ADV}}$ *discards* the transverse components of the extrapolated velocity field. This yields a MAC velocity, denoted $\tilde{\mathbf{u}}^{\mathrm{ADV},*}$. This velocity field will generally not be discretely divergence free (with respect to the MAC divergence operator, $\mathbf{D}^{f\to c}\cdot$). To enforce incompressibility, the advection velocity is defined to be the MAC projection of $\tilde{\mathbf{u}}^{\mathrm{ADV},*}$, i.e.,

$$\mathbf{u}^{\mathrm{ADV}} = P^{\mathrm{MAC}}\tilde{\mathbf{u}}^{\mathrm{ADV},*} = \tilde{\mathbf{u}}^{\mathrm{ADV},*} - \mathbf{G}^{c\to f}\tilde{\varphi},$$

where $\tilde{\varphi}$ is the solution to a discrete Poisson problem,

$$L\tilde{\varphi} = \mathbf{D}^{f\to c}\cdot\tilde{\mathbf{u}}^{\mathrm{ADV},*}. \tag{A.7}$$

This completes the procedure for computing $\mathbf{u}^{\mathrm{ADV}}$.

With $\mathbf{u}^{\mathrm{ADV}}$ in hand, we next *re-extrapolate* the timestep centered normal and transverse velocities at each cell face, using the timestep centered advection velocity, $\mathbf{u}^{\mathrm{ADV}}$, in place of $\mathbf{u}^{\mathrm{MAC},n}$. Except for this one difference, the extrapolation procedure is identical to that previously used to obtain $\tilde{u}^{n+\frac{1}{2}}$ and $\tilde{v}^{n+\frac{1}{2}}$, again making use of the Godunov procedure of Appendix A.3. Doing so yields a second approximation to the timestep centered normal and transverse velocities at each cell face in the Cartesian grid. These extrapolated values are denoted $\bar{u}^{n+\frac{1}{2}}$ and $\bar{v}^{n+\frac{1}{2}}$.

Next, the solution to (A.7), $\tilde{\varphi}$, is used to *approximately* enforce the incompressibility constraint. For the velocities normal to the cell face where they are defined, we set

$$u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \bar{u}^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - \frac{1}{h}\left(\tilde{\varphi}_{i+1,j} - \tilde{\varphi}_{i,j}\right),$$

$$v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = \bar{v}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - \frac{1}{h}\left(\tilde{\varphi}_{i,j+1} - \tilde{\varphi}_{i,j}\right),$$

whereas for the transverse components, we have

$$u^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = \bar{u}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - \frac{1}{4h}\left(\tilde{\varphi}_{i+1,j} - \tilde{\varphi}_{i-1,j} + \tilde{\varphi}_{i+1,j+1} - \tilde{\varphi}_{i-1,j+1}\right),$$

$$v^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \bar{v}^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - \frac{1}{4h}\left(\tilde{\varphi}_{i,j+1} - \tilde{\varphi}_{i,j-1} + \tilde{\varphi}_{i+1,j+1} - \tilde{\varphi}_{i+1,j-1}\right).$$

Note that in each case, the appropriate component of the discrete gradient of $\tilde{\varphi}$ is being used to approximately enforce the incompressibility constraint.

At long last, the approximation to the nonlinear advection term,

$$\mathbf{N}^{n+\frac{1}{2}}_{i,j} = \left((N_1)^{n+\frac{1}{2}}_{i,j}, (N_2)^{n+\frac{1}{2}}_{i,j}\right) \approx [(\mathbf{u}\cdot\nabla)\mathbf{u}]^{n+\frac{1}{2}}_{i,j},$$

is defined by nonconservative differencing via

$$(N_1)_{i,j}^{n+\frac{1}{2}} = \frac{1}{2h}\left(u_{i+\frac{1}{2},j}^{\mathrm{ADV}} + u_{i-\frac{1}{2},j}^{\mathrm{ADV}}\right)\left(u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - u_{i-\frac{1}{2},j}^{n+\frac{1}{2}}\right) + \frac{1}{2h}\left(v_{i,j+\frac{1}{2}}^{\mathrm{ADV}} + v_{i,j-\frac{1}{2}}^{\mathrm{ADV}}\right)\left(u_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - u_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}\right),$$

$$(N_2)_{i,j}^{n+\frac{1}{2}} = \frac{1}{2h}\left(u_{i+\frac{1}{2},j}^{\mathrm{ADV}} + u_{i-\frac{1}{2},j}^{\mathrm{ADV}}\right)\left(v_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - v_{i-\frac{1}{2},j}^{n+\frac{1}{2}}\right) + \frac{1}{2h}\left(v_{i,j+\frac{1}{2}}^{\mathrm{ADV}} + v_{i,j-\frac{1}{2}}^{\mathrm{ADV}}\right)\left(v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - v_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}\right).$$

Since $\mathbf{u}^{\mathrm{ADV}}$ is discretely divergence free, we could have employed conservative differencing here to approximate the advection term. We do not do so, however, as we find that the use of nonconservative differencing produces lower errors when we test the approximate projection method against known analytic solutions to the incompressible Navier–Stokes equations.

References

[1] M.-C. Lai, Simulations of the flow past an array of circular cylinders as a test of the immersed boundary method, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, 1998.
[2] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2) (2000) 705–719.
[3] C.S. Peskin, D.M. McQueen, A three-dimensional computational method for blood flow in the heart. I. Immersed elastic fibers in a viscous incompressible fluid, J. Comput. Phys. 81 (2) (1989) 372–405.
[4] D.M. McQueen, C.S. Peskin, A three-dimensional computational method for blood flow in the heart. II. Contractile fibers, J. Comput. Phys. 82 (2) (1989) 289–297.
[5] C.S. Peskin, D.M. McQueen, Fluid dynamics of the heart and its valves, in: H.G. Othmer, F.R. Adler, M.A. Lewis, J.C. Dallon (Eds.), Case Studies in Mathematical Modeling: Ecology, Physiology, and Cell Biology, Prentice-Hall, Englewood Cliffs, NJ, 1996, pp. 309–337.
[6] D.C. Bottino, L.J. Fauci, A computational model of ameboid deformation and locomotion, Eur. Biophys. J. 27 (5) (1998) 532–539.
[7] N.T. Wang, A.L. Fogelson, Computational methods for continuum models of platelet aggregation, J. Comput. Phys. 151 (2) (1999) 649–675.
[8] D.M. McQueen, C.S. Peskin, A three-dimensional computer model of the human heart for studying cardiac fluid dynamics, Comput. Graphics 34 (1) (2000) 56–60.
[9] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[10] R.J. Leveque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (3) (1997) 709–735.
[11] R. Cortez, M.L. Minion, The blob projection method for immersed boundary problems, J. Comput. Phys. 161 (2) (2000) 428–453.
[12] R. Cortez, The method of regularized Stokeslets, SIAM J. Sci. Comput. 23 (4) (2001) 1204–1225.
[13] L. Lee, R.J. Leveque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (3) (2003) 832–856.
[14] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.
[15] E.H. Twizell, A.B. Gumel, M.A. Arigu, Second-order, L_0-stable methods for the heat equation with time-dependent boundary conditions, Adv. Comput. Math. 6 (3–4) (1996) 333–352.
[16] P. McCorquodale, P. Colella, H. Johansen, A Cartesian grid embedded boundary method for the heat equation on irregular domains, J. Comput. Phys. 173 (2) (2001) 620–635.
[17] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1) (1989) 171–200.
[18] M.L. Minion, On the stability of Godunov-projection methods for incompressible flow, J. Comput. Phys. 123 (2) (1996) 435–449.
[19] M.L. Minion, A projection method for locally refined grids, J. Comput. Phys. 127 (1) (1996) 158–178.
[20] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (104) (1968) 745–762.
[21] A.J. Chorin, On the convergence of discrete approximations to the Navier–Stokes equations, Math. Comput. 23 (106) (1969) 341–353.
[22] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (2) (1989) 257–283.
[23] A.S. Almgren, J.B. Bell, W.Y. Crutchfield, Approximate projection methods: Part I. Inviscid analysis, SIAM J. Sci. Comput. 22 (4) (2000) 1139–1159.

[24] D.M. McQueen, C.S. Peskin, Shared-memory parallel vector implementation of the immersed boundary method for the computation of blood flow in the beating mammalian heart, J. Supercomput. 11 (3) (1997) 213–236.
[25] E.N. Jung, C.S. Peskin, Two-dimensional simulations of valveless pumping using the immersed boundary method, SIAM J. Sci. Comput. 23 (1) (2001) 19–45.
[26] L.D. Zhu, C.S. Peskin, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, J. Comput. Phys. 179 (2) (2002) 452–468.
[27] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (2) (1999) 509–534.
[28] A.S. Almgren, J.B. Bell, W.G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, SIAM J. Sci. Comput. 17 (2) (1996) 358–369.
[29] M.F. Lai, A projection method for reacting flow in the zero Mach number limit, Ph.D. thesis, University of California at Berkeley, 1993.
[30] D.F. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible Euler equations, J. Comput. Phys. 163 (2) (2000) 271–312.
[31] J.M. Stockie, Analysis and computation of immersed boundaries, with application to pulp fibres, Ph.D. thesis, Institute of Applied Mathematics, University of British Columbia, 1997.
[32] A.-K. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations, J. Comput. Phys. 200 (2) (2004) 462–488.
[33] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2) (2001) 464–499.
[34] SAMRAI: Structured adaptive mesh refinement application infrastructure, <http://www.llnl.gov/CASC/SAMRAI>.
[35] R.D. Hornung, S.R. Kohn, Managing application complexity in the SAMRAI object-oriented framework, Concurrency Comput.: Pract. Ex. 14 (5) (2002) 347–368.
[36] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Web page, <http://www.mcs.anl.gov/petsc>.
[37] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Users Manual, Technical Report ANL-95/11 – Revision 2.1.5, Argonne National Laboratory, 2004.
[38] S. Balay, V. Eijkhout, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.
[39] *hypre*: High performance preconditioners, <http://www.llnl.gov/CASC/hypre>.
[40] R.D. Falgout, U.M. Yang, *hypre*: a library of high performance preconditioners, in: P.M.A. Sloot, C.J.K. Tan, J.J. Dongarra, A.G. Hoekstra (Eds.), Computational Science – ICCS 2002 Part III, Lecture Notes in Computer Science, vol. 2331, Springer-Verlag, 2002, pp. 632–641, also available as LLNL Technical Report UCRL-JC-146175.
[41] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.
[42] W.J. Rider, Filtering nonsolenoidal modes in numerical solutions of incompressible flows, Technical Report LA-UR-3014, Los Alamos National Laboratory, 1994.
[43] R.K. Crockett, P. Colella, R.T. Fisher, R.I. Klein, C.F. McKee, An unsplit, cell-centered Godunov method for ideal MHD, J. Comput. Phys. 203 (2) (2005) 422–448.
[44] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.