IMMERSED BOUNDARY METHOD FOR VARIABLE VISCOSITY AND VARIABLE DENSITY PROBLEMS USING FAST CONSTANT-COEFFICIENT LINEAR SOLVERS I: NUMERICAL METHOD AND RESULTS

THOMAS G. FAI*, BOYCE E. GRIFFITH†, YOICHIRO MORI ‡, AND CHARLES S. PESKIN §

Abstract. We present a general variable viscosity and variable density immersed boundary method that is first-order accurate in the variable density case and, for problems possessing sufficient regularity, second-order accurate in the constant density case. The viscosity and density are considered material properties and are defined by a dynamically updated tesselation. Empirical convergence rates are reported for a test problem of a two-dimensional viscoelastic shell with spatially varying material properties. The reduction to first-order accuracy in the variable density case can be avoided by using an iterative scheme, although this approach may not be efficient enough for practical use. In our timestepping scheme, both the inertial and viscous terms are split into two parts: a constant-coefficient part that is treated implicitly, and a variable-coefficient part that is treated explicitly. This splitting allows the resulting equations to be solved efficiently using fast constant-coefficient linear solvers, and in this work, we use solvers based on the Fast Fourier transform (FFT). As an application of this method, we perform fully three-dimensional, two-phase simulations of red blood cells accounting for variable viscosity and variable density. We study the behavior of red cells during shear flow and during capillary flow.

Key words. immersed boundary method, variable viscosity, red blood cells

AMS subject classifications. 65M06, 76D05, 76Z05

1. Introduction. The immersed boundary method is a general approach for simulating fluid-structure interaction. It has been used to study diverse phenomena including animal locomotion, DNA coiling, blood clotting, and heart valve dynamics [1, 2, 3, 4, 5, 6, 7]. It has also been applied to various problems in cellular dynamics, such as biofilm dynamics [8], cellular growth [9, 10, 11], cellular blebbing [12], and platelet margination [13]. Although the original immersed boundary method was formulated for fluids with uniform density and viscosity, several variable density versions of the method have been developed for applications in which the mass density of the structure is different from that of the background fluid, such as the dynamics of parachutes and flapping flags [14, 15, 16]. Another approach, the Front-Tracking method of Unverdi and Tryggvason [17], has been used together with the immersed boundary method to simulate multiphase fluids with piecewise constant density and viscosity. Situations more general than the piecewise constant case are also of significant interest. For instance, continuously varying viscosity appears in models of stratified fluids [18] and flow through pipes with viscous heating [19] (the viscosity of water is highly temperature dependent, varying between 1.79 cP to 0.28 cP in the range 0 °C to 100 °C). To our knowledge, the immersed boundary method presented here is the first to handle the general variable viscosity case, in which the viscosity may be an arbitrary Eulerian or Lagrangian function.

To test our methodology, we calculate empirical convergence rates by simulating a thick elastic shell with smoothly varying material properties immersed in fluid. We find second-order accuracy in the constant density case and first-order accuracy in the variable density case. By applying an iterative scheme, we recover second-order accuracy in the variable density case as well, although using this iteration may not be practical since a large number of iterations is required. Note that the delta function layer forces generated by the elastic surfaces in many immersed boundary applications already limit the accuracy to first-order, and in such cases iterations are not needed.

We further apply this methodology to simulate the three-dimensional dynamics of red blood cells. Variable viscosity plays a crucial role in this situation since the viscosity of blood plasma is around 1.2 cP at body temperature, while the viscoscity of the hemoglobin solution contained in red blood cells is around 6.0 cP. Red blood cells exhibit a spectacular range of behaviors, evolution's response to the many conditions they experience while delivering oxygen throughout our bodies. At rest, they form biconcave disks with a diameter of 6–8 μm. In order to squeeze through capillaries as thin as 4 μm in diameter, they must undergo dramatic deformations. As they travel through these vessels in single file, they adopt shapes that can be parachute-like or slipper-like, as shown in Skalak and Branemark [20]. When placed in shear flow, red cells can undergo *tank treading*, in which the cell lengthens and the membrane revolves around the inner fluid, or *tumbling*, in which

*Courant Institute of Mathematical Sciences, New York University, 251 Mercer St., New York, NY, 10012 (tfai@cims.nyu.edu).

†Leon H. Charney Division of Cardiology, Department of Medicine, New York University School of Medicine, 550 First Avenue, New York, NY 10016

‡School of Mathematics, University of Minnesota, 206 Church Street, Minneapolis MN 55455

§Courant Institute of Mathematical Sciences, New York University, 251 Mercer St., New York, NY, 10012

the entire cell falls head over tail. With nearly 45% of blood's volume composed of red blood cells, knowledge of their behavior is crucial for a better understanding of physiological blood flow.

Starting with the work of Eggleton and Popel [21], the immersed boundary method has been used to simulate red blood cells under various flow conditions. The first studies used a simple membrane model that did not include bending resistance, but subsequent work has developed more realistic membrane models that capture the physiological equilibrium shapes over a wide range of parameters [22]. Recent simulations have used the Front-Tracking method to account for the different viscosities of the blood plasma and the highly viscous hemoglobin solution contained in the red cells [23, 24]. Particularly noteworthy are computations of tank treading and tumbling frequencies as functions of shear rate and viscosity that closely match the available experimental data and predict the behavior in regimes for which no experimental data yet exist [25, 26]. Other approaches developed for simulating red cells that do not use the immersed boundary method include multiparticle collision dynamics [27] and molecular dynamics [28, 29]. Mesoscale approaches have also been developed [30] that calculate the viscosity as a function of hematocrit, the volume fraction taken up by red cells.

In what follows, we begin by describing our fluid solver, which is based on the projection method of Chorin [31], in the case that the variable density $\rho(\mathbf{x},t)$ and variable viscosity $\mu(\mathbf{x},t)$ are known functions. Our timestepping scheme involves two numerical parameters that determine the splitting between explicit and implicit contributions to the viscous and inertial terms. The choice of these parameters is motivated by a theoretical analysis of the scheme contained in a companion article [32]. The mixed implicit and explicit treatment of the viscous term has been used previously in other contexts including phase separation [33], large eddy simulation [34], and miscible fluids in capillary flow [35]. In addition, Karamanos and Sherwin [36] performed von Neumann stability analysis of a similar timestepping scheme for the Navier-Stokes equations with variable viscosity. Our work extends these ideas to the case of fluid-structure interaction. In Section 3, we present the immersed boundary formulation of the problem in which $\rho(\mathbf{x},t)$ and $\mu(\mathbf{x},t)$ each satisfy an advection equation, and describe how a tesselation is used to track these material properties. In Section 5, we show empirical convergence rates obtained by applying our method to a two-dimensional test problem.

Next, we use this method to simulate red blood cells. We first study their equilibrium shapes by simulating red cell ghosts, that is, red cells that have been lysed so that their internal and external fluids are identical. After initializing the cell as a sphere and gradually decreasing its volume, using a reduced membrane model with only bending rigidity and area conservation, we recover the physiological biconcave disk shape. This equilibrium shape is subsequently used as the reference configuration for an in-plane shear resistance that is included in the full membrane model. We present two-phase simulations of red cells in shear flow and compare the tank treading and tumbling frequencies to those obtained in previous numerical studies. Further, we consider flow through thin capillaries, during which the cells assume realistic slipper-like shapes. Movies associated with these simulations can be accessed online [37].

2. Variable Coefficient Fluid Solver. Here, we describe the fluid solver that serves as the basis for our variable coefficient immersed boundary method. Consider an incompressible fluid with velocity $\mathbf{u}(\mathbf{x},t)$ and pressure $p(\mathbf{x},t)$ in a three dimensional domain with periodic boundary conditions. For now, we treat the variable density $\rho(\mathbf{x},t)$ and variable viscosity $\mu(\mathbf{x},t)$ as known functions to simplify the discussion (later on, we will take $\rho(\mathbf{x},t)$ and $\mu(\mathbf{x},t)$ to satisfy corresponding advection equations and describe how we solve for them). The Navier-Stokes equations are

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + \nabla p = \nabla\cdot\boldsymbol{\sigma} + \mathbf{f}, \tag{2.1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2.2}$$

in which the viscous stress tensor is given by $\boldsymbol{\sigma} := \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)$. Equation (2.1) expresses conservation of momentum, whereas equation (2.2) states that the volume of any given fluid parcel remains constant.

2.1. Spatial Discretization. Without loss of generality, we define the physical domain to be $[-\frac{1}{2},\frac{1}{2}]^3 \subseteq \mathbb{R}^3$ with periodic boundary conditions and coordinates $\mathbf{x} = (x_1, x_2, x_3)$. Except where otherwise noted, we use dimensionless variables. We discretize the physical domain by setting up a grid with N^3 cells and meshwidth $h = \Delta x_1 = \Delta x_2 = \Delta x_3$, where $h = 1/N$. The grid of cell centers, denoted g_0^h, is given by

$$g_0^h := \{((i+1/2)\,h, (j+1/2)\,h, (k+1/2)\,h)\,, \quad i,j,k = -N/2,\ldots,-1,0,1,\ldots,N/2-1\}\,.$$

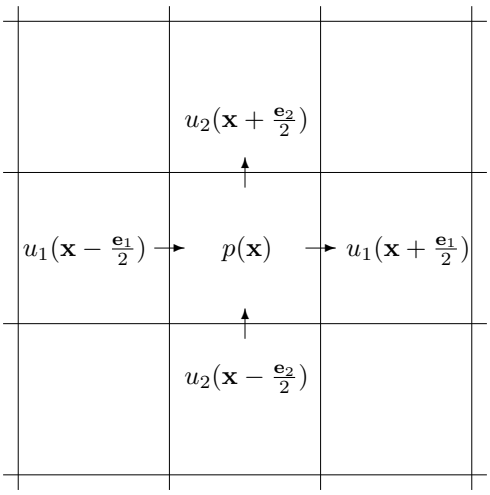


FIG. 2.1. *Locations of velocity* $\mathbf{u} = (u_1, u_2)$ *and pressure* p *on grid cell with center* $\mathbf{x} \in g_0^h$.

We employ a staggered discretization in which the components of the velocity vector $\mathbf{u} = (u_1, u_2, u_3)$ are defined on the faces of the unit cell to which they are normal. More precisely, for $\alpha = 1, 2, 3$, the component of velocity u_α is defined on the grid $g_\alpha^h := g_0^h - \frac{\mathbf{e}_\alpha}{2}$, where $\mathbf{e}_\alpha$ is a unit vector in the α direction. The pressure p is defined at cell centers. A two-dimensional version of the staggered scheme is illustrated in Figure 2.1.

We define finite difference operators D_α according to

$$(D_\alpha \phi)(\mathbf{x}) := \frac{\phi\left(\mathbf{x} + \frac{\mathbf{e}_\alpha}{2}\right) - \phi\left(\mathbf{x} - \frac{\mathbf{e}_\alpha}{2}\right)}{h},$$

for $\alpha = 1, 2, 3$. The discrete divergence operator $\mathbf{D}$ is defined by $\mathbf{D} \cdot \mathbf{u} := D_1 u_1 + D_2 u_2 + D_3 u_3$. This operator takes face-centered vector variables, such as the velocity field, to cell-centered scalars. Correspondingly, the discrete gradient operator $\mathbf{G}$ is defined by $(\mathbf{G}\phi)_\alpha := D_\alpha \phi$ and takes cell-centered to face-centered quantities. $L := \mathbf{D} \cdot \mathbf{G}$ is the standard seven-point approximation to the Laplacian in three dimensions. Let $\mathcal{S}(\mathbf{u})\phi$ denote the application of the discrete convective operator to a function ϕ. It is the discretization of $\frac{1}{2}\left((\mathbf{u} \cdot \nabla)\phi + \nabla \cdot (\mathbf{u}\phi)\right)$, and involves a combination of averaged centered differences, following Moin et al. [38]. To be precise, we define the averaging operator A_α by

$$(A_\alpha \phi)(\mathbf{x}) := \frac{\phi\left(\mathbf{x} + \frac{\mathbf{e}_\alpha}{2}\right) + \phi\left(\mathbf{x} - \frac{\mathbf{e}_\alpha}{2}\right)}{2}.$$

Then, for a vector field $\mathbf{v}$ on the staggered grid, the explicit form of the skew-symmetric convective operator is

$$\mathcal{S}(\mathbf{u})v_\alpha := \frac{1}{2}\sum_{\beta=1}^{3}\left(A_\beta\left(\left(A_\alpha u_\beta\right)\left(D_\beta v_\alpha\right)\right) + D_\beta\left(\left(A_\alpha u_\beta\right)\left(A_\beta v_\alpha\right)\right)\right), \tag{2.3}$$

for $\alpha = 1, 2, 3$.

The discretization of the viscous term is denoted by $\mathcal{L}(\mu)\mathbf{u}$. It is obtained by applying central differences to the divergence of the stress tensor $\left(\mathcal{L}(\mu)\mathbf{u}\right)_\alpha := D_\beta\left(\mu(\mathbf{x})\left(D_\alpha u_\beta + D_\beta u_\alpha\right)\right)$, here using the Einstein convention for summation over repeated indices. Expanding this expression, we have

$$\begin{aligned}\left(\mathcal{L}(\mu)\mathbf{u}\right)_\alpha(\mathbf{x}) = \frac{1}{h^2}\Big(&\mu(\mathbf{x} + \frac{\mathbf{e}_\beta}{2})\left(u_\beta(\mathbf{x} + \frac{\mathbf{e}_\alpha}{2} + \frac{\mathbf{e}_\beta}{2}) - u_\beta(\mathbf{x} - \frac{\mathbf{e}_\alpha}{2} + \frac{\mathbf{e}_\beta}{2}) + u_\alpha(\mathbf{x} + \mathbf{e}_\beta) - u_\alpha(\mathbf{x})\right)\\ &-\mu(\mathbf{x} - \frac{\mathbf{e}_\beta}{2})\left(u_\beta(\mathbf{x} + \frac{\mathbf{e}_\alpha}{2} - \frac{\mathbf{e}_\beta}{2}) - u_\beta(\mathbf{x} - \frac{\mathbf{e}_\alpha}{2} - \frac{\mathbf{e}_\beta}{2}) + u_\alpha(\mathbf{x}) - u_\alpha(\mathbf{x} - \mathbf{e}_\beta)\right)\Big).\end{aligned}$$

We evaluate $\mathcal{L}(\mu)\mathbf{u}$ at face centers for consistency with the staggered discretization of the velocity field. This requires the viscosity $\mu(\mathbf{x})$ to be evaluated at edge centers and cell centers. In the case that $\mu(\mathbf{x})$ is constant and the velocity field is discretely divergence-free, this discretization reduces to the standard seven-point approximation to the Laplacian. We note that the projection method used here ensures that the velocity field is discretely divergence free (to the tolerance of the linear solver).

2.2. Timestepping for Fluid Solver. We now consider the temporal discretization of the variable coefficient Navier-Stokes equations. We set a time step Δt and, for any mesh function $\phi(\mathbf{x}, t)$, define $\phi^n(\mathbf{x}) := \phi(\mathbf{x}, n\Delta t)$. Given an initial velocity $\mathbf{u}^n$ and a prescribed external force $\mathbf{f}^n$, we find the updated velocity $\mathbf{u}^{n+1}$. The pressure $p^{n+\frac{1}{2}}$, which is determined by requiring the velocity field to be incompressible, is also calculated at each step. We recall that, for now, $\mu(\mathbf{x}, t)$ and $\rho(\mathbf{x}, t)$ are considered to be given functions. Define

$$\overline{\mu} := \sup_{\mathbf{x},t} \mu(\mathbf{x}, t), \quad \overline{\rho} := \sup_{\mathbf{x},t} \rho(\mathbf{x}, t),$$

and set

$$r(\mathbf{x}, t) := \mu(\mathbf{x}, t) - \overline{\mu}, \quad s(\mathbf{x}, t) := \rho(\mathbf{x}, t) - \overline{\rho},$$

so that $r(\mathbf{x}, t)$ and $s(\mathbf{x}, t)$ are non-positive quantities representing the deviations from the maximum viscosity and density, respectively. The equations of motion are then discretized in time by the following predictor-corrector scheme:

$$\left.\begin{aligned} \overline{\rho}\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + s^n \frac{\mathbf{u}^n - \mathbf{u}^{n-1}}{\Delta t} + (\overline{\rho} + s^n)\,\mathcal{S}(\mathbf{u}^n)\mathbf{u}^n + \mathbf{G}\phi \\ = \mathcal{L}(r^n)\mathbf{u}^n + \overline{\mu}L\mathbf{u}^* + \mathbf{f}^*, \\ \mathbf{D}\cdot\mathbf{u}^* = 0, \end{aligned}\right\} \quad \text{predictor} \tag{2.4}$$

$$\left.\begin{aligned} \overline{\rho}\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \left(\frac{s^n + s^{n+1}}{2}\right)\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} \\ + \left(\overline{\rho} + \frac{s^n + s^{n+1}}{2}\right)\frac{\mathcal{S}(\mathbf{u}^n)\mathbf{u}^n + \mathcal{S}(\mathbf{u}^*)\mathbf{u}^*}{2} + \mathbf{G}p^{n+\frac{1}{2}} \\ = \frac{1}{2}\left(\mathcal{L}(\mu^n)\mathbf{u}^n + \overline{\mu}L\mathbf{u}^{n+1} + \mathcal{L}(r^{n+1})\mathbf{u}^*\right) + \frac{1}{2}\left(\mathbf{f}^n + \mathbf{f}^*\right), \\ \mathbf{D}\cdot\mathbf{u}^{n+1} = 0. \end{aligned}\right\} \quad \text{corrector} \tag{2.5}$$

The inertial and viscous terms are both split into two parts: a constant coefficient part that is treated implicitly, and a variable coefficient part that is treated explicitly. In the constant coefficient case, the scheme reduces to using the backward Euler rule for the viscous terms on the predictor step and the trapezoidal rule for the viscous terms on the corrector step. For the first step, we set $\mathbf{u}^{-1} = \mathbf{u}^0$.

Notice that the implicit solves involve only the standard 7-point approximation to the Laplacian, because $\mathcal{L}(\overline{\mu}) = \overline{\mu}L$. An explicit trapezoidal rule is used for the variable viscosity term involving $r^n(\mathbf{x})$. The variable density term is also treated explicitly, which on the predictor step requires the time derivative to be approximated using the previous velocity. The use of a lagged velocity is reminiscent of multistep methods such as Adams-Bashforth. Because the above scheme employs a mixture of explicit and implicit terms, and because negative density and viscosity terms appear in the formulation through $r^n(\mathbf{x})$ and $s^n(\mathbf{x})$, the stability properties of the scheme are not readily apparent. It is shown in a companion article [32], however, that a closely related scheme is stable and convergent. In Section 4, we demonstrate empirically that our timestepping method is second-order accurate for constant density problems with sufficiently smooth solutions and first-order accurate otherwise. We also show that second-order accuracy can be recovered in the variable density case by iterating the corrector step (equation (2.5)), although the iteration may not converge quickly enough to make this a practical numerical method. Although one might be tempted by the time-centered appearance of equation (2.5) to conclude second-order accuracy in the variable density case, the fact that the scheme is only first-order accuracy can be understood by analyzing a similar scheme for a simple ordinary differential equation (see Appendix B).

Next, we describe how to solve the above linear systems for the unknowns $\{\mathbf{u}^*, \phi\}$ in the predictor step and $\{\mathbf{u}^{n+1}, p^{n+\frac{1}{2}}\}$ in the corrector step. Both steps are solved using the same basic approach, which is described in detail in Devendran and Peskin [39]. (See also Griffith [40] for an alternative approach to solving these equations that is especially useful for problems that do not include periodic boundary conditions.) We summarize the procedure here in words:

1. Apply the discrete divergence operator to both sides of the momentum equation to obtain a Poisson equation for the pressure variable.

2. Solve the resulting Poisson equation in Fourier space.
3. Transform the pressure variable to physical space and calculate its gradient.
4. Solve the momentum equation in Fourier space, viewing the gradient of the pressure variable as a forcing term.

In our implementation, we use the Fourier transform in steps 2 and 4 to take advantage of the fact that the Laplacian operator is diagonal in Fourier space. In this way, we can use the FFT and avoid having to solve any linear systems through Gaussian elimination or other methods. The transformation of pressure back to physical space in step 3 may seem unnecessary. Indeed, it is possible to solve for pressure and velocity simultaneously, but this is not straightforward since pressure and velocity are located on different grids. Although here we make use of the FFT to solve the system of equations (2.4)-(2.5), note that the FFT is not an essential part of our method, and any fast Poisson solver could in principle be used.

3. Immersed Boundary Formulation. The following system of equations is the immersed boundary formulation of an elastic structure immersed in an incompressible fluid:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)+\nabla p=\nabla\cdot\left(\mu\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right)\right)+\mathbf{f}, \tag{3.1}$$

$$\nabla\cdot\mathbf{u}=0, \tag{3.2}$$

$$\frac{\partial}{\partial t}\rho(\mathbf{X}(\mathbf{q},t),t)=0, \tag{3.3}$$

$$\frac{\partial}{\partial t}\mu(\mathbf{X}(\mathbf{q},t),t)=0, \tag{3.4}$$

$$\mathbf{f}(\mathbf{x},t)=\int\mathbf{F}(\mathbf{q},t)\delta\left(\mathbf{x}-\mathbf{X}(\mathbf{q},t)\right)\mathrm{d}\mathbf{q}, \tag{3.5}$$

$$\frac{\partial\mathbf{X}}{\partial t}=\mathbf{U}(\mathbf{q},t)=\int\mathbf{u}(\mathbf{x},t)\delta\left(\mathbf{x}-\mathbf{X}(\mathbf{q},t)\right)\mathrm{d}\mathbf{x}, \tag{3.6}$$

$$\mathbf{F}(\mathbf{q},t)=\mathscr{F}[\mathbf{X}(\cdot,t)](\mathbf{q}). \tag{3.7}$$

Equations (3.1)-(3.2) are the Navier-Stokes equations with variable density $\rho(\mathbf{x},t)$ and variable viscosity $\mu(\mathbf{x},t)$. The material coordinate $\mathbf{q}$ denotes points that may be associated with either the fluid or the structure; in either case, $\mathbf{X}(\mathbf{q},t)$ gives the Cartesian position of material point $\mathbf{q}$ at time t. Equations (3.3) and (3.4) are advection equations for density and viscosity ensuring that these material properties are constant for any given fluid parcel. These assumptions hold for many problems of physical or biological interest, such as those involving immiscible fluids or different fluids separated by an impermeable elastic boundary, including the case of red blood cells to be considered in Section 6. The Lagrangian force density $\mathbf{F}(\mathbf{q},t)$ is determined by a functional of the configuration of the elastic material, denoted $\mathscr{F}[\mathbf{X}]$. For example, in some applications the material points are taken to be connected by springs and $\mathbf{F}(\mathbf{q},t)$ is computed by Hooke's law.

Equations (3.5) and (3.6) are called the interaction equations because they describe the communication between Eulerian and Lagrangian variables. Dirac delta functions are used both to transfer the Lagrangian force to the Eulerian frame and to interpolate velocities from Eulerian to Lagrangian coordinates. Although the same delta function is used, these two operations are fundamentally different in that $\mathbf{u}(\mathbf{x},t)$ and $\mathbf{U}(\mathbf{q},t)$ have the same pointwise values according to $\mathbf{U}(\mathbf{q},t)=\mathbf{u}\left(\mathbf{X}(\mathbf{q},t),t\right)$ whereas the pointwise values of $\mathbf{f}(\mathbf{x},t)$ and $\mathbf{F}(\mathbf{q},t)$ are not equal. Instead, they are related as densities, since for an arbitrary region Ω,

$$\int_\Omega\mathbf{f}(\mathbf{x},t)\,\mathrm{d}\mathbf{x}=\int_{\mathbf{X}(\mathbf{q},t)\in\Omega}\mathbf{F}(\mathbf{q},t)\,\mathrm{d}\mathbf{q}.$$

3.1. Delaunay Construction. As in many other applications of the immersed boundary method, Lagrangian coordinates $\mathbf{q}$ are used for the elastic structures. In the present method, we use a Lagrangian description for the density and viscosity as well. Let $\rho_{\mathrm{L}}(\mathbf{q})$ and $\mu_{\mathrm{L}}(\mathbf{q})$ be defined by

$$\rho_{\mathrm{L}}(\mathbf{q}):=\rho(\mathbf{X}(\mathbf{q},t),t),\quad \mu_{\mathrm{L}}(\mathbf{q}):=\mu(\mathbf{X}(\mathbf{q},t),t). \tag{3.8}$$

Although they are functions of Lagrangian coordinates, the values of ρ_{L} and μ_{L} are the Eulerian density and viscosity, respectively. This means that ρ_{L} is *not* the mass per unit $\mathbf{q}$, except in the special case that $\det\left(\frac{\partial\mathbf{X}}{\partial\mathbf{q}}\right)=1$.

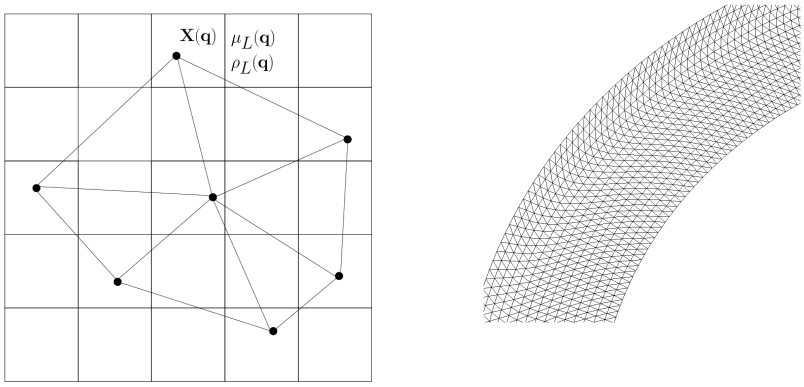


FIG. 3.1. *(a) A triangulation is used to define ρ and μ throughout the domain, (b) A close-up of the triangulated region of continuously varying viscosity and density for the test problem of the thick elastic shell. Note that this mesh is used only to define the density and viscosity.*

We next describe how we define $\rho(\mathbf{x},t)$ and $\mu(\mathbf{x},t)$ in the discrete setting. First, the Lagrangian space is overlaid with a collection of points $G_{\Delta\mathbf{q}}$, and we assign $\rho_{\mathrm{L}}(\mathbf{q})$ and $\mu_{\mathrm{L}}(\mathbf{q})$ for $\mathbf{q} \in G_{\Delta\mathbf{q}}$ based on the initial conditions. The points in $G_{\Delta\mathbf{q}}$ may represent material points in the fluid *or* the immersed structure. We then perform a Delaunay tesselation [41] in Cartesian space, with $\mathbf{X}(\mathbf{q},t)$ for $\mathbf{q} \in G_{\Delta\mathbf{q}}$ the Cartesian positions of the nodes of the elements (see Figure 3.1(a)). To calculate the density or viscosity at a given point in the domain, such as a grid point, we interpolate from the nodes of the element containing that point. Because the nodes move with the fluid, this formulation ensures that the density and viscosity are advected with the fluid as well. To interpolate from the nodes, we use linear interpolation. Let $\psi_{\mathbf{q}}(\mathbf{x},t)$ be a piecewise linear function on each element of the tesselation satisfying

$$\psi_{\mathbf{q}}(\mathbf{x},t) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{X}(\mathbf{q},t), \\ 0 & \text{if } \mathbf{x} = \mathbf{X}(\mathbf{q}',t) \text{ and } \mathbf{q}' \neq \mathbf{q}, \end{cases} \tag{3.9}$$

for $\mathbf{q}$, $\mathbf{q}' \in G_{\Delta\mathbf{q}}$. Thus, $\psi_{\mathbf{q}}(\mathbf{x},t)$ is a "hat function" peaked at $\mathbf{X}(\mathbf{q},t)$. We then define

$$\rho(\mathbf{x},t) := \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} \rho_{\mathrm{L}}(\mathbf{q})\psi_{\mathbf{q}}(\mathbf{x},t), \quad \mu(\mathbf{x},t) := \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} \mu_{\mathrm{L}}(\mathbf{q})\psi_{\mathbf{q}}(\mathbf{x},t). \tag{3.10}$$

In particular, this ensures that $\rho(\mathbf{X}(\mathbf{q},t),t) = \rho_{\mathrm{L}}(\mathbf{q})$ and $\mu(\mathbf{X}(\mathbf{q},t),t) = \mu_{\mathrm{L}}(\mathbf{q})$.

In many applications, as in the test problem of the thick viscoelastic shell in Section 5, a region of variable density and variable viscosity is contained within a constant coefficient fluid. In this case, we may reduce the computational cost by restricting the Delaunay tesselation to the region in which the density or viscosity is different from that of the ambient fluid. Additionally, as the simulation proceeds, the mesh may become distorted. It is important that the tesselation elements form a partition at all times. To avoid mesh tangling, we use a mesh generator (such as Triangle [42] in 2D or TetGen [43] in 3D) and retesselate periodically. Sometimes, especially if there is an angle constraint on the mesh, retesselation requires the insertion of additional Lagrangian points. In this case, we assign densities and viscosities to the additional Lagrangian points via linear interpolation (i.e. the equations in (3.10)) using the previous tesselation. We further note that, although the original tesselation is Delaunay, the conditions of a Delaunay tesselation are not satisfied for all time since the elements change shape as the nodes move with the fluid. Of course, if we retesselate, we have a Delaunay tesselation once more.

3.2. Regularized Delta Function. One of the characteristics of the immersed boundary method is its use of a regularized delta function to relate quantities on the Eulerian and Lagrangian grids, as made explicit in the interaction equations (3.5) and (3.6). A discussion of the properties desired from such a delta function and a derivation of the one-dimensional regularized delta function, $\phi(r)$, can be found in [44]. Here, we simply

state the result:

$$\phi(r) = \begin{cases} \frac{1}{8}\left(5 + 2r - \sqrt{-7 - 12r - 4r^2}\right), & -2 \le r < -1, \\ \frac{1}{8}\left(3 + 2r + \sqrt{1 - 4r - 4r^2}\right), & -1 \le r < 0, \\ \frac{1}{8}\left(3 - 2r + \sqrt{1 + 4r - 4r^2}\right), & 0 \le r < 1, \\ \frac{1}{8}\left(5 - 2r - \sqrt{-7 + 12r - 4r^2}\right), & 1 \le r \le 2, \\ 0, & |r| > 2. \end{cases}$$

The three-dimensional delta function is the product $\delta_h(\mathbf{x}) := \frac{1}{h^3}\phi\left(\frac{x_1}{h}\right)\phi\left(\frac{x_2}{h}\right)\phi\left(\frac{x_3}{h}\right)$. This three-dimensional regularized delta function has a support of 4 grid points in each coordinate direction. We remark that different delta functions have also been used in the immersed boundary method, and determining the properties of suitable delta functions is an area of active research [45].

3.3. Timestepping for Immersed Boundary Method. We next describe the timestepping scheme for the variable coefficient immersed boundary method, which is based on the fluid solver described above. First, we interpolate $\mathbf{u}^n$ at $\mathbf{X}^n$ with the help of the delta function to find the velocities $\mathbf{U}^n$ of the Lagrangian points. Then, we apply the forward Euler method to equation (3.6) to compute $\mathbf{X}^*$, a preliminary approximation to $\mathbf{X}^{n+1}$:

$$U_\alpha^n(\mathbf{q}) = \sum_{\mathbf{x}\in g_\alpha^h} u_\alpha^n(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}^n(\mathbf{q}))h^3, \quad \alpha = 1,\, 2,\, 3, \tag{3.11}$$

$$\mathbf{X}^*(\mathbf{q}) = \mathbf{X}^n(\mathbf{q}) + \Delta t\mathbf{U}^n(\mathbf{q}), \tag{3.12}$$

for $\mathbf{q} \in G_{\Delta\mathbf{q}}$. Notice that the summation in equation (3.11) need only be evaluated over the support of the delta function. Next, the Lagrangian force densities $\mathbf{F}^n$ and $\mathbf{F}^*$ are computed from $\mathbf{X}^n$ and $\mathbf{X}^*$, respectively, using a discrete approximation to equation (3.7). The forcing function is determined by the application, and we leave it unspecified for now. The Eulerian force densities $\mathbf{f}^n$ and $\mathbf{f}^*$ are then computed by spreading the Lagrangian force densities to the grid.

$$f_\alpha^n(\mathbf{x}) = \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} F_\alpha^n(\mathbf{q})\delta_h(\mathbf{x} - \mathbf{X}^n(\mathbf{q}))\Delta\mathbf{q}, \quad f_\alpha^*(\mathbf{x}) = \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} F_\alpha^*(\mathbf{q})\delta_h(\mathbf{x} - \mathbf{X}^*(\mathbf{q}))\Delta\mathbf{q}, \tag{3.13}$$

for $\mathbf{x} \in g_\alpha^h$ and $\alpha = 1,\, 2,\, 3$. The density and viscosity are determined by interpolating from the nodes of the tesselation. Let $\psi_\mathbf{q}^n(\mathbf{x})$ and $\psi_\mathbf{q}^*(\mathbf{x})$ be the piecewise linear functions determined from $\mathbf{X}^n$ and $\mathbf{X}^*$, respectively, as in equation (3.9). Then, set

$$\rho^n(\mathbf{x}) = \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} \rho_{\mathrm{L}}(\mathbf{q})\psi_\mathbf{q}^n(\mathbf{x}), \quad \mu^n(\mathbf{x}) = \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} \mu_{\mathrm{L}}(\mathbf{q})\psi_\mathbf{q}^n(\mathbf{x}), \tag{3.14}$$

and

$$\rho^*(\mathbf{x}) = \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} \rho_{\mathrm{L}}(\mathbf{q})\psi_\mathbf{q}^*(\mathbf{x}), \quad \mu^*(\mathbf{x}) = \sum_{\mathbf{q}\in G_{\Delta\mathbf{q}}} \mu_{\mathrm{L}}(\mathbf{q})\psi_\mathbf{q}^*(\mathbf{x}). \tag{3.15}$$

A feature of the triangulation that defines $\rho(\mathbf{x}, t)$ and $\mu(\mathbf{x}, t)$ is that these quantities make sense not only on a grid but throughout the whole domain. For practical purposes, however, we only need the density at face centers, and the viscosity at cell centers and edge centers. We find that an efficient way to compute the densities and viscosities on these grids is to loop over tesselation elements, identifying the grid points contained in each element and linearly interpolating the appropriate quantity from the nodes using barycentric coordinates. This approach parallelizes easily over the tesselation elements. Note that identifying which grid points are contained in an element is a local operation; for linear elements, we need only test grid points within the element's bounding box.

Now, let $\overline{\mu}$ be the maximum of $\mu_{\mathrm{L}}(\mathbf{q})$ over $G_{\Delta\mathbf{q}}$ and let $\overline{\rho}$ be the maximum of $\rho_{\mathrm{L}}(\mathbf{q})$ over $G_{\Delta\mathbf{q}}$. Define

$$r^n(\mathbf{x}) := \mu^n(\mathbf{x}) - \overline{\mu}, \quad s^n(\mathbf{x}) := \rho^n(\mathbf{x}) - \overline{\rho},$$

and

$$r^*(\mathbf{x}) := \mu^*(\mathbf{x}) - \overline{\mu}, \quad s^*(\mathbf{x}) := \rho^*(\mathbf{x}) - \overline{\rho}.$$

Because $\rho^n(\mathbf{x})$ and $\mu^n(\mathbf{x})$ are calculated by interpolation from the values $\rho_{\mathrm{L}}(\mathbf{q})$ and $\mu_{\mathrm{L}}(\mathbf{q})$, respectively, $r^n(\mathbf{x})$ and $s^n(\mathbf{x})$ are non-positive. We then solve the predictor-corrector scheme of equations (2.4)-(2.5) as before, except that in the corrector, we now use the predicted values of r^* and s^* time step $n+1$, since the analytical density and viscosity used in Section 2.2 are not available here. Finally, we interpolate $\mathbf{u}^{n+1}$ at $\mathbf{X}^*$ to get $\mathbf{U}^*$ and advance the Lagrangian positions to the end of the time step

$$U_\alpha^*(\mathbf{q}) = \sum_{\mathbf{x}\in g_\alpha^h} u_\alpha^{n+1}(\mathbf{x})\delta_h(\mathbf{x}-\mathbf{X}^*(\mathbf{q}))h^3, \quad \alpha = 1,\, 2,\, 3, \tag{3.16}$$

$$\mathbf{X}^{n+1}(\mathbf{q}) = \mathbf{X}^n(\mathbf{q}) + \frac{\Delta t}{2}\left(\mathbf{U}^n(\mathbf{q}) + \mathbf{U}^*(\mathbf{q})\right). \tag{3.17}$$

In the following section, we show empirical results demonstrating that this method is second-order accurate for constant density problems and first-order accurate otherwise. Thanks to the splitting we use for the constant coefficient and variable coefficient terms, this method is stable far beyond the diffusive CFL limit for purely explicit schemes, although we may still need to take small time steps in problems with nontrivial elastic stresses.

4. Empirical Convergence Rates for Variable Coefficient Fluid Without Elasticity. To test our scheme, we use a two-dimensional test problem adapted from Griffith and Peskin [46]. Consider a fluid in the domain $[-\frac{1}{2},\frac{1}{2}]^2 \subset \mathbb{R}^2$ without elasticity and with an annular region. The fluid properties vary in such a manner that they match the properties of the ambient fluid at the boundary of the annulus.

We use a Lagrangian description of the annular region of variable viscosity and density. The Lagrangian domain is $[0,1]^2$ with coordinates $\mathbf{q} = (q_1, q_2)$. We discretize the Lagrangian domain with an $N_1 \times N_2$ grid, $G_{\Delta\mathbf{q}}$, in which the spacing is given by $\Delta q_1 = 1/N_1$ and $\Delta q_2 = 1/N_2$ so that

$$G_{\Delta\mathbf{q}} = \left\{ \left(\left(m+\frac{1}{2}\right)\Delta q_1, \left(n+\frac{1}{2}\right)\Delta q_2 \right), \qquad m = 0,1,\ldots,N_1-1, \quad n = 0,1,\ldots,N_2-1 \right\}.$$

We take $N_1 = \frac{1}{8}N$ and $N_2 = \frac{50}{16}N$, where N is the number of Eulerian grid points in each coordinate direction. This makes the spacing between Lagrangian points in the fluid domain approximately half a meshwidth. At time $t = 0$, the Lagrangian coordinates are mapped to the Eulerian frame by

$$\mathbf{X}^0(q_1,q_2) = \left(\left(\alpha + \gamma\left(q_1 - \frac{1}{2}\right)\right)\cos(2\pi q_2), \left(\beta + \gamma\left(q_1-\frac{1}{2}\right)\right)\sin(2\pi q_2) \right).$$

Thus, the coordinate q_1 labels concentric bands, whereas q_2 denotes position along the circumference of such a band. As in previous empirical studies of the immersed boundary method [46], we take $\alpha = 0.2$, $\beta = 0.25$, and $\gamma = 0.3$. This choice of parameters results in an elliptical initial configuration for the annulus.

As noted in Section 3.1, we may restrict the triangulation to the region of variable viscosity and variable density. A close-up of the triangulation is shown in Figure 3.1(b). It was mentioned in Section 3.3 that the density is only required at face centers (edge centers in 2D), and the viscosity is only required at cell centers and edge centers (cell corners in 2D). In this computation, we find it convenient to evaluate the viscosity at edge centers and then interpolate to cell centers and cell corners using centered averages.

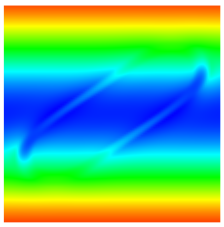
The initial velocity varies sinusoidally as $\mathbf{u}(\mathbf{x},0) = \begin{pmatrix}\sin(2\pi x_2)\\ 0\end{pmatrix}$. No initial pressure is required. We consider $N = 256$, 512, and 1048 with a time step size of $\Delta t = 0.1\Delta x$, and the simulation is run up to $t = 0.4$. The density of the surrounding fluid is $\rho = 1.0$, and the viscosity of the surrounding fluid is $\mu = 0.01$. Although dimensionless units are used here, these parameters are chosen to be representative of water in cgs units. The resulting flows have moderate Reynolds numbers in the range 10-100.

The constant density and variable density cases are investigated separately. In both cases, the viscosity is given by

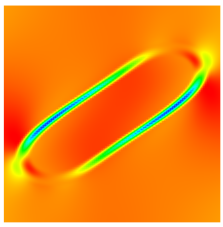
$$\mu_{\mathrm{L}}(\mathbf{q}) = \mu_0 + \frac{\mu_0}{2}\left(1 + \sin\left(2\pi q_1 - \frac{\pi}{2}\right)\right).$$

In the variable density case, the density is given by

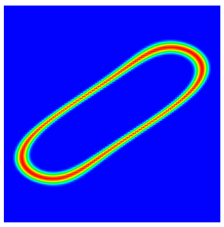
$$\rho_{\mathrm{L}}(\mathbf{q}) = \rho_0 + 5\rho_0\left(1 + \sin\left(2\pi q_1 - \frac{\pi}{2}\right)\right).$$



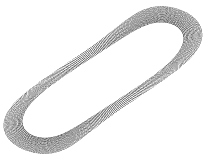
(a) Vorticity



(b) Pressure



(c) Viscosity



(d) Positions of immersed boundary points

FIG. 4.1. *Vorticity, pressure, viscosity, and Lagrangian positions at $t = 0.23$ for the variable coefficient fluid without elasticity.*

q	constant density		variable density	
	$r_1[q;256]$	$r_2[q;256]$	$r_1[q;256]$	$r_2[q;256]$
u_1	1.99	1.98	1.15	1.11
u_2	1.98	1.99	1.08	1.10
$\mathbf{X}$	1.99	1.99	1.34	1.31
p	1.99	2.00	1.16	1.20

TABLE 4.1

Empirical orders of accuracy of the predictor-corrector scheme for a fluid without elasticity in the constant density and variable density cases.

Snapshots of the vorticity, pressure field, viscosity, and position are shown in Figure 4.1 to illustrate the qualitative features of the flow.

We compute empirical convergence rates for this problem. Because we do not have an analytic form of the exact solution, we approximate the error by comparing numerical solutions on successively refined grids. More precisely, let $e_p[q^N, q^{2N}]$ be an approximation to the L^p error of the computed solution q obtained by comparison to the computed solution on the next finer mesh, i.e. $e_p[q^N, q^{2N}] := \|q^N - \mathscr{I}^{2N\to N} q^{2N}\|_p$, where $\mathscr{I}^{2N\to N}$ stands for interpolation to the coarser mesh and the discrete L^p norm for a scalar function ψ on g_α^h is defined by $\|\psi\|_p := \left(\sum_{\mathbf{x}\in g_\alpha^h} |\psi(\mathbf{x})|^p h^3\right)^{1/p}$. The empirical convergence rate, $r_p[q;N]$, is then given by

$$r_p[q;N] := \log_2\left(\frac{e_p[q^N, q^{2N}]}{e_p[q^{2N}, q^{4N}]}\right).$$

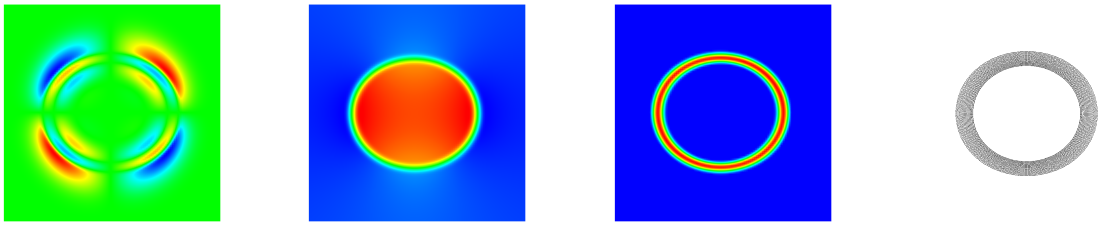
For a vector field $\mathbf{X}$, we define $\|\mathbf{X}\|_p := \|\sqrt{\mathbf{X}\cdot\mathbf{X}}\|_p$. As shown in Table 4.1, the empirical order of accuracy is close to second-order in the constant density case, and closer to first-order in the variable density case. To recover second-order accuracy in the variable density case, we iterate the corrector step (equation (2.5)) so that the timestepping scheme becomes

$$\begin{aligned}
&\overline{\rho}\frac{\mathbf{u}^{n+1,m+1}-\mathbf{u}^n}{\Delta t} + \left(\frac{s^n+s^*}{2}\right)\frac{\mathbf{u}^{n+1,m}-\mathbf{u}^n}{\Delta t} \\
&\quad + \left(\overline{\rho} + \frac{s^n+s^*}{2}\right)\frac{\mathcal{S}(\mathbf{u}^n)\mathbf{u}^n + \mathcal{S}(\mathbf{u}^{n+1,m})\mathbf{u}^{n+1,m}}{2} + \mathbf{G}p^{n+\frac{1}{2}} \\
&\qquad\qquad = \frac{1}{2}\left(\mathcal{L}(\mu^n)\mathbf{u}^n + \overline{\mu}L\mathbf{u}^{n+1,m+1} + \mathcal{L}(r^*)\mathbf{u}^{n+1,m}\right) + \frac{1}{2}\left(\mathbf{f}^n + \mathbf{f}^*\right), \\
&\mathbf{D}\cdot\mathbf{u}^{n+1,m+1} = 0,
\end{aligned} \tag{4.1}$$

where $\mathbf{u}^{n+1,m}$ is the m^{th} iterate. We have found that 10 steps of fixed-point iteration are sufficient to yield second-order accuracy, and accordingly we set $\mathbf{u}^{n+1} = \mathbf{u}^{n+1,10}$. Note that each iteration requires only the solution of the constant-coefficient incompressible Stokes equations. In particular, we do not re-evaluate the Lagrangian-Eulerian interaction equations. The predictor step is now used only to furnish an initial guess

q	iterative scheme with variable density	
	$r_1[q;256]$	$r_2[q;256]$
u_1	1.94	1.97
u_2	2.07	2.00
$\mathbf{X}$	2.01	2.00
p	1.96	1.96

TABLE 4.2
Empirical order of accuracy of the iterative scheme for a variable density fluid without elasticity.



(a) Vorticity (b) Pressure (c) Viscosity (d) Positions of immersed boundary points

FIG. 5.1. *Vorticity, pressure, viscosity, and Lagrangian positions at* $t = 0.16$ *for the problem of the elastic shell.*

$\mathbf{u}^{n+1,0}$. As shown in Table 4.2, convergence rates near second order are observed for this iterative scheme. However, even though each iteration is relatively fast, this procedure may not converge quickly enough to be practical: we find that the number of iterations required for convergence increases as the grid is refined.

5. Empirical Convergence Rates for Variable Coefficient Elastic Shell. To test the convergence of the variable density and variable viscosity immersed boundary method, we consider a thick viscoelastic shell immersed in fluid and undergoing damped oscillations. The shell can be interpreted as an idealized anisotropic neo-Hookean (rubber-like) material. The fluid properties and fiber stiffness of the shell vary continuously in such a manner that they match the properties of the fluid (in particular, zero stiffness) at the shell-fluid interface. The Lagrangian descriptions of the density and viscosity are as in Section 4. In addition to denoting a region of variable viscosity and variable density, $\mathbf{X}(\mathbf{q})$ serves as the position of a thick elastic shell whose fiber strength varies continuously. The forces generated by the fibers are given by

$$\mathbf{F}(\mathbf{q}) = \left(1 + \sin\left(2\pi q_1 - \frac{\pi}{2}\right)\right) \frac{\partial^2 \mathbf{X}}{\partial q_2^2},$$

where the standard central difference operator is used to approximate the second derivative on the Lagrangian grid. The fluid is initially at rest. Apart from the inclusion of fiber elasticity and the initial conditions for velocity, the simulations are identical to those of the previous section. See Figure 5.1 for snapshots from the simulation of the immersed elastic shell.

The resolution of both the Eulerian and Lagrangian meshes is increased, and results at successive resolutions are compared as in the previous section to compute convergence rates. As shown in Table 5.1, the empirical order of accuracy from the test problem of the thick viscoelastic shell is very near second-order in the constant density case, and closer to first-order in the variable density case. As described in the previous section, we can iterate the corrector step via equation (4.1) to obtain second-order accuracy in the variable density case (see Table 5.2). As above, we find that 10 iterations are sufficient to obtain second-order convergence rates under grid refinement.

6. Red Blood Cell Simulation. We apply our method to simulate red blood cells under various flow conditions and validate our model by comparison to previous numerical work. This is a strong test because of the computational challenges involved, including the physical stiffness of the locally area-preserving red cell membrane and the inherently three-dimensional large deformations that the red cell experiences. In addition

q	constant density		variable density	
	$r_1[q;256]$	$r_2[q;256]$	$r_1[q;256]$	$r_2[q;256]$
u_1	2.00	1.99	1.10	1.03
u_2	2.00	2.00	1.14	1.07
$\mathbf{X}$	1.98	1.97	1.28	1.29
p	1.99	1.99	1.58	1.50

TABLE 5.1

Empirical orders of accuracy of the predictor-corrector scheme for an elastic shell in the constant density and variable density cases.

q	iterative scheme with variable density	
	$r_1[q;256]$	$r_2[q;256]$
u_1	1.97	2.00
u_2	1.97	2.00
$\mathbf{X}$	1.98	1.97
p	1.99	1.99

TABLE 5.2

Empirical order of accuracy of the iterative scheme for an elastic shell with variable density.

to reproducing previous results, which included variable viscosity but not variable density, we account for the different densities inside and outside the red cells. The density of blood plasma is approximately 1.02 g/cm^3, whereas it is around 1.1 g/cm^3 for the hemoglobin solution contained by red cells [47]. We compute red cell equilibrium shapes based on our membrane model and simulate red cells in shear flow and capillary flow.

6.1. Membrane Model. We use a surface neo-Hookean elasticity model introduced by Evans and Skalak [48] to enforce the membrane's stretching and shear resistance. The neo-Hookean energy W_{nH} is defined with respect to a reference configuration that is generated in a preliminary calculation. We also include a term proportional to the total curvature squared, W_{curv}, that models bending rigidity, following Canham [49] and Helfrich [50]. The total energy W of the membrane is given by $W := W_{\mathrm{nH}} + W_{\mathrm{curv}}$. To motivate the definition of W_{nH}, we first consider the case of a three-dimensional *solid* with bulk modulus κ and shear modulus E, and we express the neo-Hookean energy in terms of generalized curvilinear coordinates in the reference configuration. Suppose that the solid is parameterized by material coordinates $\mathbf{q} = (q_1, q_2, q_3)$. Let the membrane position be denoted by $\mathbf{X}(\mathbf{q})$ and the reference position be denoted by $\mathbf{Z}(\mathbf{q})$. The neo-Hookean energy for this solid is

$$\frac{\kappa}{2}\int_{\mathbf{q}} (J-1)^2 \left(\det G_0\right)^{1/2} \mathrm{d}\mathbf{q} + \frac{E}{2}\int_{\mathbf{q}} \left(\mathrm{tr}\left(GG_0^{-1}\right)J^{-2/3} - 3\right)\left(\det G_0\right)^{1/2}\mathrm{d}\mathbf{q}, \tag{6.1}$$

where the 3×3 matrices G and G_0 are defined according to

$$G_{ij} := \frac{\partial \mathbf{X}}{\partial q_i}\cdot\frac{\partial \mathbf{X}}{\partial q_j}, \quad (G_0)_{ij} := \frac{\partial \mathbf{Z}}{\partial q_i}\cdot\frac{\partial \mathbf{Z}}{\partial q_j}, \tag{6.2}$$

and $J = \det\left(\frac{\partial \mathbf{X}}{\partial \mathbf{q}}\right) \Big/ \det\left(\frac{\partial \mathbf{Z}}{\partial \mathbf{q}}\right)$. Since $G = \left(\frac{\partial \mathbf{X}}{\partial \mathbf{q}}\right)^T\left(\frac{\partial \mathbf{X}}{\partial \mathbf{q}}\right)$ and $G_0 = \left(\frac{\partial \mathbf{Z}}{\partial \mathbf{q}}\right)^T\left(\frac{\partial \mathbf{Z}}{\partial \mathbf{q}}\right)$, G and G_0 are symmetric and positive-definite. Further, $J = \left(\frac{\det G}{\det G_0}\right)^{1/2} = \left(\det GG_0^{-1}\right)^{1/2}$.

Now, suppose that we have an elastic membrane parameterized by $\mathbf{q} = (q_1, q_2)$. We may define an analogous neo-Hookean energy for this membrane via

$$W_{\mathrm{nH}} := \frac{\kappa}{2}\int_{\mathbf{q}} (J-1)^2 \left(\det G_0\right)^{1/2} \mathrm{d}\mathbf{q} + \frac{E}{2}\int_{\mathbf{q}} \left(\mathrm{tr}\left(GG_0^{-1}\right)J^{-1} - 2\right)\left(\det G_0\right)^{1/2}\mathrm{d}\mathbf{q}, \tag{6.3}$$

where the volume integrals of equation (6.1) have been replaced by surface integrals and the matrices G and G_0 are 2×2, even though the equations in (6.2) are still applicable with $\mathbf{X}$ and $\mathbf{Z}$ in $\mathbb{R}^3$.

Equation (6.3) provides a coordinate-free version of the elastic energy of a neo-Hookean membrane. To recover the form that is more prevalent in the literature, we use the cyclic properties of trace and determinant to write W_{nH} in the symmetric form

$$\begin{aligned} W_{\mathrm{nH}} = \frac{\kappa}{2}\int_{\mathbf{q}} \left(\det\left(G_0^{-1/2} G G_0^{-1/2}\right)^{1/2} - 1\right)^2 (\det G_0)^{1/2}\,\mathrm{d}\mathbf{q} \\ + \frac{E}{2}\int_{\mathbf{q}} \left(\frac{\mathrm{tr}\left(G_0^{-1/2} G G_0^{-1/2}\right)}{\det\left(G_0^{-1/2} G G_0^{-1/2}\right)^{1/2}} - 2\right)(\det G_0)^{1/2}\,\mathrm{d}\mathbf{q}. \end{aligned} \tag{6.4}$$

Let λ_1^2 and λ_2^2 be the (positive) eigenvalues of the matrix $G_0^{-1/2} G G_0^{-1/2}$. It follows that

$$W_{\mathrm{nH}} = \frac{\kappa}{2}\int_{\mathbf{q}} (\lambda_1\lambda_2 - 1)^2 (\det G_0)^{1/2}\,\mathrm{d}\mathbf{q} + \frac{E}{2}\int_{\mathbf{q}} \left(\left(\frac{\lambda_1^2+\lambda_2^2}{\lambda_1\lambda_2}\right) - 2\right)(\det G_0)^{1/2}\,\mathrm{d}\mathbf{q}, \tag{6.5}$$

where λ_1 and λ_2 are the principal stretches. Equations (6.4) and (6.5) are given here for expository purposes only; to compute W_{nH}, we actually use the form of equation (6.3).

For our simulations, we use the values $E = 2.5 \times 10^{-3}\,\mathrm{dyn/cm}$ [51] for the shear modulus and $\kappa = CE$ for the bulk modulus, in which C is a numerical parameter reflecting the trade-off between area conservation and stiffness. As $\kappa \to \infty$, the membrane becomes incompressible in the 2D sense, i.e. area preserving, but the largest stable time step of the present scheme tends to zero. In practice, we choose C so that the average local error in area is less than 1%, with typical values in the range $C = 8 - 400$.

Lastly, the curvature energy over the surface S is given by

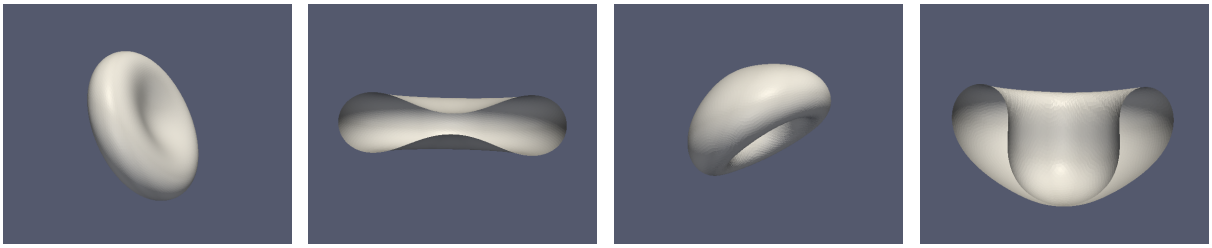
$$W_{\mathrm{curv}} := \frac{k}{2}\int_S H^2 \mathrm{d}S, \tag{6.6}$$

where H is the total curvature, i.e. the sum of the two principal curvatures, and $k = 2.0 \times 10^{-12}\,\mathrm{erg}$ [51] is the bending rigidity. The curvature energy is distinguished from the neo-Hookean energy in that it is independent of any reference configuration.

In the discrete setting, we define a collection $G_{\Delta\mathbf{q}}$ of Lagrangian points with corresponding positions in Eulerian space $\mathbf{X}(\mathbf{q})$ for $\mathbf{q} \in G_{\Delta\mathbf{q}}$. We triangulate the red cell membrane using these nodes and discretize the integrals above in terms of the resulting triangulated surface. The details of these discretizations are described in Appendix A. Once the energy W is written in terms of the $\mathbf{X}(\mathbf{q})$, it is differentiated to obtain the resulting force, $\mathbf{F}(\mathbf{q})$, via $\mathbf{F}(\mathbf{q}) = -\nabla_{\mathbf{q}} W$, where $\nabla_{\mathbf{q}}$ denotes the gradient with respect to the position $\mathbf{X}(\mathbf{q})$. We remark that $\mathbf{F}(\mathbf{q})$ is the force, not the force density, applied by node $\mathbf{q}$ to the fluid. In particular, if the Lagrangian mesh is refined indefinitely, $\mathbf{F}(\mathbf{q}) \to 0$ for any particular node $\mathbf{q}$, but the sum of $\mathbf{F}(\mathbf{q})$ over all nodes in some specified region of space remains finite. In our implementation, the differention required to evaluate $\mathbf{F}(\mathbf{q})$ is done analytically; see Appendix A for details.

6.2. Equilibrium Shapes. The surface area, A, of the red blood cell is nearly constant. In fact, the cell bursts if its area increases by as little as 2-4% [52]. If the cell were porous and its enclosed volume could change, then it would assume a spherical shape to minimize its curvature energy. However, because the enclosed volume, V, is also conserved and less than the volume of the sphere with the same surface area, the cell must find another equilibrium shape. The "reduced volume", v_{red}, is the ratio of the cell volume to the volume of a sphere with the same surface area, so that $v_{\mathrm{red}} := V \Big/ \left(\frac{A^{3/2}}{3\sqrt{4\pi}}\right)$. For red cells, $v_{\mathrm{red}} \approx 0.59$, with typical values for the volume around $V = 90\,\mu\mathrm{m}^3$ and surface area around $A = 137\,\mu\mathrm{m}^2$.

An important issue is the choice of reference configuration, denoted above by $\mathbf{Z}(\mathbf{q})$. One approach is to use a sphere with the target surface area as the reference, but this stabilizes the cup-shape over the biconcave shape and requires an artificially high bending rigidity [28]. Another, more effective approach is to use analytic functions that have been fit to experimental resting shapes, such as the widely-used function of Evans and Fung [53]. In our method, no fit to data is required. We calculate the equilibrium shape by setting the shear modulus to $E = 0$, so that the only relevant information supplied by the reference configuration is the target surface area of the membrane, in which case using a spherical reference shape does not stabilize the cup-shape over the biconcave shape. As explained by Li et al. [28], neglecting the shear modulus is justified since the cytoskeleton undergoes constant remodeling on a timescale such that the memory associated with



(a) v_{red}=0.59 (b) v_{red}=0.59 cross section (c) v_{red}=0.41 (d) v_{red}=0.41 cross section

FIG. 6.1. *Equilibrium configurations for the red cell membrane model with varying reduced volume fraction v_{red}.*

any macroscopic shape is short-lived compared to the duration of these deformations. We find that this process produces the physiological equilibrium shapes, including the characteristic biconcave disk when $v_{\text{red}} = 0.59$. The equilibrium shapes obtained using $v_{\text{red}} = 0.59$ and $v_{\text{red}} = 0.41$ are shown in Figure 6.1.

To perform these equilibrium shape computations, we initialize the cell as a sphere with area equal to the target surface area. Then, using our immersed boundary method solver, we slowly shrink the cell to the target volume by specifying that the membrane move normal to itself at a constant rate relative to the fluid velocity. The surface area is approximately conserved through the neo-Hookean elastic force acting on the membrane.

Before it reaches equilibrium, the membrane goes through many transitional stages, some of which are metastable [27]. These transitions are increasingly dramatic for smaller values of v_{red}. Once the target volume is reached, the volume is constrained by the immersed boundary method itself since the enclosed fluid is incompressible. There is another advantage to using the immersed boundary method for this application; the contact problem is avoided because the membrane cannot cross itself during the course of the simulation since it moves within a continuous velocity field. This allows for the simulation of membranes that come close to touching, as happens often when there are multiple cells, or in one cell with smaller than normal v_{red}, without requiring the introduction of nonphysical repulsive forces.

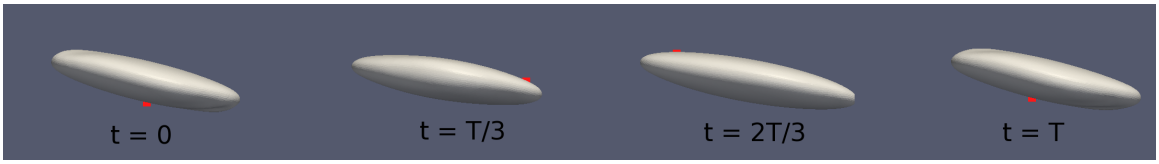
The triangulation of the sphere, which consists of $20,480$ triangles, is constructed by iteratively refining a dodecahedron. A 128^3 fluid grid is used on a $(10\,\mu\text{m})^3$ periodic domain, with $\rho = 1.0\,\text{g/cm}^3$. Because the fluid coefficients affect the dynamics and not the equilibrium shape, we focus attention on the constant viscosity case and use the decreased viscosity $\mu = 0.00012\,\text{P}$, which is about 100 times smaller than the viscosity of blood plasma and allows the membrane to reach equilibrium more quickly, thereby reducing the computational cost of the calculation. The time steps used are $\Delta t = 0.1\Delta x$ for $v_{\text{red}} = 0.59$ and $\Delta t = 0.08\Delta x$ for $v_{\text{red}} = 0.41$.

6.3. Shear Flow. Given their simplicity, it is surprising that red blood cells exhibit such a vast range of behaviors in shear flow. The transition between tank treading and tumbling has been observed both experimentally and in simulations. We first consider a red cell with equal internal and external viscosities, which is called a "ghost cell" and can be produced in the lab by hemolysis in a hypotonic solution [54]. We simulate such ghost cells in shear flow and record the tank treading frequency for several shear rates. We then add variable viscosity and observe the transition to tumbling behavior. Lastly, we account for the difference in internal and external densities of around 8% and examine the resulting changes on the tumbling frequencies. To our knowledge, these are the first simulations of red cell dynamics that account for the different densities of the internal hemoglobin solution and the surrounding plasma.

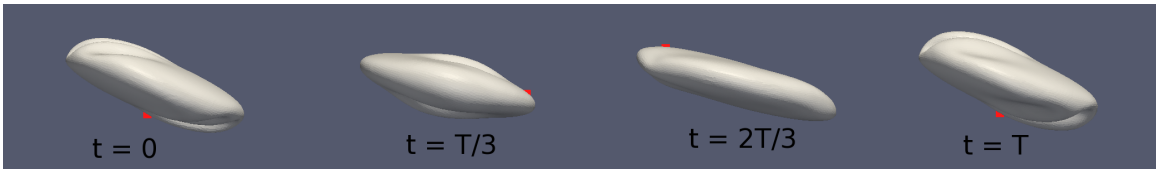
The tank treading frequency depends on the dimensionless capillary number G, defined by $G = \mu\dot{\gamma}a/E$, where μ is the viscosity of the external fluid, $\dot{\gamma}$ is the shear rate, a is the effective cell radius, and E is the shear modulus. The effective cell radius, a, is defined as $a := \left(\frac{3V}{4\pi}\right)^{1/3}$, where $V = 90\,\mu\text{m}^3$ is the cell volume so that $a = 2.78\,\mu\text{m}$. To generate a shear flow, we prescribe equal and opposite external forces on two adjacent planes of the fluid grid normal to the z-axis, as in [21]. More precisely, we specify a body force $\mathbf{f} = (f_1, 0, 0)$ such that

$$f_1(\mathbf{x}, t) = \begin{cases} K, & \text{if } x_3 = \left(-\frac{N_3}{2} + 1\right) h, \\ -K, & \text{if } x_3 = -\frac{N_3}{2} h, \\ 0, & \text{else,} \end{cases} \tag{6.7}$$

for $\mathbf{x} = (x_1, x_2, x_3)$. Because the domain is periodic, this induces a flow $\mathbf{u} = (\dot{\gamma}z, 0, 0)$ in the absence of any red cells. Inside the thin layer of thickness h between the adjacent planes, there is a very strong shear flow, and

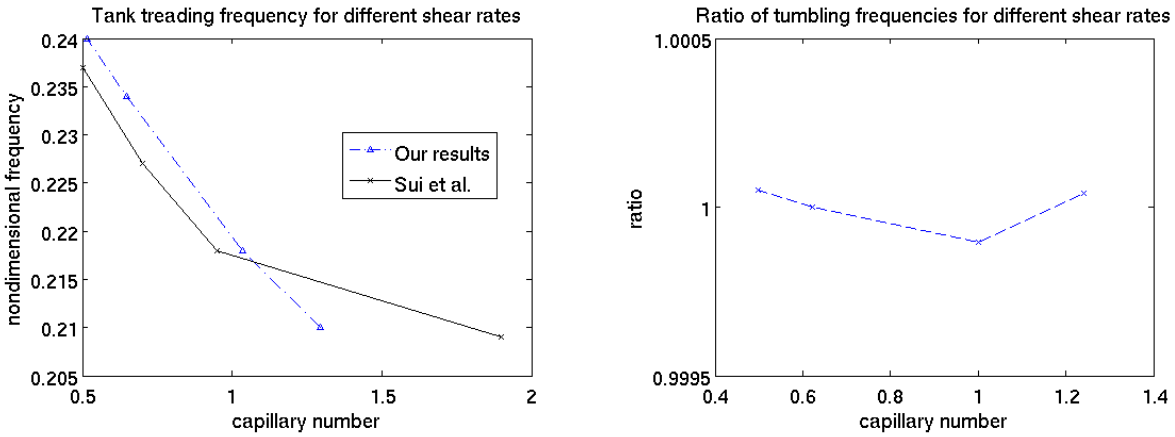


(a) $G = 1.3$



(b) $G = 0.52$

FIG. 6.2. *Snapshots of tank treading as the shear rate is decreased, with a particular material point plotted to illustrate the counterclockwise rotation of the membrane, where T denotes the tank treading period. The format of these images was inspired by [26, 55].*



(a) (b)

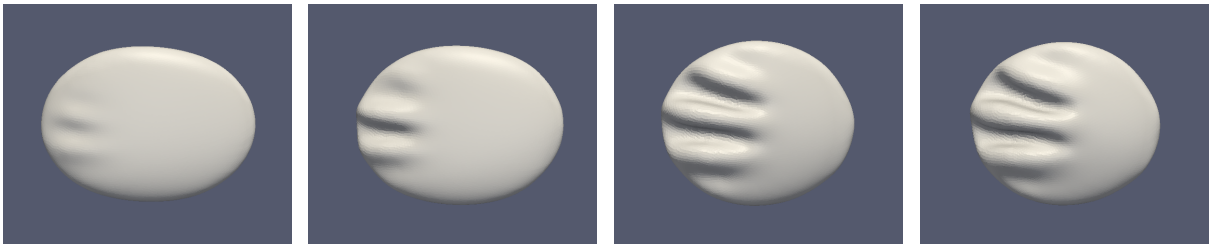
FIG. 6.3. *(a) Nondimensional tank treading frequency $f/\dot{\gamma}$ as a function of shear rate, (b) Ratio of tumbling frequencies between the constant density and variable density cases as a function of shear rate.*

we ensure that no red cells enter this thin layer by using a sufficiently large domain. The constant K required to induce a shear rate $\dot{\gamma}$ is determined empirically. In these simulations, a $128 \times 128 \times 256$ fluid grid is used on a $10\,\mu\text{m} \times 10\,\mu\text{m} \times 20\,\mu\text{m}$ periodic domain, and the time step varies from $\Delta t = 0.015\Delta x$ to $\Delta t = 0.0075\Delta x$, depending on the shear rate.

In response to shear flow, it has been observed that red cells lengthen and align with the flow. At lower shear rates, the membrane begins to exhibit some combination of tumbling and tank treading. Figure 6.2 contains snapshots of the membrane at different times for two different shear rates.

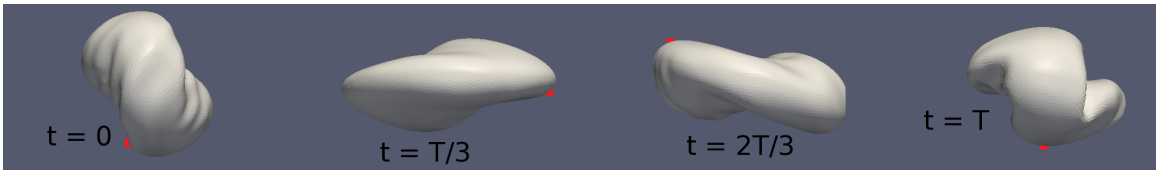
To measure the tank treading frequency, we record the period T between two subsequent crossings of a particular material point through the x-z plane. The frequency f is then given by $f = 2\pi/T$. A comparison to the numerical results of Sui et al. [25] is shown in Figure 6.3(a). We find close agreement, especially at smaller capillary numbers. Note that it would not be reasonable to expect perfect agreement since there are several differences between the two computations, such as the size of the periodic domain and lack of bending rigidity in the simulations of Sui et al.

Viewing the tank treading cells from above in Figure 6.4, we see the onset of large wavelength wrinkling that appears and disappears during the tank treading period and becomes more pronounced as the capillary number is reduced. Such a "quad-concave" wrinkling pattern has been observed in previous numerical simulations [55], but only for a bending rigidity reduced by a factor of 5. Our simulations show that, for the given membrane model, wrinkling also arises at the physiologically realistic bending rigidity of $k = 2.0 \times 10^{-12}$ erg for sufficiently

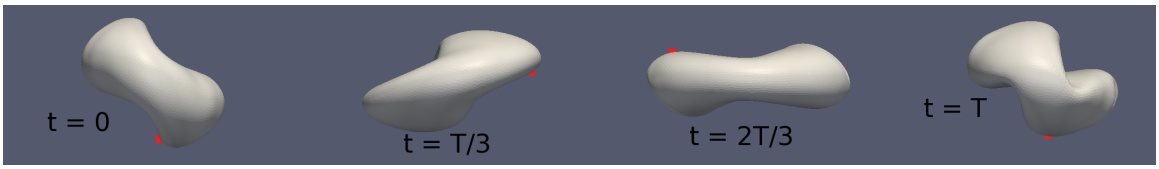


(a) $G = 1.3$ (b) $G = 1.0$ (c) $G = 0.65$ (d) $G = 0.52$

FIG. 6.4. *Appearance of large wavelength wrinkling during tank treading as capillary number is reduced. Each snapshot is taken after the membrane has completed one tank treading oscillation.*



(a) $G = 1.3$



(b) $G = 0.53$

FIG. 6.5. *Snapshots of tumbling as the shear rate is decreased.*

small capillary numbers.

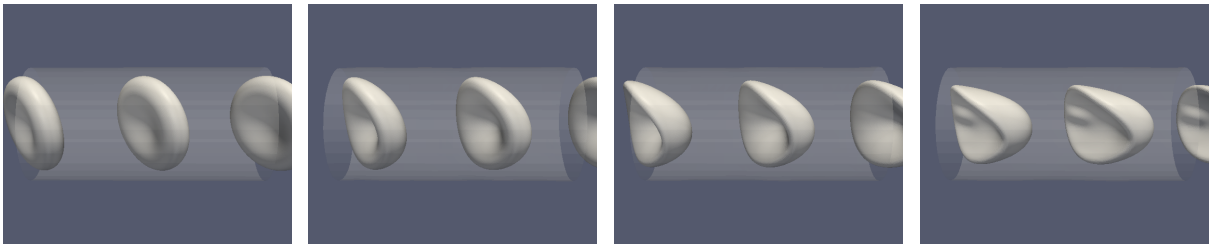
Next, we include the elevated viscosity of the hemoglobin solution inside red cells, which is approximately five times more viscous than the surrounding plasma. As predicted by Keller and Skalak [56], there is a transition from tank treading to tumbling as the internal viscosity is increased. Figure 6.5 contains images of the resulting tumbling motion.

We also account for the different densities on the inside and outside of the red cell. This is expected to be a subtle effect since the internal hemoglobin solution is only about 8% more dense than blood plasma. To investigate the influence of variable density, we compute the ratio of tumbling frequencies between constant density and variable density simulations for several shear rates. We found the difference to be less than 0.1%, as shown in Figure 6.3(b). Although its effect on tumbling frequency appears to be negligible, one clinical application in which variable density *does* play an important role is the calculation of sedimentation rate, a measure of how quickly red cells settle in a test tube that is used to help diagnose certain diseases characterized by a high level of inflammation.

6.4. Capillary Flow. We simulate the motion of two red blood cells traveling through a thin capillary in a periodic box (see Figure 6.6). The red cell membranes are modeled as above, and we account for the different densities and viscosities of the internal and external fluids. The capillary is modeled by a thin elastic membrane tethered in space. A body force of strength $2.0 \times 10^4\,\mathrm{dyn/cm^3}$, calculated through Poiseulle's law, is applied within the vessel to induce a physiologically realistic maximum flow of around $0.075\,\mathrm{cm/sec}$ [57]. The capillary is discretized and tethered in place by springs. Given a node with Lagrangian coordinate $\mathbf{q}$ and position $\mathbf{X}(\mathbf{q})$, we use Hooke's law to calculate the tethering force $\mathbf{F}(\mathbf{q})$ via

$$\mathbf{F}(\mathbf{q}) = -k_{\mathrm{teth}} \left(\mathbf{X}(\mathbf{q}) - \mathbf{X}_{\mathrm{teth}}(\mathbf{q}) \right),$$

in which $\mathbf{X}_{\mathrm{teth}}(\mathbf{q})$ is the target position and the spring constant k_{teth} is $2 \times 10^{-9}\,\mathrm{dyn/cm}$. The cells are initialized in a biconcave shape, i.e. the equilibrium shape in the absence of flow. Here, we use a $128 \times 128 \times 256$ fluid grid on a $10\,\mu\mathrm{m} \times 10\,\mu\mathrm{m} \times 20\,\mu\mathrm{m}$ periodic domain and time step $\Delta t = 0.06\Delta x$. As the cells begin to flow



(a) $t = 0$ (b) $t = 10\text{e}3\Delta t$ (c) $t = 40\text{e}3\Delta t$ (d) $t = 15\text{e}4\Delta t$

FIG. 6.6. *Red blood cells squeezing through a thin capillary.*

through the capillary, they deform into the slipper-like shapes that are observed experimentally. Note also the appearance of large wavelength wrinkles within the concave region in the last frame of Figure 6.6.

7. Implementation. These simulations have been done primarily on an NVIDIA Tesla C2070 Graphics Processing Unit (GPU), which allows for parallel computing with up to 448 cores and has 6 GB memory. Single precision was used to speed up some of the 3D simulations. The programming language used is a combination of Python and OpenCL using the wrapper PyOpenCL. Several of the computational techniques described by McQueen and Peskin [58], originally developed for use on shared-memory Cray supercomputers, were adopted to parallelize the immersed boundary routines. We also used a specialized FFT package, PyFFT. Our GPU implementation of the immersed boundary method has several features in common with those of Patel [59] and Zhang [60]. A typical 3D simulation on a 128^3 grid takes about 48 hours to run to completion. Note that the 3 minutes of real time simulated in a typical computation is a substantial duration on the scale of red cell dynamics, since red cells take only around 20 seconds to complete a cycle through the human body.

8. Conclusions. A variable viscosity and variable density immersed boundary method has been developed. This is, to our knowledge, the first description of an immersed boundary method in which the density and viscosity of the fluid and structure may be defined as general Eulerian or Lagrangian functions. By considering a smooth two-dimensional test problem, we find empirical convergence rates near second-order for constant density and closer to first-order for variable density. We can recover second-order accuracy in the variable density case, but doing so seems to require a computationally expensive iterative scheme. In a companion article [32], we prove the convergence and stability of a closely related timestepping scheme, and this analysis motivates the choices made for the numerical parameters that determine the splitting between explicit and implicit terms in our fluid solver.

We have applied our method to simulate red blood cells. After demonstrating that our membrane model produces realistic equilibrium shapes, we obtain tank treading frequencies for red cell ghosts in shear flow. We simulate tumbling by including the different viscosities of blood plasma and hemoglobin solution as well as their different densities. To our knowledge, these are the first simulations of red blood cells that account for the difference in density between the internal hemoglobin solution and the surrounding plasma. We find that accounting for the 8% difference in densities results in less than a 0.1% change in the tumbling frequency. Lastly, we have performed simulations of red blood cells traveling through a thin elastic capillary that give rise to the slipper-like shapes observed experimentally.

We see several promising avenues of future research. We would like find an *efficient* scheme, second-order accurate in the variable density case, that retains the salient features of this method, e.g. that it minimizes the need to solve large, variable coefficient, linear systems. For our red blood cell simulations, we plan to improve the membrane model by replacing the continuum neo-Hookean shear formulation with a discrete model of the spectrin cytoskeleton [61, 29, 28]. From a computational perspective, we would like to incorporate our method into an adaptive mesh refinement framework [62] to resolve the thin fluid layers between cells and capillaries, and between the cells themselves. Lastly, our GPU implementation needs to be optimized and benchmarked.

9. Acknowledgements. We acknowledge the guidance of Andreas Klockner and use of his software packages PyOpenCl, MeshPy, and PyVTK, which were key components of the GPU implementation of this method. We thank the anonymous reviewers for their valuable comments that led to the improvement of this work. T.G.F. was supported by the DOE Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308. B.E.G. and C.S.P. acknowledge research support from the National Science Foundation (NSF award OCI-1047734). B.E.G. was also supported by the American Heart Association (AHA award

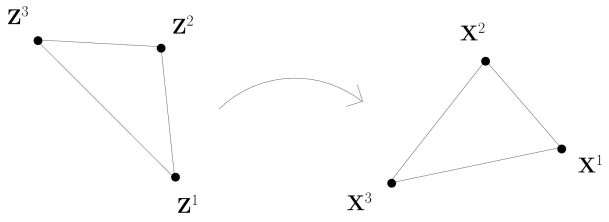


FIG. A.1. *A schematic of the transformation of a triangle from reference coordinates.*

10SDG4320049) and the NSF award DMS-1016554. Y.M. was supported by NSF award DMS-0914963, the Alfred P. Sloan Foundation, and the McKnight Foundation. Computations were performed on GPU's provided by the NYU High Performance Computing Center and the NVIDIA Academic Partnership program.

Appendix A. Discretization of the Red Cell Membrane Energy. Here, we describe the computation of the neo-Hookean energy and the bending energy on the triangulated surface that represents the red blood cell membrane.

A.1. Neo-Hookean Energy. Let us compute the neo-Hookean energy over an individual triangle t with reference triangle s. Denote the position in Eulerian space by $\mathbf{X}$ and the reference position by $\mathbf{Z}$. Let $\mathbf{X}^1$, $\mathbf{X}^2$, and $\mathbf{X}^3$ be the vertices of triangle t in Eulerian coordinates with corresponding vertices $\mathbf{Z}^1$, $\mathbf{Z}^2$, and $\mathbf{Z}^3$ for the reference triangle s. See Figure A.1 for an illustration. We use barycentric coordinates $(\gamma_1, \gamma_2, \gamma_3)$ on the triangle, so that

$$\mathbf{X} = \gamma_1 \mathbf{X}^1 + \gamma_2 \mathbf{X}^2 + \gamma_3 \mathbf{X}^3, \quad \mathbf{Z} = \gamma_1 \mathbf{Z}^1 + \gamma_2 \mathbf{Z}^2 + \gamma_3 \mathbf{Z}^3, \tag{A.1}$$

where $\gamma_1 + \gamma_2 + \gamma_3 = 1$ and $\gamma_1, \gamma_2, \gamma_3 \geq 0$. This ensures that the positions are contained in the triangles. We choose the material coordinates to be (γ_1, γ_2) and use the relation $\gamma_3 = 1 - \gamma_1 - \gamma_2$ to find the third barycentric coordinate. Let G and G_0 be defined as in Section 6.1. Now, since

$$\frac{\partial \mathbf{X}}{\partial \gamma_1} = \mathbf{X}^1 - \mathbf{X}^3, \quad \frac{\partial \mathbf{X}}{\partial \gamma_2} = \mathbf{X}^2 - \mathbf{X}^3, \tag{A.2}$$

the matrix G is constant on the triangle, and similarly for G_0. This allows us to easily calculate the trace and determinant of these matrices, and thus the energy, in terms of the vertex positions. We remark that $\det G$ and $\det G_0$ are related to the areas of the triangles t and s, respectively, via

$$\det G = \| \left(\mathbf{X}^1 - \mathbf{X}^3\right) \times \left(\mathbf{X}^2 - \mathbf{X}^3\right) \|^2 = 4 \operatorname{area}(t)^2, \tag{A.3}$$

$$\det G_0 = \| \left(\mathbf{Z}^1 - \mathbf{Z}^3\right) \times \left(\mathbf{Z}^2 - \mathbf{Z}^3\right) \|^2 = 4 \operatorname{area}(s)^2. \tag{A.4}$$

After using these relations in equation (6.3) and integrating over (γ_1, γ_2), the neo-Hookean energy of triangle t is

$$\begin{aligned} W_{\mathrm{nH}}(t) = {} & \frac{\kappa}{2} \left(\frac{\operatorname{area}(t)}{\operatorname{area}(s)} - 1 \right)^2 \operatorname{area}(s) \\ & + \frac{E}{2} \left(\frac{\|\mathbf{X}^{13}\|^2 \|\mathbf{Z}^{23}\|^2 - 2\left(\mathbf{X}^{13} \cdot \mathbf{X}^{23}\right)\left(\mathbf{Z}^{13} \cdot \mathbf{Z}^{23}\right) + \|\mathbf{Z}^{13}\|^2 \|\mathbf{X}^{23}\|^2}{4 \operatorname{area}(t) \operatorname{area}(s)} - 2 \right) \operatorname{area}(s), \end{aligned} \tag{A.5}$$

where we have used the shorthand

$$\mathbf{X}^{ij} = \mathbf{X}^i - \mathbf{X}^j, \quad \mathbf{Z}^{ij} = \mathbf{Z}^i - \mathbf{Z}^j.$$

The total energy is then given by $W_{\mathrm{nH}}(\dots \mathbf{X} \dots) = \sum_t W_{\mathrm{nH}}(t)$ The force $\mathbf{F}$ is calculated by differentiating the energy analytically. For instance, one ingredient in calculating $\frac{\partial W_{\mathrm{nH}}(t)}{\partial \mathbf{X}^1}$ is to compute $\frac{\partial \operatorname{area}(t)}{\partial \mathbf{X}^1}$. To do so,

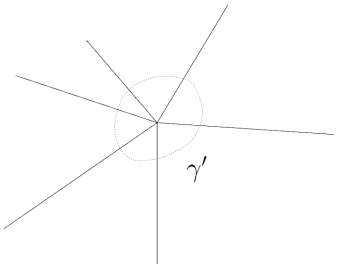


FIG. A.2. *The curve γ' surrounds a vertex in the faceted surface.*

we begin with the formula $4\,\mathrm{area}\,(t)^2 = \left(\mathbf{X}^{21} \times \mathbf{X}^{31}\right) \cdot \left(\mathbf{X}^{21} \times \mathbf{X}^{31}\right)$ and use implicit differentiation to obtain

$$8\,\mathrm{area}\,(t)\,\frac{\partial \mathrm{area}\,(t)}{\partial \mathbf{X}_i^1} = \left(\left(-\mathbf{e}_i \times \mathbf{X}^{31}\right) + \left(\mathbf{X}^{21} \times -\mathbf{e}_i\right)\right) \cdot \left(\mathbf{X}^{21} \times \mathbf{X}^{31}\right) \tag{A.6}$$

$$= \left(\mathbf{e}_i \times \mathbf{X}^{23}\right) \cdot \left(\mathbf{X}^{21} \times \mathbf{X}^{31}\right) = \mathbf{e}_i \cdot \left(\mathbf{X}^{23} \times \left(\mathbf{X}^{21} \times \mathbf{X}^{31}\right)\right), \tag{A.7}$$

for $i = 1, 2, 3$, where $\mathbf{e}_i$ is a unit vector in the i^{th} direction and we have used the invariance of the scalar triple product under cyclic permutations of the arguments. Letting $\mathbf{n}$ be the normal to triangle t defined by $\mathbf{n} := \left(\mathbf{X}^{21} \times \mathbf{X}^{31}\right) / \left(2\,\mathrm{area}\,(t)\right)$, it follows that

$$\frac{\partial \mathrm{area}\,(t)}{\partial \mathbf{X}^1} = \frac{\mathbf{X}^{23} \times \mathbf{n}}{4}. \tag{A.8}$$

The rest of the differentiation is done similarly.

A.2. Curvature. Next, we describe the discretization of the curvature energy. This appendix closely follows the exposition of Sullivan [63], but we include it here for the convenience of the reader. Consider a surface $\mathbf{X}(\alpha, \beta)$ parameterized by α, β. Let γ be a closed curve in the (α, β) plane, and let Γ be the region of the (α, β) plane that it encloses. Let γ' be the image of γ, and let Γ' be the image of Γ under the map $(\alpha, \beta) \mapsto \mathbf{X}(\alpha, \beta)$. Also, let $\mathbf{n}(\alpha, \beta)$ be the normal to the surface, given explicitly by

$$\mathbf{n}(\alpha, \beta) = \frac{\frac{\partial \mathbf{X}}{\partial \alpha}(\alpha, \beta) \times \frac{\partial \mathbf{X}}{\partial \beta}(\alpha, \beta)}{\left|\frac{\partial \mathbf{X}}{\partial \alpha}(\alpha, \beta) \times \frac{\partial \mathbf{X}}{\partial \beta}(\alpha, \beta)\right|}.$$

The total curvature, H, which is the sum of the principal curvatures, satisfies the integral relation

$$\int_{\gamma'} \mathbf{n} \times \mathrm{d}\mathbf{x} = -\iint_{\Gamma'} \mathbf{n} H \mathrm{d}A. \tag{A.9}$$

Now consider a surface made up of planar triangular facets. We can use equation (A.9) to figure out where its total curvature is supported. First, we note that the only possible locations for curvature of any kind are the edges and vertices, since the surface is otherwise flat. If we apply equation (A.9) to a curve γ' surrounding a vertex and then let γ' shrink in toward the vertex, as illustrated in Figure A.2, we see that in the limit considered, $\oint_{\gamma'} \mathbf{n} \times \mathrm{d}\mathbf{x} \to 0$, simply because the arc lengths involved approach zero, whereas $\mathbf{n}$ remains finite. Thus there is no total curvature (i.e. integrated total curvature) associated with a vertex per se. Nonetheless, we shall see later that it is useful to assign to each vertex half the integrated curvature associated with each of its edges.

Next, consider an edge e and a closed curve γ' that is contained within the two triangular faces adjacent to e and surrounds the edge e, passing once through each of its vertices as in Figure A.3(a). Let $\mathbf{Z}(e)$ be a vector aligned with the edge e such that $|\mathbf{Z}(e)| = \mathrm{length}(e)$. The sign of $\mathbf{Z}(e)$ is arbitrary, and will have no effect on the result, as we shall see.

Let $\mathbf{n}_{\mathrm{R}}(e)$ be the unit normal of the triangular facet that lies to the right of the edge e with respect to the orientation of that edge given by $\mathbf{Z}(e)$, and let $\mathbf{n}_{\mathrm{L}}(e)$ be the unit normal of the triangular facet to the left of edge e. If the orientation of $\mathbf{Z}(e)$ is reversed, then $\mathbf{n}_{\mathrm{R}}(e)$ and $\mathbf{n}_{\mathrm{L}}(e)$ are interchanged. Of course, $\mathbf{n}_{\mathrm{R}}(e)$ and

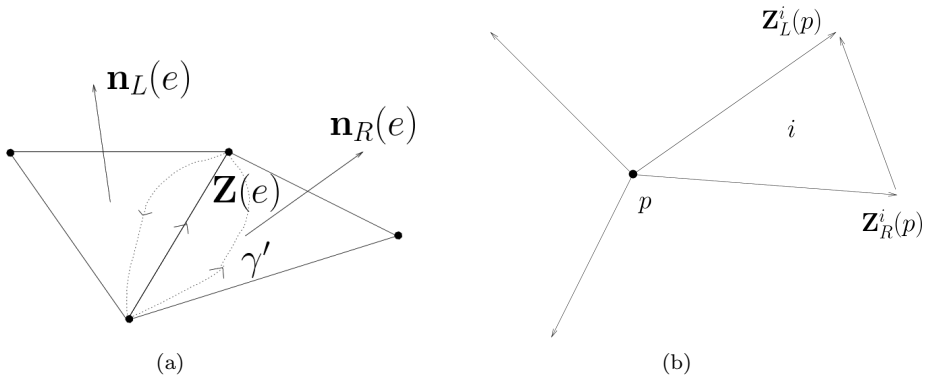


FIG. A.3. *(a) The curve γ' surrounds the edge vector $\mathbf{Z}(e)$. $\mathbf{n}_R(e)$ is the unit normal of the triangle to the right of the edge e with respect to the edge vector $\mathbf{Z}(e)$, and $\mathbf{n}_L(e)$ is the unit normal of the triangle to the left of edge e, (b) A schematic of the triangulation near vertex p.*

$\mathbf{n}_{\mathrm{L}}(e)$ point toward the same side of the surface as the global normal vector field. (The surface is assumed orientable.) Since the triangular facets are flat, we can shrink the curve γ' so that it just barely contains the edge e, and this involves no change in the integrated total curvature. Thus, from equation (A.9), the integrated total curvature vector associated with the edge e is given by

$$\mathcal{H}(e) = -\left(\mathbf{n}_{\mathrm{R}}(e) \times \mathbf{Z}(e) - \mathbf{n}_{\mathrm{L}}(e) \times \mathbf{Z}(e)\right) = \left(\mathbf{n}_{\mathrm{L}}(e) - \mathbf{n}_{\mathrm{R}}(e)\right) \times \mathbf{Z}(e). \tag{A.10}$$

Note that this is independent of the arbitrary sign of $\mathbf{Z}(e)$, since $\mathbf{n}_{\mathrm{R}}(e)$ and $\mathbf{n}_{\mathrm{L}}(e)$ interchange when $\mathbf{Z}(e)$ changes sign. As a matter of bookkeeping, we can, if we like, assign half of the integrated total curvature of each edge to each of the two vertices at the ends of that edge. For a vertex p, this gives the formula

$$\mathcal{H}(p) = \frac{1}{2} \sum_{e \in \mathrm{edges}(p)} \mathcal{H}(e) = \frac{1}{2} \sum_{e \in \mathrm{edges}(p)} \left(\mathbf{n}_{\mathrm{L}}(e) - \mathbf{n}_{\mathrm{R}}(e)\right) \times \mathbf{Z}(e). \tag{A.11}$$

It is convenient to assume that all of the edge vectors in the above sum point radially outward from p, so we make that assumption from now on. (Of course, this is only possible because we are considering only the edges that touch one particular vertex p.) Let $\mathbf{Z}_L^i(p)$ be the edge vector on the left side of triangle i belonging to the point p, and let $\mathbf{Z}_R^i(p)$ be the edge vector on the right side of triangle i belonging to the point p. Then, the edge vector opposite vertex p in triangle i is $(\mathbf{Z}_L^i(p) - \mathbf{Z}_R^i(p))$, as shown in Figure A.3(b). Now the sum in equation (A.11) can be reorganized in a triangle-centric manner, as follows

$$\mathcal{H}(p) = \frac{1}{2} \sum_{i \in \mathrm{triangles}(p)} \mathbf{n}^i \times (\mathbf{Z}_R^i(p) - \mathbf{Z}_L^i(p)) = -\frac{1}{2} \sum_{i \in \mathrm{triangles}(p)} \mathbf{n}^i \times (\mathbf{Z}_L^i(p) - \mathbf{Z}_R^i(p)), \tag{A.12}$$

where $\mathbf{n}^i$ is the normal to triangle i. The transition from equation (A.11) to equation (A.12) is essentially summation by parts. We note another feature of this integrated total curvature vector. It can be shown that it satisfies $\mathcal{H}(p) = -\nabla_p A$, where A is the area of the faceted surface and ∇_p denotes the gradient with respect to the position $\mathbf{X}(p)$. This is a desirable property since total curvature (in the continuous sense) is the first variation of area. That is, if we have a closed surface moving at a velocity given by its outward unit normal, the rate of change of area in a patch is equal to the total curvature integrated over that patch. However, $\mathcal{H}(p)$ is not itself a discretization of total curvature, since it is a vector (it points in the normal direction) and contains a factor of area associated with the vertex. Define the area-weighted normal $\mathbf{A}(p)$ by

$$\mathbf{A}(p) = \frac{1}{3} \sum_{i \in \mathrm{triangles}(p)} \mathbf{n}^i \mathrm{area}(i).$$

The area-weighted normal $\mathbf{A}(p)$ could also be used to define the normal direction, although its direction does not necessarily coincide with that of $\mathcal{H}(p)$. Thus, we have two competing notions of a normal vector to the triangulated surface. We also remark that $\mathbf{A}(p)$ satisfies $\mathbf{A}(p) = \nabla_p V$, where V is the volume enclosed by the faceted surface. We use $\mathcal{H}(p)$ and $\mathbf{A}(p)$ as ingredients for the discretization of total curvature, $h(p)$, which may be defined by $h(p) = |\mathcal{H}(p)|/|\mathbf{A}(p)|$. Then,

$$W_{\text{curv}} = \sum_p h(p)^2 a(p),$$

where $a(p)$ is one-third of the sum of the areas of the triangles attached to vertex p.

The energy is differentiated analytically to find the force $\mathbf{F}$. The most critical part of this computation involves computing the derivatives of the normal vector $\mathbf{n}$ to a triangle with vertices $\mathbf{X}^1$, $\mathbf{X}^2$, and $\mathbf{X}^3$ ordered counterclockwise with respect to the outward normal. This is used to calculate the derivative of $\mathcal{H}(p)$. Since

$$\mathbf{n} = \frac{\left(\mathbf{X}^2 - \mathbf{X}^1\right) \times \left(\mathbf{X}^3 - \mathbf{X}^1\right)}{\| \left(\mathbf{X}^2 - \mathbf{X}^1\right) \times \left(\mathbf{X}^3 - \mathbf{X}^1\right) \|},$$

and

$$\frac{\partial \left(\mathbf{X}^2 - \mathbf{X}^1\right) \times \left(\mathbf{X}^3 - \mathbf{X}^1\right)}{\partial \mathbf{X}^1_j} = \mathbf{e}_j \times \left(\mathbf{X}^2 - \mathbf{X}^3\right),$$

where $\mathbf{e}_j$ is a unit vector in the j direction, it follows from the quotient rule that

$$\frac{\partial \mathbf{n}}{\partial \mathbf{X}^1_j} = \frac{\mathbf{e}_j \times \left(\mathbf{X}^2 - \mathbf{X}^3\right)}{\| \left(\mathbf{X}^2 - \mathbf{X}^1\right) \times \left(\mathbf{X}^3 - \mathbf{X}^1\right) \|} - \frac{\mathbf{n} \left(\left(\mathbf{X}^2 - \mathbf{X}^3\right) \times \mathbf{n}\right)_j}{\| \left(\mathbf{X}^2 - \mathbf{X}^1\right) \times \left(\mathbf{X}^3 - \mathbf{X}^1\right) \|}.$$

We proceed in this manner to calculate analytically the other ingredients of the force computation.

Appendix B. Analysis of a simple ODE. To better understand the order of accuracy of the predictor-corrector scheme described by equations (2.4)-(2.5), we consider a similar scheme applied to a simple ordinary differential equation (ODE). Suppose that $u(t)$ satisfies

$$(1 + a)u' = -\lambda u, \tag{B.1}$$

for positive numbers a, λ with $a < 1$. In analogy with our variable coefficient timestepping scheme, with the number a standing in for the variable density term s, we solve

$$\frac{u^* - u^n}{\Delta t} = -\lambda u^n - a\frac{u^n - u^{n-1}}{\Delta t}, \tag{B.2}$$

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{2}\left(-\lambda u^n - \lambda u^{n+1}\right) - a\frac{u^* - u^n}{\Delta t}. \tag{B.3}$$

The predictor step used alone would yield a solver limited to first-order accuracy, but this is a characteristic of many Runge-Kutta methods, and the time-centered appearance of the corrector step may make it tempting to assume second-order accuracy for the overall scheme. However, we show in the following that this scheme is indeed limited to first-order accuracy.

We can simplify by eliminating the predictor step to get

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{-\lambda}{2}\left(u^n + u^{n+1}\right) - a\left(-\lambda u^n - a\frac{u^n - u^{n-1}}{\Delta t}\right). \tag{B.4}$$

Of course, for this simple ODE one could divide through by the factor $1+a$ and use a traditional time integration scheme such as the explicit trapezoidal rule. However, this would break the analogy with the predictor-corrector scheme whose order of accuracy we would like to illustrate. (Although in the variable density fluid equations one could similarly divide through by the density, this would lead to a variable coefficient Poisson equation for the pressure that cannot easily be solved with fast constant-coefficient linear solvers such as the FFT). We note also that a closer analogy would be to use $-\lambda u^*$ in place of $-\lambda u^n$ in the predictor step, but we use the forward Euler step here for simplicity of analysis. The conclusion is unaffected by this choice.

Now, let $\hat{u}(t)$ denote the exact solution at time t. Using the Taylor expansion, we see that

$$\hat{u}(t+\Delta t) = \hat{u}(t) - \frac{\lambda \Delta t}{1+a}\hat{u}(t) + \frac{(\Delta t)^2}{2}\left(\frac{\lambda}{1+a}\right)^2 \hat{u}(t) + \mathcal{O}(\Delta t)^3, \tag{B.5}$$

$$\hat{u}(t-\Delta t) = \hat{u}(t) + \frac{\lambda \Delta t}{1+a}\hat{u}(t) + \frac{(\Delta t)^2}{2}\left(\frac{\lambda}{1+a}\right)^2 \hat{u}(t) + \mathcal{O}(\Delta t)^3. \tag{B.6}$$

Therefore

$$\frac{\hat{u}(t+\Delta t) - \hat{u}(t)}{\Delta t} = -\frac{\lambda}{1+a}\hat{u}(t) + \frac{\Delta t}{2}\left(\frac{\lambda}{1+a}\right)^2 \hat{u}(t) + \mathcal{O}(\Delta t)^2, \tag{B.7}$$

$$\frac{\hat{u}(t) - \hat{u}(t-\Delta t)}{\Delta t} = -\frac{\lambda}{1+a}\hat{u}(t) - \frac{\Delta t}{2}\left(\frac{\lambda}{1+a}\right)^2 \hat{u}(t) + \mathcal{O}(\Delta t)^2. \tag{B.8}$$

The order of accuracy is calculated by seeing to what order in Δt the exact solution satisfies the above scheme. Using these Taylor expansions for $\hat{u}(t+\Delta t)$ and $\left(\hat{u}(t) - \hat{u}(t-\Delta t)\right)/\Delta t$ and collecting terms, the right hand side of equation (B.4) becomes

$$\left(-1 + a - \frac{a^2}{1+a}\right)\lambda \hat{u}(t) + \left(\frac{1}{2(1+a)} - \frac{a^2}{2(1+a)^2}\right)\lambda^2 \hat{u}(t)\Delta t + \mathcal{O}(\Delta t)^2 \tag{B.9}$$

$$= -\frac{\lambda}{1+a}\hat{u}(t) + (1 + a - a^2)\frac{\Delta t}{2}\left(\frac{\lambda}{1+a}\right)^2 \hat{u}(t) + \mathcal{O}(\Delta t)^2. \tag{B.10}$$

Plugging the exact solution $\hat{u}(t)$ into equation (B.4) and using equations (B.7) and (B.10), we see that

$$\frac{\hat{u}(t+\Delta t) - \hat{u}(t)}{\Delta t} - \left(\frac{-\lambda}{2}\left(\hat{u}(t) + \hat{u}(t+\Delta t)\right) - a\left(-\lambda \hat{u}(t) - a\frac{\hat{u}(t) - \hat{u}(t-\Delta t)}{\Delta t}\right)\right) \tag{B.11}$$

$$= -(a - a^2)\frac{\Delta t}{2}\left(\frac{\lambda}{1+a}\right)^2 \hat{u}(t) + \mathcal{O}(\Delta t)^2 \tag{B.12}$$

$$= \mathcal{O}(\Delta t) \tag{B.13}$$

since $a - a^2 \neq 0$ in general. This implies that the ODE scheme (B.4) is limited to first-order accuracy, which suggests that the analogous predictor-corrector PDE scheme in the variable density case (i.e. $s \neq 0$) is first-order accurate as well. Of course, if one replaces u^* by u^{n+1} in the corrector step (B.3), the scheme becomes second-order accurate, and this fact motivates the iterative scheme that we use to recover second-order accuracy in the variable density case.

REFERENCES

[1] D. C. Bottino and L. J. Fauci. A computational model of ameboid deformation and locomotion. *European Biophysics Journal With Biophysics Letters*, 27:532–539, 1998.

[2] S. Lim, A. Ferent, X. S. Wang, and Charles S. Peskin. Dynamics of a closed rod with twist and bend in fluid. *Siam Journal on Scientific Computing*, 31:273–302, 2008.

[3] A. L. Fogelson. A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting. *Journal of Computational Physics*, 56:111–134, 1984.

[4] B. E. Griffith, X. Luo, D. M. McQueen, and C. S. Peskin. Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method. *International Journal of Applied Mechanics*, 01, 2009.

[5] B. E. Griffith. Immersed boundary model of aortic heart valve dynamics with physiological driving and loading conditions. *Int J Numer Meth Biomed Eng*, 28:317–345, 2012.

[6] X. Y. Luo, B. E. Griffith, X. S. Ma, M. Yin, T. J. Wang, C. L. Liang, P. N. Watton, and G. M. Bernacca. Effect of bending rigidity in a dynamic model of a polyurethane prosthetic mitral valve. *Biomechan Model Mechanobiol*, 11:815–827, 2012.

[7] X. S. Ma, H. Gao, B. E. Griffith, C. Berry, and X. Y. Luo. Image-based fluid-structure interaction model of the human mitral valve. *Comput Fluid*. To appear.

[8] R. H. Dillon and L. J. Fauci. A microscale model of bacterial and biofilm dynamics in porous media. *Biotechnology and Bioengineering*, 68(5):536–547, 2000.

[9] K. A. Rejniak. An immersed boundary framework for modelling the growth of individual cells: An application to the early tumour development. *Journal of Theoretical Biology*, 247(1):186 – 204, 2007.

[10] K. A. Rejniak and R. H. Dillon. A single cell-based model of the ductal tumour microarchitecture. *Computational and Mathematical Methods in Medicine*, 8(1):51–69, 2007.

[11] Y. Li, A. Yun, and J. Kim. An immersed boundary method for simulating a single axisymmetric cell growth and division. *Journal of Mathematical Biology*, 65(4):653–675, 2012.

[12] W. Strychalski and R. D. Guy. A computational model of bleb formation. *Mathematical Medicine and Biology*, 2012.

[13] L. M. Crowl and A. L. Fogelson. Computational model of whole blood exhibiting lateral platelet motion induced by red blood cells. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3/4):471 – 487, 2010.

[14] L. Zhu and C. S. Peskin. Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. *Journal of Computational Physics*, 179:452–468, 2002.

[15] Y. Kim and C. S. Peskin. Numerical study of incompressible fluid dynamics with nonuniform density by the immersed boundary method. *Physics of Fluids*, 20(6), June 2008.

[16] Y. Mori and C. S. Peskin. Implicit second-order immersed boundary methods with boundary mass. *Computer Methods in Applied Mechanics and Engineering*, 197:2049–2067, 2008.

[17] S. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.

[18] M. T. Zuber and E. M. Parmentier. Finite amplitude folding of a continuously viscosity-stratified lithosphere. *Journal of Geophysical Research*, 101:5489–5498, 1996.

[19] D. D. Joseph. Variable viscosity effects on the flow and stability of flow in channels and pipes. *Physics of Fluids*, 7(11):1761–1771, 1964.

[20] R. Skalak and P. I. Branemark. Deformation of red blood cells in capillaries. *Science*, 164:717–719, 1969.

[21] C. D. Eggleton and A. S. Popel. Large deformation of red blood cell ghosts in a simple shear flow. *PHYSICS OF FLUIDS*, 10(8):1834–1845, AUG 1998.

[22] G. Lim, M. Wortis, and R. Mukhopadhyay. Stomatocyte-discocyte-echinocyte sequence of the human red blood cell: Evidence for the bilayer- couple hypothesis from membrane mechanics. *Proceedings of The National Academy of Sciences*, 99:16766–16769, 2002.

[23] P. Bagchi. Mesoscale simulation of blood flow in small vessels. *Biophysical Journal*, 92:1858–1877, 2007.

[24] S. Doddi and P. Bagchi. Three-dimensional computational modeling of multiple deformable cells flowing in microvessels. *Physical Review E*, 79, 2009.

[25] Y. Sui, Y. T. Chew, P. Roy, Y. P. Cheng, and H. T. Low. Dynamic motion of red blood cells in simple shear flow. *Physics of Fluids*, 20(11), NOV 2008.

[26] A. Z. K. Yazdani, R. M. Kalluri, and P. Bagchi. Tank-treading and tumbling frequencies of capsules and red blood cells. *Physical Review E*, 83:046305+, April 2011.

[27] H. Noguchi and G. Gompper. Shape transitions of fluid vesicles and red blood cells in capillary flows. *Proceedings of The National Academy of Sciences*, 102:14159–14164, 2005.

[28] J. Li, M. Dao, C.T. Lim, and S. Suresh. Spectrin-level modeling of the cytoskeleton and optical tweezers stretching of the erythrocyte. *Biophysical Journal*, 88(5):3707 – 3719, 2005.

[29] G. Marcelli, K. H. Parker, and C. P. Winlove. Thermal Fluctuations of Red Blood Cell Membrane via a Constant-Area Particle-Dynamics Model. *Biophysical Journal*, 89:2473–2480, 2005.

[30] M. Nakamura and S. Wada. Mesoscopic Blood Flow Simulation Considering Hematocrit-Dependent Viscosity. *Journal of Biomechanical Science and Engineering*, 5:578–590, 2010.

[31] A. J. Chorin. Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation*, 22:745–745, 1968.

[32] T. G. Fai, B. E. Griffith, Y. Mori, and C. S. Peskin. Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers II: Theory. Submitted.

[33] S. Dong and J. Shen. A time-stepping scheme involving constant coefficient matrices for phase-field simulations of two-phase incompressible flows with large density ratios. *Journal of Computational Physics*, 231(17):5788 – 5804, 2012.

[34] D. C. Leslie and S. Gao. The stability of spectral schemes for the large eddy simulation of channel flows. *International Journal for Numerical Methods in Fluids*, 8(9):1107–1116, 1988.

[35] D. Wilhelm and E. Meiburg. Three-dimensional spectral element simulations of variable density and viscosity, miscible displacements in a capillary tube. *Computers & Fluids*, 33(3):485 – 508, 2004.

[36] G.-S. Karamanos and S.J. Sherwin. A high order splitting scheme for the navierstokes equations with variable viscosity. *Applied Numerical Mathematics*, 33(14):455 – 462, 2000.

[37] Homepage of T. G. Fai. `http://www.cims.nyu.edu/~tfai`. 2012.

[38] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flow. *Journal of Computational Physics*, 143:90–124, 1998.

[39] D. Devendran and C. S. Peskin. An immersed boundary energy-based method for incompressible viscoelasticity. *Journal of Computational Physics*, 231(14):4613 – 4642, 2012.

[40] B. E. Griffith. An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as a preconditioner. *J Comput Phys*, 228:7565–7595, 2009.

[41] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf. Computational geometry: algorithms and applications. 1997.

[42] J. R. Shewchuk. *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. 1996.

[43] H. Si. A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. *Web Intelligence and Agent Systems: An International Journal*.

[44] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11, 2002.

[45] Y. Liu and Y. Mori. Properties of discrete delta functions and local convergence of the immersed boundary method. *SIAM Journal of Numerical Analysis*. To appear.

[46] B. E. Griffith and C. S. Peskin. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. *Journal of Computational Physics*, 208:75–105, 2005.

[47] T. Kenner. The measurement of blood density and its meaning. *Basic Research in Cardiology*, 84:111–124, 1989.

[48] E. A. Evans and R. Skalak. *Mechanics and Thermodynamics of Biomembranes*. CRC Press, Boca Raton, FL, 1980.

[49] P. Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *Journal of Theoretical Biology*, 26:61–81, 1970.

[50] W. Helfrich. Elastic Properties of Lipid Bi-layers: Theory and Possible Experiments. 1973.
[51] N. Mohandas and E. Evans. Mechanical Properties of the Red Cell Membrane in Relation to Molecular Structure and Genetic Defects. *Annual Review of Biophysics and Biomolecular Structure*, 23:787–818, 1994.
[52] E. Evans, R. Waugh, and L. Melnik. Elastic area compressibility modulus of red cell membrane. *Biophysical Journal*, 16:585–595, 1976.
[53] E. Evans and Y.-C. Fung. Improved measurements of the erythrocyte geometry. *Microvascular Research*, 4(4):335 – 347, 1972.
[54] J. Dodge, C. Mitchell, and D. Hanahan. The preparation and chemical characteristics of hemoglobin-free ghosts of human erythrocytes. *Archives of Biochemistry and Biophysics*, 100:119–130, 1963.
[55] A. Z. K. Yazdani and P. Bagchi. Phase diagram and breathing dynamics of a single red blood cell and a biconcave capsule in dilute shear flow. *Phys. Rev. E*, 84:026314, Aug 2011.
[56] S. R. Keller and R. Skalak. Motion of a tank-treading ellipsoidal particle in a shear flow. *Journal of Fluid Mechanics*, 120, 1982.
[57] M. K. Kalinina, Yu. I. Levkovich, K. P. Ivanov, and V. K. Trusova. Blood flow velocity in cerebral cortex capillaries (microcinephotographic study). *Neuroscience and Behavioral Physiology*, 9:185–188, 1978. 10.1007/BF01182612.
[58] D. M. McQueen and C. S. Peskin. Shared-Memory Parallel Vector Implementation of the Immersed Boundary Method for the Computation of Blood Flow in the Beating Mammalian Heart. *The Journal of Supercomputing*, 11:213–236, 1997.
[59] S. Patel. aerocuda: The gpu-optimized immersed solid code. Undergraduate Thesis, June 23, 2012.
[60] X. Zhang. An efficient gpu implementation of immersed boundary method. May 15, 2011.
[61] A. Nans, N. Mohandas, and D. L. Stokes. Native Ultrastructure of the Red Cell Cytoskeleton by Cryo-Electron Tomography. *Biophysical Journal*, 101:2341–2350, 2011.
[62] IBAMR: An adaptive and distributed-memory parallel implementation of the immersed boundary method. `http://ibamr.googlecode.com`.
[63] J. M. Sullivan. Curvatures of smooth and discrete surfaces. In Alexander I. Bobenko, John M. Sullivan, Peter Schröder, and Günter M. Ziegler, editors, *Discrete Differential Geometry*, volume 38 of *Oberwolfach Seminars*, pages 175–188. Birkhäuser Basel, 2008.