A spectral boundary integral method for flowing blood cells

Hong Zhao [a], Amir H.G. Isfahani [a], Luke N. Olson [c], Jonathan B. Freund [a,b,*]

[a] *University of Illinois at Urbana-Champaign, Department of Mechanical Science and Engineering, Urbana, IL 61801, United States*
[b] *University of Illinois at Urbana-Champaign, Department of Aerospace Engineering, Urbana, IL 61801, United States*
[c] *University of Illinois at Urbana-Champaign, Department of Computer Science, Urbana, IL 61801, United States*

ABSTRACT

A spectral boundary integral method for simulating large numbers of blood cells flowing in complex geometries is developed and demonstrated. The blood cells are modeled as finite-deformation elastic membranes containing a higher viscosity fluid than the surrounding plasma, but the solver itself is independent of the particular constitutive model employed for the cell membranes. The surface integrals developed for solving the viscous flow, and thereby the motion of the massless membrane, are evaluated using an $O(N \log N)$ particle-mesh Ewald (PME) approach. The cell shapes, which can become highly distorted under physiologic conditions, are discretized with spherical harmonics. The resolution of these global basis functions is, of course, excellent, but more importantly they facilitate an approximate de-aliasing procedure that stabilizes the simulations without adding any numerical dissipation or further restricting the permissible numerical time step. Complex geometry no-slip boundaries are included using a constraint method that is coupled into an implicit system that is solved as part of the time advancement routine. The implementation is verified against solutions for axisymmetric flows reported in the literature, and its accuracy is demonstrated by comparison against exact solutions for relaxing surface deformations. It is also used to simulate flow of blood cells at 30% volume fraction in tubes between 4.9 and 16.9 μm in diameter. For these, it is shown to reproduce the well-known non-monotonic dependence of the effective viscosity on the tube diameter.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

We develop an algorithm to efficiently simulate the detailed interactions of large numbers of blood cells flowing in complex geometry blood vessels. The nature of blood makes this inherently challenging. The red blood cells (erythrocytes) are densely packed, making up around 30% of the overall volume of the blood in the smallest blood vessels, so the cells interact continuously and strongly. Moreover, the cells are highly deformable under physiologic conditions. Unstressed red cells are well known to have a distinctive biconcave shape of around 8 μm in diameter and 2 μm in thickness, but deform easily from this minimum-energy shape. Fortunately for developing simulation models, the structure of each cell is fundamentally simple: a flexible membrane encloses a hemoglobin solution. The membrane itself has finite elastic shear modulus and bending moment but strongly resists changes in surface area. The hemoglobin within and the plasma surrounding the cells are both Newtonian, but the overall behavior of blood in small vessels is non-Newtonian due to its particulate character. Cell–cell and the cell–wall interactions lead to phenomena such as the non-monotonic change of the apparent blood viscosity with blood

* Corresponding author. Address: University of Illinois at Urbana-Champaign, Department of Mechanical Science and Engineering, Urbana, IL 61801, United States. Tel.: +1 (217) 244 7729; fax: +1 (217) 244 5705.
E-mail addresses: hongzhao@stanford.edu (H. Zhao), ghalayan@illinois.edu (A.H.G. Isfahani), lukeo@illinos.edu (L.N. Olson), jbfreund@illinois.edu (J.B. Freund).

vessel diameters [12] and the separation of red cells at vessel bifurcations [35]. The particulate character is also believed to be an important mechanical mechanism for the migration of leukocytes (white cells) towards the walls of small blood vessels [1,13].

At the low Reynolds number of flows in the microcirculation, the flow of plasma outside the cell membranes and the hemoglobin solution inside are both Stokesian. Although Stokes flows have relatively simple linear governing equations, direct numerical simulation of the cellular flow system is nevertheless challenging because of its continuously changing geometry, the deformations of the cells, and their close-range interactions. For simulation methods requiring boundary-fitted meshes, such as arbitrary Lagrangian–Eulerian methods, the mesh movement and regeneration for the complicated flow geometry is challenging and computationally expensive. Immersed boundary or immersed interface methods can avoid this by representing the changing geometry on a fixed mesh. These methods account for the cell–fluid interactions by diffusing the membrane force to the nearby fluid domain mesh points [28] or by modifying the local discrete differential operators when their stencils cross the cell membrane [21]. Both cell–cell interactions and cellular flow in small blood vessels have been simulated by the immersed boundary method in two dimensions [4,3]. The immersed finite-element method [23] and the lattice-Boltzmann method [24] similarly model the cell membranes as added body forces and have been used for studying cell aggregation and blood rheology. These fixed-mesh methods avoid re-meshing, but still require either a fine volumetric mesh or adaptive meshing to resolve close interactions between cells [39].

The mesh-based Lagrangian–Eulerian and immersed boundary or interface methods start with discretizing the differential form of the flow equations. Another approach is to use the linearity of Stokes flow to express solutions as surface integrals over interfaces and boundaries [18,32]. Such a boundary integral approach requires only discretization of the cell and vessel-wall surfaces, and the difficulties in volumetric meshing of the complex-shaped flow domain are avoided. Such an approach can not be extended to include any inertia effects as immersed boundary [3] or lattice-Boltzmann [24] schemes, but this is not restrictive since microcirculatory flows are indeed very low Reynolds number. Boundary integral formulations have been used in simulating the motion of a single unconfined red cell in shear flow [33], axisymmetric cell motion in cylindrical tubes [34], and multiple cells in two dimensions [14], but considerable development has been required to do this efficiently for large numbers of cells in general three-dimensional configurations. Our efforts to do this are the subject of this paper.

A particular challenge in simulating red cells arises from the nonlinear dependence of the membrane residual force on its deformation. As discussed in detail in Section 10, the nonlinearities stem from the membrane stress–strain relation as well as the intrinsic geometrical nonlinearities due to the finite membrane deformation. In numerical simulations, these nonlinearities can cause unphysical growth of modes of high wavenumber by an aliasing mechanism. As in any physical system, nonlinearity can move energy to smaller scales. For finite resolution, most numerical schemes will alias the unresolvable portion of this energy into the resolved solution [6]. In red-cell simulations, the aliasing errors can lead to the slow growth of surface features such as sharp corners and cusps. When the system does not have strong enough physical or numerical dissipation, the growth of these high wavenumber modes results in numerical instability of the kind that has been reported [33]. Avoiding this through some manner of numerical damping reduces the effective resolution of the scheme.

In a two-dimensional simulations of leukocyte transport, Freund [14] represented the shape of each cell with typically 16 Fourier modes, but to avoid significant aliasing errors all nonlinear operations were done with 64 collocation points interpolated by these 16 modes. The algorithm was straightforward and stable without any artificial dissipation and the time step was not limited beyond the typical restriction imposed by the 16 resolved modes. Here, we generalize this approach to develop and demonstrate the correspondingly more intricate algorithm for solving general three-dimensional cellular flows in complex geometries. Each cell surface is represented by spherical harmonics, the natural counterpart of the Fourier series for the two-dimensional cells. This spectral surface representation, in addition to its high accuracy, similarly enables de-aliasing.

We briefly review the system considered in Section 2 and discuss the various aspects of the simulation algorithm in detail in Sections 3–7. Special attention is given to the methods used for handling singular and nearly-singular integrands and for our formulation of a constraint method that provides complex geometry no-slip boundaries using formulations based upon the standard periodic-domain Green's functions of the Stokes operator. In Section 8, the expected convergence with increasing resolution is confirmed and means of controlling the accuracy with the numerical parameters are discussed. Numerical solutions are compared with exact solutions for the relaxation of small surface deformations to in Section 9 to confirm the time accuracy of the algorithm and the excellent spatial resolution of the spherical harmonics. The de-aliasing procedure is discussed and demonstrated in Section 10. Several simulations of blood cells flowing in tubes are included in Section 11. The first of these is used to verify our implementation by direct comparison of our three-dimensional algorithm with a reported axisymmetric result calculated using the same physical model. This is followed by demonstration simulations of three-dimensional flows in tubes with various diameters. These are shown to reproduce the experimentally measured non-monotonic dependence of effective blood viscosity on vessel diameter.

2. Flow solver formulation

A generic three-dimensional cellular flow domain is depicted schematically in Fig. 1. The cells D_i and vessel walls W are embedded in a rectangular domain $\Omega = [0, L_1] \times [0, L_2] \times [0, L_3]$. The Newtonian viscosity μ is that of blood plasma in Ω_0, everywhere outside of the cells. The region outside of the vessel is included in Ω_0 since it facilitates the use of periodic
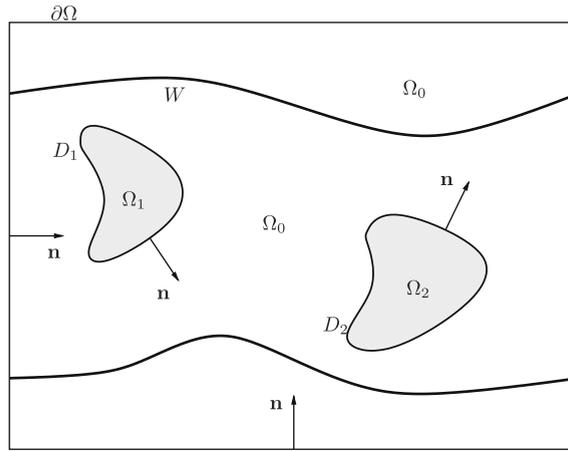
Fig. 1. A model microcirculatory vessel. The shaded area ($\Omega_{1,2,...}$) is occupied by the cells, and the rest (Ω_0) by plasma.

Green's function kernels, but the no-slip boundary condition on the vessel walls prevents this exterior flow from influencing the flow in the vessel. The hemoglobin solution within the cells has Newtonian viscosity $\lambda\mu$. Indirect experimental measurements suggest that this interior viscosity is higher than the plasma viscosity but of the same order of magnitude [10]. A viscosity ratio that has been used in past simulations of red cells is $\lambda = 5$ [33], which we also use. The flow is driven by a pressure gradient, so despite the periodic boundary conditions on the velocity, pressure has both a periodic and a linearly varying component. The periodic part can be expressed as $\mathbf{x} \cdot \langle\nabla p\rangle$, where $\langle\nabla p\rangle$ is the mean pressure gradient, which necessarily balances the net effect of any wall friction.

The theory of the boundary integral formulation for Stokes flow is well established [18,31,32,37]. We briefly outline the general formulation for completeness and concentrate on the details central to the cellular flow system and our algorithm. The formulation is built upon the solutions of the Stokes equation with a point force inhomogeneity,

$$0 = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{g}\delta(\mathbf{x} - \mathbf{x}_0), \quad 0 = \nabla \cdot \mathbf{u}, \tag{1}$$

where p is the pressure and $\mathbf{u}$ is the velocity. The solutions for the fluid velocity and stress are

$$u_i(\mathbf{x}) = \frac{1}{8\pi\mu}G_{ij}(\mathbf{x}, \mathbf{x}_0)g_j \quad \text{and} \quad \sigma_{ik}(\mathbf{x}) = \frac{1}{8\pi}T_{ijk}(\mathbf{x}, \mathbf{x}_0)g_j. \tag{2}$$

Here, the so-called *Stokeslet* $\mathbf{G}(\mathbf{x}, \mathbf{x}_0)$ is periodic in Ω on both $\mathbf{x}$ and $\mathbf{x}_0$ and satisfies the zero mean flow condition $\int_\Omega \mathbf{G}(\mathbf{x}, \mathbf{x}_0)\,d\mathbf{x} = 0$. The so-called *Stresslet* $\mathbf{T}(\mathbf{x}, \mathbf{x}_0)$ can be spatially decomposed into linear and periodic parts,

$$T_{ijk}(\mathbf{x}, \mathbf{x}_0) = -\frac{8\pi}{V}x_j\delta_{ik} + \breve{T}_{ijk}(\mathbf{x}, \mathbf{x}_0), \tag{3}$$

where $V = L_1L_2L_3$ is the volume of the domain, and the periodic part $\breve{T}_{ijk}(\mathbf{x}, \mathbf{x}_0)$ satisfies the constraint $\int_\Omega \breve{T}(\mathbf{x}, \mathbf{x}_0)\,d\mathbf{x} = 0$. The single-layer potential operator $\mathbf{N}$ and the double-layer potential operator $\mathbf{K}$ are integral operators on a surface D that involve these Green's functions as kernels. They map any surface vector distribution $\boldsymbol{\psi}$ into

$$(\mathbf{N}_D\boldsymbol{\psi})_j(\mathbf{x}_0) = \int_D \psi_i(\mathbf{x})G_{ij}(\mathbf{x}, \mathbf{x}_0)\,dS(\mathbf{x}), \tag{4}$$

and

$$(\mathbf{K}_D\boldsymbol{\psi})_j(\mathbf{x}_0) = \int_D \psi_i(\mathbf{x})T_{ijk}(\mathbf{x}, \mathbf{x}_0)n_k(\mathbf{x})\,dS(\mathbf{x}). \tag{5}$$

If the surface D is Lyapunov smooth, then both operators are weakly-singular and hence compact [20,30].

Taking $D = \cup_i D_i$ to be the set of cell membranes, the surface velocity at any point $\mathbf{x}_0 \in D$ solves

$$u_j(\mathbf{x}_0) = -\frac{1}{4\pi\mu(1 + \lambda)}(\mathbf{N}_{D\cup W}\mathbf{f})_j(\mathbf{x}_0) + \frac{1 - \lambda}{4\pi(1 + \lambda)}(\mathbf{K}_D\mathbf{u})_j(\mathbf{x}_0) + \frac{2}{1 + \lambda}\langle u_j\rangle, \tag{6}$$

where the surface normal $\mathbf{n}$ in $\mathbf{K}_D$ points into the plasma region Ω_0 (see Fig. 1), the force density $\mathbf{f}$ is the sum of hydrodynamic forces acting on D and W, and $\langle\mathbf{u}\rangle$ is the volume average of the velocity in Ω. Since it appears explicitly as a parameter, $\langle\mathbf{u}\rangle$ is convenient for setting the strength of the flow, but this average includes flow outside the vessel. The no-slip condition on the vessel wall W is

$$0 = -\frac{1}{8\pi\mu}(\mathbf{N}_{D\cup W}\mathbf{f})_j(\mathbf{x}_0) + \frac{1-\lambda}{8\pi}(\mathbf{K}_D\mathbf{u})_j(\mathbf{x}_0) + \langle u_j \rangle, \tag{7}$$

which applies for any $\mathbf{x}_0 \in W$ and prevents the exterior flow from influencing the flow in the interior of the vessel.

Since the inertia of cell membranes is negligible, the hydrodynamic traction $\mathbf{f}$ in (6) acting on the membrane balances the residual force from the membrane's internal stress. Hence $\mathbf{f}|_D$ is explicitly calculated from the cell membrane deformation and the stress–strain relation. As a result, $\mathbf{u}|_D$ and $\mathbf{f}|_W$ are solved by inverting (6) and (7) for any given $\langle\mathbf{u}\rangle$.

The volume average velocity $\langle\mathbf{u}\rangle$ is related to the mean pressure gradient $\langle\nabla p\rangle$, which is also an important parameter for studying microcirculatory flows. The average velocity implicitly determines the net wall friction force and thus the mean pressure gradient,

$$\langle\nabla p\rangle = -\frac{1}{V}\int_W \mathbf{f}(\mathbf{x},t)\,\mathrm{d}S(\mathbf{x}), \tag{8}$$

through the no-slip condition (7). The functional relation between $\langle\mathbf{u}\rangle$ and $\langle\nabla p\rangle$ thus depends on the size of the domain Ω and is therefore not generic. The explicit functional form of the dependence is not easily realized even for relatively simple geometries. Thus, unlike mesh-based methods, it is not convenient *a priori* to set either the pressure gradient or mean velocity in the vessel. However, this is not a significant limitation since a specific $\langle\mathbf{u}\rangle$ corresponds to a specific time average of $\langle\nabla p\rangle$ and mean flow in the vessel. It is thus straightforward to set either of these iteratively or more simply study a range of flow conditions by varying $\langle\mathbf{u}\rangle$ as we did previously in two dimensions [14]. Furthermore, for the kind of flow shown in Fig. 1, the time variation of domain average pressure $\langle\nabla p\rangle$ is typically small (less than 2% for the cylindrical tubes in Section 11.2) with fixed $\langle\mathbf{u}\rangle$. If a specific mean pressure gradient is indeed required, then unknowns $\mathbf{u}|_D$, $\mathbf{f}|_W$ and $\langle\nabla p\rangle$ can be solved from (6)–(8), although this increases the number of unknowns by three.

Through collocation, (6) and (7) are discretized into the linear system

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}\begin{pmatrix} \mathbf{U} \\ \mathbf{F} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}, \tag{9}$$

where $\mathbf{U}$ is all the velocity vectors for all discrete mesh points on cells and $\mathbf{F}$ is all the friction forces on the discrete mesh points on the walls. The sub-matrices in (9) clearly correspond to cell–cell, cell–wall and wall–wall interactions.

A staggered time integrator updates the cell deformations from each time step t^n to the next $t^{n+1} = t^n + \Delta t$ with the following steps:

(i) Solve the surface velocity $\mathbf{U}^n$ from

$$\mathbf{A}_{11}^n\mathbf{U}^n = \mathbf{b}_1^n - \mathbf{A}_{12}^n\mathbf{F}^n. \tag{10}$$

(ii) Advance the cell surface coordinates by the forward Euler scheme:

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t\mathbf{U}^n. \tag{11}$$

(iii) Update the wall drag force by solving

$$\mathbf{A}_{22}^{n+1}\mathbf{F}^{n+1} = \mathbf{b}_2^{n+1} - \mathbf{A}_{21}^{n+1}\mathbf{U}^n. \tag{12}$$

Euler forward time stepping is shown here for simplicity and used in our example simulations; multi-step time integration schemes such as Adams–Bashforth can be used without difficulty. Runge–Kutta schemes are also straightforward in some cases, but become more difficult to apply when a cell–cell collision removal procedure is added between sub-time-steps as seems necessary in cases (see Section 11.2). The two linear systems for $\mathbf{U}$ and $\mathbf{F}$ are solved with GMRES [40]. The submatrix $\mathbf{A}_{11}$ represents the operator $\mathbf{I} - (1-\lambda)/[4\pi(1+\lambda)]\mathbf{K}_D$, and its condition number is bounded by the viscosity ratio λ. For matched viscosity $\lambda = 1$, $\mathbf{A}_{11} = \mathbf{I}$, and the equation for $\mathbf{U}$ is explicit; when $\lambda = 5$, as for most of the demonstration simulations, the system is well conditioned and no more than 10 iterations are needed to reduce the L_2-norm of the relative residual error to about 10^{-6}. The determination of $\mathbf{F}$, on the other hand, requires solving a single-layer potential system whose condition number grows with wall mesh refinement. For the cylindrical tubes of Section 11.1, the maximum wall residual velocity is reduced to less than $O(10^{-4})$ of $\|\langle\mathbf{u}\rangle\|$ by 20 GMRES iterations. For a particular flow or geometry, it is likely that the matrix condition can be improved through effective preconditioning [45,47] or by applying the no-slip condition through an indirect formulation [16,42,48]. While it is in principle possible to solve (9) directly, the scheme we use is simpler, has been consistently stable, and is demonstrated to be accurate in the example simulations we discuss.

3. Cell surface representation and residual force calculation

The surface of each cell D_i is mapped from a unit sphere $\mathbb{S}^2$ that is parameterized by the colatitude angle $\theta \in [0,\pi]$ and the longitude angle $\phi \in [0,2\pi)$. Any function f on $\mathbb{S}^2$ is represented by a truncated series of spherical harmonic functions,

$$f(\theta, \phi) = \sum_{n=0}^{N-1} \sum_{m=0}^{n} \overline{P}_n^m(\cos\theta)(a_{nm}\cos m\phi + b_{nm}\sin m\phi),$$ (13)

yielding N^2 total number of spherical harmonic modes. The normalized associated Legendre polynomials,

$$\overline{P}_n^m(x) = \frac{1}{2^n n!}\sqrt{\frac{(2n+1)(n-m)!}{2(n+m)!}}(1-x^2)^{\frac{m}{2}}\frac{d^{n+m}}{dx^{n+m}}(x^2-1)^n,$$ (14)

satisfy

$$\int_{-1}^{1} \overline{P}_n^m(x)\overline{P}_{n'}^m(x)\,dx = \delta_{nn'}.$$ (15)

The mesh coordinates are uniform in ϕ and the roots of $P_N(\cos\theta)$ in θ. Both forward and backward transforms are computed using the SPHEREPACK library [2,44].

Besides spectral accuracy, there are two main advantages to the spherical harmonic representation. First, the differentiation of each modal function on the right-hand side of (13) is well defined, without the ambiguous behavior or singularities at the poles ($\theta = 0, \pi$) that are encountered when using, say, standard finite differences on $\mathbb{S}^2$. Secondly, the spherical harmonic representation has a uniform resolution over $\mathbb{S}^2$ in that the truncated series (13) minimizes the L_2-norm of the approximation error on $\mathbb{S}^2$. We note that the number of modes is exactly half of the number of mesh points. For an arbitrary function with values defined on the $2N^2$ mesh points, a forward and backward discrete spherical harmonic transform does not recover the original point values. Instead, it yields a distribution in which the highly oscillatory modes near the poles are removed. This also removes any time step limitation associated with the close spacing of the collocation points near the poles: the eigenvalue of the Laplacian of each spherical harmonic mode is $n(n+1)$, which is far less than the inverse square of the mesh spacing near the poles.

On the cell surface, local coordinates are defined by the two tangents $\mathbf{a}_{1,2}$ and the surface normal $\mathbf{a}_3$,

$$\mathbf{a}_1 = \frac{\partial \mathbf{x}}{\partial \theta}, \quad \mathbf{a}_2 = \frac{\partial \mathbf{x}}{\partial \phi}, \quad \mathbf{a}_3 = \mathbf{n} = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{|\mathbf{a}_1 \times \mathbf{a}_2|}.$$ (16)

The first and second fundamental forms, which are used in subsequent manipulations, have components

$$a_{ij} = \mathbf{a}_i \cdot \mathbf{a}_j \quad \text{and} \quad b_{ij} = \mathbf{a}_{i,j} \cdot \mathbf{n} \quad (i,j = 1,2).$$ (17)

Each cell we model here has a stress-free biconcave reference shape set with a standard functional form [33,34]. The local deformation of the membrane is described by the surface deformation tensor $\mathbf{F}$ that maps the two reference tangents $\mathbf{a}_k^R$ to $\mathbf{a}_k$ and has the tensor form $\mathbf{F} = \sum_{\alpha=1}^{2}\mathbf{a}_\alpha \otimes \mathbf{a}^{\alpha R}$.

For an isotropic membrane, the Piola–Kirchhoff stress only depends on the two strain invariants [43],

$$I_1 = \lambda_1^2 + \lambda_2^2 - 2 \quad \text{and} \quad I_2 = \lambda_1^2\lambda_2^2 - 1,$$ (18)

where $\lambda_{1,2}$ are the eigenvalues of the left Cauchy–Green tensor $\mathbf{V}^2 = \mathbf{F}\cdot\mathbf{F}^T$. The elastic strain energy based on a neo-Hookean model [34] is

$$W = \frac{E_S}{4}\left(\frac{1}{2}I_1^2 + I_1 - I_2\right) + \frac{E_D}{8}I_2^2,$$ (19)

where E_S is the elastic shear modulus and E_D is the dilatation modulus, which acts as a penalty parameter so that the surface dilatation is nearly unity. The in-plane Cauchy stress tensor is

$$\tau = \frac{E_S}{2J_S}(I_1+1)\mathbf{V}^2 + \frac{J_S}{2}(E_D I_2 - E_S)\mathbf{P},$$ (20)

where $\mathbf{P} = \mathbf{I} - \mathbf{n}\otimes\mathbf{n}$ is the surface projection tensor and $J_S = \lambda_1\lambda_2$ is the dilatation. The linear isotropic model used for the bending moment is

$$M_\beta^\alpha = -E_B(b_\beta^\alpha - b_\beta^{\alpha R}),$$ (21)

where E_B is the bending modulus. For a sphere of radius r, $\mathbf{b} = -\mathbf{P}/r$, hence a compression of a sphere results in a positive bending momentum according to (21) as it should.

By applying the Stokes theorem to the local torque balance, the surface transverse tensor $\mathbf{Q}$ and the in-plane tension tensor $\mathbf{N}$ are defined by

$$M_{|\alpha}^{\alpha\beta} - Q^\beta = 0,$$ (22a)

$$e_{\alpha\beta}(N_{\alpha\beta} - b_\gamma^\alpha M^{\gamma\beta}) = 0,$$ (22b)

where the subscript '$|\alpha$' denotes the covariant derivative to θ^α. Eq. (22b) determines the antisymmetric part of $\mathbf{N}$, and the symmetric part is set by the constitutive law (20). Nevertheless, because of the constitutive model (21) used for the bending

moment **M**, the term $b_{\gamma}^{\alpha}M^{\gamma\beta}$ in (22b) is symmetric about α and γ, and so the antisymmetric part of **N** is always zero. By a local force balance, the hydrodynamic surface traction **f** is determined via

$$N_{|\alpha}^{\alpha\beta} - b_{\alpha}^{\beta}Q^{\alpha} + f^{\beta} = 0, \tag{23a}$$

$$Q_{|\alpha}^{\alpha} + N^{\alpha\beta}b_{\alpha\beta} + f^3 = 0 \tag{23b}$$

for $\beta = 1, 2$. Because of the global mapping on each cell surface, all derivatives are obtained from the spherical harmonic expansion and tensor algebra.

This model for the mechanics of the cell membranes is fairly general and has been used previously as a model for blood cells [33,34], and it therefore seemed to be a good candidate for demonstrating our numerical methods. In Section 11.2, we shall see that it provides a reasonable quantitative model for flowing cells. However, the overall algorithm we develop here is only loosely tied to this particular model of the red-cell dynamics, so long as the cells remain intact. Topological changes, such as in cell lysis, would require significant modification of the scheme because the spectral basis functions are restricted to sphere-like topologies. Such cases aside, there are several other constitutive models and modeling approaches [7,22,29] that are compatible with the overall proposed algorithm and which might have better predictive capabilities in certain circumstances.

For any realistic constitutive model, the membrane will introduce nonlinearity to the linear flow system, thereby introducing the nonlinear mechanisms of numerical instability that appear to have hampered solution in the past [33]. The in-plane Cauchy stress is in general a nonlinear function of the deformation tensor and the finite membrane deformation also introduces a geometric nonlinearity. Furthermore, the covariant derivative involves projection of tensor derivatives along the local surface tangents, which is also a nonlinear operation at finite deformation. All of these nonlinearities contribute to what are called aliasing errors in the context of numerical solutions of the full nonlinear Navier–Stokes equations [6]. Energy that is moved to unresolved scales (high mode numbers) by nonlinear effects is aliased back to resolved scales, where it facilitate a spurious increase in the magnitudes of high-wavenumber energy. Our approximate de-aliasing technique, which is discussed and demonstrated in Section 10, is essential for the overall efficiency of our algorithm.

4. Discretization of the no-slip wall boundaries

No-slip wall boundaries, such as the blood vessel walls, are discretized with a triangular surface mesh, on which the surface force density is represented by second-order accurate linear boundary elements build upon 7-point Guass quadratures [32]. This is implemented in a collocation scheme whereby the residuals of (9) at wall mesh points are constrained to be zero. The singular integrations on the wall are computed by the widely-used Duffy quadrature rule [11]. For nearly-singular close-range interactions this same singular integration is used to improve accuracy, but a fine wall mesh would be needed, of course, if the cells were to approach very close to the wall. We have not found this to be restrictive in our computations because Stokesian lifting suppresses close approaches to the wall and tends to maintain the well-known near-wall cell-free layer. The cell deformation and wall force density in all our simulations are insensitive to wall mesh refinement when the wall mesh size is smaller than the gap width. An example with a wall refinement is included in Section 10 showing that our typical resolution is more than sufficient. The focus of this paper is the discretization of the cells, which as can be seen from the block matrix structure of the cell–wall system is independent of the wall discretization. If resolving close cell–wall interaction were to become a challenge in some application, it would be possible to incorporate any higher-resolution scheme for the wall [26], either locally or globally, without changing the cell discretization.

5. Smooth particle-mesh Ewald sum (SPME)

5.1. Green's function splitting

The Stokes-flow Green's functions for periodic boundary conditions are calculated as an Ewald sum [15]. To do this, both Green's functions are decomposed into short-range singular (sr) and smooth (sm) parts,

$$\mathbf{G} = \mathbf{G}^{\text{sr}} + \mathbf{G}^{\text{sm}} \quad \text{and} \quad \mathbf{T} = \mathbf{T}^{\text{sr}} + \mathbf{T}^{\text{sm}}.$$

The short-range part of the single-layer Green's function **G** is

$$G_{ij}^{\text{sr}}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{a}} \text{erfc}(\tilde{r})\left(\frac{\delta_{ij}}{r} + \frac{r_i r_j}{r^3}\right) + \frac{2}{\sqrt{\alpha}}\sum_{\mathbf{a}} e^{-\tilde{r}^2}\left(\frac{r_i r_j}{r^2} - \delta_{ij}\right), \tag{24}$$

where $\mathbf{a} = (n_1 L_1, n_2 L_2, n_3 L_3)$ for integer $n_{1,2,3}$, $\mathbf{r} = \mathbf{x} - \mathbf{y} + \mathbf{a}$ is the vector of separation, and $\tilde{\mathbf{r}} = \sqrt{\pi/\alpha}\mathbf{r}$ is non-dimensional. The corresponding smooth part of **G** is

$$G_{ij}^{\text{sm}}(\mathbf{x}, \mathbf{y}) = \frac{2\alpha}{V}\sum_{\mathbf{k}\neq 0}\Phi_1(\tilde{k}^2)(\tilde{k}^2\delta_{ij} - \tilde{k}_i\tilde{k}_j)e^{i2\pi\mathbf{k}\cdot(\mathbf{x}-\mathbf{y})}, \tag{25}$$

where $\mathbf{k} = (N_1/L_1, N_2/L_2, N_3/L_3)$ is the wave number and $\tilde{\mathbf{k}} = \sqrt{\pi\alpha}\mathbf{k}$ is non-dimensional. The function Φ_1 in (25) is an incomplete γ-function defined

$$\Phi_\gamma(z) = \int_1^\infty e^{-zt}t^\gamma \, dt. \tag{26}$$

The periodic part of the Green's function **T** has a similar decomposition:

$$\breve{T}_{ijl}^{sr}(\mathbf{x}, \mathbf{y}) = -\frac{8\sqrt{\pi}}{\alpha} \sum_{\mathbf{a}} \Phi_{\frac{3}{2}}(\tilde{r}^2)\tilde{r}_i\tilde{r}_j\tilde{r}_l \tag{27}$$

and

$$\breve{T}_{ijl}^{sm}(\mathbf{x}, \mathbf{y}) = \frac{2\alpha}{V} \sum_{\mathbf{k}\neq 0} (i2\pi)(k_i\delta_{jl} + k_j\delta_{il} + k_l\delta_{ij})\Phi_0(\tilde{k}^2)e^{i2\pi\mathbf{k}\cdot(\mathbf{x}-\mathbf{y})} + \frac{\alpha^2}{\pi V} \sum_{\mathbf{k}\neq 0} (i2\pi)^3 k_i k_j k_l \Phi_1(\tilde{k}^2)e^{i2\pi\mathbf{k}\cdot(\mathbf{x}-\mathbf{y})}. \tag{28}$$

The Ewald parameter α determines the length scale of the decomposition. The short-range part of the Green's function decays exponentially with separation distance when $\tilde{r} \gg 1$, so it is truncated at short distance without introducing significant error. The smooth part is represented by Fourier coefficients that decay exponentially fast with increasing wave numbers for $\tilde{k} \gg 1$.

5.2. Smooth component

The smooth part of the Ewald sum is calculated by the smooth particle-mesh Ewald sum (SPME) method [8,9,41]. The central idea of SPME is to calculate the smooth Fourier Ewald sum on a uniform Cartesian mesh using fast Fourier transforms. B-splines are used to distribute the source singularities from their locations on the cells and walls to this regular mesh as well as to interpolate velocities computed on the mesh back to the surfaces. That this mesh extends beyond the vessel as in Fig. 1 might appear wasteful because it amounts to discretizing regions of space where the flow is not of interest. However, where there are no cells there is little expense associated with this portion of the calculation. There are no short-range interaction to evaluate and no interpolation from the collocation points on the cells to the mesh and back. Profiling shows that the entire forward and reverse FFT portion of the solver accounts for less than 20% of the calculation for the flows in tubes discussed in Section 11.2. This will increase for more complex geometry vessels, for which more of the volume of the periodic domain would fall outside the vessel, but can be also be significantly reduced by selecting α values that further reduce the work on the FFT mesh.

5.3. Short-range component

The short-range part of the sum is computed directly to the desired accuracy, using a small but sufficient number of close interactions. Convergence is rapid with increasing the range of included interactions because of the exponential decay for $r \gg \alpha^{1/2}$. However, the singularity at $r = 0$ requires additional attention. The boundary integral of this singular kernel over a cell surface D has a general form

$$I(\mathbf{x}_0) = \int_D K(\mathbf{x}, \mathbf{x}_0)f(\mathbf{x})\,dS(\mathbf{x}) = \int_{\mathbb{S}^2} K(\mathbf{x}(\theta,\phi),\mathbf{x}_0)f(\mathbf{x}(\theta,\phi))J(\theta,\phi)\sin\theta\,d\theta\,d\phi, \tag{29}$$

where K is any Cartesian component of the short-range part of the Green's function and $f(\mathbf{x})$ is any smooth function over D. With the reference sphere $\mathbb{S}^2$ discretized by a $N \times 2N$ mesh as discussed in Section 3, the average mesh spacing is $h = \sqrt{A/N^2}$ where A is the cell surface area. When the distance d between point $\mathbf{x}_0$ and surface D is much bigger than h, the kernel $K(\mathbf{x}, \mathbf{x}_0)$ is smooth over D and is well resolved by the surface mesh. We take the threshold distance to be $h^{1/2}$, so any $\mathbf{x}_0$ whose distance to D is greater than $h^{1/2}$ is considered well separated from the surface. The surface integral (29) is thus computed by

$$I \approx \sum_{i=1}^N \sum_{j=1}^{2N} K(\mathbf{x}_{ij}, \mathbf{x}_0)f(\mathbf{x}_{ij})J_{ij}w_{ij}, \tag{30}$$

where $\mathbf{x}_{ij} = \mathbf{x}(\theta_i, \phi_j)$ is the coordinate of the quadrature point at (θ_i, ϕ_j), and w_{ij} is the quadrature weight. Using the Gauss points in θ and uniform points in ϕ, the quadrature (30) converges exponentially with mesh size h.

5.4. Singular and nearly-singular points

If $\mathbf{x}_0$ in (29) lies on or is close to D (i.e. $0 \leqslant d < h^{1/2}$), then the Green's function kernel is singular or considered to be nearly singular, for which the quadrature (30) will have poor accuracy and in general will not converge with $h \to 0$. Doing this in the nearly-singular case amounts in essence to local mesh refinement for accurate evaluation of the integrals in the 'lubrication limit' of close inter-cell spacing. In both cases, the singular surface integrals are computed by the method of floating partition of unity [5,46]. When $\mathbf{x}_0 \in D$, a local polar patch centered on $\mathbf{x}_0$ is defined on the reference sphere. For any other point

$\mathbf{x} \in D$, $\rho(\mathbf{x},\mathbf{x}_0)$ is defined to be the distance along the great circle that connects $\mathbf{x}$ and $\mathbf{x}_0$ on $\mathbb{S}^2$. This coordinate is used in the mask function

$$\eta(\rho) = \begin{cases} \exp\left(\frac{2e^{-1/t}}{t-1}\right) & \text{for } t = \rho/\rho_1 < 1, \\ 0 & \text{for } \rho \geq \rho_1, \end{cases} \tag{31}$$

where ρ_1 is a cut-off radius. Since $\eta(\rho)$ is smooth it is amenable to accurate quadrature calculations. With $\eta(\rho)$, the surface integral is split into two parts

$$I = I_1 + I_2 = \int_D K(\mathbf{x},\mathbf{x}_0)\eta(\rho(\mathbf{x},\mathbf{x}_0))f(\mathbf{x})\,dS(\mathbf{x}) + \int_D K(\mathbf{x},\mathbf{x}_0)[1 - \eta(\rho(\mathbf{x},\mathbf{x}_0))]f(\mathbf{x})\,dS(\mathbf{x}). \tag{32}$$

The integrand of I_1 has support only in the patch. To evaluate it, we first transform to the local polar coordinate system, which yields

$$I_1 = \int_0^{2\pi} \int_0^{\rho_1} K(\mathbf{x},\mathbf{x}_0)\eta(\rho)f(\rho,\varphi)\sin\rho\,d\rho\,d\varphi. \tag{33}$$

For each fixed $\varphi \in [0,2\pi)$, the integrand is finite, though in general it is discontinuous across the pole. The integral can nevertheless be calculated accurately by first using Gauss quadrature along each radial line for ρ from zero to ρ_1 and then summing over φ, in which direction the integrand is periodic and thus evaluated on a uniform mesh. The integral I_2 in (32) is smooth over D since its integrand vanishes at $\mathbf{x} = \mathbf{x}_0$, and hence is accurately computed by the quadrature rule (30). Since the mask function changes from unity at the center of the patch to zero on the patch boundary, the radius of the patch must be chosen so that the integrand of I_2 is well resolved by the surface mesh. In our calculations, we choose the patch radius on the reference sphere to be $\rho_1 = \pi/\sqrt{N}$ so that the patch radius in $\mathbb{R}^3$ is $O(h^{1/2})$. Inside the patch, $\sqrt{N}$ quadrature points are used in the ρ direction and $2\sqrt{N}$ in the φ direction, giving a point density that is thus comparable to that of the surface mesh. The singular integration error is $O(h^3)$ by this choice of patch size [46]. Higher-order accuracy is achieved with larger polar patch sizes such that $\rho_1 \propto N^{-\beta}$ and $\beta < 1/2$, albeit with more computational cost.

The quadrature points for I_1 are defined on the local polar coordinate patch, and do not coincide with the surface mesh points. Interpolation is needed to evaluate the surface coordinates and other function values on those quadrature points, for which we use bi-cubic splines. Interpolating with the global spherical harmonics would be more accurate, but spectral convergence would not be preserved because of the $O(h^3)$ limitation above, so the expense is not justified. The overall scheme is more efficient for the same formal convergence rate using a spline interpolation. To construct the bi-cubic spline approximation for a function $f(\theta,\phi)$, we take advantage of the periodicity of functions on the unit sphere, without assuming any of the additional restrictions of this particular geometry. We first compute the function values of f on a surface mesh that is uniform in both $\theta \in [0,\pi]$ and $\phi \in [0,2\pi)$ from the spherical harmonic coefficients of f. The domain of definition of f is then extended to $\theta \in [0,2\pi)$ by using the symmetry $f(2\pi - \theta, \pi + \phi) = f(\theta,\phi)$. The extended function is periodic in both directions, hence the function derivatives at nodes, needed to develop the spline, are computed by fast Fourier transform.

For nearly-singular integration, the projection point of $\mathbf{x}_0$ on surface D is found first as $\mathbf{x}_p$, and the integral is split by the same partition of unity (32) with the polar coordinate patch centering around $\mathbf{x}_p$. In all cases, to resolve the rapidly varying Green's function kernel near $\mathbf{x}_p$ in the polar patch, we use a sinh transformation in radial direction to cluster the quadrature points near its center [17].

6. Physical parameters

Following [34], all quantities are non-dimensionalized by

- The equivalent cell radius $a^* \approx 2.82\,\mu$m, where a red cell has the same volume as a sphere of radius a^*.
- A reference shear rate $k^* = 100\,\text{s}^{-1}$, which gives a reference velocity $U^* = k^*a^* = 282\,\mu$m/s.
- The plasma viscosity $\mu^* = 1.2 \times 10^{-3}\,\text{kg m}^{-1}\,\text{s}^{-1}$.

The following values are used for cell membrane shear and bending modulus,

$$E_S^* = 4.2 \times 10^{-6}\,\text{N/m} \quad \text{and} \quad E_B^* = 1.8 \times 10^{-19}\,\text{N/m} \tag{34}$$

and their non-dimensional values are

$$E_S = \frac{E_S^*}{\mu^*k^*a^*} = 12.4 \quad \text{and} \quad E_B = \frac{E_B^*}{\mu^*k^*a^{*3}} = 0.0669. \tag{35}$$

The membrane dilatational modulus is set as a penalty parameter $E_D = 200$. With this value, the local area change stays below 2% in all our simulations.

7. Numerical time step limitation

To estimate the maximum allowed time step for the explicit time integration, we consider the following linearized time evolution equation for the cell membrane coordinates $\mathbf{X}$,

$$\frac{d\delta\mathbf{X}}{dt} = \mathbf{HR}\delta\mathbf{X}, \tag{36}$$

where $\mathbf{R}\delta\mathbf{X}$ is the response of the membrane residual force to a position perturbation $\delta\mathbf{X}$, and $\mathbf{H}$ is the inverse of the boundary integral operator. The maximum allowable time step is inversely proportional to the spectral radius of the discretized right-hand side operator $\mathbf{HR}$. Since $E_D \gg E_S$, the spectral radius of the force matrix $\mathbf{R}$ is determined primarily by the dilatation and bending. Let the effective surface mesh spacing be $h_{\text{eff}} = \sqrt{A}/N$, where A is the membrane area and N is the number of latitudinal modes in (13). Then the dilatation and bending parts of the $\mathbf{R}$ operator have spectral radii that scale as $E_D h_{\text{eff}}^{-2}$ and $E_B h_{\text{eff}}^{-4}$, respectively. The scaling of the spectral radius of the discrete operator $\mathbf{H}$ with h_{eff} is however difficult to estimate, even more so for the combined $\mathbf{HR}$ operator.

The empirical time step stability limit for the simulations in Section 10 is found to be

$$\Delta t < \min\left(\frac{2.5 h_{\text{eff}}^{1.2}}{E_D}, \frac{0.19 h_{\text{eff}}^{3.2}}{E_B}\right), \tag{37}$$

where h_{eff} is computed using the estimated cell surface area $A \approx 16.8$. Therefore, the spectral radius of $\mathbf{HR}$ scales as $E_D h_{\text{eff}}^{-1.2}$ for dilatation and $E_B h_{\text{eff}}^{-3.2}$ for bending. The spectral radius of $\mathbf{R}$ alone results in $\Delta t < O(h_{\text{eff}}^2/E_D)$ and $\Delta t < O(h_{\text{eff}}^4/E_B)$. The Stokes flow, which provides $\mathbf{H}$, is dissipative, so a less restrictive stability criterion than the $\mathbf{R}$-only case is as expected.

The time step limits due to dilatation and bending in (37) are equal at $h_{\text{eff}} \approx 3.6\sqrt{E_B/E_D} = 0.066$, which translates to $N = 62$ in (13). Though not an unreasonably high resolution, this surface resolution is higher than all our simulations performed so far, so Δt is constrained primarily by the dilatational stiffness in our simulations.

8. Cell-surface quadrature convergence

The numerical errors include the truncation error of the spherical harmonic representations over the cell surface, the truncation error in the Ewald sum, the error in the surface quadratures, and the time integration. The truncation error in the spherical harmonic expansion is expected to decay exponentially with the number of latitudinal and longitudinal modes. The short-range part of the Ewald sum should also converge exponentially with the increasing cut-off distance, and the smooth part should converge algebraically with a rate equal to the order of the B-spline interpolation [41]. The error in the time integration is independent of the spatial representation and is similarly well understood. We have confirmed that small increases in cut-off distance, increasing B-spline order, or increasing the mesh for evaluation of the smooth part of the Green's function all reduce errors quickly and as expected, so we do not dwell on these errors here. For simulations with reasonable surface resolution and high-order (*e.g.* sixth-order) B-spline interpolations, the most challenging to anticipate errors are those incurred in the numerical integration, particularly the treatment of the singular term, which we consider briefly here.

We calculate the singular integrand error versus mesh density for integrations over the strained ellipsoid shown in Fig. 2. The functional form of the surface is given by

$$\frac{x^2}{a^2 s(z)^2} + \frac{y^2}{b^2 s(z)^2} + \frac{z^2}{c^2} = 1,$$
$$a = 0.8, \quad b = 0.9, \quad c = 1.0$$

with a squeezing factor

$$s(z) = 1 - 0.7\cos\left[\frac{\pi}{2}\left(\frac{z}{c} + 0.2\right)\right].$$

The north and south poles of the surface mesh are intentionally rotated away from that of the ellipsoid to avoid any symmetry in the azimuthal direction. The two analytical surface integrals we consider are the so-called single- and double-layer integrals that would be encountered in rigid body translation:

$$\mathbf{0} = \int_D G_{ij}(\mathbf{x}, \mathbf{x}_0) n_j(\mathbf{x}) \, dS(\mathbf{x}) \tag{38}$$

and

$$-4\pi U_i = \int_D U_j T_{jik}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) \, dS(\mathbf{x}), \quad \mathbf{x}_0 \in D \tag{39}$$

for constant velocity vector $\mathbf{U}$. These are representative of the integrals encountered in the full formulation.
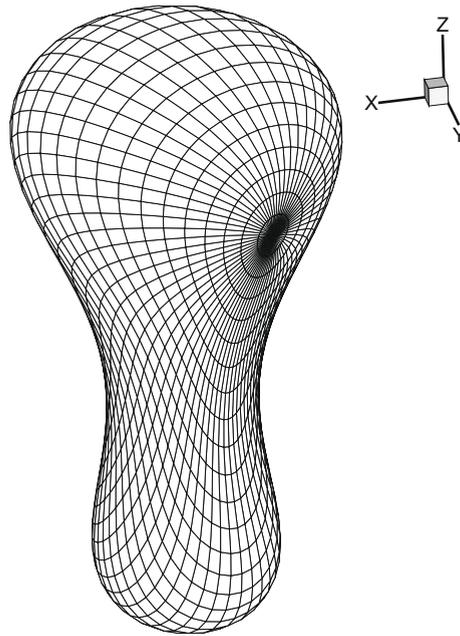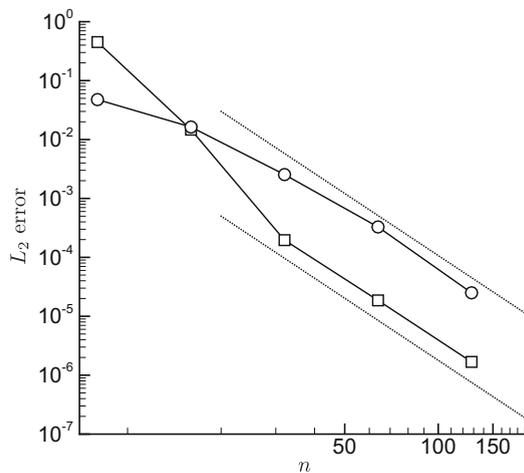
Fig. 2. Convergence study surface mesh.

Fig. 3. Error for ○ the single-layer integral (38) and □ the double-layer integral (39). The dotted lines show slope 3.5-order convergence.

The L_2-norm of the integration error is shown in Fig. 3 with the number of latitudinal mesh points increasing from $n = 8$ to 128. The error tolerance of the Ewald sum is 10^{-9} so that the measured error is entirely dominated by the surface discretization and the surface integration. The asymptotic convergence order is seen to be between 3 and 4, which is a consequence of the $O(h^{1/2})$ patch size for the singular integration [46].

9. Spatial and temporal resolution

For a spherical capsule undergoing small deformation in a quiescent fluid, its surface eigenmode evolution can be obtained analytically from a perturbation theory [38]. Specifically, the displacement of the bending mode is

$$\mathbf{u} = [A_{lm}^Y \mathbf{Y}_{lm}(\theta, \phi) + A_{lm}^Z \mathbf{Z}_{lm}(\theta, \phi)]e^{-\omega t}, \tag{40}$$

where $\mathbf{Y}_{lm} = Y_{lm}\hat{\mathbf{r}}$ and $\mathbf{Z}_{lm} = r^{-1}\nabla Y_{lm}/\sqrt{l(l+1)}$ are normalized vector spherical harmonics. The coefficients A_{lm}^Y and A_{lm}^Z can be solved from the secular equation (24) in reference [38]. When surface inertia is ignored, ω is real and positive in (40) and the
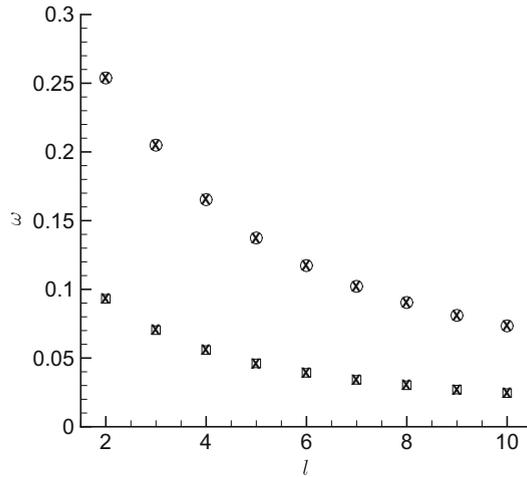
Fig. 4. The damping coefficients of surface bending modes with $2 \leqslant l \leqslant 10$ for two viscosity ratios: $\circ$ $\lambda = 1$ and $\square$ $\lambda = 5$, with corresponding $\times$ perturbation theory solutions [38].

surface mode represents a non-oscillatory exponential damping motion. It is noteworthy that ω is independent of azimuthal wave number m for bending modes.

As a validation test as well as testing the spatial and temporal resolution of the numerical scheme, we simulate the motion of a slightly deformed spherical capsule of unit radius and compare the results with theory. The elastic energy model,

$$W = \frac{1}{2}E_D[\mathrm{tr}(\epsilon)]^2 + E_S\mathrm{tr}(\epsilon^2),$$ (41)

is used in the theoretical model, where $\epsilon = (\mathbf{F}^T\mathbf{F} - \mathbf{I})/2$ is the strain tensor and E_D and E_S correspond to the dilatational and shear moduli labeled as λ and μ in Ref. [38]. The values $E_D = 2.5$ and $E_S = 1.0$ are chosen here; the bending stiffness is set to be zero. This results in the in-plane strain–stress relation

$$\mathbf{N} = J^{-1}\mathbf{F}\mathbf{F}^T[E_D\mathrm{tr}(\epsilon)\mathbf{I} + E_S(\mathbf{F}\mathbf{F}^T - \mathbf{I})].$$ (42)

For simulation, the capsule is inside a cubic periodic domain of size 20, and the initial perturbation is a bending surface eigenmode whose L_2 norm is 10^{-4}. The spherical harmonic expansion (13) has $N = 12$. For two viscosity ratios $\lambda = 1$ and 5, the damping coefficient ω of deformation eigenmodes with $2 \leqslant l \leqslant 10$ and $m = 2$ are extracted from their exponentially decaying amplitudes obtained from simulations with $\Delta t = 10^{-3}$. Since the Cartesian components of $\mathbf{Y}_{lm}$ contain modes with latitudinal wave number $(l+1)$, $l = 10$ is indeed the maximum wave number that is allowed by the expansion with $N = 12$. As shown in Fig. 4, all ω values are in good agreement with the theory, even those at the limit of the resolution of the chosen mesh. The maximum relative error is less than 0.2%, which is comparable to the numerical integration error.

The error in the forward Euler time integration can be accurately assessed using this model problem. For the linear damping system $du/dt = -\omega u$, a simple analysis shows that the error in the damping coefficient ω is

$$\frac{\Delta\omega}{\omega} = -\frac{\ln(1 - \omega\Delta t)}{\omega\Delta t} - 1 = \frac{\omega\Delta t}{2} + \frac{(\omega\Delta t)^2}{3} + O(\omega\Delta t)^3.$$ (43)

The change of $\Delta\omega/\omega$ with $\omega\Delta t$ is shown in Fig. 5, and the convergence is first order as expected.

10. Aliasing errors

10.1. Nonlinearity, aliasing, and de-aliasing

Aliasing arises from the finite resolution of any discretization, but is distinct from the truncation errors associated with, say, differentiation schemes [6,19,27] As discussed in Section 3, the cell surface residual force depends nonlinearly on the deformation, and the aliasing error introduced by this nonlinearity is a potential source of numerical instability. These non-linearities tend to broaden the spectrum of the deformations of the cells. This is physically realistic; sharper features do form on the cells. The turbulence energy cascade is a commonly cited example of this [6]. A consequence of this nonlinear spectral broadening is that energy is moved to scales that can not be resolved by the discretization. It would be consistent with the selected resolution for this energy to be removed from the solution, but instead it is aliased to the resolved scales [6,19,27], which can in turn drive additional aliasing and potentially lead to numerical instability.
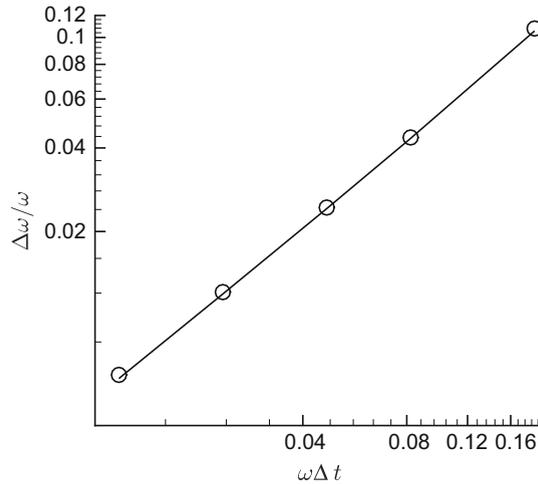
Fig. 5. The relative error in damping coefficient ω due to time integration error: ○ simulation and error estimation (43).

Dissipative numerics or explicit filtering can suppress instability due to aliasing, but both degrade the accuracy of the solution for fixed resolution. Increasing resolution also counters it, but at the cost of more points as well as a more restrictive time step stability limits for explicit time integration. As seen in Section 7, the impact of the time step restriction is potentially severe. So-called de-aliasing procedures are preferable since the resolved solution is not directly affected and more resolution is not required just for the sake of countering what is usually a slowly growing instability. For the quadratic nonlinearity of the Navier–Stokes equations, the well-known 3/2-rule eliminates aliasing errors [6,27]. In that case, the nonlinear operations are performed with 50% more mesh points, followed by a filtering of the solution. The nonlinearities in the present formulation are higher order and non-polynomial, the second of which precludes exact de-aliasing. Experimentation shows that a factor-of-three sufficiently suppresses aliasing for all our test scenarios, though most are stable with factor-of-two de-aliasing. We denote the larger mesh size M. For example, for factor-of-two de-aliasing, the nonlinear calculations are done on an $M \times 2M = 2N \times 4N$ mesh for the truncated spherical harmonic representation (13).

10.2. De-aliasing example

To demonstrate these aliasing and de-aliasing issues, we simulate the flow of a three-dimensional cell through a tight constriction: a tube of length $L = 10$ and diameter $D = 3.5$ with a constriction that necks this down to local diameter $d(z) = 3.5 - 2.2 \exp[-1.2(z-5)^2]$. A $U = 2$ mean velocity drives the flow. The cell is placed initially as seen in Fig. 6(a) at $z = 3.25$ and shifted $\Delta x = 0.25$ off the symmetry axis of the tube. A matched viscosity $\lambda = 1$ was used because it is expected to be less stable and the cell should also more quickly develop small-scale surface features which challenge the resolution of the numerical method. The subsequent evolution of the cell is seen in Fig. 6. As it flows through the constriction, the membrane appears to buckle, forming thin rib-like features.

This flow was simulated with N ranging from $N = 8$ to 64. Visually it is clear in Fig. 7 that though stable $N = 8$ only captures the gross features of the deformation. Buckling with rib-like features is seen for $N = 12$, but these ribs only achieve their large N form for $N \geqslant 16$. Visually, there is no change for $N \geqslant 24$, and the difference between $N = 16$ and 24 are hard to discern. The $N = 8$ case required factor-of-three de-aliasing to be stable, but $N \geqslant 12$ cases required only factor-of-two de-aliasing. The $N = 32$ case (and a $N = 48$ case not shown) ran with no de-aliasing up to when the rib features form at $t \approx 0.9$. The $N = 64$ case was the only one that was stable without de-aliasing (aside from the usual practice of zeroing the highest mode number which yields ambiguous derivatives). Fig. 8 shows a quantitative measure of the resolution via the energy spectra

$$E_n = \sum_{m=0}^{n} |\hat{\mathbf{x}}_n^m|^2, \quad n = 0, 1, \ldots, N-1. \tag{44}$$

Clearly, cases stabilized via de-aliasing provide an excellent model for the behavior of the modes they represent. Even the poor resolution of the $N = 8$ case provides a good model for those small n modes. Indeed, for $N = 8$ the peak increase in the pressure drop is predicted within 10% of the $N = 64$ case and to within less than a 1 percent for $N = 12$ case. Thus, with de-aliasing the accuracy can be set based upon the simulation and modeling objectives, without concern for stability.

For these simulations, the wall was discretized with 25,804 triangular elements. Increasing this to 50,996 elements for the $N = 24$ case yielded a spectrum that was indistinguishable to plotting accuracy. The net cell deformation energy $\sum E_n$ dif-
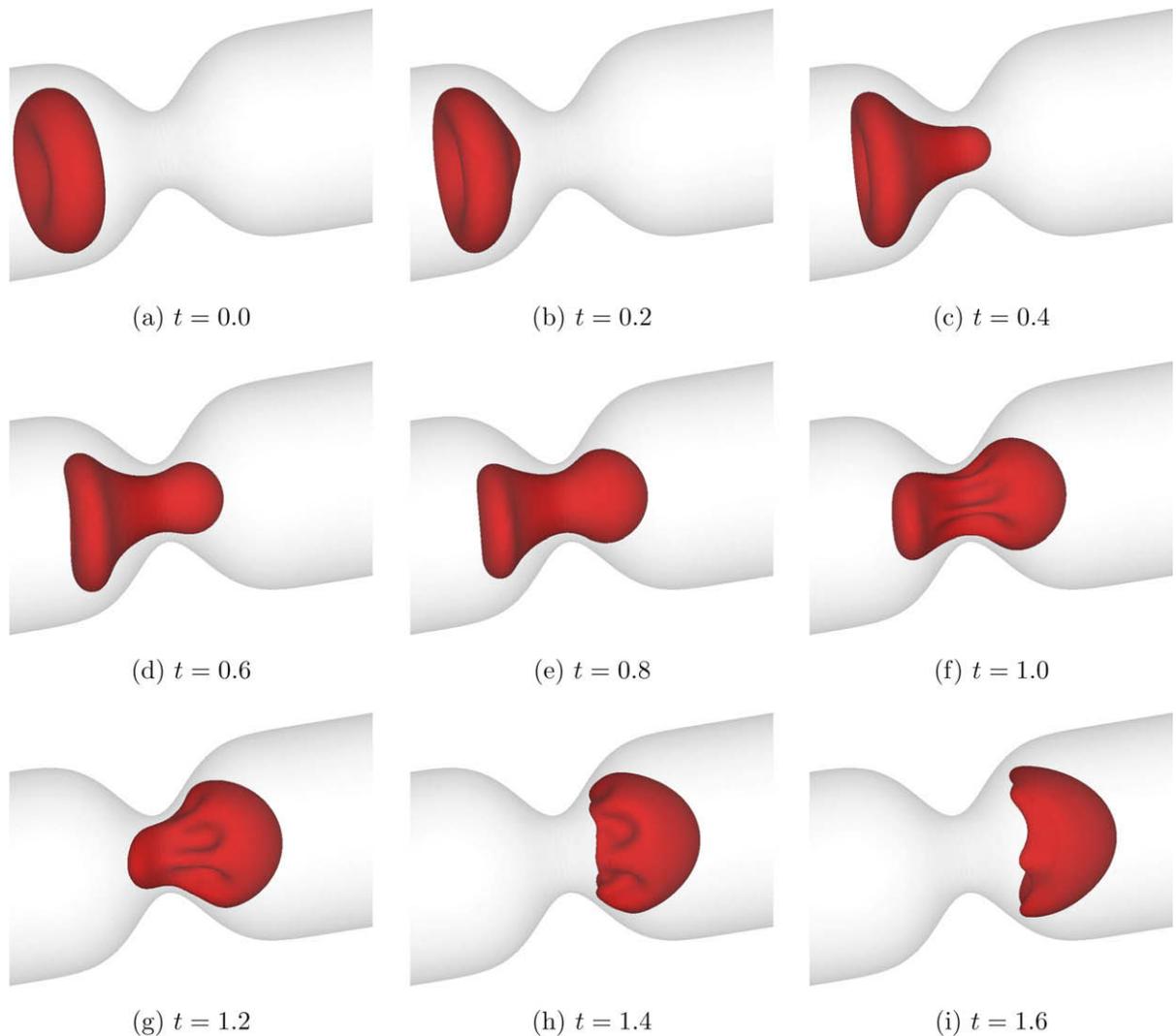
Fig. 6. Evolution of a well resolved cell ($N = 32$, $M = 64$) as it passes through a narrow neck as described in the text.

fered by 0.012% for the coarser wall simulation. Likewise, for the $N = 16$ case, halving the time step changed the net deformation energy by 0.05%. This is comparable to the time step and resolution used in many of the other examples presented.

Because the cell is significantly distorted in this example, it also is useful for illustrating the basic volume conservation of the scheme and how we enforce a constraint such that it can be exactly conserved. The boundary integral formulation is, of course, predicated on a $\nabla \cdot \mathbf{u} = 0$ condition, but its enforcement depends upon the numerical accuracy of the calculation, so volumes are not necessarily exactly preserved in the computation. The elastic reference configuration suppresses deviation from the reference volume, so it is not expected that typical flow will show a continuous drift in cell volume. However, left unconstrained the volume does vary as seen in Fig. 9 for the $N = 16$ case of the constriction flow. The unconstrained volume first increased slightly and then dropped to 99.5% of the initial volume. This is thus unlikely to be a significant factor in most simulations. Nevertheless, a variational formulation of volume enforcement as a constraint yields a correction in which the cell surface is adjusted in its normal direction. This is identical to our reported formulation for two-dimensional simulations [14], which is not repeated here. Doing this provides the small (typically $\lesssim 10^{-4}$) correction each time step to strictly maintain volume conservation, as also seen in Fig. 9.

11. Flow in cylindrical vessels

11.1. Axisymmetric flow

Pozrikidis [34] simulated axisymmetric cellular flow in a cylindrical tube by using the Green's functions for axisymmetric Stokes flows with the appropriate no-slip condition imposed on the tube wall. For comparison and validation purposes, we
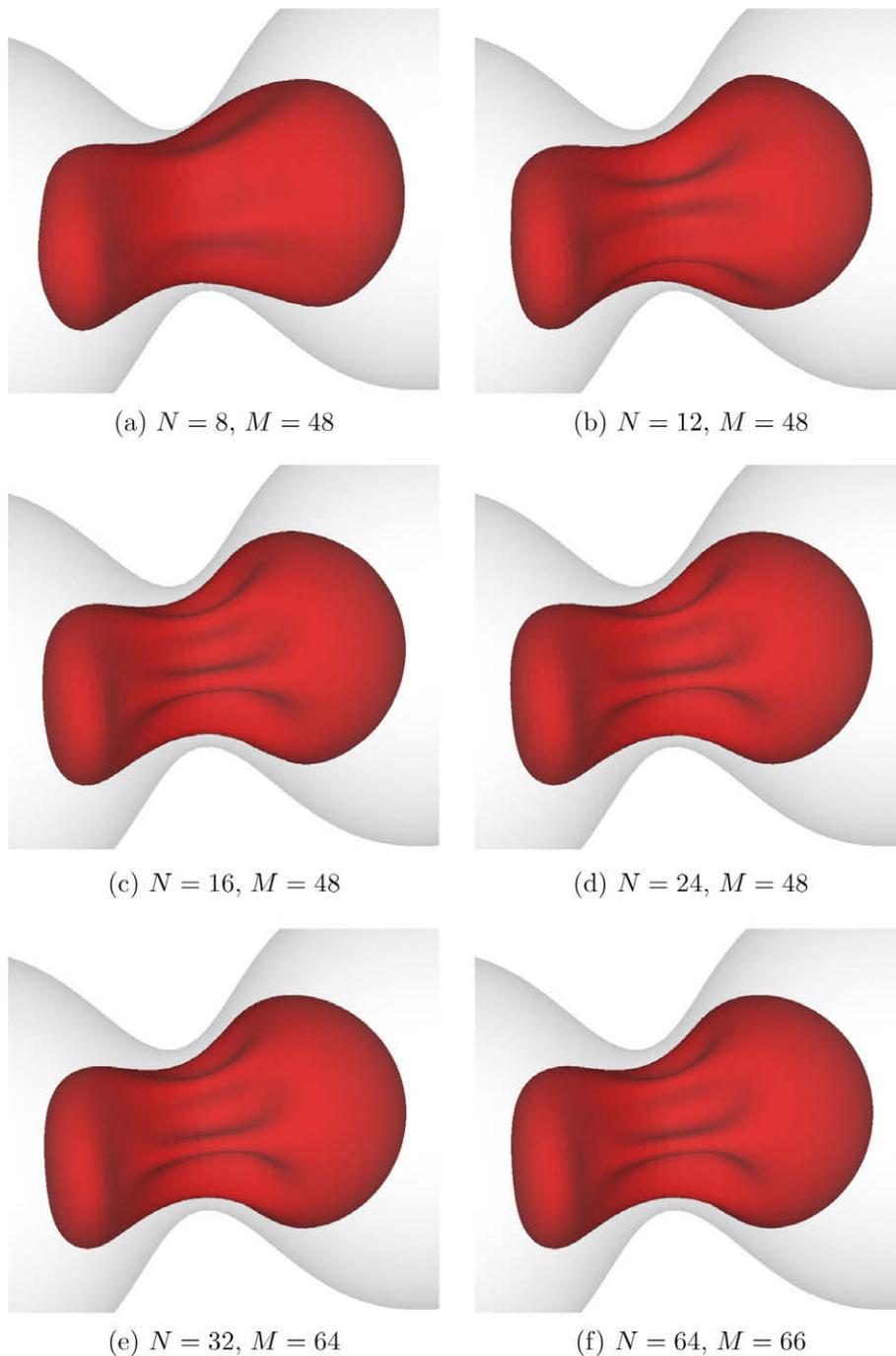
(a) $N = 8$, $M = 48$

(b) $N = 12$, $M = 48$

(c) $N = 16$, $M = 48$

(d) $N = 24$, $M = 48$

(e) $N = 32$, $M = 64$

(f) $N = 64$, $M = 66$

Fig. 7. Cell visualized in Fig. 6(f) at $t = 1$ for increasing resolutions.

simulate systems of the same geometry with our general three-dimensional formulation. The system consists of two cells which are evenly spaced along the center line of the tube whose non-dimensional diameter $D = 3.2$ corresponds to a dimensional diameter of 9.024 μm. With centers of cells separated by a distance 2, the tube hematocrit H_t (the volumetric fraction of cells) is about 26%. At $t = 0$, a nonzero background velocity $\langle \mathbf{u} \rangle$ along the tube axial z-direction is applied to induce the flow inside the tube. The cells deform into umbrella-like shapes under the action of shear, and the deformation increases with flow velocity. At steady state, each cell undergoes pure translation along the tube centerline without any relative motion of the membrane. Fig. 10 shows the increasingly deformed cell contours with flow rates.

To obtain the total tube flow rate, it is in general necessary to numerically integrate the axial velocities over a tube cross section. However, for this simple geometry, it can be calculated from the total mean axial velocity $\langle u_z \rangle$ and the mean pressure
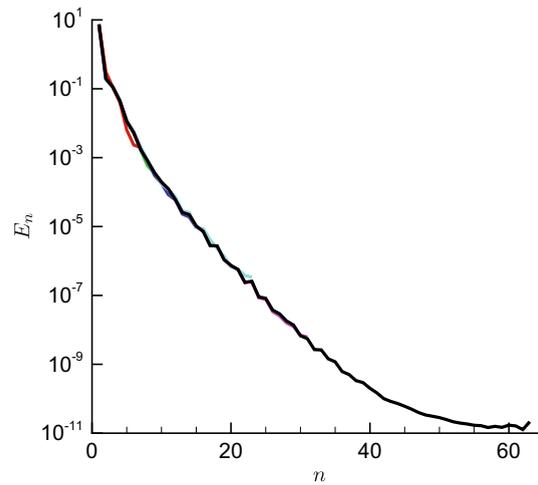
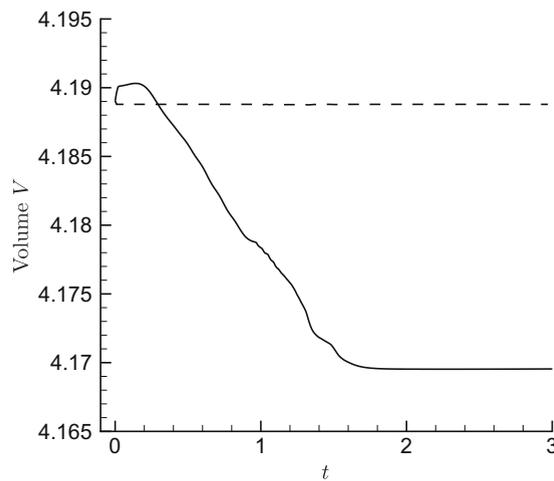**Fig. 8.** Spectra for the cases visualized in Fig. 7: $N = 8$, 12, 16, 24, 32, and 64.

Fig. 9. Volume for — unconstrained and --- volume constrained simulations of the constriction flow for $N = 16$.
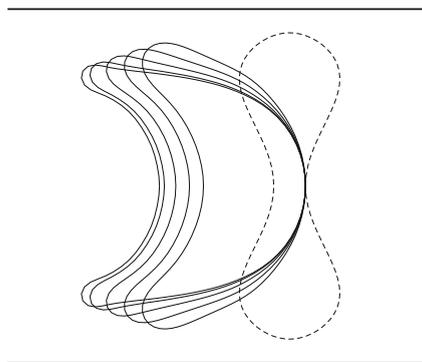

Fig. 10. The axisymmetric deformation of cells in a cylindrical tube at steady state. The straight solid line represents the tube wall boundary. The dashed line is the unstressed cell, and the solid lines are increasingly deformed cells at mean tube flow velocity of 0.62, 1.25, 2.52, 5.09 and 7.66.

gradient $\langle \partial_z p \rangle$ as follows. When the cross section of the computational domain is square ($L_1 = L_2$), the volumetric flow rate for fluid outside of the tube is [25],

$$Q_{\text{external}} = \frac{L_1^4 \langle \nabla_z p \rangle}{4\pi\mu} (\ln \chi^{-1} - 1.47644 + 2\chi - 0.5\chi^2 - 0.0509713\chi^4 + 0.077465\chi^8 - 0.109757\chi^{12}$$
$$+ 0.122794\chi^{16} - 0.146135\chi^{20} + 0.244536\chi^{24}) + O(\chi^{28}), \tag{45}$$

where $\chi = \pi a_{\text{tube}}^2 / L_1^2$ is the volume fraction of the cylindrical tube in the whole computational domain. The flow rate inside the tube is then given by $Q_{\text{tube}} = L_1^2 \langle u_z \rangle - Q_{\text{external}}$.

The effective viscosity μ_{eff} describes the increase of the wall drag coefficient due to the addition of cells and is computed from the total tube flow rate and the mean pressure gradient as

$$\mu_{\text{eff}} = \frac{\pi a_{\text{tube}}^4 \langle \nabla_z p \rangle}{8 Q_{\text{tube}}}, \tag{46}$$

where $\mu = 1$ is the non-dimensional plasma viscosity. A pure plasma flow has a parabolic Poiseuille profile with $\mu_{\text{eff}} = \mu = 1$. The tube flow has $\mu_{\text{eff}} > 1$ because of the increased viscous dissipation from the presence of cells; μ_{eff} decreases as flow rate increases because the gap between the cell and the vessel wall widens as the cell deforms more. The shear-thinning behavior is seen in Fig. 11, where our result is verified against the reported axisymmetric simulations.

11.2. Three-dimensional flow

Three-dimensional simulations are presented here for six straight circular tubes of diameter ranging from $D = 1.75(4.9\,\mu\text{m})$ up to $D = 6.0(16.9\,\mu\text{m})$, all with $H_t = 0.30$, which is a typical hematocrit in the microcirculation. The background velocities are set to be a constant $\langle u_z \rangle = 5$ except the $D = 1.75(4.9\,\mu\text{m})$ and $D = 1.9(5.4\,\mu\text{m})$ cases which had $\langle u_z \rangle = 10$. This was increased in these cases because the smaller diameter and higher effective viscosity in these smaller tubes tend to slow the flow below the shear rates considered in the corresponding experiments, which are discussed briefly below. Resolution was $N = 32$ for the $D = 1.75(4.9\,\mu\text{m})$ and $D = 1.9(5.4\,\mu\text{m})$ cases, $N = 24$ for the $D = 2.0(5.6\,\mu\text{m})$ and $D = 2.5(7.1\,\mu\text{m})$ cases, and $N = 16$ for the $D = 3.5(9.9\,\mu\text{m})$ and $D = 6.0(16.9\,\mu\text{m})$ cases. All had a de-aliasing factor of three.

Typical cell configurations are visualized for all these cases in Fig. 12. It is seen that for the smaller tubes, the red cells form the expected parachute shape seen in experiments. For the $D = 6.0$ case, the cells appear to be randomly distributed, with the $D = 3.5$ seeming to be a switch-over point to this other behavior. All cases show a near-wall cell-depleted layer which is a well-known feature of blood flow in small vessels or tubes. This is particularly thick relative to the diameter for the $D = 3.5(9.9\,\mu\text{m})$ case.

At the hematocrit simulated, the distance between neighboring cells can become small. Theoretically, the distance should remain finite as predicted by lubrication theory, but without corrective measures contact and penetration can occur due to the finite numerical errors in the surface velocity calculation and time integration. In two-dimensional simulations, short-range repulsion forces have been designed that allow close approach but prevent surface overlaps [14]. Numerical experiments with this approach also suggest that significantly larger forces are needed in three dimensions because the contact region is initially point-like, rather than line-like in two dimensions. That is, larger local forces applied at 'points' are needed to separate three-dimensional cells than the local forces applied on 'lines' in two dimensions. As the repulsion force should decay rapidly from the contact points, the mesh resolution needed for resolving the repulsion force in the contact region will be greater than that for accurate surface representation and boundary integrals. So, here a purely kinematic collision detec-
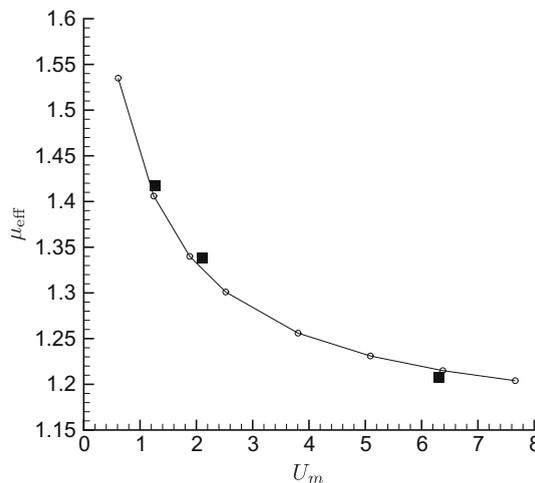

Fig. 11. Effective viscosity of axisymmetric cellular flow inside a narrow tube: ○ present three-dimensional simulations, and ■ corresponding axisymmetric simulations of Pozrikidis [34].
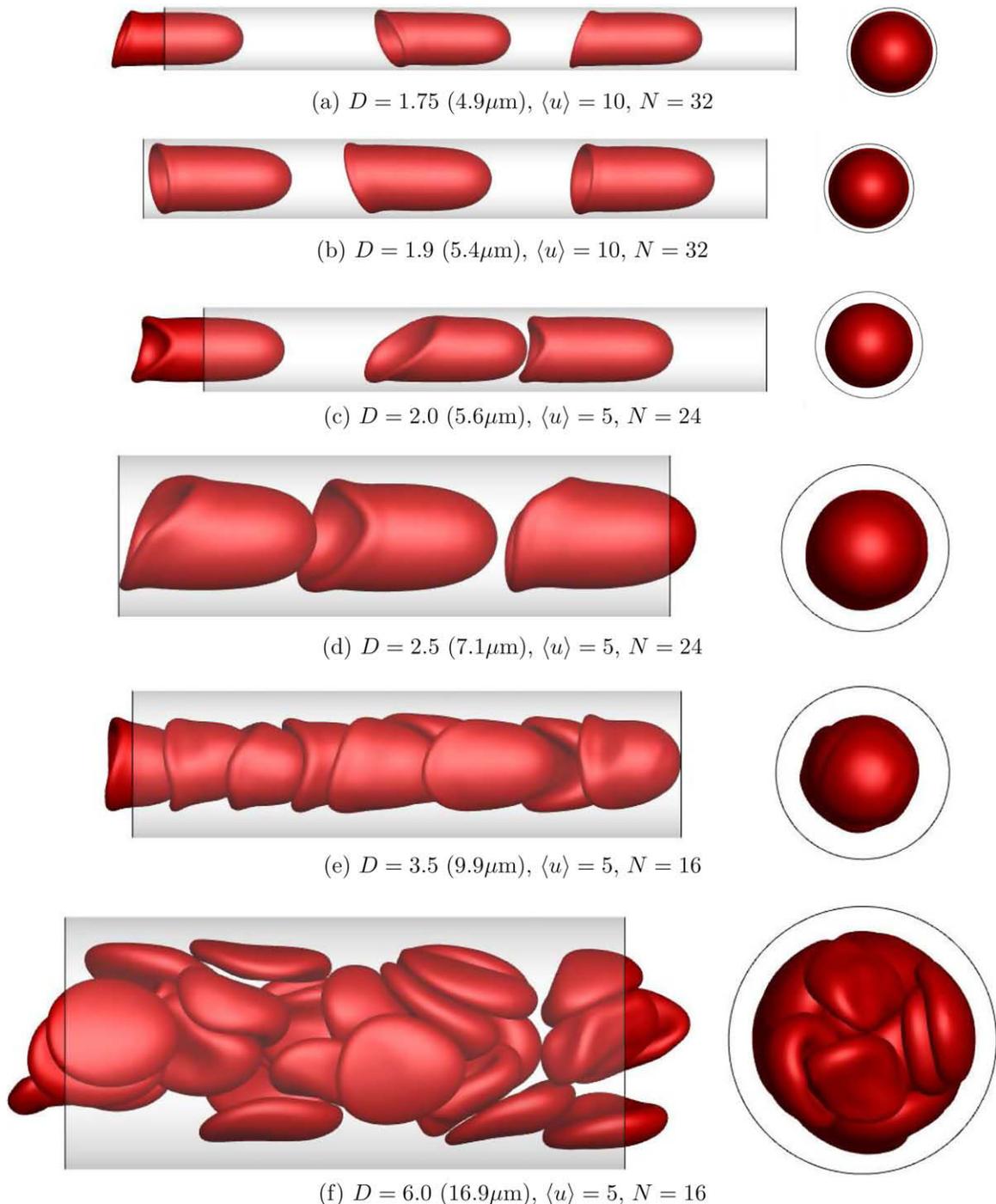
(a) $D = 1.75$ $(4.9\mu m)$, $\langle u \rangle = 10$, $N = 32$

(b) $D = 1.9$ $(5.4\mu m)$, $\langle u \rangle = 10$, $N = 32$

(c) $D = 2.0$ $(5.6\mu m)$, $\langle u \rangle = 5$, $N = 24$

(d) $D = 2.5$ $(7.1\mu m)$, $\langle u \rangle = 5$, $N = 24$

(e) $D = 3.5$ $(9.9\mu m)$, $\langle u \rangle = 5$, $N = 16$

(f) $D = 6.0$ $(16.9\mu m)$, $\langle u \rangle = 5$, $N = 16$

Fig. 12. Three-dimensional cellular flow in cylindrical tubes. Flow is from left to right in the side views and toward the viewer in the end views. The flows are streamwise periodic; only one periodic image of each cell is shown, but these in cases appear to extend beyond the tube.

tion and removal procedure is used instead of a repulsion force. A small threshold h_s separation is defined, which for our demonstration simulations is set to be 2% of the equivalent cell spherical radius. Whenever the distance of a cell surface mesh point, $\mathbf{x}$, to another cell surface is less than h_s, this point is moved in the $(\mathbf{x_p} - \mathbf{x})$ direction, where $\mathbf{x}_p$ is the projection of $\mathbf{x}$ on the other surface, until $|\mathbf{x} - \mathbf{x_p}| = h_s$. A similar approach is used for emulsion flow simulation [48]. For some of the results presented here, the time-averaged maximum correction is up to a displacement of $0.1h_s$ per time step. This h_s constraint is not activated in the smaller tubes since the cells do not come close. For the largest $D = 6$ case, doubling and halving
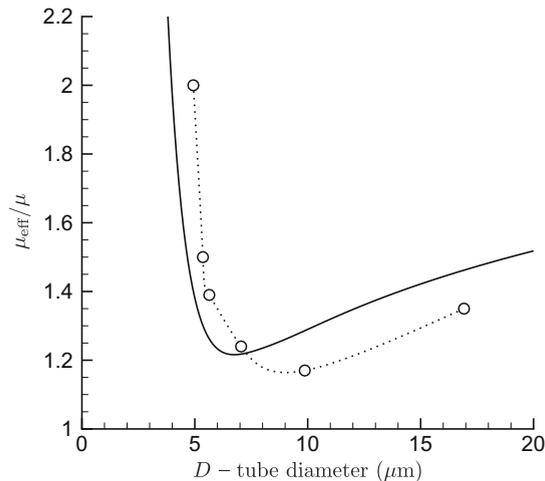
Fig. 13. Effective viscosity for simulations ∘ compared with empirical fits — of Pries et al. [36]. The data comprising this fit is scattered by approximately ±10%.

h_s produced less than a 2% change in the apparent viscosity. For all the cases simulated, no constraint was needed for close interactions between the cells and the vessel walls. The well-known near-wall cell-free layer is consistent with this.

The effective viscosities are calculated as described in the previous section, and they are compared in Fig. 13 with a standard empirical fit of experimental data [36]. A close agreement with this fit is not expected for two main reasons. The first involves a lack of complete parameterization of the flow conditions. The data leading to this fit is significantly scattered (greater than 20%), which suggests that the effective viscosity depends upon parameters other than just D and H_t. Our simulations match the criterion that the mean velocity divided by the tube diameter is greater than $50\,\mathrm{s}^{-1}$ as chosen by Pries et al. [36] for developing their fit, but it is also known that blood becomes strongly dependent upon shear rate for lower shear rates (*e.g.* the single-cell comparisons in Section 11.1), so strict independence is not to be expected for these moderate values. In addition, our constitutive model for the red-cell membranes is crude and is therefore not expected to provide a perfect match to actual red-cell behavior. It was selected for convenience for developing this flow solver, which itself is compatible with more general constitutive models. Nevertheless, it is clear in the figure that the simulations are indeed close in value to the experimental fit and show the same behavior with increasing D. There is a rapid drop in effective viscosity for small D, as the thickness of the near-wall cell-free layer increases, and then a slower increase for tubes of larger diameter than the red-cell dimension. This level of agreement suggests that the key physical mechanisms have been captured. The constitutive models can probably be improved or reparameterized to improve the apparent match, but it is not clear how to do this since many the experimental data used in developing the fit fall further from the fit than our simulation results [36]. A detailed study of the flow rheology dependence on tube diameter, tube hematocrit, and shear rates is underway.

12. Conclusions

We have presented a spectral boundary integral simulation method for Stokes flow and membrane interactions as encountered in blood flows in the microcirculation. The cell surface discretization is by a spherical harmonic expansion, which has excellent resolution and does not suffer from mesh point clustering near the two poles. However, the major strength of this spectral representation is that it enables rigorous removal of spurious high-wavenumber surface modes introduced by the nonlinearities in the flow–structure interaction system. We demonstrated excellent stability with this de-aliasing procedure. With it, the user can set the resolution so that results are mesh independent for whatever observable is of interest, and this can be set independently of any stability consideration because stability is due to de-aliasing which does not degrade the resolved solution. Periodic Green's functions are employed because they are relatively simple and fully compatible with the smooth particle-mesh Ewald algorithm used. The use of these Green's functions, however, is not limiting: complex geometry solid-wall boundaries are included within the periodic domain as constraints within the linear system that is solved as part of the time advancement algorithm. When cells approach closely, the integrands are considered to be nearly singular and a locally refined mesh is used to evaluate them.

None of the simulations presented were intensive by current standards. The largest, the 27-cell flow in the $D = 6.0$ vessel, required less than two days running on the eight processor cores of a single computer workstation. The largest part of the computational cost is from the numerical integration of the singular part of the Green's function. This cost, in turn, is mostly due to the interpolation of function values from the global cell surface mesh to the local polar coordinate patch.

Acknowledgment

References

[1] K.B. Abbitt, G.B. Nash, Rheological properties of the blood influencing selectin-mediated adhesion of flowing leukocytes, Am. J. Physiol. Heart Circ. Physiol. 285 (2003) H229.
[2] J.C. Admas, P.N. Swarztrauber, SPHEREPAKC 2.0: a model development facility, Tech. rep., NCAR, NCAR/TN-436-STR, 1997.
[3] P. Bagchi, Mesoscale simulation of blood flow in small vessels, Biophs. J. 92 (6) (2007) 1858–1877.
[4] P. Bagchi, P.C. Johnson, A.S. Popel, Computational fluid dynamic simulation of aggregation of deformable cells in a shear flow, J. Bio. Eng. 127 (2005) 1070–1080.
[5] O.P. Bruno, L.A. Kunyansky, A fast, high-order algorithm for the solution of surface scattering problems: Basic implementation, tests, and applications, J. Comput. Phys. 169 (2001) 80–110.
[6] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, Berlin, 1987.
[7] M. Dao, J. Li, S. Suresh, Molecularly based analysis of deformation of spectrin network and human erythrocyte, Mater. Sci. Eng. C 26 (2006) 1232–1244.
[8] M. Deserno, C. Holm, How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines, J. Chem. Phys. 109 (18) (1998) 7678–7693.
[9] M. Deserno, C. Holm, How to mesh up Ewald sums. II. An accurate error estimate for the particle–particle–particle–mesh algorithm, J. Chem. Phys. 109 (18) (1998) 7694–7701.
[10] L. Dintenfass, Internal viscosity of the red cell and a blood viscosity equation, Nature 219 (1968) 956–958.
[11] M.G. Duffy, Quadrature over a pyramid or cube of integrands with a singularity at a vertex, SIAM J. Numer. Anal. 19 (6) (1982) 1260–1262.
[12] R. Fåhræus, T. Lindqvist, The viscosity of blood in narrow capillary tubes, Am. J. Physiol. 96 (1931) 562–568.
[13] J.C. Firrel, H.H. Lipowsky, Leukocyte margination and deformation in mesnenteric venules of rate, Am. J. Physiol. Heart Circ. Physiol. 256 (1989) H1667.
[14] J.B. Freund, Leukocyte margination in a model microvessel, Phys. Fluids 19 (2007) 023301.
[15] H. Hashimoto, On the periodic fundamental solutions of the stokes equations and their application to viscous flow past a cubic array of spheres, J. Fluid Mech. 5 (1959) 317–328.
[16] F.K. Hebecker, Efficient boundary element methods for three-dimensional exterior viscous flows, Numer. Methods Part. Diff. Equations 2 (1986) 273–297.
[17] P.R. Johnston, D. Elliott, A sinh transformation for evaluating nearly singular boundary element integrals, Int. J. Numer. Methods Eng. 65 (2005) 564–678.
[18] S. Kim, S.J. Karrila, Microhydrodynamics: Principles and Selected Applications, Butterworth Heinemann, Stoneham, MA, 1991.
[19] A.G. Kravchenko, P. Moin, On the effect of numerical errors in large eddy simulations of turbulent flows, J. Comput. Phys. 131 (1997) 310–322.
[20] R. Kress, Linear Integral Equations, Springer-Verlag, 1989.
[21] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (3) (2003) 832–856.
[22] J. Li, M. Dao, C.T. Lim, S. Suresh, Spectrin-level modeling of the cytoskeleton and optical tweezers stretching of the erythrocyte, Biophys. J. 88 (2005) 3707–3719.
[23] W.K. Liu, Y. Liu, D. Farrell, L. Zhang, X.S. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, Immersed finite element method and its applications to biological systems, Comput. Methods Appl. Mech. Eng. 195 (2006) 1722–1749.
[24] R.M. MacMeccan, J.R. Clausen, G.P. Neitzel, C.K. Aidun, Simulating deformable particle suspensions using a coupled lattice-Boltzmann and finite-element method, J. Fluid Mech. 618 (2009) 13–39.
[25] V. Mityushev, P.M. Adler, Longitudinal permeability of spatially periodic rectangular arrays of circular cylinders. I. A single cylinder in the unit cell, Z. Angew. Math. Mech. 82 (5) (2002) 335–345.
[26] G.P. Muldowney, J.J.L. Higdon, A spectral boundary element approach to three-dimensional Stokes flow, J. Fluid Mech. 298 (1995) 167–192.
[27] S.A. Orszag, On the elimination of aliasing in finite difference schemes by filtering high-wavenumber components, J. Atoms. Sci. 28 (1971) 1074.
[28] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
[29] I.V. Pivkin, P.E. Karniadakis, Accurate coarse-grained modeling of red blood cells, Phys. Rev. Lett. 101 (2008) 118105.
[30] W. Pogorzelski, Integral Equations and Their Applications, Pergamon Press, 1966.
[31] H. Power, L.C. Wrobel, Boundary Integral Methods in Fluid Mechanics, Computational Mechanics Publications, 1994.
[32] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, Cambridge, 1992.
[33] C. Pozrikidis, Numerical simulation of the flow-induced deformation of red blood cells, Ann. Biomed. Eng. 31 (2003) 1194–1205.
[34] C. Pozrikidis, Axisymmetric motion of a file of red blood cells through capillaries, Phys. Fluids 17 (2005) 031503.
[35] A.R. Pries, K. Ley, M. Clasassen, P. Gaehtgens, Red cell distribution at microvascular bifurcations, Microvasc. Res. 38 (1989) 81–101.
[36] A.R. Pries, D. Neuhaus, P. Gaehtgens, Blood viscosity in tube flow: dependence on diameter and hematocrit, Am. J. Physiol. Heart Circ. Physiol. 263 (1992) H1770–H1778.
[37] J.M. Rallison, A. Acrivos, A numerical study of the deformation and burst of a viscous drop in an extensional flow, J. Fluid Mech. 89 (1978) 191–200.
[38] S.B. Rochal, V.L. Lorman, G. Mennessier, Viscoelastic dynamics of spherical composite vesicles, Phys. Rev. E 71 (2005) 021905.
[39] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (2) (1999) 509–534.
[40] Y. Saad, M. Schultz, A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1985) 856–869.
[41] D. Saintillan, E. Darve, E.S.G. Shaqfeh, A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: the sedimentation of fibers, Phys. Fluids 17 (2005) 033301.
[42] J.J.R. Silva, H. Power, L.C. Wrobel, A hypersingular integral equation formulation for Stokes' flow in ducts, Eng. Anal. Bound. Element 12 (1993) 185–193.
[43] R. Skalak, A. Tözeren, P.R. Zarda, S. Chien, Strain energy function of red blood cell membranes, Biophys. J. 13 (1973) 245.
[44] P.N. Swarztrauber, W.F. Spotz, Generalized discrete spherical harmonic transforms, J. Comput. Phys. 159 (2000) 213–230.
[45] H. Wang, T. Lei, J. Li, J. Huang, Z. Yao, A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations, Int. J. Numer. Methods Eng. 70 (2007) 812–839.
[46] L. Ying, G. Biros, D. Zorin, A high-order 3D boundary integral equation solver for elliptic PDEs in smooth domains, J. Comput. Phys. 219 (2006) 247–275.
[47] K. Yoshida, N. Nishimura, S. Kobayashi, Application of new fast multipole boundary integral equation method to crack problems in 3D, Eng. Anal. Bound. Elements 25 (2001) 239–247.
[48] A.Z. Zinchenko, R.H. Davis, Algorithm for direct numerical simulation of emulsion flow through a granular material, J. Comput. Phys. 227 (2008) 7841–7888.