FULLY IMPLICIT SPECTRAL BOUNDARY INTEGRAL COMPUTATION OF RED BLOOD CELL FLOW

Pei Chuan Chao  
Department of Civil, Structural and Environmental Engineering  
University at Buffalo  
212 Ketter Hall, Buffalo, NY 14260  
peichuan@buffalo.edu

Ali Gürbüz  
Department of Civil, Structural and Environmental Engineering  
University at Buffalo  
212 Ketter Hall, Buffalo, NY 14260  
aligurbu@buffalo.edu

Frederick Sachs  
Department of Physiology and Biophysics  
University at Buffalo  
304 Cary Hall, Buffalo, NY 14214  
sachs@buffalo.edu

M. V. Sivaselvan  
Department of Civil, Structural and Environmental Engineering  
University at Buffalo  
212 Ketter Hall, Buffalo, NY 14260  
mvs@buffalo.edu

April 22, 2021

Abstract

An approach is presented for implicit time integration in computations of red blood cell flow by a spectral boundary integral method. The flow of a red cell in ambient fluid is represented as a boundary integral equation (BIE), whose structure is that of an implicit ordinary differential equation (IODE). The cell configuration and velocity field are discretized with spherical harmonics. The IODE is integrated in time using a multi-step implicit method based on backward difference formulas, with variable order and adaptive time stepping controlled by local truncation error and convergence of Newton iterations. Jacobians of the IODE, required for Newton’s method, are implemented as Jacobian matrix-vector products that are nothing but directional derivatives. Their computation is facilitated by the weakly singular format of the BIE, and these matrix-vector products themselves amount to computing a second BIE. Numerical examples show that larger time steps are possible, and that the number of matrix-vector products is comparable to explicit methods.

Keywords  Weakly singular boundary integral · Fully implicit time integration · Adaptive time stepping · Spherical harmonics · Elastic thin-shell mechanics · Jacobian-vector multiplication

*Corresponding author
1 Introduction

Cell-level computational modeling of blood flow has been of great interest, because such simulations offer insight into the rheological character of blood in microcapillaries, aid in design of microfluidic devices for research and diagnosis, and have potential to guide targeted drug delivery and treatment [9,17,30]. Our interest in modeling red blood cell (RBC) flow stems from seeking to explore and test hypotheses about the role of the PIEZO1 mechanosensitive ion channel (MSC) in normal functioning of RBC; evidence of such a role is offered by clinical studies and by electrophysiological and microfluidic experiments [3,4,10,12,34]. This short note describes a byproduct of this exploration, an implicit time integration method for simulation of RBC flow. Investigating the role of PIEZOs involves simulation of RBC flow in microcapillaries of diameter less than the RBC size [19]. In this note, however, the simpler computational problem of deformation of a single RBC in ambient flow in an unbounded domain is considered, to present the formulation and implementation of the implicit method.

RBC flow at the cellular level is typically modeled using Stokes equation to represent the intra- and extracellular plasma, with the cell membrane described by elastic thin shell mechanics. Within this modeling framework, different mathematical formulations and corresponding numerical methods have been adopted (see [17] for a comprehensive review). Among these, a commonly employed approach is to formulate the Stokes flow of the intra- and extracellular plasma as boundary integral equations (BIE) [24–26]. Since RBC flow involves moving boundaries corresponding to the cell membranes, the BIE formulation, where different fields are computed only on boundaries, and not in the interiors, is beneficial. Further within the BIE setting, different schemes have been employed to discretize fields on the cell membrane, such as finite elements [5,14] and Fourier-like orthogonal spherical harmonic basis functions [33,35]. In this short note, we use a BIE formulation and spherical harmonic basis functions to discretize the RBC deformed geometry and the velocity field on the cell membrane. The cell membrane is nearly inextensible, and this is represented in elastic shell models either as a direct constraint [28,33], or by a penalty approach by means of a sufficiently large membrane dilatation modulus [16,35]. In this note, the latter approach is adopted.

The BIE for the cell membrane velocity, together with the membrane shell mechanics, constitutes an implicit ordinary differential equation (IODE) [2] in the membrane configuration (see equation (11))². Most commonly, this IODE is integrated in time using explicit schemes such as forward Euler [24,25,29,35,36]. Explicit schemes have been successfully used even in challenging scenarios, for example simulating red cells passing through very small capillaries and slits, and taking on complex configurations consisting of high buckled modes of deformation [16,35]. However, it has been demonstrated in [33] that the BIE IODE/DAE system is stiff due to inextensibility of the membrane and higher order shape derivatives that occur in computing bending deformation. Hence when explicit schemes are used, time step is restricted by stability rather than by accuracy. Furthermore, modeling the kinetics of the PIEZO1 MSC will introduce additional time scales [3,4]. Therefore, robustness with respect to time step size is desirable, and we seek to implement an implicit integration scheme.

Implicit methods for time integration of BIE arising in interface evolution problems such as RBC flow have been explored less in the literature. Dimitrakopoulos [13] presented an implicit method to compute the evolution of a three-dimensional droplet in ambient fluid of a different viscosity. The interface was modeled as having constant surface tension (unlike more general elastic shell mechanics in [16,28,33,35] and here). Special characteristics of the droplet mechanics and a perturbation approach were used to reduce the nonlinear algebraic equation in each time step to one in a scalar amplitude field of a prescribed vector field search direction. A single-step implicit Runge-Kutta method and a multi-step method based on backward difference formulas (BDF) were both considered for time discretization. Veerapaneni et al [33], building on their work on axisymmetric flows [32], proposed a semi-implicit scheme for simulation of three-dimensional vesicle flows. The fluids inside and the outside the vesicles had the same viscosity, and so the BIE contained only the single layer potential. Inextensibility of the membrane was modeled as a constraint, and the membrane mechanics was derived from a strain energy function of mean curvature. Backward Euler discretization was used, and the semi-implicit (or linearly implicit) scheme consisted of splitting the computation of the bending component of the membrane force in such a way that the new configuration can be computed by solving a linear equation. Rahimian et al [28] extended this work to account for different viscosities inside and outside the vesicle, so that the BIE also included a double layer potential. The BIE was treated using a Galerkin

---

²When a vessel is present as well in the model, it is an index-1 differential algebraic equation (DAE) [2], with the traction and velocity fields on the vessel wall being algebraic variables; if the vessel were also deformable, the BIE system would again be an IODE. This is the case when cell membrane inextensibility is modeled using a penalty approach; when inextensibility is imposed as a constraint as in [28,33], then the BIE is a DAE.
approach (as also done in this note). The implicit-explicit split was again applied to the bending force, to obtain a linearly implicit scheme. A hierarchy of iterative methods were considered for linear system solution, and their performances compared. Despite reduced overall computational cost compared to the explicit scheme, large numbers of iterations per time step were needed. Quaife and Biros [27] presented an adaptive semi-implicit method for two-dimensional vesicle suspensions. By adopting a spectral deferred correction method, the simulation was allowed to automatically select time step size and method accuracy up to third order.

In this note, an implicit method is presented that is quite simple to implement. The flow is formulated as a BIE and the membrane mechanics using elastic thin shell theory, with in-plane inextensibility of the membrane captured by a penalty approach. Spherical harmonic basis functions are used to discretize different fields on the cell membrane, and as in [35], the package SPHEREPACK [1] is used to perform computations related to spherical harmonics. The BIE, including the stress kernel, is set up in a *weakly singular* format following [22], facilitating its linearization. The IODE structure of the BIE is deliberately recognized, and an off-the-shelf IODE/DAE solver [21] is used. The solver is passed two problem-specific routines — one that evaluates the BIE producing a residual, and another that computes Jacobian matrix-vector products. Computing a Jacobian matrix-vector product amounts to evaluating another BIE (see equation (13)), and the matrix itself is not constructed. In forming the Jacobian matrix-vector product, both the BIE kernels and the membrane force are linearized. The IODE solver uses BDF for time discretization, and adapts time step as well as BDF order based on truncation error. The solver also balances truncation error and accuracy of solution of the nonlinear algebraic equations in each time step, resulting in a comparable number of Jacobian matrix-vector products per time step as needed in an iterative solution of the linear system arising in implicit methods.

This short note is organized as follows. The BIE for RBC flow is presented in Section 2 in weakly singular form. Section 3 summarizes key results from shell mechanics. The IODE structure of the BIE is discussed in Section 4, which concludes by stating the two functions — BIE evaluation and Jacobian matrix-vector product computation, to be provided to the IODE solver. The BIE directional derivatives needed for implicit solution are presented in Section 5. Numerical examples are presented in Section 6, highlighting various characteristics of the implicit solution. Finally some key observations and potential for further development of this implicit scheme implementation are summarized in Section 7.

## 2 Boundary integral equation (BIE)

With the terminology and notation defined in Figure 1, the velocity field, \(\bar{u}\), on the surface, \(\gamma\), of a red cell in a three-dimensional ambient flow, \(\bar{u}^{\infty}\), satisfies the following BIE

\[
\frac{\lambda + 1}{2} \bar{u}(\chi) + \frac{\lambda - 1}{8\pi} \int_K K(\chi, \xi) \bar{u}(\xi) \, d\gamma = \bar{u}^{\infty}(\chi) - \frac{1}{8\pi \mu} \int_\gamma G(\chi, \xi) \bar{f}(\xi) \, d\gamma \quad \forall \chi \in \gamma
\]

where \(\mu\) is the viscosity of the extracellular plasma, \(\lambda\mu\) that of the intracellular plasma, \(\bar{f}\) is the net force on the membrane applied by the fluids on either side of it per unit area of the surface \(\gamma\), and the kernels,

\[
G(\chi, \xi) = \frac{1}{\|r\|}(I + r\tilde{r}^\top); \quad K(\chi, \xi) = \frac{6}{\|r\|^2} (\tilde{r}^\top n)(\tilde{r}^\top n)
\]

are fundamental solutions for Stokes flow, with \(r = \chi - \xi\) and the unit vector, \(\tilde{r} = \frac{r}{\|r\|}\), \(n\) being the unit normal vector field on \(\gamma\) pointing outward to ambient fluid. Equation (1) is the same as equation (11) in [24] (see also equation (2) on page 143 of [26]), equation (3a) in [28], and equation (6) in [35], with appropriate mapping of notation.

Since the kernel \(G\) contains \(\|r\|\) in the denominator, the integral on the right hand side of equation (1) is weakly singular, whereas the integral on the left side is strongly singular because the kernel \(K\) contains \(\|r\|^2\) in the denominator [26]. However, a weakly singular representation of the integral on the left may be obtained by *singularity subtraction* [22]: subtracting and adding \(\left[ \int_\gamma K(\chi, \xi) \, d\gamma \right] \bar{u}(\chi)\), and using the identity, \(\int_\gamma K(\chi, \xi) \, d\gamma = -4\pi I\) [22,26], we obtain

\[
\bar{u}(\chi) + \frac{\lambda - 1}{8\pi} \int_K K(\chi, \xi)(\bar{u}(\xi) - \bar{u}(\chi)) \, d\gamma = \bar{u}^{\infty}(\chi) - \frac{1}{8\pi \mu} \int_\gamma G(\chi, \xi) \bar{f}(\xi) \, d\gamma \quad \forall \chi \in \gamma
\]

where now, the integral on the left is also weakly singular, and \((\bar{u}(\chi) - \bar{u}(\xi))\) may be thought of as “cancelling” one of the \(\|r\|s\) in the denominator of \(K\) as \(\chi \to \xi\). This form allows using the method of [18], involving
rotation of the spherical coordinate system to align the north pole with the source point, $\chi$, the eigenfunction property and the additional theorem of spherical harmonics, to compute both integrals (see [19] for further details). Furthermore, when the BIE is linearized in equation (13) below, the resulting integrals remain weakly singular, and the method of [18] can be used to compute them as well.

Introducing spherical coordinates, we have

\[
\hat{u}(\varphi_{\chi}) + \frac{\lambda - 1}{8\pi} \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} K \left( \hat{\xi}(\varphi_{\chi}), \hat{\xi}(\varphi) \right) (\hat{u}(\varphi) - \hat{u}(\varphi_{\chi})) J_{\hat{\xi}}(\varphi) \, d\theta \, d\phi
\]

\[
= \hat{u}^{\infty}(\varphi_{\chi}) - \frac{1}{8\pi\mu} \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} G \left( \hat{\xi}(\varphi_{\chi}), \hat{\xi}(\varphi) \right) \hat{f}(\varphi) \sin \theta \, d\theta \, d\phi
\]

\[
\forall \varphi_{\chi} \in [0, \pi] \times [0, 2\pi]
\]

where $\varphi_{\chi}$ denotes the spherical coordinates that map to the source point, $\chi$. Note that the force on the membrane in equation (4) is expressed per unit area of the unit sphere parent geometry, $\hat{f}$, since this can be computed readily by equation (10) below. Moreover, since the Jacobian $J_{\hat{\xi}}$ does not appear in the right hand side integral of equation (4), the linearization in equation (13) below has a simpler form.
3 Summary of elastic thin-shell mechanics results

This section summarizes the computation of the membrane force. The formulas are completely equivalent to standard ones in the literature [15, 23, 35]. However, they are arranged so that the force can be computed using only the operations of the gradient of scalar functions and the divergence of vector fields on the unit sphere, without invoking Christoffel symbols and derivatives of higher-order tensor fields. The gradient and divergence are computed in spherical coordinates as

\[
\nabla \Box = \left( \frac{\partial \Box}{\partial \theta}, \frac{1}{\sin \theta} \frac{\partial \Box}{\partial \phi} \right)^T; \quad \nabla \cdot \Box = \frac{1}{\sin \theta} \left( \frac{\partial}{\partial \theta} (\Box \sin \theta) + \frac{\partial}{\partial \phi} \Box \right)
\]

(5)

directly using the \texttt{gradgc} and \texttt{divgc} functions in SPHEREPACK. Note that the Jacobians in Figure 1 and normal vector fields to the reference and deformed surfaces can also be computed using the gradient operator,

\[
J_{\hat{\Xi}} = \left\| \nabla_{\theta} \hat{\Xi} \times \nabla_{\phi} \hat{\Xi} \right\|; \quad \hat{N}(\varphi) = \frac{1}{J_{\hat{\Xi}}} \left( \nabla_{\theta} \hat{\Xi} \times \nabla_{\phi} \hat{\Xi} \right)
\]

(6)

The force per unit area of \(S^2\) is computed by the following steps (details can be found in [8]).

1. **Kinematics**: The strain, \(\epsilon\), and curvature change, \(\kappa\), are computed as

\[
\epsilon = \frac{1}{2} A^{-1} (a - A); \quad \kappa = A^{-1} (b - B)
\]

(7)

where \(A = \nabla \hat{\Xi}^T \nabla \hat{\Xi}\) and \(a = \nabla \hat{\xi}^T \nabla \hat{\xi}\) are matrices closely related to the metric tensor in the reference and deformed configurations; similarly, \(B = \nabla \hat{\Xi}^T \nabla \hat{N}\) and \(b = \nabla \hat{\xi}^T \nabla \hat{n}\) are related to the respective curvatures.

2. **Constitutive equations**: The membrane in-plane force and moment matrices, \(n\) and \(m\), are then obtained using constitutive equations,

\[
n = E_S I + \left( \frac{E_D}{2} \log(\det(C)) - E_S \right) C^{-T}
\]

\[
m = E_B \kappa^T
\]

(8)

where \(C = 2\epsilon + I\), and following the notation of [25, 35], \(E_D\) and \(E_S\) are in-plane dilatational and shear moduli, and \(E_B\) is a bending modulus. A neo-Hookean model is used for the in-plane forces, and a linear model for the moment.

3. **Equilibrium**: The membrane out-of-plane shear, \(\tilde{n} q\) is computed as

\[
\tilde{n} q = \frac{1}{J_{\hat{\Xi}}} \left[ \nabla_{\phi} \hat{\xi} \times \mathcal{P} \tau \quad | \mathcal{P} \tau \times \nabla_{\theta} \hat{\xi} \right]
\]

(9)

where the normal projection, \(\mathcal{P} = I - \tilde{n} \tilde{n}^T\), and \(\tau = -\nabla \cdot \left( \nabla \hat{\xi} A^{-1} m J_{\hat{\Xi}} \right)\). Finally, the membrane force is calculated using

\[
\hat{f}(\varphi) = -\nabla \cdot \left( \nabla \hat{\xi} A^{-1} n + \nabla \tilde{n} A^{-1} m + \tilde{n} q \right) J_{\hat{\Xi}}
\]

(10)

4 Galerkin approach and implicit ordinary differential equation (IODE) structure

The residual of equation (4), which we call \(\text{res}(\hat{\xi}, \hat{u}; \chi)\), may be computed for any source point, \(\chi\). The BIE is satisfied if this residual is zero for all source points. Instead of enforcing the BIE at selected source points, the residual is computed using all the sample points as source points, and the spherical harmonic transform of the residual is set to zero. This amounts to a Galerkin approach with the test functions being the spherical harmonic basis functions (see [19] for details). The process is depicted in Figure 2, and may be thought of as the equation,

\[
g(\hat{\xi}, \hat{u}, t) = 0
\]

(11)
SH coefficients of $\xi$, $\dot{\xi}$

Inverse SH transform

Values of $\dot{\xi}$, $\dot{u}$
at sample points

Compute residual

Values of $\text{res}(\dot{\xi}, \dot{u}; \varphi_{\chi})$
at sample points

SH transform

SH coefficients of residual

---

Figure 2: Implicit ODE structure of BIE

---

Figure 3: Structure of function to compute directional derivatives (Jacobian matrix-vector products) of the BIE IODE

Since the velocity field, $\dot{u} = \dot{\xi}$, is the rate of change of configuration, this is an implicit ordinary differential equation (IODE).

Numerical solutions to such IODE could be computed by means of a number of solvers such as ode15s in the MATLAB suite [31], DASSL [6], DASPK [7] and IDAS in the SUNDIALS suite [21] that use multi-step methods based on backward difference formulas (BDF), and RADAU5 [20] based on an implicit Runge-Kutta method. All such solvers require the derivatives, or Jacobians, $D_{1}g$ and $D_{2}g$, to use in different variations of Newton’s method at each time-step. Computing these Jacobian matrices is cumbersome and computationally expensive. However, as outlined below in Section 5, the directional derivatives (or matrix-vector products), $D_{1}g(\dot{\xi}, \dot{u}, t) \cdot \delta \xi$ and $D_{2}g(\dot{\xi}, \dot{u}, t) \cdot \delta \dot{u}$ along $\delta \xi$ and $\delta \dot{u}$ can be readily computed. The solvers DASPK and IDAS allow a function that returns these matrix-vector products. We use IDAS with its MATLAB interface. This solver requires two functions representing the IODE,

1. A function that given the state $(\dot{\xi}, \dot{u})$ and time, computes the residual $g(\dot{\xi}, \dot{u}, t)$; this is implemented as shown in Figure 2.

2. A function that given the state, $(\dot{\xi}, \dot{u})$, time, directions $(\delta \xi, \delta \dot{u})$, and a parameter, $\Lambda$, computes the sum of the directional derivatives, $D_{1}g(\dot{\xi}, \dot{u}, t) \cdot \delta \xi + \Lambda D_{2}g(\dot{\xi}, \dot{u}, t) \cdot \delta \dot{u}$. This is discussed in the next section.

5 Directional derivatives

This section is on the computation of the directional derivatives of the IODE residual needed for implicit numerical solution. The directional derivatives, $D_{1}\text{res}(\dot{\xi}, \dot{u}; \varphi_{\chi}) \cdot \delta \xi$ and $D_{2}\text{res}(\dot{\xi}, \dot{u}; \varphi_{\chi}) \cdot \delta \dot{u}$ residual of equation (11) are calculated, and required derivative of the IODE is obtained after a transformation as shown in Figure 3.

The second of these directional derivatives is simple, since the BIE (4) is linear in the velocity field, and can be computed in the same manner as the BIE itself.

$$D_{2}\text{res}(\dot{\xi}, \dot{u}; \varphi_{\chi}) \cdot \delta \dot{u} = \delta \dot{u}(\varphi_{\chi})$$

$$\quad + \frac{\lambda - 1}{8\pi} \int_{0}^{2\pi} \int_{0}^{\pi} K \left( \xi(\varphi_{\chi}), \xi(\varphi) \right) (\delta \dot{u}(\varphi) - \delta \dot{u}(\varphi_{\chi})) J_{\xi}(\varphi) \sin \theta d\theta d\phi$$

(12)

The derivative in a direction of configuration change requires more careful consideration.

$$D_{1}\text{res}(\dot{\xi}, \dot{u}; \varphi_{\chi}) \cdot \delta \xi = \frac{\lambda - 1}{8\pi} \int_{0}^{2\pi} \int_{0}^{\pi} \left[ \delta \mathcal{K} \Delta \dot{u} J_{\xi} + \mathcal{K} \Delta \dot{u} \delta J_{\xi} \right] \sin \theta d\theta d\phi$$

$$\quad - \delta \dot{u}^{\infty}(\varphi_{\chi}) + \frac{1}{8\pi \mu} \int_{0}^{2\pi} \int_{0}^{\pi} \left[ \delta G f_{\xi} + G f_{\xi} + \delta G f_{\xi} \right] \sin \theta d\theta d\phi$$

(13)
where we have used the abbreviations, $\Delta \hat{u} = \hat{u}(\varphi) - \hat{u}(\varphi, \chi)$, $\mathcal{K} = K(\hat{\varphi}(\varphi, \chi), \hat{\chi}(\varphi))$ and $\mathcal{G} = G(\hat{\varphi}(\varphi, \chi), \hat{\chi}(\varphi))$, and $\delta \square$ to represent the directional derivative, $D \square(\hat{\chi}) \cdot \delta \hat{\chi}$. In the following, we define $\nu = \nabla \varphi \times \nabla \hat{\chi}$. The directional derivatives in equation (13), obtained by straightforward but tedious calculation, are given by

$$
\delta r = \delta \hat{\chi}(\varphi, \chi) - \delta \hat{\chi}(\varphi); \quad \delta \hat{r} = \frac{1}{\|r\|}(I - \hat{r} \hat{r}^{\top})\delta r \\
\delta \nu = \nabla \varphi \delta \hat{\chi}(\varphi) \times \nabla \hat{\chi}(\varphi) \times \nabla \varphi \delta \hat{\chi}(\varphi) \\
\delta J_{\hat{\chi}} = \hat{n}^{\top} \delta \nu; \quad \delta \hat{n} = \frac{1}{J_{\hat{\chi}}} \mathcal{P} \delta \nu; \quad \delta \mathcal{G}_{\hat{\chi}} = \frac{1}{\|r\|} \left[ -(\hat{r}^{\top} \delta r)\mathcal{G}_{\hat{\chi}} + (\hat{r}^{\top} \hat{\nu})\delta \hat{r} + (\delta \hat{r}^{\top} \hat{f})\hat{r} \right] \\
\delta \mathcal{K} \Delta \hat{u} = \frac{6}{\|r\|^2} \left[ (\hat{r}^{\top} \hat{n})(\hat{r}^{\top} \Delta \hat{u}) \left( \delta \hat{r} - \frac{2}{\|r\|}(\hat{r}^{\top} \delta r)\hat{r} \right) \\
+ \left( (\hat{r}^{\top} \Delta \hat{u})(\delta \hat{r}^{\top} \hat{n} + \hat{r}^{\top} \delta \hat{n}) + (\hat{r}^{\top} \hat{n})(\delta \hat{r}^{\top} \Delta \hat{u}) \right) \hat{r} \right] 
$$

The directional derivative of the membrane force is obtained by linearizing the shell mechanics relations from Section 3.

$$
\delta \epsilon = \frac{1}{2} A^{-1} \left( \nabla \delta \hat{\chi}^{\top} \nabla \hat{\chi} + \nabla \hat{\chi}^{\top} \nabla \delta \hat{\chi} \right) \\
\delta C = 2\delta \epsilon \\
\delta \kappa = A^{-1} \left( \nabla \delta \hat{\chi}^{\top} \nabla \hat{\chi} + \nabla \hat{\chi}^{\top} \nabla \delta \hat{\chi} \right) \\
\delta \hat{n} = \left[ \left( \frac{E_{B}}{2} \text{tr}(C^{-1}\delta C) \right) - \left( \frac{E_{B}}{2} \log(\text{det}(C)) - E_{S} \right) \delta C \right] C^{-\top} \\
\delta \hat{m} = E_{B} \delta \kappa^{\top} \\
\delta t = -\nabla \cdot \left[ \left( \nabla \delta \hat{\chi} A^{-1} \hat{m} + \nabla \hat{\chi} A^{-1} \delta \hat{m} \right) J_{\Xi} \right] \\
\delta (\mathcal{P} t) = \mathcal{P} \delta t - (\hat{n}^{\top} \hat{r}) \delta \hat{n} - (\delta \hat{n}^{\top} \hat{r}) \hat{n} \\
\delta (\mathcal{P} \hat{n}) = -\frac{\delta J_{\hat{\chi}}}{J_{\hat{\chi}}} \delta \hat{n} + \frac{1}{J_{\hat{\chi}}} \left[ \nabla \nabla \delta \hat{\chi} \times \mathcal{P} t + \nabla \nabla \hat{\chi} \times \delta (\mathcal{P} t) \right] \\
\delta \hat{f}(\varphi) = -\nabla \cdot \left[ \left( \nabla \delta \hat{\chi} A^{-1} \hat{n} + \nabla \hat{\chi} A^{-1} \delta \hat{n} + \nabla \delta \hat{n} A^{-1} \hat{m} + \nabla \hat{n} A^{-1} \delta \hat{m} + \delta (\hat{n} \hat{q}) \right) J_{\Xi} \right] 
$$

Equation (13) is itself a BIE with only weakly singular terms, and the integrals can again be computed using the method of [18]. The weakly singular representation of BIE (14) facilitates the linearization in this convenient form.

## 6 Numerical examples

Four numerical examples are presented to demonstrate the performance of the proposed fully implicit method. Various aspects of performance such as (i) time steps used to achieve the specified tolerances, (ii) order of backward difference approximation used, (iii) number of function and derivative calls needed, (iv) conservation of different quantities, and (v) comparison with the explicit time scheme, as well as the dependence of these on spatial discretization are illustrated. In each example, the motion of a red blood cell suspended in an ambient flow is considered. The four examples are

(a) Shear flow ($k = 100s^{-1}$): An ambient shear flow parallel to the $x$ direction with a shear strain rate, $k = 100s^{-1}$ (as in [24,33]), so that ambient velocity field, $u = (kz, 0, 0)^{\top}$, with an undeformed red cell placed initially making an angle of 45° with the $x$ axis. Capillary number, $Ca = \frac{k_{a}s}{E_{r}} = 0.08$.

(b) Shear flow ($k = 500s^{-1}$): Same as (a), but with a greater shear rate, $k = 500s^{-1}$ (as in [24,33]), $Ca = 0.4$. 
Table 1: Steady state time-step sizes used by the explicit and implicit time integration schemes for different flow types and spherical harmonic degrees

| Flow type | (a) | (b) | (c) | (a) | (b) | (c) | (d) |
|-----------|-----|-----|-----|-----|-----|-----|-----|
| N = 8     | 1.13E-02 | 1.16E-02 | 1.13E-02 | 7.34E-02 | 1.60E-02 | 1.22E-01 | 5.14E-02 |
| 16        | 5.07E-03 | 5.63E-03 | 5.38E-03 | 5.10E-02 | 2.26E-02 | 1.17E-01 | 6.55E-02 |
| 24        | 3.07E-03 | 3.07E-03 | 3.46E-03 | 6.60E-02 | 2.45E-02 | 8.93E-02 | 5.90E-02 |

(c) Parabolic flow: A parabolic ambient flow along the x direction, $u_x = A(B - (y^2 + z^2))$, where $A = 1$ (35.46 $\mu$m$^{-1}$s$^{-1}$) and $B = 2.3$ (18.3 $\mu$m$^2$) are parameters that control the curvature and peak velocity of the flow, $C_a = \frac{\mu A a^4}{E_B} = 14.95$.

(d) Relaxation: A highly deformed cell is suspended in ambient fluid at rest until it relaxes to its undeformed shape. A deformed shape from (b) is taken as the initial configuration.

Following common practice in the literature [24, 25, 35], non-dimensional parameters are used. The equivalent cell radius, $a \approx 2.82 \mu$m is used as characteristic length, the inverse of a shear rate $k = 100$ s$^{-1}$ as characteristic time, and plasma viscosity times this shear rate $\mu k = 0.12$ Pa as characteristic stress, where $\mu = 1.2 \times 10^{-3}$ Pa $\cdot$ s. With this nondimensionalization, the cell membrane shear modulus, $E_S = 12.4$ (4.2 $\times 10^{-6}$ N/m) and bending modulus, $E_B = 0.0669$ (1.8 $\times 10^{-19}$ N $\cdot$ m). The membrane dilatational modulus is set as a penalty parameter $E_D = 200$ (6.8 $\times 10^{-5}$ N/m) to control membrane area change while preserving numerical conditioning. Viscosity ratio of the intracellular plasma to extracellular plasma, $\lambda = 5$. The ability of the membrane to deform under viscous stress is gauged by capillary number, $C_a = \frac{\mu k a}{E_B}$, based on shear modulus for case (a) and (b), and $C_a = \frac{\mu A a^4}{E_B}$, based on bending modulus for case (c) [16]. The initial undeformed shape of the cell is a biconcave disk, which following [24, 25] is described by

$$\hat{\xi}(\varphi) = \left(\alpha \sin \theta \cos \phi; \alpha \sin \theta \sin \phi; \frac{\alpha}{2} (0.207 + 2.003 \sin^2 \theta - 1.123 \sin^4 \theta) \cos \theta \right)^\top$$

where $\alpha = 1.386$ is the non-dimensional maximum radius of the biconcave disk.

The relative tolerance for truncation error is set to $10^{-4}$ and absolute tolerance to $10^{-6}$ for the IDAS solver for all cases. A maximum time step restriction is set for each example to minimize failed steps.

Figure 4 shows deformation snapshots of the evolving cell for the different flow types with spherical harmonic degree $N = 24$. The color indicates the norm of the principal membrane stress. Figure 5 presents the profile snapshots of the cell cross section in the $xz$ plane corresponding to the deformed shapes in Figure 4. The behaviors seen in these figures are well documented in the literature [11, 24]. From Figures 4(a, b) and 5(a, b), it can be seen that the cell rotates clockwise, and is stretched or compressed depending on the angle of inclination. Compared to case (a), the cell in case (b) has larger deformation due to higher shear rate, and the period of rotation is much higher as well. A star-shape marker serves as tracker of the cell rotating. The reversed S-shape of the cell mid-plane profile shown in Figure 5(b) is also reported by Pozrikidis [24]. In Figures 4(c) and 5(c), the cell is stretched at the middle as it translates in the $x$ direction under parabolic flow. Due to the initial inclination, the upper half of the cell moves faster and drags the lower half forward. In the relaxation case, the cell recovers from the highly deformed shape to reach the stress-free state in ambient fluid, as shown in Figure 4(d) and 5(d).

In the following, various numerical performance aspects of the implicit time integration scheme are discussed:

(i) **Time step size**: The time-step sizes used by the implicit scheme and the explicit scheme for different spatial resolutions and flow types are reported in Table 1. Time-step size for implicit method is nearly independent of the spherical harmonic degree, and only depends on the flow type and capillary number. A similar observation was made in [33] for simulation of a vesicle in simple shear flow with a semi-implicit scheme.

IDAS solver statistics are provided for the shear flow case (a) in Figure 6 and for the parabolic flow case (c) in Figure 7. In each figure, solid lines show data for $N = 24$, and dotted line for $N = 8$. The data with explicit method for $N = 8$ is also reported as dash-dot line in both first and last subplots for comparison. The first subplot shows evolution of time step size during the simulation. The solver adapts the time step based on local truncation error criteria and on convergence of Newton iterations.
| Flow type | Deformation snapshots |
|-----------|-----------------------|
| (a) Shear flow $k = 100 \text{ s}^{-1}$ | ![Deformation snapshots](image) |
| (b) Shear flow $k = 500 \text{ s}^{-1}$ | ![Deformation snapshots](image) |
| (c) Parabolic flow | ![Deformation snapshots](image) |
| (d) Relaxation | ![Deformation snapshots](image) |

Figure 4: Evolution of cell deformation ($N=24$). Colors indicate the norm of the principal membrane stress.

| Flow type | Profile snapshots |
|-----------|-------------------|
| (a) Shear flow $k = 100 \text{ s}^{-1}$ | ![Profile snapshots](image) |
| (b) Shear flow $k = 500 \text{ s}^{-1}$ | ![Profile snapshots](image) |
| (c) Parabolic flow | ![Profile snapshots](image) |
| (d) Relaxation | ![Profile snapshots](image) |

Figure 5: Cell mid-plane profiles corresponding to Figure 4
Figure 6: IDAS solver statistics for example (a) for N = 24 (solid line), N = 8 (dotted line) and with explicit method for N = 8 (dash-dot line): The first subplot shows evolution of time-step size with simulation time — after adapting the time step, the solver settles at a time step of \(6.60 \times 10^{-2}\) for \(N = 24\), the second subplot shows the order of backward difference approximation, the third and fourth subplots show numbers of residual and Jacobian-vector product evaluations for implicit method and matrix-vector product for explicit method. The first and fourth subplots also show corresponding data for explicit integration.

For each example, we set a maximum step size to minimize the number of failed steps. The solver settles at a time step of \(6.60 \times 10^{-2}\) in case (a), and \(8.93 \times 10^{-2}\) in case (c), more than ten times what is possible with an explicit method based on stability considerations alone.

(ii) **Order of backward difference approximation:** The second subplot in each of Figures 6 and Figures 7 shows how the order of backward difference approximation employed by the solver. The order is adapted between 1 and 5, again based on local truncation error test and convergence. The order of approximation is at least 3, and most often is the highest possible order 5, indicating the smoothness of the solution.

(iii) **Number of function and derivative calls needed:** The third and fourth subplots in Figures 6 and Figures 7 report the numbers of residual function evaluations (Figure 2) and Jacobian matrix-vector multiplications (Figure 3). Approximately ten Jacobian multiplications are needed for every residual evaluation. Since fewer time steps are used, the total number of matrix-vector products is comparable to the number of GMRES iterations needed in an explicit method. The number of matrix-vector products can potentially be reduced by suitably preconditioning the system.

(iv) **Conservation of different quantities:** Besides local truncation error, as other measures of accuracy, we consider conservation of membrane surface area and cell volume, and equilibrium of
Figure 7: IDAS solver statistics for example (c) for $N = 24$ (solid line), $N = 8$ (dotted line) and with explicit method for $N = 8$ (dash-dot line): The first subplot shows evolution of time-step size with simulation time — after adapting the time step, the solver settles at a time step of $8.93 \times 10^{-2}$ for $N = 24$, the second subplot shows the order of backward difference approximation, the third and fourth subplots show numbers of residual and Jacobian-vector product evaluations for implicit method and matrix-vector product for explicit method. The first and fourth subplots also show corresponding data for explicit integration.

membrane forces. Figure 8 shows membrane surface area change and volume conservation error for the shear case (a). Volume conservation error decreases dramatically with increasing spherical harmonic degree due to the accuracy of representation of the BIE itself. The area changes is within 1%, indicating the extent to which the penalty parameter, $E_D$, is effective in enforcing membrane inextensibility. The largest area changes occur at the times when cell is stretched most by the shear flow when the angle of inclination of the cell is ±45°. Figure 9 demonstrates equilibrium of forces and moments on the entire cell.

(v) **Comparison with the explicit integration** Due to the variabilities in how time step size can be adapted, tolerances for nonlinear and linear solvers, truncation error control etc., it is not possible to make an exact quantitative comparison between the performances of implicit and explicit time integration schemes. Qualitatively, however, it is clear that steady state time-step size used by the implicit time integration scheme depends on the imposed flow type and is not affected by the spherical harmonics order (see Table 1). On the other hand, the steady time-step size used by the explicit method decreases with increasing spherical harmonic degree and, interestingly, is not affected by the flow strength and type. The time step dictated by stability in an explicit method is approximately 1/10th of the time step size used by IDAS for truncation error control. Comparison of computational effort between the IDAS and an explicit method is made using the number of matrix-
vector multiplications (although a Jacobian matrix-vector multiplication is more expensive than the residual matrix-vector multiplication). The number of matrix-vector multiplication performed by IDAS is approximately 1/7th of the number of GMRES iterations in an explicit method as seen in the fourth subplots of Figures 6 and 7.

7 Concluding remarks

An implicit time integration approach has been presented in connection with a spectral boundary integral method for computation of red blood cell flow. The simple problem of cell flow in an ambient fluid (rather than in a vessel) has been considered to describe of the method. The flow is represented by a boundary integral equation (BIE) expressed in weakly singular form. Cell configurations are represented by smooth deformations of the unit sphere. The cell configuration and velocity field are discretized using spherical harmonic basis functions. The cell membrane is modeled using elastic thin-shell mechanics, formulas from which are written in a manner that requires only the operations of gradient of scalar functions and divergence of vector fields on the unit sphere in spherical coordinates. This BIE representing the flow has the form of an implicit ordinary differential equation (IODE) in the cell configuration, and solutions are computed using an off-the-shelf IODE solver based on backward differences formulas. The solver adapts the time step and order of backward difference approximation to control local truncation error and to regulate convergence of Newton’s method. Newton’s method employed in implicit time integration necessitates computation of Jacobians of the IODE with respect to the configuration and the velocity field. Rather than assemble such Jacobians, Jacobian-vector products, i.e. directional derivatives are calculated. This calculation, which itself is the evaluation of a different BIE, is aided by the expression of the flow BIE in weakly singular format.
Numerical examples have been presented to demonstrate the working of the implicit approach. They illustrate that larger time steps can be used compared to explicit methods, where time-step size is limited by stability considerations. The number of Jacobian-vector products is comparable to the number of GMRES iterations of the flow BIE in explicit methods. Source code for the implementation of the implicit method is included on the supplemental material that accompanies this note.

Much further development is needed to employ this implicit method in broader problems of interest. Suitable preconditions must be constructed for the directional derivative BIEs to reduce the number of Jacobian-vector products further, and hence computational effort. Fast summation methods must also be implemented for the derivative BIEs. Most importantly, the method must be extended for the presence of other geometries such as vessels and slits. While these do not alter the mathematical formulation significantly, they raise the possibility of interpenetration when large time steps are used. In such cases, interpenetration must be flagged to the solver. Closest-point calculations used in evaluation of near-singular integrals can potentially be adapted to detect interpenetration. These are topics of current inquiry.

References

[1] John C Adams and Paul N Swarztrauber. SPHEREPACK 3.0: A model development facility. Monthly Weather Review, 127(8):1872–1878, 1999.
[2] Uri M Ascher and Linda R Petzold. Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. SIAM: Society for Industrial and Applied Mathematics, 1998.
[3] C. Bae, F. Sachs, and P. A. Gottlieb. The mechanosensitive ion channel PIEZO1 is inhibited by the peptide GsMTx4. Biochemistry, 50(29):6295–300, 2011.
[4] Chilman Bae, Radhakrishnan Gnanasambandam, Chris Nicolai, Frederick Sachs, and Philip A. Gottlieb. Xerocytosis is caused by mutations that alter the kinetics of the mechanosensitive channel PIEZO1. Proceedings of the National Academy of Sciences, 110(12):E1162–E1168, 2013.
[5] G. Boedec, M. Leonetti, and M. Jaeger. 3d vesicle dynamics simulations with a linearly triangulated surface. Journal of Computational Physics, 230(4):1020–1034, 2011.
[6] Kathryn E. Brenan, Stephen L. Campbell, and Linda R. Petzold. Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. SIAM, Philadelphia, 1996.
[7] Peter N. Brown, Alan C. Hindmarsh, and Linda R. Petzold. Using Krylov methods in the solution of large-scale differential-algebraic systems. SIAM Journal of Scientific Computing, 15(6):1467–1488, 1994.
[8] Pei-Chuan Chao. Spectral and implicit methods for computational cell mechanics. PhD thesis, University at Buffalo, 2020.
[9] Ivan Cimr`ak and Ivet`a Janˇ cigova. Computational Blood Cell Mechanics: Road Towards Models and Biomedical Applications. CRC Press, 2018.
[10] E. Cinar, S. Zhou, J. DeCourcey, Y. Wang, R. E. Waugh, and J. Wan. Piezo1 regulates mechanotransductive release of ATP from human RBCs. Proceedings of the National Academy of Sciences, 112(38):11783–11788, 2015.
[11] Daniel Cordasco and Prosenjit Bagchi. On the shape memory of red blood cells. Physics of Fluids, 29(4):041901, 2017.
[12] Jens G. Danielczok, Emmanuel Terriac, Laura Hertz, Polina Petkova-Kirova, Franziska Lautenschläger, Matthias W. Laschke, and Lars Kaestner. Red blood cell passage of small capillaries is associated with transient Ca2+-mediated adaptations. Frontiers in Physiology, 8:979, 2017.
[13] Panagiotis Dimitrakopoulos. Interfacial dynamics in stokes flow via a three-dimensional fully-implicit interfacial spectral boundary element algorithm. Journal of Computational Physics, 225(1):408–426, 2007.
[14] Sai K. Doddi and Prosenjit Bagchi. Three-dimensional computational modeling of multiple deformable cells flowing in microvessels. Physical Review E, 79:046318, Apr 2009.
[15] Wilhelm Flügge. Tensor analysis and continuum mechanics. Springer-Verlag, Berlin, New York, 1972.
[16] Jonathan B. Freund. The flow of red blood cells through a narrow spleen-like slit. Physics of Fluids, 25(11):110807, 2013.
[17] Jonathan B Freund. Numerical simulation of flowing blood cells. Annual Review of Fluid Mechanics, 46:67–95, 2014.
[18] M. Ganesh and I. G. Graham. A high-order algorithm for obstacle scattering in three dimensions. *Journal of Computational Physics*, 198(1):211–242, 2004.

[19] Ali Gürbüz. *Simulation of Pressure-Driven Red Blood Cell Flow through Constrictions by Boundary Integral Methods*. PhD thesis, University at Buffalo, 2020.

[20] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, New York, 1991.

[21] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker, and Carol S Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

[22] Yijun Liu and TJ Rudolphi. Some identities for fundamental solutions and their applications to weakly-singular boundary element formulations. *Engineering Analysis with Boundary Elements*, 8(6):301–311, 1991.

[23] Frithiof I. Niordson. *Shell theory*. North-Holland series in applied mathematics and mechanics. North-Holland, Amsterdam, New York, 1985.

[24] C Pozrikidis. Numerical simulation of the flow-induced deformation of red blood cells. *Annals of Biomedical Engineering*, 31(10):1194–1205, 2003.

[25] C Pozrikidis. Axisymmetric motion of a file of red blood cells through capillaries. *Physics of Fluids*, 17(3):031503, 2005.

[26] Constantine Pozrikidis. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press, 1992.

[27] Bryan Quaife and George Biros. Adaptive time stepping for vesicle suspensions. *Journal of Computational Physics*, 306:478–499, 2016.

[28] Abtin Rahimian, Shravan K Veerapaneni, Denis Zorin, and George Biros. Boundary integral method for the flow of vesicles with viscosity contrast in three dimensions. *Journal of Computational Physics*, 298:766–786, 2015.

[29] S Ramanujan and C Pozrikidis. Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: large deformations and the effect of fluid viscosities. *Journal of Fluid Mechanics*, 361:117–143, 1998.

[30] Timothy W. Secomb. Blood flow in the microcirculation. *Annual Review of Fluid Mechanics*, 49(1):443–461, 2017.

[31] The Mathworks. *MATLAB version 9.6.0.1072779 (R2019a)*. The Mathworks, Inc., Natick, Massachusetts, 2019.

[32] Shravan K Veerapaneni, Denis Gueyffier, George Biros, and Denis Zorin. A numerical method for simulating the dynamics of 3D axisymmetric vesicles suspended in viscous flows. *Journal of Computational Physics*, 228(19):7233–7249, 2009.

[33] Shravan K Veerapaneni, Abtin Rahimian, George Biros, and Denis Zorin. A fast algorithm for simulating vesicle flows in three dimensions. *Journal of Computational Physics*, 230(14):5610–5634, 2011.

[34] Ryan Zarychanski, Vincent P. Schulz, Brett L. Houston, Yelena Maksimova, Donald S. Houston, Brian Smith, Jesse Rinehart, and Patrick G. Gallagher. Mutations in the mechanotransduction protein PIEZO1 are associated with hereditary xerocytosis. *Blood*, 120(9):1908–1915, 2012.

[35] Hong Zhao, Amir HG Isfahani, Luke N Olson, and Jonathan B Freund. A spectral boundary integral method for flowing blood cells. *Journal of Computational Physics*, 229(10):3726–3744, 2010.

[36] Hong Zhao and Eric SG Shaqfeh. The dynamics of a vesicle in simple shear flow. *Journal of Fluid Mechanics*, 674:578–604, 2011.