Three-dimensional single framework multi-component lattice Boltzmann equation method for vesicle hydrodynamics

SPENDLOVE, James, XU, Xu <http://orcid.org/0000-0002-9721-9054>, SCHENKEL, Torsten <http://orcid.org/0000-0001-5560-1872>, SEATON, Michael, HALLIDAY, Ian and GUNN, Julian

Citation:

SPENDLOVE, James, XU, Xu, SCHENKEL, Torsten, SEATON, Michael, HALLIDAY, Ian and GUNN, Julian (2021). Three-dimensional single framework multi-component lattice Boltzmann equation method for vesicle hydrodynamics. Physics of Fluids, 33 (7), 077110. [Article]

Three-dimensional single framework multi-component lattice Boltzmann equation method for vesicle hydrodynamics

J. Spendlove,[1] X. Xu,[1, 2] T. Schenkel,[1, 2] M. A. Seaton,[3] I. Halliday,[4] and J. P. Gunn[4]
[1)] *Materials & Engineering Research Institute, Sheffield Hallam University, Howard Street, S1 1WB (UK)*
[2)] *Department of Engineering and Mathematics, Sheffield Hallam University, Howard Street, S1 1WB (UK)*
[3)] *Scientific Computing Department, UKRI STFC Daresbury Laboratory, Keckwick Lane, Warrington WA4 4AD*
[4)] *Dept of Infection, Immunity and Cardiovascular Disease, The Medical School, Beech Hill Road, Sheffield, S10 2RX*

(Dated: 15 June 2021)

We develop a three dimensional immersed boundary chromodynamic multi-component lattice Boltzmann method capable of simulating vesicles, such as erythrocytes. The presented method is encapsulated in a single framework, where the application of the immersed boundary force in the automatically adaptive interfacial region results in correct vesicle behaviour. We also set-down a methodology for computing the principal curvatures of a surface in a three-dimensional, physical space which is defined solely in terms of its surface normal vectors. The benefits of such a model are its transparent methodology, stability at high levels of deformation, automatic-adaptive interface and potential for the simulation of many erythrocytes. We demonstrate the utility of the model by examining the steady state properties, as well as dynamical behaviour within shear flow. The stability of the method is highlighted through its handling of high deformations, as well as interaction with another vesicle.

I. INTRODUCTION

A normal human red blood cell (RBC) in mechanical equilibrium is a biconcave vesicle of 6-8μm in diameter and has a maximum thickness of $2\mu m$[1]. An external membrane bounds a viscous, effectively incompressible fluid (cytoplasm)[2]. Two key micro-structural components of the membrane, a phospholipid bi-layer and a “skeletal" network of protein strands that interlink a set of anchoring points within that bilayer and act *in tandem* to resist changes in its surface area. This imparts incompressibility, the bilayer mechanics further imparting bending elasticity[3]. The thickness of the bilayer and cytoskeleton is approximately $10\,nm$. Accordingly, an RBC membrane, modelled within the continuum approximation of fluid dynamics, is considered to be a two parameter surface with constrained dynamics, which bounds otherwise independent, three-dimensional flow domains[4]. Physically, RBCs are highly resistant to changes in volume (owing to the incompressibility of the enclosed fluid) and surface area (owing to the effects outlined above) but they are, nevertheless, highly deformable[2]. One important consequence of this isochoric deformability is an ability to pass through capillaries of a diameter smaller than the RBC itself. We hypothesise, in this work, that an RBC membrane may be modelled as a quasi two-dimensional fluid (a narrow layer) with appropriately constrained dynamics, mediating interactions between an internal and external flow domain and that such a model may be implemented within a single computational framework. We neglect the role of the cytoskeleton *ab initio* and focus on the membrane for recovering appropriate dynamics, where this approximation can only be justified *a posteriori*. This article shows how to develop and parameterise a chromodynamic multi-component lattice Boltzmann equation (cMCLBE) simulation as a vehicle for this hypothesis.

We structure as follows. First, background work on the simulation of vesicles in flow is reviewed. Then, in Sec. III, the cMCLBE method for two immiscible fluids will be outlined, followed by an account of a suitable immersed boundary force with area conserving, bending rigidity and surface tension contributions, these components being necessary to enforce membrane behaviour at the boundary between the fluids. Then, an efficient method of computing bounding surface curvatures (needed in the computation of the immersed boundary force) will be outlined. Data in Sec. IV verify and validate the developed model. We present our conclusions in Sec. V.

II. BACKGROUND

In the case of a liquid-liquid interface, differentiated inter-molecular forces accumulate and give rise to a continuum scale interfacial tension force[5]. The physics of the more complex forces which *integrate-up* to a suitable continuum scale representation of the RBC membrane (in terms of interfacial tension, bending rigidity and membrane surface area conservation) is discussed from an appropriate perspective, by authors in Ref. 6. Several methodologies encapsulate these forces, essentially as immersed boundary force distributions[7,8], to describe the fluid dynamics of cellular suspensions, for applications—both practical and idealised—which require the explicit representation of suspended, deformable cells, or vesicles. The literature on this subject is diverse and we consider here only that which is most relevant to LB based approaches to the problem. That said, published research on vesicle simulation at various scales utilises lattice Boltzmann methods (LBM) quite extensively, since a mesoscale, bottom-up, approach to this continuum scale problem has much to commend it.

The first mesoscopic approach to simulating erythrocytes used a lattice spring method (LSM) to describe the membrane (surface) properties. The LSM is then *coupled* with a fluid-solver, for the blood plasma, a popular choice being single-component LBM. The resulting LSM-LBM model was shown to be capable of simulating vesicles[9–12]. The LSM captures the mechanics of an extended, elastic sheet embedded in a single fluid, through the use of point masses, connected via Hookean springs, which attempt to return a deformed object to a mechanical equilibrium- a bi-cuspid shape. At mechanical equilibrium, the set of masses (springs) lie on vertices (edges) of a quadrilaterally tessellated bi-cuspid and advect in flow. Dupin et al.[9] advanced the simulation of dense suspensions in 2007, when they simulated $o(10^2)$ erythrocytes suspension, at volume fraction of 30% using this essential apporach. The network membrane representation was later improved by Fedosov et al.[10]. For this class of coupled solver, the current state of the art is, arguably, the open-source *HemoCell* solver[11,12], which also adopts the mass-spring network approach. These coarse-grained approaches are an attractive option for the simulation of dense suspensions, due to their computational efficiency and scalability, yet there remain development opportunities relating to their accuracy and robustness i.e., their ability to handle larger levels of vesicle deformation accurately.

Another multi-framework approach employs finite element method (FEM) to describe the membrane. An FEM solver exchanges information with a fluid solver, e.g. single component LBM, dissipative particle dynamics (DPD) or boundary integral method (BIM)[13–21], which is subject to an immersed boundary force. In 2009, MacMeccan et al.[13], using LBM for the fluid solver, coupled to a FEM for the particle deformation, successfully simulated 200 fluid-filled capsules at 40% volume fraction. A year later, Kruger et al.[14] pioneered a similar approach, using immersed-boundary LBM, pioneering what is now called the lattice Boltzmann finite element method[14]. Such methods are widely recognised for their increased accuracy, compared to mass-spring network models, yet they can incur greater computational costs, making some approaches less scalable for larger numbers of particles. Recently, Kotsalos et al.[19,22], proposed a modular approach for simulating RBCs, again using LBM for the fluid solver, nodal projective FEM (npFEM) for the deformable body and the immersed boundary method (IBM) for the interaction between the deformable body and the fluid. Their model is computationally efficient and has increased accuracy at higher levels of deformation, compared to some of the spring-based meso-scopic models discussed. (We note that, in the approach advanced herein, accuracy is uniform at all levels of deformation.) However, these authors' multi-framework approach has increased model complexity. Moreover, the coupling of the deformable body to the fluid requires policing of the density of lattice points in the surface region, in order to accurately communicate a local influence of the deformable body onto the fluid solver —a problem which is automatically circumvented in this work, where there is a *de facto* adaptive mesh.

The motive for the present work is the lack of a computational model which is straightforwardly extendable to many vesicles, transparently encapsulates appropriate membrane physics even far from mechanical equilibrium, automatically solves meshing problems at all levels of vesicle deformation and, not least, relies upon a single framework methodology. Currently, some of the other available models (which are, we concede, more efficient on account of greater developmental maturity) either focus on scalability (to a large vesicle number) by sacrificing accuracy away from mechanical equilibrium, or on accuracy with diminished concern for computational cost. As we discussed above, all require multiple modelling techniques, which lead to algorithmic complexity. Here, we develop a transparent, three-dimensional multiple RBC simulation method which is: (i) based solely within MCLBE[23,24], (ii) has transparent encapsulation of appropriate membrane physics[6], (iii) is easily extended to dense RBC suspension[25] and (iv) is applicable to flow at any Reynolds number. By building on established, two-dimensional methodology[23,24] our work inherits the documented advantages of cMCLBE: exact component mass conservation for its fully immiscible independent fluids, ease of interface tracking and parallelisation, correct interface kinematics and an emergent interface with underwritten compact support (meaning resolution in the interfacial region is managed by a de facto adaptive mesh—simply, a spreading interface automatically occupies more points). Possibly our most important single step (see Section IV) is a three-dimensional MCLBE immersed boundary force which imparts physically appropriate RBC membrane behaviour in a manner consistent with continuum scale applications (i.e., as a boundary condition between the viscous flows separated by the membrane), which may be computed, efficiently, from the interface normal vectors alone.

III. METHODOLOGY

We consider the cMCLBE's red fluid to be the interior fluid, the blue fluid to be plasma and its interfacial fluid (the *de facto* membrane) to be subject to an immersed boundary force distribution, designed to impose upon that fluid the following physics: (i) constant boundary fluid volume and thickness (and hence area), and (ii) a local force in the membrane fluid that is correctly related to its principal curvatures and their spatial gradients. We start by outlining the base model (i.e. cMCLBE scheme in Sec. III A). Following this, we determine, investigate and verify an efficient and readily implementable (see below) immersed boundary force distribution fulfilling requirements (i) and (ii), in Sec. III B. The membrane bending and interfacial tension force contributions are both functions of its principal curvatures and their gradients.

Therefore, to determine an appropriate immersed boundary force, one needs readily computable expressions for the interfacial fluid region principal curvatures (note, the *de facto* membrane) requiring, as input, only a surface normal distribution; these we set out in Sec. III C. Their use in the derived membrane force is set out in Sec. III B.

A. Chromodynamic MCLBE

Since the derivation of the LBM, from the lattice Gas Automata (LGA)[26], its popularity has grown monotonically. LBM is an Eulerian solver which circumvents the Navier-Stokes equations by using kinetic scale evolution equations for a single particle distribution function. The extension of LBM to immiscible fluids[27] opened what is, undoubtedly, its most fertile field of current applications. LBM's computational and algorithmic purchase is selective. Canonically, LBM and MCLBE are an attractive option over traditional computational fluid dynamics techniques for (i) unsteady, (ii) multi-component or (iii) geometrically complex flows. The problem we address in this work is an apologist's ideal, for it spans items (i)..(iii).

The three dimensional simulation lattice we use throughout this work is designated $D3Q19$; its unit cell is defined by the set of Cartesian velocity vectors, $\underline{c}_i$, in Table I, together with discrete direction link label $i \in 0, 1, .., 18$ and two sets of lattice link weights, denoted t_i and g_i[28].

We designate internal (external) fluid as red (blue) and sum these components' link distributions, to yield a colour-blind distribution function as:

$$f_i(\underline{r},t) = R_i(\underline{r},t) + B_i(\underline{r},t), \tag{1}$$

for which the multi-relaxation time[29] forced evolution equation is:

$$\begin{aligned} f_i(\underline{r} + \underline{c}_i \Delta t, t + \Delta t) = f_i(\underline{r},t) + \\ - \sum_{j=0}^{Q-1} A_{ij} (f_j(\underline{r},t) - f_j^{(0)}(\rho, \underline{u})) + F_i, \end{aligned} \tag{2}$$

where the equilibrium distribution function ($f_i^{(0)}$) and source term (F_i) are defined as:

$$f_i^{(0)}(\rho, \underline{u}) = t_i \rho \left(1 + \frac{u_\alpha c_{i\alpha}}{c_s^2} + \frac{u_\alpha u_\beta c_{i\alpha} c_{i\beta}}{2 c_s^4} - \frac{u^2}{2 c_s^2} \right), \tag{3}$$

$$F_i = t_i \left[\frac{\underline{F}^{(t)} \cdot c_{i\alpha}}{c_s^2} + \frac{1}{2 c_s^4} \left(1 - \frac{\lambda_3}{2} \right) \left(F_\alpha^{(t)} u_\beta + F_\beta^{(t)} u_\alpha \right) \right]. \tag{4}$$

Above, $\underline{r}$, Δt, ρ, $c_{i\alpha}$, u_α, Q, c_s and λ_p correspond to the lattice position, time step, density, the α-component of the i^{th} lattice basis vector, the α-component of velocity, the number of lattice links, the colour-blind speed of sound and the pth eigenvalue corresponding to the collision matrix A_{ij}.

The MRT scheme implemented for this work (presented in[29]) is a variant of Dellar's scheme[28,30], which has enhanced density and velocity coupling (see Appendix A for details). It may be shown that to retrieve correct hydrodynamics, the collision matrix, $\underline{\underline{A}}$, and the source or forcing term, F_i, must have the following properties[31]:

$$\sum_i (1, \underline{c}_i, \underline{c}_i \underline{c}_i) F_i = \left(0, n \underline{F}^{(t)}, \frac{1}{2} \left[\underline{C} + \underline{C}^T \right] \right), \tag{5}$$

$$\sum_i (\underline{c}_i, \underline{c}_i \underline{c}_i, \underline{c}_i \underline{c}_i \underline{c}_i) A_{ij} = \left(0, \lambda_4 c_{j\alpha} c_{j\beta}, \lambda_{14} c_{j\alpha} c_{j\beta}^2 \right), \tag{6}$$

$$\sum_i (g_i, g_i \underline{c}_i, g_i \underline{c}_i \underline{c}_i) A_{ij} = \left(\lambda_{10} g_j, \lambda_{11} g_j c_{j\alpha}, \lambda_{17} g_j c_{j\alpha}^2 \right), \tag{7}$$

where $\underline{C}$ corresponds to a correction term, needed to recover the Navier-Stokes equations[31,32].

Weighted moments of the distribution functions yield the continuum scale observables. Here, we will require force-adjusted expressions for these macroscopic observables[32,33]:

$$(\rho_R, \rho_B) = \sum_i (R_i, B_i), \quad \underline{u} = \frac{\sum_i f_i(\underline{r},t) \underline{c}_i}{\rho} + \frac{\underline{F}^{(t)}}{2\rho}, \tag{8}$$

where, of course, $\rho = (\rho_R + \rho_B)$, ρ_R and ρ_B denote color-blind nodal density, red fluid nodal density and blue fluid nodal density.

A scalar, chromodynamic, or phase-field is computed from the distribution of the colour degree of freedom; it is defined as[34]:

$$\rho^N(\underline{r},t) \equiv \frac{\rho_R(\underline{r},t) - \rho_B(\underline{r},t)}{\rho_R(\underline{r},t) + \rho_B(\underline{r},t)} \in [-1, 1]. \tag{9}$$

ρ^N, identifies the component at location $\underline{r}$, with values of 1 (-1) corresponding to only red (blue) fluid at position $\underline{r}$ and the contour $\rho^N = 0$ representing the mid-interface, half way between the red and blue fluids. The local interfacial unit normal vector field can be efficiently computed from the gradient of the phase field, which is perpendicular to the interface:

$$\hat{n}(\underline{r},t) = - \frac{\underline{\nabla} \rho^N(\underline{r},t)}{|\underline{\nabla} \rho^N(\underline{r},t)|}. \tag{10}$$

It is from the above unit normal vector field that we will eventually compute the interfacial fluid (membrane) principal curvatures. Note that it exists throughout a thin shell, which is of finite volume, in the simulation domain. To prevent mixing of red and blue fluids, we apply a kinetic-scale, post collision, colour segregation rule, adapted from that of d'Ortona et al.[35]:

$$C_i^{\psi\psi}(\underline{r},t) = \frac{\rho_C(\underline{r},t)}{\rho(\underline{r},t)} f_i(\underline{r},t)^{\psi} \pm \beta t_i \frac{\rho_R(\underline{r},t) \rho_B(\underline{r},t)}{\rho(\underline{r},t)} \hat{n} \cdot \hat{\underline{c}}_i \Delta t,$$

here, superscript ψ identifies a post-collision value, $\psi\psi$ a post re-colour[35] value and $C \in [R, B]$. The $+$ ($-$) sign is used for $C = R$ (C=B) and interface width is related to the segregation parameter, β.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
c_{ix}	0	1	1	0	-1	-1	-1	0	1	0	1	0	-1	0	0	1	0	-1	0
c_{iy}	0	0	-1	-1	-1	0	1	1	1	0	0	1	0	-1	0	0	1	0	-1
c_{iz}	0	0	0	0	0	0	0	0	0	1	1	1	1	1	-1	-1	-1	-1	-1
t_i	t_0	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_2	t_2	t_2	t_1	t_2	t_2	t_2	t_2
g_i	g_0	g_1	g_2	g_1	g_2	g_1	g_2	g_1	g_2	g_1	g_2	g_2	g_2	g_2	g_1	g_2	g_2	g_2	g_2

TABLE I: $D3Q19$ lattice structure, showing lattice link (i), lattice velocity vector ($\underline{c}_i$) components and lattice weights (t_i, g_i). Here, $t_0 = \frac{1}{3}$, $t_1 = \frac{1}{18}$, $t_2 = \frac{1}{36}$ and $g_0 = 1$, $g_1 = -2$, $g_2 = 1$.

The body or immersed boundary force, $\underline{F}^{(t)}$, to be used in Eqs (4, 8), is a gradient-weighted sum (see Sec. III B 4, also Lishchuk et al.,[34]) of the moduli of the forces to be derived shortly:

$$\underline{F}^{(t)} = \frac{1}{2}\underline{\nabla}\rho^N \left(F^{(a)} + F^{(b)} + F^{(l)} \right), \tag{11}$$

specifically, the area conserving, $\underline{F}^{(a)}$, the bending, $\underline{F}^{(b)}$, and surface tension, $\underline{F}^{(l)}$ forces; see Sec. III B:

$$\underline{F}^{(a)} = -\alpha H \left(A - A_0 \right) A \hat{n}, \tag{12}$$

$$\underline{F}^{(b)} = -\kappa_B \left(\frac{3}{2}\Delta_S H + H^3 - KH \right) \hat{n}, \tag{13}$$

$$\underline{F}^{(l)} = 2\sigma H \hat{n}, \tag{14}$$

where H and K are the mean and Gaussian curvatures of the interface fluid i.e., the membrane surface. Note, that Eqs (12, 13, 14) all depend upon H, K or both. Also note that the forces in Eqs (12, 13, 14) are all parallel with $\underline{\nabla}\rho^N$ in any case. Due to the prominence of these forces, we proceed to provide further detail and to characterise their role in a flow problem.

B. Immersed Boundary or Body Force

In order to replicate correct vesicle behaviour using the MCLBE scheme, one must apply an appropriate immersed boundary force to the interface which enforces the key properties of constant interface area and a membrane force which is related to the principal curvatures of the interface. This section will outline the developed immersed boundary force, $\underline{F}^{(t)}$, introduced in Eq. (11), as well as characterise it through the use of non-dimensional groups.

1. Membrane Area Conservation Force

We derive a force density, $\underline{F}^{(a)}$, which acts to conserve the cross-sectional area of the vesicle membrane. Our force is based upon the physical compressibility of the membrane material, where it is found mostly to act to restore the overall cross sectional area of the interface fluid, should it drift from a target value. Since the interfacial area of the red and blue fluids is to be driven far from that minimum value dictated by the action of simple interfacial tension in this work, that role should not be under-estimated. The membrane material is taken to be isotropic, characterised by single "interface compressibility" α, to have zero preferred curvature and to be such that it relaxes strains very rapidly in the tangential direction. Indeed, we will consider the membrane to have a "quasi-liquid nature", the differential strain at all locations upon it being equal. As such, we neglect any component of $\underline{F}^{(a)}$ which is locally tangent to S, the fluid-fluid interface surface. In simulation, we aim to set a value of α which conserves total membrane area. It may therefore be argued that $\underline{F}^{(a)}$ is a Lagrange multiplier or "constraint force".

Let (x,y,z) span a suitable local frame for a patch of strained vesicle surface of area dA'. Patch dA' corresponds to small parameter ranges $x \in [0,dx]$, $y \in [0,dy]$. When not forming the closed surface of a vesicle, dA' is assumed flat, with area $dxdy$. For given (x,y), dA' is located about a position $\underline{u}(x,y)$, which we take to be a single-valued vector. For a schematic of this geometry, please see panel **(A)** of Figure 11. (Here, of course, we effectively treat dA' in a way which is analogous to Monge representation, where a surface would be parameterised as $(x,y,f(x,y))$, with scalar $f(x,y)$ the single-valued, local height of the vesicle surface, at given x,y.) Hence:

$$\begin{aligned} dA' &= |\underline{u}_x \times \underline{u}_y| dxdy, \\ &= |\varepsilon_{ijk}(u_j)_x(u_k)_y| dxdy, \\ &= \sqrt{\varepsilon_{ijk}\varepsilon_{ilm}(u_j)_x(u_k)_y(u_l)_x(u_m)_y} dxdy, \\ &= \sqrt{(\underline{u}_x \cdot \underline{u}_x)(\underline{u}_y \cdot \underline{u}_y) - (\underline{u}_x \cdot \underline{u}_y)^2} dxdy, \end{aligned} \tag{15}$$

where we have used the identity $\varepsilon_{ijk}\varepsilon_{ilm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl}$,[36]. The above may be written in terms of the co-efficients of the first fundamental form, I, as follows[37]:

$$dA' = \sqrt{EG - F^2} dxdy,$$

where:

$$E = \underline{u}_x \cdot \underline{u}_x, \quad G = \underline{u}_y \cdot \underline{u}_y, \quad F = \underline{u}_x \cdot \underline{u}_y. \tag{16}$$

Using this notation, the area of a finite part of the membrane is:

$$\Delta A = \iint \sqrt{EG - F^2} dxdy, \tag{17}$$

where the integration is taken over a finite range of parameters $x \in [0, \Delta x]$, $y \in [0, \Delta y]$. Before proceeding, we note a surface normal vector may be defined as $\underline{n} = \underline{u}_x \times \underline{u}_y$ and hence:

$$\hat{n} = \frac{\underline{u}_x \times \underline{u}_y}{\sqrt{EG - F^2}}, \tag{18}$$

and we state for eventual use below the approximations:

$$\Delta A \approx \sqrt{EG - F^2}\Delta x \Delta y, \quad \Delta A_0 \approx \Delta x \Delta y. \tag{19}$$

Proceeding, the free energy of our finite element of vesicle surface, associated with its deformation may be formally written as a muti-variate functional of x and y:

$$A_s = \frac{\alpha}{2} \iint \mathbb{F}(x, y, \underline{u}, \underline{u}_x(x,y), \underline{u}_y(x,y)) dx dy, \tag{20}$$

where the integration range is $x \in [0, \Delta x]$, $y \in [0, \Delta y]$, and of course:

$$\mathbb{F}(x, y, \underline{u}, \underline{u}_x(x,y), \underline{u}_y(x,y)) = \left(\sqrt{EG - F^2} - 1\right)^2. \tag{21}$$

Note that $\mathbb{F}$ does not depend, explicitly, on x, y and $\underline{u}$. Taking the variational derivative of Eq. (20) with respect to vector component $u_k(x,y)$ we obtain an expression for the k^{th} component of a membrane area-conserving force (where $k = 1, 2, 3$):

$$\begin{aligned} F_k^{(a)} =& \frac{\delta A_s}{\delta u_k(x,y)} = \frac{\partial \mathbb{F}}{\partial (u_k)} - \frac{\partial}{\partial x}\frac{\partial \mathbb{F}}{\partial (u_k)_x} - \frac{\partial}{\partial y}\frac{\partial \mathbb{F}}{\partial (u_k)_y}, \\ =& -\frac{\alpha}{2}\frac{\partial}{\partial x}\frac{\partial}{\partial (u_k)_x}\left(\sqrt{EG - F^2} - 1\right)^2 + \\ & -\frac{\alpha}{2}\frac{\partial}{\partial y}\frac{\partial}{\partial (u_k)_y}\left(\sqrt{EG - F^2} - 1\right)^2. \end{aligned} \tag{22}$$

Consider the first term, which we designate $-T_1$, on the right hand side of Eq. (22):

$$\begin{aligned} T_1 =& \frac{\alpha}{2}\frac{\partial}{\partial x}\frac{\partial}{\partial (u_k)_x}\left(\sqrt{EG - F^2} - 1\right)^2, \\ =& \alpha \frac{\partial}{\partial x}\left[\left(\frac{\sqrt{EG - F^2} - 1}{\sqrt{EG - F^2}}\right)\Big((u_k)_x G - (u_k)_y F\Big)\right], \\ =& \alpha \frac{\partial}{\partial x}\left[\left(\frac{\sqrt{EG - F^2}\Delta x \Delta y - \Delta x \Delta y}{\sqrt{EG - F^2}\Delta x \Delta y}\right) \times \right. \\ & \left. \Big((u_k)_x G - (u_k)_y F\Big)\right], \\ \approx & \alpha \left(\frac{\Delta A - \Delta A_0}{\Delta A}\right)\frac{\partial}{\partial x}\Big((u_k)_x G - (u_k)_y F\Big). \end{aligned} \tag{23}$$

The last factorisation is justified by our assumption of rapid relaxation of tangential stress which means the relative straining of a surface area element, $\Delta A = \Delta x \Delta y$ is constant over all surface locations (x, y). Similarly, the negative of the second term on the right hand side of Eq. (22) becomes:

$$T_2 \approx \alpha \left(\frac{\Delta A - \Delta A_0}{\Delta A}\right)\frac{\partial}{\partial y}\Big((u_k)_y E - (u_k)_x F\Big). \tag{24}$$

Now, substituting Eq. (23) and Eq. (24) into Eq. (22):

$$\begin{aligned} F_k^{(a)} =& -\alpha \left(\frac{\Delta A - \Delta A_0}{\Delta A}\right) \times \left[\frac{\partial}{\partial x}\Big((u_k)_x G - (u_k)_y F\Big) + \right. \\ & \left. + \frac{\partial}{\partial y}\Big((u_k)_y E - (u_k)_x F\Big)\right], \\ =& -\alpha \left(\frac{\Delta A - \Delta A_0}{\Delta A}\right) \times \Big[E(u_k)_{yy} + G(u_k)_{xx} + \\ & -2F(u_k)_{xy} + (u_k)_x(G_x - F_y) + (u_k)_y(E_y - F_x)\Big]. \end{aligned} \tag{25}$$

We now project out that component of above force which is perpendicular to the membrane, $\underline{F}^{(a)} \cdot \underline{\hat{n}}$, disregarding the tangential component. (This is consistent with the assumption that the membrane is not differentially strained.) We obtain the following:

$$\begin{aligned} & F_k^{(a)}\hat{n}_k = -\alpha \left(\frac{\Delta A - \Delta A_0}{\Delta A}\right) \times \\ & \Big[E\hat{n}_k(u_k)_{yy} + G\hat{n}_k(u_k)_{xx} - 2F\hat{n}_k(u_k)_{xy} + \\ & + \hat{n}_k(u_k)_x(G_x - F_y) + \hat{n}_k(u_k)_y(E_y - F_x)\Big] \end{aligned} \tag{26}$$

Noting that $\hat{n}_k(u_k)_x = \hat{n}_k(u_k)_y = 0$ identically (see definition in Eq. (18)), we find:

$$\begin{aligned} \underline{F}^{(a)} \cdot \underline{\hat{n}} = F_k^{(a)}\hat{n}_k =& -\alpha \left(\frac{\Delta A - \Delta A_0}{\Delta A}\right) \times \\ & (E\hat{n}_k(u_k)_{yy} + G\hat{n}_k(u_k)_{xx} - 2F\hat{n}_k(u_k)_{xy}), \\ =& -\alpha \left(\frac{\Delta A - \Delta A_0}{\Delta A}\right)(EN + GL - 2FM), \end{aligned} \tag{27}$$

where N, L and M are the coefficients of the second fundamental form of the membrane surface, usually denoted II[37]. In terms of the coefficients of I and II, we can write the mean curvature as:

$$H = \frac{1}{2}(\kappa_1 + \kappa_2) \propto \left(\frac{EN + GL - 2FM}{EG - F^2}\right), \tag{28}$$

whence, from Eq. (27) we have for the normal component of the interface area conserving force density $F_n^{(a)} \propto -\alpha H \frac{(\Delta A - \Delta A_0)}{\Delta A}(EG - F^2)$. Noting that $(EG - F^2) \propto \Delta A^2$, we absorb constants of proportionality into α and write: $F_n^{(a)} = -\alpha H (\Delta A - \Delta A_0)\Delta A$ and finally:

$$\underline{F}^{(a)} = -\alpha H (A - A_0) A \hat{n}. \tag{29}$$

In the above, the parameter α clearly now only represents the physical interface compressibility in broad terms.

2. Membrane Surface Tension Force

A membrane generates an interfacial tension force[6]. The immersed boundary force used in cMCLBE for interfacial tension was first defined Ref. 34. Formatted for the present investigation, with the weight factor included in the definition of $\underline{F}^{(t)}$, we can write for a surface tension contribution:

$$\underline{F}^{(l)} = 2\sigma H \hat{n}, \tag{30}$$

where σ is the surface tension parameter and H is the mean curvature of the surface. We shall return to the importance of σ on the outcome of vesicle shape in Sec. III B 4.

3. Membrane Bending Force

For clarity, before presenting the membrane bending force, it is first necessary to introduce notation that is used in Sections (III B 3, III C), with further explanation of the two surface frames at the time of use in Sec III C. As summarised in Tab. II, we use symbols (x,y,z) as suitable local coordinates for a patch of vesicle surface. It is now convenient to use a second local frame (x',y',z'). Eventually, expressions obtained in either system must be transformed to the coordinate system of the simulation, (X,Y,Z). Capitals are also used to identify the components of any vector measured in the frame spanned by (X,Y,Z). For example, in the frame (X,Y,Z) the unit normal is written $\hat{N} = (\hat{N}_X, \hat{N}_Y, \hat{N}_Z)$, the position vector is $\underline{X} = (X,Y,Z)$ and the gradient taken in that frame by $\underline{\nabla}_X$.

Co-ordinate Frame Notations		
Frame	Co-ordinate notation	Normal notation
Laboratory $\sum_L$	$\underline{X} = (X,Y,Z)$	$\hat{N} = (\hat{N}_X, \hat{N}_Y, \hat{N}_Z)$
Surface (General) $\sum$	$\underline{r} = (x,y,z)$	$\hat{n} = (\hat{n}_x, \hat{n}_y, \hat{n}_z)$
Surface (Specific) $\sum'$	$\underline{r}' = (x',y',z')$	$\hat{n}' = (\hat{n}'_x, \hat{n}'_y, \hat{n}'_z)$

TABLE II: The three co-ordinate frames used in this work. $\sum_L$ denotes the laboratory or simulation frame. $\sum$ is any local frame with its origin in the surface, in which the tangent vectors will not lie in the principal curvature planes of the surface, $\sum'$ is the local frame, still with its origin in the surface, in which the tangent vectors lie in the surface principal curvature planes. See panels **(B)** and **(C)** of Figure 11 for a visual representation of frames $\sum$ and $\sum'$.

Following the work reported in Ref. 23, the modulus of a bending force acting in the normal direction on a membrane is (see Eq.(A17) and Eq.(A18) of Ref. 23):

$$F^{(b)} = \kappa_B C + \underline{\nabla}_s \kappa_B \cdot \left(\underline{\nabla}_s (H - H_0)\right), \tag{31}$$

where:

$$C = H(H^2 - H_0^2) + \frac{1}{2}\Delta_s H - (H - H_0)K, \tag{32}$$

in which Δ_s is the Laplace Beltrami operator, $\underline{\nabla}_s$ is the surface gradient operator, H_0 the preferred curvature of the membrane and κ_B the membrane bending rigidity. For simplicity, we set $H_0 = 0$ and κ_B =constant. The Laplace Beltrami operator is defined as follows:

$$\Delta_s H = \underline{\nabla}_s \cdot \left(\underline{\nabla}_s H\right). \tag{33}$$

We shall use the following definition of the surface gradient operator, conveniently expressed in the coordinate frame of the simulation:

$$\underline{\nabla}_s f = \underline{\nabla}_X f - \left(\hat{N} \cdot \underline{\nabla}_X f\right) \hat{N}, \tag{34}$$

where f is some function, which has had its gradient in the local surface normal direction removed, so only its gradient tangent to the surface with local normal $\hat{N}$ remains. In Cartesian tensor notation, a component of the above vector may be written:

$$\frac{\partial}{\partial X_{i\,s}} f = \frac{\partial}{\partial X_i} f - \hat{N}_i \hat{N}_j \frac{\partial}{\partial X_j} f, \quad i,j = 1,2,3 \tag{35}$$

where the derivatives are calculated numerically using the following compact stencil which is $O(\underline{c}_i^4)$ accurate[38]:

$$\frac{\partial f}{\partial X_\alpha} = \frac{1}{c_s^2} \sum_i t_i f(\underline{r} + \underline{c}_i) c_{i\alpha} + O(\underline{c}_i^4), \tag{36}$$

where $X_\alpha \in [X,Y,Z]$ and i is the lattice link index.

Accordingly, we can compute the Laplace-Beltrami operator directly as:

$$\Delta_s f = \left(\frac{\partial}{\partial X_{i\,s}}\right)\left(\frac{\partial}{\partial X_{i\,s}}\right) f, \tag{37}$$

which, on appeal to Eq.(35) and some straightforward algebra, gives:

$$\Delta_s H = \Delta_X H - \left(\hat{N} \cdot \underline{\nabla}_X\right)^2 H + \hat{N} \cdot \underline{\nabla}_X H^2, \tag{38}$$

in which all gradient computations are made in the simulation frame, note. That is, it is not necessary to employ elaborate coordinate transformations to evaluate the terms in the right hand side of Eq.(38).

4. Characterisation of Membrane Immersed Boundary Force in Simulation Coordinates

We consider the parameterisation of equilibrium and steady-state vesicle shapes, using non-dimensional groups.

A smooth, numerically stable and convenient force weight function is $\frac{1}{2}|\underline{\nabla}_X \rho^N|$[23]. Since $\underline{\nabla}\rho^N$ defines the interface normal, we can write Eq. (11) within the simulation coordinate frame $\underline{X} = (X,Y,Z)$ as $\underline{F}^{(t)} = \frac{1}{2}\underline{\nabla}_X \rho^N \left(F^{(a)}(\underline{X}) + F^{(b)}(\underline{X}) + F^{(l)}(\underline{X})\right)$; this force appears in the dynamics of the red and blue fluids thus:

$$\frac{D}{Dt}\rho v_i = -c_s^2 \frac{\partial \rho}{\partial X_i} + 2\nu(\lambda_3) \frac{\partial}{\partial X_j}\left(\rho S_{ij}\right) + F_i^{(t)}(\underline{X}). \tag{39}$$

Flow induced deformation will be discussed shortly.

First consider equilibrium shape. $F_i^{(t)}(\underline{X})$ is significant only in the interfacial region, the membrane compressibility force contribution, $\underline{F}^{(a)}$, affects the speed of convergence to the target vesicle cross-sectional area but for a correctly parameterised simulation, vesicle area is approximately constant, so $\underline{F}^{(a)}$ presumably has a negligible effect on steady shape. Therefore, for purposes of parameterising vesicle shape in flow, we approximate:

$$\underline{F}^{(t)} \approx \frac{1}{2}\underline{\nabla}_X \rho^N \left(F^{(b)}(\underline{X}) + F^{(l)}(\underline{X})\right), \tag{40}$$

where $F^{(b)}(\underline{X})$ and $F^{(l)}(\underline{X})$ are evaluated from adaptations of Eqs. (31, 38) : $F^{(b)} = \kappa_B H(H^2 - K) + \frac{3}{2}\kappa_B \left(\Delta_X H - \left(\hat{N}\cdot\underline{\nabla}_X\right)^2 H + \hat{N}\cdot\underline{\nabla}_X H^2\right)$; $F^{(l)} = 2\sigma H$. To non-dimensionalise, now we chose the un-deformed initial drop radius, R_0 (lattice units) as a length scale: hence $\kappa_{Bi} = \frac{1}{R_0}\bar{\kappa}_{Bi}$ $(i = 1,2)$, $K = \frac{1}{R_0^2}\bar{K}$, $H = \frac{1}{R_0}\bar{H}$, $\frac{\partial}{\partial X} = \frac{1}{R_0}\frac{\partial}{\partial \bar{X}}$ etc. Weight factor $\frac{1}{2}\underline{\nabla}\rho^N$ in Eq. (40) may be written $\frac{1}{2}\beta(1-\rho^{N^2})\hat{n}^{23}$, so moduli $F^{(b)}$ and $F^{(l)}$ transform as:

$$F^{(b)} = \frac{\kappa_B}{R_0^3}\bar{F}^{(b)}, \quad F^{(l)} = \frac{\sigma}{R_0}\bar{F}^{(l)}, \tag{41}$$

where we have ignored factors of $(1-\rho^{N^2})$ (which are $O(1)$ in the interfacial region) and:

$$\begin{aligned}\bar{F}^{(b)} &= \bar{H}(\bar{H}^2 - \bar{K}) \qquad (42)\\ &+\frac{3}{2}\left(\bar{\Delta}_X \bar{H} - \left(\hat{N}\cdot\underline{\bar{\nabla}}_X\right)^2 \bar{H} + \hat{N}\cdot\underline{\bar{\nabla}}_X \bar{H}^2\right),\\ \bar{F}^{(l)} &= 2\bar{H}. \qquad (43)\end{aligned}$$

Alongside the dimensionless parameters of equilibrium shape (Eqs. (41)) we must clearly add a metric of relative target area $\frac{A_0}{R_0^2}$, to measure the equilibrium deflation. Hence, we arrive at the following dimensionless groups:

$$\frac{\kappa_B}{R_0^3}, \quad \frac{\sigma}{R_0}, \quad \frac{A_0}{R_0^2}, \tag{44}$$

i.e. the equilibrium vesicle parameter space is $\mathbb{R}^3$. From Eqs. (41), the interfacial tension and bending forces have ratio $r = \frac{\kappa}{\sigma R_0^2}$. Constraining e.g. $r =$constant, $\frac{A_0}{4\pi R_0^2}$ =constant, might reduce this to $\mathbb{R}^2$, but this was found to fix vesicle shape only within limits.

For completeness, we now characterise the steady-state vesicle deformation, induced by frictional fluid stresses. A steadily streaming vesicle will, presumably, deform subject to gradients in velocity, which are identical in the rest and lab frames. Let γ denote a steady local fluid shear rate, measured in lattice units. Additionally, we assume that a vesicle is always close to its quasi-steady shape, that compressibility and inertia are negligible. We seek the analogues for simple drop capillary number, Ca. The parameter of the steady balance between normalised deformation forces, including viscous stresses, and the restoring effect of the immersed boundary membrane force is obtained from the body-forced, weakly-compressible momentum equation Eq. (39), non-dimensionalised on its viscous term:

$$\underline{0} \approx -\frac{\rho_{\mathrm{ref}} c_s^2}{\eta\gamma}\underline{\bar{\nabla}}\bar{\rho} + \bar{\nabla}^2\underline{\bar{v}} + \frac{\kappa_B}{R_0^2 \eta\gamma}\bar{F}^{(b)}\hat{n} + \frac{\sigma}{\eta\gamma}\bar{F}^{(l)}\hat{n}, \tag{45}$$

where ρ_{ref} is some reference density. When it is possible to ignore pressure, when viscous forces dominate, there remain *two* dimensionless groups which will serve as pseudo capilary numbers for a vesicle:

$$Ca_{\mathrm{ves}} \equiv \frac{\eta R_0^2 \gamma}{\kappa_B}, \quad Ca \equiv \frac{\eta\gamma}{\sigma}. \tag{46}$$

Eq. (40) was derived in the continuum limit but it is applied in MCLBE in a discretised form at individual lattice sites. In practice, samples of the continuous force distribution are applied to the fluid at a set of discrete lattice points via the discrete kinetic equation source term. These samples, evaluated from Eq. (40), are, moreover, based upon numerical gradients and also subject to non-linearities arising from cutting off calculations outside the interfacial region. Discretisation therefore means that the only guarantee of linear and angular momentum conservation in MCLBE simulation is adequate resolution of the interfacial region.

All three immersed forces considered in this section are dependent upon the principal curvatures of the surface and their spatial gradients. Therefore, we now proceed to what is, perhaps, the key outcome of this work: an efficient, accurate and stable method of computing the principal curvatures of the surface in the simulation frame without adding significantly to the existing computational load.

C. Expressions for Vesicle Surface Principal Curvatures

In MCLBE, an immiscible fluid-fluid interface is, essentially, emergent—its location is not directly computed or tracked. In principle, a local interfacial manifold might be computed for the chromo-dynamic variant from an elaborate, expansive interpolation of ρ^N. In contradistinction, the interfacial unit normal vector field, $\hat{n}$, is required by the MCLBE algorithm and easy to compute using optimal, compact stencils. The problem addressed in this section is how to efficiently compute interface metrics. Specifically, the principal curvatures of the fluid-fluid surface, S, at any given point P solely from the pre-existing set of its normal vectors. To achieve this, we first seek expressions for surface principal curvatures in terms of the spatial gradients of $\hat{n}$, which can then be suitably discretised. In the following analysis, it will be necessary to transform between coordinate frames. The frame of the laboratory (i.e. the simulation lattice) has co-ordinates $\underline{X} = (X,Y,Z)$ and normal vector components $(\hat{N}_X, \hat{N}_Y, \hat{N}_Z)$. With the usual conventions:

$$\left[\hat{N}(\underline{X})\right]_P = \left[(\hat{N}_X, \hat{N}_Y, \hat{N}_Z)\right]_P = -\frac{\left[\underline{\nabla}_X \rho^N\right]_P}{|\underline{\nabla}_X \rho^N|_P}, \tag{47}$$

with $\underline{\nabla}_X = \left(\frac{\partial}{\partial X}, \frac{\partial}{\partial Y}, \frac{\partial}{\partial Z} \right)$.

Let us introduce local co-ordinate frame $\Sigma(x,y,z)$, which originates at P, with its x,y co-ordinates in the local tangent plane; in $\Sigma(x,y,z)$ the position and normal vector components are denoted $\underline{r} = (x,y,z)$ and $(\hat{n}_x, \hat{n}_y, \hat{n}_z)$ respectively. A second, local co-ordinate frame $\Sigma'(x',y',z')$ is also assumed to originate on P, with x',y' co-ordinates in the local tangent plane *in the principal directions* of S and in $\Sigma'(x',y',z')$, the normal vector components are denoted $(\hat{n'}_x, \hat{n'}_y, \hat{n'}_z)$ (see Table II). Specifically then, in $\Sigma(x,y,z)$ $(\Sigma'(x',y',z'))$ the z (z') axis is parallel to the normal to S at P, co-ordinate z (z') is the local elevation of S relative to the local tangent plane, and P is located at $x = y = z = 0$ $(x' = y' = z' = 0)$. (See panels **(B)** and **(C)** of Figure 11 for a visual representation of frames $\sum$ and $\sum'$.) Noting that for chosen x, y (say), the value of z is implicitly identified, let S be such that $F(x,y,z) = 0$. Then, by the implicit function theorem:

$$z = f(x,y). \tag{48}$$

Of course, this corresponds to a Monge representation of the surface, where $z = f(x,y)$ represents the single-valued displacement from a local (x-y) tangent plane of the surface S at point P. (In the present context, $z = f(x,y)$ corresponds to surface ρ^N =constant; translating in the local z direction corresponds to changing the value of this constant.) Expanding $f(x,y)$ as a polynomial in x,y, in the neighbourhood of P as follows:

$$f(x,y) = Ax^2 + By^2 + Cxy + O(x^3, y^3). \tag{49}$$

We note that there is no constant term in Eq. (49) because the origin of $\Sigma(x,y,z)$ is assumed to be located at P. There is no linear term in Eq. (49) because coordinates x,y lie in the tangent plane to S at P. We return to this issue shortly. There is a term in product xy in Eq. (49) because the co-ordinates x,y do not necessarily lie in the principal planes of S at P. We note the following identities:

$$A = \frac{1}{2}[f_{xx}]_P, \quad B = \frac{1}{2}[f_{yy}]_P, \quad C = f_{xy} = [f_{yx}]_P, \tag{50}$$

in which, we emphasise, the derivatives of f are all to be evaluated at P i.e. at the origin of Σ'. Now Eq. (49) may be written:

$$f(x,y) = \underline{r} \cdot \underline{\underline{c}} \cdot \underline{r}^T, \quad \underline{\underline{c}} = \frac{1}{2} \begin{pmatrix} 2A & C & 0 \\ C & 2B & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{51}$$

No confusion should arise between matrix $\underline{\underline{c}}$, above, and the scalar C in Eq. (49). Matrix $\underline{\underline{c}}$ is, of course, the Hessian of f at P. Now, from Eq. (48) within frame Σ, one can write $\underline{n} = \underline{\nabla}_{\underline{r}}(z - f(x,y)) = (-f_x, -f_y, 1)$ so that, normalising, we have within the environment of P:

$$\hat{n} = -\frac{1}{\sqrt{1 + f_x^2 + f_y^2}} (f_x, f_y, -1). \tag{52}$$

Note that at P itself

$$[f_x]_P = [f_y]_P = 0, \tag{53}$$

which implies $[\hat{n}]_P = (0,0,1)$ as expected.

Let us proceed to consider how Cartesian frame $\Sigma(x,y,z)$ which, we have seen, provides a basis for a local parameterisation of S at P and may be straightforwardly determined from the set for MCLBE observables which are usually computed, namely the vector field $\hat{N}$. A possible choice for the basis vectors of frame Σ may be written in terms of $\hat{N}$ as follows:

$$\begin{aligned} \hat{e}_z &= \Big([\hat{N}_X]_P, [\hat{N}_Y]_P, [\hat{N}_Z]_P \Big), \\ \hat{e}_x &= \frac{1}{\sqrt{1 - [N_Z]_P^2}} \Big(-[\hat{N}_Y]_P, [\hat{N}_X]_P, 0 \Big), \\ \hat{e}_y &= \hat{e}_z \times \hat{e}_x \\ &= \frac{1}{\sqrt{1 - [N_Z]_P^2}} \times \\ &\quad \Big(-[\hat{N}_X \hat{N}_Z]_P, -[\hat{N}_Y \hat{N}_Z]_P, [(1 - \hat{N}_Z^2)]_P \Big). \end{aligned} \tag{54}$$

Other possibilities are considered (see Appendix B for more detail). We now proceed in seeking to align the x,y directions of Σ with the principal directions of S at P.

Let us rotate frame $\Sigma(x,y,z)$, about $\hat{e}_z$, to define frame $\Sigma'(x',y',z')$, which is, recall, assumed to be such that x' and y' measure distance along the principal directions of S, at P. Note, coordinates x',y' still lie in the tangent plane to S at P. To find Σ', rotate Σ about its z axis, by an angle α such that constant C in Eq. (49) is eliminated. Using the usual orthogonal Euler rotation matrices, we have $\underline{r}'^T = \underline{\underline{r_z}}(\alpha)\underline{r}^T$, whence $\underline{r}^T = \underline{\underline{r_z}}(\alpha)^T \underline{r}'^T$, whence $\underline{r} = \underline{r}' \underline{\underline{r_z}}(\alpha)$. Eq. (51) for S may now be transformed:

$$f(x',y') = \underline{r}' \, \underline{\underline{c}}' \, \underline{r}'^T, \quad \underline{\underline{c}}' = \underline{\underline{r_z}}(\alpha) \, \underline{\underline{c}} \, \underline{\underline{r_z}}(\alpha)^T, \tag{55}$$

and, using the explicit form of $\underline{\underline{r}}_z(\alpha)^{36}$ and choosing a rotation angle:

$$\alpha = \frac{1}{2} \tan^{-1} \left(\frac{C}{A - B} \right) = \frac{1}{2} \tan^{-1} \left(\frac{2 f_{xy}}{f_{xx} - f_{yy}} \right), \tag{56}$$

(having used Eq. (50)) we find:

$$\underline{\underline{c}}' = diag(\kappa_1, \kappa_2, 0), \tag{57}$$

where:

$$\begin{aligned} \kappa_1 &= f_{xx} \cos^2(\alpha) + f_{yy} \sin^2(\alpha) + f_{xy} \sin(2\alpha), \\ \kappa_2 &= f_{xx} \sin^2(\alpha) + f_{yy} \cos^2(\alpha) - f_{xy} \sin(2\alpha), \end{aligned} \tag{58}$$

and all derivatives in the above are evaluated at P. We note that $\kappa_1 + \kappa_2 = [f_{xx} + f_{yy}]_P$, which corresponds to $-\left[\underline{\nabla}_{\underline{r}} \cdot \hat{n}\right]_P = 2H$ in frame $\Sigma(x,y,z)$ (see Eq. (62) below). Hence, as expected, we confirm $\frac{1}{2}(\kappa_1 + \kappa_2) = H$, the mean curvature of S.

In summary, by applying rotation $\underline{\underline{r_z}}(\alpha)$ with α defined in Eq. (56), we have positioned the x' and y' coordinate axes in the principal planes of S, whence we can write:

$$f(x',y') = \frac{1}{2}\kappa_1 x'^2 + \frac{1}{2}\kappa_2 y'^2, \tag{59}$$

with κ_1 and κ_2, (see Eq. (58)), clearly the principal curvatures of S at P. Henceforth, we define mean (H) and Gaussian (K) curvature of the surface as follows:

$$H = \frac{1}{2}\left(\kappa_1 + \kappa_2\right), \qquad K = \kappa_1 \kappa_2. \tag{60}$$

To fulfil our stated aim, we must now find expressions for f_{xx} etc. in terms of the the spatial derivatives of $\hat{n}$. Recall, x, y are co-ordinates on the local tangent plane at P, with directions defined in Eqs. (54). From Eq. (52), we have can define:

$$f_{xx} = \frac{\partial}{\partial x}\left(-\hat{n}_x(1+f_x^2+f_y^2)^{1/2}\right), \tag{61}$$

etc. Using the product rule and Eq. (53) above, we obtain, after some straightforward algebra:

$$\begin{aligned} [f_{xx}]_P &= -\left[(\hat{n}_x)_x\right]_P, & (62)\\ [f_{yy}]_P &= -\left[(\hat{n}_y)_y\right]_P, \\ [f_{xy}]_P &= -\left[(\hat{n}_y)_x\right]_P = -\left[(\hat{n}_x)_y\right]_P, \end{aligned}$$

where $(\hat{n}_y)_x = \frac{\partial \hat{n}_y}{\partial x}$ etc.

Clearly Eqs. (58) rely upon derivatives evaluated in a special reference frame in which the $[\hat{n}]_P$ lies in the z-direction and co-ordinates x, y span the tangent plane at P. Equivalent expressions in the simulation frame are required if costly computation is to be avoided. The reader is directed to Appendix B, where the derivatives in Eq. (62) are transformed into derivatives of $\hat{N}$, the surface normal vector measured in the lab frame $\Sigma(X,Y,Z)$, evaluated with the laboratory frame co-ordinates X, Y, Z.

IV. RESULTS

To assess the utility of the developed immersed boundary cMCLBE scheme, we now evaluate the resultant steady state vesicle data, as well as investigate and validate vesicle dynamics.

Introducing the equilibrium steady state vesicle data corresponding to zero physical flow. We first check for adequate interfacial resolution by appealing to Gauss-Bonnet theory, ensuring the immersed boundary force is appropriately communicated in the interfacial region throughout simulation. Steady state surface metrics are then discussed, as well as quantification of equilibrium vesicle shape through use of the inertia tensor, determining the accuracy of the equilibrium vesicle shapes produced. Then, a sampled phase-space characterised by non-dimensional groups and force constants discussed in Sec. III B 4 is presented, highlighting the range of steady state vesicle outcomes produced by varying the surface tension and bending rigidity.

Moving then onto vesicles dynamics, the method is validated by shearing the vesicle and comparing against experimental data, with velocity streamlines through a slice of a sheared bicuspid vesicle also being reviewed. The elasticity of the vesicle is demonstrated in the vesicles shear recovery, and then the stability and dynamics of the method is further tested via interaction with another vesicle. Through the dynamical data, we aim to demonstrate the stability, accuracy and robustness of the method at high shear rates, far from mechanical equilibrium.

In simulation, we initialise a spherical vesicle with defined cross-sectional area A_{sphere}, and then deflate with parameter Υ (i.e. increase the surface area by setting a target surface area $A_0 = \Upsilon A_{\text{sphere}}$) which is then conserved by the area conserving force (see Sec. III B). In tandem with the area conserving force, the bending and surface tension forces then form the vesicle into the desired shape, dictated by immersed boundary force constants (again see Sec. III B). Figure 1 shows the initial sphere, which is deflated and forced into a bicuspid by the described method.

A. Surface and Shape Metrics

We quantify the zero physical flow, vesicle shapes at mechanical equilibrium, produced by our immersed boundary force distribution, when applied within cMCLBE simulation, using our surface curvature measurement method of Sec. III C. We also consider the commensurate, induced micro-current activity. Throughout, we define the vesicle surface by the level set $\rho^N = 0$.

1. Interface and Simulation Resolution

We return to the matter of adequate resolution of the interfacial region for accurate immersed boundary force (see Sec. III B 4). Of course, this issue is closely related to the matter of grid independence. Here, we check for adequate resolution first by examining the vesicle surface curvature using the Gauss-Bonnet theorem, then by conducting grid convergence test.

Compliance with the Gauss-Bonnet theorem provides an elegant check on the level of surface resolution required. The surface integral of the Gaussian curvature was defined and computed as follows:

$$\langle K \rangle_{\text{S}} = \iint_S K\, dA \rightarrow \frac{\sum K(\underline{r}^*)(1-\rho^N(\underline{r}^*)^2)}{\omega}, \tag{63}$$

where the summations are taken over the simulation node set $\underline{r}^*$, symbol S denotes the membrane surface and parameter ω denotes the width of the interface (the number of MCLBE nodes, measured in the interface normal direction, for which $\rho^N < 1$). For segregation parameter $\beta = 0.67$, this width is $\omega = 2.984905565$,[23]. From the Gauss-Bonnet Theorem in $\mathbb{R}^3$,[39], we expect $\langle K \rangle_{\text{S}} = 4\pi$. Using definition Eq. (63), for the vesicles considered in the data of Table V, with initial radius $R = 15$ lattice units, we compute:

$$\frac{\langle K \rangle_{\text{S}}}{\pi} = 4.03, \tag{64}$$

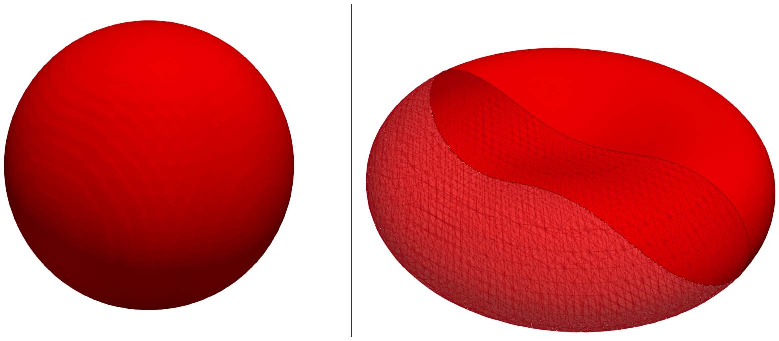


FIG. 1: Schematic depicting the process of deflation of the initial spherical vesicle (**Left**), and applying appropriate immersed boundary force resulting in a bicuspid vesicle (**Right**). The bicuspid vesicle (**Right**) is semi-transparent to help highlight the bicuspid profile. Here, the deflation value is $\Upsilon = 1.31$, and the target area of the bicuspid is thus related to the initial area of the spherical vesicle as such: $A_0 = \Upsilon A_{\text{sphere}}$.

which verifies our analysis of Sec. III C and $< 1\%$ accuracy of surface metrics. We remark, here, that MCLBE automatically provides a de facto automatically adaptive interface mesh, which means that resolution in the interfacial region will continue to be sufficient under even extreme deformation. Thus, membrane physics will be accurately enforced throughout dynamic simulation.

Our parameterisation for an initially un-deflated vesicle, radius $R = 15$, represents a balance between computational efficiency and adequate resolution. Figure 2 records data from a parameterisation for an oblate vesicle (see Sec. IV B), in which spatial resolution was scaled by factors of 1, 1.1666, 1.5 (coarse, medium, fine hereafter) —the coarse (fine) resolution corresponding to simulation with initial un-deflated initial vesicle, radius in the range $R \in [15, 22.5]$. Tab. III is a grid convergence study after Roache[40] and Celik[41], showing the grid convergence index (GCI hereafter) relative to maximum height of the sections presented in Fig. 2 i.e. the radius of the dumbbell ends. Data for all three of our resolutions, coarse, medium and fine, clearly correspond. These results give a GCI for our fine resolution of 0.39% , which is within 5% for the coarse resolution (relative to Celik's equivalent, extrapolated (infinite resolution) maximum height parameter value 8.37,[41]). Clearly, this 'internal' check complies well with our independent, analytic, 'external' verification, using Gauss-Bonnet theory (Eq. (64)). Accordingly, the grid convergence data in Tab. III permit the use of an initial drop radius of $R = 15$ (coarse resolution) for our simulations. Of course, this choice reduces computational expense.

2. Surface Metrics

Figure 3 shows surface variation of several surface metrics. Most important, it shows mean curvature H (panel (B)) and Gaussian curvature K (panel (B)) for a simulation parameterised to recover an oblate, bicuspid vesicle in equilibrium. Clearly, both H and K have qualitatively similar variation over the surface, exhibit expected trends and have values in line with their definitions in Eq. (60) (H is an order of magnitude larger than K). Panel (A) shows the magnitude of the velocity micro-current, $|\underline{u}|$ (see below), panel (D) the magnitude of the total immersed boundary force, $|\underline{F}^{(t)}|$, panel (E) the modulus of the bending and surface tension forces $|\underline{F}^{(b)} + \underline{F}^{(l)}|$ and panel (F) the magnitude of the area conservation force, $|\underline{F}^{(a)}|$. Any complex force density will excite spurious or micro-current flow with a speed $|\underline{u}|$ which scales with $|\underline{F}^{(t)}|$ (see panel (D))[42]. In flow, one could reduce micro-current noise by seeking, for a chosen regime of flow, a *minimum parameterisation* (smallest values of σ, α and κ_B etc.). Unfortunately, a minimum parameterisation is more illusive than in the case of a simple drop, for it is apparent in Sec. III B 4 that one cannot simply scale the ratio $\frac{\sigma}{\kappa_B}$ whilst maintaining a constant vesicle shape deformation. In the following, we choose appropriate σ and κ_B values by matching our model to experimental data (see Sec. IV C) and by use of the phase-diagram (see Sec. IV B).

Resolution designation	Scale factor	Maximum height	Relative estimated % (GCI)
Coarse ($R = 15$)	1	8.744	-4.46
Medium ($R = 17.5$)	1.1666	8.511	-1.68
Fine ($R = 22.5$)	1.5	8.402	-0.39

TABLE III: Grid convergence study data. Values are based on the (scaled) maximum height of the section through an oblate vesicle (the radius of its dumbbell ends; see Fig. 2), for three different mesh resolutions, designated coarse, medium and fine. Coarse resolution is achieved using an un-deflated vesicle radius $R = 15$ lattice units, medium resolution is achieved by multiplying R by 1.1666; for fine resolution multiply by 1.5.

3. Shape Metrics

Suitable vesicle shape metrics appropriate for steady-state shapes are considered as a prelude to Sec. (IV B). To quantify equilibrium vesicle shape, it is convenient to use the symmetric nine-component inertia tensor[43,44]:

$$\underline{\underline{I}} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

with computable components defined as:

$$\begin{aligned} I_{\alpha\beta} &= \int \rho_r(||\underline{r}||^2\delta_{\alpha\beta} - x_\alpha x_\beta)\underline{d^3r} \\ &\rightarrow \Sigma_{\underline{r}^*}\rho_r(\underline{r}^*)(||\underline{r}^*||^2\delta_{\alpha\beta} - x^*_\alpha x^*_\beta), \end{aligned} \tag{65}$$

where, α, $\beta \in \{1,2,3\}$ and, of course, (x, y, z) correspond to (x_1,x_2,x_3). Again, summation is taken over all discrete node positions, $\underline{r}^*$, within the simulation. The eigenvalues of $\underline{\underline{I}}$, denoted by $\lambda_1, \lambda_2, \lambda_3$ correspond to the principal moments of inertia of the vesicle. Figure 4 shows their evolution from an initially spherical vesicle to a final prolate equilibrium shape. (For the simulation parameterisation, refer to the caption.) Clearly, for this shape, degenerate eigenvalues λ_2 and λ_3 correspond to the two principal axes of rotation perpendicular to the elongated direction, the elongated direction being characterised by the smaller eigenvalue, λ_1. For other steady-state shapes, two eigenvalues are degenerate (e.g oblate shapes, bicuspids or torus shapes). For this subset, it is possible to quantify equilibrium vesicle shape using a single, dimensionless ratio $\frac{\lambda_1}{\lambda_3} = \frac{\lambda_2}{\lambda_3}$. Table IV shows the three eigenvalues of the inertia tensor corresponding to an equilibrium **(A)** oblate (torus) vesicle and **(B)** prolate vesicle. We note correct degeneracy for the vesicles, with the degenerate eigenvalues being within 1.5% (relative to λ_1) for **(A)**, and within 0.1% (relative to λ_2) for **(B)**. Tab. IV implies that (i) quantitative vesicle shape is accurate, (ii) classification of vesicle shape, using the inertia tensor eigenvalues, is possible for multiple shapes and (iii) the simulations employ resolution sufficient to recover expected eigenvalue degeneracies. Note however, a scalar shape metric is inadequate for the classification of asymmetric vesicle profiles.

TABLE IV: The three eigenvalues of the Inertia Tensor (λ_1,λ_2,λ_3), corresponding to an equilibrium oblate (*torus*) vesicle **(A)** and prolate vesicle **(B)**. Note, all eigenvalue values in the table are $\times 10^6$. For information on shape classification, see Tab. V.

(A)

Eigenvalues of the Inertia Tensor: Oblate		
λ_1	λ_2	λ_3
1.6898	1.7157	2.9444

(B)

Eigenvalues of the Inertia Tensor: Prolate		
λ_1	λ_2	λ_3
0.55612	6.0105	6.0121

B. Steady-state equilibrium phase space

Following confirmation of shape metrics and resolution, we now explore the range of vesicle shapes within a sampled phase-space. In zero flow, only shape matters. We catalogue a representative sample of the equilibrium vesicle shapes, produced by our method. All data presented in this section have $|\underline{F}^{(a)}| \approx \max\left(|\underline{F}^{(b)}|, |\underline{F}^{(l)}|\right)$.

In the developed model, the vesicle steady-state shape is determined by three parameters: deflation (Υ), bending rigidity (κ_B) and surface tension (σ). The scalars span the three-dimensional space, considered in Sec. III B 4, where we argue that one can fix deflation parameter $\Upsilon = 1.31$ in order to confine investigations to a space spanned by κ_B and σ. The value, $\Upsilon = 1.31$, was chosen based on prior art[45] to prioritise bicuspids in the sequel by using a surface area to volume ratio which had been shown to recover a bicuspid profile for $H_0 = 0$. Note, however, we can recover bicuspids with lower or higher deflation values, smaller values being favoured due to computational performance gains, arising from the ability to use a smaller lattice and yet still retain sufficient resolution.

Table V presents a sampled phase-space, showing a range of steady state outcomes (see the caption for the simulation parameters). Here, a range of bending rigidity and surface tension parameters are tested. These data show some shape outcomes have a high degree of exploitable symmetry, whereas others do not. The data in the first row of Tab. V ($\kappa_B = 0.01$) show that, even with a restricted parameterisation (κ_B fixed), a range of shapes with manifestly variable symmetry emerge. The vesicle shapes depicted in Tab. V can be categorised as (A)..(C) as follows: (A) Oblate vesicles, where the shape profile is short about the axis of rotation, such as the desired bicuspid vesicle (i.e., $\kappa_B = 1.0$, $\sigma = 0.1$) and the torus vesicle ($\kappa_B = 0.1$, $\sigma = 1 \times 10^{-3}$). Note, shapes with single and multiple holes are reported throughout vesicle literature[46,47]. (B) Prolate vesicles, which are long about the axis of rotation ($\kappa_B = 0.25$, $\sigma = 1 \times 10^{-1}$). (C) Non-axially symmetric vesicles, such as the *starfish* vesicle[48],[49] ($\kappa_B = 0.01$, $\sigma = 1 \times 10^{-5}$). It should be noted that Tab. V represents a coarse grained phase space where there are shape outcomes between these selections that are not presented.

It is appropriate to show that our method can recover a bicuspid of relevant shape (see Eq. (66) below) i.e., that it can be parameterised to represent a human erythrocyte. Put another way, we validate a stationary cMCLBE vesicle against precise human shape. Consider the bicuspid vesicle shown in Tab. V, for parameterisation $\kappa_B = 0.25, \sigma = 1 \times 10^{-5}$. We compare this against the parametric expression derived from experimental data, by Evans and Fung[50], as long ago as 1972. Taking the $x-y$ plane to span the equatorial plane of the vesicle (and the z-axis parallel with its axis of rotation), Evans and Fung give:

$$z = \frac{R_0}{2}\left[1 - \frac{r^2}{R_0^2}\right]^{\frac{1}{2}}\left[C_0 + C_1\frac{r^2}{R_0^2} + C_2\frac{r^4}{R_0^4}\right], \tag{66}$$

where, for human erythrocytes, $R_0 = 3.91\mu$m, $C_0 = 0.81$, $C_1 = 7.83$, $C_2 = -4.39$ and, off course, $r = \sqrt{x^2+y^2}$. R_0 is the size parameter, shape is determined by the constants

$C_0, .., C_2$.

The result of a comparison with Eq. (66) is shown in Fig. 5, where we have fitted only on size parameter R_0 in Eq. (66). Data in Fig. 5 show by orange circles the shape profile of the top right quadrant of a general (r,z) section, taken through our axially symmetric cMCLBE bicuspid vesicle (one containing its axis of symmetry); alongside the shape profile calculated by Evans and Fung[50], which is shown by the solid line. Clearly, the cMCLBM data fit that of the parametric expression well, providing a verification of correct biological vesicle shape at the chosen parametrisation.

We confirm that relevant and accurate erythrocyte vesicle shapes may be recovered from our cMCLBE algorithm.

C. Vesicle Shear

To verify our model's ability to correctly reproduce physically accurate vesicle dynamics, we present results for a bicuspid vesicle exposed to a simple shear. Here, following the results shown in Tab. V, we parameterise for a bicuspid shape and evolve the simulation to equilibrium. Once equilibriated, a shear is applied in the x-z plane in opposing z directions, the axis of rotation of the vesicle being parallel to the x axis. Fig. 6 superposes streaklines of steady flow and the sectioned perimeter of a vesicle (white) taken through the centre of mass in the y-z plane, i.e., the figure shows the flow in the y-z plane in and around the centre of the bicuspid. This result suggests correct physical flow with the vesicle deforming due to the applied shear. The internal circulation and the external motion (with streak-lines conforming to the vesicle perimeter) are inline with the operation of a kinematic condition.

To provide further conformation of our methodology, we compare our *in-silico* results with experiment, using the process to calibrate parameters σ and κ_B to replicate real RBC dynamics. The results of models developed by both[11] and[19] were compared and validated against Wheeler's experiment[51], in which a vesicle is subject to a simple, uni-directional shear. We adopt their approach. A Wheeler experiment positions a RBC in a fixed velocity shear flow, where the shear planes lie perpendicular to that of the axis of symmetry of the RBC. To quantify the results of the experiment, the shear flow rate (ranging between 15-200 s^{-1}) is varied and a deformation index calculated for the corresponding shear flow rate. The deformation index (DI) is defined as such:

$$DI = \frac{(D_{\max}/D_0)^2 - 1}{(D_{\max}/D_0)^2 + 1} \tag{67}$$

where $D_{\max}$ is the maximal diameter once the vesicle has deformed at given shear rate, and D_0 is the initial diameter of the bicuspid vesicle in mechanical equilibrium.

Fig. 7 shows our model (red) compared with three other models and the experimental data of Yao et al.[51]. Clearly, our model is generally consistent with all these data. As expected, the magnitude of the immersed boundary force dictates the sensitivity of the vesicle to shear-deformation, with increased sensitivity at a lower force magnitude. Accordingly, one can adjust the vesicle response to flow stresses by tuning parameters κ_B and σ. Our data show an appropriate, but not necessarily optimum, parameterisation of κ_B and σ. We show a distributed (shaded) region which bounds variation in our simulation output DI due to our value of D_0, which is an average, as follows. Our principal error in Fig. 7 is due to a small discrepancy between the semi-minor and semi-major diameter of the equilibrium vesicle (which may be reduced at the expense of additional resolution, note), and to the κ_B and σ chosen (as discussed above). Our data in Fig. 7 correspond to an average of these diameters. We remark that our model could accommodate more than 8 times the shear rates than shown in Fig. 7, which highlights the robustness of our single-framework model. However, the test does not provide information on unsteady response, such as may be obtained most simply from a shear-recovery test by considering the dynamics of the process of recovering the mechanical, bicuspid equilibrium shape when the shear is removed.

D. Shear Recovery

In Sec. IV C, we validated our model's ability to simulate the steady, shear-induced deformation of a vesicle. Clearly, the unsteady dynamic response is also of great interest. Accordingly, we proceed to consider the recovery of the vesicle from the shear by shear-deforming a vesicle, then switching off the applied shear and allowing the vesicle to relax under pseudo-elastic forces.

Figure 8 shows the evolution of the eigenvalues of the inertia tensor for when a bicuspid is exposed to a shear perpendicular to its axis of rotational symmetry then released to recover. Here, the bicuspid steadily deforms, the shear being released at 80000 lattice time steps. The dashed section of the eigenvalue locus corresponds to the time when the shear is applied. Fig. 9 shows surface plots ($\rho^N = 0$) at increasing times, relating to the data shown in Fig. 8. Fig's. (8, 9) verify broad reversibility as follows: (i) our system returns to its original bicuspid shape, i.e., the inertia tensor eigenvalues return to pre-shear values, (ii) the surface plots at 40000 and 110000 lattice time steps are identical. Furthermore, the cross-sectional area of the vesicle remained almost constant throughout simulation with a relative fluctuation of less than 0.1%. We observe that our model can handle high levels of deformation, remaining stable, and still return to a correct equilibrium shape, suggesting high stability when far from mechanical equilibrium as well as its ability to replicate vesicle behaviour qualitatively.

E. Vesicle - Vesicle Interaction

The proposed methodology scales to any number of erythrocytes using previous cMCLBE methodologies[25]. Extended simulations are clearly beyond this scope of the present methodological account. However, a verification of our method's credentials in this respect are of central significance. Therefore, we simulate two interacting vesicles propelled by opposing buoyancy forces to move past each other. This simple test should serve to verify that the method is stable for *aggressive* vesicle-vesicle contacts and deformations, whilst appropriately capturing the dynamics and physics (see below) of vesicle-vesicle interactions—at least qualitatively. Accordingly, two bicuspids were initialised with their axis of rotational symmetry parallel but offset (by an impact parameter), being separated by an appropriate initial distance. The buoyancy force was then applied (in opposing directions) parallel with their axes of rotational symmetry, propelling the vesicles towards each other. Clearly, the complex dynamics of the collision depends upon the magnitude of the buoyancy force, the impact parameter and the vesicles' parameterisation. We note that lubrication forces are emergent within the method and that no additional interface tracking whatsoever is necessary during this test. A variety of impacts have been simulated; the data presented here are representative of a set of outcomes, all of which were physically intuitive.

Fig. 10 presents a simulation output looking down the y-z plane in which two independent vesicles (Blue and Red) interact, deforming as they pass each other. A buoyancy force was applied to the Blue (Red) vesicles in the positive (negative) x direction. The data show snapshots at increasing time in alphabetical order (A)..(F). The vesicles deform and avoid contact under lubrication forces, their surface areas fluctuating by less than 0.5% throughout simulation. This simulation represents a violent *head on* interaction; as such it clearly confirms the ability of the method to produce physically intuitive dynamics under the most aggressive of impacts.

V. CONCLUSIONS

In this article, we develop and analyse (see below) a fully three-dimensional, algorithmically simple immersed boundary force which, when implemented within the chromodynamic multi-component lattice Boltzmann model (cMCLBE), furnishes a single framework methodology capable of simulating multiple vesicles in flow. This is achieved by enhancing the native MCLBE deformable Laplace interface with the additional surface-curvature dependant physics of bending rigidity and global surface area conservation or, equivalently, interface compressibility. The result is a three-parameter method with a continuum-scale interface governed by interfacial tension σ, bending rigidity κ_B and interface compressibility α. Our key methodological step is, perhaps, the provision of the means to compute requisite surface properties (curvatures) (i) locally, and (ii) based solely upon data routinely computed in the method's algorithm, specifically the MCLBE interface surface normals. However, the work advances what is also a computationally attractive, intuitive means to simulate multiple vesicles, the development of whose algorithm allows clarity of the model's physical content. Despite its algorithmic simplicity, our model retains high accuracy due to the well-documented benefits of cMCLBM, such as ease of parallelisation, strict mass conservation, fully arrested component (vesicle) coalescence, correct and verifiable interface advection (kinematic conditions) and automatic, adaptive interfacial meshing—even in regions of large deformation.

We present steady state, zero physical flow results which provide important confirmation of the methodology via the Gauss-Bonnet theorem and, using the eigenvalues of the inertia tensor as a shape proxy, we present a shape phase diagram at fixed vesicle deflation and zero applied flow, parameterised by κ_B and σ. Focusing on the biologically significant bicuspid vesicle, the method is assessed for a range of parameterisations using a Wheeler test[51], which yields results in good agreement with experimental data, with internal flow fields showing intuitively correct physical flow behaviour. The method is, moreover, very stable—it handles high shear rates and interactions with another vesicle with assurance.

Certainly, the methodology presented here is not as mature as other many-vesicle, multi-framework methods founded upon coupled, otherwise independent, flow and mechanics solvers (the most prominent being the HemoCell open-source code[11,12]) and we do not believe it will scale as effectively. On the other hand, unlike e.g. mass-spring models, our approach has validity far from mechanical equilibrium, which unlike continuum models founded upon boundary integral methodology, it is not restricted to certain regimes of flow, it is single framework and, hence, intrinsically complete (with lubrication forces and interface kinematics). Where smaller numbers of vesicles need to be simulated with great physical fidelity, our method may offer an attractive alternative.

Appendix A: Optimised Multi-relaxation-time scheme for forced fluids

A three-dimensional MRT collision scheme has been implemented within the chromodynamic MCLBM scheme outlined in Sec. III A, after that presented by Xu et. al.[29]. The benefits of an MRT scheme are its enhanced stability, compared to the LBGK model, with an inconsequential increase in computation time[53]. The succeeding work will follow the analysis of that presented in Ref. 29, now with the introduction of optimised modes. The collision matrix, $\underline{\underline{A}}$, is defined by its eigenvalues and eigenvectors, with relaxation taking place in modal space.
The reader is pointed to Table VI, which shows the left-row eigenvectors, properties, corresponding eigenvalues, the equilibria and sources relating to the following analysis.

A projection matrix consisting of left row eigenvectors,

$\underline{h}^{(p)}$, is defined as such:

$$\underline{\underline{M}} \equiv \left(\underline{h}^{(0)}, \underline{h}^{(1)}, \cdots, \underline{h}^{(18)}\right)^{T},$$

where:

$$\begin{aligned}\underline{\underline{M}}\,\underline{f} &= \left(m^{(0)}, m^{(1)}, ..., m^{(18)}\right)^{T} \\ &= \Big(\rho, \rho u_x, \rho u_y, \rho u_z, P_{xx}, P_{yy}, P_{zz}, P_{xy}, P_{xz}, P_{yz}, \\ &\qquad N, J_x, J_y, J_z, E_1, E_2, E_3, X_y, X_x\Big)^{T}, \qquad \text{(A1)}\end{aligned}$$

See Table VI, where $\underline{f} = (f_0, f_1, f_2, \cdots, f_{18})^T$.
Projecting matrix $\underline{\underline{M}}$ onto the evolution Eq. (2) gives the following transformation:

$$\underline{\underline{M}}\,\underline{f}^{\Psi} = \underline{\underline{M}}\,\underline{f} + \underline{\underline{M}}\,\underline{\underline{A}}\,\underline{\underline{M}}^{-1}\left(\underline{\underline{M}}\,\underline{f}^{(0)} - \underline{\underline{M}}\,\underline{f}\right) + \underline{\underline{M}}\,\underline{F}, \qquad \text{(A2)}$$

where $\underline{F}$ is the column vector whose elements are the source term, F_i, which encapsulates $\underline{F}^{(a)}, \underline{F}^{(b)}, \underline{F}^{(l)}$. Using the definition $\underline{\underline{M}}\ \underline{\underline{A}} = \underline{\underline{M}}\ \underline{\underline{\Lambda}}$, i.e. $\underline{\underline{\Lambda}} = \underline{\underline{M}}\ \underline{\underline{A}}\ \underline{\underline{M}}^{-1}$, with $\underline{\underline{\Lambda}} \equiv diag(\lambda_0,\ \lambda_1, ..., \lambda_{18})$, the projected evolution equation now decomposes into a set of forced scalar relaxations for each mode:

$$\begin{aligned}m^{(p)\Psi} &= m^{(p)} + \lambda_p\left(m^{(0)(p)} - m^{(p)}\right) + S^{(p)}, \\ S^{(p)} &= \sum_{j=0}^{Q-1} M_{pj}F_j, \quad p = 0, 1, \cdots, Q-1. \qquad \text{(A3)}\end{aligned}$$

Here, zero eigenvalues relate to physical modes subject to conservation principles.
The inverse transformation matrix can then be defined from the column vector, $\underline{k}^{(p)}$, constructed through Gram-Schmidt orthogonalisation:

$$\underline{\underline{M}}^{-1} = (\underline{k}^{(0)}, \underline{k}^{(1)}, \underline{k}^{(2)}, \cdots, \underline{k}^{(18)}) \qquad \text{(A4)}$$

$$\begin{aligned}6k_i^{(0)} &= t_i\left(12g_ic_{i\theta}^2 - 15c_{i\theta}^2 - 21c_{iz}^2 + 23 - 8g_i\right), && \text{(A5)} \\ k_i^{(1)} &= t_ic_{ix}\left(5 + g_i - 6c_{iy}^2\right), && \text{(A6)} \\ k_i^{(2,3)} &= t_ic_{i\gamma}\left(5 + g_i - 6c_{ix}^2\right), && \text{(A7)} \\ 2k_i^{(4,5)} &= t_i\Big(-5 + 11c_{i\zeta}^2 + c_{i\varepsilon}^2 + 3c_{iz}^2 + 2g_i && \text{(A8)} \\ &\quad - 2g_ic_{i\zeta}^2 - 4g_ic_{i\zeta}^2\Big), && \text{(A9)} \\ 2k_i^{(6)} &= t_i, \left(-6g_ic_{i\sigma}^2 + 3c_{i\sigma}^2 + 15c_{iz}^2 - 7 + 4g_i\right), && \text{(A10)} \\ k_i^{(7,8,9)} &= 9t_ic_{i\alpha}c_{i\beta}, && \text{(A11)} \\ 6k_i^{(10)} &= t_i\left(6c_{i\sigma}^2 - 12g_ic_{i\sigma}^2 + 12c_{iz}^2 - 8 + 11g_i\right), && \text{(A12)} \\ k_i^{(11)} &= t_ic_{ix}\left(1 - 3c_{iy}^2 + 2g_i\right), && \text{(A13)} \\ k_i^{(12,13)} &= t_ic_{i\gamma}\left(1 - 3c_{ix}^2 + 2g_i\right), && \text{(A14)} \\ k_i^{(14)} &= 3t_ic_{ix}\left(-2 + 6c_{iy}^2 - g_i\right), && \text{(A15)} \\ k_i^{(15,16)} &= 3t_ic_{i\gamma}\left(-2 + 6c_{ix}^2 - g_i\right), && \text{(A16)} \\ k_i^{(17,18)} &= t_i\Big(2 - 2c_{i\zeta}^2 - c_{i\varepsilon}^2 - 3c_{iz}^2 - 2g_i && \text{(A17)} \\ &\quad + 2g_ic_{i\zeta}^2 + 4g_ic_{i\varepsilon}^2\Big), && \text{(A18)}\end{aligned}$$

where $c_{i\theta}^2 = c_{ix}^2 + c_{iy}^2$, $\gamma \in [y,z]$ and is taken in alphabetical order, (ζ, ε) are taken in order as (x,y) and (y,x), and $\alpha, \beta \in [x,y,z]$ and are denoted in the pair order of (x,y), (x,z) and (y,z). Finally, using the inverted matrix, $\underline{\underline{M}}^{-1}$, the post collision distribution function can be converted from modal space to population space as such:

$$\begin{aligned}\underline{f}^{\Psi} = \underline{\underline{M}}^{-1}\Big(&\rho^{\Psi}, \rho u_x^{\Psi}, \rho u_y^{\Psi}, \rho u_z^{\Psi}, P_{xx}^{\Psi}, P_{yy}^{\Psi}, P_{zz}^{\Psi}, P_{xy}^{\Psi}, P_{xz}^{\Psi}, P_{yz}^{\Psi}, \\ &N^{\Psi}, J_x^{\Psi}, J_y^{\Psi}, J_z^{\Psi}, E_1^{\Psi}, E_2^{\Psi}, E_3^{\Psi}, X_y^{\Psi}, X_x^{\Psi}\Big)^{T}. \qquad \text{(A19)}\end{aligned}$$

Appendix B: Computation of Vesicle Surface Principal Curvatures

Following the methodology presented in Sec. (III C) for calculating the principal curvatures of the surface from the surface normal, we now, motivated by computational efficiency and the minimisation of numerical error, seek to transform the derivatives in Eq. (62) into derivatives of $\hat{N}$, the surface normal vector measured in the lab frame $\Sigma(X,Y,Z)$, evaluated with the laboratory frame co-ordinates X, Y, Z. Recalling that:

$$\hat{e}_z = \left([\hat{N}_X]_P, [\hat{N}_Y]_P, [\hat{N}_Z]_P\right). \qquad \text{(B1)}$$

When $[\hat{N}_x]_P$ and $[\hat{N}_y]_P$ in the expressions in Eq. (54) are both small (for example, near the poles of a spherical vesicle addressed using geophysical co-ordinates), the relative error in $\hat{e}_x$ and $\hat{e}_y$ will increase. Therefore, to minimise numerical error it is necessary to carefully consider the choice of frame Σ. Three straightforward choices of local co-ordinate system $\Sigma(x,y,z)$, denoted $\Sigma_1(x_1,y_1,z_1), .., \Sigma_3(x_3,y_3,z_3)$, are as follows. For the local x axis:

$$\begin{aligned}\hat{e}_{x_1} &= \frac{1}{\sqrt{1 - [N_Z]_P^2}}\left(-[\hat{N}_Y]_P, [\hat{N}_X]_P, 0\right), \\ \hat{e}_{x_2} &= \frac{1}{\sqrt{1 - [N_X]_P^2}}\left(0, [\hat{N}_Z]_P, -[\hat{N}_Y]_P\right), \\ \hat{e}_{x_3} &= \frac{1}{\sqrt{1 - [N_Y]_P^2}}\left([\hat{N}_Z]_P, 0, -[\hat{N}_X]_P\right), \qquad \text{(B2)}\end{aligned}$$

and for the corresponding right-handed $\hat{e}_{y_i} = \hat{e}_z \times \hat{e}_{x_i}$:

$$\begin{aligned}\hat{e}_{y_1} &= \frac{1}{\sqrt{1 - [N_Z]_P^2}} \times \\ &\quad \left(-[\hat{N}_X\hat{N}_Z]_P, -[\hat{N}_Y\hat{N}_Z]_P, [(1 - \hat{N}_Z^2)]_P\right), \\ \hat{e}_{y_2} &= \frac{1}{\sqrt{1 - [N_X]_P^2}} \times\end{aligned}$$

$$\Big([\hat{N}_X^2-1]_P,[\hat{N}_X\hat{N}_Y]_P,[\hat{N}_X\hat{N}_Z]_P\Big),$$

$$\hat{e}_{y_3}=\frac{1}{\sqrt{1-[N_Y]_P^2}}\times$$
$$\Big(-[\hat{N}_X\hat{N}_Y]_P,[(1-\hat{N}_Y^2)]_P,-[\hat{N}_Y\hat{N}_Z]_P\Big). \qquad \text{(B3)}$$

We consider each local co-ordinate choice $\Sigma_1..\Sigma_3$ in turn. Near P, we have:

$$\hat{n}_{x_1}=\hat{N}\cdot\hat{e}_{x_1}=\frac{-[\hat{N}_Y]_P\hat{N}_X+[\hat{N}_X]_P\hat{N}_Y}{\sqrt{1-[N_Z]_P^2}},$$
$$\hat{n}_{x_2}=\hat{N}\cdot\hat{e}_{x_2}=\frac{[\hat{N}_Z]_P\hat{N}_Y-[\hat{N}_Y]_P\hat{N}_Z}{\sqrt{1-[N_X]_P^2}},$$
$$\hat{n}_{x_3}=\hat{N}\cdot\hat{e}_{x_3}=\frac{[\hat{N}_Z]_P\hat{N}_X-[\hat{N}_X]_P\hat{N}_Z}{\sqrt{1-[N_Y]_P^2}}. \qquad \text{(B4)}$$

$$\hat{n}_{y_1}=\hat{N}\cdot\hat{e}_{y_1} \qquad \text{(B5)}$$
$$=\frac{-[\hat{N}_X\hat{N}_Z]_P\hat{N}_X-[\hat{N}_Y\hat{N}_Z]_P\hat{N}_Y+[(1-\hat{N}_Z^2)]_P\hat{N}_Z}{\sqrt{1-[N_Z]_P^2}},$$
$$\hat{n}_{y_2}=\frac{[\hat{N}_X^2-1]_P\hat{N}_X+[\hat{N}_X\hat{N}_Y]_P\hat{N}_Y+[\hat{N}_X\hat{N}_Z]_P\hat{N}_Z}{\sqrt{1-[N_X]_P^2}},$$
$$\hat{n}_{y_3}=\frac{-[\hat{N}_X\hat{N}_Y]_P\hat{N}_X+[(1-\hat{N}_Y^2)]_P\hat{N}_Y-[\hat{N}_Y\hat{N}_Z]_P\hat{N}_Z}{\sqrt{1-[N_Y]_P^2}},$$

and, accordingly, we have the following local operator equivalences:

$$\frac{\partial}{\partial x_1}=\hat{e}_{x_1}\cdot\underline{\nabla}_X$$
$$=\frac{1}{\sqrt{1-[N_Z]_P^2}}\left(-[\hat{N}_Y]_P\frac{\partial}{\partial X}+[\hat{N}_X]_P\frac{\partial}{\partial Y}\right),$$
$$\frac{\partial}{\partial x_2}=\frac{1}{\sqrt{1-[N_X]_P^2}}\left([\hat{N}_Z]_P\frac{\partial}{\partial Y}-[\hat{N}_Y]_P\frac{\partial}{\partial Z}\right),$$
$$\frac{\partial}{\partial x_3}=\frac{1}{\sqrt{1-[N_Y]_P^2}}\left([\hat{N}_Z]_P\frac{\partial}{\partial X}-[\hat{N}_X]_P\frac{\partial}{\partial Z}\right), \qquad \text{(B6)}$$

$$\frac{\partial}{\partial y_1}=\hat{e}_{y_1}\cdot\underline{\nabla}_X$$
$$=\frac{1}{\sqrt{1-[N_Z]_P^2}}$$
$$\left(-[\hat{N}_X\hat{N}_Z]_P\frac{\partial}{\partial X}-[\hat{N}_Y\hat{N}_Z]_P\frac{\partial}{\partial Y}+[1-\hat{N}_Z^2]_P\frac{\partial}{\partial Z}\right),$$
$$\frac{\partial}{\partial y_2}=\frac{1}{\sqrt{1-[N_X]_P^2}}$$
$$\left([\hat{N}_X^2-1]_P\frac{\partial}{\partial X}+[\hat{N}_X\hat{N}_Y]_P\frac{\partial}{\partial Y}+[\hat{N}_X\hat{N}_Z]_P\frac{\partial}{\partial Z}\right),$$
$$\frac{\partial}{\partial y_3}=\frac{1}{\sqrt{1-[N_Y]_P^2}} \qquad \text{(B7)}$$
$$\left(-[\hat{N}_X\hat{N}_Y]_P\frac{\partial}{\partial X}+[1-\hat{N}_Y^2]_P\frac{\partial}{\partial Y}-[\hat{N}_Y\hat{N}_Z]_P\frac{\partial}{\partial Z}\right).$$

Employing Eqs. (B4, B6) and some algebra, we can now straightforwardly find expressions for the derivative $\left[\frac{\partial\hat{n}_x}{\partial x}\right]_P$, evaluated in each of frames $\Sigma_1..\Sigma_3$:

$$\left[\frac{\partial\hat{n}_{x_1}}{\partial x_1}\right]_P=\frac{1}{1-[\hat{N}_Z]_P^2}\times$$
$$\Bigg([\hat{N}_Y]_P^2(\hat{N}_X)_X+[\hat{N}_X]_P^2(\hat{N}_Y)_Y+$$
$$-[\hat{N}_X\hat{N}_Y]_P\Big((\hat{N}_X)_Y+(\hat{N}_Y)_X\Big)\Bigg),$$
$$\left[\frac{\partial\hat{n}_{x_2}}{\partial x_2}\right]_P=\frac{1}{1-[\hat{N}_X]_P^2}\times$$
$$\Bigg([\hat{N}_Z]_P^2(\hat{N}_Y)_Y+[\hat{N}_Y]_P^2(\hat{N}_Z)_Z+$$
$$-[\hat{N}_Y\hat{N}_Z]_P\Big((\hat{N}_Z)_Y+(\hat{N}_Y)_Z\Big)\Bigg),$$
$$\left[\frac{\partial\hat{n}_{x_3}}{\partial x_3}\right]_P=\frac{1}{1-[\hat{N}_Y]_P^2}\times$$
$$\Bigg([\hat{N}_Z]_P^2(\hat{N}_X)_X+[\hat{N}_X]_P^2(\hat{N}_Z)_Z+$$
$$-[\hat{N}_X\hat{N}_Z]_P\Big((\hat{N}_Z)_X+(\hat{N}_X)_Z\Big)\Bigg), \qquad \text{(B8)}$$

and similarly, $\frac{\partial\hat{n}_y}{\partial y}$ is expressed using Eqs. (B5, B7) as follows:

$$\left[\frac{\partial\hat{n}_{y_1}}{\partial y_1}\right]_P=\frac{1}{1-[\hat{N}_Z]_P^2}\times$$
$$\Bigg([\hat{N}_X^2\hat{N}_Z^2]_P(\hat{N}_X)_X+$$
$$+[\hat{N}_X\hat{N}_Y\hat{N}_Z^2]_P\big((\hat{N}_Y)_X+(\hat{N}_X)_Y\big)+$$
$$-[\hat{N}_X\hat{N}_Z]_P[1-\hat{N}_Z^2]_P\big((\hat{N}_Z)_X+(\hat{N}_X)_Z\big)+$$
$$+[\hat{N}_Y^2\hat{N}_Z^2]_P(\hat{N}_Y)_Y+[1-\hat{N}_Z^2]_P^2(\hat{N}_Z)_Z+$$
$$-[\hat{N}_Y\hat{N}_Z]_P[1-\hat{N}_Z^2]_P\big((\hat{N}_Z)_Y+(\hat{N}_Y)_Z\big)\Bigg),$$
$$\left[\frac{\partial\hat{n}_{y_2}}{\partial y_2}\right]_P=\frac{1}{1-[\hat{N}_X]_P^2}\times$$
$$\Bigg([\hat{N}_X^2-1]_P^2(\hat{N}_X)_X+$$
$$+[\hat{N}_X\hat{N}_Y]_P[\hat{N}_X^2-1]_P\big((\hat{N}_Y)_X+(\hat{N}_X)_Y\big)+$$
$$+[\hat{N}_X\hat{N}_Z]_P[\hat{N}_X^2-1]_P\big((\hat{N}_Z)_X+(\hat{N}_X)_Z\big)+$$

$$\begin{aligned}
&+ [\hat{N}_X^2\hat{N}_Y^2]_P(\hat{N}_Y)_Y + [\hat{N}_X^2\hat{N}_Z^2]_P(\hat{N}_Z)_Z + \\
&+ [\hat{N}_X^2\hat{N}_Y\hat{N}_Z]_P\left((\hat{N}_Z)_Y + (\hat{N}_Y)_Z\right)\Bigg),
\end{aligned}$$

$$\begin{aligned}
\left[\frac{\partial \hat{n}_{y_3}}{\partial y_3}\right]_P = &\frac{1}{1-[\hat{N}_Y]_P^2}\times \qquad \text{(B9)} \\
&\Bigg([\hat{N}_X^2\hat{N}_Y^2]_P(\hat{N}_X)_X + \\
&- [\hat{N}_X\hat{N}_Y]_P[1-\hat{N}_Y^2]_P\left((\hat{N}_Y)_X + (\hat{N}_X)_Y\right) + \\
&+ [\hat{N}_X\hat{N}_Y^2\hat{N}_Z]_P\left((\hat{N}_Z)_X + (\hat{N}_X)_Z\right) + \\
&+ [1-\hat{N}_Y^2]_P^2(\hat{N}_Y)_Y + [\hat{N}_Y^2\hat{N}_Z^2]_P(\hat{N}_Z)_Z + \\
&- [\hat{N}_Y\hat{N}_Z]_P[1-\hat{N}_Y^2]_P\left((\hat{N}_Z)_Y + (\hat{N}_Y)_Z\right)\Bigg),
\end{aligned}$$

and finally, for the cross-derivatives $\frac{\partial \hat{n}_y}{\partial x}$, we have:

$$\begin{aligned}
\left[\frac{\partial \hat{n}_{y_1}}{\partial x_1}\right]_P = &\frac{1}{1-[\hat{N}_Z]_P^2}\times \\
&\Bigg([\hat{N}_X\hat{N}_Y\hat{N}_Z]_P\Big((\hat{N}_X)_X - (\hat{N}_Y)_Y\Big) + \\
&+ [\hat{N}_Y^2\hat{N}_Z]_P(\hat{N}_Y)_X - [\hat{N}_X^2\hat{N}_Z]_P(\hat{N}_X)_Y + \\
&+ [\hat{N}_X(1-\hat{N}_Z^2)]_P(\hat{N}_Z)_Y + \\
&- [\hat{N}_Y(1-\hat{N}_Z^2)]_P(\hat{N}_Z)_X\Bigg), \\
\left[\frac{\partial \hat{n}_{y_2}}{\partial x_2}\right]_P = &\frac{1}{1-[\hat{N}_X]_P^2}\times \\
&\Bigg([\hat{N}_X\hat{N}_Y\hat{N}_Z]_P\Big((\hat{N}_Y)_Y - (\hat{N}_Z)_Z\Big) + \\
&+ [\hat{N}_Z]_P[\hat{N}_X^2-1]_P(\hat{N}_X)_Y + [\hat{N}_X\hat{N}_Z^2]_P(\hat{N}_Z)_Y + \\
&- [\hat{N}_Y(\hat{N}_X^2-1)]_P(\hat{N}_X)_Z + \\
&- [\hat{N}_X\hat{N}_Y^2]_P(\hat{N}_Y)_Z\Bigg), \\
\left[\frac{\partial \hat{n}_{y_3}}{\partial x_3}\right]_P = &\frac{1}{1-[\hat{N}_Y]_P^2}\times \\
&\Bigg([\hat{N}_X\hat{N}_Y\hat{N}_Z]_P\Big((\hat{N}_Z)_Z - (\hat{N}_X)_X\Big) + \\
&+ \hat{N}_Z[1-\hat{N}_Y^2]_P(\hat{N}_Y)_X - [\hat{N}_Y\hat{N}_Z^2]_P(\hat{N}_Z)_X + \\
&+ [\hat{N}_X^2\hat{N}_Y]_P(\hat{N}_X)_Z + \\
&- [\hat{N}_X(1-\hat{N}_Y^2)]_P(\hat{N}_Y)_Z\Bigg). \qquad \text{(B10)}
\end{aligned}$$

A choice of the local co-ordinate frame, from $\Sigma_1..\Sigma_3$ will be made on the basis of local conditions. Typically, to minimise relative error, we shall choose to use Σ_1 (say) when $\hat{N}_Z$ is the largest component of $\hat{N}$.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGEMENTS

The 1st author James Spendlove would like to thank Sheffield Hallam University for his Graduate Teaching Assistant PhD scholarship, which has financially supported the research work shown in this paper.

[1] M. Diez-Silva, M. Dao, J. Han, C. Lim, and S. Suresh, "Shape and biomechanical characteristics of human red blood cells in health and disease," MRS bulletin **35(5)**, 382–388 (2010).

[2] Y. Fung, *Biomechanics : mechanical properties of living tissues* (Springer, 1993).

[3] T. Fischer, "Shape memory of human red blood cells," Biophysical Journal **86(5)**, 3304–3313 (2004).

[4] X. Li, H. Li, H. Chang, G. Lykotrafitis, and E. Karniadakis, "Computational biomechanics of human red blood cells in hematological disorders," Journal of biomechanical engineering **139(2)**, 3304–3313 (2017).

[5] J. S. Rowlinson and B. Widom, *Molecular Theory of Capillarity* (Clarendon Press, 1982).

[6] A. Guckenberger and S. Gekle, "Theory and algorithms to compute helfrich bending forces: a review," Journal of Physics: Condensed Matter **29(20)** (2017).

[7] C. Peskin, "Numerical analysis of blood flow in the heart," Journal of Computational Physics **25(3)**, 220–252 (1977).

[8] C. Peskin, "The immersed boundary method," Acta Numerica **11**, 479–517 (2002).

[9] M. Dupin, I. Halliday, C. Care, L. Alboul, and L. Munn, "Modeling the flow of dense suspensions of deformable particles in three dimensions," Phys. Rev.E **75**, 066707 (2007).

[10] D. Fedosov, B. Caswell, and G. Karniadakis, "A multiscale red blood cell model with accurate mechanics, rheology and dynamics," Biophysical Journal **98(10)**, 2215–2225 (2010).

[11] G. Zavodszky, B.Rooij, V. Azizi, and A. Hoekstra, "Cellular level in-silico modeling of blood rheology with an improved material model for red blood cells," Frontiers in Physiology **8** (2017).

[12] G. Zavodszky, B. van Rooij, V. Azizi, S. Alowayyed, and A. Hoekstra, "Hemocell: a high-performance microscopic cellular library," Procedia Computer Science **108**, 159–165 (2017).

[13] R.M.MacMeccan, J. Clausen, G.P.Neitzel, and C. Aidun, "Simulating deformable particle suspensions using a couped lattice-boltzmann and finite-element method," Journal of Fluid Mechanics **618**, 13–39 (2009).

[14] T. Krüger, F. Varnik, , and D. Raabe, "Efficient and accurate simulations of deformable particles immersed in a fluid using a combined immersed boundary lattice boltzmann finite element method," Computers & Mathematics with Applications **61(12)**, 3485–3505 (2011).

[15] T. Klöppel and W. A. Wall, "A novel two-layer, coupled finite element approach for modeling the nonlinear elastic and viscoelastic behavior of human erythrocytes," Biomechanics and Modeling in Mechanobiology **10**, 445–459 (2011).

[16] J. WALTER, A. SALSAC, and D. BARTHES-BIESEL, "Ellipsoidal capsules in simple shear flow: prolate versus oblate initial shapes," J. Fluid Mech **676**, 318–347 (2011).

[17] E. FOESSEL, J. WALTER, A.-V. SALSAC, and D. BARTHES-BIESEL, "Influence of internal viscosity on the large deformation and buckling of a spherical capsule in a simple shear flows," J. Fluid Mech **672**, 477–486 (2011).

[18] G. Boedec, M. Leonetti, and M. Jaeger, "Isogeometric fem-bem simulations of drop, capsule and vesicle dynamics in stokes flow," Journal of Computational Physics **342**, 117–138 (2017).

[19] C. Kotsalos, J. Latt, and B. Chopard, "Bridging the computational gap between mesoscopic and continuum modeling of red blood cells for fully resolved blood flow," Journal of Computational Physics (2019).

[20] F. Guglietta, M. Behr, L. Biferale, G. Falcucci, and M. Sbragaglia, "On the effects of membrane viscosity on transient red blood cell dynamics," Soft Matter **16**, 6191–6205 (2020).

[21] J. Ames, D. F. Puleri, P. Balogh, J. Gounley, E. W. Draeger, and A. Randles, "Multi-gpu immersed boundary method hemodynamics simulations," Journal of Computational Science **44**, 101153 (2020).

[22] C. Kotsalos, J. Latt, J. Beny, and B. Chopard, "Digital blood in massively parallel cpu/gpu systems for the study of platelet transport," Interface Focus **11**, 20190116 (2020), https://royalsocietypublishing.org/doi/pdf/10.1098/rsfs.2019.0116.

[23] I. Halliday, S. V. Lishchuk, T. J. Spencer, G. Pontrelli, and C. M. Care, "Multiple-component lattice boltzmann equation for fluid-filled vesicles in flow," Phys. Rev.E **87**, 023307 (2013).

[24] I. Halliday, S. V. Lishchuk, T. J. Spencer, G. Pontrelli, and P. C. Evans, "Local membrane length conservation in two-dimensional vesicle simulation using a multicomponent lattice boltzmann equation method," Phys. Rev.E **94**, 023306 (2016).

[25] T. Spencer, I. Halliday, and C. Care, "A local lattice boltzmann method for multiple immiscible fluids and dense suspensions of drops," Philosophical transactions. Mathematical, physical, and engineering sciences, **369**, 2255–2263 (2011).

[26] G. R. McNamara and G. Zanetti, "Use of the boltzmann equation to simulate lattice-gas automata," Phys. Rev.Lett **61**, 2930–2948 (1998).

[27] Gunstensen, Rothman, Zaleski, and Zanetti., "Lattice boltzmann model of immiscible fluids," Phys. Rev.A **43(8)**, 4320–4327 (1991).

[28] P. J. Dellar, "Nonhydrodynamic modes and a priori construction of shallow water lattice boltzmann equations," Phys. Rev.E **65**, 036309 (2002).

[29] X. Xu, K. Burgin, M. Ellis, and I. Halliday, "Benchmarking of three-dimensional multicomponent lattice boltzmann equation," Phys. Rev.E **96** **(5)**, 053308 (2017).

[30] P. J. Dellar, "Incompressible limits of lattice boltzmann equations using multiple relaxation times," J Comput. Phys **190**, 351 (2003).

[31] I. Halliday, X. Xu, and K. Burgin, "Shear viscosity of a two-dimensional emulsion of drops using a multiple-relaxation-time-step lattice boltzmann method," Phys. Rev.E **95**, 023301 (2017).

[32] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice boltzmann method," Phys. Rev.E **65**, 046308 (2002).

[33] X. He, S. Chen, and G. D. Doolen, "A novel thermal model for the lattice boltzmann method in incompressible limit," J. Comput. Phys **146**, 282 (1998).

[34] S. V. Lishchuk, C. M. Care, and I. Halliday, "Lattice boltzmann algorithm for surface tension with greatly reduced microcurrents," Phys. Rev.E **67**, 036701 (2003).

[35] U. D'Ortona, D. Salin, M. Cieplak, R. B. Rybka, and J. R. Banavar, Phys. Rev.E **51**, 3718 (1995).

[36] G. Arfken, *Mathematical Methods for Physicists* (Academic Press Inc, 1970).

[37] E. Abbena, S. Salamon, and A. Gray, *Modern Differential Geometry of Curves and Surfaces with Mathematica* (CRC Press, 2017).

[38] I. Halliday, S. Lishchuk, T. Spencer, K. Burgin, and T. Schenkel, "Interfacial micro-currents in continuum-scale multi-component lattice boltzmann equation hydrodynamics," Computer Physics Communications **219** (2017), 10.1016/j.cpc.2017.06.005.

[39] J. Oprea, *Modern Differential geometry and its applications* (Mathematical Association of America, 2007).

[40] P. J. Roache, "Perspective: A method for uniform reporting of grid refinement studies," Journal of Fluids Engineering; (United States) **116:3** (1994), 10.1115/1.2910291.

[41] I. Celik, U. Ghia, P. Roache, C. Freitas, H. Coloman, and P. Raad, "Procedure of estimation and reporting of uncertainty due to discretization in cfd applications," J. Fluids Eng. **130**, 078001 (2008).

[42] J. Spendlove, X. Xu, T. Schenkel, M. Seaton, and I. Halliday, "Chromodynamic multi-component lattice boltzmann equation scheme for axial symmetry," Journal of Physics A **53 (14)**, 145001 (2020).

[43] L. Landau and E. Lifschitz, *Fluid mechanics* (Oxford : Pergamon Press, 1987).

[44] H. Goldstein, *Classical mechanics* (Addison-Wesley, 1980).

[45] D. Boal, *Mechanics of the cell* (Cambridge University Press, 2012).

[46] V. C. Wimmer, H. Horstmann, A. Groh, and T. Kuner, "Donut-like topology of synaptic vesicles with a central cluster of mitochondria wrapped into membrane protrusions: A novel structure–function module of the adult calyx of held," Journal of Neuroscience **26 (1)**, 109–116 (2006).

[47] M. Mutz and D. Bensimon, "Observation of toroidal vesicles," Phys. Rev.A **43(8)**, 4525–4527 (1991).

[48] W. Wintz, H. Döbereiner, and U. Seifert, "Starfish vesicles," Europhysics Letters **33**, 5 (1996).

[49] X. Michalet, "Equilibrium shape degeneracy in starfish vesicles," Phys. Rev.E **76**, 021914 (2007).

[50] E. Evans and Y. Fung, "Improved measurements of the erythrocyte geometry," Microvascular research **4** (1972), 10.1016/0026-2862(72)90069-6.

[51] W. Yao, Z. Wen, Z. Yan, D. Sun, W. Ka, and L. Xie, "Low viscosity ektacytometry and its validation tested by flow chamber," J. Biomech **34**, 1501–1509 (2001).

[52] R. Skalak, A. Tozeren, R. P. Zarda, and S. Chien, "Strain energy function of red blood cell membranes," Biophysical journal **13**, 245–264 (1973).

[53] P. Lallemand and L.-S. Luo, "Theory of the lattice boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability," Phys. Rev.E **61**, 6546 (2000).

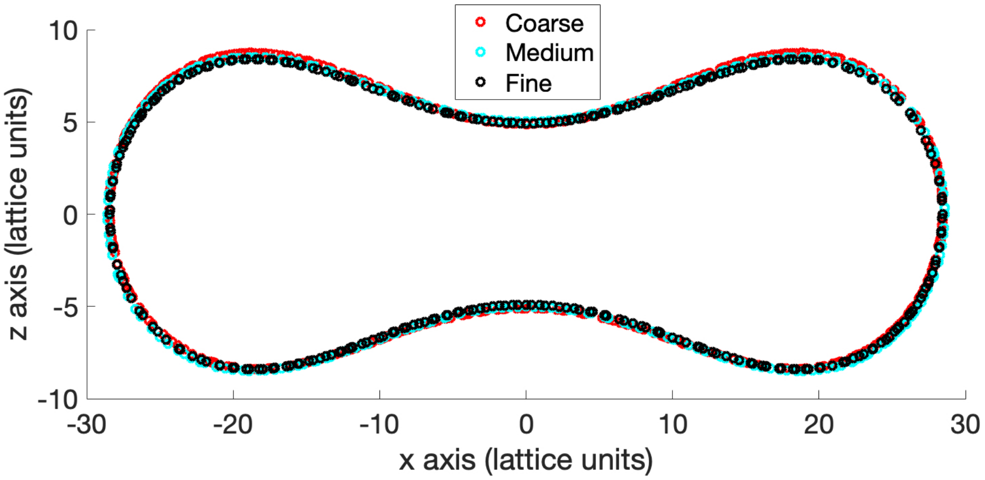


FIG. 2: The effect of resolution on equilibrium vesicle shape. Grid independence is demonstrated, here, by computing the profile of a section through an oblate vesicle, for increasing resolutions (reducing lattice link vector length). Data for three different simulation resolutions are presented with lattice scale factors of 1, 1.166, 1.333 (coarse, medium and fine), with the coarse resolution having an un-deflated initial vesicle radius of 15 lattice units. The vesicle sections were then linearly scaled to the same diameter, to facilitate visual comparison. Of course, when scaling the vesicle, the dimensionless groups in Eq. (44) must be used to adjust parameters σ and κ_B.

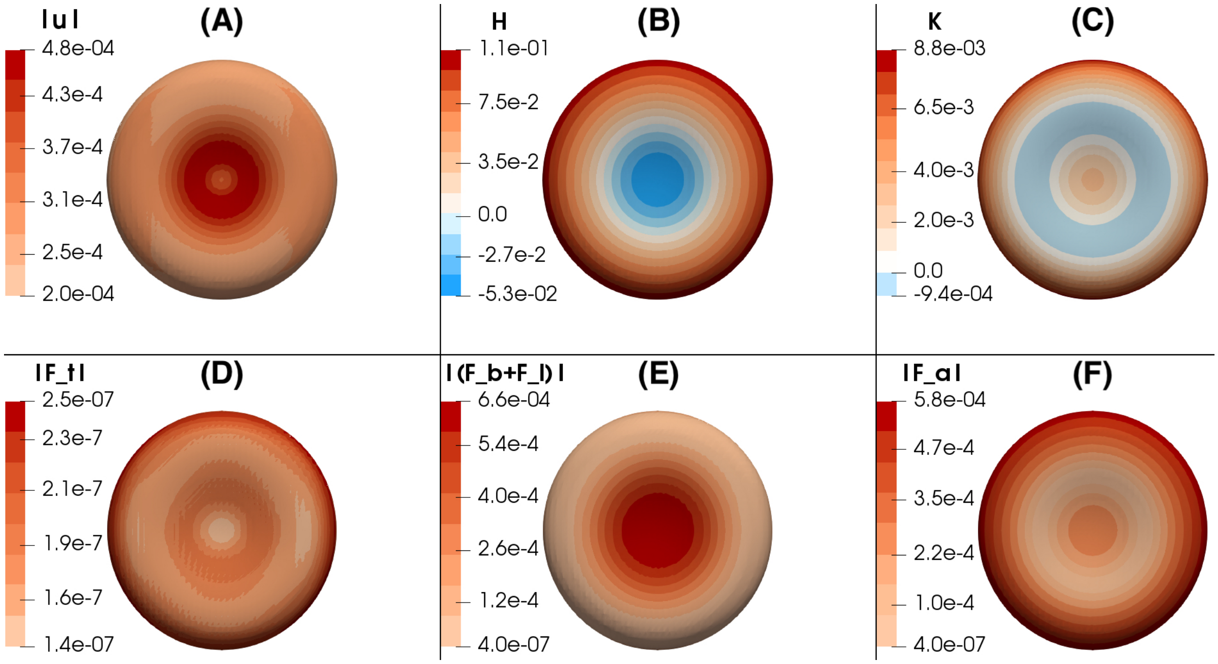


FIG. 3: Surface plots of a bicuspid vesicle over the contour $\rho^N = 0$. **(A)** Surface plot of the micro-current or magnitude of velocity $|\underline{u}|$. **(B)** Surface plot of the Mean Curvature H. **(C)** Surface plot of the Gaussian Curvature K. **(D)** Surface plot of the magnitude of the total immersed boundary force $|\underline{F}^{(t)}|$. **(E)** Surface plot of the magnitude of the bending and surface tension force $|\underline{F}^{(b)} + \underline{F}^{(l)}|$. **(F)** Surface plot of the magnitude of the area conserving force $|\underline{F}^{(a)}|$.

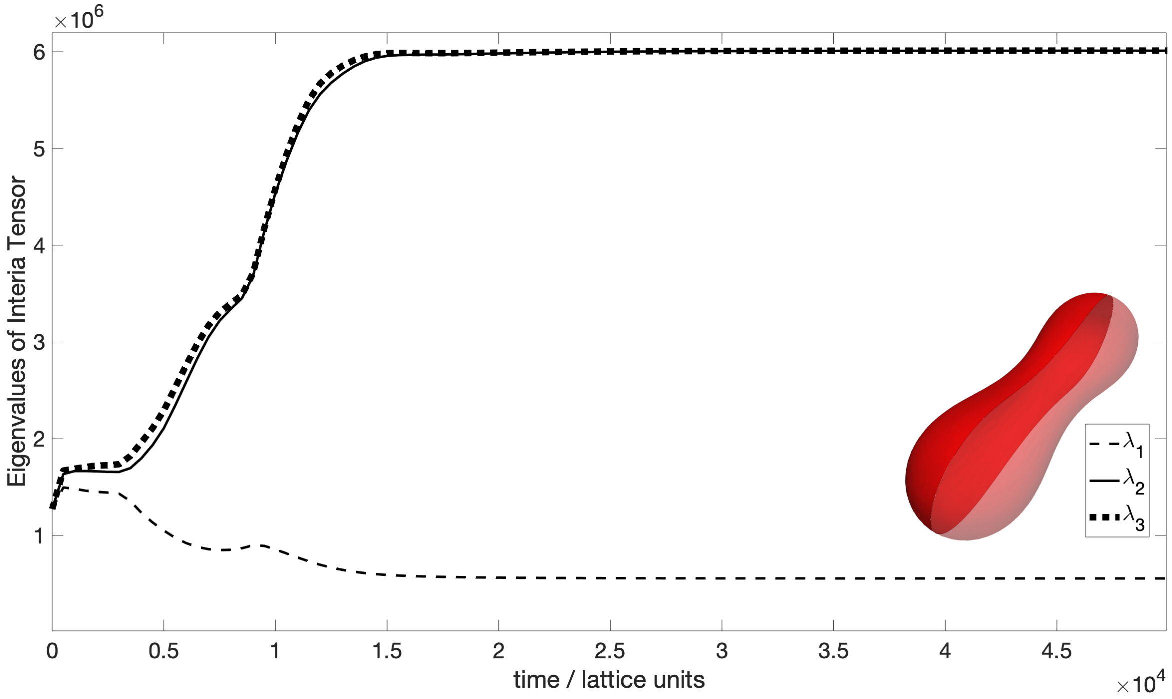


FIG. 4: Time evolution of the inertia tensor eigenvalues for a prolate vesicle, which is displayed by the sub-image (semi-transparent, to illustrate shape profile). We see divergence of the largest eigenvalues as expected. For these data, the simulation parameters are $\sigma = 0.1$, $\kappa_B = 0.25$, area increase factor $\Upsilon = 1.31$ (or system deflation), $\beta = 0.6$, initial vesicle radius= 15, lattice $= 80 \times 80 \times 80$.

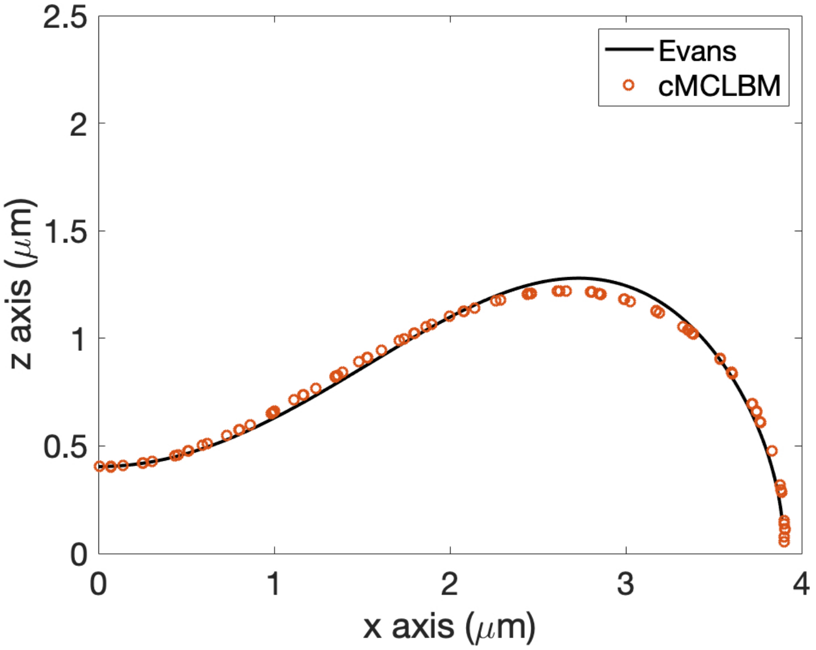


FIG. 5: Shape validation. Orange open circles show points at the centre of the cMCLBE interface, from a top-right quadrant of a vesicle section containing the axis of rotation of a three dimensional, steady state bicuspid profile. The solid black line represents fitted theory, for an identically-sized vesicle, due to Evans and Fung[50], who give a parametric expression of RBC shape profile, fitted to experimental measurement.

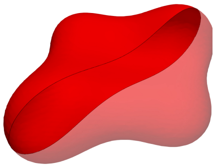
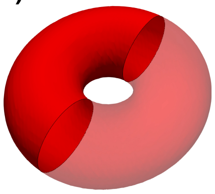
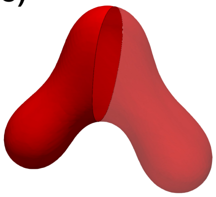
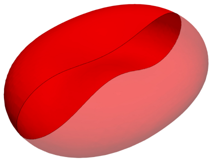
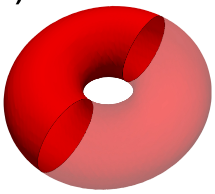
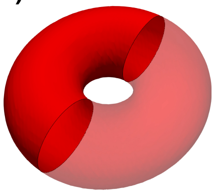
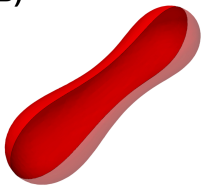
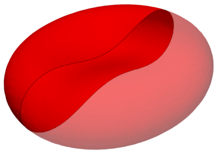
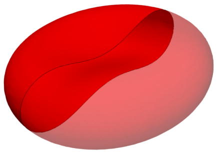
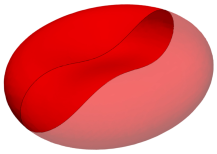
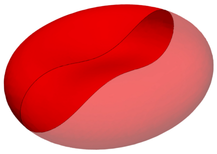
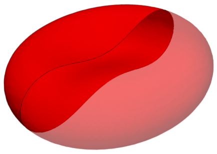
		σ		
		1×10^{-1}	1×10^{-3}	1×10^{-5}
κ_B	0.01	(C) 	(A) 	(C) 
	0.1	(A) 	(A) 	(A) 
	0.25	(B) 	(A) 	(A) 
	1.0	(A) 	(A) 	(A) 

TABLE V: A table of a sampled, pictorial phase diagram highlighting the range of vesicle profiles produced for a given bending rigidity κ_B and surface tension σ when deflation is fixed ($\Upsilon = 1.31$). **(A)** Oblate vesicles, **(B)** Prolate vesicles, **(C)** Non-axially symmetric vesicles. Here, each cell contains a semi-transparent surface plot of the vesicle, taken at contour $\rho^N = 0$.

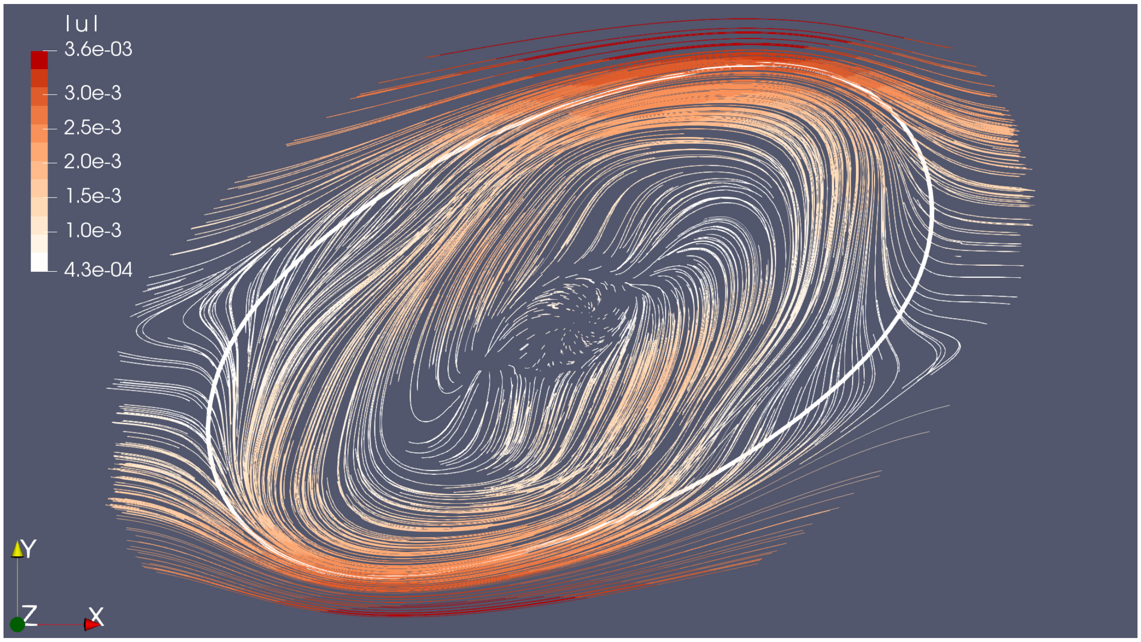


FIG. 6: A bicuspid vesicle, sheared perpendicular to the axis of symmetry in opposing z directions, corresponding to a large deformation index $DI = 29\%$ (see Eq. (67)). The white wire frame corresponds to a section of the sheared bicuspid, taken through its centre of mass, in the y-z plane. Streak-lines show interior and exterior dye paths. Streak-lines close to the boundary verify the operation of a kinematic condition. The colour map illustrates $|u|$.

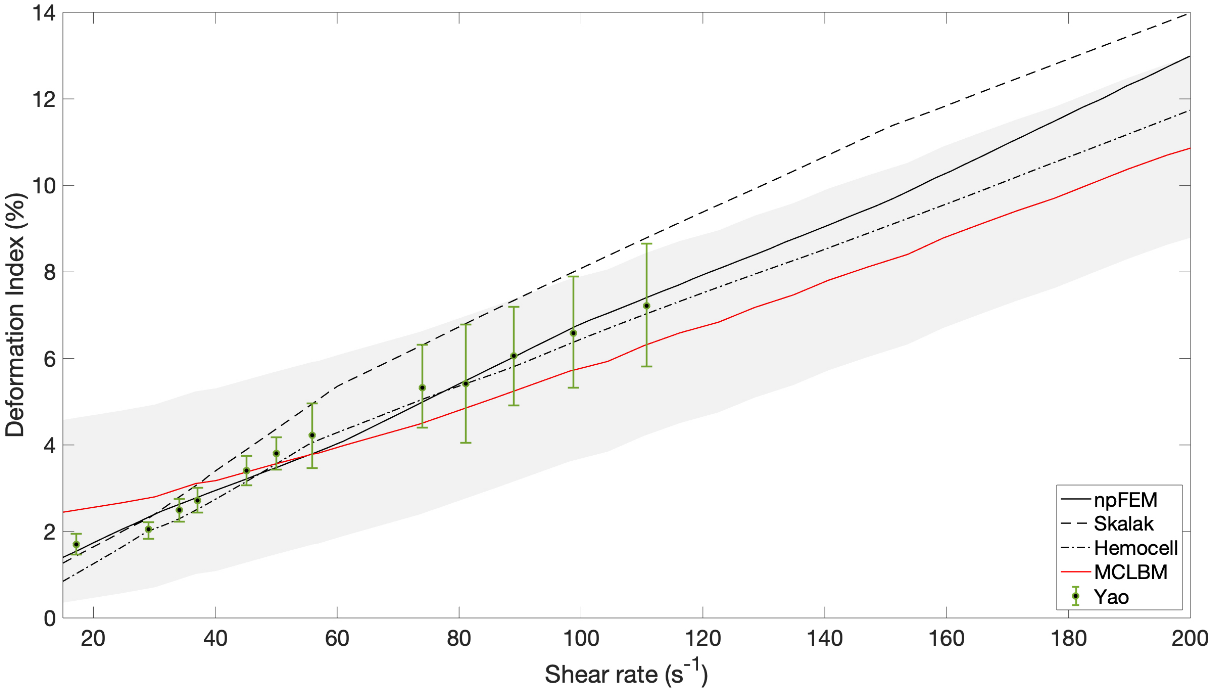


FIG. 7: Results from a single vesicle subject to a shear perpendicular to the axis of rotation over increasing shear rates, where the corresponding deformation is calculated via Eq. (67). Data show comparison of our model (Red line) with other models[11,19,52], as well as experimental data (Yao et al[51]).

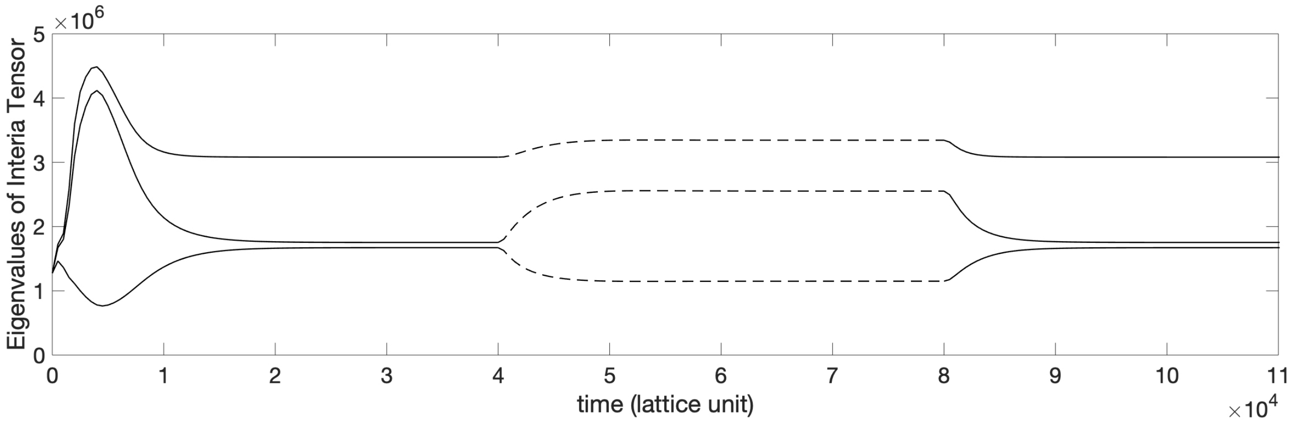


FIG. 8: Evolution of the inertia tensor eigenvalues for a bicuspid vesicle shape exposed to a uni-direction shear in opposing direction perpendicular to the axis of rotation. Here, dashed lines correspond to the period during which it was exposed to shear, starting at $40,000$ lattice time steps, and switching off at $80,000$ time steps. For these data, the simulation parameters were $\sigma = 0.00001$, $\kappa_B = 1.0$, area increase factor $\Upsilon = 1.31$ (i.e. vesicle deflation), $\beta = 0.6$, initial vesicle radius$= 15$, lattice $= 75 \times 75 \times 75$.

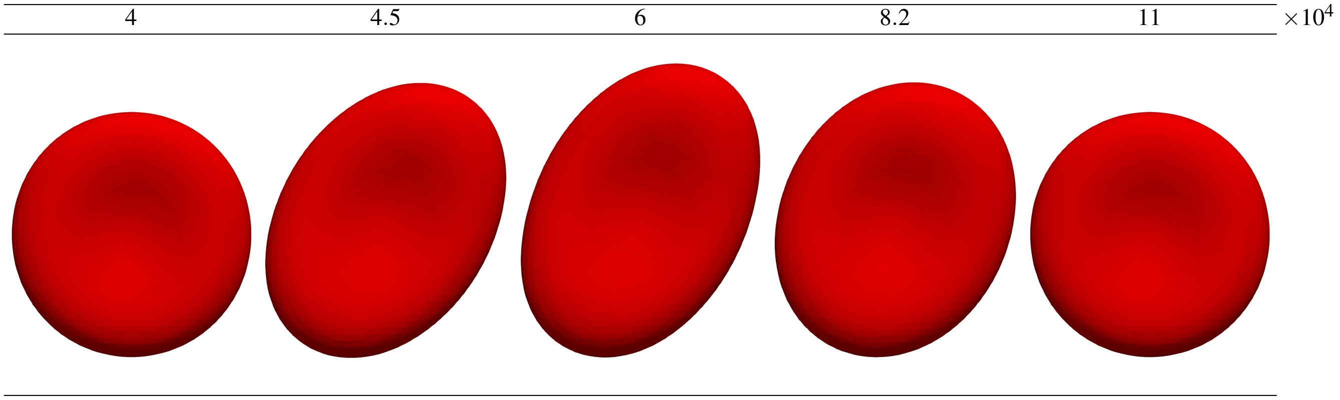


FIG. 9: Evolution surface plots of a bicuspid vesicle, exposed to and then recovering from a shear. These data relate to Fig. 8. The time corresponding to the shape is shown above the surface plots. Surface plots are taken on the level set points characterised by $\rho^N = 0$, i.e. mid-interface; the figure line of sight is down the vesicle axis of symmetry.

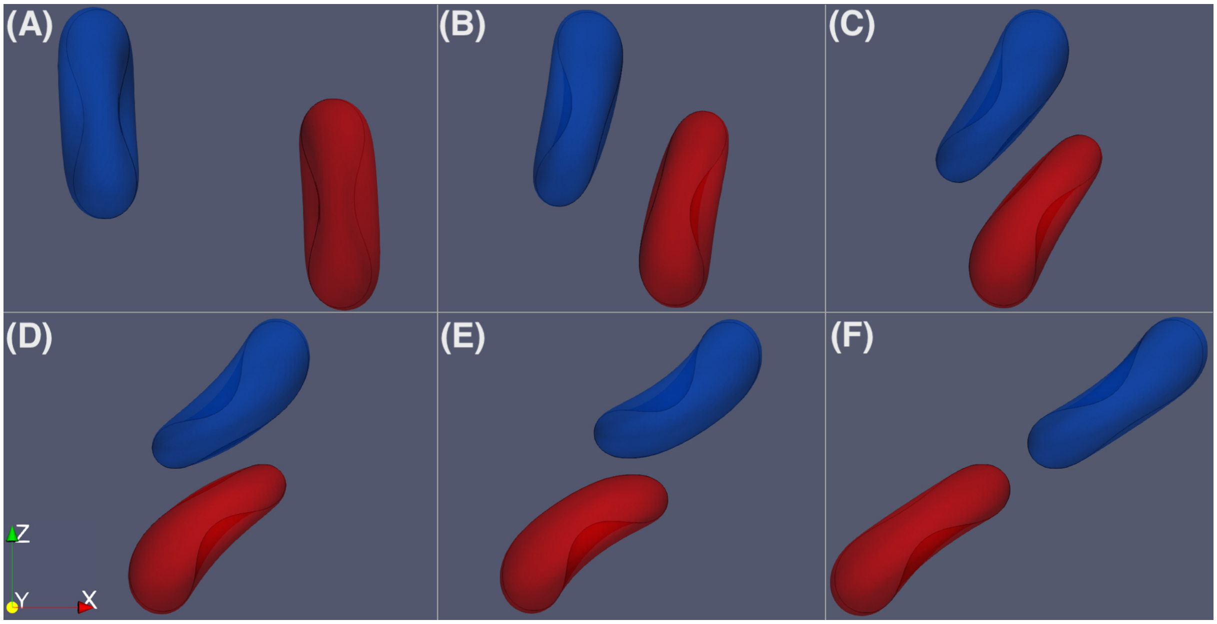


FIG. 10: Results for a vesicle-vesicle interaction, showing frames from increasing time steps (A) - (F) of two vesicles deforming and moving past each other. The line of sight is perpendicular to the scattering plane, i.e., the *y*-*z* plane; the blue (red) vesicles are propelled by a buoyancy force in the positive (negative) *x* direction. The faint black, semi-transparent line corresponds to a transect in the *x*-*z* plane through the centre of mass to help the viewer observe vesicle shape throughout vesicle-vesicle interaction.

MRT Collision Table					
mode	eigenvector component, $h_i^{(p)}$	eigenvalue, λ_p	mode, $m^{(p)}$	modal source, $S^{(p)}$	equilibrium, $m^{(0)(p)}$
$m^{(0)}$	$h_i^{(0)} = 1$	0	ρ	0	ρ
$m^{(1)}$	$h_i^{(1)} = c_{ix}$	0	ρu_x	$nF_x\Delta t$	ρu_x
$m^{(2)}$	$h_i^{(2)} = c_{iy}$	0	ρu_y	$nF_y\Delta t$	ρu_y
$m^{(3)}$	$h_i^{(3)} = c_{iz}$	0	ρu_z	$nF_z\Delta t$	ρu_z
$m^{(4)}$	$h_i^{(4)} = c_{ix}^2$	λ_4	P_{xx}	$\frac{1}{2}\left(C_{xx}+C_{xx}\right)$	$\Pi_{xx}^{(0)}$
$m^{(5)}$	$h_i^{(5)} = c_{iy}^2$	λ_4	P_{yy}	$\frac{1}{2}\left(C_{yy}+C_{yy}\right)$	$\Pi_{yy}^{(0)}$
$m^{(6)}$	$h_i^{(6)} = c_{iz}^2$	λ_4	P_{zz}	$\frac{1}{2}\left(C_{zz}+C_{zz}\right)$	$\Pi_{zz}^{(0)}$
$m^{(7)}$	$h_i^{(7)} = c_{ix}c_{iy}$	λ_4	P_{xy}	$\frac{1}{2}\left(C_{xy}+C_{yx}\right)$	$\Pi_{xy}^{(0)}$
$m^{(8)}$	$h_i^{(8)} = c_{ix}c_{iz}$	λ_4	P_{xz}	$\frac{1}{2}\left(C_{xz}+C_{zx}\right)$	$\Pi_{xz}^{(0)}$
$m^{(9)}$	$h_i^{(9)} = c_{iy}c_{iz}$	λ_4	P_{yz}	$\frac{1}{2}\left(C_{yz}+C_{zy}\right)$	$\Pi_{yz}^{(0)}$
$m^{(10)}$	$h_i^{(10)} = g_i$	λ_{10}	N	0	0
$m^{(11)}$	$h_i^{(11)} = g_ic_{ix}$	λ_{11}	J_x	0	0
$m^{(12)}$	$h_i^{(12)} = g_ic_{iy}$	λ_{11}	J_y	0	0
$m^{(13)}$	$h_i^{(13)} = g_ic_{iz}$	λ_{11}	J_z	0	0
$m^{(14)}$	$h_i^{(14)} = c_{ix}c_{iy}^2$	λ_{14}	E_1	$\frac{1}{3}\left(F_x\right)$	$E_1^{(0)} = \frac{1}{3}\rho u_x$
$m^{(15)}$	$h_i^{(15)} = c_{ix}^2c_{iy}$	λ_{14}	E_2	$\frac{1}{3}\left(F_y\right)$	$E_2^{(0)} = \frac{1}{3}\rho u_y$
$m^{(16)}$	$h_i^{(16)} = c_{ix}^2c_{iz}$	λ_{14}	E_3	$\frac{1}{3}\left(F_z\right)$	$E_3^{(0)} = \frac{1}{3}\rho u_z$
$m^{(17)}$	$h_i^{(17)} = g_ic_{iy}^2$	λ_{17}	X_y	$\left(1-\frac{\lambda_4}{2}\right)\left(F_xu_x+F_zu_z\right)$	$X_y^{(0)} = \frac{\rho}{2}\left(u_x^2+u_z^2\right)$
$m^{(18)}$	$h_i^{(18)} = g_ic_{ix}^2$	λ_{17}	X_x	$\left(1-\frac{\lambda_4}{2}\right)\left(F_yu_y+F_zu_z\right)$	$X_x^{(0)} = \frac{\rho}{2}\left(u_y^2+u_z^2\right)$

TABLE VI: Relating to the optimised MRT scheme[29], explained further in Appendix. A, containing information on the: modes, component, eigenvalues, projection, modal source and equilibrium for the implemented collision scheme.

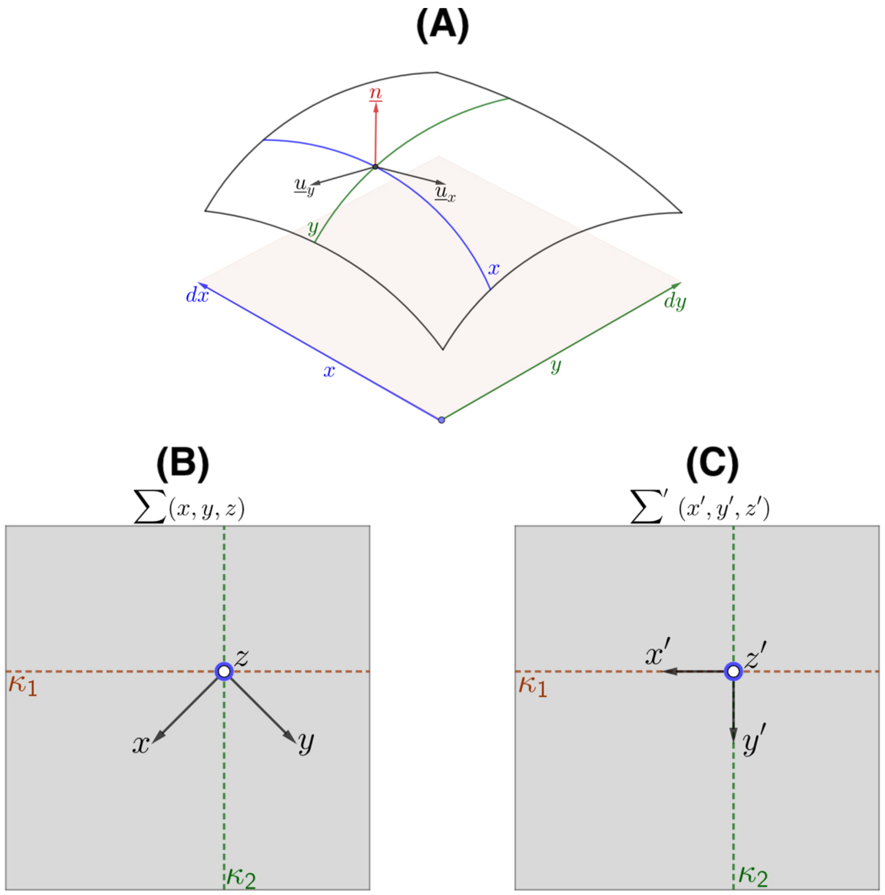


FIG. 11: Schematic to aid the differential geometry involved in sections (III B 1, III C).
(A) Excerpt of vesicle surface, which when unstrained has area $dA' = dxdy$. Position on the surface is given by $\underline{u}(x, y)$. Orthogonal tangent vectors are then calculated by partial derivatives of direction, $\underline{u}_x$ and $\underline{u}_y$, (where $\underline{u}_\alpha = \frac{\partial}{\partial \alpha}\underline{u}$). The normal at such a point is calculated via the cross of the tangent vectors, $\underline{n} = \underline{u}_x \times \underline{u}_y$ (red).
(B) and **(C)** show the surface co-ordinate frames, ($\sum(x, y, z)$ **(B)** and $\sum'(x', y', z')$ **(C)**), from a view looking down the normal to the surface (z and z'). Here, the tangent plane of frame $\sum(x, y, z)$ does not align with the principal curvatures (κ_1, κ_2) of the surface excerpt. Whereas, the tangent pane of frame $\sum'(x', y', z')$ does align with the principal curvatures of the surface excerpt.