HAL
open science

3D Numerical simulations of vesicle and inextensible capsule dynamics

Alexander Farutin, Thierry Biben, Chaouqi Misbah

To cite this version:

Alexander Farutin, Thierry Biben, Chaouqi Misbah. 3D Numerical simulations of vesicle and inextensible capsule dynamics. Journal of Computational Physics, 2014, 275, pp.539-568. 10.1016/j.jcp.2014.07.008 . hal-00841996

HAL Id: hal-00841996

https://hal.science/hal-00841996v1

Submitted on 5 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

3D Numerical simulations of vesicle and inextensible capsule dynamics

Alexander Farutin[a], Thierry Biben[b], Chaouqi Misbah[a]

[a]*Université Grenoble I/CNRS, Laboratoire Interdisciplinaire de Physique/UMR5588, Grenoble F-38041, France*
[b]*Université de Lyon, F-69000, France; Univ. Lyon 1, Laboratoire PMCN; CNRS, UMR 5586; F-69622 Villeurbanne Cedex, France*

Abstract

Vesicles are locally-inextensible fluid membranes while inextensible capsules are in addition endowed with in-plane shear elasticity mimicking the cytoskeleton of red blood cells (RBCs). Boundary integral (BI) methods based on the Green's function techniques are used to describe their dynamics, that falls into the category of highly nonlinear and nonlocal dynamics. Numerical solutions raise several obstacles and challenges that strongly impact the results. Of particular complexity is (i) the membrane inextensibility, (ii) the mesh stability and (iii) numerical precisions for evaluation of the boundary integral equations. Despite intense research these questions are still a matter of debate.

We regularize the single layer integral by subtraction of exact identities for the terms involving the normal and the tangential components of the force. In addition, the regularized kernel remains explicitly self-adjoint. The stability and precision of BI calculation is enhanced by taking advantage of additional quadrature nodes located in vertices of an auxiliary mesh, constructed by a standard refinement procedure from the main mesh. We extend the partition of unity technique to boundary integral calculation on triangular meshes: We split the calculation of the boundary integral between the original and the auxiliary mesh using a smooth weight function, which takes the distance between the source and the target as the argument and falls to zero beyond a certain cut-off distance. We provide an efficient lookup algorithm that allows us to discard most of the vertices of the auxiliary mesh lying beyond the cut-off distance from a given point without actually calculating the distances to them. The proposed algorithm offers the same treatment of near-singular integration regardless if the source and the target points belong to the same surface or not.

Additional innovations are used to increase the stability and precision of the method: The bending forces are calculated by differential geometry expressions using local coordinates defined in vicinity of each vertex. The approximation of the surface in vicinity of a vertex is obtained by fitting with a second-degree polynomial of local coordinates.

We solve for the Lagrange multiplier associated with membrane incompressibility using two penalization parameters per suspended entity: one for deviation of the global area from prescribed value and another for the sum of squares of local strains defined on each vertex. The proposed advancement is to vary the penalization parameters at each time step in such a way, that the global area of each membrane be conserved and the sum of squares of local strains be at minimum. This optimization is achieved by solving a linear system of rank three times the number of entities involved in the simulation. If no auxiliary mesh is used, the method reduces to steepest descent method thanks to the explicit self-adjointness of the regularized single-layer kernel in the boundary integral equation.

Inextensible capsules, a model of RBC, are studied by storing the position in the reference configuration for each vertex. The elastic force is then calculated by direct variation of the elastic energy. Various nonequilibrium physical examples on vesicles and capsules will be presented and the convergence and precision tests highlighted. Overall, a good convergence is observed with numerical error inversely proportional to the number of vertices used for surface discretization, the highest order of convergence allowed by piece-wise linear interpolation of the surface.

Keywords: vesicles, capsules, Stokes flow, boundary integral method, singularity subtraction

1. Introduction

Simulation of deformable and locally inextensible interfaces in Stokes flow has received an increasing attention in the last years. The upsurge of interest is motivated by understanding blood flow by taking explicitly into account blood elements (e.g. red blood cells -RBC). RBC is traditionally modeled as an inextensible visco-elastic surface containing

viscous Newtonian liquid and immersed in another viscous Newtonian liquid. In this paper, we focus on two classes of systems (i) vesicles which have a fluid and inextensible membrane that resists to bending, and (ii) inextensible capsules, which are, unlike vesicles, endowed with in-plane shear elasticity, mimicking the cytoskeleton of RBCs.

Boundary integral (BI) method represents an important tool for simulation of dynamics of deformable particles in Stokes flow, offering a very precise solution with reasonable computational cost. For example, spectral BI method[1] has asymptotic convergence that is superalgebraic in the number of spherical harmonics used for surface representation. This method is based on parametrization of the coordinates of points on the surface of a membrane by series of spherical harmonics defined on a reference sphere. For shapes that are close to a sphere, a simpler representation of the surface by a single scalar function expanded in spherical harmonics is possible[2]: the shape is parametrized by a radius function, which measures the distance from the center of mass of the closed shape to the surface in given direction. While the last parametrization is only reserved to the shapes in which a ray from the center of mass of a vesicle intersects the surface in a single point, it works well for almost spherical vesicles, showing good agreement with analytical calculations[3].

Despite the excellent convergence properties it provides, the spectral representation of the surface is not always the best option: the main disadvantage here is that it is impossible to vary independently the level of detail of representation of different regions on the same surface. That is, if a surface contains a small region of high curvature, either this region will remain under-resolved if a reasonable number of spherical harmonics is used for surface representation or other regions of the surface will be over-resolved, which usually results in prohibitively large computation times. The notable examples of vesicle shapes that appear in physical problems but are rather difficult to represent by spherical harmonics include, among others, tethered shapes of vesicles sedimenting under large Bond numbers[4, 5] or dumbbell shapes observed in shear flow[6] and in straining flow[7], when the vesicle resembles two almost-spherical parts connected by a long and thin tube, or slipper shapes of vesicles in Strong Poiseuille flow, as shown in the present study (cf. Fig.1).

In this paper, we employ an alternative approach to surface parametrization, namely, we use the discretization by a triangular mesh, which offers a much greater versatility than spectral representation, allowing to study surfaces of arbitrary shapes or even topologies. This freedom comes at a price: Piecewise-linear interpolation of the surface sets a natural lower boundary $O(h^2)$ for the precision of the calculations, where h is the characteristic length of mesh edge. Moreover, this theoretical limit is hard to achieve due to the singular behavior of the BI kernel in vicinity of the pole. Additional numerical challenges rise in simulations on triangular mesh: Calculations of curvature and its derivatives for the membrane forces can not be performed on a piece-wise flat surface and the quality of the mesh can degrade during the deformation. These complications set additional obstacles to the inherent challenges of the problem, such as conservation of the local and global area of the membrane or the stiffness of high wave-number modes that triggers numerical instabilities.

Several studies of dynamics of vesicles under flow using BI method on triangular mesh already exist in literature[8, 5, 9, 10]. Of these studies only [9] calculated BI with error $O(h^2)$, while the others are limited to the error of order $O(h)$.

In this paper, we provide an implementation of the BI method on triangular mesh. An important step here is that we are able to regularize the diverging behavior of the Green kernel by the use of exact identities. Unlike for the existing singularity removal techniques, which are restricted to the case when only normal forces are exerted by the interfaces, our method works for arbitrary distributions of interfacial forces (i.e. with normal and tangential components). The implemented method, having an error $O(h^2)$, thus reaches the theoretical limit for piece-wise linear interpolation of the surface. The proposed singularity subtraction technique contributes to computation time efficiency and higher precision. Additionally, we revisit the calculation of the force and the imposition of the membrane inextensibility in order to increase the precision and the stability of the method. Note that the integration technique presented in this paper can be used to increase the precision of other algorithms using the BI formulation, such as spectral BI method for Stokes equations[2] or even to facilitate the solution of other physical problems, such as Poisson equation in electrostatics.

The paper is organized as follows: Section 2 describes the physical problem and its mathematical formulation, Section 3 describes the topology of the mesh, Section 4 describes the calculation of the curvature force, Section 5 describes the calculation of the BIs, Section 6 describes the solution for the Lagrange multiplier enforcing the inextensibility of the membrane, Section 7 describes extension of the method to simulation of inextensible capsules to mimic RBCs, Section 8 contains validation tests. Finally, Section 9 contains discussions and conclusion.

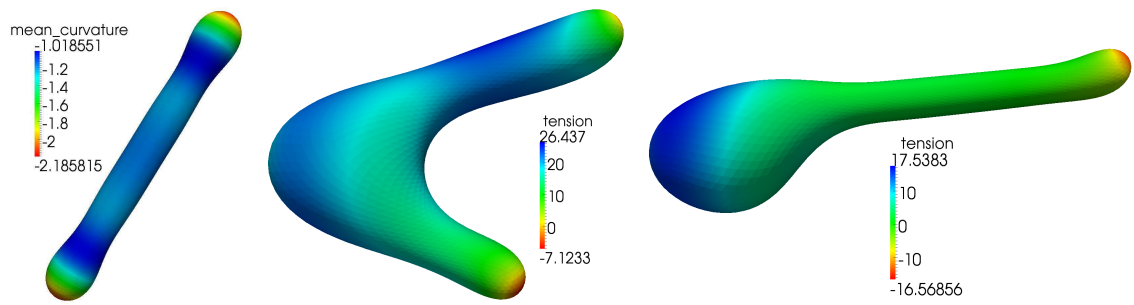

Figure 1: (color on-line) Deflated vesicles. From left to right: Equilibrium shape of a very deflated vesicle, transient shape of a deflated vesicle in Poiseuille flow, steady-state shape of the same vesicle in Poiseuille flow

2. Problem formulation

The following notations will be used throughout the text: capital Latin superscript letters (A, B, ...) are used to index the suspended entity (vesicle or capsule), the vertices are indexed by small superscript Latin letters starting from a (a, b, ...), small subscript Latin letters starting from i (i, j, ...) are used to denote coordinates in 3D space, capital Greek letter Δ is used to index triangles, and local coordinates for differential geometry will be denoted by subscript Greek letters starting from α. The convention of summation over repeated indices will be used (without distinction of covariant and contravariant indices) for coordinates but not for vertex indices. For each physical field, e.g., distribution of forces at the membrane $\boldsymbol{f}(\boldsymbol{r})$, we denote its numerical approximation at vertex a with the corresponding superscript, $\boldsymbol{f}^a$.

We focus first on vesicles. We consider the dynamics of one or more vesicles in infinite Newtonian liquid of viscosity η_{ext} subject to imposed flow $\boldsymbol{u}^\infty(\boldsymbol{r})$. A vesicle A is represented as a closed two-dimensional surface Σ^A enclosing Newtonian liquid with viscosity η^A_{int}. Due to microscopic nature of vesicles (or RBCs), the Reynolds number of the problem is usually quite small and it is safe to use the Stokes equation for the hydrodynamic part of the problem

$$\eta\Delta\boldsymbol{u}(\boldsymbol{r}) - \boldsymbol{\nabla}p(\boldsymbol{r}) + \boldsymbol{f}(\boldsymbol{r}) = 0, \quad \boldsymbol{\nabla}\cdot\boldsymbol{u}(\boldsymbol{r}) = 0, \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{r})$ is the fluid velocity in point $\boldsymbol{r}$, $p(\boldsymbol{r})$ is the pressure, and $\boldsymbol{f}(\boldsymbol{r})$ is the density of the force applied on the liquid. Equation (1) is valid at any point of the liquid, be it the suspending liquid or the one inside the vesicle. The difference is manifested in the value of viscosity η. In this study, we consider that no external force acts in the bulk of the liquid other that the forces at infinity creating the imposed flow. In this case, the density of forces is zero everywhere but at the vesicle surfaces, which allows[11] one to use the BI equation

$$\frac{(\eta_{ext}+\eta^A_{int})}{2}u_i(\boldsymbol{r}) = \eta_{ext}u_i^\infty(\boldsymbol{r}) + \sum_B \int_{\Sigma^B} G_{ij}(\boldsymbol{r},\boldsymbol{r}')f_j(\boldsymbol{r}')d^2r' + \tag{2}$$

$$+\sum_\beta(\eta_{ext}-\eta^B_{int})\int_{\Sigma^B} T_{ijk}(\boldsymbol{r},\boldsymbol{r}')u_j(\boldsymbol{r}')n_k(\boldsymbol{r}')d^2r',$$

where the indices A and B run over all vesicles involved in the problem, $\boldsymbol{r}$ is any point on the surface Σ^A, $\boldsymbol{n}(\boldsymbol{r})$ is the outward normal to the surface at the point $\boldsymbol{r}$. It is convenient to look at the two integrals on the right hand side of (2) as at linear operators with the single-layer (Stokeslet) kernel $G_{ij}(\boldsymbol{r},\boldsymbol{r}')$ and the double-layer (stresslet) kernel $T_{ijk}(\boldsymbol{r},\boldsymbol{r}')n_k(\boldsymbol{r}')$, respectively. The exact form of the kernels G and T depends on the boundary conditions of the problem. In this study, we consider only the free-space version of the problem (no boundary conditions other than the flow imposed at infinity), for which the kernels $G(\boldsymbol{r},\boldsymbol{r}')$ and $T(\boldsymbol{r},\boldsymbol{r}')$ depend only on the difference of the target position $\boldsymbol{r}$ and the source position $\boldsymbol{r}'$

$$G_{ij}(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{8\pi}\left(\frac{\delta_{ij}}{|\boldsymbol{r}-\boldsymbol{r}'|} + \frac{(\boldsymbol{r}-\boldsymbol{r}')_i(\boldsymbol{r}-\boldsymbol{r}')_j}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\right), \quad T_{ijk}(\boldsymbol{r},\boldsymbol{r}') = \frac{3}{4\pi}\frac{(\boldsymbol{r}-\boldsymbol{r}')_i(\boldsymbol{r}-\boldsymbol{r}')_j(\boldsymbol{r}-\boldsymbol{r}')_k}{|\boldsymbol{r}-\boldsymbol{r}'|^5}. \tag{3}$$

The equation (2) combines Stokes equation (1) with boundary conditions (no slip, jump of the normal component of viscous stress is equal to the force applied by the membrane, impermeability of the membrane and the imposed flow at infinity) and can be solved for the velocity field at the surface of each vesicle. The surface density of force $\boldsymbol{f}$ applied by the membrane on the adjacent liquids is calculated by variation of the bending energy[12]

$$\mathcal{E} = \sum_A \int_{\Sigma^A} \left[2\kappa^A (H(\boldsymbol{r}) - H_0^A)^2 + \zeta(\boldsymbol{r}) \right] d^2 r, \tag{4}$$

where κ^A is the bending modulus of the membrane A, H is the mean curvature (with the convention that the mean curvature of a unity sphere be equal to -1), H_0^A is a vesicle parameter usually called spontaneous curvature, and ζ is a Lagrange multiplier ensuring the incompressibility of the membrane. We will need the explicit expression for the force[13] (for an alternative derivation see [14])

$$\boldsymbol{f} = -\kappa^A \left[4(H - H_0^A)(H^2 - K + HH_0^A) + 2\Delta^s H \right] \boldsymbol{n} + 2\zeta H \boldsymbol{n} + \boldsymbol{\nabla}^s \zeta, \tag{5}$$

where K is the Gaussian curvature, $\boldsymbol{\nabla}^s$ is the surface gradient and $\Delta^s = \boldsymbol{\nabla}^s \cdot \boldsymbol{\nabla}^s$ is the Laplace-Beltrami operator. Due to the fluidity of the membrane, vesicles do not have a reference shape and the only conserved geometrical parameters are the volume V and the surface area S, which can be combined into a single scale-invariant parameter

$$\nu = \frac{3V}{4\pi (S/4\pi)^{3/2}}, \tag{6}$$

called the reduced volume.

Finally, the mathematical formulation of the local inextensibility of the membrane is written as

$$\boldsymbol{\nabla}^s \cdot \boldsymbol{u} = 0. \tag{7}$$

Together, equations (2), (5), and (7) represent a closed system, which can be solved for the unknown fields $\boldsymbol{u}(\boldsymbol{r})$ and $\zeta(\boldsymbol{r})$ for a given conformation of vesicles. The obtained velocity can be then used to update the conformation. In the present study, we use a simple explicit Euler scheme for temporal discretization of the problem. The time step is fixed and will be denoted as τ. Each iteration of the numerical scheme consists of five elementary steps: First, all relevant geometrical quantities (volume, surface area and forces) are calculated for each vesicle, then BI is calculated for obtained forces, next the surface inextensibility condition is solved and the tension Lagrange multiplier is updated, the next step is to advect the mesh vertices using the updated velocity field, and, finally, the mesh stabilization iteration is performed. Each step is described in the corresponding section of the paper.

3. Mesh topology

We use triangular mesh to track the shape of the vesicle. As the topology of the mesh is kept constant during the simulation, it is important to choose the mesh which provides an adequate description of the surface without computational overhead due to an excessive number of elements.

For almost spherical vesicles, the mesh is produced from an icosahedron, to which a refinement procedure[9] is applied N_r times: each triangle is divided into 4 smaller triangles by connecting the midpoints of the edges. The mesh is projected on the circumscribed sphere of the icosahedron after each refinement. Once the desired resolution is achieved, the surface is compressed or stretched along some direction to produce a spheroid of a given reduced volume (6). The resulting shape is uniformly rescaled and moved to obtain the desired volume and initial position.

A slightly different procedure is used for more deflated vesicles: the mesh is chosen as a sphero-cylinder: two semi-spheres (one with equatorial vertices and one without) and a cylindrical inset between them. Thus, the topology of every mesh generated by this recipe is specified by two numbers: the number of rows in the cylindrical part N_p and the number of vertices in each of the rows (number of vertices at the equator for a spherical vesicle) $5 \cdot 2^{N_r}$. The total number of vertices is $N_v = 2 + 10 \cdot 4^{N_r} + 5 \cdot 2^{N_r} N_p$. Sample meshes are presented in Fig.2. The introduction of sphero-cylindrical meshes allows us to treat many problems of dynamics of very deflated vesicles for which the icosahedral mesh fails due to excessive distortion of the triangles.

For clearness of the following, two vertices will be called neighbors if they are connected by an edge. Overall, the meshes described above have only vertices with 5 or 6 neighbors.

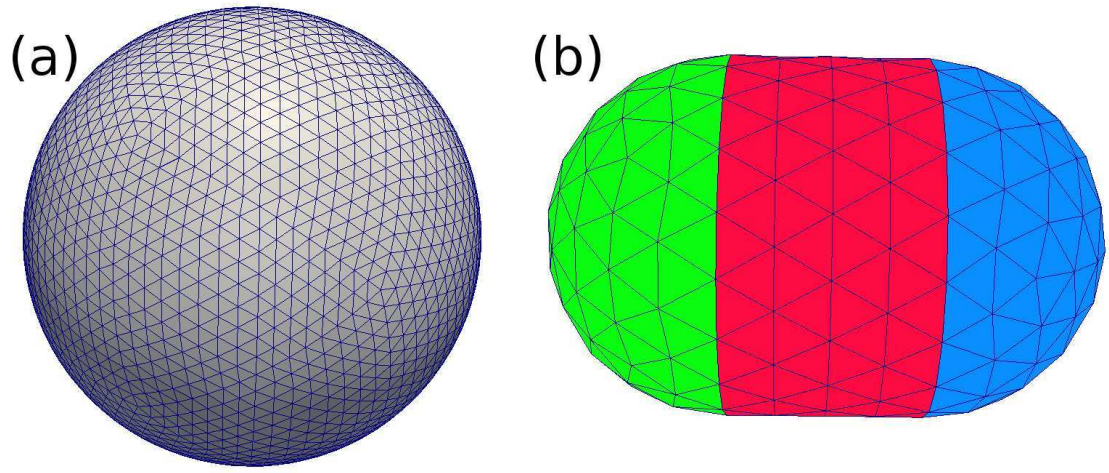

Figure 2: (color on-line) Sample meshes: (a): icosahedron after 4 refinements, $N_r = 4$, $N_p = 0$, $N_v = 2562$; (b): Sphero-cylinder, $N_r = 2$, $N_p = 4$, $N_v = 242$. For the sphero-cylinder, the colors distinguish the central inset and the semi-spheres.

4. Membrane force

The membrane force can be decomposed into two parts: the curvature part (the multiplier of κ^A in (5)) and the tension part (the rest). The former one is calculated directly, as explained in this Section, while the calculation of the latter part will be presented in Section 6.

Several approaches are used to calculate curvature and differential operators on triangular mesh: trigonometrical formulae[9, 5], direct variation of the energy[10], and quadratic interpolation[15]. It must be noted that only curvatures were calculated by quadratic interpolation in [15] while the trigonometric formulae similar to[9, 5] were used for the Laplace-Beltrami operator of the curvature. In addition, the method proposed in [15] relied on the precision of the numerical approximation of the normal, so that several iterations of the surface approximation were necessary for the normal direction to converge. We propose a rather simple but quite efficient improvement of the latter approach that eliminates aforementioned difficulties: Namely, we use the formulae of differential geometry to find the curvatures, the normal or the surface Laplacian. First, we introduce a local coordinate system related to each vertex a. We start by calculation of the approximate normal to the surface $\tilde{\boldsymbol{n}}^a$ as the average of normals to triangles adjacent to a and introduce two unit vectors $\boldsymbol{\xi}^a$ and $\boldsymbol{\eta}^a$ orthogonal to $\tilde{\boldsymbol{n}}^a$ and to each other. The local coordinates of vertex b in the coordinate system related to vertex a are then defined as

$$s^b_\xi = (\boldsymbol{r}^b - \boldsymbol{r}^a)\cdot\boldsymbol{\xi}^a, \quad s^b_\eta = (\boldsymbol{r}^b - \boldsymbol{r}^a)\cdot\boldsymbol{\eta}^a. \tag{8}$$

We use quadratic approximation for the surface in the vicinity of vertex a :

$$r_i(s_\xi, s_\eta) = r^a_i + \partial_\xi r^a_i s_\xi + \partial_\eta r^a_i s_\eta + \frac{1}{2}\left(\partial_{\xi\xi} r^a_i s^2_\xi + \partial_{\eta\eta} r^a_i s^2_\eta + 2\partial_{\xi\eta} r^a_i s_\xi s_\eta\right). \tag{9}$$

The five coefficients $\partial_\xi r^a_i$, $\partial_\eta r^a_i$, $\partial_{\xi\xi} r^a_i$, $\partial_{\eta\eta} r^a_i$, and $\partial_{\xi\eta} r^a_i$ are obtained by minimum square fitting of the approximation discrepancy at the neighbors of a :

$$\chi = \sum_{b\in U_a}\left[r^b_i - r^a_i - \partial_\xi r^a_i s^b_\xi - \partial_\eta r^a_i s^b_\eta - \frac{1}{2}\left(\partial_{\xi\xi} r^a_i (s^b_\xi)^2 + \partial_{\eta\eta} r^a_i (s^b_\eta)^2 + 2\partial_{\xi\eta} r^a_i s^b_\xi s^b_\eta\right)\right]^2, \tag{10}$$

where U_a is the set of all neighbors of vertex a.

The mean and Gaussian curvatures as well as the normal are calculated according to the usual expressions of differential geometry:

$$\boldsymbol{n}^a = \frac{\partial_\xi \boldsymbol{r}^a \times \partial_\eta \boldsymbol{r}^a}{|\partial_\xi \boldsymbol{r}^a \times \partial_\eta \boldsymbol{r}^a|}, \quad H^a = \frac{1}{2}\,\mathrm{Tr}\left[c^a.(g^a)^{-1}\right], \quad K^a = \det\left[c^a.(g^a)^{-1}\right], \quad g^a_{\alpha\beta} = \partial_\alpha r^a_i \partial_\beta r^a_i, \quad c^a_{\alpha\beta} = n^a_i \partial_{\alpha\beta} r^a_i, \tag{11}$$

where $g_{\alpha\beta}$ is the metric tensor, $c_{\alpha\beta}$ is the curvature tensor, and $\alpha, \beta \in \{\xi, \eta\}$. The normal $\boldsymbol{n}^a$ points outwards as long as the approximate normal $\tilde{\boldsymbol{n}}^a = \boldsymbol{\xi}^a \times \boldsymbol{\eta}^a$ points outwards.

Once the curvatures are calculated for every vertex, we find $\partial_\alpha H^\alpha$ and $\partial_{\alpha\beta} H^a$ by a fitting procedure analogous to (10). The Laplace-Beltrami operator is then calculated as

$$\Delta^s H = \frac{1}{\sqrt{|\det g|}} \partial_\alpha \left(\sqrt{|\det g|} g^{-1}_{\alpha\beta} \partial_\beta H \right), \tag{12}$$

which can be transformed into

$$\Delta^s H^a = \partial_{\alpha\beta} H^a (g^a)^{-1}_{\alpha\beta} - \left[(g^a)^{-1}_{\alpha\beta} \partial_{\alpha\beta} r^a_i \right] \left[(g^a)^{-1}_{\gamma\delta} \partial_\gamma H^\alpha \partial_\delta r^a_i \right], \quad \alpha, \beta, \gamma, \delta \in \{\xi, \eta\} \tag{13}$$

and then evaluated by substituting the approximations for $\partial_\alpha r^a_i$, $\partial_{\alpha\beta} r^a_i$, $\partial_\alpha H^a$, and $\partial_{\alpha\beta} H^a$ obtained in the fitting procedure.

The error of approximation of the surface by a second-order polynomial is $O(h^3)$, which gives $O(h^2)$ error for the normal, $O(h)$ error for the curvatures, and $O(h^{-1})$ error for the curvature Laplacian. Thus, *a priori*, the proposed algorithm is not sufficiently precise to calculate the curvature force for a given surface. However, thanks to the stiffness of the problem, the reverse problem of reconstruction of the shape by given values of the force can be solved with good precision by the present method. We show in Section 8 that, indeed, the force calculated by the present method for the shapes obtained *in our simulations* converges to the exact solution with error of order $O(h^2)$, which is consistent with the error order of the proposed algorithm.

5. Calculation of BIs

5.1. Exact identities

In this section we shall show how can we fully subtract singularities from the BI equation. Calculation of BI represents the most computationally expensive part of the problem. It is also the source of severe precision limitations due to the singular behavior of the Stokeslet kernel G. This explains why the highest care is required when designing an efficient method for calculation of BIs. The simplest implementation of BI calculation on a triangular mesh goes as follows. For each vertex a, all mesh triangles are divided into two groups: the "singular triangles", which contain the vertex a and the remaining "non-singular triangles". The integration over non-singular triangles can then be performed by a simple quadrature rule, while the integration over singular triangles is performed by exact analytical calculation[9], or in polar coordinates[5], or by a map from a square[16, 10], assuming linear interpolation of the forces and the shape over the triangle. It is easy to show that this rule has a numerical error of order $O(h)$: Indeed, for a simple 3 point quadrature rule, the error of integration over one triangle is of order of the second derivative of the integrand times the size of the triangle to the power 4 (i.e., $O(h^4)$). The second derivative of the Green kernel G is inversely proportional to the distance from the pole to the power 3 (cf. eq. (3)), which gives the following estimate for the surface density of the numerical error of integration e :

$$e(\boldsymbol{r}^a - \boldsymbol{r}) \propto \frac{h^2}{|\boldsymbol{r}^a - \boldsymbol{r}|^3}, \tag{14}$$

which, upon integration over surface Σ^A containing the vertex a, yields

$$\int_{\Sigma^A} e(\boldsymbol{r}^a - \boldsymbol{r}) d^2 r \propto h^2 \int_h^\infty \frac{r dr}{r^3} \propto h. \tag{15}$$

Here we used polar coordinates on the surface Σ^A in vicinity of the vertex a and replaced the upper boundary in the last integral in (15), which is of order of the size of the surface Σ, by infinity. The lower boundary of the last integral in (15) comes from the fact that singular triangles (forming a region with the size of order h surrounding the vertex a) are excluded from integration by a Gaussian quadrature rule. Despite being quite a rough estimate, eq. (15) gives the correct order of error, as will be shown in Section 8. The error estimate (15) can not be improved by increasing

the order of quadrature rule: Indeed, high-order derivatives of the Green kernels diverge in vicinity of the pole even faster than (14). As evident from (15), the main source of the error is the integration over the so-called "near-singular triangles", i.e., mesh triangles that are not singular but are still within the distance comparable to h from the pole of the Green kernel. A quadrature rule with global error of order $O(h^2)$ was used for near-singular triangles in [9] (cf. Section 8), although the details of the method were never published. The calculation of the quadrature weights was, however, quite expensive from the computational point of view.

A completely different approach is used extensively in BI simulations of droplets[17]: The main trick behind this method is to subtract from the integrand of (2) a certain function that has the same diverging behavior in vicinity of the pole but, at the same time, can be easily integrated. Hence the name "singularity subtraction" (SS) for this method. More precisely, the interfacial force for a droplet with surface tension coefficient γ is given as $2\gamma H\boldsymbol{n}$, so that

$$2\gamma \int G_{ij}(\boldsymbol{r},\boldsymbol{r}')H(\boldsymbol{r}')n_j(\boldsymbol{r}')d^2r' = 2\gamma \int G_{ij}(\boldsymbol{r},\boldsymbol{r}')\left[H(\boldsymbol{r}')-H(\boldsymbol{r})\right]n_j(\boldsymbol{r}')d^2r' + 2\gamma H(\boldsymbol{r})\int G_{ij}(\boldsymbol{r},\boldsymbol{r}')n_j(\boldsymbol{r}')d^2r', \quad (16)$$

where $H(\boldsymbol{r})$ can be extracted from the last integral because it is independent of $\boldsymbol{r}'$. If mean curvature H varies smoothly along the surface of a droplet, $H(\boldsymbol{r}')-H(\boldsymbol{r}) \propto |\boldsymbol{r}'-\boldsymbol{r}|$ (when $|\boldsymbol{r}'-\boldsymbol{r}|$ is small) and the integrand of the first integral on the right hand side of (16) remains bounded for $\boldsymbol{r}' \neq \boldsymbol{r}$ and thus can be calculated with the error of order $O(h^2)$ by a simple Gaussian quadrature rule. The second integral on the right hand side of (16) evaluates to zero for a closed surface,

$$\int_\Sigma G_{ij}(\boldsymbol{r},\boldsymbol{r}')n_j(\boldsymbol{r}')d^2r' = \int \partial'_j G_{ij}(\boldsymbol{r},\boldsymbol{r}')d^3r' = 0, \quad (17)$$

and thus does not require numerical calculation. The last integral in (17) is over the volume contained inside the surface Σ and we note by ∂'_j the derivative with respect to r'_j. The physical meaning of (17) can be traced back to the incompressibility of liquid in the Stokes regime. In other words, we have subtracted from the surface tension force a fictitious normal force that does not affect the solution due to the incompressibility of Stokes flow but reduces the net force to zero at a given point $\boldsymbol{r}$ thus compensating the diverging behavior of the Stokeslet kernel in vicinity of this point. Obviously, the subtracted force can (and should) be taken different when calculating the BI for different points $\boldsymbol{r}$. It must be noted here that the integrand of (16) is still not fully regular at $\boldsymbol{r}=\boldsymbol{r}'$: In fact, its gradient is diverging. Nevertheless, it can be seen by an argument similar to (15) that the error of numerical integration of (16) by a simple quadrature rule is $O(h^2)$.

With this simple example in mind, we proceed to the SS technique for vesicles. As readily seen from (5), the situation is much more complicated for vesicles: Indeed, the tension force due to a non-uniform Lagrange multiplier $\zeta(\boldsymbol{r})$ has a tangential component $\nabla^s_i\zeta(\boldsymbol{r}) = [\delta_{ij}-n_i(\boldsymbol{r})n_j(\boldsymbol{r})]\partial_j\zeta(\boldsymbol{r})$, which is perpendicular to the normal and thus can not be reduced to zero by subtraction of any normal force. Therefore, the exact identity (17) is not sufficient to regularize the first integral in (2) for vesicles or any other interfaces exerting tangential forces on the liquids. This limitation has troubled mankind for many years, as mentioned, e.g., in [18]. To our knowledge, this continues to be the state of the art to the present day. We are going to solve this problem by introducing a second exact identity that allows us to reduce to zero the tangential component of the force at a given point. The trick is to note that the tangential projection operator can be written as a double cross-product with the normal: $(I-\boldsymbol{n}\otimes\boldsymbol{n})\cdot\boldsymbol{f} = -[\boldsymbol{n}\times[\boldsymbol{n}\times\boldsymbol{f}]]$ and to use the identity

$$\int_\Sigma G_{ij}(\boldsymbol{r},\boldsymbol{r}')e_{jkl}n_k(\boldsymbol{r}')d^2r' = -\frac{e_{ijl}}{4\pi}\int_\Sigma \frac{(r_j-r'_j)(r_k-r'_k)n_k(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3}d^2r'. \quad (18)$$

The identity (18) can be easily verified by passing to a volume integral, which yields the same integrands for both sides of (18). Note that a tiny vicinity of the pole gives a vanishing contribution to the surface and to the volume integrals thanks to the weakly singular behavior of the kernel G. As can be seen, the identity (18) is more sophisticated than (17): The integral in the left hand side does not evaluate to zero but is expressed via another integral, which might appear singular. However, this is not the case and the integrand of the right hand side of (18) remains bounded because infinitesimal displacements along the surface are perpendicular to the normal: $(\boldsymbol{r}-\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r}') = O(|\boldsymbol{r}-\boldsymbol{r}'|^2)$.

The two exact identities (17) and (18) can be used to reduce the order of singularity at the pole of the kernel G for distribution of surface forces with arbitrary normal and tangential components by replacing the original force with a

modified one:

$$\int_\Sigma G_{ij}(\boldsymbol{r},\boldsymbol{r}')f_j(\boldsymbol{r}')d^2r' = \int_\Sigma G_{ij}(\boldsymbol{r},\boldsymbol{r}')\tilde{f}_j(\boldsymbol{r},\boldsymbol{r}')d^2r' + \frac{[\boldsymbol{n}(\boldsymbol{r})\times\boldsymbol{f}(\boldsymbol{r})]_l e_{ijl}}{4\pi}\int_\Sigma \frac{(r_j-r'_j)(r_k-r'_k)n_k(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3}d^2r', \tag{19}$$

where the modified force $\tilde{\boldsymbol{f}}$ is written as

$$\tilde{\boldsymbol{f}}(\boldsymbol{r},\boldsymbol{r}') = \boldsymbol{f}(\boldsymbol{r}') - \boldsymbol{n}(\boldsymbol{r}')\,(\boldsymbol{f}(\boldsymbol{r})\cdot\boldsymbol{n}(\boldsymbol{r})) + [\boldsymbol{n}(\boldsymbol{r}')\times[\boldsymbol{n}(\boldsymbol{r})\times\boldsymbol{f}(\boldsymbol{r})]]. \tag{20}$$

It is easy to see that (i) $\tilde{\boldsymbol{f}}(\boldsymbol{r},\boldsymbol{r}') = O(|\boldsymbol{r}-\boldsymbol{r}'|)$ for fixed $\boldsymbol{r}$ as $\boldsymbol{r}'$ approaches $\boldsymbol{r}$ and (ii) $(\boldsymbol{r}-\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r}') = O(|\boldsymbol{r}-\boldsymbol{r}'|^2)$ as $\boldsymbol{r}'$ approaches $\boldsymbol{r}$ along the surface Σ. Thus, all the integrands on the right hand side of (19) are bounded and continuous as a function of $\boldsymbol{r}'$ for any $\boldsymbol{r}' \neq \boldsymbol{r}$ (provided $\boldsymbol{f}$ is continuous) and therefore can be calculated with decent precision by a simple 3-point quadrature rule as explained below. This completes the basic idea of SS technique for arbitrary distribution of forces. We would like to stress that the transformation (19), (20) is an exact identity that holds for an arbitrary smooth closed surface with continuous normal and forces and does not involve any approximation in itself. The advantage of the SS is that the right hand side of eq. (19) has a much better behavior in vicinity of the pole than the left hand side and is thus much easier to discretize for numerical evaluation with good precision.

The question of continuity of the force and smoothness of the surface is not as evident as it might seem to be and deserves a separate discussion. The ideal mathematical solution of the problem should be sufficiently smooth (may be, except for several special points under some conditions). However, the discretized numerical problem deals with only a finite set of forces defined on the vertices of the mesh. In this case, we understand by continuity of the force or of the normal that if a more refined mesh is taken the difference of forces or normals on neighboring vertices decreases. In other words, the amplitude of excitations of modes with wavelength of the order of mesh cell size remains small in the course of simulation. For vesicles, short-wavelength excitations in the shape or in the Lagrange multiplier ζ create restoring forces (5) with a strong negative feedback. It is then important for the stability of the simulation that these restoring forces correspond to a velocity field that suppresses the initial excitation. That means that the calculation of the first integral in (2) must be positive-definite as a linear operator on vector fields defined on mesh vertices. It is true that the kernel G is positive definite as an operator on non-discretized surface. Namely,

$$D = \langle \boldsymbol{f}, \hat{G}\boldsymbol{f}\rangle = \int_\Sigma d^2r \int_\Sigma d^2r'\, f_i(\boldsymbol{r})G_{ij}(\boldsymbol{r},\boldsymbol{r}')f_j(\boldsymbol{r}') \tag{21}$$

is the viscous dissipation in flow created by the forces $\boldsymbol{f}$ in a fluid of viscosity 1 and thus can not be negative. However, it turns out that the discretized version of the operator $\hat{G}$ is not always positive-definite. In practice, the discretized operator $\hat{G}$ remains positive-definite if calculated using SS technique combined with the simple Gaussian quadrature rule on mesh vertices only if the angles of mesh triangles remain not less than $\sim 30°$. Approximately the same limitation is imposed by other simple integration techniques. We improve the stability of our method by calculating the BI over singular and almost-singular triangles using a more refined mesh, as explained below.

Before describing the refined mesh, one more question should be considered. Besides being positive-definite, the operator $\hat{G}$ is also self-adjoint:

$$\langle \boldsymbol{f}, \hat{G}\boldsymbol{g}\rangle = \langle \boldsymbol{g}, \hat{G}\boldsymbol{f}\rangle = \int_\Sigma d^2r \int_\Sigma d^2r'\, f_i(\boldsymbol{r})G_{ij}(\boldsymbol{r},\boldsymbol{r}')g_j(\boldsymbol{r}'). \tag{22}$$

This equation follows from the fact that $G_{ij}(\boldsymbol{r},\boldsymbol{r}') = G_{ji}(\boldsymbol{r}',\boldsymbol{r})$ and expresses the reciprocal theorem for Stokes flow. However, after the regularizing transformation (19), (20) is applied, this explicit reciprocity is lost. This suggests that there exists a second way to perform SS that is adjoint to (19), (20). Indeed, one can easily verify that

$$\begin{aligned}\int_\Sigma G_{ij}(\boldsymbol{r},\boldsymbol{r}')f_j(\boldsymbol{r}')d^2r' = \int_\Sigma \Big[G_{ij}(\boldsymbol{r},\boldsymbol{r}')f_j(\boldsymbol{r}') - n_i(\boldsymbol{r})f_k(\boldsymbol{r})G_{kj}(\boldsymbol{r},\boldsymbol{r}')n_j(\boldsymbol{r}') - e_{ipl}n_p(\boldsymbol{r})f_k(\boldsymbol{r})G_{kj}(\boldsymbol{r},\boldsymbol{r}')e_{jml}n_m(\boldsymbol{r}')\Big]d^2r' - \\ -\frac{1}{4\pi}\int_\Sigma \frac{[\boldsymbol{n}(\boldsymbol{r})\times[\boldsymbol{f}(\boldsymbol{r})\times(\boldsymbol{r}-\boldsymbol{r}')]]\,((\boldsymbol{r}-\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r}'))}{|\boldsymbol{r}-\boldsymbol{r}'|^3}d^2r'\end{aligned} \tag{23}$$

and that the expressions under both integrals on the right hand side of (23) remain bounded for every $\boldsymbol{r}' \neq \boldsymbol{r}$. We use the half-sum of the rules (19), (20) and (23), which gives an explicitly self-adjoint regularized kernel for the first integral on the right hand side of (2). Our plan is to capitalize on this symmetry later when we solve the membrane inextensibility constraint for the Lagrange multiplier ζ.

5.2. Refined mesh

As already discussed above, using only mesh vertices for BI discretization can result in instability of the simulation under strong deformations of mesh triangles. In addition, SS does not improve precision of integration over almost singular triangles in situations when surface is close to self-contact or when two surfaces come very close to each other. Our plan is to take advantage of additional quadrature points and partitions of unity[19, 20] to improve the stability and the precision of the integration on almost singular triangles. The partition of unity technique is widely used in spectral BI methods and consists in the following idea: a smooth weight function is used to split the singular Green kernel into two parts: a well behaved long-range part and a short-range part that is equal to zero safe for a small round patch on the reference sphere centered at the pole of the Green kernel. The first part is then easily integrated by traditional quadrature rules, while the second one is integrated in polar coordinates.

Because we do not use an explicit reference configuration, some modifications to the partition of unity technique are necessary. It turns out that these modifications will, actually, provide certain advantages over the original technique. Namely, instead of using the distance between the source and the target *on the reference sphere* as the argument of the splitting weight function, we use the distance between the source and the target *in the coordinate space*. As we will show below, this little modification leads to the same treatment of integration on all almost-singular triangles, regardless whether they belong to the same surface as the target point or not, and whether the distance between the target and the source triangle is small if measured along the surface in the first case. It must be noted here that the case when the source and the target of the Green kernel are well separated on the reference sphere but not in the real space is usually denied a proper treatment if the splitting weight function is based on the distance along the reference sphere, which can lead to severe precision limitations for surfaces that are close to self-contact.

The second difference here is that instead of using a separate set of additional quadrature nodes for each target point, as is usually done during calculation of the almost-singular contribution in spectral methods[19, 20], we use the same refined mesh for all target points. This allows us to compute the interpolated forces and multiply them by the quadrature weights in advance for each vertex of the refined mesh, which requires a negligible fraction of BI computation time. It is then sufficient to select the vertices which lie within the cut-off distance from a given target vertex and multiply the precalculated forces by the Green kernel during the integration. The selection is done by a simple lookup algorithm that allows to discard the vertices of the refined mesh that are well separated from a given target vertex without actually calculating the distance to them.

The refined mesh is obtained by cutting each triangle of the original mesh into 16 equal triangles (cf. Fig. 3, left). The coordinates and forces on the refined mesh are obtained by piece-wise linear interpolation from the corresponding values on the original mesh. More precisely, for a triangle with vertices a, b, and c belonging to the original mesh, the positions and forces of the refined mesh are defined as

$$\boldsymbol{r}(s_1, s_2, s_3) = s_1\boldsymbol{r}^a + s_2\boldsymbol{r}^b + s_3\boldsymbol{r}^c, \quad \boldsymbol{f}(s_1, s_2, s_3) = s_1\boldsymbol{f}^a + s_2\boldsymbol{f}^b + s_3\boldsymbol{f}^c, \quad s_1+s_2+s_3 = 1, \quad s_1, s_2, s_3 \in \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}, \tag{24}$$

where s_1, s_2, and s_3 are barycentric coordinates of the vertex of the refined mesh. Interpolation (24) is consistent on the edges and vertices of the original mesh.

We denote cut-off distance R_c, and introduce a splitting weight function

$$w(r) = \begin{cases} 1 - 4\left(\frac{r}{R_c}\right)^3 + 3\left(\frac{r}{R_c}\right)^4 & \text{if } r \leq R_c \\ 0 & \text{if } r > R_c \end{cases} \tag{25}$$

in order to accelerate the calculations (cf. Fig.3, right). As can be seen from (25), $w(r)$ is continuously differentiable. We use the weight function $w(r)$ to split the BI into two parts:

$$\int_\Sigma G_{ij}(\boldsymbol{r}, \boldsymbol{r}')\, f_j(\boldsymbol{r}') d^2r' = \int_\Sigma \left[1 - w\left(|\boldsymbol{r} - \boldsymbol{r}'|\right)\right] G_{ij}(\boldsymbol{r}, \boldsymbol{r}')\, f_j(\boldsymbol{r}') d^2r' + \int_\Sigma w\left(|\boldsymbol{r} - \boldsymbol{r}'|\right) G_{ij}(\boldsymbol{r}, \boldsymbol{r}')\, f_j(\boldsymbol{r}') d^2r'. \tag{26}$$

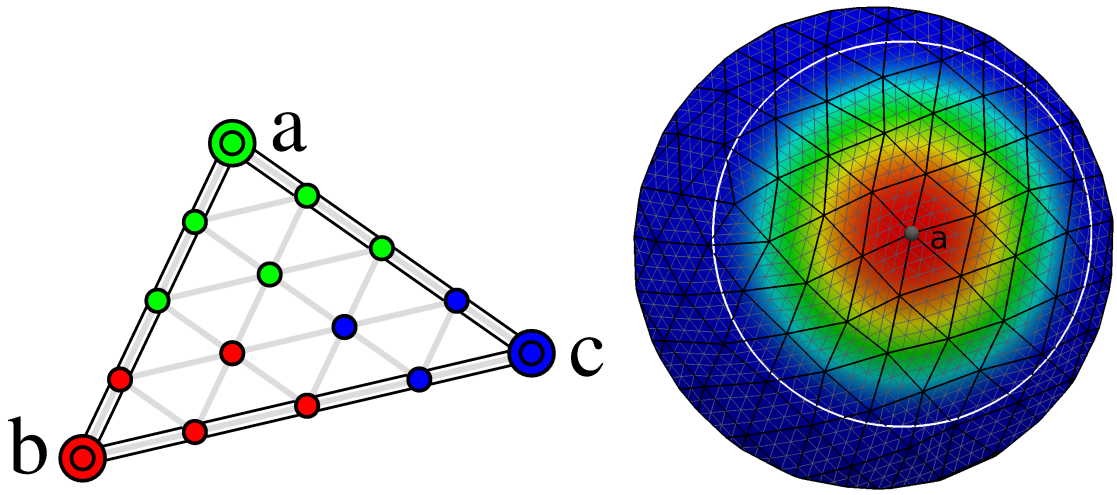

Figure 3: (color on-line) Left: A triangle of the original mesh subdivided into 16 triangles of the refined mesh. Large circles denote the vertices of the original mesh. Small circles denote the vertices of the refined mesh. Dark double lines denote the edges of the original mesh, light single lines denote the edges of the refined mesh, forming 16 small triangles. The colors of small vertices are chosen according to the lookup set to which they are attributed (as explained in Subsection 5.6). In this case $a < b < c$ (assuming the lookup set with smaller index has advantage in tie breaking). Right: A mesh fragment with the original (thick black lines) and the refined meshes (thin gray lines) shown. A small sphere marks vertex a. White circle marks the cut-off distance from the vertex a. Only vertices of the refined mesh which lie within the white circle contribute to the numerical estimate of BI at vertex a. Color by the weight function w of distance from the vertex a. The weight function decreases smoothly from 1 at vertex a to 0 beyond the white circle.

The first integral on the right hand side of (26) is regular at $\boldsymbol{r} = \boldsymbol{r}'$ and thus is calculated on the original mesh. The integrand of the second integral on the right hand side of (26) differs from 0 only if $|\boldsymbol{r} - \boldsymbol{r}'| < R_c$. and thus can be calculated on the refined mesh without taking excessive toll on the computation times. If $\boldsymbol{r} \in \Sigma$, the second integral of the right hand side of (26) is regularized by singularity subtraction, as explained below.

5.3. Exact identities with reduced support

The force (20) is small only when $\boldsymbol{r}'$ is close enough to $\boldsymbol{r}$, while for other values of $\boldsymbol{r}'$ the SS technique does not contribute to the precision of the BI calculations. However, the simple identities (17) and (18), must be calculated over the whole surface of the vesicle, which takes a significant portion of the time spent on BI computation. In order to reduce the computational cost of the problem, we generalize the identities (17) and (18) to include an arbitrary differentiable function $\tilde{w}$ of distance between $\boldsymbol{r}$ and $\boldsymbol{r}'$:

$$\int_\Sigma \left\{ \tilde{w}\left(|\boldsymbol{r}-\boldsymbol{r}'|\right) G_{ij}(\boldsymbol{r},\boldsymbol{r}') n_j(\boldsymbol{r}') + \frac{\tilde{w}'\left(|\boldsymbol{r}-\boldsymbol{r}'|\right)}{8\pi} \left[\delta_{ij} - \frac{(\boldsymbol{r}-\boldsymbol{r}')_i (\boldsymbol{r}-\boldsymbol{r}')_j}{|\boldsymbol{r}-\boldsymbol{r}'|^2} \right] n_j(\boldsymbol{r}') \right\} d^2 r' = 0, \tag{27}$$

$$\int_\Sigma \tilde{w}\left(|\boldsymbol{r}-\boldsymbol{r}'|\right) \left\{ G_{ij}(\boldsymbol{r},\boldsymbol{r}') e_{jkl} n_k(\boldsymbol{r}') + \frac{e_{ijl}}{4\pi} \frac{(r_j - r'_j)(r_k - r'_k) n'_k}{|\boldsymbol{r}-\boldsymbol{r}'|^3} \right\} d^2 r' + \tag{28}$$

$$+ \int_\Sigma \frac{\tilde{w}'\left(|\boldsymbol{r}-\boldsymbol{r}'|\right)}{8\pi} \left\{ \frac{(\boldsymbol{r}-\boldsymbol{r}')_i (\boldsymbol{r}-\boldsymbol{r}')_j}{|\boldsymbol{r}-\boldsymbol{r}'|^2} e_{jkl} n_k(\boldsymbol{r}') + e_{ijl} \left[\delta_{jk} - \frac{(\boldsymbol{r}-\boldsymbol{r}')_j (\boldsymbol{r}-\boldsymbol{r}')_k}{|\boldsymbol{r}-\boldsymbol{r}'|^2} \right] n_k \right\} d^2 r' = 0,$$

where $\tilde{w}'$ is the derivative of function $\tilde{w}$ with respect to its single argument. It is convenient to choose $\tilde{w} = w$, so that the integrand of the identities (27) and (28) is different from zero only for $|\boldsymbol{r} - \boldsymbol{r}'| < R_c$. This way, it is sufficient to apply SS technique only to the second integral in the right hand side of (26), as explained below.

5.4. Implementation for the Stokeslet kernel

For the sake of simplicity of notations, the sets of vertices of the original and the refined meshes on surface Σ will be called $M(\Sigma)$ and $M^r(\Sigma)$, respectively. Note that the points of the original mesh are also included in the refined mesh ($M(\Sigma) \subset M^r(\Sigma)$). The set of all triangles of the original and the refined meshes on surface Σ will be called $T(\Sigma)$ and $T^r(\Sigma)$, respectively. The set of vertices of a triangle Δ is called $M(\Delta)$ or $M^r(\Delta)$, depending whether $\Delta \in T(\Sigma)$ or $\Delta \in T^r(\Sigma)$. Conversely, the set of all triangles of the original (or refined) mesh that are adjacent to a vertex a will be called $T(a)$ (or $T^r(a)$). We denote the vector and the scalar area of triangle Δ as $\boldsymbol{N}(\Delta)$ and $S(\Delta)$, respectively. Finally, the quadrature weights at vertex a are defined as

$$S^r(a) = \frac{1}{3}\sum_{\Delta\in T^r(a)} S(\Delta), \quad S(a) = \frac{1}{3}\sum_{\Delta\in T(a)} S(\Delta), \quad \boldsymbol{N}^r(a) = \frac{1}{3}\sum_{\Delta\in T^r(a)} \boldsymbol{N}(\Delta), \quad \boldsymbol{N}(a) = \frac{1}{3}\sum_{\Delta\in T(a)} \boldsymbol{N}(\Delta), \quad \boldsymbol{\nu}^a = \frac{\boldsymbol{N}(a)}{|\boldsymbol{N}(a)|}. \tag{29}$$

Note that the vector $\boldsymbol{\nu}(a)$ is generally different from the approximation for the normal $\boldsymbol{n}^a$ defined in Section 4.

We employ the simple 3-point quadrature rules to calculate integrals:

$$\int_\Sigma \phi(\boldsymbol{r}) d^2r \approx \frac{1}{3}\sum_{\Delta\in T(\Sigma)} S(\Delta) \sum_{a\in M(\Delta)} \phi(\boldsymbol{r}^a) = \sum_{a\in M(\Sigma)} S(a)\phi(\boldsymbol{r}^a), \tag{30}$$

$$\int_\Sigma \phi(\boldsymbol{r})\boldsymbol{n} d^2r \approx \frac{1}{3}\sum_{\Delta\in T(\Sigma)} \boldsymbol{N}(\Delta) \sum_{a\in M(\Delta)} \phi(\boldsymbol{r}^a) = \sum_{a\in M(\Sigma)} \boldsymbol{N}(a)\phi(\boldsymbol{r}^a), \tag{31}$$

Using these notations, we write the discretized BI if $\boldsymbol{r}^a \notin \Sigma$ as

$$\int_\Sigma G_{ij}(\boldsymbol{r}^a - \boldsymbol{r}') f_j(\boldsymbol{r}') d^2r' \approx \sum_{b\in M(\Sigma)} (1 - w^{ab}) G^{ab}_{ij} f^b_j S(b) + \sum_{b\in M^r(\Sigma)} w^{ab} G^{ab}_{ij} f^b_j S^r(b), \tag{32}$$

where $\boldsymbol{r}^{ab} = \boldsymbol{r}^a - \boldsymbol{r}^b$, $G^{ab} = G(\boldsymbol{r}^{ab})$, $w^{ab} = w(|\boldsymbol{r}^{ab}|)$. If $\boldsymbol{r}^a \in \Sigma$, SS is performed:

$$\int_\Sigma G_{ij}(\boldsymbol{r}^a - \boldsymbol{r}') f_j(\boldsymbol{r}') d^2r' \approx \sum_{b\in M(\Sigma)\setminus\{a\}} (1 - w^{ab}) G^{ab}_{ij} f^b_j S(b) + \sum_{b\in M^r(\Sigma)\setminus\{a\}} w^{ab} G^{ab}_{ij} f^b_j S^r(b) - \tag{33}$$

$$-\mu\left[\nu^a_k f^a_k (G\cdot\boldsymbol{N})^a_i + (G\times\boldsymbol{N})^a_{ij}[\boldsymbol{f}^a \times \boldsymbol{\nu}^a]_j\right] - (1-\mu)\left[\nu^a_i f_k (G\cdot\boldsymbol{N})^a_k + e_{ijl}\nu^a_j f^a_k (G\times\boldsymbol{N})^a_{kl}\right],$$

where the regularizing contributions are calculated as

$$(G\cdot\boldsymbol{N})^a_i = \sum_{b\in M^r(\Sigma)\setminus\{a\}} \left[w^{ab} G^{ab}_{ij} N^r_j(b) + \frac{w'^{ab}}{8\pi}\left(\delta_{ij} - \frac{r^{ab}_i r^{ab}_j}{|r^{ab}|^2}\right) N^r_j(b)\right], \tag{34}$$

$$(G\times\boldsymbol{N})^a_{ij} = \sum_{b\in M^r(\Sigma)\setminus\{a\}} \left\{w^{ab}\left(e_{jkl} G^{ab}_{ik} - \frac{e_{ijk}}{4\pi}\frac{r^{ab}_k r^{ab}_l}{|r^{ab}|^3}\right) N^r_l(b) + \frac{w'^{ab}}{8\pi}\left[\frac{r^{ab}_i r^{ab}_k}{|r^{ab}|^2} e_{jkl} - e_{ijk}\left(\delta_{kl} - \frac{r^{ab}_k r^{ab}_l}{|r^{ab}|^2}\right)\right] N^r_l(b)\right\}. \tag{35}$$

The constant μ can be chosen arbitrary (preferably within the interval $[0; 1]$). In our calculations, we set $\mu = 1/2$, which makes the regularized kernel explicitly self-adjoint.

The exact value of R_c is usually chosen to be one half of the volume-equivalent radius of the vesicle. In this case, the computational times spent for calculation of each integral in the right hand side of (26) are approximately the same for a single almost-spherical vesicle.

5.5. Implementation for the stresslet kernel

The stresslet kernel is regularized thanks to a well-known identity

$$\int_{\Sigma} T_{ijk}(\boldsymbol{r}, \boldsymbol{r}')n_k(\boldsymbol{r}')d^2r' = \frac{\delta_{ij}}{2} \tag{36}$$

if $\boldsymbol{r} \in \Sigma$. The same weight function (25) is used to split the calculations between the original and the refined meshes. The velocity is linearly interpolated on the refined mesh in the same way as the coordinates or the forces (eq. 24). If $\boldsymbol{r}^a \notin \Sigma$, we obtain

$$\int_{\Sigma} T_{ijk}(\boldsymbol{r}^a - \boldsymbol{r}')u_j(\boldsymbol{r}')n_k(\boldsymbol{r}')d^2r' \approx \sum_{b\in M(\Sigma)} (1 - w^{ab})T^{ab}_{ijk}u^b_j N_k(b) + \sum_{b\in M^r(\Sigma)} w^{ab}T^{ab}_{ijk}u^b_j N^r_k(b), \tag{37}$$

where $T^{ab} = T(\boldsymbol{r}^{ab})$. If $\boldsymbol{r}^a \in \Sigma$,

$$\int_{\Sigma} T_{ijk}(\boldsymbol{r}^a - \boldsymbol{r}')u_j(\boldsymbol{r}')n_k(\boldsymbol{r}')d^2r' \approx \frac{u^a_i}{2} + \sum_{b\in M(\Sigma)\setminus\{a\}} (1 - w^{ab})T^{ab}_{ijk}(u^b_j - u^a_j)N_k(b) + \sum_{b\in M^r(\Sigma)\setminus\{a\}} w^{ab}T^{ab}_{ijk}(u^b_j - u^a_j)N^r_k(b). \tag{38}$$

5.6. Lookup algorithm for the refined mesh

As already mentioned, the splitting weight function (25) allows to reduce dramatically the computational cost of the BI calculation on the refined mesh. Namely, if a vertex b of the refined mesh lies outside the sphere of radius R_c with the center in vertex a of the original mesh, there is no contribution of the vertex b to the numerical estimate of the BI at the vertex a because the weight function $w(|\boldsymbol{r}^a - \boldsymbol{r}^b|)$ is equal to zero (cf. Fig. 4). Nevertheless, the verifications which vertices of the refined mesh lie within the distance R_c from a given vertex a of the original mesh can take significant amount of time due to the vast number of vertices in the refined mesh. We introduce a simple lookup algorithm that greatly reduces the number of verifications. The idea of the algorithm is to distribute all vertices of the refined mesh between disjoint lookup sets indexed by the vertices of the original mesh:

$$M^r(\Sigma) = \bigcup_{a\in M(\Sigma)} L(a), \quad L(a) \cap L(b) = \varnothing, \; a \neq b. \tag{39}$$

The decision, to which lookup set a given vertex $b \in M^r(\Sigma)$ is attributed, is taken upon examination of the barycentric coordinates of the vertex b (used in Section 5.2): The vertex b of the refined mesh is attributed to the lookup set $L(a)$ if the vertex a of the original mesh has the largest weight in the barycentric coordinates of the vertex b (cf. Fig. 3). The ties are broken by the global index of the vertices of the original mesh. This definition of the lookup sets allows us to calculate them upon mesh generation and does not require their update during the whole course of simulation.

At each time step, we precalculate so-called lookup radius for each lookup set, defined as

$$L_{max}(a) = \max_{b\in L(a)} |\boldsymbol{r}^b - \boldsymbol{r}^a|. \tag{40}$$

The lookup procedure goes as follows: For each couple of vertices on the original mesh a and b, we calculate the distance between them $r^{ab} = |\boldsymbol{r}^a - \boldsymbol{r}^b|$ (actually, this distance must be calculated anyway as a part of the BI calculation on the original mesh). The distance r^{ab} is then compared with $R_c + L_{max}(b)$. If $r^{ab} > R_c + L_{max}(b)$ then for any vertex $c \in L(b)$

$$|\boldsymbol{r}^a - \boldsymbol{r}^c| \geq \big||\boldsymbol{r}^a - \boldsymbol{r}^b| - |\boldsymbol{r}^b - \boldsymbol{r}^c|\big| \geq r^{ab} - L_{max}(b) > R_c \tag{41}$$

by a consequence of the triangle inequality. We see that if $r^{ab} > R_c + L_{max}(b)$, there is no contribution of the vertices in $L(b)$ to the numerical estimate of BI at the vertex a. On the contrary, if $r^{ab} < R_c + L_{max}(b)$, each vertex in $L(b)$ must be checked for distance from the vertex a individually. Likewise, r^{ab} must be compared with $R_c + L_{max}(a)$ to check if the vertices of $L(a)$ contribute to the numerical estimate of the BI at the vertex b. Finally, for each vertex a of the original mesh, contribution of the vertices from $L(a)$ to the numerical estimate of BI at the vertex a must be calculated. With the present definition of the lookup sets, their lookup radii turn out to be rather small compared to the cut-off distance (and even smaller if compared to the size of the vesicle) and are of order $O(h)$. That is, only a negligible fraction of vertices of the refined mesh lying beyond the distance R_c from a given vertex a are not discarded by the lookup algorithm.

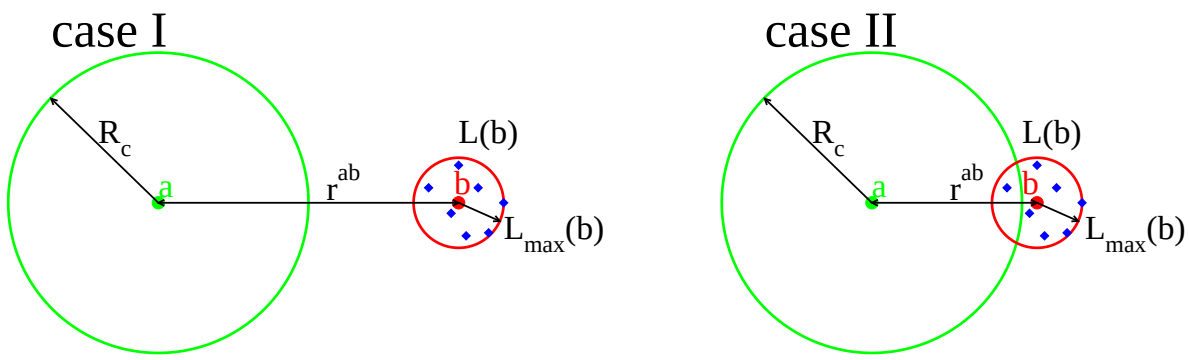

Figure 4: (color on-line) Schematic view of the lookup algorithm for the refined mesh (in 2D for simplicity): Vertices a and b belong to the original mesh. Small diamonds show vertices of the refined mesh attributed to the lookup set $L(b)$ of the vertex b. The small diamonds are shown scattered in a random way in order to demonstrate the general concept of the lookup algorithm, which works even if the vertices of the refined mesh are not arranged in the regular fashion shown in Fig. 3. The smaller red circle marks the largest distance $L_{max}(b)$ from the vertex b to a point of its lookup set $S(b)$. The larger green circle marks cut-off distance for calculations on refined mesh around the vertex a : If a vertex of the refined mesh is outside the larger green circle, it does not contribute to the numerical estimate of the BI at the vertex a. Case I: $r^{ab} > R_c + L_{max}(b)$. The red and green circles do not intersect: It is safe to assume that no point in the lookup set of vertex b contributes to the BI estimate at the vertex a. Case II: $r^{ab} < R_c + L_{max}(b)$. The red and green circles intersect: Each point in the lookup set of vertex b must be checked individually for contribution to the numerical estimate of BI at the vertex a. One diamond actually got inside the larger green circle and thus this vertex of the refined mesh gives a non-zero contribution to the numerical estimate of the BI at the vertex a.

5.7. *Volume conservation*

It follows from the incompressibility of Stokes flow that

$$\int_{\Sigma^A} u_i(\boldsymbol{r}) n_i(\boldsymbol{r}) d^2 r = 0 \tag{42}$$

for each vesicle A, which is equivalent to conservation of the volume inside that vesicle. It is known that the exact solution of the BI (2) satisfies the condition (42), provided the imposed flow has no sources inside the vesicle A. Unfortunately, the discretization (32), (33), (37), (38) is not compatible with (42), which creates drift of vesicle volumes. In order to resolve this problem, we project the velocity field calculated from BI equation on the subspace of velocity fields that conserve the volume of each vesicle in linear approximation:

$$\hat{P}_V \boldsymbol{u}^a = \boldsymbol{u}^a - \boldsymbol{\nu}(a) \frac{\sum\limits_{b \in M(\Sigma^A)} \boldsymbol{u}^b \cdot \boldsymbol{N}(b)}{\sum\limits_{b \in M(\Sigma^A)} \boldsymbol{\nu}(b) \cdot \boldsymbol{N}(b)}, \quad a \in M(\Sigma^A), \tag{43}$$

where $\hat{P}_V$ is the projection operator. After this projection, the change of the volume of a vesicle during one time step τ is of order $O(\tau^2)$, which is eliminated by rescaling as described in Section 6.2.

6. Membrane inextensibility

6.1. *Local and global inextensibility*

One of the most difficult challenges in simulation of dynamics of vesicles and inextensible capsules lies in the fact that inextensibility of the membrane must be enforced locally, which usually requires to resolve a linear system whose size is proportional to the combined number of vertices of all vesicles. In addition, dynamics of a vesicle is very sensitive to its total volume V and surface area S. Because these parameters are physical properties of a vesicle, we develop a numerical procedure that allows us to conserve these 2 parameters with great precision, as explained in this section.

Two different strategies are possible when solving for the Lagrange multiplier ζ : either a direct resolution of the linear system (7)[5] or iterative approach when the tension is slightly adjusted at each time step to ensure the surface incompressibility during variations of the shape of the vesicle. GMRES method is traditionally employed to solve for the Lagrange multiplier in spectral BI methods[1, 2]. Alternative approach is to use penalization technique[9]. The advantage of the latter approach is that the penalization is imposed not only on the local strains but also on deviations of the global surface area (or equivalently the reduced volume) from the prescribed value, which prevented numerical artifacts due to sensitivity of the vesicle dynamics to the reduced volume. The drawback of the last method is that a rather large penalization constant has to be used in order to ensure local incompressibility of the membrane with good precision, which sets a rather stiff limitation for the choice of numerically stable time steps. The proposed method consists in selecting the optimal values for the local and global strain penalization parameters at each time step. That is, the penalization parameters are chosen to minimize the local strains and to keep the global area of each vesicle at the prescribed value. By doing so, despite the fact that the penalization parameters can attain rather high values, no additional limitations are imposed on the time step. The inner volumes of vesicles are kept constant due to projection (43) and rescaling at each time step.

The volume and the surface area of a vesicle A are calculated as (using notations of subsection 5.4):

$$V^A \approx \frac{1}{3} \sum_{a \in M(\Sigma^A)} \boldsymbol{r}^a \cdot \boldsymbol{N}(a), \quad S^A \approx \sum_{a \in M(\Sigma^A)} S(a). \tag{44}$$

We define the strain field ρ^a and the tension field ζ^a for each vertex a. The strain field ρ^a is used to measure the local stretching of the membrane in vicinity of vertex a, while the tension field ζ^a is used as a Lagrange multiplier. Our goal is to minimize the strains $|\rho^a|$ by finding optimal values of the tension ζ^a.

The tension force $\boldsymbol{\phi}^a(\zeta^b)$ for arbitrary scalar field ζ^b is obtained by energy variation:

$$\phi_i^a(\zeta^b) = -\frac{1}{S(a)} \frac{\delta \mathcal{E}_{tens}(\zeta^b)}{\delta r_i^a}, \quad \mathcal{E}_{tens}(\zeta^b) = \sum_a \zeta^a S(a). \tag{45}$$

Because the BI technique described in Section 5 requires interpolation of the force on the refined mesh, we calculate the tension force explicitly: consider a triangle Δ defined by 3 vertices r^a, r^b, and r^c, then

$$\frac{\delta S(\Delta)}{\delta r_i^a} = \frac{1}{2} \frac{(\boldsymbol{r}^b - \boldsymbol{r}^c) \times \left[(\boldsymbol{r}^a - \boldsymbol{r}^b) \times (\boldsymbol{r}^a - \boldsymbol{r}^c)\right]}{\left|(\boldsymbol{r}^a - \boldsymbol{r}^b) \times (\boldsymbol{r}^a - \boldsymbol{r}^c)\right|} = \pm \frac{(\boldsymbol{r}^b - \boldsymbol{r}^c) \times \boldsymbol{N}(\Delta)}{2\,|\boldsymbol{N}(\Delta)|}. \tag{46}$$

The sign is '+' if the cycle $a \to b \to c \to a$ is counterclockwise viewed from outside the vesicle.

In order to simplify the notations, we denote the full force (tension part and the curvature part) for a given tension field ζ^b as $\boldsymbol{f}^a(\zeta^b)$. We denote the curvature force as $\boldsymbol{f}^a(0)$ (the full force when the tensions are equal to zero). We thus can write $\boldsymbol{f}^a(\zeta^b) = \boldsymbol{f}^a(0) + \boldsymbol{\phi}^a(\zeta^b)$. After calculation of the BI, we obtain the full velocity field $\boldsymbol{u}^a(\zeta^b) = \boldsymbol{u}^a(0) + \boldsymbol{v}^a(\zeta^b)$, where

$$u_i^a(0) = \hat{P}_V \left[\frac{2\eta_{ext}}{\eta_{int} + \eta_{ext}} u_i^\infty(\boldsymbol{r}^a) + \frac{2}{\eta_{int} + \eta_{ext}} \hat{G}_i^a(\boldsymbol{f}^c(0)) + \frac{2(\eta_{ext} - \eta_{int})}{\eta_{int} + \eta_{ext}} \hat{T}_i^a(\boldsymbol{u}^c(t_{n-1})) \right], \tag{47}$$

$$v_i^a(\zeta^b) = \frac{2}{\eta_{int} + \eta_{ext}} \hat{P}_V \hat{G}_i^a(\boldsymbol{\phi}^c(\zeta^b)). \tag{48}$$

Here we denote by $\hat{P}_V \hat{G}_i^a(\boldsymbol{f}^c)$ and $\hat{P}_V \hat{T}_i^a(\boldsymbol{u}^c)$ the BIs calculated using the recipes of Section 5 for the force field $\boldsymbol{f}^c$ and velocity field $\boldsymbol{u}^c$, respectively. Finally, we denote by $\boldsymbol{u}^c(t_{n-1})$ the full velocity field at previous time step.

We denote as $\rho^a(\boldsymbol{v}^b)$ the local strain rate of arbitrary velocity field $\boldsymbol{v}^b$ measured at vertex a. This value is defined as the variation of $S(a)$ if all vertices are displaced using the velocity field $\boldsymbol{v}^b$:

$$\rho^a(\boldsymbol{v}^b) = \sum_b \frac{\delta S(a)}{\delta r_i^b} \boldsymbol{v}_i^b. \tag{49}$$

As noted in [10], the linear operators (45) and (49) can be considered adjoint. We can see from (44) that

$$\frac{dS^A}{dt} = \sum_{a \in M(\Sigma^A)} \rho^a(\boldsymbol{u}^b), \tag{50}$$

which means that zero local strain ensures conservation of global area of the vesicle. This conservation is exact only for infinitesimal time steps: For finite time step τ, there is a small drift of global area. This drift has two origins: non-linear effects and changes of the surface area and volume of the vesicle during mesh advection. The former contribution is of order $O(\tau^2)$ and thus can be reduced by taking sufficiently small time step τ, while the latter is of order $O(\tau)O(h^2)$ and thus can be reduced by taking sufficient number of discretization points. We compensate the drift of surface area by adding a small isotropic correction to surface tension. Namely, we take the following compensation of the area drift for each vesicle A at time step t_n :

$$\delta S^A(t_n) = \frac{S^A - S^A(t_n)}{\tau} - \frac{S^A(t_n) - S^A(t_{n-1}) - \tau\delta S^A(t_{n-1})}{\tau}, \tag{51}$$

where S^A is the imposed area for vesicle A, $S^A(t_n)$ is the surface area of vesicle A at time step t_n, $S^A(t_{n-1})$ is the surface of the same vesicle at the previous time step, and $\delta S^A(t_{n-1})$ is the compensation of drift for the vesicle A at the time step t_{n-1}. The first part of the right hand side of (51) extends the surface area of the vesicle from $S^A(t_{n-1})$ to S^A, while the second one is equal to the difference between the expected and the actual areas of the vesicle A at the time step t_n and represents the non-linear effects and changes of the area during mesh advection. Because the latter part does not change significantly between subsequent time steps, the compensation (51) results in a very precise conservation of the global area of each vesicle. In practice, the correction to the tensions due to the drift compensation turned out to be negligible.

Thus, at each time step, we impose the dilatation of the total area by δS^A for of each vesicle A, which gives the following condition on the local strains for vertices $a \in M(\Sigma^A)$:

$$\sum_{a\in M(\Sigma^A)} \rho^a(\boldsymbol{u}^b(\zeta^c)) = \delta S^A. \tag{52}$$

Because condition (52) is incompatible with zero local strain (as follows from (50)), we impose the following system on local strains:

$$\forall A:\ \forall a \in M(\Sigma^A):\ \rho^a(\boldsymbol{u}^b(\zeta^c)) = \frac{S(a)}{S^A(t_n)}\delta S^A, \tag{53}$$

which gives a system of equations (53) linear in unknowns ζ^c. Direct resolution of this system being impractical, we use the iterative approach: We take tension field $\zeta^b(t_{n-1})$ from the time step t_{n-1}, and perform an optimization iteration to obtain the tension field $\zeta^b(t_n)$ at the time step t_n. Two trial directions per vesicle are used for the change of the tension: For each vesicle A, we take (i) the tension field equal to residual strain at previous time step for vertices of vesicle A and equal to zero for vertices of other vesicles (denoted as $\rho^a(t_{n-1})|_A$) and (ii) the tension field equal to 1 on vertices of A and equal to zero for vertices of other vesicles (denoted as $1|_A$). In other words, we chose the following ansatz for the tension $\zeta^a(t_n)$:

$$\zeta^a(t_n) = \zeta^a(t_{n-1}) + \alpha^A + \beta^A\rho^a(t_{n-1}),\ \ a \in M(\Sigma^A). \tag{54}$$

The values of α^A and β^A are chosen to satisfy two conditions: (i) the fixed global area constraint (52) and (ii) the minimal sum of squares of local strains. This gives the following problem of constrained minimization:

$$\begin{aligned}
\chi &= \sum_A \sum_{a\in\Sigma^A} \left[\rho^a(\boldsymbol{u}^b(\zeta^c(t_n))) - \frac{S(a)}{S^A(t_n)}\delta S^A\right]^2 - \sum_A \gamma^A \left\{\left[\sum_{a\in\Sigma^A} \rho^a(\boldsymbol{u}^b(\zeta^c(t_n)))\right] - \delta S^A\right\} = \\
&= \sum_A \sum_{a\in\Sigma^A} \left[\tilde{\rho}^a(\boldsymbol{u}^b(\zeta^c(t_{n-1}))) + \sum_B \alpha_B\rho^a(\boldsymbol{v}^b(1^c|_B)) + \sum_B \beta_B\rho^a(\boldsymbol{v}^b(\rho^c(t_{n-1})|_B))\right]^2 - \\
&- \sum_A \gamma^A \left[\sum_{a\in\Sigma^A} \tilde{\rho}^a(\boldsymbol{u}^b(\zeta^c(t_{n-1}))) + \sum_B \alpha_B\rho^a(\boldsymbol{v}^b(1^c|_B)) + \sum_B \beta_B\rho^a(\boldsymbol{v}^b(\rho^c(t_{n-1})|_B))\right],
\end{aligned} \tag{55}$$

where γ^A is the Lagrange multiplier enforcing the constraint (52) during minimization (55) and the corrected strain is defined as

$$\tilde{\rho}^a(\boldsymbol{u}^b(\zeta^c(t_{n-1}))) = \rho^a(\boldsymbol{u}^b(\zeta^c(t_{n-1}))) - \frac{S(a)}{S^A(t_n)}\delta S^A,\ \ a \in M(\Sigma)^a. \tag{56}$$

The minimization of (55) is performed by solving a linear system for the unknowns α^C, β^C, and γ^A : Two sets of equations

$$\forall B: \sum_A \sum_{a\in M(\Sigma^A)} \left[\tilde{\rho}^a(\boldsymbol{u}^b(\zeta^c(t_{n-1}))) + \sum_C \alpha_C \rho^a(\boldsymbol{v}^b(1^c|_C)) + \sum_C \beta_C \rho^a(\boldsymbol{v}^b(\rho^c(t_{n-1})|_C)) - \gamma^A\right] \rho^a(\boldsymbol{v}^b(1^c|_B)) = 0, \tag{57}$$

$$\forall B: \sum_A \sum_{a\in M(\Sigma^A)} \left[\tilde{\rho}^a(\boldsymbol{u}^b(\zeta^c(t_{n-1}))) + \sum_C \alpha_C \rho^a(\boldsymbol{v}^b(1^c|_C)) + \sum_C \beta_C \rho^a(\boldsymbol{v}^b(\rho^c(t_{n-1})|_C)) - \gamma^A\right] \rho^a(\boldsymbol{v}^b(\rho^c(t_{n-1})|)) = 0 \tag{58}$$

are supplemented with the inextensibility conditions for each vesicle (52).

In order to calculate the coefficients of the system (52), (57), and (58), the following force and velocity fields are calculated at each time step: the full force $\boldsymbol{f}^a(\zeta^b(t_{n-1}))$ and velocity $\boldsymbol{u}^a(\zeta^b(t_{n-1}))$ fields are calculated for the tension field at the previous time step $\zeta^b(t_{n-1})$, tension forces $\boldsymbol{\phi}^a(\rho^b(t_{n-1})|_A)$ and $\boldsymbol{\phi}^a(1|_A)$, and velocity fields associated to them $\boldsymbol{v}^a(\rho^b(t_{n-1})|_A)$ and $\boldsymbol{v}^a(1|_A)$, are calculated using local strains at previous time step $\rho^b(t_{n-1})|_A$ and uniform tension $1|_A$ for each vesicle A. Because the trial tension fields $\rho^b(t_{n-1})|_A$ and $1|_A$ are equal to zero but on the vesicle A, the calculation of velocity fields associated to the trial tensions increases the computational complexity of one iteration no more than by a factor of 3. In fact, the regularizing integrals for the SS as well as the inverse distance for each pair of points can be calculated only once per iteration, which further reduces the computational cost. If the number of vesicles is not large, it is convenient to store the velocity fields and local strains corresponding to each trial direction. This way, the velocity field and the local strains for the tension field (54) can be obtained without BI recalculation thanks to the linearity. Otherwise, two passes of force calculation and integration are required: one to adjust the tension and one to calculate the velocity for the adjusted tension.

Note that if no refined mesh is used for BI calculation and the projection operator (43) is omitted, the system (53) is characterized by a symmetric matrix thanks to the explicitly symmetric singularity subtraction discussed in the Section 5 and the explicitly adjoint definitions of the tension force (45) and the local strain (49). In this case, the proposed method reduces to steepest descent method with additional constraints of prescribed global areas of the vesicles. If the refined mesh is used for BI calculations, the symmetry of the matrix characterizing the system (53) is only approximate. In practice, this does not pose a problem and the penalization technique presented above works well unless the BI operator ceases to be positive-definite.

6.2. Time advancement and mesh advection

We use a modification of mesh advection technique[17] in order to stabilize the shape of triangles: A fictitious tangential displacement is added to the velocity field calculated from the BI (2) in order to prevent excessive distortion of mesh triangles. Additionally, the vesicle is rescaled at each time step to keep the prescribed volume.

7. Inextensible capsules

Several previous numerical studies for capsules [18, 21, 22, 23, 20, 24, 25, 15] have dealt with more or less extensible capsules. Real capsules, made of polymers, are indeed extensible [26]. However, having in mind that a capsule model is to be built in order to mimic RBCs, the question of observing a strict inextensibility becomes natural. Our wish is thus to extend our model of vesicles to the case where the membrane is inextensible and is endowed with in-plane shear elasticity.

7.1. Elastic energy of capsules

We feel that it is necessary to give a brief introduction to the elasticity of 2D surfaces before actually describing the numerical algorithm for capsule simulation. The invariants of 2D relative displacement gradient[27] are traditionally used to characterize large deformations of 2D surfaces. We find it more convenient to use the metric tensor on the surface to characterize its deformation, as discussed, e.g., in[18]. While both approaches are equivalent and lead to the same analytical expressions for elastic forces, metric tensors are easier to calculate on triangular meshes.

Unlike for vesicles, reference configuration is indispensable to characterize the elastic properties of capsules. For convenience, we choose the reference configuration to be the state corresponding to the minimum of elastic energy.

Thus, for every point $\boldsymbol{r}$ on the capsule surface, we have the reference point $\boldsymbol{x}(\boldsymbol{r})$. Alternatively, we can consider dependence of the actual point $\boldsymbol{r}$ on the reference coordinate $\boldsymbol{x}$. Now the major assumption in the classical theory of elasticity is that the elastic energy of a material depends only on the first derivatives of $\boldsymbol{r}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$. This assumption is justified if the characteristic wave vectors of the deformation are much smaller than the characteristic inverse distance of the microscopic structure of the material. In order to establish the general form of the dependence of the elastic energy on the partial derivatives $\partial r_i/\partial x_j$ for an isotropic 2D material, we consider a parametrization of the reference configuration $\boldsymbol{x}(s^1, s^2)$ by local coordinates s_1 and s_2 in the vicinity of a point $\boldsymbol{x}^0 = \boldsymbol{x}(s_0^1, s_0^2)$, which induces a parametrization of the deformed capsule $\boldsymbol{r}(s^1, s^2) = \boldsymbol{r}(\boldsymbol{x}(s^1, s^2))$ in vicinity of the point $\boldsymbol{r}^0 = \boldsymbol{r}(\boldsymbol{x}^0)$. This parametrization induces two metric tensors: one on the deformed capsule

$$g_{\alpha\beta}(\boldsymbol{r}^0) = \frac{\partial r_i}{\partial s^\alpha}\frac{\partial r_i}{\partial s^\beta}\bigg|_{s^1=s_0^1, s^2=s_0^2}, \quad \alpha,\beta \in \{s^1, s^2\} \tag{59}$$

and one on the reference configuration

$$g^0_{\alpha\beta}(\boldsymbol{x}^0) = \frac{\partial x_i}{\partial s^\alpha}\frac{\partial x_i}{\partial s^\beta}\bigg|_{s^1=s_0^1, s^2=s_0^2}, \quad \alpha,\beta \in \{s^1, s^2\}. \tag{60}$$

The principal extensions $\epsilon_1(\boldsymbol{x}^0)$ and $\epsilon_2(\boldsymbol{x}^0)$ in the reference point $\boldsymbol{x}^0$ can be calculated by diagonalizing simultaneously the two matrices $g_{\alpha\beta}(\boldsymbol{r}^0)$ and $g^0_{\alpha\beta}(\boldsymbol{x}^0)$:

$$\det\left[g_{\alpha\beta}(\boldsymbol{r}^0) - \epsilon_{1,2}(\boldsymbol{x}^0)^2 g^0_{\alpha\beta}(\boldsymbol{x}^0)\right] = 0, \quad \alpha,\beta \in \{1, 2\}. \tag{61}$$

The surface density of the elastic energy of the capsule is an arbitrary symmetric function of ϵ_1 and ϵ_2 :

$$\mathcal{E}_{el} = \int E_{el}[\epsilon_1(\boldsymbol{x}), \epsilon_2(\boldsymbol{x})]d^2x. \tag{62}$$

Note that the ratio between the actual and the reference area is given by

$$\frac{d^2r}{d^2x} = \epsilon_1\epsilon_2, \tag{63}$$

which allows us to rewrite the energy (62) as an integral over the actual shape with a slightly different energy density (for inextensible capsules, there is no difference between integration over the reference configuration and over the actual surface). As specified from the beginning, we choose the reference configuration as the equilibrium state of the elastic energy. That is, the minimum of (62) corresponds to $\epsilon_1 = \epsilon_2 = 1$ everywhere on the capsule. Thus small deformations are characterized by the values of ϵ_1 and ϵ_2 close to 1. It is reasonable to make an assumption that the elastic energy be analytical in small deformations and make an expansion in Taylor's series:

$$E_{el}(\epsilon_1, \epsilon_2) = E_{el,0} + \sum_{i=2}^{\infty}\sum_{k=0}^{i} E_{k,i-k}(\epsilon_1 - 1)^k(\epsilon_2 - 1)^{i-k}, \tag{64}$$

where the terms linear in $\epsilon_1 - 1$ and $\epsilon_2 - 1$ are zero to ensure the equilibrium at $\epsilon_1 = \epsilon_2 = 1$. The symmetry of the elastic energy of a capsule with respect to a permutation of ϵ_1 and ϵ_2 implies that

$$E_{i,j} = E_{j,i}. \tag{65}$$

The simplest elasticity model would result from truncating the expansion (64) at $i = 2$ (a neo-Hookean model of 2D material). Various more sophisticated models exist in the literature with different combinations of high-order terms taken into account. Our idea is to keep the elasticity model as simple as possible. Unlike standard models used for capsules where (weakly) extensible membranes are treated [28], we shall start directly with inextensible capsules, and this will further reduce the number of possible terms in (64). First let us exploit the symmetry (65), which readily

tells us that the energy (64) can be rewritten as an expansion in powers of two elementary symmetric polynomials of $\epsilon_1 - 1$ and $\epsilon_2 - 1$, which are[1]

$$\epsilon_1 + \epsilon_2 - 2 \tag{66}$$

and

$$(\epsilon_1 - 1)(\epsilon_2 - 1) = 1 - \epsilon_1 - \epsilon_2 + \epsilon_1\epsilon_2. \tag{67}$$

Local inextensibility of a capsule imposes additional relation $\epsilon_1\epsilon_2 = 1$ on the principal stretches, as suggested by (63). The polynomials (66) and (67) are linearly dependent in this case, which means that the energy cost of local in-plane deformation of the capsule membrane can be represented as a function of only one variable, measuring the local shearing distortion. If the polynomial (66) were taken to be this variable, then the elastic energy (62,64) of the capsule in its reference configuration would not have been explicitly at equilibrium without the local inextensibility constraint (that is enforced only via a Lagrange multiplier), which would have resulted in position dependent Lagrange multiplier ζ at equilibrium. To avoid this inconvenience, we use another measure of local shearing distortion,

$$\varepsilon^2 = \epsilon_1^2 + \epsilon_2^2 - 2\epsilon_1\epsilon_2 = (\epsilon_1 - \epsilon_2)^2, \tag{68}$$

which is equivalent to (66) for inextensible capsules. Indeed,

$$\epsilon_1 + \epsilon_2 - 2 = \sqrt{(\epsilon_1 - \epsilon_2)^2 + 4\epsilon_1\epsilon_2} - 2 = \sqrt{4 + \varepsilon^2} - 2 = \frac{\varepsilon^2}{4} - \frac{\varepsilon^4}{64} \pm ... \tag{69}$$

Rewriting the energy density (64) as a function of ε^2 yields the following expression for the elastic energy of an inextensible capsule

$$\mathcal{E}_{el} = \int \left[\tilde{E}_{el}(\varepsilon^2(\boldsymbol{x})) + 2\kappa\left(H(\boldsymbol{x}) - H_0\right)^2 \frac{d^2 r}{d^2 x} + \zeta(\boldsymbol{x})\left(\frac{d^2 r}{d^2 x} - 1\right)\right] d^2 x. \tag{70}$$

The sum of last two terms is the Helfrich energy (4), which allows to calculate their associated force by the methods described in Sections 4 and 6. The first term is absent for vesicles and is evaluated by direct variation of energy, as explained below. We will need the explicit expression of ε^2 for calculation of the elastic force:

$$\varepsilon^2 = \frac{g_{11}g^0_{22} + g_{22}g^0_{11} - 2g_{12}g^0_{12}}{\det g^0_{\alpha\beta}} - 2\sqrt{\frac{\det g_{\alpha\beta}}{\det g^0_{\alpha\beta}}}. \tag{71}$$

The exact dependence $E_{el}(\varepsilon^2)$ can be chosen either in the simplest form

$$E_{el}(\varepsilon^2) = \frac{\mu_s}{2}\varepsilon^2 \tag{72}$$

or in a more sophisticated way, for example, using a finite extensibility non-linear elasticity model (FENEM) in order to prevent mesh collapse under strong flows (see section 8.8).

7.2. Calculation of elastic force on triangular mesh

Numerical simulation of inextensible capsules is not much different form that of vesicles. The reference configuration is parametrized by storing the reference coordinates for each mesh vertex. The deformation parameter ε^2 is calculated for each triangle of the mesh as if it was stretched uniformly: Consider a triangle with vertices $\boldsymbol{r}^a$, $\boldsymbol{r}^b$, and $\boldsymbol{r}^c$, having the reference coordinates $\boldsymbol{x}^a$, $\boldsymbol{x}^b$, and $\boldsymbol{x}^c$, respectively. The metric tensors are calculated as

$$g_{11} = (\boldsymbol{r}^b - \boldsymbol{r}^a)\cdot(\boldsymbol{r}^b - \boldsymbol{r}^a),\ \ g_{12} = g_{21} = (\boldsymbol{r}^b - \boldsymbol{r}^a)\cdot(\boldsymbol{r}^c - \boldsymbol{r}^a),\ \ g_{22} = (\boldsymbol{r}^c - \boldsymbol{r}^a)\cdot(\boldsymbol{r}^c - \boldsymbol{r}^a), \tag{73}$$

$$g^0_{11} = (\boldsymbol{x}^b - \boldsymbol{x}^a)\cdot(\boldsymbol{x}^b - \boldsymbol{x}^a),\ \ g^0_{12} = g^0_{21} = (\boldsymbol{x}^b - \boldsymbol{x}^a)\cdot(\boldsymbol{x}^c - \boldsymbol{x}^a),\ \ g^0_{22} = (\boldsymbol{x}^c - \boldsymbol{x}^a)\cdot(\boldsymbol{x}^c - \boldsymbol{x}^a). \tag{74}$$

[1]This is nothing but to say that the energy is a function of invariants which are the trace ($\epsilon_1 + \epsilon_2$) and the determinant ($\epsilon_1\epsilon_2$).

It must be noted that the value of ε^2 does not change upon any permutation of a, b, or c if calculated from (71) and (73 – 74). The elastic force is calculated by variation of (70) with respect to $\boldsymbol{r}^a$

$$f^a_{el,i} = -\frac{1}{S(a)}\sum_b \frac{\partial E_{el}((\varepsilon^2)^b)}{\partial r^a_i} S^{ref}(b) = -\frac{1}{S(a)}\sum_b \left.\frac{dE_{el}}{d\varepsilon^2}\right|_{\varepsilon^2=(\varepsilon^2)^b} \frac{\partial(\varepsilon^2)^b}{\partial r^a_i} S^{ref}(b), \tag{75}$$

where $S^{ref}(b)$ is the quadrature weight of the vertex b calculated for the reference configuration (in fact, it should be equal to $S(b)$ for an inextensible surface, however $S^{ref}(b)$ is independent of $\boldsymbol{r}^a$ and thus acts as a constant during the variation). The variation of ε^2 with respect to r^a_i is calculated from (71) using the following expressions:

$$\frac{\partial g_{11}}{\partial r^a_i} = 2r^a_i - 2r^b_i, \quad \frac{\partial g_{12}}{\partial r^a_i} = \frac{\partial g_{21}}{\partial r^a_i} = 2r^a_i - r^b - r^c \quad \frac{\partial g_{22}}{\partial r^a_i} = 2r^a_i - 2r^c_i. \tag{76}$$

The curvature and tension parts of the force, as well as the integration technique for capsules are the same as for the vesicles. Minor changes are required for the calculation of the lagrange multiplier ζ^a from the local inextensibility condition: The system of equations (53) must be replaced by

$$\forall A : \forall a \in M(\Sigma^A) : \rho^a(\boldsymbol{u}^b(\zeta^c)) = \frac{S^{ref}(a) - S(a)}{2\tau}, \tag{77}$$

which is resolved as described in Section 6. The time advancement is executed in the straightforward way

$$r^a(t_n) = r^a(t_{n-1}) + \tau v^a(t_{n-1}). \tag{78}$$

8. Validation

8.1. Non-dimensionalization

We provide several examples concerning both vesicles and capsules. In our validation tests, we consider a single vesicle or capsule of volume V, surface area S, that encloses liquid of viscosity η_{int} and is submersed in liquid of viscosity η_{ext}. Two dimensionless numbers can be formed using these parameters: The reduced volume (6) and the viscosity contrast

$$\lambda = \frac{\eta_{int}}{\eta_{ext}}. \tag{79}$$

Both vesicles and capsules are endowed with bending modulus κ, while two additional parameters will be used to describe the shear elasticity of capsules, as is explained below.

Two kinds of imposed flow are considered: the unbounded linear shear flow, characterized by the shear rate $\dot{\gamma}$:

$$v^\infty_x(\boldsymbol{r}) = \dot{\gamma} r_y, \quad v^\infty_y(\boldsymbol{r}) = v^\infty_z(\boldsymbol{r}) = 0 \tag{80}$$

and the unbounded axial Poiseuille flow, which is characterized by the flow curvature $\dot{\alpha}$:

$$v^\infty_x(\boldsymbol{r}) = \dot{\alpha}(r^2_y + r^2_z), \quad v^\infty_y(\boldsymbol{r}) = v^\infty_z(\boldsymbol{r}) = 0 \tag{81}$$

and coincides with the flow profile in a cylindrical channel up to constant, which does not affect the solution thanks to the Galilean invariance of the problem. The flow strength is reflected by a dimensionless capillary number C_a, which we define as

$$C_a = \frac{\dot{\gamma}\eta R^3_v}{\kappa}, \quad R_v = \left(\frac{3V}{4\pi}\right)^{1/3} \tag{82}$$

for shear flow and

$$C_a = \frac{\dot{\alpha}\eta R^4_v}{\kappa}, \quad R_v = \left(\frac{3V}{4\pi}\right)^{1/3} \tag{83}$$

for a Poiseuille flow. Here R_v is the volume-equivalent radius of the vesicle. In order to facilitate comparison with other works, we present our results in explicitly dimensionless form.

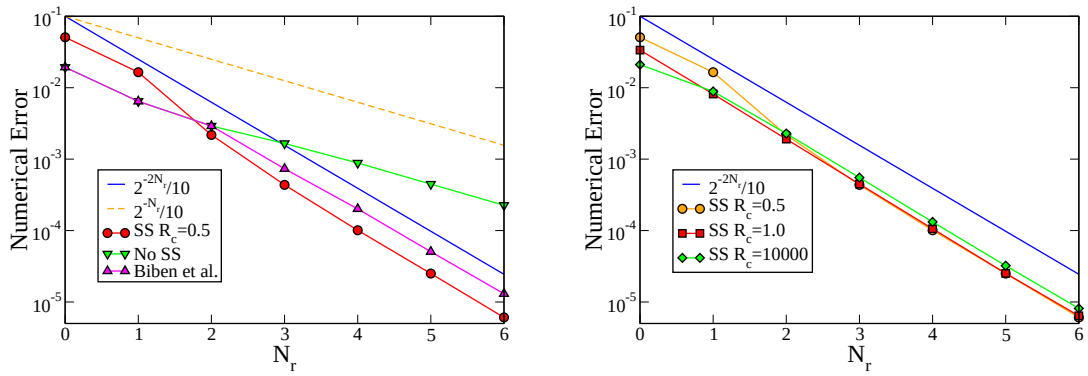

Figure 5: (color on-line) Difference between BI calculated numerically and the analytical result (85). Sphere of radius one, sample force field (84). Right: Numerical error of BI calculation as a function of number of refinements N_r, comparison of different integration techniques: SS technique with refined mesh, as described in the present study ($R_c = 0.5$, red circles), analytical integration for singular triangles and point-wise quadrature for all others (green triangles down), and the integration method used in [9] (violet triangles up). The solid line is the guide for the eyes that demonstrates that error scales as $O(h^2)$ with refinement for the SS technique and the method [9]. The dashed line is the guide for the eyes that demonstrates that error scales as $O(h)$ if no special treatment of almost-singular triangles is performed. Right: Effect of the refined mesh and of the value of the cut-off distance R_c on the precision of BI calculation.

8.2. Singularity subtraction tests

We start the validation of the problem by simple tests of the SS technique. We take for the test surface a sphere of radius one, for which the BIs can be calculated analytically. We choose the sample distribution of forces as

$$f_x(\boldsymbol{r}) = r_y r_z, \quad f_y(\boldsymbol{r}) = r_z r_x, \quad f_z(\boldsymbol{r}) = r_x r_y. \tag{84}$$

As can be seen from (84), the sample force has both a normal and a tangential components. In addition, the sample force (84) is not a linear function of coordinates, which will allow us to see the error due to force interpolation on the refined mesh. In Fig.5, we present the difference between BI calculated using the technique presented in Section 5 and the analytical result

$$\int G_{ij}(\boldsymbol{R} - \boldsymbol{r}) f_j(\boldsymbol{r}) d^2 r = 4 f_i(\boldsymbol{R})/35 \tag{85}$$

as a function of number of mesh vertices. It is evident from the slope of the curves in Fig.5 (left), that the numerical error in calculation of BI decreases asymptotically as $O(4^{-N_r}) = O(h^2)$ for the SS technique and the method [9], while the error decreases asymptotically only as $O(2^{-N_r}) = O(h)$ for the simple technique of combined analytical integration over singular triangles and numerical integration using 3 point quadrature over non-singular triangles. Figure 5 (right) shows that increasing the cut-off distance improves the precision of calculation for a small number of mesh vertices ($N_r \leq 2$). By contrast, moderate values of the cut-off distance ($R_c = 0.5$) give the best precision for larger numbers of mesh vertices ($N_r \geq 3$). This suggests that the piece-wise linear interpolation, which is used here to calculate the quantities on the refined mesh, generates a noticeable contribution to the numerical error. Overall, we see that the SS method becomes advantageous for $N_r \geq 2$. It must be noted here that a factor of 2 is not a negligible increase in precision over the method [9] due to the low-order nature of the convergence. Namely, for second order methods, increasing the precision by a factor of four requires increasing the refinement N_r by a unity, which gives fourfold increase in the number of mesh vertices and consequent increase by a factor of sixteen in BI computation time, which could make the simulation practically impossible. A sample distribution of numerical error for $R_c = 0.5$, $N_r = 3$, $N_p = 0$ is shown in Fig. 6 (left).

8.3. Hydrodynamic interactions between almost-touching surfaces

Another challenge for the numerical calculation of BIs is the situation when two interfaces are close to contact. This situation arises when hydrodynamic interactions of several interfaces are studied or one interface is close to self-contact. In any case, precision of BI calculation plays an important role in prevention of interpenetration of the

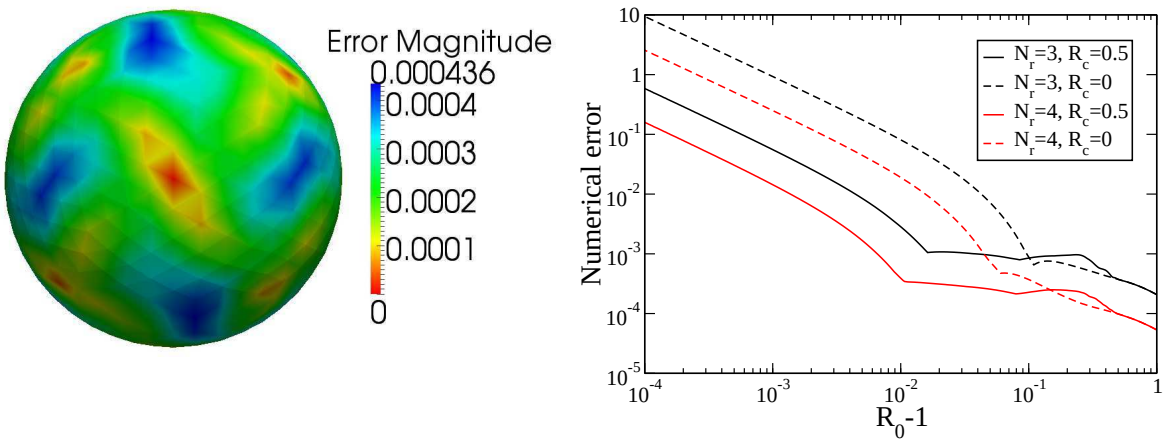

Figure 6: (color on-line) Left: Sample distribution of numerical error ($R_c = 0.5$, $N_r = 3$, $N_p = 0$). Right: Numerical error of BI calculation when target lies outside of the surface on which the forces are distributed. The effect of use of refined mesh. The sample force field (84) is distributed over a unity sphere. The numerical error is calculated on a sphere of radius $|R|$. The maximum of error for a given value of $|R|$ is plotted as a function of $|R| - 1$ for two meshes ($N_r = 3$, $N_p = 0$ and $N_r = 4$, $N_p = 0$). Solid lines are results of numerical integration with the use of refined mesh ($R_c = 0.5$), the dashed lines are the results of numerical integration without the use of refined mesh ($R_c = 0$). For the reference, edge lengths lie between 0.12 and 0.18 for $N_r = 3$ and between 0.04 and 0.10 for $N_r = 4$. Edges of the refined mesh are 4 times shorter.

interfaces. Here we show that the introduction of the refined mesh greatly improves the precision of BI calculation when two surfaces are almost touching. The same distribution of forces (84) on a unity sphere is considered. For each mesh point $\boldsymbol{r}^a$ on the unity sphere, we calculate the numerical estimate of the BI in the point $|R|\boldsymbol{r}^a$ and compare the result with the exact analytical expression

$$\begin{aligned}
\int G_{xj}(\boldsymbol{R}-\boldsymbol{r})f_j(\boldsymbol{r})d^2r &= R_yR_z\frac{105R_x^2(|R|^2-1)+15-7|R|^2}{70|R|^5},\\
\int G_{yj}(\boldsymbol{R}-\boldsymbol{r})f_j(\boldsymbol{r})d^2r &= R_zR_x\frac{105R_y^2(|R|^2-1)+15-7|R|^2}{70|R|^5},\\
\int G_{zj}(\boldsymbol{R}-\boldsymbol{r})f_j(\boldsymbol{r})d^2r &= R_xR_y\frac{105R_z^2(|R|^2-1)+15-7|R|^2}{70|R|^5},
\end{aligned} \tag{86}$$

for $|R| \geq 1$. Scanning through all mesh vertices a, we find the maximum discrepancy and plot it as a function of $|R|-1$. The results are presented in Fig. 6 (right) for two different mesh refinements with and without the auxiliary refined mesh for BI calculation. The general trend is that if $|R| \gg 1$, the numerical error is relatively low regardless if the refined mesh is used or not for the BI calculation. However, when $|R| - 1$ is less than the characteristic size of the mesh cell, a strong increase of numerical error is observed if BI is calculated without the use of refined mesh. For the calculation with refined mesh, the numerical error remains relatively low as long as $|R| - 1$ is not less than the characteristic size of a cell of the refined mesh. Overall, a clear advantage of the use of the refined mesh for calculation of BI is seen. It must be noted here that when two surfaces are separated by a distance greater than the cut-off distance for calculations on the refined mesh, the excess time cost of the use of the refined mesh for BI calculation is negligible thanks to the lookup algorithm.

8.4. Membrane shape, curvature, tension, and force

In order to check the convergence of the force calculation, we consider a steady state of a vesicle without viscosity contrast ($\lambda = 1$) in shear flow. We calculate the force $\boldsymbol{f}$, mean curvature H, and the Laplace-Beltrami operator of the curvature $\Delta^s H$. The results are plotted for the cross-section by the shear plane $z = 0$ as a function of the polar angle φ, where φ is the orientation angle in the shear plane (Fig. 7). The results obtained by high-order analytical expansions[29] are included as a reference. In order to achieve good convergence of analytical expansions, an almost

spherical vesicle ($\nu = 0.97$) and a strong flow ($C_a = 10$) are used. As can be seen in Fig. 7, a good convergence is achieved for the membrane force and the steady-state shape. The tension and the curvature converge to the analytical result as well. Surprisingly, even the curvature Laplacian converges to the analytical value everywhere but at the 12 exceptional vertices having only five neighbors. It must be noted here that the algorithm for the curvature force calculation presented in Section 4 works equally well for vertices with any number of neighbors higher than 4, and that the exceptional behavior in vicinity of the five-neighbor points comes from the tension force. That is, a continuous distribution of Lagrange multiplier ζ creates a strong jump of the normal component of the tension force at the five-neighbor vertices, which is compensated by an opposite jump in the curvature part of the force, in such a way that the distribution of the full force is continuous and converges to the theoretical value. Under strong flows, the Lagrange multiplier ζ is large and therefore creates a large jump of the tension force at the exceptional vertices, which, in turn, leads to large jumps in curvature force and, consequently, curvature Laplacian. The only way to remove this behavior is to design a method for tension force calculation that is less sensitive to the number of neighbors of a given vertex. Because the diverging results for tension force and the curvature Laplacian occur only at a small number of exceptional vertices and do not affect the vesicle dynamics, we leave this question for future studies.

8.5. Deflated vesicles

In order to verify the ability of the proposed method to treat very deflated vesicles we calculate the bending energy $\mathcal{E}_{curv} = 2\kappa \int H^2 dS$ and compare the measured values with the results reported in [1]. As can be seen in Fig. 8, The convergence is rather good for all reduced volumes. Nevertheless, some discrepancy is observed for very deflated vesicles, especially in a strong flow. In order to achieve good resolution of the surface, sphero-cylindrical meshes were used here.

8.6. Test for time-dependent dynamics

As a test for time-dependent dynamics, we consider a vesicle with reduced volume $\nu = 0.95$ and viscosity contrast λ ranging from 9 to 20 in shear flow. As known from analytical studies[3], the inplane dynamics of such a vesicle relaxes to a periodic vacillating-breathing (VB) or tumbling (TB) motion, depending on λ and the capillary number C_a. Two values of C_a will be considered for which independent measurements are available[2]: a rather weak flow $C_a = 0.19$, and a moderate flow $C_a = 5.7$. An additional analytical result is available[30] in the limit of infinitely weak flows, which we put as a reference for $C_a = 0.19$.

The following observations can be made by inspecting Fig.9: First of all, the results of [30] and of [2] agree very well, second, the results of the present code lie within 7% from the results of [2] for $N_r = 3$ and within 2% for $N_r = 4$. A decrease of numerical error by a factor of about 5 is observed when N_r is increased from 2 to 3 or from 3 to 4, thus giving a slightly higher improvement of precision than suggested by the theoretical estimation of error of order $O(h^2)$.

8.7. Shape and migration of deflated vesicles in unbounded Poiseuille flow

As the ultimate test for the BI calculation technique, we consider lateral migration of a vesicle in Poiseuille flow. Indeed, while the local shape of a small patch of a membrane is mostly defined by membrane forces, lateral migration velocity of a vesicle in Poiseuille flow is a very subtle effect entirely defined by the BI. Comparison between numerical and analytical study of lateral migration for almost spherical vesicles in Poiseuille flow can be found in [29]. Here we present some benchmarks for a more deflated vesicle, with reduced volume $\nu = 0.85$, under a strong flow $C_a = 100$. We consider only the case without viscosity contrast ($\lambda = 1$). We fix the lateral distance of the center of mass of the vesicle to the value $y_0/R_v = 0.5$ and study the subsequent vesicle dynamics. The vesicle assumes a rather peculiar slipper shape (Fig. 11, left). The section of this shape by the $z = 0$ plane as well as the y component of the average velocity of the membrane V_{my} are then measured as a function of the number of vertices N_v.

The sections of the vesicle by the $z = 0$ plane are presented in Fig. 10 (left). A notable feature of this shape is a small "tail", where the mean curvature reaches very high absolute value ($H \sim -10/R_v$ for $C_a = 100$). The "tail" is presented in detail in Fig. 10 (right).

The non-dimensionalized migration velocity $V_{my}/(\dot{\alpha} R_v^2)$ is shown in Fig.11. We performed simulations for several refinement numbers N_r and several numbers of inset rows N_p. As can be seen in Fig. 11, sphero-cylindrical meshes are advantageous in this case, giving a better estimate for the migration velocity than the simulation with $N_p = 0$ while using a lower number of mesh vertices.

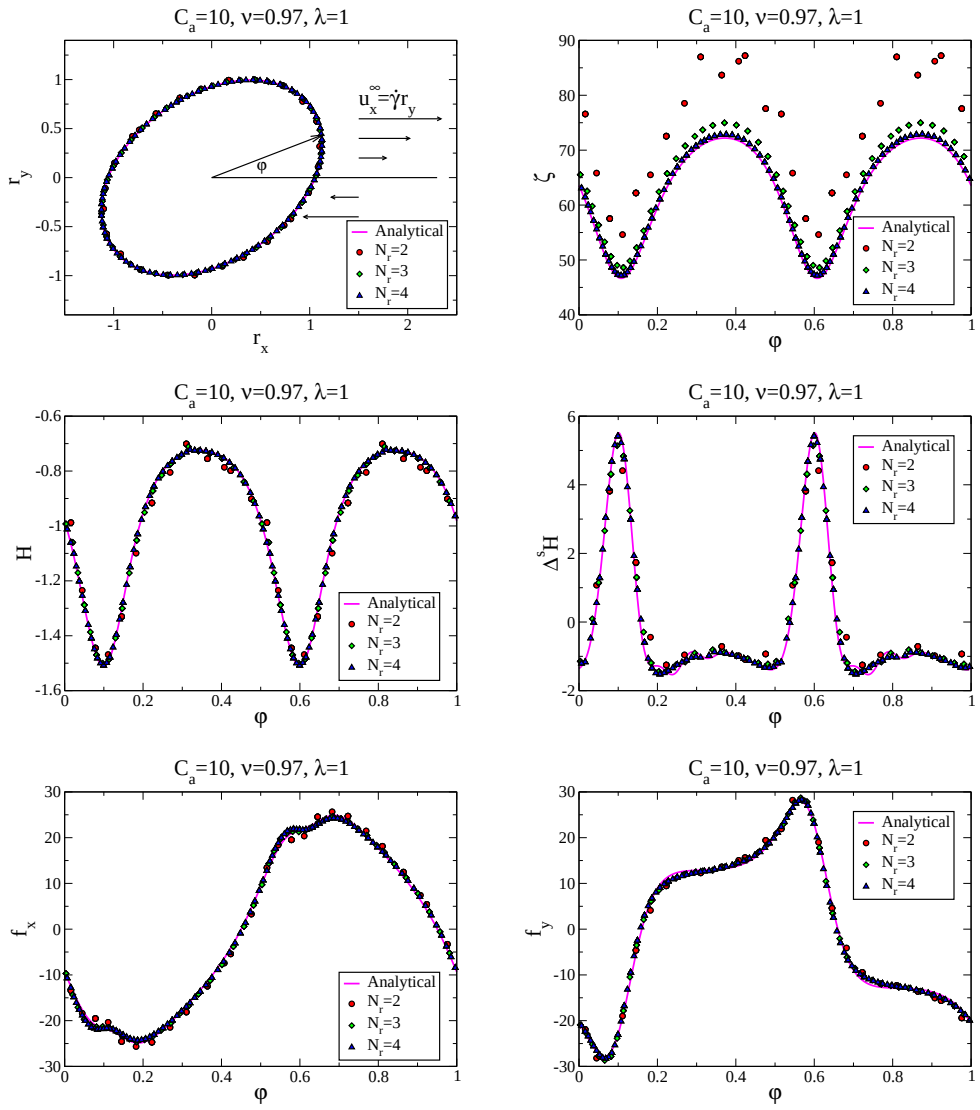

Figure 7: (color on-line) Various geometrical properties calculated for a steady state of a vesicle in simple shear flow. Numerical simulations with various degrees of refinement compared with high-order analytical expansions. $\nu = 0.97$, $C_a = 10$, $\lambda = 1$, $N_p = 0$. First row: Cross-section of the vesicle by the shear plane $r_z = 0$, imposed shear flow shown by horizontal arrows, polar angle φ in the shear plane is marked (left); Lagrange multiplier ζ in the cross-section by the shear plane $z = 0$, as a function of the polar angle φ (right). Second row: Mean curvature H (left) and its surface Laplacian $\Delta^s H$ (right) in the shear plane section as a function of the polar angle φ. Last row: membrane force f_x (left) and f_y (right) in the shear-plane section as a function of the polar angle φ. For the curvature Laplacian, the exceptional values at vertices with only 5 neighbors are excluded.

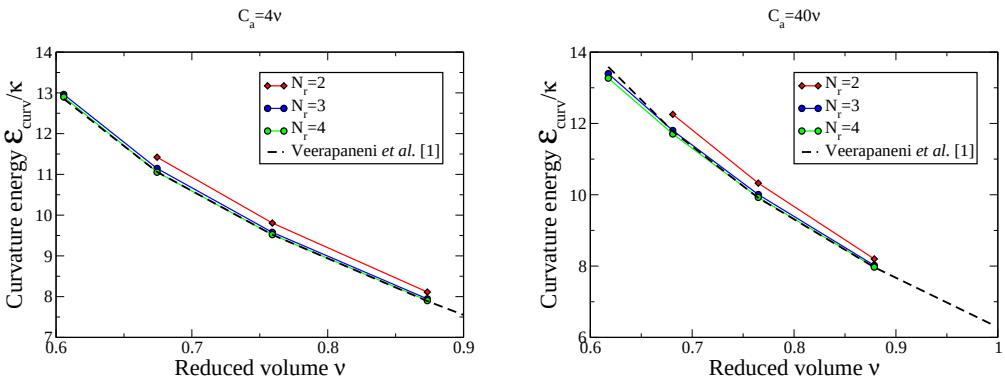

Figure 8: (color on-line) Energy of a vesicle in shear flow as a function of reduced volume for different number of vertices. $\lambda = 1$. Left: weak flow $C_a = 4\nu$. Right: Strong flow $C_a = 40\nu$. Dashed line: results of spectral BI method [1].

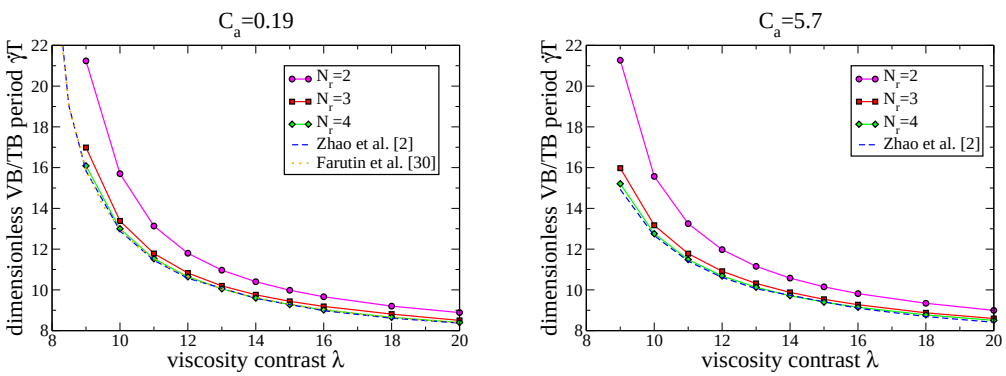

Figure 9: (color on-line) Dimensionless VB/TB period as a function of viscosity contrast for 3 different numbers of mesh vertices. Reduced volume $\nu = 0.95$. Capillary number $C_a = 0.19$ (left) and $C_a = 5.7$ (right). The numbers of mesh vertices are: $N_r = 2$, $N_p = 0$, $N_v = 162$ (purple circles), $N_r = 3$, $N_p = 0$, $N_v = 642$ (green squares), and $N_r = 4$, $N_p = 0$, $N_v = 2562$ (red diamonds). The results of spectral code [2] (blue dashed curve) and of high-order analytical expansions[30] (orange dotted curve for $C_a = 0.19$) are provided as a reference.

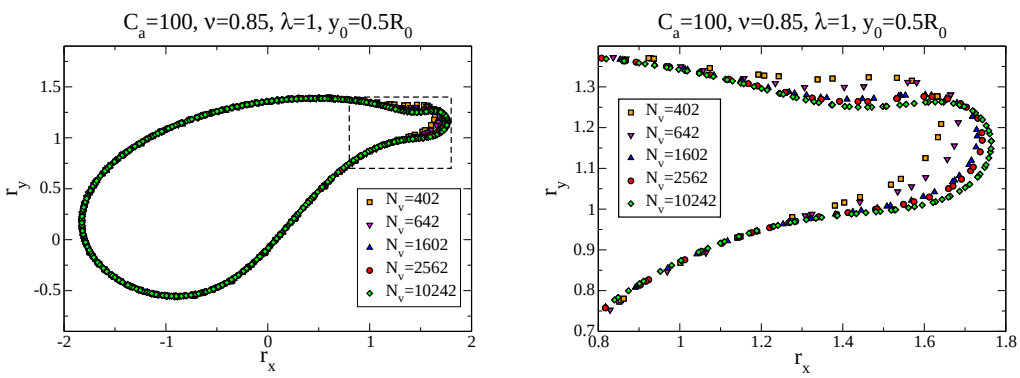

Figure 10: (color on-line) Cross-section of a vesicle in Poiseuille flow by a symmetry plane. $\nu = 0.85$, $C_a = 100$, $\lambda = 1$, $y_0 = 0.5R_v$. Left: The whole cross-section. Dashed box marks the tail region. Right: The tail region shown in detail.

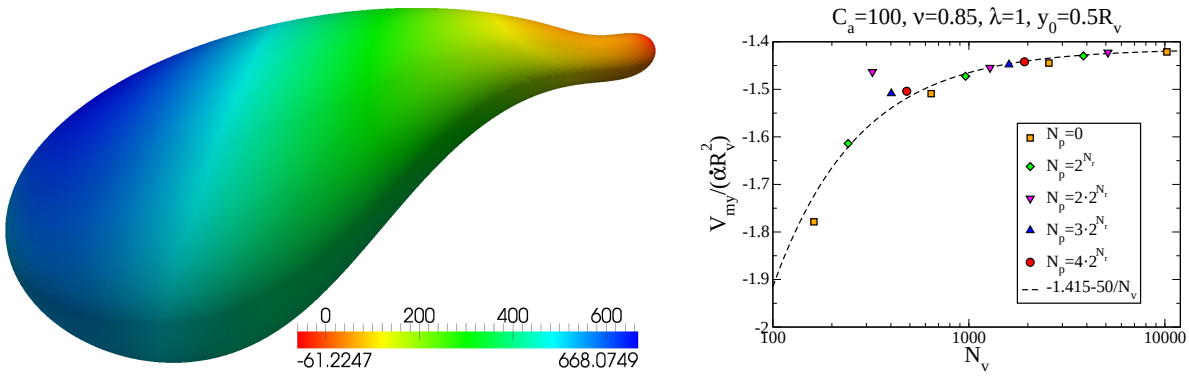

Figure 11: (color on-line) Vesicle fixed at distance $y_0 = 0.5R_v$ from the symmetry axis of a Poiseuille flow. $\nu = 0.85$, $C_a = 100$, $\lambda = 1$. Left: steady-state shape, color by Lagrange multiplier ζ. Right: Steady-state value of dimensionless migration velocity $V_{my}/(\dot{\alpha}R_v^2)$.

8.8. Inextensible capsules

Two simple tests are provided for simulations of inextensible capsules: stretching by optical tweezers and dynamics under simple shear flow. Because the quantitatively correct modelization of RBCs is an open question, we employ a rather simple model, which, however, allows us to reproduce many important features of RBC dynamics. In our model, the elastic properties of RBC membrane are characterized by two numbers: the shear modulus for small deformations μ_s and the maximum deformation parameter ε_m^2. The elastic energy of a capsule is written in the following "FENEM-like" form:

$$E_{el}(\varepsilon^2) = \frac{\mu_s \varepsilon_m^2}{2} \frac{1}{1 - \varepsilon^2/\varepsilon_m^2}. \tag{87}$$

At small deformation the energy is quadratic in ε (neo-Hookean) and becomes very stiff when ε^2 approaches ε_m^2. The following parameters are taken in all simulations: the volume-equivalent radius $R_v = 2.7\mu m$, (corresponding to volume $V \sim 82\mu m^3$, which is slightly lower than the median value reported in the literature[31]), the reduced volume $\nu = 0.65$, (one of the most commonly used values[31]), $\mu_s = 1.9\mu N/m$, which is slightly lower than the median value reported in the literature (e.g. in [32]), the maximum of allowed shear distortion is chosen as $\varepsilon_m^2 = 2.2$, and, finally, the bending modulus is chosen as $\kappa = 8.1 \cdot 10^{-19} J$, which is about 2.5 times higher than the value reported in experimental studies (e.g. in [33]). These values were chosen based on a simple trial and error fitting of the results of a stretching experiment on RBC[32] (Fig. 12, as explained below). The reference configuration is chosen to be the oblate equilibrium shape of a vesicle of given reduced volume (0.65 in this case). With this definition, the reference configuration remains at equilibrium when the full energy of the capsule is considered.

8.9. Stretching the RBCs by optical tweezers

The first test is inspired by an experiment[32], in which two small beads were attached at two opposite points of the rim of a RBC. One of the beads was then pulled in order to stretch the RBC. The axial (along the stretching force) and the transverse (perpendicular to the stretching force) diameters of the disc-like shape of the RBC were measured as a function of the stretching force. We reproduce the experiment by applying two concentrated forces in opposite regions of the rim of the RBC. In order to avoid numerical artifacts due to discontinuities of the applied force, the forces are taken to decrease smoothly from the maximum value in the epicenter vertex a to zero beyond the action radius R_a :

$$f_{ext}(\boldsymbol{r}) = \begin{cases} f_{ext}^0 \left[\frac{|\boldsymbol{r}-\boldsymbol{r}^a|^2}{R_a^2} - 1 \right]^2 & \text{if } |\boldsymbol{r} - \boldsymbol{r}^a| \le R_a \\ 0 & \text{if } |\boldsymbol{r} - \boldsymbol{r}^a| > R_a \end{cases}, \tag{88}$$

where $f_{ext}(\boldsymbol{r})$ is the amplitude of the applied force in a given point $\boldsymbol{r}$ and f_{ext}^0 is a parameter which allows to vary the applied force. Once the shape reaches saturation, the total force pulling on the RBC from each side is calculated

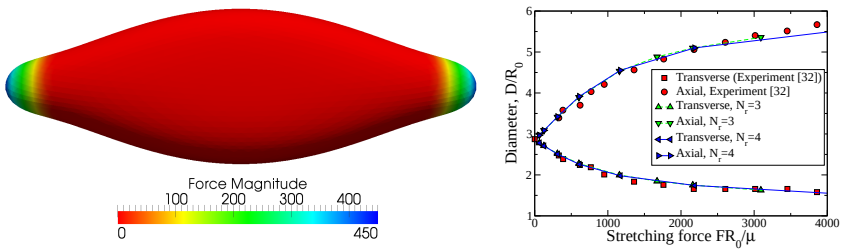

Figure 12: (color on-line) RBC stretched by optical tweezers. Left: Shape of a stretched RBC. Right: Cell diameters as a function of the stretching force.

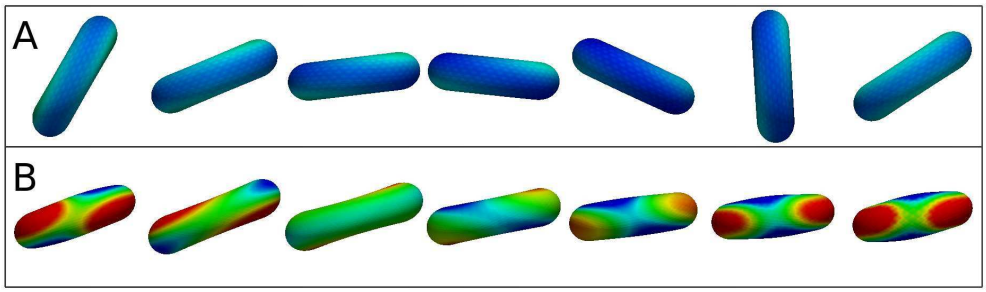

Figure 13: (color on-line) Dynamics of RBCs in simple shear flow. Color by Lagrange multiplier ζ. Snapshot A: tumbling. Snapshot B: tank-treading (with some breathing and oscillations of orientation). Changes in the Lagrange multiplier ζ due to membrane tank-treading are clearly seen.

and plotted against the resulting diameters (Fig. 12, right). In our simulations, $R_a = 0.7R_v$ was chosen. A sample distribution of applied forces on the surface of RBC is shown in Fig. 12, left.

As can be seen (Fig. 12, right) the sensitivity of the results to the vertex number is hardly noticeable. In addition, a good agreement with experiment is observed. In general, the agreement can be improved even further by choosing more carefully the values of physical parameters, the constitutive equation (87), and the distribution of the applied forces. This task, however, goes beyond the purpose of the present study.

8.10. Dynamics of RBCs in shear flow

Next, we consider dynamics of a RBC in shear flow. Unlike vesicles, RBCs are known to exhibit different behavior in shear flow depending on the shear rate even if the viscosities of internal and external liquids are equal ($\lambda = 1$). We use our model to reproduce this behavior. Figure 14 shows time dependence of the angle, which the longest direction from center of the capsule makes with the flow velocity. For high shear rate, the capsule in our simulations aligns with certain fixed direction showing small oscillations of elongation and orientation as the membrane tank-treads (tank-treading motion). For small shear rate, the longest direction from the center makes complete turns (tumbling motions). Snapshots of different motions are presented in Fig. 13.

9. Discussion and conclusion

We have presented a numerical algorithm for simulation of vesicles and inextensible capsules under flow. While being relatively simple and fast, the algorithm allows numerical treatment of many important problems in dynamics of vesicles and red bloods cells, including simulation of very deflated vesicles and capsules under flow. During various stages of development, the algorithm was employed with success to solve several physical problems[34, 30, 6, 29]. When available, analytical solutions have shown very good agreement with the results of numerical simulations.

The proposed algorithm is characterized by several important advances whose applicability is by no means limited to simulations of vesicles and inextensible capsules. Most notably, singularity subtraction for tangential forces

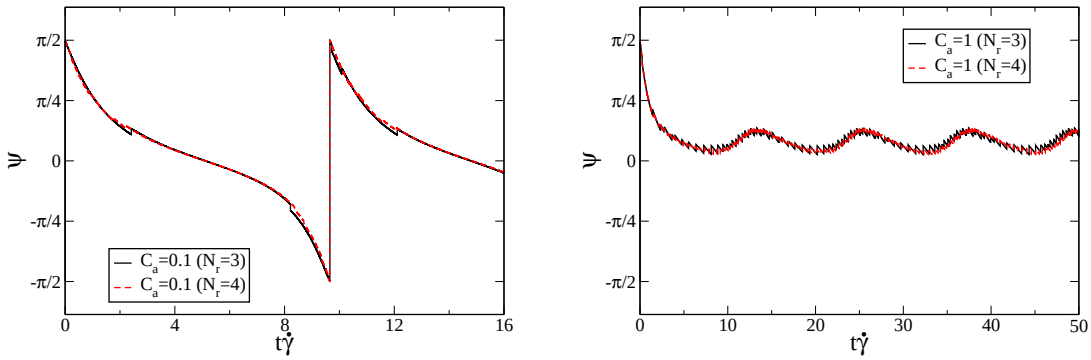

Figure 14: (color on-line) Dynamics of RBCs in simple shear flow. Transition form tumbling to tank-treading by increasing the shear rate. Angle between the longest projection of capsule shape on the shear plane and the flow velocity. Left: weak flow ($C_a = 0.1$). Right: stronger flow ($C_a = 1$). Small jumps in the orientation angle appear when the vertex whose projection on the *xy* plane was the furthest from the center at some time step is not the furthest from the center at the next time step.

has not been used before (to the best of our knowledge). Nevertheless, we have shown that this technique can be used efficiently to calculate BIs with low numerical error. The use of refined mesh greatly increases the stability of the algorithm and the precision of calculation when distance between two surfaces becomes comparable to the characteristic length scale of spatial resolution. The use of cut-off distance allows one to save computation time and the continuously differentiable weight function allows smooth splitting of calculations between the original and the refined meshes. Note that unlike for several previous implementations of the BI method[20, 2], where the splitting weight was a function of the distance on the reference sphere between the source and the target points, here we use the distance in the actual 3D space as the argument of the splitting weight function. This allows effectively the same treatment of the close-to-singularity situations, regardless if the source and the target belong to the same surface or not, including the cases when the surface is close to self-contact.

It must be noted, however, that singularity subtraction does not fully regularize the single-layer kernel, but only decreases the order of singularity by one. In other words, the application of singularity-subtracting transformation gives ”a free” multiplier of order $O(h)$ (provided, of course, that all non-singular calculations are performed with sufficient precision) to the numerical error (recall that h is the characteristic spatial resolution of the mesh). That is, by applying singularity subtraction, we have decreased the error of simple quadrature rule from $O(h)$ to $O(h^2)$. Likewise, we can speculate that applying the singularity-subtracting transformation in spectral method [20] could reduce the error from $O(h^3)$ to $O(h^4)$ while keeping the same number of quadrature nodes. However, we conjecture that using SS together with several refined meshes with different cut-off distances and a cascade lookup algorithm would be a more efficient way to implement a spectral method for boundary integral calculation with numerical error of order $O(h^4)$.

The singularity subtraction techniques developed here can be used with success in solving other types of Boundary integral equations, such as the 2D version of boundary integral equation for Stokes flow (as explained in Appendix A) or Poisson equation in electrostatics (as explained in Appendix B).

Several additional improvements contribute to the increased precision of the algorithm: instead of needing several iterations of quadratic fitting of the surface in order to find a good approximation for the normal and curvatures, the use of differential geometry allows one to find all the relevant quantities after just one iteration. The use of optimized penalization method to solve for the tension is somewhat more expensive with respect to the usual penalization technique (and clearly less expensive than the explicit resolution of the linear system), but ensures much better local and global inextensibility of the membrane and eliminates the constraint on the time-step related to the stiffness of the tension force. With correct implementation, calculation of BI for three force fields takes less than triple time of BI calculation for a single force field because the regularizing identities and the inverse distance between the source and the target points can be calculated only once. The residual local strain can be further decreased by performing more than one optimization iteration at each time step, however, decreasing the time step seems to be a better option.

While the proposed algorithm works well for a wide variety of situations, there exist problems to which it can

not be applied. The major limitation comes from the fixed topology of the mesh: Under strong deformations of the vesicle, the mesh triangles inevitably become distorted to the point when one of the angles becomes extremely acute. The penalization method loses its stability in this case, which results in excitation of high-wave-number modes in tension and subsequent failure of the simulation (this usually happens when the smallest angle of mesh triangles becomes about $5-10^\circ$). The explicit time-advancement scheme is stable only if the time step is small enough. The relaxation times of the bending force are proportional to h^{-3}, which results in severe limitations on the time step, especially if a large amount of discretization points is used and neighbor vertices come close to each other because of mesh distortion. Finally, while the results of the simulation are usually precise enough for most physical applications, a prohibitively large number of discretization points is required to obtain a reliable result in some cases. Higher order of spatial and temporal resolution could improve the precision of the algorithm. Solving these problems represents a promising task for future research.

We would like to thank P.-Y. Gires, M. Léonetti, S. Mendez, E.S.G. Shaqfeh, M. Thiébaud, and S. Veerapaneni for helpful discussions. We gratefully acknowledge financial support from CNES (Centre National d'Etudes Spatiales) and ESA (European Space Agency).

Appendix A. Singularity subtraction for 2D Stokes flow

Dynamics of a 2D suspension in Stokes limit is governed[11] by a BI similar to (2),

$$\frac{(\eta_{ext}+\eta^A_{int})}{2}u_i(\boldsymbol{r}) = \eta_{ext}u_i^\infty(\boldsymbol{r}) + \sum_B \oint_{\Sigma^B} G_{ij}(\boldsymbol{r},\boldsymbol{r}')f_j(\boldsymbol{r}')ds(\boldsymbol{r}')+ \tag{A.1}$$

$$+\sum_B(\eta_{ext}-\eta^B_{int})\oint_{\Sigma^B} T_{ijk}(\boldsymbol{r},\boldsymbol{r}')u_j(\boldsymbol{r}')n_k(\boldsymbol{r}')ds(\boldsymbol{r}').$$

Here, and throughout this Section we assume that the coordinate indices i, j, and k can take only two values, x and y. We denote by $ds(\boldsymbol{r})$ the arc length along the contour Σ resulting from a section of an infinite cylindrical vesicle by a plane perpendicular to $\boldsymbol{z}$. The difference with the 3D case is manifested only in the expression of the Green kernels[11]:

$$G_{ij}(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{4\pi}\left(-\delta_{ij}ln|\boldsymbol{r}-\boldsymbol{r}'| + \frac{(\boldsymbol{r}-\boldsymbol{r}')_i(\boldsymbol{r}-\boldsymbol{r}')_j}{|\boldsymbol{r}-\boldsymbol{r}'|^2}\right),\quad T_{ijk}(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{\pi}\frac{(\boldsymbol{r}-\boldsymbol{r}')_i(\boldsymbol{r}-\boldsymbol{r}')_j(\boldsymbol{r}-\boldsymbol{r}')_k}{|\boldsymbol{r}-\boldsymbol{r}'|^4}. \tag{A.2}$$

The kernels (A.2) are obtained by taking the principal value of the integral of (3) over z from $-\infty$ to ∞, assuming translational invariance of the problem along the z direction. As can be seen from (A.2), the kernel G has a special point for $\boldsymbol{r}'=\boldsymbol{r}$, where it diverges. It is thus advantageous for the precision of numerical calculation of the BI to perform the SS.

In general, it is possible to obtain the regularizing identities by the same integration over z of (17) and (18). There is, however, a simpler way to perform the SS in 2D case. Namely, we consider the following integral identities:

$$\boldsymbol{n}(\boldsymbol{r})\cdot\oint G(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r}')ds(\boldsymbol{r}') = 0, \tag{A.3}$$

$$\boldsymbol{t}(\boldsymbol{r})\cdot\oint_\Sigma G(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{t}(\boldsymbol{r}')ds(\boldsymbol{r}') + \frac{\boldsymbol{n}(\boldsymbol{r})}{2\pi}\oint_\Sigma \frac{(\boldsymbol{r}-\boldsymbol{r}')[(\boldsymbol{r}-\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r}')]}{|\boldsymbol{r}-\boldsymbol{r}'|^2}ds(\boldsymbol{r}') = 0, \tag{A.4}$$

where $\boldsymbol{t}$ is the tangent vector to the contour Σ. The SS transformation for the single-layer kernel then goes as follows:

$$\oint_\Sigma G(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{f}(\boldsymbol{r}')ds(\boldsymbol{r}') = \oint_\Sigma[G(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{f}(\boldsymbol{r}') - \boldsymbol{f}_n(\boldsymbol{r})\boldsymbol{n}(\boldsymbol{r})\cdot G(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r}') - \boldsymbol{f}_t(\boldsymbol{r})\boldsymbol{t}(\boldsymbol{r})\cdot G(\boldsymbol{r},\boldsymbol{r}')\cdot\boldsymbol{t}(\boldsymbol{r}')]ds(\boldsymbol{r}')- \tag{A.5}$$

$$-\frac{\boldsymbol{f}_t(\boldsymbol{r})}{2\pi}\oint_\Sigma\frac{[(\boldsymbol{r}-\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r})][(\boldsymbol{r}-\boldsymbol{r}')\cdot\boldsymbol{n}(\boldsymbol{r}')]}{|\boldsymbol{r}-\boldsymbol{r}'|^2}ds(\boldsymbol{r}'),$$

where $\boldsymbol{f}_n = (\boldsymbol{f}\cdot\boldsymbol{n})\boldsymbol{n}$ and $\boldsymbol{f}_t = (\boldsymbol{f}\cdot\boldsymbol{t})\boldsymbol{t}$ are, respectively, the normal and the tangential parts of the force $\boldsymbol{f}$. It can be easily checked that the integrands of both integrals on the right hand side of (A.5) go to zero as $\boldsymbol{r}'$ approaches $\boldsymbol{r}$ along the contour Σ.

Appendix B. Singularity subtraction for Poisson equation in 3D

A similar technique of singularity subtraction can be applied to other physical problems: Consider a problem finding the distribution of surface charges $\sigma(\boldsymbol{x})$ in a conducting body placed into an external potential $\varphi^\infty(\boldsymbol{r})$. This problem would require solving a boundary integral equation

$$\phi^\infty(\boldsymbol{r}) - \frac{1}{4\pi}\int_\Sigma \frac{\sigma(\boldsymbol{r}')d^2r'}{|\boldsymbol{r}-\boldsymbol{r}'|} = C \tag{B.1}$$

for every point $\boldsymbol{r}$ on the surface Σ of the conducting body. Here C is a constant acting as a Lagrange multiplier associated to the charge neutrality condition

$$\int_\Sigma \sigma(\boldsymbol{r}')d^2r' = 0. \tag{B.2}$$

The boundary integral in (B.1) has an integrand that diverges for $\boldsymbol{r}' = \boldsymbol{r}$ but can be regularized by the following transformation:

$$\int_\Sigma \frac{\sigma(\boldsymbol{r}')d^2r'}{|\boldsymbol{r}-\boldsymbol{r}'|} = \int_\Sigma \frac{\sigma(\boldsymbol{r}')-\sigma(\boldsymbol{r})[\boldsymbol{n}(\boldsymbol{r})\cdot\boldsymbol{n}(\boldsymbol{r}')]d^2r'}{|\boldsymbol{r}-\boldsymbol{r}'|} - \int_\Sigma \frac{\sigma(\boldsymbol{r})[\boldsymbol{n}(\boldsymbol{r})\cdot(\boldsymbol{r}-\boldsymbol{r}')][\boldsymbol{n}(\boldsymbol{r}')\cdot(\boldsymbol{r}-\boldsymbol{r}')]d^2r'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}, \tag{B.3}$$

where we have used the fact (cf. (17)) that

$$\sigma(\boldsymbol{r})\boldsymbol{n}(\boldsymbol{r})\int\left\{\frac{\boldsymbol{n}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} + \frac{(\boldsymbol{r}-\boldsymbol{r}')[\boldsymbol{n}(\boldsymbol{r}')\cdot(\boldsymbol{r}-\boldsymbol{r}')]}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\right\}d^2r' = 0. \tag{B.4}$$

It is easy to check that the integrands of both integrals on the right hand side of (B.3) remain continuous and bounded as $\boldsymbol{r}'$ approaches $\boldsymbol{r}$ along the surface $\boldsymbol{\Sigma}$, which allows one to use a simple quadrature rule to calculate the integral (B.3) with precision of order $O(h^2)$.

References

[1] S. K. Veerapaneni, A. Rahimian, G. Biros, D. Zorin, A fast algorithm for simulating vesicle flows in three dimensions, Journal of Computational Physics 230 (2011) 5610 – 5634.
[2] H. Zhao, E. S. G. Shaqfeh, The dynamics of a vesicle in simple shear flow, Journal of Fluid Mechanics 674 (2011) 578–604.
[3] A. Farutin, T. Biben, C. Misbah, Analytical progress in the theory of vesicles under linear flow, Phys. Rev. E 81 (2010) 061904.
[4] Z.-H. Huang, M. Abkarian, A. Viallat, Sedimentation of vesicles: from pear-like shapes to microtether extrusion, New Journal of Physics 13 (2011) 035026.
[5] G. Boedec, M. Leonetti, M. Jaeger, 3d vesicle dynamics simulations with a linearly triangulated surface, Journal of Computational Physics 230 (2011) 1020 – 1034.
[6] A. Farutin, C. Misbah, Squaring, parity breaking, and S tumbling of vesicles under shear flow, Phys. Rev. Lett. 109 (2012) 248106.
[7] V. Kantsler, E. Segre, V. Steinberg, Critical dynamics of vesicle stretching transition in elongational flow, Phys. Rev. Lett. 101 (2008) 048101.
[8] S. Sukumaran, U. Seifert, Influence of shear flow on vesicles near a wall: A numerical study, Phys. Rev. E 64 (2001) 011916.
[9] T. Biben, A. Farutin, C. Misbah, Three-dimensional vesicles under shear flow: Numerical study of dynamics and phase diagram, Phys. Rev. E 83 (2011) 031921.
[10] H. Zhao, A. P. Spann, E. S. G. Shaqfeh, The dynamics of a vesicle in a wall-bound shear flow, Physics of Fluids 23 (2011) 121901.
[11] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, 1992.
[12] W. Helfrich, Elastic properties of lipid bilayers: Theory and possible experiments, Z. Naturforsch. C 28 (1973) 693.
[13] O.-Y. Zhong-can, W. Helfrich, Bending energy of vesicle membranes: General expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders, Phys. Rev. A 39 (1989) 5280–5288.
[14] A. Laadhari, C. Misbah, P. Saramito, On the equilibrium equation for a generalized biological membrane energy by using a shape optimization approach, Physica D 239 (2010) 1567–1572,.
[15] M. Zhao, P. Bagchi, Dynamics of microcapsules in oscillating shear flow, Physics of Fluids 23 (2011) 111901.

[16] M. G. Duffy, Quadrature over a pyramid or cube of integrands with a singularity at a vertex, SIAM Journal on Numerical Analysis 19 (1982) 1260–1262.
[17] M. Loewenberg, E. J. Hinch, Numerical simulation of a concentrated emulsion in shear flow, Journal of Fluid Mechanics 321 (1996) 395–419.
[18] C. Pozrikidis, Interfacial dynamics for stokes flow, Journal of Computational Physics 169 (2001) 250 – 301.
[19] O. P. Bruno, L. A. Kunyansky, A fast, high-order algorithm for the solution of surface scattering problems: Basic implementation, tests, and applications, Journal of Computational Physics 169 (2001) 80 – 110.
[20] H. Zhao, A. H. Isfahani, L. N. Olson, J. B. Freund, A spectral boundary integral method for flowing blood cells, Journal of Computational Physics 229 (2010) 3726 – 3744.
[21] S. K. Doddi, P. Bagchi, Lateral migration of a capsule in a plane poiseuille flow in a channel, Int. J. Multiphase Flow 34 (2008) 966–986.
[22] E. Lac, A. Morel, D. Barthès-Biesel, Hydrodynamic interaction between two identical capsules in simple shear flow, J. Fluid Mech. 573 (2007) 149–169.
[23] P. Bagchi, R. M. Kalluri, Dynamics of nonspherical capsules in shear flow, Physical Review E 80 (2009) 016307.
[24] D.-V. Le, Z. Tan, Large deformation of liquid capsules enclosed by thin shells immersed in the fluid, J. Comput, Phys. 229 (2010) 4097–4116.
[25] J. Walter, A.-V. Salsac, D. Barthès-Biesel, P. Le Tallec, Coupling of finite element and boundary integral methods for a capsule in a stokes flow, International Journal for Numerical Methods in Engineering 83 (2010) 829–850.
[26] D. Barthès-Biesel, Capsule motion in flow: Deformation and membrane buckling, C.R. Physique 10 (2009) 764–774.
[27] D. Barthès-Biesel, J. M. Rallison, The time-dependent deformation of a capsule freely suspended in a linear shear flow, Journal of Fluid Mechanics 113 (1981) 251–267.
[28] R. Skalak, A. Tozeren, R. Zarda, S. Chien, Strain energy function of red blood cell membranes, Biophys. J. 13 (1973) 245–264.
[29] A. Farutin, C. Misbah, Analytical and numerical study of three main migration laws for vesicles under flow, Phys. Rev. Lett. 110 (2013) 108104.
[30] A. Farutin, O. Aouane, C. Misbah, Vesicle dynamics under weak flows: Application to large excess area, Phys. Rev. E 85 (2012) 061922.
[31] P. B. Canham, A. C. Burton, Distribution of size and shape in populations of normal human red cells, Circulation Research 22 (1968) 405–422.
[32] J. P. Mills, L. Qie, M. Dao, C. T. Lim, S. Suresh, Nonlinear elastic and viscoelastic deformation of the human red blood cell with optical tweezers, Mechanics and chemistry of biosystems 1 (2004) 169–180.
[33] T. Betz, M. Lenz, J.-F. Joanny, C. Sykes, Atp-dependent mechanics of red blood cells, Proceedings of the National Academy of Sciences 106 (2009) 15320–15325.
[34] G. Coupier, A. Farutin, C. Minetti, T. Podgorski, C. Misbah, Shape diagram of vesicles in poiseuille flow, Phys. Rev. Lett. 108 (2012) 178106.