

A Cartesian-octree adaptive front-tracking solver for immersed biological capsules in large complex domains

Damien P. Huet^a, Anthony Wachs^{a,b,*}

^a*Department of Mathematics, University of British Columbia, 1984 Mathematics Road, BC V6T 1Z2, Vancouver, Canada*

^b*Department of Chemical & Biological Engineering, University of British Columbia, 2360 E Mall, BC V6T 1Z3, Vancouver, Canada*

Abstract

We present an open-source adaptive front-tracking solver for biological capsules in viscous flows. The membrane elastic and bending forces are solved on a Lagrangian triangulation using a linear Finite Element Method and a paraboloid fitting method. The fluid flow is solved on an octree adaptive grid using the open-source platform Basilisk. The Lagrangian and Eulerian grids communicate using an Immersed Boundary Method by means of Peskin-like regularized Dirac delta functions. We demonstrate the accuracy of our solver with extensive validations: in Stokes conditions against the Boundary Integral Method, and in the presence of inertia against similar (but not adaptive) front-tracking solvers. Excellent qualitative and quantitative agreements are shown. We then demonstrate the robustness of the present solver in a challenging case of extreme membrane deformation, and illustrate its capability to simulate inertial capsule-laden flows in complex STL-defined geometries, opening the door for bioengineering applications featuring large three-dimensional channel structures. The source code and all the test cases presented in this paper are freely available.

1. Introduction

The numerical study of membrane-enclosed fluid objects, or capsules, has seen tremendous interest over the past three decades due to the wide range of applications in the biomedical and bioengineering world. Indeed, numerical simulation of capsule dynamics in viscous flows is crucial to better characterize and understand blood flow through capillary microcirculation and develop applications such as targeted drug delivery [1], migration of cancerous leukocytes through the microvascular network [2, 3] and cell sorting and cell characterization in microfluidic devices [4, 5]. In particular, the latter application has

*Corresponding author

Email addresses: huet@math.ubc.ca (Damien P. Huet), wachs@math.ubc.ca (Anthony Wachs)

the potential to speed-up labour-intensive diagnosis procedures or to extract relevant components of biofluids. For instance inertial centrifugation in spiral-shaped microchannels has been shown to efficiently and accurately segregate cells based on their size, and could be applied to perform non-destructive blood plasma extraction [6, 7, 8].

The study of capsules from a mechanical point of view was paved in 1981 by the pioneering analytical work of Barthès-Biesel & Rallison [9], who derived from the thin-shell theory a time-dependant expression for the deformation of an elastic capsule in a shear flow in the limit of small deformations. A decade later, Pozrikidis went beyond the assumption of small deformations, using the Boundary Integral Method (BIM) to investigate finite deformations of elastic capsules in a shear flow [10, 11]. This work was quickly followed by Eggleton & Popel who simulated spherical and biconcave capsules in shear flows using the Front-Tracking Method (FTM) [12]. Capitalizing on the advantages of the BIM – such as a lower computing cost compared to the FTM, and the ability to simulate true Stokes conditions – Pozrikidis investigated the bending resistance of capsules and proposed a simplified bending model for biological membranes valid for small deformations [13], leading to the first numerical simulation of an RBC based on the thin-shell theory [14]. The work of Pozrikidis was later extended by Zhao et al., who proposed a BIM able to simulate RBCs in complex geometries with up to 30% volume fraction [15]. In the 2000s, Barthès-Biesel and Lac also used the BIM and studied finite deformations of capsules devoid of bending resistance: they considered the effect of the membrane constitutive law and exhibited buckling instabilities [16] as well as the dynamics of two interacting capsules in a shear flow [17].

Despite the major success of the BIM to simulate capsules and biological cells, the FTM is still being developed. Indeed, while the FTM is more computationally intensive than the BIM because it necessitates meshing the whole 3D fluid domain, and while it can require very small time steps to satisfy stability conditions depending on the considered membrane forces; the FTM can handle inertial regimes, thus allowing to examine a wider range of applications. As such, Bagchi uses the FTM to perform two-dimensional simulations of several thousand RBCs in a shear flow [18], allowing the study of RBC interactions at the mesoscale. In the next years, Doddi & Bagchi [19] and Yazdani & Bagchi [20, 21, 22] develop respectively three-dimensional implementations of the elastic membrane stress and of the Helfrich’s bending stress for biological membranes, the latter not being limited to small deformations as was the case for the formulation of Pozrikidis. The ability to consider finite Reynolds numbers allowed Doddi & Bagchi to extend the work of Lac et al. on capsule interactions to inertial flows [23]. Their framework was later extended to complex geometries by Balogh & Bagchi [24], enabling them to study the dynamics of hundreds of RBCs in a microvascular network with a hematocrit (volume fraction of RBCs) of 30% [25]. Another variant of the FTM is to use a Lattice-Boltzman fluid solver rather than the traditional PDE-based Navier-Stokes solver: this can bring significant performance improvement especially at low Reynolds number where the Lattice-Boltzman Method (LBM) performs well. For instance, Li &

Sarkar extend the work of Barthès-Biesel and Lac on the instabilities of elastic capsules in shear flows using an FTM-LBM solver [26], and Zhang et al. describe a similar framework able to simulate RBCs [27], including cell-cell aggregation phenomena [28]. More recently, Ames et al. [29] harnessed the performance improvements of GPUs and demonstrated an impressive 17 million RBCs simulated in a microvascular network using a similar FTM-LBM framework.

Other methods to simulate biological capsules and vesicles include the RBC model of Fedosov et al. [30]. In the work of Fedosov, the RBC membrane mechanics is not governed by the thin shell theory but rather by a coarse-grained molecular dynamics model. The membrane of the RBC is discretized, with each edge representing several nonlinear springs which correspond to elastic and viscoelastic properties of the membrane of the RBC. The model parameters are found for an extremely fine mesh of over 27000 nodes, where the lengths of the edges correspond to that of biological spectrins. Yet, in Fedosov’s model practical RBC simulations are conducted with different model parameters which are intended to display the same mechanical behavior as that obtained with a fine mesh, but using a number of nodes orders of magnitude lower. These coarse-grained mechanical properties of the membrane are shown to lead to results which accuracy lies within the range of experimental measurement errors for the specific cases considered. However the range of validity of this coarse-graining step is not obvious and the spatial convergence can be non-monotonous or even not exist (see the transverse diameter plot in figure 1 in [30]), indicating to use this model with care. A last approach to capsule simulations is to adopt a fully Eulerian framework, where the membrane is not discretized with Lagrangian nodes, edges and faces. Instead, the capsule configuration is described using the Eulerian grid employed to solve the Navier-Stokes equations. Removing the need of a Lagrangian grid is a desirable description as the IBM can reduce the spatial and temporal accuracies to first order if no special treatment is implemented. In the Eulerian capsule description, Volume-Of-Fluid (VOF), level-set or phase-field methods can be utilized to track the position of the membrane, similar to what is done in the context of fluid-fluid interfaces [31, 32, 33, 34]. If the considered membrane mechanical behavior is independent of the past configuration of the membrane, for instance if there is no resistance to shear and high resistance to bending, the membrane forces can be computed using techniques developed for surface-tension flows: the local curvature can be computed using height-functions in the case of a VOF description, or by numerically differentiating the level-set or phase-field function near the interface [35]. However, in most biological applications the membrane properties do depend on the past membrane configurations due to its elastic behavior. In such cases a quantity representing the membrane stretch needs to be initialized and advected in the vicinity of the membrane, for instance the left Cauchy-Green deformation tensor. Ii et al. have demonstrated that this approach is possible and scalable [36, 37, 38], although more comparisons with the FTM are needed in order to evaluate the performance and the accuracy of the Eulerian methods, especially for long-lasting simulation.

Concomitant to these developments of capsule simulations, the IBM gained

great popularity in the particle-laden flows community [39, 40, 41], and in the past two decades some adaptive IBM have been proposed in cases of immersed solid particles. In this context, Roma et al. [42] present a two-dimensional adaptive IBM implementation where the Adaptive Mesh Refinement (AMR) is achieved by means of overlapping rectangles – or “patches” – of finer grid cells in the regions of interest, i.e. where higher accuracy of the flow field is needed. This method was later improved and proved second-order accurate by Griffith et al. [43] and Vanella et al. [44]. Previously, Agreasar et al. [45] had used a non-patched adaptive FTM-IBM method in order to simulate axisymmetric circular cells. Their IBM implementation did not use Peskin-like regularized Dirac delta functions: instead the Lagrangian grid on the membrane communicates with the background Eulerian grid via an area-weighted extrapolation. More recently, Cheng & Wachs [46] used the IBM coupled with an LBM solver to achieve adaptive simulations in the case of a single rigid sphere in various flow conditions.

The goal of this paper is to present an efficient framework to study the dynamics of dilute suspensions of capsules in complex geometries, not limited to non-inertial regimes. As the BIM cannot be used at finite Reynolds numbers, we use the FTM and therefore the whole 3D fluid domain is discretized. Since a vast range of realistic applications consider geometries of sizes orders of magnitude larger than the typical size of a capsule, requiring hundreds of millions to billions of Eulerian grid cells when the Cartesian grid has a constant grid size, we develop an adaptive FTM solver rendering achievable to simulate configurations that were previously out of reach with a constant grid size. We provide the open-source code as part of the Basilisk platform [47, 34, 48, 49].

The paper is organized as follows: in Section 2 we present the problem formulation and the governing equations for both the fluid and the capsule dynamics. We describe the implementation of our numerical model in Section 3, emphasizing the finite element membrane model and the FTM method. Numerous validation cases are shown in Section 4, for increasingly difficult configurations: validations are performed by comparing our computed results against accurate BIM data available in literature whenever possible, otherwise against other FTM results. Section 5 contains new results generated with the present method, where the adaptive mesh capability dramatically improves computational efficiency. In Section 6 we summarize our work and discuss the strengths, weaknesses and possible improvements of the present method as well as future perspectives.

2. Governing equations

2.1. Fluid motion

The fluid phase is assumed Newtonian and incompressible: the fluid surrounding and enclosed by the elastic membranes is described using the mixture Navier-Stokes equations:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot (\mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) + \mathbf{f}_b \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

where \mathbf{u} is the velocity, p is the pressure, ρ is the constant density and μ is the variable viscosity field, since we will consider non-unity viscosity ratios $\lambda_\mu = \mu_i/\mu_e \neq 1$, with μ_i and μ_e the internal and external viscosities. \mathbf{f}_b denotes the body force containing the membrane elastic and bending force densities acting on the fluid: $\mathbf{f}_b = \mathbf{f}_{\text{elastic}} + \mathbf{f}_{\text{bending}} = (\mathbf{F}_{\text{elastic}} + \mathbf{F}_{\text{bending}})/V$, with V a relevant control volume and $\mathbf{F}_{\text{elastic}}$ and $\mathbf{F}_{\text{bending}}$ the integrated membrane force densities.

2.2. Membrane mechanics

We assume that the lipid-bilayer membrane is infinitely thin: please note that this is not a strong assumption for most biological cells, as thickness of the biological membrane (lipid-bilayer) is $5nm$ while an RBC characteristic size is $10\mu m$. A biological membrane undergoing deformation responds with elastic and bending stresses, described with two distinct mechanical models.

The elastic strains and stresses are described using the theory of thin shells[50]. We summarize this framework here, but the interested reader is referred to the work of [9] for more details. In this continuous description of the capsule, we first introduce the projectors $\mathbf{P} = \mathbf{I} - \mathbf{n}\mathbf{n}$ and $\mathbf{P}_R = \mathbf{I} - \mathbf{n}_R\mathbf{n}_R$ onto the current and reference (stress-free) membranes shapes, with \mathbf{I} the identity tensor and \mathbf{n} and \mathbf{n}_R the unit normal vectors to the current and reference membranes configurations, which are both oriented outward. The membrane strains are described using the surface deformation gradient tensor \mathbf{F}_s , derived from the classical deformation gradient tensor \mathbf{F} as follows:

$$\mathbf{F}_s = \mathbf{P} \cdot \mathbf{F} \cdot \mathbf{P}_R. \quad (3)$$

The surface right Cauchy-Green deformation tensor \mathbf{C}_s is then defined from \mathbf{F}_s :

$$\mathbf{C}_s = \mathbf{F}_s^T \cdot \mathbf{F}_s. \quad (4)$$

Let the three eigenvalues of \mathbf{F}_s be $\lambda_1, \lambda_2, 0$ associated with the eigenvectors $\mathbf{t}_1, \mathbf{t}_2, \mathbf{n}$. Then the eigenbasis of \mathbf{C}_s is the same as that of \mathbf{F}_s , associated with eigenvalues $\lambda_1^2, \lambda_2^2, 0$. Note that in the stress-free configuration, at the beginning of a typical simulation, $\lambda_1 = \lambda_2 = 1$, and $\mathbf{F}_s = \mathbf{C}_s = \mathbf{P} = \mathbf{P}_R$.

The above quantities are useful to compute the membrane elastic stress, which can be expressed using a surface strain-energy function $W_s(\lambda_1, \lambda_2)$:

$$\sigma_i = \frac{1}{\lambda_j} \frac{\partial W_s}{\partial \lambda_i}, \quad i \neq j. \quad (5)$$

In this work, two distinct strain-energy functions corresponding to two membrane elastic laws are used to describe several types of lipid bilayers in various conditions:

1. The Neo-Hookean law, used to describe vesicles and artificial capsules, and which corresponding strain-energy function is

$$W_s^{NH} = \frac{E_s}{6} \left(\lambda_1^2 + \lambda_2^2 + \frac{1}{\lambda_1^2 \lambda_2^2} - 3 \right), \quad (6)$$

where E_s denotes the shear modulus.

2. The Skalak law, used to describe the elastic response of RBC membranes, and which strain energy function is

$$W_s^{Sk} = \frac{E_s}{4} (I_1^2 + 2I_1 - 2I_2 + CI_2^2), \quad (7)$$

where the invariants $I_1 = \lambda_1^2 + \lambda_2^2 - 2$ and $I_2 = \lambda_1^2 \lambda_2^2 - 1$ have been introduced, as well as the area dilatation modulus C preventing strong area changes and is taken “large” [10, 51] in order to describe the strong area incompressibility of RBCs. Unless otherwise stated, the value $C = 10$ is used in the simulation results presented below.

Once the elastic stress is known, the elastic force exerted by the membrane onto the fluid is simply

$$\mathbf{F}_{\text{elastic}} = \nabla \cdot \boldsymbol{\sigma}, \quad (8)$$

although we will follow the approach of [52] and use the principle of virtual work instead of directly computing the divergence of the stress, as explained in Section 3.4.1.

The bending stresses are described using Helfrich’s bending energy per unit area \mathcal{E}_B [53]:

$$\mathcal{E}_B = 2E_b (\kappa - \kappa_0)^2 + E_g \kappa_g \quad (9)$$

where $\kappa = (\kappa_1 + \kappa_2)/2$ is the local mean curvature, $\kappa_g = \kappa_1 \kappa_2$ is the Gaussian curvature, and E_b and E_g are their associated bending moduli. κ_1 and κ_2 are the two principal curvatures at a given point of the two-dimensional membrane sheet, and κ_0 is the reference curvature. Then, the bending stresses are derived from the total bending energy $\int_{\Gamma} (2E_b (\kappa - \kappa_0)^2 + E_g \kappa_g) dS$ by means of a variational derivative, to yield the normal bending force per unit area [54]:

$$\mathbf{F}_{\text{bending}}/A = -2E_b (\Delta_s (\kappa - \kappa_0) + 2(\kappa - \kappa_0)(\kappa^2 - \kappa_g + \kappa_0 \kappa)) \mathbf{n}, \quad (10)$$

where the operator Δ_s is the surface Laplacian – or Laplace-Beltrami operator – defined as $\Delta_s = \nabla_s \cdot \nabla_s = ((\mathbf{I} - \mathbf{n}\mathbf{n}) \cdot \nabla) \cdot ((\mathbf{I} - \mathbf{n}\mathbf{n}) \cdot \nabla)$, and A is a relevant control area. Note how E_g has disappeared in the variational formulation because κ_g is a topological invariant [55, 54].

At this point a parallel with surface tension forces is enlightening: the bending energy related to surface tension acting on a droplet is proportional to the area of the interface, and leads to surface tension forces proportional to the curvature, i.e. to the second derivative of the interface geometry. In contrast, as stated above the bending energy related to lipid-bilayer membranes is proportional to the curvature, and thus the corresponding bending force depends

on the second derivative of the curvature, i.e. to fourth-order derivatives of the geometry. As such, the numerical simulations of biological capsules subject to bending stresses is a formidable challenge, and the interested reader is referred to the reviews of [56, 54]. Our approach to computing the bending force is described in Section 3.4.2, and corresponds to method E in [56].

3. Numerical method

3.1. Adaptive Finite Volume solver for the Navier-Stokes equations

Assuming the body force field \mathbf{f}_b is known, Eq. (1–2) are solved using the open-source platform Basilisk [48]. The viscous term $\nabla \cdot (\mu(\nabla u + (\nabla u)^T))$ is treated implicitly using a multigrid Poisson solver [34, 47], the incompressibility condition is satisfied by the classical projection method of Chorin [57], and the advection term $\mathbf{u} \cdot \nabla \mathbf{u}$ is solved using the second-order Bell-Colella-Glaz upwind advection scheme [58]. To this end, the divergence-free velocity field and the viscosity field are located on cell faces while a velocity field approximately divergence-free, the pressure field, and the body force field are all defined on the cell centers.

Basilisk computes the solution of the Navier-Stokes equations on an octree grid, which allows to coarsen and refine computational cells throughout the simulations while keeping a structured mesh. The coarsening and refinement of grid cells is implemented using a wavelet-based algorithm: for the sake of completeness we present here a short overview of the adaptivity process, and the interested reader is referred to [47, 34, 59] for more in-depth descriptions. Let f be a field of interest which variations in space will govern the size of the grid cells. First, f is downsampled onto a lower level grid by volume-averaging: we call this downsampled field f_d . Then, f_d is upsampled back to the original grid using second-order interpolations, resulting in the downsampled-then-upsampled field f_{du} . Since f and f_{du} are defined on the same (original) grid, the sampling error field $\epsilon^i = \|f^i - f_{du}^i\|$ can be defined in each computational cell i . Finally, ϵ^i is used to decide if cell i should be coarsened or refined based on an adaptivity criterion $\zeta > 0$:

$$\begin{cases} \epsilon^i > \zeta & \Rightarrow \text{refine cell } i \\ \epsilon^i < \frac{2\zeta}{3} & \Rightarrow \text{coarsen cell } i \\ \frac{2\zeta}{3} \leq \epsilon^i \leq \zeta & \Rightarrow \text{leave cell } i \text{ at its current level.} \end{cases} \quad (11)$$

This wavelet-based algorithm is very versatile as any field of interest can be used to influence the refinement level of the octree grid. In this study, the adaptivity is based on the velocity field and on both the presence of domain boundaries and the capsule membrane. In other words, the fields of interest that the wavelet adaptivity algorithm considers are: u_x, u_y, u_z, c_s and ξ , where the first three scalar fields are the three components of the velocity field \mathbf{u} , c_s is the fluid volume fraction field (in case of complex geometries), and ξ is a scalar field which varies strongly in the vicinity of the membrane and is constant

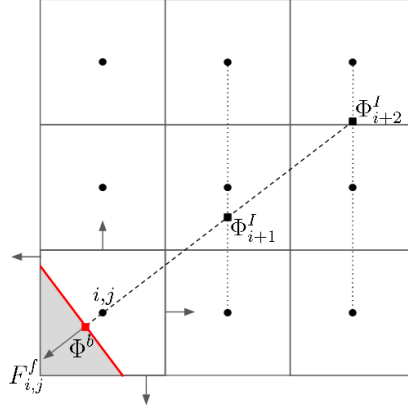


Figure 1: A two-dimensional example of an interpolation stencil estimating the boundary flux Φ^b with second-order accuracy. The gray area denotes the solid, the red line denotes the solid boundary, and the arrows denote the five face fluxes that are computed in the cut cell of interest i, j . The circle dots denote the Eulerian cell centers while the square dots show the locations of the data points Φ from which the boundary flux $F_{i,j}^f$ is interpolated. The interpolation line normal to the solid boundary is represented by a dashed line. In case of Dirichlet boundary conditions, Φ^b is known but the data points Φ_{i+1}^I and Φ_{i+2}^I are themselves interpolated with second-order accuracy along the dotted lines using the centers of the Eulerian grid cells.

elsewhere – see Section 3.3.2 for a definition of ξ . The refinement criterion ζ can be different for different fields of interest. In this study, we choose ζ to be very small when applied to c_s and ξ in order to impose the maximum level of refinement in the vicinity of walls and of the membrane (typically we choose $\zeta < 10^{-10}$); while we found by trial and error that having ζ of the order of 1% of the characteristic velocity when applied to u_x , u_y and u_z leads to satisfactory refinement and coarsening of computational cells in the rest of the fluid domain.

3.2. Second-order treatment of solid embedded boundaries

In Basilisk, complex geometries are handled using the sharp, second-order and conservative embedded boundaries method of Johansen and Colella [60]. In this method, the solid boundaries are assumed to cut the Eulerian cells in a piecewise linear fashion. Boundary conditions are enforced by estimating the flux on the solid boundary using second-order interpolations on a stencil involving the surrounding fluid cells. An example of such interpolation stencil in two dimensions is shown in figure 1. This method can be implemented so that only the volume and face fluid fractions are necessary to describe the boundary and recover the boundary flux [47]. As such, at the beginning of the simulation these two fluid fraction fields are generated from either a user-defined level-set function describing the geometry or an STL file.

If no additional treatment is done, it is well known that this class of cut-cell methods suffers from unreasonably strict CFL restrictions due to cells with

small fluid volume fractions. Indeed, when a cell is cut by a solid boundary the effective CFL condition becomes $\Delta t < c\Delta x/(f|\mathbf{u}|)$, where c and f are the volume and face fractions, and $|\mathbf{u}|$ is the velocity norm in the considered cell. If c/f is close to zero, the time step Δt may become arbitrarily small. To alleviate this issue, a “flux redistribution” technique is carried out, where the fluxes of problematic small cells are redistributed to their neighbors, thus preventing Δt from becoming arbitrarily small [61, 62, 63].

3.3. Front-Tracking Method (FTM)

3.3.1. Standard FTM formulation

The capsule configuration is described using the Front Tracking Method: we adopt a Lagrangian representation of the capsule, which we cover by a triangulated, unstructured mesh [64, 33]. This Lagrangian mesh communicates with the Eulerian octree grid used to describe the background fluid by means of regularized Dirac-delta functions introduced by Peskin [65], which role is to interpolate velocities from the Eulerian grid to a Lagrangian node; and to spread membrane forces from a Lagrangian node to the Eulerian grid. In this paper we use a cosine-shaped regularized delta function:

$$\delta(\mathbf{x}_0 - \mathbf{x}) = \begin{cases} \frac{1}{64\Delta^3} \prod_{i=1}^3 \cos\left(\frac{\pi}{2\Delta}(x_{0,i} - x_i)\right) & \text{if } |x_{0,i} - x_i| < 2\Delta \\ 0 & \text{otherwise} \end{cases}, \quad (12)$$

where Δ is the length of an Eulerian cell and $\mathbf{x}_0 = [x_0, y_0, z_0]$ corresponds in practice to the coordinates of a Lagrangian node. The prefactor $1/(64\Delta^3)$ ensures that the discrete integral over the whole space $\int_{\Omega} \delta(\mathbf{x}_0 - \mathbf{x}) d\mathbf{x}$ is equal to 1. Then, the velocity \mathbf{u}_0 of a given Lagrangian node located at \mathbf{x}_0 is interpolated from the Eulerian velocity field \mathbf{u} using:

$$\mathbf{u}_0 = \int_{\Omega} \mathbf{u}(\mathbf{x}) \delta(\mathbf{x}_0 - \mathbf{x}) d\mathbf{x} \iff \mathbf{u}_0 = \sum_{i \in \text{stencil}} \mathbf{u}_i \delta(\mathbf{x}_0 - \mathbf{x}_i) \Delta^3, \quad (13)$$

where “stencil” denotes the Eulerian cells which center \mathbf{x}_i is such that $\delta(\mathbf{x}_0 - \mathbf{x}_i) \neq 0$, and \mathbf{u}_i is the velocity of a given fluid cell. Similarly, the membrane force \mathbf{F}_0 at a Lagrangian node is spread to a force density field \mathbf{f} using:

$$\mathcal{M}(\text{supp}(\delta)) \mathbf{f} = \int_{\Omega} \mathbf{F}_0 \delta(\mathbf{x}_0 - \mathbf{x}) d\mathbf{x} \iff \mathbf{f}_i = \mathbf{F}_0 \delta(\mathbf{x}_0 - \mathbf{x}_i), \quad (14)$$

where $\mathcal{M}(\text{supp}(\delta))$ is the measure of the support of the regularized Dirac-delta function, and is in practice equal to $64\Delta^3$ in three dimensions for the regularized Dirac-delta function we choose in Eq. (12).

Once the Lagrangian velocities of all the capsule nodes have been interpolated from the Eulerian velocity field using Eq. (13), the position of each node is updated using a second-order Runge-Kutta time integration scheme. The

membrane stresses are then computed from the new configuration of the capsule using the methods described later in Section 3.4, and transferred to the background fluid using Eq. (14). In the context of particle-laden flows, the use of the IBM requires sub-time stepping for the particle advection due to the orders of magnitude difference in the fluid time scale and the solid-solid interactions time scale [39]. This is not necessary for capsule-laden flows described by the FTM, i.e. the time step for the advection of Lagrangian nodes is equal to that of the fluid solver. It follows that the trajectories of each Lagrangian node coincide with the streamlines of the flow and that the triangulations of two interacting capsules can never overlap – provided that the time step is sufficiently small. As such, there is no need for any ad-hoc repulsive force between two approaching capsules or between a capsule approaching a wall, as was the case in e.g. [66]: the non-penetration condition is seamlessly handled by the local flow field. In the latter case of close interaction between a capsule and a wall, however, the definition of the IBM stencil must be altered, as described in the next subsection.

Additionally, the case of capsules of inner viscosity μ_i different from the viscosity of the surrounding fluid μ_e needs special treatment. We adopt the approach developed in the original FTM by Unverdi & Tryggvason [64]: a discrete indicator function I is computed from a discrete “grid-gradient” field $\mathbf{G}(\mathbf{x})$:

$$\mathbf{G}(\mathbf{x}) = \sum_{i \in \mathcal{T}} S_i \delta(\mathbf{x} - \mathbf{x}_i) \mathbf{n}_i, \quad (15)$$

where \mathcal{T} denotes the set of all triangles of the discretization of the surface of all the capsules, S_i is the surface area of triangle i , \mathbf{x}_i is the position vector of its centroid and \mathbf{n}_i is its unit *inward* normal vector. In practice, $\mathbf{G}(\mathbf{x})$ is computed by looping over all triangles of the discretizations of all capsules and spreading the quantity $S_i \mathbf{n}_i$ using the regularized Dirac-delta functions introduced previously. As such, $\mathbf{G}(\mathbf{x})$ is non-zero in the union of all the IBM stencils. The discrete indicator function $I(\mathbf{x})$ is computed by solving the following Poisson problem:

$$\Delta I = \nabla \cdot \mathbf{G}. \quad (16)$$

Since I is a regularized step function, it should have constant values away from the capsule membranes. To guarantee this property, we only update I in the cells where \mathbf{G} is non-zero and we re-initialize I to 0 or 1 elsewhere, as suggested in [33].

3.3.2. Adaptive FTM strategy

Our current implementation of the FTM requires all cells in an IBM stencil to be the same size Δ . As a result, all the Eulerian cells around the membrane must be the same size as well, although future studies may lift this restriction. In practice, since the flow physics is happening in the vicinity of the capsule, we need the cell sizes around the membrane to be the smallest grid size of the fluid domain. To enforce this condition, we create a scalar field ξ initialized at each

time step to be: (i) 0 if the Eulerian cell does not belong to any IBM stencil; or (ii) a randomly generated value between 0 and 1 otherwise. In other words, the scalar field ξ tags the IBM stencils with noise while the rest of the domain is set to a constant value. Feeding this scalar field to Basilisk’s wavelet adaptation algorithm ensures that all the stencil cells are defined at the finest level, and that no IBM stencil contains Eulerian cells of different cell levels.

3.4. Computation of the membrane forces

3.4.1. Computation of the elastic force with the Finite Element Method

In order to compute the nodal elastic forces given by Eq. (8), we employ a Finite Element Method (FEM). In most FEM solvers from Engineering applications, the sought quantity is the displacement of a structure under a known applied stress. In the case of biological membranes, we rather seek the internal stress of the membrane under a known displacement [55]. Charrier et al. [52] have been the first to design this specific FEM framework: we base our implementation on their work as well as that of Doddi & Bagchi [19].

Consider an arbitrary triangle T_i on the discretized membrane: in order to compute the elastic force of its three vertices, we first rotate it to a common plane – e.g. the x, y -plane – using a rotation matrix \mathbf{R}_i from the current orientation of the triangle to its orientation in the common plane. Then, we assume the position of the triangle vertices in a stress-free configuration is known in the common plane, and we compute the displacements \mathbf{v}_k of each of the three vertices of T_i . Using linear shape functions, the deformation gradient tensor and the Cauchy-Green deformation gradient tensor attached to T_i can be computed:

$$\mathbf{F} = \frac{\partial \mathbf{v}_k}{\partial \mathbf{x}^p}, \quad \mathbf{C} = \mathbf{F}^T \mathbf{F}, \quad (17)$$

where $(\mathbf{x}^p, \mathbf{y}^p)$ is the basis of the common plane. Note that \mathbf{F} and \mathbf{C} are two-dimensional tensors and correspond to the tangential components of \mathbf{F}_s and \mathbf{C}_s in Eq. (3–4). By diagonalizing \mathbf{C} and taking the square root of its eigenvalues (\mathbf{C} is symmetric positive definite), we can access the two principal stretch ratios λ_1, λ_2 attached to T_i . Following Charrier et al. [52], the principle of virtual work yields the expression linking the nodal force and nodal displacement at node j :

$$\mathbf{F}_{\text{elastic},j}^P = A_i \frac{\partial W}{\partial \lambda_1} \frac{\partial \lambda_1}{\partial \mathbf{v}_j} + A_i \frac{\partial W}{\partial \lambda_2} \frac{\partial \lambda_2}{\partial \mathbf{v}_j}, \quad (18)$$

where A_i is the area of T_i . Rotating Eq. (18) back to the current reference frame of T_i , we get the final expression of the contribution of triangle T_i to the elastic force of node j :

$$\begin{aligned} \mathbf{F}_{\text{elastic},j} &= \mathbf{R}^T \mathbf{F}_{\text{elastic},j}^P \\ &= A_i \mathbf{R}^T \left(\frac{\partial W}{\partial \lambda_1} \frac{\partial \lambda_1}{\partial \mathbf{v}_j} + \frac{\partial W}{\partial \lambda_2} \frac{\partial \lambda_2}{\partial \mathbf{v}_j} \right). \end{aligned} \quad (19)$$

This FEM implementation is summarized in algorithm 1.

Algorithm 1 Pseudocode for the Finite Element Method

```

loop over all triangles  $i$ 
  loop over the three nodes  $j$  of  $T_i$ 
    Compute  $\mathbf{x}_j^P = \mathbf{R}\mathbf{x}_j$ 
    Compute the nodal displacement  $\mathbf{v}_j = \mathbf{x}_j^P - \mathbf{x}_{j,t=0}^P$ 
  end loop
  Compute  $\mathbf{F}$ ,  $\mathbf{C}$  from Eq. (17)
  Compute the eigenvalues of  $\mathbf{C}$  and  $\mathbf{F}$ , i.e.  $\lambda_1^2$ ,  $\lambda_2^2$ ,  $\lambda_1$ ,  $\lambda_2$ 
  loop over the three nodes  $j$  of  $T_i$ 
    Compute  $\partial\lambda_1/\partial\mathbf{v}_j$ ,  $\partial\lambda_2/\partial\mathbf{v}_j$ 
    Compute  $\mathbf{F}_{\text{elastic},j}^P$  from Eq. (18)
    Rotate  $\mathbf{F}_{\text{elastic},j}^P$  to the current orientation of  $T_i$ 
    Add  $\mathbf{F}_{\text{elastic},j}$  to the total elastic force of node  $j$ 
  end loop
end loop

```

3.4.2. Computation of the bending force using paraboloid fits

The computation of $\mathbf{F}_{\text{bending}}$ relies on the local evaluation of: (i) the mean and Gaussian curvatures κ and κ_g , (ii) the Laplace-Beltrami operator of the mean curvature $\Delta_s\kappa$, and (iii) a relevant control area A .

To evaluate κ and κ_g at node i , we blend the approaches of Farutin et al. [67] and Yazdani & Bagchi [21]. A local reference frame is attached to node i , with the z -direction coinciding with the approximate normal vector \mathbf{n}_i . Then, a paraboloid is fitted to node i and its one-ring neighbors. In our triangulated surface, most nodes have six neighbors¹, making the system overdetermined and a least-squares method is used. From this paraboloid fitting, we can derive the local mean and Gaussian curvatures – see equations (12) and (13) in [21] –, as well as a refined approximation of \mathbf{n}_i . This procedure is iterated using the newest normal vector approximation to define the local frame of reference, until satisfactory convergence of \mathbf{n}_i is reached. Our numerical experimentations show that between three to five iterations usually suffice to obtain a converged normal vector.

The same paraboloid fitting method is used to compute $\Delta_s\kappa$, or $\Delta_s(\kappa - \kappa_0)$ in the case of non-zero reference curvature. This time, a paraboloid is fitted to the *curvatures* of node i and its neighbors, and then differentiated to obtain the desired surface Laplacian.

The last term A is necessary to obtain a bending force as opposed to a bending force per surface area. Let A_i denote the nodal area attached to node i : at any time the sum of all nodal areas need to equal the total area of the

¹Exactly twelve nodes have five neighbors, since we discretize a spherical membrane by subdividing each triangle of an icosahedron. Each newly created node is projected back to a sphere, and if necessary projected onto a more complex shape, e.g. a biconcave membrane.

capsule, i.e. $\sum_{i=0}^N A_i = A_{tot}$ with N the number of Lagrangian nodes and A_{tot} the total area of the discretized surface of the capsule. The Voronoi area of node i enforces this property only for non-obtuse triangles. As such, we adopt the “mixed-area” of Meyer et al. [56, 68] which treats the special case of obtuse triangles separately: if a triangle j is not obtuse, its contribution to the nodal area of its vertices is the standard Voronoi area; while if j is an obtuse triangle, the nodal area of its obtuse vertex is $A_j/2$ while the nodal area of the remaining two vertices is $A_j/4$, where A_j is the area of triangle j .

4. Validation Cases

4.1. Elastic and bending forces of an isolated membrane

4.1.1. Elongation of a flat elastic membrane

Our first validation case focuses on the computation of the elastic stress in the membrane. To this end, we stretch an isolated flat membrane devoid of bending resistance in one of its principal direction \mathbf{e}_1 while ensuring the principal stress T_2 in the second principal direction \mathbf{e}_2 remains zero. We then analyze the non-zero principal stress $T_1 = (\partial W/\partial \lambda_1)/\lambda_2$ as a function of the principal stretch $e_1 = (\lambda_1^2 - 1)/2$. This test is repeated for two membranes: the former obeying the neo-Hookean law and the latter obeying the Skalak law. Note that in order to set the principal stress T_2 equal to zero, we impose λ_2 to a value strictly lower than 1, i.e. the membrane is shrunk in the second principal direction:

$$\lambda_2 = \begin{cases} 1/\sqrt{\lambda_1} & \text{for the neo-Hookean law} \\ \sqrt{(1 + C\lambda_1^2)/(1 + C\lambda_1^4)} & \text{for the Skalak law} \end{cases} \quad (20)$$

We compare our results to the exact stress derived by Barthès-Biesel et al. [51] in figure 2, with $E_s = C = 1$. The data we generate overlaps perfectly with the analytical stress-strain relations, thus validating the implementation of our Finite-Element solver for the elastic membrane stresses. The source code to reproduce this validation case is available online [69].

4.1.2. Bending force of a curved membrane

In order to validate our bending force, we follow the procedure of Guckenberger et al. [56]: considering a biconcave membrane with zero reference curvature, we compare the mean and Gaussian curvatures, Laplace-Beltrami operator of the mean curvature, and total nodal bending force density to analytical expressions derived using a symbolic calculus software. Since the biconcave capsule has a rotational symmetry around the z -axis and a symmetry with respect to the (x, y) -plane, we plot our results according to the angle θ defined in figure 3, with θ varying from 0 to $\pi/2$. This biconcave shape is a good candidate to test the bending force since its two principal curvatures are in general not equal to each other and are varying along the surface of the biconcave shape – even changing sign. The following results are obtained with a biconcave membrane discretized by a triangulation containing 5120 triangular elements.

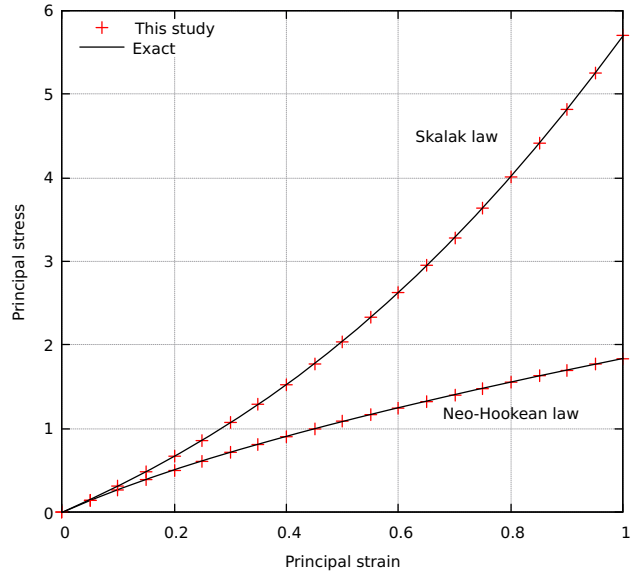


Figure 2: Stress-strain response of an isolated flat membrane for the neo-Hookean and Skalak elastic laws. The results from this study are compared to exact expressions derived in [51].

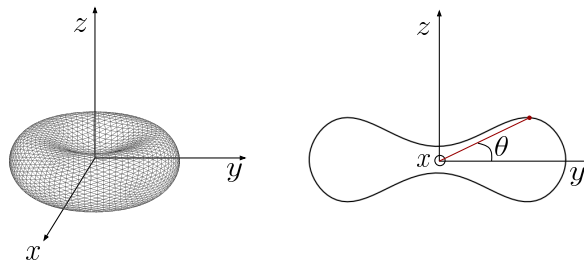


Figure 3: Schematic of a biconcave capsule centered at the origin, and definition of the polar angle θ .

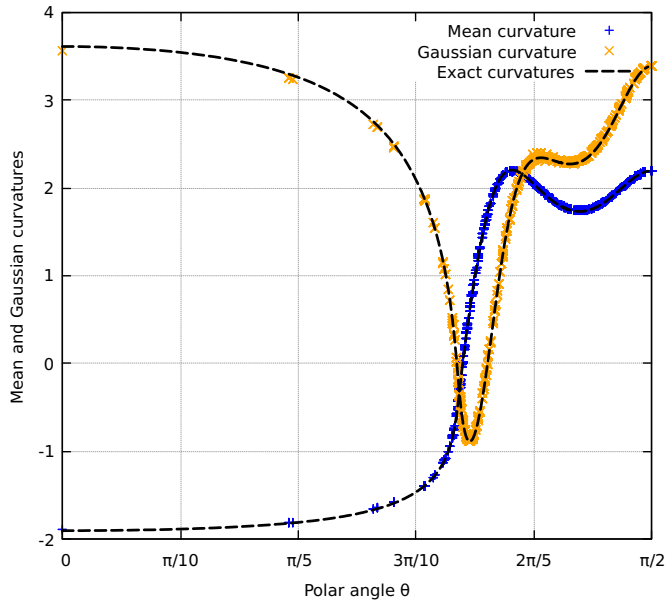
We compare our computed mean and Gaussian curvatures to their respective analytical expressions in figure 4a: the agreement is very satisfactory for both curvatures. Figure 4b shows the Laplace-Beltrami operator of the curvature against its analytical expression: the general trend still matches that of the analytical expression very well, but a few outliers deviate from it by a few percents. The same behavior is observed in figure 4c which shows the nodal bending force density. The fact that the behavior of figure 4b and figure 4c is similar is not surprising, as the nodal bending force density plotted in figure 4c directly involves the Laplace-Beltrami operator of the mean curvature shown in figure 4b. It is expected to see some small deviations to the theory when taking the Laplace-Beltrami operator of the mean curvature, as we are essentially taking a fourth-order derivative of the geometry of the membrane, and Guckenberger et al. [56] observe a similar noise when performing the same tests (see figures 6c and 8e in [56]). In fact, they show that most other methods perform much worse at computing the Laplace-Beltrami operator of the mean curvature, and hence at computing the total bending force. As such, our implementation of the bending force shows the expected performance. The code to reproduce this test case is available at [70].

4.2. Initially spherical capsule in an unbounded shear flow

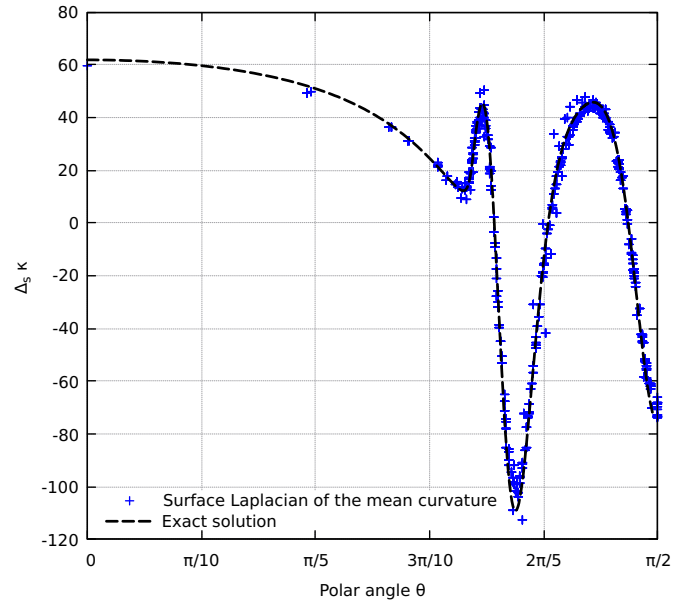
4.2.1. Neo-Hookean elasticity without bending resistance

We now seek validation of the coupling between the membrane solver and the fluid solver. To this end, we consider an initially spherical capsule of radius a in an unbounded shear flow. The elasticity is governed by the neo-Hookean law, and the flow field is initialized to be that of an undisturbed shear flow. As the capsule deforms, we plot the Taylor deformation parameter $D = (a_{max} - a_{min}) / (a_{max} + a_{min})$ as a function of the non-dimensional time $\dot{\gamma}t$, with a_{max} and a_{min} the maximum and minimum radii of the capsule at a given time, and $\dot{\gamma}$ the shear rate. We perform this simulation for various Capillary numbers $Ca = \mu a^2 \dot{\gamma} / E_s$, with E_s the elastic modulus. In this test case, Stokes conditions are intended so we set the Reynolds number to 0.01. At time $t = 0$, the flow field is set to that of a fully developed shear flow: $u_x = \dot{\gamma}y$. The computational box is bi-periodic in the x and z directions, while Dirichlet boundary conditions for the velocity are imposed in the y direction. The length of the computational box is equal to 8 initial radii, the size of the most refined Eulerian cells is set to $1/128$ that of the domain length, and the membrane is discretized by 1280 triangles. The non-dimensional time step $\dot{\gamma}\Delta t$ is set to 10^{-3} , except for the Capillary numbers $Ca = 0.025$ and $Ca = 0.0125$ where the time step is decreased to $\dot{\gamma}\Delta t = 10^{-4}$ to stabilize the elastic force computation.

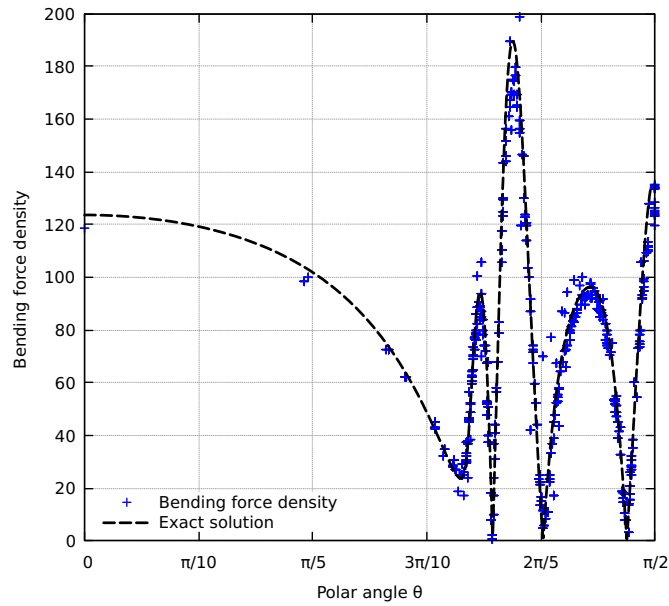
This case has been widely studied in the literature: in figure 5 we compare our results to those of [10, 11] who used the BIM, as well as [19] who used the FTM. The agreement is very satisfactory: the steady-state value we obtain for D is well within the range of the reported data, both in the transient regime and once a steady-state is reached. We also show in figure 5 the results for a finer triangulation of the membrane with 5120 triangles, and a refined Eulerian



(a)



(b)



(c)

Figure 4: Comparison of the computed mean and Gaussian curvatures (top left), Laplace-Beltrami operator of the mean curvature (top right) and nodal bending force density (bottom) to their analytical expressions. All quantities are plotted against the polar angle θ defined in figure 3.

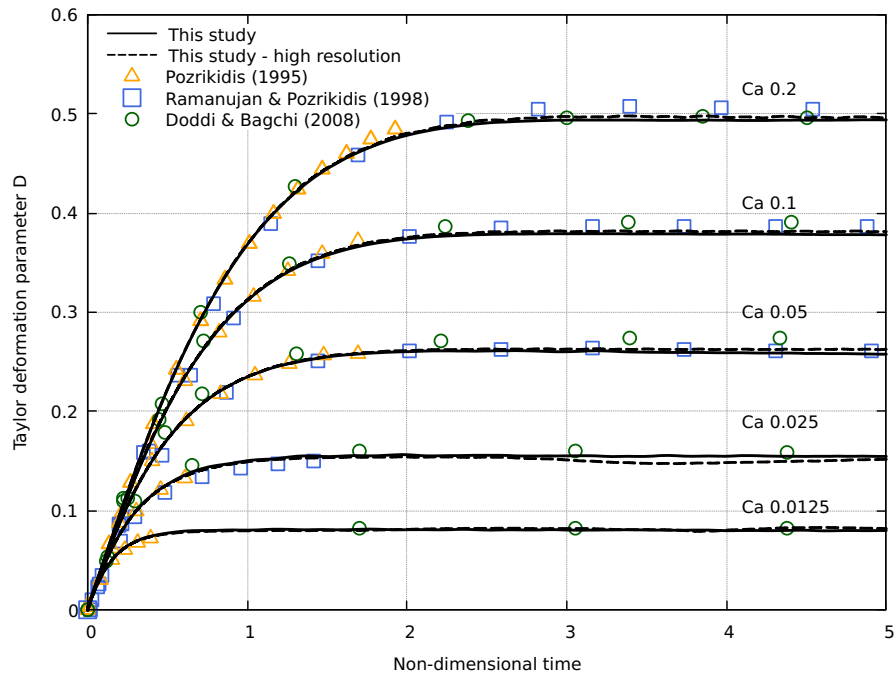


Figure 5: Taylor deformation parameter as a function of the non-dimensional time of an initially spherical Neo-Hookean capsule in an unbounded shear flow. The solid line corresponds to 32 Eulerian cells per initial diameter and a Lagrangian discretization using 1280 triangles, while the dashed line corresponds to 64 Eulerian cells per initial diameter and a Lagrangian discretization using 5120 triangles.

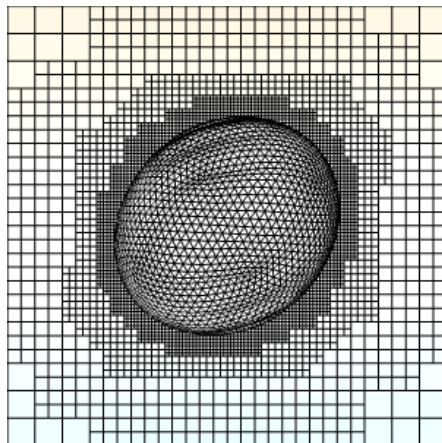


Figure 6: A zoomed in snapshot of a buckling membrane at $Ca = 0.0125$ with a Lagrangian discretization comprising 5120 triangles. This behavior is arising due to the absence of bending stresses, and the buckling wavelength is dependant on the Lagrangian discretization. The color field represents the x -component of the velocity.

mesh with the finest Eulerian cell size corresponding to $1/256$ that of the domain length. The only difference is that the steady state is longer to reach for $Ca = 0.025$ and $Ca = 0.0125$, due to the apparition of buckling instabilities on the membrane as a result of the absence of bending stresses. This buckling instability has been observed both experimentally [71] and numerically [72, 22], although in numerical simulations the wavelength is unphysical and determined by the size of the mesh discretizing the capsule [55]. In our simulations, we do observe the same dependance of the wavelength of the membrane buckles on the Lagrangian mesh element size. An example of this buckling instability is shown in figure 6. The code to reproduce this test case is available at [73].

4.2.2. Including bending resistance

We further validate our solver by considering the similar case of a capsule deforming in a shear flow, this time with the addition of a bending force. As in the previous case, an initially spherical, unstressed capsule is placed in a shear flow where the initial velocity field is fully developed. The Capillary number is $Ca = 0.05$, and the non-dimensional bending coefficient $\tilde{E}_b = E_s/(a^2 E_b)$ is chosen equal to 0.0375. The membrane is discretized with 5120 triangles and the same Eulerian resolution as in the previous case is chosen. Due to the stiffness of the bending force, we set the time step to $\Delta t = 10^{-4}$. The Taylor deformation parameter is compared to that of various studies in the literature in figure 7. The capsule deforms under the action of the flow field and the Taylor deformation parameter quickly attains a steady state of about $D = 0.15$. We remark that the data reported in the literature is scattered by about 20% which underlines the challenges to simulate Helfrich's bending force, as was previously noted by [56]. We also note that our results are situated well within the range

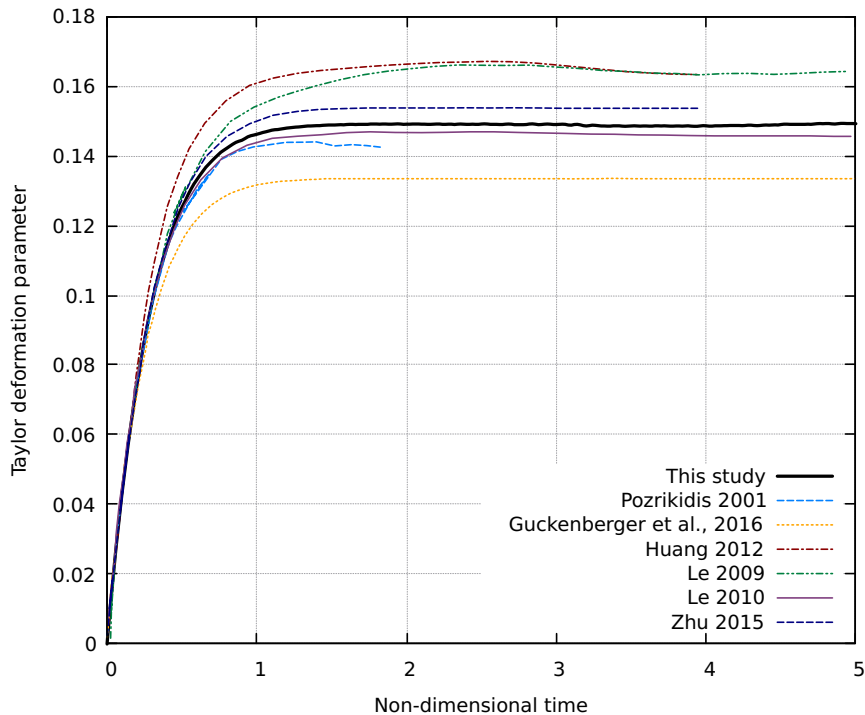


Figure 7: Taylor deformation parameter of an isolated capsule undergoing elastic and bending stresses in a shear flow. The capillary number is $Ca = 0.05$ and the non-dimensional bending coefficient is $\tilde{E}_b = 0.0375$.

of the reported data: we are close to the results of Zhu & Brandt [74] and Le et al. [75], and our curve is located in the middle of the reported range that we borrowed from [56, 54]. Given such a wide range of reported literature data, it is difficult to conduct a rigorous quantitative analysis. Nevertheless, we conclude from figure 7 that our bending force shows a similar behavior as that of other studies, a claim also supported by the validation case in Section 4.4. The code to reproduce this test case is available at [76]

4.3. Initially spherical capsule flowing through a constricted channel

To validate our implementation for an elastic capsule in the presence of complex boundaries, we consider the case of a capsule flowing through a constricted square channel proposed by Park & Dimitrakopoulos [77]. The elasticity of the membrane is governed by the Skalak law with the area dilatation modulus C set to 1, the capsule is initially pre-inflated such that its circumference is increased by 5%, and the flow is driven by an imposed uniform velocity field at the inlet and outlet boundaries. We follow [77] and choose the Capillary number to be 0.1, and since Stokes conditions are intended we set $Re = 0.01$.

The results are presented in figure 8 and figure 9. The qualitative agreement in figure 8 is very satisfactory as the capsule shape is visually identical to that of [77]. We draw the reader’s attention to the adaptive Eulerian mesh on the right-hand side of figure 8: the cells size is imposed to be minimal at the solid boundaries and around the capsule, while everywhere else the adaptivity criterion is governed by the velocity gradients. As a result, the grid cells away from the membrane and from the walls quickly coarsen to up to three levels lower, except in the vicinity of the corners where stronger velocity gradients occur. Figure 9 shows the non-dimensional lengths of the capsule in the x -, y - and z -directions with respect to the non-dimensional position of its center x_c/H_c , with x_c and H_c the x -position of the center of the capsule and the half-height of the constriction, respectively. As found by Park & Dimitrakopoulos, the final shape of the capsule is not exactly spherical as it remains shrunked in the x -direction downstream of the constriction. Despite some small deviations during the extreme deformation of the capsule, around $x_c/H_c = -1$, the overall quantitative agreement of the transient shape of the capsule is also satisfactory, especially considering that other authors have reproduced this case with similar or larger deviations from the results reported by Park & Dimitrakopoulos [24, 38]. The code to reproduce this test case is available at [78].

4.4. Red blood cell in an unbounded shear flow

The next test case aims at validating the membrane solver when a viscosity ratio $\lambda_\mu = \mu_i/\mu_e$ is different than 1. To this end, we consider an RBC in an unbounded shear flow, with $\lambda_\mu = 5$. The membrane forces include the Skalak elastic law and the Helfrich’s bending force. The Capillary number is $Ca = 0.1$, the area dilatation modulus C is chosen equal to 50 and the non-dimensional bending coefficient is $\tilde{E}_b = 0.01$. The reference curvature is $c_0 a = -2.09$ [79, 20], where $a = (3V/4\pi)^{1/3}$ is the radius of the sphere of equal volume as that of the RBC. The initial shape of the RBC is biconcave and is described by the following equations, for an RBC which largest radius is orthogonal to the y direction [80]:

$$\begin{cases} x = ac \cos \phi \sin \psi \\ y = \frac{ac}{2} \sin \phi (\alpha_1 + \alpha_2 \cos^2 \phi + \alpha_3 \cos^4 \phi) \\ z = ac \cos \phi \sin \psi, \end{cases} \quad \text{with } \phi \in [0, 2\pi], \psi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \quad (21)$$

with $\alpha_1 = 0.207$, $\alpha_2 = 2.003$ and $\alpha_3 = -1.123$. Since we consider a viscosity ratio, we also define the initial indicator function I as the volume fraction of inner fluid:

$$I(\mathbf{x}) = \begin{cases} 1 & \text{if } \Phi(\mathbf{x}) < 0 \\ 0 & \text{if } \Phi(\mathbf{x}) > 0 \\ \text{between 0 and 1} & \text{otherwise,} \end{cases} \quad (22)$$

where Φ is the level-set alternative formulation of Eq. (21):

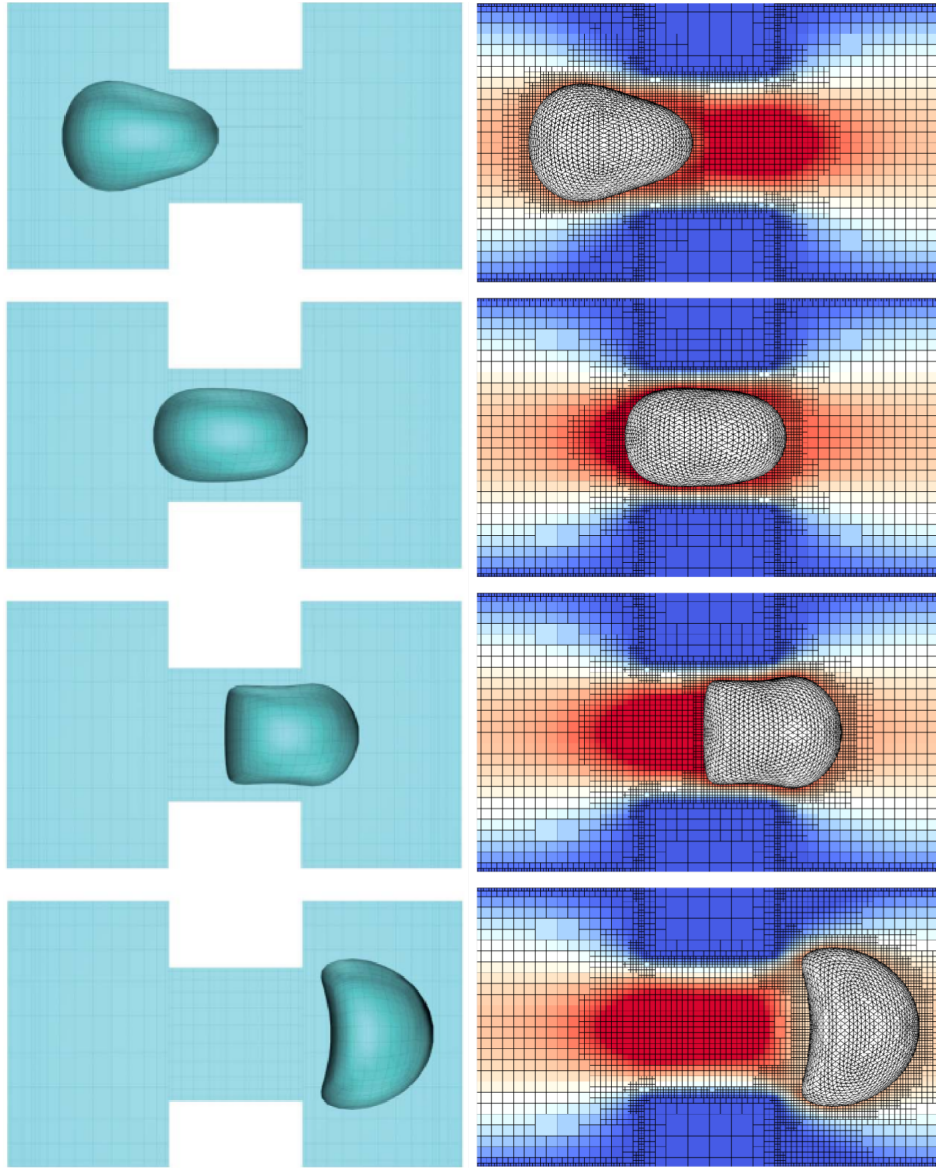


Figure 8: Snapshots of the capsule as it flows through the constriction. Left: Park & Dimitrakopoulos [77]; Right: this study.

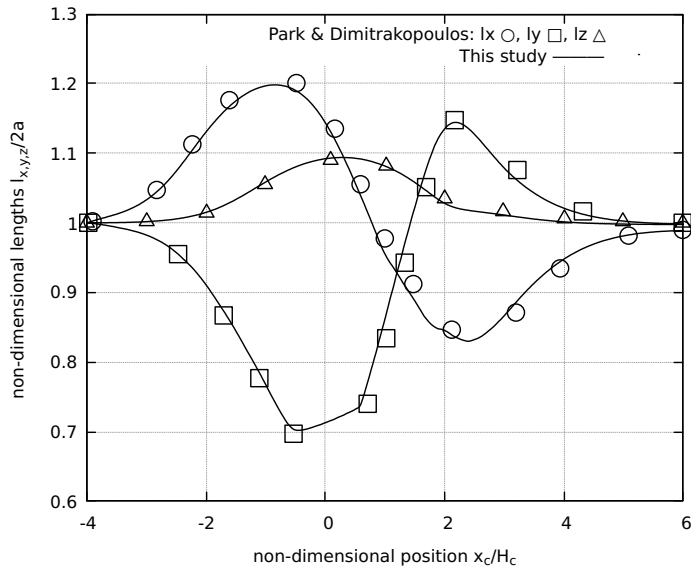


Figure 9: Non-dimensional lengths of the capsules in the three directions x , y and z with respect to the non-dimensional x -position of the center of the capsule. Results are compared to [77].

$$\Phi(x, y, z) = \frac{x^2 + z^2}{(ac)^2} + \frac{4y^2}{(ac)^2} \left(\alpha_1 + \alpha_2 \frac{x^2 + z^2}{(ac)^2} + \alpha_3 \left(\frac{x^2 + z^2}{(ac)^2} \right)^2 \right). \quad (23)$$

The initial fluid velocity is set to that of an unbounded shear flow of shear rate $\dot{\gamma}$ with the velocity gradient in the direction of the greater axis of the RBC. The dimensionless time step we use is $\dot{\gamma}\Delta t = 10^{-4}$ and is determined from trial and error.

The qualitative results are presented in figure 10, where we include snapshots of the same case from Yazdani & Bagchi [20]. We observe that the RBC is undergoing a tumbling motion, a behavior of RBCs that is not seen without viscosity ratio in this range of Capillary numbers [20, 80]. Moreover, the deformation of the RBC matches qualitatively well that of [20]. However our tumbling period seems slightly shorter than that of [20]: we attribute this small discrepancy to the fact that we may have set different values for the area dilatation modulus C , as [20] only provides a range of values: $C \in [50, 400]$. Nevertheless, the results in figure 10 show that in our implementation, the combination of elastic forces, bending forces and viscosity ratio matches well the qualitative results observed in the literature, and that the overall agreement is satisfactory. The code to reproduce this case is available at [81].

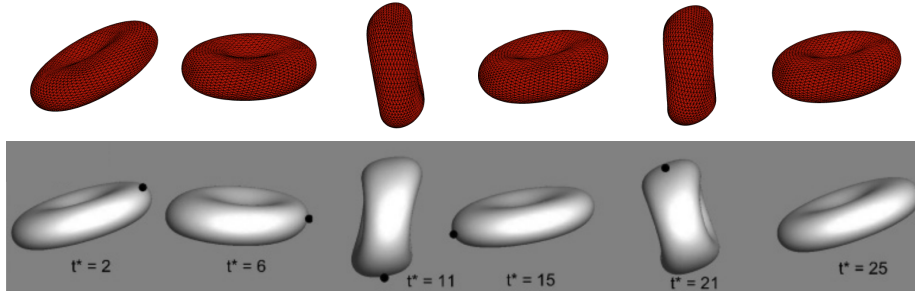


Figure 10: Tumbling motion of an isolated RBC in a shear flow. Top: this study; bottom: Yazdani & Bagchi [20].

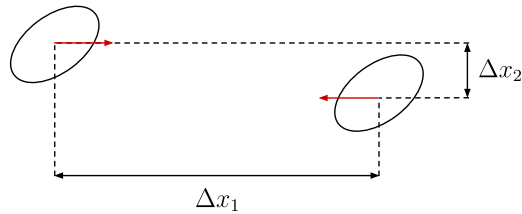


Figure 11: Schematic of the two capsules in the shear flow, prior to the interception. The horizontal and vertical gaps Δx_1 and Δx_2 are defined, and the red arrows represent the velocities of the centers of the capsules.

4.5. Capsules interception in a shear flow

Our last validation case focuses on the interactions of two capsules. Two initially spherical, pre-inflated neo-Hookean capsules are placed in an unbounded shear flow with their initial positions offset in the horizontal and vertical directions as shown in figure 11. Since the capsules are offset in the vertical direction, they gain horizontal velocities of opposite signs and their trajectories eventually intercept. This configuration is a good validation candidate since we can compare our results to those obtained by Lac et al. using the boundary integral method [17]. We consider a computational box of size $16a$ where a is the initial radius of the capsules. The finest Eulerian resolution corresponds to the domain being discretized by 512 cells in each direction, and the two membranes are discretized with 5120 triangles. A non-dimensional time step of $\dot{\gamma}\Delta t = 2.5 \cdot 10^{-4}$ is chosen. The Reynolds number is set to 0.01.

Figure 12 shows the qualitative comparison of the shape of the two capsules at several stages of the interception. In our simulations, the color field corresponds to the vertical component of the velocity. At each stage, there is visually no difference in the shape of the capsules. If we track the center of each capsule throughout the simulation, we can compute their difference Δx_2 in the vertical direction and their difference in the horizontal direction Δx_1 . Normalizing by the initial diameter $2a$ of the capsules, we plot in figure 13 the vertical gap between the two capsules as their intercept, and we compare our results to those of

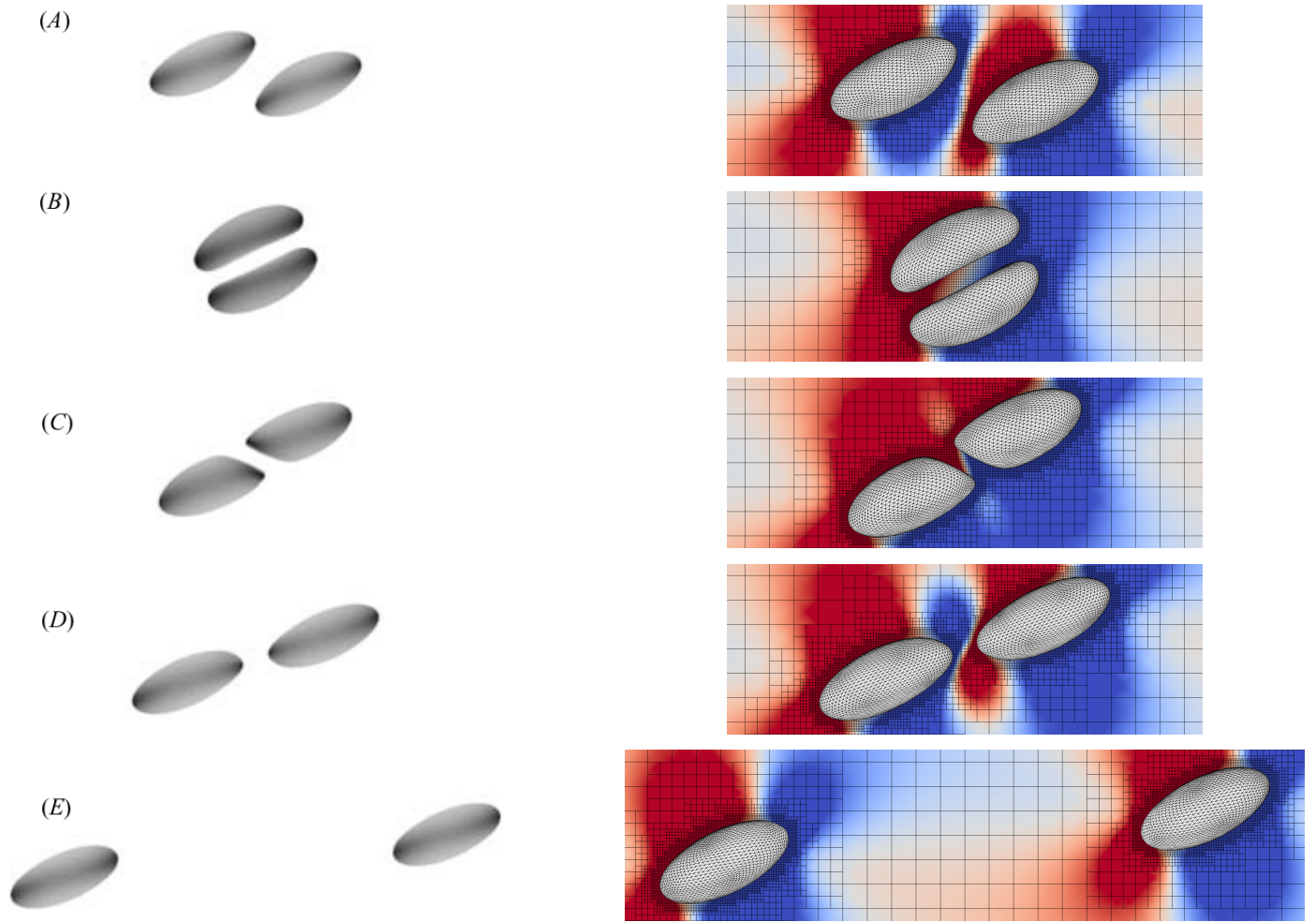


Figure 12: Snapshots of the interception of two neo-Hookean capsules in a shear flow. Left: Boundary Integral results of Lac et al. [17]. Right: This study. The color field corresponds to the vertical component of the velocity (rescaled for each snapshot).

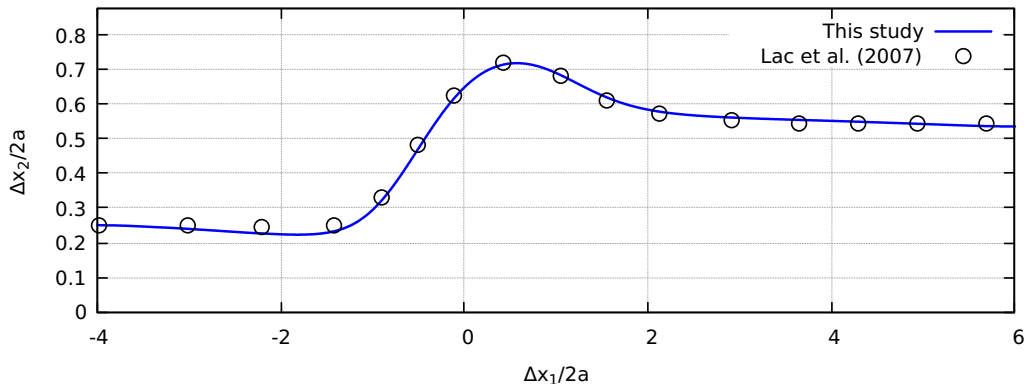


Figure 13: Non-dimensional vertical gap $\Delta x_2/2a$ against the non-dimension horizontal gap $\Delta x_1/2a$ between the centers of the two capsules. The results from this study are compared to Lac et al. [17].

Lac et al. [17]. The agreement is very satisfactory: the transient regime is very well captured, both methods showing a maximum non-dimensional vertical gap of about 0.72; and the steady-state reached is about 0.54. Small discrepancies can be observed around $\Delta x_1/2a = -2$ where our vertical gap is slightly lower than that of Lac et al.; and for $\Delta x_1/2a$ between 4 and 6 where our slope is still slightly negative while that of Lac et al. is essentially zero. Those discrepancies are minor and could be explained by our choice of Reynolds number $Re = 10^{-2}$ while the boundary integral method operates in true Stokes conditions. Regarding the adaptive mesh, as stated above we perform this simulation using an equivalent fluid resolution of 64 cells per initial diameter, in a cubic box 8 diameters in length. Our simulation requires about $4.5 \cdot 10^5$ fluid cells, while using a constant mesh size would require about $1.3 \cdot 10^8$ cells. For this specific case, using an adaptive grid therefore reduces the number of fluid cells by a factor of about 300.

A similar configuration was later examined by Doddi & Bagchi [23] in the presence of inertia. In a cubic box of size $H = 4a\pi$, periodic in the x and z directions, for a Capillary number of 0.05 and an initial vertical distance $\Delta x_2/2a = 0.2$, they observed that the capsules don't intercept when the Reynolds number $Re = \rho\dot{\gamma}a^2/\mu$ is greater or equal to 3. Instead, the vertical component of the center of the capsules changes sign and the direction of movement is reversed. We reproduce these results from Doddi & Bagchi (figure 8c in [23]): each capsule is discretized with 5120 triangles, the Eulerian equivalent resolution is 40 points per initial diameter, and the non-dimensional time step is $\Delta t = 10^{-3}$. Figure 14 shows the vertical position of the centers of the two capsules with respect to time, for Reynolds numbers of 3, 10 and 50, and the generated data is compared to [23]. Our results superimpose very well with [23], in particular for $Re = 3$ and $Re = 10$. For $Re = 50$, the agreement is still very satisfactory although we notice that we predict a slightly larger overshoot

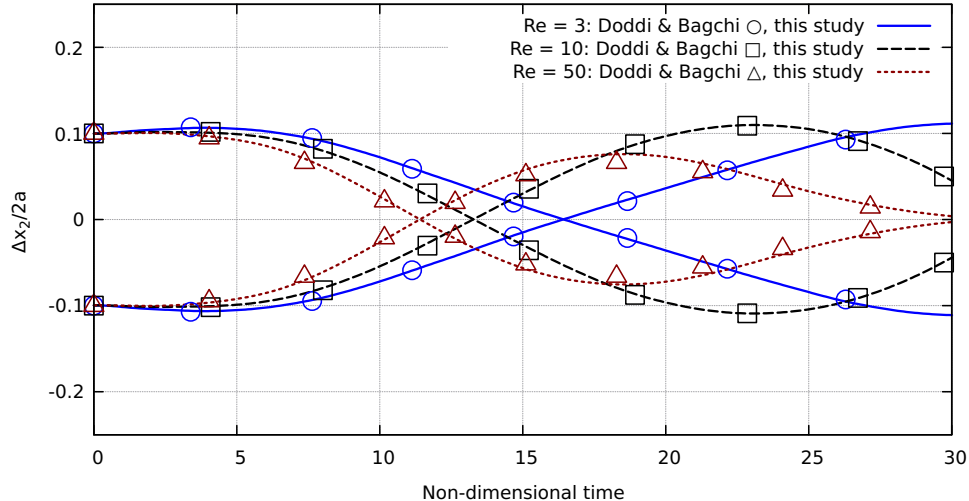


Figure 14: Non-dimensional vertical gap $\Delta x_2/2a$ against the non-dimensional time $\hat{\gamma}t$ for Reynolds number of 3, 10 and 50. The results of this study are compared to Doddi & Bagchi [23].

around $t = 18$ compared to the results of [23].

Overall, the quantitative agreement in figure 13 and figure 14 is very satisfactory and it validates our adaptive front-tracking solver for several capsules, both in Stokes conditions and in the presence of inertia. The code for these two cases is available at [82, 83].

5. Results

5.1. Capsule flowing through a narrow constriction

In their original study, Park & Dimitrakopoulos [77] investigated the case presented in Section 4.3 for relatively wide constriction sizes – the half size of the constriction H_c was greater than or equal to the capsule radius a . In this subsection we instead decrease the constriction size to $H_c = a/2$ in order to demonstrate the robustness of our solver in cases of extreme deformations, including close to domain boundaries. As in Section 4.3, the capsule is initially circular and pre-inflated such that each distance on the capsule surface is increased by 5%. It obeys the Skalak elastic law with $C = 1$ and $Ca = 0.1$. Since the capsule undergoes extreme deformations, we also consider a bending force in order to suppress unphysically sharp corners: the non-dimensional bending coefficient is set to $\bar{E}_b = 10^{-3}$. In order to resolve well the capsule deformation, we increase the resolution of the triangulation, which now comprises 20480 triangles. We also perform this simulation for two finest Eulerian grid resolutions

of 50 and 100 grid cells per initial diameter respectively. The flow is driven by an imposed uniform velocity field at the inlet and outlet boundaries, and the Reynolds number is set to 0.01 to model Stokes flow conditions. For this case the non-dimensional time step is set to $a\Delta t/U = 2.5 \cdot 10^{-5}$, with a the initial radius of the capsule and U the characteristic velocity of the fluid. It appears that this strict restriction on the time step is due to the stiff bending force combined to the very fine discretization we choose.

Qualitative results are shown in figure 15. As expected, the deformation of the capsule is considerably greater when the constriction size is halved: the capsule becomes almost flat as it reaches the center of the constriction. Figure 15 also confirms that the Eulerian mesh is refined only in the region of interest, as the grid size quickly increases with the distance from the constriction and from the capsule. We show quantitative results in figure 16, where the non-dimensional x -, y - and z -lengths of the capsule are plotted against the non-dimensional x -position of the center of the capsule. Unsurprisingly, the capsule vertical length l_y decreases by over a factor of two when the capsule reaches the center of the constriction, before sharply increasing again as the front of the capsule expands while leaving the constriction. The sharp point observed for l_y at $x_c/H_c = 0$ is simply due to the non-locality of the variable l_y : for $x_c/H_c \leq 0$ the maximum height of the capsule is located at its rear, while for $x_c/H_c \geq 0$ it is located at its front. We also note that the maximum decrease of the capsule height $l_{y,\min}^N$ in the case of a narrow constriction is much more pronounced than its counterpart $l_{y,\min}^W$ in the wider constriction. However, the minimum capsule height is not halved when the constriction size is halved, i.e. $l_{y,\min}^N > l_{y,\min}^W/2$. On the other hand, the maximum x -length of the capsule more than doubles in the case of a narrow constriction when compared to the wider constriction size, and the maximum z -length more than triples. Therefore, the capsule preference to elongate in the streamwise direction rather than the spanwise direction is reduced when the constriction is narrower. Finally, the capsule reaches a steady shape after the constriction for $x_c > 6H_c$: the constriction size does not appear to affect this steady shape, which is a slightly deformed sphere compressed in the streamwise direction.

In figure 16 the x -, y - and z -lengths of the capsule are shown for two maximum Eulerian resolutions: $D/\Delta x = 50$ and $D/\Delta x = 100$, with D the initial diameter of the capsule and Δx the smallest Eulerian cell size. Relatively minor differences are observed between the two mesh resolutions, indicating that a maximum Eulerian resolution of $D/\Delta x = 50$ is sufficient to capture the underlying physics of this configuration. In terms of performance, conducting the previous convergence study up to $D/\Delta x = 100$ with a constant grid size would have required about $4.2 \cdot 10^7$ fluid cells, while our simulation used less than $4.6 \cdot 10^6$ fluid cells, thus allowing a tenfold reduction in the number of fluid grid cells, and likely reducing the computational resources by a factor of 7 to 10 when accounting for the computational overload due to the complex tree structure of the grid and the adaptivity algorithm.

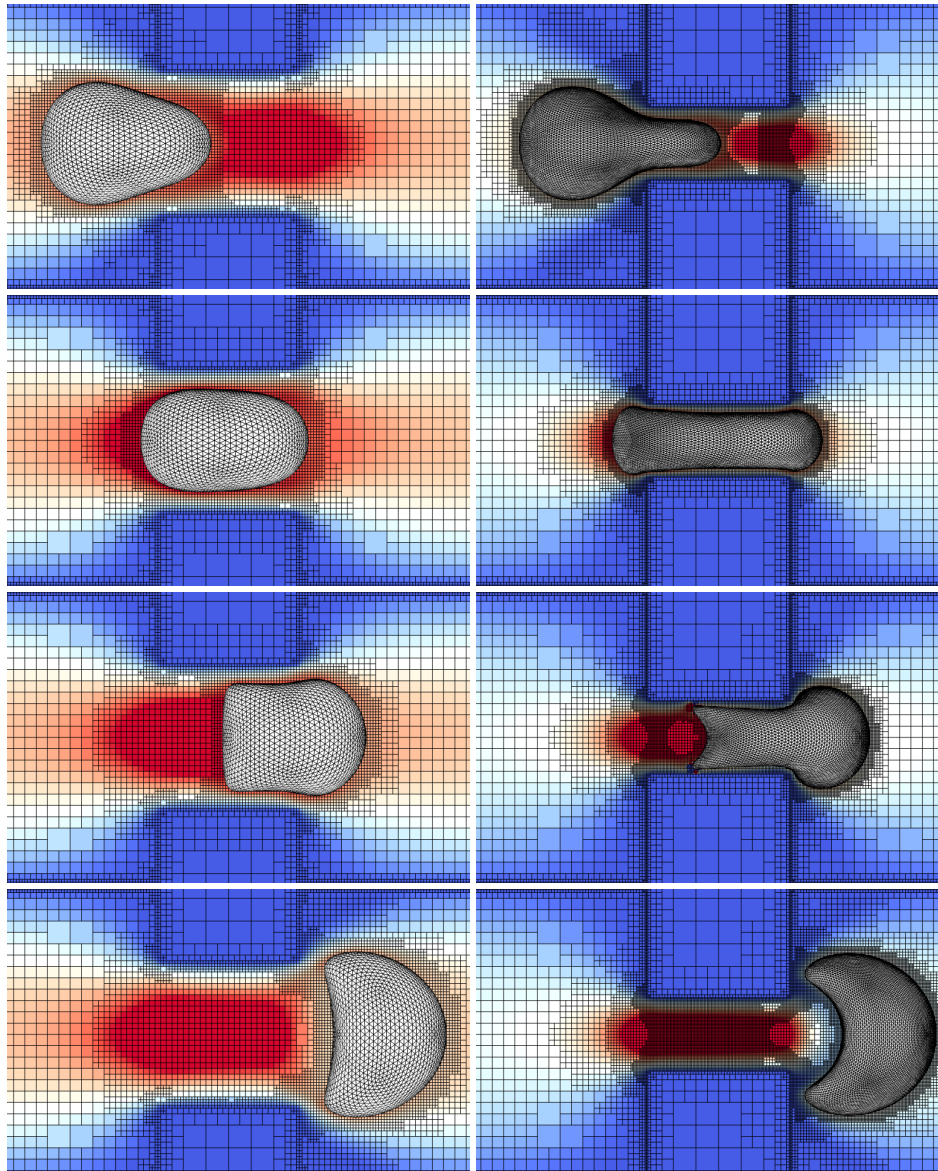


Figure 15: Qualitative comparison of a capsule flowing through a constricted channel: when the constriction size is equal to the initial diameter of the capsule (left); and when the constriction size is equal to half of the initial diameter of the capsule, with a finest Eulerian grid resolution of 100 cells per initial diameter (right). Color field: x -component of the velocity field.

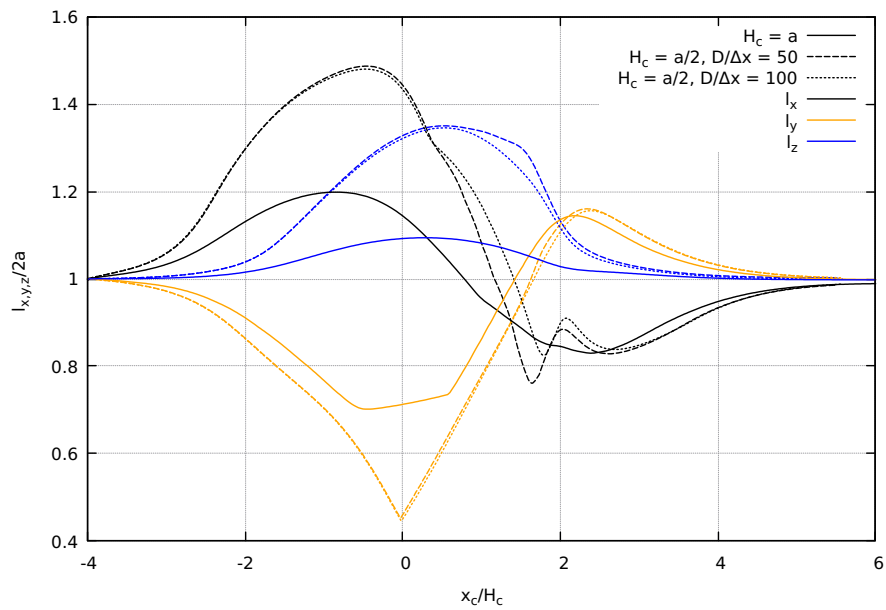


Figure 16: Non-dimensional x -, y - and z -lengths of the capsule $l_x/2a$, $l_y/2a$ and $l_z/2a$ with respect to the non-dimensional x -position of the center of the capsule x_c/H_c as it flows through a constriction size equal to: (i) the initial diameter of the capsule (solid line); and (ii) half of the initial diameter of the capsule (dashed line). The dotted line corresponds to a finest Eulerian resolution of 100 Eulerian grid cells per initial diameter (shown for $x_c/H_c = a/2$ only).

5.2. Capsule-laden flows in large complex channel geometries

It is clear from the previous simulation results presented in this paper that the adaptive mesh refinement is useful to lower the number of cells inside the fluid domain, and thus the amount of computations per time step. However, it can also be desirable to reduce the number of cells *outside* the fluid domain, as they can also be associated with a large computational and memory cost. This is because in cases of complex geometries the computational domain of Cartesian grid methods is by design a bounding box that surrounds the fluid domain. As such, if the volume fraction of the fluid domain in this bounding box is low and if a constant grid size is used, most of the computational cells are located inside the solid walls and a significant amount of memory and computational resources are allocated for these “solid” cells where no physics happens. This is especially true in cases of large, three-dimensional channel geometries. For instance, let us consider a helical pipe of radius R_{pipe} connecting an upper and a lower arm of length L_{arm} , where a capsule of radius $R_c = R_{\text{pipe}}/4$ is placed at the top of the geometry. If we assume this geometry is embedded in a bounding box of depth $2R_{\text{helix}} = 10R_{\text{pipe}}$, height $H_{\text{helix}} = 12R_{\text{helix}}$ and length $2L_{\text{arm}} = H_{\text{helix}}$, as shown in figure 17a, ensuring 16 grid cells per initial capsule diameter using a uniform Eulerian grid would require over 1.2 billion grid cells, rendering the computation extremely expensive. In contrast, using Basilisk’s adaptive mesh as shown in figure 17a allows to reduce the number of computational cells by a factor of about 200, down to less than 6 million grid cells. If only the helix itself is of interest and not the connecting arms, using a uniform Eulerian grid would require about 200 million grid cells, and using our adaptive solver would still reduce the number of grid cells by over a factor of 30.

As a demonstration of the capability of the present solver to handle such large and complex geometries, simulations are carried out in the helical geometry described above and shown in figure 17. A neo-Hookean capsule of radius R_c with $Ca = 0.1$ is placed on the upper straight pipe centerline at a distance of $1.5R_{\text{helix}}$ prior to entering the helix. We propose to study the inertial migration of this capsule for three Reynolds numbers: $Re = 10, 50$ and 100 . We set the non-dimensional time step $R_{\text{helix}}\Delta t/U$ to 10^{-3} , with U the characteristic velocity of the fluid; and we choose a finest Eulerian grid resolution corresponding to 16 grid cells per initial diameter. Each case ran for about three days on 96 processors, with around $6 \cdot 10^4$ cells per processor. To analyze the trajectories, we define the distance r^* of the capsule centroid to the helix centerline normalized by the pipe radius R_{pipe} , as well as the number of helical periods $N^* = (H_{\text{helix}} - z)/H_{1p}$, with H_{1p} the height of one vertical period of the helix, i.e. its pitch distance. A slice of the flow field in the helix for $Re = 100$ is shown in figure 17c, and a movie of this simulation as well as the code to reproduce it are available at [84]. The trajectories of the capsule are shown in the three-dimensional space in figure 17b, in one dimension by showing r^* as a function of N^* in figure 18a, and in a cross-section of the pipe orthogonal to the helix centerline in figure 18b. Only the path corresponding to the capsule located inside the helix is shown in these figures. Immediately after release, the capsule moves away from the centerline for all Reynolds numbers. The initial overshoot of r^*

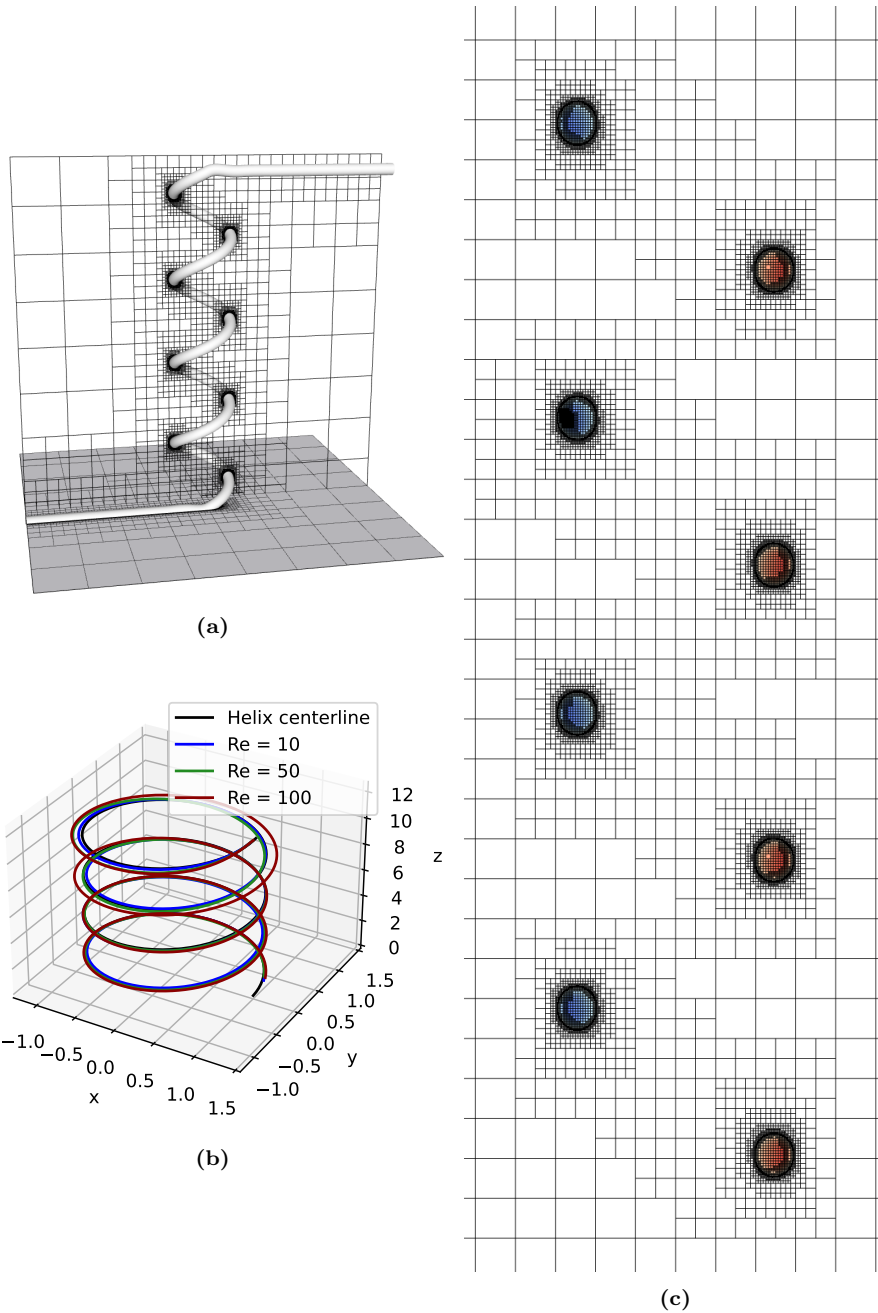
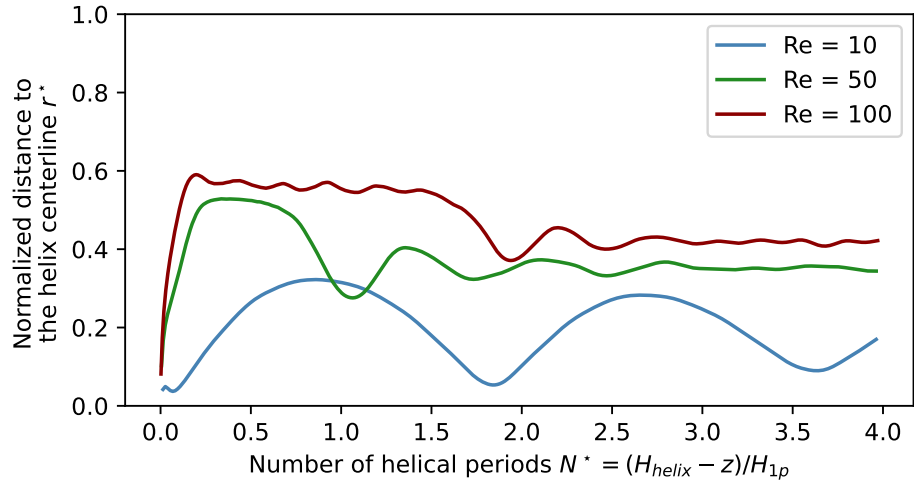
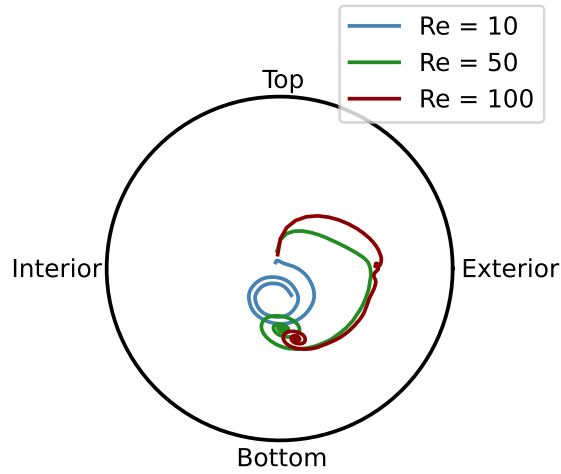


Figure 17: Adaptive mesh around a helical geometry and its connecting pipes: (a) full computational box around the whole geometry; (b) three-dimensional trajectory of a capsule at $Re = 10, 50, 100$ (for visual clarity the helix is shrunk in the vertical direction); (c) adaptive mesh and velocity field in the vertical plane. In (c), the color field corresponds to the x component of the velocity, where blue is into the page and orange is out of the page. The capsule is about to cross the vertical plane for the third time, hence the additional small cells inside the third circular cross-section from the top.



(a)



(b)

Figure 18: Trajectory of a solitary capsule in the helix at $Re = 10, 50, 100$: (a) normalized distance from the helix centerline; (b) radial migration in a plane orthogonal to the helix centerline. N^* corresponds to the number of revolutions around the vertical axis of the helix.

increases with the Reynolds number, from $r_{\max}^* \approx 0.3$ at $Re = 10$ to $r_{\max}^* \approx 0.6$ at $Re = 100$. After four helical revolutions, the capsule exits the helix with a steady position of $r_{\infty}^* \approx 0.38$ and $r_{\infty}^* \approx 0.45$ for $Re = 50$ and $Re = 100$ respectively. In the case of $Re = 10$, however, a steady state is not yet reached when the capsule exits the helix, but we can extrapolate the capsule trajectory to find that $r_{\infty}^* \approx 0.18$ for this Reynolds number. For all Reynolds numbers, the steady position is located in the lower half of the cross-section, at an angle θ from the horizontal line of about $-\pi/2$, although θ increases slightly with the Reynolds number. Interestingly, we note that the capsule transient path is longer for $Re = 100$ than that for $Re = 50$: as can be seen in figure 18, for $Re = 100$ the capsule seems to reach an unstable equilibrium at $r^* \approx 0.58$ and $\theta \approx 0$ for as long as 1.5 helical periods, before continuing its spiralling motion towards a stable steady-state location. Further investigation would be necessary to characterize this behavior and determine if, for instance, this unstable equilibrium corresponds to the center of a vortex, but this is not the focus of the present paper. The purpose of this simulation is to show that the present solver is able to simulate large three-dimensional channel geometries, and has the potential to simulate full microfluidic devices.

6. Conclusion and perspectives

We have presented an adaptive front-tracking solver to simulate deformable capsules in viscous flows at zero and finite inertia. The membrane mechanics is governed by an elastic and a bending resistance, and a non-unity viscosity ratio is allowed. Moreover, the present solver is compatible with complex STL-defined geometries, thus providing all the ingredients needed to simulate realistic flow configurations of biological cells, including red blood cells, both in-vivo and in-vitro. Numerous validation cases are presented against data available in the literature: we compare our results mainly to the highly accurate boundary integral method in Stokes conditions, and to other front-tracking methods at non zero Reynolds numbers. Very good qualitative and quantitative agreement is shown in all cases. We then demonstrate the robustness of the present solver in more challenging configurations, as well as its potential to tackle very large, three-dimensional channel geometries relevant to inertial microfluidic applications. Moreover, the present implementation is open-source as part of the Basilisk platform: the documented source code is freely available, as well as the source files to reproduce all the simulations presented in this paper [49].

Although the present adaptive front-tracking solver can simulate all the range of Reynolds numbers, the non-inertial limit is challenging because of the computation of the viscous term. The simulations we show in this paper at $Re = 0.01$ are several times slower to complete than their counterpart at, e.g., $Re = O(1)$ or $Re = O(10)$. As a result, if only the non-inertial regime is sought, boundary integral solvers likely remain the most efficient method by far. Another challenge is the stiffness of the bending stresses: since the Helfrich's bending formulation involves such high-order derivatives of the membrane geometry, and since the time integration of the membrane problem is explicit, the time

step is controlled by the time scale associated with the bending force whenever bending effects are included. This is a known challenge of computing bending stresses on elastic membranes (see, e.g., p. 40 of [55]). As a result, we have to decrease our time step by one order of magnitude (sometimes even more) whenever the bending stresses are included. Unfortunately, to our knowledge the stability condition associated with the bending force is unknown and it is therefore not possible to stabilize simulations by employing an adaptive time-step strategy, as is already the case in Basilisk with the CFL condition and with the celerity of capillary waves for surface tension stresses. One could investigate the implicit or “approximately implicit” treatment of the immersed boundary method as done by Roma, Peskin & Berger [42].

On the implementation side, the fluid solver from Basilisk is compatible with shared and distributed memory systems – i.e. using OpenMP and MPI libraries –, allowing to run simulations on large supercomputers. Naturally, we have enabled our front-tracking solver to be compatible with MPI as well. However, ensuring a good scaling with the number of capsules is not trivial as the domain decomposition of the Eulerian adaptive mesh is governed by a Z-ordering algorithm. As a result, when the Eulerian mesh is adaptive, the stencils attached to the Lagrangian nodes of a given capsule are most likely containing Eulerian cells handled by several distinct processors. In other words, a single capsule has to exist on many different processors in order to communicate with the background Eulerian mesh. Interpolating velocities from the fluid cells to the Lagrangian nodes, and spreading the Lagrangian forces to the fluid cells thus requires expensive inter-processor communications. Investigating efficient strategies to simulate a large number of capsules with an adaptive mesh is left for future works. That being said, in all the simulations shown in this study with one or two capsules considered, at most 5% of the total computation time is spent in the front-tracking solver. Consequently, unless simulating a large number of capsules – e.g., $O(100)$ –, the bottleneck is still the Navier-Stokes solver. Moreover when a large number of capsules is considered, one could argue that a uniform Eulerian mesh can be more efficient than its adaptive counterpart because a large number of capsules would likely result in a high volume fraction of capsules. An efficient parallel implementation of the present front-tracking solver restricted to uniform Eulerian grids is straightforward and could be implemented in future studies if dense volume fractions are considered. Another extension to the present solver could be to allow the triangulation of the membrane to be adaptive as well. Such adaptive triangulations have been considered to simulate fluid-fluid interfaces [33], but in the case of elastic membranes special care needs to be given to coarsen or refine the shape functions of a triangle. However only simulations featuring extreme membrane deformation would benefit from an adaptive membrane triangulation, as the front-tracking solver is only taking a few percents of the total computing time. For the applications we seek where reasonable membrane deformations are expected, the gain of an adaptive membrane triangulation is likely close to zero.

Another possible improvement to the present solver would be to allow the support of the regularized Dirac-delta functions to include grid cells of different

sizes. Indeed, the current method imposes a constant grid size in the vicinity of the membrane in order to apply the IBM in a straightforward manner, in a similar fashion to our Distributed Lagrange Multiplier/Fictitious Domain method implemented in Basilisk to simulate flows laden with rigid particles [85], but this can result in imposing a finer Eulerian grid resolution around the membrane than what is necessary to properly resolve some parts of the membrane. This scenario typically happens when the capsule is close or will come close to a sharp boundary, such as in the case of the narrow constriction in Section 5.1. In figure 15, for instance, as the capsule enters the narrow constriction, the Eulerian grid resolution around its tail is much finer than necessary due to the very fine grid resolution needed at the front of the capsule, which is located in a flow with strong gradients and close to sharp boundaries. A conceptually simple way to allow the size of the Eulerian grid cells to change along the membrane would be: (i) to propagate the forcing term or the averaging operator to smaller grid cells located inside the stencil of interest; and (ii) to increase the stencil size if a large grid cell is encountered, such that any stencil size is always four times larger in each direction than its largest grid cell. This adaptive extension of the immersed boundary method is under current investigation and shows promising results.

As demonstrated in Section 5.2, the current state of our adaptive solver allows resolved inertial simulations of capsule-laden flows in large three-dimensional geometries for a fraction of the computational cost of that of uniform front-tracking solvers implemented on uniform Cartesian grids. As such, our solver has the potential to provide valuable insight to help develop inertial migration microfluidic devices. The present solver may even allow the simulation of full microfluidic geometries consisting in several stacked layers of microfluidic channels, such as the spiralling geometries in the experimental work of Fang et al. [8]. This has the potential to provide valuable qualitative and quantitative information about the flow field, capsule dynamics and sorting efficiency of a given realistic microfluidic geometry, thus reducing the number of manufacturing iterations during the design process of such devices. Our medium-term objective is to tackle this type of virtual design problem, considering sub-domains of the full geometry as a first step, while concomitantly investigating the possible improvements stated above.

Acknowledgements

Damien P. Huet thanks Antoine Morente for designing the helix STL geometry and for helpful discussions. The authors greatly appreciate the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) via Anthony Wachs' Discovery Grant RGPIN-2016-06572. This research was enabled by support provided by Compute Canada (<http://www.computecanada.ca>) through Anthony Wachs's 2020 and 2021 Computing Resources for Research Groups allocation qpf-764-ac.

References

- [1] M. W. Dewhurst, T. W. Secomb, Transport of drugs from blood vessels to tumour tissue, *Nature Reviews Cancer* 17 (12) (2017) 738–750.
- [2] D. F. Puleri, P. Balogh, A. Randles, Computational models of cancer cell transport through the microcirculation, *Biomechanics and Modeling in Mechanobiology* 20 (4) (2021) 1209–1230.
- [3] P. Balogh, J. Gounley, S. Roychowdhury, A. Randles, A data-driven approach to modeling cancer cell mechanics during microcirculatory transport, *Scientific Reports* 11 (1) (2021) 1–18.
- [4] E. Islamzada, K. Matthews, Q. Guo, A. T. Santoso, S. P. Duffy, M. D. Scott, H. Ma, Deformability based sorting of stored red blood cells reveals donor-dependent aging curves, *Lab on a Chip* 20 (2) (2020) 226–235.
- [5] S. R. Bazaz, A. Mihandust, R. Salomon, H. A. N. Joushani, W. Li, H. A. Amiri, F. Mirakhorli, S. Zhand, J. Shrestha, M. Miansari, et al., Zigzag microchannel for rigid inertial separation and enrichment (z-rise) of cells and particles, *Lab on a Chip* (2022).
- [6] N. Takeishi, H. Yamashita, T. Omori, N. Yokoyama, S. Wada, M. Sugihara-Seki, Inertial migration of red blood cells under a newtonian fluid in a circular channel, *arXiv preprint arXiv:2209.11933* (2022).
- [7] A. Gangadhar, S. A. Vanapalli, Inertial focusing of particles and cells in the microfluidic labyrinth device: Role of sharp turns, *Biomicrofluidics* 16 (4) (2022) 044114.
- [8] Y. Fang, S. Zhu, W. Cheng, Z. Ni, N. Xiang, Efficient bioparticle extraction using a miniaturized inertial microfluidic centrifuge, *Lab on a Chip* 22 (18) (2022) 3545–3554.
- [9] D. Barthes-Biesel, J. Rallison, The time-dependent deformation of a capsule freely suspended in a linear shear flow, *Journal of Fluid Mechanics* 113 (1981) 251–267.
- [10] C. Pozrikidis, Finite deformation of liquid capsules enclosed by elastic membranes in simple shear flow, *Journal of Fluid Mechanics* 297 (1995) 123–152.
- [11] S. Ramanujan, C. Pozrikidis, Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: large deformations and the effect of fluid viscosities, *Journal of fluid mechanics* 361 (1998) 117–143.
- [12] C. D. Eggleton, A. S. Popel, Large deformation of red blood cell ghosts in a simple shear flow, *Physics of fluids* 10 (8) (1998) 1834–1845.
- [13] C. Pozrikidis, Effect of membrane bending stiffness on the deformation of capsules in simple shear flow, *Journal of Fluid Mechanics* 440 (2001) 269.

- [14] C. Pozrikidis, Numerical simulation of the flow-induced deformation of red blood cells, *Annals of biomedical engineering* 31 (10) (2003) 1194–1205.
- [15] 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 (10) (2010) 3726–3744.
- [16] E. Lac, D. Barthes-Biesel, N. Pelekasis, J. Tsamopoulos, Spherical capsules in three-dimensional unbounded stokes flows: effect of the membrane constitutive law and onset of buckling, *Journal of Fluid Mechanics* 516 (2004) 303–334.
- [17] E. Lac, A. Morel, D. Barthès-Biesel, Hydrodynamic interaction between two identical capsules in simple shear flow, *Journal of Fluid Mechanics* 573 (2007) 149–169.
- [18] P. Bagchi, Mesoscale simulation of blood flow in small vessels, *Biophysical journal* 92 (6) (2007) 1858–1877.
- [19] S. K. Doddi, P. Bagchi, Lateral migration of a capsule in a plane poiseuille flow in a channel, *International Journal of Multiphase Flow* 34 (10) (2008) 966–986.
- [20] A. Z. Yazdani, P. Bagchi, Phase diagram and breathing dynamics of a single red blood cell and a biconcave capsule in dilute shear flow, *Physical Review E* 84 (2) (2011) 026314.
- [21] A. Yazdani, P. Bagchi, Three-dimensional numerical simulation of vesicle dynamics using a front-tracking method, *Physical Review E* 85 (5) (2012) 056308.
- [22] A. Yazdani, P. Bagchi, Influence of membrane viscosity on capsule dynamics in shear flow, *Journal of fluid mechanics* 718 (2013) 569–595.
- [23] S. K. Doddi, P. Bagchi, Effect of inertia on the hydrodynamic interaction between two liquid capsules in simple shear flow, *International journal of multiphase flow* 34 (4) (2008) 375–392.
- [24] P. Balogh, P. Bagchi, A computational approach to modeling cellular-scale blood flow in complex geometry, *Journal of Computational Physics* 334 (2017) 280–307.
- [25] P. Balogh, P. Bagchi, Analysis of red blood cell partitioning at bifurcations in simulated microvascular networks, *Physics of Fluids* 30 (5) (2018) 051902.
- [26] X. Li, K. Sarkar, Front tracking simulation of deformation and buckling instability of a liquid capsule enclosed by an elastic membrane, *Journal of Computational Physics* 227 (10) (2008) 4998–5018.

- [27] J. Zhang, P. C. Johnson, A. S. Popel, An immersed boundary lattice boltzmann approach to simulate deformable liquid capsules and its application to microscopic blood flows, *Physical biology* 4 (4) (2007) 285.
- [28] J. Zhang, P. C. Johnson, A. S. Popel, Red blood cell aggregation and dissociation in shear flows simulated by lattice boltzmann method, *Journal of biomechanics* 41 (1) (2008) 47–55.
- [29] J. Ames, D. F. Puleri, P. Balogh, J. Gounley, E. W. Draeger, A. Randles, Multi-gpu immersed boundary method hemodynamics simulations, *Journal of computational science* 44 (2020) 101153.
- [30] D. A. Fedosov, B. Caswell, G. E. Karniadakis, A multiscale red blood cell model with accurate mechanics, rheology, and dynamics, *Biophysical journal* 98 (10) (2010) 2215–2225.
- [31] C. W. Hirt, B. D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, *Journal of computational physics* 39 (1) (1981) 201–225.
- [32] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, *Journal of computational physics* 100 (2) (1992) 335–354.
- [33] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, *Journal of computational physics* 169 (2) (2001) 708–759.
- [34] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, *Journal of Computational Physics* 228 (16) (2009) 5838–5866.
- [35] G.-H. Cottet, E. Maitre, A level set method for fluid-structure interactions with immersed surfaces, *Mathematical models and methods in applied sciences* 16 (03) (2006) 415–438.
- [36] S. Ii, X. Gong, K. Sugiyama, J. Wu, H. Huang, S. Takagi, A full eulerian fluid-membrane coupling method with a smoothed volume-of-fluid approach, *Communications in Computational Physics* 12 (2) (2012) 544.
- [37] S. Ii, K. Sugiyama, S. Takagi, Y. Matsumoto, A computational blood flow analysis in a capillary vessel including multiple red blood cells and platelets, *Journal of Biomechanical Science and Engineering* 7 (1) (2012) 72–83.
- [38] S. Ii, K. Shimizu, K. Sugiyama, S. Takagi, Continuum and stochastic approach for cell adhesion process based on eulerian fluid-capsule coupling with lagrangian markers, *Journal of Computational Physics* 374 (2018) 769–786.
- [39] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *Journal of Computational Physics* 209 (2) (2005) 448–476.

- [40] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, *Journal of Computational Physics* 231 (9) (2012) 3663–3684.
- [41] W.-P. Breugem, A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows, *Journal of Computational Physics* 231 (13) (2012) 4469–4498.
- [42] A. Roma, C. Peskin, M. Berger, An adaptive version of the immersed boundary method, *Journal of Computational Physics* 153 (2) (1999) 509–534.
- [43] B. E. Griffith, R. D. Hornung, D. M. McQueen, C. S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, *Journal of computational physics* 223 (1) (2007) 10–49.
- [44] M. Vanella, A. Posa, E. Balaras, Adaptive mesh refinement for immersed boundary methods, *Journal of Fluids Engineering* 136 (4) (2014).
- [45] G. Agresar, J. Linderman, G. Tryggvason, K. Powell, An adaptive, cartesian, front-tracking method for the motion, deformation and adhesion of circulating cells, *Journal of Computational Physics* 143 (2) (1998) 346–380.
- [46] Z. Cheng, A. Wachs, An immersed boundary/multi-relaxation time lattice boltzmann method on adaptive octree grids for the particle-resolved simulation of particle-laden flows, *Journal of Computational Physics* (2022) 111669.
- [47] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries, *Journal of Computational Physics* 190 (2) (2003) 572–600.
- [48] S. Popinet, A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations, *Journal of Computational Physics* 302 (2015) 336–358.
- [49] D. P. Huet, <http://basilisk.fr/sandbox/huet>, accessed: 2022-10-05 (2022).
- [50] A. E. Green, J. E. Adkins, Large elastic deformations and non-linear continuum mechanics (1960).
- [51] D. Barthes-Biesel, A. Diaz, E. Dhenin, Effect of constitutive laws for two-dimensional membranes on flow-induced capsule deformation, *Journal of Fluid Mechanics* 460 (2002) 211–222.
- [52] J. Charrier, S. Shrivastava, R. Wu, Free and constrained inflation of elastic membranes in relation to thermoforming—non-axisymmetric problems, *The Journal of Strain Analysis for Engineering Design* 24 (2) (1989) 55–74.
- [53] W. Helfrich, Elastic properties of lipid bilayers: theory and possible experiments, *Zeitschrift für Naturforschung c* 28 (11-12) (1973) 693–703.

- [54] A. Guckenberger, S. Gekle, Theory and algorithms to compute Helfrich bending forces: A review, *Journal of Physics: Condensed Matter* 29 (20) (2017) 203001.
- [55] D. Barthes-Biesel, Motion and deformation of elastic capsules and vesicles in flow, *Annual Review of fluid mechanics* 48 (2016) 25–52.
- [56] A. Guckenberger, M. P. Schraml, P. G. Chen, M. Leonetti, S. Gekle, On the bending algorithms for soft objects in flows, *Computer Physics Communications* 207 (2016) 1–23.
- [57] A. J. Chorin, Numerical solution of the Navier-Stokes equations, *Mathematics of computation* 22 (104) (1968) 745–762.
- [58] J. B. Bell, P. Colella, H. M. Glaz, A second-order projection method for the incompressible Navier-Stokes equations, *Journal of Computational Physics* 85 (2) (1989) 257–283.
- [59] J. A. Van Hooff, S. Popinet, C. C. Van Heerwaarden, S. J. Van der Linden, S. R. de Roode, B. J. Van de Wiel, Towards adaptive grids for atmospheric boundary-layer simulations, *Boundary-layer meteorology* 167 (3) (2018) 421–443.
- [60] H. Johansen, P. Colella, A cartesian grid embedded boundary method for Poisson’s equation on irregular domains, *Journal of Computational Physics* 147 (1) (1998) 60–85.
- [61] P. Colella, D. T. Graves, B. J. Keen, D. Modiano, A cartesian grid embedded boundary method for hyperbolic conservation laws, *Journal of Computational Physics* 211 (1) (2006) 347–366.
- [62] S. Popinet, <http://basilisk.fr/src/embed.h#lifting-the-small-cell-cfl-restriction>, accessed: 2022-10-05 (2018).
- [63] A. Ghigo, <http://basilisk.fr/sandbox/ghigo/src/myembed.h>, accessed: 22-11-2022 (2021).
- [64] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *Journal of computational physics* 100 (1) (1992) 25–37.
- [65] C. Peskin, Numerical analysis of blood flow in the heart, *Journal of Computational Physics* 25 (3) (1977) 220–252.
- [66] L. Lu, M. J. Morse, A. Rahimian, G. Stadler, D. Zorin, Scalable simulation of realistic volume fraction red blood cell flows through vascular networks, in: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019, pp. 1–30.

- [67] A. Farutin, T. Biben, C. Misbah, 3d numerical simulations of vesicle and in-extensible capsule dynamics, *Journal of Computational Physics* 275 (2014) 539–568.
- [68] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: *Visualization and mathematics III*, Springer, 2003, pp. 35–57.
- [69] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/uniaxial_stretch.c, accessed: 2022-10-05 (2022).
- [70] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/biconcave_curvatures.c, accessed: 2022-10-05 (2022).
- [71] I. Koleva, H. Rehage, Deformation and orientation dynamics of polysiloxane microcapsules in linear shear flow, *Soft Matter* 8 (13) (2012) 3681–3693.
- [72] 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 (7) (2010) 829–850.
- [73] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/nh_shear.c, accessed: 2022-10-05 (2022).
- [74] L. Zhu, L. Brandt, The motion of a deforming capsule through a corner, *Journal of Fluid Mechanics* 770 (2015) 374–397.
- [75] D. V. Le, Effect of bending stiffness on the deformation of liquid capsules enclosed by thin shells in shear flow, *Physical Review E* 82 (1) (2010) 016318.
- [76] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/bending_shear.c, accessed: 2022-10-05 (2022).
- [77] S.-Y. Park, P. Dimitrakopoulos, Transient dynamics of an elastic capsule in a microfluidic constriction, *Soft matter* 9 (37) (2013) 8844–8855.
- [78] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/constricted_channel.c, accessed: 2022-10-05 (2022).
- [79] C. Pozrikidis, Resting shape and spontaneous membrane curvature of red blood cells, *Mathematical Medicine and Biology: A Journal of the IMA* 22 (1) (2005) 34–52.
- [80] C. Pozrikidis, *Computational hydrodynamics of capsules and biological cells* (2010).
- [81] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/rbc_shear.c, accessed: 2022-10-05 (2022).

- [82] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/caps_interception.c, accessed: 2022-10-05 (2022).
- [83] D. P. Huet, http://basilisk.fr/sandbox/huet/tests/lagrangian_caps/caps_interception_inertia.c, accessed: 2022-10-05 (2022).
- [84] D. P. Huet, http://basilisk.fr/sandbox/huet/cases/lagrangian_caps/helix.c, accessed: 2022-10-05 (2022).
- [85] C. Selçuk, A. R. Ghigo, S. Popinet, A. Wachs, A fictitious domain method with distributed lagrange multipliers on adaptive quad/octrees for the direct numerical simulation of particle-laden flows, *Journal of Computational Physics* 430 (2021) 109954.