An ALE method for simulations of axisymmetric elastic surfaces in flow

Marcel Mokbel | Sebastian Aland

Faculty of Informatics and Mathematics, HTW Dresden, Dresden, Germany

Correspondence
Sebastian Aland, Faculty of Informatics and Mathematics, HTW Dresden, Friedrich-List-Platz 1, 01069 Dresden, Germany. Email: sebastian.aland@htw-dresden.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Number: AL1705/3, AL1705/5, FOR3013

Summary
The dynamics of membranes, shells, and capsules in fluid flow has become an active research area in computational physics and computational biology. The small thickness of these elastic materials enables their efficient approximation as a hypersurface, which exhibits an elastic response to in-plane stretching and out-of-plane bending, possibly accompanied by a surface tension force. In this work, we present a novel arbitrary Lagrangian-Eulerian (ALE) method to simulate such elastic surfaces immersed in Navier-Stokes fluids. The method combines high accuracy with computational efficiency, since the grid is matched to the elastic surface and can, therefore, be resolved with relatively few grid points. The focus of this work is on axisymmetric shapes and flow conditions, which are present in a wide range of biophysical problems. We formulate axisymmetric elastic surface forces and propose a discretization with surface finite-differences coupled to evolving finite elements. We further develop an implicit coupling strategy to reduce time step restrictions. We show in several numerical test cases that accurate results can be achieved at computational times on the order of minutes on a single core CPU. We demonstrate two state-of-the-art applications which to our knowledge cannot be simulated with any other numerical method so far: we present first simulations of the observed shape oscillations of novel microswimming shells and the uniaxial compression of the cortex of a biological cell during an AFM experiment.

KEYWORDS
ALE, fluid-structure interaction, surface elasticity, thin shell, two-phase flow

1 INTRODUCTION

The dynamics of elastic membranes, shells and capsules in fluid flow has become an active research area in computational physics and computational biology. Example systems include vesicle membranes immersed in fluids,1-4 red, and white blood cells transported with the blood plasma,5-8 general biological cells including cytoplasmic flows,9,10 or even man-made elastic thin shells to deliver cargo through a fluid.11 Therefore, there exists a broad interest in advanced modeling and simulation technologies that enable the understanding of such systems.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. International Journal for Numerical Methods in Fluids published by John Wiley & Sons Ltd.

1604 | wileyonlinelibrary.com/journal/fld

Int J Numer Meth Fluids. 2020;92:1604–1625.
In all these examples, the elastic material is typically very thin such that its direct numerical resolution with continuum-based discretization techniques requires prohibitively fine mesh sizes. This problem can be circumvented by replacing volumes of thin layers by dimensionally reduced surfaces, that is, the elastic material is approximated as a hypersurface of zero thickness. Throughout this article, we are particularly interested in elastic surfaces with a finite shear modulus which prevents the surface from strong tangential deformations (while out-of-plane deformation might still be sizable). We, therefore, explicitly exclude pure lipid vesicles and focus on the more relevant case of thin shells. One prominent example of which is cell surfaces. These are composed of a membrane connected to a thin elastic actin cortex, and the membrane/cortex complex thus forms a thin elastic sheet with strong resistance to surface shear and dilation. Mechanical properties of the surface include in-plane stretch elasticity and out-of-plane bending elasticity. Additionally, surface tension forces may arise, for example, in cellular membranes and these forces stem from microscopic motor proteins that permanently try to contract the surface. Together with the hydrodynamics of the surrounding fluids, all the forces lead to a tightly coupled system of flow and surface evolution.

Several methods have been developed to numerically simulate such systems. Most popular among them are the boundary integral method, the immersed boundary method, and particle collision methods. Boundary element methods and boundary integral methods couple the Stokes equations to thin shell theory for the elastic surface. The methods are very efficient as they reduce the system to a pure surface problem, yet the limitation to the Stokes regime restricts these methods to small length scales and small flow/shear rates. An alternative approach is particle methods. Recently, many studies used the multiparticle collision dynamics model to describe elastic capsules. This numerical scheme is very flexible but remains partly phenomenological. The immersed boundary method in its original form couples different numerical grids for the surface and the fluid domain by use of smeared-out delta functions. Main problems of such methods are the loss in accuracy associated to this interpolation between grid and the handling of high viscosity ratios. These problems can be overcome by coupling immersed boundaries with technically far more complicated cut cell methods.

Alternative to the above methods, interface capturing methods can be used to track interface movement. The most prominent of which are the level set method and the phase field method. However, the inclusion of shear and dilational surface elasticity is traditionally not considered in these approaches, as it is not clear how to carry the reference coordinates along the elastic structure. Notably, some first steps have been done in this direction recently for level set and phase-field methods.

The arbitrary Lagrangian-Eulerian (ALE) method is another approach to discretize moving domains and moving boundaries. The method uses a body-fitted grid to couple the advantages of Eulerian and Lagrangian description of the material. To achieve a certain flexibility in terms of solvers, partitioned approaches of ALE are very common. These approaches explicitly couple two separate momentum equations, one for the fluid and one for the solid. This leads to stability problems, the most famous of which being the added mass effect. A further drawback of ALE methods is the necessity of re-triangulation when strong grid deformations are involved. However, in 3D fluid-structure interaction (FSI) problems, these drawbacks are compensated by some distinct advantages of ALE methods. These include their efficiency, the high accuracy representation of the domains by a body-fitted grid, and the simplicity of implementation. While the ALE method is the standard method for the interaction of 3D elastic structures with surrounding fluids, it is rarely used for elastic surfaces immersed in fluids, see References for examples. In these methods, the thickness of the surface is typically resolved by use of special shell or beam elements, which have to be carefully chosen to prevent shear locking. Typically, separate momentum equations are solved in the fluid and the structure.

In this article, we aim to propose a different ALE method that is particularly suited for very thin elastic materials. This slenderness implies that no significant momentum is carried by the structure, which allows neglecting the corresponding momentum balance by including the elastic response as a surface force in the fluid. Accordingly, problems like the added mass effect or shear locking are eliminated. We restrict our model to the axisymmetric setting, which applies to many surfaces in nature and technology and reduces the system effectively to a two-dimensional problem. This makes the method particularly attractive for problems involving long-term computations (many time steps) or extremely fine grids, for which full 3D simulations are illusive. Along these lines, an axisymmetric boundary element method and an axisymmetric immersed boundary method have been proposed recently to simulate vesicles and elastic hyper-surfaces. Here, we propose an axisymmetric ALE method, which complements these methods by some distinct advantages. In the context of two-phase flows, axisymmetric ALE methods have been developed and benchmarked, see, eg, References, proving superior efficiency and accuracy in comparison to other two-phase flow methods. Our work adds the elastic forces at the material interface to obtain an efficient and accurate method for elastic surfaces in fluid flows. Accordingly, we provide a formulation for the elastic surface forces and propose a discretization with surface finite-differences, which is coupled to
evolving finite elements (FEs) of the bulk problems. We further develop an implicit coupling strategy to reduce time step restrictions. While the method is presented here for surfaces that are surrounded by fluids from both sides, it is straightforward to omit one of the fluid phases. Similarly, the method can easily account for completely different physics in both surrounding domains. We illustrate this by simulating the uniaxial compression of a biological cell filled with intracellular fluid and the propulsion of novel microswimmers having gaseous/liquid fluids inside/outside their elastic shell.

The rest of this article is structured as follows. In Section 2, the governing equations for the model are introduced along with the suitable boundary conditions to ensure force balance across the membrane. Details about the membrane forces are provided in Section 3. The numerical discretization of the problem is presented in Section 4. Numerical test cases are presented in Section 5. Some details on the implementation can be found in the Appendix.

2 | GOVERNING EQUATIONS

Consider a (closed) surface of revolution immersed in a cylindrical 3D domain, representing an elastic membrane immersed in a fluid (Figure 1A). Half of the domains cross section acts as the two-dimensional computational domain $\Omega$ (Figure 1B). $\Omega$ is composed of two separate parts, the exterior $\Omega_0(t)$ and the interior $\Omega_1(t)$, both describing incompressible viscous fluids: $\Omega = \text{int}(\Omega_0 \cup \Omega_1)$. The interface $\Gamma(t) = \Omega_0(t) \cap \Omega_1(t)$, which separates both domain parts, corresponds physically to an elastic shell and is described by the parametric form

$$X(s, t) = (X(s, t), R(s, t))^T,$$

with the arc length parameter $s$. The $x$-axis is chosen to be the symmetry axis and the $r$-axis is the distance to the symmetry axis in the considered 2D meridian $x - r$ domain. With the divergence operator for cylindrical coordinate systems of the form\cite{8}

$$\tilde{\nabla} \cdot = \left( \frac{\partial}{\partial r}, \frac{1}{r} \frac{\partial}{\partial \theta} \right),$$

the Navier-Stokes equations for the external ($i = 0$) and internal ($i = 1$) fluid read

$$\rho_i (\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) + \nabla p_i = \tilde{\nabla} \cdot \left( \eta_i \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right) \right) + \frac{2\eta_i}{r^2} v_r \cdot \left( \begin{array}{c} 0 \\ 1 \end{array} \right), \quad \text{in } \Omega_i$$

$$\tilde{\nabla} \cdot \mathbf{v} = 0, \quad \text{in } \Omega_i,$$

with the velocity $\mathbf{v} = (v_x, v_r)^T$ in $\Omega \cup \Gamma$, pressures $p_i$, viscosities $\eta_i$, and densities $\rho_i$ in $\Omega_i$, respectively. The velocity is continuous across $\Gamma$ and can hence be defined in the whole domain, while the pressure is discontinuous across $\Gamma$ and has therefore to be defined separately in both phases. The following jump condition holds at the interface to ensure the force
balance at the elastic membrane

\[ [-p I + \eta (\nabla v + (\nabla v)^T)] \cdot n = -\frac{\partial E}{\partial \Gamma}, \]  

(5)

where \( n \) is the interface normal pointing to \( \Omega_0 \) and \( [f] = f_0 - f_1 \) denotes the jump operator across the interface \( \Gamma \). The surface force term \( \frac{\partial E}{\partial \Gamma} \) is defined in the following. At the symmetry axis we specify the usual free slip condition.

3 | SHELL MEMBRANE FORCES

The membrane is assumed to be an isotropic thin shell with a thickness \( d \). The force response of a membrane to elastic deformations can be described with in-plane and out-of-plane forces acting to minimize the corresponding elastic energies. The in-plane energy is also referred as stretching energy \( E_{\text{stretch}} \) and the out-of-plane energy as bending energy \( E_{\text{bend}} \). Additionally, a surface tension energy \( E_{\text{tension}} \) can be present. The membrane forces are then given by the first variation of these energies with respect to changes in \( \Gamma \)

\[ \frac{\partial E}{\partial \Gamma} = \frac{\partial E_{\text{tension}}}{\partial \Gamma} + \frac{\partial E_{\text{bend}}}{\partial \Gamma} + \frac{\partial E_{\text{stretch}}}{\partial \Gamma}. \]  

(6)

The surface tension energy, tending to minimize the membrane area, reads

\[ E_{\text{tension}} = \int_{\Gamma} \gamma \, dA, \]  

(7)

with the material specific surface tension \( \gamma [N/m] \), which is assumed to be constant on the whole membrane.

Bending stiffness is the resistance of the membrane against bending (out-of-plane) deformations. The bending energy tends to minimize the deviation of the local curvature from the material’s reference curvature (also termed spontaneous curvature). The bending energy reads

\[ E_{\text{bend}} = \int_{\Gamma} \frac{c_b}{8} (\kappa - \kappa_{\text{ref}})^2 \, dA, \]  

(8)

with the material specific bending stiffness \( c_b [Nm] \) and the total curvature \( \kappa = \nabla \cdot n \), which is twice the mean curvature of the membrane. The spontaneous curvature \( \kappa_{\text{ref}} \) is in many practical cases either zero or the total curvature in the initial state, depending on the physical context of the problem.

The stretching energy \( E_{\text{stretch}} \) minimizes in-plane stretching and compression of the membrane compared with the reference state. In the axisymmetric setting, it is useful to describe the stretching energy in terms of the two principal stretches \( \lambda_1 \) and \( \lambda_2 \), which provide information about relative changes of surface lengths in lateral and circumferential direction, respectively. An illustration can be found in Figure 2A. The principal stretches read

\[ \lambda_1 = \frac{ds}{ds_{\text{ref}}}, \quad \lambda_2 = \frac{R}{R_{\text{ref}}}, \]  

(9)

with the subscript \( \text{ref} \) corresponding to the quantities at the same material point in the reference state.

In 3D elasticity theory, the response of an isotropic elastic body to elastic deformations can be described by two material specific parameters: Young’s modulus \( E \) and the Poisson ratio \( \nu \). For a thin elastic material of thickness \( d \), these parameters are typically reformulated into surface parameters, for example, the area dilation modulus \( K_A \) and area shear modulus \( K_S \). Considering a rectangular surface element, \( K_A \) describes the response of the membrane to in-plane area changes with constant aspect ratio of the surface element (Figure 2B). \( K_S \) provides information about the response to in-plane shear deformations with constant area of the surface element (Figure 2C). All elastic surface parameters can be calculated directly from Young’s modulus, Poisson ratio, and shell thickness:

\[ K_A = \frac{dE}{2(1 - \nu)}, \quad K_S = \frac{dE}{2(1 + \nu)}, \quad c_b = \frac{d^3E}{24(1 - \nu^2)}. \]  

(10)
In terms of $K_A$ and $K_S$, the stretching energy can in the linear regime be written as

$$E_{\text{stretch}} = \int_{\Gamma} \frac{K_A + K_S}{2} \left( (\lambda_1 - 1)^2 + (\lambda_2 - 1)^2 \right) + (K_A - K_S) (\lambda_1 - 1) (\lambda_2 - 1) \, dA. \quad (11)$$

Alternatively, the stretching energy of a nonlinear material can be used. For example, the neo-Hookean energy can be expressed by the left Cauchy-Green strain tensor, which can be easily constructed from the principle stretches $\lambda_1$, $\lambda_2$. The corresponding forces to the respective energies can be obtained by calculating the first variations of the three surface energies:

$$\frac{\partial E_{\text{tension}}}{\partial \Gamma} = g \kappa \mathbf{n}, \quad (12)$$

$$\frac{\partial E_{\text{bend}}}{\partial \Gamma} = c_0 \left[ \Delta \Gamma (\kappa - \kappa_{\text{ref}}) + (\kappa^2 - 2K)(\kappa - \kappa_{\text{ref}}) - \frac{1}{2} \kappa (\kappa - \kappa_{\text{ref}})^2 \right] \cdot \mathbf{n}, \quad (13)$$

$$\frac{\partial E_{\text{stretch}}}{\partial \Gamma} = (\kappa \mathbf{n} - \nabla \Gamma) \left[ (K_A + K_S) (\lambda_1 - 1) + (K_A - K_S) (\lambda_2 - 1) \right] - \frac{2K_S}{R} (\lambda_1 - \lambda_2) \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (14)$$

where $K$ is the Gaussian curvature, $\Delta \Gamma$ is the surface Laplacian, and $\nabla \Gamma$ is the surface gradient. Derivations can be found in References 9 and 35.

### 4 | NUMERICAL SCHEME

#### 4.1 | Time discretization

The problem is discretized in time with equidistant time steps of size $\tau$. The complete system is split into several sub-problems that are solved subsequently in each time step. At first, the membrane forces are computed from the current membrane shape. Afterward, the Navier-Stokes equations are solved by an implicit Euler method using the previously computed membrane forces. The resulting velocity is then used to advect the surface and the domains.

Using the ALE approach, the material derivative of the velocity $\partial_t^* \mathbf{v} = d_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}$ is discretized in the nth time step as follows

$$\partial_t^* \mathbf{v}^n = \frac{\mathbf{v}^n - \mathbf{v}^{n-1}_{\text{moved}}}{\tau} + \left( \mathbf{v}^{n-1} - \mathbf{v}^{n-1}_{\text{grid}} \right) \cdot \nabla \mathbf{v}^n,$$

where $\mathbf{v}^{n-1}_{\text{grid}}$ is the velocity of the grid movement from the previous time step calculated with one of the mesh smoothing algorithms presented in Section 4.4. This term has to be subtracted from the convection term due to the mesh update. The
FIGURE 3 Illustration of the workflow of the numerical solution procedure in each time step

velo \text{cy } v_{\text{prev}} \text{ moved} is the velocity of the last time step, but after the mesh update, that is, the grid point coordinates have been moved without changing the velocity values in each degree of freedom (DOF). The complete workflow of the numerical solution procedure is illustrated in Figure 3.

4.2 Space discretization

We use a finite-element method where the grids to represent the domains are matched at the immersed interface, that is, they share the same grid points. Accordingly, let \( T_h, i = 0, 1 \) be the triangulations of \( \Omega_i \) such that \( T_h = T_{h,0} \cup T_{h,1} \) is a conforming triangulation of \( \Omega \). The triangulation of the membrane is given by \( \Gamma_h = T_{h,0} \cap T_{h,1} \). An example for the corresponding numerical mesh is shown in Figure 4. The triangulations \( T_{h,0} \) and \( T_{h,1} \) are separated, that is, \( \Gamma_h \) acts as a boundary for both. This definition of the mesh ensures the possibility of continuous velocity and non-zero pressure jump across \( \Gamma_h \).

In order to obtain the membrane force, the tangential \( t \) and the values for \( n, \kappa, K, \Delta_r \kappa, \lambda_1, \) and \( \lambda_2 \) have to be calculated. Consider the parametrization \( X(s) \) with the arc length parameter \( s \) (hence, \( \| X_s \| = 1 \)), we obtain from

\[
t = (X_s, R_s)^T, \quad n = (-R_s, X_s)^T, \quad \nabla_{\Gamma} \lambda_k = t \partial_s \lambda_k, k \in \{0, 1\}
\]  

(15)

\[
\kappa = \frac{X_s}{R} + R_s X_{ss} - R_{ss} X_s, \quad K = \frac{X_s (R_s X_{ss} - R_{ss} X_s)}{R}, \quad \Delta_r \kappa = \frac{\kappa R_s}{R} + \kappa_{ss}.
\]  

(16)

In the discrete case, \( X_i = (X^i, R^i) \) for \( i = 0, \ldots, N - 1 \) is the sequence of membrane grid points ordered counterclockwise, approximating \( \Gamma \) by piecewise linear line segments \( (X^i, X^{i+1}) \). Then, the derivatives of \( X^i, R^i \) and \( \kappa^i \) with respect to \( s \) can be calculated with finite differences, here shown for \( X^i \), where \( i = 1, \ldots, N - 2 \)

\[
X^i_s = \frac{X^{i+1} - X^{i-1}}{\| X^{i+1} - X^{i-1} \|}, \quad X^i_{ss} = \frac{2}{\| X^{i+1} - X^{i-1} \|} \left( \frac{X^{i+1} - X^i}{\| X^{i+1} - X^{i-1} \|} - \frac{X^i - X^{i-1}}{\| X^i - X^{i-1} \|} \right).
\]  

(17)
The approximations for the principal stretches read
\[
\lambda_1^i = \frac{\left( \|X^{i+1} - X^i\| + \|X^i - X^{i-1}\| \right)}{\left( \|X_{\text{ref}}^{i+1} - X_{\text{ref}}^i\| + \|X_{\text{ref}}^i - X_{\text{ref}}^{i-1}\| \right)}, \quad \lambda_2^i = \frac{R_i}{R_{\text{ref}}^i}.
\] (18)

While the above definition of \( \lambda_1^i \) works well to obtain values for \( \lambda_1 \) at the vertices, it turned out to be numerically unstable to compute the derivative \( \partial_s \lambda_1 \). The problem can be illustrated as follows. The definition of \( \lambda_1 \) in Equation (18) effectively tracks changes in the distance between the left and right neighbor of a vertex. Hence, it is invariant to displacements where exactly every second vertex moves with the same velocity in tangential direction. Using these values to compute \( \partial_s \lambda_1 \) in practice leads to oscillations of every second vertex as soon as the time step size is large. This problem can be circumvented by replacing the vertex-based values \( \lambda_1^i \) by the following values, which represent the stretching at the edge midpoints between vertices \( i \) and \( i + 1 \).
\[
\lambda_1^{i+\frac{1}{2}} = \frac{\|X^{i+1} - X^i\|}{\|X_{\text{ref}}^{i+1} - X_{\text{ref}}^i\|}, \quad \lambda_1^{i-\frac{1}{2}} = \frac{\|X^i - X^{i-1}\|}{\|X_{\text{ref}}^i - X_{\text{ref}}^{i-1}\|}.
\] (19)

With these quantities, the derivatives of the principal stretches are computed,
\[
\partial_s \lambda_1^i = \frac{\lambda_1^{i+\frac{1}{2}} - \lambda_1^{i-\frac{1}{2}}}{\frac{1}{2} \left( \|X^{i+1} - X^i\| + \|X^i - X^{i-1}\| \right)}, \quad \partial_s \lambda_2^i = \frac{\lambda_2^{i+1} - \lambda_2^{i-1}}{\|X^{i+1} - X^{i-1}\|}.
\] (20)

To calculate the surface quantities on the symmetry axis (e.g., \( i = 0 \) and \( i = N - 1 \)), one takes advantage of the fact that \( R = R_{ss} = R_{sss} = 0 \) and \( X_0 = X_{ss} = 0 \) and applies l'Hospital's rule, see section 3.1 of Reference 24 for details.

### 4.3 Weak form

The fully discrete system in weak form is presented in the following. The FE spaces read
\[
V_h = \left\{ v \in H^1_0(\Omega) \mid v_{jk} \in P_2(k), k \in T_h \right\}
\] (21)

\[
M_{h,i} = \left\{ q \in L^2_0(\Omega) \cap C(\overline{\Omega}) \mid q_{jk} \in P_1(k), k \in T_{h,i} \right\}, \quad i = 0, 1,
\] (22)

where \( V_h \) is the FE space for the velocity. The resulting velocity will be continuous across \( \Gamma_h \). The FE spaces \( M_{h,i} \) for the pressures \( p_i \) are defined separately in \( T_{h,i} \) to allow discontinuous pressure across \( \Gamma_h \). Assuming constant viscosities \( \eta_i \) in \( T_{h,i} \), the weak form of the system given in Section 2 reads:

Find \( (v^n, p^n_0, p^n_1) \in V_h^2 \times M_{h,0} \times M_{h,1} \), s.t. \( \forall (w, q_0, q_1) \in V_h^2 \times M_{h,0} \times M_{h,1} : \)

\[
0 = \int_{\Omega_h^{(e)}} \rho_0 \left( \frac{v^n - v_0^{\text{moved}}}{\tau} + (v^{n-1} - v_0^{\text{grid}}) \cdot \nabla v^n \right) \cdot w + \eta_0 \nabla v^n \cdot \nabla w - p^n_0 v \cdot w \\
+ \left( 0, \frac{\eta_0}{R} \right) \cdot (\nabla v + (v)^T) \cdot w + \frac{2\eta_0}{R^2} v_r \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot w \ dx \\
+ \int_{\Omega^{(c)}} \rho_1 \left( \frac{v^n - v_0^{\text{moved}}}{\tau} + (v^{n-1} - v_0^{\text{grid}}) \cdot \nabla v^n \right) \cdot w + \eta_1 \nabla v^n \cdot \nabla w - p^n_1 v \cdot w \\
+ \left( 0, \frac{\eta_1}{R} \right) \cdot (\nabla v + (v)^T) \cdot w + \frac{2\eta_1}{R^2} v_r \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot w \ dx
\]
\[- \int_{\Gamma} \left( \frac{\partial E_{\text{stretch}}}{\partial \Gamma} + \frac{\partial E_{\text{tension}}}{\partial \Gamma} + \frac{\partial E_{\text{bend}}}{\partial \Gamma} \right)^{n-1} \cdot \mathbf{w} \, dA \]
\[0 = \int_{\Omega_h(t)} \hat{\mathbf{v}} \cdot \mathbf{v}^* q_0 \, dx + \int_{\Omega_h(t)} \hat{\mathbf{v}} \cdot \mathbf{v}^* q_1 \, dx, \tag{23}\]

with surface forces \( \frac{\partial E}{\partial \Gamma} \) defined by Equations (12) to (20) and the time step \( \tau \).

## 4.4 Mesh movement

The basic idea of the ALE approach is to move the grid of the elastic structure with the material velocity, while the fluid grid is moved with an arbitrary velocity keeping the mesh in a proper shape. Accordingly, in the present work, the membrane grid points are displaced with the velocity \( \mathbf{v} \) in every time step. As the membrane is moving, it is necessary to extend this movement to every grid point in \( T_h \) to keep the mesh well in shape. Two different strategies for rearranging grid points on a mesh with given boundary movement have been used for the simulations in the present work.

The first strategy (Section 4.4.1) is the usual harmonic extension of solid displacement into the fluid mesh. While this strategy is suitable for most practical cases, it might involve the accumulation of small errors. Especially for problems where many time steps are necessary (e.g., the simulations in Section 5.6), these errors lead to degenerated elements (e.g., the smallest angle of some triangles shrinks uninhibited). For such cases, we propose a second method in Section 4.4.2, which is designed to preserve element areas and edge lengths.

### 4.4.1 Solving Laplace problem

The first strategy involves the harmonic extension of interface movement by solving the Laplace problem

\[ \Delta \mathbf{v}_{\text{grid}} = 0 \quad \text{in } \Omega, \]
\[ \mathbf{v}_{\text{grid}} = 0 \quad \text{on } \partial \Omega \setminus \Gamma, \]
\[ \mathbf{v}_{\text{grid}} = \mathbf{v} \quad \text{on } \Gamma, \tag{24} \]

where \( \mathbf{v}_{\text{grid}} \) is the velocity of the grid points in \( T_h \). The mesh is then moved with \( \mathbf{v}_{\text{grid}} \). In some cases, it can be helpful to use the bi-Laplace problem instead:

\[ \Delta^2 \mathbf{v}_{\text{grid}} = 0 \quad \text{in } \Omega, \]
\[ \Delta \mathbf{v}_{\text{grid}} = \mathbf{v}_{\text{grid}} = 0 \quad \text{on } \partial \Omega \setminus \Gamma, \]
\[ \mathbf{v}_{\text{grid}} = \mathbf{v}, \quad \mathbf{n} \cdot \nabla \Delta \mathbf{v}_{\text{grid}} = 0 \quad \text{on } \Gamma. \tag{25} \]

Both approaches can be used in most cases. However, for strong and/or periodic deformations, some problems can occur using the Laplace Problem for mesh smoothing, for example, elements near the interface can degenerate slowly and cause a crash of the simulation. In this case, the mesh smoothing approach shown in the following may be a better choice.

### 4.4.2 Element area and length conservation

The second strategy is motivated by the fact that solving the Laplace problem equation 24 or the bi-Laplace problem equation 25 can lead to degenerated elements. To avoid this, it is helpful to penalize changes in area and edge length of the elements in order to prevent large deformations in both, the area and the aspect ratio of the elements. How to preserve element surface areas or grid point distances is described in Reference 50.

Assume that the boundary has been moved already. Iterate over all grid points \( x_i \). Check whether the length of each edge and/or the area of each element, which has \( x_i \) as a vertex, has changed (e.g., is \( A_{ref}^k/A^k \neq 1 \) for the area \( A^k \) of the \( k \)th element containing \( x_i \) as a vertex). If, for example, the area of element \( k \), which has the vertex \( x_i \), decreased after the movement of \( \Gamma \) (\( A_{ref}^k/A^k > 1 \)), \( x_i \) needs to be moved in order to re-increase the area of this element. This amounts to
moving $x_i$ in the direction $(1 - A^k_{ref}/A^k_n)\mathbf{n}_n^{k,i}$, where $\mathbf{n}_n^{k,i}$ is the normal of the opposite edge of $x_i$ in the $k$th element. Using the same approach to conserve edge lengths leads to Equation (26) below. The complete scheme for calculating the grid point movement in order to get the new grid reads:

1. Calculate areas $A$ and edge lengths $l_a$, $l_b$, $l_c$ of all elements using the point coordinates.
2. Compute the velocity of every grid point $x_i$ on $\partial T_h \cup \Gamma_h$, using the following formula

$$v_{grid}^{i} = \frac{1}{\tau} \sum_{e \in N(i)} c_a \left(1 - \frac{A^k_{ref}}{A^k_n}\right) \mathbf{n}_n^{k,i} + c_l \left(1 - \frac{l_b^{k,i}_{ref}}{l_b^{k,i}}\right) \mathbf{t}_b^{k,i} + c_l \left(1 - \frac{l_c^{k,i}_{ref}}{l_c^{k,i}}\right) \mathbf{t}_c^{k,i},$$

(26)

where $N(i) = \{e \in T_h \mid x_i \text{ is a vertex of } e\}$ is the set of all elements that share the vertex $x_i$, $l_b^{k,i}$ and $l_c^{k,i}$ are the two edge lengths of vertex $x_i$ in element $k$, and $\mathbf{t}_b^{k,i}$ and $\mathbf{t}_c^{k,i}$ are two the corresponding tangential vectors pointing away from $x_i$. The parameters $c_a$ and $c_l$ control the strength of area and edge length preservation, respectively. See also Figure 5 for a visualization of the quantities introduced in Equation (26).

3. Move all mesh points (except for the already moved surface points) by $\tau v_{grid}^{i}$.

Figure 6 shows an example for both, the Laplace smoothing approach and the area and length conservation approach. In both cases, a strong deformation has been imposed within a total amount of 500 time steps. The membrane shape at the end is circular with perfect agreement in both cases. The Laplace smoothing result (Figure 6B) seems a little more even in the internal fluid. However, the area and length conservation approach (Figure 6C) produces a better result: In the external fluid, the triangles near the membrane are less deformed. Around the symmetry axis, elements have been less compressed in $x$ direction whereas at the poles, they have been less stretched in $r$ direction (the latter is also visible in the internal fluid). These rather small improvements of the second approach can be quite important, for example, when due to a periodic deformation small errors that occur in the Laplace smoothing accumulate over time. Nevertheless, in this article, if not mentioned otherwise, the Laplace smoothing approach is used as it is independent of additional problem-specific parameters (like $c_a$, $c_l$).

## 5 NUMERICAL TESTS

Test case simulations were performed to show the quality and capability of the presented model, to verify the correctness of the interfacial forces and to analyze the mesh and time step stability.

### 5.1 Verification of the interfacial forces

In the following, we prescribe the elastic membrane as an initially oblate-shaped object, that is, its cross-section as the combination of two parallel lines and a semicircle (Figure 8A). The radius of the semicircle amounts to 0.1, the parallel
FIGURE 6  Example images for the two mesh smoothing approaches. A, The mesh near the membrane before the deformation. B and C, The mesh after a strong deformation during 500 time steps. B is the result of the Laplace smoothing approach, where C is the result of the area and length conservation approach [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 7  Shell volume over time for the stretching dominant case

lines have a length of 0.9, s.t. the surface of revolution has an equatorial radius of 0.55 and a height of 0.2. The membrane is located in the center of the computational domain $\Omega = [-2, 2] \times [0, 1]$. To verify the correctness of the three main interfacial forces, we consider the following three configurations:

1  Surface tension dominant case: $\gamma = 0.003$, $c_b = 0$, $K_A = 0$, $K_S = 0$.
2  Bending stiffness dominant case: $\gamma = 0$, $c_b = 11.1$, $K_A = 0.025$, $K_S = 0$.
3  Stretching dominant case: $\gamma = 0$, $c_b = 0$, $K_A = 0.025$, $K_S = 0.008333$. Here, the initial conditions for the principal stretches $\lambda_1$ and $\lambda_2$ are changed from 1 to 1.05 to induce a 5% pre stretch to the membrane. This causes non-zero stretching energy on the membrane.

The viscosities and densities are chosen equally in both phases and amount to $\eta_i = 1$, $\rho_i = 10^3$. Figure 7 shows that volume conservation is perfectly ensured in the simulations. The figure shows the volume over time for the stretching dominant case. The volume decreases in the beginning for a maximum amount of 0.085%. The other cases show similar behavior with even smaller volume changes in the beginning of the deformation.

In case 1, the membrane is expected to evolve into a sphere, since surface tension tends to minimize the surface area. This behavior is verified in the simulations, see Figure 8B.

In case 2, the stationary state is expected to yield a red blood cell like shape, since bending stiffness tends to minimize the curvature locally, while the additional influence of area dilation prevents the membrane to deform (or stretch) strong enough to get spherical. In this sense, case 2 is similar to a lipid vesicle, as the finite $K_A$ leads to approximate conservation of surface area. The stationary shapes of these simulations fit qualitatively well with theory Figure 8C.24

In case 3, in-plane elasticity penalizes stretching in tangential direction to the surface. Hence, the initial condition of $\lambda_1$ and $\lambda_2$ being larger than 1.0 causes a deformation of the membrane such that the radius of the oblate should decrease, where the thickness should somehow increase a bit. Note that, in the absence of volume conservation, the membrane would contract in both directions such that the principal stretches would approach 1.0 everywhere on the membrane.
However, the conservation of enclosed volume prevents the principal stretches from reaching 1.0, in general. The stationary state of the stretching dominant case is shown in (Figure 8C). The principal stretches in the stationary state are illustrated in Figure 9A. The equilibrium configuration of the elastic surface shows a significant stretch ($\lambda_1 > 1$) in lateral direction, which is necessary to accommodate the excess volume. This is accompanied by a compression in circumferential direction ($\lambda_2 < 1$) to minimize surface dilation.

In the numerical model, the Navier-Stokes equations are solved with separately defined pressures in $\Omega_0$ and $\Omega_1$, which leads to a discontinuous pressure field along $\Gamma$. The pressure field together with velocity glyphs is shown in Figure 9B for the bending stiffness dominant case.

5.2 Mesh resolution study

Three different mesh resolutions have been chosen to investigate the dependence of the simulation results on the mesh (Figure 10). In the following, we denote the mesh size by $h_i$ and number of surface grid points by $N_i$ for $i \in \{1, 2, 3\}$. The coarsest mesh has a mesh size of $h_1 = 0.055$ at the interface, and hence, the membrane is resolved by $N_1 = 23$ grid points. The complete mesh has 228 grid points. Refining this mesh by two triangle bisections leads to an intermediate mesh ($h_2 = 0.0275, N_2 = 45, 820$ total grid points). Two further bisections lead to the finest mesh ($h_3 = 0.01375, N_3 = 89, 3102$ total grid points).

Figure 11 shows the membrane points of the respective case for all chosen mesh resolutions. As can be seen, the method produces accurate results even with relatively coarse meshes. There is no visible difference between the shapes of all three meshes. To quantify that, we introduce two different error measures in the following.
The mean distance error between membrane points on the $h_1$- and $h_2$-mesh and the corresponding points on the next finer mesh are defined as

$$E^{hi} = \frac{1}{N_1} \sum_{j=0}^{N-1} \| X_{2i+j}^{h_i} - X_{2i+1}^{h_{i+1}} \|, \quad i \in \{1, 2\}. \quad (27)$$

For the $h_2$ (or $h_3$) mesh, every other (or 4th) membrane point is used, so only corresponding grid points (existing in the coarsest mesh) are compared.

As a second error indicator, we use the membrane cross section perimeter. Given that membrane points are sorted, we define the perimeter

$$P^h = \sum_{j=0}^{N-1} \| X_j^{h_i} - X_{j+1}^{h_i} \|, \quad i \in \{1, 2, 3\}, \quad (28)$$

where $X_{N_i}^{h_i} := X_0^{h_i}$. Then, the error is calculated using

$$E_P^h = \left| P^h - P^{h_{i+1}} \right|, \quad i \in \{1, 2\}. \quad (29)$$

With these values, the experimental order of convergence ($\text{EOC}_E$ and $\text{EOC}_P$) can be obtained

$$\text{EOC}_E = \frac{\ln E^{h_1}}{\ln \frac{h_1}{h_2}}, \quad \text{EOC}_P = \frac{\ln P^h}{\ln \frac{h_1}{h_2}}. \quad (30)$$
The obtained values are shown in Table 1. The point coordinates converge with order 1 and the areas/perimeters converge with order 2. A similar order of convergence is reached for the velocity using the $L_2$-norm of the velocity difference in $\Omega$. For example, in the stretching dominant case, we obtain EOC 1.43 for the velocity in $x$-direction ($E^{h_1}_v = 6.6412 \cdot 10^{-8}$, $E^{h_1}_e = 2.4178 \cdot 10^{-8}$).

### 5.3 Time step study

According to the previous section, it seems sufficient to do further studies using the $h_1$-mesh. The three different cases are now being analyzed using different time step sizes. Figure 12A-C illustrate the shape evolution of the surface tension case (tested with timesteps $\tau = 2.5 \cdot 10^{-4}$, $2.5 \cdot 10^{-5}$, and $2.5 \cdot 10^{-6}$). During the evolution, slight differences in the shapes can be observed. The stationary state shows very good agreement with the expected spherical shape (Figure 12C), with only slight tangential differences (surface tension works in normal direction only), for larger time steps. The simulation with the largest time step required only 80 time steps to reach the stationary state. The necessary compute time of less than 1 minute on a single core CPU (Intel Haswell, 2.50 GHz) illustrates the efficiency of the proposed method.

In the bending stiffness dominant case (case 2), the shape differences are also comparatively small. Time step sizes $10^{-5}$, $10^{-6}$, and $10^{-7}$ have been tested. Figure 12D-F shows the time evolution of the cross section shape. The agreement of point coordinates is good enough to have no visible difference between the different time step sizes. This is also observable when analyzing the membrane energy (Figure 12J,K)

$$E_{\text{membrane}} = 2\pi \int_{\Gamma} (E_{\text{tension}} + E_{\text{bend}} + E_{\text{stretch}}) dA. \quad (31)$$

Figure 12J shows $E_{\text{membrane}}$ for the bending stiffness dominant case for all tested time step sizes. A close up view of the increase at $t = 0.001$ is shown in Figure 12K, for the largest and smallest time steps. Additionally, the time step $\tau = 1.2 \cdot 10^{-5}$ is included. This value was the largest time step size, where stable simulations were possible, as the explicit coupling between flow and membrane elasticity introduces a numerical stiffness. The membrane energy in the case of $\tau = 1.2 \cdot 10^{-5}$ oscillates for $t < 0.0015$. These oscillations smooth out when the energy dissipates, leading to the exact same behavior as for the smallest time step. Consequently, even with large time steps, the membrane energy dissipates in the same manner as with small time steps and the resulting shapes show nearly no differences. However, the coupling of bending stiffness and surface elasticity makes the membrane movement more subtle such that it takes relatively long to reach the stationary state. With the largest time step, 2000 time steps were required. The total compute time in this case was $\approx 30$ minutes.

The stretching dominant case has been tested with time steps $\tau = 4 \cdot 10^{-5}$, $4 \cdot 10^{-6}$, $4 \cdot 10^{-7}$. Images are shown in Figure 12G-I. For the bending stiffness dominant case, the membrane energy dissipates in the same manner for small and large time steps and there are only very small shape differences between the shapes for the respective time step sizes. The simulation required 250 time steps to reach the stationary state. The total compute time in this case was $\approx 3$ minutes on a single core CPU.

A convergence study is done also for the time step analysis. The results can be seen in Table 1 and show the expected first order convergence. A similar order of convergence is reached for the fluid state variables ($v, p$) using the $L_2$-norm of
FIGURE 12  Evolution of the shell cross section shapes for, A-C, the surface tension stiffness dominant case, D-F, the bending stiffness dominant case, and, G-I, the stretching dominant case with different time step sizes. Membrane energy over time is shown for the bending stiffness dominant case (J), with a close-up on the local peak (K) [Colour figure can be viewed at wileyonlinelibrary.com]

the velocity and pressure differences in \( \Omega \). For example, in the surface tension dominant case, we obtain EOC 1.03 for the velocity \((E_{\tau_1}^v = 7.1588 \cdot 10^{-5}, E_{\tau_2}^v = 6.6811 \cdot 10^{-6})\) and 1.22 for pressure \((E_{p_1}^v = 1.1276 \cdot 10^0, E_{p_2}^v = 6.8318 \cdot 10^{-2})\).

5.4  Timestep stabilization of the stretching force

The explicit coupling of fluid flow equations and interface movement leads in general to time step restrictions. For two-phase flow, it is well known that this coupling introduces a stiffness to the system, which results in a time step restriction proportional to the surface tension coefficient. In this case, linear predictions of the curvature term are typically used to relax this restriction, see Reference 51 for a detailed discussion and Reference 52 for an example of an ALE method. Also, for the bending force, the stiffness can be overcome by implicit and semi-implicit schemes. 53 The idea is to include the interface movement monolithically in the flow solver to evaluate the surface forces implicitly at the newly computed time step. This approach has been shown to work extremely well in the immersed boundary method and can in principle be combined with our present ALE method.

However, while this methodology has been proposed for surface tension and bending forces, we are not aware of a similar method for the in-plane elastic stretching force. Accordingly, in this section, we present a novel approach to monolithically couple the in-plane stretching force to the flow.

The crucial issue in the stretching force (Equation (14)) is the implicit treatment of the principal stretches \( \lambda_1 \) and \( \lambda_2 \). A prediction for the values of the principal stretches in the upcoming time step can be obtained from the following...
evolution equations (see appendix of Reference 9):

\[ \frac{\partial}{\partial t} \lambda_1^n - \lambda_1 \nabla_{\Gamma} \cdot \mathbf{v} = 0, \quad \frac{\partial}{\partial t} \lambda_2^n - \frac{\lambda_2}{R} \frac{\mathbf{v}}{R} = 0 \quad \text{on } \Gamma, \quad (32) \]

where \( \nabla_{\Gamma} \) is the (non-axisymmetric) surface divergence in the two-dimensional domain. Accordingly, we can approximate the new values of the principle stretches by

\[ \lambda_1^n = \lambda_1^{n-1} + \tau \lambda_1^{n-1} \nabla_{\Gamma} \cdot \mathbf{v}, \quad \lambda_2^n = \lambda_2^{n-1} + \tau \lambda_2^{n-1} \frac{\mathbf{v}}{R} = 0 \quad \text{on } \Gamma, \quad (33) \]

where the old solution \( \lambda_i^{n-1} \) is calculated, as before, from the current point coordinates (see Equation (18)) on \( \Gamma \).

These equations can now be solved together with the Navier-Stokes equation in one system. To complete the monolithic coupling, the term \( \left( \frac{\partial E_{\text{stretch}}}{\partial t} \right)_n \) in Equation (23) is replaced by

\[ \left( \frac{E_{\text{stretch}}}{\partial t} \right)_n := \left( \kappa^{n-1} n^{n-1} - \nabla_{\Gamma} \right) \left[ (K_A + K_S) (\lambda_1^n - 1) + (K_A - K_S) (\lambda_2^n - 1) \right] - \frac{2K_S}{R} (\lambda_1^n - \lambda_2^n) \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (34) \]

which includes the implicitly calculated values of the principal stretches.

The capability of this implicit strategy has been tested with the parameters of the stretching dominant case. Figure 13 shows the grid points of the membrane for the three different meshes with sizes \( h_1, h_2, \) and \( h_3 \) and different time step sizes. The time step sizes have been chosen to be maximal in each case, that is, such that the usage of a larger time step would lead to crashing simulations due to oscillations on the membrane points.

As shown in Figure 13, the implicit approach permits an enlargement of the time step size by a factor of 8-10. Also, note that the maximum time step is roughly proportional to the grid size, for both, implicit and explicit approaches. Only for the coarsest mesh (Figure 13A), differences between the explicit and implicit approach are visible, which can be attributed to the large time step sizes. Hence, in particular for higher mesh resolution at the membrane, the implicit approach can lead to high quality results with up to 10 times faster simulations.

5.5 Uniaxial compression of a biological cell

As mentioned in Section 1, the ALE method presented in this work offers the opportunity to restrict the simulation to only one of the fluid phases. This capability, which is not present in many other methods (eg, immersed boundary methods), can be very useful if the viscosity ratio between the two fluid phases is large such that the phase with the smaller viscosity has little influence on the results and can be neglected. This does not only speed up the simulations but can also greatly simplify the coupling of the elastic surface to exterior forces, as, for example, in a contact problem.

We illustrate this in the following, as we consider a fluid-filled elastic shell under external compression. The application behind this test case is the uniaxial compression of a biological cell during an atomic force microscopy (AFM) experiment, which is used to measure cell mechanical properties. The viscosity of the intracellular fluid is typically several orders of magnitude larger than the viscosity of the surrounding medium. A detailed description of the problem can be found in our work.49
Consider a rounded biological cell confined between two parallel plates. During the experiment, the upper plate moves with a prescribed velocity $v_{\text{compress}}$ until the initial plate distance $h(t) = h_0$ is decreased over time to $h(t) = h_0 - \Delta h$, see Figure 14A. The necessary force $F$ is constantly measured during compression. Matching experimental data with simulations can be used to extract the surface properties (i.e., $K_A$ and $K_S$) of the cell’s elastic shell (the actin cortex). The experimental setup along with the simulation domain is illustrated in Figure 14A,B.

In the simulations, the interior of the cell is denoted by $\Omega_1$, which is bounded by the elastic cell cortex $\Gamma$ and the symmetry axis. $\Gamma$ is subdivided into the area touching the plates $\Gamma_p$ and the free surface area $\Gamma_f$. During compression, a part of the free surface will touch the plate, accordingly $\Gamma_p$ and $\Gamma_f$ are time-dependent:

$$\Gamma_p(t) = \{ x = (x, r) \in \Gamma : x = 0 \lor x = h(t) \}, \quad \Gamma_f(t) = \Gamma / \Gamma_p(t).$$

A simple contact algorithm is implemented in the numerical simulation: Surface grid points are initially marked to belong to either $\Gamma_p$ or $\Gamma_f$, as soon as a grid point of $\Gamma_f$ touches the upper plate ($x \geq h(t)$), it is shifted to $\Gamma_p$. The interface curve of $\Gamma$ for the initial meshes with plate distance $h = h_0$ is given by a minimal surface calculated according to equations described in Reference 13.

The surrounding medium is neglected to avoid the complicated numerical handling of it being squeezed out of the contact region. Accordingly, the Navier-Stokes equations (3)-(4) are only solved in $\Omega_1$. The corresponding boundary conditions, Equation (5), have to be adapted

$$(-p_1 I + \eta_1 (\nabla v + (\nabla v)^T)) \cdot n = \frac{\partial E}{\partial t} \quad \text{on } \Gamma_f$$

$$t \cdot (-p_1 I + \eta_1 (\nabla v + (\nabla v)^T)) \cdot n = t \cdot \frac{\partial E}{\partial t} \quad \text{on } \Gamma_p$$

$\nu_x = \delta_{c>0} v_{\text{compress}}, \quad \text{on } \Gamma_p,$

where $t$ is the tangential vector to $\Gamma_p$. 

**Figure 14** A, Experimental setup of a biological cell under uniaxial compression. B, The simulation domain. The plates here are on the top and bottom sides of the cell with the boundary $\Gamma_p$ (red dotted line). The free part of the cell boundary is $\Gamma_f$ (blue line). C, Streamlines during the compression process, colored by $|v_r|$. D and E, Compressed cell colored by the values of the principal stretches $\lambda_1$ (D) and $\lambda_2$ (E) together with tangential lines in the corresponding stretch directions.
Numerical studies have been conducted in the realistic physical parameter regime $h_0 \in [6 \, \mu m, 16 \, \mu m], \gamma \in [0.5 \, mN/m, 3 \, mN/m], K_4 = 25 \, nN/\mu m, K_5 \in [8 \, nN/\mu m, 25 \, nN/\mu m], K_6 \in [0.11 \, nN/\mu m, 0.17 \, nN/\mu m], \eta_1 = 1 \, Pa\cdot s$.

Figure 14C shows exemplary streamlines during compression, as fluid is driven toward the free boundary $\Gamma_f$, extending the cell’s radius. Detailed numerical results can be found in Reference 49.

As shown in Reference 49, the simulations reproduce the characteristic force response of biological cells. Matching simulations with experiments can provide new insights in cell mechanical properties. To this end, an extremely fine grid resolution with approximately 2000 grid points along the surface contour is necessary to disentangle contributions from surface tension and surface elasticity (see Reference 49). This fine grid resolution makes the problem unfeasible to full 3D simulations and underlines the relevance of the proposed method.

5.6 Shape oscillations of novel microswimming shells

Finally, we present the first numerical simulations of this process, which can help to gain a better understanding of the buckling dynamics and the influence of parameters on the swimming efficiency. The sudden release of stored elastic energy during the buckling of the shell leads to very high flow rates such that the hydrodynamics are significantly influenced by inertial forces. Hence, even at the microscale, the full Navier-Stokes equations are necessary to describe the process.\(^{11}\) The simulation of the inner phase $\Omega_i$ is not necessary due to the low density and viscosity of air. Instead, we assume a homogeneous air pressure $p_1$ inside and use adiabatic gas theory, to relate this pressure to the inner shell volume $V$ by $p_1 = p_{1,0} (V_0/V)^{1.4}$, where $p_{1,0}$ and $V_0$ denote the respective initial values. Accordingly, the stress exerted by the air is reduced to $p_1 \, \mathbf{n}$ whereupon the stress boundary condition (5) becomes

\[
( -p_0 \mathbf{I} + \eta_0 \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right) ) \cdot \mathbf{n} = -p_{1,0} \left( V_0/V \right)^{1.4} \mathbf{n} - \frac{\partial E}{\partial T}
\]

The weak spot is introduced by slightly decreasing the elastic surface moduli locally around the membrane point touching the symmetry axis ($R = 0$) on the right. Note that the numerical results are independent of the exact amount of this reduction, as the weak spot only serves as a locator for the occurring buckling instability. Numerical parameters are as in Reference 11 where a macroscopic shell diameter of $\approx 5 \, cm$ and increased viscosity (glycerol $\eta = 1 \, Pa$ or oil $\eta = 37.5 \, Pa$) was used to mimic flow conditions at the microscale. The initial air pressure is $p_{1,0} = 1 \, bar$.

During the simulation, an external pressure difference is imposed using a sinusoidal function (oscillating between $p_0 = 1.0 \, bar$ and $p_{0} = 1.77 \, bar$) at the outer boundaries of the computational domain. The change in outer pressure leads to periodic buckling and debuckling of the microswimmer shell. Images of different stages of a buckling cycle are shown in Figure 15C-F together with the surrounding fluid velocities. Increasing pressure in the beginning leads to uniform shrinkage (Figure 15C) and buckling (Figure 15D,E). The concomitant complex flow patterns induce a thrust of the shell, propelling it to the right. Decreasing pressure leads to inflation and debuckling and a further propulsion to the right, see Figure 15F. Shape evolution and flow patterns are comparable to the experiments in Reference 11. Note that the large out-of-plane deformations during buckling only involve small in-plane stretching of the surface, which justifies to use the linear elasticity model for the in-plane stretching presented here.

The rich dynamics during this swimming process require a great number of time steps to be resolved. This is illustrated in Figure 15B, which shows the position and velocity of the microswimmer shell during the first 0.25 seconds.
FIGURE 15 A, Experimental setup of the shell under pressure oscillations, adapted from. 54 B, Position and velocity of the shell’s center of mass in x-direction during the first 0.25 second of acceleration. The frequency of the outer pressure oscillations is 75 Hz. C-F, Snapshots of the buckling shape dynamics during one cycle in the simulations, together with the velocity field [Colour figure can be viewed at wileyonlinelibrary.com]

elastic surface oscillations occur due to the high eigenfrequency of the swimmer (see Figure 15B, red curve). To resolve these oscillations that are accompanied by complex flow patterns, we use a time steps size of \( \tau = 2.5 \mu \text{s} \). Accordingly, \( 10^5 \) time steps are needed to accurately resolve the acceleration process for the given parameters. For other excitation frequencies and amplitudes, the acceleration process may even take significantly longer as a more complex interplay of excitation frequency and eigenfrequency develops. The corresponding number of necessary time steps makes full 3D simulations illusive. The axisymmetric immersed boundary method in Reference 24 cannot cope with different physical processes inside and outside the shell (compressible/incompressible fluid) and axisymmetric boundary element methods are restricted to the Stokes regime. Hence the method proposed here is, to our knowledge, the only available method to simulate the microscopic swimming in reasonable times.

Our simulations shall be used in the future in collaboration with the authors of Reference 11 to gain understanding of the complex dynamical coupling of surface shape deformations, pressure oscillations, and shell propulsion to create efficient novel microswimmers.

6 | CONCLUSION

In this article, we have presented a novel ALE method to simulate axisymmetric elastic surfaces immersed in Navier-Stokes fluids. As inherent to ALE methods, the grid is matched to the elastic material, which can, therefore, be resolved with relatively few grid points. The axisymmetric setting reduces the system effectively to a two-dimensional
problem. Elastic surface forces are discretized with surface finite-differences and coupled to evolving FE s of the bulk problems. An implicit coupling strategy reduces time step restrictions induced by the stiffness of the stretching elasticity.

The method combines high accuracy with computational efficiency, which is confirmed in several numerical test cases dominated either by surface tension, bending stiffness or in-plane elasticity. In all these cases, we find that numerical errors are relatively small even for coarse grids and converge with order of 1-2 with respect to grid size and time step size. The computational times of the test problems are on the order of minutes on a single core CPU. While such a high computational speed is not required in most typical applications, it does enable otherwise unfeasible simulations as soon as the problems involve a high number of grid points or rich dynamics demanding many time steps.

As examples, we present first simulations of the observed shape oscillations of novel microswimming shells and the uniaxial compression of biological cells filled with cytoplasm. In collaboration with (bio)physicists, the method is currently used to get more insight into the dynamics of microswimming shells and the cellular cortex.

ACKNOWLEDGEMENTS
The authors acknowledge support from the German Research Foundation DFG (grants AL1705/3, AL1705/5) and the DFG Research Unit FOR 3013. Simulations were performed at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden.

ORCID
Sebastian Aland https://orcid.org/0000-0002-5258-0006

REFERENCES
1. Aland S, Egerer S, Lowengrub J, Voigt A. Diffuse interface models of locally inextensible vesicles in a viscous fluid. *J Comput Phys*. 2014;277:32-47.
2. Baumgart T, Hammond AT, Sengupta P, et al. Large-scale fluid/fluid phase separation of proteins and lipids in giant plasma membrane vesicles. *Proc Nat Acad Sci*. 2007;104(9):3165-3170.
3. Noguchi H, Gompper G. Fluid vesicles with viscous membranes in shear flow. *Phys Rev Lett*. 2004;93:258102.
4. Kraus M, Wintz W, Seifert U, Lipowsky R. Fluid vesicles in shear flow. *Phys Rev Lett*. 1996;77:3685-3688.
5. Marth W, Aland S, Voigt A. Margination of white blood cells - a computational approach by a hydrodynamic phase field model. *J Fluid Mech*. 2016;790:389-406.
6. Ye T, Phan-Thien N, Khoo BC, Lim CT. A file of red blood cells in tube flow: a three-dimensional numerical study. *J Appl Phys*. 2014;116(12):124703.
7. Vlahovska PM, Barthes-Biesel D, Misbah C. Flow dynamics of red blood cells and their biomimetic counterparts. *Comptes Rendus Physique*. 2013;14(6):451-458. Living Fluids / Fluids Vivants.
8. Dupire J, Socol M, Villat J. Full dynamics of a red blood cell in shear flow. *Proc Nat Acad Sci*. 2012;109(51):20808-20813.
9. Mokbel M, Mokbel D, Mietke A, et al. Numerical simulation of real-time deformability cytometry to extract cell mechanical properties. *ACS Biomat Sci Eng*. 2017;3(11):2962-2973.
10. Mietke A, Lülicher F, Salazarini IF. Self-organized shape dynamics of active surfaces. *Proc Nat Acad Sci*. 2019;116(1):29-34.
11. Djellouli A, Marmottant P, Djeridi H, Quilliet C, Coupiet G. Buckling instability causes inertial thrust for spherical swimmers at all scales. *Phys Rev Lett*. 2017;119:224501.
12. Givoli D. Finite element modeling of thin layers. *CMES Comput Model Eng Sci*. 2004;5:497-514.
13. Fischer-Friedrich E, Hyman AA, Lülicher F, Müller DJ, Helenius J. Quantification of surface tension and internal pressure generated by single mitotic cells. *Sci Rep*. 2014;4:6213.
14. Pozrikidis C. Numerical simulation of the flow-induced deformation of red blood cells. *Ann Biomed Eng*. 2003;31(10):1194-1205.
15. Veerapaneni SK, Gueyffier D, Zorin D, Biros G. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2d. *J Comput Phys*. 2009;228(7):2334-2353.
16. Veerapaneni SK, Gueyffier D, Biros G, Zorin D. A numerical method for simulating the dynamics of 3d axisymmetric vesicles suspended in viscous flows. *Journal of Computational Physics*. 2009;228(19):7233-7249.
17. Boedec G, Leonetti M, Jaeger M. 3d vesicle dynamics simulations with a linearly triangulated surface. *J Comput Phys*. 2011;230(4):1020-1034.
18. Veerapaneni SK, Rahimian A, Biros G, Zorin D. A fast algorithm for simulating vesicle flows in three dimensions. *J Comput Phys*. 2011;230(14):5610-5634.
19. Zhao H, Shaqueh E. The dynamics of a vesicle in simple shear flow. *J Fluid Mech*. 2011;674:578-604.
20. Noguchi H, Gompper G. Swinging and tumbling of fluid vesicles in shear flow. *Phys Rev Lett*. 2007;98:128103.
21. Paulose J, Vliegenthart GA, Gompper G, Nelson DR. Fluctuating shells under pressure. *Proc Nat Acad Sci*. 2012;109(48):19551-19556.
22. Winkler RG, Fedosov DA, Gompper G. Dynamical and rheological properties of soft colloid suspensions. *Curr Opinion Colloid Interface Sci*. 2014;19(6):594-610.
23. Le D, White J, Peraire J, Lim K, Khoo B. An implicit immersed boundary method for three-dimensional fluid-membrane interactions. *J Comput Phys.* 2009;228(22):8427-8445.

24. Hu W-F, Kim Y, Lai M-C. An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in Navier-Stokes flows. *J Comput Phys.* 2014;257(Part A):670-686.

25. Givelberg E. Modeling elastic shells immersed in fluid. Communications on Pure and Applied Mathematics 57.

26. Fai TG, Griffith BE, Mori Y, Peskin CS. Immersed boundary method for variable viscosity and variable density problems using fast constant-coefficient linear solvers I: numerical method and results. *SIAM J Sci Comput.* 2013;35(5):B1132-B1161.

27. Peskin CS. The immersed boundary method. *Acta Numer.* 2002;11:479-517.

28. Han Z, Stoter SK, Wu C-T, et al. Consistent discretization of higher-order interface models for thin layers and elastic material surfaces, enabled by isogeometric cut-cell methods. *Comput Methods Appl Mech Eng.* 2019;350:245-267.

29. Maitre E, Milcent T, Cottet G-H, Raoult A, Usson Y. Applications of level set methods in computational biophysics. *Mathematical and Computer Modelling.* 2009;49(11):2161-2169. Trends in Application of Mathematics to Medicine.

30. Salac D, Miksis M. A level set projection model of lipid vesicles in general flows. *J Comput Phys.* 2011;230(22):8192-8215.

31. Laadhari A, Saramito P, Misbah C. Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods. *J Comput Phys.* 2014;263:328-352.

32. Maitre E, Misbah C, Peyla P, Raoul T. Comparison between advected-field and level-set methods in the study of vesicle dynamics. *Phys D Nonlin Phenomena.* 2012;241(13):1146-1157.

33. Doeyeu V, Chabannes V, Prud’Homme C, Ismail M. Simulation of vesicle using level set method solved by high order finite element. *EAIIM Proc.* 2012;38:335-347.

34. Aland S, Voigt A. Benchmark computations of diffuse interface models for two-dimensional bubble dynamics. *Int J Numer Methods Fluids.* 2012;69(3):747-761.

35. Lowengrub J, Allard J, Aland S. Numerical simulation of endocytosis: viscous flow driven by membranes with non-uniformly distributed curvature-inducing molecules. *J Comput Phys.* 2016;309:112-128.

36. Aland S. Phase field models for two-phase flow with surfactants and biomembranes. In: Bothe D, Reusken A, eds. *Transport Processes at Fluidic Interfaces.* Cham: Springer International Publishing; 2017:271-290.

37. Cottet G-H, Maitre E. A level set method for fluid-structure interactions with immersed surfaces. *Math Models Methods Appl Sci.* 2006;16:415-438.

38. Mokbel D, Habib S, Aland S. A phase-field model for fluid-structure interaction. *J Comput Phys.* 2018;372:823-840.

39. Degroote J, Haelterman R, Annerel S, Bruggeman P, Vierendeels J. Performance of partitioned procedures in fluid-structure interaction. *Comput Struct.* 2010;88(7-8):446-457.

40. van Brummelen EH. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *J Appl Mech.* 2009;76(2):021206.

41. Souli M, Ouahsine A, Lewin L. Ale formulation for fluid-structure interaction problems. *Comput Methods Appl Mech Eng.* 2000;190(5):659-675.

42. Sawada T, Hisada T. Fluid–structure interaction analysis of the two-dimensional flag-in-wind problem by an interface-tracking ale finite element method. *Comput Fluids.* 2007;36(1):136-146.

43. Farhat C, Rallu A, Wang K, Belytschko T. Robust and provably second-order explicit–explicit and implicit–explicit staggered time-integrators for highly non-linear compressible fluid–structure interaction problems. *Int J Numer Methods Eng.* 2010;84(1):73-107.

44. Zhang Q, Hisada T. Analysis of fluid–structure interaction problems with structural buckling and large domain changes by ale finite element method. *Comput Methods Appl Mech Eng.* 2001;190(48):6341-6357.

45. Bletzinger K-U, Bischoff M, Ramm E. A unified approach for shear-locking-free triangular and rectangular shell finite elements. *Comput Struct.* 2000;75(3):321-334.

46. Trozzo R, Boedec G, Leonetti M, Jaeger M. Axisymmetric boundary element method for vesicles in a capillary. *J Comput Phys.* 2015;289:62-82.

47. Li J, Hesse M, Ziegler J, Woods AW. An arbitrary lagrangian eulerian method for moving-boundary problems and its application to jumping over water. *J Comput Phys.* 2005;208(1):289-314.

48. Aland S, Boden S, Hahn A, Klingbeil F, Weismann M, Weller S. Quantitative comparison of taylor flow simulations based on sharp-interface and diffuse-interface models. *Int J Numer Methods Fluids.* 2013;73(4):344-361.

49. Mokbel D, Hosseini K, Aland S, Fischer-Friedrich E. The poison ratio of the cellular actin cortex is frequency-dependent. *Biophys J.* 2020. https://doi.org/10.1016/j.bpj.2020.03.002

50. Teschner M, Heidelberger B, Müller M, Gross M. A versatile and robust model for geometrically complex deformable solids. *Comput Graph Int Conf.* 2004;312-319.

51. Aland S. Time integration for diffuse interface models for two-phase flow. *J Comput Phys.* 2014;262:58-71.

52. Bänsch E, Weller S. A comparison of several time discretization methods for free surface flows. *Proc ALGORITMY.* 2012;2012:331-341.

53. Lai M-C, Ong KC. Unconditionally energy stable schemes for the inextensible interface problem with bending. *SIAM J Sci Comput.* 2019;41(4):B649-B668.

54. Djellouli A. Swimming through spherical shell buckling (Dissertation). Universitécopy; Grenoble-Alpes; 2017.

55. Vey S, Voigt A. AMDIS: adaptive multidimensional simulations. *Comput Visual Sci.* 2006;10(1):57-67.

56. Witkowski T, Ling S, Praetorius S, Voigt A. Software concepts and numerical algorithms for a scalable adaptive parallel finite element method. *Adv Comput Math.* 2015;41(6):1145-1177.
SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Mokbel M, Aland S. An ALE method for simulations of axisymmetric elastic surfaces in flow. *Int J Numer Meth Fluids*. 2020;92:1604–1625. https://doi.org/10.1002/fld.4841

APPENDIX A.

Implementation details I—mesh and boundary conditions

The presented method is implemented in the FE toolbox AMDiS.\(^{55,56}\) While AMDiS does support Taylor-Hood (P2/P1) elements, it does not support different solution variables being defined on different meshes, as necessary for the pressure \(p_i\) here. Accordingly, we use some technical workarounds to implement the proposed method, described in the following.

A single mesh that contains both domains \(\Omega_0, \Omega_1\) as separate disconnected parts is constructed in gmsh.\(^{57}\) To easily distinguish between internal and external fluid, the internal part is translated below the symmetry axis \((r < 0)\). Figure A1 shows the mesh in the initial state. At the beginning of the simulation an indicator function is created discriminating between the two mesh parts based on the \(r\)-value (grid points with negative \(r\) belong to the internal fluid). Immediately afterward, the internal fluid mesh is shifted upward to obtain a spatially matched grid, which is yet unconnected at the interface.

The Navier-Stokes equations are assembled on both mesh partitions separately using Lagrangian P2/P1 elements. The implementation of interfacial stress balance equation (5) and velocity continuity are technically realized as follows. The discrete membrane \(\Gamma_h = \Gamma_{h,0} \cup \Gamma_{h,1}\) is composed of the membrane boundaries of the mesh for the external and internal fluid, respectively. Due to the disconnected grid, we can only enforce one-sided stress conditions, which we prescribe as follows

\[
(-p_0 \mathbf{I} + \eta_0 \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right)) \cdot \mathbf{n} = -\frac{\partial E}{\partial \Gamma} \quad \text{on} \quad \Gamma_{h,0}
\]

\[
-(-p_1 \mathbf{I} + \eta_1 \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right)) \cdot \mathbf{n} = 0 \quad \text{on} \quad \Gamma_{h,1}.
\]

Define \(n\) as the number of \((P2-)\)DOFs on the membrane. Let \(j = 0, \ldots, n - 1\), be an ordered (eg, counterclockwise) numbering of the DOFs on \(\Gamma_{h,0}\) and \(\Gamma_{h,1}\), such that the numbers of both boundaries are equal.

At the coordinate of each membrane DOF \(j\) there exist exactly two P2-test-functions \(\psi_0^j, \psi_1^j\) which are non-zero, with \(\text{supp}(\psi_0^j) \subset \Omega_0\) and \(\text{supp}(\psi_1^j) \subset \Omega_1\). These test-functions are each used twice, once for testing with the \(v_x\)-equation and once for testing with the \(v_r\)-equation. Each of these test cases corresponds to a single row in the assembled Finite-Element matrix. The correct stress jump condition can be realized by adding the two rows related to \(\psi_1^j\) to the two respective

![FIGURE A1](https://wileyonlinelibrary.com)  
**FIGURE A1** Mesh as it is imported by the simulation  
[Colour figure can be viewed at wileyonlinelibrary.com]
FIGURE A2 1D example for the test-functions across $\Gamma$. The grid coordinates are denoted with $x'$, where the membrane coordinate has the superscript 0. Since it exists for both meshes, there is $x'^0_0$ for $T_{h,0}$ and $x'^0_1$ for $T_{h,1}$ at the same position. The red colour refers to $T_{h,0}$, the black colour refers to $T_{h,1}$. $\psi^0_j$ and $\psi^1_j$ are the test-functions at the membrane coordinate $x'^0_j$ and $x'^1_j$, respectively. Adding up both one-sided test-functions by addition of corresponding matrix lines effectively mimics that the momentum equations were tested with a single test-function of an interfacially connected grid [Colour figure can be viewed at wileyonlinelibrary.com]

rows related to $\psi^j_0$, for every $j = 0, \ldots, n - 1$. This adds both one-sided test-functions and effectively mimics that the momentum equations were tested with a single test-function of an interfacially connected grid. Also the boundary condition equations (A1) and (A2) are effectively added up to recover the correct stress jump condition equation (5). As an illustration, Figure A2 shows the test-functions in the case of a 1D mesh.

The obsolete rows related to $\psi^j_1$ can then be overwritten to enforce the continuity of the velocities at the interface

$$v^j_1 - v^0_1 = 0, \quad \text{on } \Gamma_{h,1}. \quad (A3)$$

As $v = (v_x, v_r)$, the above describes two continuity conditions that can be assembled in the $v_x$- and $v_r$-rows tested with $\psi^j_1$. The continuity of velocity ensures, in particular, that corresponding DOFs on $\Gamma_{h,0}$ and $\Gamma_{h,1}$ share the same coordinate points for all times.

Implementation details II—extension of surface equations to $\Omega$

In AMDiS, it is not possible to solve 2D bulk equations and 1D surface equations all in one system. To compute the principal stretches implicitly, it is, therefore, necessary to extend Equation (32) to $\Omega$. The equations for the principal stretches then read

$$\partial_t \lambda_1 - \lambda_1 \nabla \cdot \mathbf{v}_{\Gamma, \text{ext}} = 0, \quad \partial_t \lambda_2 - \frac{(\mathbf{v}_{\Gamma, \text{ext}})_r}{R_{\text{ext}}} = 0 \quad \text{in } \Omega, \quad (A4)$$

where $\mathbf{v}_{\Gamma, \text{ext}}$ is the extension of the interfacial velocity to $\Omega$ constant in normal direction. $R_{\text{ext}}$ is the distance to the symmetry axis for any point in $\Omega$. The equations for the principal stretches are solved using $P1$ elements. The calculation of the normal extension is based on a Hopf-Lax algorithm described in Reference 58. The surface divergence then reads

$$\nabla \cdot \mathbf{v}_{\Gamma, \text{ext}} = \mathbf{P} : \mathbf{v}_{\Gamma, \text{ext}}, \quad (A5)$$

with the projection onto the tangent space $\mathbf{P} = \mathbf{I} - \mathbf{n n}^T$. 