membranes

Article

A Lagrangian Thin-Shell Finite Element Method for Interacting Particles on Fluid Membranes

Sanjay Dharmavaram [1,*], Xinran Wan [2] and Luigi E. Perotti [3]

[1] Department of Mathematics, Bucknell University, 1 Dent Drive, Lewisburg, PA 17837, USA
[2] Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[3] Mechanical and Aerospace Engineering Department, University of Central Florida, 12760 Pegasus Drive, Orlando, FL 32816, USA
* Correspondence: sd045@bucknell.edu

Abstract: A recurring motif in soft matter and biophysics is modeling the mechanics of interacting particles on fluid membranes. One of the main outstanding challenges in these applications is the need to model the strong coupling between the substrate deformation and the particles' positions as the latter freely move on the former. This work presents a thin-shell finite element formulation based on subdivision surfaces to compute equilibrium configurations of a thin fluid shell with embedded particles. We use a variational Lagrangian framework to couple the mechanics of the particles and the substrate without having to resort to ad hoc constraints to anchor the particles to the surface. Unlike established methods for such systems, the particles are allowed to move between elements of the finite element mesh. This is achieved by parametrizing the particle locations on the reference configuration. Using the Helfrich–Canham energy as a model for fluid shells, we present the finite element method's implementation and an efficient search algorithm required to locate particles on the reference mesh. Several analyses with varying numbers of particles are finally presented reproducing symmetries observed in the classic Thomson problem and showcasing the coupling between interacting particles and deformable membranes.

Keywords: interacting particles; lipid membranes; subdivision finite element; Helfrich–Canham model; model for protein–membrane interaction

check for updates

Citation: Dharmavaram, S.; Wan, X.; Perotti, L.E. A Lagrangian Thin-Shell Finite Element Method for Interacting Particles on Fluid Membranes. *Membranes* **2022**, *12*, 960. https://doi.org/10.3390/membranes12100960

Academic Editor: Hsin-Hui Shen

Received: 25 August 2022
Accepted: 26 September 2022
Published: 30 September 2022

1. Introduction

A two-dimensional interacting assembly of particles embedded, yet free to move, on a deformable fluid membrane is a recurring motif in material science, soft-matter, and biophysical systems. This type of assembly serves as a model to understand diverse phenomena, such as protein packing on viral capsids [1], cellular processes such as endo-/exo-cytosis [2], and crystallization on curved geometries [3–6]. In these applications, the two-dimensional (2D) nature of the structure is often critical to the application function. Ever since the discovery of graphene, it has been well known that 2D materials have unique electronic, mechanical, and optical properties, which distinguish them from their bulk 3D counterparts [7]. Consequently, in recent decades there has been significant interest in synthesizing and characterizing novel 2D materials with tailorable properties. Colloidal particles, owing to their versatility and tunability, are playing a significant role in this regard [8–10].

In addition to the practical applications noted above, 2D particle assemblies on thin substrates are also adopted to probe fundamental questions on the structure of matter. Numerous, particle-based experimental studies have focused on understanding crystal structures and defects on curved geometries [11–17]. Curvature and geometrical frustration induce crystalline states not found in flat geometries. For instance, while on flat infinite substrates, particles under isotropic repulsive interactions arrange themselves in a regular hexagonal lattice, on curved substrates (especially for a sufficiently large number of particles) the arrangements invariably contain crystal defects [11,18].

Despite growing interest and advances in numerical methods to model colloids on two-dimensional surfaces, most studies have focused on rigid geometries, i.e., the substrate cannot deform in response to the forces exerted by the particles. Some studies do consider the adhesion effects of colloidal particles on the shape of the substrate [19,20], but remain focused on the *local* distortion effects. *Long-range* and *cooperative* interactions between particles and substrates have largely not been explored.

A significant challenge in studying such particle systems using computational models is the coupled interaction between the particles and the deformable substrate. Since the particles are constrained to move on the substrate whose shape is itself determined by particle interactions, the configuration of the particles and the substrate's shape cannot be independently determined. This poses unique computational challenges. Current approaches impose constraints to anchor the particles on the substrate [21] or artificially restrict the particles to lie at the nodes of the mesh used to discretize the substrate [22,23]. Relying on constraints to force particles on the substrate is computationally expensive and cumbersome to implement, while restricting the particles' positions to the nodes of the substrate mesh imposes undue restrictions on the allowed particle/substrate equilibrium configurations. In a recent work [24], we developed an alternative method that is efficient and does not introduce approximations or spurious constraints in modeling the particles–substrate interactions. In this *Lagrangian* approach, we parametrize the location of the particles using coordinates mapped on the *reference* configuration. Together with the deformation mapping of the substrate, these reference coordinates constitute the system's degrees-of-freedom. To obtain the actual particles' locations on the deformed substrate, we compose the reference coordinates with the substrate's deformation map. Thus, constraints are not required to anchor the particles to the surface. We applied this framework to study interacting particles on a spherical fluid shell, employing a spectral Galerkin method and accordingly discretizing the surface deformation map using a spherical harmonic expansion. To circumvent computational issues stemming from the in-plane fluidity of the substrate [25–27], a *radial graph* ansatz [28] was implemented. According to this ansatz, the displacement is parametrized along the radial direction from the center of the reference sphere. This is similar to the Monge representation used for flat membranes [29–31].

Although novel and effective in modeling the particle–substrate coupled interactions correctly, this approach presented two key limitations. The radial graph ansatz implies that only radial displacements from the spherical reference state can be computed. This restricts the equilibrium configurations that can be obtained, precluding severely distorted shapes as the ones appearing during endo-/exo-cytosis of cells and budding of viral capsids. An example of such a state in multi-phase lipid vesicles is shown in Figure 2h of [32]. The second major limitation regards the use of the spectral Galerkin method. As spherical harmonics have to be computed using recursive algorithms, this approach is computationally expensive, especially when high order terms are required to better represent the deformed configuration. In addition, the number of spherical harmonics needed up to order ℓ grows quadratically (i.e., $(\ell+1)^2$), leading to a rapid rise in computational cost as higher accuracy is sought. The global nature of these functions and the coordinate singularities at the sphere's poles also requires very high-order quadrature rules. Finally, due to Gibbs-type oscillations, the method is inadequate when equilibrium configurations present sharp shape changes.

Even though our studies have so far focused on "zero temperature" equilibrium states, where thermal fluctuations do not play a role, the ultimate application of the computational models lies in finite-temperature studies, often simulated using Monte Carlo methods. Indeed, for many biophysical and soft-matter systems at physiologically relevant temperatures, thermal fluctuations play an important role. However, applying the spectral Galerkin method in Monte Carlo simulations is not straightforward. Monte

Carlo simulations typically perturb the system's degrees-of-freedom with "moves" that must satisfy the *detailed balance* condition. In spectral methods, the degrees-of-freedom are the mode coefficients that do not have a clear physical interpretation and enforcing *detailed balance* is not straightforward. A finite element approach with nodal displacement degrees of freedom, on the other hand, is amenable to standard Monte Carlo approaches, and can, therefore, be easily adapted to simulate finite-temperature systems.

In this article, we extend our previous work and remove the limitations described above. We adapt the Lagrangian particle formulation to the context of C^1 conforming thin shell finite elements [33] that use interpolation functions based on a Loop subdivision scheme [34,35]. The Loop subdivision finite element scheme requires only displacement degrees of freedom while retaining C^1 continuity of the shape functions across elements. This is necessary to compute the curvatures terms appearing in the thin-shell elastic energy and the mapping of the particles from the reference to the deformed mesh (as described in Sections 2.2 and 2.3, this step may require the evaluation of the deformation gradient tensor at the interface between elements). Furthermore, this finite element method (FEM) is widely used in lipid membrane studies [25–27] and including the Lagrangian particle approach into this framework can help other researchers in the area.

In addition to merging the Lagrangian particle formulation with thin-shell FEM and removing the radial graph ansatz, an essential contribution of the present work are the details of the search algorithm used to locate the particles on the reference mesh. Since, in our Lagrangian approach, the particle positions are parametrized in the reference configuration, the search for the elements to which each particle belongs is carried out only on the reference mesh.

This paper is organized as follows. In Section 2.1, we present the variational formulation of the substrate-particles system. We model the fluid substrate using the Helfrich–Canham energy and the particle interactions using pair-wise isotropic potentials. The in-plane fluidity of the substrate poses computational challenges, which are addressed using the *gauge-fixing procedure* proposed in [27]. In Section 2.2, we present the Lagrangian formulation for the gauge-fixed formulation of the Helfrich–Canham energy. We present the weak form of the equilibrium equations which are then used to discretize the system. We briefly comment on the implementation aspects of the Loop subdivision FEM method relevant to the present problem in Section 2.3. In Section 2.4, the details for the proposed search algorithm are provided. Finally, in Section 3, we discuss the details of a validation study and the results of representative analyses to showcase the presented method.

2. Materials and Methods

This work considers a system consisting of a topologically spherical fluid shell with N mutually interacting particles. The particles are embedded in the substrate but can freely move along its surface. We focus on equilibrium configurations and assume that the particles do not experience in-plane viscous forces. Since the particles cannot leave the surface, the substrate can exert only normal forces on the particles and vice versa. This model system (Figure 1) is motivated, for example, by the study of interacting proteins embedded on lipid membranes. Note that, following studies in the computational biophysics literature, in this work the term "membrane" is often used to refer to a fluid shell, despite the difference in connotations of the two terms in continuum mechanics.

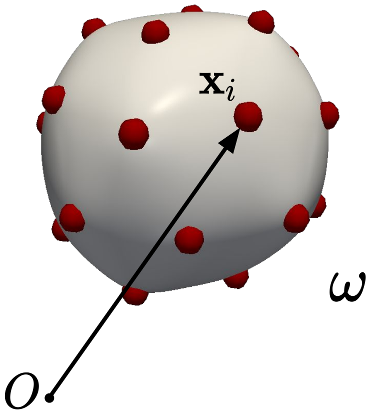

Figure 1. Schematic showing the deformed configuration of the membrane embedded with particles. Particle i has position vector $\mathbf{x}_i$.

2.1. Model Formulation

Without loss of generality, we assume that the substrate's undeformed configuration is a unit sphere (S^2). Let us denote the surface of the deformed configuration as $\omega \subset \mathbb{R}^3$ and $\mathbf{f} : S^2 \to \omega$ as the deformation map of the deformed surface. We model the bending elastic energy $\mathcal{E}_{HC}$ of the substrate using the Helfrich–Canham energy [36]:

$$\mathcal{E}_{HC} = \int_{\omega} \kappa (H - C_0)^2 + \kappa_g K \, da \,, \tag{1}$$

where H and K are the mean and Gaussian curvatures of ω with κ and κ_g their associated bending stiffnesses, and C_0 is the preferred curvature. Helfrich–Canham energy is a widely used model for lipid membranes, and the energy (1) is expressed as an integral on the current configuration ω with da being its area measure. This is in line with the observation that a lipid membrane behaves like a two-dimensional fluid, and, consequently, has no preferred in-plane reference configuration. Since the stretching modulus of a lipid membrane is significantly larger than the bending moduli, the membrane is usually approximated as area-preserving [37]. This incompressibility condition is enforced through the area constraint:

$$\int_{\omega} da = 4\pi \,. \tag{2}$$

We model particle interactions with a pair potential $\Phi(r)$, where r represents the 3D Euclidean distance between the interacting particles. The total interaction energy is thus given by:

$$U(\mathbf{x}_1, \cdots, \mathbf{x}_N) = \sum_{i=1}^{N} \sum_{j>i}^{N} \Phi(||\mathbf{x}_i - \mathbf{x}_j||) \,, \tag{3}$$

where $\mathbf{x}_i$ represents the 3D position of particle i on the membrane and $|| \cdot ||$ denotes the standard Euclidean norm. In this work, we will explore various potentials, including Coulombic, harmonic, and Lennard–Jones interaction potentials (see Section 3).

The (Helmholtz) free energy $\mathcal{E}$ of the system is given by:

$$\mathcal{E} = \mathcal{E}_{HC} + U(\mathbf{x}_1, \cdots, \mathbf{x}_N) - pV \,, \tag{4}$$

where p is the internal osmotic pressure on the substrate and V is the enclosed volume. The area constraint in Equation (2) must be additionally imposed. In the simulations presented in this work, we do so using a penalty method. Alternatively, the area constraint can be enforced using a Lagrange multiplier, which can be numerically implemented using the Augmented Lagrangian method.

Remark 1. *Instead of a fixed pressure applied on the membrane, it is common to constrain the volume enclosed in its interior [25,26]. Under such a formulation, we define a non-dimensional parameter $v := V/V_0$ that measures the fraction of volume enclosed by ω with respect to the volume enclosed by the unit (reference) sphere ($V_0 = 4\pi/3$). Thus, we impose the condition:*

$$\frac{1}{3}\int_{\omega} \mathbf{f}\cdot\mathbf{d}\, da = \frac{4\pi v}{3}\,, \tag{5}$$

where $\mathbf{d}$ *is the outward pointing (unit) normal to surface ω and* $\mathbf{f}$ *is the deformation map. The integral on the left side computes the total volume enclosed by ω, and has been obtained using the divergence theorem: $\int_V div\ \mathbf{f}\ dv = \int_\omega \mathbf{f}\cdot\mathbf{d}\ da$, where $div\ \mathbf{f} = 3$. When required, this volume constrained formulation is employed in the simulations presented in Section 3.*

2.1.1. In-Plane Fluidity

In-plane fluidity, also known as *reparametrization invariance*, is a defining material symmetry for fluid shells. For instance, in lipid membranes it arises because lipid molecules can freely move on the surface facing very little resistance, conferring to the membrane a 2D fluid-like behavior. In the Helfrich–Canham model, fluidity manifests as reparametrization invariance of the energy shown in Equation (1). This can be inferred from the dependence of the energy solely on differential geometrical quantities, which are independent of the chosen parameterization of the surface. Thus, unlike the case of solid shells, in-plane fluidity of fluid membranes implies that the latter does not have a prescribed "reference configuration".

A consequence of in-plane fluidity is that the tangential components of the Euler-Lagrange equations vanish identically [37,38]. Only the normal components of the Euler–Lagrange equations contribute to the equilibrium equations and, thus, the equations are inherently underdetermined. It can be shown that equilibrium configurations, when they exist, belong to an *equivalence class* of solutions [27]. Members of this class are different parametric representations for the *same* surface and are related to each other by a diffeomorphism map between the parameter spaces. Since there are infinitely many diffeomorphisms between spherical surfaces, they give rise to a grossly redundant solution space. This causes significant challenges in numerical simulations where spurious zero-energy shear modes [26] and severe mesh distortions [25] are commonly reported.

Remark 2. *Note that, although the system considered here includes particles in addition to a fluid substrate, the analysis presented above still applies. In-plane fluidity remains the material symmetry for the problem. This is because the particles are free to move on the surface of the membrane and they lack preferred reference positions. This is evidenced by Equation (3), where the particles' interaction energy only depends on the current particles' locations, i.e., $\{\mathbf{x}_i\}$. Thus, reparametrizing the surface will neither change the positions of the particles nor their interaction energy. Therefore, U is reparametrization invariant, and so is $\mathcal{E}$ in Equation (4). As the numerical issues noted in the previous paragraph will also plague the membrane–particle system, in the next section we describe a strategy to circumvent these issues.*

2.1.2. Gauge-Fixing Procedure

Several numerical schemes have been developed to address computational issues stemming from reparametrization invariance. Some methods impose local area incompressibility (instead of global area constraint) [26] to suppress zero-energy modes, while others [25,39] dampen tangential motion by introducing ad hoc in-plane energies whose contributions are iteratively decreased to zero. Monge representation [29] or a *radial graph* ansatz [28,40]—where the surface is parametrized using a single unknown function—have also been used to circumvent reparametrization invariance. In this work, we employ the *gauge-fixing procedure* recently proposed in [27], which is computationally efficient for topologically spherical surfaces and does not require any iterative reduction in ad hoc energy terms that change the physics of the model. In this approach, reparametrization invariance is viewed as a form of *gauge symmetry*, a symmetry of a physical theory whose energy/action functional is invariant under certain continuous group of transformations. Gauge theories always contain redundant degrees of freedom and have underdetermined Euler–Lagrange equations. Additional constraints ought to be imposed to break this redundancy in what constitutes the *gauge-fixing procedure*. We shall now summarize this procedure for spherical fluid membranes as presented in [27].

The gauge-fixing procedure entails supplementing the free energy $\mathcal{E}$ (cf., Equation (4)) with an additional term, i.e.,:

$$\tilde{\mathcal{E}} = \mathcal{E} + \mathcal{E}_{\mathrm{HM}}\,, \tag{6}$$

where the second term is the *harmonic map energy* given by:

$$\mathcal{E}_{\mathrm{HM}} = \int_{\omega} \frac{1}{2} g^{\alpha\beta} h_{\alpha\beta}\, da\,. \tag{7}$$

Here, $g_{\alpha\beta}$ $(\alpha, \beta = 1, 2)$ are the (contravariant) components of the metric tensor of the deformed surface ω, and $h_{\alpha\beta}$ are the (covariant) components of the metric tensor of the reference surface S^2. Although it might seem that by adding $\mathcal{E}_{\mathrm{HM}}$ (Equation (7)) to the free energy, we are modifying the surface constitutive law, this is not the case. Despite the additional harmonic map term in Equation (6), at an equilibrium state, $\mathcal{E}_{\mathrm{HM}}$ does not alter the Euler-Lagrange equations of the Helfrich–Canham energy. To see this, let us first note that the tangential components of the Euler-Lagrange equation due to $\mathcal{E}$ are trivially zero [37,41]. Thus, the only contribution to the Euler–Lagrange equations of (6) in the tangential direction is due to $\mathcal{E}_{\mathrm{HM}}$. The tangential variation of $\mathcal{E}_{HM}$ is explicitly given by [27]:

$$\delta_{||}\mathcal{E}_{\mathrm{HM}} = \int_{\omega} h_{\alpha\beta} \Big[\frac{1}{\sqrt{g}} \partial_{\mu} \Big(\sqrt{g} g^{\alpha\mu} \Big) + g^{\mu\gamma} \Upsilon^{\alpha}_{\mu\gamma} \Big] v^{\beta}\, da\,, \tag{8}$$

where $\Upsilon^{\cdot}_{\cdot\cdot}$ is the Christoffel symbol corresponding to the metric tensor h and v^{β} is the tangential variation along ω. The term in the brackets in Equation (8) (for each $\beta \in \{1, 2\}$) is the tangential component of the Euler-Lagrange equations of the harmonic map. The normal component is given by:

$$\delta_{\perp}\mathcal{E}_{\mathrm{HM}} = \frac{1}{2} \int_{\omega} h_{\alpha\beta} \Big[2b^{\alpha\beta} - 2Hg^{\alpha\beta} \Big] w\, da\,, \tag{9}$$

where w is a smooth variation normal to ω.

It has been shown in Theorem 3, Appendix D of [27] that

Theorem 1. *If*

$$\frac{1}{\sqrt{g}} \partial_{\mu} \Big(\sqrt{g} g^{\alpha\mu} \Big) + g^{\mu\gamma} \Upsilon^{\alpha}_{\mu\gamma} = 0, \ \textit{for } \alpha \in \{1, 2\}, \tag{10}$$

then $\delta_{\perp}\mathcal{E}_{HM} \equiv 0$ *for all smooth variations* w.

That is, if an equilibrium state satisfies the tangential component of the Euler–Lagrange equations due to $\mathcal{E}_{\mathrm{HM}}$, then the contribution to the normal component of the Euler–Lagrange equations due to $\mathcal{E}_{\mathrm{HM}}$ is trivially zero. Thus, the normal component of the equilibrium equation that determines the lipid membrane's shape remains unmodified. In other words, at equilibrium, the harmonic map energy only contributes to the Euler–Lagrange equations' tangential components and, in doing so, provides the constraints necessary to prevent arbitrary tangential motions. The equilibria for the gauge-fixed and the original formulation are identical.

The principal advantage of the gauge-fixing procedure is that adding the harmonic map energy $\mathcal{E}_{\mathrm{HM}}$ in Equation (6) does not change the physics of the problem. Furthermore, the gauge-fixed formulation is computationally more efficient than the previously mentioned iterative formulations, since no additional energy terms are included that need to iteratively be reduced to zero.

The gauge-fixing procedure developed in [27] only includes $\mathcal{E}_{\text{HC}}$ while $\mathcal{E}$ (c.f., Equation (4)) also includes the particle interaction energy, U. However, this inclusion does not affect the gauge-fixing procedure as the only relevant fact is that the energy is reparametrization invariant. In light of Remark 2, we note that since the particles are also "fluid-like", $\tilde{\mathcal{E}}$ is also reparametrization invariant and the gauge-fixing procedure can be applied to the present problem.

One notable feature of Equation (6) is that unlike $\mathcal{E}$ (which was invariant under arbitrary reparametrizations of the surface), the *gauge-fixed* formulation, i.e., $\tilde{\mathcal{E}}$, is invariant under conformal reparametrizations of the surface. Thus, the gauge-fixing procedure does not break completely the reparametrization invariance symmetry. It has been shown in [27] that configurations that remain in the equivalence class are related by the six-dimensional Möbius group of transformations. These six modes can be viewed as the rigid translation and rotation modes of the sphere [42]. The three translational modes can be constrained by imposing the "zero-mass" constraint [27]:

$$\int_\omega \mathbf{R}(X^1, X^2)\, da = \mathbf{0}\,, \tag{11}$$

where $\mathbf{R}(X^1, X^2)$ is a parametrization of the reference sphere. The three rotational modes can, in theory, be fixed by landmark constraints [43]. In practice, however, we found that reliable results (presented in Section 3) could be obtained without any landmark constraints when we used the L-BFGS [44] numerical minimization algorithm.

2.2. Lagrangian Particle Formulation

One of the challenges in discretizing $\tilde{\mathcal{E}}$ is that the particles must always lie on the surface ω that is itself determined by the configuration of the particles. The shape of the surface and the arrangement of the particles are inextricably coupled, and one cannot be solved independently of the other. In [24], we have developed a variational *Lagrangian* formulation to address this challenge. In this formulation, the particles' 3D positions on ω are parametrized by the particles' coordinates $\mathbf{X}_i$ on the reference surface S^2, which, therefore, become the particles' degrees of freedom. Since the particles are fluid-like and do not have a preferred reference configuration, $\mathbf{X}_i$ have no physical significance. To obtain the actual 3D position $\mathbf{x}_i$ of the particles, we compose $\mathbf{X}_i$ with the deformation map $\mathbf{f} : S^2 \to \omega$, i.e., $\mathbf{x}_i = \mathbf{f}(\mathbf{X}_i)$. This is illustrated schematically in Figure 2. Here, $\mathbf{x}_i$ is the position of particle i on ω, the relevant variable that appears in the interaction energy U (see Equation (3)). Using the Lagrangian formulation, we can write the free energy in Equation (4) with respect to the reference surface as:

$$\mathcal{E}[\mathbf{X}_1, \cdots, \mathbf{X}_n, \mathbf{f}] = \int_{S^2} \Big[\kappa (H - C_0)^2 + \kappa_g K\Big] J\, \mathrm{d}A - pV + U(\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N)\,, \tag{12}$$

where $J = \sqrt{g}/\sqrt{G}$ is the Jacobian determinant, expressed in terms of g and G, the determinants of the metric tensor of ω and S^2, respectively, and we adopted the notation $\mathbf{f}_i := \mathbf{f}(\mathbf{X}_i)$. The area measure on S^2 is denoted as dA. The degrees-of-freedom for the system are $\mathbf{X}_1, \cdots \mathbf{X}_N$, and $\mathbf{f}$, which are explicitly indicated in (12).

Equilibrium configurations are determined by finding critical points of (12) (or, the gauge-fixed formulation (6)). In a variational setting, this requires taking the variations of (12) with respect to the degrees-of-freedom, viz., $\mathbf{f}$ and $\mathbf{X}_i$ ($i = 1, \cdots N$). Let $\boldsymbol{\eta}(\mathbf{X})$ denote the variation in $\mathbf{f}(\mathbf{X})$ and $\boldsymbol{\theta}_i$ the variation in $\mathbf{X}_i$. As discussed in [24], care must be taken while computing the variation in $\mathbf{f}_i = \mathbf{f}(\mathbf{X}_i)$. For each $i \in \{1, 2, \cdots, N\}$ we obtain:

$$\delta \mathbf{f}_i = \delta[\mathbf{f}(\mathbf{X}_i)] = \boldsymbol{\eta}(\mathbf{X}_i) + \nabla \mathbf{f}(\mathbf{X}_i) \boldsymbol{\theta}_i\,, \tag{13}$$

where $\nabla\mathbf{f}$ is the deformation gradient tensor expressed in terms of the surface gradient $\nabla(\cdot)$. Using basis vectors $\mathbf{a}_\alpha := \partial\mathbf{f}/\partial X^\alpha$ ($\alpha = 1, 2$) for the tangent space to ω at $\mathbf{X}$, we can write $\nabla\mathbf{f}(\mathbf{X}) = g^{\alpha\beta}\mathbf{a}_\alpha \otimes \mathbf{a}_\beta$, where $g^{\alpha\beta}$ are the contravariant components of the metric tensor (defined explicitly in the next section). We see from (13) that $\delta\mathbf{f}_i$ has two contributions: the first one accounts for variation in the position of the substrate, and the second one accounts for the variation of the particle positions in the reference configuration $\boldsymbol{\theta}_i$, which are pushed forward to the current configuration by the deformation gradient tensor. Note that all the tensor and vector fields in (13) are evaluated at $\mathbf{X}_i$.

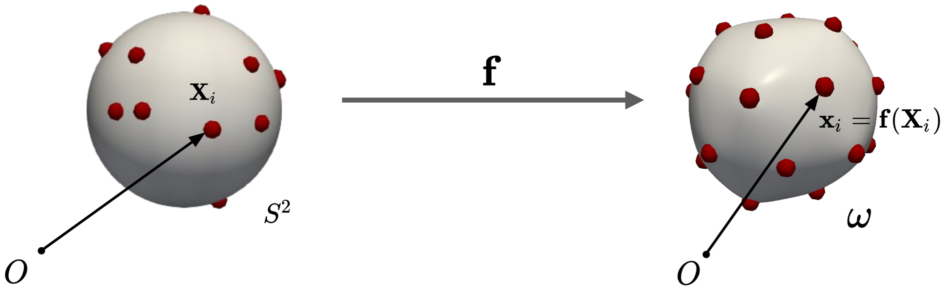

Figure 2. Reference and deformed configurations. The formulation degrees of freedom are the deformation mapping $\mathbf{f}(\mathbf{X})$ and the particle positions $\mathbf{X}_i$ ($i \in \{1, 2, \cdots, N\}$) in the reference configuration.

Weak Form

In this section, we derive the weak-form for the gauge-fixed formulation, cf. (6), (7), and (12). For clarity, we shall begin by recalling some notation and important differential geometry identities that we will use to derive the weak form. Much of what we present in this section, especially concerning the weak form for the gauge-fixed Helfrich–Canham energy, can be found in [27]. The computation of the first variation of the particles-substrate energies using the Lagrangian formulation discussed in the previous section is based on our previous work [24].

The basis for the tangent space of ω at $\mathbf{X}$ (parametrized by X^1 and X^2) is spanned by

$$\mathbf{a}_\alpha := \mathbf{f}_{,\alpha} = \frac{\partial\mathbf{f}}{\partial X^\alpha} \in \mathbb{R}^3,\ \alpha \in \{1, 2\}\,, \tag{14}$$

where we use $(\cdot)_{,\alpha} = \partial(\cdot)/\partial X^\alpha$. The components of the metric tensor of ω are given in terms of the Euclidean dot product in $\mathbb{R}^3$ by

$$g_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta\,.$$

The dual basis vectors of $\mathbf{a}_\alpha$ are denoted as

$$\mathbf{a}^\alpha := g^{\alpha\beta}\mathbf{a}_\beta\,,$$

where $g^{\alpha\beta}$ are the contravariant components of the metric tensor and we used the Einstein summation convention on the repeated indices. This convention is employed in the following discussion as well. The unit normal field to ω is given by

$$\mathbf{d} = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\sqrt{g}}\,,$$

where we have used the identity $\sqrt{g} = ||\mathbf{a}_1 \times \mathbf{a}_2||$.

The components of the second fundamental form of ω are given by

$$b_{\alpha\beta} = -\mathbf{d}_{,\alpha} \cdot \mathbf{a}_\beta = \mathbf{d} \cdot \mathbf{a}_{\alpha,\beta}\,.$$

The mean and Gaussian curvatures can then be computed as, respectively:

$$H = \frac{1}{2} b^{\alpha}_{\alpha} = -\frac{1}{2} \mathbf{a}^{\alpha} \cdot \mathbf{d}_{,\alpha}\,, \quad K = \det(b^{\alpha}_{\beta})\,, \tag{15}$$

where, as per convention, tensor indices are lowered or raised by multiplying with $g_{\alpha\beta}$ or its inverse $g^{\alpha\beta}$.

To derive the weak form of the gauge-fixed functional, $\tilde{\mathcal{E}}$, we compute its variation $\delta\tilde{\mathcal{E}}$ with respect to $\boldsymbol{\eta}$ (the variation in $\mathbf{f}$) and $\boldsymbol{\theta}_i$ (the variation in $\mathbf{X}_i$). It follows from Equation (6) that

$$\delta\tilde{\mathcal{E}} = \delta\mathcal{E} + \delta\mathcal{E}_{\mathrm{HM}}\,. \tag{16}$$

The first variation on the right side is computed by taking the variations of all the contributing terms in Equation (4). To do so, let us first define the following quantities:

$$\begin{aligned}\mathbf{n}^{\alpha} = {} & [\kappa(H - C_0) + 2\kappa_g H] g^{\alpha\beta} \mathbf{d}_{,\beta} + [\kappa(H - C_0)^2 + \kappa_g(K - b^{\nu}_{\beta} \mathbf{a}^{\beta} \cdot \mathbf{d}_{,\nu})] \mathbf{a}^{\alpha} + \\ & \gamma\sqrt{a}\,\mathbf{a}^{\alpha} - \frac{1}{3} pV[(\mathbf{f} \cdot \mathbf{d})\mathbf{a}^{\alpha} - (\mathbf{f} \cdot \mathbf{a}^{\alpha})\mathbf{d}]\,,\end{aligned} \tag{17}$$

$$\mathbf{m}^{\alpha} = [-\kappa(H - C_0)\delta^{\alpha}_{\beta} + \kappa_g(b^{\alpha}_{\beta} - 2H\delta^{\alpha}_{\beta})]\mathbf{a}^{\beta}\,, \tag{18}$$

where δ^{α}_{β} is the Kronecker delta and γ is the Lagrange multiplier enforcing the area constraint (see Equation (2)). It can be shown (see [25,27] for details) that:

$$\begin{aligned}\delta\mathcal{E} = {} & \int_{S^2} \left[\mathbf{n}^{\alpha} \cdot \delta\mathbf{a}_{\alpha} + \mathbf{m}^{\alpha} \cdot \delta\mathbf{d}_{,\alpha} - \frac{1}{3} pV \mathbf{d} \cdot \boldsymbol{\eta} \right] J dA + \\ & \sum_{i=1}^{N} \sum_{j>i}^{N} \Phi'(r_{ij}) \frac{(\mathbf{f}_i - \mathbf{f}_j)}{||\mathbf{f}_i - \mathbf{f}_j||} \cdot (\delta\mathbf{f}_i - \delta\mathbf{f}_j)\,,\end{aligned} \tag{19}$$

where $r_{ij} = ||\mathbf{f}_i - \mathbf{f}_j||$, and the variations $\delta\mathbf{a}_{\alpha}$ and $\delta\mathbf{d}_{,\alpha}$ are given by [26]:

$$\delta\mathbf{a}_{\alpha} = \boldsymbol{\eta}_{,\alpha}\,, \tag{20}$$

$$\delta\mathbf{d}_{,\alpha} = -[\mathbf{a}^{\beta}_{,\alpha} \otimes \mathbf{d} + \mathbf{a}^{\beta} \otimes \mathbf{d}_{,\alpha}] \cdot \boldsymbol{\eta}_{,\beta} - [\mathbf{a}^{\beta} \otimes \mathbf{d}] \cdot \boldsymbol{\eta}_{,\beta\alpha}\,, \tag{21}$$

where $\otimes$ is the tensor product of vectors in $\mathbb{R}^3$. The last term in Equation (19) was obtained by taking the variation of U, cf. Equation (3), using $\mathbf{x}_i = \mathbf{f}(\mathbf{X}_i)$. Equation (19) can be further simplified using Equation (13) to obtain:

$$\begin{aligned}\delta\mathcal{E} = {} & \int_{S^2} \left[\mathbf{n}^{\alpha} \cdot \delta\mathbf{a}_{\alpha} + \mathbf{m}^{\alpha} \cdot \delta\mathbf{d}_{,\alpha} - \frac{1}{3} pv \mathbf{d} \cdot \boldsymbol{\eta} \right] J dA + \\ & \sum_{i=1}^{N} \sum_{j>i}^{N} \left[[\mathbf{g}_{ij} \cdot \boldsymbol{\eta}(\mathbf{X}_i) - \mathbf{g}_{ij} \cdot \boldsymbol{\eta}(\mathbf{X}_j)] + [\mathbf{g}_{ij} \cdot \mathbf{a}_{\alpha}(\mathbf{X}_i)\theta^{\alpha}_i - \mathbf{g}_{ij} \cdot \mathbf{a}_{\alpha}(\mathbf{X}_j)\theta^{\alpha}_j] \right]\,,\end{aligned} \tag{22}$$

where θ^{α}_i $(\alpha = 1, 2)$ are the components of $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta}_i := \theta^{\alpha}_i \mathbf{a}_{\alpha}$, and

$$\mathbf{g}_{ij} := \Phi'(r_{ij}) \frac{(\mathbf{f}_i - \mathbf{f}_j)}{||\mathbf{f}_i - \mathbf{f}_j||}\,. \tag{23}$$

The second term on the right side of Equation (16) can be evaluated by taking the variation of Equation (7), leading to [27],

$$\delta\mathcal{E}_{\mathrm{HM}} = \frac{1}{2} \int_{S^2} h_{\alpha\beta} \Big(-2g^{\alpha\gamma}\mathbf{a}^{\beta} + g^{\alpha\beta}\mathbf{a}^{\gamma} \Big) \cdot \boldsymbol{\eta}_{,\gamma} J\, dA\,. \tag{24}$$

Finally, the weak form of the equilibrium equations is given by the condition:

$$\delta\mathcal{E} + \delta\mathcal{E}_{\text{HM}} = 0\,, \tag{25}$$

for all admissible variations $\boldsymbol{\eta} : S^2 \to \mathbb{R}$ and $\theta_i^\alpha \in \mathbb{R}$, $(i = 1, 2, \cdots, N$ and $\alpha = 1, 2)$. If requested, the area constraint (Equation (2)) must also be imposed.

We conclude this section with a remark. A minimization solver cannot be employed to compute the equilibria of $\tilde{\mathcal{E}}$ using the weak form in Equation (25). Indeed, since $\mathcal{E}_{\text{HM}}$ does not need to be positive, a minimizer of $\tilde{\mathcal{E}}$ is not necessarily the minimizer of $\mathcal{E}$. That is, the gauge-fixing procedure of adding $\mathcal{E}_{\text{HM}}$ to the energy could potentially turn local minima into saddle points and local maxima. To circumvent this problem, in the numerical results presented in Section 3, we minimize instead:

$$\hat{\mathcal{E}} := \mathcal{E} + \lambda_g(\mathcal{E}_{\text{HM}})^2 + \mu_1\Big(\int_{S^2} J\,dA - 4\pi\Big)^2 + \mu_2\Big|\int_{S^2} \mathbf{R}(X^1, X^2) J\,dA\Big|^2\,, \tag{26}$$

where the harmonic map energy appears as the second (squared) term. The coefficient λ_g is a parameter that controls the strength of the harmonic map energy. The last two terms are the area (Equation (2)) and "zero-mass" (Equation (11)) constraints, which are imposed using a penalty formulation with parameters μ_1 and μ_2. The weak form associated with this modified functional is given by:

$$\delta\mathcal{E} + (2\lambda_g\mathcal{E}_{\text{HM}})\delta\mathcal{E}_{\text{HM}} + \lambda_1\int_{S^2} \mathbf{a}^\alpha\cdot\boldsymbol{\eta}_\alpha J\,dA + \lambda_2\int_{S^2} \mathbf{R}(X^1, X^2)\mathbf{a}^\alpha\cdot\boldsymbol{\eta}_\alpha J\,dA = 0\,, \tag{27}$$

where $\lambda_1 = 2\mu_1(\int_{S^2} J\,dA - 4\pi)$ and $\lambda_2 = 2\mu_2\int_{S^2} \mathbf{R}J\,dA$. The last two terms in Equation (27) were obtained by taking the variations of the last two terms in Equation (26) and using the identity for variation in J, viz., $\delta J = J\mathbf{a}^\alpha\cdot\boldsymbol{\eta}_\alpha$. The terms $\delta\mathcal{E}$ and $\delta\mathcal{E}_{\text{HM}}$ appearing in Equation (27) are given by Equation (22) and Equation (24), respectively. If instead of a fixed pressure p, a volume constraint (Equation (5)) was to be imposed, then the corresponding penalty term must be added to Equations (26) and (27).

2.3. Loop Subdivision Finite Element Method for Thin Shells

To discretize Equation (26) and its weak form Equation (27), we employ a Ritz–Galerkin approach. Due to the dependence of $\mathcal{E}_{HC}$ on the mean curvature H, this discretization step requires the use of shape functions that ensure C^1 continuity across the elements. Furthermore C^1 continuity across the elements is also required due to Lagrangian particle formulation adopted here. Indeed, mapping particles, described in terms of their reference positions, to the deformed mesh (where the interaction energies must be calculated) requires the evaluation of the deformation gradient tensor, potentially also at the interface between elements. To satisfy the C^1 continuity requirement, we employ the thin-shell finite element method based on (Loop) subdivision surfaces first proposed in [33] for solid shells. This approach achieves C^1 continuity solely using the nodal displacement degrees-of-freedom of a triangular mesh. The shape functions are box-splines defined in terms of *barycentric coordinates* on an element. Unlike traditional finite elements, these functions are supported on a ring of neighbors around a given element (e.g., see Figure 2 in [26]). Despite having non-local support, this method is very efficient and it has been used to solve a variety of shell problems, including problems on fluid shells [25–27]. Explicit details of this method, including its implementation, can be found, for example, in [26,33].

A quirky feature of the method is that the shape functions are explicitly defined only in a triangular element belonging to a *regular patch*, i.e., a (triangular) element whose vertices have all valence equal to six. Elements that are not regular are called *irregular*. Evaluating shape functions at a point inside an irregular element is not straightforward. The element must be subdivided (using the Loop scheme [34]) as many times as required so that the point of interest (e.g., the location of a quadrature point or of a particle) is inside a regular (subdivision) element. Repeated subdivisions may be performed by explicitly multiplying nodal coordinates with appropriate matrix operators. However, this subdivision step can be computationally cumbersome depending on the number of required subdivisions. In [35], an elegant and efficient implementation has been proposed to evaluate shape functions on irregular elements without any explicit matrix operator multiplications. Sets of functions called *eigenbasis* required for these calculations have been tabulated in [45] for a variety of nodal valences.

The distinction between regular versus irregular elements is particularly relevant to our problem for two reasons. Firstly, since we consider topologically spherical surfaces, it is well known that it is impossible to triangulate such a surface only with regular elements. According to Euler's theorem, at least 12 nodes with valence five are needed to cover topologically spherical surfaces. Secondly, while computing a particle's 3D coordinates, it is necessary to evaluate the shape functions at arbitrary points on the surface. Some of these points could potentially be inside an irregular element or at the edges of one. Explicit subdivision by multiplying nodal degrees-of-freedom by the subdivision matrix (see Equation (70) in [33]) would be inefficient as this step must be performed for every particle on an irregular element and at each step of the minimization iteration. In our implementation, we therefore employ the eigenbasis discussed in [35,45] to evaluate shape functions efficiently.

2.4. Search Algorithm

Computing the discretized energy and the weak form requires to evaluate the shape functions at the coordinates of the particles. For example, this can be seen in Equation (22), where the variation $\boldsymbol{\eta}$ and function $\mathbf{a}_\alpha$ appearing inside the summations must be evaluated at $\mathbf{X}_i$ ($i = 1 \cdots, N$). Since the shape functions are defined piece-wise over each element, an important step in the finite element implementation consists in identifying the element to which a given particle belongs. Mapping the particles to their corresponding element will necessarily involve searching over the mesh elements. This section will present an efficient search algorithm for this purpose.

We distinguish between two types of coordinates to parametrize particle positions—*global coordinates* and *local coordinates*. Global coordinates are defined continuously on the entire reference surface. In the formulation presented in Section 2.2, $\mathbf{X}_i$ are the particles' global coordinates. Since the reference surface is spherical, we can use spherical coordinates for the global coordinates, i.e., $\mathbf{X}_i = (\vartheta_i, \phi_i)$, where ϑ_i is the co-latitude angle of particle i measured from the positive z-axis and ϕ_i is its azimuthal angle. Local coordinates, on the other hand, are defined per element. For the Loop subdivision finite element scheme, we use barycentric coordinates (u, v, w) (with $u + v + w = 1$) as local coordinates to locate particles on a given triangular element.

The goal of the search algorithm is to map the global coordinates of a given particle to its local (barycentric) coordinates on an element. The inputs to the algorithm are the mesh (nodes and connectivity information) and global coordinates of a given particle. The algorithm's outputs are the local coordinates and the index of the triangular element. The local coordinate and the index will then be used to evaluate the shape functions needed in computing $\mathcal{E}$ and $\partial\mathcal{E}$ (Equations (12) and (22)).

Note that if we determine the (index of the) element containing the particle, then we can determine the local coordinates using the ray-triangle intersection algorithm (see Appendix A). Thus, the critical step in the search algorithm is to determine the element containing a given particle. This can be naively achieved using a brute-force approach: loop over all elements for each particle and determine if an element contains the particle using the ray-triangle intersection method. However, this approach is inefficient as it involves many failed and unnecessary searches.

Instead, a computationally efficient approach would be to bin the elements into subgroups depending on their relative proximity in 3D (i.e., construct a hash table). Then, given a particle position, we can use its coordinates to quickly identify the subgroup it belongs to and only search in the elements belonging to that subgroup using the ray-triangle intersection algorithm. For all the elements in the subgroup to which the particle does not belong, the ray-triangle intersection algorithm will return barycentric coordinates outside the expected $[0,1]$ interval.

To construct the hash table, we divide $\mathbf{R}^3$ into $M_X \times M_Y \times M_Z$ rectangular boxes, in the X, Y, and Z direction, respectively. We begin by scaling the mesh such that it spans the boxes. The scale factors must be stored for use later in the ray-triangle intersection step. Although optional, this step allows to use integer arithmetic in the next steps, therefore improving efficiency. We index each box with a tuple of integers (p,q,r), where p, q, and r are the minimum X, Y, and Z coordinates of points in the box. We then associate to each box the list of triangular elements that intersect the box. This is done using two loops as illustrated in Algorithm 1. We first loop over all the elements in the mesh and determine all the boxes a given element intersects. This list is called $faceList$ in Algorithm 1. Next, we invert $faceList$ to associate each box with all the elements contained (even partially) in that box. Note that an empty box (i.e., not containing any face) is not listed in the final hash table. The resulting hash table that associates a given box with all the faces it intersects is stored in $boxList$.

Algorithm 1 Hash table algorithm.

```
Input: $ver \leftarrow$ vertices, $con \leftarrow$ connectivity, $M_X$, $M_Y$, $M_Z \leftarrow$ Integers
Output: A hash table with entries $\{box : face1, face2, \dots\}$ for each box.
  scale $ver$ s.t. the mesh spans $M_X M_Y M_Z$ boxes
  for $face$ in $con$ do
      $v1, v2, v3 \leftarrow$ coordinates of three vertices of $face$
      $X_{\min} \leftarrow$ floor[min$(v1_X, v2_X, v3_X)$], $X_{\max} \leftarrow$ ciel[max$(v1_X, v2_X, v3_X)$]
      $Y_{\min} \leftarrow$ floor[min$(v1_Y, v2_Y, v3_Y)$], $Y_{\max} \leftarrow$ ciel[max$(v1_Y, v2_Y, v3_Y)$]
      $Z_{\min} \leftarrow$ floor[min$(v1_Z, v2_Z, v3_Z)$], $Z_{\max} \leftarrow$ ciel[max$(v1_Z, v2_Z, v3_Z)$]
      listOfBoxes = all boxes with index $(p,q,r)$ with $p \in [X_{\min}, X_{\max}]$, $q \in [Y_{\min}, Y_{\max}]$, $r \in [Z_{\min}, Z_{\max}]$.
      Append $\{face : listOfBoxes\}$ to the temporary dictionary data-structure $faceList$.
  end for
  for box in all boxes do
      for $f$ in $faceList$ do
          if $box \in f$.VALUES() then
              Construct a hash table $boxList$ with entries $\{box : f\}$ or append $f$ to $box$.VALUES() if entry with $box$ key already exists.
          end if
      end for
  end for
  Return $boxList$
```

Once the hash table $boxList$ is generated, the search algorithm that converts global to local coordinates consists of the following steps. We first use the global coordinates to determine the box in which the particle lies. This is a straightforward calculation as it entails using integer arithmetic. Then, the identified box is passed to the hash function, which returns the list of triangles associated with it. Finally, the ray intersection algorithm is used to search through the list of elements in the identified box. This final search returns the index of the triangle which contains the particle and the particle barycentric coordinates in the identified element. The overall search strategy is schematically shown in Figure 3.

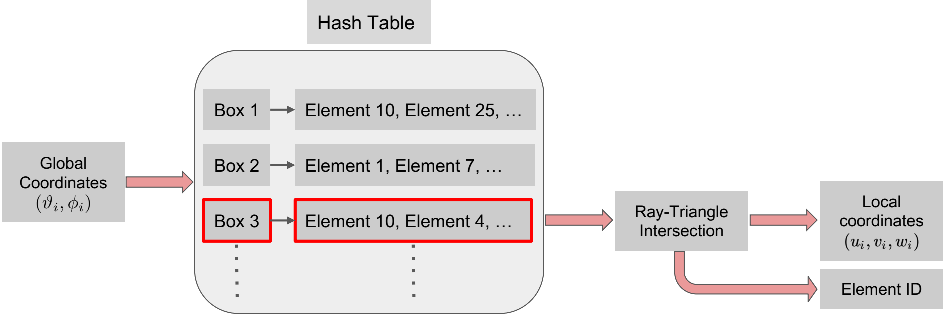
Hash Table
Global Coordinates (ϑ_i, ϕ_i)
Box 1
Element 10, Element 25, …
Box 2
Element 1, Element 7, …
Box 3
Element 10, Element 4, …
Ray-Triangle Intersection
Local coordinates (u_i, v_i, w_i)
Element ID

Figure 3. Schematic of the search algorithm used to convert the global coordinates (ϑ_i, ϕ_i) of particle i to its local barycentric coordinates (u_i, v_i, w_i). The algorithm first locates the list of faces/element IDs (schematically highlighted in red) in the hash table (generated by Algorithm 1) and then searches this list with the ray-triangle intersection algorithm to compute the barycentric coordinates and finds the element ID.

The choice of the number of boxes (determined by M_X, M_Y, and M_Z) will determine the algorithm's efficiency. At one extreme, $M_X = M_Y = M_Z = 1$ (i.e., the mesh is contained in a single box) corresponds to a hash table with only one box containing all the faces. In this case, determining the particle barycentric coordinates and element would be equivalent to a brute-force search. The other extreme, consists in choosing large values for M_X, M_Y, and M_Z, so that few elements are contained in each box. In this case, the hash table will contain a large number of entries with the same elements contained in many boxes. This is also inefficient. Although M_X, M_Y, and M_Z should be chosen depending on the number of elements used to discretize the domain of interest, in our numerical examples we have noticed that the efficiency of the search algorithm is not highly sensitive to these parameters. For most of the results presented in Section 3, we use a finite element mesh with 2562 nodes and 5120 triangular elements, and we adopt $M_X = M_Y = M_Z = 15$. The resulting hash table (which does not include empty boxes) contained 1111 entries, each entry containing an average of 16 elements.

We conclude this section by noting that, due to the Lagrangian particle formulation, the particles' global positions in the reference configuration are the input to the search algorithm. As a result, the hash table generated using Algorithm 1 needs to be *computed only once* for a given discretization of the reference sphere. Even as model parameters are modified, which, in turn, changes the substrate shape and the organization of the particles, the hash table does not need to be updated. Thus, the Lagrangian particle approach combined with this search algorithm offers a significant reduction in computational effort.

3. Results and Discussion

We now present computational results generated using the finite element formulation described above. We obtain these results by minimizing the discretized form of (26) using the weak form (27) with a gradient-based L-BFGS minimization solver [44]. A random initial state was used to initiate minimization. Unless otherwise stated, for all the results presented here, we used an icosahedrally symmetric spherical triangular mesh with 2562 nodes and 5120 elements. The chosen penalty parameters (cf., (26)) were $\mu_1 = 1000$, $\mu_2 = 1000$ and the preferred curvature C_0 was set equal to zero. Unless indicated otherwise, no volume constraint was imposed and the internal pressure was set to zero. Since the system's topology was not allowed to change, the value of κ_g does not influence the equilibrium state, a fact attributed to the Gauss–Bonnet theorem. Recall that the parameter λ_g weighs the effect of the harmonic map energy and since this parameter appears as a penalty in Equation (26), a larger value is ideal but can cause slow convergence. In the following simulations, the value λ_g was chosen to aid convergence. We remark that, once computed, the equilibrium configuration is not affected by the value λ_g.

Before presenting our results we note that caution should be exercised while visualizing the substrate/particle system, especially when coarser meshes are used. As shown in

Figure 4a, when 642 nodes (and 1280 elements) were used to simulate a system with $N = 3$ particles, the particles seem to lose contact with the surface. That is, the particles appear to lie inside the triangular enclosure, hovering below the surface, despite the Lagrangian particle framework being designed to prevent such situations. The cause of this apparent discrepancy stems from the visualization scheme used. In Figure 4a, only the nodal positions are used to generate the surface. However, the actual surface used by the Loop subdivision finite element scheme is the surface generated using box splines [34]. When this fact is taken into account, the discrepancy is resolved (see Figure 4b).

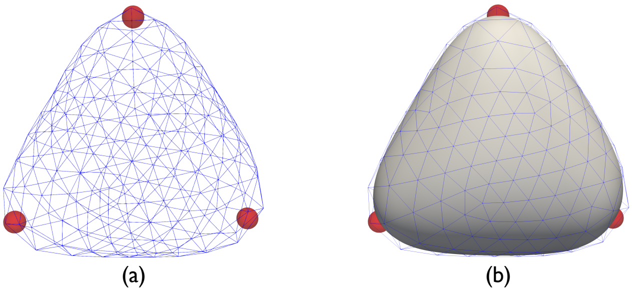

Figure 4. (**a**) Apparent loss of contact between particle and surface while visualizing the system using the underlying triangular mesh. (**b**) Actual surface shape after interpolation using Loop subdivision box splines (the underlying triangular mesh, shown in blue, is reported for reference).

To validate the proposed method, we computed equilibrium configurations of particles interacting via pair-wise Coulombic interactions, i.e., $\Phi(r) = 1/r$. The results of our simulations are presented in Figure 5. For these simulations, we chose $\kappa = 1$ and $\lambda_g = 10$. We observe that the system replicates the symmetry configurations of the classical Thomson problem [46], which would correspond to the large bending stiffness limit. In the figure, we have employed the Schönflies notation for the symmetry groups: D_{nh} represents an n-fold dihedral symmetry with a horizontal mirror plane, O_h represents octahedral symmetry, etc.

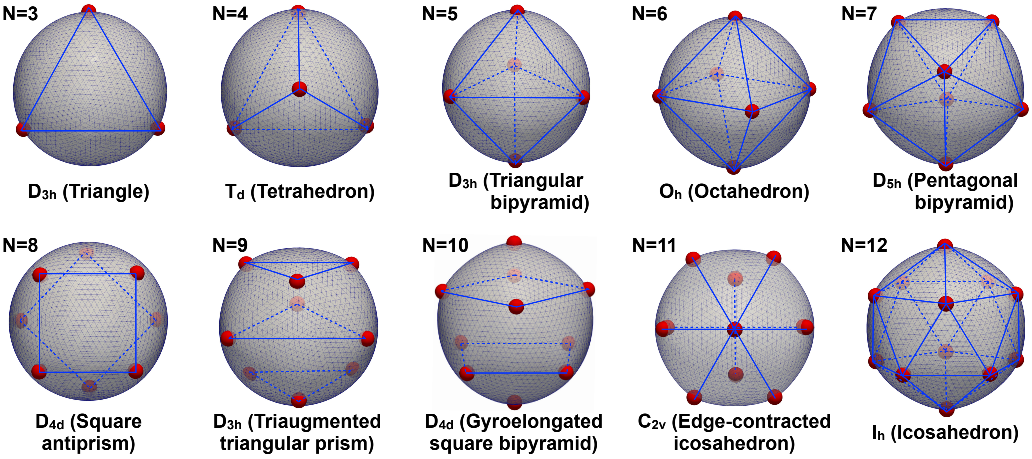
N=3
N=4
N=5
N=6
N=7
D3h (Triangle)
Td (Tetrahedron)
D3h (Triangular bipyramid)
Oh (Octahedron)
D5h (Pentagonal bipyramid)
N=8
N=9
N=10
N=11
N=12
D4d (Square antiprism)
D3h (Triaugmented triangular prism)
D4d (Gyroelongated square bipyramid)
C2v (Edge-contracted icosahedron)
Ih (Icosahedron)

Figure 5. N particles interacting via Coulomb electric potential over a spherical surface with *high bending stiffness*. This case study approaches the classic Thomson problem where electric charges interact on a rigid sphere. The same symmetries computed in the Thomson problem are obtained for $N = 3$ to $N = 12$ [46], providing validation of the proposed approach. The finite element mesh used to discretize the underlying deformable substrate is shown together with lines (blue) to highlight the symmetry state of the computed particles (red) positions.

Figure 6 shows the effect of changing the bending stiffness κ for $N = 12$ particles interacting via a Lennard–Jones potential $\Phi(r) = \epsilon[(r_e/r)^{12} - 2(r_e/r)^6]$. In these simulations $\epsilon = 0.1$, $\lambda_g = 2$, $r_e = 1.3$, and the particles are initially arranged in an icosahedrally symmetric configuration. κ was initially set equal to 3 and was gradually decreased while a numerical continuation scheme used the equilibrium configuration computed at each step as the initial guess for the subsequent step. As the bending stiffness κ decreases, a "stellated" particles/substrate configuration emerges, demonstrating the ability of the proposed formulation to model large shape changes and deformation of the coupled system.

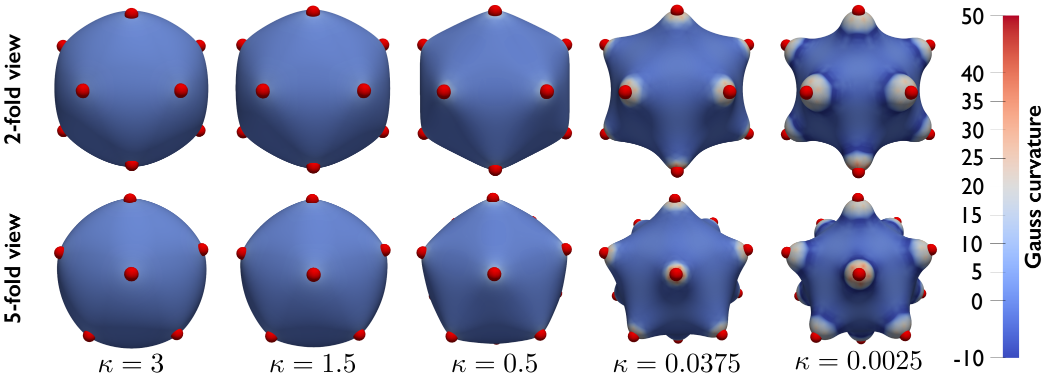

Figure 6. Particle—substrate configurations obtained while gradually decreasing the bending stiffness κ. The system is composed of $N = 12$ particles interacting via Lennard–Jones potential and originally arranged in an icosahedral configuration. As the substrate bending stiffness decreases, the system configuration evolves from spherical to "stellated", demonstrating that the proposed formulation can be employed to study large changes in the deformation and configurations of the particle/substrate system.

The proposed formulation could be employed to study pinching of the substrate due to the particle interactions. We demonstrate this ability by analyzing a system with $N = 40$ particles interacting via a harmonic interaction potential $V = (r - r_e)^2/2$, with $r_e = 0.8$. The other simulations parameters are $\kappa = 1$, $\lambda_g = 5$, and a volume constraint is also included. The state shown on the left of Figure 7 corresponds to a reduced volume $v = 0.95$ (cf., Equation (5)), while the state on the right corresponds to a reduced volume $v = 0.90$. These pinched states are reminiscent of viral budding or exo-cytosis. We note that it would not be possible to model the pinched neck state (shown in Figure 7, right) using the radial graph ansatz presented previously in [24] as the radial vector will no longer be a well-defined, single-valued function.

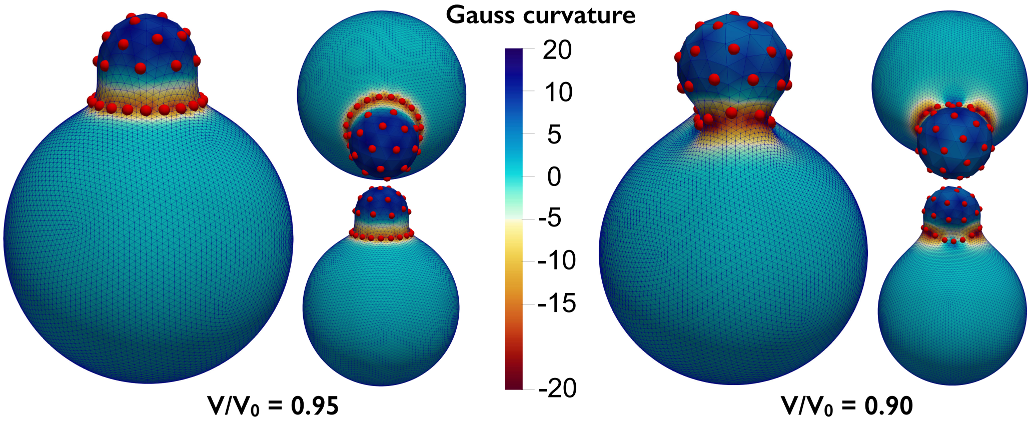

Figure 7. Particles interacting via harmonic potential over a deformable substrate with decreasing volume constraint. The volume is first decreased to 0.95 (**left**) and then to 0.9 (**right**) of the reference volume of a unit sphere ($4/3\pi$) producing budding of the underlying substrate. Lateral views from different angles (large panel and small bottom panel) and top/inclined view (small top panel) are provided. The mesh shown has been subdivided once according to the Loop scheme.

We conclude by presenting an example including a larger number of particles ($N = 200$) interacting via the Lennard–Jones potential described above. Initially, particles are randomly distributed on a half of the reference sphere. Subsequently, their equilibrium distance r_e is gradually increased from $r_e = 0.15$ to $r_e = 0.31$ in steps $\Delta r_e = 0.01$, while the remaining model parameters remain constant ($\kappa = 2$, $\epsilon = 0.1$, and $\lambda_g = 10$). As r_e increases, the particles move apart and start enveloping the substrate, until the full substrate is covered and deformed (Figure 8). This final example shows, once more, the ability of the proposed formulation to model the particles/substrate coupling and the effect of the particle arrangement on the substrate's curvature.

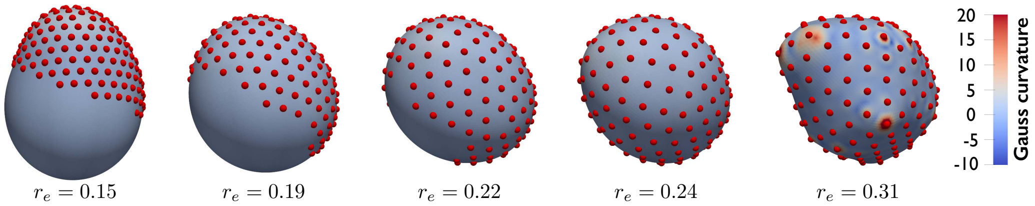

Figure 8. $N = 200$ particles interacting via Lennard−Jones potential and gradually covering a deformable substrate. In the system's initial configuration, the particles are present on a half of the reference sphere. The particles envelope and deform the full substrate as r_e increases showing the strong particle−substrate coupling.

4. Conclusions and Future Applications

This work presents a thin-shell finite element formulation combined with a Lagrangian particle framework to compute equilibrium states of interacting particles moving on a deformable substrate. We present the implementation of the formulation in the context of fluid shells, which we model with the Helfrich–Canham energy. Fluid shells are intrinsically degenerate structures, as any tangential motion along the surface of the shell does not contribute to its elastic energy. As a result, spurious zero-energy modes are present and commonly hinder the simulations of these systems. In this work, we adopt the *gauge-fixing* procedure recently developed in [27] to resolve the computational problems stemming from this degeneracy.

Since the particle positions are parametrized in the reference configuration, an attractive feature of the proposed formulation is that computation of terms in Equations (26) and (27) only requires the shape functions at locations on the reference mesh. An efficient search algorithm has been implemented to locate particles on the reference mesh by binning the mesh elements into boxes. Because of the referential description, the hash table generated for this purpose only needs to be created once.

This work generalizes and overcomes limitations of previous approaches to study particles moving of deformable substrates. Importantly, particles are not artificially pinned to the nodes of the substrate's mesh, lifting a spurious constraint that limited the configurations achievable by the particles/substrate system. Furthermore, a radial graph ansatz was not necessary due to the gauge-fixing formulation that we have employed.

In this work, we disregard the dynamics of the system and solely focus on equilibrium configurations. Moreover, by employing the Helfrich model for the membrane, the present work focuses on idealized fluid membranes where in-plane viscosity is zero. However, it has been known that lipid membranes are viscoelastic [47,48]. In this regard, it is possible to extend the approach presented here to incorporate viscoelastic effects. Time dependence of the deformation map of the membrane must be assumed, i.e., $\mathbf{f}(X, t)$. Since the particle positions are parametrized in the reference configuration (see Section 2.2), they must also explicitly depend on time, i.e., $\mathbf{X}_i(t)$. The absolute velocity of the particle is thus given by

$$\dot{\mathbf{x}}_i(t) = \frac{d}{dt}\mathbf{f}(\mathbf{X}_i(t), t) = \nabla\mathbf{f}(\mathbf{X}_i(t), t)\dot{X}_i(t) + \frac{\partial}{\partial t}\mathbf{f}(\mathbf{X}_i(t), t)\,, \tag{28}$$

where the dot represents the time derivative. Note that the second term in Equation (28) represents the velocity at the material point $\mathbf{X}_i$ on the membrane, thus the relative velocity of the particle with respect to the membrane is given by the first term of Equation (28), i.e., $\nabla \mathbf{f}(\mathbf{X}_i(t), t)\dot{X}_i(t)$. To model viscous drag on the particle due to the membrane, suitable models that depend on the relative velocity can be used. Recall that the gauge-fixing procedure had to be employed to circumvent problems due to the in-plane fluidity of the membrane. A viscoelastic model for the membrane obviates the need for this procedure.

Even though the finite element formulation has been presented for topologically spherical shells, the Lagrangian particle formulation and the search algorithm can be easily adapted to other topologies. The restriction to spherical surfaces is warranted by the gauge-fixing procedure that is employed to deal with fluid shells. As discussed in [27], this procedure is guaranteed to work only for spherical surfaces. However, the formulation presented above can be easily extended to other topologies in the case of solid shells, where material laws that include an in-plane stretching energy would prevent the tangential zero-energy modes typical of fluid shells.

Although in this work we only consider "zero temperature" equilibrium states, where thermal fluctuations are not accounted, the ultimate application of the presented formulation is to study the finite-temperature effects. Indeed, for many biophysical and soft-matter systems at physiologically relevant temperatures, thermal fluctuations play a key role [49,50]. The method presented here—where degrees of freedom are the nodal displacements of the mesh and the particles' positions in the reference configuration—can be easily combined with standard Monte Carlo schemes used to simulate finite temperature systems. As the particles' Monte Carlo moves occur in the reference configuration, locating the elements to which the particles belong is, once again, computationally inexpensive and based on the hash table generated only once for this purpose.

Author Contributions: Conceptualization, L.E.P. and S.D.; software, S.D. and X.W.; validation, L.E.P., S.D. and X.W.; writing—original draft preparation, L.E.P., S.D. and X.W.; writing—review and editing, L.E.P. and S.D.; visualization, L.E.P. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank their respective universities for offering funds to support this research.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Ray-Triangle Intersection Algorithm

Consider the schematic shown in Figure A1. Let a particle be located at P with global coordinates (ϑ, ϕ) and lying on the element ABC, whose vertices have position vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$, respectively. To determine the barycentric coordinates (u, v, w) with respect to the triangle ABC, we adopt the efficient Möller–Trumbore ray-triangle intersection algorithm [51].

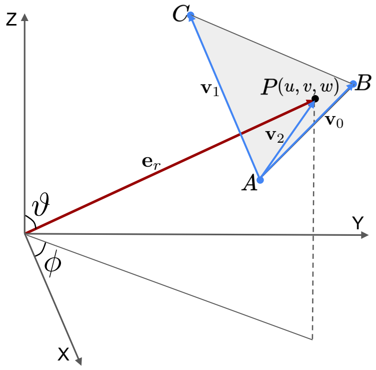

Figure A1. Schematic with key quantities used in the ray-triangle intersection algorithm.

Let the unit radial vector from the origin to P be parametrized as:

$$\mathbf{e}_r = \langle \cos\phi \sin\vartheta, \sin\phi \sin\vartheta, \cos\vartheta \rangle \,.$$

The edges of the triangle are given by:

$$\mathbf{v}_0 = (\mathbf{b} - \mathbf{a})\,, \quad \mathbf{v}_1 = (\mathbf{c} - \mathbf{a})\,,$$

and the normal vector to the triangle is:

$$\mathbf{n} = \mathbf{v}_0 \times \mathbf{v}_1 \,.$$

In addition, let us define the following variables:

$$t := \frac{\mathbf{a} \cdot \mathbf{n}}{\mathbf{e}_r \cdot \mathbf{n}}\,;$$

$$\mathbf{p} := t\mathbf{e}_r\,;$$

$$\mathbf{v}_2 := \mathbf{p} - \mathbf{a} = t\mathbf{e}_r - \mathbf{a}\,;$$

$$d_{00} := \mathbf{v}_0 \cdot \mathbf{v}_0\,;$$

$$d_{11} := \mathbf{v}_1 \cdot \mathbf{v}_1\,;$$

$$d_{20} := \mathbf{v}_2 \cdot \mathbf{v}_0\,;$$

$$d_{21} := \mathbf{v}_2 \cdot \mathbf{v}_1\,.$$

The local barycentric coordinates are then given by:

$$v = \frac{d_{11}d_{20} - d_{01}d_{21}}{(d_{00}d_{11} - d_{01}^2)}\,;$$

$$w = \frac{d_{00}d_{21} - d_{01}d_{20}}{(d_{00}d_{11} - d_{01}^2)}\,;$$

$$u = 1 - v - w\,.$$

References

1. Dharmavaram, S.; She, S.B.; Lázaro, G.; Hagan, M.F.; Bruinsma, R. Gaussian curvature and the budding kinetics of enveloped viruses. *PLoS Comput. Biol.* **2019**, *15*, e1006602. [CrossRef] [PubMed]
2. Oh, N.; Park, J.H. Endocytosis and exocytosis of nanoparticles in mammalian cells. *Int. J. Nanomed.* **2014**, *9*, 51.
3. Vitelli, V.; Lucks, J.B.; Nelson, D.R. Crystallography on curved surfaces. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 12323–12328. [CrossRef]
4. Šarić, A.; Cacciuto, A. Self-assembly of nanoparticles adsorbed on fluid and elastic membranes. *Soft Matter* **2013**, *9*, 6677–6695. [CrossRef]
5. Bradley, L.C.; Chen, W.H.; Stebe, K.J.; Lee, D. Janus and patchy colloids at fluid interfaces. *Curr. Opin. Colloid Interface Sci.* **2017**, *30*, 25–33. [CrossRef]
6. Brant, J.A.; Childress, A.E. Colloidal adhesion to hydrophilic membrane surfaces. *J. Membr. Sci.* **2004**, *241*, 235–248. [CrossRef]
7. Dubertret, B.; Heine, T.; Terrones, M. The rise of two-dimensional materials. *Accounts Chem. Res.* **2015**, *48*, 1–2. [CrossRef]
8. Wood, M. Colloidal lithography and current fabrication techniques producing in-plane nanotopography for biological applications. *J. R. Soc. Interface* **2007**, *4*, 1–17. [CrossRef]
9. Colson, P.; Henrist, C.; Cloots, R. Nanosphere lithography: A powerful method for the controlled manufacturing of nanomaterials. *J. Nanomater.* **2013**, *2013*, 21. [CrossRef]
10. Wang, Y.; Zhang, M.; Lai, Y.; Chi, L. Advanced colloidal lithography: From patterning to applications. *Nano Today* **2018**, *22*, 36–61. [CrossRef]
11. Irvine, W.T.; Vitelli, V.; Chaikin, P.M. Pleats in crystals on curved surfaces. *Nature* **2010**, *468*, 947. [PubMed]
12. Irvine, W.T.; Bowick, M.J.; Chaikin, P.M. Fractionalization of interstitials in curved colloidal crystals. *Nat. Mater.* **2012**, *11*, 948. [PubMed]
13. García, N.A.; Register, R.A.; Vega, D.A.; Gómez, L.R. Crystallization dynamics on curved surfaces. *Phys. Rev. E* **2013**, *88*, 012306. [CrossRef]

14. Meng, G.; Paulose, J.; Nelson, D.R.; Manoharan, V.N. Elastic instability of a crystal growing on a curved surface. *Science* **2014**, *343*, 634–637. [CrossRef]
15. Manoharan, V.N. Colloidal matter: Packing, geometry, and entropy. *Science* **2015**, *349*, 1253751. [CrossRef]
16. Brojan, M.; Terwagne, D.; Lagrange, R.; Reis, P.M. Wrinkling crystallography on spherical surfaces. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 14–19.
17. Stoop, N.; Dunkel, J. Defect formation dynamics in curved elastic surface crystals. *Soft Matter* **2018**, *14*, 2329–2338. [CrossRef]
18. Bausch, A.; Bowick, M.J.; Cacciuto, A.; Dinsmore, A.; Hsu, M.; Nelson, D.; Nikolaides, M.; Travesset, A.; Weitz, D. Grain boundary scars and spherical crystallography. *Science* **2003**, *299*, 1716–1718.
19. Liu, I.B.; Sharifi-Mood, N.; Stebe, K.J. Capillary assembly of colloids: Interactions on planar and curved interfaces. *Annu. Rev. Condens. Matter Phys.* **2018**, *9*, 283–305.
20. Soligno, G.; Dijkstra, M.; van Roij, R. Self-assembly of cubic colloidal particles at fluid–fluid interfaces by hexapolar capillary interactions. *Soft Matter* **2018**, *14*, 42–60. [CrossRef]
21. Elliott, C.M.; Stinner, B. Modeling and computation of two phase geometric biomembranes using surface finite elements. *J. Comput. Phys.* **2010**, *229*, 6585–6612. [CrossRef]
22. Perotti, L.; Dharmavaram, S.; Klug, W.; Marian, J.; Rudnick, J.; Bruinsma, R. Useful scars: Physics of the capsids of archaeal viruses. *Phys. Rev. E* **2016**, *94*, 012404. [CrossRef]
23. Singh, A.R.; Perotti, L.E.; Bruinsma, R.F.; Rudnick, J.; Klug, W.S. Ground state instabilities of protein shells are eliminated by buckling. *Soft Matter* **2017**, *13*, 8300–8308. [CrossRef]
24. Dharmavaram, S.; Perotti, L.E. A Lagrangian formulation for interacting particles on a deformable medium. *Comput. Methods Appl. Mech. Eng.* **2020**, *364*, 112949. [CrossRef]
25. Ma, L.; Klug, W.S. Viscous regularization and r-adaptive remeshing for finite element analysis of lipid membrane mechanics. *J. Comput. Phys.* **2008**, *227*, 5816–5835. [CrossRef]
26. Feng, F.; Klug, W.S. Finite element modeling of lipid bilayer membranes. *J. Comput. Phys.* **2006**, *220*, 394–408. [CrossRef]
27. Dharmavaram, S. A gauge-fixing procedure for spherical fluid membranes and application to computations. *Comput. Methods Appl. Mech. Eng.* **2021**, *381*, 113849. [CrossRef]
28. Zhao, S.; Healey, T.; Li, Q. Direct computation of two-phase icosahedral equilibria of lipid bilayer vesicles. *Comput. Methods Appl. Mech. Eng.* **2017**, *314*, 164–179. [CrossRef]
29. Deserno, M. Fluid Lipid Membranes–A Primer. 2007. Available online: http://www.cmu.edu/biolphys/deserno/pdf/membrane_theory.pdf (accessed on 24 August 2022).
30. Rangamani, P.; Agrawal, A.; Mandadapu, K.K.; Oster, G.; Steigmann, D.J. Interaction between surface shape and intra-surface viscous flow on lipid membranes. *Biomech. Model. Mechanobiol.* **2013**, *12*, 833–845. [CrossRef]
31. Rangarajan, R.; Gao, H. A finite element method to compute three-dimensional equilibrium configurations of fluid membranes: Optimal parameterization, variational formulation and applications. *J. Comput. Phys.* **2015**, *297*, 266–294. [CrossRef]
32. Baumgart, T.; Hess, S.T.; Webb, W.W. Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension. *Nature* **2003**, *425*, 821–824. [CrossRef]
33. Cirak, F.; Ortiz, M.; Schröder, P. Subdivision surfaces: A new paradigm for thin-shell finite-element analysis. *Int. J. Numer. Methods Eng.* **2000**, *47*, 2039–2072. [CrossRef]
34. Loop, C. Smooth Subdivision Surfaces Based on Triangles. Ph.D. Thesis, University of Utah, Salt Lake City, UT, USA, 1987.
35. Stam, J. Evaluation of loop subdivision surfaces. In Proceedings of the SIGGRAPH'98 CDROM Proceedings, Orlando, FL, USA, 19–24 July 1998.
36. Helfrich, W. Elastic properties of lipid bilayers: Theory and possible experiments. *Z. Für Naturforschung C* **1973**, *28*, 693–703. [CrossRef]
37. Jenkins, J. Static equilibrium of configurations of a model red blood cell membrane. *Biophys. J.* **1973**, *13*, 926–939.
38. Capovilla, R.; Guven, J.; Santiago, J. Deformations of the geometry of lipid vesicles. *J. Phys. A Math. Gen.* **2003**, *36*, 6281. [CrossRef]
39. Elliott, C.M.; Stinner, B. Computation of Two-Phase Biomembranes with Phase Dependent Material Parameters Using Surface Finite Elements. *Commun. Comput. Phys* **2013**, *13*, 325–336. [CrossRef]
40. Healey, T.J.; Dharmavaram, S. Symmetry-breaking global bifurcation in a surface continuum phase-field model for lipid bilayer vesicles. *SIAM J. Math. Anal.* **2017**, *49*, 1027–1059. [CrossRef]
41. Steigmann, D.; Baesu, E.; Rudd, R.E.; Belak, J.; McElfresh, M. On the variational theory of cell-membrane equilibria. *Interfaces Free Bound.* **2003**, *5*, 357–366. [CrossRef]
42. Arnold, D.N.; Rogness, J. Möbius Transformations Revealed. *Notes AMS* **2008**, *55*, 1226–1231.
43. Gu, X.; Wang, Y.; Chan, T.F.; Thompson, P.M.; Yau, S.T. Genus Zero Surface Conformal Mapping and Application to Brain Sufrace Mapping. *IEEE Trans. Med Imaging* **2004**, *23*, 949–958. [CrossRef]
44. Zhu, C.; Byrd, R.H.; Lu, P.; Nocedal, J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.* **1997**, *23*, 550–560. [CrossRef]
45. Stam, J. Exact Evaluation of Subdivision Surfaces. Available online: https://www.dgp.toronto.edu/~stam/reality/Research/SubdivEval/ (accessed on 24 August 2022).
46. Glasser, L.; Every, A. Energies and spacings of point charges on a sphere. *J. Phys. A Math. Gen.* **1992**, *25*, 2473. [CrossRef]

47. Puig-de Morales-Marinkovic, M.; Turner, K.T.; Butler, J.P.; Fredberg, J.J.; Suresh, S. Viscoelasticity of the human red blood cell. *Am. J. Physiol.-Cell Physiol.* **2007**, *293*, C597–C605. [CrossRef] [PubMed]
48. Rahimi, M.; Arroyo, M. Shape dynamics, lipid hydrodynamics, and the complex viscoelasticity of bilayer membranes. *Phys. Rev. E* **2012**, *86*, 011932. [CrossRef]
49. Lin, Y.; Inamdar, M.; Freund, L. The competition between Brownian motion and adhesion in soft materials. *J. Mech. Phys. Solids* **2008**, *56*, 241–250. [CrossRef]
50. Michalet, X.; Bensimon, D.; Fourcade, B. Fluctuating vesicles of nonspherical topology. *Phys. Rev. Lett.* **1994**, *72*, 168. [CrossRef]
51. Möller, T.; Trumbore, B. Fast, minimum storage ray-triangle intersection. *J. Graph. Tools* **1997**, *2*, 21–28. [CrossRef]