Spring 1-21-2017

Fluid Membrane Mechanics

Eduard Benet Cerdà
University of Colorado at Boulder, eduard.benetcerda@colorado.edu

Fluid Membrane Mechanics

by

E. Benet

B.A., Technical University of Catalonia, 2013

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Civil Engineering

2015

This thesis entitled:
Fluid Membrane Mechanics
written by E. Benet
has been approved for the Department of Civil Engineering

_____

Prof. Franck Vernerey

_____

Prof. Ronald Pak

_____

Prof. Bengt Fonberg

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Benet, E. (M.S., Engineering Science)

Fluid Membrane Mechanics

Thesis directed by Prof. Prof. Franck Vernerey

Due to their biological relevance, lipid coated vesicles, such as liposomes, microbubbles or microdroplets are used in many areas of medicine such as targeted drug delivery or ultrasound imaging. Hence, a deep knowledge of the mechanics of those vesicles will provide a better understanding on many processes, aid in the design of new drugs or improve the imaging quality among others.

A vesicle can usually be described as an enclosed fluid body where its interface with the outer environment consists of lipid molecules which are generally assembled in single or double layers. Within those layers, lipids are in constant motion and, from a continuum point of view, the interface can be idealized as a compressible viscous fluid-like film constrained to a two dimensional manifold, which is embedded in a three dimensional space. The stretchability comes from the fact that the spacing between lipids can be increased, or reduced, and the inertial forces are disregarded due to the small length scale of the lipids.

This fluid model of the film behavior is a result of several studies at the molecular level that were able to successfully describe many processes present on a lipid coating, such as phase separation, transport or diffusion, through the individual motion of lipid molecules. Thanks to that, we can use a continuum theory for the mechanics of a fluid coating and describe how this activity on the interface affects the overall behavior of the vesicle. To that end, the object of the project is to develop a multiscale, numerical model of a lipid coated vesicle where its response to external inputs accounts for small scale mechanics of its fluid interface. The model will use a meshfree particle method that will provide a highly accurate description of the membrane while being able to handle large deformations without the hindering steps of creating a mesh or refining it.

Dedication

To Teresa for always encouraging me to give the best of myself.

Acknowledgements

First and foremost, I would like to express my very great gratitude to Pere Balsells and the University of Colorado, who made possible this experience by trusting me with one of their fellowships. Without their help and spirit to keep this program alive, this thesis, and many others, would not have been possible.

I would like to express my very great appreciation to professor Franck Vernerey, my research supervisor, for their patient guidance, enthusiastic encouragement and useful critiques during all my stage at the University of Colorado at Boulder. But above all of this, for the chance of joining his research group, and for introducing me to the biophysical world, which is gripping and full of opportunities. I really value the chance he has given to me, for working in an project like this; I was able to see how a real investigation is done, and taught me to be more ambitious and exigent with my work.

Thanks to all the lab gang for sharing all those everlasting evenings in the office. Your company and always cheerful mood made all the investigation much more pleasant.

I would also like to express my gratitude to the Boulder crew to share with me so many American experiences and for their support in all those moments where not everything was going well.

Thank, of course, to my girlfriend and family, who have always given and unconditional support and that encouraged me to pursue this project, as well as continue my investigation in this same field.

Contents

Chapter

Figures

Figure

Chapter 1

Introduction

The deformability of lipid coated vesicles such as liposomes, microbubbles or microdroplets is of crucial relevance in many processes on our everyday life. For instance, the transport of those particles through porous media is used in the pharmaceutical industry on the design of new filtration membranes for drug purification or blood filtration [30, 19]. Food and fossil fuel industries use it in residual water purification where they need to separate oil emulsions from the water [6]. And in medicine, vesicle deformability has a big role in ultrasound imaging or the development of new drug delivery techniques [42]. Therefore in order to improve all those fields is absolutely necessary to understand the mechanics of those particles: how and why do they deform, what does affect its deformability and how can we used that in our own benefit.

A vesicle can usually be described as an enclosed fluid body where its interface with the outer environment consists of lipid molecules which are generally assembled in single or double layers. Within those layers, lipids are in constant motion and, from a continuum point of view, the interface can be idealized as a compressible viscous fluid-like film constrained to a two dimensional manifold, which is embedded in a three dimensional space. This lipid interface is the responsible agent for the deformation characteristics of all those particles by controlling its stretchability, curvature, impermeability and the interactions with the outer environment among others.

The mathematical modeling of those membranes has been a subject of study for a long time. There are now many different mathematical models to describe different processes where those membranes are involved, such as cell endocytosis [21], phase separation of the membrane lipids

[14], cell deformability [7], etc. Which have been solved using a variety of numerical approaches, like finite elements [14], particle methods [37] or molecular dynamics [32] among others.

However, even though there are many studies and models on fluid interfaces, there are still very few approaches that are able to successfully describe a multiphase fluid flow with an elastic interface and none of them accounts for the viscoelasticity of lipid coated vesicles [2]. Jenny et al. [27, 28] proposed a finite volume approach for the multiphase transport on porous media with fluid-fluid interfaces. Helmig [25] used a multiscale approach to study the effect heterogeneity of porous media in multiphase transports using finite elements. Similarly, Ataie-Ashtiani et al. [3] used homogenization techniques and finite differences to study the transport of two phase flows. Nevertheless, none of these existing methods is able to handle elastic interfaces and they are mostly limited to the study of fluid-fluid and fluid-gas systems.

It was not until most recently when Foucard et al. [18] proposed a method to describe the transport of soft, elastic vesicles through porous media at the microscale. They use a combination of finite elements to describe the fluid flow, coupled with a particle method that solves the elastic membrane equations. However, they are still limited to purely elastic, such as a rubber balloon, or fluid-fluid interfaces driven purely by surface tension, such as an oil droplet in water.

Hence, the objective of this thesis is to develop a numerical model to describe the mechanics of a fluid membrane in a way that improves the current formulations so it can easily be incorporated in the work of Foucard [18]. With this objective, we developed a new approach that it is able to describe a fluid membrane with high accuracy and lower computational cost. Moreover, even though this work is purely in 2D, we tried to a method that it is capable of a 3D representation without major changes on the formulation.

In this thesis, we present a meshfree particle method to solve the governing equations of a fluid membrane. We use a meshless method that eases the future implementation of a 3D membrane where meshing can hinder our solutions, and because it is a good approach to describe the division (or fusion) between to vesicles. The method has been combined with the use of radial basis functions (RBFs) to interpolate the surface and the field described on it. RBFs have proved to be an excellent

tool to solve PDE's in a curved surface with high accuracy [39], and we believe that they are the best tool to represent our surface.

The outline of this thesis is as follows: in section 2 we define the necessary differential geometry needed to describe the kinematics of a fluid interface. Section 3 derives the governing equations for a compressible fluid membrane. In section 4 we describe the numerical method used to solve the system of partial differential equations (PDE's). And sections 5, 6 and 7 cover the full solution of both 1D and 3D axisymmetric fluid membranes.

Chapter 2

Geometrical description of a 2D manifold in a 3D space

The object of this section is to provide a clear and detailed description on the kinematics of a two dimensional (2D) fluid surface in a three dimensional (3D) space. This topic is widely known and is commonly used in many areas such as shell theory [38, 23, 35], where is usually needed to compute the gradient of the velocity on the surface. For completeness we provide here the key mathematical background and notation that are needed for the reminder of this thesis.

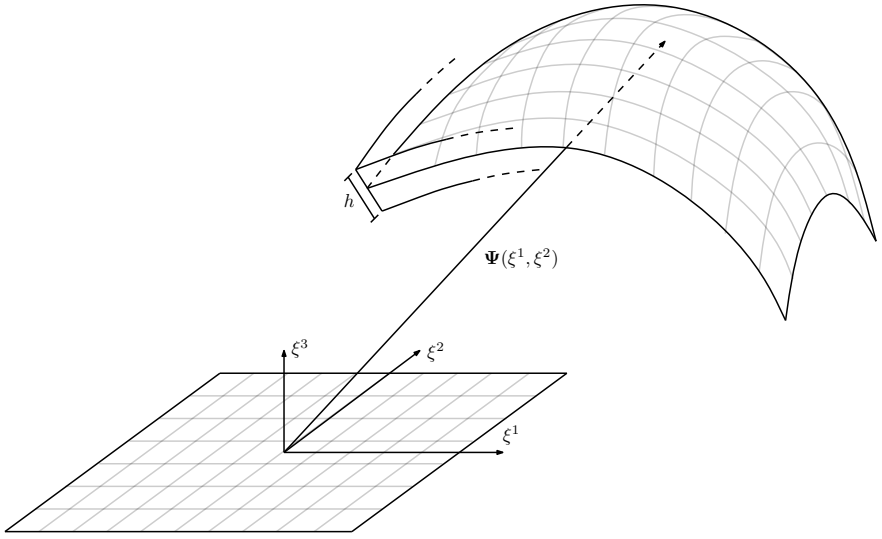


Figure 2.1: *Scheme of a 2D manifold embedded on a 3D space*

2.1 Surface Parametrization

Shells and membranes are three dimensional entities that, because of their smaller thickness compared with the other two dimensions, are usually treated like two dimensional surfaces existing

in a three dimensional space. Also, it is usually assumed that we are dealing with Kirchoff-Love membranes, where a fiber which is initially normal to the middle surface of the shell remains normal to the deformed middle surface, which is reasonable as long as the hypothesis of small thickness remains true.

Due to the fact that membranes are flat surfaces, every variable or field can be parametrized in a 2D parametric space (figure 2.1), which makes much easier to deal with them. However, because of the curvature of the real space, every calculation such as gradients, divergences or order fields must be redefined. This section will provide the necessary tools in order to operate with any field within the membrane, as well as establish a nomenclature convention that will be followed through all the thesis.

Mathematically speaking, a 2D manifold has zero thickness in its normal direction. However that is not true for a real membrane where the thickness can play an important role in may different scenarios which might be important in future problems. In order to avoid this, many efforts have been done to capture the relevance of the third coordinate [31, 40], and develop a geometrically exact description of a membrane withouth overcomplicating the mathematics of the system. A generic flat surface in curvilinear coordinates ξ^1, ξ^2 can be described as:

$$\Psi = \boldsymbol{\Psi}(\xi^1, \xi^2). \tag{2.1}$$

However, using this equation does not allow us to capture the physics behind the thickness of the membrane. Therefore, we extend the formulation to another one where Ψ becomes the mid surface of a thin membrane of thickness h such that:

$$\boldsymbol{X}(\xi^1, \xi^2, \xi^3) = \boldsymbol{\Psi}(\xi^1,\ \xi^2) + \xi^3 \hat{\boldsymbol{n}}(\xi^1, \xi^2), \tag{2.2}$$

where the first term characterizes the membrane mid surface, $\hat{\boldsymbol{n}}$ is the normal to it and ξ^3 is a value between $-h/2$ and $h/2$. Next, we define the tangent vectors on the mid surface and the normal just as it is done in regular shell theory:

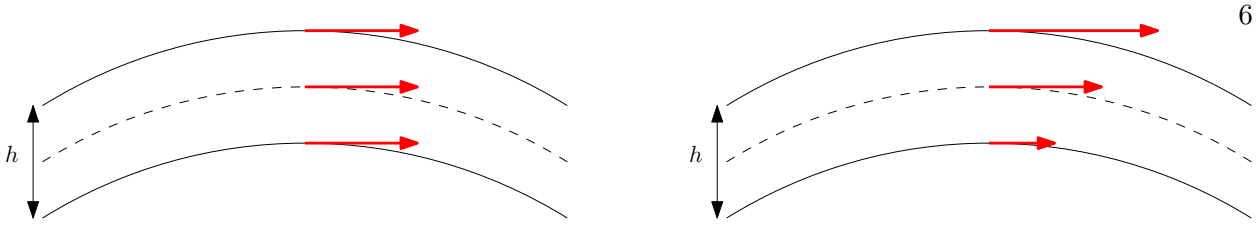


Figure 2.2: *On the left, the velocity field is constant across the surface, which means that velocity gradient will show an out of plane deformation. On the right, the same velocity field accounting for the normal coordinate ensures that the membrane will not suffer shear deformations unless specifically defined.*

$$\boldsymbol{a}_\alpha = \frac{\partial \boldsymbol{\Psi}}{\partial \xi^\alpha} = \boldsymbol{\Psi}_{,\alpha} \tag{2.3}$$

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{a}_1 \times \boldsymbol{a}_2}{\|\boldsymbol{a}_1 \times \boldsymbol{a}_2\|}, \tag{2.4}$$

where from now on we will use Greek indices for values ranging from 1 to 2, and Latin indices to go from 1 to 3. We can also define the local covariant basis in 3D, which physically represents the tangent vectors at any parallel surface:

$$\boldsymbol{g}_\alpha = \frac{\partial \boldsymbol{X}}{\partial \xi^\alpha} = \boldsymbol{a}_\alpha + \xi^3 \hat{\boldsymbol{n}}_{,\alpha} \tag{2.5}$$

$$\boldsymbol{g}_3 = \frac{\partial \boldsymbol{X}}{\partial \xi^3} = \hat{\boldsymbol{n}}. \tag{2.6}$$

On the midplane of the membrane, the vectors 2.3 and 2.4 are no different from 2.4 and 2.5. However, as shown in figure 2.2, if one moves outside the membrane the latter set of vectors is able to capture the fact that the normal must remain normal after deformation. All this might seem useless for now since we are only concerned about the results on the mid plane, but in further calculations we will see how the derivatives are indeed affected by this modification.

From this, we can also define the first fundamental form of the surface, or the covariant metric tensor, in the mid plane and in the whole membrane:

$$g_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j \qquad a_{\alpha\beta} = \boldsymbol{a}_\alpha \cdot \boldsymbol{a}_\beta. \tag{2.7}$$

And the contravariant basis, which is the basis perpendicular to the natural one:

$$\boldsymbol{g}^i \cdot \boldsymbol{g}_j = \delta^j_i \qquad \boldsymbol{a}^\alpha \cdot \boldsymbol{a}_\beta = \delta^\beta_\alpha \tag{2.8}$$

$$g^{ij} = \boldsymbol{g}^i \cdot \boldsymbol{g}^j \qquad a^{\alpha\beta} = \boldsymbol{a}^\alpha \cdot \boldsymbol{a}^\beta. \tag{2.9}$$

The derivative of these basis can be computed as:

$$\boldsymbol{g}_{i,j} = \Gamma^k_{ij} \boldsymbol{g}_k \tag{2.10}$$

$$\boldsymbol{g}^i_{,j} = -\Gamma^k_{ij} \boldsymbol{g}^k, \tag{2.11}$$

where Γ^k_{ij} are the components of the Christoffel symbol of the second kind, which takes into account the change in the curvilinear basis due to the deformation of the membrane and can be computed as:

$$\Gamma_{\alpha\beta\gamma} = \frac{1}{2} \left(g_{\alpha\gamma,\beta} + g_{\beta\gamma,\alpha} - g_{\alpha\beta,\gamma} \right) \tag{2.12}$$

$$\Gamma^\lambda_{\alpha\beta} = g^{\lambda\gamma} \Gamma_{\alpha\beta\gamma}. \tag{2.13}$$

Next, we define the second fundamental form of the surface, or curvature tensor, in its covariant, contravariant and mixed forms:

$$\kappa_{\alpha\beta} = \kappa_{\beta\alpha} = -\boldsymbol{a}_\alpha \cdot \hat{\boldsymbol{n}}_{,\beta} = \hat{\boldsymbol{n}} \cdot \boldsymbol{a}_{\alpha,\beta} \tag{2.14}$$

$$\kappa^{\alpha\beta} = a^{\alpha\mu} a^{\lambda\beta} \kappa_{\mu\lambda} \tag{2.15}$$

$$\kappa_{\beta}^{\alpha} = a^{\alpha\mu}\kappa_{\mu\beta}. \tag{2.16}$$

Note that this last definition has only been done for the mid plane of the membrane, and this equations are not valid outside of it. In this document we will not pursue further details on the description of the manifold since it is not necessary for the present work. For further study refer to [41].

Having described the surface, the next step consist in defining how we operate with different scalar and vector fields on the surface. Because our surface is no longer flat and our basis vectors not orthonormal, all the operators such as the gradient or the divergence must be redefined.

2.2 Gradient of a Scalar Field

Starting with the calculation of the gradient of a scalar field in general curvilinear coordinates, we obtain:

$$\nabla\phi = \frac{\partial\phi}{\partial x^i}\boldsymbol{g}^i = \phi_{,i}\boldsymbol{g}^i \qquad i = 1, 2, 3. \tag{2.17}$$

But since this we are not considering the third coordinate this can be rewritten as:

$$\nabla\phi = \frac{\partial\phi}{\partial x^\alpha}\boldsymbol{g}^\alpha = \phi_{,\alpha}\boldsymbol{g}^\alpha \qquad \alpha = 1, 2. \tag{2.18}$$

In a physical sense, this two equations represent the same variation. However, while the first one is doing it in a volumetric sense, the second one is only taking care of the gradient on the surface and it can be interpreted as the surface projection of 2.17

2.3 Divergence a Vector Field

We define a vector field $\boldsymbol{u}$ in curvilinear coordinates as:

$$\boldsymbol{u} = u^i\boldsymbol{g}_i, \tag{2.19}$$

and its divergence can be computed with the expression bellow:

$$\nabla \cdot \boldsymbol{u} = \left(\boldsymbol{g}^k \frac{\partial}{\partial x^k} \right) \cdot (u_i \boldsymbol{g}_i) = u^i_{,i} + u^i \Gamma^k_{ik}. \tag{2.20}$$

Note that this definition only applies the chain rule on the first equation. The first term accounts for the derivative of the local components of the vector u respect to the curvilinear coordinates and Γ^k_{ik} considers the variation on the basis vector. This definition can be extrapolated to calculate the divergence of a tensor field $\boldsymbol{p}$ as:

$$\nabla \cdot \boldsymbol{p} = \left(\boldsymbol{g}^k \frac{\partial}{\partial x^k} \right) \cdot (p^{ij} \boldsymbol{g}_i \boldsymbol{g}_j) = \left(p^{ki}_{,k} + p^{ij} \Gamma^k_{jk} + p^{kj} \Gamma^i_{jk} \right) \boldsymbol{g}_i. \tag{2.21}$$

2.4 Gradient of a Vector Field

Finally the last step is to calculate the gradient of a vector field on the surface. With this we will be able to calculate the velocity gradient at any point on the surface which is a very important result for the remainder of this thesis. The general equation to calculate the (volumetric) gradient of a vector is:

$$\nabla \boldsymbol{u} = \frac{\partial u^i \boldsymbol{g}_i}{\partial \xi^j} \otimes \boldsymbol{g}^j = u^i \big|_j \, \boldsymbol{g}_i \otimes \boldsymbol{g}^j \qquad i, j = 1, 2, 3, \tag{2.22}$$

where the first term is called the covariant derivative of $\boldsymbol{v}$:

$$u^i \big|_j = \frac{\partial u^i}{\partial \xi^j} + \Gamma^i_{jk} u^k. \tag{2.23}$$

2.4.1 Gradient of a velocity field

To illustrate this result, and due to its relevance on further calculations, we will compute the gradient of a velocity field on a surface. Consider a vector field separated in tangential and normal components:

$$\boldsymbol{v} = v^{\alpha}\boldsymbol{g}_{\alpha} + v^{3}\hat{\boldsymbol{n}} = \boldsymbol{v}^{t} + \boldsymbol{v}^{n}. \tag{2.24}$$

If we calculate the gradient of this vector, we obtain:

$$\nabla\boldsymbol{v} = \frac{\partial \boldsymbol{v}^{t}}{\partial \xi^{\beta}} \otimes \boldsymbol{g}^{\beta} + \frac{\partial \boldsymbol{v}^{t}}{\partial \xi^{3}} \otimes \hat{\boldsymbol{n}} + \frac{\partial \boldsymbol{v}^{n}}{\partial \xi^{\beta}} \otimes \boldsymbol{g}^{\beta} + \frac{\partial \boldsymbol{v}^{n}}{\partial \xi^{3}} \otimes \hat{\boldsymbol{n}} \tag{2.25}$$

and then, analysing term by term we get:

$$\begin{aligned}
\boldsymbol{v}^{t}_{,\beta} \otimes \boldsymbol{g}^{\beta} &= \left(v^{\alpha}\boldsymbol{g}_{\alpha}\right)_{,\beta} \otimes \boldsymbol{g}^{\beta} = \left(v^{\alpha}_{,\beta}\boldsymbol{g}_{\alpha} + v^{\alpha}\boldsymbol{g}_{\alpha,\beta}\right) \otimes \boldsymbol{g}^{\beta} \\
&= v^{\alpha}_{,\beta}\left(\boldsymbol{g}_{\alpha} \otimes \boldsymbol{g}^{\beta}\right) + v^{\alpha}\left(\boldsymbol{a}_{\alpha} + \xi^{3}\hat{\boldsymbol{n}}_{,\alpha}\right)_{,\beta} \otimes \boldsymbol{g}^{\beta} \\
&= v^{\alpha}_{,\beta}\left(\boldsymbol{g}_{\alpha} \otimes \boldsymbol{g}^{\beta}\right) + \left(v^{\alpha}\Gamma^{\lambda}_{\alpha\beta}\boldsymbol{a}_{\lambda} + v^{\alpha}b_{\alpha\beta}\hat{\boldsymbol{n}}\right) \otimes \boldsymbol{g}^{\beta} + \left(\xi^{3}\left(\hat{\boldsymbol{n}}_{,\alpha}\right)_{,\beta}\right) \otimes \boldsymbol{g}^{\beta} \\
&= v^{\alpha}_{,\beta}\left(\boldsymbol{a}_{\alpha} \otimes \boldsymbol{g}^{\beta}\right) + v^{\alpha}_{,\beta}\xi^{3}\left(\hat{\boldsymbol{n}}_{,\alpha} \otimes \boldsymbol{g}^{\beta}\right) + v^{\lambda}\Gamma^{\alpha}_{\lambda\beta}\left(\boldsymbol{a}_{\alpha} \otimes \boldsymbol{g}^{\beta}\right) \\
&\quad + v^{\alpha}b_{\alpha\beta}\left(\hat{\boldsymbol{n}} \otimes \boldsymbol{g}^{\beta}\right) + \xi^{3}\left(\left(\hat{\boldsymbol{n}}_{,\alpha}\right)_{,\beta}\right) \otimes \boldsymbol{g}^{\beta}
\end{aligned} \tag{2.26}$$

$$\begin{aligned}
\boldsymbol{v}^{t}_{,3} \otimes \hat{\boldsymbol{n}} &= \left(v^{\alpha}\boldsymbol{g}_{\alpha}\right)_{,3} \otimes \hat{\boldsymbol{n}} = \left(v^{\alpha}_{,3}\boldsymbol{g}_{\alpha} + v^{\alpha}\boldsymbol{g}_{\alpha,3}\right) \otimes \hat{\boldsymbol{n}} \\
&= v^{\alpha}_{,3}\left(\boldsymbol{g}_{\alpha} \otimes \hat{\boldsymbol{n}}\right) + v^{\alpha}\left(\hat{\boldsymbol{n}}_{,\alpha} \otimes \hat{\boldsymbol{n}}\right) \\
&= v^{\alpha}_{,3}\left(\boldsymbol{a}_{\alpha} \otimes \hat{\boldsymbol{n}}\right) + v^{\alpha}_{,3}\xi^{3}\left(\hat{\boldsymbol{n}}_{,\alpha} \otimes \hat{\boldsymbol{n}}\right) + v^{\alpha}\left(\hat{\boldsymbol{n}}_{,\alpha} \otimes \hat{\boldsymbol{n}}\right)
\end{aligned} \tag{2.27}$$

$$\begin{aligned}
\boldsymbol{v}^{n}_{,\beta} \otimes \boldsymbol{g}^{\beta} &= \left(v^{3}\hat{\boldsymbol{n}}\right)_{,\beta} \otimes \boldsymbol{g}^{\beta} = \left(v^{3}_{,\beta}\hat{\boldsymbol{n}} + v^{3}\hat{\boldsymbol{n}}_{,\beta}\right) \otimes \boldsymbol{g}^{\beta} = \\
&= v^{3}_{,\beta}\left(\hat{\boldsymbol{n}} \otimes \boldsymbol{g}^{\beta}\right) - v^{3}b^{\alpha}_{\beta}\left(\boldsymbol{a}_{\alpha} \otimes \boldsymbol{g}^{\beta}\right)
\end{aligned} \tag{2.28}$$

$$\boldsymbol{v}^{n}_{,3} \otimes \hat{\boldsymbol{n}} = \left(v^{3}\hat{\boldsymbol{n}}\right)_{,3} \otimes \hat{\boldsymbol{n}} = v^{3}_{,3}\left(\hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}}\right). \tag{2.29}$$

so the gradient of a vector field can finally be rewritten as:

$$\begin{aligned}\nabla \boldsymbol{v} &= \left(v^{\alpha}|_{\beta} - b^{\alpha}_{\beta} v^{3}\right)\left(\boldsymbol{a}_{\alpha} \otimes \boldsymbol{g}^{\beta}\right) + \left(v^{3}_{,\beta} + b_{\alpha\beta} v^{\alpha}\right)\left(\hat{\boldsymbol{n}} \otimes \boldsymbol{g}^{\beta}\right) \\ &\quad - v^{\lambda} b_{\lambda\alpha}\left(\boldsymbol{a}^{\alpha} \otimes \hat{\boldsymbol{n}}\right) + v^{\alpha}_{,3}\left(\boldsymbol{a}_{\alpha} \otimes \hat{\boldsymbol{n}}\right) + v^{3}_{,3}\left(\hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}}\right) \\ &\quad + \xi^{3}\left[v^{\alpha}_{,\beta}\left(\hat{\boldsymbol{n}}_{,\alpha} \otimes \boldsymbol{g}^{\beta}\right) + \left(\left(\hat{\boldsymbol{n}}_{,\alpha}\right)_{,\beta}\right) \otimes \boldsymbol{g}^{\beta} + \hat{\boldsymbol{n}}_{,\alpha} \otimes \hat{\boldsymbol{n}}\right].\end{aligned} \tag{2.30}$$

Then, if we consider only the mid section ($\xi^3 = 0$ and $\boldsymbol{g}^{\alpha} = \boldsymbol{a}^{\alpha}$) we obtain the following result:

$$\begin{aligned}\nabla \boldsymbol{v}|_{\xi^3=0} &= \left(v^{\alpha}|_{\beta} - b^{\alpha}_{\beta} v^{3}\right)\left(\boldsymbol{a}_{\alpha} \otimes \boldsymbol{a}^{\beta}\right) + \left(v^{3}_{,\beta} + b_{\alpha\beta} v^{\alpha}\right)\left(\hat{\boldsymbol{n}} \otimes \boldsymbol{a}^{\beta}\right) \\ &\quad - v^{\lambda} b_{\lambda\alpha}\left(\boldsymbol{a}^{\alpha} \otimes \hat{\boldsymbol{n}}\right) + v^{\alpha}_{,3}\left(\boldsymbol{a}_{\alpha} \otimes \hat{\boldsymbol{n}}\right) + v^{3}_{,3}\left(\hat{\boldsymbol{n}} \otimes \hat{\boldsymbol{n}}\right),\end{aligned} \tag{2.31}$$

which is the same that was derived by Pop et al. [35] plus three additional terms accounting for the variation of all velocities across the surface as well as the change of magnitude of the tangent vector across it. Those terms are the ones capturing the physics of a thin, Kirchoff-Love membrane ensuring that a normal fiber to the membrane remains normal after deformation. They will also allows us to define a cross section gradient on any field in case that is needed in the future.

The remaining terms represent the variation in normal and tangential velocity on the membrane. The first two account for the variation in normal velocity, which is composed of the pure tangential variation along a curve $v^{\alpha}|_{\beta}$ plus a correction on the curvature induced by the normal velocity $-b^{\alpha}_{\beta} v^{3}$. In a similar way, the term $v^{3}_{,\beta}$ accounts for the variation in normal velocity along the curve and the term $b_{\alpha\beta} v^{\alpha}$ is correcting it by incorporating the change in curvature on the system.

Chapter 3

Governing equations

In this section the objective is to characterize a compressible fluid membrane by defining the set of equations controlling its behavior. Membranes are 3D bodies where two of their dimensions are much larger that the third one, making them look like a 2D body embedded in a 3D space. Due to that, the governing equations of the membranes are in general simplified by assuming that they are infinitely thin.

As any other 3D body, the mechanical response of a membrane can be calculated by determining the balance of mass and the balance of forces. However, with the objective reducing the total amount of degrees of freedom, we will derive them by using the geometrical description described before. Moreover, we will only assume that the membrane is a flat surface after we derived all the equations, this way we can always recover the full three dimensionality of the problem.

Fluid membranes are only different from a regular solid membrane by the fact that the molecules on its surface are able to move giving the surface the properties of a fluid. Therefore, there will be no difference in the development of the regular governing equations, and their fluid nature will come in the constitutive equations

3.1 Balance of Mass

Consider a body with initial configuration Ω_0 and density $\rho_0(\boldsymbol{x}, 0)$, that is deformed into a current configuration Ω and density $\rho(\boldsymbol{x}, t)$. Its mass can be computed as:

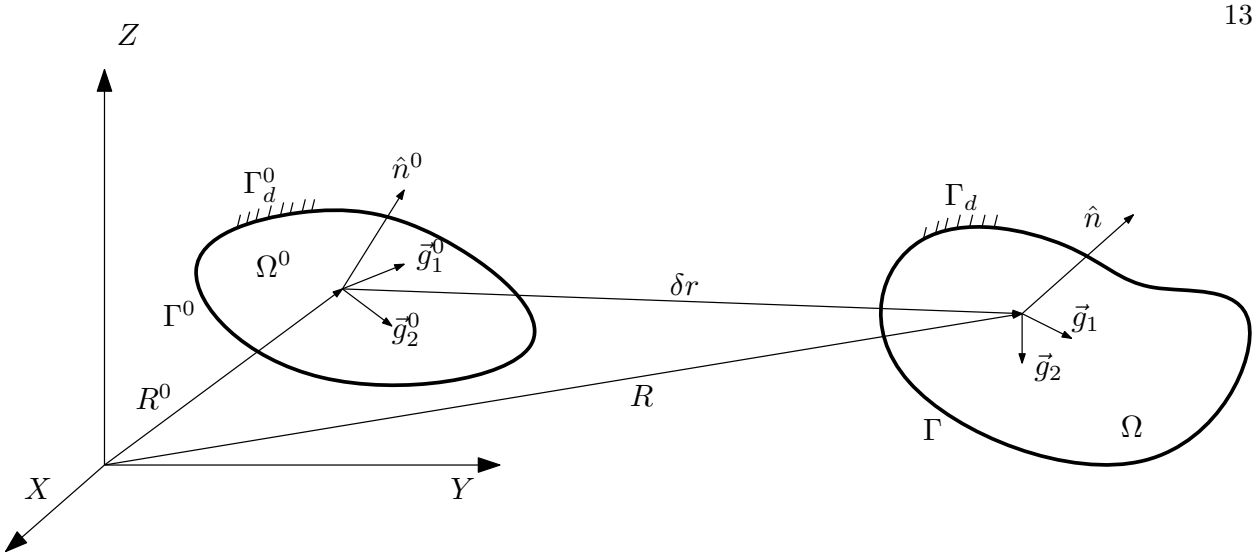


Figure 3.1: *We define a local basis* $\vec{\boldsymbol{g}}_1^0, \vec{\boldsymbol{g}}_2^0, \hat{\boldsymbol{n}}$ *on a body* Ω^0 *surrounded by a boundary* Γ^0*, with Dirichlet boundary conditions applied on* Γ_d^0*. This body is deformed by a vector* $\vec{r}+\delta\vec{r}$ *to a deformed configuration* Ω, Γ, Γ_d

$$m = \int_{\Omega} \rho(\boldsymbol{x}, t) dv = \int_{\Omega_0} \rho(\boldsymbol{x}, t) J dV. \tag{3.1}$$

Taking the material time derivative of the mass we obtain the net flux of mass in the body $\gamma(\boldsymbol{x}, t)$:

$$\frac{Dm}{Dt} = \gamma(\boldsymbol{x}) = \int_{\Omega_0} \left[\frac{D\rho}{Dt} J + \rho \frac{DJ}{Dt} \right] dV. \tag{3.2}$$

where the Jacobian is the change in area, or the determinant of the deformation gradient tensor $\boldsymbol{F}$:

$$\frac{DJ}{Dt} = \frac{\partial J}{\partial F_{iI}} \frac{\partial F_{iI}}{\partial t} = \frac{\partial J}{\partial \boldsymbol{F}} : \dot{\boldsymbol{F}}. \tag{3.3}$$

Considering then the following relationship [26]:

$$\frac{\partial \det(A)}{\partial A} = \det(A) A^{-T}, \tag{3.4}$$

the material time derivative of the Jacobian becomes:

$$\frac{DJ}{Dt} = J\boldsymbol{F}^{-T} : \boldsymbol{L}\boldsymbol{F} = JF^{-1}_{iI}L_{ij}F_{jI} = J\delta_{ij}L_{ij} = JL_{ii} = J\boldsymbol{1} : \boldsymbol{L} = J\mathrm{tr}(\boldsymbol{L}), \tag{3.5}$$

so the balance of mass results:

$$\frac{D\rho}{Dt} + \rho\mathrm{tr}(\boldsymbol{L}) = \gamma(\boldsymbol{x}). \tag{3.6}$$

According to section 2.4.1, the velocity gradient on the membrane is:

$$\begin{aligned} \boldsymbol{L} &= \left(v^{\alpha}|_{\beta} - \kappa^{\alpha}_{\beta}v^{3}\right)\left(\boldsymbol{a}_{\alpha}\otimes\boldsymbol{a}^{\beta}\right) + \left(v^{3}_{,\beta} + \kappa_{\alpha\beta}v^{\alpha}\right)\left(\hat{\boldsymbol{n}}\otimes\boldsymbol{a}^{\beta}\right) \\ &\quad -v^{\lambda}\kappa_{\lambda\alpha}\left(\boldsymbol{a}^{\alpha}\otimes\hat{\boldsymbol{n}}\right) + v^{\alpha}_{,3}\left(\boldsymbol{a}_{\alpha}\otimes\hat{\boldsymbol{n}}\right) + v^{3}_{,3}\left(\hat{\boldsymbol{n}}\otimes\hat{\boldsymbol{n}}\right), \end{aligned} \tag{3.7}$$

and therefore the divergence becomes:

$$tr(\boldsymbol{L}) = v^{\alpha}|_{\alpha} - v^{3}\kappa^{\alpha}_{\alpha} + v^{3}_{,3}, \tag{3.8}$$

so we finally obtain the full balance of mass on the membrane:

$$\boxed{\frac{D\rho}{Dt} + \rho\left(v^{\alpha}|_{\alpha} - v^{3}\kappa^{\alpha}_{\alpha} + v^{3}_{,3}\right) = \gamma(\xi^{1},\xi^{2},\xi^{3})} \tag{3.9}$$

Note that if we do not consider the change in thickness we recover the equation from Marsden and Hughes [34] of the balance of mass for a 2D manifold.

3.2 Balance of Linear Momentum

The linear momentum of a body is defined as:

$$\boldsymbol{L}(t) \overset{\text{def}}{=} \int_{\Omega_0} \rho(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)JdV, \tag{3.10}$$

and therefore its variation in time is:

$$\frac{D\boldsymbol{L}}{Dt} = \int_{\Omega}\boldsymbol{b}dv + \int_{\Gamma}\boldsymbol{t}da, \tag{3.11}$$

where $\boldsymbol{b} = b^{\alpha}\boldsymbol{g}_{\alpha} + b^{3}\hat{\boldsymbol{n}}$ are the body forces and $\boldsymbol{t}$ are the external forces applied on the surface. Then, applying the chain rule to the LHS we get:

$$\frac{D\boldsymbol{L}}{Dt} = \left[\frac{D}{Dt}(\rho J)\boldsymbol{v} + \rho J\frac{D}{Dt}\boldsymbol{v}\right] dV, \tag{3.12}$$

and introducing the balance of mass we can write the first term as:

$$\int_{\Omega_0} \frac{D}{Dt}(\rho J)dV = \int_{\Omega_0} \left(\frac{D\rho}{Dt} + \rho \mathrm{tr}(\boldsymbol{L})\right) JdV = 0, \tag{3.13}$$

so the final expression for the balance of linear momentum is:

$$\frac{D\boldsymbol{L}}{Dt} = \int_{\Omega_0} [\rho\boldsymbol{a}]\, JdV. \tag{3.14}$$

Going back to the RHS of 3.11 and using the divergence theorem we get:

$$\int_{\Gamma} t^{i}\boldsymbol{g}_{i}da = \int_{\Gamma} \sigma^{ij}n_{j}\boldsymbol{g}_{i}da = \int_{\Omega} \left(\frac{\partial\left(\sigma^{ij}\boldsymbol{g}_{i}\boldsymbol{g}_{j}\right)}{\partial\xi^{k}} \cdot \boldsymbol{g}^{k}\right) dv. \tag{3.15}$$

Note that in curvilinear coordinates if we expand the divergence of the stress tensor it is easy to prove that:

$$\frac{\partial\left(\sigma^{ij}\boldsymbol{g}_{i}\boldsymbol{g}_{j}\right)}{\partial\xi^{k}} \cdot \boldsymbol{g}^{k} = \sigma^{ij}\big|_{j}\,\boldsymbol{g}_{i}. \tag{3.16}$$

Therefore the balance of linear momentum in curvilinear coordinates reads:

$$\rho\boldsymbol{a} = \boldsymbol{b} + \mathrm{div}(\boldsymbol{\sigma}) \qquad \rho a^{i} = b^{i} + \sigma^{ij}|_{j}, \tag{3.17}$$

or

$$\rho\left(\frac{\partial\boldsymbol{v}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{v}\right) = \boldsymbol{b} + \mathrm{div}(\boldsymbol{\sigma}). \tag{3.18}$$

This is valid for any 3D solid, however, this is a membrane therefore is convenient to simplify the problem using a 2D surface on the mid plane of the membrane. In doing so, we must define the

resultant of the stresses across the thickness of the membrane that will be applied on the surface. For the sake of simplicity, we only consider the zeroth and first moments of the stress, defined as follows:

$$\bar{\sigma}^{ij}\boldsymbol{a}_i\boldsymbol{a}_j = \int_{-h/2}^{h/2} \sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j d\xi \tag{3.19}$$

$$\bar{m}^{ij}\boldsymbol{a}_i\boldsymbol{a}_j = \int_{-h/2}^{h/2} \xi\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j d\xi. \tag{3.20}$$

Then, for the $\underline{0^{th}\text{ moment}}$ equation we get:

$$\int_{-h/2}^{h/2} (\sigma^{ij}|_j + b^i)\boldsymbol{g}_i d\xi^3 = \int_{-h/2}^{h/2} \left(\frac{\partial\left(\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j\right)}{\partial\xi^k} \cdot \boldsymbol{g}^k + b^i\boldsymbol{g}_i \right) d\xi^3 = 0. \tag{3.21}$$

The divergence of the stress tensor can be expanded by separating the normal and tangential components of the derivatives:

$$\frac{\partial\left(\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j\right)}{\partial\xi^k} \cdot \boldsymbol{g}^k = \frac{\partial\left(\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j\right)}{\partial\xi^\alpha} \cdot \boldsymbol{g}^\alpha + \frac{\partial\left(\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j\right)}{\partial\xi^3} \cdot \boldsymbol{g}^3. \tag{3.22}$$

Therefore the integral can be rewritten as:

$$\int_{-h/2}^{h/2} \frac{\partial\left(\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j\right)}{\partial\xi^\alpha} \cdot \boldsymbol{g}^\alpha d\xi^3 + \int_{-h/2}^{h/2} \frac{\partial\left(\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j\right)}{\partial\xi^3} \cdot \hat{\boldsymbol{n}} d\xi^3 + \int_{-h/2}^{h/2} b^i\boldsymbol{g}_i d\xi^3 = 0. \tag{3.23}$$

Pulling out the derivative of the integral, and integrating the second term we obtain:

$$\frac{\partial}{\partial\xi^\gamma}\left(\int_{-h/2}^{h/2} \sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j d\xi^3 \right) \cdot \boldsymbol{a}^\gamma + \sigma^{i3}\boldsymbol{g}_i\Big|_{h^-}^{h^+} + \int_{-h/2}^{h/2} b^i\boldsymbol{g}_i d\xi^3 = 0, \tag{3.24}$$

so going back to the integral and using the definition of the resultant 3.19 it becomes:

$$\frac{\partial(\bar{\sigma}^{ij}\boldsymbol{a}_i\boldsymbol{a}_j)}{\partial\xi^\gamma} \cdot \boldsymbol{a}^\gamma + \sigma^{i3}\boldsymbol{g}_i\Big|_{h^-}^{h^+} + \int_{-h/2}^{h/2} b^i\boldsymbol{g}_i d\xi^3 = 0. \tag{3.25}$$

Then we consider the derivatives of the first term:

$$\begin{aligned}
\frac{\partial(\bar{\sigma}^{\alpha\beta}\boldsymbol{a}_\alpha\boldsymbol{a}_\beta)}{\partial\xi^\gamma}\cdot\boldsymbol{a}^\gamma &= \bar{\sigma}^{\alpha\beta}_{,\gamma}\boldsymbol{a}_\alpha\delta^\gamma_\beta + \bar{\sigma}^{\alpha\beta}\boldsymbol{a}_{\alpha,\gamma}\delta^\gamma_\beta + \bar{\sigma}^{\alpha\beta}\boldsymbol{a}_\alpha(\boldsymbol{a}_{\beta,\gamma}\cdot\boldsymbol{a}^\gamma) \\
&= \bar{\sigma}^{\alpha\beta}_{,\beta}\boldsymbol{a}_\alpha + \bar{\sigma}^{\alpha\beta}\left(\Gamma^\gamma_{\alpha\beta}\boldsymbol{a}_\gamma + \kappa_{\alpha\beta}\hat{\boldsymbol{n}}\right) + \bar{\sigma}^{\alpha\beta}\boldsymbol{a}_\alpha\left(\Gamma^\lambda_{\beta\gamma}\boldsymbol{a}_\lambda + \kappa_{\beta\gamma}\hat{\boldsymbol{n}}\right)\cdot\boldsymbol{a}^\gamma \\
&= \bar{\sigma}^{\alpha\beta}_{,\beta}\boldsymbol{a}_\alpha + \bar{\sigma}^{\alpha\beta}\left(\Gamma^\gamma_{\alpha\beta}\boldsymbol{a}_\gamma + \kappa_{\alpha\beta}\hat{\boldsymbol{n}}\right) + \bar{\sigma}^{\alpha\beta}\Gamma^\gamma_{\beta\gamma}\boldsymbol{a}_\alpha \qquad (3.26)\\
&= \bar{\sigma}^{\alpha\beta}|_\beta\boldsymbol{a}_\alpha + \bar{\sigma}^{\alpha\beta}\kappa_{\alpha\beta}\hat{\boldsymbol{n}}
\end{aligned}$$

$$\frac{\partial(\bar{\sigma}^{\alpha 3}\boldsymbol{a}_\alpha\hat{\boldsymbol{n}})}{\partial\xi^\gamma}\cdot\boldsymbol{a}^\gamma = \bar{\sigma}^{\alpha 3}\boldsymbol{a}_\alpha(\hat{\boldsymbol{n}}_{,\gamma}\cdot\boldsymbol{a}^\gamma) = -\bar{\sigma}^{\alpha 3}\kappa^\gamma_\gamma\boldsymbol{a}_\alpha \qquad (3.27)$$

$$\begin{aligned}
\frac{\partial(\bar{\sigma}^{3\beta}\hat{\boldsymbol{n}}\boldsymbol{a}_\beta)}{\partial\xi^\gamma}\cdot\boldsymbol{a}^\gamma &= \bar{\sigma}^{3\beta}_{,\gamma}\hat{\boldsymbol{n}}\delta^\gamma_\beta + \bar{\sigma}^{3\beta}\hat{\boldsymbol{n}}_{,\gamma}\delta^\gamma_\beta + \bar{\sigma}^{3\beta}\hat{\boldsymbol{n}}(\boldsymbol{a}_{\beta,\gamma}\cdot\boldsymbol{a}^\gamma) \\
&= \bar{\sigma}^{3\beta}_{,\beta}\hat{\boldsymbol{n}} - \bar{\sigma}^{3\beta}\kappa^\gamma_\beta\boldsymbol{a}_\gamma + \bar{\sigma}^{3\beta}\hat{\boldsymbol{n}}\left(\Gamma^\lambda_{\beta\gamma}\boldsymbol{a}_\lambda + \kappa_{\beta\gamma}\hat{\boldsymbol{n}}\right)\cdot\boldsymbol{a}^\gamma \\
&= \bar{\sigma}^{3\beta}_{,\beta}\hat{\boldsymbol{n}} - \bar{\sigma}^{3\beta}\kappa^\gamma_\beta\boldsymbol{a}_\gamma + \bar{\sigma}^{3\beta}\Gamma^\gamma_{\beta\gamma}\hat{\boldsymbol{n}} \qquad (3.28)\\
&= \bar{\sigma}^{3\beta}|_\beta\hat{\boldsymbol{n}} - \bar{\sigma}^{3\beta}\kappa^\gamma_\beta\boldsymbol{a}_\gamma
\end{aligned}$$

$$\frac{\partial(\bar{\sigma}^{33}\hat{\boldsymbol{n}}\hat{\boldsymbol{n}})}{\partial\xi^\gamma}\cdot\boldsymbol{a}^\gamma = \bar{\sigma}^{33}\hat{\boldsymbol{n}}(\hat{\boldsymbol{n}}_{,\gamma}\cdot\boldsymbol{a}^\gamma) = -\bar{\sigma}^{33}\kappa^\gamma_\gamma\hat{\boldsymbol{n}}. \qquad (3.29)$$

Therefore the 0^{th} moment equation becomes:

$$\left(\bar{\sigma}^{\alpha\beta}|_\beta - \bar{\sigma}^{3\beta}\kappa^\alpha_\beta - \bar{\sigma}^{\alpha 3}\kappa^\gamma_\gamma\right)\boldsymbol{a}_\alpha + \left(\bar{\sigma}^{3\beta}|_\beta + \bar{\sigma}^{\alpha\beta}\kappa_{\alpha\beta} - \bar{\sigma}^{33}\kappa^\gamma_\gamma\right)\hat{\boldsymbol{n}} + \sigma^{i3}\boldsymbol{g}_i\big|^{h^+}_{h^-} + \int_{-h/2}^{h/2} b^i\boldsymbol{g}_i d\xi^3 = 0. \qquad (3.30)$$

which is the same expression derived in [40, 23]. The first two terms correspond to the variation of all the stresses in the tangential directions, while the third term is the tangential and normal shear applied on the inner and outer side of the membrane as a tangential and normal forces in its surface (figure 3.2). It represents the kind of force that we would apply if we blew on the surface of a soap bubble, or if we press it normal to its surface.

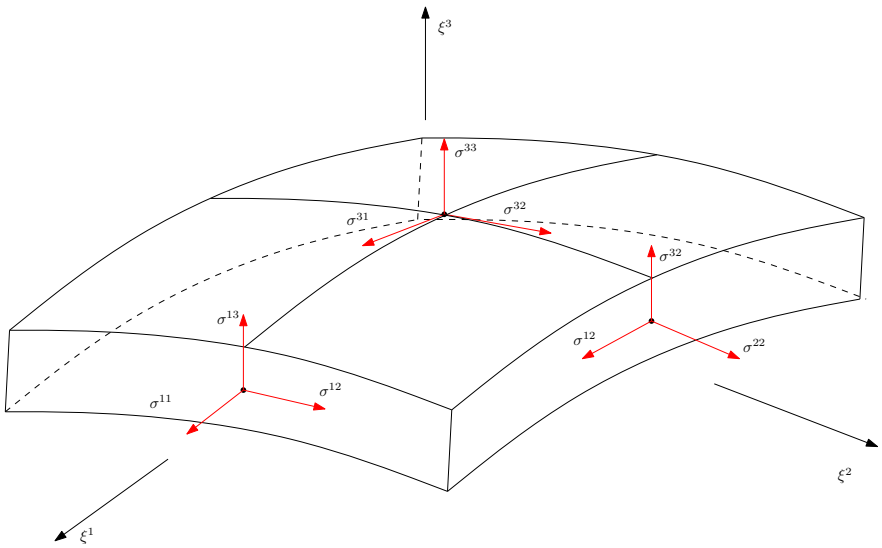


Figure 3.2: *Forces on a differential section of the membrane*

Similarly we proceed to derive the equation of the 1^{st} moment:

$$\int_{-h/2}^{h/2} \xi^3 \left(\frac{\partial \left(\sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^k} \cdot \boldsymbol{g}^k + b^i \boldsymbol{g}_i \right) d\xi^3 = 0 \tag{3.31}$$

and expanding as we did before:

$$\int_{-h/2}^{h/2} \left(\xi^3 \frac{\partial \left(\sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^\alpha} \cdot \boldsymbol{g}^\alpha \right) d\xi^3 + \int_{-h/2}^{h/2} \left(\xi^3 \frac{\partial \left(\sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^3} \cdot \hat{\boldsymbol{n}} \right) d\xi^3 + \int_{-h/2}^{h/2} \xi^3 b^i \boldsymbol{g}_i d\xi^3 = 0 \tag{3.32}$$

.

Then we consider:

$$\frac{\partial \left(\xi^3 \sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^\alpha} \cdot \boldsymbol{g}^\alpha = \xi^3 \frac{\partial \left(\sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^\alpha} \cdot \boldsymbol{g}^\alpha \tag{3.33}$$

$$\frac{\partial \left(\xi^3 \sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^3} \cdot \boldsymbol{g}^3 = \left(\sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right) \cdot \boldsymbol{g}^3 + \xi^3 \frac{\partial \left(\sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^3} \cdot \boldsymbol{g}^3 = \sigma^{i3} \boldsymbol{g}_i + \xi^3 \frac{\partial \left(\sigma^{ij} \boldsymbol{g}_i \boldsymbol{g}_j \right)}{\partial \xi^3} \cdot \boldsymbol{g}^3 \tag{3.34}$$

and substituting into equation 3.32 we get:

$$\frac{\partial}{\partial \xi^\gamma}\left(\int_{-h/2}^{h/2}(\xi^3\sigma^{ij}\boldsymbol{g}_i\boldsymbol{g}_j)d\xi^3\right)\cdot\boldsymbol{a}^\gamma+\xi^3\sigma^{i3}\boldsymbol{g}_i\big|_{-h/2}^{h/2}-\int_{-h/2}^{h/2}(\sigma^{i3}\boldsymbol{g}_i)d\xi^3+\int_{-h/2}^{h/2}\xi^3 b^i\boldsymbol{g}_i d\xi=0. \quad (3.35)$$

Again, using the definition of the resultant moment 3.20 we get:

$$\frac{\partial\left(\bar{m}^{ij}\boldsymbol{a}_i\boldsymbol{a}_j\right)}{\partial\xi^\gamma}\cdot\boldsymbol{a}_\gamma+\xi^3\sigma^{i3}\boldsymbol{g}_i\big|_{-h/2}^{h/2}-\int_{-h/2}^{h/2}(\sigma^{i3}\boldsymbol{g}_i)d\xi+\int_{-h/2}^{h/2}\xi^3 b^i\boldsymbol{g}_i d\xi=0 \quad (3.36)$$

so finally, using a similar derivation we obtain for the 1^{st} moment:

$$\begin{aligned}\left(\bar{m}^{\alpha\beta}|_\beta-\bar{m}^{3\beta}\kappa_\beta^\alpha-\bar{m}^{\alpha 3}\kappa_\gamma^\gamma\right)\boldsymbol{a}_\alpha+\left(\bar{m}^{3\beta}|_\beta+\bar{m}^{\alpha\beta}\kappa_{\alpha\beta}-\bar{m}^{33}\kappa_\gamma^\gamma\right)\hat{\boldsymbol{n}}+\\ m^{i3}\boldsymbol{g}_i\big|_{h^-}^{h^+}-\int_{-h/2}^{h/2}(\sigma^{i3}\boldsymbol{g}_i)d\xi+\int_{-h/2}^{h/2}\xi^3 b^i\boldsymbol{g}_i d\xi^3 \;=\; 0. \quad (3.37)\end{aligned}$$

As before, the first two terms are the variation of the moments in the tangential directions, and the third one is the moment applied on the inner and outer faces of the membrane. Note that this formulation matches the description of an elastic membrane done by [40].

3.3 Constitutive Relations

Our problem is now defined by equations 3.9, 3.29 and 3.37 which are valid to describe a continuous 2D membrane of small thickness with not singularities and embedded in a 3D space. Next we will introduce the constitutive equations that will particularize those equations to a specific fluid membrane.

Fluid membranes are a specific type of membranes where the molecules that compose them can exchange positions among themselves and give the membrane the properties of a fluid enclosed within the membrane body. For instance, a soap bubble is an air vesicle whose interface is made of water surrounded by a surfactant and where there is a fluid flow only within its surface. Most importantly, fluid membranes can be found in biological systems such as cells, bacteria or liposomes where the membrane is formed by a surfactant molecule called lipid.

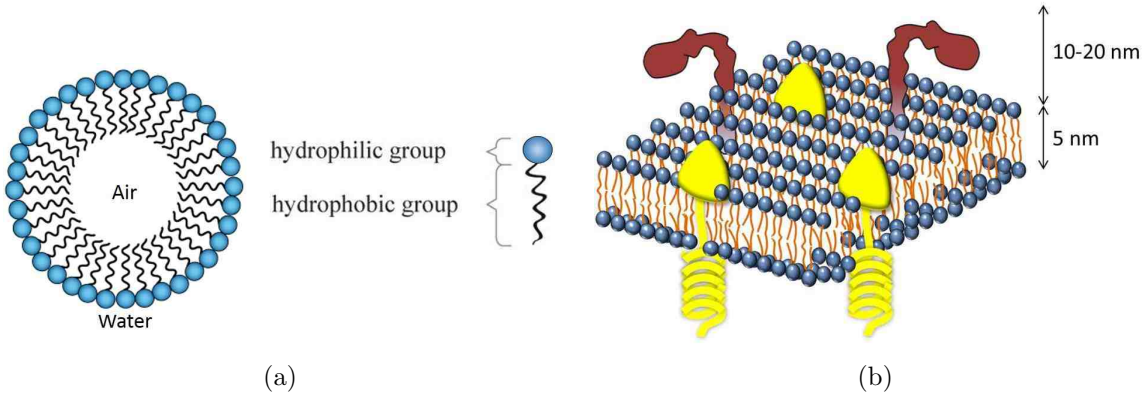


Figure 3.3: *Left. Due to the lipid properties a micelle has an air interior and it is surrounded by an aqueous environment. Right. Scheme of the outer layer of a cell membrane which is mostly formed by lipid molecules and different types of proteins. The fluidity of those membranes comes from the fact that the lipids are in constant motion within the interface giving it the overall behavior of a viscous fluid constrained to the membrane body.*

Lipids are a specific type of molecule formed by and hydrophilic head and an hydrophobic tail which, in an aqueous environment, arrange themselves in different structures such that the exposure of the tails to the water is minimized. This results in the formation of common structures such as micelles (figure 3.3a), or bilayers (figure 3.3b). In those layers, the lipids can exchange positions between lipids on the same layer, or in the case of the bilayers, with the opposite one, and this behavior is responsible for the fluid-like behavior of the membrane.

Even thought the membrane has a fluid behaviour, it is usually not a Newtonian fluid. Lipid membranes are often folded or wrinkled so they can be easily stretched or compressed and each single layer can also be either compressed or stretched, so the membrane is left with a behavior similar to the one of a rubber balloon. Furthermore, some complex entities such as cells can store pieces of the membrane in their interior and they have the capability of changing the total mass at any time.

Due to this complexity on the description of a fluid membrane, we will make two assumptions to simplify the balance of mass: 1) we consider that the total mass of the system is constant and 2) the components of the velocity are constat across the membrane (the membrane experiences no shear), which is safe since they are usually very thin. Hence, the balance of mass results:

$$\frac{D\rho}{Dt} + \rho\left(v^{\alpha}|_{\alpha} - v^{3}\kappa^{\alpha}_{\alpha}\right) = 0. \tag{3.38}$$

Expanding the material time derivative of the density we can write:

$$\frac{\partial\rho}{\partial t} + \boldsymbol{v}\cdot\nabla\rho + \rho\left(v^{\alpha}|_{\alpha} - v^{3}\kappa^{\alpha}_{\alpha}\right) = 0, \tag{3.39}$$

and taking into account that $\rho\mathrm{div}(\boldsymbol{v}) + \boldsymbol{v}\mathrm{grad}(\rho) = \mathrm{div}(\rho v^{\alpha})$ the final balance of mass becomes:

$$\boxed{\frac{\partial\rho}{\partial t} + \left(\rho v^{\alpha}\right)|_{\alpha} - \rho v^{3}\kappa^{\alpha}_{\alpha} = 0.} \tag{3.40}$$

Those membranes are in general very thin, usually around 5 nanometers in thickness. Combining this fact with the fluidity that the molecules provide it makes the shear stresses on the membrane very small, and it is save to assume that they are negligible. Hence, the equation for the $0th$ moment can be simplified and decomposed in its normal and tangential parts resulting in:

$$\sigma^{\alpha\beta}|_{\beta} + b^{\alpha} = 0 \tag{3.41}$$

$$\sigma^{\alpha\beta}\kappa_{\alpha\beta} + b^{3} = 0. \tag{3.42}$$

Following [2] the Cauchy stress tensor on a fluid membrane in curvilinear coordinates can be written as:

$$\sigma^{\alpha\beta} = 2\mu D^{\alpha\beta} + \lambda g^{\alpha\beta}\left(g^{\mu\lambda}v_{\mu|\lambda} - v_{n}g^{\mu\lambda}\kappa_{\mu\lambda}\right) - pg^{\alpha\beta}, \tag{3.43}$$

where $D^{\alpha\beta}$ is the deformation gradient tensor [34]:

$$2D^{\alpha\beta} = g^{\alpha\lambda}g^{\beta\mu}\left(v_{\lambda|\mu} + v_{\mu|\lambda} - 2\kappa_{\lambda\mu}v^{n}\right), \tag{3.44}$$

μ, λ are the Lame coefficients and p is the pressure. Hence the Cauchy stress tensor is composed by a term arising from the viscous forces (first two components), and a term coming from the pressure.

According to this relationship, the stresses are a function of the pressure but not the density. However, in the case of a fluid membrane those two magnitudes are related, since a change of pressure can not be explained without considering a change in density; when the pressures increases at one point, the lipids are being compressed together incrementing its density and viceversa. As a first approximation we use the pressure-density relationship proposed by [11], but in future work we might look for a relationship that better represents the physics of the system:

$$p(\rho) = a(\rho - \rho_0)^c, \tag{3.45}$$

here a and c are positive constants. This relationship makes the pressure proportional to the difference between the current density and a base density ρ_0 that would be the case where the lipids are neither compressed nor stretched, and it seems reasonable for this initial model. Substituting this relationship into equation 3.43 the Cauchy stress tensor we obtain:

$$\sigma^{\alpha\beta} = 2\mu D^{\alpha\beta} + \lambda g^{\alpha\beta}\left(g^{\mu\lambda}v_{\mu|\lambda} - v_n g^{\mu\lambda}\kappa_{\mu\lambda}\right) - a(\rho - \rho_0)^c g^{\alpha\beta}. \tag{3.46}$$

Finally, substituting 3.46 into 3.41 and 3.42 we get the final expressions for the tangent and normal components of the $0th$ moment:

$$\boxed{2\mu D^{\alpha\beta}|_\beta + \lambda g^{\alpha\beta}\left(g^{\mu\lambda}v_{\mu|\lambda} - v_n g^{\mu\lambda}\kappa_{\mu\lambda}\right)|_\beta - g^{\alpha\beta}(a(\rho - \rho_0)^c)|_\beta + b^\alpha = 0} \tag{3.47}$$

$$\boxed{\left(2\mu D^{\alpha\beta} + \lambda g^{\alpha\beta}\left(g^{\mu\lambda}v_{\mu|\lambda} - v_n g^{\mu\lambda}\kappa_{\mu\lambda}\right) - a(\rho - \rho_0)^c g^{\alpha\beta}\right)\kappa_{\alpha\beta} + b^3 = 0.} \tag{3.48}$$

If we simplified the system to one where there were no velocities, we would see that the second equation simply becomes Laplace law in the vesicle. In our system, whenever there is an internal pressure caused by a change in density, the system will try to reach equilibrium by both creating a tangential flow and decreasing its curvature. Note that due to the fact that the pressure is related to the density, a system with an arbitrary shape (in absence of curvature energy) will be in equilibrium as long as the density is equal to ρ_0 everywhere, and it will not tend to minimize its

curvature. However, if there is a change in density, the system will tend to minimize its curvature until the lipids are completely distributed, which is not necessarily the same as reaching a spherical or cylindrical form. Further examples will discuss this issue in detail.

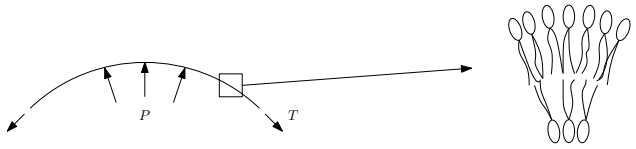


Figure 3.4: *In pure fluid vesicles, the position of the membrane is determined by Laplace law. However, vesicles with lipid membranes have stretching and bending resistance due to the stretching and compression of the lipid heads and therefore its equilibrium state is more complex.*

Finally, if a lipid membrane is bent, the heads of the lipids on the bottom are compressed while the lipids on the top are stretched (figure 3.4). This creates a bending resistance to the external moments represented by the $1st$ moment equations of the linear momentum. Again, because of the fluidity of the membrane, the out of plane moments do not play a big role and will be simplified. Applying this to equations 3.37 and separating between the tangential and normal moments we obtain:

$$m^{\alpha\beta}|_{\beta}\boldsymbol{a}_{\alpha} + \int_{-h/2}^{h/2} \xi^3 b^{\alpha}\boldsymbol{g}_{\alpha} d\xi^3 = 0 \tag{3.49}$$

$$m^{\alpha\beta}\kappa_{\alpha\beta}\hat{\boldsymbol{n}} = 0. \tag{3.50}$$

We did not consider any moment in the normal direction. Those moments would create a vortex within the interface which would be hard to represent due to the high viscosity model of our system. The curvature energy of biological incompressible membranes is generally assumed to be given by the minimization of Helfrich energy [24]. From this energy, the corresponding bending moment has been derived in [9, 8, 43], resulting in:

$$m^{\alpha\beta} = -[K_c(H + C_0)g^{\alpha\beta} + \bar{K}\kappa^{\alpha\beta}] \tag{3.51}$$

With this we have defined all the governing equations needed to describe our problem, hence the next step will be to propose a numerical scheme to solve the equations.

Chapter 4

Numerical Strategy

There are many different numerical schemes that have been used to solve similar problems to the one concerned in this thesis [19]. In Rosolen et al. [37] they use a meshfree method to solve the membrane equations using a phase field approach. Rahimi et al. [36] used B-splines combined with a finite elements to solve the surface equations and similarly Dolbow [12, 14] added a phase field on this same system to solve the equations of a biomembrane in phase separation. All of this methods have their advantages and problems and all of them seem suitable for each case. When choosing the numerical strategy to solve the system of PDE's previously described we took into account mainly three different aspects: the ability to obtain highly accurate approximations that can describe the curvature and its derivatives, the potential to implement the strategy in 3D without being hindered by the remeshing of the system, and the potential speed of the method.

In the end we decided to use a meshfree particle method. This way, there is absolutely no need for meshing our membrane, which can be fairly complicated when there are great deformations present. Second, a particle method is great if the structure can break or merge with another, as it happens when sometimes two vesicles fuse into one. On top of that, we decided to use radial basis functions to interpolate our fields using scattered particles.

Radial basis functions have proved to be an excellent tool to approximate and solve PDE's on 2D manifolds immersed in a 3D space with high accuracy and without the need for a mesh of the body [15, 16, 20, 39]. Hence in order to solve our system of equations we used a completely meshfree approach called local radial point interpolation method (LRPIM) developed by Liu et al.

[33], which is an extension of the more common meshless local petrov galerking (MLPG) [4, 5]. This method uses RBF's to construct the shape functions that will interpolate the fields in the surface and then solves the system of PDE's using a local weak from of the problem.

4.0.1 Field interpolation: Radial Basis Functions

Consider a body Ω described by a set of N scattered particles with coordinates $\boldsymbol{x} = \left\{ \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \right\}$, like the one shown in figure 4.1, where we want to interpolate a function $u(\boldsymbol{x})$ defined over the total domain. Since there is no underlying mesh telling us what nodes we have to use to interpolate the system at one particular point, for each point $\boldsymbol{x}$ where we want to interpolate a field we define a support domain, or an influence region, that collects the n neighbour particles and that will act similarly to our single finite element. Based on those collected particles and the distance r between each of them and the point of interest we will construct the necessary shape functions to interpolate any field.

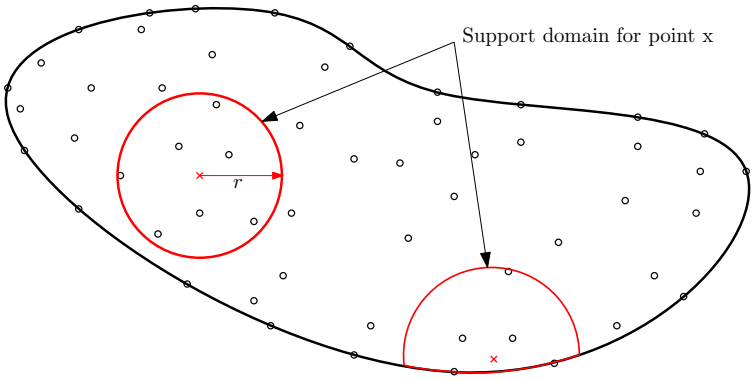


Figure 4.1: *Scheme of the support domain and particle gathering for the interpolation at a point x*

A function defined on the surface is interpolated using a combination of radial basis functions ϕ and polynomials p of degree m such as:

$$u(\boldsymbol{x}) = \sum_{i=1}^{n} \phi_i(\boldsymbol{x}) a_i + \sum_{i=1}^{m} p_j(\boldsymbol{x}) b_j = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{a} + \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{b} = \left\{ \begin{array}{cc} \boldsymbol{\phi}^T(\boldsymbol{x}) & \boldsymbol{p}^T(\boldsymbol{x}) \end{array} \right\} \left\{ \begin{array}{c} \boldsymbol{a} \\ \boldsymbol{b} \end{array} \right\} \tag{4.1}$$

The objective now is to use the information of closest particles at the point x to create some basis functions that interpolate a field in a similar way that it is done in finite elements. In order to determine the coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$, a support domain is created around the point $\boldsymbol{x}$ collecting the n closest particles where the field of interest is known. This leads to a set of equations that can be expressed as:

$$\boldsymbol{U}_s = \boldsymbol{\phi}_0 \boldsymbol{a} + \boldsymbol{P}_m \boldsymbol{b} \tag{4.2}$$

where $\boldsymbol{U}_s$ is the vector containing the values of the functions at the n collected points:

$$\boldsymbol{U}_s^T = \left\{ \begin{array}{cccc} u_1 & u_2 & \dots & u_n \end{array} \right\}, \tag{4.3}$$

$\boldsymbol{\phi}_0$ is the matrix of radial basis functions:

$$\boldsymbol{\phi}_0 = \begin{bmatrix} \phi_1(r_1) & \phi_2(r_1) & \dots & \phi_n(r_1) \\ \phi_1(r_2) & \phi_2(r_2) & \dots & \phi_n(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(r_n) & \phi_2(r_n) & \dots & \phi_n(r_n) \end{bmatrix}_{n \times n}, \tag{4.4}$$

and $\boldsymbol{P}_m^T$ is the polynomial matrix of order m:

$$\boldsymbol{P}_m^T = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ \vdots & \vdots & \ddots & \vdots \\ p_m(\boldsymbol{x}_1) & p_m(\boldsymbol{x}_2) & \dots & p_m(\boldsymbol{x}_n) \end{bmatrix}_{m \times n}, \tag{4.5}$$

so the vector coefficients for RBFs and polynomials respectively are:

$$\boldsymbol{a}^T = \left\{ \begin{array}{cccc} a_1 & a_2 & \dots & a_n \end{array} \right\} \tag{4.6}$$

$$\boldsymbol{b}^T = \left\{ \begin{array}{cccc} b_1 & b_2 & \dots & b_m \end{array} \right\}. \tag{4.7}$$

Now we have $n+m$ variables but only n equations. Hence, we add the following additional m constraint equations as described in [22] and used to enforce the partition of unity:

$$\sum_{i=1}^{n} p_j(\boldsymbol{x}_i) a_i = \boldsymbol{P}_m^T \boldsymbol{a} = 0, \qquad j = 1, 2, \dots, m \tag{4.8}$$

so we obtain the following system of equations:

$$\tilde{\boldsymbol{U}}_s = \left[\begin{array}{c} \boldsymbol{U}_s \\ \boldsymbol{0} \end{array} \right] = \left[\begin{array}{cc} \boldsymbol{\phi}_0 & \boldsymbol{P}_m \\ \boldsymbol{P}_m^T & \boldsymbol{0} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{a} \\ \boldsymbol{b} \end{array} \right\} = \boldsymbol{G} \left\{ \begin{array}{c} \boldsymbol{a} \\ \boldsymbol{b} \end{array} \right\} \tag{4.9}$$

and from this, it is easy to see that:

$$\left\{ \begin{array}{c} \boldsymbol{a} \\ \boldsymbol{b} \end{array} \right\} = \boldsymbol{G}^{-1} \tilde{\boldsymbol{U}}_s \tag{4.10}$$

The matrix $\boldsymbol{G}$ is guaranteed to be non-singular as long as there is no repeated point for any set of scattered points and dimension [16]. Hence, substituting the coefficients into equation 4.1 we obtain:

$$u(\boldsymbol{x}) = \left\{ \begin{array}{cc} \boldsymbol{\phi}^T(\boldsymbol{x}) & \boldsymbol{p}^T(\boldsymbol{x}) \end{array} \right\} \boldsymbol{G}^{-1} \tilde{\boldsymbol{U}}_s = \tilde{\boldsymbol{N}}(\boldsymbol{x}) \tilde{\boldsymbol{U}}_s \tag{4.11}$$

where

$$\tilde{\boldsymbol{N}}^T(\boldsymbol{x}) = \left\{ \begin{array}{ccccccc} N_1(\boldsymbol{x}) & N_2(\boldsymbol{x}) & \dots & N_n(\boldsymbol{x}) & N_{n+1}(\boldsymbol{x}) & \dots & N_m(\boldsymbol{x}) \end{array} \right\} \tag{4.12}$$

but since the vector $\tilde{\boldsymbol{U}}_s$ contains zeros from $n+1$ to m, the interpolation of $u(x)$ can be expressed as:

$$u(\boldsymbol{x}) = \boldsymbol{N}(\boldsymbol{x})^T \boldsymbol{U}_s \tag{4.13}$$

where the vector of shape functions is:

$$\boldsymbol{N}^T(\boldsymbol{x}) = \left\{ \begin{array}{cccc} N_1(\boldsymbol{x}) & N_2(\boldsymbol{x}) & \dots & N_n(\boldsymbol{x}) \end{array} \right\} \tag{4.14}$$

and its derivatives can be easily obtained as:

$$u_{,i}(\boldsymbol{x}) = \boldsymbol{N}_{,i}^T(\boldsymbol{x})\boldsymbol{U}_s = \frac{\partial}{\partial x}\left(\left\{ \begin{array}{cc} \boldsymbol{\phi}^T(\boldsymbol{x}) & \boldsymbol{p}^T(\boldsymbol{x}) \end{array} \right\} \right) \boldsymbol{G}^{-1}\tilde{\boldsymbol{U}}_s \tag{4.15}$$

One of the main difference between this shape functions and the ones obtained using the regular point interpolation method (PIM), or the MLPG is the fact that this functions enforce perfectly the partition of unity and this makes it very easy to enforce Dirichlet boundary conditions.

There are many different radial basis functions that can be used [16]: Multiquadric, Gaussian, T-spline, etc. and each of them is better suited for a specific problem. However, for the sake of simplicity all this work is done using Gaussian RBF's and is left as future work to find what are the advantages of other functions, and which are better suited for this problem.

$$\phi(r) = e^{-(\epsilon r)^2} \tag{4.16}$$

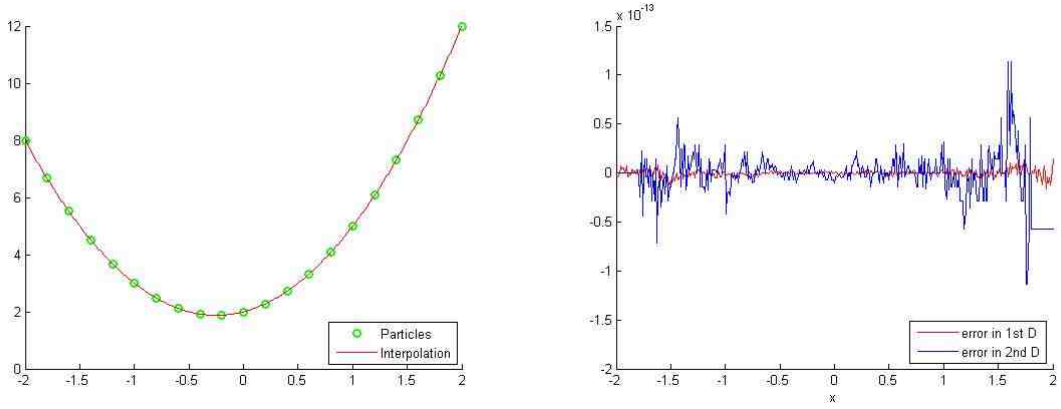
where ϵ is a shape parameter to control the smoothness of the solution and r is the cartesian distance between two points.

4.0.2 Accuracy of the shape functions

Once defined the interpolation of a field we will check the results by interpolating a polynomial function using a set of scattered particles. The radial point interpolation shape functions used a combination of radial basis functions and a polynomial basis to interpolate the fields on the surface. Hence, they should be able to perfectly interpolate a polynomial. In that case all the terms affecting the radial basis functions should come to zero, while the coefficients on the polynomial terms should be exactly the ones of the polynomial. Hence we will try to fit a function such as:

$$f(x) = ax^2 + bx + c \tag{4.17}$$

To interpolate the function first we place several pairs (x,f(x)) that will be our field nodes, or particles. Using these nodes we interpolate the polynomial and its first two derivatives. Figure 4.2a shows the results of interpolating a second degree polynomial with coefficients $a = 1, b = 1, c = 2$. As expected, the maximum absolute error was $3.5 \cdot 10^{-15}$ which is just MATLAB's accuracy.



(a) (b)

Figure 4.2: *Left. Interpolation of a second order polynomial with parameters a = 1, b = 1, c = 2 Right. Absolute error in the first and second derivatives of the same polynomial*

The error increased as we moved towards the edge of the domain, which is due to the fact that the value of the polynomial increases, but relatively it stayed the same. Figure 4.2b shows the absolute error on the first and second derivatives which is still very low but increasing with each derivative. This is normal as in every interpolation the error increases with the grade of the derivative. In case that this error became a problem in larger problems, we would consider the implementation of techniques that reduce the error in the derivatives when using RBF's [13].

4.0.3 Local Radial Point Interpolation Method

Having defined the interpolation of a field based on several neighbouring particles, the next step consists in defining the strategy to solve the system of PDE's that conform the governing

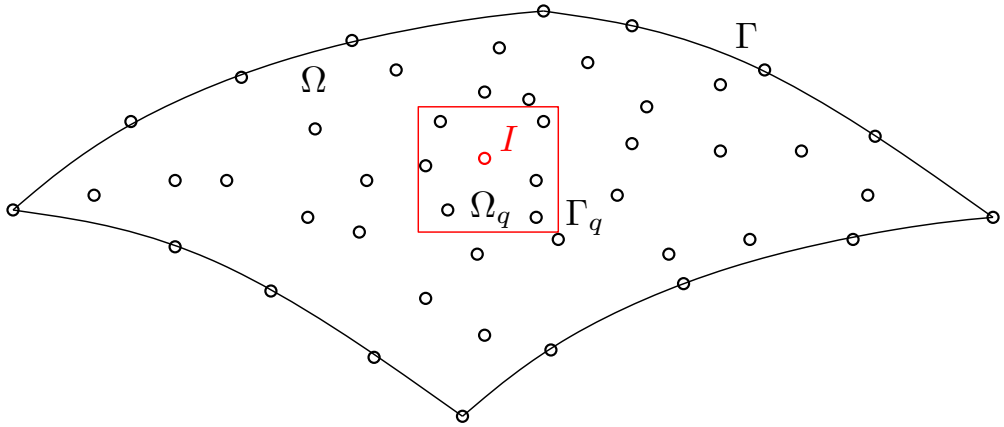


Figure 4.3: *A body* Ω *with a boundary* Γ *is discretized via particles or field nodes. For each node we construct a local weak from on a local subdomain* Ω_q *with boundary* Γ_q

equations of the problem (3.40,3.47,3.48). For this purpose we will use a locally weighted residual approach called local radial point interpolation method [33], that leads to a local weak from for each node I. However, instead of going straightforward to the full problem, with the purpose of illustrating how the method works, we will formulate and solve the diffusion equation in 1D:

$$\frac{\partial c(\boldsymbol{x},t)}{\partial t} = D\frac{\partial^2 c(\boldsymbol{x},t)}{(\partial x)^2} \tag{4.18}$$

with boundary conditions on Dirichlet Γ_d and Neumann Γ_n boundaries:

$$c - c_0 \qquad \text{on} \quad \Gamma_d \tag{4.19}$$

$$D\frac{\partial c}{\partial x} = q \qquad \text{on} \quad \Gamma_n \tag{4.20}$$

Consider a body Ω with boundary Γ discretized by a total of N particles or field nodes. For each particle we define a subdomain, or quadrature domain, Ω_q bounded by Γ_q. Because the governing equations are satisfied in any domain, we can define a local weak form for each particle I as:

$$\int_{\Omega_q} W_I \left(\frac{\partial c}{\partial t} - D \frac{\partial^2 c}{(\partial x)^2} \right) d\Omega = 0 \tag{4.21}$$

where W^I is the weight or test function centered at node I and $c = c(\boldsymbol{x}, t)$. One of the main advantage of the local meshfree method is that it only requires the compatibility in the local quadrature domain, which means that using local weighted method, the solution will exist as long as the field approximation is continuous within the quadrature domain, the shape function is differentiable and the resultant integrand is integrable.

Using integration by parts on the second term we obtain:

$$\frac{\partial}{\partial x} \left(W_I \frac{\partial c}{\partial x} \right) = \frac{\partial W_I}{\partial x} \frac{\partial c}{\partial x} + W_I \frac{\partial^2 c}{\partial x^2}, \tag{4.22}$$

substituting it into equation 4.21 and applying the divergence theorem we obtain:

$$\int_{\Omega_q} \left[W_I \frac{\partial c}{\partial t} + D \frac{\partial W_I}{\partial x} \right] d\Omega - \int_{\Gamma_I} D W_I \frac{\partial c}{\partial x} d\Gamma = 0 \tag{4.23}$$

In order to do the boundary integral we have to consider where is our particle and how is the quadrature domain. If the local domain does not intersect with the global boundary of the body $\Gamma_I = \Gamma_q$. However, if the local domain does intersect with the global domain we must consider:

$$\Gamma_I = \Gamma_q \bigcup \Gamma_d \bigcup \Gamma_n \bigcup \Gamma_r \tag{4.24}$$

where d, n, r stand for Dirichlet, Neumann and Robin boundary conditions. For the sake of simplicity we chose a step function as our test function in this problem such as:

$$W_I(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \Omega_q^I \\ 0 & \boldsymbol{x} \notin \Omega_q^I \end{cases} \tag{4.25}$$

and introducing this into equation 4.23 we obtain:

$$\int_{\Omega_q} \frac{\partial c}{\partial t} d\Omega - \int_{\Gamma_q} D \frac{\partial c}{\partial x} d\Gamma - \int_{\Gamma_d} D \frac{\partial c}{\partial x} d\Gamma - \int_{\Gamma_n} q d\Gamma = 0 \tag{4.26}$$

Next we introduce the interpolation of our field c at a point $\boldsymbol{x}$:

$$c(\boldsymbol{x}, t) = \sum_{k=1}^{n} N_k(\boldsymbol{x}) c_k(t) \tag{4.27}$$

for the n collected particles at point $\boldsymbol{x}$. Introducing this into equation 4.26 we obtain the following differential equation:

$$\boldsymbol{C}\dot{\boldsymbol{U}}(t) + \boldsymbol{K}\boldsymbol{U}(t) = \boldsymbol{F}(t) \tag{4.28}$$

where

$$\boldsymbol{C}_{IJ} = \int_{\Omega_q} W_I N_J(\boldsymbol{x}) d\Omega \tag{4.29}$$

$$\boldsymbol{K}_{IJ} = -\int_{\Gamma_q} W_I D \frac{\partial N_J(\boldsymbol{x})}{\partial x} n_x d\Omega - \int_{\Gamma_d} W_I D \frac{\partial N_J(\boldsymbol{x})}{\partial x} n_x d\Omega \tag{4.30}$$

$$\boldsymbol{F}_I = -\int_{\Gamma_n} W_I q \Gamma \tag{4.31}$$

$$\boldsymbol{U}^T = \left\{ \begin{array}{cccc} c_1 & c_2 & \dots & c_N \end{array} \right\} \tag{4.32}$$

$$\dot{\boldsymbol{U}}^T = \left\{ \begin{array}{cccc} \dot{c}_1 & \dot{c}_2 & \dots & \dot{c}_N \end{array} \right\} \tag{4.33}$$

Note that the matrices in 4.28 are assembled differently than in finite elements or other global weak form methods. In FEM the local matrix constructed for each element or node is assembled on a global matrix based on the global connectivity of the elements. In this method, like in MLPG, the global matrices are constructed by stacking row by row the nodal matrices constructed for each particle, or just adding together the equations found for each weak form. This way, we end up

with a non-symmetric banded (in general) matrix. In this particular problem, if we collected all particles at each point, both $\boldsymbol{C}$ and $\boldsymbol{K}$ would be assembled as:

$$\begin{bmatrix} K_{11} & K_{12} & \dots & K_{1(N-1)} & K_{1N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{I1} & K_{I2} & \dots & K_{(I-1)(N-1)} & K_{IN} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{N1} & K_{N2} & \dots & K_{N(N-1)} & K_{NN} \end{bmatrix} \begin{Bmatrix} c_1 \\ \vdots \\ c_I \\ \vdots \\ c_N \end{Bmatrix} = \begin{bmatrix} F_1 \\ \vdots \\ F_I \\ \vdots \\ F_N \end{bmatrix} \tag{4.34}$$

However, since each point only collects a few particles, most of the terms of this matrix become zero. The time integration of equation 4.28 is solved using a simple Crank-Nicholson two point difference:

$$\frac{1}{2}\left(\frac{\partial c}{\partial t}\right)_{t+\Delta t} + \frac{1}{2}\left(\frac{\partial c}{\partial t}\right)_{t} = \frac{c_{t+\Delta t} - c_t}{\Delta t} \tag{4.35}$$

and substituting it into equation 4.28 one obtains:

$$\left[\frac{\boldsymbol{C}}{\Delta t} + \frac{1}{2}\boldsymbol{K}_{t+\Delta t}\right] c_{t+\Delta t} = \frac{1}{2}\left(\boldsymbol{F}_{t+\Delta t} + \boldsymbol{F}_t\right) + \left[\frac{\boldsymbol{C}}{\Delta t} - \frac{1}{2}\boldsymbol{K}_t\right] c_t \tag{4.36}$$

Up to this point we obtained a full scheme to solve the original PDE of the diffusion equation using a meshfree method based on the radial basis functions interpolation. To check the validity of the method, the next section solves the equation and compares it to the analytical solution.

4.0.4 Solving the diffusion equation

Consider a 1D bar with length $L = 1$ and a constant diffusion coefficient $D = 0.1$ (Figure 4.4). This bar is subjected to Dirichlet boundary conditions of $c = 0$ at both ends, and it has an initial concentration at time zero of:

$$c(x, 0) = x(x - 1)(x - 0.5)(x - 0.25) \tag{4.37}$$

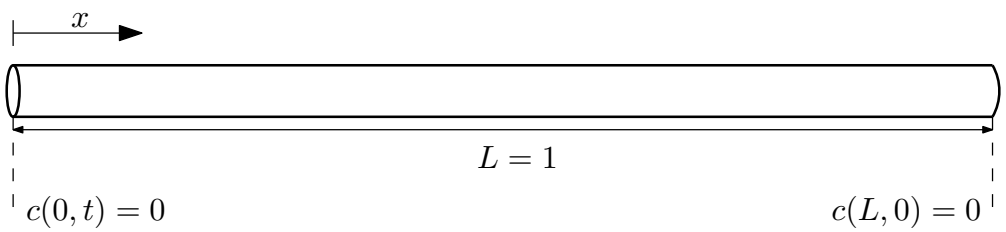


Figure 4.4: *Scheme of the 1D diffusion problem*

This problem can be solved analytically using Fourier series with the following result:

$$c(x,t) = \frac{2}{L}\sum_{n=1}^{\infty}\left[\left(\int_0^L c(y,0)\sin\left(\frac{n\pi y}{L}\right)dy\right)\sin\left(\frac{n\pi x}{L}\right)e^{\frac{-n^2\pi^2 Dt}{L^2}}\right] \tag{4.38}$$

The solution is compared to the system implemented using the previous method. For this example, we set a total time of $t = 0.1s$ in increments of $dt = 0.0001s$, and we used a total of 41 particles equally distributed along the distance. Figure 4.5 shows the initial and final concentrations at every point and the location of the field nodes used in the problem.

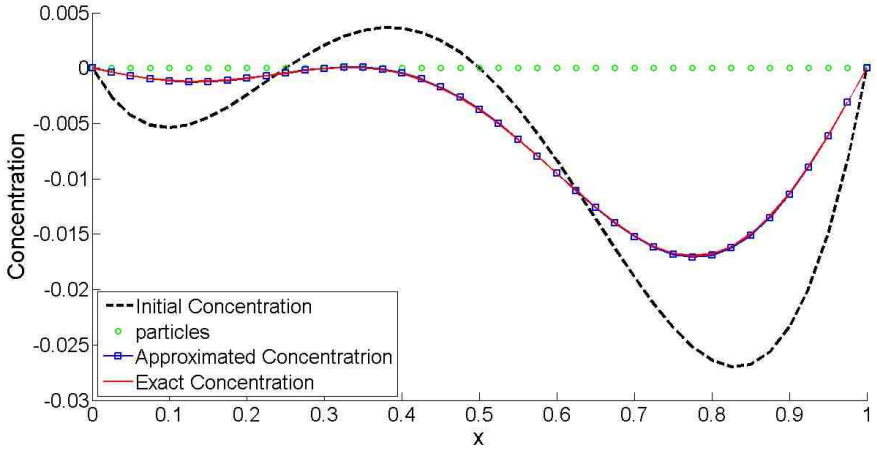


Figure 4.5: *Analytical and approximate solution of the diffusion equation with a coefficient $D = 0.1$.*

After a thousand iterations, the cumulated absolute error is plotted in figure 4.6 where the maximum relative error was $6 \cdot 10^{-3}\%$. Taking into account that we used a second order CN scheme and that our test function is just a Heaviside jump the results are very accurate. The solution could

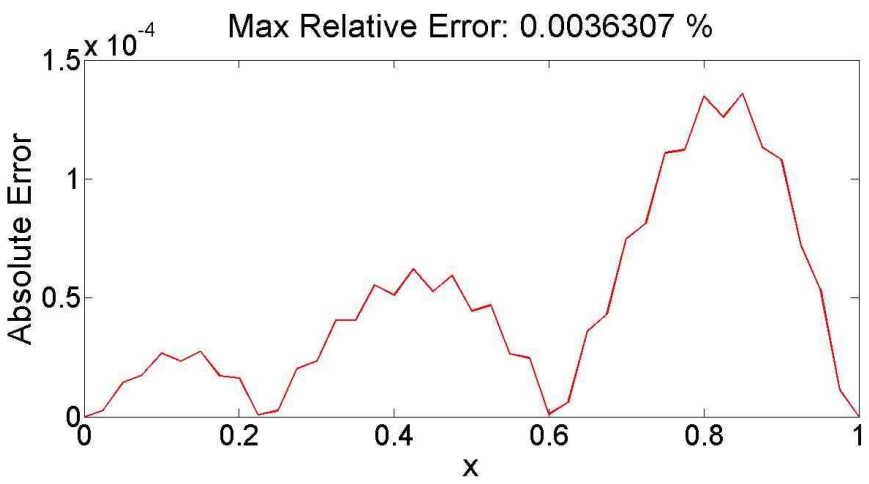
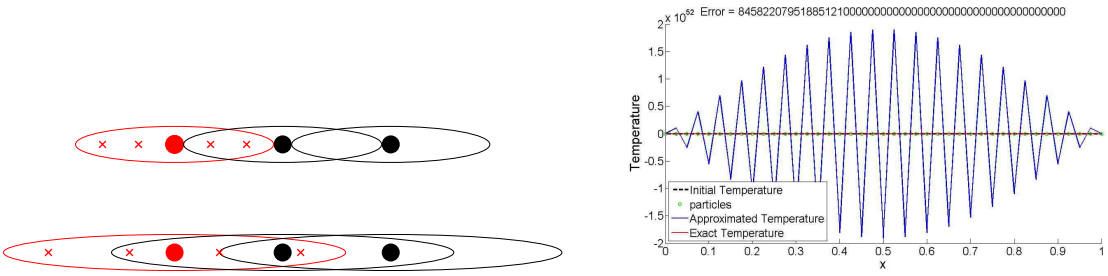


Figure 4.6: *Cumulative absolute error at each point*

be easily improved by increasing the number of particles, using a better time integration scheme or using a more smooth test function. Nonetheless, none of this analysis will be done for this problem.

Finally, we want to notice the importance of choosing the right quadrature domain. Because we are using a local weak form formulation, the problem is locally solved for each particle in an arbitrary subdomain called quadrature domain. This domains are theoretically arbitrary and they intersect each other, and the only constraint is that they should cover the whole domain. However, we observed that when any point of the domain is overlapped by more than two domains (Figure 4.7a) the solution becomes unstable, and for the same problem described before we obtain what is shown in figure 4.7b. The fact that the solution becomes unstable due to this condition is just an hypothesis, and it could be due to some other condition that occurs at the same time. Nonetheless, the problem was always solved when we avoided this particular scenario.



(a) (b)

Figure 4.7: *Left. In the top scheme, each point of the domain is covered by a support domain but non of them is overlapped more than 2 times. On the bottom, because the domains are two big, the central point is covered by the three domains. Right. ATemperature profile when some point is covered by more than two integration domains*

Chapter 5

Mixed Formulation

The main particularity of this membrane problem is the fact that we are dealing with a deformable fluid membrane. In terms of numerically solving this problem, this poses and obvious question. Should we pursue a Lagrangian or Eulerian approach?

In our opinion the answer is a mixed of those. As a body that is being deformed under external (or internal) inputs it should be treated from a Lagrangian point of view, because the interface is changing positions, curvature, etc. On the other side, treating a fluid flow in a Lagrangian point of view is very complicated. For instance, think about a soap bubble such as the one depicted in figure 5.1; we can clearly see how the fluid in the membrane is creating vortex, moving from once place to another, and that would be very almost impossible to describe from a Lagrangian viewpoint.

If we did that, our particles would be completely distorted and we would have to reconstruct our shape functions all the time. Hence, we see that it is not possible to follow each point of a

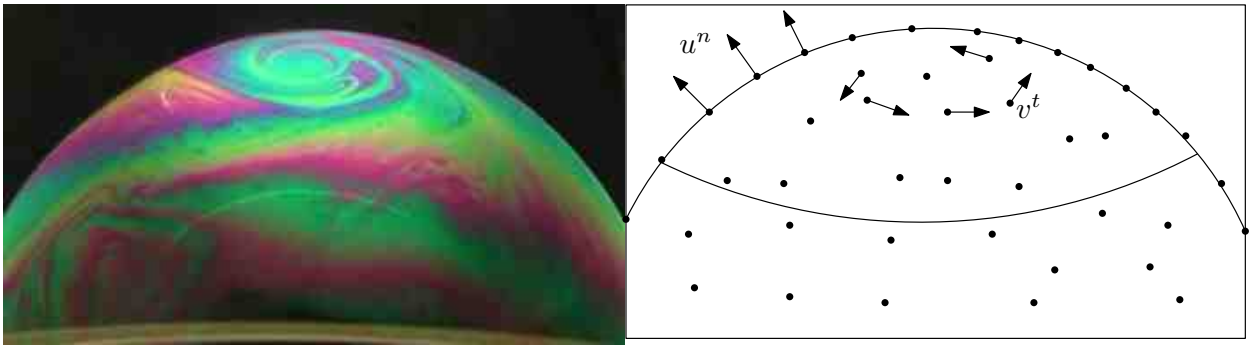


Figure 5.1: *A soap bubble is a clear example of a fluid membrane where the surface flow is clearly visible*

fluid but it is much more efficient to look at it from an Eulerian viewpoint, where we look at the velocities from a fixed point in space, and it does not matter where the actual particles move but what is happening at that point.

For this reason, we decided to pursue a mixed approach to the problem where the velocity of each node will be decomposed in its tangential and normal component:

$$\boldsymbol{v} = v^t \boldsymbol{g} + v^n \hat{\boldsymbol{n}} \tag{5.1}$$

From a tangential perspective, we will solve the problem in an Eulerian point of view, solving for the tangential velocity on each of the nodes, and never updating their position. This way, we will know how the fluid in the membrane looks like without having any problems on the particle position.

On the other side, we will be solving for the displacement on the normal direction, as it is commonly done in level set problems, and the particles will indeed be updated in time. Hence we will end up with a deforming membrane with a fluid flow in its surface.

This method has just one single inconvenient, since we are updating the particles only in a normal position, we must do it in very small time increments, so the normal has time to change and adapt to point to the actual position of the vesicle. An update done too fast would lead to wrong results.

Chapter 6

Compressible fluid 1D membrane (no bending resistance)

Up to this point, we derived the governing equations that describe a compressible fluid membrane and suitable numerical strategy was proposed to solve them. Hence, the next step will be to find the weak form and combine it with the meshfree method to solve the problem.

As we have seen, the mechanical problem described by equations 3.40, 3.47 and 3.48 is a complex nonlinear system of partial differential equations. Adding this to the fact that we have to deal with differential geometry, we are left with a very complex problem. Therefore, in order to ease the implementation, we subdivided the main problem in simpler questions that will help us develop the model, find errors, and set the basis for the future three dimensional implementation of a compressible fluid membrane.

First of all, we assumed that none of the membranes in the following examples has bending resistance, so we did not consider equation 3.37 in our calculations. This is a major simplification and it is definitely not true in real life systems, so it will be one of the first things to incorporate in future research. However, adding bending rigidity to the membranes adds a next level of complexity. One one hand, we need to add two new partial differential equations to our system, which will have to be properly derived and error free. And on the other hand, those equations involve the derivative of the curvature tensor, which implies having to interpolate the third order derivative of our surface. Hence, we decided not to incorporate all this equations until we know that everything else is correctly derived.

As it was mentioned before, not having any bending rigidity on the problem implies that the

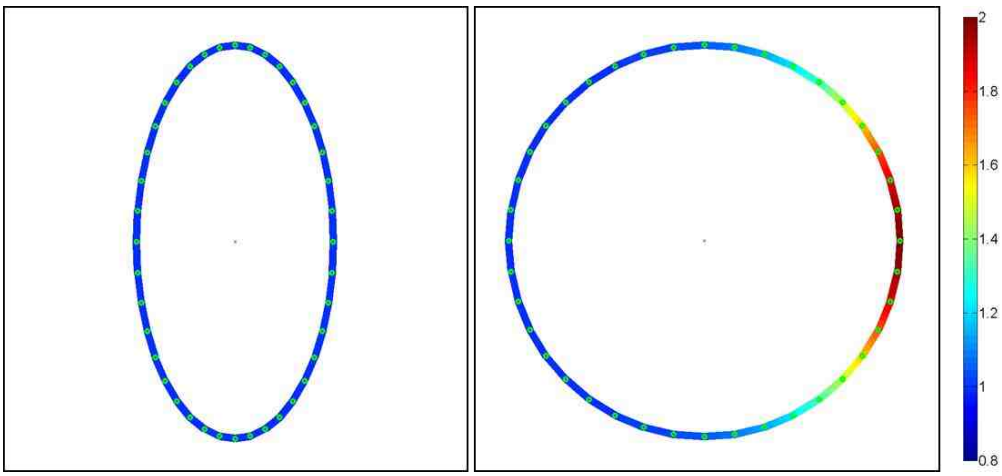


Figure 6.1: *The effect of not accounting for the curvature energy. The membrane on the left has a constant density (line color) everywhere $\rho = \rho_0$, so it is in equilibrium. While the membrane on the right has a density variation so it will deform until the density equilibrates everywhere.*

system will not seek for a minimum curvature shape unless there is a specific pressure difference in its surface. To set an example, only the ellipsoidal membrane depicted in figure 6.1 will be in equilibrium, since its density is constant everywhere. The circular membrane of the right will deform until it reaches equilibrium although it will probably tend to keep a circular shape.

The second simplification was not to implement the full 3D problem at once but to start with a simpler 1D one and build up from there. This way, the differential geometry gets extremely simplified, for instance, there are less Christoffel symbols to be computed, less curvature components, and less variables in general that could be introducing errors. Furthermore, this is the model that we will incorporate in the general 2D model for the transport of soft colloids that our groups is currently developing [17]. Recall that one of the main reasons for developing this model, was to improve our understanding on the transport of soft colloids through porous media by improving the physics on our current formulation, so we'll need to develop a 1D membrane as well as an axisymmetric one (section 7) on top of the full 3D goal.

6.1 Weak Form

With all the previous assumptions our fluid membrane problem can be written as:

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t} + (\rho v^{\alpha})\,|_{\alpha} - \rho v^{3}\kappa^{\alpha}_{\alpha} = 0 && \forall \boldsymbol{x} \in \Omega \\
&2\mu D^{\alpha\beta}|_{\beta} + \lambda g^{\alpha\beta}\left(g^{\mu\lambda}v_{\mu|\lambda} - v_n g^{\mu\lambda}\kappa_{\mu\lambda}\right)|_{\beta} - g^{\alpha\beta}(a(\rho-\rho_0)^c)|_{\beta} + b^{\alpha} = 0 && \forall \boldsymbol{x} \in \Omega \\
&\left(2\mu D^{\alpha\beta} + \lambda g^{\alpha\beta}\left(g^{\mu\lambda}v_{\mu|\lambda} - v_n g^{\mu\lambda}\kappa_{\mu\lambda}\right) - a(\rho-\rho_0)^c g^{\alpha\beta}\right)\kappa_{\alpha\beta} + b^{3} = 0 && \forall \boldsymbol{x} \in \Omega \\
&v^{3} = v^{d} && \forall \boldsymbol{x} \in \Gamma_d
\end{aligned}
\tag{6.1}
$$

Where the last equation is an optional Dirichlet boundary condition to prescribe the normal velocity at some points of the membrane. Due to the mixed formulation that we are using, and be cause we are dealing with a closed surface, our system only moves through the normal direction of the membrane, so any problem will be well posed regardless of the existence or not of an essential boundary condition.

Having said that, we will derive the weak form of the problem. Introducing the test functions ζ_α, θ and η, we can write:

$$
\begin{aligned}
&\int_{\Omega}\left[2\mu D^{\alpha\beta}|_{\beta} + \lambda g^{\alpha\beta}\left(v^{\lambda}|_{\lambda} - v_n\kappa^{\lambda}_{\lambda}\right)|_{\beta} - g^{\alpha\beta}(a(\rho-\rho_0)^c)|_{\beta} + b^{\alpha}\right]\zeta_{\alpha}d\Omega \quad + \\
&\int_{\Omega}\left[\dot{\rho} + (\rho v^{\alpha})\,|_{\alpha} - \rho v^{3}\kappa^{\alpha}_{\alpha}\right]\theta d\Omega \quad + \\
&\int_{\Omega}\left[\left(2\mu D^{\alpha\beta} + \lambda g^{\alpha\beta}\left(v^{\lambda}|_{\lambda} - v_n\kappa^{\lambda}_{\lambda}\right) - a(\rho-\rho_0)^c g^{\alpha\beta}\right)\kappa_{\alpha\beta} + b^{3}\right]\eta \quad = \quad 0,
\end{aligned}
\tag{6.2}
$$

and using the chain rule followed by the divergence theorem the first term can be rewritten as:

$$
\begin{aligned}
\int_{\Omega}\left(D^{\alpha\beta}|_{\beta}\zeta_{\alpha}\right)d\Omega &= \int_{\Omega}\left(D^{\alpha\beta}\zeta_{\alpha}\right)|_{\beta}d\Omega - \int_{\Omega}\left(D^{\alpha\beta}\zeta_{\alpha}|_{\beta}\right)d\Omega && (6.3)\\
&= \underbrace{\int_{\Gamma}\left(D^{\alpha\beta}\zeta_{\alpha}\right)n_{\beta}d\Gamma}_{0} - \int_{\Omega}\left(D^{\alpha\beta}\zeta_{\alpha}|_{\beta}\right)d\Omega && (6.4)
\end{aligned}
$$

where the first term becomes zero because the normal vector is perpendicular to the surface. Similarly, for the next two terms we can write

$$\int_\Omega \left(\lambda g^{\alpha\beta}\left(v^\lambda|_\lambda - v_n\kappa_\lambda^\lambda\right)|_\beta\zeta_\alpha\right)d\Omega = -\int_\Omega \left(\lambda g^{\alpha\beta}\left(v^\lambda|_\lambda - v_n\kappa_\lambda^\lambda\right)\zeta_\alpha|_\beta\right)d\Omega \tag{6.5}$$

$$\int_\Omega \left(g^{\alpha\beta}(a(\rho-\rho_0)^c)|_\beta\zeta_\alpha\right)d\Omega = -\int_\Omega \left(g^{\alpha\beta}(a(\rho-\rho_0)^c)\zeta_\alpha|_\beta\right)d\Omega \tag{6.6}$$

and introducing them back into 6.2 we obtain the final expression for the weak form 6.7:

$$\begin{aligned} &\int_\Omega \left(2\mu D^{\alpha\beta} + \lambda g^{\alpha\beta}\left(v^\lambda|_\lambda - v_n\kappa_\lambda^\lambda\right) - g^{\alpha\beta}a(\rho-\rho_0)^c\right)\zeta_\alpha|_\beta d\Omega \\ &\qquad - \int_\Omega b^\alpha\zeta_\alpha - \int_\Omega \left[\dot\rho + (\rho v^\alpha)\,|_\alpha - \rho v^3\kappa_\alpha^\alpha\right]\theta d\Omega \\ &-\int_\Omega \left[\left(2\mu D^{\alpha\beta} + \lambda g^{\alpha\beta}\left(v^\lambda|_\lambda - v_n\kappa_\lambda^\lambda\right) - a(\rho-\rho_0)^c g^{\alpha\beta}\right)\kappa_{\alpha\beta}\right]\eta d\Omega - \int_\Omega b^3\eta d\Omega \quad = \quad 0 \end{aligned} \tag{6.7}$$

6.2 Linearization

The previous problem led to a non-linear expression of the weak form, therefore we must linearize the equations in a context of a Newton-Raphson solution for the incremental term. In order to organize the following steps, the weak form is divided in three equations G,H,I:

$$G = \int_\Omega \left(\underbrace{2\mu D^{\alpha\beta}}_{G_1} + \underbrace{\lambda g^{\alpha\beta}\left(v^\lambda|_\lambda - v_n\kappa_\lambda^\lambda\right)}_{G_2} - \underbrace{g^{\alpha\beta}a(\rho-\rho_0)^c}_{G_3}\right)\zeta_\alpha|_\beta d\Omega = 0 \tag{6.8}$$

$$H = \int_\Omega \left[\underbrace{\dot\rho}_{H1} + \underbrace{(\rho v^\alpha)\,|_\alpha}_{H_2} - \underbrace{\rho v^3\kappa_\alpha^\alpha}_{H3}\right]\theta d\Omega = 0 \tag{6.9}$$

$$I = \int_\Omega \left[\underbrace{2\mu D^{\alpha\beta}\kappa_{\alpha\beta}}_{I1} + \underbrace{\lambda g^{\alpha\beta}\left(v^\lambda|_\lambda - v_n\kappa_\lambda^\lambda\right)\kappa_{\alpha\beta}}_{I2} - \underbrace{a(\rho-\rho_0)^c g^{\alpha\beta}\kappa_{\alpha\beta}}_{I3} + b^3\right]\eta = 0 \tag{6.10}$$

From now on we will use the subindex $n+1$ to indicate the t_{n+1}. Using the linearization operator, the functions G,H,I at t_{n+1}, can be written as:

$$\mathfrak{L}X_{n+1} = X^0_{n+1} + \delta X_{n+1} = 0 \qquad X = G, H, I \tag{6.11}$$

where

$$\delta X = \frac{\partial X}{\partial v^\alpha}\delta v^\alpha + \frac{\partial X}{\partial u^n}\delta u^n + \frac{\partial X}{\partial \rho}\delta\rho \qquad X = G, H, I \tag{6.12}$$

This is an explicit problem where the deformed geometry will be updated in very small time steps, hence, the updated values of the metric, the curvature, and other geometrical parameters will be very close between two time increments. Using this fact, and for sake of simplicity, we assumed that the change on the metric and curvature is small enough so that they can treated as constants (within the time increment) and the linearization of those terms is zero:

$$\delta g^{\alpha\beta} = \delta\kappa^{\alpha\beta} = 0 \tag{6.13}$$

Once the surface is updated, those values are obviously recomputed to match the current surface. Besides this fact, the linearization was carried normally to all the terms, the full details of which can be found in Appendix A. Following those calculations, the incremental terms in 6.8, 6.9 and 6.10 can be written as:

$$\delta G = \int_\Omega \Big[\mu g^{\alpha\gamma} g^{\beta\lambda} \left((\delta v_\gamma)\,|_\lambda + (\delta v_\lambda)\,|_\gamma\right) + \lambda g^{\alpha\beta}\left(\left(\delta v^\lambda\right)|_\lambda - v_n\kappa^\lambda_\lambda\right) - g^{\alpha\beta} ac\,(\rho-\rho_0)^{c-1}\,\delta\rho\Big]\,\zeta_\alpha|_\beta d\Omega \tag{6.14}$$

$$\delta H = \int_\Omega \left[\delta\dot\rho + \rho(\delta v^\alpha)|_\alpha + v^\alpha|_\alpha\delta\rho + \rho_{,\alpha}\delta v^\alpha + v^\alpha(\delta\rho)_{,\alpha}\right]\theta d\Omega \tag{6.15}$$

$$\delta I = \int_\Omega \Big[\mu g^{\alpha\gamma} g^{\beta\lambda}\left((\delta v_\gamma)|_\lambda + (\delta v_\lambda)|_\gamma\right)\kappa_{\alpha\beta} + \lambda g^{\alpha\beta}\kappa_{\alpha\beta}\left(\delta v^\lambda\right)|_\lambda - g^{\alpha\beta}\kappa_{\alpha\beta} ac(\rho-\rho_0)^{c-1}\delta\rho\Big]\,\eta d\Omega. \tag{6.16}$$

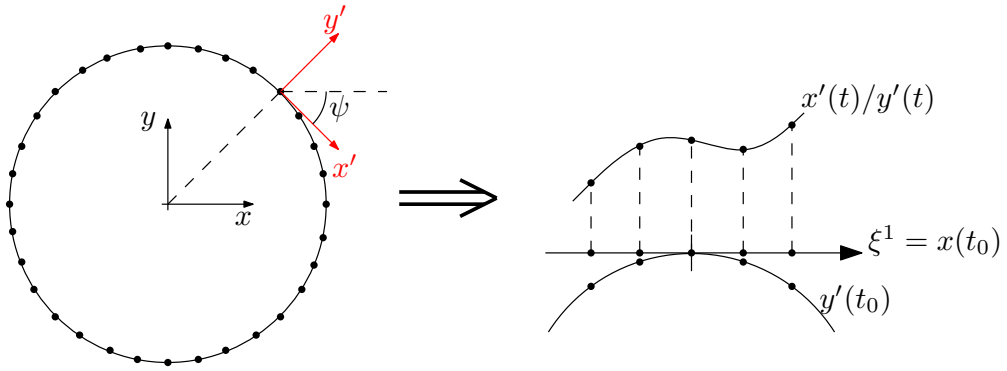


Figure 6.2: *A local coordinate system is defined at each particle based on the initial position of the particles and it will be kept the same during the time evolution of the surface. Each local weak form will be constructed and solved in its corresponding basis.*

Once we have a full linearized problem, we need to couple it with our numerical scheme that will allow us to solve the system.

6.3 Numerical Interpolation (1D)

In order to couple our equations with the numerical approximation, we need to introduce a parametrization of the membrane. Because this a 1D problem, we could introduce a global parametrization of the system using a variable like the arc length that covered the full circular membrane, and always map backwards to this initial shape. However, this would only be valid whenever the initial shape of our system is known and mappable with a global coordinate, so we would be restricting ourselves to very specific initial shapes like spheres, circles, ellipses, etc. For instance, bacteria usually have rod-like shapes and would be left outside our method.

For this reason, we introduced a local parametrization for each particle where we will build the local weak forms (figure 6.2), and that will be used to construct our shape functions and locally interpolate the surface. Moreover, this will be the same procedure used on a full 3D body so it will help on its future implementation.

On each particle, we translate and rotate the global coordinate system $x-y$, to define a local system $x'-y'$ such that y' has the direction of the normal at that point.

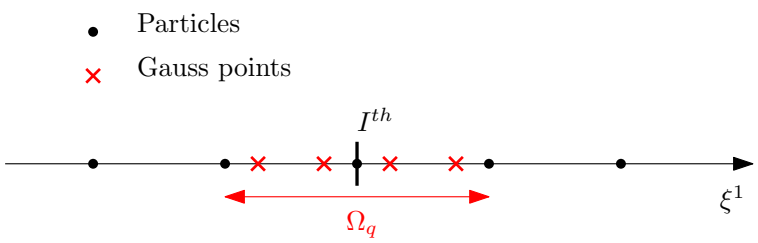


Figure 6.3: *The quadrature domain is defined between the two closes particles around the I^{th} one*

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} + \begin{Bmatrix} x_I \\ y_I \end{Bmatrix} \tag{6.17}$$

Then, the initial particles ($t = t_0$), are projected on the x' axis that will act as our parent coordinate ξ^1, and that will be used to calculate our shape functions. To avoid recomputing the shape functions each time the particles are deformed, we will keep the same initial rotated axis and the only difference will be that we will be interpolating the functions $x'(t)$ and $y'(t)$, using the parent coordinate ξ^1. The membrane is locally parametrized by a coordinate ξ^1, so its equation at any point can be written as:

$$\boldsymbol{x} = \boldsymbol{x}(\xi^1) = \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} f(\xi^1) \\ g(\xi^1) \end{Bmatrix} \tag{6.18}$$

To construct a local weak form for each particle we define a local subdomain Ω_q around the particle where we are solving the weak form, and we introduce a gauss quadrature for its integration.

The shape functions are constructed at each gauss point following the LRPIM procedure. This is done by collecting its n closest nodes, so in the end, the contribution of each gauss point to the local weak form at the node I will require the use of the n particles needed to interpolate each field. Therefore, we define the following variable vector for each gauss point:

$$\boldsymbol{u}^T = \{u^1, ..., u^n\} \tag{6.19}$$

and after constructing the shape functions and its derivatives, we define:

$$\boldsymbol{N}^T = \left\{N^1, N^2, \ldots, N^n\right\} \tag{6.20}$$

$$\boldsymbol{N}_{,1}^T = \left\{N_{,1}^1, N_{,1}^2, \ldots, N_{,1}^n\right\} \tag{6.21}$$

$$\boldsymbol{N}_{,11}^T = \left\{N_{,11}^1, N_{,11}^2, \ldots, N_{,11}^n\right\} \tag{6.22}$$

Hence the interpolation of a general field and its derivatives reads:

$$u = u(\xi^1) = \sum_{I=1}^{n} N^I(\xi^1) u^I = \boldsymbol{N}^T \boldsymbol{u} \tag{6.23}$$

$$u_{,1} = \frac{\partial u(\xi^1)}{\partial \xi^1} = \sum_{I=1}^{n} N_{,1}^I(\xi^1) u^I = \boldsymbol{N}_{,1}^T \boldsymbol{u} \tag{6.24}$$

$$u_{,11} = \frac{\partial^2 u(\xi^1)}{\partial \left(\xi^1\right)^2} = \sum_{I=1}^{n} N_{,11}^I(\xi^1) u^I = \boldsymbol{N}_{,11}^T \boldsymbol{u} \tag{6.25}$$

Having defined the interpolation of each variable and its derivative, we can compute the local curvilinear basis by doing $\partial \boldsymbol{x} / \partial \xi^1$.

$$\boldsymbol{g}_1 = \frac{\partial \boldsymbol{x}}{\partial \xi^1} = \begin{bmatrix} \boldsymbol{N}_{,s}^T \boldsymbol{x} & \boldsymbol{N}_{,s}^T \boldsymbol{y} \end{bmatrix} \tag{6.26}$$

$$\boldsymbol{g}_{1,1} = \begin{bmatrix} \boldsymbol{N}_{,ss}^T \boldsymbol{x} & \boldsymbol{N}_{,ss}^T \boldsymbol{y} \end{bmatrix} \tag{6.27}$$

which can be completed by introducing the unit normal at each subdomain as:

$$\hat{\boldsymbol{n}} = \frac{\begin{bmatrix} -\boldsymbol{N}_{,1}^T \boldsymbol{y} & \boldsymbol{N}_{,1}^T \boldsymbol{x} \end{bmatrix}}{\| \begin{bmatrix} -\boldsymbol{N}_{,1}^T \boldsymbol{y} & \boldsymbol{N}_{,1}^T \boldsymbol{x} \end{bmatrix} \|} \tag{6.28}$$

As an example, in figure 6.4 we plotted the local basis obtained using a four gauss points quadrature on a circular membrane.

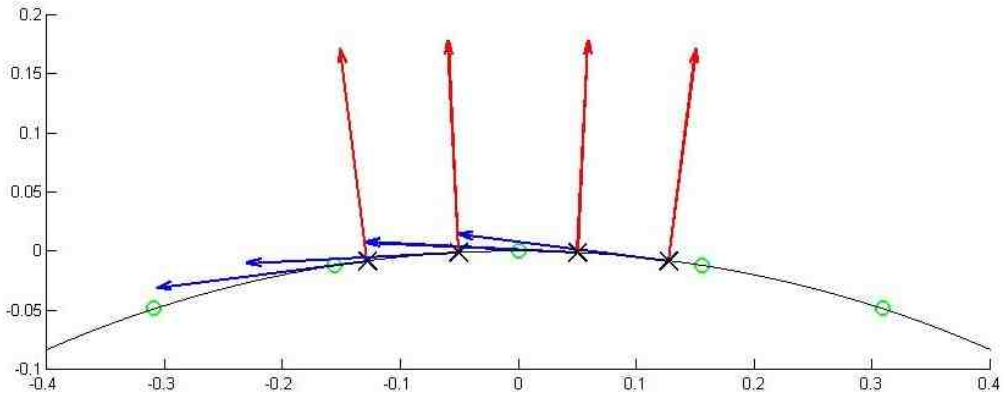


Figure 6.4: *Local basis at each gauss point.*

Having defined the local basis on our surface, we can define its metric as:

$$g_{11} = \boldsymbol{g}_1 \cdot \boldsymbol{g}_1 = \boldsymbol{x}^T \boldsymbol{N}_{,1} \boldsymbol{N}_{,1}^T \boldsymbol{x} + \boldsymbol{y}^T \boldsymbol{N}_{,1} \boldsymbol{N}_{,1}^T \boldsymbol{y} \qquad \Rightarrow \qquad g^{11} = \frac{1}{g_{11}} \tag{6.29}$$

and the second fundamental form, or curvature tensor, becomes:

$$\kappa_{11} = \frac{-\boldsymbol{x}^T \boldsymbol{N}_{,11} \boldsymbol{N}_{,1}^T \boldsymbol{y} + \boldsymbol{y}^T \boldsymbol{N}_{,11} \boldsymbol{N}_{,1}^T \boldsymbol{x}}{\| \begin{bmatrix} -\boldsymbol{N}_{,1}^T \boldsymbol{y} & \boldsymbol{N}_{,1}^T \boldsymbol{x} \end{bmatrix} \|} \tag{6.30}$$

so we can finally calculate H, or twice the mean curvature:

$$H = g^{\alpha\beta} \kappa_{\beta\alpha} = g^{11} k_{11} \tag{6.31}$$

Next we introduce the Christoffel symbols of the first kind which, since we are using a 1D membrane, are reduced to just one:

$$\Gamma_{111} = \frac{1}{2} \frac{\partial g_{11}}{\partial \xi^1} = \boldsymbol{x}^T \boldsymbol{N}_{,11} \boldsymbol{N}_{,1}^T \boldsymbol{x} + \boldsymbol{y}^T \boldsymbol{N}_{,11} \boldsymbol{N}_{,1}^T \boldsymbol{y} \tag{6.32}$$

and from this, the Christoffel symbol of the second kind is:

$$\Gamma_{11}^1 = g^{11} \Gamma_{111} \tag{6.33}$$

Before we continue to construct the weak form, we need to introduce the interpolation of the test functions. Hence, we define:

$$\zeta_1(\xi^1) = \sum_{I=1}^{n} W^I(\xi^1)\zeta_1^I \tag{6.34}$$

$$\theta(\xi^1) = \sum_{I=1}^{n} W^I(\xi^1)\theta^I \tag{6.35}$$

$$\eta(\xi^1) = \sum_{I=1}^{n} W^I(\xi^1)\eta^I \tag{6.36}$$

Following [33], the test function ζ_α, η and θ are chosen to be cubic splines centered at node I which vanish at the edge of the quadrature domain (figure 6.5). That means that as long as our quadrature domain does not intersect with the global boundary Γ, the integration over the local boundary Γ_q will be 0.

$$W_i(\boldsymbol{x}) = \begin{cases} \frac{2}{3} - 4r_i^2 + 4r_i^3 & r_i \leq \frac{r_q}{2} \\ \frac{4}{3} - 4r_i + 4r_i^2 - \frac{4}{3}r_i^3 & \frac{r_q}{2} \leq r_i \leq r_q \\ 0 & r_i > r_q \end{cases} \tag{6.37}$$

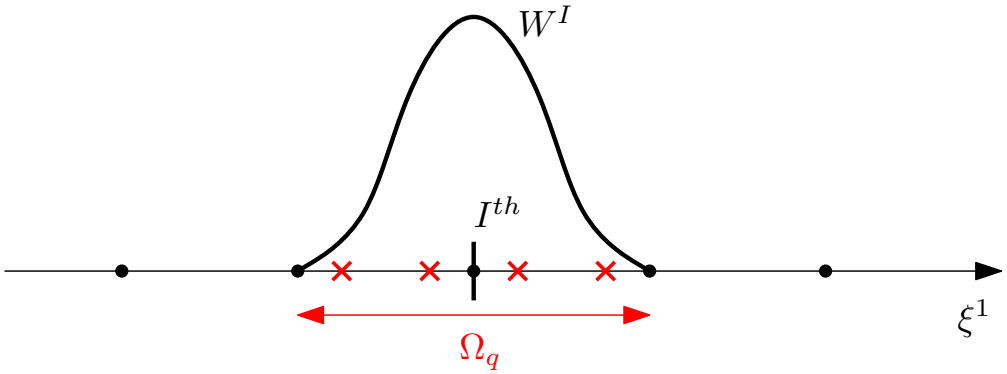


Figure 6.5:

As seen in the final form of the weak form, we need to take the covariant derivative of the test functions. Hence we define:

$$\left[\ \zeta_{1|1}\ \right] = \underbrace{\left[W_{,s} - \Gamma^1_{11} W\right]}_{\boldsymbol{s}_I} \underbrace{\left[\ \zeta^I_1\ \right]}_{\boldsymbol{\zeta}^I} \tag{6.38}$$

so we can finally proceed to assemble our global vectors and matrices. Our system is Eulerian in the tangent direction, and Lagrangian in the normal one, hence our global variable and unknowns vectors are:

$$\boldsymbol{U}^T = \left[\ \ldots \quad v^1_I \quad u^n_I \quad \rho_I \quad \ldots\ \right]^T_{1\times 3N}, \tag{6.39}$$

where each variable is expressed in the local basis of its node. Then, the variation or incremental vector is:

$$\delta\boldsymbol{U}^T = \left[\ \ldots. \quad \delta v^1_I \quad \delta u^n_I \quad \delta\rho_I \quad \ldots^T\ \right]_{1\times 3N}. \tag{6.40}$$

With the purpose of making the interpolation more clear in each of the terms in $G^0, H^0, I^0, \delta G, \delta H, \delta I$ we define the following matrices:

$$\begin{aligned} \boldsymbol{B}_1 &= \left[\ g_{11}N^1_{,1} \quad 0 \quad 0 \quad \ldots \quad g_{11}N^n_{,1} \quad 0 \quad 0\ \right]_{(1\times 3n)} \\ &+ \left[\ g_{11}\Gamma^1_{11}N^1 \quad 0 \quad 0 \quad \ldots \quad g_{11}\Gamma^1_{11}N^n \quad 0 \quad 0\ \right]_{(1\times 3n)} \end{aligned} \tag{6.41}$$

$$\boldsymbol{B}_2 = \left[\ N^1_{,1} + \Gamma^1_{11}N^1 \quad 0 \quad 0 \quad \ldots \quad N^n_{,1} + \Gamma^1_{11}N^n \quad 0 \quad 0\ \right]_{(1\times 3n)} \tag{6.42}$$

$$\boldsymbol{B}_3 = \left[\ N^1 \quad 0 \quad 0 \quad \ldots \quad N^n \quad 0 \quad 0\ \right]_{(1\times 3n)} \tag{6.43}$$

$$\boldsymbol{B}_4 = \left[\ 0 \quad N^1 \quad 0 \quad \ldots \quad 0 \quad N^n \quad 0\ \right]_{(1\times 3n)} \tag{6.44}$$

$$\boldsymbol{B}_5 = \left[\begin{array}{cccccccc} 0 & 0 & N^1 & \dots & 0 & 0 & N^n \end{array}\right]_{(1\times 3n)} \tag{6.45}$$

$$\boldsymbol{B}_6 = \left[\begin{array}{cccccccc} 0 & 0 & N^1_{,1} & \dots & 0 & 0 & N^n_{,1} \end{array}\right]_{(1\times 3n)} \tag{6.46}$$

The details of the interpolation of each expression can be found in Appendix B. Finally, going back to 6.11, the linearization of $G_{n+1}, H_{n+1}, I_{n+1}$ can be written in matrix form as:

$$\boxed{\begin{aligned}\mathfrak{L}G_{n+1} = \underbrace{\left[G1 + G2 - G3\right]^0}_{\boldsymbol{K}_G^0} + \left(\boldsymbol{\zeta}^I\right)^T \underbrace{\int_\Omega \boldsymbol{S}_I^T \left[2\mu \boldsymbol{G}\boldsymbol{B}_1 + \lambda \boldsymbol{g}_c^v \boldsymbol{B}_2 - ac(\rho-\rho_0)^{c-1}\boldsymbol{g}_c^v \boldsymbol{B}_5\right] d\Omega}_{\boldsymbol{K}_G} \delta\boldsymbol{U} \\ + \left(\boldsymbol{\zeta}^I\right)^T \underbrace{\int_\Omega \boldsymbol{S}_I^T \left[-2\mu \boldsymbol{G}\boldsymbol{\kappa}^v \boldsymbol{B}_4 - \kappa_\lambda^\lambda \boldsymbol{B}_4\right] d\Omega}_{\boldsymbol{C}_G} \delta\dot{\boldsymbol{U}}\end{aligned}} \tag{6.47}$$

$$\boxed{\begin{aligned}\mathfrak{L}H_{n+1} = \underbrace{\left[H1 + H2 - H3\right]^0}_{\boldsymbol{K}_H^0} + \theta^I \underbrace{\int_\Omega W_I \left[\rho \boldsymbol{B}_2 + \rho_{,s}\boldsymbol{B}_3 + v^\alpha|_\alpha \boldsymbol{B}_5 + v^1 \boldsymbol{B}_6 - v^3 \kappa_\alpha^\alpha \boldsymbol{B}_5\right] d\Omega}_{\boldsymbol{K}_H} \delta\boldsymbol{U} \\ + \theta^I \underbrace{\int_\Omega W_I \left[\boldsymbol{B}_5 - \rho \kappa_\alpha^\alpha \boldsymbol{B}_4\right] d\Omega}_{\boldsymbol{C}_H} \delta\dot{\boldsymbol{U}} = 0\end{aligned}} \tag{6.48}$$

$$\boxed{\begin{aligned}\mathcal{L}I_{n+1} = \underbrace{\left[I1 + I2 - I3\right]^0}_{\boldsymbol{K}_I^0} + \eta^I \underbrace{\int_\Omega W_I \left[(\boldsymbol{\kappa}^v)^T \left(2\mu \boldsymbol{G}\boldsymbol{B}_1 + \lambda \boldsymbol{g}_c^v \boldsymbol{B}_2 - ac(\rho-\rho_0)^{c-1}\boldsymbol{g}_c^v \boldsymbol{B}_5\right)\right] d\Omega}_{\boldsymbol{K}_I} \delta\boldsymbol{U} \\ + \eta^I \underbrace{\int_\Omega W_I \left[(\boldsymbol{\kappa}^v)^T \left(-2\mu \boldsymbol{G}\boldsymbol{\kappa}^v \boldsymbol{B}_4 - \kappa_\lambda^\lambda \boldsymbol{B}_4\right)\right] d\Omega}_{\boldsymbol{C}_I} \delta\dot{\boldsymbol{U}}\end{aligned}} \tag{6.49}$$

$$\boxed{\mathcal{L}F_{n+1} = -\underbrace{\int_\Omega b^\alpha \zeta_\alpha d\Omega}_{\boldsymbol{F}_G} - \underbrace{\int_\Omega b^3 \eta d\Omega}_{\boldsymbol{F}_I}} \tag{6.50}$$

Finally, going back to the weak form we obtain the following system of equations to be solved with a Newton-Raphson algorithm:

$$\begin{pmatrix} \boldsymbol{K}_G \\ \boldsymbol{K}_H \\ \boldsymbol{K}_I \end{pmatrix} \delta \boldsymbol{U} + \begin{pmatrix} \boldsymbol{C}_G \\ \boldsymbol{C}_H \\ \boldsymbol{C}_I \end{pmatrix} \delta \dot{\boldsymbol{U}} = - \begin{pmatrix} \boldsymbol{K}_G^0 + \boldsymbol{F}_G \\ \boldsymbol{K}_H^0 \\ \boldsymbol{K}_I^0 + \boldsymbol{F}_I \end{pmatrix} \tag{6.51}$$

6.4 Time Integration

The previous problem is time dependent, so we must integrate in time in order to solve it. Using the generalized trapezoidal rule, with an integration parameter α we define:

$$v_{i(n+1)} = v_{i(n)} + \Delta t(1-\alpha)\ \dot{v}_{i(n)} + \alpha \Delta t\ \dot{v}_{i(n+1)} \tag{6.52}$$

$$\rho_{(n+1)} = \rho_{(n)} + \Delta t(1-\alpha)\ \dot{\rho}_{(n)} + \alpha \Delta t\ \dot{\rho}_{(n+1)}, \tag{6.53}$$

and applying the linearziation operator on $v_{i(n+1)}, \rho_{(n+1)}$ we get:

$$\delta v_{i(n+1)} = \alpha \Delta t \left(\delta \dot{v}_{i(n+1)} \right) \tag{6.54}$$

$$\delta \rho_{(n+1)} = \alpha \Delta t \left(\delta \dot{\rho}_{(n+1)} \right). \tag{6.55}$$

Hence we can define:

$$\delta \boldsymbol{U} = \alpha \Delta t \delta \dot{\boldsymbol{U}} \tag{6.56}$$

and going back to 6.51 we get:

$$\left(\alpha \Delta t \left(\boldsymbol{K} \right)_{t+\Delta t}^{i-1} + \boldsymbol{C}_{t+\Delta t}^{i-1} \right) \delta \dot{\boldsymbol{U}}^i = -(\boldsymbol{F})_{t+\Delta t}^{i-1} \tag{6.57}$$

6.5 Particle Update

Since this is a moving interface problem, we must update our membrane after each time step. As shown before, we proposed a mixed formulation, where the tangential velocity is solved from an Eulerian point of view and the normal movement is treated from a Lagrangian one. Because of that, the update of the interface must be done in such a way that the material position of the particles is conserved. In other words, we must ensure that the updated particles (our calculation nodes) correspond to the same points in an updated position.

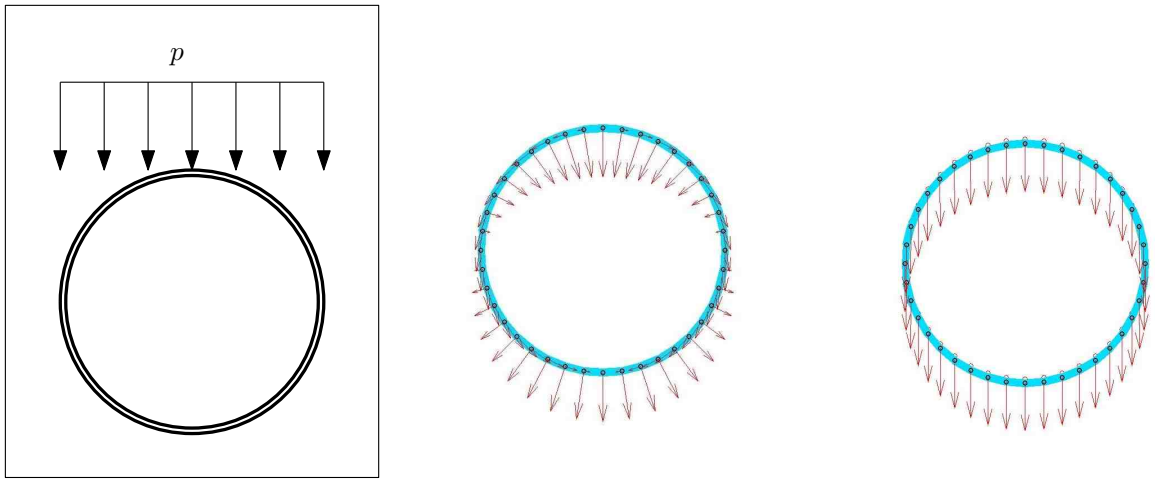
To understand better this concept, imagine a problem where we membrane is subjected to a pressure resulting in a pure translation. In figure 6.6b, we see the solution for the normal velocity after solving a vesicle under the mentioned field. As it can be seen, the updated position of the membrane corresponds to a translation of the interface, so the solution seems to be correct. However, if we moved the particles using those vectors, the particles on the top would be clutching together while the particles on the bottom would be spread more and more with time. This is caused because when we update the particles using the normal vector the updated points do not correspond to the same material point where the original particle was.

For this reason, we decided to separate the two movements of our body, the solid body motion and the deformation. Note that sice we have an Eulerian tangential velocity, our particles will never experience a rotation, but a tangential flow of some sort. Hence, the velocity of any point in the membrane can be decomposed in a solid body motion which corresponds to the velocity of the center of mass v_{CM}, and the real deformation v_d:

$$v_I = v_{CM} + v_d. \tag{6.58}$$

The resultant is the total velocity of each particle v_I, which in our case is the sum of the normal and tangential velocities that we obtained by solving the system of equations.

$$v_I = v_t + v_n. \tag{6.59}$$



(a) (b) (c)

Figure 6.6: *On the left we see a scheme of the problem that leads to a uniform translation of the vesicle. In the middle, the membrane is updated using only the normal direction. In the right the update velocity has been decomposed with a solid body motion and a deformation and the update velocity (in red) is the combination of the translation plus the deformation in the normal direction*

The velocity of the center of mass is calculated by the mass ponderated average of all the velocities of the system.

$$v_{CM} = \frac{1}{M} \sum_{i=1}^{n} v_I \rho_I. \tag{6.60}$$

Hence the final effective deformation in the normal direction on each time step, can be expressed as:

$$v_n^{eff} = v_d \cdot \hat{\boldsymbol{n}} = (v_I - v_{CM}) \cdot \hat{\boldsymbol{n}} \tag{6.61}$$

and the resultant effective tangential velocity (Eulerian):

$$v_t^{eff} = v_d \cdot \hat{\boldsymbol{n}} = (v_I - v_{CM}) \cdot \boldsymbol{g} \tag{6.62}$$

If we apply this to the previous problem, we observe that after a few time steps the updated velocity of the vesicle becomes the one of a pure translation (figure 6.6c) which is because the

updated nodes, where we constructed the weak form, correspond to the updated material points of the membrane.

This procedure, however, is valid as long as none of the particles has Dirichlet boundary condition affecting its normal displacement. If, for instance, there were particles with their normal displacement fixed, the previous statement would not be valid, since there is no solid body motion, just a deformation.

6.6 Results

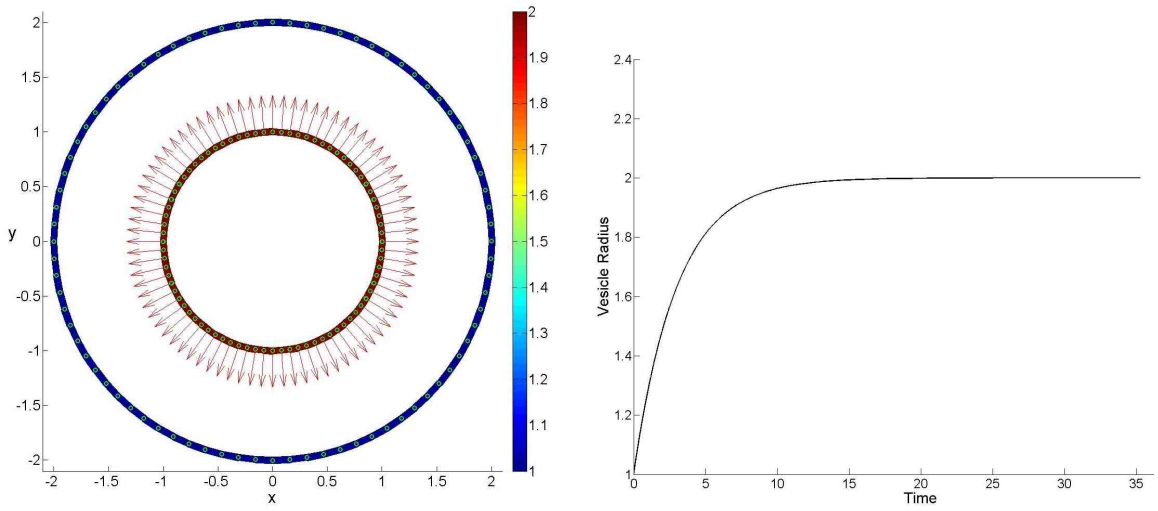
6.6.1 Convergence test: initially compressed vesicle

The first example that we will show is what happens when we have a spherical vesicle of radius $R = 1$, with an initial density twice the equilibrium value $\rho(t = 0) = \rho_0$. Obviously, the vesicle will suffer a constant expansion until the density reaches the equilibrium value everywhere, but due to the pressure-density law that we introduced (equation 3.45), it will expand more and more slowly as we reach the equilibrium state. For this reason we introduce a variable time step that will keep a constant minimum increment of $u^n_{min} = 5 \cdot 10^{-4}$. Hence, for every time step where the maximum displacement falls bellow this magnitude the time step will be doubled.

On figure 6.7, the difference between the initial and final step is shown. The green circles are the position of the field nodes used to interpolate the vesicle in both the first step and their final positions. The red arrows show the magnitude of the normal displacement in both positions, and the line color represents the value of the density at each particle.

As said, the expansion rate decreases when the vesicle radius increases and the density is less distant from the equilibrium value. For this reason, we cannot see the velocity arrows on the large vesicle on figure 6.7. This condition can be seen even more clearly on the right, where the radius of the vesicle is plotted at each time step, and it becomes very clear that the increment is more and more slow with time.

To prove the convergence of the system, we run the same problem with an increasing number



(a) (b)

Figure 6.7: *Left. 2D vesicle under constant expansion due to an initial compressed state with* $\rho = 2\rho_0$. *Right. Vesicle mean radius vs time*

of particles to observe if the error on the solution did indeed tend to 0 value. The error was computed using the fact the total mass must be conserved and that we are using a circle which remains a circle in time, which means that the following expression is true at any time step:

$$\rho(t=0)2\pi R(t=0) = \rho(t=T)2\pi R(t=T) \qquad \Rightarrow \qquad \rho(t=0)R(t=0) = \rho(t=T)R(t=T) \tag{6.63}$$

so the realative error was computed the following way:

$$Error = \frac{(\rho(t=0)R(t=0) - \bar{\rho}(t=T)\bar{R}(t=T))}{\rho(t=0)R(t=0)} \times 100 \tag{6.64}$$

where $\bar{\rho}$, $\bar{R}$, are the average value of density and radius on all the particles.

As we can see, the more particles we use the less error that we observe on the solution, so it is clear that it converges to the right solution of the problem. However, using more than 40 particles for this type of problem seems to much, and the error obtained in that particular case was not as low as we might expect. Probably, this could be related to the time integration scheme that we

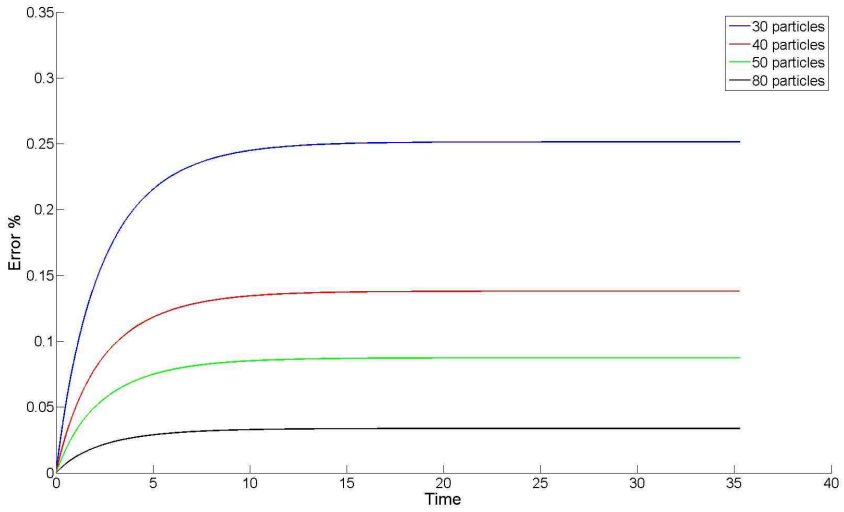


Figure 6.8: *Relative error on the vesicle expansion under initial compressed state.*

used. A Crank Nicholson or a Backward Euler schemes have relatively low convergence rate and could be introducing a lot of error. It is left as a future work the implementation of an integration scheme with a higher convergence level, like a Runge-Kutta.

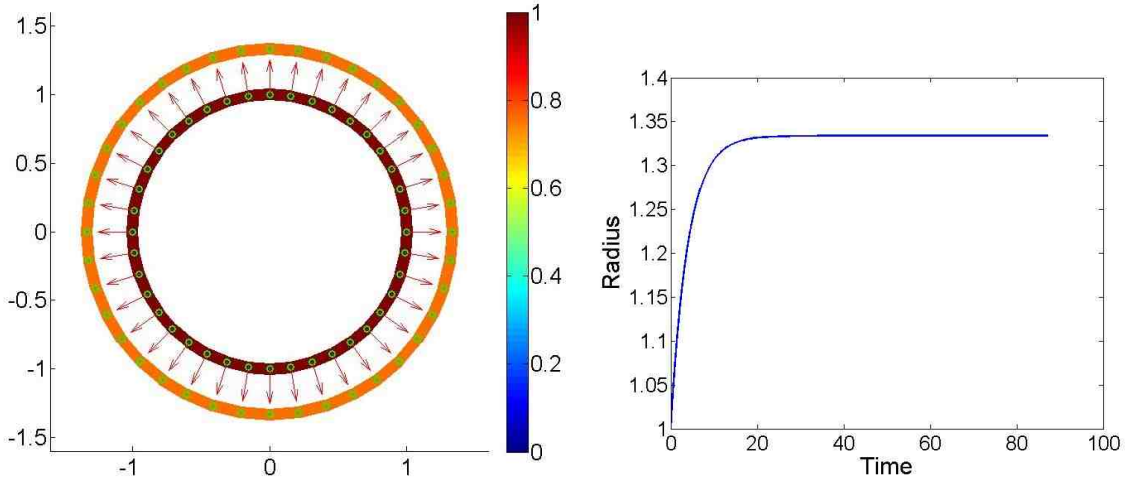
6.6.2 Membrane mechanical response: vesicle under internal pressure

With the first example we were able to test the response of a vesicle subjected to an internal pressure on the membrane. Next we will show the mechanical response of the vesicle to an external force. In this case we will introduce an internal pressure on the vesicle that will inflate it until it reaches the equilibrium forced by Laplace's law (equation 6.20). According to this equation, an equilibrium situation ($v_n = 0$) will be reached when the following relationship is accomplished:

$$a\left(\rho - \rho_0\right)^c H + b^3 = 0, \tag{6.65}$$

therefore, we introduce a constant internal pressure on the vesicle with the form:

$$b^3 = P \tag{6.66}$$



(a) (b)

Figure 6.9: *Left. 2D vesicle under constant expansion due to an internal pressure of $p = 0.25$. Right. Vesicle mean radius vs time*

Because there is nothing that breaks the symmetry of the system and all the fields are constant, the tangent velocity will remain zero at each time step, and equation 6.65 simply becomes Laplace's law. Hence, from equation 6.1 the normal velocity can be analytically calculated as:

$$v_n = \frac{a(\rho - \rho_0)^c H + b^3}{\lambda H^2 + 2\mu H^2} \tag{6.67}$$

To run this problem we chose a vesicle of initial radius $R_0 = 1$ discretized by 40 evenly distributed particles. For this example the value of $p = 0.15$, $\rho_0 = 1$ and $\mu = \lambda = 1$ were chosen. Hence by solving the previous equations for the values of a circle, it is easy to see that it reaches the equilibrium state when the density becomes $\rho = 0.75\rho 0$, and this corresponds to an equilibrium radius of $R_f = R_0/0.75$. In the figure bellow both the initial and the last step can be seen, where the colors represent the pressure and the arrows indicate both normal and tangential velocity both in the same scale.

Initially the time step was chosen to be 0.001 seconds and again, due to the profile of the pressure, the time step was adapted to maintain a minimum normal displacement of $u_{min} = 0.5 \cdot 10^{-4}$. After 502 iterations the system reached equilibrium with an error respect the real of 0.05%,

calculated as the difference between the final radius and the analytical equilibrium position.

To prove that the system actually follows the analytical expression shown in 6.67, we run the same problem using three different values for μ and λ. Figure 6.10 shows the different evolutions of the normal velocities obtained for each of the densities. Each of the solution is plotted on top of the analytical expression but it is hard to see them because they are on top of each other.

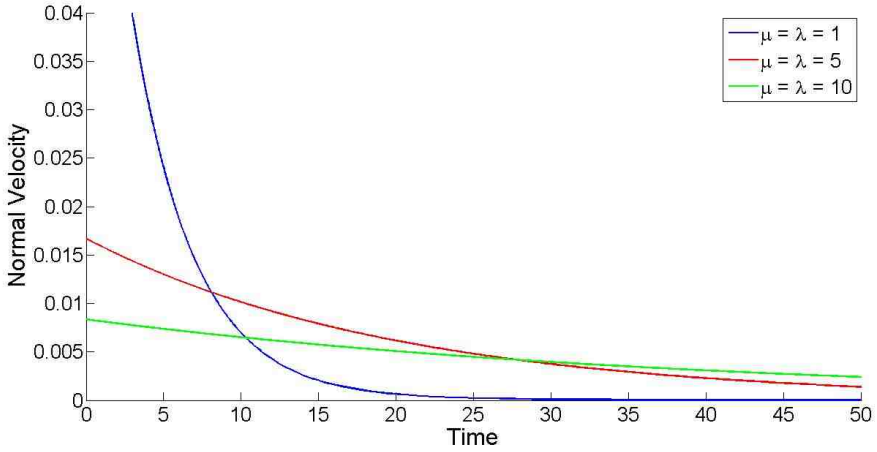
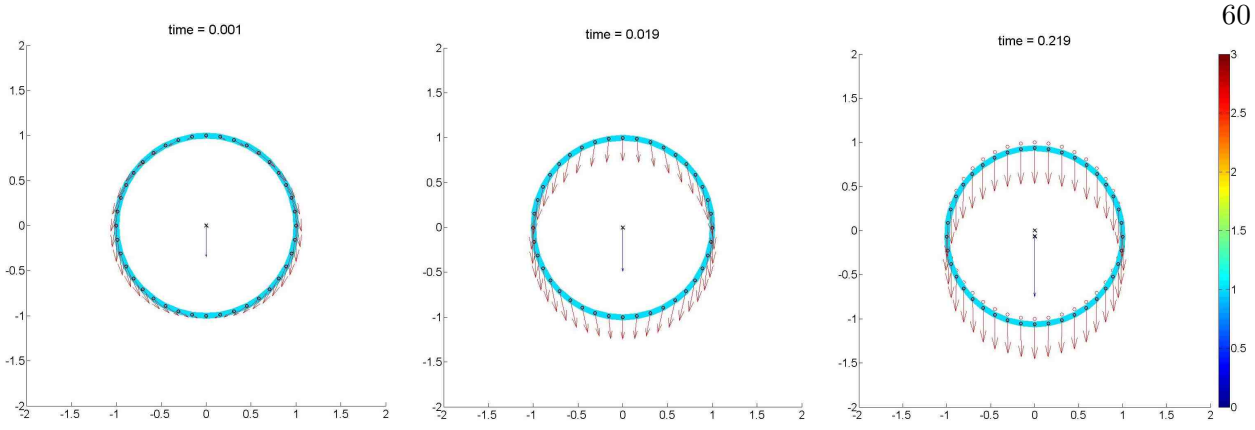


Figure 6.10: *Evolution of the normal velocity on the vesicle with time for different values of the density*

As expected, as we increase the density of the system, the normal velocity is slower in time and it takes longer for the system to reach equilibrium, however, since all of them must end at the same position, the area bellow each of the curves is the same.

6.6.3 Membrane mechanical response: vesicle translation

Finally, we show the same problem used for the discussion on the update of the particles. The problem shown in figure 6.6a is a vesicle initially under equilibrium under a constant external vertical pressure pushing it down. I assumed that the pressure only acts on the normal direction and there is no tangential force applied, as if the fluid causing the pressure perfectly slipped around the membrane. This problem is very interesting since its results depend greatly the parameters a and c of the pressure-density relationship.



(a) (b) (c)

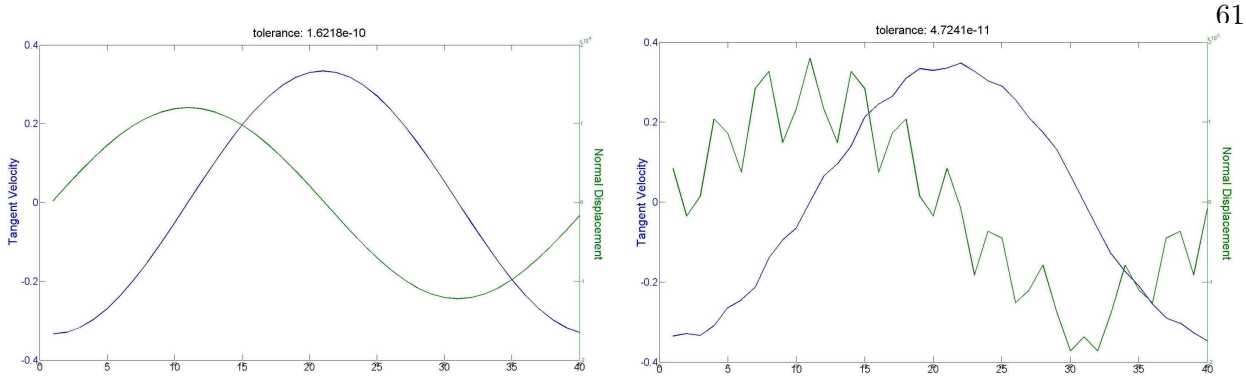
Figure 6.11: *At the beginning (left) a local tangential velocity appears trying to compress the particle on the front. As the pressure builds up, this velocity disappears and the system starts almost a pure translation (right).*

Small values of a, or high values of c, allow the system to easily get compressed or stretched without opposing much resistance. So in those cases, a lot of the energy introduced on the vesicle by the external force is wasted on recirculating the fluid flow on the membrane rather than actually moving the membrane. Therefore, in order to get almost a pure translation we decided to use a very high value of a ($a = 100$). This way, the system will not allow to build up much pressure and it will start translating very fast.

On figure 6.11 we see the evolution of the vesicle in three periods of time. The red arrows represent updated direction of the particles at each time step which is a combination of the solid body motion, depicted by the blue arrow in the center, and the deformation normal velocity. As it can be seen, at the very first time step the system has translation ($v_{CM} \neq 0$), but since it still has no opposition from the in-plane pressure it tries to recirculate the fluid on the membrane compressing it at the front part.

Once the density starts rising on the top and reducing on the bottom, the system tends to stop recirculating the fluid and it becomes a pure translation figure 6.11c.

Although the problem seemed to work fine, we encountered a major problem occurring after a few time steps. From the start, our system has very smooth solutions: figure 6.12a shows the



(a) (b)

Figure 6.12: *Left. tangential velocity and normal displacement at each particle at each time step. Right. Oscillations appearing on the same fields after a few iterations on the system.*

tangential velocity and normal displacement obtained for the first time step on all the 41 particles used to compute this vesicle. However, after several iterations, the system builds up some error which is propagated as an instability and makes the system crash.

Based on the discussion done at the end of the thesis the errors are most likely coming from the fact that we did not linearize the geometry, in certain conditions this assumption makes the equations impossible to solve and therefore the system is likely to oscillate and fail.

Chapter 7

Compressible fluid axisymmetric membrane

Once the 1D problem is solved, we will move on the the solution of a two dimensional membrane embedded on a 3D space. However, we will only solve an axisymmetric problem for now. First of all, the axisymmetric problem is very similar to the 1D problem since both of them are only represented by a line, so the implementation will be very straightforward. Second, it will be very useful to have a working 3D solution, so when we develop the full 3D problem we will be able to compare them and make sure that the solutions are correct.

The procedure to solve this problem will be exactly the same that we followed for the previous case and only the details where both formulations are different will be detailed. The weak form of a fluid membrane as well as the linearization of the solution was derived for a general three dimensional body, so the reader is referred to section 6.1 and 6.2 for the details. Hence, in a similar way as it was done in [1] we proceed to introduce the numerical interpolation of the linearized weak form for an axisymmetric fluid membrane.

7.1 Numerical Interpolation

Even though we are representing the membrane by a one dimensional line, this is a full 2D membrane embedded on a 3D space, which means that we need at least two parametric coordinates to describe it instead of one. Because it is an axisymmteric problem, we will use the axisymmetric properties of the body to simplify the math.

We start from a body discretized by N equidistant particles where we define the geometrical

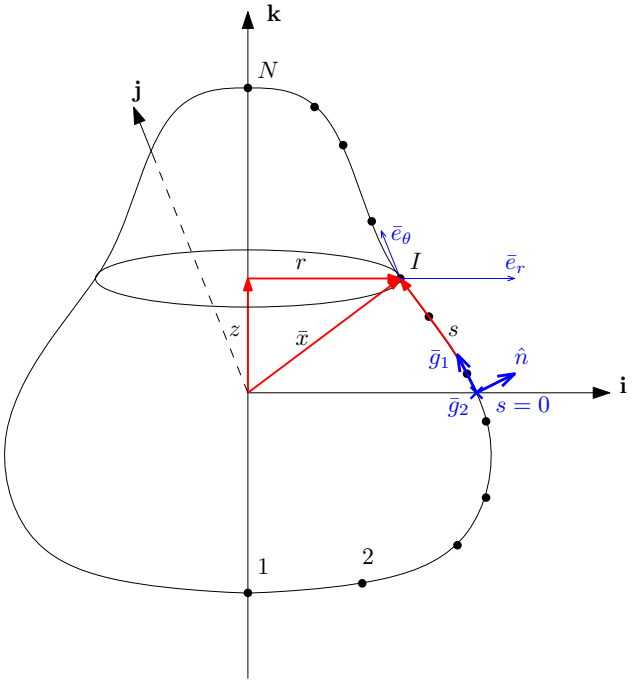


Figure 7.1: *Scheme of the axisymmetric body* Ω *and its parametrization*

properties of the body Ω using the parametric coordinates arc length s, and the revolution angle θ, figure 7.1.

$$\xi^1 = s \qquad \xi^2 = \theta \tag{7.1}$$

So the position vector in both the cartesian basis $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ and the cylindrical coordinate system r, θ, z can be expressed as:

$$\boldsymbol{x} = \boldsymbol{x}(\xi^1, \xi^2) = \left\{ \begin{array}{c} x \\ y \\ z \end{array} \right\} = \left\{ \begin{array}{l} r(s)\cos(\theta) \\ r(s)\sin(\theta) \\ z(s) \end{array} \right. \tag{7.2}$$

Note that, due to the axisymmtery of the problem, the parameters r, z that describe the surface are only a function of the arc length s, and the radial vector $\boldsymbol{e}_r$ is only a function of θ. Then, to introduce the particle method we create an integration subdomain Ω_q centered in each particle where the gauss points will be defined. For every gauss point we will collect n particles

to create the shape functions and interpolate each field. Hence we can define the following vectors with their global r, z coordinates:

$$\boldsymbol{r}^T = \{r^1, ..., r^n\} \tag{7.3}$$

$$\boldsymbol{z}^T = \{z^1, ..., z^n\}, \tag{7.4}$$

the vectors containing the shape functions and its derivatives are analogous to the 1D case in equations 6.20,6.21,6.22 and the interpolation of any field $u(s)$ is written in the same way as in equations 6.23,6.24,6.25. Hence we can define the tangential local curvilinear basis by just taking the derivative of the surface $\partial\boldsymbol{x}/\partial s, \partial\boldsymbol{x}/\partial\theta$, and the third vector as the unit normal to those two figure 7.1

$$\boldsymbol{g}_1 = \frac{\partial \boldsymbol{x}}{\partial s} = \begin{bmatrix} \boldsymbol{N}_{,s}^T\boldsymbol{r}\cos(\theta) & \boldsymbol{N}_{,s}^T\boldsymbol{r}\sin(\theta) & \boldsymbol{N}_{,s}^T\boldsymbol{z} \end{bmatrix} \tag{7.5}$$

$$\boldsymbol{g}_2 = \frac{\partial \boldsymbol{x}}{\partial \theta} = \begin{bmatrix} -\boldsymbol{N}^T\boldsymbol{r}\sin(\theta) & \boldsymbol{N}^T\boldsymbol{r}\cos(\theta) & 0 \end{bmatrix} \tag{7.6}$$

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{g}_1 \times \boldsymbol{g}_2}{|\boldsymbol{g}_1 \times \boldsymbol{g}_2|} \tag{7.7}$$

The derivatives of the tangent vectors will be needed to compute the curvature and can also be computed directly as:

$$\boldsymbol{g}_{1,1} = \begin{bmatrix} \boldsymbol{N}_{,ss}^T\boldsymbol{r}\cos(\theta) & \boldsymbol{N}_{,ss}^T\boldsymbol{r}\sin(\theta) & \boldsymbol{N}_{,ss}^T\boldsymbol{z} \end{bmatrix} \tag{7.8}$$

$$\boldsymbol{g}_{1,2} = \frac{\partial \boldsymbol{x}}{\partial s} = \begin{bmatrix} -\boldsymbol{N}_{,s}^T\boldsymbol{r}\sin(\theta) & \boldsymbol{N}_{,s}^T\boldsymbol{r}\cos(\theta) & 0\boldsymbol{z} \end{bmatrix} \tag{7.9}$$

$$\boldsymbol{g}_{2,1} = \begin{bmatrix} -\boldsymbol{N}_{,s}^T\boldsymbol{r}\sin(\theta) & \boldsymbol{N}_{,s}^T\boldsymbol{r}\cos(\theta) & 0 \end{bmatrix} \tag{7.10}$$

$$\boldsymbol{g}_{2,2} = \left[\begin{array}{ccc} -\boldsymbol{N}^T \boldsymbol{r} \cos(\theta) & -\boldsymbol{N}^T \boldsymbol{r} \sin(\theta) & 0 \end{array} \right] \tag{7.11}$$

so the metric results:

$$\begin{aligned} g_{11} &= \boldsymbol{g}_1 \cdot \boldsymbol{g}_1 = \left(\frac{\partial r}{\partial s}\right)^2 + \left(\frac{\partial z}{\partial s}\right)^2 \\ g_{12} &= \boldsymbol{g}_1 \cdot \boldsymbol{g}_1 = 0 = g_{21} \\ g_{22} &= \boldsymbol{g}_2 \cdot \boldsymbol{g}_2 = r^2 \end{aligned} \tag{7.12}$$

Introducing the shape functions into the metric we obtain:

$$g_{11} = \left(\sum_{I=1}^{n} N^I_{,s} r^I\right)\left(\sum_{I=1}^{n} N^J_{,s} r^J\right) + \left(\sum_{I=1}^{n} N^I_{,s} z^I\right)\left(\sum_{I=1}^{n} N^J_{,s} z^J\right) \tag{7.13}$$

$$g_{22} = \left(\sum_{I=1}^{n} N^I r^I\right)\left(\sum_{I=1}^{n} N^J r^J\right), \tag{7.14}$$

and defining the following matrices:

$$\boldsymbol{N}\boldsymbol{N}^T = \begin{bmatrix} N^1N^1 & N^1N^2 & \dots & N^1N^n \\ & N^2N^2 & \dots & N^2N^n \\ & & \ddots & \vdots \\ & \text{sym.} & & N^nN^n \end{bmatrix}_{(n\times n)} \tag{7.15}$$

$$\boldsymbol{N}_{,s}\boldsymbol{N}^T_{,s} = \begin{bmatrix} N^1_{,s}N^1_{,s} & N^1_{,s}N^2_{,s} & \dots & N^1_{,s}N^n_{,s} \\ & N^2_{,s}N^2_{,s} & \dots & N^2_{,s}N^n_{,s} \\ & & \ddots & \vdots \\ & \text{sym.} & & N^n_{,s}N^n_{,s} \end{bmatrix}_{(n\times n)}, \tag{7.16}$$

the metric finally becomes:

$$g_{11} = \boldsymbol{r}^T \boldsymbol{N}_{,s}\boldsymbol{N}^T_{,s}\boldsymbol{r} + \boldsymbol{z}^T \boldsymbol{N}_{,s}\boldsymbol{N}^T_{,s}\boldsymbol{z} \tag{7.17}$$

$$g_{22} = \boldsymbol{r}^T \boldsymbol{N} \boldsymbol{N}^T \boldsymbol{r}. \tag{7.18}$$

For convenience we will now use voids notation, so we can define the metric in a more compact form as:

$$\boldsymbol{g}^v = \left\{ \begin{array}{c} g_{11} \\ g_{22} \\ 2g_{12} \end{array} \right\} \tag{7.19}$$

The second fundamental form, or curvature tensor is defined as:

$$\kappa_{\alpha\beta} = \hat{\boldsymbol{n}} \cdot \boldsymbol{g}_{\alpha,\beta} \tag{7.20}$$

and particularizing for the case $\theta = 0$, the values of the four components are:

$$\begin{aligned} \kappa_{11} &= \frac{\boldsymbol{r}^T \boldsymbol{N}_{,ss} \boldsymbol{N}_{,s}^T \boldsymbol{z} - \boldsymbol{z}^T \boldsymbol{N}_{,ss} \boldsymbol{N}_{,s}^T \boldsymbol{r}}{|\boldsymbol{g}_1 \times \boldsymbol{g}_2|} \\ \kappa_{12} &= 0 = \kappa_{21} \\ \kappa_{22} &= -\frac{\boldsymbol{r}^T \boldsymbol{N} \boldsymbol{N}_{,s}^T \boldsymbol{z}}{|\boldsymbol{g}_1 \times \boldsymbol{g}_2|}. \end{aligned} \tag{7.21}$$

Again, using voids notation we define:

$$\boldsymbol{\kappa}^v = \left\{ \begin{array}{c} \kappa_{11} \\ \kappa_{22} \\ 2\kappa_{12} \end{array} \right\} \tag{7.22}$$

and since we know that $\kappa_\beta^\alpha = g^{\alpha\gamma} \kappa_{\gamma\beta}$, it is easy to see that at $\theta = 0$:

$$\kappa_1^1 = g^{11} \kappa_{11} \qquad \kappa_2^1 = \kappa_1^2 = 0 \qquad \kappa_2^2 = g^{22} \kappa_{22} \tag{7.23}$$

Next we introduce the Christoffel symbols of the first kind:

$$\Gamma_{\alpha\beta\gamma} = \frac{1}{2}\left(g_{\alpha\gamma,\beta} + g_{\beta\gamma,\alpha} - g_{\alpha\beta,\gamma}\right), \tag{7.24}$$

from which only are non zero the following four:

$$\Gamma_{111} = \frac{1}{2}\frac{\partial g_{11}}{\partial s} = \left(\boldsymbol{r}^T \boldsymbol{N}_{,ss}\boldsymbol{N}_{,s}^T\boldsymbol{r} + \boldsymbol{z}^T \boldsymbol{N}_{,ss}\boldsymbol{N}_{,s}^T\boldsymbol{z}\right) \tag{7.25}$$

$$\Gamma_{122} = \Gamma_{212} = -\Gamma_{221} = \frac{1}{2}\frac{\partial g_{22}}{\partial s} = \boldsymbol{r}^T \boldsymbol{N}_{,s}\boldsymbol{N}^T\boldsymbol{r} \tag{7.26}$$

so the Christoffel symbols of the second kind become:

$$\Gamma^{\lambda}_{\alpha\beta} = g^{\lambda\gamma}\Gamma_{\alpha\beta\gamma} \qquad g^{\lambda\gamma} = [g_{\lambda\gamma}]^{-1} \tag{7.27}$$

$$\Gamma^{1}_{11} = \frac{1}{g_{11}}\Gamma_{111} \tag{7.28}$$

$$\Gamma^{2}_{12} = \frac{1}{g_{22}}\Gamma_{122} \tag{7.29}$$

$$\Gamma^{1}_{22} = -\frac{1}{g_{11}}\Gamma_{122} \tag{7.30}$$

As we did for the 1D problem, before we continue to interpolate the weak form, we need to introduce the expression of the test functions. Which since we are only interpolating a 1D line they will be the same that we proposed in the previous section with $1 \Rightarrow \alpha$ and $\xi^1 \Rightarrow s$. For the covariant derivative of these test functions we define:

$$\begin{bmatrix} \zeta_{1|1} \\ \zeta_{2|2} \\ \left(\zeta_{1|2} + \zeta_{2|1}\right) \end{bmatrix} = \underbrace{\begin{bmatrix} W_{,s} - \Gamma^1_{11}W & 0 \\ -\Gamma^1_{22}W & 0 \\ 0 & -2\Gamma^2_{12}W \end{bmatrix}}_{\boldsymbol{S}_I} \underbrace{\begin{bmatrix} \zeta^I_1 \\ \zeta^I_2 \end{bmatrix}}_{\boldsymbol{\zeta}^I} \tag{7.31}$$

For a general axisymmetric case, we know that the local velocity $v^2 = constant$ and $\partial/\partial\xi^2 = 0$. However, to avoid losing any generality, we will keep the degree of freedom in the formulation and remove it eventually when we code it. Our system is Eulerian in the tangent direction, and Lagrangian in the normal one, hence our unknowns and incremental term vectors are:

$$\boldsymbol{U}^T = \begin{bmatrix} \ldots & v_I^1 & v_I^2 & u_I^n & \rho_I & \ldots \end{bmatrix}^T \tag{7.32}$$

$$\delta\boldsymbol{U}^T = \begin{bmatrix} \ldots & \delta v_I^1 & \delta v_I^2 & \delta u_I^n & \delta\rho_I & \ldots \end{bmatrix}^T \tag{7.33}$$

Before we interpolate each of the terms in $G^0, H^0, I^0, \delta G, \delta H, \delta I$ we define the following matrices:

$$\begin{aligned} \boldsymbol{B}_1 &= \begin{bmatrix} g_{11}N^1_{,s} & 0 & 0 & 0 & \ldots & g_{11}N^n_{,s} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \\ 0 & g_{22}N^1_{,s} & 0 & 0 & \ldots & 0 & g_{22}N^n_{,s} & 0 & 0 \end{bmatrix}_{(3\times 4n)} \\ &+ \begin{bmatrix} g_{11}\Gamma^1_{11}N^1 & 0 & 0 & 0 & \ldots & g_{11}\Gamma^1_{11}N^n & 0 & 0 & 0 \\ g_{22}\Gamma^2_{21}N^1 & 0 & 0 & 0 & \ldots & g_{22}\Gamma^2_{21}N^n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \end{bmatrix}_{(3\times 4n)} \end{aligned} \tag{7.34}$$

$$\boldsymbol{B}_2 = \begin{bmatrix} N^1_{,s} + (\Gamma^1_{11} + \Gamma^2_{12})N^1 & 0 & 0 & 0 & \ldots & N^n_{,s} + (\Gamma^1_{11} + \Gamma^2_{12})N^n & 0 & 0 & 0 \end{bmatrix}_{(1\times 4n)} \tag{7.35}$$

$$\boldsymbol{B}_3 = \begin{bmatrix} N^1 & 0 & 0 & 0 & \ldots & N^n & 0 & 0 & 0 \end{bmatrix}_{(1\times 4n)} \tag{7.36}$$

$$\boldsymbol{B}_4 = \begin{bmatrix} 0 & 0 & N^1 & 0 & \ldots & 0 & 0 & N^n & 0 \end{bmatrix}_{(1\times 4n)} \tag{7.37}$$

$$\boldsymbol{B}_5 = \begin{bmatrix} 0 & 0 & 0 & N^1 & \ldots & 0 & 0 & 0 & N^n \end{bmatrix}_{(1\times 4n)} \tag{7.38}$$

$$\boldsymbol{B}_6 = \begin{bmatrix} 0 & 0 & 0 & N^1_{,s} & \dots & 0 & 0 & 0 & N^n_{,s} \end{bmatrix}_{(1\times 4n)} \tag{7.39}$$

The full details of the interpolation can be found in Appendix B, and we are left with equations 7.41, 7.42 and 7.43. However, this still does not allow us to integrate the full 3D body. Because we are dealing with an axisymmetric problem a differential volume ring can be expressed as:

$$d\Omega = 2\pi r ds = 2\pi \boldsymbol{N}^T \boldsymbol{r} ds \tag{7.40}$$

where ds is a differential portion of arch length. Hence, all the integrals over the volume become integrals over the arc-length $\mathcal{S}$. Finally, going back to 6.11, the linearization of $G_{n+1}, H_{n+1}, I_{n+1}$ can be written as:

$$\begin{aligned} \mathfrak{L}G_{n+1} = \underbrace{\overset{0}{[G1 + G2 - G3]^0}}_{\boldsymbol{K}_G} + 2\pi\boldsymbol{\zeta}^{IT} \underbrace{\int_{\mathcal{S}} \boldsymbol{S}_I^T \left[2\mu \boldsymbol{G}\boldsymbol{B}_1 + \lambda \boldsymbol{g}_c^v \boldsymbol{B}_2 - ac(\rho-\rho_0)^{c-1}\boldsymbol{g}_c^v \boldsymbol{B}_5\right] ds}_{\boldsymbol{K}_G} \delta\boldsymbol{U} \\ + 2\pi\boldsymbol{\zeta}^{IT} \underbrace{\int_{\mathcal{S}} \boldsymbol{S}_I^T \left[-2\mu \boldsymbol{G}\boldsymbol{\kappa}^v \boldsymbol{B}_4 - \kappa_\lambda^\lambda \boldsymbol{B}_4\right] ds}_{\boldsymbol{C}_G} \delta\dot{\boldsymbol{U}} \end{aligned} \tag{7.41}$$

$$\begin{aligned} \mathfrak{L}H_{n+1} = \underbrace{\overset{0}{[H1 + H2 - H3]^0}}_{\boldsymbol{K}_H} + 2\pi\theta^I \underbrace{\int_{\mathcal{S}} W_I \left[\rho \boldsymbol{B}_2 + \rho_{,s}\boldsymbol{B}_3 + v^\alpha|_\alpha \boldsymbol{B}_5 + v^1 \boldsymbol{B}_6 - v^3 \kappa_\alpha^\alpha \boldsymbol{B}_5\right] ds}_{\boldsymbol{K}_H} \delta\boldsymbol{U} \\ + 2\pi\theta^I \underbrace{\int_{\mathcal{S}} W_I \left[\boldsymbol{B}_5 - \rho\kappa_\alpha^\alpha \boldsymbol{B}_4\right] ds}_{\boldsymbol{C}_H} \delta\dot{\boldsymbol{U}} = 0 \end{aligned} \tag{7.42}$$

$$\begin{aligned}\mathcal{L}I_{n+1} = \underbrace{\overset{0}{[I1 + I2 - I3]}^{0}}_{\boldsymbol{K}_I} + \\ 2\pi\eta^I \underbrace{\int_{\mathcal{S}} W_I \left[\left(\boldsymbol{\kappa}^v\right)^T \left(2\mu \boldsymbol{G}\boldsymbol{B}_1 + \lambda \boldsymbol{g}_c^v \boldsymbol{B}_2 - ac(\rho-\rho_0)^{c-1}\boldsymbol{g}_c^v \boldsymbol{B}_5\right)\right] ds}_{\boldsymbol{K}_I} \delta\boldsymbol{U} \\ + 2\pi\eta^I \underbrace{\int_{\mathcal{S}} W_I \left[\left(\boldsymbol{\kappa}^v\right)^T \left(-2\mu \boldsymbol{G}\boldsymbol{\kappa}^v \boldsymbol{B}_4 - \kappa_\lambda^\lambda \boldsymbol{B}_4\right)\right] ds}_{\boldsymbol{C}_I} \delta\dot{\boldsymbol{U}}\end{aligned} \tag{7.43}$$

$$\mathcal{L}F_{n+1} = -\underbrace{2\pi \int_{\mathcal{S}} b^\alpha \zeta_\alpha r ds}_{\boldsymbol{F}_G} - \underbrace{2\pi \int_{\mathcal{S}} b^3 \eta ds}_{\boldsymbol{F}_I}, \tag{7.44}$$

and we obtain the following system of equations to be solved on a Newton Raphson algorithm:

$$\begin{pmatrix} \boldsymbol{K}_G \\ \boldsymbol{K}_H \\ \boldsymbol{K}_I \end{pmatrix} \delta\boldsymbol{U} + \begin{pmatrix} \boldsymbol{C}_G \\ \boldsymbol{C}_H \\ \boldsymbol{C}_I \end{pmatrix} \delta\dot{\boldsymbol{U}} = -\begin{pmatrix} \boldsymbol{K}_G^0 + \boldsymbol{F}_G \\ \boldsymbol{K}_H^0 \\ \boldsymbol{K}_I^0 + \boldsymbol{F}_I \end{pmatrix} \tag{7.45}$$

It is worth noting that for the interpolation of the Gauss points near the symmetry axis, we used the mirror particles to avoid losing any precision. However, whenever one of the mirrored particles is collected its tangential velocity must be assembled as the $-v^T$ of its corresponding node since this particles exist at $\theta = \pi$, and therefore their tangent basis points in the opposite direction.

After that, the time integration of the equations, as well as the particle update after each time step is done exactly the same way as described for the 1D case. With all we can proceed to show some examples of how the axisymmetric code works.

7.1.1 Results

With this 3D formulation, we want to show first that the code has exactly the expected behavior, and second how the code is able to handle a more complex problem with large deformations and a non-zero tangential velocity. In the first example we show the same problem showed before where the vesicle expanded under an initially compressed state, so will be able to calculate the

error and the convergence. On the second example, the vesicle will be subjected to a lateral force that will compress it.

7.1.1.1 Inflation

First of all we consider a initially compressed spherical vesicle with $\rho = 2\rho_0$. This is the exact same problem that we showed for the 1D case, which is quite suitable to compute the error, convergence, and other factors on how the code performs.

However, the results are slightly different than what we obtained on the previous problem. In this case we have an initially flat spherical membrane with density ρ, which will deform maintaining a spherical shape since there is nothing breaking the symmetry. Since the mass of the system has to conserved we can always write:

$$4\pi R^2(t)\rho(t) = 4\pi R^2(t_0)\rho(t_0) \tag{7.46}$$

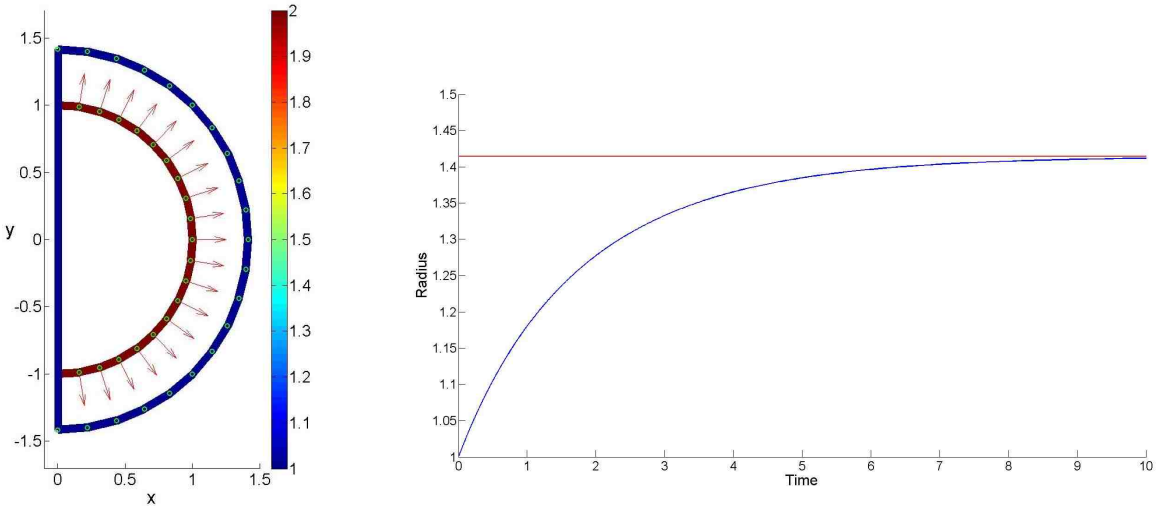
Hence, given a density at any time t the previous equation tells us what the radius should be, so we can compute the relative error at any time using the following expression:

$$Error_{\%} = \frac{\left|R^2(t_0)\rho(t_0) - R^2(t)\rho(t)\right|}{R^2(t_0)\rho(t_0)} \times 100 \tag{7.47}$$

On figure 7.2a we can see the initial and final shape of the vesicles after the inflation. Note that while in the previous case the density was proportional to the inverse of the radius, now it has the same behavior with the square of it, which means that the final shape in this problem when the system reaches equilibrium will be a sphere with radius:

$$R_{eq} = R(t_0)\sqrt{\frac{\rho(t_0)}{\rho_0}} = 1\sqrt{\frac{2}{1}} = \sqrt{2} \tag{7.48}$$

And its evolution in time has been plotted in 7.2b. As it happened before, the system never reaches equilibrium since the radius will start growing more and more slowly as the density



(a) (b)

Figure 7.2: *Left. Initially compressed vesicle with a density $\rho = 2\rho_0$ (color) which expands until it reaches equilibrium. Red arrows point the direction of the normal displacement in the same scale. Right. Evolution of the radius of the vesicle with time*

approaches ρ_0, so we used a variable time step and considered that we reached the equilibrium state when the difference between this two magnitudes is bellow 10^{-3}.

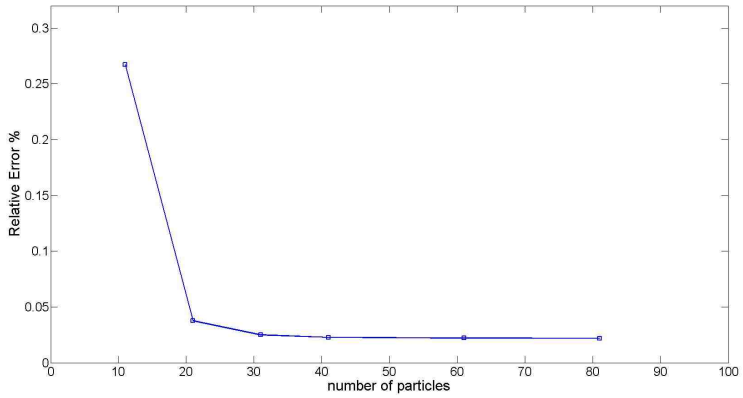


Figure 7.3: *Relative error in the solution for different number of particles used.*

Finally, to prove that the system converged to the right solution, we run the same problem using and increasing number of particles and we plotted the relative error obtained in the equilibrium state versus the number of particles used (figure 7.3). As one might see, there is a clear

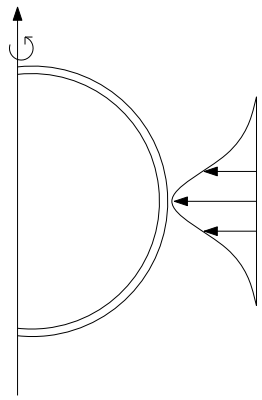


Figure 7.4: *Lateral indentation problem scheme*

convergence and reduction on the error when we run the problem from a low particle number (10) to a larger one (40). After that, the reduction in the error obtained when solving the problem was much smaller. This is probably caused because the integration scheme of the system is not good enough. Nonetheless, using 40 particles the error becomes quite small and it would not make sense to use more in a problem like that.

7.1.1.2 Ring Deformation

Finally, int his last example we would like to show how does the fluidity of the membrane affect its deformation. Since this is the main difference between a regular membrane and a fluid membrane, it is important to know how different a fluid membrane would behave than a regular one.

With this purpose, we propose a lateral indentation test on an initial spherical vesicle using two different parameters for the pressure-density relationship, and a further comparison of the results with special atention on the tangential velocity. The lateral indentation will be done using a force per unit area located around the equator of the sphere on a gauss distribution, (figure 7.4) but that only acts on the normal direction of the membrane. Hence, we will end up compressing the vesicle as if we were tightening a belt around its center.

The objective is both to show that the model is capable of enduring high deformations and,

demonstrate the importance of the membrane fluidity on the deformation. The initial membrane will be a sphere discretized by 21 equally distributed particles around its surface with $\mu = \lambda = 1$ and $c = 1$. The force will be a normal distribution on the z direction with $\mu = 0$, and $\sigma = 0.1$. We will again account for the density relationship $\frac{\rho_0}{\rho}$ that will take care of the change in area.

$$p = 5N(z, \mu, \sigma)\frac{\rho_0}{\rho} \tag{7.49}$$

The time step will be set fix to $dt = 0.001$ to make sure that the comparison is done in the exact same conditions and the case will be run for the values $a = 10$ and $a = 0.01$. The values a and c in equation 3.45, have a similar effect for they are the ones telling how difficult is to compress or stretch the vesicle. Higher values of a result in very high internal pressures for small changes in the density, so they will create a strong tangential flux to keep the difference at is minimum possible. Hence, low values of a have an opposite effect and will allow the vesicle to easily compress or stretch. The constant c has a similar effect but it can be more complex to understand due to its power nature.

First of all, we ran the problem using $a = 10$, (figure 7.5). As we can see, while the vesicle is deforming, there is never a big change on the density and is always kept close to the original stability number ρ_0, depicted in green in the figures bellow. This means that there is a strong tangential flow recirculating the molecules around the membrane to make sure that their compression is the minimum possible one.

Recall that since we do not have curvature energy in this problem, a system without external force will be in equilibrium as long as the density is kept $\rho = \rho_0$. However, as long as there is an unevenly distributed external force, the system will never be in equilibrium for it can always redistribute the molecules.

Next we run the exact same problem using a value of $a = 0.01$ (figure 7.6). As expected, in this situation we observe the cumulation of much higher densities which are concentrated around the area where we apply the force. Because the pressure built up inside the membrane is very low,

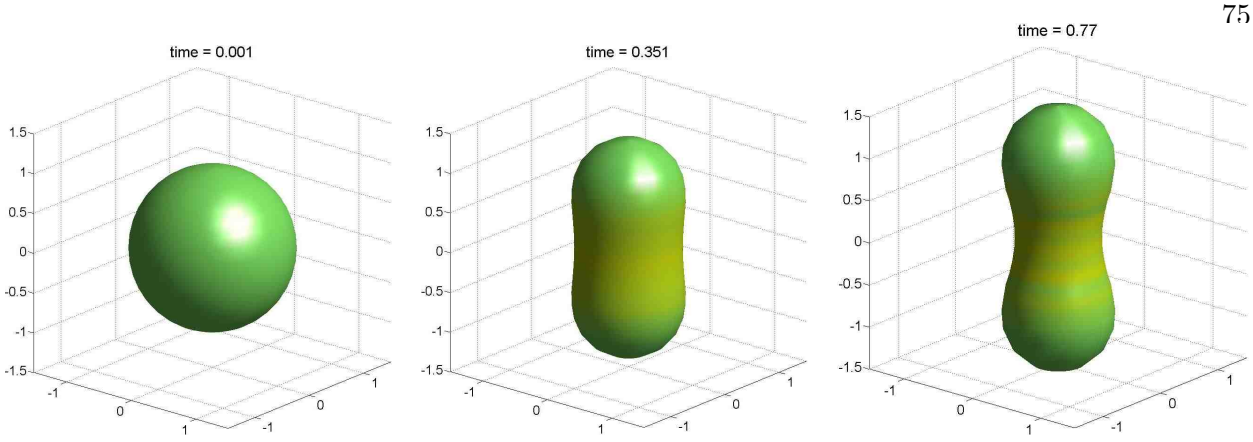


Figure 7.5: *Ring indentation of an originally spherical vesicle with parameter* $a = 10$ $c = 1$. *The colors represent the molecular density on the surface, being the green color from the figure on the left* ρ_0

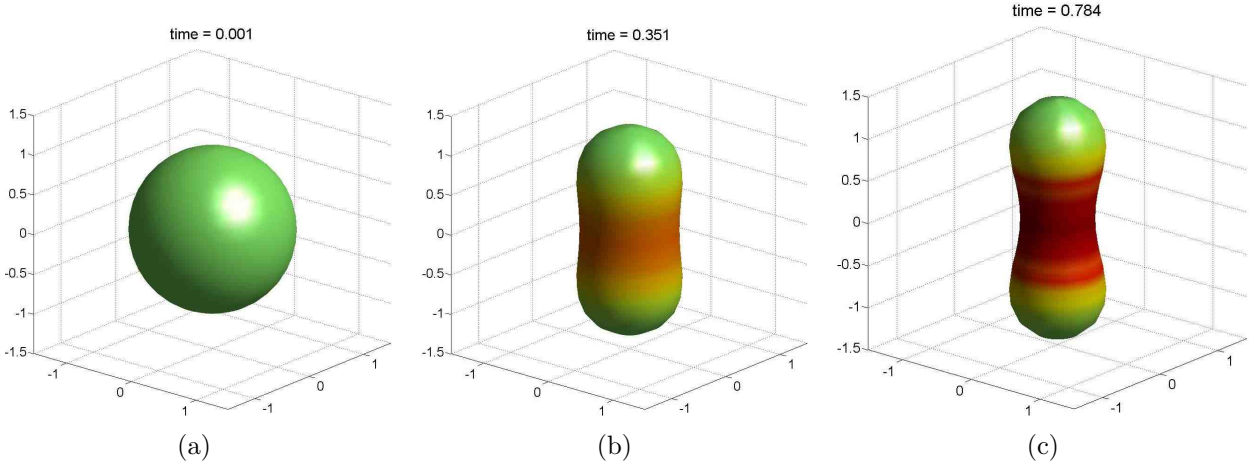


Figure 7.6: *Ring indentation of an originally spherical vesicle with parameter* $a = 0.01$ $c = 1$. *The colors represent the molecular density on the surface, being the green color from the figure on the left* ρ_0

the system rather compress the membrane than redistribute the molecules via tangential flux.

This two situations should lead to different deformed shapes with time. However, the difference between those to is not only determined by the difference on the a, parameter but also by the other parameters involved in the problem. For example, the values μ and λ are the viscosities of the system which determine the magnitude of both the normal and tangential velocities. So very

high values would lead to almost a solid membrane and very low ones would lead to fast velocities both normal and tangential. Hence, in order to compare the deformation between the two cases this values had to be kept the same.

In figure 7.7 we can see the actual difference in shape between the two situations exposed before. It is very clear that the case where the pressures inside the membrane are lower $a = 0.01$ leads to higher deformations, which is caused because the force is encountering less resistance to the external force, and by Laplace's law it can get higher curvatures.

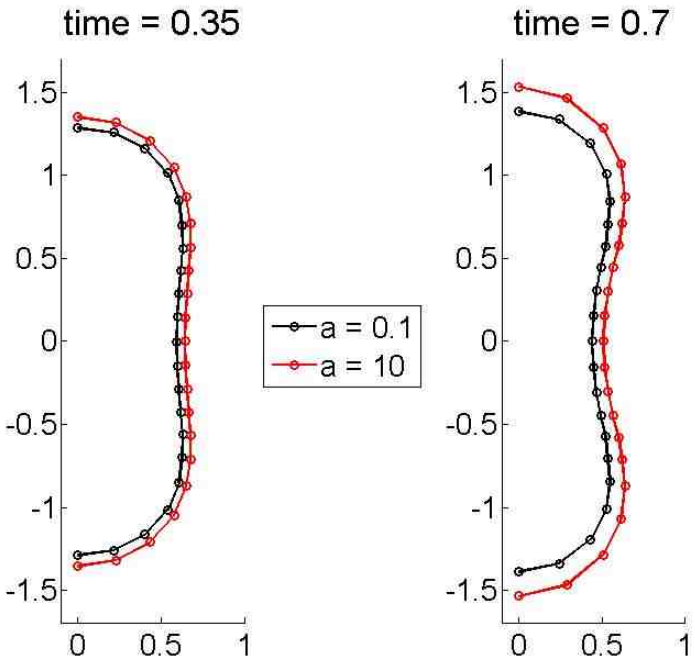


Figure 7.7: *Comparison of the shape difference for the two cases shown before.*

It is also worth noticing that since we do not account for the curvature energy, if we stopped applying the force the first case scenario would be almost in an equilibrium position while the second one would be enduring some more deformation until it reached a new equilibrium position.

Another way to look at the differences between the two cases is to look at the tangential velocity profile in time. For this purpose we look at how this velocity looks like at time $t = 0.35$. In the figure bellow on the left we can see the actual velocity profile for the case $a = 0.01$, while in the

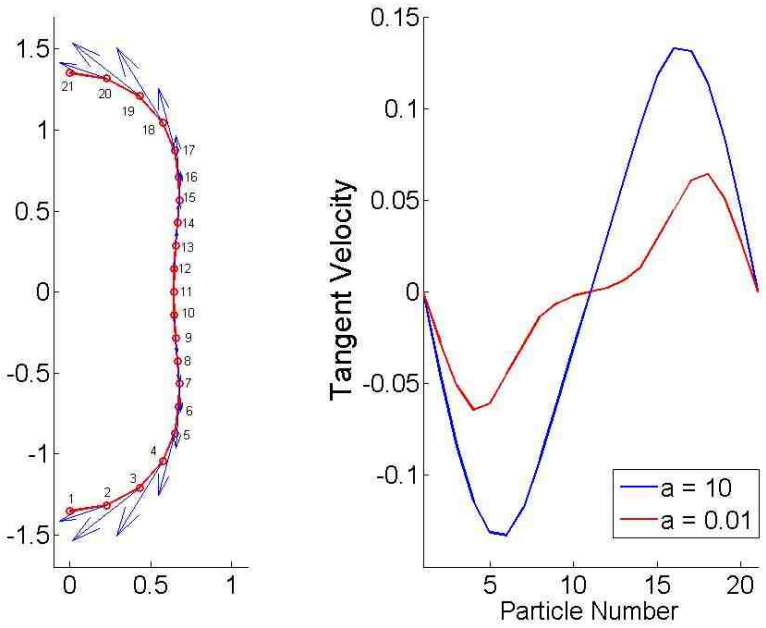


Figure 7.8: *Left. Tangential flow at the intermediate situation for* $a = 0.01$. *Right. Comparison of the tangential velocity particle by particle for the two intermediate cases.*

figure on the right, this tangential velocity has been plotted particle by particle for both situations. At first glance we can see how the high pressure case has a much higher tangential velocity which follows the density gradient.

In the second case, we first note that this velocity is quite smaller as we expected. However, we also note this flatter region at the center, which is produced by a more uniform density on the region. Remember that we are applying an equally distributed force, so the higher the density, the smaller the force is and the density becomes more uniform.

As final comment, note that on figure 7.6c we can observe some oscillations on the density because the colors are not completely uniform. This problem could probably be caused by the same instability that caused the oscillations on the translation 1D problem and which must be solved before we can run more interesting problems with this code. However, based on the results it seems more plausible that we are talking about a numerical issue of the method rather than an actual mistake on the formulation.

Chapter 8

Discussion and Conclusions

In this thesis we have presented a full new procedure to solve the mechanical equations describing a lipid coated vesicles. Unlike every other publications in this topic, we solved the membrane equations using a completely meshfree method, which not only allows us to interpolate the surface with high accuracy, but is also capable to solve the differential equations very fast and without much computational power.

On the numerical part, the fact that we use a meshfree method is crucial on our next step to the development of a full 3D model of fluid membrane. There are still very few models that are capable to solve a fluid membrane in 3D dimensions and most of them use finite elements as a numerical solver. Hence, it is hard to account for the large deformations of the systems not only because of the deformation of the mesh, but for the constant remeshing that has to be done. With this method we never have the need of refining the mesh and adding or removing particles when needed is fairly simple which makes the method very convenient for this type of problem.

Second, the fact that we use a mixed Eulerian-Lagrangian formulation is very useful in terms of avoiding the mesh (or particle position) deformation. We observed that some problems can produce great tangential flows which would move or group the particles and hinder our interpolation on a pure Lagrangian description. However, thanks to this new approach, a simple tangential flow keeps the calculation nodes in the same place avoiding the creation of clutches of particles or areas deserted of them.

Regarding the membrane formulation, we derived the full fluid interface problem from the very

basic balance of mass and balance of forces and we proved that the resulting equations matched the current models for a fluid membrane. Moreover, we added the compressibility of the lipid molecules to our system which was neglected in all previous models but experimentally it is known to play a crucial role in the system [29].

However, there are still some issues in our formulation causing our solution to oscillate that must be resolved before the model is complete. With the objective of ruling out any computational mistake in the formulation, we implemented the problem using the extended particle difference method developed by Song et al. [45], where the equations are solved directly without the need of constructing the weak form. In this case, we reached the exact same point where the problem oscillates and does not reach a solution.

Having ruled out the possibility of an implementation error, and with the aim of stabilizing our solution, we decided to add a volume constraint on the system by the means of the divergence theorem:

$$\int_S v_n dA = 0, \tag{8.1}$$

which consists of an integral constraint added directly to the problem in the same way as it was done by Vernerey et al [44]. Consequently an extra unknown corresponding to the internal pressure of the vesicle has to be added to equation 3.48 resulting in:

$$\sigma^{\alpha\beta}\kappa_{\alpha\beta} + b^3 + P = 0. \tag{8.2}$$

Adding this constraint did not solve the oscillation of the solution, therefore as next step we decided to go from a viscoelastic fluid interface to a simple elastic interface such that $\lambda = \mu \to 0$. In this context the governing equations of the problem become:

$$\dot{\rho} - \rho v_n H = 0 \tag{8.3}$$

$$-a(\rho - \rho_0)^c H + P + b^n = 0, \tag{8.4}$$

where the vesicle has a certain force distribution b^n along its surface. With this problem it is clear that one of our assumptions was fundamentally wrong. We assumed the linearization of the metric and curvature was negligible and this impedes the mean curvature H to change within a single time step. Hence, since the pressure P is constant everywhere, any point without an external force b^n was only one solution for ρ:

$$\rho = \left(\frac{P}{aH}\right)^{\frac{1}{c}} + \rho_0, \tag{8.5}$$

which cannot reproduce a smooth transition with the points where a force b^n is applied. This poses a clear inconsistency with our formulation and is therefore the most probable cause of our oscillations. Trying to resolve this problem, we started adding the variational terms of first order in the metric and curvatures following the formulation from Deserno [10], and we also changed our Dirchlet boundary condition from an external force to a prescribed displacement to avoid problems such as buckling.

Despite improving the behavior of our system, this changes did not resolve all our issues completely and we are still observing a sharp jump in the solution at the edge of our Dirichlet boundary condition. This could be caused by the fact that adding this type of condition on a membrane does create a discontinuity that can not be captured by a continuum description of the system such as the RBF interpolation. Hence, our next step will consist in adding the necessary mechanism, in terms of physics and numerics, to allow the existence of a discontinuity in curvature.

Overall, we believe that our approach will enable us to solve the equations of a fluid membrane in a much more efficient way than all currently existing methodologies and it will set the basis to add more physics to the system.

Bibliography

[1] Ashutosh Agrawal and David J. Steigmann. A model for surface diffusion of trans-membrane proteins on lipid bilayers. Zeitschrift für angewandte Mathematik und Physik, 62:549–563, 2011.

[2] Marino Arroyo and Antonio DeSimone. Relaxation dynamics of fluid membranes. Phys. Rev. E, 79(3):31915, mar 2009.

[3] B. Ataie-Ashtiani, S.M. Hassanizadeh, M. Oostrom, M.a. Celia, and M.D. White. Effective parameters for two-phase flow in a porous medium with periodic heterogeneities. Journal of Contaminant Hydrology, 49:87–109, 2001.

[4] S. N. Atluri and T. Zhu. A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. Computational Mechanics, 22(2):117–127, 1998.

[5] S.N. Atlut and T.L. Zhu. A new MLPG approach to nonlinear problems in computer modeling and simulation, 1998.

[6] Ruth E. Baltus, Appala Raju Badireddy, Wendong Xu, and Shankararaman Chellam. Analysis of Configurational Effects on Hindered Convection of Nonspherical Bacteria and Viruses across Microfiltration Membranes. Industrial & Engineering Chemistry Research, 48(5):2404–2413, mar 2009.

[7] F. Campelo and a. Hernández-Machado. Dynamic model and stationary shapes of fluid vesicles. The European Physical Journal E, 20(1):37–45, apr 2006.

[8] R Capovilla and J Guven. Stress and geometry of lipid vesicles. Journal of Physics: Condensed Matter, 16(22):S2187–S2191, 2004.

[9] Riccardo Capovilla and Jemal Guven. Stresses in lipid membranes. 6233:16, 2002.

[10] Markus Deserno. Notes on Differential Geometry with special emphasis on surfaces in R 3. Area, pages 1–64, 2004.

[11] Chi-Kun Desjardins, Benot; Lin. A SURVEY OF THE COMPRESSIBLE NAVIER-STOKES EQUATIONS Beno t Desjardins and Chi-Kun Lin. Taiwanese Journal of Mathematics, 3(2):123–137, 1999.

[12] John Dolbow. Modeling Microdomain Evolution on Giant Unilamellar Vesicles using a Phase-Field Approach Abstract Modeling Microdomain Evolution on Giant Unilamellar Vesicles using a Phase-Field Approach. 2013.

[13] N. Mai Duy and Thanh Tran Cong. Approximation of function and its derivatives using radial basis function networks. Applied Mathematical Modelling, 27(3):197–220, 2003.

[14] Anand Embar, John Dolbow, and Eliot Fried. Microdomain evolution on giant unilamellar vesicles. Biomechanics and Modeling in Mechanobiology, 12(3):597–615, 2013.

[15] Natasha Flyer, Bengt Fornberg, and Gregory A Barnett. On the Role of Polynomials in RBF-FD approximations. pages 1–16, 2014.

[16] Bengt Fornberg and Natasha Flyer. Solving PDEs with radial basis functions, volume 24. 2015.

[17] Louis Foucard, Anup Aryal, Ravindra Duddu, and Franck Vernerey. A coupled Eulerian-Lagrangian extended finite element formulation for moving interface problems in hyperelastic media.

[18] Louis Foucard and Franck J Vernerey. Theoretical and computational treatment of the mechanics of immersed interfaces. pages 1–41, 2013.

[19] Jonathan B. Freund. Numerical Simulation of Flowing Blood Cells. Annual Review of Fluid Mechanics, 46(1):130812142306005, 2013.

[20] Edward J. Fuselier and Grady B. Wright. A high-order kernel method for diffusion and reaction-diffusion equations on surfaces. Journal of Scientific Computing, 56(3):535–565, 2013.

[21] Huajian Gao, Wendong Shi, and Lambert B Freund. Mechanics of receptor-mediated endocytosis. Proceedings of the National Academy of Sciences of the United States of America, 102(27):9469–74, jul 2005.

[22] M.a. Golberg, C.S. Chen, and H. Bowman. Some recent results and proposals for the use of radial basis functions in the BEM. Engineering Analysis with Boundary Elements, 23(4):285–296, 1999.

[23] Albert Edward Green and Wolfgang Zerna. Theoretical Elasticity. Courier Corporation, 2002.

[24] Wolfrang Helfrich. Bending energy of vesicle membranes: General expressions for the first, second and third variation of the shape energy and applications to spheres and cylinders. Physical Review, 39(10):5280–5288, 1989.

[25] R Helmig, C T Miller, H Jakobs, H Class, M Hilpert, C E Kees, and J Niessner. Multiphase flow and transport modeling in heterogeneous porous media. Progress in Industrial Mathematics at ECMI 2004, 8:449–488, 2006.

[26] G Holzapfel. Nonlinear solid mechanics: A continuum approach for engineering, volume First Edit. 2000.

[27] P. Jenny, S. H. Lee, and H. a. Tchelepi. Adaptive Multiscale Finite-Volume Method for Multiphase Flow and Transport in Porous Media. Multiscale Modeling & Simulation, 3(1):50–64, 2005.

[28] P. Jenny, S. H. Lee, and H. a. Tchelepi. Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media. Journal of Computational Physics, 217:627–641, 2006.

[29] James J. Kwan and Mark A. Borden. Lipid monolayer dilatational mechanics during microbubble gas exchange, 2012.

[30] a. Laadhari, C. Misbah, and P. Saramito. On the equilibrium equation for a generalized biological membrane energy by using a shape optimization approach. Physica D: Nonlinear Phenomena, 239(16):1567–1572, aug 2010.

[31] Duc-Vinh Le and Zhijun Tan. Large deformation of liquid capsules enclosed by thin shells immersed in the fluid. Journal of Computational Physics, 229(11):4097–4116, jun 2010.

[32] Yang Li, Xin Chen, and Ning Gu. Computational investigation of interaction between nanoparticles and membranes: hydrophobic/hydrophilic effect. The journal of physical chemistry. B, 112(51):16647–53, dec 2008.

[33] G.R. Liu, L. Yan, J.G. Wang, and Y.T. Gu. Point interpolation method based on local residual formulation using radial basis functions. Structural Engineering and Mechanics, 14(6):713–732, 2002.

[34] Jerrold E Marsden and Thomas J R Hughes. MATHEMATICAL FOUNDATIONS OF ELASTICITY. Dover Publications, INC, 1983.

[35] Julian J. Pop and C. C. Wang. Acceleration waves in isotropic elastic membranes. Archive for Rational Mechanics and Analysis, 77(1):47–93, 1981.

[36] Mohammad Rahimi and Marino Arroyo. Shape dynamics, lipid hydrodynamics, and the complex viscoelasticty of bilayer membranes. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics, 86(1):1–15, 2012.

[37] a. Rosolen, C. Peco, and M. Arroyo. An adaptive meshfree method for phase-field models of biomembranes. Part I: Approximation with maximum-entropy basis functions. Journal of Computational Physics, 249:303–319, 2013.

[38] Evariste; Sanchez-Palencia, Olivier; Millet, and Fabien Béchet. Singular Problems in Shell Theory. Springer-Verlag, 2010.

[39] Varun Shankar, Grady B. Wright, Robert M. Kirby, and Aaron L. Fogelson. A Radial Basis Function (RBF)-Finite Difference (FD) Method for Diffusion and ReactionDiffusion Equations on Surfaces. Journal of Scientific Computing, 63(3):745–768, 2015.

[40] J.C. Simo and D.D. Fox. On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parametrization. Computer Methods in Applied Mechanics and Engineering, 72(3):267–304, mar 1989.

[41] Victor Andreevich Toponogov. Differential Geometry of Curves and Surfaces. Birkhäuser Boston.

[42] Vladimir P Torchilin. Recent advances with liposomes as pharmaceutical carriers. Nature reviews. Drug discovery, 4(2):145–60, feb 2005.

[43] Z. C. Tu and Z. C. Ou-Yang. Recent theoretical advances in elasticity of membranes following Helfrich's spontaneous curvature model. Advances in Colloid and Interface Science, 208:66–75, 2014.

[44] Franck Vernerey and Ronald Y S Pak. Analysis of Soft Fibers with Kinematic Constraints and Cross-Links by Finite Deformation Beam Theory. Journal of Engineering Mechanics, (August):1–11, 2011.

[45] Young-Cheol Yoon and Jeong-Hoon Song. Extended particle difference method for weak and strong discontinuity problems: part I. Derivation of the extended particle derivative approximation for the representation of weak and strong discontinuities. Computational Mechanics, 53(6):1087–1103, 2014.

Appendix A

Linearization of the Weak Form

In this appendix each step carried out to get the linearized version of the weak form (eq. 6.7) is shown in detail. The reader is referred to chapter 2 for most of the definitions used in differential geometry.

- G1

$$\delta G1 = \int_\Omega \left[\delta \left(2\mu D^{\alpha\beta} \right) \zeta_\alpha|_\beta + \left(2\mu D^{\alpha\beta} \right) \delta\zeta_\alpha|_\beta \right] d\Omega \tag{A.1}$$

The linearization of the test function is 0 because it does not depend on either v or ρ. Also, since we are not considering the normal velocity, the vesicle can not deform and the metric is assumed to be constant within a timestep, hence its linearization is zero and we only need to linearize the first term. Using the definition of the deformation rate tensor (eq. 3.44) we get:

$$\delta(2\mu D^{\alpha\beta}) = 2\mu\delta D^{\alpha\beta} = 2\mu g^{\alpha\gamma} g^{\beta\lambda} \delta D_{\gamma\lambda} = \mu g^{\alpha\gamma} g^{\beta\lambda} \left((\delta v_\gamma)|_\lambda + (\delta v_\lambda)|_\gamma - 2\kappa_{\gamma\lambda}\delta v_n \right) \tag{A.2}$$

Hence we obtain:

$$\delta G1 = \int_\Omega \left[\mu g^{\alpha\gamma} g^{\beta\lambda} \left((\delta v_\gamma)|_\lambda + (\delta v_\lambda)|_\gamma - 2\kappa_{\gamma\lambda}\delta v_n \right) \right] \zeta_\alpha|_\beta d\Omega \tag{A.3}$$

- G2

$$\delta G2 = \int_\Omega \left[\delta\left(\lambda g^{\alpha\beta} v^\lambda|_\lambda - v_n \kappa^\lambda_\lambda\right)\zeta_\alpha|_\beta\right] d\Omega + \int_\Omega \left[\left(\lambda g^{\alpha\beta} v^\lambda|_\lambda - v_n \kappa^\lambda_\lambda\right)\delta\zeta_\alpha|_\beta\right] d\Omega \tag{A.4}$$

Similarly, we only need to linearize the first term, where the lame parameter λ and the metric are constant:

$$\delta(\lambda g^{\alpha\beta} v^\lambda|_\lambda - v_n \kappa^\lambda_\lambda) = \lambda g^{\alpha\beta}\delta\left(v^\lambda|_\lambda - v_n\kappa^\lambda_\lambda\right) = \lambda g^{\alpha\beta}\left(\left(\delta v^\lambda\right)|_\lambda - \kappa^\lambda_\lambda \delta v_n\right) \tag{A.5}$$

And going back to A.4 we get:

$$\delta G2 = \int_\Omega \left[\lambda g^{\alpha\beta}\left(\left(\delta v^\lambda\right)|_\lambda - \kappa^\lambda_\lambda \delta v_n\right)\right]\zeta_\alpha|_\beta d\Omega \tag{A.6}$$

- G3

$$\delta G3 = \int_\Omega \left[\delta\left(g^{\alpha\beta} a(\rho-\rho_0)^c\right)\zeta_\alpha|_\beta\right] d\Omega + \int_\Omega \left[\left(g^{\alpha\beta} a(\rho-\rho_0)^c\right)\delta\zeta_\alpha|_\beta\right] d\Omega \tag{A.7}$$

The parameters a and c in the pressure relationship are constants, hence we obtain:

$$\delta(g^{\alpha\beta} a(\rho-\rho_0)^c) = g^{\alpha\beta} ac(\rho-\rho_0)^{c-1}\delta\rho \tag{A.8}$$

$$\delta G3 = \int_\Omega \left[g^{\alpha\beta} ac(\rho-\rho_0)^{c-1}\delta\rho\right]\zeta_\alpha|_\beta d\Omega \tag{A.9}$$

In the same way, we proceed to linearize the rest of the terms.

- H1

$$\delta H1 = \int_\Omega \left[\delta\dot{\rho}\theta\right] d\Omega + \int_\Omega \left[\dot{\rho}\delta\theta\right] d\Omega = \int_\Omega \left[\delta\dot{\rho}\theta\right] d\Omega \tag{A.10}$$

- H2

$$\delta H2 = \int_{\Omega} \left[\delta((\rho v^{\alpha})\,|_{\alpha})\theta\right] d\Omega + \int_{\Omega} \left[(\rho v^{\alpha})\,|_{\alpha}\delta\theta\right] d\Omega \tag{A.11}$$

$$\delta((\rho v^{\alpha})\,|_{\alpha}) = \rho(\delta v^{\alpha})|_{\alpha} + v^{\alpha}|_{\alpha}\delta\rho + \rho_{,\alpha}\delta v^{\alpha} + (\delta\rho)_{,\alpha}v^{\alpha} \tag{A.12}$$

$$\delta H2 = \int_{\Omega} \left[\rho(\delta v^{\alpha})|_{\alpha} + v^{\alpha}|_{\alpha}\delta\rho + \rho_{,\alpha}\delta v^{\alpha} + (\delta\rho)_{,\alpha}v^{\alpha}\right]\theta d\Omega \tag{A.13}$$

- H3

$$\delta H3 = \int_{\Omega} \left[\delta\left(\rho v^{3}\kappa_{\alpha}^{\alpha}\right)\theta\right] d\Omega + \int_{\Omega} \left[\left(\rho v^{3}\kappa_{\alpha}^{\alpha}\right)\delta\theta\right] d\Omega \tag{A.14}$$

$$\delta\left(\rho v^{3}\kappa_{\alpha}^{\alpha}\right) = v^{3}\kappa_{\alpha}^{\alpha}\delta\rho + \rho\kappa_{\alpha}^{\alpha}\delta v^{3} \tag{A.15}$$

$$\delta H3 = \int_{\Omega} \left[\left(v^{3}\kappa_{\alpha}^{\alpha}\delta\rho + \rho\kappa_{\alpha}^{\alpha}\delta v^{3}\right)\theta\right] d\Omega \tag{A.16}$$

For the last three terms, using A.2, A.5 and A.8, and recalling that we assumed the linearization of the metric tensor and the curvature to be zero, we directly obtain:

- I1

$$\delta I1 = \int_{\Omega} \left[\mu g^{\alpha\gamma}g^{\beta\lambda}\left((\delta v_{\gamma})|_{\lambda} + (\delta v_{\lambda})|_{\gamma} - 2\kappa_{\gamma\lambda}\delta v_{n}\right)\kappa_{\alpha\beta}\right]\eta d\Omega \tag{A.17}$$

- I2

$$\delta I2 = \int_{\Omega} \left[\lambda g^{\alpha\beta}\left(\left(\delta v^{\lambda}\right)|_{\lambda} - \kappa_{\lambda}^{\lambda}\delta v_{n}\right)\kappa_{\alpha\beta}\right]\eta d\Omega \tag{A.18}$$

- I3

$$\delta I3 = \int_{\Omega} \left[g^{\alpha\beta} \kappa_{\alpha\beta} ac(\rho - \rho_0)^{c-1} \delta\rho \right] \eta d\Omega \tag{A.19}$$

Appendix B

Numerical interpolation of the weak form

Having derived each of the linearized terms in the weak form, and using all the definitions on chapters 6 and 7, we can interpolate each of the terms in $G^0, H^0, \delta G, \delta H$ obtaining:

- G1

$$\delta G1 = \int_\Omega \left[\mu \underbrace{g^{\alpha\gamma} g^{\beta\lambda} \left((\delta v_\gamma)|_\lambda + (\delta v_\lambda)|_\gamma - 2\kappa_{\gamma\lambda}\delta v_n \right)}_{\delta D^{\alpha\beta}} \right] \zeta_\alpha|_\beta d\Omega \tag{B.1}$$

Using voids notation $\boldsymbol{D}$ can be rewritten as:

$$\boldsymbol{D}^v = \begin{Bmatrix} \delta D^{11} \\ \delta D^{22} \\ 2\delta D^{12} \end{Bmatrix} = \underbrace{\begin{bmatrix} [g^{11}]^2 & 0 & 0 \\ 0 & [g^{22}]^2 & 0 \\ 0 & 0 & g^{11}g^{22} \end{bmatrix}}_{\boldsymbol{G}} \begin{Bmatrix} \delta D_{11} \\ \delta D_{22} \\ 2\delta D_{12} \end{Bmatrix} \tag{B.2}$$

Using 3.44 and 6.21 the covariant form of $\delta \boldsymbol{D}$ can be expressed as:

$$\delta D_{\gamma\lambda} = \frac{1}{2} \left(g_{\mu\gamma} \sum \frac{\partial N^I}{\partial \xi^\lambda} \delta v_I^\mu + g_{\mu\lambda} \sum \frac{\partial N^J}{\partial \xi^\gamma} \delta v_J^\mu + g_{\mu\gamma}\Gamma^\mu_{\lambda\omega} \sum N^K \delta v_K^\omega + g_{\mu\lambda}\Gamma^\mu_{\gamma\omega} \sum N^K \delta v_K^\omega \right) \tag{B.3}$$

So finally, $\delta G1$ becomes:

$$\delta G1 = \int_{\Omega} \boldsymbol{\zeta}^{IT} \boldsymbol{S}_I^T \left[2\mu \boldsymbol{G} \left(\boldsymbol{B}_1 \delta \boldsymbol{U} - \boldsymbol{\kappa}^v \boldsymbol{B}_4 \delta \dot{\boldsymbol{U}} \right) \right] d\Omega \tag{B.4}$$

Similarly we can obtain:

$$G1^0 = \int_{\Omega} \boldsymbol{\zeta}^{IT} \boldsymbol{S}_I^T \left[2\mu \boldsymbol{G} \left(\boldsymbol{B}_1 \boldsymbol{U}^0 - \boldsymbol{\kappa}^v \boldsymbol{B}_4 \dot{\boldsymbol{U}}^0 \right) \right] d\Omega \tag{B.5}$$

- G2

$$\delta G2 = \int_{\Omega} \left[\lambda g^{\alpha\beta} \left(\left(\delta v^{\lambda} \right) |_{\lambda} - \kappa_{\lambda}^{\lambda} \delta v_n \right) \right] \zeta_{\alpha}|_{\beta} d\Omega \tag{B.6}$$

Therefore $\delta G2, G2^0$ can be written as:

$$\delta G2 = \int_{\Omega} \boldsymbol{\zeta}^{IT} \boldsymbol{S}_I^T \left[\lambda \boldsymbol{g}_c^v (\boldsymbol{B}_2 \delta \boldsymbol{U} - \kappa_{\lambda}^{\lambda} \boldsymbol{B}_4 \delta \dot{\boldsymbol{U}}) \right] d\Omega \tag{B.7}$$

$$G2^0 = \int_{\Omega} \boldsymbol{\zeta}^{IT} \boldsymbol{S}_I^T \left[\lambda \boldsymbol{g}_c^v (\boldsymbol{B}_2 \boldsymbol{U}^0 - \kappa_{\lambda}^{\lambda} \boldsymbol{B}_4 \dot{\boldsymbol{U}}^0) \right] d\Omega \tag{B.8}$$

- G3

$$\delta G3 = \int_{\Omega} \left[g^{\alpha\beta} ac(\rho - \rho_0)^{c-1} \delta\rho \right] \zeta_{\alpha}|_{\beta} d\Omega \tag{B.9}$$

The expression for $G3^0, \delta G3$ are:

$$\delta G3 = \int_{\Omega} \boldsymbol{\zeta}^{IT} \boldsymbol{S}_I^T \left[ac(\rho - \rho_0)^{c-1} (\boldsymbol{g}_c^v \boldsymbol{B}_5) \delta \boldsymbol{U} \right] d\Omega \tag{B.10}$$

$$G3^0 = \int_{\Omega} \boldsymbol{\zeta}^{IT} \boldsymbol{S}_I^T \left[a \boldsymbol{g}_c^v \left[\boldsymbol{B}_5 \boldsymbol{U}^0 \right]^c \right] d\Omega \tag{B.11}$$

- H1

$$\delta H1 = \int_{\Omega} \left[\delta\dot{\rho}\right] \theta d\Omega \tag{B.12}$$

$$\delta H1 = \int_{\Omega} \theta^I W_I \left[\boldsymbol{B}_5 \delta\dot{\boldsymbol{U}}\right] d\Omega \tag{B.13}$$

Also:

$$H1^0 = \int_{\Omega} \theta^I W_I \left[\boldsymbol{B}_5 \dot{\boldsymbol{U}}^0\right] d\Omega \tag{B.14}$$

- H2

$$\delta H2 = \int_{\Omega} \left[\rho(\delta v^\alpha)|_\alpha + v^\alpha|_\alpha \delta\rho + \rho_{,\alpha}\delta v^\alpha + (\delta\rho)_{,\alpha} v^\alpha\right] \theta d\Omega \tag{B.15}$$

And expanding each term separately we obtain:

*

$$\rho(\delta v^\alpha)|_\alpha = \rho \boldsymbol{B}_2 \delta\boldsymbol{U} \tag{B.16}$$

*

$$v^\alpha|_\alpha \delta\rho = \boldsymbol{B}_5 \delta\boldsymbol{U} \tag{B.17}$$

*

$$\rho_{,\alpha}\delta v^\alpha = \rho_{,1}\delta v^1 + \underbrace{\rho_{,2}}_{0}\delta v^2 = \rho_{,1}\boldsymbol{B}_3\delta\boldsymbol{U} \tag{B.18}$$

*

$$(\delta\rho)_{,\alpha} v^\alpha = (\delta\rho)_{,1} v^1 + \underbrace{(\delta\rho)_{,2}}_{0} v^2 = v^1 \boldsymbol{B}_6 \delta\boldsymbol{U} \tag{B.19}$$

Therefore, $\delta H2$ can be finally written as:

$$\delta H2 = \int_{\Omega} \theta^I W_I \left[\left(v^{\alpha}|_{\alpha} \boldsymbol{B}_5 + v^1 \boldsymbol{B}_6 + \rho \boldsymbol{B}_2 + \rho_{,1} \boldsymbol{B}_3 \right) \delta \boldsymbol{U} \right] d\Omega \tag{B.20}$$

For the term $H2^0$ we have:

$$H2^0 = \int_{\Omega} \theta \left[\left(\rho v^{\alpha} \right) |_{\alpha} \right] d\Omega = \int_{\Omega} \theta \left[v^{\alpha} \rho_{,\alpha} + \rho v^{\alpha}|_{\alpha} \right] d\Omega \tag{B.21}$$

Hence:

$$H2^0 = \int_{\Omega} \theta^I W_I \left[\left(\boldsymbol{B}_3 \boldsymbol{U}^0 \right) \left(\boldsymbol{B}_6 \boldsymbol{U}^0 \right) + \left(\boldsymbol{B}_5 \boldsymbol{U}^0 \right) \left(\boldsymbol{B}_2 \boldsymbol{U}^0 \right) \right] d\Omega \tag{B.22}$$

- H3

$$\delta H3 = \int_{\Omega} \left[\left(v^3 \kappa^{\alpha}_{\alpha} \delta \rho + \rho \kappa^{\alpha}_{\alpha} \delta v^3 \right) \theta \right] d\Omega \tag{B.23}$$

$$\delta H3 = \int_{\Omega} \theta^I W_I \left[v^3 \kappa^{\alpha}_{\alpha} \boldsymbol{B}_5 \delta \boldsymbol{U} + \rho \kappa^{\alpha}_{\alpha} \boldsymbol{B}_4 \delta \dot{\boldsymbol{U}} \right] d\Omega \tag{B.24}$$

$$H3^0 = \int_{\Omega} \theta^I W_I \left[\kappa^{\alpha}_{\alpha} \left(\boldsymbol{B}_5 \boldsymbol{U}^0 \right) \left(\boldsymbol{B}_4 \dot{\boldsymbol{U}}^0 \right) \right] d\Omega \tag{B.25}$$

The last three terms can be done easily since they are exactly the same as the G terms multiplied by $k_{\alpha\beta}$. Hence

- I1

$$\delta I1 = \int_{\Omega} \left[\mu g^{\alpha\gamma} g^{\beta\lambda} \left((\delta v_{\gamma})|_{\lambda} + (\delta v_{\lambda})|_{\gamma} - 2\kappa_{\gamma\lambda} \delta v_n \right) \kappa_{\alpha\beta} \right] \eta d\Omega \tag{B.26}$$

$$\delta I1 = \int_{\Omega} \eta^I W_I \left[2\mu \left(\boldsymbol{\kappa}^v \right)^T \boldsymbol{G} \left(\boldsymbol{B}_1 \delta \boldsymbol{U} - \boldsymbol{\kappa}^v \boldsymbol{B}_4 \delta \dot{\boldsymbol{U}} \right) \right] d\Omega \tag{B.27}$$

$$I1^0 = \int_{\Omega} \eta^I W_I \left[2\mu \left(\boldsymbol{\kappa}^v \right)^T \boldsymbol{G} \left(\boldsymbol{B}_1 \boldsymbol{U}^0 - \boldsymbol{\kappa}^v \boldsymbol{B}_4 \dot{\boldsymbol{U}}^0 \right) \right] d\Omega \tag{B.28}$$

- I2

$$\delta I2 = \int_{\Omega} \left[\lambda g^{\alpha\beta} \left(\left(\delta v^{\lambda} \right) |_{\lambda} - \kappa^{\lambda}_{\lambda} \delta v_n \right) \kappa_{\alpha\beta} \right] \eta d\Omega \tag{B.29}$$

$$\delta I2 = \int_{\Omega} \eta^I W_I \left[\lambda \left(\boldsymbol{\kappa}^v \right)^T \boldsymbol{g}^v_c (\boldsymbol{B}_2 \delta \boldsymbol{U} - \kappa^{\lambda}_{\lambda} \boldsymbol{B}_4 \delta \dot{\boldsymbol{U}}) \right] d\Omega \tag{B.30}$$

$$I2^0 = \int_{\Omega} \eta^I W_I \left[\lambda \left(\boldsymbol{\kappa}^v \right)^T \boldsymbol{g}^v_c (\boldsymbol{B}_2 \boldsymbol{U} - \kappa^{\lambda}_{\lambda} \boldsymbol{B}_4 \dot{\boldsymbol{U}}) \right] d\Omega \tag{B.31}$$

- I3

$$\delta I3 = \int_{\Omega} \left[g^{\alpha\beta} \kappa_{\alpha\beta} ac (\rho - \rho_0)^{c-1} \delta \rho \right] \eta d\Omega \tag{B.32}$$

$$\delta I3 = \int_{\Omega} \eta^I W_I \left[ac (\rho - \rho_0)^{c-1} ((\boldsymbol{\kappa}^v)^T \boldsymbol{g}^v_c \boldsymbol{B}_5) \delta \boldsymbol{U} \right] d\Omega \tag{B.33}$$

$$\delta I3 = \int_{\Omega} \eta^I W_I \left[a \left(\boldsymbol{\kappa}^v \right)^T \boldsymbol{g}^v_c \left[\boldsymbol{B}_5 \boldsymbol{U}^0 \right]^c \right] d\Omega \tag{B.34}$$