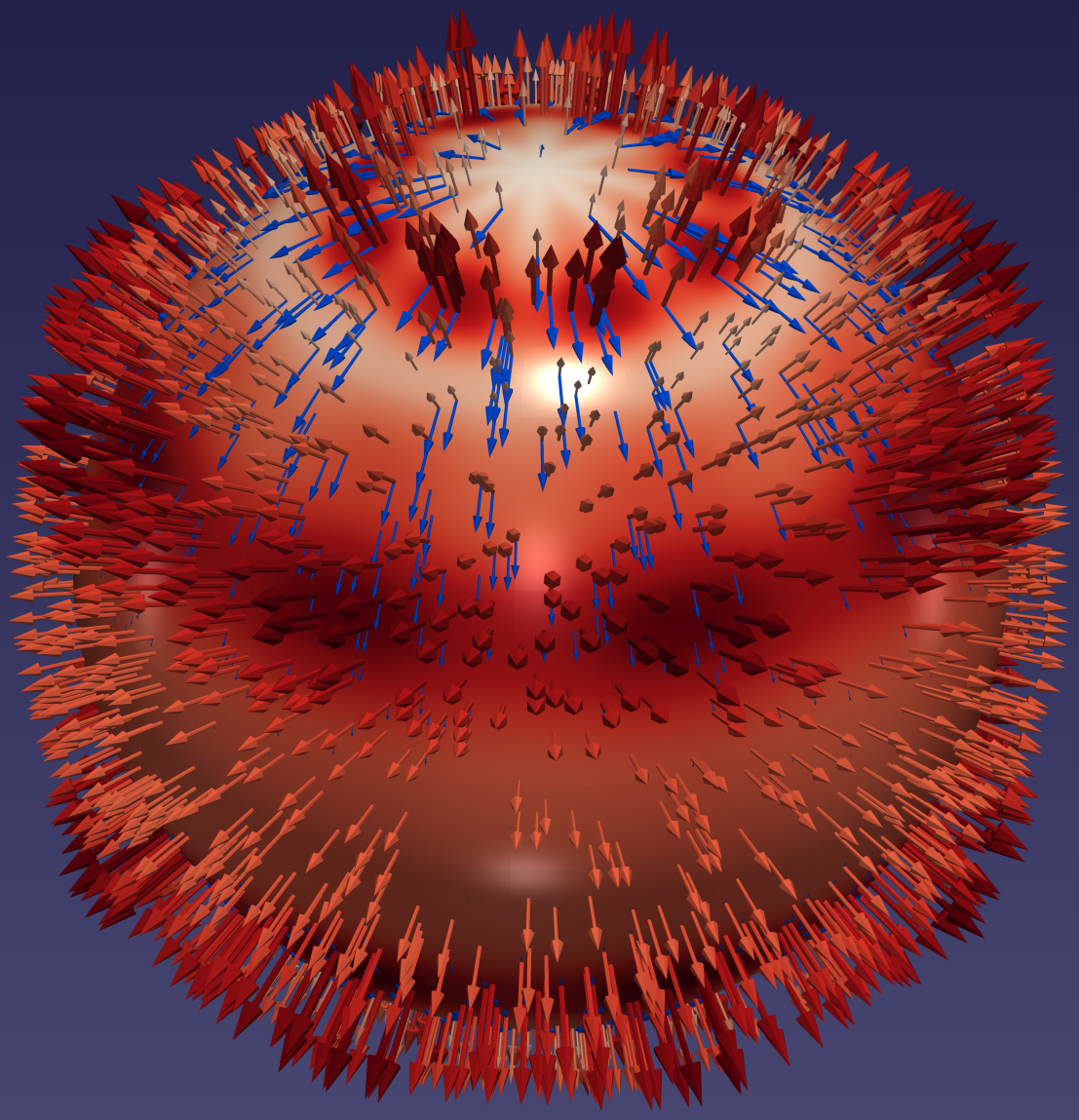
Isogeometric analysis of fluid cellular membranes

Application of discrete exterior calculus and isogeometric analysis to Stokes flow on time-evolving surfaces

Master Thesis Applied Mathematics

Douwe Bosma



Delft University of Technology

TU Delft

Isogeometric analysis of fluid cellular membranes

Application of discrete exterior calculus and isogeometric analysis to Stokes flow on time-evolving surfaces

by

Douwe Bosma

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday June 1, 2022 at 10:30 AM.

Student number:	4379624	
Project duration:	February, 2021 – June, 2022	
Thesis committee:	Prof. Dr. Ir. K. Vuik	TU Delft
	Prof. Dr. Ir. A. Heemink	TU Delft
	Dr. Ir. D. Toshniwal	TU Delft, supervisor
	H. Verhelst Msc	TU Delft, daily supervisor

Cover: A visualization of the solution to the thin film equation. It is described in the results section of this report.

An electronic version of this thesis is available at http://repository.tudelft.nl/.

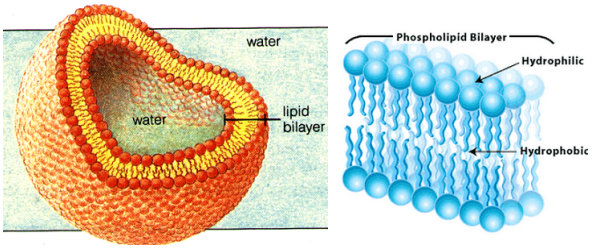
TU Delft

Abstract

An isogeometric finite element method for incompressible fluid film equations is presented. The method can be applied to numerically model the behaviour of thin cellular membranes, such as lipid bilayers. The membranes are represented by infinitely thin closed surfaces. Both the surface parametrization and analysis are based on state-of-the-art polar spline spaces. These spaces are defined such that a C^1 continuous genus 0 surface can be constructed. At the discrete setting, point-wise conservation of mass is attained, using the framework of discrete exterior calculus. Therefore, the polar spline spaces are called divergence conforming. Time discretization of the highly non-linear system is done via the fixed-point iterations. It is found that for certain non-uniformly curved domains, the iterations converge and time stepping can be performed. However, for surfaces that are closely resembling a perfect sphere, the iterations are not stable for any Δt. The solution is parameter-dependent and this indicates a possible bug in the Matlab code.

Contents

1 | Introduction



In this master thesis a computational method that can be used to study the behavior of thin fluid membranes is investigated. A biological example of these membranes is the cell boundary of eukaryotic cells. The eukaryotic cell is very common in almost all non-bacterial living organisms [1] and its membrane consists of two layers of lipids, called lipid bilayers. Lipid bilayers possess a very interesting fluid-solid duality: in-plane, the lipids are able to flow, while out-of-plane, the surface behaves like a hyperelastic solid material [2]. The cell membrane plays a key role in cell behavior and interaction with the cell surroundings. It can undergo vast shape deformations to enable all kinds of cellular processes that often involve budding [3]. The importance of the cell membrane can be seen, for example, in endocytosis, which is the process of bringing both nutrients and pathogens into the cell [4–6] and in cytokinesis, the process where a mother cell divides into two daughter cells [7]. Additionally, the behavior of the cell membrane is crucial for membrane repair [8, 9] and cell adhesion [10, 11]. Another interesting example of the role of the cell membrane is the red blood cell, where a skeleton-bilayer interaction results in a very flexible but tough membrane [12–14]. Furthermore, cell migration, is closely related to the properties of the cell membrane and the presence of local curvature [15–18].

The motivation for this research is twofold. On the one hand, it is useful to get experience with the equations related to thin fluid membranes and how to solve these in an elegant and reliable way. Among others this concerns the circumvention of the use of Lagrange multiplayer's or penalty methods to enforce in-plane flows or divergence free flows on curved surfaces and how to parameterize these surfaces. These experiences can be deployed later to extend the methods. On the other hand, these models find different applications including numerical weather prediction [19] and the already mentioned cellular membranes. modeling the behaviour of fluid cell membranes and cell interaction finds its use in, for example, the understanding of the process of the HIV virus fusion with immune cells [20].

Typically, the in-plane behavior of lipid bilayers is that of a viscous and almost incompressible fluid [2, 21]. As a consequence of the incompressiblility constraint, the in-plane and out-of-plane behavior is intertwined, leading to a coupled system of equations. For the in-plane flow description the surface Stokes equation is very suited, as a consequence of the low Reynolds number. On top of that, the surface itself is time-evolving, described by an out-of-plane shape equation. Taking the limit to infinitely thin membranes, leads to the Stokes equation on a two dimensional curved surface that evolves in time. It is also known as the fluid film equation [22]. The fluid film equation is used in modeling foam behavior and foam life time [23], which is valuable in multiple engineering branches such as material design [24–26]. Fluid film equations also find usage in numerically modeling fluid interfaces, a phenomenon that has a very wide range of applications, including lipid bilayers [27–30].

Numerical solutions to partial differential equations (PDEs) do not, in general, satisfy fundamental conservation laws, possibly leading to instability and inconsistency of the numerical method [31, 32]. Instabilities cause failure of the method, while inconsistencies often lead to conservation-violating solutions that can easily remain unnoticed [33]. Therefore, the formulation of numerical methods that mimic important conservation properties of the physical problem at the discrete level, are of great importance [34]. This importance is emphasized by the high complexity of surface PDEs compared to PDEs in Euclidean space, especially if the domain itself is time-evolving. A framework that can be utilized to represent conservation laws in a discrete but point-wise exact manner, is called discrete exterior calculus (DEC) [35, 36]. Exterior calculus is the study of n-dimensional differentiable manifolds, and it relies heavily on differential forms to express scalar and vector fields. It is the foundation of finite element exterior calculus [37, 38], where the properties of differential forms are maintained on the discrete level. Examples of differential forms used in surface (Navier-)Stokes finite element formulations can be found

in Refs. [39–41]. Formulating a PDE in terms of differential forms has the benefit of an obvious separation of conservation laws (that one wants to discretize exactly) and constitutional models (that will be approximated) [34].

The cell membrane will be represented by a 2-dimensional midsurface with zero thickness, much akin to the Kirchoff-Love (KL) thin shell model [42–45]. The use of this model is popular because of its simplicity and efficiency in terms of degrees of freedom per control point. A downside of the KL model is the minimum requirement of C^1 surface continuity, which guarantees that surface quantities, such as the curvature tensor, are well defined. An elegant and effective way to overcome this problem is by representing the geometry with C^k continuous ($k \in \mathbb{N}$) B-spline basis functions. These replace the standard piece-wise polynomial basis functions, which are only C^0 smooth on element boundaries. Additionally, the same B-spline basis functions that represent the geometry are used for analysis. Exactly this is the concept of isogeometric analysis (IGA) as introduced in Refs. [46, 47]. The term IGA refers to 're-using' the geometry function spaces for analysis and not necessarily to the use of smooth functions such as B-splines. Nonetheless, in this research IGA will also incorporate the employment of these B-spline basis functions. While the higher order continuity of the geometry is the main reason of adopting IGA in this research, IGA has other desirable features. In general, IGA is more efficient [48, 49] and more robust [50] than ordinary finite element analysis (FEA). Moreover, as a consequence of the higher order smoothness, the existence of higher order derivatives of discretized FE functions is guaranteed. State-of-the art polar spline spaces [41, 51] will be used to construct sphere-like surfaces that are C^1 everywhere, even at the poles where element boundaries are collapsed to singular points.

The goal of this research is to numerically model the surface Stokes equation coupled to the out-of-plane deformation of a thin sphere-like surface, using divergence conforming spline spaces. The direct focus will be on the development of a numerical method that is naturally consistent with the continuous conservation laws, without resorting to penalty methods or Lagrange multipliers to enforce the desired properties. Specifically, the governing fluid film differential equations for cell membranes and lipid bilayers will be solved, adopting previously derived constitutional models [39, 52]. The research question reads

> *How can thin membranes be numerically modeled using divergence conforming isogeometric analysis?*

To answer this main question a few different subjects need to be addressed. These can be formulated as two sub-questions of the research question:

1. *How to formulate and implement a divergence conforming computational method based on discrete exterior calculus and isogeometric analysis?*
2. *How to perform time integration for this highly non-linear system of discrete equations?*

The field of research is a combination of viscous flows on curved surfaces, out-of-plane deformations of thin shells, divergence conforming exterior finite elements and state-of-the-art C^1 smooth polar spline spaces. The novelty of this research lies in the time-integration of the coupled system of equations on a polar spline defined geometry.

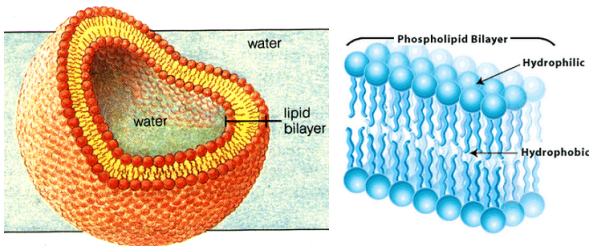
In this report, chapter 2 reviews the available literature concerning computational methods for lipid bilayers and corresponding models. Next, chapter 3 will introduce several preliminary mathematical concepts. Subsequently, in chapter 4 the strong formulations and their continuous weak formulations will be presented. Chapter 5 will then describe the domain- and time-discretizations of the problem, and in chapter 6 results will be discussed. Finally, concluding remarks and recommendations for future work are given in chapter 7.

2 | Literature review

This section gives a brief overview of the relevant literature related to this research. Firstly, lipid bilayers are introduced and a few examples of its role in cellular processes in presented. Subsequently, in section 2.2 various constitutional models that have been proposed are summarized. This includes models for interfacial flows, elastic thin solid shells and coupled equations for both. Lastly, in section 2.3 an overview of computational methods applied to lipid bilayers is given.

2.1. The role of lipid bilayers in biology

The cell boundary membrane of non-bacterial cells, such as the human red blood cell, almost always consists of a lipid bilayer [53]. The lipid bilayer separates the interior fluid from an exterior bulk fluid in which the cell lives, and is vital for the surviving of the cell. Lipid bilayers are made out of two layers of lipids that consist of a head and a tail part. The heads are hydrophilic and are oriented towards the water-like fluids on the interior (for the inner layer) and towards the exterior of the cell (for the outer layer). The tails are hydrophobic and are oriented towards the interior of both layers. See the figure on the right for visualization. This cell membrane plays an important role in cellular processes such as endocytosis [4], cytokinesis [7] or cell migration [16]. Endocytosis, which is the process of nutrients traveling toward the cell interior, plays a key role in the transportation and effectivity of medicine at cell level [6]. In addition, endocytosis is used by pathogens to enter the cell.



Interestingly, lipid bilayers are out-of-plane elastic materials, while in-plane, the material behaves like a nearly area-incompressible fluid [54]. Due to this incompressibility constraint, the liquid needs to flow in-plane to accommodate any shape changes of the cell membrane. This indicates the importance of the coupling between out-of-plane deformation and in-plane flow while modeling lipid bilayers. Would the membrane be modeled using out-of-plane elastic effects only, then this does not capture the full properties of the membrane [55].

2.2. Constitutional models for cell membranes

Several models are available for both the in-plane and the out-of-plane behavior of cell membranes. Included are solid hyperelastic thin shells, thin fluid film equations and the coupling of in-plane flow with out-of-plane deformations. This section will provide a brief overview of theoretical models that have been proposed for lipid bilayers

2.2.1. Interfacial flows

The pioneering work of Scriven [22] was crucial in understanding the behavior of interfacial flows. He used differential geometry to represent the flows and other quantities on arbitrarily curved surfaces, instead of using Cartesian-, cylindrical- or spherical-coordinates. The choice of curvilinear coordinates led to a very general set of equations. This elegant setup was extended to three-dimensional flows by Aris [56]. The framework proposed by these two papers is still widely used in today's theoretical models for lipid bilayers [39, 57–59]. An introduction to numerical methods related to interfacial flows is given in Ref. [60].

2.2.2. Elastic thin shells

The first models for thin biological membranes only incorporated the out-of-plane elastic effects, neglecting any liquid behavior. Examples are the bending energy minimisation models of Helfrich and Canham [61, 62], proposed in the early 70s. The extension to a general Euler-Lagrange equation corre-

sponding to the Helfrich-Canham bending model, called the shape equation, was derived more than a decade later [63, 64] and numerical efforts to solve it can be found in Refs. [65, 66]. The models and Euler-Lagrange equations where extended to the non-axisymmetric setting by [67] and continued to be developed [68, 69]. A special case of Helfrich bending energy is the Willmore energy, which describes the total bending energy of a surface shape. A thorough description of the theory and computational techniques with regards to Helfrich bending is described in [70].

Closely connected to the mentioned models of the past decades are thin shell models which describe the behavior of thin and very thin structures. Cell membranes typically are very thin compared to their diameter ($R/r \approx 1000$ for R the diameter length and r the membrane thickness [62, 71]) and can therefore be considered thin shells. Thin shell theories capture the elastic effects of shells with respect to bending and membrane stiffness. One of the simplest and oldest models for thin shells is the Kirchhoff-Love (KL) thin shell theory, which is a product of the plate theory of Kirchhoff [72] and its extension to shells by Love in 1888 [73]. The theory knows an enormous amount of variations and extensions, see Ref. [74] for an overview of the historical impact of the papers of Kirchhoff and Love. In the theory the shell is completely described by a midsurface and the thickness coordinate is given by the normal direction. It is assumed that transverse normal stresses are neglected and that each normal vector remains straight and normal during deformation. The consequence of these central assumptions is the fact that transverse shear strains dissolve, with the advantage that each control point only has three degrees of freedom (DOFs), in stead of five or more. The downside is that the surface kinematics of the KL theory requires C^1 continuity throughout the shell elements and boundaries. This constraint is not easily met by the standard piece-wise polynomial finite elements method. This is probably the reason why, despite the theoretical simplicity of the model, the KL theory is not widely adopted in (older) finite element codes.

Other popular shell models are the Reissner–Mindlin (RM) [75] model and its extensions. The RM model only requires C^0 continuity throughout the elements and their boundaries, which is easily satisfied making use of classical finite element analysis (FEA). It is more complex than the KL model: in addition to the three DOFs associated with the displacement, two other parameters are introduced, describing the rotation of the shell elements. This explains why the RM model is also referred to as a 5-parameter shell model and allows for shear stresses. In case of thicker shells (with $R/r < 20$ for length scale R and shell thickness r), the model is more suitable than the KL model [45]. Despite the additional complexity, the RM model has been more popular due to the more relaxed continuity constraints. Extensions to 7-parameter models are proposed, based on additional strain variables [76] or displacement variables [77] among others. An extensive description of these models can be found in [45]. A 6-parameter solid shell model is recently proposed as an extension of the KL model [78]. In this model both the midsurface and the normal vector are treated as unknown quantities of three DOFs each, releasing the need for C^1 continuous elements.

In this research the KL theory will be adopted to describe the out-of-plane deformation of the system. It is a simple and efficient model and it is well suited for thin structures such as cellular membranes. In section 2.3 it is described how the C^1 continuity requirement will be met.

2.2.3. Coupled equations

A coupling of the elastic effects to the in-plane flow, the diffusion of proteins and/or the surrounding bulk fluid was made for simple geometries [79–81]. In 2009, the general equations for a single component, arbitrarily curved and deforming lipid membranes coupled to in-plane viscous flow were determined for the first time [39]. The authors of this last work use variational methods and exterior calculus. Their elastic out-of-plane component is based on the previously mentioned Helfrich bending energy [61]. A mistake in Ref. [39] concerning the presence of inertia terms, is corrected in the work of Yavari [58] and the momentum conservation is formulated more sharply in [82]. Variations on the model have been made by introduction of an area difference elastic model [83], an extra coupling to in-plane protein diffusion [55] and by using balance laws to derive the equations [59, 82]. A more general approach is taken in [84–86], were a theoretical framework is established that leaves room for different solid and liquid constitutional models for biological membranes. A pure membrane version of these papers is described in [44]. Other constitutive models in context of the KL theory, applicable to liquid shells and membranes, is presented in Ref. [87]

2.3. Computational methods for cell membranes

In this section a brief overview is given of computational methods for lipid bilayers. Initially, isogeometric analysis (IGA) is introduced and a brief overview of papers that use IGA as a computational method is provided. Subsequently, papers related to discrete exterior calculus are mentioned. Lastly, a general overview of computational methods for lipid bilayers is provided.

2.3.1. Isogeometric analysis

The concept of isogeometric analysis (IGA) was first introduced by Hughes et al. in 2005 [46]. Its original idea came from a desire to integrate computer aided design CAD and the finite element method (FEM) more tightly with each other. In stead of re-meshing an object when switching from the design process to analysis, it would be possible to work with the exact same geometry, saving lot of time. In CAD software Non-uniform rational B-splines (NURBS) have become a standard method to build geometries. By replacing the polynomial shape functions of classical FEM with NURBS, both the geometry and the analysis use the same basis functions while the geometry is represented in an exact manner. Especially for complex shaped geometries, this would lead to significantly improved convergence properties [46, 47]. For the mathematical description of the IGA concept and the usage of B-splines or NURBS, we refer to these last to papers. Additionally, IGA is superior to standard piecewise-polynomial based FEA in terms of robustness and efficiency [48–50]

In this research the main reason for adopting the IGA method is directly related to the C^1 continuity requirements that follow from the KL thin shell theory. This requirement is easily attained using IGA, while with standard FEA it is not so straight forward. As a consequence, many (older) FEM implementations only approximately satisfy the C^1 smoothness [88, 89], use discontinuous Galerkin [90] or triangular loop subdivision FEA [91, 92]. Often these kind of methods are less efficient as additional degrees of freedom need to be introduced. One of the first applications of IGA to KL models is done by Kiendl et al. [42, 43] and further extended to elasto-plastic systems in Ref. [93]. These pioneering papers are used in theoretical models of, for example, structure analysis [94], solid phase thin shells [85] and thin floating object [95]. IGA and KL based computational methods are also applied to lipid bilayers [86, 96, 97]. Furthermore, IGA is even used for the RM shell model [98–100] due to its more appealing accuracy and efficiency properties.

2.3.2. Discrete exterior calculus

Conservation laws of continuous differential equation are not necessarily attained in their discrete counterparts used in computational methods. If this is not the case, for example if mass conservation of a incompressible fluid is not attained at the discrete setting, the method is called *inconsistent*. Inconsistent computational methods run the risk of systematically converging to wrong solutions. Especially in complex systems this is dangerous, as it is hard to notice the error with the human eye. A coupled system of surface Stokes flow on a time-evolving domain is an example of such a complex system and one would like to prevent any inconsistencies. A way of doing this is by means of exterior FEA [37, 38], relying on the framework of discrete exterior calculus (DEC) [35, 36].

2.3.3. Review of computational methods

A few of the very first examples of computational methods for elastic membranes include a consistent FE formulation [101] and a model of the behavior of (sometimes incompressible) rubber membranes [102, 103]. Other, more recent publications in the category of elastic membranes, examine wrinkling behavior in thin shells [104], find an application in woven fibre membranes [105] or use C^1-conforming subdivision surfaces to attain the C^1 smoothness requirement from the KL theory [91, 92]. Phase changes and phase separation of lipid membranes have been numerically modeled [97, 106, 107]. Among others, this can be applied to modeling the fusion of HIV with cell membranes. Concerning curvature elasticity, Willmore flow is the geometric flow of a surface shape to minimal Willmore energy (for genus 0 surfaces this is a sphere). The Willmore energy is a special case of the Helfrich bending energy [108]. The Willmore energy functional is used to numerically model Willmore flow [106, 109–112] (were Ref. [112] uses IGA) and to model the behaviour of fluid membranes [12, 85, 113]. The Willmore and Helfrich energy have also been used for computational methods that model budding behaviour of liquid cell membranes [86, 96, 107, 114] and solid cell membranes [115]. Apart from these out-of-plane deformation models, in-plane fluid behaviour of membranes is examined using the surface (Navier-)Stokes equation. FE

implementations of the surface (Navier-)Stokes equation for static surfaces include the usage of the discrete exterior calculus framework [40, 116] and standard FEA [30, 117]. However, Refs. [30, 116, 117] use Lagrange multiplayer's or a penalty method to enforce tangential surface flow. Additionally, there are some examples of papers that model the surface (Navier-)Stokes equation, without using FEA [118, 119].

The coupled out-of-plane deformation and in-plane flow of liquid membranes are numerically modeled using the thin film equations (without bending) [52, 120] or by incorporating bending via the Helfrich or Willmore energies [13, 55, 59, 121]. An extension of the latter setting also involves a coupling to the surrounding bulk (Navier-)Stokes equation [14, 39, 83, 122]. Ref. [52] is the only mentioned paper of this paragraph, that employs IGA for the coupled equations. However, in the paper an additional projection step is needed in order to attain the inf-sub conditions [123] and full genus 0 surfaces are not addressed. The coupling of in- and out-of-plane effect is also modeled for an instantaneous out-of-plane velocity, using a special case of IGA that relies on polar splines [41]. It is emphasized that this paper does not include time-evolving surfaces.

3 | Mathematical Preliminaries

This section introduces several concepts that are used extensively in this research. Firstly, B-spline basis functions are introduced in section 3.1. In section 3.4.3 B-splines are then extended to *polar splines*. Subsequently, sections 3.2 and 3.3 introduce the surface kinematics and differential calculus framework, respectively. The latter is used in defining divergence conforming discrete spline spaces.

3.1. B-splines

Especially in the area of computer-aided design, B-spline curves, surfaces and solids are used to represent the shapes of objects in development. In this research B-splines are used both to represent the arbitrarily curved domain of cellular membranes and to perform analysis. Exactly the same B-spline function spaces will be used for both. This is called the *isogeometric* concept: the basis for the geometry is also used to approximate the solution.

Firstly, the univariate B-spline basis functions will be introduced (see section 3.1.1 - 3.1.3), which are used to define arbitrary curves and surfaces. Also, two different refinement types, h- and p-refinement, are described. Note that B-spline solids will be excluded from this introduction as these will not be needed throughout the research. This section is mainly based on [46, 47] and the references therein.

3.1.1. Knot vectors & basis functions

The B-spline basis functions (also called *shape functions*) live on a parametric space, where their corresponding knot vector is defined. A knot vector Ξ is a sequence of increasing values, which indicate the boundaries of elements in parametric space. A general knot vector is given by

$$\Xi = [\xi_1,\ \xi_2,\ \xi_3,\ \ldots,\ \xi_{n_{\text{sh}}+r},\ \xi_{n_{\text{sh}}+r+1}] \tag{3.1}$$

with $\xi_1 \leq \xi_2 \leq \xi_3 \leq \ldots \leq \xi_{n_{\text{sh}}+r} \leq \xi_{n_{\text{sh}}+r+1}$. Here n_{sh} indicates the number of basis functions (or *shape functions*), while r indicates their order. A knot vector for basis functions of order r is said to be *open* if the first and last knots are repeated $r+1$ times, that is, the first and last $r+1$ knots are equal.

Definition. *B-spline basis functions are defined by the Cox-deBoor recursive formula's [47, 124],*

$$\text{for } r = 0: \quad B_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.2a}$$

$$\text{for } r > 0: \quad B_i^r(\xi) = \frac{\xi - \xi_i}{\xi_{i+r} - \xi_i} B_i^{r-1}(\xi) + \frac{\xi_{i+r+1} - \xi}{\xi_{i+r+1} - \xi_{i+1}} B_{i+1}^{r-1}(\xi). \tag{3.2b}$$

These formulas use the definition $\frac{0}{0} := 0$. The formulas are easily understood, but are not so suited for practical uses as far more efficient algorithms exist to generate the basis functions [51, 125–127].

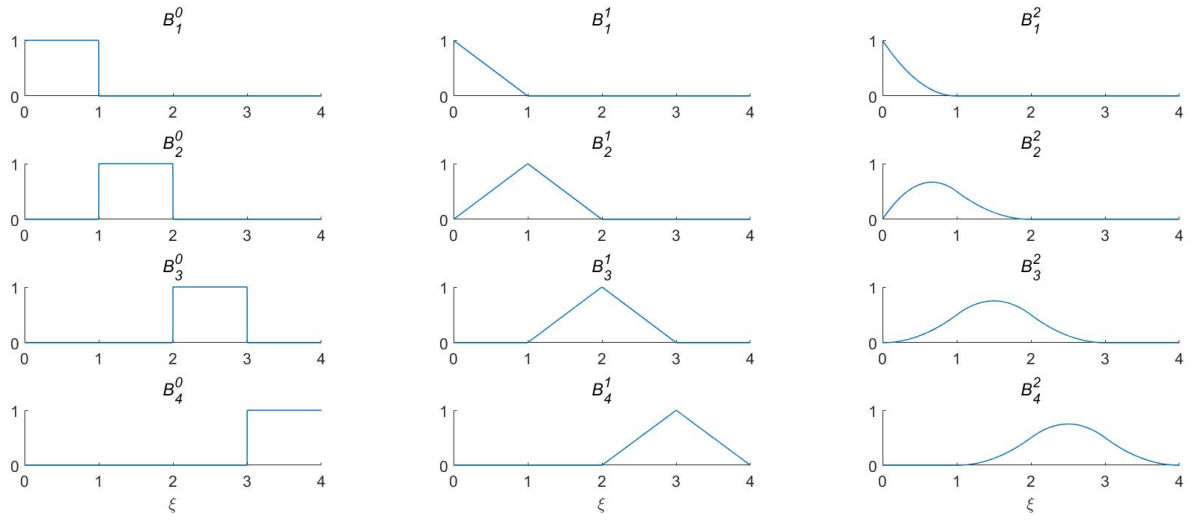
B-spline basis functions have the following appealing properties. First of all, assuming maximal smoothness, basis functions sum up to unity,

$$\sum_{i=1}^{n} B_i^r(\xi) = 1 \quad \forall \xi. \tag{3.3}$$

The basis functions have local support, that is, basis function $B_i^r(\xi)$ is non-zero on the interval $[\xi_i,\ \xi_{i+r+1}]$ only. Another very important property is that each B-spline basis functions of order r has $r-1$ continuous derivatives and can be differentiated r times on the whole domain, including the element boundaries. This property is one of the main reason B-splines are adopted in this research. Furthermore, the basis functions are linearly independent.

To make all this more concrete, observe the basis functions of order $r = 0, 1, 2$ that are presented in figure 3.1. Their corresponding open knot vectors are given by

$$\Xi = [\underbrace{0, \ldots, 0}_{r+1 \text{ times}}, 1, 2, 3, 4, 5, \ldots] \tag{3.4}$$



(a) Order $r = 0$ **(b)** Order $r = 1$ **(c)** Order $r = 2$

Figure 3.1: Order 1, 2, 3 B-spline basis functions corresponding to knot vector of Eq. (3.4)

As can be seen, the basis functions of order 0 and 1 are identical to the standard FE basis functions of the same order. However, from order 2 onward this does not hold.

3.1.2. B-spline curves

To obtain a B-spline curve $\mathcal{C}$, each basis function is multiplied with a *control point* (essentially a vector). The summation over all basis functions and control point pairs defines the curve,

$$\mathcal{C}(\xi) = \sum_{i=1}^{n_{\text{sh}}} B_i^r(\xi) \boldsymbol{R}_i, \tag{3.5}$$

with control points $\boldsymbol{R}_i$ (vectors). An example of such a curve can be found in figure 3.2, with knot vector $\Xi = [0, 0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6]$. In this figure an important property of the control points is visualised: in general, the control points do not interpolate the curve. In the figure there are three exceptions. The two outer control points do interpolate the curve due to the repeated boundary knots of the open knot vector. Also, control point $\xi = 4$ interpolates the curve due to double occurrence. This is the only location where the curve is C^0 smooth in stead of C^1 smooth. In general, a B-spline curve is C^{r-m_i} smooth for basis functions of order r and multiplicity of m_i of knot ξ_i.

3.1.3. Refinement types

There exist different refinement possibilities for B-spline based functions spaces. Some of these have an analogous operation in standard, piecewise polynomial based FEA.

Knot insertion (h-refinement)

Let $\Xi = [\xi_1, \xi_2, \ldots, \xi_{n_{\text{sh}}+r+1}]$ be a knot vector and let $\bar{\xi} \in [\xi_k, \xi_{k+1})$. Define a new knot vector $\Xi = \left[\xi_1, \xi_2, \ldots, \xi_k, \bar{\xi}, \xi_{k+1}, \ldots, \xi_{n_{\text{sh}}+r+1}\right]$. The new basis functions corresponding to this new knot vector are defined by Eq. (3.2). This refinement type enriches the solution space and produces an improved approximation of the exact solution. The support of each individual function is decreased, leading to a relatively more sparse mass matrix. In order to keep the geometry identical, the original control points $\{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_{n_{\text{sh}}}\}$ are substituted by new control points $\{\bar{\boldsymbol{R}}_1, \bar{\boldsymbol{R}}_2, \ldots, \bar{\boldsymbol{R}}_{n_{\text{sh}}+1}\}$, where n_{sh} is the original number of basis functions. These new control points are defined by

$$\overline{\boldsymbol{R}}_i = \alpha_i \boldsymbol{R}_i + (1 - \alpha_i)\, \boldsymbol{R}_{i-1} \quad \text{with} \tag{3.6}$$

$$\alpha_i = \begin{cases} 1, & 1 \leqslant i \leqslant k - r \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+r} - \xi_i}, & k - r + 1 \leqslant i \leqslant k, \\ 0, & k + 1 \leqslant i \leqslant n + r + 2 \end{cases} \tag{3.7}$$

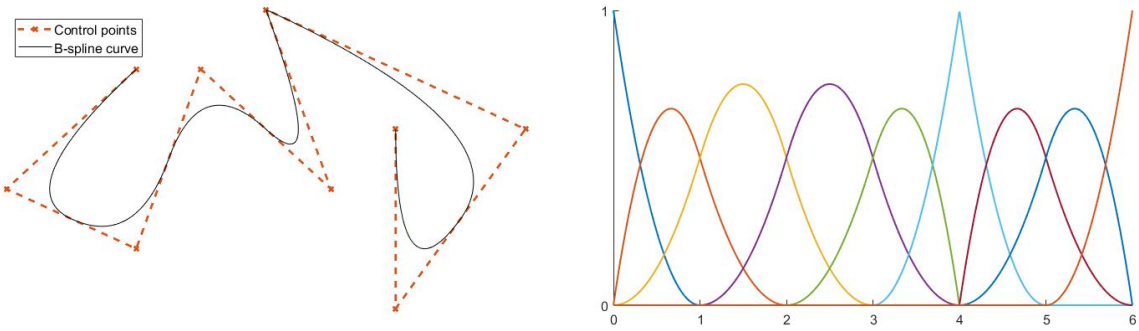


Figure 3.2: Quadratic B-spline curve and corresponding basis functions. Knot vector:
$\Xi = [0, 0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6]$

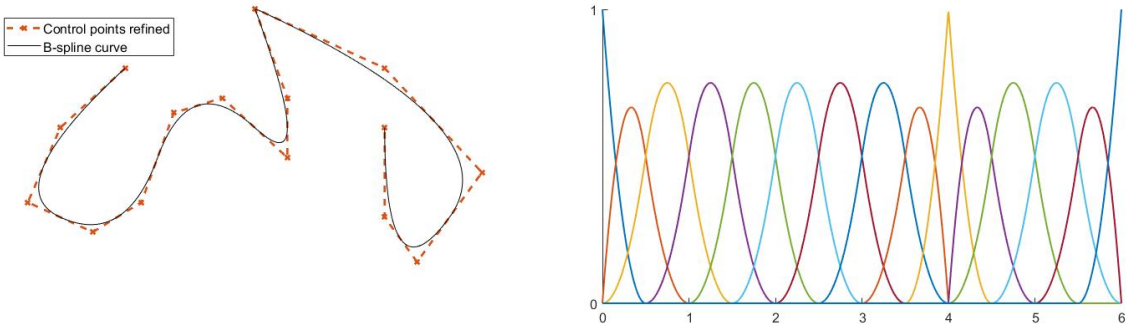


Figure 3.3: h-refined B-spline curve and corresponding basis functions. Knot vector:
$\Xi = \left[0,\ 0,\ 0,\ \frac{1}{2},\ 1,\ 1\frac{1}{2},\ 2,\ 2\frac{1}{2},\ 3,\ 3\frac{1}{2},\ 4,\ 4,\ 4\frac{1}{2},\ 5,\ 5\frac{1}{2},\ 6,\ 6,\ 6\right]$

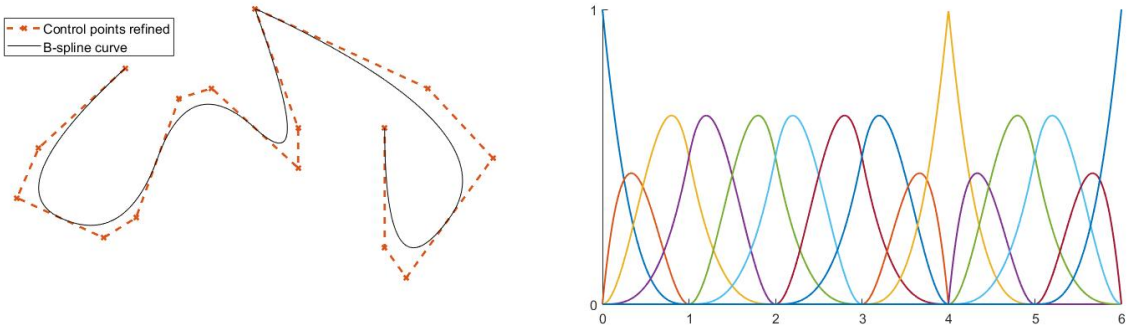


Figure 3.4: p-refined B-spline curve and corresponding basis functions. Knot vector:
$\Xi = [0, 0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 4, 5, 5, 6, 6, 6, 6]$

It is possible to insert a knot that already exists. This is the same as repeating a knot, and it lowers the order of continuity of the curve at this knot. For example, see knot $\xi = 4$ of figure 3.2. The process of knot insertion of very analogous to h-refinement of standard FEA, where the mesh is refined. An example of knot refinement is performed on the curve of figure 3.2 resulting in figure 3.3. The corresponding knot vectors are

$$\begin{aligned} &\text{Original :} \quad \Xi = [0,0,0, \quad 1, \quad 2, \quad 3, \quad 4,4, \quad 5, \quad 6,6,6] \\ &h\text{-refined :} \quad \Xi = [0,0,0, \ \tfrac{1}{2}, \ 1, \ 1\tfrac{1}{2}, \ 2, \ 2\tfrac{1}{2}, \ 3, \ 3\tfrac{1}{2}, \ 4, \ 4, \ 4\tfrac{1}{2}, \ 5, \ 5\tfrac{1}{2}, \ 6,6,6]. \end{aligned} \tag{3.8}$$

Note that the curves remain identical to each other.

Order elevation (p-refinement)
Elevating the order of the basis functions is another way to obtain an enriched solution space. In most literature, the character p (and q) is reserved to indicate the order. However, in this document the order is indicated by r (and s for the second coordinate), as p and q will refer to test or trial functions of the weak formulation. Elevating the original order r_0 to order $r_1 > 0$ leads to $C^{r_1 - m_i}$ smooth functions, where m_i indicates the multiplicity of knot ξ_i. This is an important difference with p-refinement of standard FEA. The smoothness on the element boundaries of standard FEA will always remain C^0, independent of the order of the piecewise polynomial basis functions.

To keep the B-spline geometry identical after order elevation, each knot of the corresponding knot vector needs to be repeated $r_1 - r_0$ times. For example, if the order of the curve of figure 3.2 is to be elevated to $r_1 = r_0 + 1$ (one order higher), it will result in figure 3.4. The corresponding knot vectors are

$$\begin{aligned} &\text{Original :} \quad \Xi = [0,0,0, \quad 1, \quad 2, \quad 3, \quad 4, \ 4, \quad 5, \quad 6,6,6] \\ &h\text{-refined :} \quad \Xi = [0,0,0,0 \ , \ 1, \ 1, \ 2, \ 2, \ 3, \ 3, \ 4, \ 4, \ 4 \ 5, \ 5, \ 6,6,6,6]. \end{aligned} \tag{3.9}$$

3.1.4. B-spline surfaces and geometric mapping

Let $\hat{\Omega} = [0,1] \times [0,1]$ be a 2D square domain with coordinates $(\boldsymbol{\xi}) = (\xi^1, \xi^2)$. The above concept of B-spline curves can be extended to surfaces. The multivariate basis functions are defined by means of the tensor product of univariate basis functions:

$$B_{ij}^{(r,s)}(\boldsymbol{\xi}) = B_i^r(\xi^1) B_j^s(\xi^2) \tag{3.10}$$

Here $B_j^s(\xi^2)$ are the order s basis functions corresponding to knot vector $\mathcal{H} = \{\eta_1, \eta_2, \ldots, \eta_{m_{\text{sh}}+s+1}\}$ and m_{sh} indicates the number of basis functions in the η direction. A B-spline surface is now defined by

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{i=1}^{n_{\text{sh}}} \sum_{j=1}^{m_{\text{sh}}} B_{ij}^{(r,s)}(\boldsymbol{\xi}) \boldsymbol{R}_{ij}. \tag{3.11}$$

The concept of B-spline surfaces is applied to define the geometric domain Ω, using a B-spline geometric mapping $\boldsymbol{x}(\boldsymbol{\xi}) : \hat{\Omega} \longrightarrow \Omega$ from the parametric to the geometric domain. The mapping $\boldsymbol{x}$ is defined as in Eq. (3.11). The canonical basis of Ω reads as $(\boldsymbol{x}) = (x^1, x^2)$, but the domain is parametrized by the parametric coordinates, called curvilinear coordinates. The inverse $\boldsymbol{x}^{-1}(\boldsymbol{x}) = \boldsymbol{\xi}(\boldsymbol{x})$ is assumed to exist. For 2D vectors $\boldsymbol{R}_{ij}$, an example of this mapping is found in figure 3.5. Note that the gradient of this mapping $\nabla \boldsymbol{x}$, the Jacobian matrix J, is given by

$$\begin{aligned} \nabla \boldsymbol{x} = J &= \begin{bmatrix} \boldsymbol{x}_{,1}(\boldsymbol{\xi}) & \boldsymbol{x}_{,2}(\boldsymbol{\xi}) \end{bmatrix} = \begin{bmatrix} x^1_{,1}(\boldsymbol{\xi}) & x^1_{,2}(\boldsymbol{\xi}) \\ x^2_{,1}(\boldsymbol{\xi}) & x^2_{,2}(\boldsymbol{\xi}) \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^{n_{\text{sh}}} \sum_{j=1}^{m_{\text{sh}}} B_{ij,1}^{(r,s)}(\boldsymbol{\xi}) \boldsymbol{R}_{ij} & \sum_{i=1}^{n_{\text{sh}}} \sum_{j=1}^{m_{\text{sh}}} B_{ij,2}^{(r,s)}(\boldsymbol{\xi}) \boldsymbol{R}_{ij} \end{bmatrix}, \end{aligned} \tag{3.12}$$

where each element of this matrix can be expressed in terms of the derivatives of the basis functions. Extension to a surface embedded into 3D space is done by simply extending the 2D control points $\boldsymbol{R}_{i,j}$ to 3D. Consequently, the corresponding Jacobian matrix of Eq. (3.11) is not square anymore, as $\boldsymbol{x} = \begin{bmatrix} x^1 & x^2 & x^3 \end{bmatrix}^T$.

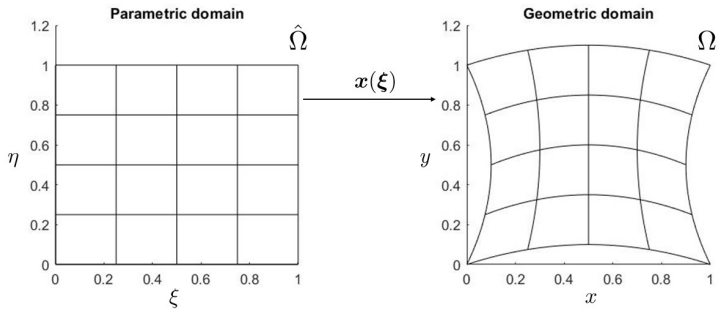


Figure 3.5: Mapping $\boldsymbol{x}(\boldsymbol{\xi})$ defining a geometric domain Ω.

3.1.5. B-spline functions

B-spline functions are build using the same basis functions as for the geometry. A B-spline function, living on $\hat{\Omega}$, is given by

$$\hat{u}(\boldsymbol{\xi}) = \sum_{i=1}^{n_{\text{sh}}} \sum_{j=1}^{m_{\text{sh}}} B_{ij}^{(r,s)}(\boldsymbol{\xi}) \hat{u}_{ij}. \tag{3.13}$$

$\hat{u}_{ij}$ denote the *control variables* of the function and the hat indicates functions and control variables on the parametric domain $\hat{\Omega}$. The same function $u(\boldsymbol{x})$, but of different form, defined on the geometric domain Ω can be constructed using a composition of $\hat{u}(\boldsymbol{\xi})$ and the inverse of the mapping. This will be further discussed in section A.3.3.

3.2. Surface kinematics

Here, surface quantities that are needed to express PDEs on surfaces embedded into 3D space are presented. An important ingredient is the curvilinear coordinates framework with parameters $\boldsymbol{\xi}$ that describe the surface position $\boldsymbol{x}(\boldsymbol{\xi})$ [45, 128]. Among others, this section introduces surface quantities that are later used in the fluid film equations. Introductory section of [42, 43, 52, 84, 87, 129, 130] are followed in the description.

Take a point $\boldsymbol{x}$ on the surface Ω. There exists a natural covariant basis for the tangential plane at $\boldsymbol{x}$, given by

$$\boldsymbol{a}_\alpha := \boldsymbol{x}_{,\alpha} = \frac{\partial \boldsymbol{x}}{\partial \xi^\alpha}, \tag{3.14}$$

and unit-length normal vector $\boldsymbol{n} = (\boldsymbol{a}_1 \times \boldsymbol{a}_2)/|\boldsymbol{a}_1 \times \boldsymbol{a}_2|$. Here and from now on Greek sub- and superscripts span the set $\{1, 2\}$. The two parametric derivatives are denoted by $(\cdot)_{,\alpha}$ for readability.

Using the natural basis $\boldsymbol{a}_\alpha$, the metric tensor, $g_{\alpha\beta} = \boldsymbol{a}_\alpha \cdot \boldsymbol{a}_\beta$, is introduced. It is also known as the *first fundamental form*). It is concisely written as

$$\boldsymbol{g} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}. \tag{3.15}$$

It is easy to see (using Eq. (3.12)) that $\boldsymbol{g} = J^T J$ holds. A dual *contravariant* basis $\boldsymbol{a}^\alpha$ for the tangent plane at $\boldsymbol{x}$ is defined by $\boldsymbol{a}_\alpha \cdot \boldsymbol{a}^\beta = \delta_\alpha^\beta$, where δ_α^β is the Kronecker delta. The corresponding contravariant metric tensor reads

$$g^{\alpha\beta} := \boldsymbol{a}^\alpha \cdot \boldsymbol{a}^\beta, \quad \text{with } \boldsymbol{g}^{-1} = \begin{bmatrix} g^{11} & g^{12} \\ g^{21} & g^{22} \end{bmatrix}, \tag{3.16}$$

and it is the matrix inverse of the covariant metric tensor, $[g^{\alpha\beta}] = [g_{\alpha\beta}]^{-1}$. It follows that the sets $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{n}\}$ and $\{\boldsymbol{a}^1, \boldsymbol{a}^2, \boldsymbol{n}\}$ form a basis for $\mathbb{R}^3$. Note that both pairs $\boldsymbol{a}_\alpha$ and $\boldsymbol{a}^\alpha$ are not necessarily an orthogonal or unit-length pair of vectors, but $\boldsymbol{n}$ is always orthogonal to both $\boldsymbol{a}_\alpha$ and $\boldsymbol{a}^\alpha$.

Any surface vector $\boldsymbol{v}$ can now be expressed in terms of both in-plane bases,

$$\begin{aligned} \boldsymbol{v} &= v^\alpha \boldsymbol{a}_\alpha \\ \boldsymbol{w} = \boldsymbol{v}^\flat &= v_\alpha \boldsymbol{a}^\alpha. \end{aligned} \tag{3.17}$$

Here, and throughout the rest of the report, Einsteins summation convention is adopted: for every Greek sub or super script that occurs twice in the same term, a summation over the span of this index is suppressed for readability. For example, $f_\alpha g_\alpha$ reads as

$$f_\alpha g_\alpha = \sum_{\alpha \in \{1,2\}} f_\alpha g_\alpha. \tag{3.18}$$

In Eq. (3.17) v^α and v_α are contravariant and covariant components of $\boldsymbol{v}$ and $\boldsymbol{v}^\flat$, respectfully. The 'flat' symbol $(\cdot)^\flat$ indicates that the indices of in-plane vector components are lowerd. Raising the component indices is denoted by the 'sharp' symbol $(\cdot)^\sharp$ and so $\boldsymbol{w}^\sharp = \boldsymbol{v}$. Lowering and raising indices involve the components of the metric tensor $\boldsymbol{g}$: $v_\alpha = g_{\alpha\beta} v^\beta$ and $v^\alpha = g^{\alpha\beta} v_\beta$. Note that physically, there is no difference between $\boldsymbol{v}$ and $\boldsymbol{v}^\flat$. Both vectors represent the same 'arrow', but are expressed in terms of a different basis. A general 3D vector $\boldsymbol{V}$, can be expressed as

$$\boldsymbol{V} = \boldsymbol{v} + v^{\mathrm{m}} \boldsymbol{n} = \boldsymbol{v}^\flat + v^{\mathrm{m}} \boldsymbol{n}. \tag{3.19}$$

v^{m} is the normal component of $\boldsymbol{V}$. This component is not dependent on the used in-plane basis, due to the orthogonality of $\boldsymbol{n}$.

As the coordinate systems $\{\boldsymbol{a}_\alpha, \boldsymbol{n}\}$ and $\{\boldsymbol{a}^\alpha, \boldsymbol{n}\}$ are dependent on the position $\boldsymbol{x}$ on the manifold, the derivatives of $\boldsymbol{a}_\alpha$ do not have the desired properties that fixed orthonormal coordinate systems do have. A new notion of differentiation is therefore introduced, called the covariant derivative,

$$\begin{aligned} v^\alpha_{;\beta} &= v^\alpha_{,\beta} + \Gamma^\alpha_{\gamma\beta} v^\gamma \\ v_{\alpha;\beta} &= v_{\alpha,\beta} - \Gamma^\gamma_{\alpha\beta} v_\gamma, \end{aligned} \tag{3.20}$$

where the Christoffel symbol $\Gamma^\gamma_{\alpha\beta}$ is given by

$$\Gamma^\gamma_{\alpha\beta} = \boldsymbol{a}^\gamma \cdot \boldsymbol{a}_{\alpha,\beta}. \tag{3.21}$$

Note that $\Gamma^\gamma_{\alpha\beta}$ is symmetric in its lower indices.

Another surface quantity is the *curvature tensor* (or *second fundamental form*) $\boldsymbol{k}$ in covariant components,

$$[k_{\alpha\beta}] = \boldsymbol{n} \cdot \boldsymbol{a}_{\alpha,\beta} = \boldsymbol{n}_{,\beta} \cdot \boldsymbol{a}_\alpha. \tag{3.22}$$

It can also be expressed in mixed component and contravariant component form, $k^\alpha_\beta = k_{\gamma\beta} g^{\gamma\alpha}$ and $k^{\alpha\beta} = k^\beta_\gamma g^{\gamma\alpha}$, respectively. Each tensor is symmetric. The *mean curvature* of the surface is given by

$$H := \frac{1}{2} k^\alpha_\alpha = \frac{1}{2} g^{\alpha\beta} k_{\alpha\beta} \tag{3.23}$$

while the *Gaussian curvature* reads as

$$K = \frac{k}{g}. \tag{3.24}$$

g and k are the determinants of $\boldsymbol{g}$ and $\boldsymbol{k}$, respectively.

3.3. Introduction to differential calculus

Differential forms are objects that represent integrands over curves, surfaces or solids. It is a framework that generalises the fundamental theorem of calculus, the Stokes' circulation theorem and Gauss' divergence theorem [34]. Differential forms have a close relationship with exterior calculus, which is the study of n-dimensional ($n \in N$) differentiable manifolds. In this research only surfaces embedded into 3D Euclidean space are observed and, therefore, this introduction is restricted to $n = 2$. For a more complete and more general introduction of exterior calculus and differential forms see Refs. [37, 38]. In this introduction mostly Refs. [34, 41] are followed.

For a 2D manifold Ω, there are three different types of differential forms, called k-forms for $k = 0, 1, 2$. They correspond to spaces $\Lambda^k(\Omega)$ for $k = 0, 1, 2$, respectively. Elements of $\Lambda^k(\Omega)$ are skew-symmetric

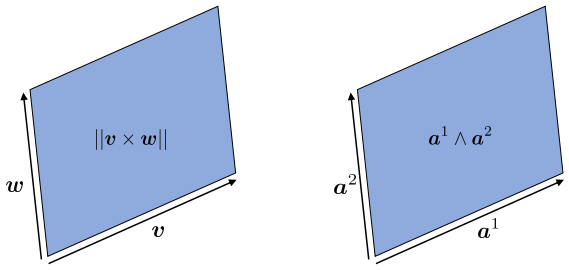


Figure 3.6: The length of the cross product $\boldsymbol{v} \times \boldsymbol{w}$ is equal to the surface of the parallelogram defined by the two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$. In much the same sense, the wedge product of basis vectors $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$ defines a surface 2-form $\boldsymbol{a}^1 \wedge \boldsymbol{a}^2$.

k-linear forms on $T_{\boldsymbol{x}}\Omega \times \ldots \times T_{\boldsymbol{x}}\Omega$, where $T_{\boldsymbol{x}}\Omega$ is the tangent 2D space at $\boldsymbol{x} \in \Omega$. Let $\omega^{(0)} \in \Lambda^0(\Omega)$ be a 0-form. $\omega^{(0)}(\boldsymbol{x})$ actually is a very well known object, namely a scalar function. A 1-form $\Lambda^1(\Omega) \ni \omega^{(1)} = \omega_\alpha \boldsymbol{a}^\alpha$ is identical to a covector as introduced in Eq. (3.17), with ω_α the covariant components. The contravariant basis $\boldsymbol{a}^\alpha$ ensures that $\omega^{(1)}$ lies in the tangent plane of Ω at $\boldsymbol{x} \in \Omega$. Lastly, let $\omega^{(2)} \in \Lambda^2(\Omega)$ be a 2-form. This form can be written as $\omega^{(2)} = \omega_{12}\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2$, where $\boldsymbol{a}^1 \wedge \boldsymbol{a}^2$ is called the *volume form*. Since only surfaces ($n = 2$) are used, this form will be referred to as the *surface form*. From now on $\omega^{(k)}$ will denote a k-form in $\Lambda^k(\Omega)$.

The operator $\wedge$ is borrowed from the framework of exterior calculus, and its is called the *exterior product* or *wedge product*. It is given by the mapping $\wedge\colon \Lambda^j \times \Lambda^k \longrightarrow \Lambda^{j+1}$. The best way to intuitively understand its working is by interpreting the exterior product as a cross product between vectors in the 3D space. The length of the resulting vector is the surface area of the parallelogram spanned by the two input vectors, hence the name surface form. Its direction is a vector normal to the induced surface. (See figure 3.6) The exterior product has properties similar to the cross product:

Property. *The exterior product has the following properties:*

1. *It is skew symmetric, that is,* $\boldsymbol{a}^1 \wedge \boldsymbol{a}^2 = -\boldsymbol{a}^2 \wedge \boldsymbol{a}^1$.
2. *The exterior self product* $\boldsymbol{a}^i \wedge \boldsymbol{a}^i = 0$ *is zero for* $i = 1, 2$.
3. *It has the property of distribution over addition and multiplication.*
4. *It is associative, such that* $(\omega^{(i)} \wedge \beta^{(j)}) \wedge \gamma^{(k)} = \omega^{(i)} \wedge (\beta^{(j)} \wedge \gamma^{(k)})$. *(Contrary to the cross product.)*

As an example, take two 1-forms $\omega^{(1)} = \omega_\alpha \boldsymbol{a}^\alpha$ and $\beta^{(1)} = \beta_\alpha \boldsymbol{a}^\alpha$. Using the mentioned properties, their exterior product is given by

$$\left(\omega_1 \boldsymbol{a}^1 + \omega_2 \boldsymbol{a}^2\right) \wedge \left(\beta_1 \boldsymbol{a}^1 + \beta_2 \boldsymbol{a}^2\right) = (\omega_1 \beta_2 - \omega_2 \beta_1)\, \boldsymbol{a}^1 \wedge \boldsymbol{a}^2. \tag{3.25}$$

There are a few other important operations on differential forms that will be introduced. Firstly, the *Hodge star* $\star\colon \Lambda^k \longrightarrow \Lambda^{n-k}$ is defined in terms of the metric and its inverse [41]. For 0-, 1- and 2-forms it is given by, respectively,

$$\begin{aligned} \star\omega &= \sqrt{g}\omega_{12}\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 \\ \star\omega_\alpha\ \boldsymbol{a}^\alpha &= \sqrt{g}\omega_\alpha g^{\alpha\gamma}\epsilon_{\gamma\delta}\ \boldsymbol{a}^\delta \\ \star\omega_{12}\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 &= \frac{1}{\sqrt{g}}\omega. \end{aligned} \tag{3.26}$$

Here, $\epsilon_1 2 = 1$, $\epsilon_{21} = -1$, and 0 otherwise.

Essentially the Hodge star is the bridge between the (sub)space $\Lambda^k(\Omega)$ and its orthonormal (sub)space $\left(\Lambda^k\right)^\perp(\Omega) = \Lambda^{n-k}(\Omega)$ for $k = 0, 1, 2$. It has the following properties

$$\star\star\,\omega^{(k)} = (-1)^k \omega^{(k)}, \tag{3.27a}$$

$$\beta^{(k)} \wedge \star\omega^{(n-k)} = \langle \beta^{(k)}, \omega^{(n-k)} \rangle_{\boldsymbol{g}}\ dS, \tag{3.27b}$$

for surface form $dS = \boldsymbol{a}^1 \wedge \boldsymbol{a}^2$. See figure 3.7 for an example of the Hodge star operator applied to a 1-form $\omega^{(1)}$. The figure provides an intuitive understanding of property (3.27a).

Another important operator is the *exterior derivative* $\mathbf{d}$: $\Lambda^k(\Omega) \longrightarrow \Lambda^{k+1}(\Omega)$. The exterior derivative is a coordinate independent generalization of vector calculus gradient, divergence and curl operators [34]. Applying $\mathbf{d}$ to a k-form produces a unique $k+1$-form given by

$$\mathbf{d}\,\omega^{(0)} = \frac{\partial \omega^{(0)}}{\partial \xi^\alpha}\ \boldsymbol{a}^\alpha \tag{3.28}$$

$$\mathbf{d}\,\omega^{(1)} = \left(\frac{\partial \omega_2}{\partial \xi^1} - \frac{\partial \omega_1}{\partial \xi^2}\right)\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2. \tag{3.29}$$

Furthermore, $\mathbf{d}\,\mathbf{d}\,\omega^{(k)} = 0$ for any k-form and for $\omega^{(i)} \in \Lambda^i(\Omega)$ and $\beta^{(j)} \in \Lambda^j(\Omega)$ the Leibniz rule with respect to the wedge product is satisfied:

$$\mathbf{d}(\omega^{(i)} \wedge \beta^{(j)}) = \mathbf{d}\,\omega^{(i)} \wedge \beta^{(j)} + (-1)^i \omega^{(i)} \wedge \mathbf{d}\,\beta^{(j)} \tag{3.30}$$

The co-differential operator $\mathbf{d}^\star : \Lambda^k(\Omega) \longrightarrow \Lambda^{k-1}(\Omega)$ is given by $\mathbf{d}^\star = -\star\,\mathbf{d}\,\star$ such that

$$\left(\omega^{(k-1)}, \mathbf{d}^\star \beta^{(k)}\right) = \left(\mathbf{d}\,\omega^{(k-1)}, \beta^{(k)}\right) - \int_{\partial\Omega} \omega^{(k-1)} \wedge \star\,\beta^{(k)}. \tag{3.31}$$

It follows that $\mathbf{d}^\star$ is the Hilbert space adjoint of the exterior derivative, up to a boundary term.

Differential forms can be interpreted as integrand objects inducing a natural integral. Let $\omega^{(k)}$ be a k-form. Its natural integral is given by

$$\int_{\Omega^k} \omega^{(k)} \tag{3.32}$$

for Ω^k a k-dimensional (sub)manifold. For $k = 0$ this boils down to the function evaluation

$$\int_{\{\boldsymbol{x}_0\}} \omega^{(0)} = \omega^{(0)}(\boldsymbol{x}_0). \tag{3.33}$$

For $k = 1, 2$ the integral can be written as, respectively, a Riemann line- or surface integral

$$\int_{\mathcal{C}} \omega^{(1)} = \int_{\mathcal{C}} f_\alpha\ a^\alpha = o(\mathcal{C}) \int_{\mathcal{C}} \omega_1\ dx^1 + \omega_2\ dx^2 \tag{3.34a}$$

$$\int_{\mathcal{S}} \omega^{(2)} = \int_{\mathcal{S}} \omega\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 = o(\mathcal{S}) \iint_{\mathcal{S}} \omega\ dx^1 dx^2. \tag{3.34b}$$

Here $\mathcal{C}$ and $\mathcal{S}$ are a curve and a surface, respectively. $o(\cdot) = \pm 1$ is a function indicating the orientation of the domain. Note that this is convention and is related to the chosen direction of the tangent or normal vector.

A L^2 inner product on $\Lambda^k(\Omega)$ can be defined using the Hodge star operator. Let $\omega^{(k)}, \beta^{(k)} \in \Lambda^k(\Omega)$

$$\left(\omega^{(k)}, \beta^{(k)}\right)_\Omega = \int_\Omega \left(\omega^{(k)}, \beta^{(k)}\right)\ d\Omega = \int_\Omega \omega^{(k)} \wedge \star\,\beta^{(k)}. \tag{3.35}$$

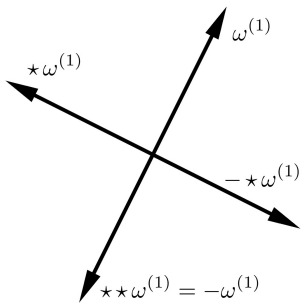


Figure 3.7: The Hodge star operator applied to the 1-form $\omega^{(1)}$. For simplicity it is assumed that the metric tensor $\boldsymbol{g}$ is the identity matrix. From the figure it is clear why $-\star\star\omega^{(1)} = \omega^{(1)}$.

It is worth noting that the integrand of the last integral is a 2-form, regardless of the value for k.

Lastly, let $G : \hat{\Omega} \longrightarrow \Omega$ be the geometric mapping defining Ω (see section 3.1.4). The *pullback* $G^\star : \Lambda^i(\Omega) \longrightarrow \Lambda^i(\hat{\Omega})$ of G is defined such that the following holds

$$\int_\Omega \omega^{(k)} = \int_{G(\hat{\Omega})} \omega^{(k)} = \int_{\hat{\Omega}} G^\star(\omega^{(k)}) \tag{3.36}$$

for f a k-form. $G^\star$ commutes with both the exterior product $\wedge$ and the exterior derivative $\mathbf{d}$, which makes it extremely useful as will be seen.

3.3.1. Outer oriented differential forms

There are two different interpretations of a system of differential forms, the inner oriented (or ordinary) differential forms and the outer oriented (or twisted) differential forms. Their spaces (for a 2-dimensional manifold) are given schematically by

$$\begin{array}{lc} \text{Inner oriented} & \mathbb{R} \longrightarrow \Lambda^0(\Omega) \xrightarrow{\mathbf{d}} \Lambda^1(\Omega) \xrightarrow{\mathbf{d}} \Lambda^2(\Omega) \longrightarrow 0 \\ & \quad\;\; \updownarrow \star \quad\;\; \overset{-\mathbf{d}^\star}{\curvearrowright} \quad\;\; \updownarrow \star \quad\;\; \overset{-\mathbf{d}^\star}{\curvearrowright} \quad\;\; \updownarrow \star \\ \text{Outer oriented} & 0 \longleftarrow \Lambda^2(\Omega) \xleftarrow{\mathbf{d}} \Lambda^1(\Omega) \xleftarrow{\mathbf{d}} \Lambda^0(\Omega) \longleftarrow \mathbb{R}. \end{array} \tag{3.37}$$

Both horizontal sequences of differential forms, connected by the exterior derivative $\mathbf{d}$ are called *de Rham* complexes. In formulating differential equation in terms of differential forms, one of both options for the de Rham complex should be chosen. Depending on the problem that will be discretized, the inner and outer oriented complexes have different properties. This has everything to do with the fact that $\mathbf{d}$ is a metric free operation, while $\mathbf{d}^\star = -\star\mathbf{d}\star$ is not. It means that $\mathbf{d}$ is independent of the underlying coordinates and allows for an exact discrete representation [34].

3.4. Discrete differential forms and polar spline spaces

For isogeometric based FEA the discrete versions of the introduced differential forms are used. Discrete differential forms are represented by B-spline basis functions as introduced in section 3.1. However, the basis functions will be slightly modified. This section summarises some aspects of the construction of polar spline spaces as described by Toshniwal and Hughes [41]. For details it is referred to this paper and to Ref. [51], where only polar spline surfaces of C^1 smoothness are discussed.

3.4.1. Periodic univariate B-spline basis functions

Periodic B-spline basis functions are defined such that the end and starting point of a domain are identical. The periodic basis functions are denoted by $P_i^{(r)}$ to distinguish them from the non-periodic functions. In the knot vector, each knot occurs only once, unless lower continuity is desired. There is no boundary and the number of basis functions is equal to the number of knots. As an example of a periodic B-spline curve, see figure 3.8 which is a variation of the original non-periodic curve given in figure 3.2. Note that even at the starting point of the curve, the C^1 smoothness (for quadratic basis functions) is attained. Again, the only C^0 continuous point is the consequence of a repeated knot vector.

3.4.2. Semi-periodic tensor product space

The semi-periodic bilinear space is defined by combining the periodic basis functions of the previous section with the non-periodic basis function of section 3.1.1. It leads to a shape with the topology of a cylinder as visualised in figure 3.9a.

$$\begin{aligned} N_{ij}^{(0,0)}(\boldsymbol{\xi}) &= P_i(\xi)B_j(\eta), \quad N_{ij}^{(1,0)}(\boldsymbol{\xi}) = \overline{P}_i(\xi)B_j(\eta), \\ N_{ij}^{(0,1)}(\boldsymbol{\xi}) &= P_i(\xi)\overline{B}_j(\eta), \quad N_{ij}^{(1,1)}(\boldsymbol{\xi}) = \overline{P}_i(\xi)\overline{B}_j(\eta). \end{aligned} \tag{3.38}$$

The scaled basis functions are defined as

$$\overline{P}_i^2 = 2P_i^1, \quad \overline{B}_i^2 = 2B_i^1 \tag{3.39}$$

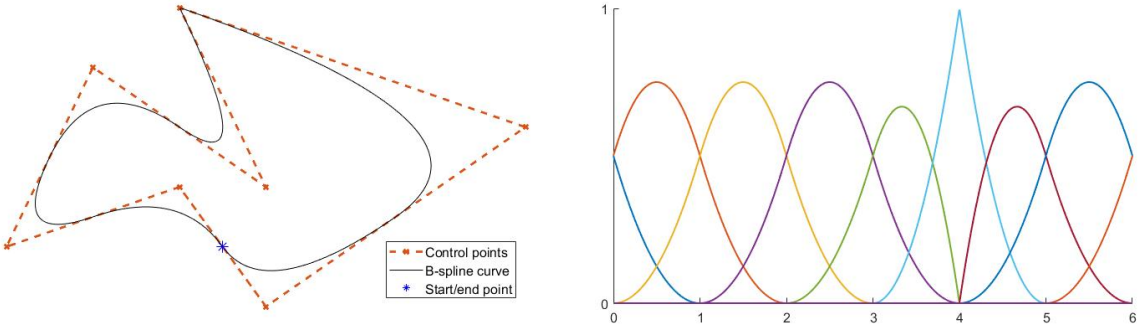


Figure 3.8: Quadratic periodic B-spline curve and corresponding periodic basis functions.
Knot vector: $\Xi = [0, 1, 2, 3, 4, 4, 5, 6 \equiv 0]$

In Eq. (3.38) it is recognised that the highest order of the basis functions is always in superscript. In this report the order will always be equal to 2, and it will be suppressed for readability.

Let

$$\mathcal{S}^{(0,0)} := \operatorname{span}\left\{ N_{ij}^{(0,0)} : \begin{matrix} i = 0, \dots, n^1 - 1, \\ j = 0, \dots, n^2 - 1 \end{matrix} \right\}, \qquad n^{(0,0)} := \dim\left(\mathcal{S}^{(0,0)}\right) = n^1 \times n^2 \tag{3.40a}$$

$$\mathcal{S}^{(1,0)} := \operatorname{span}\left\{ N_{ij}^{(1,0)} : \begin{matrix} i = 0, \dots, n^1 - 1, \\ j = 0, \dots, n^2 - 1 \end{matrix} \right\}, \qquad n^{(1,0)} := \dim\left(\mathcal{S}^{(1,0)}\right) = \overline{n}^1 \times n^2 \tag{3.40b}$$

$$\mathcal{S}^{(0,1)} := \operatorname{span}\left\{ N_{ij}^{(0,1)} : \begin{matrix} i = 0, \dots, n^1 - 1, \\ j = 0, \dots, n^2 - 1 \end{matrix} \right\}, \qquad n^{(0,1)} := \dim\left(\mathcal{S}^{(0,1)}\right) = n^1 \times \overline{n}^2 \tag{3.40c}$$

$$\mathcal{S}^{(1,1)} := \operatorname{span}\left\{ N_{ij}^{(1,1)} : \begin{matrix} i = 0, \dots, n^1 - 1, \\ j = 0, \dots, n^2 - 1 \end{matrix} \right\}, \qquad n^{(1,1)} := \dim\left(\mathcal{S}^{(1,1)}\right) = \overline{n}^1 \times \overline{n}^2. \tag{3.40d}$$

The following spaces are defined for tensor product 0-, 1- and 2-forms, respectively:

$$\Lambda^0_{\mathrm{T}} = \mathcal{S}^{(0,0)}, \quad \Lambda^1_{\mathrm{T}} = \left\{ f_\alpha \boldsymbol{a}^\alpha : f_1 \in \mathcal{S}^{(1,0)},\ f_2 \in \mathcal{S}^{(0,1)} \right\}, \quad \Lambda^2_{\mathrm{T}} = \left\{ f\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 : f \in \mathcal{S}^{(1,1)} \right\} \tag{3.41}$$

Let $\omega \in \Lambda^0_{\mathrm{T}}$, $\boldsymbol{u} \in \Lambda^1_{\mathrm{T}}$, and $q \in \Lambda^2_{\mathrm{T}}$. In terms of basis functions, these differential forms read as

$$\omega = \sum_{i,j} P_i B_j \omega_{ij} \qquad = \left[\boldsymbol{N}^{(0,0)}\right] \cdot [\boldsymbol{\omega}] \tag{3.42a}$$

$$\boldsymbol{u} = \sum_{i,j} \overline{P}_i B_j \boldsymbol{u}^1_{ij}\ \boldsymbol{a}^1 + \sum_{i,j} N_i \overline{B}_j \boldsymbol{u}^2_{ij}\ \boldsymbol{a}^2 \qquad = \left[\boldsymbol{N}^{(1,0)}\right] \cdot \left[\boldsymbol{u}^1\right]\ \boldsymbol{a}^1 + \left[\boldsymbol{P}^{(0,1)}\right] \cdot \left[\boldsymbol{u}^2\right]\ \boldsymbol{a}^2 \tag{3.42b}$$

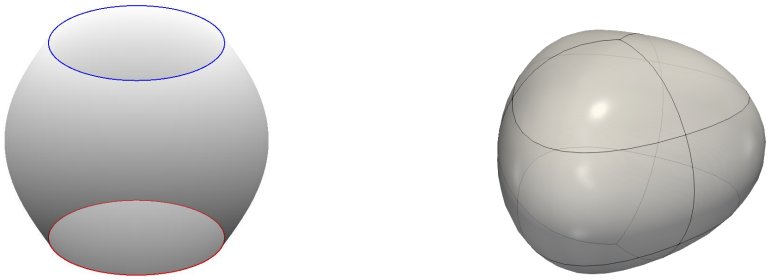
$$q = \sum_{i,j} \overline{N}_i \overline{B}_j q_{ij}\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 \qquad = \left[\boldsymbol{N}^{(1,1)}\right] \cdot [\boldsymbol{q}]\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2. \tag{3.42c}$$

Without going into details, it can be shown that for $\Lambda^1_{\mathrm{T}} \ni \boldsymbol{f} := \mathbf{d}\,\omega$ and $\Lambda^2_{\mathrm{T}} \ni h := \mathbf{d}\,\boldsymbol{u}$ the discrete exterior derivatives can be represented by a few matrices, $\boldsymbol{D}^{(1,0)}$, $\boldsymbol{D}^{(0,1)}$, $\boldsymbol{D}^{(2,0)}$ and $\boldsymbol{D}^{(0,2)}$ [41], such that

$$\begin{aligned} \boldsymbol{f} = \mathbf{d}\,\omega &:= \left[\boldsymbol{N}^{(1,0)}\right] \cdot \boldsymbol{D}^{(1,0)}[\boldsymbol{\omega}]\ \boldsymbol{a}^1 + \left[\boldsymbol{N}^{(0,1)}\right] \cdot \boldsymbol{D}^{(0,1)}[\boldsymbol{\omega}]\ \boldsymbol{a}^2 \quad \text{and} \\ h = \mathbf{d}\,\boldsymbol{u} &:= \left[\boldsymbol{N}^{(1,1)}\right] \cdot \left(-\boldsymbol{D}^{(2,0)}[\boldsymbol{u}^1] + \boldsymbol{D}^{(0,2)}[\boldsymbol{u}^2]\right)\ \boldsymbol{a}^1 \wedge \boldsymbol{a}^2. \end{aligned} \tag{3.43}$$

3.4.3. Polar splines

Polar spline spaces are used to construct sphere-like surfaces (with genus 0) that are C^1 continuous everywhere. This is done by collapsing two lines of the parametric domain to singular points called



(a) A cylinder-like surface

(b) A 3x3 element grid with two collapsed poles.

Figure 3.9: Collapsing both the red and the blue edges of the cylinder to two points, leads to two singularities called 'poles' as depicted on the right

poles. (See figure 3.9.) Again, a geometric mapping similar to Eq. (3.11) defines the surface Ω, but the basis functions are different. Following Ref. [41], there exist matrices

$$\boldsymbol{E}^{(0),\text{pol}}, \quad \boldsymbol{E}^{(1),\text{pol}}, \quad \boldsymbol{E}^{(2),\text{pol}} \tag{3.44}$$

that will be used to define the polar basis functions. Using the semi-periodic basis functions of the previous paragraph, the polar basis functions are defined by,

$$\begin{aligned} \boldsymbol{N}^{(0),\text{pol}} &= \boldsymbol{E}^{(0),\text{pol}} \boldsymbol{N}^{(0,0)}, \\ \boldsymbol{N}^{(2),\text{pol}} &= \boldsymbol{E}^{(2),\text{pol}} \boldsymbol{N}^{(1,1)}, \end{aligned} \qquad \boldsymbol{N}^{(1),\text{pol}} = \begin{bmatrix} \boldsymbol{N}^{(1,0),\text{pol}} \\ \boldsymbol{N}^{(0,1),\text{pol}} \end{bmatrix} = \boldsymbol{E}^{(1),\text{pol}} \begin{bmatrix} \boldsymbol{N}^{(1,0)} \\ \boldsymbol{N}^{(0,1)} \end{bmatrix}. \tag{3.45}$$

The geometric mapping $\boldsymbol{x}$ is now given by

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{A=1}^{n^{(0),\text{pol}}} N_A^{(0),\text{pol}} \boldsymbol{R}_A^{\text{pol}}. \tag{3.46}$$

Note that this mapping is singular at the poles. This does not result in any problems, as the implementation of this setting generally involves the evaluation of quadrature points in the element interiors only.

The polar basis functions define the corresponding spaces

$$\Lambda_S^0(\Omega) := \text{span}\left\{ N_A^{(1),\text{pol}} : A = 0, \dots, n^{(0),\text{pol}} - 1 \right\} \tag{3.47a}$$

$$\Lambda_S^1(\Omega) := \text{span}\left\{ N_A^{(1,0),\text{pol}} \boldsymbol{a}^1 + N_A^{(0,1),\text{pol}} \boldsymbol{a}^2 : A = 0, \dots, n^{(1),\text{pol}} - 1 \right\} \tag{3.47b}$$

$$\Lambda_S^2(\Omega) := \text{span}\left\{ N_A^{(2),\text{pol}} \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 : A = 0, \dots, n^{(2),\text{pol}} - 1 \right\}, \tag{3.47c}$$

and their functions can be written as,

$$\omega = \sum_A N^{(0),\text{pol}} \omega_A \qquad = \left[\boldsymbol{N}^{(0),\text{pol}} \right] \cdot [\boldsymbol{\omega}] \tag{3.48a}$$

$$\boldsymbol{u} = \sum_A N^{(1,0),\text{pol}} \boldsymbol{u}_A^1 \; \boldsymbol{a}^1 + \sum_A N^{(0,1),\text{pol}} \boldsymbol{u}_A^2 \; \boldsymbol{a}^2 \qquad = \left[\boldsymbol{N}^{(1,0),\text{pol}} \right] \cdot \left[\boldsymbol{u}^1 \right] \; \boldsymbol{a}^1 + \left[\boldsymbol{N}^{(0,1),\text{pol}} \right] \cdot \left[\boldsymbol{u}^2 \right] \; \boldsymbol{a}^2 \tag{3.48b}$$

$$q = \sum_A N^{(2)\text{pol}} q_A \; \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 \qquad = \left[\boldsymbol{N}^{(2),\text{pol}} \right] \cdot [\boldsymbol{q}] \; \boldsymbol{a}^1 \wedge \boldsymbol{a}^2 \tag{3.48c}$$

Similar to the discrete representation of the exterior derivative of Eq. (3.43) in the semi-periodic setting, there is a similar representation for the polar setting [41]. The relations read as

$$\boldsymbol{f} = \mathbf{d}\,\omega := \begin{bmatrix} \boldsymbol{N}^{(1,0),\mathrm{pol}}\,\boldsymbol{a}^1 \\ \boldsymbol{N}^{(0,1),\mathrm{pol}}\,\boldsymbol{a}^2 \end{bmatrix} \cdot \boldsymbol{D}^{(0),\mathrm{pol}}\,[\boldsymbol{\omega}]\,, \qquad h = \mathbf{d}\,\boldsymbol{u} := \boldsymbol{N}^{(2),\mathrm{pol}} \cdot \boldsymbol{D}^{(1),\mathrm{pol}}\,[\boldsymbol{u}]\;\; \boldsymbol{a}^1 \wedge \boldsymbol{a}^2. \tag{3.49}$$

It is important to note that the polar basis functions $\left\{N_A^{(k),\mathrm{pol}}\right\}_{A=1}^{n^{(k),\mathrm{pol}}}$ have properties similar to the basis functions defined in section 3.1. That is, they form a partition of unity $\sum_{A=1}^{n_P} P_A = 1$, each P_A is C^1 continuous and they are linear independent from each other [41, 127]. Furthermore, the local support of the polar basis function is analogous to the ordinary B-spline basis function. Only at the poles, the functions are quite different.

4 | Continuous strong and weak forms

This chapter presents the differential equations corresponding to fluid films. The full system is rewritten in terms of differential forms and the continuous weak formulation is derived.

4.1. Strong formulation of the fluid film equations

The strong formulation as derived in Ref. [39] is presented. These equations are equivalent to the fluid film equations of Ref. [52]. On the domain Ω the following holds

$$-\operatorname{grad}_\Omega p + \mu\left[-\left(\mathbf{d}^\star\,\mathbf{d}\,\boldsymbol{v}^\flat\right)^\sharp - 2(\boldsymbol{k}-2H\boldsymbol{g})^\sharp\cdot\operatorname{grad}_\Omega v^{\mathrm{m}} + 2K\boldsymbol{v}\right] + \boldsymbol{f} = \boldsymbol{0} \tag{4.1a}$$

$$\operatorname{div}_\Omega\cdot\boldsymbol{v} - 2v^{\mathrm{m}}H = 0 \tag{4.1b}$$

$$-2Hp + 2\mu\left(\boldsymbol{\nabla v} : \boldsymbol{k} - \left(4H^2-2K\right)v^{\mathrm{m}}\right) + f^{\mathrm{n}} = 0. \tag{4.1c}$$

subject to Dirichlet boundary conditions,

$$\boldsymbol{v} = \boldsymbol{v}_\partial \quad \text{on } \partial\Omega. \tag{4.2}$$

The strong formulation contains an in-plane equation (Eq. (4.1a)), which, together with the incompressibility constraint (Eq. (4.1b)), forms the surface Stokes equation. Eq. (4.1c) describes the out-of-plane or normal behavior of the domain. In the equations, p is the pressure, which is the negative of the surface tension. $\boldsymbol{v}$ is the in-plane velocity vector field and v^{m} the mesh velocity normal to the surface. Note that $\boldsymbol{v}$ and v^{m} appear in both the in-plane and the out-of-plane equations. This indicates the coupling between surface flow and normal velocities. $\boldsymbol{f}$ is the surface forcing, while f^{n} is the pressure difference between the inner and outer bulk fluids. Both of them are known. The surface quantities $\boldsymbol{g}, \boldsymbol{k}, H, K$ are the metric tensor, the curvature tensor, the mean curvature and the Gaussian curvature, respectively. All these quantities depend on the shape of the surface. Note that in some papers (like in Refs. [39, 41]), H denotes twice the mean curvature, in stead of just the mean curvature. The term containing the double dot product ':' in Eq. (4.1c) reads as,

$$\boldsymbol{\nabla v} : \boldsymbol{k} = \operatorname{Tr}(\nabla\boldsymbol{v}\cdot\boldsymbol{k}) = \operatorname{Tr}\left(\begin{bmatrix} g^{1\gamma}v^1_{;\gamma} & g^{2\gamma}v^1_{;\gamma} \\ g^{1\gamma}v^2_{;\gamma} & g^{2\gamma}v^2_{;\gamma}\end{bmatrix}\cdot\begin{bmatrix} k_{11} & k_{12} \\ k_{12} & k_{22}\end{bmatrix}\right) = g^{\alpha\gamma}v^\beta_{;\gamma}k_{\beta\alpha} = v_{\alpha;\beta}k^{\alpha\beta} \tag{4.3}$$

Lastly, μ is the viscosity of the fluid.

4.2. The fluid film equations in terms of differential forms

The Strong formulation presented at the start of this section is now rewritten in terms of differential forms. The choice of outer oriented differential forms will first be motivated in section 4.2.1. Subsequently the final strong formulation that is used in this research is presented.

4.2.1. Outer oriented differential forms

For the Stokes discretization of this research, a pointwise satisfaction of the incompressibility constraint is desired to prevent numerical inconsistencies in the IGA formulation. In order to do so, the outer oriented complex is adopted [34, 131], such that the expression $\operatorname{div}_\Omega\cdot\boldsymbol{v} = 0$ can be represented by

$$\mathbf{d}\,\boldsymbol{u} = 0 \tag{4.4}$$

in a discrete, but point-wise exact manner (see section 3.3.1). Here $\boldsymbol{u} = \star\,\boldsymbol{v}^\flat$ is an outer oriented 1-form. If one observes figure 3.7, it becomes clear why $\boldsymbol{u}$ is called a twisted form: $\boldsymbol{u}$ is orthogonal to $\boldsymbol{v}$. It is noted that $\boldsymbol{v}^\flat$ is an inner oriented 1-form, and that $\boldsymbol{v}$ is a *proxy field* (that is, corresponding vector field) of $\boldsymbol{u}$.

The variables in the strong formulation of Eq. (4.1) are: the fluid surface velocity given by the vector field $\boldsymbol{v} = v^\alpha \boldsymbol{a}_\alpha$; the pressure function p; the surface normal velocity component v^m. The originals $\boldsymbol{v}, p, v^\mathrm{m}$ are in fact proxy fields (that is, scalar of vector field corresponding to 0-, 1-, or 2-forms) for the corresponding outer oriented differential forms $\boldsymbol{u} \in \Lambda^1(\Omega)$, $q, \nu \in \Lambda^2(\Omega)$, respectively. Their relationships are given by $\boldsymbol{u} = \star \boldsymbol{v}^\flat$, $q = \star p$, $\nu = \star v^\mathrm{m}$. Lastly, a new unknown is introduced: the voriticity 0-form,

$$\Lambda^0 \ni \omega = \mathbf{d}^\star \boldsymbol{u}. \tag{4.5}$$

This is the last ingredient needed for the vorticity-velocity-pressure formulation of the Stokes equation and the out-of-plane shape equation expressed in terms of the outer oriented differential forms. It is noted that in the discrete setting, Eq. (4.5) will not be represented exactly, a sacrifice for exact satisfaction of Eq. (4.4).

4.2.2. Restatement of Strong form

The strong formulation as in Eq. (4.1) expressed in terms of the proxy fields is now restated in terms of the corresponding differential forms. The restated strong formulation will be the final version used in this research. Set $\mu = 1$ and $\boldsymbol{f} = 0$ for the surface forcing. Substituting the outer oriented differential forms into Eq. (4.1a) results in

$$-\operatorname{grad}_\Omega \star q + (\mathbf{d}^\star \, \mathbf{d} \star \boldsymbol{u})^\sharp - 2(\boldsymbol{k} - 2H\boldsymbol{g})^\sharp \cdot \operatorname{grad}_\Omega \star \nu - 2K(\star \boldsymbol{u})^\sharp = \mathbf{0}. \tag{4.6}$$

Now $(\cdots)^\flat$ and $\star$ are consecutively applied to each term of the Eq. (4.6) which gives

$$\begin{aligned} -\star\mathbf{d}\star q - \star\star\mathbf{d}\star\mathbf{d}\star\boldsymbol{u} - \star 2(\boldsymbol{k} - 2H\boldsymbol{g}) \cdot \mathbf{d}\star\nu - \star\star 2K\boldsymbol{u} &= \mathbf{0} \\ \implies \qquad \mathbf{d}^\star q - \mathbf{d}\,\mathbf{d}^\star \boldsymbol{u} - \star 2(\boldsymbol{k} - 2H\boldsymbol{g}) \cdot \mathbf{d}\star\nu + 2K\boldsymbol{u} &= \mathbf{0}. \end{aligned} \tag{4.7}$$

Similarly, the outer oriented differential forms are substituted into Eq. (4.1b),

$$\begin{aligned} \star\mathbf{d}\star\boldsymbol{v} - 2Hv^\mathrm{m} &= 0, \\ \implies \quad \star\mathbf{d}\,\boldsymbol{u} - 2H\star\nu &= 0 \\ \implies \quad \mathbf{d}\,\boldsymbol{u} - 2H\nu &= 0. \end{aligned} \tag{4.8}$$

and into Eq. (4.1c),

$$\begin{aligned} -2Hp + 2\boldsymbol{\nabla}\boldsymbol{v} : \boldsymbol{k} - \left(H^2 - 2K\right) v^\mathrm{m} &= -f^\mathrm{n} \\ \implies \quad -2H\star q - 2\boldsymbol{\nabla}\star\boldsymbol{u} : \boldsymbol{k} - 2\left((2H)^2 - 2K\right)\star\nu &= -f^\mathrm{n}. \end{aligned} \tag{4.9}$$

The vorticity (Eq. (4.5)) is introduced and the static strong formulation reads as

$$\omega - \mathbf{d}^\star \boldsymbol{u} = 0 \tag{4.10a}$$

$$-\mathbf{d}\,\omega + 2K\boldsymbol{u} + \mathbf{d}^\star q - \star(2(\boldsymbol{k} - 2H\boldsymbol{g}) \cdot \mathbf{d}\star\nu) = \mathbf{0} \tag{4.10b}$$

$$\mathbf{d}\,\boldsymbol{u} - 2H\nu = 0 \tag{4.10c}$$

$$-2H\star q - 2\boldsymbol{\nabla}\star\boldsymbol{u} : \boldsymbol{k} - 2\left(4H^2 - 2K\right)\star\nu = -f^\mathrm{n}. \tag{4.10d}$$

The surface is closed and, therefore, no boundary terms are needed.

Since the surface evolves in time, the surface evolution equation is added to the system. Note that all known and unknwon quantities of the above strong formulation are depending on both the position $\boldsymbol{x}$ (or $\boldsymbol{\xi}$) and time t. The surface position $\boldsymbol{x}(\boldsymbol{\xi}, t)$ is defined by its initial position $\boldsymbol{x}(\boldsymbol{\xi}, t_0)$ at time t_0 and the surface velocity $v^\mathrm{m}(\boldsymbol{\xi}, t) = \star\nu(\boldsymbol{\xi}, t)$ integrated over time,

$$\boldsymbol{x}(\boldsymbol{\xi}, t) = \boldsymbol{x}(\boldsymbol{\xi}, t_0) + \int_{t_0}^{t} v^\mathrm{m}(\boldsymbol{\xi}, s)\boldsymbol{n}(\boldsymbol{\xi}, s)ds \tag{4.11}$$

Note that Eqs. (4.10a)-(4.10c) are identical to the strong formulation of [41].

4.2.3. Weak formulation

The weak formulation corresponding to Eq. (4.10) will now be derived. The equations of the strong formulation correspond to a 0-form, 1-form, 2-form and another 0-form equation, respectively. These indicate the corresponding test functions spaces. Multiplying with the test functions and integrating the strong formulation leads to:

Find $\omega \in \Lambda^0(\Omega)$, $\boldsymbol{u} \in \Lambda^1(\Omega)$, $q \in \Lambda^2(\Omega)$ *and* $\nu \in \Lambda^2(\Omega)$ *such that for all* $\zeta_0, \zeta_0' \in \Lambda^0(\Omega)$, $\zeta_1 \in \Lambda^1(\Omega)$ *and* $\zeta_2 \in \Lambda^2(\Omega)$ *the following holds:*

$$(\zeta_0, \omega)_\Omega - (\zeta_0, \mathbf{d}^\star \boldsymbol{u})_\Omega = 0 \tag{4.12a}$$

$$-(\boldsymbol{\zeta}_1, \mathbf{d}\,\omega)_\Omega + (\boldsymbol{\zeta}_1, 2K\boldsymbol{u})_\Omega + (\boldsymbol{\zeta}_1, \mathbf{d}^\star q)_\Omega - (\boldsymbol{\zeta}_1, \star 2[\boldsymbol{k} - 2H\boldsymbol{g}] \cdot \mathbf{d} \star \nu)_\Omega = 0 \tag{4.12b}$$

$$(\zeta_2, \mathbf{d}\,\boldsymbol{u})_\Omega + (\zeta_2, \sigma)_\Omega - (\zeta_2, H\nu)_\Omega = 0 \tag{4.12c}$$

$$\left(\zeta_2^\Sigma, q\right)_\Omega = 0 \tag{4.12d}$$

$$-(\zeta_0', 2\boldsymbol{\nabla} \star \boldsymbol{u} : \boldsymbol{k})_\Omega - (\zeta_0', 2H \star q)_\Omega - 2\left(\zeta_0', \left[(2H)^2 - 2K\right] \cdot \star \nu\right)_\Omega = -(\zeta_0', f^{\mathrm{n}})_\Omega \tag{4.12e}$$

Note that two extra terms are included. Eq. (4.12d) counteracts the rank deficiency of the associated linear system by requiring a mean surface tension of 0. The term $(\zeta_2, \sigma)_\Omega$ is simply the transpose of $\left(\zeta_2^\Sigma, q\right)_\Omega$.

5 | Discretization

In this section the discretization process will be described. In section 5.1 the discrete weak formulation is obtained, while time is discretized in section 5.2. In appendix A.3, the simpler 2D Stokes equation is also discretized. In the appendix section, the reader can get more familiar with the transformations that are used for the 0-, 1-, and 2-forms.

5.1. Space discretization of the fluid film equations

Now remember the relation between $\mathbf{d}$ and $\mathbf{d}^\star$ given by Eq. (3.31). Note that $\partial\Omega = \emptyset$. This will be used to express the weak formulation fully in terms of the exterior derivative $\mathbf{d}$:

Find $\omega \in \Lambda^0_S(\Omega)$, $\boldsymbol{u} \in \Lambda^1_S(\Omega)$, $q \in \Lambda^2_S(\Omega)$ *and* $\nu \in \Lambda^2_S(\Omega)$ *such that for all* $\zeta_0, \zeta_0' \in \Lambda^0_S(\Omega)$, $\zeta_1 \in \Lambda^1_S(\Omega)$ *and* $\zeta_2 \in \Lambda^2_S(\Omega)$ *the following holds:*

$$(\zeta_0, \omega)_\Omega - (\mathbf{d}\,\zeta_0, \boldsymbol{u})_\Omega = 0 \tag{5.1a}$$

$$-(\boldsymbol{\zeta}_1, \mathbf{d}\,\omega)_\Omega + (\boldsymbol{\zeta}_1, 2K\boldsymbol{u})_\Omega + (\mathbf{d}\,\boldsymbol{\zeta}_1, q)_\Omega - (\boldsymbol{\zeta}_1, \star 2[\boldsymbol{k} - 2H\boldsymbol{g}] \cdot \mathbf{d}\star\nu)_\Omega = 0 \tag{5.1b}$$

$$(\zeta_2, \mathbf{d}\,\boldsymbol{u})_\Omega + (\zeta_2, \sigma)_\Omega - (\zeta_2, H\nu)_\Omega = 0 \tag{5.1c}$$

$$\left(\zeta_2^\Sigma, q\right)_\Omega = 0 \tag{5.1d}$$

$$-(\zeta_0', 2\boldsymbol{\nabla}\star\boldsymbol{u} : \boldsymbol{k})_\Omega - (\zeta_0', 2H\star q)_\Omega - 2\left(\zeta_0', \left[(2H)^2 - 2K\right]\star\nu\right)_\Omega = -(\zeta_0', f^{\mathrm{n}})_\Omega \tag{5.1e}$$

The associated linear system (for fixed Ω) reads as

$$\boldsymbol{M}\boldsymbol{\psi} = \boldsymbol{f} \tag{5.2}$$

with known

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}^{\zeta_0\omega} & \boldsymbol{M}^{\zeta_0\boldsymbol{u}} & & & \\ \boldsymbol{M}^{\zeta_1\omega} & \boldsymbol{M}^{\zeta_1\boldsymbol{u}} & \boldsymbol{M}^{\zeta_1 q} & & \boldsymbol{M}^{\zeta_1\nu} \\ & \boldsymbol{M}^{\zeta_2\boldsymbol{u}} & & \boldsymbol{M}^{\zeta_2\sigma} & \boldsymbol{M}^{\zeta_2\nu} \\ & & \boldsymbol{M}^{\zeta_2^\Sigma q} & & \\ & \boldsymbol{M}^{\zeta_0'\boldsymbol{u}} & \boldsymbol{M}^{\zeta_0' q} & & \boldsymbol{M}^{\zeta_0'\nu} \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 0 \\ \boldsymbol{f}^{\zeta_0'} \end{bmatrix} \tag{5.3}$$

and unknown control variables

$$\boldsymbol{\psi} = \begin{bmatrix} [\boldsymbol{\omega}] \\ [\boldsymbol{u}] \\ [\boldsymbol{q}] \\ [\sigma] \\ [\boldsymbol{\nu}] \end{bmatrix}. \tag{5.4}$$

The submatrices of matrix $\boldsymbol{M}$ are expressed as $\boldsymbol{M}^{AB}$ where A and B indicate the related trial functions (rows) and test functions (columns), respectively. The submatrices are constructed using Gaussian quadrature on each of the elements to evaluate the integrals of the weak form. The matrices are not constructed entry by entry, rather for each element the corresponding integrals are calculated using Gaussian quadrature points. Integrals of integrands that include basis function that are known to be zero a-priori (due to the local support) are omitted for efficiency.

Observe the submatrix of $\boldsymbol{M}$,

$$\boldsymbol{M}_{\text{Stokes}} = \begin{bmatrix} \boldsymbol{M}^{\zeta_0\omega} & \boldsymbol{M}^{\zeta_0\boldsymbol{u}} & & \\ \boldsymbol{M}^{\zeta_1\omega} & \boldsymbol{M}^{\zeta_1\boldsymbol{u}} & \boldsymbol{M}^{\zeta_1 q} & \\ & \boldsymbol{M}^{\zeta_2\boldsymbol{u}} & & \boldsymbol{M}^{\zeta_2\sigma} \\ & & \boldsymbol{M}^{\zeta_2^\Sigma q} & \end{bmatrix}. \tag{5.5}$$

It is exactly equal to the left-hand side matrix associated with the final weak formulation of Ref. [41] (Eq. (121)) and $M_{\text{Stokes}} = M_{\text{Stokes}}^T$.

5.2. Time discretization of surface evolution

Remember that the surface evolution equation of is given by Eq. (4.11). In the weak formulation of Eq. (5.1) the occurring surface characteristics $K, H, \boldsymbol{k}, \boldsymbol{g}$, the differential calculus operators $\star$, $\mathbf{d}$ and the inner products depend on the surface position $\boldsymbol{x}(\boldsymbol{\xi}, t)$ of $\Omega(t)$ at time t. As a consequence, the system is non-linear in time. However, keeping the surface momentarily fixed, leads to a linear system of equations. The time discretization of the surface evolution equation (Eq. (4.11)) will be performed using an implicit-Euler scheme. Is described in the following subsections, the system is solved by alternating between solving the shape equation and the Stokes equation. The solutions for both the shape equation and the Stokes equation at a new time step, which directly influences the surface deformation, will be computed iteratively using the fixed-point method.

A summary of time stepping and of the the fixed-point iterations can be found in the pseudo code of algorithm 1. Here subscripts n denotes the time, while subscripts i indicates the iteration. A schematic summary of the algorithm, including data flow, can be found in figure 5.1. Lastly, the online repository for the Matlab code can be found here. The following subsection go into details of the algorithm. For readability, the time and iteration subscripts are sometimes suppressed in the text.

5.2.1. Notation

Let Δt be the size of the time step, such that the sequence $t_0 < t_1 < \ldots < t_n < \ldots < t_{\text{end}-1} < t_{\text{end}}$ increases with Δt. Ω^n denotes the surface at time t_n defined by geometric mapping,

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{A=1} N_A^{(0),\text{pol}} \boldsymbol{R}_A^{\text{pol}}, \tag{5.6}$$

with time dependent control points $[\boldsymbol{R}^n] = [\boldsymbol{R}](t_n)$. The discrete fundamental unknown functions at time t_n are given by ω^n, $\boldsymbol{u}^n$, q^n, σ^n (the Stokes variables) and ν^n the normal velocity defining the surface deformation. The control variables and the control variable offsets of the Stokes unknowns are stored in, respectively,

$$[\boldsymbol{\phi}_n] = \begin{bmatrix} [\boldsymbol{\omega}_n] \\ [\boldsymbol{u}_n] \\ [\boldsymbol{q}_n] \\ [\sigma_n] \end{bmatrix}, \quad [\Delta\boldsymbol{\phi}_{n+1}] = \begin{bmatrix} [\Delta\boldsymbol{\omega}_{n+1}] \\ [\Delta\boldsymbol{u}_{n+1}] \\ [\Delta\boldsymbol{q}_{n+1}] \\ [\Delta\sigma_{n+1}] \end{bmatrix}, \tag{5.7}$$

such that

$$[\boldsymbol{\phi}_{n+1}] = [\boldsymbol{\phi}_n] + [\Delta\boldsymbol{\phi}_{n+1}]\,. \tag{5.8}$$

Additionally, $[\boldsymbol{\nu}_n]$ represents the control variables of the normal velocity 2-form, and $[\boldsymbol{v}_n^{\text{m}}]$ and $[\Delta\boldsymbol{v}_{n+1}^{\text{m}}]$ represent the control variables of the corresponding 0-form mesh velocity and its offset function, respectively.

5.2.2. Adjustments of the linear system

In the fixed-point iterations, solving the Stokes equation and the shape equation is done in two separate steps (figure 5.1). First, the shape equation is solved given the control points of the in-plane velocity $[\boldsymbol{u}_{n+1}]_i$ and the pressure $[\boldsymbol{q}_{n+1}]_i$ at the previous iteration, expressed by the vectors

$$[\boldsymbol{h}_{n+1}^{\boldsymbol{u}}]_i = [\boldsymbol{M}_{n+1}^{\zeta_0'\boldsymbol{u}}]_i \cdot [\boldsymbol{u}_{n+1}]_i \quad \text{and} \quad [\boldsymbol{h}_{n+1}^{q}]_i = [\boldsymbol{M}_{n+1}^{\zeta_0' q}]_i \cdot [\boldsymbol{q}_{n+1}]_i\,, \tag{5.9}$$

respectively. Furthermore, the unknown normal velocity 2-form ν, replaced by the mesh velocity 0-form v^{m}, such that the third term of Eq. (5.1e) becomes

$$-2\left(\zeta_0', \left[(2H)^2 - 2K\right] \star \nu\right)_\Omega \longrightarrow -2\left(\zeta_0', \left[(2H)^2 - 2K\right] v^{\text{m}}\right)_\Omega, \tag{5.10}$$

leading to a square matrix $\boldsymbol{M}^{\zeta_0' v^{\text{m}}}$. The corresponding linear system reads,

$$\boxed{[\boldsymbol{M}_{n+1}^{\zeta_0' v^{\text{m}}}]_i \cdot \left[v_{n+1}^{\text{m}}\right]_{i+1} = \left[\boldsymbol{f}_{n+1}^{\text{shape}}\right]_i}, \tag{5.11}$$

where

$$\left[\boldsymbol{f}_{n+1}^{\text{shape}}\right]_i = \boldsymbol{f}_{n+1}^{\zeta_0'} - [\boldsymbol{h}_{n+1}^{\boldsymbol{u}}]_i - [\boldsymbol{h}_{n+1}^{q}]_i. \tag{5.12}$$

Subsequently, the Stokes equation is solved given the control variables $[\nu_{n+1}]_{i+1}$ of the normal velocity. For the Stokes equation there is no adjustment needed. However, since the shape equation is solves for the mesh velocity $\left[v_{n+1}^{\text{m}}\right]_{i+1}$, the input variable

$$[\nu_{n+1}]_{i+1} = \left[\star\, v_{n+1}^{\text{m}}\right]_{i+1} \tag{5.13}$$

is obtained by applying the Hodge star. This operation can not be performed exactly and in stead, $\star\, v^{\text{m}}$ is projected on to 2-form space Λ_S^2. It is performed by multiplying Eq. (5.13) with a test function and integrating, leading to a weak form:

$$\begin{gathered}\textit{Find } \nu \in \Lambda_S^2(\hat{\Omega}) \textit{ such that for all } \zeta_2 \in \Lambda_S^2(\hat{\Omega}), \textit{ the following holds:} \\ (\zeta_2, \nu)_{\hat{\Omega}} = (\zeta_2, \star\, v^{\text{m}})_{\hat{\Omega}}.\end{gathered} \tag{5.14}$$

It can be expressed in terms of a linear system,

$$\left[\boldsymbol{M}_{n+1}^{\zeta_2 q}\right]_{i+1} \cdot [\boldsymbol{\nu}_{n+1}]_{i+1} = \left[\boldsymbol{h}_{n+1}^{v^{\text{m}}}\right]_{i+1}, \tag{5.15}$$

where $\boldsymbol{M}^{\zeta_2 q}$ and $\boldsymbol{h}^{v^{\text{m}}}$ correspond to $(\zeta_2, \nu)_{\hat{\Omega}}$ and $(\zeta_2, \star\, v^{\text{m}})_{\hat{\Omega}}$, respectively.

The Stokes system now reads as

$$\boxed{\left[\boldsymbol{M}_{n+1}^{\text{Stokes}}\right]_i [\phi_{n+1}]_i = \left[\boldsymbol{f}_{n+1}^{\text{Stokes}}\right]_{i+1}}, \tag{5.16}$$

where

$$\left[\boldsymbol{f}_{n+1}^{\text{Stokes}}\right]_{i+1} = \begin{bmatrix} \boldsymbol{0} \\ \left[\boldsymbol{M}_{n+1}^{\zeta_1 \nu}\right]_i \cdot [\nu_n]_{i+1} \\ \left[\boldsymbol{M}_{n+1}^{\zeta_2 \nu}\right]_i \cdot [\nu_n]_{i+1} \\ 0 \end{bmatrix}. \tag{5.17}$$

5.2.3. Geometry evolution

The movement of the surface (Eq. (4.11)) is discretized in time by the backwards-Euler method,

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta t \Delta v_{n+1}^{\text{m}} \boldsymbol{n}, \tag{5.18}$$

with normal vector $\boldsymbol{n}$ and time step $\Delta t = t_{n+1} - t_n$. Rewriting Eq. (5.18) in terms of the geometry control points gives

$$[\boldsymbol{R}_{n+1}]_{i+1} = [\boldsymbol{R}_{n+1}]_i + \Delta t \left[\Delta v_{n+1}^{\text{m}}\right]_{i+1} \odot [\boldsymbol{n}_{n+1}]_i. \tag{5.19}$$

Here, $[\boldsymbol{R}] \in \mathbb{R}^{n_P \times 3}$ are the control vectors of Ω_n, $[\Delta v^{\text{m}}] \in \mathbb{R}^{n_P \times 1}$ are the mesh velocity offset control points (0-form) and $[\boldsymbol{n}] \in \mathbb{R}^{n_P \times 3}$ are the control vectors of the discrete normal vector (0-form). $\odot$ indicates a row-wise multiplication. $[\boldsymbol{n}]$ is obtained in much the same way as $[\nu]$ is obtained in Eq. (5.15). However, this time a projection onto the 0-form space $\Lambda^0(\hat{\Omega})$ is performed using the system,

$$\boldsymbol{M}^{\zeta_0 \omega} [\boldsymbol{n}] = \boldsymbol{h}^{\boldsymbol{n}}. \tag{5.20}$$

Here, $\boldsymbol{h}^{\boldsymbol{n}}$ is a three-column matrix corresponding to

$$(\zeta_0, n^i)_{\Omega} \quad \text{for } i \in \{1, 2, 3\} \tag{5.21}$$

Note that this left-hand-side matrix is identical to the top left matrix of $\boldsymbol{M}$ (of Eq. (5.2)).

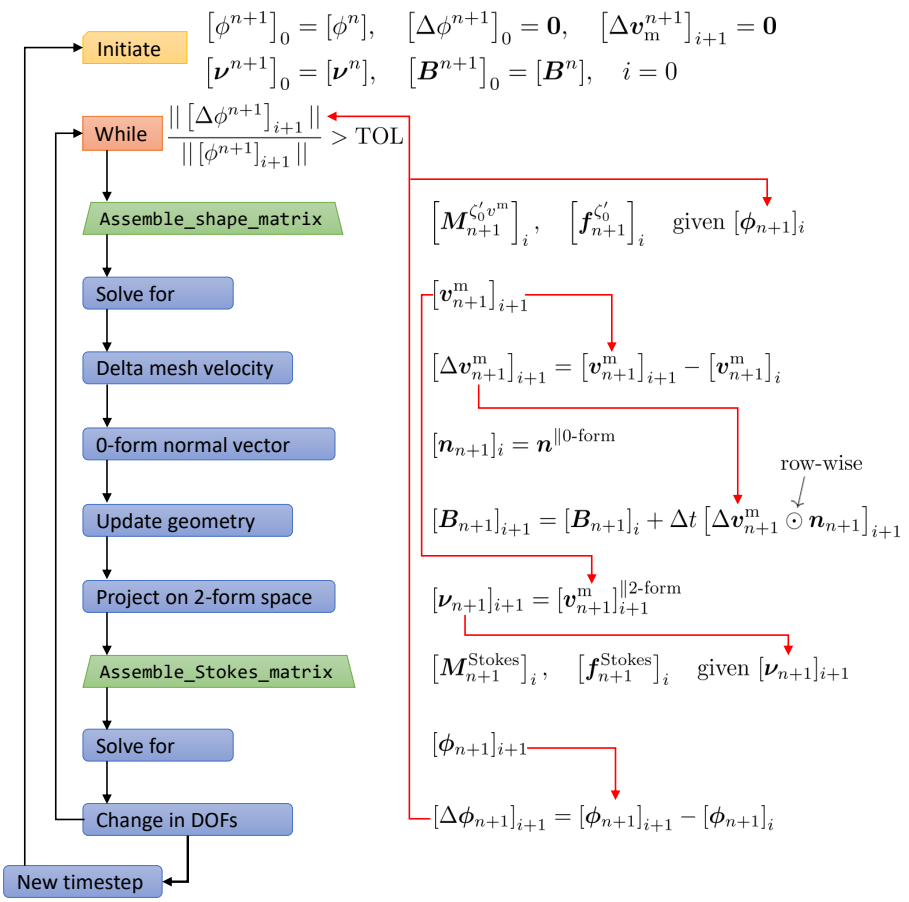


Figure 5.1: Flow chart of the fixed point loop. The black arrows indicate the sequence of steps, while the red arrows visualize the data dependencies.

Algorithm 1: Matlab pseudo code for time step and fixed point loops

```
for t_step = 1:t_end
    time = time + D_t;

    % Initiate DOFs:
    % - phi (Stokes DOFs: vorticity (0-form), in-plane vel (1-form), pressure (2-form))
    % - v_m (out-of-plane vel (0-form))
    % - spline_space (geometry spline space)
    [phi_new, v_m_new, spline_space] = initiate_DOFs();
    [D_phi, D_v_m] = initate_deltas();

    while norm(D_phi)/norm(phi_new) > TOL
        itCount = itCount + 1;

        % Set old DOFs
        phi_old = phi_new;
        v_m_old = v_m_new;

        % Assemble & solve shape Eq given the Stokes variables, phi
        [LHS_shape, RHS_shape] = assemble_shape_matrix(spline_space, params, phi_old);
        v_m_new = LHS_shape\RHS_shape; % 0-form
        D_v_m  = v_m_new - v_m_old; % Delta vel

        % Project normal vector --> 0-form space
        Mass    = CS_assemble_mass_matrix(spline_space, params);
        RHS_nrml = L2_projection_normal_RHS(spline_space, params);
        normal  = Mass{1}\RHS_nrml; % 0-form

        % Update geometry & spline space
        v_m_vector = D_v_m.*normal; % element-wise multiplication
        spline_space = update_geometry(spline_space, D_t, v_m_vector);

        % Project v_m (0-form) --> nnu (2-form)
        Mass   = CS_assemble_mass_matrix(spline_space, params);
        RHS_nnu = L2_projection_2form_RHS(spline_space, params, v_m_new;
        nnu    = Mass{3}\RHS_nnu; % 2-form

        % Assemble & solve Stokes Eq given nnu
        [LHS_Stokes, RHS_Stokes] = assemble_Stokes_matrix(spline_space, params, nnu);
        phi_new   = LHS_Stokes\RHS_Stokes;

        % Delta Stokes DOFs
        D_phi = phi_new - phi_old;
    end
end
```

6 | Results

In this chapter the results of Stokes flow on a curved polar-spline geometry, with and without out-of-plane deformation, is discussed. After the gained experience with the self-written code (see appendix A.4), the Matlab implementation of Ref. [41] was utilized. This code is first used to reproduce the results of the paper for static surfaces with a prescribed instantaneous normal velocity field. Subsequently, in section 6.2, the code is extended to the full discrete system as presented in Eq. (5.1) by incorporating time evolution of the domain.

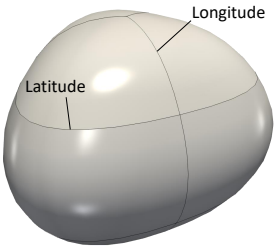


Figure 6.1: The basic domain on the coarsest level. The latitudinal and longitudinal coordinate directions are indicated.

6.1. Stokes flow on a static closed surface

In this research, the code corresponding to Ref. [41] is adopted and extended. Therefore, it is only natural to include the bi-polar surface as constructed in this paper (for order $p = 2$). The domain is visualized for the coarsest level in figure 6.1 (3×3 elements) and will from now on be called the *basic domain*. The latitudinal and longitudinal coordinate directions are also indicated in the figure.

The Stokes flow on the basic domain with a prescribed instantaneous velocity is now presented. A result of Ref. [41] is reproduced, using the weak formulation of Eq. (5.1a)-(5.1d) with a prescribed normal velocity 2-form,

$$\nu_{\text{ex}} = \sqrt{g} f_{12}\, \boldsymbol{a}^1 \wedge \boldsymbol{a}^1 \quad \text{with} \quad f_{12} = \cos\left(2\pi\xi^1\right)\left(\cos\left(2\pi\xi^2\right) - 1\right)^2 . \tag{6.1}$$

However, the result presented in figure 6.2 differs form the result found in Ref. [41] (Fig. 15a), because of a resolved error in the original code. The implementation error had effect on the mean curvature H in the mesh velocity term of Eq. (5.1b). The mean curvature of Eq. (5.1c) was implemented correctly. The error led to an inconsistency in the normal velocity source terms, with a non-physical solution as the consequence.

Observing figure 6.2, positive and negative instantaneous normal velocities lead to low and high pressures, respectively. Note that the negative of the pressure equals the surface tension. As expected, the in-plane fluid flows away from high pressures areas and towards low pressure areas and the flow supports the out-of-plane deformation.

6.2. Time-evolving fluid film

The problem of the Stokes equation on a time-evolving domain is described in sections 5.1 and 5.2. The system is driven by a uniform pressure difference, $b^{\text{n}} = 1$, which corresponds to a higher interior pressure compared to the exterior of the surface. As pointed out in Ref. [59], this makes sense if one does not account for viscous bulk interactions. Time stepping is performed using the iterative fixed-point method.

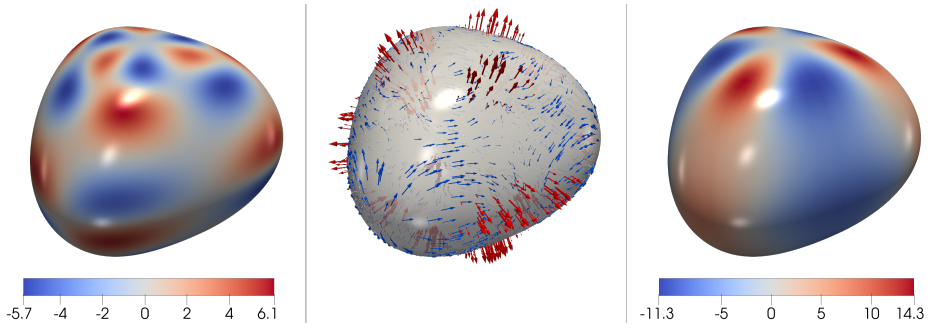


Figure 6.2: The solution to the Stokes equation on the static basic domain is visualized here. Left shows the vorticity ω; in the middle the blue arrows indicate the in-plane fluid flow $\boldsymbol{v} = (-\star \boldsymbol{u})^\sharp$ and the red arrows show the prescribed instantaneous normal velocity $(\star \nu_{\text{ex}})\boldsymbol{n}$; the right figure corresponds with the pressure q. Note that the pressure equals the negative of the surface tension.

6.2.1. The basic domain

For different refinement levels of the basic domain, the results for the iterations of the first time step can be found in figure 6.3. For convenience the grids of i, h-refinements is denoted by H_i. The convergence plots show the relative norm of the solution increment. For example, the value for the vorticity at time step n and iteration i is reads

$$\frac{\left|\left|[\omega_n]_i - [\omega_n]_{i-1}\right|\right|_2}{\left|\left|[\omega_n]_{i-1}\right|\right|_2}. \tag{6.2}$$

Iterations

Observing the results of figure 6.3, it stands out that the convergence rate is highest for the H_1 grid and lowers as the grid becomes finer. Note that for the fixed-point iteration we expect a convergence rate of order $O(1)$, while for the H_3 grid, it takes about 7 iterations to reach order $O(1)$ improvement. Nevertheless, the convergence rates are steady and do not flatten after a certain number of iterations. For the coarse domain, the normal velocity v^{m} is constant along the latitudinal direction. This is not what is expected, as a relation between the curvature of the surface and the normal velocity should be visible, and is due to low solution resolution on the coarse mesh. For the H_2 and H_3 grids, the solution is good in the sense that the normal velocity v^{m} is inversely related to the curvature depending term $4H^2 - 2K$ form Eq. (4.10d). (See also figure (6.4)). It is therefore expected that the normal velocity is higher at the more 'flat' parts of the surface. This is reflected in the figure. The solutions on these grids do differ slightly: the normal velocity at the poles is a little higher for the H_3 grid compared to the H_2 grid.

Time steps

For the basic domain with the H_2 grid, time steps are carried out. The results can be found in figure 6.4. As time increases the surface grows and gets closer to a sphere-like domain. The normal velocity v^{m} should be inversely related to the term $4H^2 - 2K$, due to Eq. (4.10d), leading to a higher velocity at the more flat parts of the surface. This is exactly what happens in the figure.

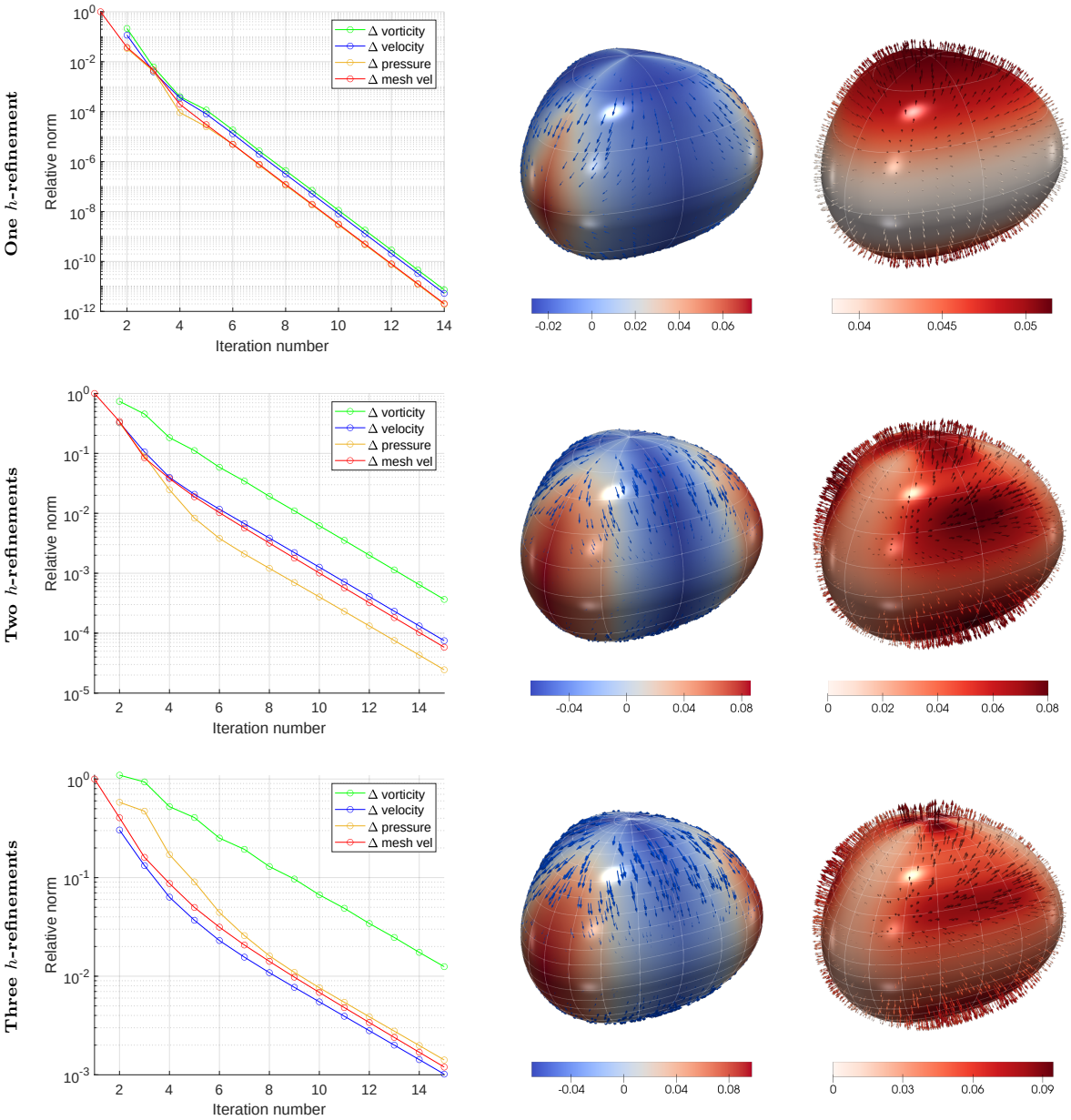
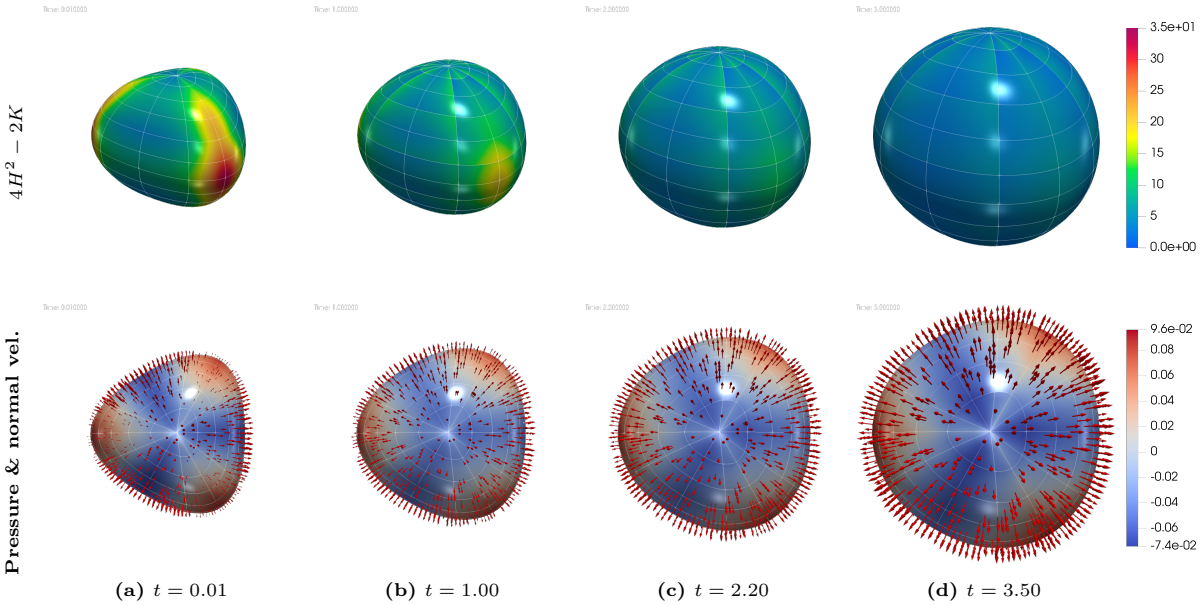
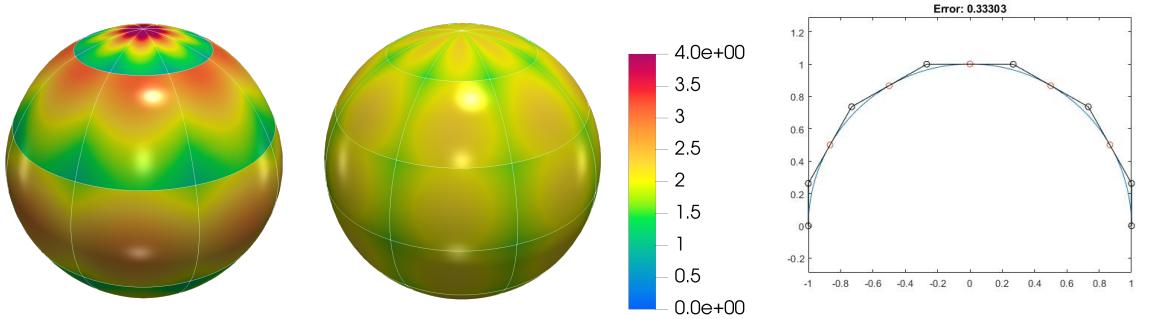


Figure 6.3: The results on the basic domain for different refinement levels. One, two or three h-refinements are performed corresponding to the first, second and third row, respectively. The left most figure of each row shows the iteration number versus the relative norms of the solution increments as described in Eq. (6.2). The other figures show the solution at the final time step. The middle figures are colored by the pressure, and the blue arrows indicate the in-plane flow. In the right figures both the surface color and the red arrows represent the out-of-plane velocity of the domain.



(a) $t = 0.01$ **(b)** $t = 1.00$ **(c)** $t = 2.20$ **(d)** $t = 3.50$

Figure 6.4: The basic domain at four different time steps. The top figures show the domain from a side angle and the surfaces are colored by the value of $4H^2 - 2K$. The bottom row shows the same domains seen from above. These figures are colored by the pressure q (minus the surface tension). As time increases the domain grows in size and slowly approaches a more sphere-like domain. The animation corresponding to this figure can be found here.



(a) Simple sphere **(b)** Best sphere **(c)** Least squares approximation of a half circle.

Figure 6.5: Two approximations of a sphere on the coarsest level. The surfaces are colored by the term $4H^2 - 2K$. In the latitudinal direction the control points are spaced uniformly on a circle. In the longitudinal direction the best sphere approximates a half circle using least squares.

6.2.2. Comparison of two sphere-like domains

The convergence rates of the solutions on the basic domain of the previous section are not perfect.Therefore, it is interesting to investigate the behaviour of other domains. For simplicity two sphere-like domains are introduced (see figure 6.5), called the *simple sphere* (8×5 elements) and the *best sphere* (8×6 elements). The surfaces of figure 6.5 are colored by the term $4H^2 - 2K$ that occurs in Eq. (4.10d). For a perfect sphere the mean curvature $H = \frac{1}{r}$ and the Gaussian curvature $K = \frac{1}{r^2}$, such that $4H^2 - 2K = 2$ for a sphere of radius 1. As it is not possible to construct a perfect sphere using polar splines, both geometries are approximations only. The control points of both domains are spaced uniformly on circles in the latitudinal direction (circling around the poles). The domains differ in the longitudinal direction (toward the poles): for the simple sphere the domain is initiated as two layers of control points that define the poles. Subsequently, new layers are created in the longitudinal direction until the right number of elements is reached. For the best sphere, least squares approximates the half circle as shown in figure 6.5c in the longitudinal direction.

The simple sphere
The results on the simple sphere are presented in figure 6.6. H_0, H_1, H_2 refer to the coarse mesh, and the h-refined meshes, respectively. The solutions behave analogous to the solutions on the basic domain. The convergence rate of the coarse geometry is close to order $O(1)$, as expected. As the grid becomes finer, the convergence rate worsens. For the H_0 grid, the solution is constant along the latitudinal direction. Refining the mesh, leads to solutions that also vary in the latitudinal direction. This is what one expects, as the domain is an imperfect approximation of the sphere.

The best sphere
The results of the best sphere are presented in figure 6.7. It is easy to see that the iterations do not converge. Contrary, the solution blows up in a few iterations. To get a little intuition for the initial behaviour of the solution, the first three iterations of the once-refined grid are presented in figure 6.8. The normal velocity of the first iteration is something that one would expect. Similar to the simple sphere the normal velocity is constant along the latitudinal direction, and points outwards. However, the range of the normal velocity is more narrow. In the second iteration, the in-plane flow and the pressure of the previous iteration have a high impact on the normal velocity. Subsequently, the in-plane flow reverses and is much stronger. Note that the solution as a whole is more or less flipped in sign. The solution at the third iteration is comparable in sign to the solution of the first iteration. The scales are very different, though.

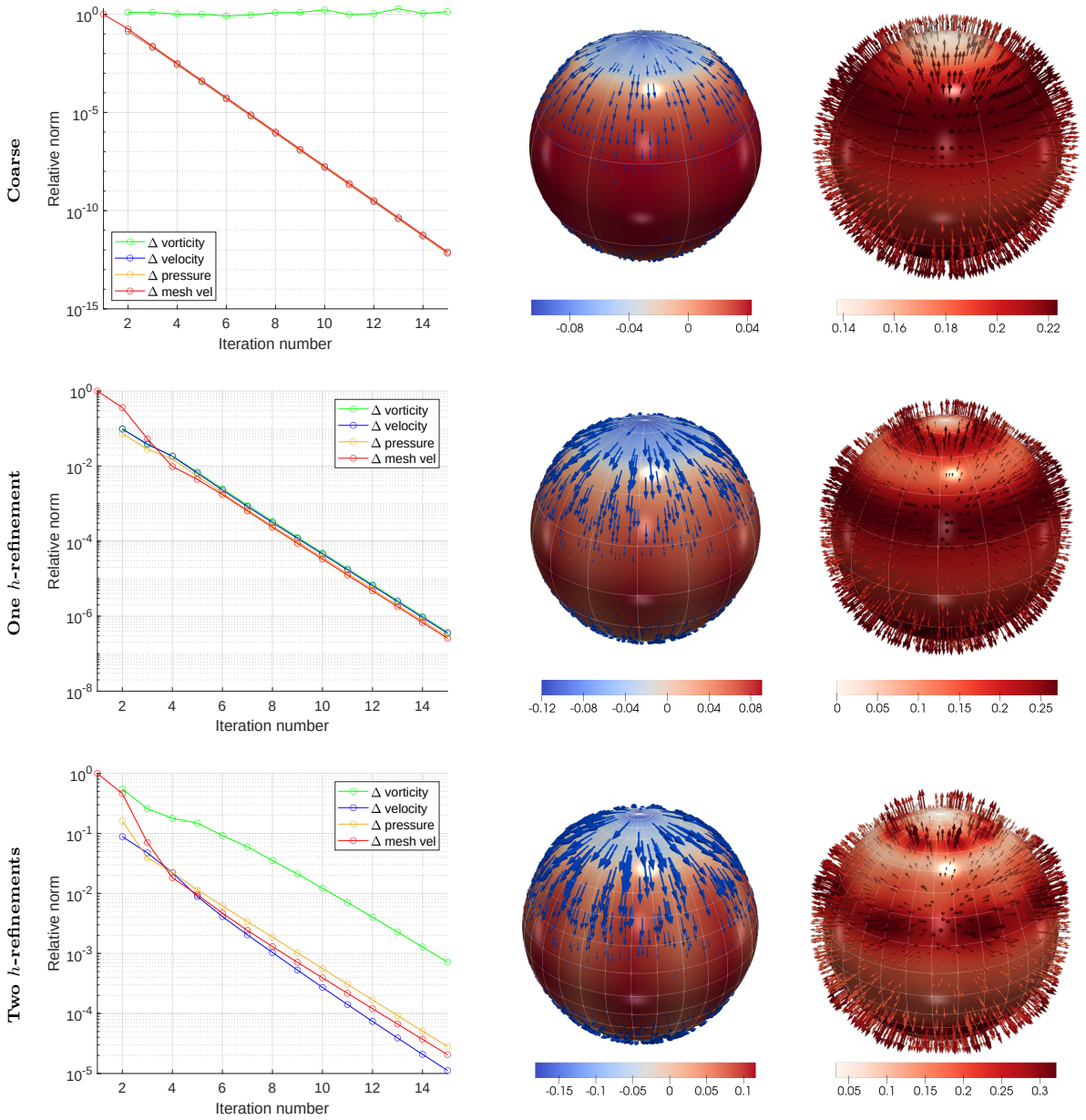
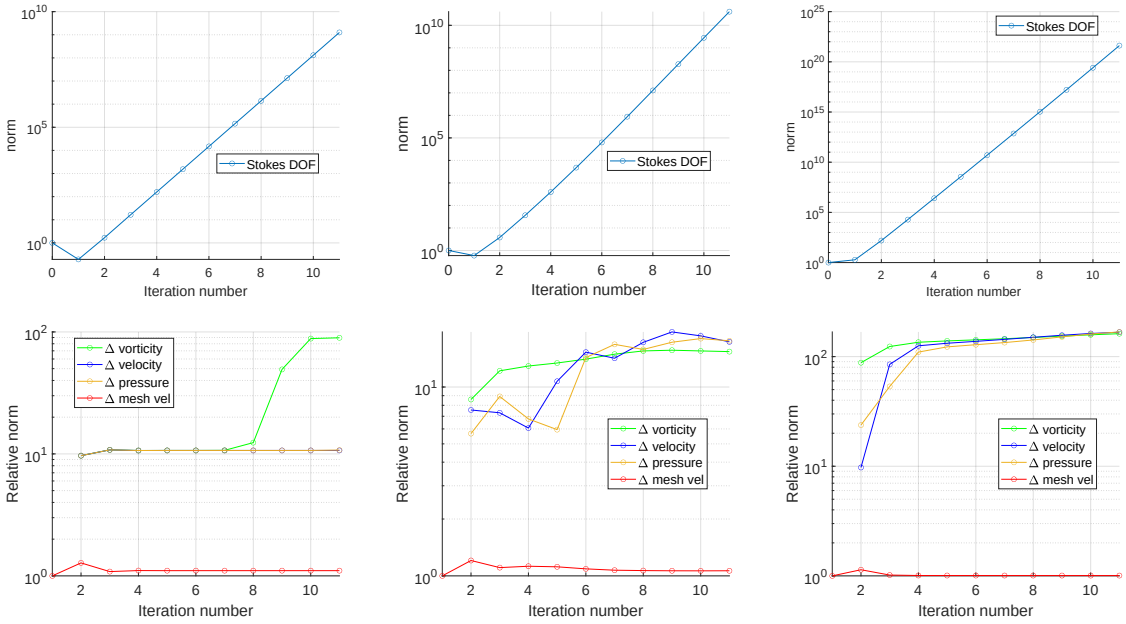
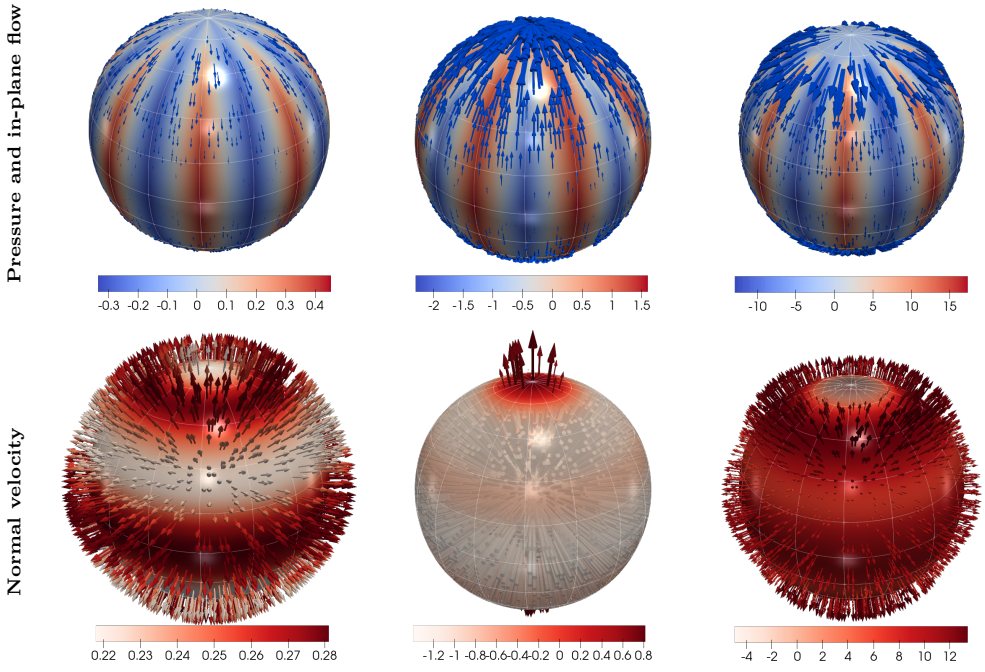


Figure 6.6: The results on the simple sphere. The coarse mesh, the mesh after one or two h-refinements, correspond to the first, second and third row of figures. The left most figure of each row shows the iteration number versus the relative norms of the solution increments as described in Eq. (6.2). The other figures show the solution at the final time step. The middle figures are colored by the pressure, and the blue arrows indicate the in-plane flow. In the right figures both the surface color and the red arrows represent the out-of-plane velocity of the domain.



(a) Coarse domain **(b)** One h-refinement **(c)** Two h-refinements

Figure 6.7: Error plots for the best sphere with $\Delta t = 0$, pratically leading to no geometry update. The top row corresponds to the absolute values of the Stokes DOFs (ϕ from Eq. (5.7)). The bottom row presents the relative solutions increments as seen before.



(a) Iteration 1 **(b)** Iteration 2 **(c)** Iteration 3

Figure 6.8: Three subsequent iterations of the best sphere after one h-refinement. Δt equals 0. The top surfaces are colored by the pressure, and the blue arrow indicate the in-plane flow (not scaled to each other). In the bottom figures, both the surface color and the arrows indicate the normal velocity.

7 | Conclusion

In this thesis a computational method that solves the fluid film equation has been proposed. This has its applications in, for example, the behavior of biological membranes such as lipid bilayers. The framework of discrete exterior calculus was utilized to formulate a consistent numerical method, where the unknowns are expressed as differential forms. Specifically, the properties of differential forms are applied to construct divergence conforming spline spaces for incompressible flows on curved surfaces. The big advantage of this setup is, that one does not have to resort to penalty methods or Lagrange multipliers to enforce tangential surface flow or mass conservation. Additionally, the concept of isogeometric analysis was used to construct C^1 continuous geometries. This smoothness restriction is a necessity for the existence of the weak formulation corresponding to the problem. A specific type of B-splines, called polar splines, have been adopted to construct C^1 smooth genus 0 surfaces with two collapsed singular points at both poles. Starting from the strong formulation, the weak formulation has been derived and has been discretized in space using the polar spline spaces. Time-discretization was done using the backwards (or implicit) Euler method, resulting in a non-linear system of equations. Using the iterative fixed-point method the system was solved for different time steps.

It can be concluded that the solutions for the basic domain and the simple sphere behave well. As expected, for a uniform pressure difference, the domains grow in size. The deformation is highest at relatively flat (low curvature) locations and the in-plane fluid supports the deformation. The convergence rates of these solutions vary from good (order $O(1)$) to poor (7 iterations for order $O(1)$ improvement) and seem to be related to the fineness of the grid. Interestingly, for the most uniform geometry thinkable, the best sphere, the iterations are not stable. Yet, it is unclear where this instability comes from. As the other sphere-like domain does behave well, it seems that the instability is parametrization dependent. This indicates that there might be a bug in the implementation. Assuming that the original Stokes implementation is correct, the bug must be in the implementation of the shape equation. Since the inverse relation between $4H^2 - 2K$ and the normal velocity v^{m} can be confirmed from the results, the problem could possibly be laying with the implementation vectors $[\boldsymbol{h}^{\boldsymbol{u}}_{n+1}]_i$ and $[\boldsymbol{h}^{q}_{n+1}]_i$ of Eq. (5.11). Note that from literature it is known that the fluid film is unstable due to lack of bending terms. Although this might influence the results of this research, it can not explain the parametrization dependent solutions and diverging results.

Returning to the research questions, and sub-questions, that were stated in the introduction of this report, it can be concluded that all the steps of formulating and implementing a divergence conforming method to solve the fluid film equations have been carried out. Time integration is performed using the fixed-point method. Issues are encountered concerning low convergence rates and non-converging solutions for some geometries. However, time stepping has successfully been carried out for certain geometries.

7.1. Recommendations for future work

In this sections follow recommendations for future work. A distinction is made between solving the possible implementation error and extending the present research.

7.1.1. Implementation error

As is concluded above, there is probably an implementation error in the Matlab code corresponding to this research. Unfortunately, the problem has not been solved, and due to lack of time efforts had to be put on hold. The following is recommended:

- The Stokes implementation of [41] has been adopted in this research. Assuming that this part of the implementation does not contain any errors, the current bug must be in the shape equation. Since the relation between the shape of the domain and the normal deformation is in order, matrix $[\boldsymbol{M}^{\zeta_0' v^{\mathrm{m}}}_{n+1}]_i$ of Eq. (5.11) should be fine. It is recommended to focus on the implementation of the vectors $[\boldsymbol{h}^{\boldsymbol{u}}_{n+1}]_i$ and $[\boldsymbol{h}^{q}_{n+1}]_i$ of this same equation.

- The results of this report show that the solutions on two comparable approximations of a unit sphere behave very differently. It is interesting to investigate the cause for this deviant behaviour. Experimenting with geometries that lie in-between both sphere-like shapes might lead to new insights. For example constructing a simple sphere and a best sphere using the same number of elements in both directions gives two sets of control points. Taking the the mean position of each matching control point pair creates such an in-between surface. One can also mutate both sphere-like geometries.
- As presented in algorithm 1, the equations are solved in two alternating steps. Solving the equations at once, might lead to new insights.
- A last extension might be to perform time steps for the simple sphere and observe whether the iterations will become instable after a while or not.

7.1.2. Extensions

The research can be extended in different ways. Based on the results and the literature the following recommendations for future work are given:

- The fluid film equation is known to be unstable due to the absence of a stabilising term. Incorporating bending into the equations is a possible way to overcome this instability. For example, combining the fluid film equation with Willmore flow, which minimizes the bending energy of a surface, could be promising. Computational methods that combine Willmore flow with the (Navier-)Stokes equations are found in Refs. [13, 55, 59, 121]. Other examples of Willmore flow are found in Refs. [82, 109, 110, 112].
- In this research the pressure difference b^{n} is set to a constant value. However, it may also depend on the interior volume enclosed by the surface, such that the the pressure difference becomes smaller when the volume grows. As is suggested in Ref. [82], Willmore energy can be incorporated into b^{n} by including a dependency on the domain shape.
- A different way to extend the model is by incorporating the 3D bulk fluid flow into the model. The bulk flow is coupled to the in-plane surface flow and makes the model more realistic in terms of, for example, cell interaction with the surrounding [83].
- Another improvement might be to elevate the degree of the basis functions. It can be observed what effects are of higher order of smoothness of the surface and the solution.
- To deal with the non-linearity's in the system of equations, the fixed-point iterative method has been used. It is an easy-to-implement method, but generally it converges very slowly. Setting up a Newton-Raphson iteration would lead to faster convergence.
- In the results section of this report, geometries keep on growing due to a pressure difference. The incompressibility constraint is still attained since it also depends on the normal velocity. Adding a surface area constraint [84] would lead to total area preserving solutions, applicable to in-extensible membranes.
- Concerning biological applications, one could compare mathematical results with biological observations for validation. For example, the shape of the red blood cell can be compared with real world observations as is done in [13].

References

1. Pasenkiewicz-Gierula, M., Murzyn, K., Róg, T. & Czaplewski, C. Molecular dynamics simulation studies of lipid bilayer systems. *Acta Biochimica Polonica* **47,** 601–611 (Sept. 30, 2000).
2. Evans, E. A. Bending Resistance and Chemically Induced Moments in Membrane Bilayers. *Biophysical Journal* **14,** 923–931 (Dec. 1, 1974).
3. Hurley, J. H., Boura, E., Carlson, L.-A. & Różycki, B. Membrane Budding. *Cell* **143,** 875–887 (Dec. 10, 2010).
4. Higgins, M. K. & McMahon, H. T. Snap-shots of clathrin-mediated endocytosis. *Trends in Biochemical Sciences* **27,** 257–263 (May 1, 2002).
5. Matlin, K. S., Reggio, H., Helenius, A. & Simons, K. Infectious entry pathway of influenza virus in a canine kidney cell line. *Journal of Cell Biology* **91,** 601–613 (Feb. 22, 1981).
6. McMahon, H. T. & Boucrot, E. Molecular mechanism and physiological functions of clathrin-mediated endocytosis. *Nat Rev Mol Cell Biol* **12,** 517–533 (Aug. 2011).
7. Schroeder, T. E. THE CONTRACTILE RING : II. Determining its Brief Existence, Volumetric Changes, and Vital Role in Cleaving Arbacia Eggs. *Journal of Cell Biology* **53,** 419–434 (May 1, 1972).
8. Blazek, A. D., Paleo, B. J. & Weisleder, N. Plasma Membrane Repair: A Central Process for Maintaining Cellular Homeostasis. *Physiology* **30,** 438–448 (Nov. 1, 2015).
9. Boye, T. L. *et al.* Annexin A4 and A6 induce membrane curvature and constriction during cell membrane repair. *Nat Commun* **8,** 1623 (Nov. 20, 2017).
10. Das, S. & Du, Q. Adhesion of vesicles to curved substrates. *Phys. Rev. E* **77,** 011907 (Jan. 15, 2008).
11. Alberts, B. *et al. Molecular biology of the cell* Sixth edition. 1 online resource : illustrations (chiefly color), color portrait vols. (Garland Science, Taylor and Francis Group, New York, NY, 2015).
12. Peng, Z., Asaro, R. J. & Zhu, Q. Multiscale simulation of erythrocyte membranes. *Phys. Rev. E* **81,** 031904 (Mar. 4, 2010).
13. Klöppel, T. & Wall, W. A. A novel two-layer, coupled finite element approach for modeling the nonlinear elastic and viscoelastic behavior of human erythrocytes. *Biomech Model Mechanobiol* **10,** 445–459 (July 1, 2011).
14. Barrett, J. W., Garcke, H. & Nürnberg, R. A stable numerical method for the dynamics of fluidic membranes. *Numer. Math.* **134,** 783–822 (Dec. 1, 2016).
15. Losserand, S., Coupier, G. & Podgorski, T. Migration velocity of red blood cells in microchannels. *Microvascular Research* **124,** 30–36 (July 1, 2019).
16. Zhao, Y. *et al.* Exo70 Generates Membrane Curvature for Morphogenesis and Cell Migration. *Developmental Cell* **26,** 266–278 (Aug. 12, 2013).
17. Arroyo, M., Heltai, L., Millán, D. & DeSimone, A. Reverse engineering the euglenoid movement. *Proceedings of the National Academy of Sciences* **109,** 17874–17879 (Oct. 30, 2012).
18. McMahon, H. T. & Gallop, J. L. Membrane curvature and mechanisms of dynamic cell membrane remodelling. *Nature* **438,** 590–596 (Dec. 2005).
19. Cotter, C. J. & McRae, A. T. T. Compatible finite element methods for numerical weather prediction. *arXiv:1401.0616 [physics]* (Jan. 3, 2014).
20. Yang, S.-T., Kiessling, V. & Tamm, L. K. Line tension at lipid phase boundaries as driving force for HIV fusion peptide-mediated fusion. *Nat Commun* **7,** 11401 (Apr. 26, 2016).

21. Tasso, I. V. & Buscaglia, G. C. A finite element method for viscous membranes. *Computer Methods in Applied Mechanics and Engineering* **255,** 226–237 (Mar. 1, 2013).
22. Scriven, L. E. Dynamics of a fluid interface Equation of motion for Newtonian surface fluids. *Chemical Engineering Science* **12,** 98–108 (May 1, 1960).
23. Stone, H. A., Koehler, S. A., Hilgenfeldt, S. & Durand, M. Perspectives on foam drainage and the influence of interfacial rheology. *J. Phys.: Condens. Matter* **15,** S283–S290 (Dec. 2002).
24. Prud'homme, R. K. & Khan, S. A. *Foams: theory, measurements, and applications Surfactant science series* **v. 57.** viii, 596 (Marcel Dekker, Inc., New York, 1996).
25. Saye, R. I. & Sethian, J. A. Multiscale Modeling of Membrane Rearrangement, Drainage, and Rupture in Evolving Foams. *Science* **340,** 720–724 (May 10, 2013).
26. Binks, B. P. & Horozov, T. S. *Colloidal Particles at Liquid Interfaces* 505 pp. (Cambridge University Press, Aug. 17, 2006).
27. Stone, H. A. Interfaces: in fluid mechanics and across disciplines. *Journal of Fluid Mechanics* **645,** 1–25 (Feb. 2010).
28. Pozrikidis, C. A finite-element method for interfacial surfactant transport, with application to the flow-induced deformation of a viscous drop. *Journal of Engineering Mathematics* **49,** 163–180 (June 1, 2004).
29. Pozrikidis, C. Interfacial Dynamics for Stokes Flow. *Journal of Computational Physics* **169,** 250–301 (May 20, 2001).
30. Fries, T.-P. Higher-order surface FEM for incompressible Navier-Stokes flows on manifolds. *International Journal for Numerical Methods in Fluids* **88,** 55–78 (2018).
31. Fonn, E., van Brummelen, H., Kvamsdal, T. & Rasheed, A. Fast divergence-conforming reduced basis methods for steady Navier–Stokes flow. *Computer Methods in Applied Mechanics and Engineering* **346,** 486–512 (Apr. 2019).
32. Zienkiewicz, O. C., Taylor, R. L. & Zhu, J. Z. *The Finite Element Method: Its Basis and Fundamentals* 753 pp. (Elsevier, May 26, 2005).
33. Perot, J. B. Discrete Conservation Properties of Unstructured Mesh Schemes. *Annu. Rev. Fluid Mech.* **43,** 299–318 (Jan. 21, 2011).
34. Hiemstra, R. R., Toshniwal, D., Huijsmans, R. H. M. & Gerritsma, M. I. High order geometric methods with exact conservation properties. *Journal of Computational Physics. Physics-compatible numerical methods* **257,** 1444–1471 (Jan. 15, 2014).
35. Desbrun, M., Hirani, A. N., Leok, M. & Marsden, J. E. Discrete Exterior Calculus. *arXiv:math/0508341* (Aug. 18, 2005).
36. Desbrun, M., Kanso, E. & Tong, Y. in *Discrete Differential Geometry* (eds Bobenko, A. I., Sullivan, J. M., Schröder, P. & Ziegler, G. M.) 287–324 (Birkhäuser, Basel, 2008).
37. Arnold, D. N., Falk, R. S. & Winther, R. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica* **15,** 1–155 (May 2006).
38. Arnold, D., Falk, R. & Winther, R. Finite element exterior calculus: From hodge theory to numerical stability. *Bulletin of the American Mathematical Society* **47,** 281–354 (2010).
39. Arroyo, M. & DeSimone, A. Relaxation dynamics of fluid membranes. *Phys. Rev. E* **79,** 031915 (Mar. 24, 2009).
40. Nitschke, I., Reuther, S. & Voigt, A. in *Transport Processes at Fluidic Interfaces* (eds Bothe, D. & Reusken, A.) 177–197 (Springer International Publishing, Cham, 2017).
41. Toshniwal, D. & Hughes, T. J. R. Isogeometric discrete differential forms: Non-uniform degrees, Bézier extraction, polar splines and flows on surfaces. *Computer Methods in Applied Mechanics and Engineering* **376,** 113576 (Apr. 1, 2021).
42. Kiendl, J., Bletzinger, K. .-., Linhard, J. & Wüchner, R. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering* **198,** 3902–3914 (Nov. 1, 2009).

43. Kiendl, J., Hsu, M.-C., Wu, M. C. H. & Reali, A. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering* **291,** 280–303 (July 1, 2015).
44. Sauer, R. A., Duong, T. X. & Corbett, C. J. A computational formulation for constrained solid and liquid membranes considering isogeometric finite elements. *Computer Methods in Applied Mechanics and Engineering* **271,** 48–68 (Apr. 1, 2014).
45. Bischoff, M., Bletzinger, K.-U., Wall, W. A. & Ramm, E. in *Encyclopedia of Computational Mechanics* (American Cancer Society, 2004).
46. Hughes, T. J. R., Cottrell, J. A. & Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* **194,** 4135–4195 (Oct. 1, 2005).
47. Cottrell, J. A., Hughes, T. J. R. & Bazilevs, Y. *Isogeometric Analysis: Toward Integration of CAD and FEA* 356 pp. (John Wiley & Sons, Aug. 11, 2009).
48. Akkerman, I., Bazilevs, Y., Calo, V. M., Hughes, T. J. R. & Hulshoff, S. The role of continuity in residual-based variational multiscale modeling of turbulence. *Comput Mech* **41,** 371–378 (Dec. 3, 2007).
49. Morganti, S. *et al.* Patient-specific isogeometric structural analysis of aortic valve closure. *Computer Methods in Applied Mechanics and Engineering. Isogeometric Analysis Special Issue* **284,** 508–520 (Feb. 1, 2015).
50. Lipton, S., Evans, J. A., Bazilevs, Y., Elguedj, T. & Hughes, T. J. R. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering. Computational Geometry and Analysis* **199,** 357–373 (Jan. 1, 2010).
51. Speleers, H. & Toshniwal, D. A General Class of C1 Smooth Rational Splines: Application to Construction of Exact Ellipses and Ellipsoids. *Computer-Aided Design* **132,** 102982 (Mar. 1, 2021).
52. Sahu, A., Omar, Y. A. D., Sauer, R. A. & Mandadapu, K. K. Arbitrary Lagrangian–Eulerian finite element method for curved and deforming surfaces: I. General theory and application to fluid interfaces. *Journal of Computational Physics* **407,** 109253 (Apr. 15, 2020).
53. Verkleij, A. J. *et al.* The asymmetric distribution of phospholipids in the human red cell membrane. A combined study using phospholipases and freeze-etch electron microscopy. *Biochimica et Biophysica Acta (BBA) - Biomembranes* **323,** 178–193 (Oct. 11, 1973).
54. Evans, E. A. *Mechanics and Thermodynamics of Biomembranes* 263 pp. (CRC Press, Jan. 18, 2018).
55. Mahapatra, A., Saintillan, D. & Rangamani, P. Transport Phenomena in Fluid Films with Curvature Elasticity. *J. Fluid Mech.* **905,** A8 (Dec. 25, 2020).
56. Aris, R. *Vectors, tensors, and the basic equations of fluid mechanics* Dover ed. 1 online resource (xiv, 286 pages) : illustrations. Vols. (Dover Publications, New York, 1989).
57. Rahimi, M. & Arroyo, M. Shape dynamics, lipid hydrodynamics, and the complex viscoelasticity of bilayer membranes. *Phys. Rev. E* **86,** 011932 (July 31, 2012).
58. Yavari, A., Ozakin, A. & Sadik, S. Nonlinear Elasticity in a Deforming Ambient Space. *J Nonlinear Sci* **26,** 1651–1692 (Dec. 1, 2016).
59. Rangamani, P., Agrawal, A., Mandadapu, K. K., Oster, G. & Steigmann, D. J. Interaction between surface shape and intra-surface viscous flow on lipid membranes. *Biomech Model Mechanobiol* **12,** 833–845 (Aug. 1, 2013).
60. Gross, S. & Reusken, A. *Numerical Methods for Two-phase Incompressible Flows* 487 pp. (Springer Science & Business Media, Apr. 26, 2011).
61. Helfrich, W. Elastic Properties of Lipid Bilayers: Theory and Possible Experiments. *Zeitschrift für Naturforschung C* **28,** 693–703 (Dec. 1, 1973).
62. Canham, P. B. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *Journal of Theoretical Biology* **26,** 61–81 (Jan. 1, 1970).

63. Zhong-can, O.-Y. & Helfrich, W. Instability and Deformation of a Spherical Vesicle by Pressure. *Phys. Rev. Lett.* **59,** 2486–2488 (Nov. 23, 1987).

64. Zhong-can, O.-Y. & Helfrich, W. Bending energy of vesicle membranes: General expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders. *Phys. Rev. A* **39,** 5280–5288 (May 1, 1989).

65. Seifert, U. & Lipowsky, R. in *Handbook of Biological Physics* 403–463 (Elsevier, 1995).

66. Seifert, U., Berndl, K. & Lipowsky, R. Shape transformations of vesicles: Phase diagram for spontaneous- curvature and bilayer-coupling models. *Phys. Rev. A* **44,** 1182–1202 (July 1, 1991).

67. Steigmann, D. Fluid films with curvature elasticity. *Archive for Rational Mechanics and Analysis* **150,** 127–152 (Dec. 1999).

68. Capovilla, R. & Guven, J. Stresses in lipid membranes. *J. Phys. A: Math. Gen.* **35,** 6233–6247 (Aug. 2, 2002).

69. Rahimi, M., DeSimone, A. & Arroyo, M. Curved fluid membranes behave laterally as effective viscoelastic media. *Soft Matter* **9,** 11033–11045 (Nov. 6, 2013).

70. Guckenberger, A. & Gekle, S. Theory and algorithms to compute Helfrich bending forces: a review. *J. Phys.: Condens. Matter* **29,** 203001 (Apr. 2017).

71. Nagle, J. F. & Tristram-Nagle, S. Structure of lipid bilayers. *Biochim Biophys Acta* **1469,** 159–195 (Nov. 10, 2000).

72. Kirchhoff, G. Über das Gleichgewicht und die Bewegung einer elastischen Scheibe. *Journal für die reine und angewandte Mathematik (Crelles Journal)* **1850,** 51–88 (July 1, 1850).

73. Love, A. E. H. & Darwin, G. H. XVI. The small free vibrations and deformation of a thin elastic shell. *Philosophical Transactions of the Royal Society of London. (A.)* **179,** 491–546 (Jan. 1, 1888).

74. Calladine, C. R. The Theory of Thin Shell Structures 1888–1988. *Proceedings of the Institution of Mechanical Engineers, Part A: Power and Process Engineering* **202,** 141–149 (Aug. 1, 1988).

75. Naghdi, P. M. in *Linear Theories of Elasticity and Thermoelasticity: Linear and Nonlinear Theories of Rods, Plates, and Shells* (ed Truesdell, C.) 425–640 (Springer, Berlin, Heidelberg, 1973).

76. Büchter, N., Ramm, E. & Roehl, D. Three-dimensional extension of non-linear shell formulation based on the enhanced assumed strain concept. *International Journal for Numerical Methods in Engineering* **37,** 2551–2568 (1994).

77. Simo, J. C., Rifai, M. S. & Fox, D. D. On a stress resultant geometrically exact shell model. Part IV: Variable thickness shells with through-the-thickness stretching. *Computer Methods in Applied Mechanics and Engineering* **81,** 91–126 (July 1, 1990).

78. Leonetti, L., Liguori, F., Magisano, D. & Garcea, G. An efficient isogeometric solid-shell formulation for geometrically nonlinear analysis of elastic shells. *Computer Methods in Applied Mechanics and Engineering* **331,** 159–183 (Apr. 1, 2018).

79. Seifert, U. Configurations of fluid membranes and vesicles. *Advances in Physics* **46,** 13–137 (Feb. 1, 1997).

80. Leitenberger, S. M., Reister-Gottfried, E. & Seifert, U. Curvature Coupling Dependence of Membrane Protein Diffusion Coefficients. *Langmuir* **24,** 1254–1261 (Feb. 1, 2008).

81. Agrawal, A. & Steigmann, D. J. A model for surface diffusion of trans-membrane proteins on lipid bilayers. *Z. Angew. Math. Phys.* **62,** 549–563 (June 2011).

82. Jankuhn, T., Olshanskii, M. A. & Reusken, A. Incompressible fluid problems on embedded surfaces: Modeling and variational formulations. *Interfaces Free Bound.* **20,** 353–377 (Nov. 5, 2018).

83. Barrett, J. W., Garcke, H. & Nürnberg, R. Numerical computations of the dynamics of fluidic membranes and vesicles. *Phys. Rev. E* **92,** 052704 (Nov. 3, 2015).

84. Sauer, R. A. & Duong, T. X. On the theoretical foundations of thin solid and liquid shells. *Mathematics and Mechanics of Solids* **22,** 343–371 (Mar. 1, 2017).

85. Duong, T. X., Roohbakhshan, F. & Sauer, R. A. A new rotation-free isogeometric thin shell formulation and a corresponding continuity constraint for patch boundaries. *Computer Methods in Applied Mechanics and Engineering. Special Issue on Isogeometric Analysis: Progress and Challenges* **316,** 43–83 (Apr. 1, 2017).

86. Sauer, R. A., Duong, T. X., Mandadapu, K. K. & Steigmann, D. J. A stabilized finite element formulation for liquid shells and its application to lipid bilayers. *Journal of Computational Physics* **330,** 436–466 (Feb. 1, 2017).

87. Sahu, A., Sauer, R. A. & Mandadapu, K. K. Irreversible thermodynamics of curved lipid membranes. *Phys. Rev. E* **96,** 042409 (Oct. 17, 2017).

88. Oñate, E. & Zárate, F. Rotation-free triangular plate and shell elements. *International Journal for Numerical Methods in Engineering* **47,** 557–603 (2000).

89. Linhard, J., Wüchner, R. & Bletzinger, K.-U. "Upgrading" membranes to shells—The CEG rotation free shell element and its application in structural analysis. *Finite Elements in Analysis and Design* **44,** 63–74 (Dec. 1, 2007).

90. Wells, G. N., Kuhl, E. & Garikipati, K. A discontinuous Galerkin method for the Cahn–Hilliard equation. *Journal of Computational Physics* **218,** 860–877 (Nov. 1, 2006).

91. Ma, L. & Klug, W. S. Viscous regularization and r-adaptive remeshing for finite element analysis of lipid membrane mechanics. *Journal of Computational Physics* **227,** 5816–5835 (May 10, 2008).

92. Feng, F. & Klug, W. S. Finite element modeling of lipid bilayer membranes. *Journal of Computational Physics* **220,** 394–408 (Dec. 20, 2006).

93. Ambati, M., Kiendl, J. & De Lorenzis, L. Isogeometric Kirchhoff–Love shell formulation for elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering* **340,** 320–339 (Oct. 1, 2018).

94. Goyal, A., Dörfel, M. R., Simeon, B. & Vuong, A.-V. Isogeometric shell discretizations for flexible multibody dynamics. *Multibody Syst Dyn* **30,** 139–151 (Aug. 1, 2013).

95. Verhelst, H. Modelling Wrinkling Behaviour of Large Floating Thin Offshore Structures: An application of Isogeometric Structural Analysis for Post-Buckling Analyses (2019).

96. Auddya, D. *et al.* Biomembranes undergo complex, non-axisymmetric deformations governed by Kirchhoff-Love kinematics and revealed by a three dimensional computational framework. *arXiv:2101.11929 [cond-mat, q-bio]* (Jan. 28, 2021).

97. Zimmermann, C. *et al.* An isogeometric finite element formulation for phase transitions on deforming surfaces. *Computer Methods in Applied Mechanics and Engineering* **351,** 441–477 (July 1, 2019).

98. Benson, D. J., Bazilevs, Y., Hsu, M. C. & Hughes, T. J. R. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering. Computational Geometry and Analysis* **199,** 276–289 (Jan. 1, 2010).

99. Dornisch, W., Klinkel, S. & Simeon, B. Isogeometric Reissner–Mindlin shell analysis with exactly calculated director vectors. *Computer Methods in Applied Mechanics and Engineering* **253,** 491–504 (Jan. 1, 2013).

100. Echter, R., Oesterle, B. & Bischoff, M. A hierarchic family of isogeometric shell finite elements. *Computer Methods in Applied Mechanics and Engineering* **254,** 170–180 (Feb. 1, 2013).

101. Oden, J. T. & Sato, T. Finite strains and displacements of elastic membranes by the finite element method. *International Journal of Solids and Structures* **3,** 471–488 (July 1, 1967).

102. Fried, I. Finite element computation of large rubber membrane deformations. *International Journal for Numerical Methods in Engineering* **18,** 653–660 (1982).

103. Ibrahimbegović, A. & Gruttmann, F. A consistent finite element formulation of nonlinear membrane shell theory with particular reference to elastic rubberlike material. *Finite Elements in Analysis and Design* **13,** 75–86 (Apr. 1, 1993).

104. Stanuszek, M. FE analysis of large deformations of membranes with wrinkling. *Finite Elements in Analysis and Design* **39,** 599–618 (Apr. 1, 2003).

105. Ambroziak, A. & Kłosowski, P. A four-node 3D isoparametric membrane element. *TASK Quarterly* **10,** 35–47 (Jan. 1, 2006).

106. Elliott, C. M. & Stinner, B. Modeling and computation of two phase geometric biomembranes using surface finite elements. *Journal of Computational Physics* **229,** 6585–6612 (Sept. 1, 2010).

107. Mercker, M., Marciniak-Czochra, A., Richter, T. & Hartmann, D. Modeling and Computing of Deformation Dynamics of Inhomogeneous Biological Surfaces. *SIAM J. Appl. Math.* **73,** 1768–1792 (Jan. 1, 2013).

108. Dall'Acqua, A. & Pozzi, P. *Willmore-Helfrich L^{2}-flow of curves with natural boundary conditions* Feb. 3, 2013.

109. Bonito, A., Nochetto, R. H. & Sebastian Pauletti, M. Parametric FEM for geometric biomembranes. *Journal of Computational Physics* **229,** 3171–3188 (May 1, 2010).

110. Dziuk, G. Computational parametric Willmore flow. *Numer. Math.* **111,** 55–80 (Nov. 2008).

111. Barrett, J. W., Garcke, H. & Nürnberg, R. On the parametric finite element approximation of evolving hypersurfaces in R3. *Journal of Computational Physics* **227,** 4281–4307 (Apr. 20, 2008).

112. Bartezzaghi, A., Dedè, L. & Quarteroni, A. Isogeometric Analysis of geometric Partial Differential Equations. *Computer Methods in Applied Mechanics and Engineering* **311,** 625–647 (Nov. 1, 2016).

113. Rangarajan, R. & Gao, H. A finite element method to compute three-dimensional equilibrium configurations of fluid membranes: Optimal parameterization, variational formulation and applications. *Journal of Computational Physics* **297,** 266–294 (Sept. 15, 2015).

114. Omar, Y. A. D., Sahu, A., Sauer, R. A. & Mandadapu, K. K. Nonaxisymmetric Shapes of Biological Membranes from Locally Induced Curvature. *Biophysical Journal* **119,** 1065–1077 (Sept. 15, 2020).

115. Hassinger, J. E., Oster, G., Drubin, D. G. & Rangamani, P. Design principles for robust vesiculation in clathrin-mediated endocytosis. *PNAS* **114,** E1118–E1127 (Feb. 14, 2017).

116. Reuther, S. & Voigt, A. Solving the incompressible surface Navier-Stokes equation by surface finite elements. *Physics of Fluids* **30** (2018).

117. Olshanskii, M. A., Quaini, A., Reusken, A. & Yushutin, V. A Finite Element Method for the Surface Stokes Problem. *SIAM J. Sci. Comput.* **40,** A2492–A2518 (Jan. 1, 2018).

118. Gross, B. J. & Atzberger, P. J. Hydrodynamic flows on curved surfaces: Spectral numerical methods for radial manifold shapes. *Journal of Computational Physics* **371,** 663–689 (Oct. 15, 2018).

119. Sigurdsson, J. K. & Atzberger, P. J. Hydrodynamic coupling of particle inclusions embedded in curved lipid bilayer membranes. *Soft Matter* **12,** 6685–6707 (Aug. 10, 2016).

120. Reuther, S., Nitschke, I. & Voigt, A. A numerical approach for fluid deformable surfaces. *Journal of Fluid Mechanics* **900** (Oct. 2020).

121. Mietke, A., Jülicher, F. & Sbalzarini, I. F. Self-organized shape dynamics of active surfaces. *Proc Natl Acad Sci USA* **116,** 29–34 (Jan. 2, 2019).

122. Barrett, J., Garcke, H. & Nürnberg, R. Finite element approximation for the dynamics of asymmetric fluidic biomembranes. *Math. Comp.* **86,** 1037–1069 (May 2017).

123. Brezzi, F. & Bathe, K.-J. A discourse on the stability conditions for mixed finite element formulations. *Computer Methods in Applied Mechanics and Engineering. Proceedings of the Workshop on Reliability in Computational Mechanics* **82,** 27–57 (Sept. 1, 1990).

124. Piegl, L. & Tiller, W. *The NURBS Book* 2nd ed. (Springer-Verlag, Berlin Heidelberg, 1997).

125. Scott, M. A., Borden, M. J., Verhoosel, C. V., Sederberg, T. W. & Hughes, T. J. R. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering* **88,** 126–156 (2011).

126. Borden, M. J., Scott, M. A., Evans, J. A. & Hughes, T. J. R. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering* **87,** 15–47 (2011).

127. Toshniwal, D., Speleers, H., Hiemstra, R. R. & Hughes, T. J. R. Multi-degree smooth polar splines: A framework for geometric modeling and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering. Special Issue on Isogeometric Analysis: Progress and Challenges* **316,** 1005–1061 (Apr. 1, 2017).

128. Ciarlet, P. G. *An introduction to differential geometry with applications to elasticity* 209 pp. (Springer, Dordrecht, 2005).

129. Carroll, S. M. Lecture Notes on General Relativity. *arXiv:gr-qc/9712019* (Dec. 2, 1997).

130. Deserno, M. Fluid lipid membranes: From differential geometry to curvature stresses. *Chemistry and Physics of Lipids. Membrane mechanochemistry: From the molecular to the cellular scale* **185,** 11–45 (Jan. 1, 2015).

131. Kreeft, J. & Gerritsma, M. Mixed mimetic spectral element method for Stokes flow: A pointwise divergence-free solution. *Journal of Computational Physics* **240,** 284–309 (May 1, 2013).

132. Donea, J. & Huerta, A. *Finite Element Methods for Flow Problems* 366 pp. (John Wiley & Sons, June 2, 2003).

133. Dziuk, G. & Elliott, C. M. Finite elements on evolving surfaces. *IMA Journal of Numerical Analysis* **27,** 262–292 (Apr. 1, 2007).

A | Stokes on a static and strictly 2D domain

In this appendix the general Stokes flow on a 2D domain is described, along with its weak formulation and corresponding function spaces. The weak formulation is discretized in time, it is pulled back onto parametric space. Lastly, preliminary results for this setting are presented.

A.1. Strong formulation

Let Ω be a flat (that is strictly 2D) surface. Figure 3.5 gives an example of such a domain. We assume that Ω does not evolve in time and will therefore have zero normal velocity, i.e. $v^{\mathrm{m}} = 0$. This also means the the shape equation of Eq. (4.1c) is redundant. These assumptions will simplify the strong formulation of Eq. (4.1) to

$$-\operatorname{grad}_\Omega p - \left(\mathbf{d}^\star \,\mathbf{d}\, \boldsymbol{v}^b\right)^\sharp + 2K\boldsymbol{v} + \boldsymbol{f} = \mathbf{0} \tag{A.1a}$$

$$\operatorname{div}_\Omega \cdot \boldsymbol{v} = 0, \tag{A.1b}$$

where the superscript of $\boldsymbol{f}^{\mathrm{s}}$ is suppressed for readability.

As Ω is assumed to be flat, $K = 0$ holds and the surface gradient and divergence operators simplify to the 2D gradient and divergence operators, $\operatorname{div}_\Omega \cdot \boldsymbol{v} = v^\alpha_{,\alpha} = \nabla \cdot \boldsymbol{v}$ and $\operatorname{grad}_\Omega p = \nabla p$, respectively. The term $\left(\mathbf{d}^\star \,\mathbf{d}\, \boldsymbol{v}^b\right)^\sharp$ is a generalization of the curl-curl operator [39]. In the case of a flat 2D surface this expression is equivalent to

$$-\left(\mathbf{d}^\star \,\mathbf{d}\, \boldsymbol{v}^b\right)^\sharp = -\boldsymbol{\nabla} \times \nabla \times \boldsymbol{v} = \nabla^2 \boldsymbol{v}. \tag{A.2}$$

The last equality follows from the fact that $\nabla \cdot \boldsymbol{v} = 0$. By substituting these simplifications into Eq. (A.1), the well known 2D stokes equation is recovered

$$-\nabla^2 \boldsymbol{v} + \nabla p = \boldsymbol{f}, \qquad \text{on } \Omega \tag{A.3a}$$

$$\nabla \cdot \boldsymbol{v} = 0, \qquad \text{on } \Omega \tag{A.3b}$$

$$\boldsymbol{v} = \boldsymbol{v}_\partial, \qquad \text{on } \partial\Omega. \tag{A.3c}$$

In Eq. (A.3) the first vector equation is the momentum balance, the second equality corresponds to the mass conservation and the last vector equation is the boundary conditions of the problem. $\boldsymbol{v}$ is the velocity of the fluid while p is the pressure. Together, these form the unknowns of the problem.

The system can alternatively be described by introducing an additional variable: the vorticity $\omega := \nabla \times \boldsymbol{v}$. The strong form of the vorticity-velocity-pressure formulation of the Stokes equation is given by

$$\omega = \nabla \times \boldsymbol{v}, \qquad \text{on } \Omega \tag{A.4a}$$

$$\boldsymbol{\nabla} \times \omega + \nabla p = \boldsymbol{f}, \qquad \text{on } \Omega \tag{A.4b}$$

$$\nabla \cdot \boldsymbol{v} = 0, \qquad \text{on } \Omega \tag{A.4c}$$

$$\boldsymbol{v} = \boldsymbol{v}_\partial, \qquad \text{on } \partial\Omega, \tag{A.4d}$$

The advantage of this additional vorticity variable is a lower smoothness requirement of the velocity $\boldsymbol{v}$ in the corresponding continuous and discrete weak formulations.

A.2. Continuous weak formulation

In order to derive the continuous weak formulation of the Stokes equation, Eqs. (A.4) are multiplied with with a corresponding test function, after which the equations are integrated. The weak formulation is given by:

Find $\omega \in \mathcal{W}$, $\boldsymbol{v} \in \mathcal{V}_{\boldsymbol{v}_\partial}$ *and* $p \in \mathcal{P}$ *such that for all* $\alpha \in \mathcal{W}$, $\boldsymbol{w} \in \mathcal{V}_0$ *and* $q \in \mathcal{P}$ *the following holds:*

$$\int_\Omega \alpha\omega \, d\Omega + \int_\Omega \alpha_{,1} v_2 - \alpha_{,2} v_1 \, d\Omega = \int_{\partial\Omega} \alpha \boldsymbol{v}_\partial \cdot d\Gamma \tag{A.5a}$$

$$\int_\Omega w_1 \omega_{,2} - w_2 \omega_{,1} \, d\Omega - \int_\Omega p \nabla \cdot \boldsymbol{w} \, d\Omega = \int_\Omega \boldsymbol{w} \cdot \boldsymbol{f} \, d\Omega \tag{A.5b}$$

$$\int_\Omega q \nabla \cdot \boldsymbol{v} \, d\Omega = 0. \tag{A.5c}$$

To obtain Eq. (A.5a),

$$\int_\Omega \alpha \nabla \times \boldsymbol{v} \, d\Omega = \int_\Omega \nabla \times (\alpha \boldsymbol{v}) \, d\Omega - \int_\Omega \nabla\alpha \times \boldsymbol{v} \, d\Omega \tag{A.6}$$

and Stokes' circulation theorem are used. Eq. (A.5b) is obtained using

$$\begin{aligned} \int_\Omega \boldsymbol{w} \cdot \nabla p \, d\Omega &= -\int_\Omega p \nabla \cdot \boldsymbol{w} \, d\Omega + \int_\Omega p \boldsymbol{w} \cdot \boldsymbol{\nu} \, d\Omega \\ &= -\int_\Omega p \nabla \cdot \boldsymbol{w} \, d\Omega. \end{aligned} \tag{A.7}$$

$\boldsymbol{\nu}$ is the outward pointing normal vector, defined on the boundary $\partial\Omega$.

Minimum smoothness requirement for test and trial functions, as implied by the weak formulation, can be found in table A.1. For ω, α and p, q this is quite obvious. Recognizing that $\nabla \cdot \boldsymbol{v} = v^1_{,1} + v^2_{,2}$ indicates that of component v^α only the derivative in the x_α direction is needed.

Function	Minimum smoothness requirement	
	First coordinate	Second coordinate
ω, α	C^0	C^0
v_1, w_1	C^0	C^{-1}
v_2, w_2	C^{-1}	C^0
p, q	C^{-1}	C^{-1}

Table A.1: Smoothness requirements for the test and trial functions as implied by the occurring derivatives in the weak formulation of Eq. (A.5).

The smoothness requirements for the test and trial functions have a direct consequence for the function spaces they live in. In order to characterize these spaces, first the Sobolev space $H^k(\Omega)$ is introduced [132, 133]. Let $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2\}$ be a multi-index with $|\boldsymbol{\sigma}| = \sigma_1 + \sigma_2$. Define

$$D^{\boldsymbol{\sigma}} f = \frac{\partial^{|\boldsymbol{\sigma}|} f}{\partial (x^1)^{\sigma_1} \partial (x^2)^{\sigma_2}} \tag{A.8}$$

the Sobolev space of order k is given by

$$H^k(\Omega) = \left\{ f : D^{\boldsymbol{\sigma}} f \in L^2(\Omega), |\boldsymbol{\sigma}| \leq k \right\}. \tag{A.9}$$

The function spaces of the test and trial functions of Eq. (A.5) are given by

$$\mathcal{W} := H^1(\Omega), \tag{A.10a}$$

$$\mathcal{V} := \left\{ \boldsymbol{f} : \nabla \cdot \boldsymbol{f} \in H^0(\Omega) \right\}, \tag{A.10b}$$

$$\mathcal{P} := H^0(\Omega), \tag{A.10c}$$

and

$$\mathcal{V}_{\boldsymbol{g}} = \{ \boldsymbol{f} \in \mathcal{V} : \boldsymbol{f} \cdot \boldsymbol{\nu} = \boldsymbol{g} \cdot \boldsymbol{\nu} \text{ on } \partial\Omega \}. \tag{A.11}$$

Note that these are the minimum requirements for the function spaces. There are no restrictions with regards to higher order spaces.

A.3. Discretization of the weak formulation

In the following subsection it is described how the continuous weak formulation of the flat 2D Stokes equation as introduced in Eq. (A.5), is discretized. However, before the actual discretization can be carried out, a few things need to be introduced. First of all, notations and definitions are introduced. Next, the discrete function spaces of the parametric domain will be defined (see section A.3.2). Lastly, three functions transformations that connect the discrete parametric spaces with their geometric counterparts are given in section A.3.3. These basics are utilised to obtain the discrete weak formulation as defined on the parametric domain, and to obtain the linear system to be solved.

A.3.1. Notation

As discussed in Eq. (3.10), the bivariate basis functions $B_{ij}^{(r,s)}$ are given by the tensor product of two univariate functions of order $r, s \in \{0, 1, 2\}$. From now on, if a linear combination of basis functions is expressed, the lower indices are denoted by a single capital letter and the lower and upper limits of the sum are suppressed. For example, Eq. (3.11) is abbreviated by

$$\begin{aligned} \boldsymbol{x}(\boldsymbol{\xi}) &= \sum_{i=1}^{n_{\text{sh}}} \sum_{j=1}^{m_{\text{sh}}} B_{ij}^{(r,s)}(\boldsymbol{\xi}) \boldsymbol{R}_{ij} \\ &= \sum_{A} B_{A}^{(r,s)} \boldsymbol{R}_{A} \end{aligned} \tag{A.12}$$

Here A denotes the global index of the basis functions such that $A = n_{\text{sh}} \cdot (j-1) + i$. Note that the upper limits n_{sh} and m_{sh} are depending on both the order and the length of the corresponding knot vector. The upper limit of A can therefore be deduced from the multivariate B-spline function order and the number of elements. Throughout this section capital letters A and B will indicate the global index related to the coordinate specific coordinates i, j, k, l. The coordinate specific sub scripts of control points or control variables exhibit the same global index relation. Note that n_{sh} and m_{sh} can always be deduced using the order of the basis functions.

A.3.2. Discrete function spaces

The discrete function spaces corresponding to the parametric domain will be introduced. The to-be introduced transformations will then define the functions spaces counterparts living on the geometric domain. Define $\breve{r} := r - 1$, $\breve{s} := s - 1$, $\breve{n}_{\text{sh}} := n_{\text{sh}} - 1$ and $\breve{m}_{\text{sh}} := m_{\text{sh}} - 1$. This notation will be used to improve readability. The discrete spaces on the **parametric** domain $\hat{\Omega}$ are given by,

$$\hat{\mathcal{W}}_h = \left\{ \hat{f} : \hat{f}(\boldsymbol{\xi}) = \sum_{A} B_{A}^{(r,s)}(\boldsymbol{\xi}) f_A \right\}, \tag{A.13a}$$

$$\hat{\mathcal{V}}_h^1 = \left\{ \hat{f} : \hat{f}(\boldsymbol{\xi}) = \sum_{A} B_{A}^{(r,\breve{s})}(\boldsymbol{\xi}) f_A \right\}, \tag{A.13b}$$

$$\hat{\mathcal{V}}_h^2 = \left\{ \hat{f} : \hat{f}(\boldsymbol{\xi}) = \sum_{A} B_{A}^{(\breve{r},s)}(\boldsymbol{\xi}) f_A \right\}, \tag{A.13c}$$

$$\hat{\mathcal{P}}_h = \left\{ \hat{f} : \hat{f}(\boldsymbol{\xi}) = \sum_{A} B_{A}^{(\breve{r},\breve{s})}(\boldsymbol{\xi}) f_A \right\}, \tag{A.13d}$$

and $\hat{\mathcal{V}}_h = \hat{\mathcal{V}}_h^1 \times \hat{\mathcal{V}}_h^2$. Furthermore, let $\hat{\boldsymbol{\nu}}$ be the outward pointing normal vector of $\hat{\Omega}$. Define

$$\hat{\mathcal{V}}_{\boldsymbol{\phi},h} = \left\{ \boldsymbol{f} \in \hat{\mathcal{V}}_h : \boldsymbol{f} \cdot \hat{\boldsymbol{\nu}} = \boldsymbol{\phi} \cdot \hat{\boldsymbol{\nu}} \text{ on } \partial\hat{\Omega} \right\} \tag{A.14a}$$

The corresponding knot vectors are now introduced. Define the knot vectors in the first parametric coordinate direction as

$$\begin{aligned} \Xi^{r} &= [\xi_1^r, \ \xi_2^r, \ \xi_3^r, \ \ldots, \ \xi_{n_{\text{sh}}+r-1}^r, \ \xi_{n_{\text{sh}}+r}^r, \ \xi_{n_{\text{sh}}+r+1}^r] \\ \Xi^{\breve{r}} &= [\quad \xi_1^{\breve{r}}, \ \xi_2^{\breve{r}}, \ \ldots, \ \xi_{\breve{n}_{\text{sh}}+\breve{r}}^{\breve{r}}, \ \xi_{\breve{n}_{\text{sh}}+\breve{r}+1}^{\breve{r}} \quad] \ . \end{aligned} \tag{A.15}$$

Here $\xi_1^r = \ldots = \xi_{r+1}^r$ and $\xi_{n_{\text{sh}}}^r = \ldots = \xi_{n_{\text{sh}}+r+1}^r$ are the first and last $r+1$ repeated knots of Ξ^r, respectively, making it an open knot vector. The same holds for $\Xi^{\check{r}}$ where $\xi_1^{\check{r}} = \ldots = \xi_{\check{r}+1}^{\check{r}}$ and $\xi_{\check{n}_{\text{sh}}}^{\check{r}} = \ldots = \xi_{\check{n}_{\text{sh}}+\check{r}+1}^{\check{r}}$ are the first and last $\check{r}+1 = r$ repeated knots, respectively. Note the alignment of the knots of both knot vectors in Eq. (A.16), this indicates that the aligned knots are equal. The only difference between knot vectors Ξ^r and $\Xi^{\check{r}}$ is one extra repetition of the first and last knot of the former.

The exact analogue is defined for the second parametric coordinate direction:

$$\begin{array}{rcl} \mathcal{H}^s &=& [\eta_1^s, \quad \eta_2^s, \quad \eta_3^s, \quad \ldots, \quad \eta_{m_{\text{sh}}+s-1}^s, \quad \eta_{m_{\text{sh}}+s}^s, \quad \eta_{m_{\text{sh}}+s+1}^s] \\ \mathcal{H}^{\check{s}} &=& [\quad \eta_1^{\check{s}}, \quad \eta_2^{\check{s}}, \quad \ldots, \quad \eta_{\check{m}_{\text{sh}}+\check{s}}^{\check{s}}, \quad \eta_{\check{m}_{\text{sh}}+\check{s}+1}^{\check{s}} \quad] \quad . \end{array} \tag{A.16}$$

As above, the first, respetively, last $s+1$ knots of $\mathcal{H}^s$ and the first, respectively, last $\check{s}+1 = \check{s}$ knots of $\mathcal{H}^{\check{s}}$ are equal.

A.3.3. Transformation of discrete functions in 2D

In this section the relation between the discrete function spaces of (A.13) and their geometric counterparts is discussed. This relation is a key element in translating the weak formulation of Eq. (A.29) to an equivalent weak from, defined on the parametric domain, such that the incompressibility constrain on $\boldsymbol{v}$ is maintained. Furthermore, the function spaces of equation (A.13) are only valid divergence conforming spline spaces for certain function relations. This section gives these relations.

Let $\hat{\omega} \in \hat{\mathcal{W}}_h$, $\hat{\boldsymbol{v}} \in \hat{\mathcal{V}}_h$ and $\hat{p} \in \hat{\mathcal{P}}_h$

$$\hat{\omega}(\boldsymbol{\xi}) : \hat{\Omega} \to \mathbb{R}, \qquad \hat{\boldsymbol{v}}(\boldsymbol{\xi}) : \hat{\Omega} \to \mathbb{R}^2, \qquad \hat{p}(\boldsymbol{\xi}) : \hat{\Omega} \to \mathbb{R}. \tag{A.17}$$

The geometric forms, $\omega(\boldsymbol{x})$, $\boldsymbol{v}(\boldsymbol{x})$ and $p(\boldsymbol{x})$, corresponding to the parametric functions of Eq. (A.17), are constructed such that the following holds.

$$\begin{array}{rcrcl} \hat{\omega}(\boldsymbol{\xi}) &:=& T_0\omega(\boldsymbol{x}) &=& \omega \circ \boldsymbol{x}(\boldsymbol{\xi}) : \quad \hat{\Omega} \to \mathbb{R} \\ \hat{\boldsymbol{v}}(\boldsymbol{\xi}) &:=& T_1\boldsymbol{v}(\boldsymbol{x}) &=& \sqrt{g}J^{-1}\boldsymbol{v} \circ \boldsymbol{x}(\boldsymbol{\xi}) : \quad \hat{\Omega} \to \mathbb{R}^2, \\ \hat{p}(\boldsymbol{\xi}) &:=& T_2p(\boldsymbol{x}) &=& \sqrt{g}p \circ \boldsymbol{x}(\boldsymbol{\xi}) : \quad \hat{\Omega} \to \mathbb{R}. \end{array} \tag{A.18}$$

Here T_0, T_1 and T_2 are the transformations corresponding to the spaces $\hat{\mathcal{W}}_h$, $\hat{\mathcal{V}}_h$ and $\hat{\mathcal{P}}_h$, respectively.

The geometric counterparts of $\hat{\mathcal{W}}_h, \hat{\mathcal{V}}_h, \hat{\mathcal{P}}_h$ as introduced in Eq. (A.13), are defined by

$$\mathcal{W}_h := \left\{ f : T_0 f \in \hat{\mathcal{W}}_h \right\}, \tag{A.19a}$$

$$\mathcal{V}_h := \left\{ \boldsymbol{f} : T_1 \boldsymbol{f} \in \hat{\mathcal{V}}_h \right\}, \tag{A.19b}$$

$$\mathcal{P}_h := \left\{ f : T_2 f \in \hat{\mathcal{P}}_h \right\} \tag{A.19c}$$

and

$$\mathcal{V}_{\boldsymbol{g},h} = \{ \boldsymbol{f} \in \mathcal{V}_h : \boldsymbol{f} \cdot \boldsymbol{\nu} = \boldsymbol{g} \cdot \boldsymbol{\nu} \text{ on } \Omega \} \tag{A.20}$$

The transformations of Eq. (A.18) can be inverted in order to obtain the relations

$$\begin{array}{rcrcl} \mathcal{W}_h \ni \omega(\boldsymbol{x}) &=& T_0^{-1}\hat{\omega}(\boldsymbol{\xi}) &=& \hat{\omega} \circ \boldsymbol{\xi}(\boldsymbol{x}) : \quad \Omega \to \mathbb{R} \\ \mathcal{V}_h \ni \boldsymbol{v}(\boldsymbol{x}) &=& T_1^{-1}\hat{\boldsymbol{v}}(\boldsymbol{\xi}) &=& \frac{1}{\sqrt{g}}J\hat{\boldsymbol{v}} \circ \boldsymbol{\xi}(\boldsymbol{x}) : \quad \Omega \to \mathbb{R}^2 \\ \mathcal{P}_h \ni p(\boldsymbol{x}) &=& T_2^{-1}\hat{p}(\boldsymbol{\xi}) &=& \frac{1}{\sqrt{g}}\hat{p} \circ \boldsymbol{\xi}(\boldsymbol{x}) : \quad \Omega \to \mathbb{R}, \end{array} \tag{A.21}$$

and note that

$$\sqrt{g}J^{-1} = \sqrt{g}\frac{1}{\sqrt{g}}J^{\text{adj}} = J^{\text{adj}} = \begin{bmatrix} x^2_{,2}(\boldsymbol{\xi}) & -x^1_{,2}(\boldsymbol{\xi}) \\ -x^2_{,1}(\boldsymbol{\xi}) & x^1_{,1}(\boldsymbol{\xi}) \end{bmatrix} \tag{A.22a}$$

$$\frac{1}{\sqrt{g}}J = \left(J^{\text{adj}}\right)^{-1} = \begin{bmatrix} \xi^2_{,2}(\boldsymbol{x}) & -\xi^1_{,2}(\boldsymbol{x}) \\ -\xi^2_{,1}(\boldsymbol{x}) & \xi^1_{,1}(\boldsymbol{x}) \end{bmatrix} \tag{A.22b}$$

for $J^{\text{adj}} = \text{adjoint}(J(\boldsymbol{\xi}))$. Note that the parametric and the geometric functions can be evaluated on both the parametric and the geometric via the mapping. For readability, the function argument will from now on be omitted where possible.

The following holds for a function transforming according to T_0.

$$\frac{\partial}{\partial x^\alpha}\omega = \frac{\partial \xi^\beta}{\partial x^\alpha}\frac{\partial}{\partial \xi^\beta}\hat{\omega} = \xi^\beta_{,\alpha}\hat{\omega}_{,\beta} = \xi^\beta_{,\alpha}\sum_A B^{(r,s)}_{A,\beta}\omega_A, \tag{A.23}$$

where the multi-dimensional product rule for differentiation is used. ω is expressed as a linear combination of basis functions by

$$\omega = \hat{\omega} = \sum_A B^{(r,s)}_A \omega_A. \tag{A.24}$$

For functions $\boldsymbol{v}$ transforming according to T_1 the following equality holds.

$$2 \qquad \hat{\boldsymbol{v}} = \begin{bmatrix} \sum_A B^{(r,\check{s})}_A v^1_A \\ \sum_A B^{(\check{r},s)}_A v^2_A \end{bmatrix} \tag{A.25a}$$

$$\begin{aligned} \boldsymbol{v}(\boldsymbol{x}) &= \frac{1}{\sqrt{g}} J \begin{bmatrix} \hat{v}^1 \\ \hat{v}^2 \end{bmatrix} = \frac{1}{\sqrt{g}} \begin{bmatrix} x^1_{,1}\hat{v}^1 + x^1_{,2}\hat{v}^2 \\ x^2_{,1}\hat{v}^1 + x^2_{,2}\hat{v}^2 \end{bmatrix} \\ &= \left(J^{\text{adj}}\right)^{-1} \begin{bmatrix} \hat{v}^1 \\ \hat{v}^2 \end{bmatrix} = \begin{bmatrix} \xi^2_{,2}\hat{v}^1 - \xi^1_{,2}\hat{v}^2 \\ -\xi^2_{,1}\hat{v}^1 + \xi^1_{,1}\hat{v}^2 \end{bmatrix} \end{aligned} \tag{A.25b}$$

Using the same mapping, the following equality can be deduced for the parametric and geometric divergence:

$$\begin{aligned} \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} &= \nabla_{\boldsymbol{x}} \cdot \begin{bmatrix} \xi^2_{,2}\hat{v}^1 - \xi^1_{,2}\hat{v}^2 \\ -\xi^2_{,1}\hat{v}^1 + \xi^1_{,1}\hat{v}^2 \end{bmatrix} = \\ &= \xi^2_{,12}\hat{v}^1 + \xi^2_{,2}\hat{v}^1_{,\alpha}\xi^\alpha_{,1} - \xi^1_{,12}\hat{v}^2 - \xi^1_{,2}\hat{v}^2_{,\alpha}\xi^\alpha_{,1} \\ &- \xi^2_{,12}\hat{v}^1 - \xi^2_{,1}\hat{v}^1_{,\alpha}\xi^\alpha_{,2} + \xi^1_{,12}\hat{v}^2 + \xi^1_{,1}\hat{v}^2_{,\alpha}\xi^\alpha_{,2} \\ &= \hat{v}^1_{,\alpha}\left(\xi^2_{,2}\xi^\alpha_{,1} - \xi^2_{,1}\xi^\alpha_{,2}\right) + \hat{v}^2_{,\alpha}\left(\xi^1_{,1}\xi^\alpha_{,2} - \xi^1_{,2}\xi^\alpha_{,1}\right) \\ &= \frac{1}{\sqrt{g}}\left(\hat{v}^1_{,1} + \hat{v}^2_{,2}\right) = \frac{1}{\sqrt{g}}\nabla_{\boldsymbol{\xi}} \cdot \hat{\boldsymbol{v}}. \end{aligned} \tag{A.26}$$

This is a very important identity, as it guarantees mass conservation to also hold in the parametric function spaces. It follows that

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = \frac{1}{\sqrt{g}}\nabla_{\boldsymbol{\xi}} \cdot \hat{\boldsymbol{v}} = \frac{1}{\sqrt{g}}\left(\sum_A B^{(r,\check{s})}_{A,1} v^1_A + \sum_A B^{(\check{r},s)}_{A,2} v^2_A\right). \tag{A.27}$$

Lastly, a general function p transforming according to T_2 is described. The derivatives of these functions are not needed for the discretization of the Stokes equation. Only the expression in terms of basis function is of interest:

$$p = \frac{1}{\sqrt{g}}p = \frac{1}{\sqrt{g}}\sum_A B^{(\check{r},\check{s})}_A p_A \tag{A.28}$$

A.3.4. Discrete weak formulation on parametric space

In this section the continuous weak formulation of Eq. (A.5) is discretized. The integrals are pulled back to the parametric domain using the transformations T_0 , T_1 , T_2 that were described in section (A.3.3). Subsequently, it is indicated how the corresponding matrix is composed.

The discrete weak formulation on Ω is given as follows:

Find $\omega \in \mathcal{W}_h$, $\boldsymbol{v} \in \mathcal{V}_{\boldsymbol{v}_\partial,h}$ *and* $p \in \mathcal{P}_h$ *such that for all* $\alpha \in \mathcal{W}_h$, $\boldsymbol{w} \in \mathcal{V}_{\boldsymbol{0},h}$ *and* $q \in \mathcal{P}_h$, *it holds that*

$$\int_\Omega \alpha\omega \; d\Omega + \int_\Omega \alpha_{,1} v^2 - \alpha_{,2} v^1 \; d\Omega = \int_{\partial\Omega} \alpha \boldsymbol{v}_\partial \cdot d\Gamma \tag{A.29a}$$

$$\int_\Omega w^1 \omega_{,2} - w^2 \omega_{,1} \; d\Omega - \int_\Omega p \nabla \cdot \boldsymbol{w} \; d\Omega = \int_\Omega \boldsymbol{w} \cdot \boldsymbol{f} \; d\Omega \tag{A.29b}$$

$$\int_\Omega q \nabla \cdot \boldsymbol{v} \; d\Omega = 0. \tag{A.29c}$$

Here $\Gamma = \boldsymbol{x}(\boldsymbol{\gamma})$ is a parametrization of the boundary $\partial\Omega$ and $\boldsymbol{\gamma} : [0,4] \to \mathbb{R}^2$ is the parametrization of the parametric boundary $\partial\hat{\Omega}$. $\boldsymbol{\gamma} = \boldsymbol{\gamma}_1 \oplus \boldsymbol{\gamma}_2 \oplus \boldsymbol{\gamma}_3 \oplus \boldsymbol{\gamma}_4$ is given by

$$\begin{aligned}
&\boldsymbol{\gamma}_1 : [0,1) \to \hat{\Omega}; \quad \boldsymbol{\gamma}_1(\ell) = (\ell, 0), \\
&\boldsymbol{\gamma}_2 : [1,2) \to \hat{\Omega}; \quad \boldsymbol{\gamma}_2(\ell) = (1, \ell - 1), \\
&\boldsymbol{\gamma}_3 : [2,3) \to \hat{\Omega}; \quad \boldsymbol{\gamma}_3(\ell) = (3 - \ell, 1), \\
&\boldsymbol{\gamma}_4 : [3,4) \to \hat{\Omega}; \quad \boldsymbol{\gamma}_4(\ell) = (0, 4 - \ell).
\end{aligned} \tag{A.30}$$

In order to keep things readable the inner product on $\hat{\Omega}$ is introduced.

Definition. *For arbitrary* C^0 *continuous functions* $\hat{f}(\boldsymbol{\xi})$ *and* $\hat{g}(\boldsymbol{\xi})$ *the inner product defined on* $\hat{\Omega}$ *is denoted by*

$$(\hat{f}, \hat{g})_{\hat{\Omega}} = \int_{\hat{\Omega}} \hat{f}\hat{g} \; d\hat{\Omega}. \tag{A.31}$$

In appendix B the integrals of Eq. (A.29) are pulled back to the parametric domain. The resulting weak formulation defined on $\hat{\Omega}$ reads as

Find $\hat{\omega} \in \hat{\mathcal{W}}_h$, $\hat{\boldsymbol{v}} \in \hat{\mathcal{V}}_{\boldsymbol{v}_\partial,h}$ *and* $\hat{p} \in \hat{\mathcal{P}}_h$ *such that for all* $\hat{\alpha} \in \hat{\mathcal{W}}_h$, $\hat{\boldsymbol{w}} \in \hat{\mathcal{V}}_{0,h}$ *and* $\hat{q} \in \hat{\mathcal{P}}_h$,

$$(\hat{\alpha}, \; \hat{\omega} \; \sqrt{g})_{\hat{\Omega}} - \left(\hat{\alpha}_{,\beta} \sqrt{g} g^{\beta 2}, \; \hat{v}^1\right)_{\hat{\Omega}} + \left(\hat{\alpha}_{,\beta} \sqrt{g} g^{\beta 1}, \; \hat{v}^2\right)_{\hat{\Omega}} = \int_0^4 \hat{\alpha} \boldsymbol{v}_\partial \; J\boldsymbol{\gamma}' \tag{A.32a}$$

$$\begin{aligned}
\left(\hat{w}^1, \; \sqrt{g} g^{\beta 2} \hat{\omega}_{,\beta}\right)_{\hat{\Omega}} - \left(\hat{w}^1_{,1}, \; \frac{1}{\sqrt{g}} \hat{p}\right)_{\hat{\Omega}} - \left(\hat{w}^2, \; \sqrt{g} g^{\beta 1} \hat{\omega}_{,\beta}\right)_{\hat{\Omega}} - \left(\hat{w}^2_{,2}, \; \frac{1}{\sqrt{g}} \hat{p}\right)_{\hat{\Omega}} &= \left(\hat{w}^1, \; x^\alpha_{,1} f^\alpha\right)_{\hat{\Omega}} \\
&\quad + \left(\hat{w}^2, \; x^\alpha_{,2} f^\alpha\right)_{\hat{\Omega}}
\end{aligned} \tag{A.32b}$$

$$\left(\frac{1}{\sqrt{g}} \hat{q}, \; \hat{v}^1_{,1}\right)_{\hat{\Omega}} + \left(\frac{1}{\sqrt{g}} \hat{q}, \; \hat{v}^2_{,2}\right)_{\hat{\Omega}} = 0. \tag{A.32c}$$

Substituting the basis functions and recognising that the test function control variables can be chosen arbitrarily, leads to the following discrete weak formulation on parametric space $\hat{\Omega}$:

$$\begin{aligned}
\sum_B \omega_B \left(B^{(r,s)}_A, \; B^{(r,s)}_B \; \sqrt{g}\right)_{\hat{\Omega}} - \sum_B v^1_B \left(B^{(r,s)}_{A,\beta} \sqrt{g} g^{\beta 2}, \; B^{(r,\check{s})}_B\right)_{\hat{\Omega}} \\
+ \sum_B v^2_B \left(B^{(r,s)}_{A,\beta} \sqrt{g} g^{\beta 1}, \; B^{(\check{r},s)}_B\right)_{\hat{\Omega}} = \int_0^4 B^{(r,s)}_A \boldsymbol{v}_\partial \; J\boldsymbol{\gamma}' \; ds
\end{aligned} \tag{A.33a}$$

$$\sum_B \omega_B \left(B^{(r,\check{s})}_A, \; \sqrt{g} g^{\beta 2} B^{(r,s)}_{B,\beta}\right)_{\hat{\Omega}} - \sum_B p_B \left(B^{(r,\check{s})}_{A,1}, \; \frac{1}{\sqrt{g}} B^{(\check{r},\check{s})}_B\right)_{\hat{\Omega}} = \left(B^{(r,\check{s})}_A, \; x^1_{,1} f^1 + x^2_{,1} f^2\right)_{\hat{\Omega}} \tag{A.33b}$$

$$-\sum_B \omega_B \left(B^{(\check{r},s)}_A, \; \sqrt{g} g^{\beta 1} B^{(r,s)}_{B,\beta}\right)_{\hat{\Omega}} - \sum_B p_B \left(B^{(\check{r},s)}_{A,2}, \; \frac{1}{\sqrt{g}} B^{(\check{r},\check{s})}_B\right)_{\hat{\Omega}} = \left(B^{(\check{r},s)}_A, \; x^1_{,2} f^1 + x^2_{,2} f^2\right)_{\hat{\Omega}} \tag{A.33c}$$

$$\sum_B v^1_B \left(\frac{1}{\sqrt{g}} B^{(\check{r},\check{s})}_A, \; B^{(r,\check{s})}_{B,1}\right)_{\hat{\Omega}} + \sum_B v^2_B \left(\frac{1}{\sqrt{g}} B^{(\check{r},\check{s})}_A, \; B^{(\check{r},s)}_{B,2}\right)_{\hat{\Omega}} = 0 \tag{A.33d}$$

Eq. (A.33) forms a linear system of equations

$$M\boldsymbol{d}_h = \boldsymbol{r}_h, \tag{A.34}$$

with

$$M = \begin{bmatrix} F & C^1 & C^2 & \\ A^1 & & & G^1 \\ A^2 & & & G^2 \\ & D^1 & D^2 & \end{bmatrix}, \quad \boldsymbol{d}_h = \begin{bmatrix} \boldsymbol{\omega}_h \\ \boldsymbol{v}_h^1 \\ \boldsymbol{v}_h^2 \\ \boldsymbol{p}_h \end{bmatrix}, \quad \boldsymbol{r}_h = \begin{bmatrix} \boldsymbol{t}_h \\ \boldsymbol{f}_h^1 \\ \boldsymbol{f}_h^2 \\ \boldsymbol{0} \end{bmatrix}. \tag{A.35}$$

The matrix entries are defined by the inner products on the left-hand sides of the equations in (A.33). The position of the inner products is matched with the position of the submatrices. For example,

$$\begin{aligned} [F]_{AB} &= \left(B_A^{(r,s)},\ B_B^{(r,s)}\ \sqrt{g}\right)_{\hat{\Omega}}, \\ \left[C^1\right]_{AB} &= \left(B_{A,\beta}^{(r,s)}\sqrt{g}g^{\beta 2},\ B_B^{(r,\check{s})}\right)_{\hat{\Omega}}, \\ \left[C^2\right]_{AB} &= \left(B_{A,\beta}^{(r,s)}\sqrt{g}g^{\beta 1},\ B_B^{(\check{r},s)}\right)_{\hat{\Omega}}. \end{aligned} \tag{A.36}$$

The analogue holds for the other submatrices. Note that for each submatrix the ranges of A and B depend on the specific basis functions associated with it.

Due to the Dirichlet boundary conditions some of the control variables in the vectors $\boldsymbol{v}_h^1$ and $\boldsymbol{v}_h^2$ are in fact known. They are obtained by projecting $\boldsymbol{v}_\partial$ onto the space of discrete velocities. These boundary terms will then contribute to the right-hand-side of Eq. (A.34), and the corresponding rows and columns at the left-hand-side are eliminated.

A.4. Preliminary results

The steady state Stokes flow on a flat 2D geometry has been implemented from scratch using Matlab. It uses the discrete weak formulation of Eq. (A.32) on parametric space and the 2-dimensional geometric mapping as defined in Eq. (3.11). A manufactured solution is created in order to verify the convergence rates of the implementation. The manufactured solution on geometric space is given by

$$\begin{aligned} \omega &= 2\pi \sin(x^1\pi)\sin(x^2\pi), \quad & \boldsymbol{v} &= \begin{bmatrix} \sin(x^1\pi)\cos(x^2\pi) \\ -\cos(x^1\pi)\sin(x^2\pi) \end{bmatrix}, \\ p &= \pi\cos(x^1\pi)\cos(x^2\pi) + 1, \quad & \boldsymbol{f} &= \begin{bmatrix} \pi^2\sin(x^1\pi)\cos(x^2\pi) \\ -3\pi^2\cos(x^1\pi)\sin(x^2\pi) \end{bmatrix} \end{aligned} \tag{A.37}$$

and the geometry is found in figure 3.5. The results can be found in figure A.1. Furthermore, a convergence plot of each of the discrete functions can be found in figure A.2. As the figures indicate, the discrete solution converges to the the exact solution with order $p_{\min}$ for the H^1-norm and order $p_{\min+1}$ for the L_2-norm, as expected.

Unfortunately, in some cases, an implementation error interferes with the solution. For example, taking the same manufactured solution as before, but changing the geometry to the one found in figure A.3, leads to the error in that same figure.

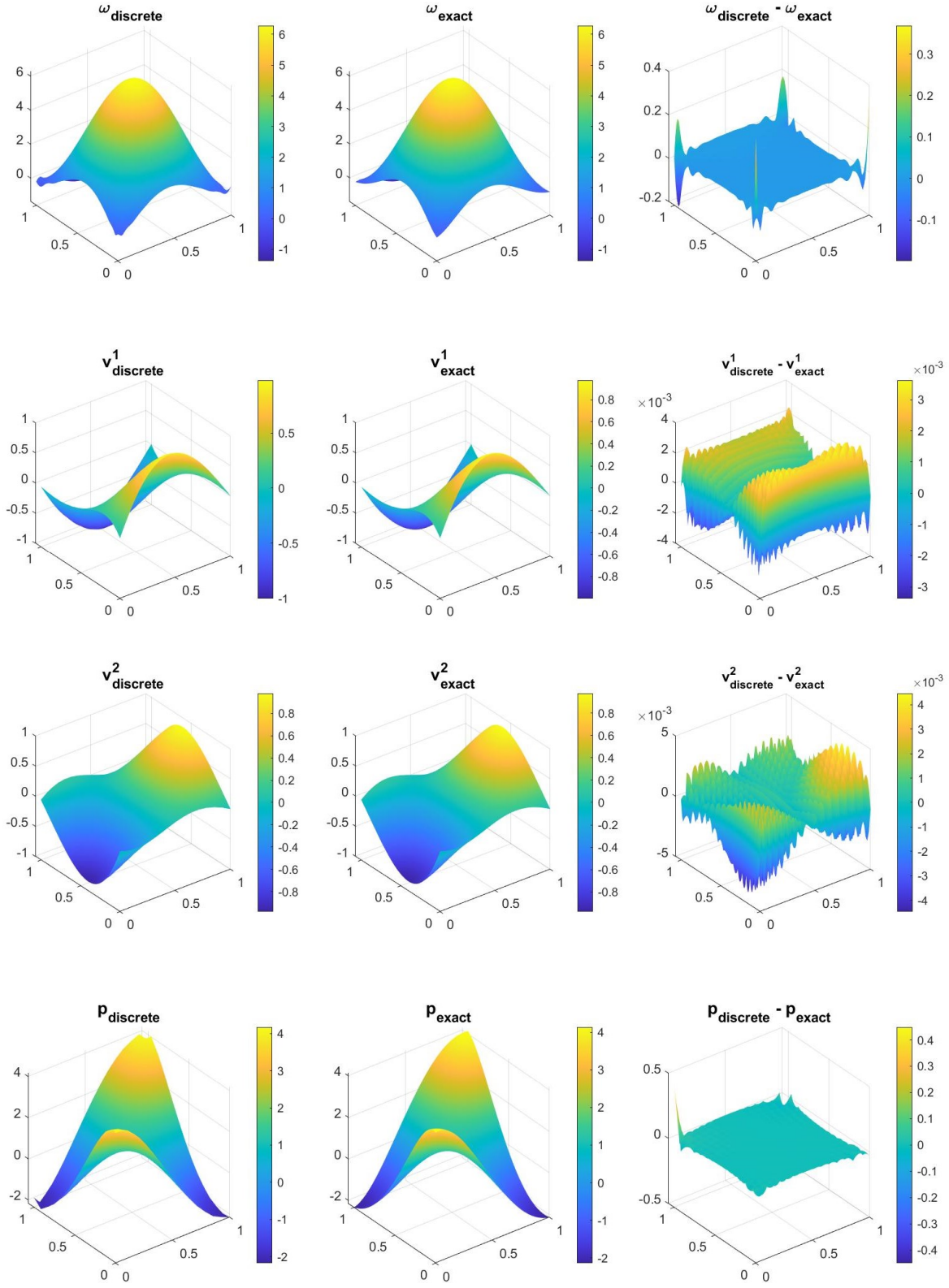


Figure A.1: Resulting plots of (from left to right) the discrete solutions, the exact solutions and their difference. The plots show the vorticity ω, the velocity $\boldsymbol{v}$ and the pressure p on the geometric domain.

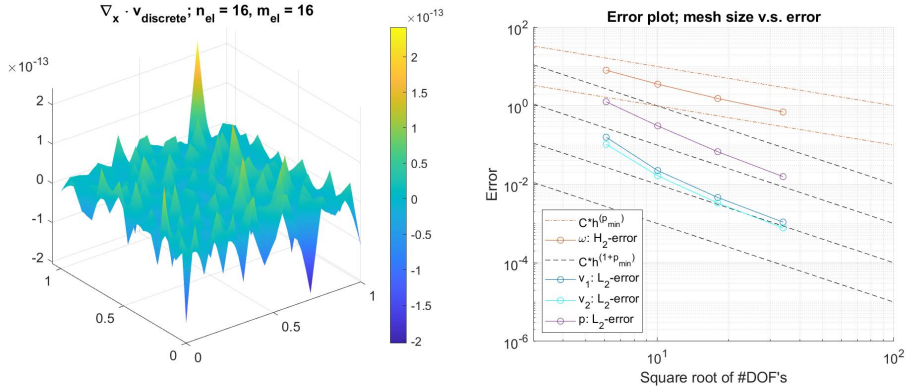
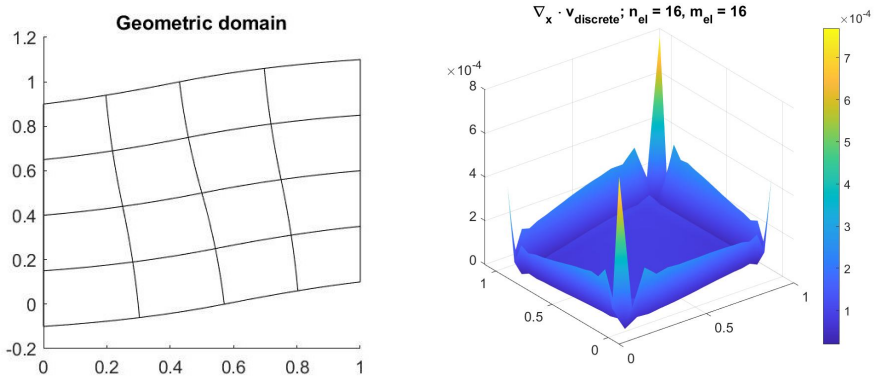


Figure A.2: On the left a plot of the discrete divergence which is 0 up to computer precision. On the right a convergence plot of the vorticity, both coordinates of the velocity and the pressure.



(a) A 2D domain

(b) Divergence of v

Figure A.3: The combination of the manufactured solution of Eq. (A.37) and the geometry on the left, gives a non-zero divergence of v.

B | Pullback of the 2D weak formulation

In order to derive the weak formulation on $\hat{\Omega}$, the integrals of Eq. (A.29) need to be pulled back to the parametric domain. First, note that

$$\begin{aligned}
\xi^\beta_{,1}x^2_{,2} - \xi^\beta_{,2}x^1_{,2} &= \begin{cases} \sqrt{g}\left(\xi^1_{,1}\xi^1_{,1} + \xi^1_{,2}\xi^1_{,2}\right) = & \sqrt{g}\ \boldsymbol{a}^1\cdot\boldsymbol{a}^1 = \quad \sqrt{g}g^{11}, \quad \text{for } \beta = 1 \\ \sqrt{g}\left(\xi^2_{,1}\xi^1_{,1} + \xi^2_{,2}\xi^1_{,2}\right) = & \sqrt{g}\ \boldsymbol{a}^2\cdot\boldsymbol{a}^2 = \quad \sqrt{g}g^{12}, \quad \text{for } \beta = 2, \end{cases} \\
-\xi^\beta_{,1}x^2_{,1} + \xi^\beta_{,2}x^1_{,1} &= \begin{cases} \sqrt{g}\left(\xi^1_{,1}\xi^2_{,1} + \xi^1_{,2}\xi^2_{,2}\right) = & \sqrt{g}\ \boldsymbol{a}^1\cdot\boldsymbol{a}^2 = \quad \sqrt{g}g^{12}, \quad \text{for } \beta = 1 \\ \sqrt{g}\left(\xi^2_{,1}\xi^2_{,1} + \xi^2_{,2}\xi^2_{,2}\right) = & \sqrt{g}\ \boldsymbol{a}^2\cdot\boldsymbol{a}^2 = \quad \sqrt{g}g^{22}, \quad \text{for } \beta = 2. \end{cases}
\end{aligned} \tag{B.1}$$

Transformations T_0, T_1, T_2 will be used to obtain the pulled back integrals. Starting with Eq. (A.29a), the pull-back of the left-hand side is equivalent to

$$\begin{aligned}
&(\hat{\alpha},\ \hat{\omega}\ |J|)_{\hat{\Omega}} + \left(\xi^\beta_{,1}\hat{\alpha}_{,\beta},\ \frac{1}{|J|}\left[x^2_{,\alpha}\hat{v}^\alpha\right]\ |J|\right)_{\hat{\Omega}} - \left(\xi^\beta_{,2}\hat{\alpha}_{,\beta},\ \frac{1}{|J|}\left[x^1_{,\alpha}\hat{v}^\alpha\right]\ |J|\right)_{\hat{\Omega}} \\
\Longrightarrow\ &(\hat{\alpha},\ \hat{\omega}\ |J|)_{\hat{\Omega}} - \left(\hat{\alpha}_{,\beta}[-\xi^\beta_{,1}x^2_{,1} + \xi^\beta_{,2}x^1_{,1}],\ \hat{v}^1\right)_{\hat{\Omega}} + \left(\hat{\alpha}_{,\beta}[\xi^\beta_{,1}x^2_{,2} - \xi^\beta_{,2}x^1_{,2}],\ \hat{v}^2\right)_{\hat{\Omega}} \\
\Longrightarrow\ &(\hat{\alpha},\ \hat{\omega}\ |J|)_{\hat{\Omega}} - \left(\sqrt{g}g^{\beta 2}\hat{\alpha}_{,\beta},\ \hat{v}^1\right)_{\hat{\Omega}} + \left(\sqrt{g}g^{\beta 1}\hat{\alpha}_{,\beta},\ \hat{v}^2\right)_{\hat{\Omega}}.
\end{aligned} \tag{B.2}$$

To obtain the right-hand side, the boundary integral is represented as an one dimensional integral over parameters $\boldsymbol{\gamma}$. The following identity is used

$$\begin{aligned}
\int_{\partial\Omega} \alpha\boldsymbol{v}_\partial\cdot d\Gamma &= \int_0^4 \alpha(\boldsymbol{x}(\boldsymbol{\gamma}(s)))\ \boldsymbol{v}_\partial(\boldsymbol{x}(\boldsymbol{\gamma}(s)))\cdot\boldsymbol{x}_{,s}(\boldsymbol{\gamma}(s))\ ds \\
&= \int_0^4 \hat{\alpha}(\boldsymbol{\gamma}(s))\ \boldsymbol{v}_\partial(\boldsymbol{x}(\boldsymbol{\gamma}(s)))\cdot J\boldsymbol{\gamma}'(s)\ ds \\
&= \int_0^4 \hat{\alpha}\boldsymbol{v}_\partial\cdot J\boldsymbol{\gamma}'\ ds.
\end{aligned} \tag{B.3}$$

Note that

$$J\boldsymbol{\gamma}' = \begin{cases} \boldsymbol{a}_1, & \text{on } \partial\hat{\Omega}_B, \quad \ell\in[0,1) \\ \boldsymbol{a}_2, & \text{on } \partial\hat{\Omega}_R, \quad \ell\in[1,2) \\ -\boldsymbol{a}_1, & \text{on } \partial\hat{\Omega}_T, \quad \ell\in[2,3) \\ -\boldsymbol{a}_2, & \text{on } \partial\hat{\Omega}_L, \quad \ell\in[3,4). \end{cases} \tag{B.4}$$

The pullback of Eq. (A.29b) is equivalent to

$$\begin{aligned}
&\left(\frac{1}{|J|}\left[x^1_{,\alpha}\hat{w}^\alpha\right],\ \xi^\beta_{,2}\hat{\omega}_{,\beta}\ |J|\right)_{\hat{\Omega}} - \left(\frac{1}{|J|}\left[x^2_{,\alpha}\hat{w}^\alpha\right],\ \xi^\beta_{,1}\hat{\omega}_{,\beta}\ |J|\right)_{\hat{\Omega}} - \left(\frac{1}{|J|}\nabla_{\boldsymbol{\xi}}\cdot\hat{\boldsymbol{w}},\ \frac{1}{|J|}\hat{p}\ |J|\right)_{\hat{\Omega}} \\
&= \left(\frac{1}{|J|}\left[x^1_{,\alpha}\hat{w}^\alpha\right],\ f^1\ |J|\right)_{\hat{\Omega}} + \left(\frac{1}{|J|}\left[x^2_{,\alpha}\hat{w}^\alpha\right],\ f^2\ |J|\right)_{\hat{\Omega}}.
\end{aligned} \tag{B.5}$$

Rearranging terms gives

$$\begin{aligned}
&\left(\hat{w}^1,\ [-\xi^\beta_{,1}x^2_{,1} + \xi^\beta_{,2}x^1_{,1}]\hat{\omega}_{,\beta}\right)_{\hat{\Omega}} - \left(\hat{w}^1_{,1},\ \frac{1}{|J|}\hat{p}\right)_{\hat{\Omega}} - \left(\hat{w}^2,\ [\xi^\beta_{,1}x^2_{,2} - \xi^\beta_{,2}x^1_{,2}]\hat{\omega}_{,\beta}\right)_{\hat{\Omega}} - \left(\hat{w}^2_{,2},\ \frac{1}{|J|}\hat{p}\right)_{\hat{\Omega}} \\
&= \left(\hat{w}^1,\ x^\alpha_{,1}f^\alpha\right)_{\hat{\Omega}} + \left(\hat{w}^2,\ x^\alpha_{,2}f^\alpha\right)_{\hat{\Omega}}
\end{aligned} \tag{B.6}$$

and so

$$\begin{aligned}&\left(\hat{w}^1,\ \sqrt{g}g^{\beta 2}\hat{\omega}_{,\beta}\right)_{\hat{\Omega}} - \left(\hat{w}^1_{,1},\ \frac{1}{|J|}\hat{p}\right)_{\hat{\Omega}} - \left(\hat{w}^2,\ \sqrt{g}g^{\beta 1}\hat{\omega}_{,\beta}\right)_{\hat{\Omega}} - \left(\hat{w}^2_{,2},\ \frac{1}{|J|}\hat{p}\right)_{\hat{\Omega}} \\ &= \left(\hat{w}^1,\ x^\alpha_{,1}f^\alpha\right)_{\hat{\Omega}} + \left(\hat{w}^2,\ x^\alpha_{,2}f^\alpha\right)_{\hat{\Omega}}.\end{aligned} \tag{B.7}$$

Lastly, the pullback of (A.29c) reads as

$$\begin{aligned}\left(\frac{1}{|J|}\hat{q},\ \frac{1}{|J|}\nabla_{\boldsymbol{\xi}}\cdot\hat{\boldsymbol{v}}\ |J|\right)_{\hat{\Omega}} &= 0 \\ \Longrightarrow \left(\frac{1}{|J|}\hat{q},\ \hat{v}^1_{,1}\right)_{\hat{\Omega}} + \left(\frac{1}{|J|}\hat{q},\ \hat{v}^2_{,2}\right)_{\hat{\Omega}} &= 0.\end{aligned} \tag{B.8}$$