Isogeometric Analysis for High Order Geometric Partial Differential Equations with Applications

THÈSE N^{O} 7950 (2017)

PRÉSENTÉE LE 20 SEPTEMBRE 2017
À LA FACULTÉ DES SCIENCES DE BASE
CHAIRE DE MODÉLISATION ET CALCUL SCIENTIFIQUE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Andrea BARTEZZAGHI

acceptée sur proposition du jury:

Prof. V. Panaretos, président du jury
Prof. A. Quarteroni, Prof. L. Dede', directeurs de thèse
Prof. A. Bonito, rapporteur
Prof. S. Bartels, rapporteur
Prof. A. Buffa, rapporteuse

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2017

Research is what I'm doing
when I don't know what I'm doing

— Wernher von Braun

Acknowledgements

Firstly, I would like to thank my advisor, Prof. Alfio Quarteroni, for giving me the huge opportunity of working in the *Chair of Modeling and Scientific Computing* (CMCS) at EPFL. I am thankful for his guidance and support throughout these years, during which I have learned and developed a lot, both personally and professionally. Moreover, CMCS is an highly inspirational and friendly environment and it has been a real pleasure working there.

I am also very grateful to my coadvisor, Dr. Luca Dedè, for his constant support and patience, at every hour of the day, which made it possible for me to overcome all problems I had to face during the PhD and learn from them. Thanks to his precious advices I have been able to grow from all points of view. In particular, he taught me to *think* (i.e. write on scratch paper) before *acting* (i.e. writing the code).

I would also like to thank MER Simone Deparis and Dr. Andrea Manzoni for the many fruitful discussions and the many lunches we had at EPFL.

I am thankful to the members of the jury, Prof. Sören Bartels, Prof. Andrea Bonito, and Prof. Annalisa Buffa, for reviewing this thesis and having provided me with insightful comments and feedbacks; I thank also Prof. Victor Panaretos for being president of the jury.

I am very grateful to all my past and present colleagues at CMCS and, in general, EPFL. In particular, to Anto and Nico, my fellow doctoral mates, with whom I shared a lot of discussions, jokes, beers, and a fridge; to Ste e Pego, for the dinners and for giving me driving lessons, the hard way; to Luca (Vasa) and the powerful Brotherhood; to Paolo, for the cynic jokes and advices; to Matteo, for the beers; to Claudia, Elena and Marco, for sharing the divine pearls of Prof. Umiera; to Davide, Federico, and Anna, for the countless aids and the nice discussions.

A big thank to the *Bussigny* volley team, the *Sfigatos* team, the *Tuesday*'s team, and to all the volley and beachvolley mates I had the pleasure to play with: without sports, I would not have been able to survive these years without becoming crazy.

Many thanks also to Ruggero, Boes, and the proud members of the historical *Clan Boe*, to Gio, for the life endangering adventures we shared, to Ivan, for the absurd laughs and cocktails we had, to Gaia and Giorgio, for the super deep discussions and nice time around Europe, to the *222* warmates, for the fruitful exchanges of bullets, and to the countless other colleagues and friends I met during these years, with whom I shared a lot of laughs and had a great time.

A sincerely huge thank goes to all my family: it is needless to say that without their support I would not have been able to achieve anything. I thank them for constantly guiding and helping me; my communication skills are not the best, and I am sorry for that, but they have a place in the first line in my heart, always.

Last, but not least, I am extremely thankful to Arianna. Since when we met, she has always supported me, unconditionally, in all situations and through all the difficulties. Her help was invaluable for me, especially at the end of the thesis, to not give up to the accumulated stress and anxiety, but to remain calm and focused. I really loved these last two years, in great part because of her and all our adventures.

I gratefully acknowledge the financial support of the Swiss National Science Foundation through the project “Isogeometric Analysis for Partial Differential Equations: surface models and optimization problems in Haemodynamics” (project ♯ 147033).

Lausanne, August 2017 Andrea Bartezzaghi

Abstract

In this thesis, we consider the numerical approximation of high order geometric Partial Differential Equations (PDEs). We first consider high order PDEs defined on surfaces in the 3D space that are represented by single-patch tensor product NURBS. Then, we spatially discretize the PDEs by means of NURBS-based Isogeometric Analysis (IGA) in the framework of the Galerkin method. With this aim, we consider the construction of periodic NURBS function spaces with high degree of global continuity, even on closed surfaces. As benchmark problems for the proposed discretization, we propose Laplace–Beltrami problems of the fourth and sixth orders, as well as the corresponding eigenvalue problems, and we analyze the impact of the continuity of the basis functions on the accuracy as well as on computational costs. The numerical solution of two high order phase field problems on both open and closed surfaces is also considered: the fourth order Cahn–Hilliard equation and the sixth order crystal equation, both discretized in time with the generalized-α method. We then consider the numerical approximation of geometric PDEs, derived, in particular, from the minimization of shape energy functionals by L^2-gradient flows. We analyze the mean curvature and the Willmore gradient flows, leading to second and fourth order PDEs, respectively. These nonlinear geometric PDEs are discretized in time with Backward Differentiation Formulas (BDF), with a semi-implicit formulation based on an extrapolation of the geometry, leading to a linear problem to be solved at each time step. Results about the numerical approximation of the two geometric flows on several geometries are analyzed. Then, we study how the proposed mathematical framework can be employed to numerically approximate the equilibrium shapes of lipid bilayer biomembranes, or vesicles, governed by the Canham–Helfrich curvature model. We propose two numerical schemes for enforcing the conservation of the area and volume of the vesicles, and report results on benchmark problems. Then, the approximation of the equilibrium shapes of biomembranes with different values of reduced volume is presented. Finally, we consider the dynamics of a vesicle, e.g. a red blood cell, immersed in a fluid, e.g. the plasma. In particular, we couple the curvature-driven model for the lipid membrane with the incompressible Navier–Stokes equations governing the fluid. We consider a segregated approach, with a formulation based on the Resistive Immersed Surface method applied to NURBS geometries. After analyzing benchmark fluid simulations with immersed NURBS objects, we report numerical results for the investigation of the dynamics of a vesicle under different flow conditions.

Key words: High order Partial Differential Equation, Geometric Partial Differential Equation, Surface, NURBS, Isogeometric Analysis, Biomembrane

Résumé

Dans cette thèse, nous considérons l'approximation numérique d'Équations aux Dérivées Partielles (EDPs) d'ordre élevé. Nous considérons tout d'abord des EDPs d'ordre élevé, définies sur des surfaces dans l'espace 3D, représentées par single-patch NURBS. Ensuite, les EDPs sont discrétisées dans l'espace grâce à l'Analyse Isogéométrique (IGA) basée sur les NURBS dans le cadre de la méthode de Galerkin. Dans ce but, nous considérons la construction d'espaces de fonctions NURBS périodiques avec un degré élevé de continuité globale, même sur des surfaces fermées. Comme benchmarks pour la discrétisation proposée, nous proposons les problèmes de Laplace–Beltrami d'ordre quatre et six, ainsi que les problèmes de valeurs propres correspondants, et nous analysons l'impact de la continuité des fonctions de bases sur la précision ainsi que sur les coûts computationels. La solution numérique de deux problèmes de phase field sur des surfaces à la fois ouvertes et fermées est aussi considérée : l'équation de Cahn–Hilliard, d'ordre quatre, et l'équation du chrystal, d'ordre six, discrétisées par rapport au temps avec la α-méthode généralisée. Nous considérons ensuite l'approximation numérique d'EDPs géométriques, obtenues, en particulier, à partir de la minimisation des fonctionnelles d'énergie par le flux de gradient L^2. Nous analysons les flux de la courbure moyenne et de Willmore, menant à des EDPs de second et de quatrième ordre, respectivement. Ces EDPs géométriques non linéaires sont discrétisées par rapport au temps avec des Formules de Différentiation Rétrograde (BDF), grâce à une formulation semi-implicite basée sur une extrapolation de la géométrie, menant à un problème linéaire à résoudre à chaque pas de temps. Des résultats sur l'approximation numérique de deux flux géométriques sur différentes géométries sont analysés. Ensuite, nous regardons comment le cadre mathématique proposé peut être employé pour approximer numériquement les formes d'équilibre de biomembranes lipidiques, ou vésicules, gouvernées par le modèle de Canham–Helfrich. Nous proposons deux schémas numériques pour imposer la conservation de l'aire et du volume des vésicules, et nous rapportons les résultats sur des problèmes benchmark. Par la suite, l'approximation de la forme d'équilibre de biomembranes avec différentes de valeurs de volume réduit est présentée. Finalement, nous considérons la dynamique d'un vésicule immergé dans un fluide. En particulier, nous couplons le modèle guidé par la courbure pour la membrane lipidique avec les équations de Navier–Stokes incompressibles gouvernant le fluide. Nous considérons une approche ségréguée, avec une formulation basée sur la méthode des Surfaces Immergées Résistives appliqué sur des géométries NURBS. Après avoir analysé des simulations benchmark de fluides avec des objets NURBS immergés, nous reportons les résultats numériques pour l'investigation des dynamiques d'un vésicule soumis à différentes conditions de flux.

Mots clefs : Équations aux Dérivées Partielles d’ordre élevé, Équations aux Dérivées Partielles géométriques, Surface, NURBS, Analyse Isogéométrique, Biomembrane

Contents

Acknowledgements

Abstract

Introduction 1

1 High order surface PDEs 7
- 1.1 Geometrical representation of surfaces . . . 8
  - 1.1.1 Surface differential operators . . . 10
  - 1.1.2 Identity function and surface curvatures . . . 11
- 1.2 High order surface PDEs . . . 14
  - 1.2.1 Laplace–Beltrami biharmonic problem . . . 15
  - 1.2.2 Laplace–Beltrami triharmonic problem . . . 15
  - 1.2.3 High order Laplace–Beltrami eigenvalue problems . . . 16
- 1.3 Phase field models . . . 18
  - 1.3.1 Cahn–Hilliard equation . . . 18
  - 1.3.2 Phase field crystal equation . . . 20

2 Geometric PDEs 23
- 2.1 Evolving surfaces . . . 24
- 2.2 Shape differential calculus . . . 26
- 2.3 Geometric PDEs as geometric gradient flows . . . 28
- 2.4 Mean curvature flow . . . 29
- 2.5 Willmore flow . . . 30
  - 2.5.1 Willmore flow with spontaneous curvature . . . 31

3 Isogeometric Analysis 33
- 3.1 Univariate B-Spline and NURBS . . . 35
- 3.2 Multivariate B-Spline and NURBS . . . 38
- 3.3 NURBS function spaces . . . 40
  - 3.3.1 Periodic NURBS function spaces . . . 43
- 3.4 NURBS-based Isogeometric Analysis . . . 45
  - 3.4.1 A priori error estimate . . . 47

4 IGA for high order surface PDEs **49**

4.1 Numerical approximation of steady PDEs 50

4.1.1 Test 4.1. Laplace–Beltrami biharmonic problem 50

4.1.2 Test 4.1. Comparison with isoparametric FEM 55

4.1.3 Test 4.2. Laplace–Beltrami triharmonic problem 55

4.1.4 Test 4.2. Comparison with isoparametric FEM 57

4.1.5 Test 4.3. High order Laplace–Beltrami eigenvalue problems 59

4.2 Phase field models 61

4.2.1 Time discretization by the generalized-α method 61

4.2.2 Test 4.4. Approximation of the Cahn–Hilliard equation 63

4.2.3 Test 4.5. Approximation of the phase field crystal equation 69

5 IGA for geometric PDEs **73**

5.1 Spatial discretization with IGA 74

5.2 Time discretization with Backward Differentiation Formulas 77

5.3 Approximation of the mean curvature flow 81

5.3.1 Numerical results 82

5.4 Approximation of the Willmore flow 93

5.4.1 Numerical results 94

6 Equilibrium shapes of lipid vesicles **103**

6.1 Mathematical Model 104

6.1.1 Energy minimization 106

6.2 Numerical Approximation 108

6.2.1 Space discretization 108

6.2.2 Time discretization 109

6.2.3 Enforcement of the area and volume constraints 110

6.3 Numerical Results 114

6.3.1 Benchmark cases 114

6.3.2 Comparison of equilibrium shapes 119

7 Dynamics of lipid vesicles in fluids **127**

7.1 Description of the domains 128

7.2 The fluid equations 129

7.2.1 Non-dimensionalization of the Navier–Stokes equations 130

7.2.2 Weak formulation of the fluid equations 132

7.2.3 The Resistive Immersed Surface method 132

7.2.4 Computation of forces on an immersed object 137

7.2.5 Space discretization of the fluid equations 138

7.2.6 SUPG stabilization 139

7.2.7 Time discretization of the fluid equations 141

7.2.8 Numerical results: lid-driven cavity 142

7.2.9 Numerical results: flow past cylinder 145

7.3 Signed distance and resistive method with NURBS 152
7.3.1 Selection of candidate points 153
7.3.2 Point projection on NURBS curves 156
7.3.3 Point projection on surfaces 157
7.3.4 Resistive NURBS immersed object 159
7.3.5 Numerical example: NURBS curves immersed in a 2D domain 160
7.4 Fluid-membrane interaction 163
7.4.1 Space discretization of the coupled problem 167
7.4.2 Time discretization of the coupled problem 169
7.4.3 Numerical results 173

Conclusions **189**

Bibliography **209**

Curriculum Vitae **211**

Introduction

The focus of this thesis is the numerical approximation of Partial Differential Equations (PDEs) defined on surfaces and of geometric PDEs, which involve high order differential operators.

In several fields, mathematical models often involve PDEs defined on manifolds of lower dimensionality with respect to the physical space in which they are hosted. This is the case, for instance, of PDEs defined on curves embedded in the two-dimensional space, or defined on surfaces and curves in the three-dimensional space [1]. Examples of *surface PDEs* are in structural mechanics, where thin structures are often modeled by employing plates and shells [2]; in fluid dynamics, where shallow water models are often formulated on surfaces [3], as in the context of atmospheric simulations [4,5]; in biology, where electrophysiology models can be formulated as surface PDEs to simulate the propagation of the cardiac electric signal on the atria, that can be regarded as 2D surfaces due to the small thickness of the atrial walls [6,7]; in image processing and computer graphics, where problems defined on surfaces arise, for example, in the segmentation of medical images or the generation of procedural textures [8,9].

Another family of problems, usually formulated on manifolds, is that of *geometric PDEs*. With this term we indicate those problems whose main unknown drives also the geometrical evolution of the domain where the equations are defined [10]. Geometric PDEs are usually formulated on curves or surfaces which evolve, in time or pseudo-time, towards the minimization of a shape energy functional, driven by a gradient flow process. Formulations of this kind arise in many applications, for instance when modeling the structure and properties of materials [11,12,13], in image processing and segmentation [14,15,16], or shape reconstruction and surface modeling [17,18,19,20,21].

Over the years, several techniques for the numerical approximation of surface and geometric PDEs have been developed. Among the others, approaches based on the Finite Element Method (FEM) have been considered for instance in [22,23,24], or based on the Adaptive FEM in [25] for approximating elliptic surface PDEs, and in [26,27,28] in the context of geometric PDEs and shape optimization. Surfaces can also be treated implicitly, for example by means of level set methods [29,30] or through diffuse interfaces [31,32]. In general, FEM-based approaches are not error free when considering the geometrical representation of the computational domain and for the evaluation of the geometric quantities. Surface PDEs, and in particular geometric PDEs, require an accurate representation of the geometry, since geometric quantities, as the normal and the curvatures, as well as tangential differential operators, are usually involved. Therefore, particular care must be devoted to the choice of a suitable geometrical representation and space discretization method, as the error introduced by an approximation of the geometry can affect the accuracy of the numerical solutions [33]; this could also potentially lead to the adoption of inefficient computational meshes, requiring large amounts of Degrees Of Freedom (DOFs), in order to control the geometrical error.

As the focus of this thesis consists in the numerical approximation of high order surface and geometric PDEs, we choose to represent the geometries by NURBS and to spatially discretize the PDEs by means of Isogeometric Analysis (IGA). Based on the seminal work [34], IGA builds upon the isogeometric concept for which the same basis functions used for representing the geometry of the computational domain are also employed for building the approximation spaces of the PDEs [35]. IGA is generally aimed at filling the gap between the modeling of geometries in industrial contexts, usually performed by means of Computer Aided Design (CAD) software, and the use of such geometries directly for analysis purposes, e.g. the numerical approximation of PDEs of interest. Ideally, the costly process of building an approximation of the geometry to be used as computational mesh suited for the numerical analysis would thus be avoided. In this thesis, we consider geometries represented by single-patch NURBS [36, 37]. By using NURBS mappings, a small amount of DOFs is required for the exact representation of several geometries of interest, e.g. conic sections. Moreover, geometrical features, such as curvatures and normal vectors, are evaluated exactly. This plays a fundamental role when dealing with mathematical models which rely on these quantities. While IGA-based discretizations provide several advantages over the standard FEM [35], this exactness of the geometrical representation provided by NURBS-based IGA is one of the major features with regard to the problems considered in this thesis. Moreover, we benefit also from the properties of the NURBS function spaces used for approximating the solution: by using NURBS basis functions, it is possible to control and tune the degree of continuity of the functions used for approximating the solution [35, 37]. With respect to the standard FEM, this allows to achieve approximations of particular accuracy with a low amount of DOFs [38, 39, 40, 41], as well as to discretize high order PDEs in the framework of the Galerkin method, without reverting to mixed formulations required by the standard Lagrangian isoparametric FEM [42].

While the approximation of second order surface PDEs by NURBS-based IGA has been extensively analyzed in [43], in this thesis we focus instead on higher order PDEs defined on surfaces. As computational domains, both open and closed surfaces are considered, all represented by single-patch NURBS mappings. Specific care is taken in the construction of periodic NURBS function spaces [44]: this allows the usage of basis functions meeting the wanted degree of continuity almost everywhere, even on closed surfaces. We consider the approximation of elliptic PDEs involving high order Laplace–Beltrami operators, specifically of the fourth and sixth order. We show the convergence rate of the errors, both on open and closed surfaces, and we study the corresponding eigenvalue problems. With the described approach, geometrical mappings presenting pointwise singularities, e.g. the poles when considering the sphere, does not affect the convergence of the solution, provided that the quadrature points used to evaluate the integrals are chosen not to lie in singular points. As further applications, we consider the numerical approximation of two high order phase field problems defined on surfaces: the Cahn–Hilliard equation [44, 45], of the fourth order, and the phase field crystal equation [46, 47], of the sixth order, both being nonlinear, time dependent PDEs.

The approaches employed for handling high order surface PDEs are then extended for dealing with the geometric PDEs, for which we propose their numerical approximation by means of NURBS-based IGA. In this context, the exactness of the geometrical representation is crucial. Conversely, when using FEM-based discretizations on approximated geometries, these quantities need to be approximated as well, reconstructed, or treated weakly [48, 49]. In particular, we consider the numerical approximation of two geometric PDEs: the mean curvature flow, derived from the minimization of the area, and the Willmore flow, for which the Willmore energy, i.e. the integral of the total mean curvature squared, is minimized. Being the geometry itself the unknown, these problems are highly nonlinear. In this regard, we propose a time discretization of the geometric PDEs based on the Backward Differentiation Formulas (BDF) [50, 51]. In our formulations the nonlinearities are treated semi-implicitly through extrapolation in time of the geometric terms, compatibly with the considered BDF scheme. This leads to a linear system to be solved at each time step.

One of the applications where geometric PDEs are extensively used is the modeling of lipid vesicles [49, 52, 53, 54]. A vesicle consists in a lipid membrane which is spontaneously formed when lipid molecules, consisting in a hydrophilic head group and hydrophobic hydrocarbon tails, are immersed in aqueous environment with certain levels of concentration and temperature [55]. Lipid biomembranes are ubiquitous in biology [56]. A noticeable example is represented by the red blood cell, a biomembrane consisting in a lipid double layer, with a cytoskeleton of spectrin proteins (which determines the elasticity properties of the cell) linked by filaments of actin, and filled with a suspension of hemoglobin [57]. Red blood cells are generally considered to be elastic or nearly visco-elastic and area-preserving membranes [57, 58, 59, 60, 61], whose default configuration resembles a biconcave disc. In humans, the thickness of the layers combined is less than 100 nm, while the diameter of the red blood cell is approximatively 6.2–8.2 µm [62]. This justifies the common assumption of treating red blood cells, and biomembranes in general, as surfaces embedded in the 3D space. Since the study of lipid bilayers is of particular interest in biology, several mathematical models have been proposed over the years. These models can be divided into two major classes: microscopic discrete molecular-based models and macroscopic continuum models. Within the first class are, for example, models based on spring networks [63, 64, 65], molecular dynamics [66], or multiscale techniques [67, 68, 69, 70]. A different approach, purely continuous in modeling the equilibrium shape of vesicles, consists in minimizing the bending or curvature energy, subject to appropriate volume and area constraints [58, 60], an approach supported also by [57, 71], where the problem is studied from a continuum mechanics prospective. Within this approach, the problem is formulated as a geometric PDE, with the addition of geometric constraints. Several methods for its numerical approximation have been proposed, for example based on Adaptive FEM [26, 49, 52], parameric FEM [54], FEM with suitable remeshing strategy [53], level sets [72,73], or phase field approaches [74,75,76].

In this thesis, we propose the numerical study of the equilibrium shapes of vesicles by means of NURBS-based IGA. In particular, we consider the spontaneous curvature model [55,60]. We follow the approach considered for the approximation of the Willmore flow problem and we extend the formulation by introducing the constraints on total area and volume, treated by means of Lagrange multipliers. As in the mean curvature and Willmore flow problems, we represent the vesicle as a NURBS surface and discretize the geometric PDE in space with IGA and in time with BDF schemes, in a semi-implicit formulation with extrapolation of the geometric quantities. We propose two methods for the enforcement of the geometric constraints. Firstly, we consider an iterative scheme in the outline of the one proposed in [28,52] and adapted to our discretization method. With this scheme, the area and volume are conserved up to a chosen tolerance. As an alternative, we propose to fulfill the constraints directly on the extrapolated geometry. This approach relies on the accurate representation of the geometry due to the IGA discretization with NURBS and is computationally less demanding than the first method.

While the equilibrium shapes of vesicles can be modeled by using curvature-based formulations, in order to study the dynamics of vesicles in situations of interest it is necessary to take into account the interaction of the membrane with both the outer [55] and the inner fluids [52]. Modeling the fluid-membrane interaction at cellular level is useful for understanding the rheology of dense suspensions of cells. Considering for example the blood, which is a suspension of red blood cells and other particles in the plasma, its rheology is significantly influenced by the biophysical characteristics of the biomembranes and changes with the vessel diameter; while in vessels larger than 200 µm blood can be modeled as a homogeneous fluid governed by the incompressible Navier–Stokes equations, in vessels with smaller diameter, especially when their size is comparable to that of the suspended cells (around 8 µm), it is essential to model the blood at cell-scale level [55]. As an example, when the vessels are very small, the cellular constitution of the blood is responsible for a significant drop of flow viscosity; this is known as the *Fåhræus–Lindqvist* effect [77] and is due to the tendency of red blood cells of flowing along the centerline of the capillary, leaving a thin layer of plasma near the walls free of cells, and thus lowering the effective viscosity of the blood when the thickness of the layer is significant with respect to the diameter of the capillary [78]. Moreover, several pathologies and diseases, as malaria or sickle-cell disease, are able to modify the shape of red blood cells, affecting their mechanics, and thus the overall flow behavior [70]. Over the years, a wide variety of numerical approaches have been employed for the numerical study of the biomembrane-fluid interaction; see e.g. [52,70,79,80,81,82]. In particular, the choices of geometrical representation and the discretization of the computational domains are important, as the biomembranes may be subject to large deformations and rigid movements. As examples, in [83,84,85] approaches based on the immersed boundary methods are considered; a phase field approach is used in [86]; level-set based geometrical descriptions of the vesicles are considered in [73,87,88]; a parametric mesh, following the vesicle and updated with the fluid velocity field, is employed in [79,89]; solid capsules, as representations for red blood and nucleated cells, are considered instead in [82].

In this respect, we extend in this thesis the numerical approach considered for studying the shapes of vesicles at equilibrium to model also the dynamical behavior of a biomembrane stemming from the coupling with the fluid. An incompressible Newtonian fluid, governed by the Navier–Stokes equations, is considered. The vesicle is subject both to an internal force, driven by the bending energy as in the static case, as well as to the forces exerted by the fluid. The fluid takes into account the presence of the biomembrane by means of an additional penalization term for the velocity in the Navier–Stokes momentum equation, based on the resistive method [90, 91, 92]. Both the fluid domain and the vesicle are represented by NURBS mappings. In this respect, we propose the construction of the signed distance field generated by a NURBS geometry, to obtain an implicit description of the immersed vesicle for applying the resistive method. The fluid and membrane equations are discretized in space with NURBS-based IGA and in time with BDF schemes; the fluid equations are stabilized by means of the SUPG stabilization [93, 94, 95]. The interaction between the fluid and the membrane equations is performed by considering a partitioned approach, based on a strongly coupled Dirichlet–Neumann scheme [96, 97], with under-relaxation of the displacements of the vesicle [98]. We then apply the described approach for the numerical study of a vesicle immersed in a fluid at different regimes.

isoGlib

All the numerical simulations and results presented in this thesis have been obtained by using a research code developed in collaboration with Luca Dedè. *isoGlib* is a C++ framework providing the tools for the numerical approximation, by means of NURBS-based IGA, of PDEs defined on curves, surfaces, and volumes, potentially involving high order differential operators. The modularity and object-oriented nature of the library allows the application of isoGlib to a variety of problems, ranging from cardiac electrophysiology [7], blood flow dynamics [95], and phase-field models [99], to fluid dynamics and fluid-structure interaction [100, 101], including the applications presented in this thesis. The generation of the geometries and preprocessing of the meshes are performed in Matlab, with the support of the NURBS toolbox [102]. MPI communications, as well as assembling and solution of the linear systems, including the application of preconditioners, are performed with the aid of *Trilinos* [103]. Post-processing of the results, including the ones presented in this thesis, is performed with *Paraview* [104].

Outline of the thesis

In Chapter 1 we introduce the basic concepts and notation regarding parametric surfaces and the surface differential operators; then, we introduce an abstract formulation for high order scalar elliptic PDEs, describing also the benchmark Laplace–Beltrami biharmonic, triharmonic, and eigenvalue problems. Two high order phase field models are then introduced, namely the Cahn–Hilliard and the phase field crystal equations.

In Chapter 2, after having introduced the notation required for dealing with evolving surfaces and shape differentials, we introduce the abstract formulation for geometric PDEs based on the gradient of shape energy functionals, with details about the mean curvature and the Willmore flow problems.

NURBS geometrical mappings, function spaces, and NURBS-based IGA are described in Chapter 3. The construction of periodic NURBS function spaces, for dealing with closed geometries, is also explained.

Chapter 4 deals with the numerical approximation of high order surface PDEs by means of NURBS-based IGA in the framework of the Galerkin method. We provide convergence results for different discretizations under h-refinement regarding the approximation of the high order Laplace–Beltrami problems introduced in Chapter 1, on open surfaces as well as closed surfaces, employing periodic NURBS function spaces; a comparison against a discretization based on the standard isoparametric FEM is also provided. We conclude the chapter with the numerical approximation of the Canh–Hilliard and phase field equations, providing results both on open and closed surfaces. The majority of the presented results have already been reported in our paper [105].

In Chapter 5, we numerically approximate the geometric PDEs introduced in Chapter 2. In particular, we propose weak formulations for the mean curvature and Willmore flows and we discretize the equations by means of NURBS-based IGA and BDF schemes in semi-implicit formulation, with extrapolation of the geometric quantities. We show results on several geometries, performing an analysis of the errors obtained, as well as the condition numbers. These results are also published in our work [106].

Chapter 6 is devoted to the numerical approximation of the equilibrium shapes of lipid vesicles. After an introduction to the spontaneous curvature model for lipid biomembranes, we provide a weak formulation for the problem and we discretize it with NURBS-based IGA and BDF schemes; we then describe the two proposed schemes for the enforcement of the geometric constraints. Results of the approximated geometric flow on different initial shapes are provided, the majority of which have been submitted in the paper [107].

In Chapter 7 we deal with the fluid-membrane interaction problem and vesicle dynamics. First, the Navier–Stokes equations are recalled, together with the resistive method employed to deal with immersed obstacles; the NURBS-based IGA discretization of the Navier–Stokes equations, with SUPG stabilization, is then described and results concerning the approximation of the fluid equations in two benchmark cases are reported. Then, we deal with the construction of a signed distance field associated with a NURBS curve or surface, providing a numerical example of fluid flow past a NURBS obstacle. Finally, we introduce the coupling between the fluid and the membrane and we present numerical results concerning a vesicle in parabolic and shear flows.

Finally, conclusions and possible future developments follow.

1 High order surface PDEs

Several mathematical models are formulated through Partial Differential Equations (PDEs) defined on lower dimensional manifolds [1]. Examples of three dimensional problems represented on surfaces can be found in Fluid Dynamics, Mechanics, Biology, Electromagnetism, and image processing [30, 31, 33, 52]: this is the case, for instance, for thin geometries, which are often modeled as membranes, plates, or shells [2], depending on the structure of the original domain. This leads to define surface PDEs, which often involve high order differential operators to account for the dimensionality reduction [108].

The main focus of this thesis is the numerical approximation of models yielding PDEs defined on bidimensional manifolds immersed in the three dimensional space, described through a geometrical *mapping* from a parametric domain to the physical domain. These manifolds are usually called *parametric surfaces* [109]. In this chapter, the notation for defining the geometrical mapping characterizing the parametric surfaces adopted throughout this thesis is introduced, together with the associated geometrical quantities and the differential operators on surfaces. Then, high order elliptic surface PDEs are introduced. In this regard, benchmark problems of fourth and sixth order are considered: in particular, the fourth order (biharmonic) Laplace–Beltrami problem, the sixth order (triharmonic) Laplace–Beltrami problem, known as the triharmonic problem, and the general high order Laplace–Beltrami eigenvalue problem, which can be defined with both the biharmonic or the triharmonic operators. Finally, extension to unsteady nonlinear problems is considered. In particular, two high order phase field models are introduced: the fourth order Cahn–Hilliard equation and the sixth order phase field crystal equation. These high order Laplace–Beltrami and phase field problems constitute a reference for the treatment of steady and unsteady high order surface PDEs; this framework is then used as a base for studying the geometric PDEs, which will be discussed in Chapter 2.

1.1 Geometrical representation of surfaces

Let the subset $\Omega \subset \mathbb{R}^3$ be a C^2-*hypersurface*, i.e. for each point $\tilde{\mathbf{p}} \in \Omega$ there exists an open set $U \in \mathbb{R}^3$, such that $\tilde{\mathbf{p}} \in U$, and a function $u \in C^2(U)$ such that [10]:

$$U \cap \Omega = \left\{ \mathbf{p} \in U \mid u(\mathbf{p}) = 0 \right\}, \qquad \text{and} \quad \nabla u(\mathbf{p}) \neq 0 \qquad \forall \mathbf{p} \in U \cap \Omega. \tag{1.1.1}$$

The tangent space of Ω in $\mathbf{p} \in \Omega$, denoted with $T_{\mathbf{p}}\Omega$, is the three-dimensional linear subspace of $\mathbb{R}^3$ orthogonal to $\nabla U(\mathbf{p})$. Then, the C^2-hypersurface Ω is *orientable* if there exists a vector field $\boldsymbol{\nu} \in C^1(\Omega, \mathbb{R}^3)$ (i.e. differentiable in an open neighborhood of Ω), such that $\boldsymbol{\nu} \perp T_{\mathbf{p}}\Omega$ and $|\boldsymbol{\nu}| = 1$, for all $\mathbf{p} \in \Omega$ [10].

Now, let $\widehat{\Omega} \subset \mathbb{R}^2$ be the *parametric domain*. The *parametric coordinate* is a vector-valued independent variable $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \mathbb{R}^2$, used to refer to the points in the parametric domain $\widehat{\Omega}$. Let $\Omega \subset \mathbb{R}^3$ be a compact, oriented C^2-hypersurface defined by means of a geometrical mapping $\mathbf{X}$ as:

$$\mathbf{X} : \widehat{\Omega} \to \Omega \subset \mathbb{R}^3, \qquad \boldsymbol{\xi} \to \mathbf{X}(\boldsymbol{\xi}) = \mathbf{p}. \tag{1.1.2}$$

The manifold Ω, defined through $\mathbf{X}$, represents a parametric surface in $\mathbb{R}^3$, which can be defined with or without boundary $\partial\Omega$. Indeed, if $\partial\Omega \equiv \emptyset$, then Ω is a *closed surface*; in the opposite case, for which $|\partial\Omega| > 0$, Ω is an *open surface*.

The geometrical mapping (1.1.2) is assumed to be sufficiently smooth, e.g. at least $C^1(\widehat{\Omega})$, and invertible almost everywhere (a.e.) in $\widehat{\Omega}$. The inverse mapping $\mathbf{X}^{-1}$ is denoted as:

$$\mathbf{X}^{-1} : \Omega \to \widehat{\Omega} \subset \mathbb{R}^2, \qquad \mathbf{p} \to \mathbf{X}^{-1}(\mathbf{p}) = \boldsymbol{\xi} \qquad \forall \mathbf{p} \in \Omega. \tag{1.1.3}$$

The geometrical mapping and its inverse permit to recast the quantities and functions defined on the parametric domain onto the manifold and viceversa. Indeed, any sufficiently regular function $\psi : \Omega \to \mathbb{R}$ defined on the surface Ω, e.g. $\psi \in C^0(\Omega)$, can be expressed in the parametric domain $\widehat{\Omega}$ by the corresponding function $\widehat{\psi} : \widehat{\Omega} \to \mathbb{R}$ defined as:

$$\widehat{\psi}(\boldsymbol{\xi}) := \psi(\mathbf{X}(\boldsymbol{\xi})) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}. \tag{1.1.4}$$

The operation performed in Eq. (1.1.4) is usually called *pull-back* of ψ from Ω to $\widehat{\Omega}$. Similarly, for any function $\widehat{\psi} : \widehat{\Omega} \to \mathbb{R}$ defined on the parametric domain $\widehat{\Omega}$, the corresponding function $\psi : \Omega \to \mathbb{R}$ on the manifold is written by means of the *push-forward* operation as:

$$\psi(\mathbf{p}) := \widehat{\psi}(\mathbf{X}^{-1}(\mathbf{p})) \qquad \forall \mathbf{p} \in \Omega. \tag{1.1.5}$$

In this thesis, because of the kind of surfaces considered (in particular parametric surfaces represented by NURBS geometrical mappings, described in Chapter 3), the pull-back and push-forward operations allow the description of the same quantities and functions both on the physical and the parametric domain, and the easy exchange between the two. The

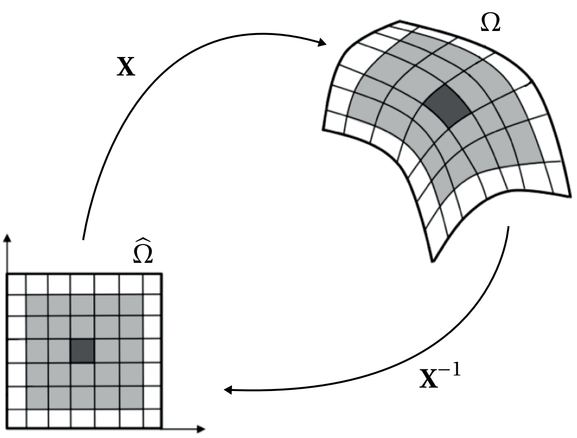


Figure 1.1 – Sketch of the physical domain Ω and the parametric domain $\widehat{\Omega}$, together with the geometrical mapping $\mathbf{X}$ and its inverse $\mathbf{X}^{-1}$.

parametric domain $\widehat{\Omega}$ and the domain Ω, together with the geometric mapping $\mathbf{X}$ and its inverse $\mathbf{X}^{-1}$, are sketched in Figure 1.1.

The Jacobian of the mapping $\widehat{F}:\widehat{\Omega}\to\mathbb{R}^{3\times 2}$ is defined as:

$$\widehat{F}_{i,\alpha}(\boldsymbol{\xi}) := \frac{\partial X_i}{\partial \xi_\alpha}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi}\in\widehat{\Omega}, \qquad \text{for } i=1,2,3 \text{ and } \alpha=1,2, \tag{1.1.6}$$

with $\alpha = 1,2$ indicating the parametric direction. The tangent directions $\hat{\mathbf{t}}_{\Omega,\alpha}(\boldsymbol{\xi})$, for $\alpha = 1,2$, represent the unit vectors tangent to the surface Ω in $\mathbf{p}=\mathbf{X}(\boldsymbol{\xi})$:

$$\hat{\mathbf{t}}_{\Omega,\alpha}(\boldsymbol{\xi}) := \frac{\partial \mathbf{X}}{\partial \xi_\alpha}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi}\in\widehat{\Omega}, \qquad \text{for } \alpha=1,2; \tag{1.1.7}$$

then, $\mathbf{t}_{\Omega,\alpha}$ is obtained as the push-forward of $\hat{\mathbf{t}}_{\Omega,\alpha}$ from $\widehat{\Omega}$ onto Ω, for $\alpha = 1,2$. The unit vector $\mathbf{n}_\Omega(\mathbf{p})$ normal to the surface in $\mathbf{p}\in\Omega$ is the push-forward of the normal in the parametric domain $\hat{\mathbf{n}}_\Omega(\boldsymbol{\xi})$, with $\boldsymbol{\xi}=\mathbf{X}^{-1}(\mathbf{p})$, which is defined as:

$$\hat{\mathbf{n}}_\Omega(\boldsymbol{\xi}) := \frac{\hat{\mathbf{t}}_{\Omega,1}(\boldsymbol{\xi})\times\hat{\mathbf{t}}_{\Omega,2}(\boldsymbol{\xi})}{|\hat{\mathbf{t}}_{\Omega,1}(\boldsymbol{\xi})\times\hat{\mathbf{t}}_{\Omega,2}(\boldsymbol{\xi})|}; \tag{1.1.8}$$

note that $\mathbf{n}_\Omega$ represents the normal to the surface Ω and not its boundary $\partial\Omega$. Together, $\mathbf{t}_{\Omega,1}(\mathbf{p})$, $\mathbf{t}_{\Omega,2}(\mathbf{p})$, and $\mathbf{n}_\Omega(\mathbf{p})$ form a covariant basis at the surface point $\mathbf{p}\in\Omega$.

The first fundamental form, or metric tensor, is the symmetric and positive definite second order tensor $\widehat{G}:\widehat{\Omega}\to\mathbb{R}^{2\times 2}$, $\boldsymbol{\xi}\to\widehat{G}(\boldsymbol{\xi})$, defined as:

$$\widehat{G}(\boldsymbol{\xi}) := \left(\widehat{F}(\boldsymbol{\xi})\right)^T \widehat{F}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi}\in\widehat{\Omega}. \tag{1.1.9}$$

The determinant of the mapping $\widehat{g}:\widehat{\Omega}\to\mathbb{R}$, $\boldsymbol{\xi}\to\widehat{g}(\boldsymbol{\xi})$, corresponds to the area of the parallelogram lying on the tangent plane defined by $\hat{\mathbf{t}}_{\Omega,1}$ and $\hat{\mathbf{t}}_{\Omega,2}$:

$$\widehat{g}(\boldsymbol{\xi}) := |\hat{\mathbf{t}}_{\Omega,1} \times \hat{\mathbf{t}}_{\Omega,2}| \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega} \tag{1.1.10}$$

and it is such that

$$\widehat{g}(\boldsymbol{\xi}) = \sqrt{\det\left(\widehat{G}(\boldsymbol{\xi})\right)} \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}. \tag{1.1.11}$$

Assuming the invertibility of the geometrical mapping a.e. in $\widehat{\Omega}$ means that the determinant $\widehat{g}$ is allowed to be zero only in subsets of $\widehat{\Omega}$ with zero measure in the topology of $\mathbb{R}^2$, thus requiring $\widehat{g}$ to be positive elsewhere. The determinant $\widehat{g}$ permits also the definition of the notion of integration over the surface Ω: considering a function $\psi : \Omega \to \mathbb{R}$, with corresponding pull-back function $\widehat{\psi} : \widehat{\Omega} \to \mathbb{R}$ defined through Eq. (1.1.4), the integral of ψ over Ω is written as:

$$\int_{\Omega} \psi \, d\Omega := \int_{\widehat{\Omega}} \widehat{\psi} \, \widehat{g} \, d\widehat{\Omega}. \tag{1.1.12}$$

Finally, by exploiting the invertibility of the geometrical mapping, the geometric quantities $\widehat{F}$, $\widehat{G}$ and $\widehat{g}$ can be pushed-forward and expressed directly on the manifold Ω as:

$$\begin{aligned} &F : \Omega \to \mathbb{R}^{3\times 2}, && F(\mathbf{p}) := \left(\widehat{F} \circ \mathbf{X}^{-1}\right)(\mathbf{p}) && \forall \mathbf{p} \in \Omega, \\ &G : \Omega \to \mathbb{R}^{2\times 2}, && G(\mathbf{p}) := \left(\widehat{G} \circ \mathbf{X}^{-1}\right)(\mathbf{p}) && \forall \mathbf{p} \in \Omega, \\ &g : \Omega \to \mathbb{R}, && g(\mathbf{p}) := \left(\widehat{g} \circ \mathbf{X}^{-1}\right)(\mathbf{p}) && \forall \mathbf{p} \in \Omega. \end{aligned} \tag{1.1.13}$$

1.1.1 Surface differential operators

In order to define PDEs on the surface Ω it is necessary to define a set of differential operators on the manifold. Thanks to the invertibility of the geometrical mapping (1.1.2), surface differential operators can be defined as projection onto the surface of the corresponding differential operators defined in the physical space [1,28,43,110]. Let now $\phi \in C^1(\Omega)$ be a generic function, together with its smooth prolongation $\tilde{\phi}(\mathbf{x})$ from Ω into a "tubular" neighborhood $\widetilde{\Omega} \subset \mathbb{R}^3$, which is such that $\Omega \subset \widetilde{\Omega}$. It is possible to define the projection tensor $\mathrm{P}(\mathbf{p}) \in \mathbb{R}^{3\times 3}$ as:

$$\mathrm{P}(\mathbf{p}) := \mathbf{I} - \mathbf{n}_{\Omega}(\mathbf{p}) \otimes \mathbf{n}_{\Omega}(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.14}$$

where $\mathbf{I}$ is the identity tensor in $\mathbb{R}^{3\times 3}$. Then, the surface gradient ∇_{Ω} is defined as the projection of the standard gradient onto the manifold, reading:

$$\nabla_{\Omega}\phi(\mathbf{p}) := \mathrm{P}\left[\nabla\tilde{\phi}(\mathbf{p})\right] = \nabla\tilde{\phi}(\mathbf{p}) - \left[\nabla\tilde{\phi}(\mathbf{p}) \cdot \mathbf{n}_{\Omega}(\mathbf{p})\right]\mathbf{n}_{\Omega}(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega; \tag{1.1.15}$$

indeed, $\nabla_{\Omega}\psi(\mathbf{p})$ is the orthogonal projection of $\nabla\psi$ onto $T_{\mathbf{p}}\Omega$. The surface gradient defined in Eq. (1.1.15) is equivalently expressed using the geometrical mapping (1.1.2) and Eqs. (1.1.6) and (1.1.9) as:

$$\nabla_\Omega \phi(\mathbf{p}) = \left[\widehat{F}(\boldsymbol{\xi})\,\widehat{G}^{-1}(\boldsymbol{\xi})\,\widehat{\nabla}\widehat{\phi}(\boldsymbol{\xi})\right] \circ \mathbf{X}^{-1}(\boldsymbol{\xi}) \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.16}$$

where $\widehat{\nabla}\widehat{\phi}(\boldsymbol{\xi}) : \widehat{\Omega} \to \mathbb{R}^2$ is the gradient operator in the parameter domain. Similarly, by considering a vector field $\boldsymbol{\varphi} \in \left[C^1(\Omega)\right]^3$, the surface divergence of $\boldsymbol{\varphi}$ is defined as:

$$\nabla_\Omega \cdot \boldsymbol{\varphi}(\mathbf{p}) := \operatorname{trace}\left[\nabla_\Omega \boldsymbol{\varphi}(\mathbf{p})\right] \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.17}$$

which is rewritten using Eqs. (1.1.6), (1.1.9), and (1.1.10) as:

$$\nabla_\Omega \cdot \boldsymbol{\varphi}(\mathbf{p}) = \left[\frac{1}{\widehat{g}(\boldsymbol{\xi})}\widehat{\nabla} \cdot \left(\widehat{g}(\boldsymbol{\xi})\,\widehat{G}^{-1}(\boldsymbol{\xi})\,\widehat{F}^T(\boldsymbol{\xi})\,\widehat{\boldsymbol{\varphi}}(\boldsymbol{\xi})\right)\right] \circ \mathbf{X}^{-1}(\boldsymbol{\xi}) \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.18}$$

where $\widehat{\boldsymbol{\varphi}}(\boldsymbol{\xi}) := \boldsymbol{\varphi}(\mathbf{X}(\boldsymbol{\xi}))$, $\forall \boldsymbol{\xi} \in \widehat{\Omega}$. By considering a function $\phi \in C^2(\Omega)$, the Laplace–Beltrami operator on the manifold Ω can be defined by composing the surface divergence operator (1.1.17) with the surface gradient operator (1.1.15) as [1]:

$$\Delta_\Omega \phi(\mathbf{p}) := \nabla_\Omega \cdot \left(\nabla_\Omega \phi(\mathbf{p})\right) \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.19}$$

which, by using Eqs. (1.1.16) and (1.1.18), reads:

$$\Delta_\Omega \phi(\mathbf{p}) = \left[\frac{1}{\widehat{g}(\boldsymbol{\xi})}\widehat{\nabla} \cdot \left(\widehat{g}(\boldsymbol{\xi})\,\widehat{G}^{-1}(\boldsymbol{\xi})\,\widehat{\nabla}\widehat{\phi}(\boldsymbol{\xi})\right)\right] \circ \mathbf{X}^{-1}(\boldsymbol{\xi}) \qquad \forall \mathbf{p} \in \Omega. \tag{1.1.20}$$

Finally, by considering a general function $\phi \in C^3(\Omega)$, the gradient of the Laplace–Beltrami operator $\nabla_\Omega \Delta_\Omega \phi$, expressed by using Eqs. (1.1.16) and (1.1.20), reads:

$$\nabla_\Omega \Delta_\Omega \phi(\mathbf{p}) = \left[\widehat{F}(\boldsymbol{\xi})\,\widehat{G}^{-1}(\boldsymbol{\xi})\,\widehat{\nabla}\left(\frac{1}{\widehat{g}(\boldsymbol{\xi})}\widehat{\nabla} \cdot \left[\widehat{g}(\boldsymbol{\xi})\,\widehat{G}^{-1}(\boldsymbol{\xi})\,\widehat{\nabla}\widehat{\phi}(\boldsymbol{\xi})\right]\right)\right] \circ \mathbf{X}^{-1}(\boldsymbol{\xi}) \qquad \forall \mathbf{p} \in \Omega. \tag{1.1.21}$$

1.1.2 Identity function and surface curvatures

Geometric PDEs formulations, which will be introduced in Chapter 2, make extensively use of the definition of curvatures of the surface. With this aim, the second fundamental form $\mathscr{H}$ associated to the geometric mapping (1.1.2) is introduced as:

$$\mathscr{H}(\mathbf{p}) := \nabla_\Omega \mathbf{n}_\Omega(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.22}$$

which is also called *shape operator* [110]. Since Ω is a surface in $\mathbb{R}^3$, the shape operator $\mathscr{H}$ is a second order tensor which possesses a null eigenvalue associated to the eigenvector along the normal to the surface $\mathbf{n}_\Omega$ and two other non zero eigenvalues. In each $\mathbf{p} \in \Omega$, the two non zero eigenvalues of $\mathscr{H}(\mathbf{p})$ are called *principal curvatures* and are denoted in this thesis with $\kappa_\alpha(\mathbf{p})$, for $\alpha = 1, 2$. The *total mean curvature* H is then defined as the sum of the principal curvatures:

$$H(\mathbf{p}) := \kappa_1(\mathbf{p}) + \kappa_2(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.23}$$

while the *Gauss curvature* K is their product:

$$K(\mathbf{p}) := \kappa_1(\mathbf{p})\kappa_2(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega. \tag{1.1.24}$$

In this work, the normal $\mathbf{n}_\Omega$ introduced in Eq. (1.1.8) is considered to be oriented such that H is positive for spherical surfaces with the normal directed outward with respect to the origin. Furthermore, the *total mean curvature vector* $\mathbf{H}$ is defined as:

$$\mathbf{H}(\mathbf{p}) := H(\mathbf{p})\mathbf{n}_\Omega(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega. \tag{1.1.25}$$

It is important to highlight that the definition of the mean curvature in Eq. (1.1.23) which is used in this work differs from other common definitions in the literature. Sometimes with *mean curvature* one means the average of k_1 and k_2, i.e. $(k_1 + k_2)/2$, which differs from the definition (1.1.23) by the factor $1/2$; this is, however, just a change of notation. Moreover, from Eqs. (1.1.22), (1.1.23), and (1.1.24), the following relation holds:

$$|\mathscr{H}(\mathbf{p})|^2 = \kappa_1^2(\mathbf{p}) + \kappa_2^2(\mathbf{p}) = H^2(\mathbf{p}) - 2K(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega. \tag{1.1.26}$$

Let $\mathbf{x} : \Omega \to \mathbb{R}^3$ be the identity function on Ω, reading:

$$\mathbf{x}(\mathbf{X}(\boldsymbol{\xi})) = \mathbf{X}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}. \tag{1.1.27}$$

This map is convenient from the point of view of the notation when formulating equations defined directly on the physical surface. In the rest of this work, the surface Ω is considered to be equivalently identified either by its geometrical mapping $\mathbf{X}$ or by the identity function $\mathbf{x}$. A useful relation which links the surface Laplace–Beltrami operator applied to the identity function $\mathbf{x}$ on Ω to the total mean curvature vector $\mathbf{H}$ is the following [110]:

$$(-\Delta_\Omega \mathbf{x})(\mathbf{p}) = \mathbf{H}(\mathbf{p}) \qquad \forall \mathbf{p} \in \Omega, \tag{1.1.28}$$

which will be used extensively in this work.

When integrating on the surface Ω, there is a set of integration by parts formulae that we summarize in the following theorems [1, 10].

Theorem 1. *Consider a C^1 scalar function $\psi : \Omega \to \mathbb{R}$; then:*

$$\int_\Omega \nabla_\Omega \psi \, d\Omega = \int_\Omega \psi \, (\nabla_\Omega \cdot \mathbf{n}_\Omega)\mathbf{n}_\Omega \, d\Omega + \int_{\partial\Omega} \psi (\boldsymbol{\tau}_\Omega \times \mathbf{n}_\Omega) d\Omega, \tag{1.1.29}$$

where the unit tangent $\boldsymbol{\tau}_\Omega$ on $\partial\Omega$ is such that $\boldsymbol{\tau}_\Omega \times \mathbf{n}_\Omega$ points outside of Ω.

Theorem 2. *Consider a C^1 vector-valued function $\boldsymbol{\varphi} : \Omega \to \mathbb{R}^3$; the following divergence theorem holds:*

$$\int_\Omega \nabla_\Omega \cdot \boldsymbol{\varphi} \, d\Omega = \int_\Omega H \boldsymbol{\varphi} \cdot \mathbf{n}_\Omega \, d\Omega + \int_{\partial\Omega} \boldsymbol{\varphi} \cdot (\boldsymbol{\tau}_\Omega \times \mathbf{n}_\Omega) d\Omega. \tag{1.1.30}$$

Theorem 3. *Consider a C^1 scalar function $\psi : \Omega \to \mathbb{R}$ and the C^1 vector-valued function $\boldsymbol{\varphi} : \Omega \to \mathbb{R}^3$; then:*

$$\int_\Omega \nabla_\Omega \psi \cdot \boldsymbol{\varphi}\, d\Omega = -\int_\Omega \psi \nabla_\Omega \cdot \boldsymbol{\varphi}\, d\Omega + \int_\Omega H \psi \boldsymbol{\varphi} \cdot \mathbf{n}_\Omega\, d\Omega + \int_{\partial\Omega} \psi \boldsymbol{\varphi} \cdot (\boldsymbol{\tau}_\Omega \times \mathbf{n}_\Omega) d\Omega. \tag{1.1.31}$$

Proofs of Theorems 1, 2 and 3 are reported in [1]. Finally, as an example of useful application of Theorem 3 for integration by parts, let $\psi_1, \psi_2 : \Omega \to \mathbb{R}$ be C^2 scalar or vector-valued functions; then:

$$-\int_\Omega \Delta_\Omega \psi_1 \psi_2\, d\Omega = \int_\Omega \nabla_\Omega \psi_1 \cdot \nabla_\Omega \psi_2\, d\Omega - \int_{\partial\Omega} \psi_2 \nabla_\Omega \psi_1 \cdot (\boldsymbol{\tau}_\Omega \times \mathbf{n}_\Omega) d\Omega, \tag{1.1.32}$$

which is the usual rule used for integrating by part in the weak formulation of the Laplace–Beltrami operator. Furthermore, by applying Eq. (1.1.32) to the Laplace–Beltrami of the identity map **x** and following the relation in Eq. (1.1.28), the following holds, on closed surfaces:

$$\int_\Omega \nabla_\Omega \mathbf{x} \cdot \nabla_\Omega \boldsymbol{\varphi}\, d\Omega = \int_\Omega H \boldsymbol{\varphi} \cdot \mathbf{n}_\Omega\, d\Omega, \tag{1.1.33}$$

for generic C^1 functions $\boldsymbol{\varphi} : \Omega \to \mathbb{R}^3$. This results is very important for the weak formulation of geometric PDEs and it will be used extensively in the related chapters of this work.

1.2 High order surface PDEs

Let $\Omega \subset \mathbb{R}^3$ be a parametric surface described by the geometric mapping $\mathbf{X}$ defined in Eq. (1.1.2). A generic scalar elliptic PDE of order $2m$, with $m = 1, 2$, or 3, can be expressed in weak form as:

$$\begin{aligned} &\text{find } u \in V_0 \text{ such that} \\ &\qquad a(\psi, u) = F(\psi) \qquad \forall \psi \in V_0, \end{aligned} \tag{1.2.1}$$

where $a : V_0 \times V_0 \to \mathbb{R}$ is a continuous bilinear form, strongly coercive on $V_0 \times V_0$, associated to one of the surface differential operators (1.1.16), (1.1.20), or (1.1.21), $F : V_0 \to \mathbb{R}$ is a continuous linear functional, and V_0 is a suitable Hilbert space, subspace of $H^m(\Omega)$ and yielding the homogeneous counterpart of the essential boundary conditions. In particular, the function space $H^m_0(\Omega)$ is defined as [111]:

$$H^m_0(\Omega) := \left\{\psi : \Omega \to \mathbb{R}, \, \psi \in H^m(\Omega), \, \Gamma_0 \psi = \ldots = \Gamma_{m-1}\psi = 0 \text{ on } \partial\Omega\right\}, \tag{1.2.2}$$

where Γ_k denotes the trace operator of order k, for $k = 0, 1, \ldots, m-1$. Thanks to the Lax–Milgram Lemma [112], the solution of Eq. (1.2.1) exists and is unique. If the domain Ω is a closed surface, Ω does not possess boundary and thus $V_0 \equiv V$, with $V \equiv H^m(\Omega)$; in this case, the weak formulation of a generic scalar elliptic PDE of order $2m$ reads:

$$\begin{aligned} &\text{find } u \in V \text{ such that} \\ &\qquad a(\psi, u) = F(\psi) \qquad \forall \psi \in V, \end{aligned} \tag{1.2.3}$$

where it is assumed that the form $a : V \times V \to \mathbb{R}$ carries a zero-th order linear operator corresponding to a reaction term, which makes $a(\cdot, \cdot)$ strongly coercive in V.

By recalling the invertibility of the geometric mapping (1.1.2), problems (1.2.1) and (1.2.3) are pulled-back into the parametric domain $\widehat{\Omega}$ by exploiting the differential operators (1.1.16), (1.1.18), (1.1.20), and (1.1.21). Problems (1.2.1) and (1.2.3) written into the parametric domain $\widehat{\Omega}$ read:

$$\begin{aligned} &\text{find } \widehat{u} \in \widehat{V}_0 \text{ such that} \\ &\qquad \widehat{a}(\widehat{\psi}, \widehat{u}) = \widehat{F}(\widehat{\psi}) \qquad \forall \widehat{\psi} \in \widehat{V}_0 \end{aligned} \tag{1.2.4}$$

and

$$\begin{aligned} &\text{find } \widehat{u} \in \widehat{V} \text{ such that} \\ &\qquad \widehat{a}(\widehat{\psi}, \widehat{u}) = \widehat{F}(\widehat{\psi}) \qquad \forall \widehat{\psi} \in \widehat{V}, \end{aligned} \tag{1.2.5}$$

respectively, where $\widehat{V}_0$ and $\widehat{V}$ correspond to the spaces V_0 and V defined over the parametric domain $\widehat{\Omega}$.

1.2.1 Laplace–Beltrami biharmonic problem

Let the domain Ω be sufficiently smooth. The *biharmonic operator* $\Delta^2_\Omega\cdot$ is the fourth order Laplace–Beltrami differential operator on the surface Ω defined as $\Delta^2_\Omega\cdot := \Delta_\Omega\,\Delta_\Omega\cdot$. The biharmonic problem on Ω with homogeneous essential boundary conditions reads:

$$\begin{aligned}&\text{find } u : \Omega \to \mathbb{R} \text{ such that}\\ &\qquad\begin{cases}\mu\Delta^2_\Omega u + \gamma u = f & \text{in } \Omega,\\ u = 0 & \text{on } \partial\Omega,\\ \nabla_\Omega u \cdot \mathbf{n}_{\partial\Omega} = 0 & \text{on } \partial\Omega,\end{cases}\end{aligned} \tag{1.2.6}$$

where μ and $\gamma \in \mathbb{R}$ are positive constants, $\mathbf{n}_{\partial\Omega}$ is the outward directed unit vector normal to the boundary $\partial\Omega$, and f is a sufficiently regular function. If the domain Ω is a closed surface, problem (1.2.6) reduces to:

$$\begin{aligned}&\text{find } u : \Omega \to \mathbb{R} \text{ such that}\\ &\qquad\mu\Delta^2_\Omega u + \gamma u = f \qquad \text{in } \Omega.\end{aligned} \tag{1.2.7}$$

Problems (1.2.6) and (1.2.7) read in weak formulation as in Eqs. (1.2.1) and (1.2.3), for which:

$$a(\psi, u) := \int_\Omega \mu\,\Delta_\Omega u\,\Delta_\Omega \psi\, d\Omega + \int_\Omega \gamma\, u\,\psi\, d\Omega, \tag{1.2.8}$$

and

$$F(\psi) := \int_\Omega f\,\psi\, d\Omega, \tag{1.2.9}$$

with $V_0 \equiv H^2_0(\Omega)$, defined in Eq. (1.2.2) for $m = 2$, and $V \equiv H^2(\Omega)$. Problem (1.2.6) is well posed with $\gamma = 0$, while problem (1.2.7) necessitates $\gamma > 0$. The problems are then recast into the parametric domain $\widehat{\Omega}$ as in Eqs. (1.2.4) and (1.2.5), with the bilinear form $\widehat{a}(\cdot,\cdot)$ and linear operator $\widehat{F}(\cdot)$ obtained by pulling-back $a(\cdot,\cdot)$ and $F(\cdot)$ into the parametric domain $\widehat{\Omega}$ as:

$$\widehat{a}(\widehat{\psi}, \widehat{u}) := \int_{\widehat{\Omega}} \mu\,\frac{1}{\widehat{g}}\widehat{\nabla}\cdot\left(\widehat{g}\,\widehat{G}^{-1}\widehat{\nabla}\widehat{u}\right)\widehat{\nabla}\cdot\left(\widehat{g}\,\widehat{G}^{-1}\widehat{\nabla}\widehat{\psi}\right) d\widehat{\Omega} + \int_{\widehat{\Omega}} \gamma\,\widehat{u}\,\widehat{\psi}\,\widehat{g}\, d\widehat{\Omega}, \tag{1.2.10}$$

and

$$\widehat{F}(\widehat{\psi}) := \int_{\widehat{\Omega}} \widehat{f}\,\widehat{\psi}\,\widehat{g}\, d\widehat{\Omega}, \tag{1.2.11}$$

respectively.

1.2.2 Laplace–Beltrami triharmonic problem

The *triharmonic operator* $\Delta^3_\Omega\cdot$ is the sixth order Laplace–Beltrami differential operator on the surface Ω, defined as $\Delta^3_\Omega\cdot := \Delta_\Omega\,\Delta_\Omega\,\Delta_\Omega\cdot$. Then, the triharmonic problem on Ω with homogeneous essential boundary conditions reads:

$$\text{find } u : \Omega \to \mathbb{R} \text{ such that} \\ \begin{cases} -\mu\Delta_\Omega^3 u + \gamma u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \\ \nabla_\Omega u \cdot \mathbf{n}_{\partial\Omega} = 0 & \text{on } \partial\Omega, \\ \Delta_\Omega u = 0 & \text{on } \partial\Omega, \end{cases} \tag{1.2.12}$$

where μ and $\gamma \in \mathbb{R}$ are positive constants and f is a sufficiently regular function. When considering closed surface domains, the problem becomes:

$$\text{find } u : \Omega \to \mathbb{R} \text{ such that} \\ -\mu\Delta_\Omega^3 u + \gamma u = f \qquad \text{in } \Omega. \tag{1.2.13}$$

Problems (1.2.12) and (1.2.13) read in weak form as in Eqs. (1.2.1) and (1.2.3), being:

$$a(\psi, u) = \int_\Omega \mu \nabla_\Omega (\Delta_\Omega u) \cdot \nabla_\Omega \left(\Delta_\Omega \psi\right) d\Omega + \int_\Omega \gamma\, u \psi \, d\Omega \tag{1.2.14}$$

and

$$F(\psi) = \int_\Omega f \, \psi \, d\Omega, \tag{1.2.15}$$

respectively, with $V_0 \equiv H_0^3(\Omega)$, defined in Eq. (1.2.2) for $m = 3$, and $V \equiv H^3(\Omega)$. Problem (1.2.12) is well posed for $\gamma = 0$, while problem (1.2.13) requires $\gamma > 0$. When recast into the parametric domain $\widehat{\Omega}$, the weak formulations of problems (1.2.12) and (1.2.13) read as in Eqs. (1.2.4) and (1.2.5), with the bilinear form $\widehat{a}(\cdot,\cdot)$ and linear operator $\widehat{F}(\cdot)$ defined as:

$$\widehat{a}(\widehat{\psi}, \widehat{u}) := \int_{\widehat{\Omega}} \mu \widehat{\nabla} \left[\frac{1}{\widehat{g}} \widehat{\nabla} \cdot \left(\widehat{g}\, \widehat{G}^{-1} \widehat{\nabla} \widehat{u} \right) \right] \cdot \left\{ \widehat{G}^{-1} \widehat{\nabla} \left[\frac{1}{\widehat{g}} \widehat{\nabla} \cdot \left(\widehat{g}\, \widehat{G}^{-1} \widehat{\nabla} \widehat{\psi} \right) \right] \right\} \widehat{g}\, d\widehat{\Omega} + \int_{\widehat{\Omega}} \gamma\, \widehat{u}\, \widehat{\psi}\, \widehat{g}\, d\widehat{\Omega} \tag{1.2.16}$$

and

$$\widehat{F}(\psi) := \int_{\widehat{\Omega}} \widehat{f}\, \widehat{\psi}\, \widehat{g}\, d\widehat{\Omega}, \tag{1.2.17}$$

respectively.

1.2.3 High order Laplace–Beltrami eigenvalue problems

The Laplace–Beltrami eigenvalue problem defined on a closed surface Ω, as e.g. the sphere, is defined as:

$$\text{find } u \in V \text{ and } \lambda \in \mathbb{R} \text{ such that} \\ a(\psi, u) = \lambda\, b(\psi, u) \qquad \forall \psi \in V, \tag{1.2.18}$$

where $a(\cdot,\cdot)$ is the bilinear form associated with either the biharmonic problem (1.2.8) or the triharmonic problem (1.2.14) for $\gamma = 0$, while $b : V \times V \to \mathbb{R}$ is the bilinear form representing the mass term defined as:

$$b(\psi, u) := \int_\Omega u\,\psi\, d\Omega, \tag{1.2.19}$$

with $V \equiv H^m(\Omega)$. Due to the symmetry of the problem, all the eigenvalues λ are expected to be real valued and non negative, i.e. $\lambda \in \mathbb{R}$, $\lambda \geq 0$.

Problem (1.2.18) can be recast into the parametric domain $\widehat{\Omega}$ as:

$$\begin{aligned} &\text{find } \widehat{u} \in \widehat{V} \text{ and } \lambda \in \mathbb{R} \text{ such that} \\ &\qquad \widehat{a}(\widehat{\psi}, \widehat{u}) = \lambda\, \widehat{b}(\widehat{\psi}, \widehat{u}) \qquad \forall \widehat{\psi} \in \widehat{V}, \end{aligned} \tag{1.2.20}$$

where $\widehat{V}$ is the function space associated to V over the parametric domain $\widehat{\Omega}$ and $\widehat{a}(\cdot,\cdot)$ corresponds to the bilinear form $a(\cdot,\cdot)$ pulled-back into the parametric domain, thus being either (1.2.10), for the biharmonic problem, or (1.2.16), for the triharmonic problem, for $\gamma = 0$. The bilinear form $\widehat{b} : \widehat{V} \times \widehat{V} \to \mathbb{R}$ reads:

$$\widehat{b}(\widehat{\psi}, \widehat{u}) := \int_{\widehat{\Omega}} \widehat{u}\,\widehat{\psi}\,\widehat{g}\, d\widehat{\Omega}. \tag{1.2.21}$$

1.3 Phase field models

Phase field models [113] are mathematical models describing the evolution of mixtures of phases in a domain Ω together with their mutual interaction at the interfaces. This is possible through the introduction of a scalar field $u : \Omega \to \mathbb{R}$, called *phase field*, which indicates the presence of the phases. In the case of two phases, to indicate the presence of the first phase or the second phase in the domain the field can assume, for instance, the pair of values $u_1 = 0$ and $u_2 = 1$, respectively. Between the phases, the phase field smoothly changes value, generating a smooth interface between them. The equations governing the phase field usually contain some parameters which tune the diffusivity of the interface and hence its width; in the limit of no diffusivity, the correct sharp interface should be recovered. Therefore, these models permit the approximation of evolving interfaces without the need of front tracking techniques.

In this thesis, both steady and time-dependent high order surface PDEs are considered. Steady benchmark problems have been introduced in Section 1.2. In addition to those, as examples of time-dependent high order surface PDEs, two phase field models based on the minimization of a free energy functional through gradient flow are also considered: the fourth order Cahn–Hilliard equation, described in Section 1.3.1, and the sixth order crystal equation, described in Section 1.3.2.

1.3.1 Cahn–Hilliard equation

Spinodal decomposition is the process of phase separation undergone by a quenched homogeneous fluid mixture [114]. The Cahn–Hilliard equation is a stiff, nonlinear, fourth order parabolic equation which describes spinodal decomposition for a binary fluid [44, 45]. Let $\Omega \subset \mathbb{R}^3$ be an arbitrary surface domain. The binary fluid mixture lays in Ω and the concentration of one of its components is denoted by $u = u(x, t) : \Omega \times (0, T) \to [0, 1]$; the concentration of the other phase is then $1 - u$. The Cahn–Hilliard equation models the dissipation of the Ginzburg–Landau free energy $\mathscr{G}$ within a mass conservative system [115]:

$$\mathscr{G}(u) = \mathscr{G}_c(u) + \mathscr{G}_s(u), \tag{1.3.1}$$

where $\mathscr{G}_c$ refers to the chemical free energy (bulk energy) and $\mathscr{G}_s$ is the surface free energy. The chemical free energy describes the immiscibility of the mixture's components and, as far as isothermal binary mixtures are concerned, it can be written as:

$$\mathscr{G}_c(u) := \frac{1}{2\theta}\left[u \log(u) + (1-u)\log(1-u)\right] + u(1-u), \tag{1.3.2}$$

where θ is the ratio between the critical and absolute temperatures [115]; in this work, it is always set to $\theta = 3/2$ (which refers to a physically relevant case [45, 116]), for which $\mathscr{G}_c$ assumes the form of a double well in the variable u, as shown in Figure 1.2. The minimization of the chemical free energy $\mathscr{G}_c$ leads to the separation of the phases. The surface free energy describes instead the attractive long-range interactions between the molecules of the binary

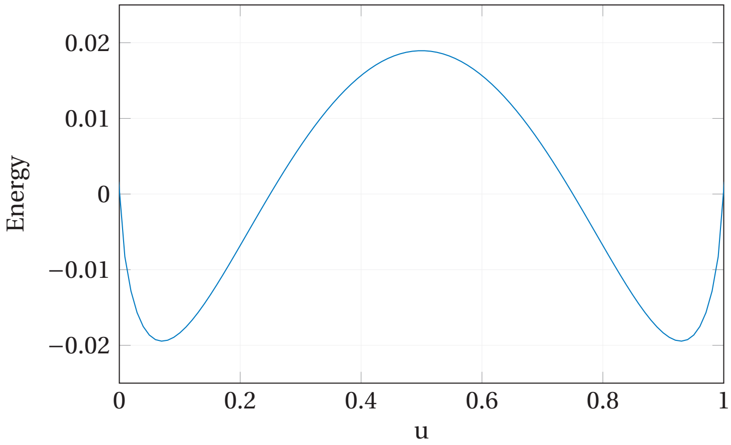


Figure 1.2 – Chemical free energy $\mathscr{G}_c$ in function of u, with $\theta = 3/2$. The energy forms a double well, whose minimization favors the separation of the phases.

mixture [44] and reads:

$$\mathscr{G}_s(u) := \frac{1}{2}\lambda \nabla_\Omega u \cdot \nabla_\Omega u, \tag{1.3.3}$$

where the differential operators are defined on the surface Ω and the parameter λ characterizes the interface thickness between the phases, which is proportional to $\sqrt{\lambda}$. The minimization of the surface free energy leads to the coarsening of the phases; in fact, solutions u of the Cahn–Hilliard equation for which $\mathscr{G}(u)$ is minimum correspond to the solutions of the isoperimetric problem [117].

In order to minimize the free energy $\mathscr{G}$ while maintaining the constraint of mass conservation, the H^{-1} gradient flow of $\mathscr{G}$ on the surface Ω is considered, as in [113]. By assuming suitable boundary conditions (or simply $\partial\Omega \equiv \emptyset$), the Fréchet derivative [118] of the free energy reads:

$$\frac{\delta \mathscr{G}}{\delta u} = \mu_u - \lambda \Delta_\Omega u, \tag{1.3.4}$$

where μ_u is the chemical potential, corresponding to:

$$\mu_u := \frac{d\mathscr{G}_c}{du} = \frac{1}{2\theta}\log\left(\frac{u}{1-u}\right) + 1 - 2u. \tag{1.3.5}$$

Then, the H^{-1} gradient flow of $\mathscr{G}$ leads to the Cahn–Hilliard problem, reading [45]:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla_\Omega \cdot \left(M_u \nabla_\Omega \left(\mu_u - \lambda \Delta_\Omega u\right)\right) & \text{in } \Omega \times (0,T)\,, \\ M_u \nabla_\Omega (\mu_u - \lambda \Delta_\Omega u) \cdot \mathbf{n}_\Omega = h & \text{on } \partial\Omega_N \times (0,T)\,, \\ M_u \lambda \nabla_\Omega u \cdot \mathbf{n}_\Omega = 0 & \text{on } \partial\Omega \times (0,T)\,, \\ u = g & \text{on } \partial\Omega_D \times (0,T)\,, \\ u(0) = u_0 & \text{in } \Omega \times \{0\}\,, \end{cases} \tag{1.3.6}$$

where $M_u = M_0 u(1-u)$ is the degenerate mobility [114] and the functions $g : \partial\Omega_D \to \mathbb{R}$ and $h : \partial\Omega_N \to \mathbb{R}$ form suitable conditions on the boundary decomposed as $\partial\Omega = \overline{\partial\Omega_D \cup \partial\Omega_N}$. If the surface Ω is closed, i.e. if $\partial\Omega \equiv \emptyset$, the boundary conditions in Eq. (1.3.6) are dropped and the problem reads:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla_\Omega \cdot \left(M_u \nabla_\Omega \left(\mu_u - \lambda \Delta_\Omega u\right)\right) & \text{in } \Omega \times (0,T)\,, \\ u(0) = u_0 & \text{in } \Omega \times \{0\}\,. \end{cases} \tag{1.3.7}$$

The term $u_0 : \Omega \to [0,1]$ refers to a suitable initial solution. Following [44], problems (1.3.6) and (1.3.7) are non-dimensionalized and written in weak form as:

for a.e. $t \in (0,T)$, find $u \in L^2(0,T;V) \cap H^1\left(0,T;L^2(\Omega)\right)$ such that

$$\begin{cases} \displaystyle\int_\Omega \frac{\partial u}{\partial t} \psi \, d\Omega + a_{CH}(u(t))(\psi) = 0 & \forall \psi \in V, \\ u(0) = u_0 & \text{in } \Omega, \end{cases} \tag{1.3.8}$$

where $V = H^2(\Omega)$ and:

$$a_{CH}(u)(\psi) := \int_\Omega \left(\mathbb{N}_1 M_u \nabla_\Omega \mu_u + \nabla_\Omega M_u \Delta_\Omega u\right) \cdot \nabla_\Omega \psi \, d\Omega + \int_\Omega M_u \Delta_\Omega u \Delta_\Omega \psi \, d\Omega, \tag{1.3.9}$$

where the boundary conditions (if $\partial\Omega \neq \emptyset$) are considered homogeneous. All the quantities and differential operators are now dimensionless and $\mathbb{N}_1 = \dfrac{L_0^2}{\lambda}$ is a dimensionless parameter, with L_0, M_0 and $T_0 = \dfrac{L_0^4}{\lambda M_0}$ being the characteristic length, mobility, and time scale, respectively.

1.3.2 Phase field crystal equation

Materials are often characterized by the properties of their structure at the micro-scale level, which defines their behavior at macro-scale and, usually, their traits depend on topological defects at atomic length scale. In order to account for their complex structure, most material characterizations are based on discrete Molecular Dynamics models, which are accurate but subject to severe computational time constraints since dealing with atomistic scales and phonon time scales [46]. On the contrary, continuum models, for their nature, permit the modeling of physically larger domains for longer time lengths, but with a loss in accuracy of the physical description of the materials, since they lack control on the microscale. In this respect, the phase field crystal equation [46,47,119,120] is a mathematical model for the study of crystal growth in a pure supercooled liquid, for epitaxial growth and for crack propagation in ductile materials. The model describes a two phase system at atomic length scales, thus embedding the physical properties of the microstructure in a diffusive time scale. The phase field crystal equation is based on the definition of a free energy functional $\mathscr{E}(u)$, which is minimized, by construction, by a periodic density field. In this way, the free energy functional “embeds” the periodicity nature of crystal structures directly into its formulation and naturally includes the elastic energy and symmetry properties of the periodic crystal field [47]. The solution of the crystal equation is then obtained by the minimization of the energy functional, under the constraint of mass preservation.

By considering a surface $\Omega \subset \mathbb{R}^3$ and the variable $u = u(x,t) : \Omega \times (0,T) \to \mathbb{R}$ as the local atomistic density describing the two-phase system, the liquid phase is characterized by spatially uniform values of u, while the zone with solid crystals presents the typical symmetric and periodic structures of the crystal lattice. The free energy functional $\mathscr{C}$ describing the two-phase system reads [46]:

$$\mathscr{C}(u) := \int_\Omega \left(\Phi(u) + \frac{D}{2} k^4 u^2 - D k^2 |\nabla_\Omega u|^2 + \frac{D}{2} (\Delta_\Omega u)^2 \right) d\Omega, \tag{1.3.10}$$

where D and k are positive constants, Φ is defined as:

$$\Phi(u) := -\frac{\varepsilon}{2} u^2 - \frac{g}{3} u^3 + \frac{1}{4} u^4, \tag{1.3.11}$$

with ε and g positive physical parameters. The Fréchet derivative of the free energy $\mathscr{C}$ is obtained, under suitable boundary conditions on $\partial\Omega$ (or $\partial\Omega \equiv \emptyset$), as:

$$\frac{\delta \mathscr{C}(u)}{\delta u} = \phi(u) + D k^4 u + 2 D k^2 \Delta_\Omega u + D \Delta_\Omega^2 u, \tag{1.3.12}$$

where $\phi(u) := \Phi'(u) = -\varepsilon u - g u^2 + u^3$. In order to minimize the free energy, the evolution of u is governed by the H^{-1} gradient flow of $\mathscr{C}$. The minimization problem yields the following nonlinear time-dependent sixth order PDE on the surface Ω:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta_\Omega \left(\phi(u) + D k^4 u + 2 D k^2 \Delta_\Omega u + D \Delta_\Omega^2 u \right) & \text{in } \Omega \times (0,T), \\ \nabla_\Omega \left(\phi(u) + D k^4 u + 2 D k^2 \Delta_\Omega u + D \Delta_\Omega^2 u \right) \cdot \mathbf{n}_\Omega = 0 & \text{on } \partial\Omega \times (0,T), \\ \nabla_\Omega \left(2 D k^2 \Delta_\Omega u + D \Delta_\Omega^2 u \right) \cdot \mathbf{n}_\Omega = 0 & \text{on } \partial\Omega \times (0,T), \\ \nabla_\Omega (\Delta_\Omega u) \cdot \mathbf{n}_\Omega = 0 & \text{on } \partial\Omega \times (0,T), \\ u(0) = u_0 & \text{in } \Omega, \end{cases} \tag{1.3.13}$$

where $u_0 : \Omega \to \mathbb{R}$ is a suitable initial solution. When considering fully periodic boundary conditions or closed surfaces, problem (1.3.13) becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta_\Omega \left(\phi(u) + D k^4 u + 2 D k^2 \Delta_\Omega u + D \Delta_\Omega^2 u \right) & \text{in } \Omega \times (0,T), \\ u(0) = u_0 & \text{in } \Omega. \end{cases} \tag{1.3.14}$$

Problems (1.3.13) and (1.3.14) are conveniently rewritten in dimensionless form, for which the weak formulation reads:

for a.e. $t \in (0,T)$, find $u \in L^2(0,T;V) \cap H^1(0,T;L^2(\Omega))$ such that

$$\begin{cases} \displaystyle\int_\Omega \frac{\partial u}{\partial t}\psi \, d\Omega + a_{PFC}(u(t))(\psi) = 0 & \forall \psi \in V, \\ u(0) = u_0 & \text{in } \Omega, \end{cases} \tag{1.3.15}$$

where $V = H^3(\Omega)$ and:

$$\begin{aligned} a_{PFC}(u)(\psi) :=& \int_\Omega \left(\phi'(u) + \mathbb{N}_1\right) \nabla_\Omega u \cdot \nabla_\Omega \psi \, d\Omega \\ & - \mathbb{N}_2 \int_\Omega \Delta_\Omega u \, \Delta_\Omega \psi \, d\Omega + \mathbb{N}_3 \int_\Omega \nabla_\Omega \left(\Delta_\Omega u\right) \cdot \nabla_\Omega \left(\Delta_\Omega \psi\right) d\Omega. \end{aligned} \tag{1.3.16}$$

The differential operators are now dimensionless and $\mathbb{N}_1$, $\mathbb{N}_2$, and $\mathbb{N}_3$ are dimensionless parameters defined as:

$$\mathbb{N}_1 := \frac{Dk^4}{\phi_0}, \qquad \mathbb{N}_2 := \frac{2Dk^2}{\phi_0 L_0^2}, \qquad \text{and} \quad \mathbb{N}_3 := \frac{D}{\phi_0 L_0^4}, \tag{1.3.17}$$

where L_0, ϕ_0 and $T_0 = \dfrac{L_0^2}{\phi_0}$ are characteristic values for length, the function Φ of Eq. (1.3.11), and time, respectively.

2 Geometric PDEs

Geometric Partial Differential Equations indicate those equations whose unknown represents the evolution of the geometrical domain which the equations are defined in [10]. These equations are usually defined on surfaces in 3D; the surface itself represents the unknown of the problem. An energy functional characterizes the specific geometric problem under consideration, directly depending on the shape of the surface. A surface geometric PDEs is derived by minimizing this energy. Surface geometric PDEs and, more generally, interface evolution problems arise in several applications; examples are material Science, where the crystalline structure and the properties of the materials are described via models based on geometric properties [13], as for the Stefan problem [11], or the growth of snow crystals [12], or in biomembrane modeling [26,49,52] and, more recently, in image processing, for instance for automatic contours detection or image segmentation [14, 15, 16], as well as for surface reconstruction and restoration [17,18,19,20,21].

In this chapter, two problems formulated as geometric PDEs defined on 3D surfaces are introduced. The first one is the *mean curvature flow*, for which the considered surface moves along its mean curvature vector. This causes the surface to evolve towards the minimization of its area [121] and it is of fundamental interest for the study of minimal surfaces. Problems of this kind arise, for example, when studying grain boundary motion in alloys or modeling physical systems involving surface tension, such as biological cells and membranes, bubbles, capillarity, and others, and have been extensively studied theoretically [121, 122, 123, 124].

The second problem under consideration is the *Willmore flow* problem [125], which leads to the minimization of the Willmore (or bending) energy, used, for example, in optimal surface modeling [126], surface restoration [18], and in physical models for biomembranes [55,58,60]. Theoretical results about the existence, uniqueness, and regularity of the solutions of the Willmore flow problem can be found in [127, 128, 129, 130].

2.1 Evolving surfaces

The geometric PDEs considered in this thesis are unsteady problems, where the surface domain in which they are defined evolves from an initial state. Therefore, a time (or pseudo-time) variable t is introduced, bounded between an initial time $t_0 = 0$ and a final time $T \in \mathbb{R}^+$. Let now $\{\Omega_t\}_{t\in(0,T)}$ be a family of $C^{2,1}$-hypersurfaces, i.e. such that, for each point $(\tilde{\mathbf{p}}_0, \tilde{t}) \in \mathbb{R}^3 \times (0,T)$ with $\tilde{\mathbf{p}} \in \Omega_{\tilde{t}}$, there exists an open set $U \subset \mathbb{R}^3$ and a function $u \in C^{2,1}\left(U \times (\tilde{t}-\delta, \tilde{t}+\delta)\right)$, for a constant $\delta > 0$, such that [10]:

$$U \cap \Omega_t = \left\{\mathbf{p} \in U \mid u(\mathbf{p},t) = 0\right\} \qquad \text{and} \quad \nabla u(\mathbf{p},t) \neq 0 \qquad \forall \mathbf{p} \in U \cap \Omega_t. \tag{2.1.1}$$

Now, let $\Omega_0 \subset \mathbb{R}^3$ represent the initial surface, at time $t = 0$, described through the geometrical mapping $\mathbf{X}_0 : \widehat{\Omega} \to \mathbb{R}^3$, where $\widehat{\Omega}$ is the parametric domain introduced in Section 1.1. At each time instance t, the corresponding manifold Ω_t, evolved up to time t, is represented by the geometrical mapping $\mathbf{X}_t : \widehat{\Omega} \to \mathbb{R}^3$. The evolution of the surface domain under consideration, represented by the family of compact, oriented $C^{2,1}$-hypersurfaces $\{\Omega_t\}_{t\in(0,T)}$, is thus described through the corresponding family of geometrical mappings $\{\mathbf{X}_t\}_{t\in(0,T)}$. It is convenient, from the notational point of view, to represent the evolving surface through the time-dependent geometric mapping:

$$\mathbf{X}(\boldsymbol{\xi},t) = \mathbf{X}_t(\boldsymbol{\xi}) \qquad \boldsymbol{\xi} \in \widehat{\Omega},\ \forall t \in (0,T), \tag{2.1.2}$$

which, together with the represented surfaces, is assumed to be sufficiently smooth both in space and time. Therefore, for each $t \in (0,T)$ the surface Ω_t is represented by $\mathbf{X}(t) = \mathbf{X}(\boldsymbol{\xi},t)$, for $\boldsymbol{\xi} \in \widehat{\Omega}$. The boundary of Ω_t, if present, is denoted by $\partial\Omega_t$. In general, Ω_t depends on the mapping $\mathbf{X}(t)$, thus it is possible to consider $\Omega_t = \Omega(\mathbf{X}(t))$.

For convenience of notation, let G_T be the topological cylinder defined as [48]:

$$G_T := \bigcup_{t\in(0,T)} \Omega_t \times \{t\}. \tag{2.1.3}$$

Then, as done in Eq. (1.1.27) for non-evolving surfaces, the identity function $\mathbf{x} : G_T \to \mathbb{R}^3$ for time-dependent surfaces is introduced as:

$$\mathbf{x}(\mathbf{X}(\boldsymbol{\xi},t),t) := \mathbf{X}(\boldsymbol{\xi},t) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \quad t \in [0,T]. \tag{2.1.4}$$

Let $\mathbf{V} : \widehat{\Omega} \times (0,T) \to \mathbb{R}^3$ be defined as:

$$\mathbf{V}(\boldsymbol{\xi},t) := \frac{\partial \mathbf{X}}{\partial t}(\boldsymbol{\xi},t) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \quad t \in (0,T). \tag{2.1.5}$$

For a general function $\psi : G_T \to \mathbb{R}$, its material derivative $\dot{\psi}$ reads:

$$\dot{\psi}(\mathbf{X}(\boldsymbol{\xi},t),t) := \frac{\partial \psi}{\partial t}(\mathbf{X}(\boldsymbol{\xi},t),t) + \mathbf{V}(\boldsymbol{\xi},t) \cdot \nabla\psi(\mathbf{X}(\boldsymbol{\xi},t),t) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \quad t \in (0,T), \tag{2.1.6}$$

where $\nabla\psi$ is the three-dimensional spatial gradient of a smooth prolongation of ψ from Ω_t into a tubular region in $\mathbb{R}^3$ containing Ω_t. Let $\mathbf{v} : G_T \to \mathbb{R}^3$ denote the velocity of the surface Ω_t:

$$\mathbf{v}(\mathbf{X}(\boldsymbol{\xi}, t), t) := \dot{\mathbf{x}}(\mathbf{X}(\boldsymbol{\xi}, t), t) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \quad t \in (0, T). \tag{2.1.7}$$

Following Eq. (2.1.6), the material derivative of the identity function is written has:

$$\dot{\mathbf{x}}(\mathbf{X}(\boldsymbol{\xi}, t), t) := \left(\frac{\partial \mathbf{x}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{x}\right)(\mathbf{X}(\boldsymbol{\xi}, t), t) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \quad t \in (0, T); \tag{2.1.8}$$

since $\partial \mathbf{x}/\partial t \equiv 0$ and $\nabla \mathbf{x} = \mathrm{I}$ [48], the velocity of the points of the surface, defined on G_T in Eq. (2.1.8), is simplified into:

$$\mathbf{v}(\mathbf{X}(\boldsymbol{\xi}, t), t) = \frac{\partial \mathbf{x}}{\partial t}(\mathbf{X}(\boldsymbol{\xi}, t), t) = \mathbf{V}(\boldsymbol{\xi}, t) \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \quad t \in (0, T). \tag{2.1.9}$$

The normal component of the velocity of the evolving surface is denoted with $v : G_T \to \mathbb{R}$, and reads:

$$v(\mathbf{X}(\boldsymbol{\xi}, t)) := \mathbf{v}(\mathbf{X}(\boldsymbol{\xi}, t)) \cdot \mathbf{n}_{\Omega_t}(\mathbf{X}(\boldsymbol{\xi}, t)), \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \quad t \in (0, T). \tag{2.1.10}$$

In order to simplify the notation, for quantities defined on G_T or Ω_t, as e.g. $\mathbf{x}$, $\mathbf{v}$ and v, here henceforth the arguments $\mathbf{X}(\boldsymbol{\xi}, t)$ and t will be dropped, since it is always assumed that $\boldsymbol{\xi} \in \widehat{\Omega}$ and $t \in (0, T)$.

A useful way of describing the local evolution (or perturbation) of Ω_t in a small interval of time can be accomplished by looking at the evolution of the individual points of the surface. In particular, by considering a small final time $\widetilde{T}$, the points of the surface evolve according to a system of ordinary differential equations [110]:

$$\begin{cases} \dfrac{d}{dt}\mathbf{p}(t) = \boldsymbol{\varphi}(\mathbf{p}(t)) & t \in (0, \widetilde{T}), \\ \mathbf{p}(0) = \mathbf{p}, \end{cases} \tag{2.1.11}$$

for each given point $\mathbf{p} \in \Omega_t$; $\boldsymbol{\varphi} : \Omega_t \to \mathbb{R}^3$ is a velocity (or perturbation) field. The trajectories of the points of the surfaces, governed by the velocity $\boldsymbol{\varphi}$, can be expressed by the family of transformations $T_\varepsilon^{\boldsymbol{\varphi}} : \Omega_t \to \mathbb{R}^3$, for $\varepsilon \in [0, \widetilde{T})$, defined as [110]:

$$T_\varepsilon^{\boldsymbol{\varphi}}(\mathbf{p}) := \mathbf{p}(\varepsilon) \qquad \forall \mathbf{p} \in \Omega_t. \tag{2.1.12}$$

This is a convenient way to describe $\Omega_{t+\varepsilon}$ as a transformation of the surface Ω_t, reading:

$$\Omega_{t+\varepsilon} = T_\varepsilon^{\boldsymbol{\varphi}}(\Omega_t) = \left\{T_\varepsilon^{\boldsymbol{\varphi}}(\mathbf{p}), \mathbf{p} \in \Omega_t\right\} \qquad \forall \varepsilon \in [0, \widetilde{T}). \tag{2.1.13}$$

Finally, the transport theorems, useful for computing the time derivative of integral quantities [10, 110], are recalled in Theorems 4 and 5.

Theorem 4. *Let $\psi \in C^1(\widetilde{\Omega}_t)$, where the open set $\widetilde{\Omega}_t$ is such that the surface $\Omega_t \subset \widetilde{\Omega}_t$ for all $t \in (0,T)$. Then, the material derivative of the integral of ψ over the surface Ω_t is:*

$$\frac{d}{dt}\int_{\Omega_t} \psi \, d\Omega_t = \int_{\Omega_t} \frac{\partial \psi}{\partial t} \, d\Omega_t + \int_{\Omega_t} \psi v H \, d\Omega_t + \int_{\Omega_t} \frac{\partial \psi}{\partial \mathbf{n}_\Omega} v \, d\Omega_t, \tag{2.1.14}$$

where H is the mean curvature defined in Eq. (1.1.23) *and v is the normal velocity defined in Eq.* (2.1.10).

Theorem 5. *If the surface Ω_t is the boundary of an open, bounded subset $\Theta_t \subset \mathbb{R}^3$, then, given a function $\psi \in C^1(\Theta_t)$, the material derivative of the integral of ψ in the volume Θ_t satisfies:*

$$\frac{d}{dt}\int_{\Theta_t} \psi \, d\Theta_t = \int_{\Theta_t} \frac{\partial \psi}{\partial t} \, d\Theta_t + \int_{\Omega_t} \psi v \, d\Omega_t. \tag{2.1.15}$$

2.2 Shape differential calculus

Let Ψ be a scalar function depending on Ω_t and $\mathbf{p} \in \Omega_t$, i.e. $\Psi = \Psi(\Omega_t, \mathbf{p})$. The Lagrangian derivative of Ψ with respect to the vector field $\boldsymbol{\varphi} : \Omega_t \to \mathbb{R}^3$ is obtained as [110]:

$$\dot{\Psi}(\Omega_t, \mathbf{p}; \boldsymbol{\varphi}) := \lim_{\varepsilon \to 0} \frac{\Psi\left(\Omega_{t+\varepsilon}, T_\varepsilon^{\boldsymbol{\varphi}}(\mathbf{p})\right) - \Psi(\Omega_t, \mathbf{p})}{\varepsilon} \qquad \forall \mathbf{p} \in \Omega_t, \tag{2.2.1}$$

where the transformation flow $T_\varepsilon^{\boldsymbol{\varphi}}$, for the vector field $\boldsymbol{\varphi}$, is defined in Eq. (2.1.12). Now, let the open, bounded subset $\Theta_t \subset \mathbb{R}^3$ be the volume enclosing Ω_t, for each $t \in (0,T)$. The shape derivative of a scalar function $\widetilde{\Psi}$ depending on the volume Θ_t, i.e. $\widetilde{\Psi} = \widetilde{\Psi}(\Theta_t, \mathbf{p})$, in Θ_t along the direction $\boldsymbol{\varphi}$ is defined as [110]:

$$\widetilde{\Psi}'(\Theta_t, \mathbf{p}; \boldsymbol{\varphi}) := \dot{\widetilde{\Psi}}(\Theta_t, \mathbf{p}; \boldsymbol{\varphi}) - \nabla \widetilde{\Psi} \cdot \boldsymbol{\varphi}, \tag{2.2.2}$$

where the Lagrangian derivative $\dot{\widetilde{\Psi}}$ is defined as in Eq. (2.2.1) for functions depending on the volume Θ_t. In particular, by returning to functions Ψ depending on the surface Ω_t and by considering vector fields $\boldsymbol{\varphi}$ which are normal to the surface, then the shape derivative of Ψ in Ω_t along $\boldsymbol{\varphi}$ sees the last term of Eq. (2.2.2) vanishing, reading:

$$\Psi'(\Omega_t, \mathbf{p}; \boldsymbol{\varphi}) := \dot{\Psi}(\Omega_t, \mathbf{p}; \boldsymbol{\varphi}). \tag{2.2.3}$$

Now, let J be a shape energy functional depending on the surface Ω_t, i.e. $J = J(\Omega_t)$. By considering a fixed time $t \in (0,T)$, the shape differential of J, i.e. the first variation of J corresponding to a deformation of its argument Ω_t along the direction $\boldsymbol{\varphi} : \Omega_t \to \mathbb{R}^3$, is denoted with $dJ(\Omega_t)(\boldsymbol{\varphi})$ and reads [110]:

$$dJ(\Omega_t)(\boldsymbol{\varphi}) := \lim_{\varepsilon \to 0} \frac{J(\Omega_{t+\varepsilon}) - J(\Omega_t)}{\varepsilon}, \tag{2.2.4}$$

where $\Omega_{t+\varepsilon}$ is the domain perturbed along $\boldsymbol{\varphi}$, as defined in Eq. (2.1.13). In Eq. (2.2.4), the functional J depends on Ω_t, but it can also be defined to depend instead on the volume Θ_t, with a definition of shape differential analogue to Eq. (2.1.13). In general, three forms of energy functionals are usually considered. If J takes the form:

$$J(\Theta_t) = \int_{\Theta_t} \widetilde{\Psi}(\Theta_t, \mathbf{p})\, \mathrm{d}\Theta_t, \tag{2.2.5}$$

i.e. J consists in a volume integral over Θ_t of a function $\widetilde{\Psi}$ depending on the shape of Θ_t and the position $\mathbf{p}$, then its shape derivative in the direction $\boldsymbol{\varphi}$ assumes the form [110]:

$$dJ(\Theta_t)(\boldsymbol{\varphi}) = \int_{\Theta_t} \widetilde{\Psi}'(\Theta_t, \mathbf{p}; \boldsymbol{\varphi})\, d\Theta_t + \int_{\Omega_t} \widetilde{\Psi}(\Theta_t, \mathbf{p}) \boldsymbol{\varphi} \cdot \mathbf{n}_{\Omega_t}\, d\Omega_t. \tag{2.2.6}$$

If instead J has the form:

$$J(\Omega_t) = \int_{\Omega_t} \widetilde{\Psi}(\Theta_t, \mathbf{p})\, d\Omega_t, \tag{2.2.7}$$

then its shape derivative in the direction $\boldsymbol{\varphi}$ reads:

$$dJ(\Omega_t)(\boldsymbol{\varphi}) = \int_{\Omega_t} \widetilde{\Psi}'(\Theta_t, \mathbf{p}; \boldsymbol{\varphi})\, d\Omega_t + \int_{\Omega_t} \left(\frac{\partial \widetilde{\Psi}}{\partial \mathbf{n}_{\Omega_t}} (\Theta_t, \mathbf{p}) + \widetilde{\Psi}(\Theta_t, \mathbf{p}) H \right) \boldsymbol{\varphi} \cdot \mathbf{n}_{\Omega_t}\, d\Omega_t. \tag{2.2.8}$$

Finally, if J is a surface integral of a function depending on the shape of Ω_t, i.e.:

$$J(\Omega_t) = \int_{\Omega_t} \Psi(\Omega_t, \mathbf{p})\, d\Omega_t, \tag{2.2.9}$$

then its shape derivative is calculated by considering a normal extension of Ψ as a constant into a neighborhood of Ω_t, resulting in:

$$dJ(\Omega_t)(\boldsymbol{\varphi}) = \int_{\Omega_t} \Psi'(\Omega_t, \mathbf{p})\, d\Omega_t + \int_{\Omega_t} \Psi(\Omega_t, \mathbf{p}) H \boldsymbol{\varphi} \cdot \mathbf{n}_{\Omega_t}\, d\Omega_t. \tag{2.2.10}$$

An important result, that gives a common formulation to the shape derivatives, is described in the *Structure theorem* (Theorem 3.6) and its corollary in [110], whose contents are condensated in the following theorem.

Theorem 6. *Consider the shape functional* J *and assume that* $dJ(\Theta_t)(\boldsymbol{\varphi})$ *exists for all* $\boldsymbol{\varphi} \in \mathscr{D}(\mathbb{R}^3; \mathbb{R}^3)$. *Then, there exists an element* $\mathscr{G} \in \mathscr{D}'(\mathbb{R}^3; \mathbb{R}^3)$, *called* shape gradient, *with support in* Ω_t *and such that:*

$$dJ(\Theta_t)(\boldsymbol{\varphi}) = \int_{\Omega_t} \mathscr{G} \boldsymbol{\varphi} \cdot \mathbf{n}_{\Omega_t}\, d\Omega_t \qquad \forall \boldsymbol{\varphi} \in \mathscr{D}(\mathbb{R}^3; \mathbb{R}^3). \tag{2.2.11}$$

Theorem 6 states, in particular, that the shape derivative dJ along $\boldsymbol{\varphi}$, for each direction $\boldsymbol{\varphi}$, only depends on the normal component of $\boldsymbol{\varphi}$ on Ω_t.

2.3 Geometric PDEs as geometric gradient flows

A geometric PDE defined on a surface is an equation, or a system of equations, describing the evolution of the points of the surface, on which the equations are defined. Starting from an initial surface Ω_0, identified by $\mathbf{X}_0 : \widehat{\Omega} \to \mathbb{R}^3$, the aim is finding, for all the times $t \in (0, T)$, the family of surfaces $\{\Omega_t\}_{t\in(0,T)} \subset \mathbb{R}^3$ identified by their geometrical mapping $\mathbf{X}(t) : \widehat{\Omega} \to \mathbb{R}^3$, whose evolution obeys a differential law of the form:

$$\begin{cases} \dot{\mathbf{X}} = \mathscr{F}(t, \mathbf{X}, \mathbf{n}_\Omega, H, K, \ldots) & \text{in } \widehat{\Omega}, \text{ for } t \in (0, T), \\ \mathbf{X}(0) = \mathbf{X}_0, \end{cases} \tag{2.3.1}$$

possibly with additional boundary conditions in case the domain is not a closed surface. The law $\mathscr{F}$ identifies the problem at hand and potentially depends on several geometrical quantities associated with the geometry. In general, the geometric PDEs treated in this thesis derive from the minimization of a shape energy functional $J(\Omega)$. This functional can be seen as the objective functional in an optimization process, where the design variable is represented by the surface Ω itself. J usually depends on geometrical quantities associated to the geometrical mapping of the surface Ω, such as the surface normal and the curvatures described in Section 1.1. Moreover, in this thesis changes of topology are neglected and the parametric domain $\widehat{\Omega}$ is kept invariant throughout the geometric evolution process.

In order to minimize the energy, the gradient flow of J is considered [113, 131]. Let $V(\Omega_t)$ be a Hilbert space induced by the scalar product $\mathscr{M}(\cdot,\cdot)$ on Ω_t. The gradient flow of the energy functional J aims at minimizing J by seeking, for all $t \in (0, T)$, a velocity $\dot{\mathbf{x}} \in V(\Omega_t)$ such that:

$$\mathscr{M}(\dot{\mathbf{x}}, \boldsymbol{\varphi}) = -dJ(\Omega_t)(\boldsymbol{\varphi}) \qquad \forall \boldsymbol{\varphi} \in V(\Omega_t),\ t \in (0, T). \tag{2.3.2}$$

In particular, by using Theorem 6, problem (2.3.2) is equivalent to:

$$\mathscr{M}(\dot{\mathbf{x}}, \boldsymbol{\varphi}) = -\int_{\Omega_t} \mathscr{G}\, \boldsymbol{\varphi} \cdot \mathbf{n}_{\Omega_t}\, d\Omega_t \qquad \forall \boldsymbol{\varphi} \in V(\Omega_t),\ t \in (0, T). \tag{2.3.3}$$

A common choice for $\mathscr{M}(\cdot,\cdot)$ is the $L^2(\Omega_t)$ scalar product, for which:

$$\mathscr{M}(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2) := \int_{\Omega_t} \boldsymbol{\varphi}_1 \cdot \boldsymbol{\varphi}_2\, d\Omega_t \qquad \forall \boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2 \in L^2(\Omega_t),\ t \in (0, T). \tag{2.3.4}$$

Therefore, the L^2-gradient flow of the energy functional J reads: given an initial surface $\Omega_0 \subset \mathbb{R}^3$, represented by $\mathbf{x}_0$, for a.e. $t \in (0, T)$ find $\Omega_t \subset \mathbb{R}^3$ such that:

$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t = -\mu\, dJ(\Omega_t)(\boldsymbol{\varphi}) & \forall \boldsymbol{\varphi} \in V(\Omega_t), \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0, \end{cases} \tag{2.3.5}$$

where $\mu \in \mathbb{R}^+$ is a constant representing a mobility and V is a suitable function space for the trial and test functions, which will be defined later specifically for each problem under consideration. Problem (2.3.5) can be expressed also in strong form as:

$$\begin{cases} \dot{\mathbf{x}} = -\mu \mathscr{G}\, \mathbf{n}_{\Omega_t} & \text{in } \Omega_t,\ t \in (0,T), \\ \text{b.c.s.} & \text{on } \partial\Omega_t, \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0, \end{cases} \tag{2.3.6}$$

where “b.c.s” stands for the (possible) boundary conditions and $\mathscr{G}$ is the shape gradient associated with J. Problems (2.3.5) and (2.3.6) represent, in general, highly nonlinear systems of PDEs, whose steady states correspond to local minima of the energy J. Indeed, regardless of the choice of scalar product $\mathscr{M}$, the gradient flow in Eq. (2.3.2) is always evolving towards the (local) minimization of the energy J, as:

$$dJ(\Omega_t)(\dot{\mathbf{x}}) = -\mathscr{M}(\dot{\mathbf{x}}, \dot{\mathbf{x}}) = -\|\dot{\mathbf{x}}\|^2_{V(\Omega_t)} \leq 0 \qquad t \in (0,T). \tag{2.3.7}$$

2.4 Mean curvature flow

The first geometric PDE considered in this thesis is the *mean curvature flow*. Let J_A be the energy functional defined as:

$$J_A(\Omega_t) := \int_{\Omega_t} 1\, d\Omega_t, \tag{2.4.1}$$

which corresponds to the area of the surface Ω_t. By applying the formula in Eq. (2.2.10) to the functional J_A, the shape derivative of J_A in Ω_t along the direction $\boldsymbol{\varphi} : \Omega_t \to \mathbb{R}^3$ is obtained as [48, 121]:

$$dJ_A(\Omega_t)(\boldsymbol{\varphi}) = \int_{\Omega_t} \mathbf{H} \cdot \boldsymbol{\varphi}\, d\Omega, \tag{2.4.2}$$

where $\mathbf{H}$ is the total mean curvature vector defined in Eq. (1.1.25). Following the definition in Eq. (2.2.11), the shape gradient of the area functional $\mathscr{G}_A$ is the mean curvature, i.e. :

$$\mathscr{G}_A := H. \tag{2.4.3}$$

Therefore, the *mean curvature flow* is the problem associated with the minimization of the area functional J_A by means of a L^2-gradient flow. Following the prototype Eq. (2.3.5), the problem reads: given an initial surface $\Omega_0 \subset \mathbb{R}^3$ identified by the map $\mathbf{x}_0 : \widehat{\Omega} \to \mathbb{R}^3$, for all $t \in (0,T)$ find the surface Ω_t, identified by $\mathbf{x} : \Omega_t \to \mathbb{R}^3$, such that:

$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t = -\mu\, dJ_A(\Omega_t)(\boldsymbol{\varphi}) & \forall \boldsymbol{\varphi} \in V(\Omega_t), \\ \mathbf{x}(0) = \mathbf{x}_0 \quad \text{in } \Omega_0, \end{cases} \tag{2.4.4}$$

where $V(\Omega_t)$ is a suitable Hilbert function space, which will be specified in Chapter 5. The mean curvature flow problem in strong flow can be written as in Eq. (2.3.6), reading: for all $t \in (0, T)$, find the surface Ω_t such that:

$$\begin{cases} \dot{\mathbf{x}} = -\mu \mathbf{H} & \text{in } \Omega_t, \\ \text{b.c.s.} & \text{on } \partial\Omega_t, \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0. \end{cases} \tag{2.4.5}$$

The mean curvature flow models the evolution of the surface towards the local minimization of its area [10]. This can be verified by using the transport Theorem 4 and by setting $\psi = 1$ and normal velocity $v = -H$ from Eq. (2.4.5), thus obtaining:

$$\frac{d}{dt} J_A(\Omega_t) = -\frac{1}{\mu} \int_{\Omega_t} v^2 \, d\Omega_t \leq 0, \tag{2.4.6}$$

indicating a decrease of the surface area in time.

The mean curvature flow problem has been extensively studied, from different points of view. In case of mono-dimensional parametric space $\widehat{\Omega}$, then Ω_t represents a curve in the physical space. The mean curvature flow problem for curves is known as *curve shortening flow*. In this regard, smooth solutions in finite time have been studied for instance in [132,133]. Approaches for handling the cusp-like singularities that can arise if the initial curve is not embedded are described in [134, 135, 136]. Regarding surfaces, in [121] an approach to the surface mean curvature flow based on the measure theory is considered. In [123] existence of a solution is studied for the parametric evolution of an initial smooth convex surface without boundary. Singularities arising from the surface evolution according to the mean curvature are analyzed in [122]. The mean curvature flow for non-parametric surfaces with boundary is analyzed in [124]. For more details and references see [10,121,123].

2.5 Willmore flow

The Willmore energy functional J_W is defined as [125]:

$$J_W(\Omega_t) := \frac{1}{2} \int_{\Omega_t} H^2 d\Omega_t \tag{2.5.1}$$

The energy J_W expresses the bending energy associated to the surface Ω_t [125]. The *Willmore flow problem* is the L^2-gradient flow of J_W. Under suitable hypotheses, the shape derivative of J_W at Ω_t along $\boldsymbol{\varphi} : \Omega_t \to \mathbb{R}^3$ is given by [125]:

$$dJ_W(\Omega_t)(\boldsymbol{\varphi}) = -\int_{\Omega_t} \left(\Delta_\Omega H - \frac{1}{2} H^3 + H |\mathscr{H}|^2 \right) \boldsymbol{\varphi} \cdot \mathbf{n}_{\Omega_t} \, d\Omega_t, \tag{2.5.2}$$

which, for three dimensional surfaces is equivalent to:

$$dJ_W(\Omega_t)(\boldsymbol{\varphi}) = -\int_{\Omega_t} \left(\Delta_\Omega H + \frac{1}{2}H^3 - 2HK\right) \boldsymbol{\varphi} \cdot \mathbf{n}_{\Omega_t}\, d\Omega_t, \tag{2.5.3}$$

in virtue of Eq. (1.1.26). The shape gradient $\mathscr{G}_W$ of the Willmore energy then reads:

$$\mathscr{G}_W := -\Delta_\Omega H - \frac{1}{2}H^3 + 2HK. \tag{2.5.4}$$

In this thesis, the Willmore flow is considered only for closed surfaces, for which $\partial\Omega_t \equiv \emptyset$. Then, by following the prototype Eq. (2.3.5), the Willmore flow problem reads: given an initial surface $\Omega_0 \subset \mathbb{R}^3$ identified by the mapping $\mathbf{x}_0 : \Omega_0 \to \mathbb{R}^3$, find, for all $t \in (0, T)$, the surface Ω_t, identified by $\mathbf{x} : \Omega_t \to \mathbb{R}^3$, such that:

$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t = -\mu\, dJ_W(\Omega_t)(\boldsymbol{\varphi}) & \forall \boldsymbol{\varphi} \in V(\Omega_t), \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0, \end{cases} \tag{2.5.5}$$

with a suitable Hilbert function space $V(\Omega_t)$. Problem (2.5.5) can be expressed in strong form as follows: find, for all $t \in (0, T)$, the surface Ω_t such that:

$$\begin{cases} \dot{\mathbf{x}} = \mu \left(\Delta_\Omega H + \frac{1}{2}H^3 - 2HK\right) \mathbf{n}_{\Omega_t} & \text{in } \Omega_t, \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0. \end{cases} \tag{2.5.6}$$

When considering curves, the problem is generally known as the *elastic flow of curves*. In this case, global existence in time of a solution for curves in $\mathbb{R}^n$ was studied in [137] and [138], for $n = 2$ and $n \geq 3$ respectively. For surfaces, which represent the focus of this thesis, the Willmore flow problem has been studied analytically mainly on closed surfaces. Existence of a solution up to the finite time $T < +\infty$ for two-dimensional surfaces in $\mathbb{R}^n$, with $n \geq 3$, is proven in [127], with T depending on the curvature of the initial surface Ω_0. Existence and uniqueness of the local solution of problem (2.5.6) under the hypothesis that the initial geometry Ω_0 is a compact, closed, immersed, and orientable $C^{2,\alpha}$-surface in $\mathbb{R}^3$ is proven in [130], together with the global existence of the solution in time where Ω_0 is "sufficiently" close to a sphere. In [128] global existence of solutions is shown under the assumption that $\int_{\Omega_0} |\mathring{\mathscr{H}}|^2$ is sufficiently small, being $\mathring{\mathscr{H}}$ the trace-free part of the second fundamental form $\mathscr{H}$. In [129] the authors proved that if Ω_0 is a smooth immersion of a sphere in $\mathbb{R}^3$ and it is such that its Willmore energy $J_W(\Omega_0) \leq 16\pi$, then its Willmore flow smoothly exists for all times and converges to a sphere.

2.5.1 Willmore flow with spontaneous curvature

When modeling cell membranes, more general energies than the pure Willmore energy in Eq. (2.5.1) are usually considered. The spontaneous curvature is an extension of the Willmore energy functional, which reads:

$$J_{W_0}(\Omega_t) := \frac{1}{2}\int_{\Omega_t} (H - H_0)^2 d\Omega_t, \tag{2.5.7}$$

where $H_0 : \Omega_t \to \mathbb{R}$ represents the prescribed spontaneous curvature over the surface Ω_t. H_0 is used in biophysics to model physical asymmetries of the membranes, due, for instance, to chemical differences in the environment in which the cell is immersed [60]. The shape derivative of J_{W_0} in Ω_t along the direction $\boldsymbol{\varphi} : \Omega_t \to \mathbb{R}^3$ reads [125]:

$$dJ_{W_0}(\Omega_t)(\boldsymbol{\varphi}) = -\int_{\Omega_t} \left[\Delta_\Omega(H - H_0) - \frac{1}{2}H(H - H_0)^2 + (H - H_0)\left(|\nabla_\Omega \mathbf{n}_{\Omega_t}|^2 + \frac{\partial H_0}{\partial \mathbf{n}_{\Omega_t}}\right)\right] \boldsymbol{\varphi}\cdot\mathbf{n}_{\Omega_t}\, d\Omega_t. \tag{2.5.8}$$

In case of constant spontaneous curvatures, i.e. $H_0 \in \mathbb{R}$, the shape derivative of J_{W_0} in Eq. (2.5.8) is simplified into the following [72, 139]:

$$\begin{aligned} dJ_{W_0}(\Omega_t)(\boldsymbol{\varphi}) &= -\int_{\Omega_t} \left[\Delta_\Omega H - \frac{1}{2}H(H - H_0)^2 + (H - H_0)\left(H^2 - 2K\right)\right] \boldsymbol{\varphi}\cdot\mathbf{n}_{\Omega_t}\, d\Omega_t \\ &= -\int_{\Omega_t} \left[\Delta_\Omega H + H\left(\frac{1}{2}H^2 - \frac{1}{2}H_0^2 - 2K\right) - 2H_0 K\right] \boldsymbol{\varphi}\cdot\mathbf{n}_{\Omega_t}\, d\Omega_t. \end{aligned} \tag{2.5.9}$$

The Willmore flow problem with spontaneous curvature on closed surfaces is written in weak and strong form as Eqs. (2.5.5) and (2.5.6), respectively, with the shape derivative replaced by dJ_{W_0} of Eq. (2.5.9).

3 Isogeometric Analysis

When using the Finite Element Method (FEM) to approximate surface and geometric PDEs, a challenge for obtaining an accurate numerical approximation is the construction of a suitable computational mesh, which still represents an approximation of the original geometry [112]. Indeed, generating a mesh of "good quality" is necessary not only to accurately represent the surface, but also to accurately evaluate the differential operators which are associated to the geometrical properties of the manifold. In particular, this requires the accurate computation of several geometrical quantities, such as the normal and curvature of the surface introduced in Sections 1.1 and 1.1.2. Moreover, besides being time consuming, the process of mesh generation may require a large number of Degrees Of Freedom (DOFs) for the PDE approximation. In the FEM context, different approaches have been introduced aiming at controlling the approximation error induced by the discretization of the geometry; examples are the surface FEM [22, 23], or geometrically consistent Adaptive FEM [25, 26, 27, 49]. Other approaches are based on modeling the surface as immersed in the 3D domain or treated implicitly, as, for example, for level set formulations [29, 30] or diffuse and resistive interface methods [31, 32].

In alternative to the above mentioned methods, this thesis focuses on the use of Isogeometric Analysis [34, 35] for the numerical approximation of the PDEs. Isogeometric Analysis (IGA) is a discretization method for approximating PDEs based on the isogeometric paradigm, for which the same basis functions are used first for the geometrical description of the domain and then for the numerical approximation of the solution of the PDEs [34, 35]; in this respect, IGA was developed with the goal of filling the gap between Computer Aided Design (CAD) and FEM, by providing a unified representation of the geometrical design, the computational domain, and the approximation function spaces. One potential advantage of IGA is its ability to directly exploit the description of the geometry for the spatial discretization of the PDEs, without requiring the time-consuming process of generating a computational mesh, which often only represents an approximation of the geometry.

While IGA is nowadays adopted for several geometrical representations [140, 141], including T-splines [142], in this thesis the focus is devoted to tensor-product B-splines and NURBS surfaces [37] built as single patches. This choice is mainly motivated by the ability of NURBS and B-splines to exactly represent several geometries of practical interest, especially in industrial applications, at the coarsest level of discretization. Moreover, additional refinement procedures permit the enhancement of the approximation properties of the finite dimensional spaces, while not affecting the geometrical representation.

The focus of this thesis is mainly to address high order PDEs. For equations of order $2m$, with $m \geq 1$, besides the already mentioned geometrical advantages, IGA allows the use of the standard Galerkin formulation, without invoking the mixed formulations required by the isoparametric FEM [42] with the standard Lagrange polynomial basis functions when $m \geq 2$. The possibility of using globally C^k-continuous NURBS basis functions, with $m-1 \leq k \leq p-1$ and p being the polynomial degree, yields IGA finite dimensional spaces that are subspaces of the trial and test Sobolev spaces H^m that naturally arise when dealing with PDEs of order $2m$, with $m \geq 1$ [40]. In addition, periodic NURBS basis functions can be built on surfaces with the goal of obtaining globally high order continuous NURBS function spaces [44]. This in turn allows the construction of NURBS function spaces of the required regularity and thus the solution of high order PDEs defined on closed surfaces.

In this chapter, the B-Spline and NURBS basis functions are introduced, which define the geometry and build the approximation function spaces; then, spatial discretization of generic elliptic PDEs with NURBS-based IGA within the Galerkin method is described. The concepts which are introduced in this chapter act as a framework for the numerical approximation of the PDEs of interest covered in this thesis.

3.1 Univariate B-Spline and NURBS

Let $\widehat{I} \subset \mathbb{R}$ be a reference interval, acting as parametric domain. The reference interval $\widehat{I}$ is divided into sub-domains called *patches*. Each patch is defined through a *knot vector*, a collection of non-negative values reading $\Xi := \{\xi_i\}_{i=1}^{n} \in \mathbb{R}^n$, with $n \in \mathbb{N}$, $n \geq 2$. The single values ξ_i, for $i = 1, \dots, n$, are the *knots* and are assumed to be ordered in ascending order. The knots divide the knot vector into the *knot spans*, specifically into the intervals $[\xi_i, \xi_{i+1}] \subset \mathbb{R}$, for $i = 1, \dots, n-1$. In analogy with the FEM, the knot spans are also called *mesh elements* and the parametric domain acts as the (reference) *computational mesh*. A *uniform* knot vector is defined by knots which are equally spaced; otherwise, it is called *non-uniform*. A knot can be repeated inside the knot vector. In particular, if the knot ξ_i is repeated r times, it is said to have *multiplicity* r and it is written, in this thesis, with the notation $\{\xi_i\}^r$ inside the knot vector. We focus on single-patch mappings; this means that the parametric interval $\widehat{I}$ is considered to host a single knot vector. Moreover, the parametric domain is represented by the interval $\widehat{I} \equiv [0,1] \subset \mathbb{R}$ and therefore the knot vector is normalized, i.e. the knots ranges from 0 to 1.

The B-Spline basis functions of degree p on the reference interval $\widehat{I}$ are defined through the Cox-de Boor recursion formula. In particular, the piecewise constant basis functions, i.e. for which $p = 0$, read:

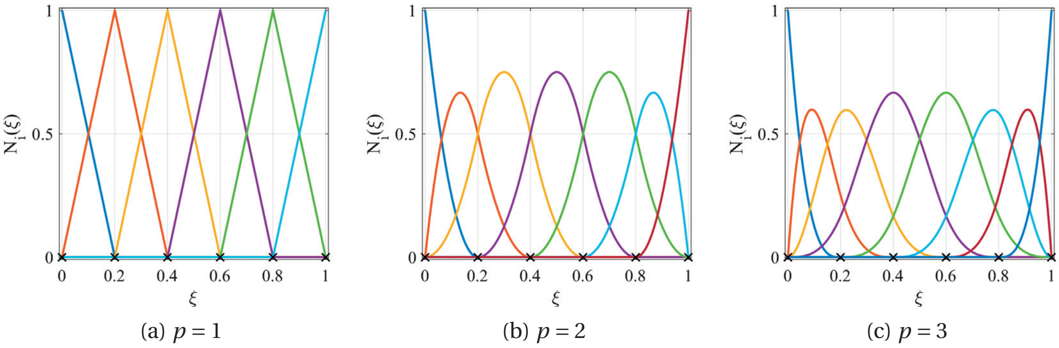
$$\widehat{N}_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{3.1.1}$$

for $i = 1, \dots, n-1$. This represents the initialization of the recursion formula, for which each basis function of degree p is defined as linear combination of two basis functions of degree $p-1$ as follows:

$$\widehat{N}_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \widehat{N}_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \widehat{N}_{i+1,p-1}(\xi), \tag{3.1.2}$$

for $i = 1, \dots, n_{bf}$, $n_{bf} = n - p - 1$; n_{bf} represents the number of basis functions. Note that, if a denominator in Eq. (3.1.2) is equal to zero, then the corresponding quotient is considered to be zero. The B-Spline basis functions, defined in Eqs. (3.1.1) and (3.1.2), enjoy several interesting properties; the reader is referred to the book [37] for an in-depth analysis. In particular, the B-Spline basis functions are pointwise non-negative over the entire parametric domain and constitute a partition of unity. Moreover, the support of each basis function of degree p extends over $p+1$ knot spans. Therefore, with a basis of degree p, to evaluate all basis functions in a certain point $\xi \in [0,1]$ it is necessary to evaluate just $p+1$ of them.

The continuity across the knots of B-Spline basis functions can be easily achieved by changing the multiplicity of the knots inside the knot vector. In particular, by considering basis functions of order p, if the knot ξ_i has multiplicity $1 \leq r_i \leq p$ then the B-Spline basis functions are C^{p-r_i}-continuous across that knot. Therefore, if the knot has multiplicity $r_i = p$ then the basis is interpolatory in ξ_i. Nevertheless, inside the knot spans the basis functions are C^{p-1}-



(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

Figure 3.1 – Univariate B-spline basis functions of polynomial degrees $p = 1, 2$, and 3 obtained from the knot vectors $\Xi = \left\{\{0\}^{p+1}, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \{1\}^{p+1}\right\}$ and globally C^{p-1}-continuous in $\widehat{I} = (0, 1)$.

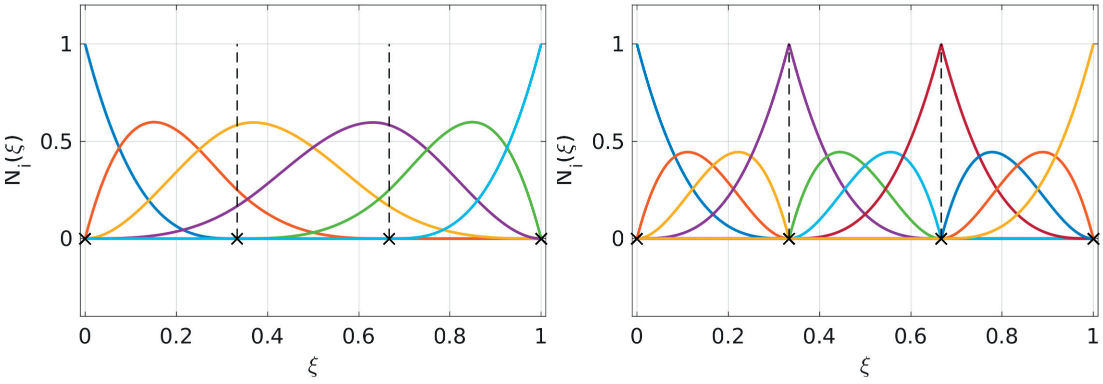
continuous. In this thesis, only *open* knot vectors are considered, for which the first and last knots are always repeated $p+1$ times, regardless of the multiplicity of the other knots (and thus also of the global continuity of the basis functions); this also means that the basis is interpolatory at the boundaries of the knot vector. Examples of univariate B-Spline basis functions of different polynomial degrees and maximal global continuity are represented in Figure 3.1, while in Figure 3.3 an example of B-Spline basis functions built from a knot vector composed by knots with different multiplicity is reported instead; the different multiplicity of the knots defines the degree of continuity of the basis functions across the knots. In addition, in order to visually compare the B-Spline basis functions with the FEM basis, in Figure 3.2 basis functions over three mesh elements are drawn: Figures 3.2a and 3.2b show C^2- and C^0-continuous B-Spline basis functions, respectively, of polynomial order $p = 3$ and Figure 3.2c the standard Lagrangian $\mathbb{P}_3$ FEM basis.

The NURBS basis functions (an acronym for Non-Uniform Rational B-Splines) of degree p are obtained as projective transformations of the B-Spline basis functions $\widehat{N}_{i,p}$ defined in Eqs. (3.1.1) and (3.1.2), reading:

$$\widehat{R}_{i,p}(\boldsymbol{\xi}) := \frac{w_i}{\sum\limits_{j=1}^{n_{bf}} w_j \widehat{N}_{j,p}(\boldsymbol{\xi})} \widehat{N}_{i,p}(\boldsymbol{\xi}), \tag{3.1.3}$$

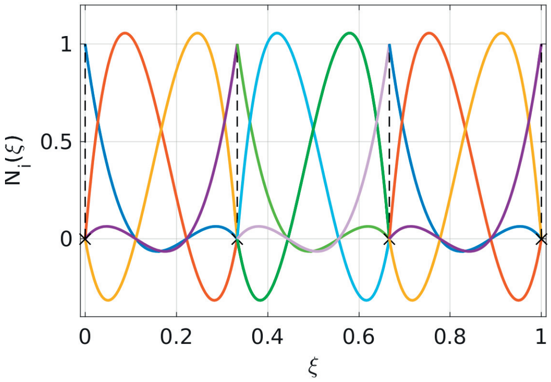
where $w_i \in \mathbb{R}$, for $i = 1, \ldots, n_{bf}$ are the weights of the transformation. From now on, the subscript p will be omitted from the basis functions $\widehat{N}_i$ and $\widehat{R}_i$, to simplify the notation. The NURBS basis functions share the basic properties of the B-Spline basis functions; for more, including implementation details, see [37].

By using the basis functions defined in Eq. (3.1.3), a NURBS curve in $\mathbb{R}^3$ is defined through the



(a) B-Spline basis, C^2-continuous

(b) B-Spline basis, C^0-continuous



(c) FEM basis

Figure 3.2 – Comparison of univariate B-Spline and FEM basis functions of degree $p = 3$: in (a), B-Spline basis functions obtained from the knot vector $\Xi = \left\{\{0\}^4, \frac{1}{3}, \frac{2}{3}, \{1\}^4\right\}$ and globally C^2-continuous; in (b), B-Spline basis built from $\Xi = \left\{\{0\}^4, \left\{\frac{1}{3}\right\}^3, \left\{\frac{2}{3}\right\}^3, \{1\}^4\right\}$ and C^0-continuous at the element boundaries; in (c), standard Lagrangian FEM basis. 3 elements are shown, separated by black dashed lines.

geometrical parametrization:

$$\mathbf{C} : \widehat{I} \subset \mathbb{R} \to \mathbb{R}^3, \qquad \mathbf{C}(\xi) = \sum_{i=1}^{n_{bf}} \widehat{R}_i(\xi)\mathbf{P}_i \qquad \forall \xi \in \widehat{I}, \tag{3.1.4}$$

where $\mathbf{P}_i \in \mathbb{R}^3, i = 1, \ldots, n_{bf}$, are the *control points* in the physical space. The knot vector together with the control points completely define the curve and its properties. In particular, the continuity properties of the curve are defined through its knot vector and are not affected by the control points. Moreover, by moving a single control point one affects the curve only in $p+1$ knot spans. This is of particular interest in the CAD world and it is one of the main reasons for the popularity of NURBS geometries in industrial design. The linear combination of the control points is called *control polygon*. Thanks to the non-negative and compactly

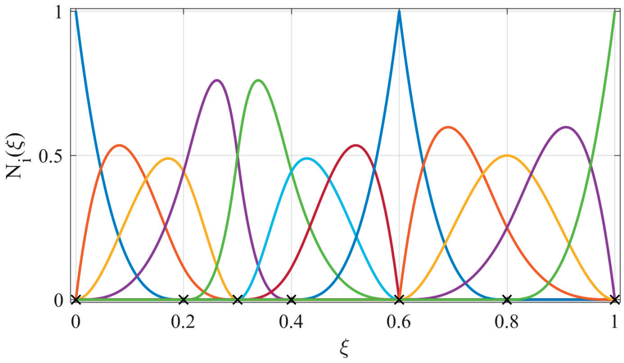


Figure 3.3 – Univariate B-spline basis functions of degree $p = 3$ obtained from the knot vector $\Xi = \left\{\{0\}^4, \frac{1}{5}, \left\{\frac{3}{10}\right\}^2, \frac{2}{5}, \left\{\frac{3}{5}\right\}^3, \frac{4}{5}, \{1\}^4\right\}$, in which the knots have different multiplicity. In particular, the basis functions are C^2-continuous across the knots $\frac{1}{5}$, $\frac{2}{5}$, and $\frac{4}{5}$, C^1-continuous across the knot $\frac{3}{10}$, and C^0-continuous (i.e. interpolatory) at the knot $\frac{3}{5}$.

supported basis functions, which form a partition of unity, the control polygon completely contains the curve; this is known as the *convex hull* property. An example of NURBS curve is sketched in Figure 3.4a.

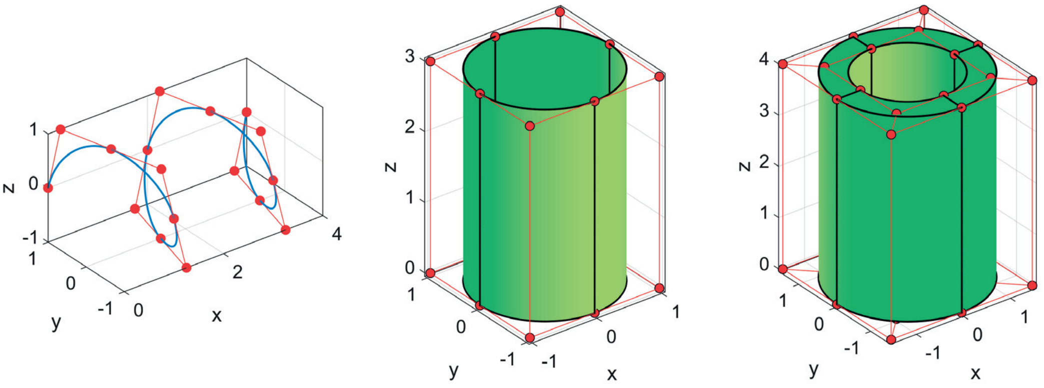
3.2 Multivariate B-Spline and NURBS

The construction adopted for monodimensional parametric domains can be extended to the multidimensional case. In particular, consider the d-dimensional parametric domain $\widehat{\Omega}$ built as product of d reference intervals. Each reference interval $\alpha = 1,\ldots,d$ is divided into knots, which define the knot vector $\Xi_\alpha = \{\xi_{\alpha,i}\}_{i=1}^{n_\alpha} \in \mathbb{R}^{n_\alpha}$. The knot vector Ξ_α defines a univariate B-Spline basis $\{\widehat{R}_{\alpha,i}\}_{i=1}^{n_{bf,\alpha}}$ of polynomial degree p_α, with $n_{bf,\alpha} = n_\alpha - p_\alpha - 1$ being the cardinality of the basis functions. The tensor product of the knot vectors defines a natural partition of the parametric domain into subregions, which represent the *mesh elements* in the multivariate case. Specifically, in the case $d = 2$, by considering a rectangular parametric domain $\widehat{\Omega} \subset \mathbb{R}^2$ (which, in case of normalized knot vectors, is defined as $\widehat{\Omega} \equiv [0,1]^2$), the bivariate B-Spline basis functions in each point $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \widehat{\Omega}$ are defined by the tensor product of the monodimensional basis functions on each parametric direction, reading:

$$\widehat{N}_{i,j}(\boldsymbol{\xi}) := \widehat{N}_{1,i}(\xi_1)\,\widehat{N}_{2,j}(\xi_2), \tag{3.2.1}$$

for $i = 1,\ldots,n_{bf,1}$ and $j = 1,\ldots,n_{bf,2}$, for a total of $n_{bf} = n_{bf,1}\,n_{bf,2}$ basis functions. The corresponding bivariate NURBS basis functions are then defined as:

$$\widehat{R}_{i,j}(\boldsymbol{\xi}) := \frac{w_{i,j}}{\sum_{k=1}^{n_{bf,1}} \sum_{l=1}^{n_{bf,2}} w_{k,l}\,\widehat{N}_{k,l}(\boldsymbol{\xi})}\widehat{N}_{i,j}(\boldsymbol{\xi}), \tag{3.2.2}$$



(a) Curve (b) Surface (c) Volume

Figure 3.4 – Different NURBS geometries. (a) is a NURBS curve of degree $p = 2$. (b) is a NURBS surface, defined through the tensor product of the two univariate basis of degrees $p = 1$ and $p = 2$, built by the knot vectors $\Xi_1 = \{\{0\}^2, \{1\}^2\}$ and $\Xi_2 = \left\{\{0\}^3, \{\frac{1}{4}\}^2, \{\frac{1}{2}\}^2, \{\frac{3}{4}\}^2, \{1\}^3\right\}$, respectively. (c) is a NURBS volume, defined through the tensor product of the three univariate basis of degrees $p = 1$, $p = 2$, and $p = 1$, built by the knot vectors $\Xi_1 = \{\{0\}^2, \{1\}^2\}$, $\Xi_2 = \left\{\{0\}^3, \{\frac{1}{4}\}^2, \{\frac{1}{2}\}^2, \{\frac{3}{4}\}^2, \{1\}^3\right\}$, and $\Xi_3 = \{\{0\}^2, \{1\}^2\}$, respectively. In all the three cases, the control points are represented by red dots, the control polygon/net/lattice is highlighted in red, and the mesh elements are divided by black lines.

for $i = 1, \ldots, n_{bf,1}$ and $j = 1, \ldots, n_{bf,2}$. By using the basis in Eq. (3.2.2), a NURBS surface in $\mathbb{R}^3$ is then defined through the parametrization:

$$\mathbf{S} : \widehat{\Omega} \subset \mathbb{R}^2 \to \mathbb{R}^3, \qquad \mathbf{S}(\boldsymbol{\xi}) := \sum_{i=1}^{n_{bf,1}} \sum_{j=1}^{n_{bf,2}} \widehat{R}_{i,j}(\boldsymbol{\xi})\, \mathbf{P}_{i,j} \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \tag{3.2.3}$$

where the control points $\left\{\mathbf{P}_{i,j}\right\}_{i \in [1, n_{bf,1}], j \in [1, n_{bf,2}]}$ describe a *control net*. Example of a NURBS surface is shown in Figure 3.4b. Similarly to Eq. (3.2.1), by considering the parallelepipedic parametric domain $\widehat{\Theta} \subset \mathbb{R}^3$ (being $\widehat{\Theta} \equiv [0,1]^3$ in case of normalized knot vectors), trivariate B-Spline basis functions in each point $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3) \in \widehat{\Theta}$ are defined as:

$$\widehat{N}_{i,j,k}(\boldsymbol{\xi}) := \widehat{N}_{1,i}(\xi_1)\, \widehat{N}_{2,j}(\xi_2)\, \widehat{N}_{k,3}(\xi_3), \tag{3.2.4}$$

for $i = 1, \ldots, n_{bf,1}$, $j = 1, \ldots, n_{bf,2}$, and $k = 1, \ldots, n_{bf,3}$, for a total of $n_{bf} = n_{bf,1}\, n_{bf,2}\, n_{bf,3}$ basis functions. The corresponding trivariate NURBS basis functions then read:

$$\widehat{R}_{i,j,k}(\boldsymbol{\xi}) := \frac{w_{i,j,k}}{\sum_{l=1}^{n_{bf,1}} \sum_{m=1}^{n_{bf,2}} \sum_{o=1}^{n_{bf,3}} w_{l,m,o}\, \widehat{N}_{l,m,o}(\boldsymbol{\xi})} \widehat{N}_{i,j,k}(\boldsymbol{\xi}), \tag{3.2.5}$$

for $i = 1, \dots, n_{bf,1}$, $j = 1, \dots, n_{bf,2}$, and $k = 1, \dots, n_{bf,3}$. By using the basis in Eq. (3.2.5), a NURBS volume in $\mathbb{R}^3$ is then defined through the parametrization:

$$\mathbf{V} : \widehat{\Theta} \subset \mathbb{R}^3 \to \mathbb{R}^3, \qquad \mathbf{V}(\boldsymbol{\xi}) := \sum_{i=1}^{n_{bf,1}} \sum_{j=1}^{n_{bf,2}} \sum_{k=1}^{n_{bf,3}} \widehat{R}_{i,j,k}(\boldsymbol{\xi})\, \mathbf{P}_{i,j,k} \qquad \forall \boldsymbol{\xi} \in \widehat{\Theta}, \tag{3.2.6}$$

where the control points $\{\mathbf{P}_{i,j,k}\}_{i \in [1, n_{bf,1}], j \in [1, n_{bf,2}], k \in [1, n_{bf,3}]}$ define a *control lattice*. Example of a NURBS volume is reported in Figure 3.4c. In general, the inclusion of weights in the construction of the NURBS basis functions is necessary to exactly represent geometries such as conics, for which B-Spline basis functions cannot be used.

Regardless of the dimensionality of the parametric domain, the basis functions built from the tensor product of the univariate basis can be enumerated such that they are indexed by a single index $1 \leq i \leq n_{bf}$; the same enumeration can be applied also to the control points building the control net/lattice. In this way, by choosing a suitable parametric space $\widehat{\Omega}$, all NURBS geometries, defined by the parametrizations in Eqs. (3.1.4), (3.2.3), and (3.2.6), can be described by the general NURBS geometrical mapping:

$$\mathbf{X} : \widehat{\Omega} \to \mathbb{R}^3, \qquad \mathbf{X}(\boldsymbol{\xi}) := \sum_{i=1}^{n_{bf}} \widehat{R}_i(\boldsymbol{\xi}) \mathbf{P}_i \qquad \forall \boldsymbol{\xi} \in \widehat{\Omega}, \tag{3.2.7}$$

for $i = 1, \dots, n_{bf}$. Moreover, multivariate B-Spline and NURBS basis functions share the main properties of their univariate counterparts; for more details, see [37].

3.3 NURBS function spaces

Let $\widehat{\Omega} \subset \mathbb{R}^d$ be the parametric domain, with $d = 1$, 2, or 3, and $\Omega \subset \mathbb{R}^3$ the physical domain, described by a geometrical mapping $\mathbf{X} : \widehat{\Omega} \to \Omega$ of the type defined in Eq. (3.2.7). The NURBS function space $\widehat{\mathcal{N}_h}$ over the parametric domain $\widehat{\Omega}$ is defined as:

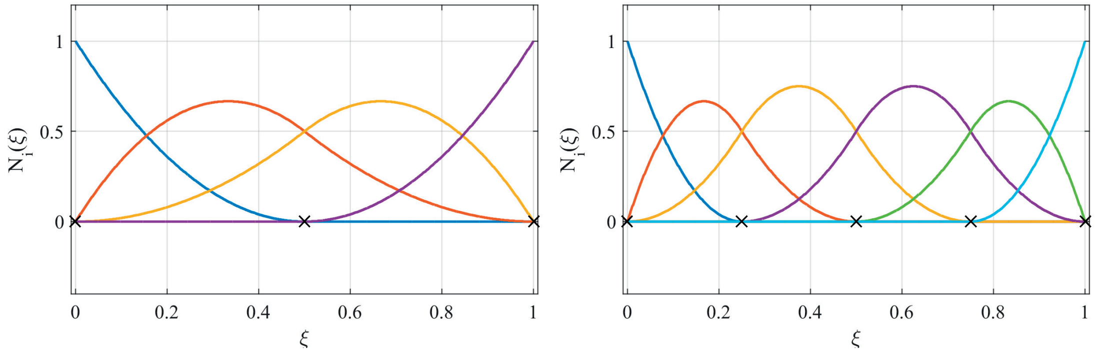
$$\widehat{\mathcal{N}_h} := \operatorname{span}\{\widehat{R}_i, \quad i = 1, \dots, n_{bf}\} \tag{3.3.1}$$

and its counterpart in the physical domain Ω as:

$$\mathcal{N}_h := \operatorname{span}\{\widehat{R}_i \circ \mathbf{X}^{-1}, \quad i = 1, \dots, n_{bf}\}. \tag{3.3.2}$$

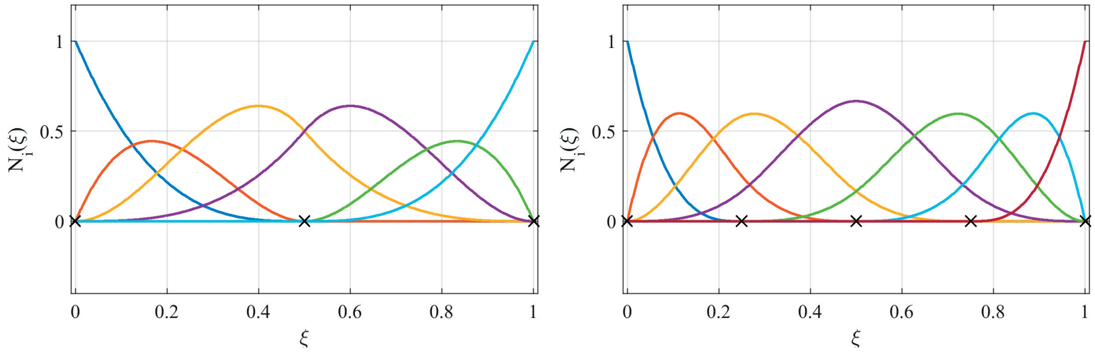
According to the isogeometric concept, these spaces will be used to build the trial function spaces for the approximation of PDEs. The subscript h refers to the characteristic size of the mesh elements, and is usually defined as the maximum diameter of the mesh elements in the physical space [35, 43]. The computational domain Ω is usually represented at its coarsest level of discretization, from which the spaces $\widehat{\mathcal{N}_h}$ and $\mathcal{N}_h$ are built; the coarsest approximation is often already suitable to reproduce exactly the surface geometry.

For NURBS, there are three different kinds of refinement procedures which permit the enrich-



(a) Original basis

(b) Basis after knot insertion



(c) Basis after order elevation

(d) Basis after *k*-refinement

Figure 3.5 – Different refinement strategies applied to a B-Spline basis originally of degree $p = 2$ and built from the knot vector $\Xi = \left\{\{0\}^3, \frac{1}{2}, \{1\}^3\right\}$ (shown in (a)): 1 level of knot insertion, 1 level of order elevation, and 1 level of k-refinement are shown in (b), (c), and (d), respectively.

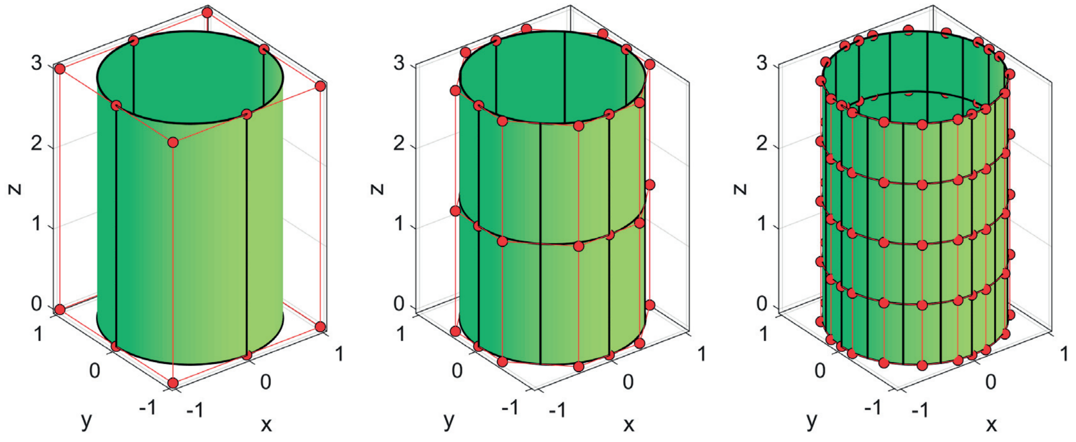
ment of the NURBS function spaces [35]. Consider, for instance, the knot vector:

$$\Xi = \left\{\{0\}^3, \frac{1}{2}, \{1\}^3\right\}, \tag{3.3.3}$$

which defines the B-Spline basis functions of degree $p = 2$ shown in Figure 3.5a. The *knot insertion* refers to the procedure of splitting the knot spans by introducing new knots, effectively increasing the number of mesh elements and basis functions; when the continuity between elements is preserved, inserting knots with the correct multiplicity can be compared to the *h-refinement* procedure of the FEM. As an example, after one level of knot insertion, the knot vector Ξ, originally defined in Eq. (3.3.3), becomes:

$$\Xi_h = \left\{\{0\}^3, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \{1\}^3\right\}, \tag{3.3.4}$$

for which the (nontrivial) knot spans are split in two. The corresponding B-Spline basis functions are shown in Figure 3.5b. The process of increasing the polynomial degree of the



(a) Original (b) 1 level of h-ref. (c) 2 level of h-ref.

Figure 3.6 – h-refinement (knot insertion) applied to the NURBS geometry in (a): 1 level of h-ref. in (b) and 2 levels of h-ref. in (c). Under refinement, the original geometrical representation is maintained exactly, while the number of mesh elements (knot spans), basis functions, and control points is increased.

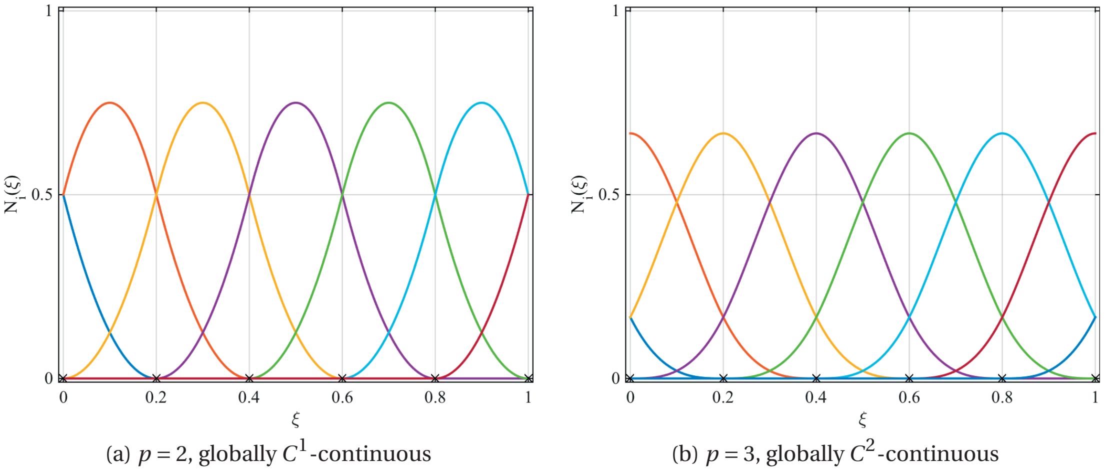
basis functions while preserving the existing continuity across the edges of the elements, is called *order elevation* and it is closely related to the FEM *p-refinement*. Applied to Ξ of Eq. (3.3.3), it results in the following knot vector:

$$\Xi_p = \left\{ \{0\}^4, \left\{ \frac{1}{2} \right\}^2, \{1\}^4 \right\}. \tag{3.3.5}$$

The basis functions defined by Ξ_p and shown in Figure 3.5c are now of polynomial degree $p = 3$. Moreover, to preserve the internal continuity of the original basis, the knot $\frac{1}{2}$ has been duplicated, so that the resulting basis functions are globally C^{p-2} $(= C^1)$ -continuous. Finally, B-splines and NURBS benefit from another form of refinement, which does not have counterparts in FEM, the *k-refinement*, for which the degree of the basis functions is firstly elevated and then new unique knots are inserted, maintaining the highest possible continuity across the elements. One level of k-refinement applied to the knot vector Ξ results in:

$$\Xi_k = \left\{ \{0\}^4, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \{1\}^4 \right\}, \tag{3.3.6}$$

which defines the B-Spline basis functions shown in Figure 3.5d. All these refinement procedures must preserve the original representation of the underlying geometry while enriching the dimension and approximation properties of the function space; in this regard, an example of refinement of a cylindrical geometry is reported in Figure 3.6. For a detailed description about these refinement procedures, the reader is referred to [34, 35] and, for implementation details and algorithms, to [37].



(a) $p = 2$, globally C^1-continuous

(b) $p = 3$, globally C^2-continuous

Figure 3.7 – Periodic univariate B-spline basis functions.

3.3.1 Periodic NURBS function spaces

To build closed curves and closed or partially closed surfaces, especially conics, such as circles, spheres, cylinders, or tori, the usual NURBS construction is such that the basis functions are only globally C^0-continuous in $\widehat{\Omega}$ and Ω [35, 37]. As an example, consider the cylinder shown in Figures 3.4b and 3.6. It is represented as a NURBS surface defined through the tensor product of two univariate basis of polynomial degree $p_1 = 1$ and $p_2 = 2$ in the two parametric directions, characterized by the knot vectors:

$$\Xi_1 = \left\{\{0\}^2, \{1\}^2\right\} \qquad \text{and} \qquad \Xi_2 = \left\{\{0\}^3, \left\{\frac{1}{4}\right\}^2, \left\{\frac{1}{2}\right\}^2, \left\{\frac{3}{4}\right\}^2, \{1\}^3\right\}, \tag{3.3.7}$$

respectively. The NURBS basis along the second parametric direction is of degree $p_2 = 2$ but just C^0-continuous at the patch and element boundaries, as evident from the multiplicities of the internal knots. This is the standard procedure for representing conics with single-patch NURBS, see [35, 37]. Nevertheless, for the approximation of PDEs over the surface, one may be interested in considering a NURBS space $\mathcal{N}_h$ for which the basis functions feature higher global continuity degree over the physical domain, especially for the the numerical approximation of high order PDEs. In order to build such basis functions, we consider an approach based on algebraic constraints and local linear transformations of the NURBS basis functions originally used to represent the surface, leading to a subparametric approach, as in [105]. Such smooth basis functions can be defined by suitably using the k-refinement procedure and enforcing periodic conditions on the original basis functions. For example, consider again the cylinder whose knot vectors are reported in Eq. (3.3.7); the knot vector Ξ_2, characterizing the NURBS basis along the second parametric direction, is replaced by the following:

$$\Xi_2 = \left\{ \{0\}^3, \left\{\frac{1}{4}\right\}^3, \left\{\frac{1}{2}\right\}^3, \left\{\frac{3}{4}\right\}^3, \{1\}^3 \right\}, \tag{3.3.8}$$

and then appropriately h-refined. The increment of multiplicity of the internal knots in (3.3.8) leads the corresponding NURBS basis functions to be discontinuous at the (original, before the h-refinement) element boundaries. The requested degree of continuity is then restored at each of these discontinuities and at the patch boundaries by enforcing algebraic constraints between the DOFs and by transforming the NURBS basis functions which have support near the discontinuities and the boundaries, as done in [44] for planar surfaces and in [105] for 3D surfaces. In particular, we define a periodic NURBS function space by applying a linear transformation operator $\mathbf{T}^{per} \in \mathbb{R}^{n_{bf} \times n_{bf}}$ to the basis functions $\widehat{\mathbf{R}} := \{\widehat{R}_i\}_{i=1}^{n_{bf}}$ which define the NURBS space $\widehat{\mathcal{N}_h}$, as:

$$\widehat{\mathbf{R}}^{per} := \mathbf{T}^{per} \widehat{\mathbf{R}}, \tag{3.3.9}$$

thus obtaining, together with the necessary constraints among the DOFs, a set of periodic basis functions $\widehat{\mathbf{R}}^{per}$. The periodic NURBS function spaces are then constructed as:

$$\widehat{\mathcal{N}_h^{per}} := \text{span}\left\{\widehat{R}_i^{per}, \quad i = 1, \ldots, n_{bf}\right\} \tag{3.3.10}$$

and

$$\mathcal{N}_h^{per} := \text{span}\left\{\widehat{R}_i^{per} \circ \mathbf{X}^{-1}, \quad i = 1, \ldots, n_{bf}\right\}. \tag{3.3.11}$$

Such procedure allows the construction of periodic basis functions preserving high order continuity internally and across the boundaries of the NURBS patch. Examples of univariate periodic B-Spline basis functions of polynomial degrees $p = 2$ and $p = 3$ and globally C^1- and C^2-continuous, respectively, are shown in Figure 3.7.

3.4 NURBS-based Isogeometric Analysis

Let $\Omega \subset \mathbb{R}^3$ be a domain defined as a NURBS curve, surface, or volume through a geometrical mapping of the form (3.2.7). Now, consider the prototype elliptic PDE of the $2m$-th order in Eq. (1.2.1) for $m = 2,3$ defined on Ω and whose variational form is reported here for convenience:

$$\begin{aligned} &\text{find } u \in V_0 \text{ such that} \\ &\qquad a(\psi, u) = F(\psi) \qquad \forall \psi \in V_0, \end{aligned} \tag{3.4.1}$$

where $V_0 \subset H^m(\Omega)$. In this thesis, the numerical approximation of such PDEs is obtained by means of NURBS-based IGA in the framework of the Galerkin method. By considering the NURBS function space $\mathcal{N}_h$ defined in Eq. (3.3.2), used also for the geometrical representation in Eq. (3.2.7), an approximate solution $u_h \in \mathcal{N}_h$ is sought such that:

$$u_h(\mathbf{p}) = \sum_{i=1}^{n_{bf}} \left(\widehat{R}_i \circ \mathbf{X}^{-1}\right)(\mathbf{p})\, U_i \qquad \forall \mathbf{p} \in \Omega, \tag{3.4.2}$$

where $\mathbf{U} = \left(U_1, \ldots, U_{n_{bf}}\right)^T \in \mathbb{R}^{n_{bf}}$ is the vector of *control variables*, corresponding to the unknowns of the discrete problem. For IGA within the Galerkin framework, u_h is obtained by solving the finite dimensional problem:

$$\begin{aligned} &\text{find } u_h \in V_h \text{ such that} \\ &\qquad a(\psi_h, u_h) = F(\psi_h) \qquad \forall \psi_h \in V_h, \end{aligned} \tag{3.4.3}$$

where $V_h := V_0 \cap \mathcal{N}_h$. Then, following Eq. (1.2.4), problem (3.4.3) is rewritten on the parametric domain $\widehat{\Omega}$ as:

$$\begin{aligned} &\text{find } \widehat{u}_h \in \widehat{V}_h \text{ such that} \\ &\qquad \widehat{a}\left(\widehat{\psi}_h, \widehat{u}_h\right) = \widehat{F}\left(\widehat{\psi}_h\right) \qquad \forall \widehat{\psi}_h \in \widehat{V}_h, \end{aligned} \tag{3.4.4}$$

where $\widehat{V}_h := \widehat{V}_g \cap \widehat{\mathcal{N}_h}$. The solution $\widehat{u}_h$ defined on the parametric domain $\widehat{\Omega}$ reads:

$$\widehat{u}_h(\boldsymbol{\xi}) = \sum_{i=1}^{n_{bf}} \widehat{R}_i(\boldsymbol{\xi})\, U_i. \tag{3.4.5}$$

The same procedure is followed when considering problems defined on closed surfaces, like for Eqs. (1.2.3) and (1.2.5).

The high order Laplace–Beltrami eigenvalue problem defined in Eq. (1.2.18) is discretized by means of NURBS-based IGA in a similar way. On the parametric domain $\widehat{\Omega}$, the discrete problem reads:

$$\begin{aligned} &\text{find } \widehat{u}_h \in \widehat{V}_h \text{ and } \lambda_h \in \mathbb{R} \text{ such that} \\ &\qquad \widehat{a}(\widehat{\psi}_h, \widehat{u}_h) = \lambda_h \widehat{b}(\widehat{\psi}_h, \widehat{u}_h) \qquad \forall \widehat{\psi}_h \in \widehat{V}_h, \end{aligned} \tag{3.4.6}$$

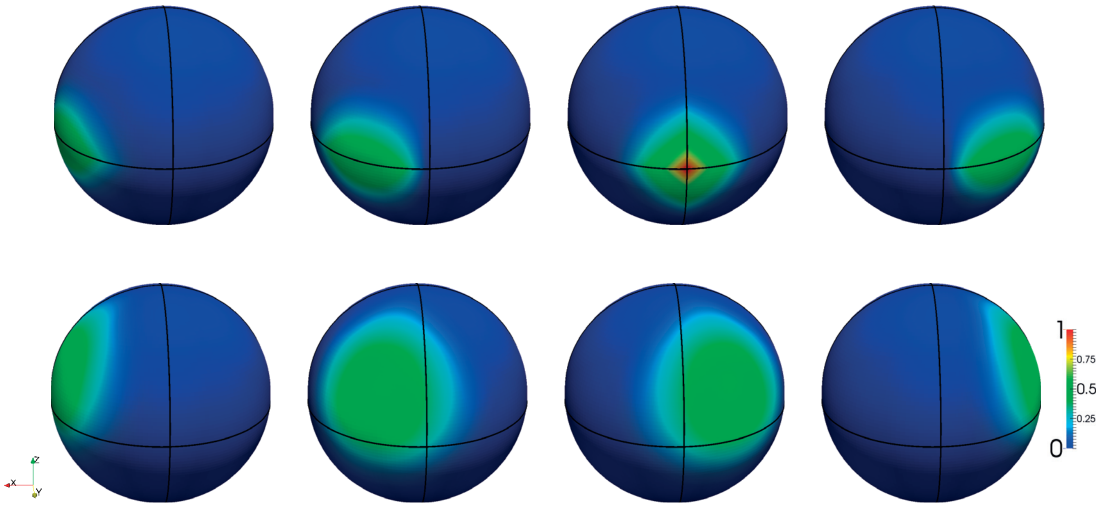


Figure 3.8 – Examples of NURBS basis functions of degree $p = 2$ on the sphere; globally C^0-continuous basis functions (top row) and C^2-continuous a.e. basis functions (bottom row).

where $\widehat{a}(\cdot,\cdot)$ is either the bilinear form (1.2.10) or (1.2.16), $\widehat{b}(\cdot,\cdot)$ is the bilinear form of Eq. (1.2.21), and $\widehat{V}_h$ is a suitable NURBS function space whose elements satisfy the differentiability requirements associated to the operators under consideration (that is $H_0^m(\widehat{\Omega})$ or $H^m(\widehat{\Omega})$ for an original differential operator of order $2m$). The eigenvalues λ_h are real since the considered bilinear forms are symmetric.

We remark that, depending on the order of the PDE, it is necessary to satisfy different differentiability requirements on u_h and ψ_h. Specifically, for PDEs of order $2m$ the trial and test function spaces should be subspaces of the function space $H^m(\Omega)$, for $m \geq 1$. This requirement is satisfied for example if the basis functions are at least globally C^{m-1}-continuous on the surface Ω. This is a distinguishing feature of NURBS-based IGA: the global continuity of the NURBS basis functions can be tuned directly as they are constructed or enforced by means of suitable refinement procedures, as seen in Sections 3.3.1. As mentioned before, if the original basis functions representing the geometry do not meet the necessary continuity requirement, by using a subparametric approach globally C^{m-1}-continuous basis functions on closed or partially closed surfaces can be defined, through suitable local linear transformations and algebraic constraints. In these cases, the NURBS spaces for the IGA approximation are built as e.g. $\widehat{V}_h^{per} = \widehat{V} \cap \widehat{\mathcal{N}}_h^{per}$ and $V_h^{per} = V \cap \mathcal{N}_h^{per}$ from Eqs. (3.3.10) and (3.3.11). In this respect, for fourth order problems ($m = 2$) at least globally C^1-continuous basis functions are used and at least globally C^2-continuous basis functions for sixth order problems ($m = 3$). However, some closed surfaces of practical interest, as e.g. the sphere, can only be built by NURBS basis functions which do not possess the required degree of global continuity, as the presence of localized singularities of the geometrical mapping prevents the basis functions to be globally C^{m-1}-continuous on Ω. For example, the standard single patch construction of the sphere involves the presence of two singularities at the poles, for which the subparametric approach leads to the use of NURBS spaces where the functions are only C^0-continuous at the poles and up to C^{m-1}-continuous elsewhere. In this case, the basis functions are C^{m-1}-continuous a.e. in Ω and globally C^{m-1}-continuous in $\widehat{\Omega}$. Nevertheless, even in presence of these pointwise singularities, we show by numerical evidence that the spaces V_h^{per} obtained in this manner yield the same convergence rates for the errors of the standard conformal Galerkin method using subspaces of $H^m(\Omega)$. This is true even if the basis functions are not pointwise C^{m-1}-continuous across the poles of the sphere, as it will be shown in Sections 4.1.1 and 4.1.3. Nevertheless, the basis functions and their derivatives are in any case evaluated in the Gauss–Legendre quadrature nodes, which are internal to the mesh elements and therefore not laying in the singularity points corresponding to the poles of the sphere. As example, in Figure 3.8 eight NURBS basis functions of degree $p = 2$ on the sphere are reported: the top row shows the original non-transformed NURBS basis functions which are only C^0-continuous across the equator and four meridians (indicated as black lines); the bottom row shows four periodic basis functions which have been transformed as in Eq. (3.3.9) and are C^1-continuous on the sphere except at the poles.

3.4.1 A priori error estimate

In the next chapter, we will test the accuracy of the NURBS-based IGA approximation of problem (3.4.3) under h-refinement. In this respect, in [40] error estimates for the approximation of elliptic high order PDEs by means of NURBS-based IGA are available for 2D and 3D domains. In particular, Theorem 3.3 of [40] is reported for reference in the following:

Theorem 7. *For the elliptic PDE of order* $2m$ *in* (3.4.3) *defined in* $\Omega \subset \mathbb{R}^d$*, for* $d = 2, 3$*, endowed with homogeneous essential boundary conditions, let* $u \in H^r(\Omega)$*, for* $r \geq m$*, be the exact solution of problem* (3.4.3) *and* u_h *be the approximate solution, obtained by means of NURBS-based IGA in the framework of the Galerkin method. Then, the following a priori error estimate in lower order norms* $H^\sigma(\Omega)$*, with* $0 \leq \sigma \leq m$*, holds:*

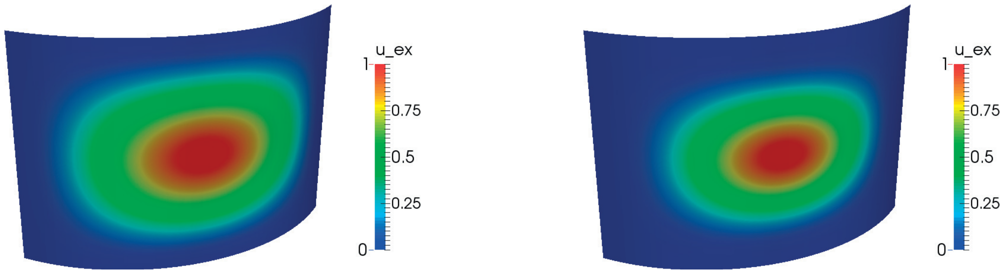
$$\|u - u_h\|_{H^\sigma(\Omega)} \leq C_{shape} h^\beta \|u\|_{H^r(\Omega)}, \tag{3.4.7}$$

where $\beta := \min\{\delta - \sigma, 2(\delta - m)\}$*, with* $\delta := \min\{r, p+1\}$*,* p *is the polynomial degree of the basis functions, and* C_{shape} *is a constant independent of* h*.*

Theorem 7 is stated and proven in [40] for problems for which the dimension of the parametric domain $\widehat{\Omega} \subset \mathbb{R}^k$ is equal to the dimension of the physical domain $\Omega \subset \mathbb{R}^d$, i.e. $k = d$. It can, however, be extended to the case of surfaces with boundaries, similarly to the results in [43] for second order surface PDEs. The case of fully closed surfaces is not covered by the theorem. Nevertheless, in the next chapter we will report numerical error analysis also for the case of closed surfaces (as the sphere), showing consistent convergence rates.

4 IGA for high order surface PDEs

In this chapter, we consider the numerical approximation of the high order surface PDEs introduced in Chapter 1, by means of NURBS-based IGA, which we briefly described in Chapter 3. Firstly, we study the approximation of the steady problems described in Section 4.1, in particular the Laplace–Beltrami biharmonic and triharmonic problems and the Laplace–Beltrami eigenvalue problem. Then, we consider the time-dependent phase field problems introduced in Section 4.2, in particular the Cahn–Hilliard and phase field crystal equations. The solution of these problems aims at establishing a computational framework for the geometric PDEs solved in Chapter 5. We provide numerical results both on open and closed surfaces. Most of these results were published in [105], of which this chapter represents an overview.



(a) Solution of Test 4.1.1

(b) Solution of Test 4.2.1

Figure 4.1 – Exact solutions u of the biharmonic and triharmonic problems (Tests 4.1 and 4.2) on a quarter of cylinder.

4.1 Numerical approximation of steady PDEs

In this section, we propose three test problems: the numerical approximation of the Laplace–Beltrami biharmonic (Test 4.1) and triharmonic (Test 4.2) problems on a quarter of cylinder, a cylinder, and a unit sphere. Then, we approximate high order Laplace–Beltrami eigenvalue problems on the unit sphere (Test 4.3).

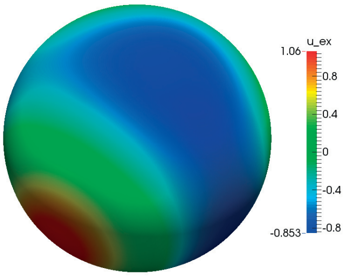
4.1.1 Test 4.1. Laplace–Beltrami biharmonic problem

We consider the numerical approximation of the Laplace–Beltrami biharmonic problem described in Section 1.2.1 on different geometries. In particular, we consider different domains Ω, namely:

- a quarter of cylinder in Test 4.1.1;
- a cylinder, closed on the lateral surface, but of finite length in Test 4.1.2;
- a unit sphere (a fully closed surface) in Test 4.1.3.

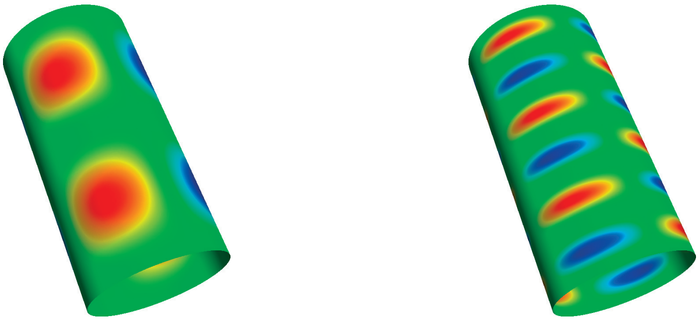
We consider problem (1.2.6) if Ω has a boundary, otherwise, for closed surfaces, problem (1.2.7) is solved instead. Since $V \subseteq H^2(\Omega)$, for the discretization of the problem (and the exact representation of the geometry) we consider NURBS bases of degree $p \geq 2$ and at least globally C^1-continuous a.e. in the parametric domain $\widehat{\Omega}$. Regarding the cylinder and the sphere, we enforce global H^2-regularity in the physical space across the closed surface by means of the periodic transformations of the NURBS basis functions described in Section 3.3.1. We remark that, in this way, we obtain basis functions which are C^{p-1}-continuous everywhere on the closed lateral surface of the cylinder; on the sphere, the basis functions are C^{p-1}-continuous a.e., with exception at the poles.

We consider a right-hand-side function f hand-crafted in such a way that the exact solution u is known and is globally C^∞-continuous on Ω. Specifically, for Test 4.1.1, with the domain $\Omega = (0, \pi/2) \times (0, 1)$ representing, in cylindrical coordinates (θ, z), a quarter of cylinder with



(a) Solution of Tests 4.1.3 and 4.2.3

Figure 4.2 – Exact solution u of the biharmonic and triharmonic problems (Tests 4.1 and 4.2) on a unit sphere.



(a) Solution of Test 4.1.2 (b) Solution of Test 4.2.2

Figure 4.3 – Exact solutions u of the biharmonic and triharmonic problems (Tests 4.1 and 4.2) on a cylinder.

unitary radius and centered at the origin, we consider the following exact solution in cylindrical coordinates (Figure 4.1a):

$$u(\theta, z) = \sin^2(2\theta)\sin^2(\pi z), \tag{4.1.1}$$

where $\theta = \operatorname{atan}(y/z)$, with $\mu = 1$ and $\gamma = 0$. On the cylinder (Test 4.1.2), with domain in cylindrical coordinates being $\Omega = (0, 2\pi) \times (0, 1)$, we consider the following exact solution instead (Figure 4.3a):

$$u(\theta, z) = \sin\left(\sqrt{2} + 2\theta\right)\sin^2\left(\frac{\pi}{2}z\right). \tag{4.1.2}$$

On the sphere of unitary radius centered at the origin (Test 4.1.3) we use (Figure 4.2a):

$$u(x, y, z) = (x - x_0)(y - y_0)^2 - (y - y_0)(z - z_0)^2 + (x - x_0)^2(z - z_0), \tag{4.1.3}$$

with $x_0 = 0.05$, $y_0 = 0.1$, and $z_0 = 0.15$.

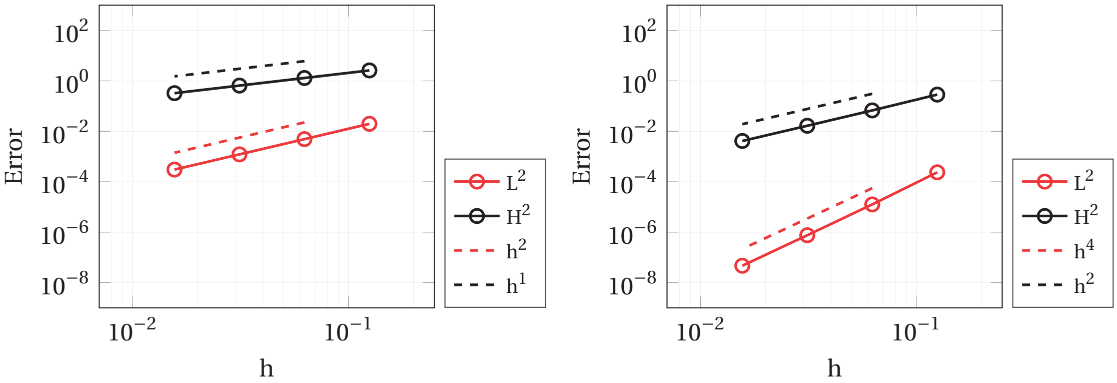


Figure 4.4 – Test 4.1.1. Biharmonic problem on the quarter of cylinder. Errors in norms $H^2(\Omega)$ and $L^2(\Omega)$ vs. mesh size h for NURBS bases of degrees $p = 2$ and 3, globally C^1- and C^2-continuous, respectively (logarithmic scales are used on both the axes).

The biharmonic problem is governed by a fourth order operator (for which, by using the notation of Section 1.2, $m = 2$). By applying Theorem 7, when considering exact solutions $u \in H^r(\Omega)$, with $r \geq p+1 \geq m$ (as it is the case with $u \in C^\infty(\Omega)$), the following estimates hold for problem (1.2.6), following directly from the inequality (3.4.7):

$$\begin{aligned} \|u - u_h\|_{H^2(\Omega)} &\leq C_{shape} h^{p-1} \|u\|_{H^r(\Omega)} \\ \|u - u_h\|_{L^2(\Omega)} &\leq \tilde{C}_{shape} h^{\min\{p+1, 2p-2\}} \|u\|_{H^r(\Omega)}. \end{aligned} \tag{4.1.4}$$

In order to compute the norm $H^2(\Omega)$, instead of employing the full seminorm, we use the equivalent norm $L^2(\Omega)$ of the Laplace–Beltrami surface operator. Indeed, for closed surfaces or open surfaces on which the essential boundary conditions are enforced, one has:

$$\|\Delta_\Omega \varphi\|_{L^2(\Omega)} \simeq |\varphi|_{H^2(\Omega)}, \tag{4.1.5}$$

for $\varphi \in H^2(\Omega)$ on a closed surface or $\varphi \in H^2_0(\Omega)$. In Figure 4.4, we report the errors in norms $H^2(\Omega)$ and $L^2(\Omega)$ obtained by the IGA approximation of the biharmonic problem on the quarter of cylinder (Test 4.1.1) under h-refinement, having used NURBS bases of degree $p = 2$ and $p = 3$, which are globally C^1- and C^2-continuous, respectively. We observe that the convergence rates are in agreement with the error estimates (4.1.4). Indeed, the convergence rates are 1 and 2 for the errors in norm $H^2(\Omega)$ using basis of degree $p = 2$ and 3, respectively; similarly, the rates are 2 and 4 for the errors in the norm $L^2(\Omega)$ for $p = 2$ and 3, respectively.

In Figure 4.5, we plot the errors obtained on the cylinder (Test 4.1.2). Differently from Test 4.1.1, the cylinder possesses a closed lateral surface (even if it has a top and bottom boundary), which needed transformation of the NURBS basis functions to enforce the required degree of continuity; nevertheless, the estimates in Eq. (4.1.4) for the errors hold and same convergence rates as Test 4.1.1 are observed.

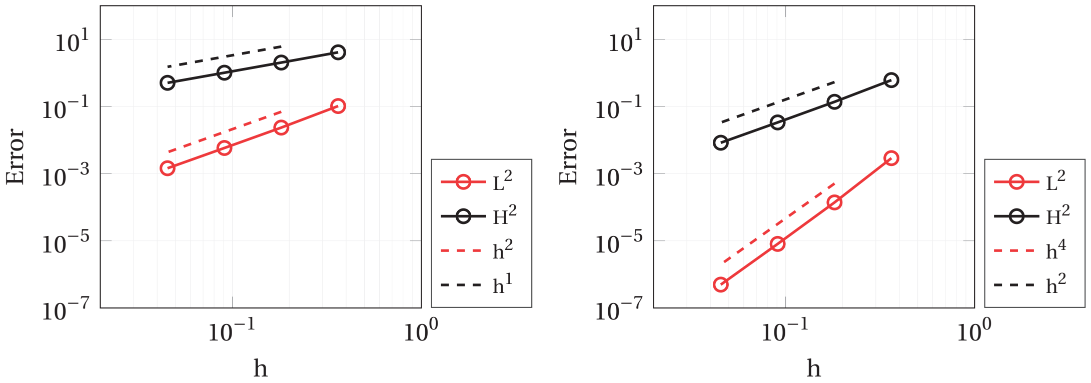


Figure 4.5 – Test 4.1.2. Biharmonic problem on a cylinder. Errors in norms $H^2(\Omega)$ and $L^2(\Omega)$ vs. mesh size h for NURBS bases of degrees $p = 2$ and 3, globally C^1- and C^2-continuous, respectively (logarithmic scales are used on both the axes).

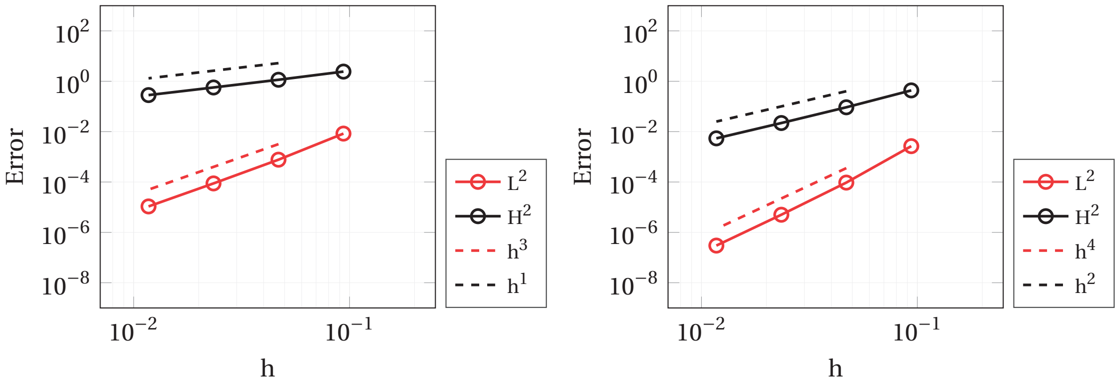


Figure 4.6 – Test 4.1.3. Biharmonic problem on the sphere. Errors in norms $H^2(\Omega)$ and $L^2(\Omega)$ vs. mesh size h for NURBS bases of degrees $p = 2$ and 3, C^1- and C^2-continuous a.e. on Ω, respectively (logarithmic scales are used on both the axes).

The errors obtained for the approximation of the biharmonic problem on the unit sphere (Test 4.1.3) are reported instead in Figure 4.6. In this case, the convergence rates still satisfy the estimate in Eq. (4.1.4). Actually, the convergence rate for the error in norm $L^2(\Omega)$ is higher than predicted from Eq. (4.1.4) since for $p = 2$ it is equal to 3; we provide an explanation of this behavior in the following remark.

Remark 1. *The error estimate* (4.1.4)*, and more in general* (3.4.7)*, is not optimal in lower order norms, at least for problems defined on closed surfaces. Specifically, one would rather expect the error estimate in* $L^2(\Omega)$ *norm to be:*

$$\|u - u_h\|_{L^2(\Omega)} \leq \tilde{C}_{shape} h^{p+1} \|u\|_{H^r(\Omega)}. \tag{4.1.6}$$

This is precisely what we obtain in Figure 4.6 for $p = 2$*, but not for* $p = 3$*. By returning to the error estimate of Eq.* (4.1.4)*, we remark that the threshold* h^{2p-2} *arises from a regularity result*

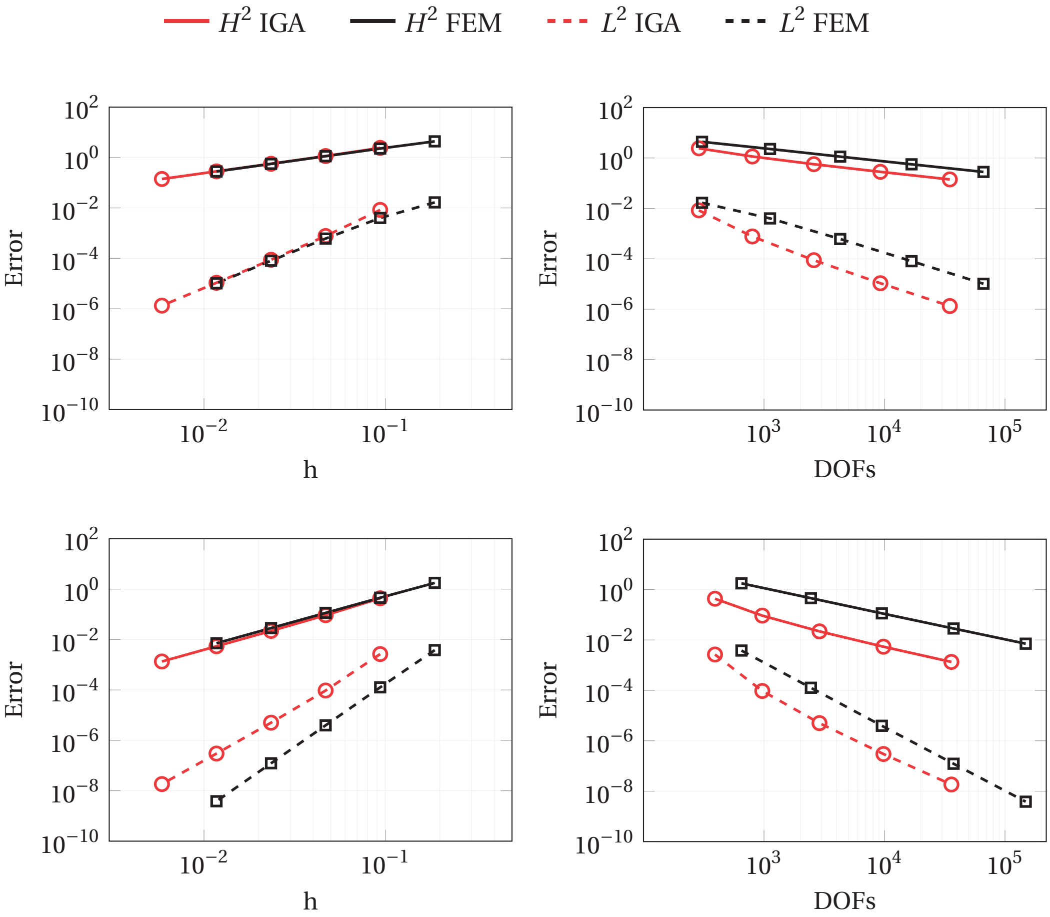


Figure 4.7 – Test 4.1.3. Biharmonic problem on the sphere. Errors in norms $H^2(\Omega)$ and $L^2(\Omega)$ vs. mesh size h (left) and number of DOFs (right), obtained with IGA and FEM for degrees $p = 2$ (top) and 3 (bottom); the IGA approximation uses NURBS basis functions C^1- and C^2-continuous a.e. on Ω, respectively (logarithmic scales are used on both the axes).

for problems with boundaries in the track of the Aubin–Nitsche duality argument [40], as it is actually confirmed for the approximation on the quarter of cylinder in Figure 4.4 and 4.5. The same argument can be applied also in the case of the cylinder as reported in Figure 4.5, which has closed lateral surface but still possesses boundaries at the top and bottom. However, for fully closed surfaces, we numerically find the optimal convergence rate of Eq. (4.1.6) *for the error in norm* $L^2(\Omega)$ *irrespective of the value of* p. *Although we do not have a rigorous proof, we speculate that on closed surfaces the exact solution does not suffer the regularity limitation featured instead for open surfaces, i.e. endowed with boundary* $\partial\Omega$, *and hence the optimal convergence rate* h^{p+1} *can be achieved.*

4.1.2 Test 4.1. Comparison with isoparametric FEM

For the sake of comparison, we also consider the approximation of the biharmonic problem (1.2.7) on the sphere Ω by means of a standard isoparametric FEM discretization. In this case, in view of the FEM approximation based on C^0-continuous Lagrangian basis functions, the following mixed formulation is considered:

$$\text{find } u, v : \Omega \to \mathbb{R} \text{ such that}$$
$$\begin{cases} -\Delta_\Omega u - v = 0 & \text{in } \Omega, \\ -\mu \Delta_\Omega v + \gamma u = f & \text{in } \Omega, \end{cases} \tag{4.1.7}$$

where $v : \Omega \to \mathbb{R}$ is an auxiliary unknown. Problem (4.1.7) in weak form reads:

$$\text{find } u, v \in V \text{ such that}$$
$$\begin{cases} \displaystyle\int_\Omega \nabla_\Omega u \cdot \nabla_\Omega \varphi \, d\Omega - \int_\Omega v \varphi \, d\Omega = 0 & \forall \varphi \in V, \\ \displaystyle\mu \int_\Omega \nabla_\Omega v \cdot \nabla_\Omega \psi \, d\Omega + \gamma \int_\Omega u \psi \, d\Omega = \int_\Omega f \psi \, d\Omega & \forall \psi \in V, \end{cases} \tag{4.1.8}$$

with $V \equiv H^1(\Omega)$. Problem (4.1.8) is discretized using isoparametric FEM of degrees $p = 2$ and 3 on successively finer meshes of the unit sphere. In this case, the geometry is not represented exactly, as it happens instead with NURBS. We remark that, due to the difficulty of achieving high order continuity across the elements, standard isoparametric FEM discretizations [42] require a system of equations with approximately twice the number of DOFs than those of IGA, being therefore potentially less efficient. A comparison between the approximation errors obtained with IGA and these FEM discretizations is shown in Figure 4.7, for which both the errors vs. the mesh size and the number of DOFs involved in the IGA and FEM approximations are reported. As we notice from Figure 4.7, the IGA approximation requires a smaller number of DOFs than its FEM counterpart of the same polynomial degree to achieve the same accuracy.

4.1.3 Test 4.2. Laplace–Beltrami triharmonic problem

We consider the numerical approximation of the triharmonic problem described in Section 1.2.2 on different geometries. In particular, we consider the approximation of problem (1.2.12) on the quarter of cylinder (Test 4.2.1) and the cylinder (Test 4.2.2), and of problem (1.2.13) on the unit sphere (Test 4.2.3). These geometries are the same considered for the biharmonic problem of Section 4.1.1, but, this time, represented by NURBS basis of degree $p \geq 3$ and at least globally C^2-continuous a.e. in the parametric domain $\widehat{\Omega}$ (for the cases of closed surfaces, the periodic NURBS function spaces described in Section 3.3.1 are employed). Then, the same function spaces are used for the discretization of the problems. Regarding Test 4.2.1, defined on the quarter of cylinder, we consider the following exact solution in cylindrical coordinates (θ, z) (Figure 4.1b):

$$u(\theta, z) = \sin^3(2\theta)\,\sin^3(\pi z), \tag{4.1.9}$$

where $\theta = \operatorname{atan}(y/z)$, with $\mu = 1$, $\gamma = 0$, and f suitably chosen. For Test 4.2.2, defined on the cylinder, we consider the following exact solution instead (Figure 4.3b):

$$u(\theta, z) = \sin\left(\sqrt{2} + 2\theta\right)\,\sin^3\left(\frac{3}{2}\pi z\right). \tag{4.1.10}$$

For problem (1.2.13) defined on the sphere (Test 4.2.3), we consider a right-hand-side function f such that the exact solution u is the one reported in Eq. (4.1.3) and shown in Figure 4.2a, using the same parameters.

By applying Theorem 7 to problem (1.2.12), an elliptic PDE of order 6 endowed with homogeneous essential boundary conditions for which $u \in H^r(\Omega)$ with $r \geq p+1 \geq m$, the following estimates hold:

$$\begin{aligned} \|u - u_h\|_{H^3(\Omega)} &\leq C_{shape} h^{p-2} \|u\|_{H^r(\Omega)} \\ \|u - u_h\|_{L^2(\Omega)} &\leq C_{shape} h^{\min\{p+1, 2p-4\}} \|u\|_{H^r(\Omega)}. \end{aligned} \tag{4.1.11}$$

Similarly to the case of the biharmonic problem, we do not compute the norm $H^3(\Omega)$ by using the seminorm $H^3(\Omega)$, but rather we use the norm $L^2(\Omega)$ of the third order Laplace–Beltrami operator. Indeed, for closed surfaces and problems with essential boundary conditions, the former seminorm and norm are equivalent, i.e.:

$$|\varphi|_{H^3(\Omega)} \simeq \|\nabla_\Omega \Delta_\Omega \varphi\|_{L^2(\Omega)}, \tag{4.1.12}$$

for $\varphi \in H^3(\Omega)$ with Ω closed or $\varphi \in H^3_0(\Omega)$. Figure 4.8 shows the errors in norms $H^3(\Omega)$ and $L^2(\Omega)$ obtained by the IGA approximation of the triharmonic problem on the quarter of cylinder (Test 4.2.1) under h-refinement, with C^2-continuous NURBS basis functions of degrees $p = 3$ and 4. We observe that the convergence rates obtained are in agreement with the error estimate (4.1.11); indeed, the rates are 1 and 2 for the errors in norm $H^3(\Omega)$ using basis of degrees $p = 3$ and 4, respectively, and 2 and 4 for the errors in norm $L^2(\Omega)$ for $p = 3$ and 4, respectively. Similarly, as shown in Figure 4.9, the numerical approximation the triharmonic problem on the cylinder (Test 4.2.2) yields the same convergence rates as in Test 4.2.1 on the quarter of cylinder, i.e. in agreement with the estimate of Eq. (4.1.11).

Figure 4.10 shows the errors obtained by the numerical approximation of the triharmonic problem on the sphere (Test 4.2.3). As for the biharmonic problem on the sphere, the convergence rates for the errors in norm $H^3(\Omega)$ are in agreement with Eq. (4.1.11), while the errors in norm $L^2(\Omega)$ converge at a higher rate, precisely 4 and 5 for basis functions of degrees $p = 3$ and 4, respectively. We explain this result in the track of Remark 1 for which we expect the optimal convergence rate $p+1$ for problems on closed surfaces, as indicated in Eq. (4.1.6).

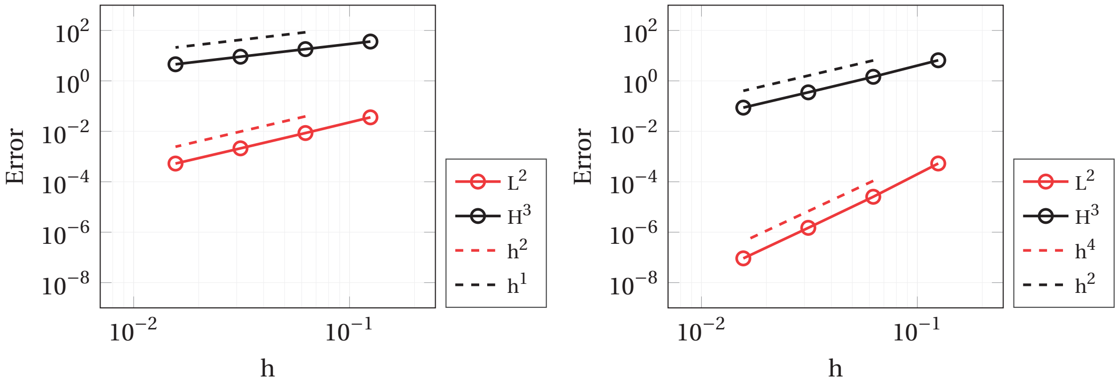


Figure 4.8 – Test 4.2.1. Triharmonic problem on the quarter of cylinder. Errors in norms $H^3(\Omega)$ and $L^2(\Omega)$ vs. mesh size h for NURBS bases of degree $p = 3$ and 4 and globally C^2-continuous in both cases (logarithmic scales are used on both the axes).

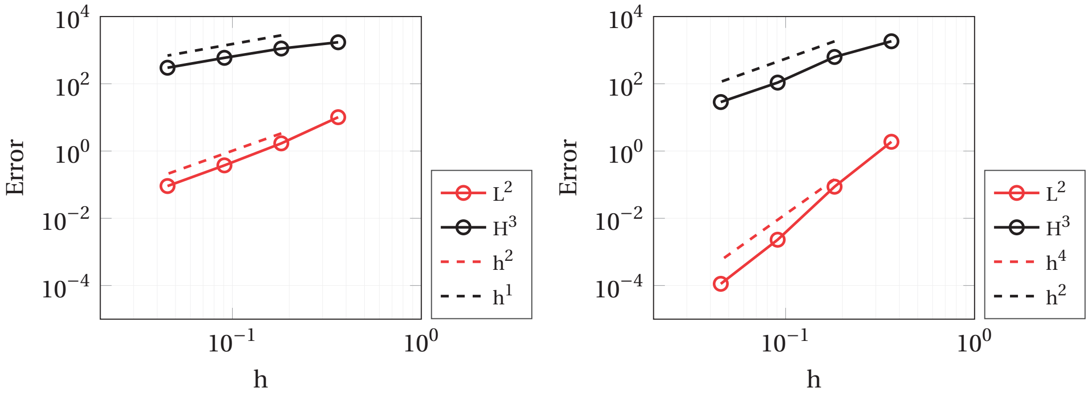


Figure 4.9 – Test 4.2.2. Triharmonic problem on the cylinder. Errors in norms $H^3(\Omega)$ and $L^2(\Omega)$ vs. mesh size h for NURBS bases of degree $p = 3$ and 4 and globally C^2-continuous in both cases (logarithmic scales are used on both the axes).

4.1.4 Test 4.2. Comparison with isoparametric FEM

In addition, we consider the approximation of the triharmonic problem (1.2.13) defined on the sphere Ω by means of a standard isoparametric FEM approximation. In this respect, we consider the problem in mixed formulation:

$$\text{find } u, v, w : \Omega \to \mathbb{R} \text{ such that}$$

$$\begin{cases} -\Delta_\Omega u + w = 0 & \text{in } \Omega, \\ -\Delta_\Omega w + v = 0 & \text{in } \Omega, \\ -\mu\Delta_\Omega v + \gamma u = f & \text{in } \Omega, \end{cases} \tag{4.1.13}$$

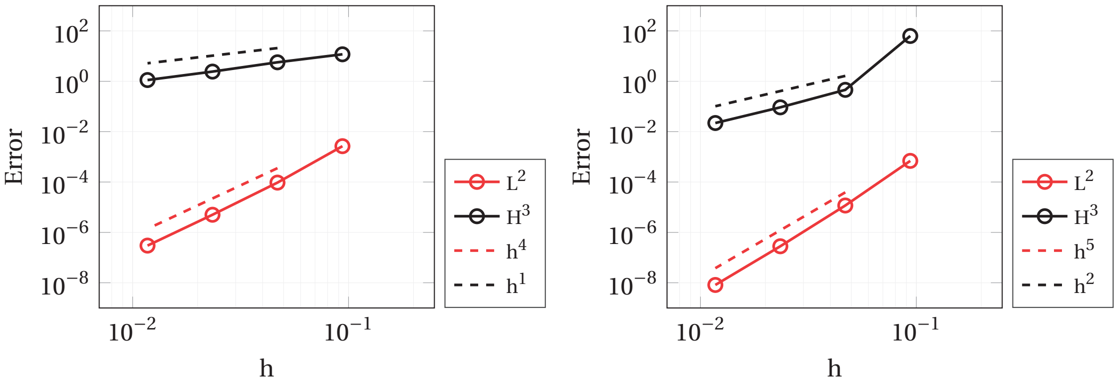


Figure 4.10 – Test 4.2.3. Triharmonic problem on the sphere. Errors in norms $H^3(\Omega)$ and $L^2(\Omega)$ vs. mesh size h for NURBS bases of degrees $p = 3$ and 4 and C^2-continuous a.e. on Ω in both cases (logarithmic scales are used on both the axes).

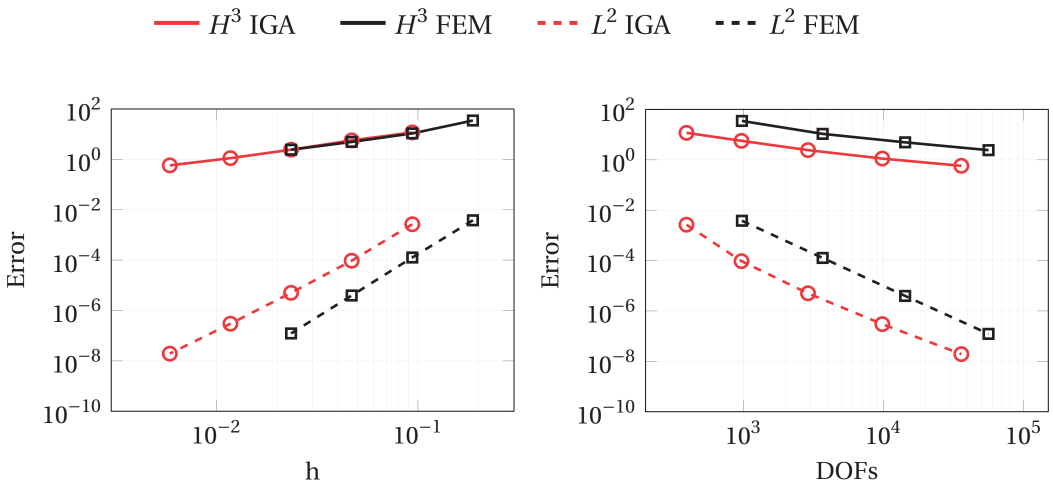
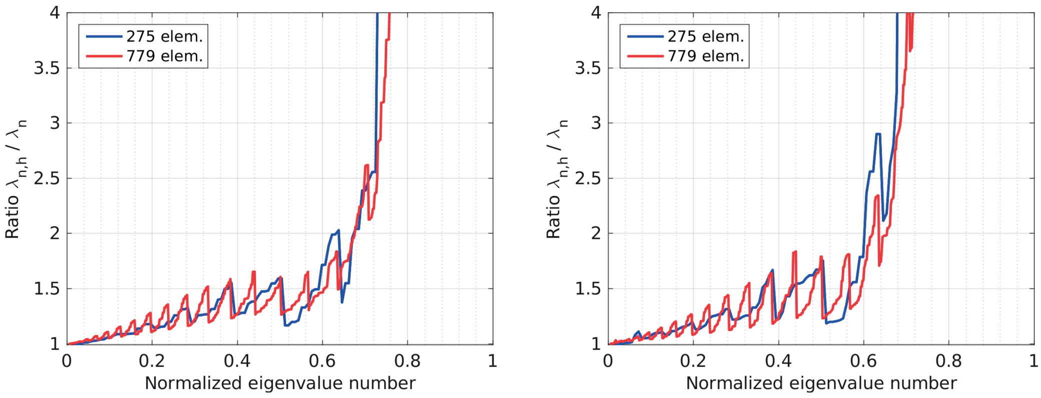


Figure 4.11 – Test 4.2.3. Triharmonic problem on the sphere. Errors in norm $H^3(\Omega)$ and $L^2(\Omega)$ vs. mesh size h (left) and number of DOFs (right), obtained with IGA and FEM for degree $p = 3$; for the IGA approximation NURBS basis functions C^2-continuous a.e. on Ω are used (logarithmic scales are used on both the axes).

where v and $w : \Omega \to \mathbb{R}$, are auxiliary unknowns. Problem (4.1.13) in weak form reads:

find $u, v, w \in V$ such that

$$\begin{cases} \int_\Omega \nabla_\Omega u \cdot \nabla_\Omega \varphi \, d\Omega + \int_\Omega w \varphi \, d\Omega = 0 & \forall \varphi \in V, \\ \int_\Omega \nabla_\Omega w \cdot \nabla_\Omega \psi \, d\Omega + \int_\Omega v \psi \, d\Omega = 0 & \forall \psi \in V, \\ \mu \int_\Omega \nabla_\Omega v \cdot \nabla_\Omega \vartheta \, d\Omega + \gamma \int_\Omega u \vartheta \, d\Omega = \int_\Omega f \vartheta \, d\Omega & \forall \vartheta \in V, \end{cases} \tag{4.1.14}$$



(a) Bilaplacian, $p = 2$, C^1-continuous a.e.

(b) Trilaplacian, $p = 3$, C^2-continuous a.e.

Figure 4.12 – Test 4.3. Laplace–Beltrami eigenvalue problems on the sphere. Normalized spectra (ratio $\lambda_{n,h}/\lambda_n$ vs. n/n_{bf}) computed by solving the bilaplacian and trilaplacian eigenvalue problems, with NURBS bases of degree $p = 2$ and C^1-continuous a.e. in Ω (left) and degree $p = 3$ and C^2-continuous a.e. in Ω (right), respectively.

with $V \equiv H^1(\Omega)$. Problem (4.1.14) is discretized using isoparametric FEM of degree $p = 3$ on successively finer meshes on the unit sphere. In this case, the FEM discretizations yield systems of equations with approximately three times the amount of DOFs with respect to the corresponding IGA discretizations. The errors obtained with IGA and FEM are reported in Figure 4.11 vs. the mesh size and the number of DOFs. As we can observe, IGA yields the same level of error with a much smaller number of DOFs, for which the same accuracy can be obtained more efficiently.

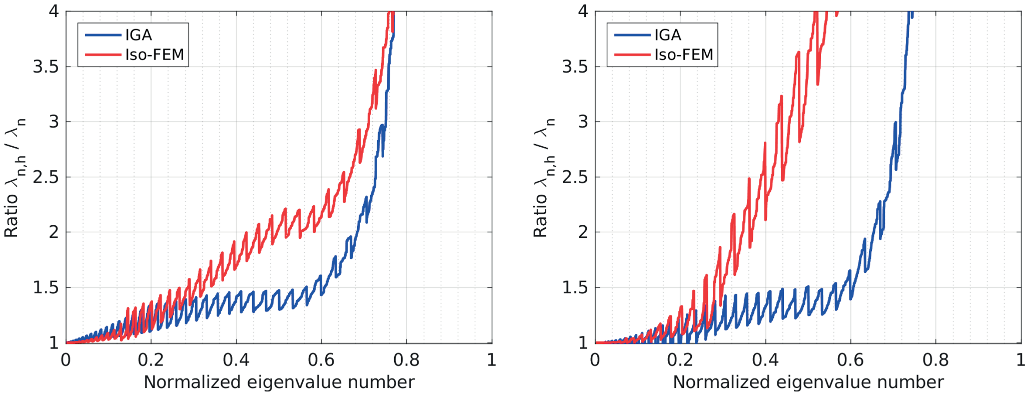
4.1.5 Test 4.3. High order Laplace–Beltrami eigenvalue problems

We consider the numerical approximation of the eigenvalue problem (1.2.20) associated to the Laplace–Beltrami operators of the fourth ($m = 2$) and sixth ($m = 3$) orders on the unit sphere. The exact eigenvalues, solution of the eigenvalue problem governed by an operator of order $2m$, with $m \geq 1$, are:

$$\lambda_n = (n(n+1))^m \qquad \text{for } n = 0, 1, \ldots, \tag{4.1.15}$$

where each eigenvalue λ_n has multiplicity $2n + 1$ (see e.g. [33]).

For the numerical approximation of the eigenvalue problem for $m = 2$ we employ NURBS bases of degree $p = 2$ and C^1-continuous a.e. in Ω, while for $m = 3$ we use NURBS bases of degree $p = 3$ and C^2-continuous a.e. in Ω. In Figure 4.12 we report the normalized spectra, i.e. the ratio $\lambda_{n,h}/\lambda_n$ vs. the normalized eigenvalue number n/n_{bf}, considering two different mesh sizes h for both the bilaplacian and trilaplacian eigenvalue problems.



(a) Bilaplacian, $p = 2$

(b) Trilaplacian, $p = 3$

Figure 4.13 – Test 4.3. Laplace–Beltrami eigenvalue problems on the sphere. Comparison of the normalized spectra computed by solving the bilaplacian (left) and trilaplacian (right) eigenvalue problems with IGA and isoparametric FEM for degrees $p = 2$ and 3, respectively; the IGA approximation uses NURBS basis functions C^1- and C^2-continuous a.e. on Ω, respectively.

Then, we compare the normalized spectra obtained with IGA against the ones obtained using isoparametric FEM discretizations. The bilaplacian and trilaplacian eigenvalue problems are approximated using FEM in mixed formulation, similarly to Eqs. (4.1.7) and (4.1.13), respectively. For the comparison of the bilaplacian problem we use for IGA NURBS of degree $p = 2$ and C^1-continuous a.e. in Ω with 2,048 DOFs, while for FEM we consider Lagrangian basis functions of degree $p = 2$ and C^0-continuous for a total of 3,972 DOFs. Regarding the trilaplacian problem, we consider an IGA discretization with NURBS basis functions of degree $p = 3$ and C^2-continuous a.e. in Ω for a total of 2,048 DOFs against a FEM discretization with Lagrangian basis functions of degree $p = 3$ and C^0-continuous with 3,318 DOFs. In Figure 4.13 we report the normalized spectra obtained for the Laplace–Beltrami eigenvalue problems of orders 4 and 6 on the sphere approximated by IGA and isoparametric FEM. It is quite evident that the IGA approximation with high order continuous basis functions yields more accurate results than its FEM counterpart even with a lower number of DOFs involved. Still, we remark that the isoparametric FEM do not allow the exact representation of the sphere.

4.2 Phase field models

In this section we consider the numerical approximation of the Cahn–Hilliard (Test 4.4) and phase field crystal equations (Test 4.5), introduced in Sections 1.3.1 and 1.3.2, respectively. As these are time-dependent problems, we firstly describe our approach to the time discretization using the α-method. Then, we proceed with the description of the IGA spatial discretization and we report numerical results for the approximation of these phase field models on open and closed surfaces.

4.2.1 Time discretization by the generalized-α method

As far as the time discretization is concerned, we employ for both the phase field models under consideration the generalized-α method [44, 45, 143]. This is a fully implicit time integration scheme, allowing the control on the numerical dissipation by suitably tuning a parameter. Let us consider a partition of the interval $[0, T] \subset \mathbb{R}$ into $N \in \mathbb{N}$, $N > 0$, time steps, with time instances denoted as t_n, for $n = 0, \dots, N$, for which $t_0 = 0$ and $t_N = T$. We denote the size of time step n with $\Delta t_n = t_{n+1} - t_n$. Let us consider the approximated solution $\widehat{u}_h(t_n) = \widehat{u}_h^n$ at time t_n and discretized by means of NURBS-based IGA, such that:

$$\widehat{u}_h^n = \sum_{i=1}^{n_{bf}} \widehat{R}_i U_i(t_n) \qquad \text{and} \qquad \dot{\widehat{u}}_h^n = \sum_{i=1}^{n_{bf}} \widehat{R}_i \dot{U}_i(t_n), \tag{4.2.1}$$

the latter being the time derivative. We denote with $\mathbf{U}^n$ and $\dot{\mathbf{U}}^n$ the vectors of control variables associated with the solution and their time derivatives at time t_n, i.e. $\mathbf{U}^n = \{U_i(t_n)\}_{i=1}^{n_{bf}}$ and $\dot{\mathbf{U}}^n = \{\dot{U}_i(t_n)\}_{i=1}^{n_{bf}}$. The generalized-$\alpha$ method consists in solving, at each time instance t_n, with $n \geq 0$, the following system of equations:

$$\begin{cases} \widehat{\mathbf{Res}}(\dot{\mathbf{U}}^{n+\alpha_m}, \mathbf{U}^{n+\alpha_f}) = \mathbf{0} \\ \mathbf{U}^{n+\alpha_f} = \mathbf{U}^n + \alpha_f \left(\mathbf{U}^{n+1} - \mathbf{U}^n\right) \\ \dot{\mathbf{U}}^{n+\alpha_m} = \dot{\mathbf{U}}^n + \alpha_m \left(\dot{\mathbf{U}}^{n+1} - \dot{\mathbf{U}}^n\right) \\ \mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t_n \dot{\mathbf{U}}^n + \delta \Delta t_n \left(\dot{\mathbf{U}}^{n+1} - \dot{\mathbf{U}}^n\right), \end{cases} \tag{4.2.2}$$

where the residual vector $\widehat{\mathbf{Res}}(\cdot,\cdot)$ is associated with the weak residuals (which, in our case, will be defined later in Eqs. (4.2.15) and (4.2.21) for the problems under consideration); the vectors $\mathbf{U}^0$ and $\dot{\mathbf{U}}^0$ are associated with the initial solution $\widehat{u}_h^0$ and initial velocity $\dot{\widehat{u}}_h^0$, as:

$$\widehat{u}_h^0 = \sum_{i=1}^{n_{bf}} \widehat{R}_i U_i(t_0) \qquad \text{and} \qquad \dot{\widehat{u}}_h^0 = \sum_{i=1}^{n_{bf}} \widehat{R}_i \dot{U}_i(t_0). \tag{4.2.3}$$

The parameters α_m, α_f and δ are chosen such that:

$$\delta = \frac{1}{2} + \alpha_m - \alpha_f \qquad \text{and} \quad \alpha_m \geq \alpha_f \geq \frac{1}{2}, \tag{4.2.4}$$

leading to an unconditionally stable and second order accurate method when employed for linear problems [144]. Moreover, they can be tuned to control the numerical dissipation of the high frequencies. By denoting with $\rho_\infty \in [0,1]$ the limit of the spectral radius of the amplification matrix for $\Delta t \rightarrow \infty$, the parameters can be chosen in terms of ρ_∞ as:

$$\alpha_m = \frac{1}{2}\left(\frac{3-\rho_\infty}{1+\rho_\infty}\right), \qquad \alpha_f = \frac{1}{1+\rho_\infty}, \qquad \text{and} \quad \delta = \frac{1}{1+\rho_\infty}. \tag{4.2.5}$$

In order to solve the nonlinear system of equations (4.2.2) resulting from time and space discretizations, we employ the Newton method [50], as in [44]. We denote with the subscript k the current Newton iterate, starting from $k = 0$, and with k_{max} the maximum number of sub-iterations allowed. Then, we have the following predictor–multicorrector scheme at each discrete time instance t_n, with $n \geq 0$:

Predictor (initialization phase of Newton method):

$$\begin{aligned} \mathbf{U}_0^{n+1} &= \mathbf{U}^n \\ \dot{\mathbf{U}}_0^{n+1} &= \frac{\delta-1}{\delta}\dot{\mathbf{U}}^n. \end{aligned} \tag{4.2.6}$$

Corrector:

1. Interpolate the solution and time derivative vectors at the intermediate time instances $t_{n+\alpha_f}$ and $t_{n+\alpha_m}$, respectively:

$$\begin{aligned} \mathbf{U}_{k+1}^{n+\alpha_f} &= \mathbf{U}^n + \alpha_f\left(\mathbf{U}_k^{n+1} - \mathbf{U}^n\right), \\ \dot{\mathbf{U}}_{k+1}^{n+\alpha_m} &= \dot{\mathbf{U}}^n + \alpha_m\left(\dot{\mathbf{U}}_k^{n+1} - \dot{\mathbf{U}}^n\right). \end{aligned} \tag{4.2.7}$$

2. Calculate the problem-specific residual vector:

$$\mathbf{Q}_{k+1} := \mathbf{Res}\left(\dot{\mathbf{U}}_{k+1}^{n+\alpha_m}, \mathbf{U}_{k+1}^{n+\alpha_f}\right). \tag{4.2.8}$$

   Then, check if the relative norm of the residual is below a certain tolerance τ:

$$\frac{\|\mathbf{Q}_{k+1}\|}{\|\mathbf{Q}_0\|} \leq \tau. \tag{4.2.9}$$

   If this is the case, the scheme has converged; otherwise, continue with step 3.

3. Assemble the tangent stiffness matrix:

$$\mathbf{K}_{k+1} := \alpha_m \frac{\partial \mathbf{Res}\left(\dot{\mathbf{U}}_{k+1}^{n+\alpha_m}, \mathbf{U}_{k+1}^{n+\alpha_f}\right)}{\partial \dot{\mathbf{U}}^{n+\alpha_m}} + \alpha_f \delta \Delta t_n \frac{\partial \mathbf{Res}\left(\dot{\mathbf{U}}_{k+1}^{n+\alpha_m}, \mathbf{U}_{k+1}^{n+\alpha_f}\right)}{\partial \mathbf{U}^{n+\alpha_f}}. \tag{4.2.10}$$

4. Solve the following linear system:

$$\mathbf{K}_{k+1}\Delta\dot{\mathbf{U}}_{k+1}^{n+1} = -\mathbf{Q}_{k+1}. \tag{4.2.11}$$

5. Correct the time derivative vector with the solution of the linear system (4.2.11):

$$\dot{\mathbf{U}}_{k+1}^{n+1} = \dot{\mathbf{U}}_{k}^{n+1} + \Delta\dot{\mathbf{U}}_{k+1}^{n+1}, \tag{4.2.12}$$

   then update the solution vector:

$$\mathbf{U}_{k+1}^{n+1} = \mathbf{U}_{k}^{n+1} + \delta\Delta t_n \Delta\dot{\mathbf{U}}_{k+1}^{n+1}. \tag{4.2.13}$$

6. Set $k = k+1$. If $k < k_{max}$ continue from step 1.

For the numerical tests of the next sections, we set $\rho_\infty = 0.5$ and the solution of the linear system (4.2.11) of step 4 is performed with the GMRES method [145] with incomplete LU factorization as preconditioner.

4.2.2 Test 4.4. Approximation of the Cahn–Hilliard equation

We consider the numerical approximation of the Cahn–Hilliard equations with boundary conditions (1.3.6) on a quarter of cylinder (Test 4.4.1) and its counterpart for closed surfaces on a unit sphere (Test 4.4.2). Since the Cahn–Hilliard equation is a fourth order PDE, we require the NURBS basis functions used for the discretization of the problem to be at least globally C^1-continuous a.e., eventually by employing the periodic NURBS function space construction described in Section 3.3.1. Finally, problem (1.3.8) is rewritten in the parametric domain $\widehat{\Omega}$ and discretized using NURBS-based IGA, reading:

for a.e. $t \in (0,T)$, find $\widehat{u}_h \in L^2(0,T;\widehat{V}_h) \cap H^1\left(0,T;L^2(\widehat{\Omega})\right)$ such that

$$\begin{cases} \widehat{\mathrm{Res}}(\widehat{u}_h(t))(\widehat{\psi}_h) = 0 & \forall\psi_h \in \widehat{V}_h,\ t \in (0,T), \\ + \text{ possible B. C.} & \text{on } \partial\widehat{\Omega} \text{ (if } \partial\Omega \neq \emptyset), \\ \widehat{u}_h(0) = \widehat{u}_{h,0} & \text{in } \widehat{\Omega}, \end{cases} \tag{4.2.14}$$

where $\widehat{V}_h := H^2(\widehat{\Omega}) \cap \widehat{\mathcal{N}_h}$, with, eventually, $\widehat{\mathcal{N}_h} := \widehat{\mathcal{N}}_h^{per}$ for closed surfaces. $\widehat{\mathrm{Res}}$ is the weak residual defined as:

$$\widehat{\mathrm{Res}}(\widehat{u}_h(t))(\widehat{\psi}_h) := \int_{\widehat{\Omega}} \frac{\partial \widehat{u}_h}{\partial t}\,\widehat{\psi}_h\, d\widehat{\Omega} + \widehat{a}_{CH}(\widehat{u}_h(t))(\widehat{\psi}_h) \qquad \forall\widehat{\psi}_h \in \widehat{V}_h, \tag{4.2.15}$$

with $\widehat{a}_{CH}(\cdot)(\cdot)$ obtained by a pull-back operation on the form (1.3.9) and $\widehat{u}_{h,0}$ as the pull-back of the L^2-projection of u_0 onto $\mathcal{N}_h$. The weak residual, defined on Ω, reads:

$$\mathrm{Res}(u_h(t))(\psi_h) := \int_{\Omega} \frac{\partial u_h}{\partial t}\, v_h\, d\Omega + a_{CH}(u_h(t))(\psi_h) \qquad \forall\psi_h \in V_h, \tag{4.2.16}$$

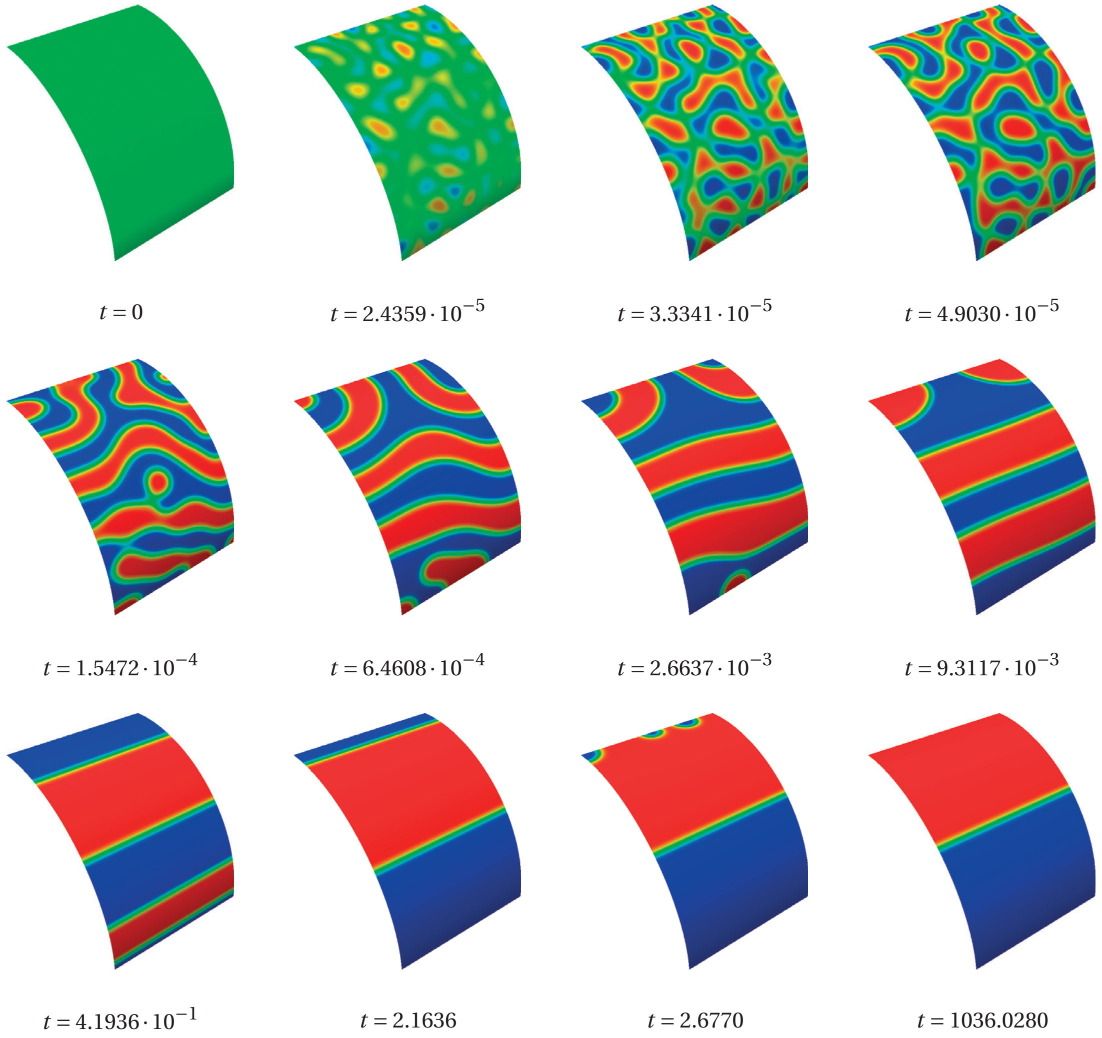
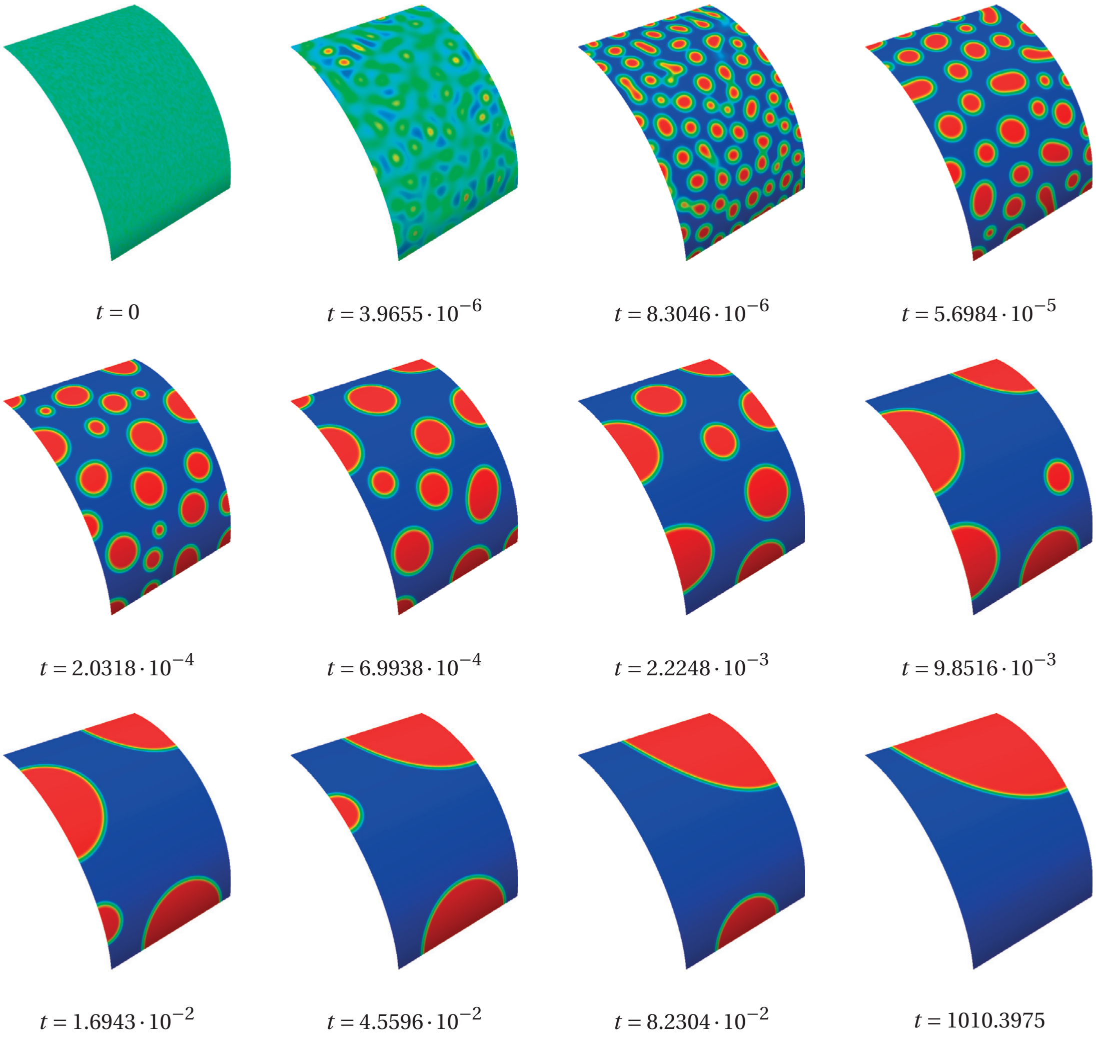

$t = 0$
$t = 2.4359 \cdot 10^{-5}$
$t = 3.3341 \cdot 10^{-5}$
$t = 4.9030 \cdot 10^{-5}$
$t = 1.5472 \cdot 10^{-4}$
$t = 6.4608 \cdot 10^{-4}$
$t = 2.6637 \cdot 10^{-3}$
$t = 9.3117 \cdot 10^{-3}$
$t = 4.1936 \cdot 10^{-1}$
$t = 2.1636$
$t = 2.6770$
$t = 1036.0280$

Figure 4.14 – Test 4.4.1a. Cahn–Hilliard equation on a quarter of cylinder: evolution of the solution with volume fraction $v_f = 0.5$.

where $V_h := H^2(\Omega) \cap \mathcal{N}_h$, with, eventually, $\mathcal{N}_h := \mathcal{N}_h^{per}$.

Solutions of the Cahn–Hilliard equation are characterized by extremely variable time scales. In order to obtain simulations reaching a steady state by considering all the time scales to a sufficiently long final time T, it is necessary to employ an adaptive time stepping procedure. We resort to the scheme proposed in [44, 45, 146]. The adaptive scheme relies on comparing the solutions obtained by solving the problem firstly with the generalized-α method and then with the Backward Euler method. Specifically, at each time step n the following iterative procedure is performed, indicating with i the iteration:

1. Set $i = 0$ and $\Delta t_{n,0} = \Delta t_{n-1}$.
2. Solve problem (4.2.14) at time step n with the generalized-α method using the time step


$t = 0$ $t = 3.9655 \cdot 10^{-6}$ $t = 8.3046 \cdot 10^{-6}$ $t = 5.6984 \cdot 10^{-5}$

$t = 2.0318 \cdot 10^{-4}$ $t = 6.9938 \cdot 10^{-4}$ $t = 2.2248 \cdot 10^{-3}$ $t = 9.8516 \cdot 10^{-3}$

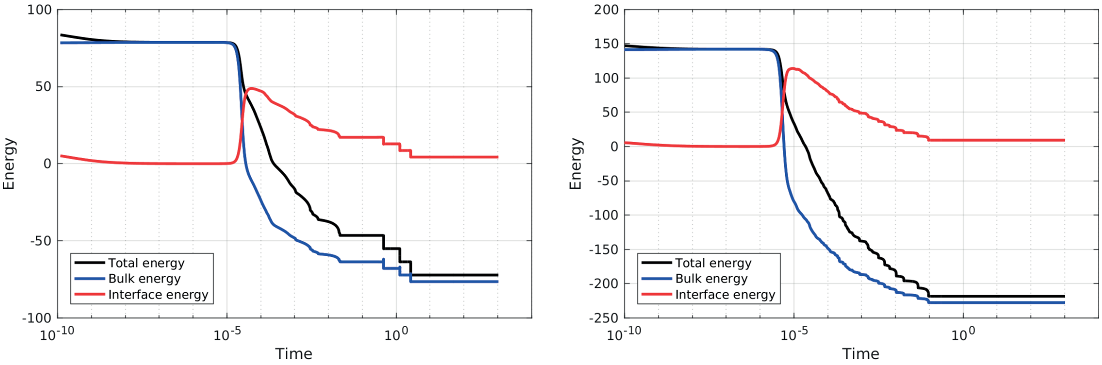
$t = 1.6943 \cdot 10^{-2}$ $t = 4.5596 \cdot 10^{-2}$ $t = 8.2304 \cdot 10^{-2}$ $t = 1010.3975$

Figure 4.15 – Test 4.4.1b. Cahn–Hilliard equation on a quarter of cylinder: evolution of the solution with volume fraction $v_f = 0.35$.

size $\Delta t_{n,i}$, obtaining the solution vector $\mathbf{U}_{\alpha,i}^{n+1}$.

3. Solve problem (4.2.14) at time step n with the Backward Euler method using the time step size $\Delta t_{n,i}$, obtaining the solution vector $\mathbf{U}_{BE,i}^{n+1}$.
4. Calculate the relative error between the solution vectors obtained using the two time discretization schemes:

$$e_{n+1,i} := \frac{\|\mathbf{U}_{\alpha,i}^{n+1} - \mathbf{U}_{BE,i}^{n+1}\|}{\|\mathbf{U}_{\alpha,i}^{n+1}\|}. \tag{4.2.17}$$



(a) $v_f = 0.5$ (Test 4.4.1a)

(b) $v_f = 0.35$ (Test 4.4.1b)

Figure 4.16 – Test 4.4.1. Cahn-Hilliard equation on the quarter of cylinder: energies $\mathcal{G}(t)$, $\mathcal{G}_c(t)$, and $\mathcal{G}_s(t)$ vs. time, with volume fractions $v_f = 0.5$ (Test 4.4.1a, left) and $v_f = 0.35$ (Test 4.4.1b, right).

5. Calculate the new time step size as:

$$\Delta t_{n,i+1} = \gamma_a \sqrt{\frac{\tau}{e_{n+1,i}}} \Delta t_{n,i}, \tag{4.2.18}$$

   where $\gamma_a \in \mathbb{R}^+$ is a safety parameter.
6. If $e_{n+1,i} < \tau$, with $\tau \in \mathbb{R}^+$ being a tolerance, or the maximum number of iterations i_{max} is reached, stop the iterations; otherwise, continue from step 2.

For the results presented in this thesis, we set $\tau = 10^{-3}$ and $\gamma_a = 0.85$.

We consider the numerical solution of the Canh–Hilliard equation with initial data u_0 representing a random mixture of the phases. In particular, the *volume fraction* is a measure of the quantity of a phase with respect to the other one, reading:

$$v_f := \frac{\int_\Omega u_0 \, d\Omega}{|\Omega|}; \tag{4.2.19}$$

this ratio is conserved throughout the evolution in time of the mixture.

Regarding the approximation of the Canh–Hilliard equation on the quarter of cylinder, we consider the case with $v_f = 0.5$ (Test 4.4.1a) and the case with $v_f = 0.35$ (Test 4.4.1b), with homogeneous natural boundary conditions and considering $M_0 = 1$, $\lambda = 1.3144 \cdot 10^{-3}$, $L_0 = 1$, and initial time step size $\Delta t_0 = 10^{-12}$. The IGA discretization is performed with NURBS basis functions of degree $p = 2$, globally C^1-continuous, for a total of 16,384 elements, yielding 16,384 DOFs. The evolution of the phases for Tests 4.4.1a and 4.4.1b are outlined in Figures 4.14 and 4.15, respectively, where we highlight the phase transition from the initial mixed condition to the steady state. For both the cases, the evolution of the total free energy $\mathcal{G}(t)$, as well as the chemical (bulk) $\mathcal{G}_c(t)$ and the surface $\mathcal{G}_s(t)$ energies, are reported in Figure 4.16. In Test 4.4.1a,

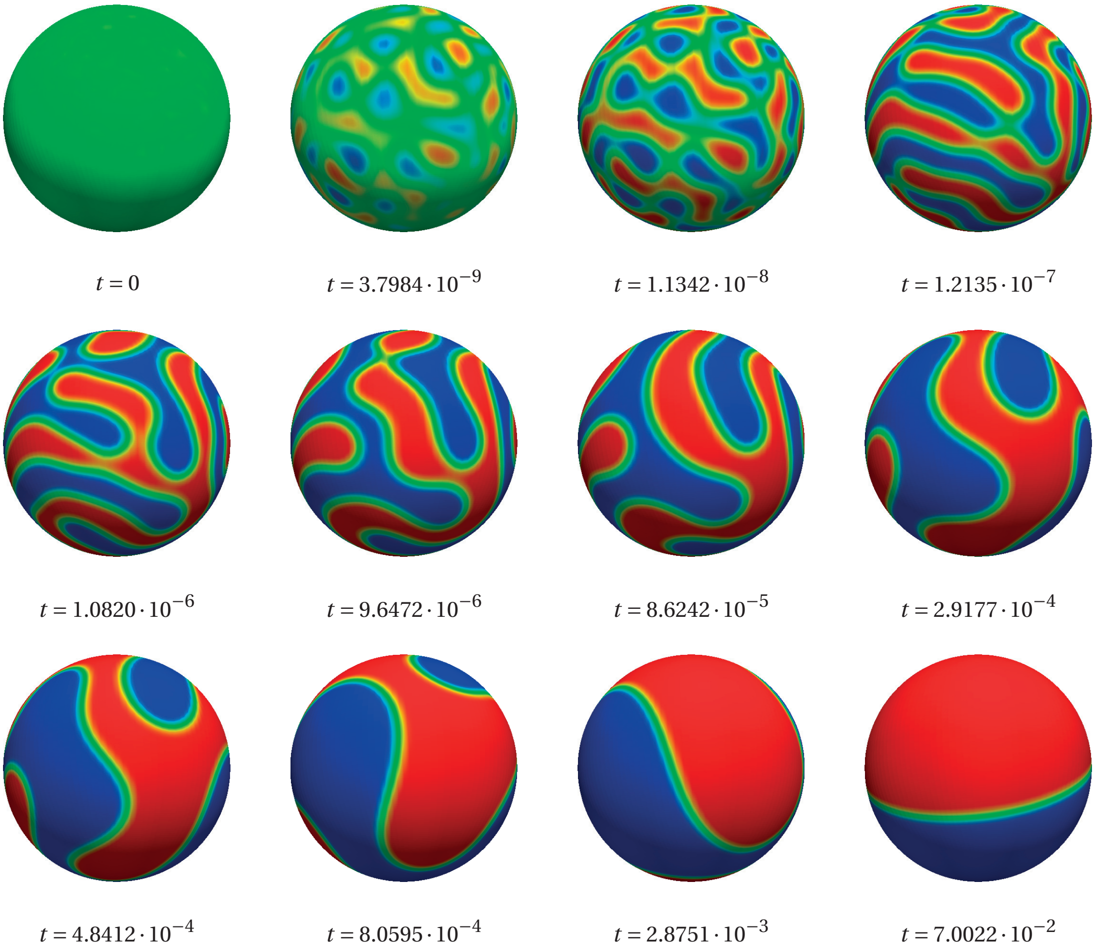
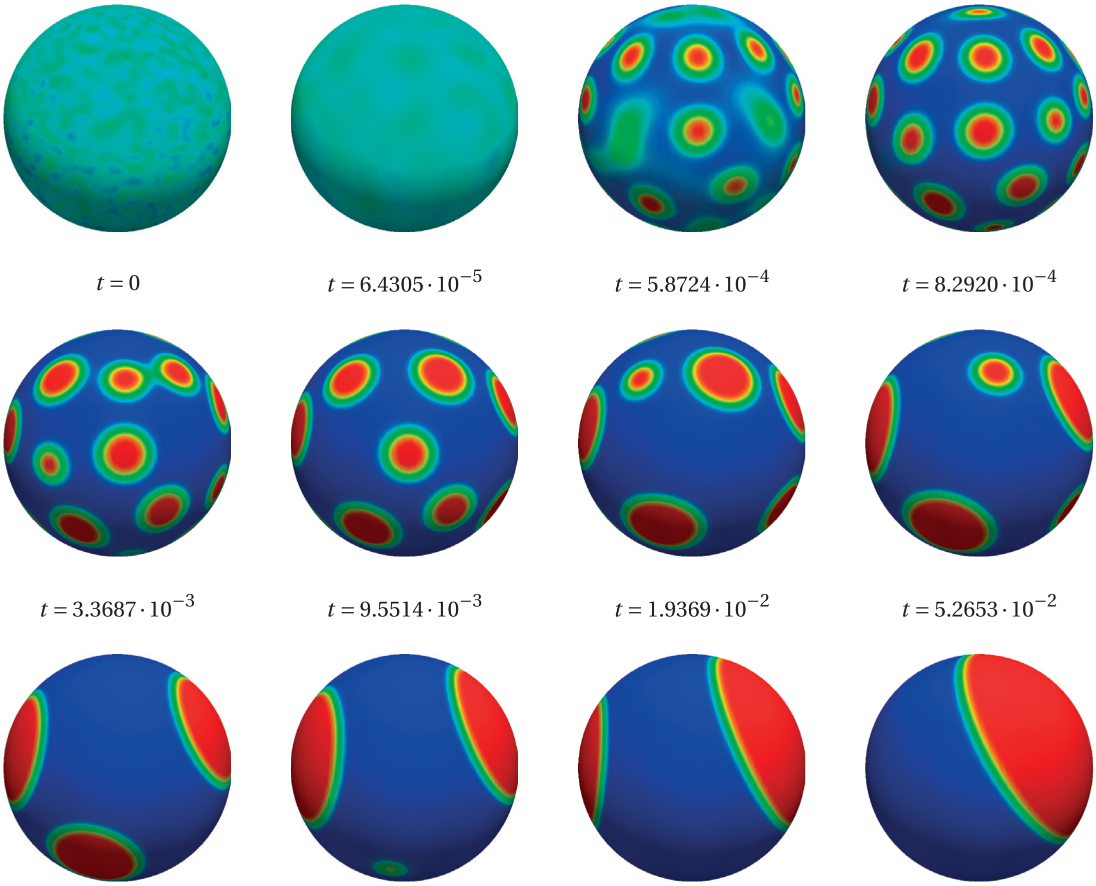

$t = 0$
$t = 3.7984 \cdot 10^{-9}$
$t = 1.1342 \cdot 10^{-8}$
$t = 1.2135 \cdot 10^{-7}$
$t = 1.0820 \cdot 10^{-6}$
$t = 9.6472 \cdot 10^{-6}$
$t = 8.6242 \cdot 10^{-5}$
$t = 2.9177 \cdot 10^{-4}$
$t = 4.8412 \cdot 10^{-4}$
$t = 8.0595 \cdot 10^{-4}$
$t = 2.8751 \cdot 10^{-3}$
$t = 7.0022 \cdot 10^{-2}$

Figure 4.17 – Test 4.4.2a. Cahn–Hilliard equation on the sphere: evolution of the solution with volume fraction $v_f = 0.5$.

at the beginning the majority of the energy is represented by the chemical one and the phases undergo an initial separation. Then, the energy becomes mostly interfacial and, as it gets minimized, it leads to the coarsening of the phases, until they reach the equilibrium status for which the interface is minimal. Instead, in Test 4.4.1b the prevalence of one phase in the initial mixture, tuned through the volume fraction v_f, leads to the scarcer phase to confine itself on a corner of the geometry and the resulting minimal interface is circular.

Regarding the approximation on the sphere, we consider the case with $v_f = 0.5$ (Test 4.4.2a) and the case with $v_f = 0.3$ (Test 4.4.2b). In both the cases, we set $M_0 = 1$, $\lambda = 1.3144 \cdot 10^{-3}$, $L_0 = 1$ and we consider a spatial IGA discretization based on NURBS bases of degree $p = 2$ and C^1-continuous a.e. on Ω; the mesh is comprised of 8,844 elements, for a total number of 8,192 DOFs. We initially set $\Delta t_0 = 10^{-14}$. The results are reported in Figure 4.17 (for Test 4.4.2a) and Figure 4.18 (for Test 4.4.2b), for several time instances. The evolutions of the total, the chemical, and the interface energies are reported in Figure 4.19. Finally, we report in Figure 4.20 the



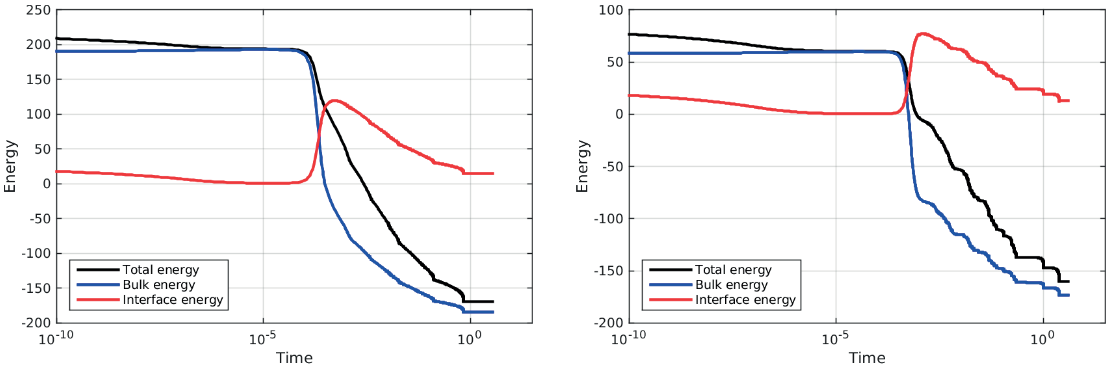
$t = 0$ $t = 6.4305 \cdot 10^{-5}$ $t = 5.8724 \cdot 10^{-4}$ $t = 8.2920 \cdot 10^{-4}$

$t = 3.3687 \cdot 10^{-3}$ $t = 9.5514 \cdot 10^{-3}$ $t = 1.9369 \cdot 10^{-2}$ $t = 5.2653 \cdot 10^{-2}$

$t = 1.0873 \cdot 10^{-1}$ $t = 2.1818 \cdot 10^{-1}$ $t = 1.9039$ $t = 4.0040$

Figure 4.18 – Test 4.4.2b. Cahn–Hilliard equation on the sphere: evolution of the solution with volume fraction $v_f = 0.3$.

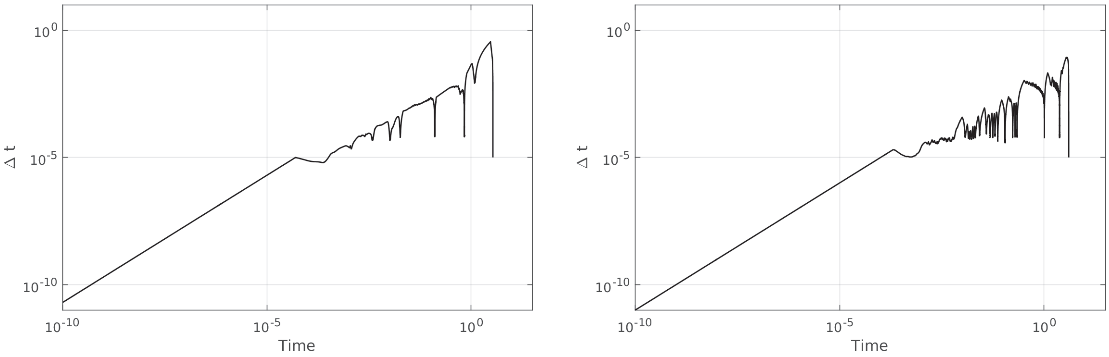
evolution in time of the time step size Δt for Tests 4.4.2a and 4.4.2b, to show how the adaptive scheme adapt the time step size to reach the equilibrium states in less computational time.



(a) $v_f = 0.5$ (Test 4.4.2a)

(b) $v_f = 0.3$ (Test 4.4.2b)

Figure 4.19 – Test 4.4.2. Cahn-Hilliard equation on the sphere: energies $\mathcal{G}(t)$, $\mathcal{G}_c(t)$, and $\mathcal{G}_s(t)$ vs. time, with volume fractions $v_f = 0.5$ (Test 4.4.2a, left) and $v_f = 0.3$ (Test 4.4.2b, right).



(a) $v_f = 0.5$ (Test 4.4.2a)

(b) $v_f = 0.3$ (Test 4.4.2b)

Figure 4.20 – Test 4.4.2. Cahn-Hilliard equation on the sphere: time step size Δt vs. time, for volume fractions $v_f = 0.5$ (Test 4.4.2a, left) and $v_f = 0.3$ (Test 4.4.2b, right).

4.2.3 Test 4.5. Approximation of the phase field crystal equation

We consider the approximation of the phase field crystal equation, described in Section 1.3.2, on an open and a closed surface. In particular, we consider problem (1.3.13) with natural boundary conditions on the quarter of cylinder and problem (1.3.14) on a torus. The geometry is always represented exactly by NURBS with a single patch. Problems (1.3.13) and (1.3.14) are characterized both by a sixth order PDE; therefore, we consider the function space $V \subseteq H^3(\Omega)$, with function at least globally C^2-continuous. We rewrite problem (1.3.15) in the parametric domain $\widehat{\Omega}$ and spatially discretize it with NURBS-based IGA, obtaining the following semi-discretized problem:

$$\text{find } \widehat{u}_h(t) \in L^2(0,T;\widehat{V}_h) \cap H^1(0,T;L^2(\widehat{\Omega})) \text{ such that}$$

$$\begin{cases} \widehat{\text{Res}}(\widehat{u}_h(t))(\widehat{\psi}_h) = 0 & \forall \widehat{\psi}_h \in \widehat{V}_h,\ t \in (0,T), \\ \widehat{u}_h(0) = \widehat{u}_{h,0} & \text{in } \widehat{\Omega} \times \{0\}, \end{cases} \tag{4.2.20}$$

where $\widehat{V}_h := H^3(\widehat{\Omega}) \cap \widehat{\mathscr{N}}_h^{per}$ and $\widehat{\text{Res}}$ is obtained performing the pull-back operation on the weak residual:

$$\text{Res}(u_h(t))(\psi_h) := \int_\Omega \frac{\partial u_h}{\partial t} \psi_h \, d\Omega + a_{PFC}(u_h(t))(\psi_h), \tag{4.2.21}$$

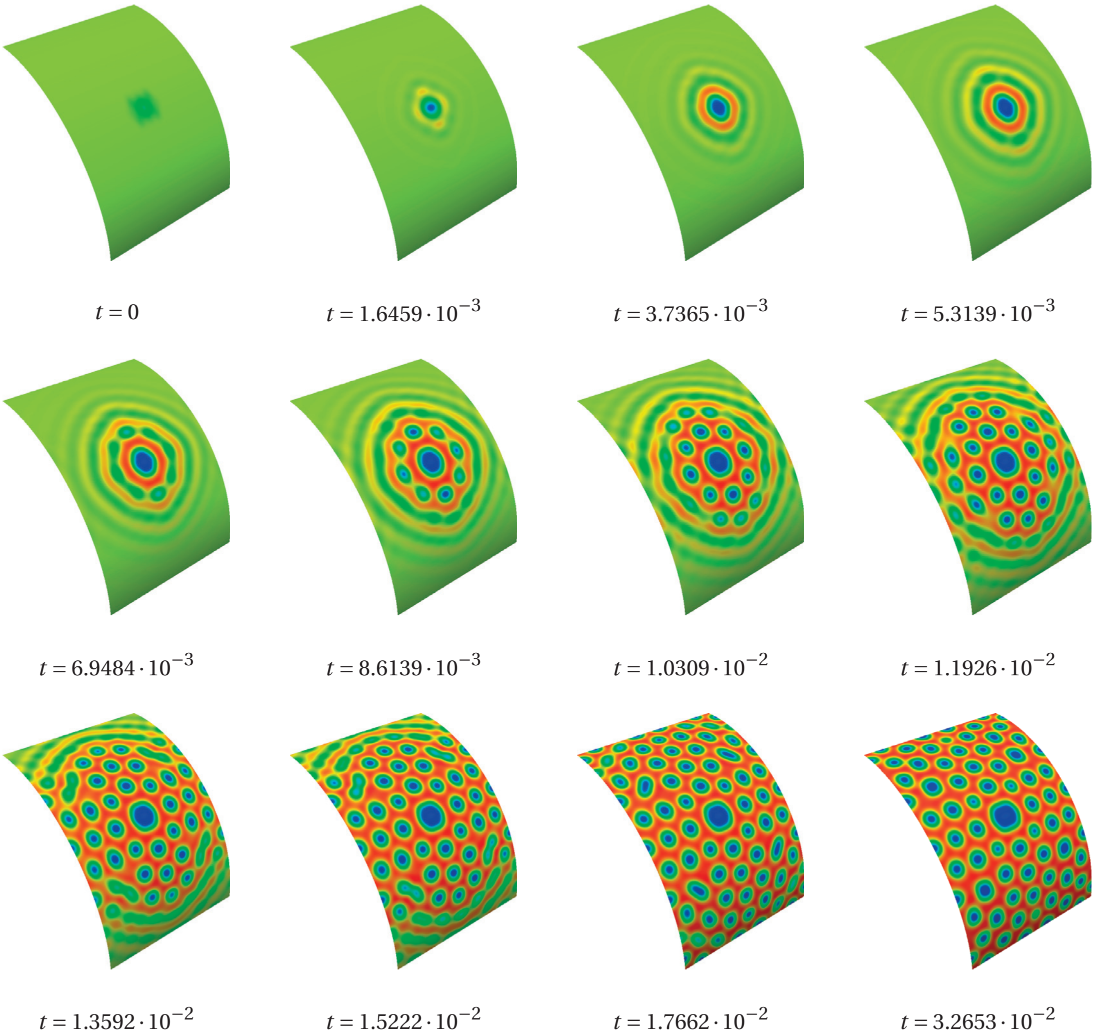
with $a_{PFC}(\cdot)(\cdot)$ being the form associated with the crystal equation and reported in Eq. (1.3.16).

Unlike the Cahn–Hilliard equation, the phase field crystal equation does not involve large variations of the time scales; nevertheless, an ad hoc empirical time step size adaptivity scheme has been employed, in order to reduce the overall computational cost of the simulation: at each time step the successive time step size is calculated as the actual one rescaled by a factor depending on the number of Newton sub-iterations N_{newton} carried at the corrector stage of the generalized-α method. Specifically, $\Delta t_{n+1} = \min\{\beta_n \Delta t_n, \Delta t_{max}\}$, where:

$$\beta_n = \begin{cases} 1.2 & \text{if } N_{newton} < 3, \\ 1.1 & \text{if } N_{newton} = 3, \\ 0.8 & \text{if } N_{newton} = 4, \\ 0.5 & \text{if } N_{newton} > 4. \end{cases} \tag{4.2.22}$$

This is intended to keep the number of Newton iterations between 3 and 4 at each time step, which represents a compromise between the computational cost and the accuracy of the solution; moreover, with respect to the adaptivity scheme described in Section 4.2.2, it does not require the assembly and solution of two linear systems at each time step.

We consider an initial condition u_0 representing a single crystal immersed in a uniform liquid field. We choose the parameters $D = 10^6$, $k = 10^{-3/2}$, $L_0 = 1$, $\phi_0 = 5$, $g = 0$, and $\varepsilon = 1$, for which the dimensionless parameters are $\mathbb{N}_1 = 0.2$, $\mathbb{N}_2 = 4 \cdot 10^{-3}$, and $\mathbb{N}_3 = 2 \cdot 10^{-5}$, and the initial time step size $\Delta t_0 = 5 \cdot 10^{-5}$. We consider the spatial IGA approximation of the phase field crystal problem by using NURBS basis functions of degree $p = 3$ and C^2-continuous on Ω; since also the geometrical mapping of the torus does not have singularities, we are able to enforce the required degree of continuity globally. For Test 4.5.1 on the quarter of cylinder, the mesh is comprised of 1,024 elements, yielding 1,024 DOFs. Regarding Test 4.5.2 on the torus, the mesh is made of 36,305 elements, for a total of 32,768 DOFs. The results for Tests 4.5.1 and 4.5.2 are reported in Figures 4.21 and 4.23, respectively. The evolutions of the energy $\mathscr{E}(t)$ for both cases are reported in Figure 4.22 and 4.24, from which we observe that it is monotonically decreasing in time.



$t = 0$ $t = 1.6459 \cdot 10^{-3}$ $t = 3.7365 \cdot 10^{-3}$ $t = 5.3139 \cdot 10^{-3}$

$t = 6.9484 \cdot 10^{-3}$ $t = 8.6139 \cdot 10^{-3}$ $t = 1.0309 \cdot 10^{-2}$ $t = 1.1926 \cdot 10^{-2}$

$t = 1.3592 \cdot 10^{-2}$ $t = 1.5222 \cdot 10^{-2}$ $t = 1.7662 \cdot 10^{-2}$ $t = 3.2653 \cdot 10^{-2}$

Figure 4.21 – Test 4.5.1. Phase field crystal equation on the quarter of cylinder: evolution of the solution.

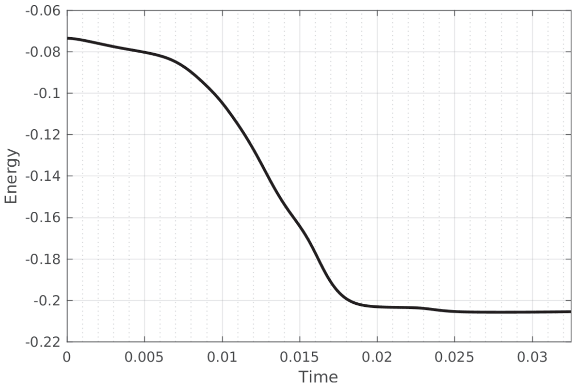
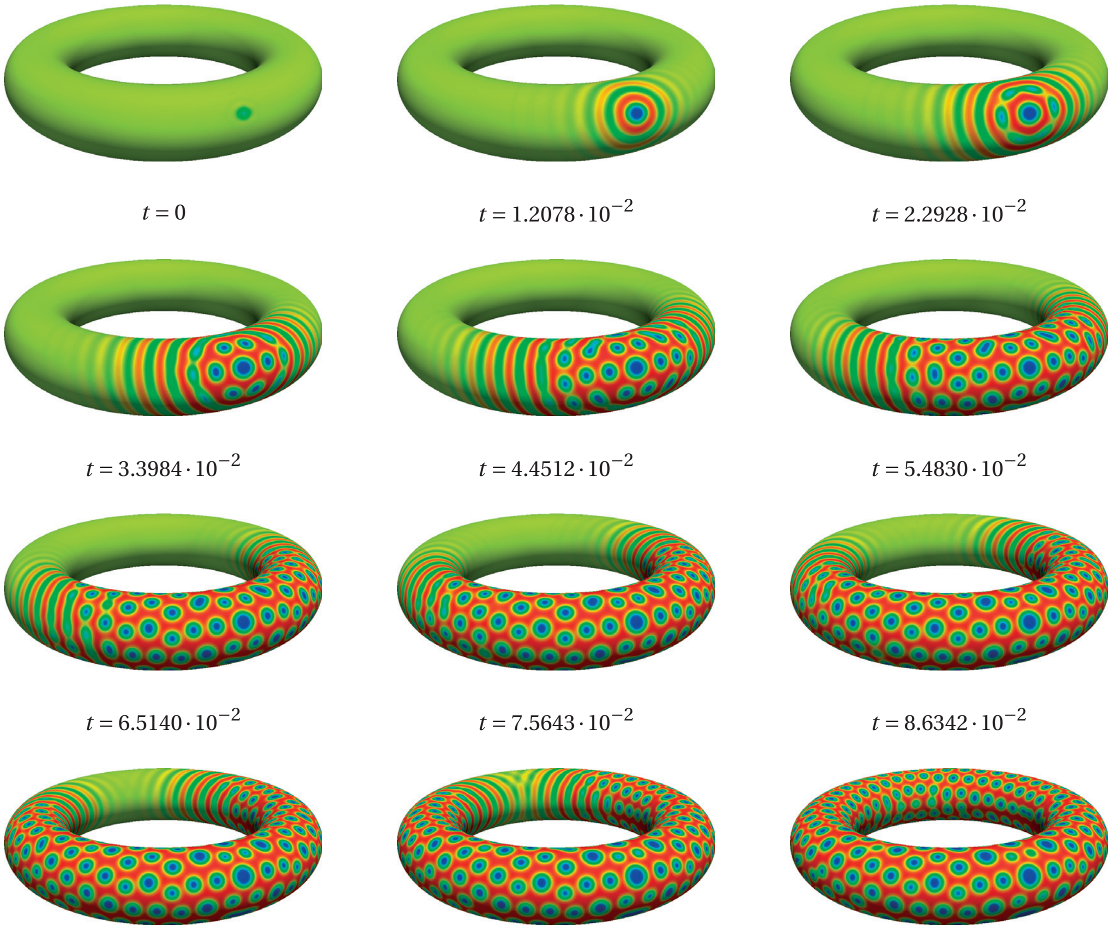


Figure 4.22 – Test 4.5.1. Phase field crystal equation on the quarter of cylinder: energy $\mathscr{C}(t)$ vs. time.



$t = 0$ $t = 1.2078 \cdot 10^{-2}$ $t = 2.2928 \cdot 10^{-2}$

$t = 3.3984 \cdot 10^{-2}$ $t = 4.4512 \cdot 10^{-2}$ $t = 5.4830 \cdot 10^{-2}$

$t = 6.5140 \cdot 10^{-2}$ $t = 7.5643 \cdot 10^{-2}$ $t = 8.6342 \cdot 10^{-2}$

$t = 9.7245 \cdot 10^{-2}$ $t = 1.0941 \cdot 10^{-1}$ $t = 1.2884 \cdot 10^{-1}$

Figure 4.23 – Test 4.5.2. Phase field crystal equation on the torus: evolution of the solution.

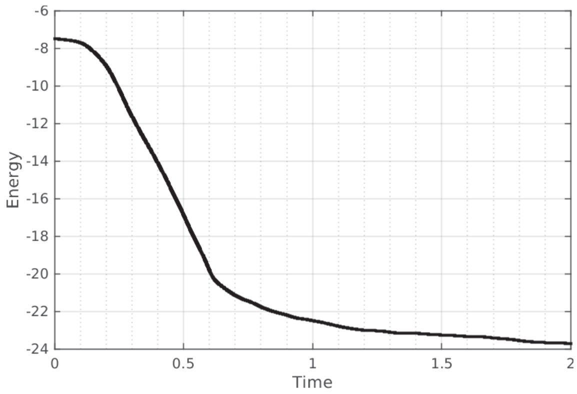


Figure 4.24 – Test 4.5.2. Phase field crystal equation on the torus: energy $\mathscr{E}(t)$ vs. time.

5 IGA for geometric PDEs

In this chapter, based on the paper [106], we consider the numerical approximation of the geometric PDEs introduced in Chapter 2, namely the mean curvature and the Willmore flow problems.

Being ubiquitous in many applications, the approximation of geometric PDEs has been extensively analyzed using different approaches in literature, which we briefly recall hereafter. Considering the mean curvature flow problem described in Section 2.4, several formulations based on the FEM have been considered by various authors (see e.g. [147, 148]), with the level set method (e.g. in [149,150]), or based on a phase field approach (see [10]). The approximation of the Willmore flow is also of particular interest as it can be considered as a simplified model for more complex curvature-based geometric flows. The seminal work [151] considers a general surface evolver, which has been applied to the Willmore energy using the FEM. Approximations based on finite difference schemes for axisymmetric solutions are proposed in [152]. In general, the numerical approximation of the Willmore flow on parametric surfaces is based on the FEM, as e.g. in [48,153,154]; in [155] a formulation based on the level set method is used, while in [138, 156, 157] approximations of the Willmore flow for curves — also called elastic flow of curves — are studied. For a general review on the numerical approximation of geometric PDEs we refer the interested reader to [10], while, for approximating PDEs on evolving surfaces, to the review work [158]. Nevertheless, all these approaches generally involve geometric approximations of quantities which may lead to accuracy issues or complex numerical schemes, which we instead aim at removing by using NURBS-based IGA.

In this chapter, we describe our approach to the numerical approximation of the mean curvature and Willmore flow problems. Firstly, we describe our approach to the discretization in space and time of a generic geometric PDE. As in Chapter 4, we discretize in space with NURBS-based IGA in Section 5.1. Time discretization is performed by employing high order Backward Differentiation Formulas (BDF) in a semi-implicit formulation and is introduced in Section 5.2. Applications of these techniques to the mean curvature and Willmore flow problems follow in Sections 5.3 and 5.4, respectively, with details regarding the formulation adopted for each of the two problems and numerical results on several common geometries.

5.1 Spatial discretization with IGA

Let us consider a generic shape energy functional $J(\Omega)$ and the associated geometric gradient flow problem of Eq. (2.3.5), in weak form, which we rewrite here for convenience: given $\Omega_0 \subset \mathbb{R}^d$,

$$\text{for a.e. } t \in (0,T),\ \text{find } \Omega_t \subset \mathbb{R}^d \text{ such that}$$
$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t = -\mu\, dJ(\Omega_t)(\boldsymbol{\varphi}) & \forall \boldsymbol{\varphi} \in V(\Omega_t), \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0. \end{cases} \tag{5.1.1}$$

The unknown surface Ω_t is defined through the identity map $\mathbf{x}$ at time t, as described in Section 2.1. This problem can be written in a (more) general way as follows:

$$\text{for a.e. } t \in (0,T),\ \text{find } \mathbf{x} \in V_{g,t} \text{ such that}$$
$$\begin{cases} m(\dot{\mathbf{x}};\boldsymbol{\varphi}) + a(\mathbf{x};\boldsymbol{\varphi}) = 0 & \forall \boldsymbol{\varphi} \in V_{0,t}, \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0, \end{cases} \tag{5.1.2}$$

where $a(\cdot;\cdot)$ is the form associated with the right-hand-side of Eq. (5.1.1) and defining the problem under consideration and $m(\cdot;\cdot)$ is the mass form:

$$m(\dot{\mathbf{x}};\boldsymbol{\varphi}) := \int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t. \tag{5.1.3}$$

The Hilbert spaces $V_{g,t}$ and $V_{0,t}$ depend on the order of the spatial differential operators involved in the form $a(\cdot;\cdot)$; in the case that Ω_t is open (that is with boundary), $V_{g,t}$ accounts for the non-homogeneous essential boundary conditions and $V_{0,t}$ is its homogeneous counterpart. If the problem is defined on closed geometries, then $\partial\Omega_t \equiv \emptyset$ for all $t \in (0,T)$ and the spaces $V_{0,t}$ and $V_{g,t}$ coincide and are identified with the Hilbert space indicated as V_t, resulting in the following problem:

$$\text{for a.e. } t \in (0,T),\ \text{find } \mathbf{x} \in V_t \text{ such that}$$
$$\begin{cases} m(\dot{\mathbf{x}};\boldsymbol{\varphi}) + a(\mathbf{x};\boldsymbol{\varphi}) = 0 & \forall \boldsymbol{\varphi} \in V_t, \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0. \end{cases} \tag{5.1.4}$$

The forms $a(\cdot;\cdot)$ and $m(\cdot;\cdot)$, the function spaces, and the geometric quantities depend on the current computational domain Ω_t, which itself depends on $\mathbf{x}(t)$. Therefore, problems (5.1.2) and (5.1.4) are highly non-linear.

Since we deal with parametric geometries defined by geometrical mappings of the form (1.1.2), which are invertible a.e., following Section 3.4, we *pull-back* problem (5.1.2) into the parametric domain $\widehat{\Omega}$ (problem (5.1.4) can be treated in the same way), thus obtaining:

$$\text{for a.e. } t \in (0,T), \text{ find } \mathbf{X} \in \widehat{V}_g \text{ such that}$$
$$\begin{cases} \widehat{m}(\dot{\mathbf{X}}, \widehat{\boldsymbol{\varphi}}) + \widehat{a}(\mathbf{X}; \widehat{\boldsymbol{\varphi}}) = 0 & \forall \widehat{\boldsymbol{\varphi}} \in \widehat{V}_0, \\ \mathbf{X}(0) = \mathbf{X}_0 & \text{in } \widehat{\Omega}, \end{cases} \tag{5.1.5}$$

where $\mathbf{X}$ is the time-dependent geometric mapping defining Ω_t, as introduced in Eq. (2.1.2), and $\widehat{V}_g$ and $\widehat{V}_0$ correspond to the pull-back of the function spaces $V_{g,t}$ and $V_{0,t}$ on the parametric domain $\widehat{\Omega}$, respectively; the forms $\widehat{m}(\cdot,\cdot)$ and $\widehat{a}(\cdot;\cdot)$ result from the pull-back operation applied to $m(\cdot;\cdot)$ and $a(\cdot;\cdot)$, respectively. In particular, we have:

$$\widehat{m}(\dot{\mathbf{X}}, \widehat{\boldsymbol{\varphi}}) = \int_{\widehat{\Omega}} \dot{\mathbf{X}} \cdot \widehat{\boldsymbol{\varphi}} \, \widehat{g} \, d\widehat{\Omega} \qquad \text{for } \widehat{\boldsymbol{\varphi}} \in \widehat{V}_0, \tag{5.1.6}$$

where $\widehat{g}$ is the determinant of the first fundamental form of the mapping, defined in Eq. (1.1.10). The form $a(\cdot;\cdot)$, which depends on the problem under consideration, is pulled-back in a similar fashion, using the geometric quantities introduced in Section 1.1.

Then we proceed with the spatial discretization of problem (5.1.5). We consider trial and test functions in the form of (3.2.7) choosing suitable NURBS function spaces $\mathcal{N}_h$ and $\widehat{\mathcal{N}_h}$ accordingly with (3.3.2) and (3.3.1). Indeed, for all $t \in (0,T)$, we seek solutions in the form:

$$\mathbf{x}_h(t) = \sum_{i=1}^{n_{bf}} \left(\widehat{R}_i \circ \mathbf{X}^{-1} \right) \mathbf{P}_i(t) \tag{5.1.7}$$

in the physical space; $\mathbf{P}_i(t)$, for $i = 1, \ldots, n_{bf}$, are the time-dependent control points introduced in Section 3.2 which describe the evolving surface and, in this context of geometric PDEs, also play the role of vector-valued control variables. The semi-discretized problem reads:

$$\text{for a.e. } t \in (0,T), \text{ find } \mathbf{x}_h \in V_{g,t,h} \text{ such that}$$
$$\begin{cases} m(\dot{\mathbf{x}}_h; \boldsymbol{\varphi}_h) + a(\mathbf{x}_h; \boldsymbol{\varphi}_h) = 0 & \forall \boldsymbol{\varphi}_h \in V_{0,t,h}, \\ \mathbf{x}_h(0) = \mathbf{x}_{0,h} & \text{in } \Omega_0, \end{cases} \tag{5.1.8}$$

which can be pulled-back into the parametric domain $\widehat{\Omega}$, thus obtaining:

$$\text{for a.e. } t \in (0,T), \text{ find } \mathbf{X}_h \in \widehat{V}_{g,h} \text{ such that}$$
$$\begin{cases} \widehat{m}(\dot{\mathbf{X}}_h, \widehat{\boldsymbol{\varphi}}_h) + \widehat{a}(\mathbf{X}_h; \widehat{\boldsymbol{\varphi}}_h) = 0 & \forall \widehat{\boldsymbol{\varphi}}_h \in \widehat{V}_{0,h}, \\ \mathbf{X}_h(0) = \mathbf{X}_{0,h} & \text{in } \Omega_0. \end{cases} \tag{5.1.9}$$

The finite-dimensional function spaces $V_{g,t,h}$, $V_{0,t,h}$, $\widehat{V}_{g,h}$ and $\widehat{V}_{0,h}$ are subsets of the NURBS function spaces defined as $V_{g,t,h} := V_{g,t} \cap [\mathcal{N}_h]^d$, $V_{0,t,h} := V_{0,t} \cap [\mathcal{N}_h]^d$, $\widehat{V}_{g,h} := \widehat{V}_g \cap [\widehat{\mathcal{N}_h}]^d$ and $\widehat{V}_{0,h} := \widehat{V}_0 \cap [\widehat{\mathcal{N}_h}]^d$, respectively. In this setting, the family of unknown geometries $\{\Omega_t\}_{t\in(0,T)}$ is described through the time-dependent mapping $\mathbf{X}_h$, as well as from the corresponding time-dependent identity map $\mathbf{x}_h$, following the notation introduced in Section 2.1.

As already outlined in Section 3.3.1, while simple geometries can straightforwardly be represented by using C^{p-1}-continuous NURBS basis functions in a single patch, this may be not the case of more complex geometries, as the closed ones, for which the single patch NURBS representation involves bases which are only globally C^0-continuous in Ω and $\widehat{\Omega}$ (e.g. a circle for $d = 2$ or a sphere for $d = 3$). Since we are interested in the approximation of geometric PDEs of order two or higher (as the Willmore flow), we need to use trial and test function spaces built from high order globally continuous basis functions over the whole geometry. Therefore, when dealing with closed geometries, we consider the construction of periodic NURBS function spaces $\mathcal{N}_h^{per}$ as defined in Eq. (3.3.11). However, since the control points $\{\mathbf{P}_i\}_{i=1}^{n_{bf}}$ describe the geometry but at the same time represent the unknown of the problem, we cannot use a subparametric approach as described in Section 3.4 and applied for the approximation of high order PDEs in Chapter 4. Indeed, we also need to apply the same transformations to the control points in order to use the same NURBS function space for both the solution and the geometrical representation, i.e. a pure isoparametric approach. Specifically, using the same linear operator of Eq. (3.3.9), we apply the transformation [44]:

$$\mathbf{P}^{per} := (\mathbf{T}^{per})^{-T}\mathbf{P} \tag{5.1.10}$$

to obtain the transformed control points $\mathbf{P}_i^{per} \in \mathbb{R}^3$, for $i = 1, \dots, n_{bf}$. We stress the fact that the representation of Ω given by the periodic NURBS basis functions $\widehat{\mathbf{R}}^{per}$ together with the control points $\mathbf{P}^{per}$ is equivalent to the one given by the original NURBS basis functions $\widehat{\mathbf{R}}$ with the control points $\mathbf{P}$, i.e.:

$$\mathbf{X}_h = \sum_{i=1}^{n_{bf}} \widehat{R}_i^{per}\mathbf{P}_i^{per} = \sum_{i=1}^{n_{bf}} \widehat{R}_i\mathbf{P}_i. \tag{5.1.11}$$

Therefore, when dealing with closed or partially closed geometries, the quantities $\widehat{\mathbf{R}}^{per}$, $\mathbf{P}^{per}$ and $\mathcal{N}_h^{per}$ are used instead of the original ones, for example in Eqs. (5.1.7), (5.1.8), and (5.1.9). In order to simplify the notations, from now on we will drop the superscript "*per*", referring indifferently to both the non-transformed or the transformed NURBS function space and control points depending on the situation at hand (either open or closed geometries).

5.2 Time discretization with Backward Differentiation Formulas

Problems governed by geometric PDEs are quite often nonlinear. Considering for example the general problem (5.1.1), all the geometric quantities, such as curvatures and normal vectors, as well as the tangent differential operators and the function spaces are defined with regard to the geometry Ω; therefore, they depend themselves on the unknown $\mathbf{x}$. In literature [26, 48, 49, 52, 147], such problems are typically discretized in time with semi-implicit, first order schemes: at each time instance, the geometrical terms are evaluated using the solution computed at the previous time instance, thus leading to the solution of a linear system. Here, we propose the use of high order implicit Backward Differentiation Formulas (BDF) [50, 159, 160].

Let us consider the time interval $[0, T]$ and divide it into $N > 0$ (for $N \in \mathbb{N}$) time steps of size $\Delta t_n = t_{n+1} - t_n$, with $n = 0, \dots, N-1$, and such that $t_0 = 0$ and $t_N = T$. We introduce the approximate geometry Ω_{n+1} as the geometry defined by the NURBS mapping:

$$\mathbf{X}_h^{n+1}(\boldsymbol{\xi}) = \sum_{i=1}^{n_{bf}} \widehat{R}_i(\boldsymbol{\xi})\,\mathbf{P}_i^{n+1} \tag{5.2.1}$$

from Eq. (3.2.7), where $\{\mathbf{P}_i^{n+1}\}_{i=1}^{n_{bf}}$ are the control points coordinates computed at the time instance t_{n+1}. Considering a fixed time step size Δt, i.e. $\Delta t_n = \Delta t$ for $n = 0, \dots, N-1$, the time discretization using a k-th order BDF scheme consists in approximating the time derivative $\dot{\mathbf{X}}_h$ at time instance t_{n+1} through a linear combination of the mappings $\mathbf{X}_h$ at the time step $n+1$ and the k previous time steps, as:

$$\dot{\mathbf{X}}_h^{n+1} \simeq \frac{1}{\Delta t}\left(\alpha_0 \mathbf{X}_h^{n+1} - \sum_{i=1}^{k} \alpha_i \mathbf{X}_h^{n+1-i}\right), \tag{5.2.2}$$

for $n \geq k-1$, with the coefficients $\alpha_i \in \mathbb{R}$, for $i = 0, \dots, k$, suitably chosen such that the approximation is of order k.

In order to avoid to solve a nonlinear system of equations at each time step, we recast it in a semi-implicit formulation. This is actually the usual procedure considered in the literature (see e.g. [48, 52]), for which the geometric terms inducing the nonlinearities are treated explicitly, i.e. are evaluated at the previous time instance. Instead, in this thesis we exploit the high order accuracy of the BDF schemes together with the simplicity of semi-implicit formulations by introducing a geometry Ω_* extrapolated from the previous time steps, according to Gregory–Newton extrapolation, which provides a guess of the approximated geometry at the next time instance. The extrapolation performed is of the k-th order, the same order as the BDF scheme considered and it is derived in the same manner as the BDF discretization itself (see e.g. [159, 160]). The extrapolated geometry Ω_* is defined by the NURBS mapping:

$$\mathbf{X}_h^{*}(\boldsymbol{\xi}) = \sum_{i=1}^{n_{bf}} \widehat{R}_i(\boldsymbol{\xi})\,\mathbf{P}_i^{*}, \tag{5.2.3}$$

where $\{\mathbf{P}_i^*\}_{i=1}^{n_{bf}}$ are the control points obtained from the sets $\left\{\mathbf{P}_i^{n+1-k}\right\}_{i=1}^{n_{bf}}, \dots, \{\mathbf{P}_i^n\}_{i=1}^{n_{bf}}$ of the control point coordinates computed at the previous time steps as:

$$\mathbf{P}_i^* := \sum_{j=1}^{k} \beta_j \mathbf{P}_i^{n+1-j}, \tag{5.2.4}$$

for $i = 1, \dots, n_{bf}$, with $\beta_j \in \mathbb{R}$, for $j = 1, \dots, k$, being appropriate coefficients guaranteeing an extrapolation of the k-th order. Then, we rewrite the problem to be solved such that it lies on the extrapolated geometry, i.e. the unknowns, as well as the integrals and the geometric quantities, become defined on Ω_*.

By referring now to the time derivative $\dot{\mathbf{x}}_h$ at time step $n+1$ of the identity function $\mathbf{x}_h$, defined on Ω_*, following Eqs. (5.2.1), (5.2.2), and (5.2.3), we approximate it with the k-th order BDF scheme as:

$$\dot{\mathbf{x}}_h^{n+1} \simeq \frac{1}{\Delta t}\left[\alpha_0 \mathbf{x}_h^{n+1} - \sum_{i=1}^{k} \alpha_i \left(\mathbf{x}_h^{n+1-i} \circ \mathbf{X}_h^{n+1-i}\right) \circ (\mathbf{X}_h^*)^{-1}\right], \tag{5.2.5}$$

for $n \geq k-1$. Then, we introduce the velocity $\mathbf{v}_h^{n+1} : \Omega_* \to \mathbb{R}^d$ at the time instance t_{n+1}, defined as:

$$\mathbf{v}_h^{n+1} := \alpha_0 \frac{\mathbf{x}_h^{n+1} - \mathbf{x}_h^{bdf,n}}{\Delta t}, \tag{5.2.6}$$

where $\mathbf{x}_h^{bdf,n} : \Omega_* \to \mathbb{R}^d$ is defined as:

$$\mathbf{x}_h^{bdf,n} := \sum_{i=1}^{k} \frac{\alpha_i}{\alpha_0} \left(\mathbf{x}_h^{n+1-i} \circ \mathbf{X}_h^{n+1-i}\right) \circ (\mathbf{X}_h^*)^{-1} \tag{5.2.7}$$

for $n \geq k-1$; moreover, we define the extrapolated solution $\mathbf{x}_h^* : \Omega_* \to \mathbb{R}^d$ at time t_{n+1} following Eqs. (5.2.3) and (5.2.4):

$$\mathbf{x}_h^* := \sum_{j=1}^{k} \beta_j \left(\mathbf{x}_h^{n+1-j} \circ \mathbf{X}_h^{n+1-j}\right) \circ (\mathbf{X}_h^*)^{-1}, \tag{5.2.8}$$

for $n \geq k-1$. We now consider, at each time instance t_{n+1}, the velocity $\mathbf{v}_h^{n+1}$ as an unknown of the problem. Then, after having computed the velocity $\mathbf{v}_h^{n+1}$, the new identity map $\mathbf{x}_h^{n+1} : \Omega_* \to \mathbb{R}^d$ is obtained as:

$$\mathbf{x}_h^{n+1} = \mathbf{x}_h^{bdf,n} + \frac{\Delta t}{\alpha_0} \mathbf{v}_h^{n+1}, \tag{5.2.9}$$

which correspond to the new geometrical mapping $\mathbf{X}_h^{n+1}$, which defines the new geometry Ω_{n+1} as in Eq. (5.2.1) and approximating $\Omega_{t_{n+1}}$.

In this framework, problem (5.1.8) is therefore discretized in time and rewritten with respect

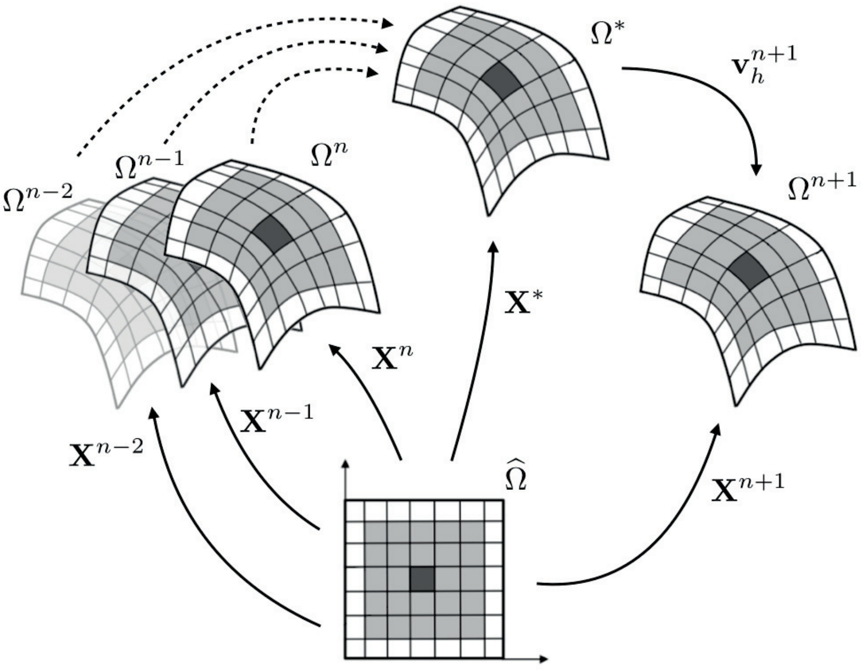


Figure 5.1 – Scheme of the geometric mappings involved in problem (5.2.12). In this example we consider a BDF scheme of order $k = 3$ for the time discretization. From the parametric domain $\widehat{\Omega}$, we define the geometries from the current and previous 2 time instances, say Ω_n, Ω_{n-1} and Ω_{n-2}, through the geometric mappings $\mathbf{X}_h^n$, $\mathbf{X}_h^{n-1}$ and $\mathbf{X}_h^{n-2}$, respectively. With a linear combination of these, we define the mapping $\mathbf{X}^*$ for the extrapolated geometry Ω_*. Then, we solve the problem on the extrapolated geometry, obtaining the velocity $\mathbf{v}_h^{n+1}$ as result. The next surface Ω_{n+1} at time instance t_{n+1}, defined through the mapping $\mathbf{X}_h^{n+1}$, is obtained by advecting the extrapolated geometry Ω_* with the calculated velocity $\mathbf{v}_h^{n+1}$.

to the unknown velocity $\mathbf{v}_h^{n+1}$ as follows:

for $n = k, \dots, N-1$, find $\mathbf{v}_h^{n+1} \in V_{g,h}^*$ such that

$$\begin{cases} m_h^*(\mathbf{v}_h^{n+1}, \boldsymbol{\varphi}_h) + a_h^*\left(\mathbf{x}_h^{bdf,n} + \dfrac{\Delta t}{\alpha_0}\mathbf{v}_h^{n+1}; \boldsymbol{\varphi}_h\right) = 0 & \forall \boldsymbol{\varphi}_h \in V_{0,h}^*, \\ \mathbf{x}_h^{bdf,0} = \mathbf{x}_{0,h}^{bdf,0}, \end{cases} \tag{5.2.10}$$

where $V_{g,h}^*$ and $V_{0,h}^*$ are the function spaces, corresponding to $V_{g,t,h}$ and $V_{0,t,h}$, defined on the extrapolated geometry Ω_*, $\mathbf{x}_{0,h}^{bdf,0}$ is obtained by Eq. (5.2.7) from suitable initial conditions $\mathbf{x}_h^n$, for $n = 0, \dots, k-1$, and $m_h^*(\cdot,\cdot)$ and $a_h^*(\cdot;\cdot)$ are forms corresponding to $m(\cdot,\cdot)$ and $a(\cdot;\cdot)$, respectively, in which the differential operators and domain of integrations are defined in Ω_*. For example, the form $m_h^*(\cdot,\cdot)$ reads:

$$m_h^*(\mathbf{v}_h^{n+1}, \boldsymbol{\varphi}_h) := \int_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \boldsymbol{\varphi}_h \, d\Omega_*. \tag{5.2.11}$$

By exploiting the extrapolated geometry Ω_* and its geometrical quantities (e.g. the curvatures and the normal) in the formulation of $a_h^*(\cdot;\cdot)$ and by performing the integrals on Ω_*, the form $a_h^*(\cdot;\cdot)$ becomes bilinear and problem (5.2.10) can be rewritten as:

for $n = k, \dots, N-1$, find $\mathbf{v}_h^{n+1} \in V_{g,h}^*$ such that

$$\begin{cases} m_h^*(\mathbf{v}_h^{n+1}, \boldsymbol{\varphi}_h) + \dfrac{\Delta t}{\alpha_0} a_h^*(\mathbf{v}_h^{n+1}, \boldsymbol{\varphi}_h) = -a_h^*(\mathbf{x}_h^{bdf,n}, \boldsymbol{\varphi}_h) & \forall \boldsymbol{\varphi}_h \in V_{0,h}^*, \\ \mathbf{x}_h^{bdf,0} = \mathbf{x}_{0,h}^{bdf,0}. \end{cases} \tag{5.2.12}$$

In Figure 5.1 we report a sketch of the scheme leading to the computation of the surface Ω_{n+1} to highlight the time discretization procedure described in this section. We remark that problem (5.2.12) is linear. Moreover, at each time instance the extrapolation of the geometry Ω_* is potentially an accurate guess of the unknown geometry of the next time instance, aided by the exactness of the geometrical representation of NURBS in the IGA-based spatial discretization.

5.3 Approximation of the mean curvature flow

Let us consider the mean curvature flow problem defined in Eq. (2.4.5). By assuming sufficient regularity for all the geometric quantities involved, by using the relation of Eq. (1.1.28), and integrating by parts the Laplace-Beltrami operator, we recast the mean curvature problem (2.4.5) in the general formulation of Eq. (5.1.2) with the form $a(\cdot;\cdot)$ being defined as [147]:

$$a_{MCF}(\mathbf{x};\boldsymbol{\varphi}) := \mu \int_{\Omega} \nabla_{\Omega}\mathbf{x} : \nabla_{\Omega}\boldsymbol{\varphi}\, d\Omega. \tag{5.3.1}$$

The mean curvature flow problem in weak form then reads:

for a.e. $t \in (0,T)$, find $\mathbf{x} \in V_{g,t}$ such that

$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t + \mu \int_{\Omega_t} \nabla_{\Omega_t}\mathbf{x} : \nabla_{\Omega_t}\boldsymbol{\varphi}\, d\Omega_t = 0 & \forall \boldsymbol{\varphi} \in V_{0,t}, \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0 \end{cases} \tag{5.3.2}$$

with $V_{g,t}$ and $V_{0,t}$ being subsets of $\left[H^1(\Omega)\right]^3$.

The semi-discretized problem obtained by the NURBS-based IGA approximation of the mean curvature flow problem is in the form of Eq. (5.1.8), with the function spaces $V_{g,t,h}$ and $V_{0,t,h}$ being subsets of $V_{t,h} := [H^1(\Omega_t)]^3 \cap [\mathcal{N}_h]^3$. To rewrite the semi-discretized problem into the parametric domain $\widehat{\Omega}$ performing a pull-back operation, the form of Eq. (5.3.1) becomes:

$$\widehat{a}_{MCF}(\mathbf{X}_h;\widehat{\boldsymbol{\varphi}}_h) := \mu \int_{\widehat{\Omega}} \left(\widehat{F}\widehat{G}^{-1}\widehat{\nabla}\mathbf{X}_h\right) : \left(\widehat{F}\widehat{G}^{-1}\widehat{\nabla}\widehat{\boldsymbol{\varphi}}_h\right) \widehat{g}\, d\widehat{\Omega} \tag{5.3.3}$$

for $\widehat{\boldsymbol{\varphi}}_h \in \widehat{V}_{0,h}$, having used the relation in Eq. (1.1.16), with the function spaces $\widehat{V}_{g,h}$ and $\widehat{V}_{0,h}$ being subsets of $\widehat{V}_h := [H^1(\widehat{\Omega})]^d \cap [\widehat{\mathcal{N}_h}]^d$.

Following Section 5.2, we propose to discretize in time the mean curvature flow problem employing the BDF schemes of order k, together with the extrapolation of the geometry. The resulting full discrete problem, written on the extrapolated surface Ω_*, reads:

for $n = k, \ldots, N-1$, find $\mathbf{v}_h^{n+1} \in V_{g,h}^*$ such that

$$\begin{cases} \displaystyle\int_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \boldsymbol{\varphi}_h\, d\Omega_* + \mu\frac{\Delta t}{\alpha_0} \int_{\Omega_*} \nabla_{\Omega_*}\mathbf{v}_h^{n+1} : \nabla_{\Omega_*}\boldsymbol{\varphi}_h\, d\Omega_* = -\int_{\Omega_*} \nabla_{\Omega_*}\mathbf{x}_h^{bdf,n} : \nabla_{\Omega_*}\boldsymbol{\varphi}_h\, d\Omega_* \\ \hfill \forall \boldsymbol{\varphi}_h \in V_{0,h}^*, \\ \mathbf{x}_h^{bdf,0} = \mathbf{x}_{0,h}^{bdf,0}, \end{cases} \tag{5.3.4}$$

with the function spaces $V_{g,h}^*$ and $V_{0,h}^*$ being subsets of $V_h^* := [H^1(\Omega_*)]^3 \cap [\mathcal{N}_h]^3$.

5.3.1 Numerical results

We consider the numerical approximation of the mean curvature flow problem of Eq. (2.4.5), in particular by using the formulation proposed in Eq. (5.3.4) described in Section 5.3. In all out tests we set $\mu = 1$.

Test 5.1.1. We consider the mean curvature flow of an initial unit sphere $\Omega_0 \subset \mathbb{R}^3$. By exploiting the radial symmetry of the sphere, the geometry Ω_t under mean curvature flow remains a sphere for each $t \in (0, T)$ with evolution in spherical coordinates described by the ordinary differential equation [152]:

$$\begin{cases} \dot{r} = -\dfrac{2}{r} & \text{for } t \in (0, T), \\ r(0) = r_0, \end{cases} \tag{5.3.5}$$

where $r(t)$ is the radius of the sphere at time t and r_0 the radius of Ω_0. This equation admits the exact solution:

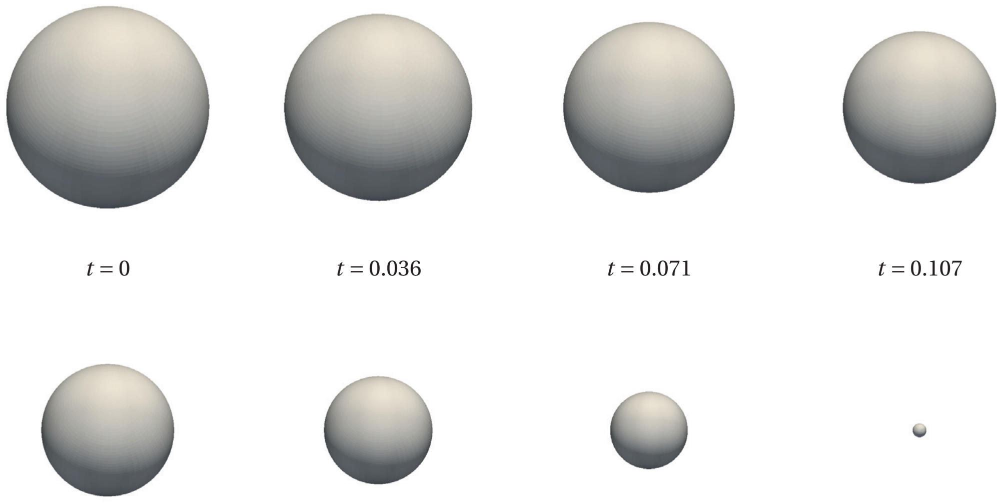
$$r(t) = \sqrt{r_0^2 - 4t} \qquad \text{for } t \in [0, T), \tag{5.3.6}$$

from which it is evident that the sphere degenerates into a point at $t = \dfrac{r_0^2}{4}$. Thus, considering an initial sphere Ω_0 of radius $r_0 = 1$, we expect the solution of problem (2.4.5) to be represented by a shrinking sphere with radius described by Eq. (5.3.6) and collapsing into a single point at time $T = 0.25$.

Figure 5.2 displays the evolution of the sphere Ω_{n+1} obtained by the numerical approximation of problem (5.3.4), at different time instances; the evolution of the area $|\Omega_n|$ is reported in Figure 5.3 together with the evolution of the exact area $|\Omega_{t_n}|$ computed with Eq. (5.3.6). The sphere is represented by NURBS of degree $p = 2$ and C^1-continuous a.e., for a total of 220 elements, yielding 384 DOFs[1]. Time discretization is performed by employing a BDF scheme of order $k = 2$ with fixed time step size $\Delta t = 0.001$. The linear systems arising from the full discretization of the PDEs at each time step are solved by using the GMRES method with the ILUT preconditioner [50], with the stopping criterion being the relative residual (in Euclidean norm) below a tolerance of 10^{-9}.

---

[1] The amount of DOFs reported corresponds to the size of the linear system solved at each time instance; therefore, it takes into account for the constraints set to build the periodic basis functions and the fact that the solution is vector-valued (the velocity $\mathbf{v} \in \mathbb{R}^3$ for each control point).



$t = 0$ $t = 0.036$ $t = 0.071$ $t = 0.107$

$t = 0.142$ $t = 0.178$ $t = 0.213$ $t = 0.249$

Figure 5.2 – Test 5.1.1. Mean curvature flow of a sphere. Solution at different time instances.

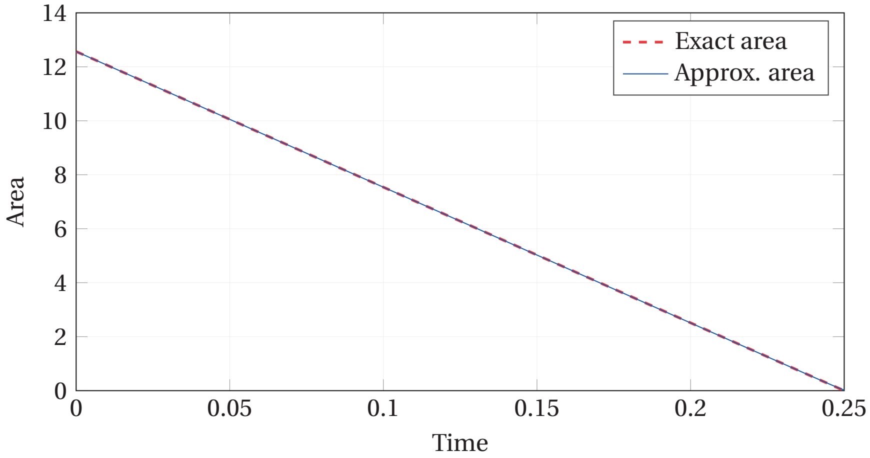
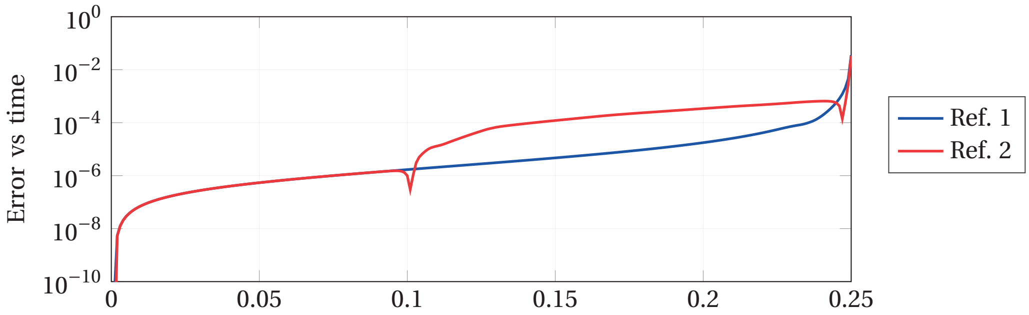
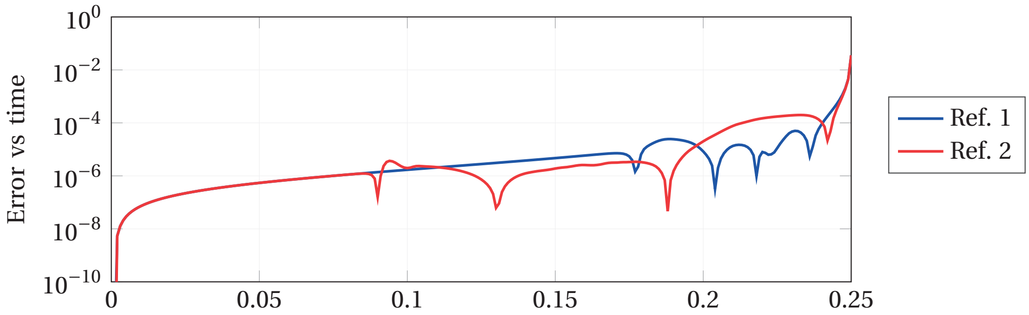


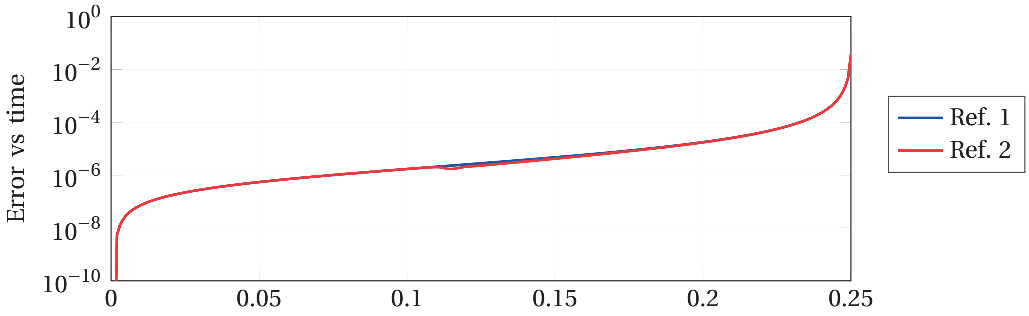
Figure 5.3 – Test 5.1.1. Mean curvature flow of a sphere. Evolution of the approximated area $|\Omega_n|$ and exact area $|\Omega_{t_n}|$ vs. time t; NURBS of degree $p = 2$ and C^1-continuous a.e. with 220 mesh elements are used.



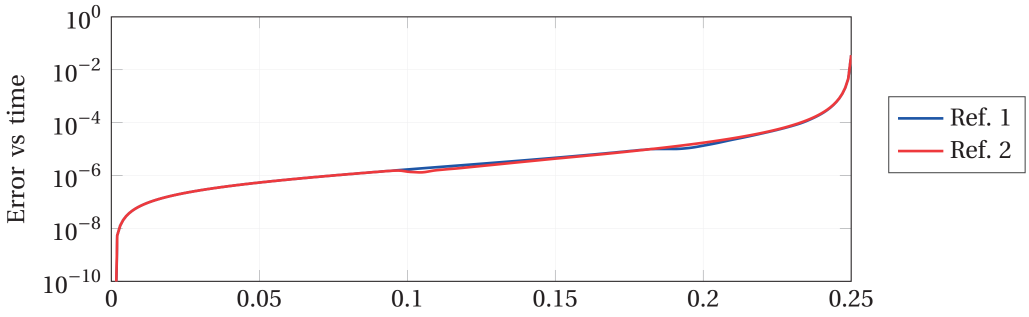
(a) $p = 2$, C^1-continuous a.e., 3^2 quad. pts.



(b) $p = 3$, C^2-continuous a.e., 4^2 quad. pts.

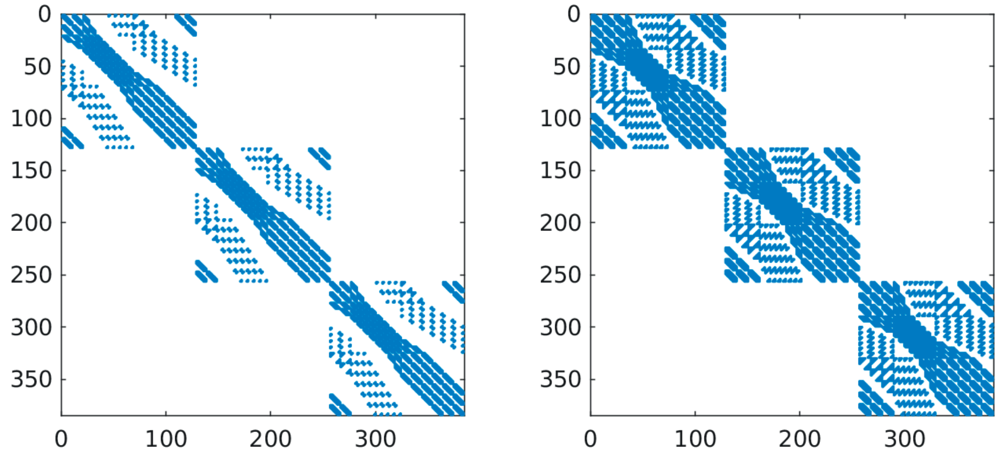


(c) $p = 2$, C^1-continuous a.e., 7^2 quad. pts.



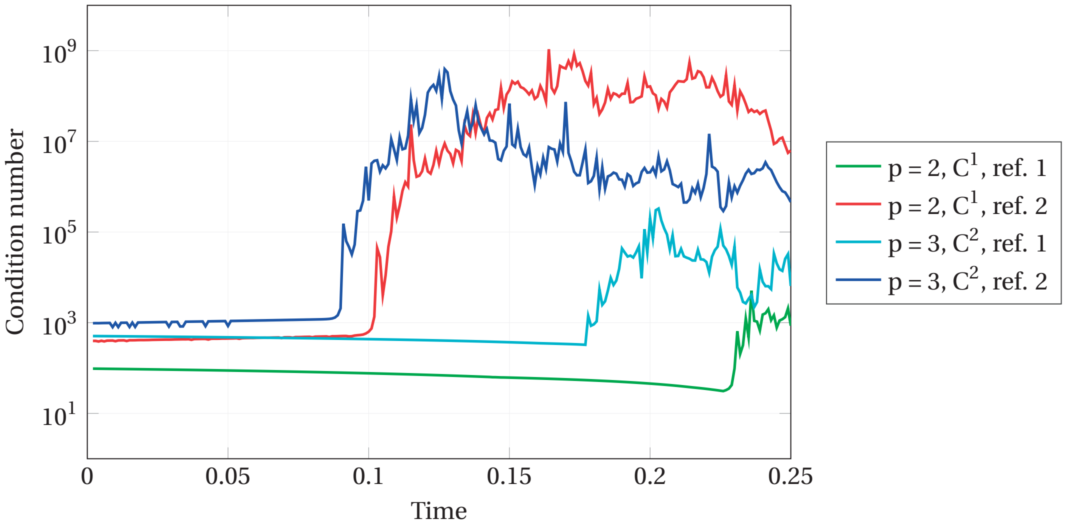
(d) $p = 3$, C^2-continuous a.e., 7^2 quad. pts.

Figure 5.4 – Test 5.1.1. Mean curvature flow of a sphere. Behavior of the error on the area err_n vs time t for meshes with different NURBS basis functions (ref. 1 with 384 DOFs and ref. 2 with 6,144 DOFs) and using $(p+1)^2$ (in (a) and (b)) and 7^2 (in (c) and (d)) quadrature points per mesh element.



(a) $p = 2$, C^1-continuous a.e.

(b) $p = 3$, C^2-continuous a.e.



(c) Evolution of the condition number vs. time t

Figure 5.5 – Test 5.1.1. Mean curvature flow of a sphere. Sparsity patterns (a) and (b) and evolution of the condition number $\kappa(A)$ of the matrix associated to the fully discrete problem (5.2.12) vs. time t (c) using NURBS basis functions of degrees $p = 2$ and 3, C^1- and C^2-continuous a.e., respectively, and two refinement levels yielding 384 and 6,144 DOFs, respectively.

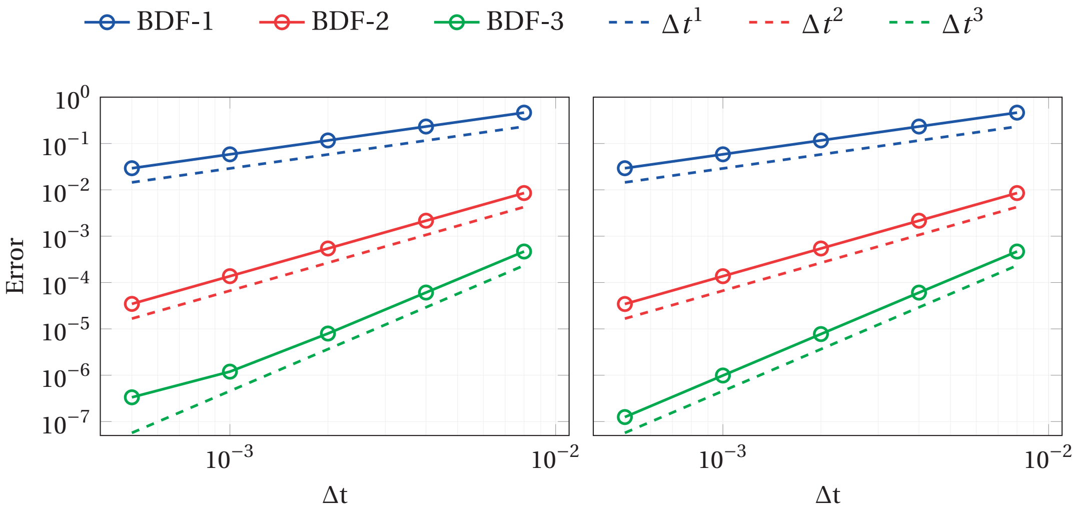
We report in Figure 5.4 the behavior of the errors on the numerically approximated area vs time, say $err_n := |\Omega_{t_n} - \Omega_n|$, obtained by solving problem (5.2.12) with NURBS of degree $p = 2$ and 3, which are C^{p-1}-continuous a.e., respectively. We compare the errors obtained using meshes of 220 and 2,380 elements for the $p = 2$, C^1-continuous NURBS basis, while 275 and 2,555 elements for the $p = 3$, C^2-continuous basis (yielding 384 and 6,144 DOF for both $p = 2$ and 3). In particular, Figures 5.4a and 5.4b show the errors obtained using the standard Gauss–Legendre quadrature rule with $(p+1)^2$ points per mesh element for the approximation of the integrals, while Figures 5.4c and 5.4d show the errors obtained using 7^2 quadrature points per element, thus with a significantly increased accuracy of the numerical integration. We observe that the errors are very small in all the cases and only increase when the geometry tends to degenerate in a point, as expected from the exact solution of Eq. (5.3.6). In addition, a smoother behavior of the error err_n is observed when using a large number of quadrature nodes. Nevertheless, the errors remain very small, even for the standard Gauss–Legendre quadrature formulas with $(p+1)^2$ points typically used in IGA and employed in the rest of this thesis.

We report in Figures 5.5a and 5.5b the sparsity patterns of the matrix A arising from the fully discrete problem (5.2.12) with NURBS of degree $p = 2$ and 3, respectively, with 384 DOFs in both the cases. The evolutions of the condition number of the matrices associated to problem (5.2.12) at each time step are reported in Figure 5.5c for the NURBS already considered for the results in Figure 5.4; the condition number $\kappa(A)$ is actually a lower bound of the 1-norm condition number of the matrix A. The condition number κ increases with the degree of the NURBS basis functions and when the mesh is refined. We remark that, for this specific problem, the NURBS mapping is singular at the poles of the sphere, which leads to high values of the condition number. Moreover, the sphere shrinks according to the mean curvature flow and eventually degenerates in a point, another reason for the increment of the condition numbers. Nevertheless, in the case under consideration, the condition numbers $\kappa(A)$ are never high enough to significantly interfere with the accuracy of the GMRES solver (for the chosen tolerance).

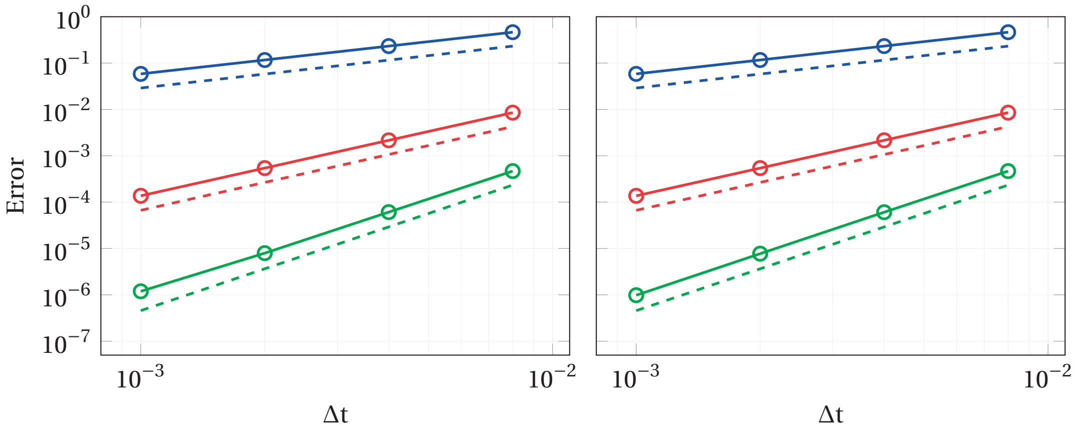
In order to compare the performance of the proposed scheme with BDFs of different orders with respect to the time step size Δt, simulations with BDFs of orders $k = 1$, 2, and 3 have been performed for meshes comprised of NURBS basis functions of degrees $p = 2$ and 3, which are C^0- and C^{p-1}-continuous a.e. on the surface. Errors on the area with respect to the exact solution are reported in Figure 5.6 in logarithmic scale; the reported errors are computed as:

$$err_{\tilde{n}} := |\Omega_{t_{\tilde{n}}} - \Omega_{\tilde{n}}| \tag{5.3.7}$$

at fixed time $t_{\tilde{n}} = 0.016$, with $\tilde{n} = t_{\tilde{n}}/\Delta t$. The BDF schemes are initialized with the corresponding numbers of exact time steps in order to bootstrap the time integration method correctly, such that order of convergence of k is maintained for each BDF used. The meshes considered are built out of NURBS basis functions of degrees $p = 2$ and 3, with 220 and 275 elements, respectively; for the degree $p = 2$, meshes with basis functions C^1-continuous a.e. and C^0-



(a) $p = 2$, C^1-continuous a.e. (b) $p = 3$, C^2-continuous a.e.

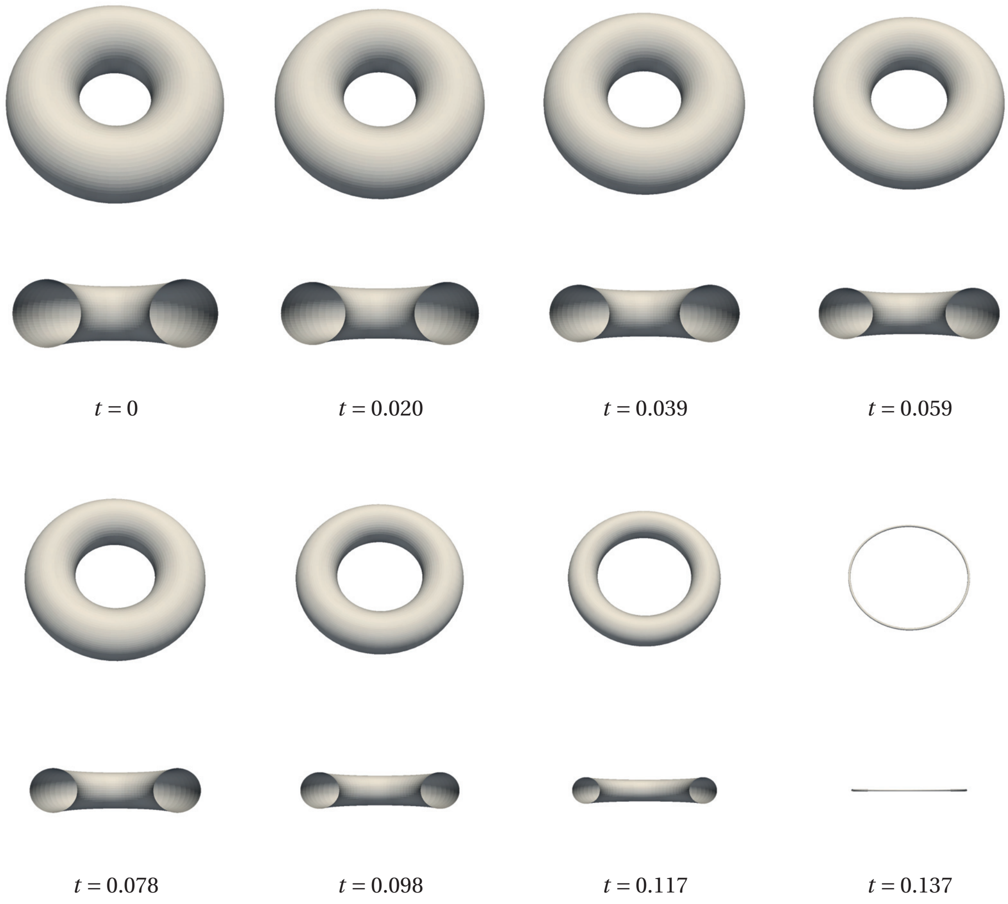


(c) $p = 2$, C^0-continuous a.e. (d) $p = 3$, C^0-continuous a.e.

Figure 5.6 – Test 5.1.1. Mean curvature flow of a sphere. Absolute errors on the area at time $t_{\tilde{n}} = 0.016$ $err_{\tilde{n}}$ vs. Δt, for different BDF schemes (BDF of orders $k = 1$, 2, and 3) and NURBS basis functions ($p = 2$ and 3, which are C^0- and C^{p-1}-continuous a.e.).

continuous a.e. are considered, with 384 and 600 DOFs, respectively; for the degree $p = 3$, meshes with basis functions C^2-continuous a.e. and C^0-continuous a.e. are considered, with 384 and 864 DOFs respectively. We remark that the errors corresponding to the spatial discretization are significantly small, even when approximating the problem with a low amount of mesh elements; this fact permits us to employ high order temporal discretizations and recover the full rate of convergence. Also, since the smooth C^{p-1}-continuous basis functions lead to a smaller number of DOFs than their C^0-continuous counterpart, the former generally lead to more efficient and accurate discretizations.

Test 5.1.2. Next, we study the evolution of a torus subject to mean curvature flow. We consider



$t = 0$ $t = 0.020$ $t = 0.039$ $t = 0.059$

$t = 0.078$ $t = 0.098$ $t = 0.117$ $t = 0.137$

Figure 5.7 – Test 5.1.2a. Mean curvature flow of a torus with $R_0 = 1$ and $r_0 = 0.5$. Solution at different time instances.

a family of toric surfaces $\Omega_0 \subset \mathbb{R}^3$ described by:

$$\left(R_0 - \sqrt{x^2 + y^2}\right)^2 + z^2 = r_0^2 \qquad (x, y, z) \in \mathbb{R}^3, \tag{5.3.8}$$

in a standard Cartesian coordinate system, where R_0 and r_0 are the torus' major and minor radii, respectively, of the initial configuration corresponding to Ω_0, being $R_0 > 0$ and $r_0 \in (0, R_0)$. Depending on the ratio between the two radii R_0/r_0, the torus is evolving either to collapse into a circle or to self-penetrate and merge into a disk. Figures 5.7 and 5.9 show the evolution of tori with $R_0 = 1$, $r_0 = 0.5$ and $R_0 = 1$, $r_0 = 0.7$, respectively, subject to mean curvature flow. The first torus has an aspect ratio R_0/r_0 such that it collapses into a circle, while the second one tends to merge into an ellipsoid; since we adopt a parametric representation of the geometry and we do not support topology changes, we let the geometry evolve until a self-intersection of the surface occurs. The evolution of the toric areas is plotted in Figures 5.8 and 5.10, respectively. We consider NURBS with basis functions of degree $p = 2$ and globally C^1-continuous, with 836 elements and 1,536 DOFs for both the cases; we used a BDF scheme of order $k = 2$ for

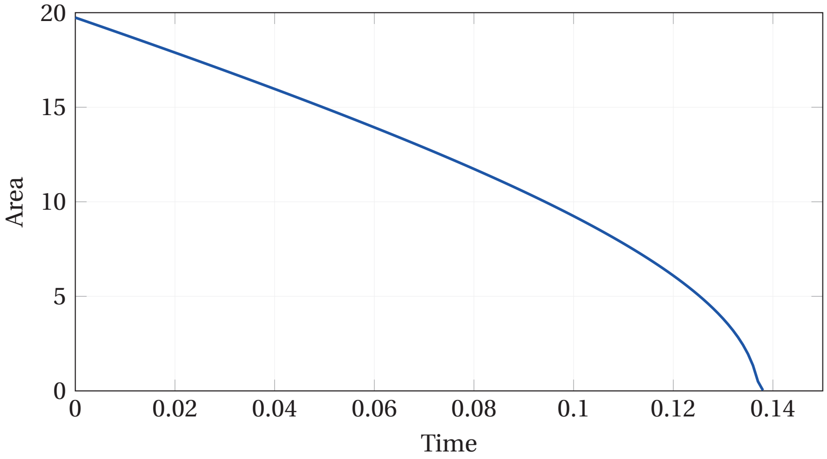


Figure 5.8 – Test 5.1.2a. Mean curvature flow of a torus with $R_0 = 1$ and $r_0 = 0.5$. Evolution of the approximated area $|\Omega_n|$ vs. time t; NURBS of degree $p = 2$ and globally C^1-continuous with 836 mesh elements, yielding 1,536 DOFs, are used.

integration in time with time step size $\Delta t = 0.001$.

Test 5.1.3. We consider the mean curvature flow of an open surface, in particular a cylindrical shell. We parametrize the cylinder by its radius R_0 and height L_0. The bottom and top bases of the cylinder (two circles of radius R_0) are fixed (i.e. we set $\mathbf{x}_h = \mathbf{x}_{0,h}$ on $\partial\Omega$), while the lateral surface (Ω_t) is free to evolve according to the mean curvature flow. The geometry minimizing the area depends on the aspect ratio L_0/R_0 of the initial cylinder Ω_0. In particular, the final solution may either be discontinuous, consisting in two circles at the bases of the cylinder, and thus involving a topology change (known as *Goldschmidt solution* [161]), or exhibit a *catenoid* as local minimum, generated by rotating the catenary of equation:

$$x = a\cosh\left(\frac{y}{a}\right) \tag{5.3.9}$$

along the y-axis, with $a \in \mathbb{R}$ being solution of the nonlinear relation:

$$\cosh\left(\frac{L_0}{2a}\right) - \frac{R_0}{a} = 0. \tag{5.3.10}$$

Such catenoid has area equal to:

$$A_{cat} = \pi a^2 \left[\sinh\left(\frac{L_0}{a}\right) + \frac{L_0}{a}\right]. \tag{5.3.11}$$

Figures 5.11 and 5.12 show the evolution of two cylinders, the first with radius $R_0 = 1$ and height $L_0 = 1$, while the second one with $R_0 = 1$ and $L_0 = 2$. The meshes considered in the spatial approximation are both NURBS built of basis functions of degree $p = 2$ and globally C^1-continuous, with 456 elements, for a total of 1,152 DOFs; time integration is performed employing a BDF scheme of order 2, with time step size $\Delta t = 0.001$. The evolutions of the

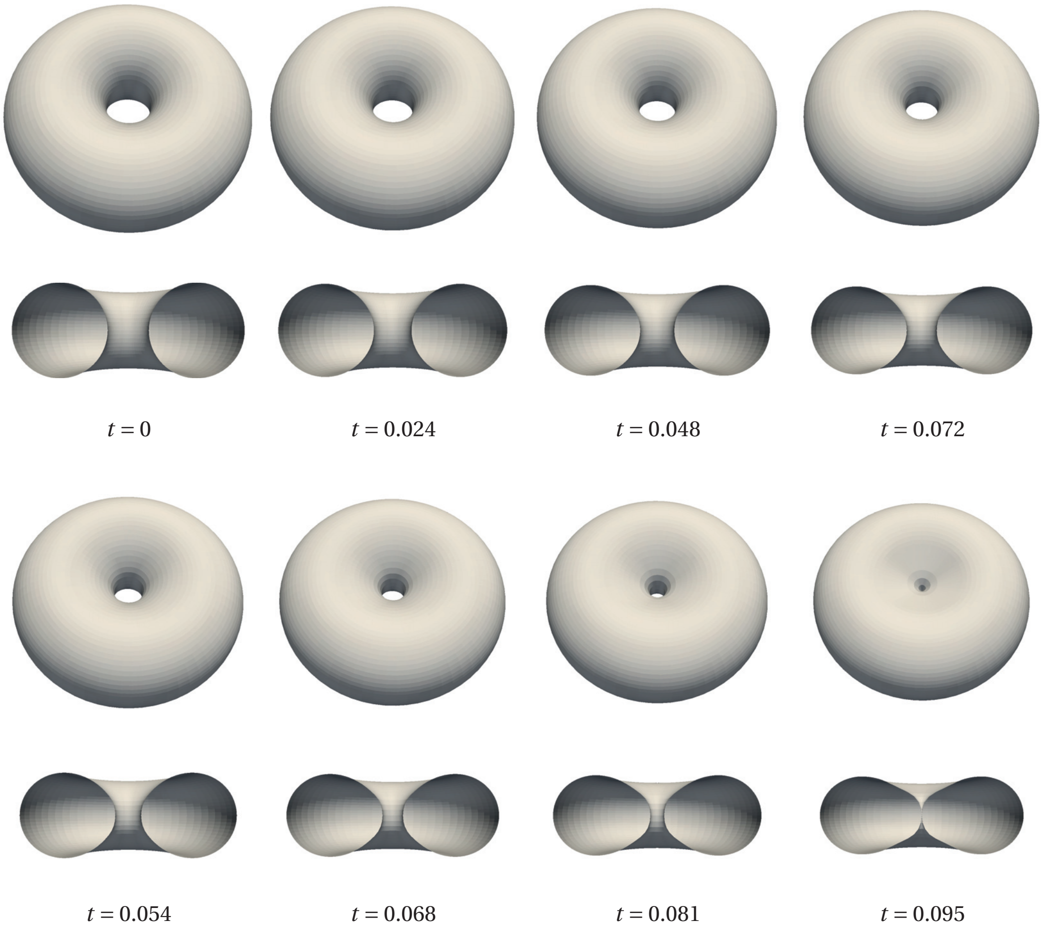


Figure 5.9 – Test 5.1.2b. Mean curvature flow of a torus with $R_0 = 1$ and $r_0 = 0.7$. Solution at different time instances.

areas $|\Omega_t|$ are also plotted in Figures 5.11 and 5.12, respectively. The first cylinder has aspect ratio $L_0/R_0 = 1$ such that a catenoid is a local minimum and the numerical solution effectively converges to such geometry. The second cylinder (for $L_0/R_0 = 2$) does not feature a catenoid as local minimum, therefore the minimization process continues towards the solution with topology changes, which we stop when a singularity in the geometrical mapping (an indicator of a topology change) is reached. In both the cases, we obtain accurate solutions even by employing spatial discretizations involving a very small amount of mesh elements and DOFs.

We remark that the formulation for the mean curvature flow problem based on Eq. (5.3.2) can suffer from tangential motion of the control points. In some situations, this may lead to deterioration of the mesh quality. Nevertheless, in the cases analyzed in this work, also thanks to the exactness of the NURBS geometrical representation and the calculation of the geometrical quantities, the tangential motions are negligible and do not affect the outcome of the simulations. If necessary, in order to eliminate or at least to control such tangential motions, a splitting similar to the one which will be proposed for the Willmore flow in Section 5.4 can be

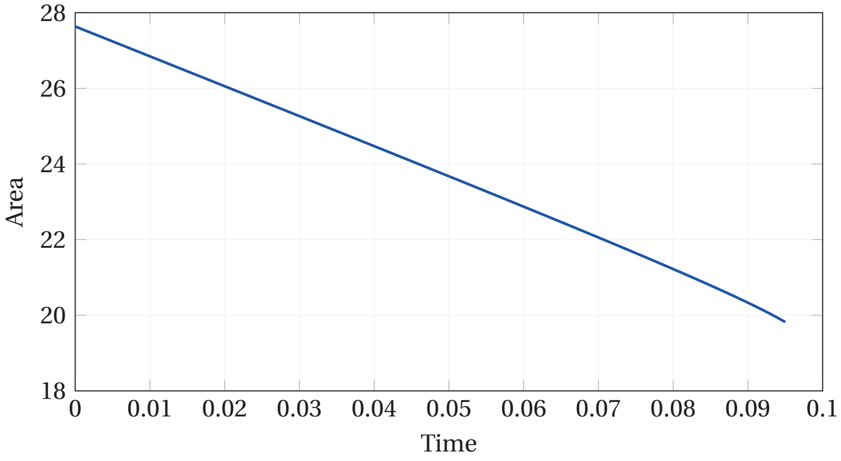


Figure 5.10 – Test 5.1.2b. Mean curvature flow of a torus with $R_0 = 1$ and $r_0 = 0.7$. Evolution of the approximated area $|\Omega_n|$ vs. time t; NURBS of degree $p = 2$ and globally C^1-continuous with 836 mesh elements, yielding 1,536 DOFs, are used.

devised for the mean curvature flow too.

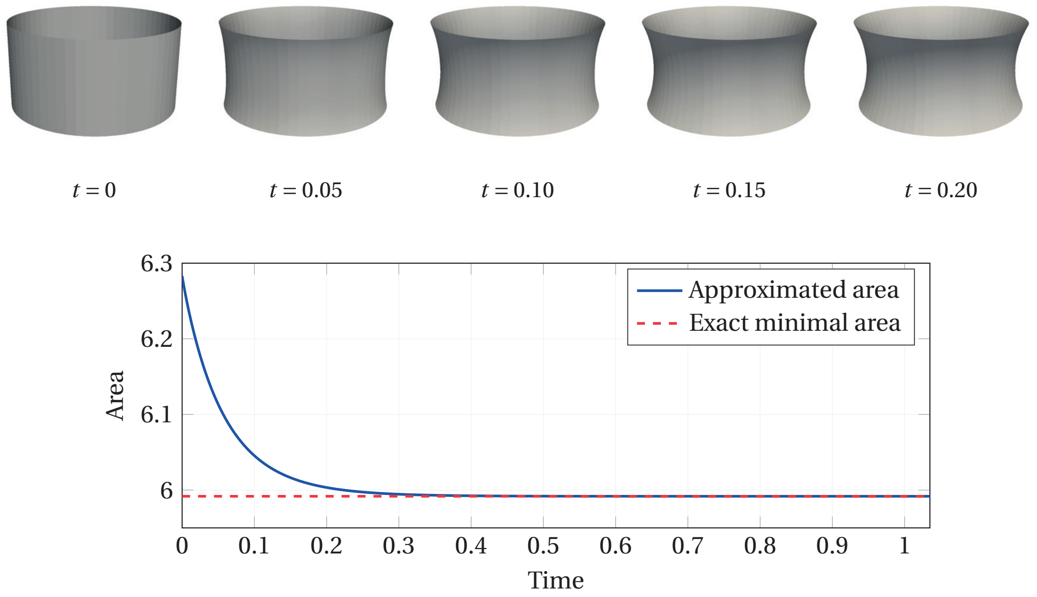


Figure 5.11 – Test 5.1.3a. Mean curvature flow of a cylinder with $R_0 = 1$ and $L_0 = 1$. Evolution of the approximated area $|\Omega_n|$ vs. time t for a cylinder with $R_0 = 1$ and $L_0 = 1$ (a) and with $R_0 = 1$ and $L_0 = 2$ (b); NURBS of degree $p = 2$ and globally C^1-continuous with 456 mesh elements, yielding 1,152 DOFs, are used.

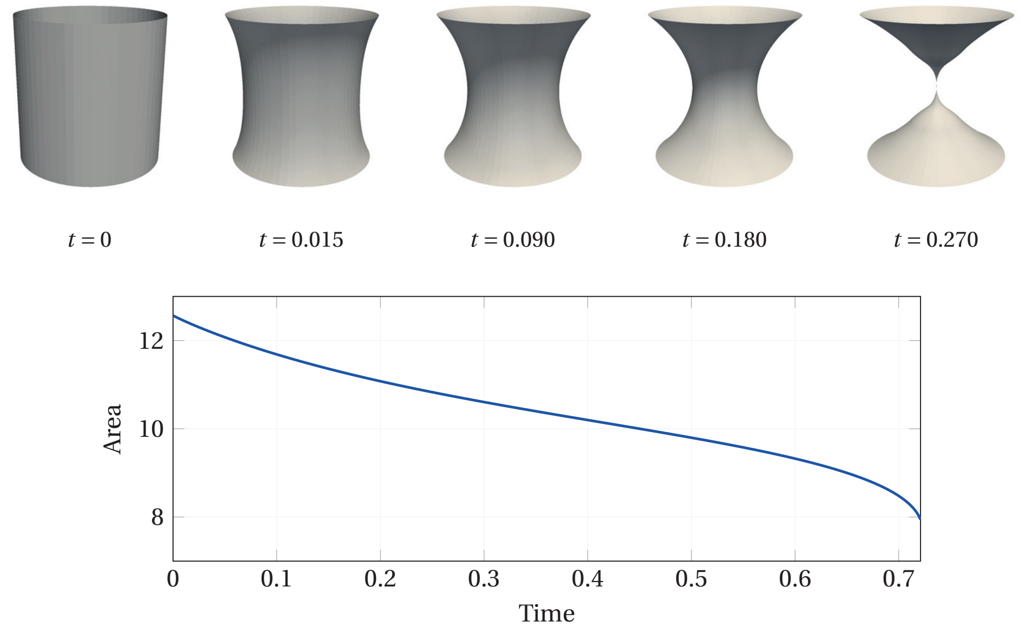


Figure 5.12 – Test 5.1.3b. Mean curvature flow of a cylinder with $R_0 = 1$ and $L_0 = 2$. Evolution of the approximated area $|\Omega_n|$ vs. time t; NURBS of degree $p = 2$ and globally C^1-continuous with 456 mesh elements, yielding 1,152 DOFs, are used.

5.4 Approximation of the Willmore flow

The Willmore flow problem of Eq. (2.5.6) is a nonlinear time dependent high order PDE. For spatial discretizations based on the standard FEM with C^0-continuous basis functions, mixed formulations to decrease the order of the PDE are commonly used [48, 49, 153]. In addition, the term HK in the shape derivative of the Willmore energy of Eq. (2.5.3), which involves both the mean (H) and the Gauss (K) curvatures, depends nonlinearly on the principal curvatures and it is difficult to treat with variational methods [48]; therefore, terms as K or the normal to the surface $\mathbf{n}_\Omega$ are usually avoided through suitable manipulations in the weak formulation of the problem. Regardless of the order of the differential problem, these considerations lead to the adoption of mixed formulations where additional unknowns are introduced, usually being the mean curvature H or the mean curvature vector $\mathbf{H}$ [48, 49, 153, 154], for which the resulting PDEs are of the second order, and the (approximate) curvature usually treated in a weak sense.

In the framework of NURBS-based IGA, one benefits both from the exact representation of the initial geometry Ω, with the possibility of calculating the geometrical quantities directly from the NURBS representation, and the ability to treat high order tangential PDEs in a straightforward manner, as outlined in Chapter 4. Therefore, by considering closed surfaces, we propose the following weak formulation of the Willmore flow problem:

for a.e. $t \in (0,T)$, find $\mathbf{x} \in V_t$ and $v \in W_t$ such that

$$\begin{cases} \displaystyle\int_{\Omega_t} v\,\psi \, d\Omega_t + \mu \int_{\Omega_t} (\Delta_{\Omega_t}\mathbf{x}\cdot\mathbf{n}_{\Omega_t})\Delta_{\Omega_t}\psi \, d\Omega_t \\ \displaystyle\qquad\qquad + \mu \int_{\Omega_t} \left(\frac{1}{2}H^2 - 2K\right)\left(\Delta_{\Omega_t}\mathbf{x}\cdot\mathbf{n}_{\Omega_t}\right)\psi \, d\Omega_t = 0 & \forall \psi \in W_t, \\ \displaystyle\int_{\Omega_t} \dot{\mathbf{x}}\cdot\boldsymbol{\varphi}\, d\Omega_t - \int_{\Omega_t} v\,\mathbf{n}_{\Omega_t}\cdot\boldsymbol{\varphi}\, d\Omega_t = 0 & \forall \boldsymbol{\varphi} \in V_t, \\ \mathbf{x}(0) = \mathbf{x}_0, \end{cases} \tag{5.4.1}$$

where $V_t := \left[H^2(\Omega_t)\right]^3$, while $W_t := H^2(\Omega_t)$; the normal velocity v (defined in Eq. (2.1.10)) is also an unknown of the problem.

We consider NURBS-based IGA for the approximation of (5.4.1). We therefore discretize the equations following the same procedure described in Section 3.4, thus seeking the trial and test functions for the unknowns $\mathbf{x}_h$ and v_h in the function spaces $V_{t,h} := V_t \cap [\mathcal{N}_h]^3$ and $W_{t,h} := W_t \cap \mathcal{N}_h$, respectively. We remark that, with IGA, the evaluation of the terms involving H and K is straightforward, since the curvatures can be computed directly and “exactly” from the NURBS mapping $\mathbf{x}_h$. Problem (5.4.1) is rewritten into the parametric domain $\widehat{\Omega}$ through a pull-back operation as described in Section 3.4 and similarly to the approximation of the mean curvature flow problem of Section 5.3; in this case, we also use the relation of Eq. (1.1.20) for the treatment of the Laplace-Beltrami operator in the parametric domain $\widehat{\Omega}$. We remark that, since we need to ensure that the test and trial function spaces are subsets of H^2, we consider NURBS function spaces with basis functions at least globally C^1-continuous a.e. in Ω_t, for all $t \in (0, T)$. Moreover, since Ω is closed, we consider NURBS periodic function spaces, as mentioned in Section 3.4. Finally, we highlight that, with the formulation of Eq. (5.4.1), the amount of tangential motions is limited, since the velocity $\mathbf{v}$ evolving the surface is aligned, in the "weak sense", along the normal $\mathbf{n}_\Omega$ to the current surface Ω.

Considering a time discretization performed with the BDF schemes as described in Section 5.2, the full discrete Willmore flow problem reads:

$$\text{for } n = 0, \dots, N-1, \text{ find } \mathbf{v}_h^{n+1} \in V_h^* \text{ and } v_h^{n+1} \in W_h^* \text{ such that}$$

$$\begin{cases} \displaystyle\int_{\Omega_*} v_h^{n+1} \psi_h \, d\Omega_* + \mu \frac{\Delta t}{\alpha_0} \int_{\Omega_*} (\Delta_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \mathbf{n}_{\Omega_*}) \Delta_{\Omega_*} \psi_h \, d\Omega_* \\ \qquad\qquad \displaystyle + \mu \frac{\Delta t}{\alpha_0} \int_{\Omega_*} \left[\frac{1}{2} (H_h^*)^2 - 2K_h^* \right] \left(\Delta_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \mathbf{n}_{\Omega_*} \right) \psi_h \, d\Omega_* \\ \qquad \displaystyle = -\mu \int_{\Omega_*} \left(\Delta_{\Omega_*} \mathbf{x}_h^{bdf,n} \cdot \mathbf{n}_{\Omega_*} \right) \Delta_{\Omega_*} \psi_h \, d\Omega_* \\ \qquad\qquad \displaystyle - \mu \int_{\Omega_*} \left[\frac{1}{2} (H_h^*)^2 - 2K_h^* \right] \left(\Delta_{\Omega_*} \mathbf{x}_h^{bdf,n} \cdot \mathbf{n}_{\Omega_*} \right) \psi_h \, d\Omega_* & \forall \psi_h \in W_h^*, \\ \displaystyle\int_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \boldsymbol{\varphi}_h \, d\Omega_* - \int_{\Omega_*} v_h^{n+1} \mathbf{n}_{\Omega_*} \cdot \boldsymbol{\varphi}_h \, d\Omega_* = 0 & \forall \boldsymbol{\varphi}_h \in V_h^*, \\ \mathbf{x}_h^{bdf,0} = \mathbf{x}_{0,h}^{bdf,0}, \end{cases} \tag{5.4.2}$$

where V_h^* and W_h^* are the function spaces $V_{t,h}$ and $W_{t,h}$ built on Ω_*, respectively, and H_h^* and K_h^* represent the mean curvature and Gauss curvature of the extrapolated geometry Ω_*, respectively. Problem (5.4.2) is then solved after being recast into the parametric domain $\widehat{\Omega}$.

5.4.1 Numerical results

We consider the numerical approximation of the Willmore flow problem defined in Eq. (2.5.6) using the numerical scheme (5.4.1) proposed in Section 5.4. For all the tests we set $\mu = 1$.

Test 5.2.1. As initial geometry Ω_0 we consider ellipsoids described by the following relation:

$$\frac{x^2}{a_0^2} + \frac{y^2}{b_0^2} + \frac{z^2}{c_0^2} = 1 \qquad (x, y, z) \in \mathbb{R}^3, \tag{5.4.3}$$

where $a_0, b_0, c_0 \in \mathbb{R}$ are positive constants denoting its aspect ratio. It is known that an ellipsoid should converge to a sphere under Willmore flow [129], which has Willmore energy J_W equal to 8π. For our numerical test, we consider the approximation of the Willmore flow applied to an initial ellipsoid Ω_0 with parameters $a_0 = 4$, $b_0 = 4$, and $c_0 = 1$. The geometry Ω_0 is

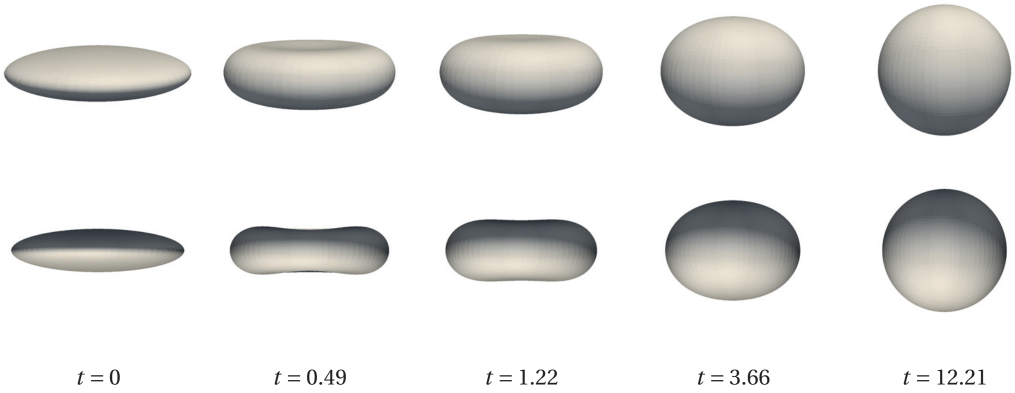


Figure 5.13 – Test 5.2.1. Willmore flow of an ellipsoid with $a_0 = 4$, $b_0 = 4$ and $c_0 = 1$. Solution at different time instances.

represented as a NURBS surface with basis functions of degrees $p = 2$ and 3, being C^1- and C^2-continuous a.e., respectively, with two h-refinement levels for each degree. The considered meshes with NURBS of degree $p = 2$ are made of 684 and 2,380 elements, respectively for the two refinement levels; the meshes with basis functions of degree $p = 3$ are instead made of 779 and 2,555 elements, respectively. With respect to the two h-refinement levels, the total number of DOFs amounts to 2,048 and 8,192, independently of the degree p of the NURBS basis functions[2]. Integration in time is performed by employing the BDF scheme of order $k = 2$ with a fixed time step size $\Delta t = 0.01$.

Figure 5.13 shows the solution obtained at different time instances, with the mesh comprised of 779 elements. The evolution in time of the Willmore energy, together with the Willmore energy associated to a sphere (indicated as *Exact final energy*), is reported in Figure 5.14, together with the evolution in time of the area and the volume of the approximated geometry Ω_n. We remark that problem (2.5.6) does not involve any constraint on the area and the volume of the surface, which are in principle free to evolve while the Willmore energy J_W is being minimized; as a matter of fact, we notice that the evolutions of the area and the volume are sensitive to the discretization under consideration. We obtain, with the coarsest mesh built of NURBS of degree $p = 2$ a final error on the Willmore energy equal to 0.6496 (2.585%); when refining the mesh, we obtain a significant reduction of such error, being equal to 0.1696 (0.675%). Instead, using NURBS of degree $p = 3$ yields better results, with errors equal to 0.0237 (0.094%) and 0.0055 (0.022%) for the first and second h-refinement levels, respectively.

Finally, we report in Figures 5.15a and 5.15b the sparsity patterns of the matrix associated to the full discrete problem (5.4.2) with NURBS of degrees $p = 2$ and 3, with 2,048 DOFs for both the cases. In Figure 5.15c the condition number $\kappa(A)$ is reported at each time instance, for each NURBS considered for the results of Figure 5.14. The behavior of $\kappa(A)$ is similar

---

[2] The number of DOFs accounts for both a vector valued unknown (the velocity $\mathbf{v}$) and a scalar unknown (the normal velocity v).

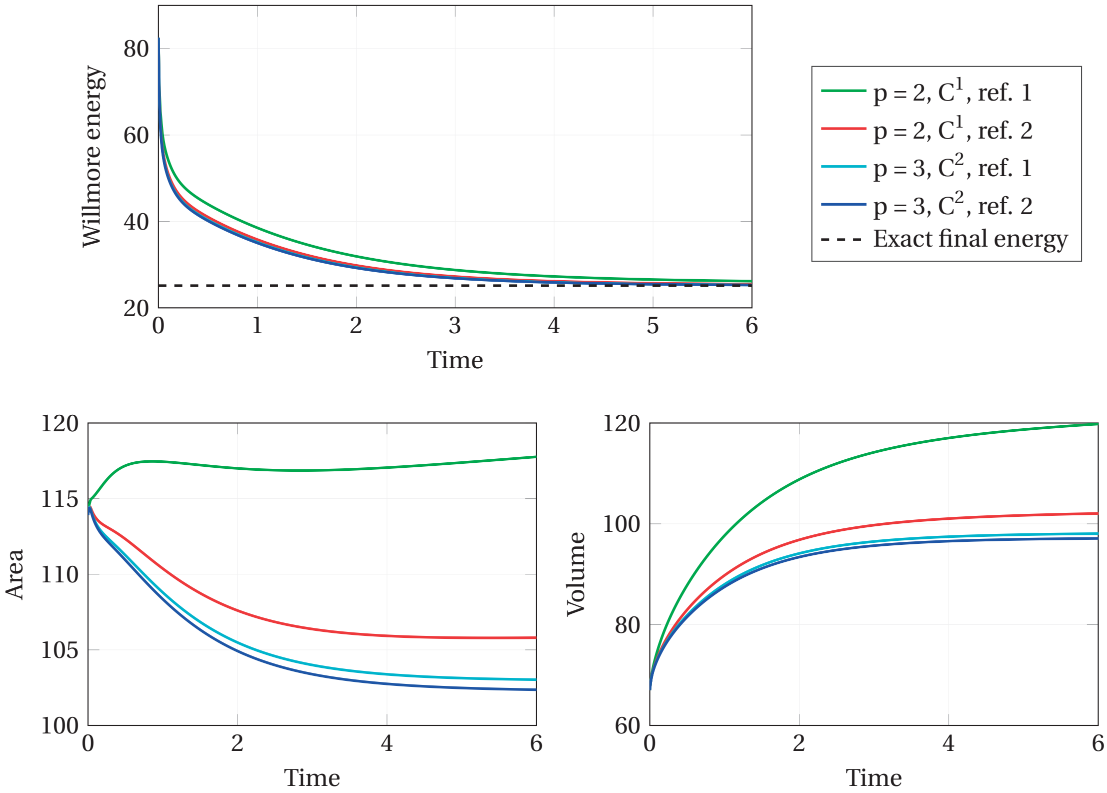
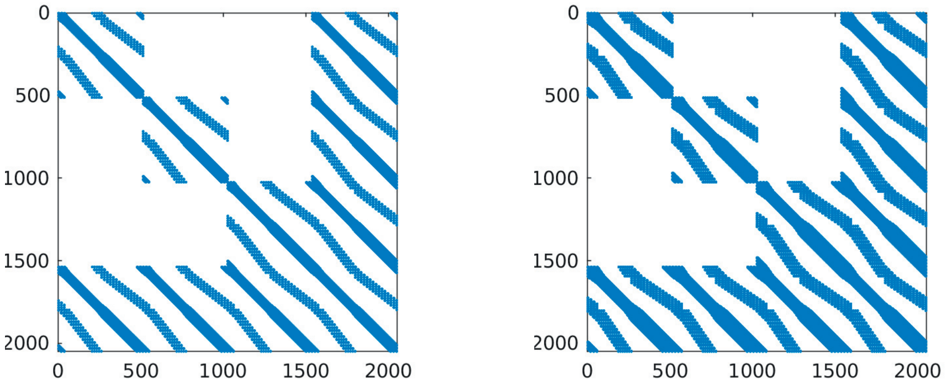


Figure 5.14 – Test 5.2.1. Willmore flow of an ellipsoid with $a_0 = 4$, $b_0 = 4$ and $c_0 = 1$. Evolution of the Willmore energy J_W, area and volume vs. time t (zoom) for NURBS of degrees $p = 2$ and C^1-continuous a.e. with 684 and 2,380 elements, yielding 2,048 and 8,192 DOFs, respectively, and $p = 3$ and C^2-continuous a.e. with 779 and 2,555 elements, again yielding 2,048 and 8,192 DOFs, respectively.

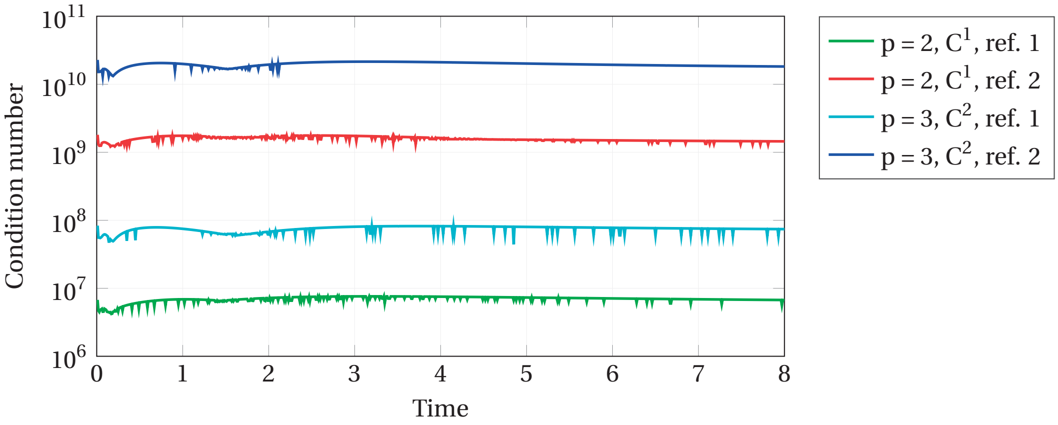
to what experienced for Test 5.1.1, in the sense that, the higher degree of the NURBS basis functions and the finer the mesh, the higher the condition number. With respect to Test 5.1.1, the condition number is generally higher, due to the high order of derivatives involved in the Willmore flow problem with respect to the mean curvature problem.

Test 5.2.2. We consider the numerical approximation of the Willmore flow of a torus, described by the relation in Eq. (5.3.8). In particular, Clifford tori, which are characterized by a ratio between the outer R_0 and inner r_0 radii equal to $R_0/r_0 = \sqrt{2}$, are stationary geometries for the Willmore flow, with Willmore energy J_W equal to $4\pi^2$; tori with different aspect ratios tend to converge to Clifford tori. We numerically simulate the Willmore flow of a initial torus Ω_0 with $R_0 = 1$ and $r_0 = 0.2$ (i.e. for which $R_0/r_0 = 5$), represented as a NURBS surface with basis functions of degrees $p = 2$ and 3, being globally C^1- and C^2-continuous, and two h-refinement levels. Solutions at different time steps are reported in Figure 5.16. Time discretization is based on the BDF scheme of order $k = 2$ and time step size $\Delta t = 0.001$. By employing NURBS basis functions of degree $p = 2$ and globally C^1-continuous, we consider a NURBS mesh built of 836 elements, yielding 2,048 DOFs, and a mesh of 2,660 elements, yielding 8,192 DOFs.



(a) $p = 2$, C^1-continuous a.e.

(b) $p = 3$, C^2-continuous a.e.



(c) Evolution of the condition number vs. time t

Figure 5.15 – Test 5.2.1. Willmore flow of an ellipsoid with $a_0 = 4$, $b_0 = 4$ and $c_0 = 1$. Sparsity patterns in (a) and (b) and evolution of the condition number $\kappa(A)$ of the matrix associated to the full discrete problem (5.4.2) vs. time t ((c)), using NURBS basis functions of degrees $p = 2$ and 3, C^1- and C^2-continuous a.e., respectively, and two refinement levels yielding 2,048 and 8,192 DOFs, respectively, both for $p = 2$ and $p = 3$.

Additionally, by employing NURBS basis functions of degree $p = 3$ and globally C^2-continuous, we consider a mesh built of 1,025 elements, yielding 2,880 DOFs, and a mesh of 2,993 elements, yielding 9,792 DOFs.

In Figure 5.17 the evolution of the Willmore energy is reported, together with the Willmore energy of the Clifford torus (indicated as *Exact final energy*). If we compare the final Willmore energy of the approximated solution with the Willmore energy of the Clifford torus we obtain the following errors with the above mentioned meshes, in order: 0.1146 (0.290%), 0.1130 (0.286%), 0.0114 (0.029%), and 0.0010 (0.003%). Therefore, the best compromise between accuracy and number of DOFs employed is obtained for NURBS basis functions of degree $p = 3$ and globally C^2-continuous, which guarantee a good accuracy even with a small amount

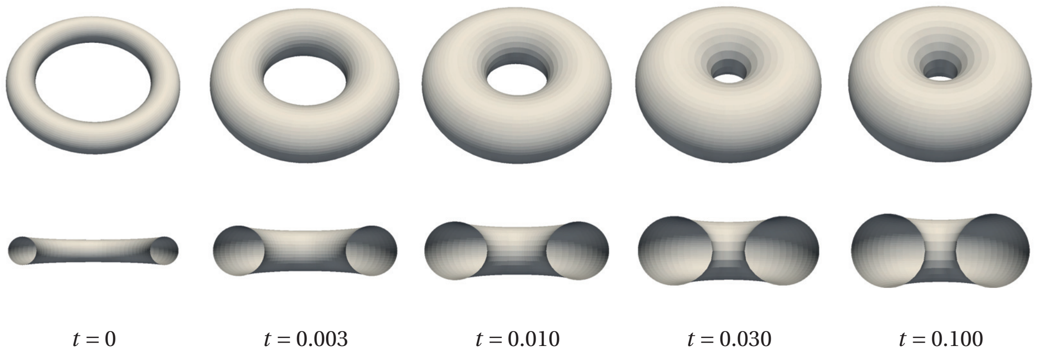


Figure 5.16 – Test 5.2.2. Willmore flow of a torus with $R_0 = 1$ and $r_0 = 0.2$. Solution at different time instances.

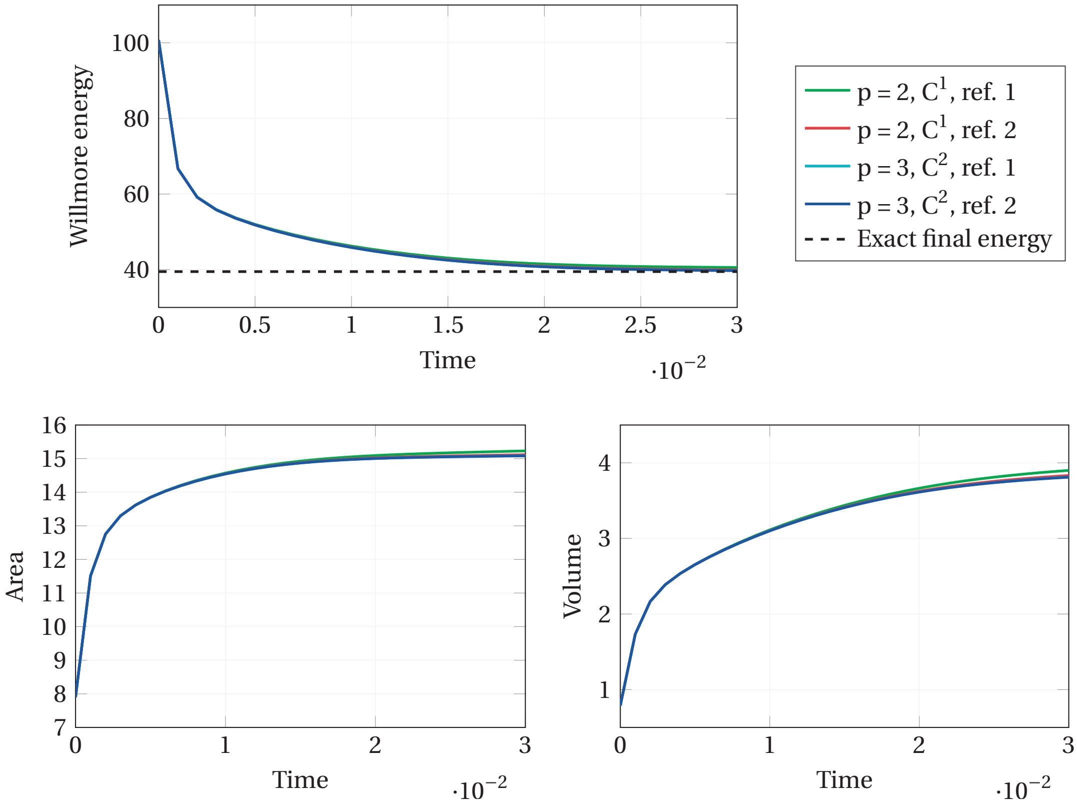
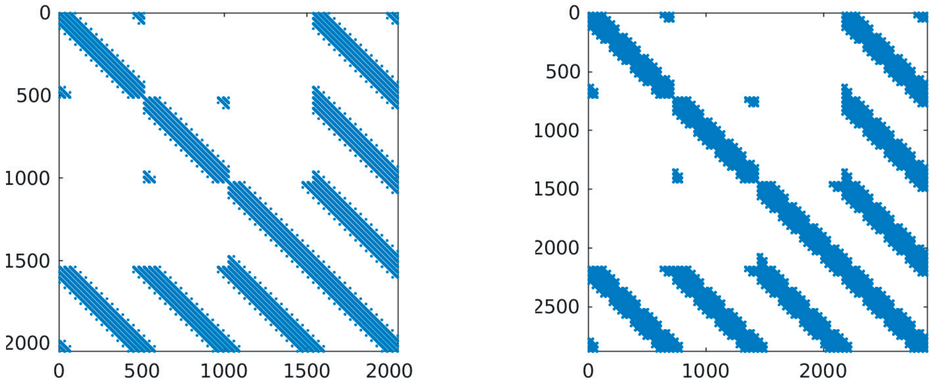
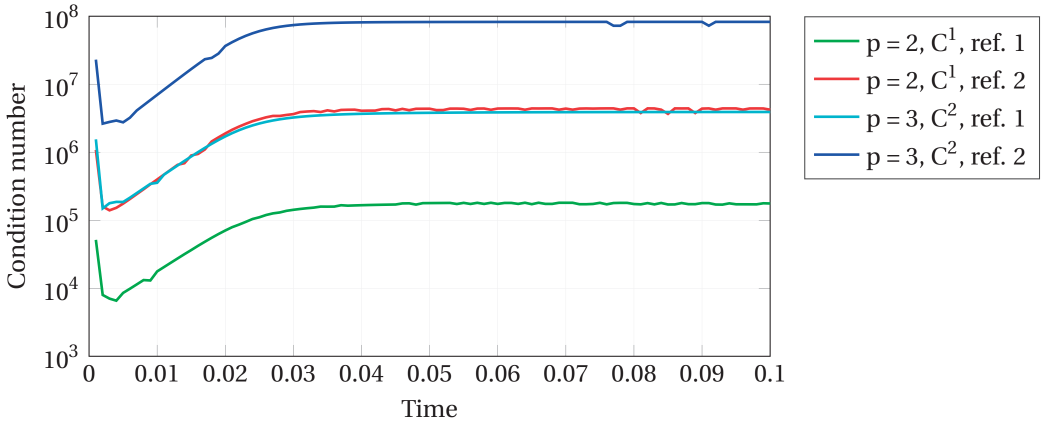


Figure 5.17 – Test 5.2.2. Willmore flow of a torus with $R_0 = 1$ and $r_0 = 0.2$. Evolution of the Willmore energy J_W, area and volume vs. time t (zoom) for meshes of two refinement levels built of NURBS of degrees $p = 2$ (for 836 and 2,660 elements) and $p = 3$ (for 1,025 and 2,993 elements), C^1- and C^2-continuous a.e., respectively.



(a) $p = 2$, C^1-continuous a.e.

(b) $p = 3$, C^2-continuous a.e.



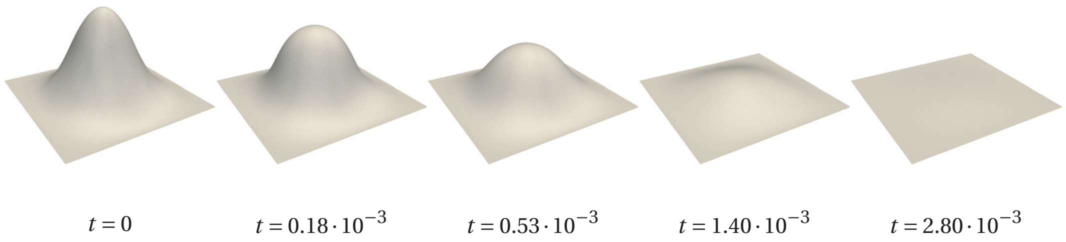
(c) Evolution of the condition number vs. time t

Figure 5.18 – Test 5.2.2. Willmore flow of a torus with $R_0 = 1$ and $r_0 = 0.2$. Sparsity patterns in (a) and (b) and evolution of the condition number $\kappa(A)$ of the matrix associated to the full discrete problem (5.4.2) vs. time t ((c)), using NURBS basis functions of degrees $p = 2$ and 3, C^1- and C^2-continuous a.e., respectively, and two refinement levels yielding 2,048 and 8,192 DOFs for $p = 2$, and 2,880 and 9,792 DOFs for $p = 3$, respectively.

of DOFs.

Finally, we report in Figures 5.18a and 5.18b the sparsity patterns of the matrices associated to the fully discrete problem (5.4.2) with NURBS of degrees $p = 2$ and 3, with 2,048 and 2,880 DOFs, respectively. We report in Figure 5.18c the evolution of the condition number $\kappa(A)$ at each time step, for each NURBS already considered in Figure 5.17. As usual, the condition number increases with the degree p of the NURBS basis functions and with the refinement of the mesh, but, for each discretization, these follow the same overall behavior in time. With respect to Test 5.2.1, the condition number tends to be smaller, since the mapping does not present any singularity.

Test 5.2.3. As a final example, we consider an open surface Ω. We still consider the formulation of Eq. (5.4.2), even if it is tailored to model closed geometries, since it remains valid if we consider a problem on an open surface with homogeneous essential boundary conditions on the unknown velocity and normal velocity; the function spaces employed for the approximation of the velocity and the normal velocity in Eq. (5.4.2) are replaced by $V_h^* := [H_0^2(\Omega_*)]^3 \cap [\mathcal{N}_h]^3$ and $W_h^* := H_0^2(\Omega_*) \cap \mathcal{N}_h$, respectively. We consider the evolution of an initially deformed quadrilateral surface under Willmore flow, with homogeneous essential boundary conditions, which evolves toward a flat plane. In Figure 5.19 we report the computed geometry at different time instances. The NURBS mesh is built of 2,304 elements, yielding 2,116 DOFs, with NURBS basis functions of degree $p = 3$ and globally C^2-continuous; the BDF scheme of order 2 is employed for the time discretization, with time step size $\Delta t = 10^{-5}$. Evolution of the Willmore energy, surface area, and volume are reported in Figure 5.20. Even if designed for closed geometries, the scheme outlined in Section 5.4 performs well also for open surfaces, involving large deformations, without changes to the formulation.



$t = 0$ $t = 0.18 \cdot 10^{-3}$ $t = 0.53 \cdot 10^{-3}$ $t = 1.40 \cdot 10^{-3}$ $t = 2.80 \cdot 10^{-3}$

Figure 5.19 – Test 5.2.3. Willmore flow of a deformed rectangular surface with fixed borders. Solution at different time instances.

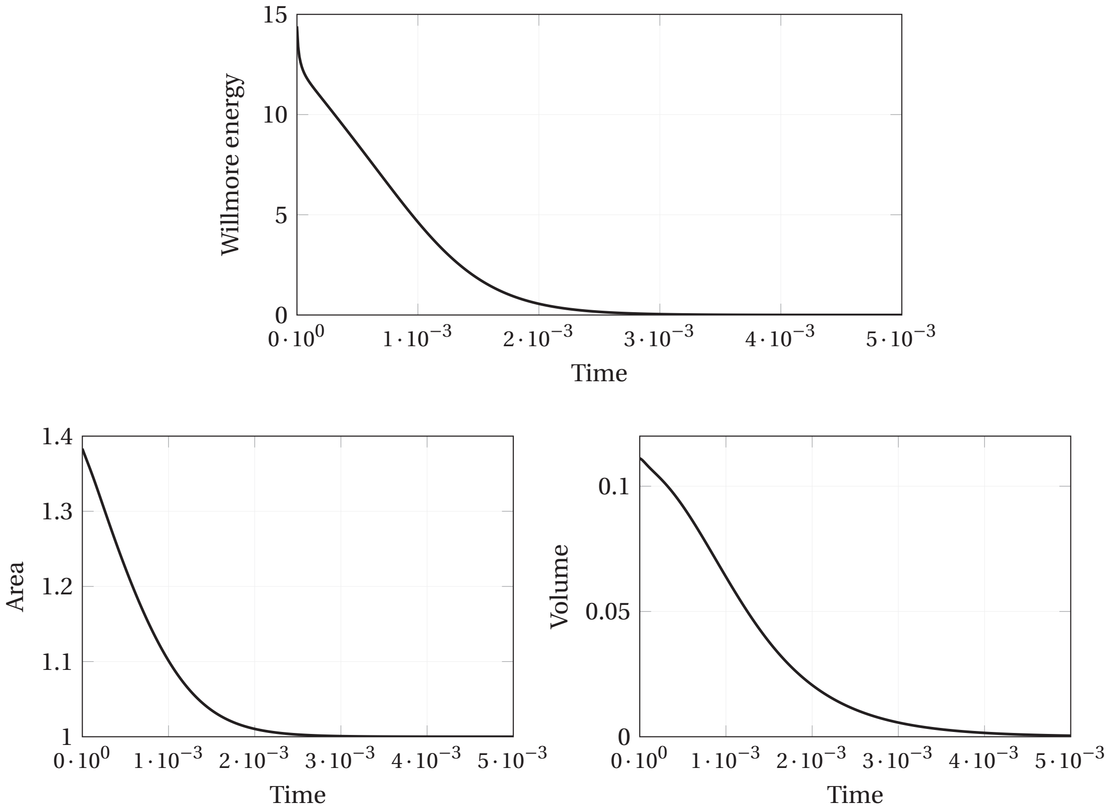


Figure 5.20 – Test 5.2.3. Willmore flow of an open quadrilateral surface. Evolution of the Willmore energy J_W, area and volume vs. time t (zoom) for a mesh of 2,304 elements built of NURBS of degree $p = 2$ and globally C^1-continuous.

6 Equilibrium shapes of lipid vesicles

In this chapter, we apply the framework developed in Chapter 5 for the numerical approximation of geometric PDEs to the problem of finding the shapes of lipid vesicles in equilibrium conditions; the discussion and results that follow are based on our paper [107].

A lipid *vesicle*, as e.g. a red blood cell, is a biomembrane consisting of a lipid bilayer, made of molecules with a hydrophilic head group and two hydrophobic hydrocarbon chains which spontaneously aggregate in closed shapes when set in an aqueous environment [55]. Lipid bilayers are of great interest in biology since they are the fundamental components of the membrane of cells and organelles [56]; a variety of mathematical models describing their shape and dynamic behavior has been proposed in recent years. In general, these can be classified into microscopic discrete molecular based models as for example in [63, 64, 66, 80], multiscale models as in [67, 68, 70], and macroscopic continuum models as e.g. in [26, 27, 49, 52, 72, 82]. A common assumption consists in treating the membranes as surfaces embedded into the 3D space since the combined layer thickness is small compared to the diameter of the vesicle; for example, for red blood cells, the membrane thickness is less than 100 nm, while their diameter is about 8 µm wide [61]. In these mathematical models the bending elasticity (or curvature energy) is the driving factor for the configuration of the vesicles, a conjecture that has been confirmed also by different experiments on isolated vesicles [162].

In Section 6.1, we will introduce two classical and widely used continuum models: the spontaneous curvature model [58, 60] and the bilayer coupling model [163, 164, 165]. We refer the interested reader to [162] for an in-depth analysis of equilibrium shapes obtained with these two models and to [55] for a general discussion about the modeling of fluid membranes and vesicles. These models, based on the minimization of the bending energy subject to geometric contraints, are usually approximated numerically with finite element-based discretizations of the membrane surface, as in [26, 27, 49, 52, 53, 54], or by using phase field approaches, as e.g. in [74, 75, 76].

In this chapter, we consider the numerical approximation of the spontaneous curvature model to seek the equilibrium shapes of lipid vesicles, described in Section 6.1. We consider the biomembranes being represented as single-patch NURBS surfaces and we propose the numerical approximation of their equilibrium shapes by discretizing the equations in space by means of NURBS-based IGA (see Section 6.2.1) and in time by means of BDF schemes (see Section 6.2.2). Handling of the geometric constraints is described in Section 6.2.3. Numerical results follow.

6.1 Mathematical Model

Let us consider a vesicle represented by a compact, connected, oriented and "sufficiently" smooth closed surface $\Omega \subset \mathbb{R}^3$, parametrized by means of a geometrical mapping as of Eq. (1.1.2), as described in Section 1.1. In the classical curvature model, the bending energy of a vesicle depends only on its curvature. In particular, the bending energy related to the vesicle represented by Ω reads [58]:

$$J_B(\Omega) := \frac{1}{2} k_c \int_\Omega H^2 \, d\Omega + k_g \int_\Omega K \, d\Omega, \tag{6.1.1}$$

where k_c and k_g are positive constants representing the bending and the Gaussian rigidities, respectively. These parameters are difficult to obtain experimentally. In general, the bending rigidity can be determined mechanically by applying a force to the vesicle and measuring its response. Alternatively, instead of the mechanical approach, thanks to the membranes' "softness" the reaction of the vesicles to thermal fluctuations can be obtained by methods based on image processing. A brief summary of these approaches, with references therein, is described in an Appendix of [55]. The classical curvature model, however, has no actual physical realization and cannot describe some behaviors of the vesicles observed experimentally, such as budding [55], which are due to the bilayer nature of the membrane.

In the spontaneous-curvature model, initially introduced in [60], the bending energy related to the membrane Ω reads:

$$J_B(\Omega) := \frac{1}{2} k_c \int_\Omega (H - H_0)^2 \, d\Omega + k_g \int_\Omega K \, d\Omega, \tag{6.1.2}$$

which, by recalling the definition of Willmore energy with spontaneous curvature J_{W_0} of Eq. (2.5.7), can be rewritten as:

$$J_B(\Omega) := k_c J_{W_0}(\Omega) + k_g J_G(\Omega). \tag{6.1.3}$$

The term H_0 in Eq. (6.1.2) describes the spontaneous curvature of the vesicle and it accounts for the unbalance of the membrane due to a different chemical environment on the two sides of the vesicle, or to different chemical composition of the two layers [55, 166, 167, 168]. In virtue of the Gauss-Bonnet theorem [109], the second energy term J_G of Eq. (6.1.3) does not depend on the shape of Ω, but it is a topological invariant which only depends on the genus

g [125] of the surface Ω and it is equal to:

$$J_G(\Omega) = 4\pi(1-g). \tag{6.1.4}$$

Although the majority of vesicles have spherical topology, some vesicles of higher genus have been observed experimentally [55]. Nevertheless, as in this thesis we deal with parametric geometries and we do not consider topological changes, the term J_G will be neglected, by setting $k_g = 0$.

Moreover, energetic considerations [55] lead to two common assumptions when modeling lipid membranes. Firstly, the fact that the membrane has a fixed number of lipid molecules and energetically allows bending deformation more easily than stretching or compressing [169,170] leads to the consideration of a constraint on the surface area. Secondly, the biomembrane is impermeable to molecules dispersed in the solution at the time scales of interest, and, for typical vesicles, the curvature energy can balance only small osmotic differences across the biomembrane. Therefore, the vesicle adapts to the difference of osmotic pressure by effectively changing its volume such that the resulting osmotic pressure difference is negligible; then, the volume is kept constant. This behavior can be effectively approximated by introducing a constraint on the vesicle volume.

By referring to the surface area of the vesicle with J_A, as defined in Eq. (2.4.1), and the enclosed volume with J_V, defined as:

$$J_V(\Omega) := \int_\Omega \mathbf{x}\cdot\mathbf{n}_\Omega \, d\Omega, \tag{6.1.5}$$

in order to express the constraints on the area and the volume two Lagrange multipliers are introduced in the energy of Eq. (6.1.3), thus obtaining the so called *Canham–Helfrich* energy J_{CH} [49]:

$$J_{CH}(\Omega,\delta p,\Pi_\Omega) = k_c J_{W_0}(\Omega) + \Pi_\Omega\Big(J_A(\Omega) - J_A(\Omega_0)\Big) + \delta p\Big(J_V(\Omega) - J_V(\Omega_0)\Big), \tag{6.1.6}$$

where Ω_0 is a reference (or initial) surface and Π_Ω and δp are the Lagrange multipliers associated with the area and volume constraints, respectively. More precisely, δp is interpreted as an osmotic pressure jump across the inner and outer sides of the biomembrane, while Π_Ω as the tensile stress required to maintain the inextensibility of the membrane [139,167].

Another assumption that can be made is that the two layers of the biomembrane hardly exchange molecules between each other. This aspect can be taken into account by exploiting the fact that the differences between the densities of the layers and the mean density at the mid-surface depend locally on the mean curvature and, at the leading order, the area difference between the two layers can be approximated in function of the integrated mean curvature [55]. Therefore, if no exchange of molecules between the two layers is assumed, a constraint on the integrated mean curvature can be considered by introducing an energetic penalization term for the deviations from a given area difference. These models are usually called *Area Difference Elasticity* models [54,55]. The *bilayer coupling* model is instead based on enforcing, with a Lagrange multiplier, a constraint on the area difference [163,164], that is the difference:

$$\Delta A = A^{ext} - A^{int} \tag{6.1.7}$$

between the area of the external layer and the area of the internal layer. It can be approximated as:

$$\Delta A \approx 2dM, \tag{6.1.8}$$

where d is the distance between the two layers and M is defined as:

$$M(\Omega) := \frac{1}{2}\int_{\Omega} H \, d\Omega; \tag{6.1.9}$$

the approximation in Eq. (6.1.8) holds up to the order d^2/A [162]. By considering the Willmore energy defined in Eq. (2.5.1), as bending energy and by introducing the Lagrange multiplier q for the integrated mean curvature, the bilayer coupling model then reads [162]:

$$J_{BC}(\Omega) := k_c J_W(\Omega) + \Pi'_{\Omega}\left(J_A(\Omega) - J_A(\Omega_0)\right) + \delta p'\left(J_V(\Omega) - J_V(\Omega_0)\right) + q\,M(\Omega). \tag{6.1.10}$$

However, if the following relation between the Lagrange multipliers is considered:

$$\Pi'_{\Omega} = \Pi_{\Omega} + \frac{k_c}{2}H_0^2, \qquad \delta p' = \delta p, \qquad \text{and} \qquad q = -2k_c H_0, \tag{6.1.11}$$

we see that the spontaneous curvature model of Eq. (6.1.6) and the bilayer coupling model of Eq. (6.1.10) lead to the same equilibrium shapes [165].

With this remark, we now consider the spontaneous curvature model governed by the Canham–Helfrich energy J_{CH} of Eq. (6.1.6) and we focus, in the next sections, on its numerical approximation.

6.1.1 Energy minimization

We study the equilibrium shapes of the lipid biomembranes by minimizing the Canham–Helfrich energy J_{CH} of Eq. (6.1.6). With this aim, we formulate the *Canham–Helfrich flow problem* as the L^2-gradient flow of J_{CH}. Given an initial closed surface $\Omega_0 \subset \mathbb{R}^3$, described by $\mathbf{x}_0$, for a.e. $t \in (0,T)$, find $\mathbf{x} \in V_t$, $\Pi_{\Omega t} \in \mathbb{R}$, and $\delta p_t \in \mathbb{R}$ such that

$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t = -dJ_{CH}\left(\Omega(\mathbf{x}), \Pi_{\Omega t}, \delta p_t; \boldsymbol{\varphi}\right) & \forall \boldsymbol{\varphi} \in V_t, \\ J_A(\Omega_t) = J_A(\Omega_0), \qquad J_V(\Omega_t) = J_V(\Omega_0), & \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0, \end{cases} \tag{6.1.12}$$

where $V_t := [H^2(\Omega_t)]^3$ and dJ_{CH} is the shape derivative of J_{CH} at Ω_t along $\boldsymbol{\varphi} \in V_t$, reading [139]:

$$dJ_{CH}(\Omega, \Pi_\Omega, \delta p; \boldsymbol{\varphi}) = k_c\, dJ_{W_0}(\Omega; \boldsymbol{\varphi}) + \Pi_\Omega\, dJ_A(\Omega; \boldsymbol{\varphi}) + \delta p\, dJ_V(\Omega; \boldsymbol{\varphi}), \tag{6.1.13}$$

where the shape derivatives of J_V and J_A along $\boldsymbol{\varphi}$ are given by:

$$dJ_A(\Omega; \boldsymbol{\varphi}) = \int_\Omega H \boldsymbol{\varphi} \cdot \mathbf{n}_\Omega \, d\Omega \tag{6.1.14}$$

(as in Eq. (2.4.2)) and

$$dJ_V(\Omega; \boldsymbol{\varphi}) = \int_\Omega \boldsymbol{\varphi} \cdot \mathbf{n}_\Omega \, d\Omega, \tag{6.1.15}$$

respectively. By considering a constant spontaneous curvature, i.e. $H_0 \in \mathbb{R}$, we rewrite also the shape derivative of the energy J_{W_0} from Eq. (2.5.9) as:

$$dJ_{W_0}(\Omega; \boldsymbol{\varphi}) = -\int_\Omega \left[\Delta_\Omega H + H\left(\frac{1}{2}H^2 - \frac{1}{2}H_0^2 - 2K\right) - 2H_0 K\right] \boldsymbol{\varphi} \cdot \mathbf{n}_\Omega \, d\Omega. \tag{6.1.16}$$

By using Eqs. (6.1.14), (6.1.15), and (6.1.16) in Eq. (6.1.13) we finally obtain:

$$dJ_{CH}(\Omega, \Pi_\Omega, \delta p; \boldsymbol{\varphi}) = \int_\Omega \left\{k_c \left[-\Delta_\Omega H - H\left(\frac{1}{2}H^2 - \frac{1}{2}H_0^2 - 2K\right) + 2H_0 K\right] + \Pi_\Omega H + \delta p\right\} \boldsymbol{\varphi} \cdot \mathbf{n}_\Omega \, d\Omega. \tag{6.1.17}$$

In particular, the Canham–Helfrich flow problem in strong form reads:

$$\text{for a.e. } t \in (0, T),\ \text{find } \Omega_t \subset \mathbb{R}^3,\ \Pi_{\Omega t} \in \mathbb{R}, \text{ and } \delta p_t \in \mathbb{R} \text{ such that}$$
$$\begin{cases} \dot{\mathbf{x}} = -\left\{k_c \left[-\Delta_{\Omega_t} H - H\left(\frac{1}{2}H^2 - \frac{1}{2}H_0^2 - 2K\right) + 2H_0 K\right] + \Pi_{\Omega t} H + \delta p_t\right\} \mathbf{n}_{\Omega_t} & \text{in } \Omega_t, \\ J_A(\Omega_t) = J_A(\Omega_0), \qquad J_V(\Omega_t) = J_V(\Omega_0), & \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0. \end{cases} \tag{6.1.18}$$

The Canham–Helfrich flow represents an extension of the Willmore flow problem, already discussed in Section 5.4, with the addition of the area and volume constraints, as well as the spontaneous curvature terms. We then apply the same splitting introduced in Eq. (5.4.1), for which we treat the velocity $\mathbf{v} = \dot{\mathbf{x}}$ and the normal velocity v as unknowns. Therefore, we consider the following weak formulation:

for a.e. $t \in (0,T)$, find $\mathbf{x} \in V_t$, $v \in W_t$, $\Pi_{\Omega t} \in \mathbb{R}$, and $\delta p_t \in \mathbb{R}$ such that

$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}} \cdot \boldsymbol{\varphi}\, d\Omega_t - \int_{\Omega_t} v\, \mathbf{n}_{\Omega_t} \cdot \boldsymbol{\varphi}\, d\Omega_t = 0 & \forall \boldsymbol{\varphi} \in V_t, \\ \displaystyle\int_{\Omega_t} v\psi\, d\Omega_t + k_c \int_{\Omega_t} (\Delta_{\Omega_t}\mathbf{x}\cdot\mathbf{n}_{\Omega_t})\Delta_{\Omega_t}\psi\, d\Omega_t - 2k_c \int_{\Omega_t} H_0 K \psi\, d\Omega_t & \\ \qquad\qquad \displaystyle + k_c \int_{\Omega_t} (\Delta_{\Omega_t}\mathbf{x}\cdot\mathbf{n}_{\Omega_t}) \left(\frac{1}{2}H^2 - \frac{1}{2}H_0^2 - 2K\right)\psi\, d\Omega_t & \\ \qquad\qquad \displaystyle + \Pi_{\Omega t} \int_{\Omega_t} H\psi\, d\Omega_t + \delta p_t \int_{\Omega_t} \psi\, d\Omega_t = 0 & \forall \psi \in W_t, \\ \displaystyle\int_{\Omega_t} H\mathbf{x}\cdot\mathbf{n}_{\Omega_t}\, d\Omega_t = A_0, \qquad \int_{\Omega_t} \mathbf{x}\cdot\mathbf{n}_{\Omega_t}\, d\Omega_t = V_0, & \\ \mathbf{x}(0) = \mathbf{x}_0, & \end{cases} \tag{6.1.19}$$

with initial area $A_0 = J_A(\Omega_0)$ and volume $V_0 = J_V(\Omega_0)$. In particular, we consider, for any given $t \in (0,T)$, the function spaces $V_t = \left[H^2(\Omega_t)\right]^3$ and $W_t = H^2(\Omega_t)$, since the formulation of Eq. (6.1.19) involves second order surface differential operators applied to the trial and test functions.

6.2 Numerical Approximation

In this section, we consider the numerical approximation of the Canham–Helfrich flow problem (6.1.19). We introduce both the space and time discretizations and we compare two numerical approaches to enforce the area and volume constraints through Lagrange multipliers.

6.2.1 Space discretization

As for the geometric PDEs approximated in Chapter 5, for the spatial discretization of problem (6.1.19) we consider the Galerkin method using NURBS-based IGA subspaces. The biomembrane evolution is represented by the family of surfaces $\{\Omega_t\}_{t\in(0,T)}$ described by single patch NURBS mappings as in Eq. (3.2.7). We follow the procedure outlined in Section 5.1. The NURBS function spaces $\widehat{\mathcal{N}_h}$ and $\mathcal{N}_h$, defined over the parametric domain $\widehat{\Omega}$ and the physical domain Ω, respectively, and introduced in Chapter 3 are considered; since the biomembranes are represented as closed surfaces, the chosen NURBS spaces are periodic and exhibit high order global continuity of the basis functions. In an isoparametric approach, these spaces describe the geometries as well as the trial and test functions. In particular, for any $t \in (0,T)$, we choose $V_{t,h} := V_t \cap [\mathcal{N}_h]^3$ and $W_{t,h} := W_t \cap \mathcal{N}_h$ as trial and test function spaces, with V_t and W_t defined in Section 6.1.1. Hence, by considering the formulation of Eq. (6.1.19), the semi-discrete problem reads:

for a.e. $t \in (0,T)$ find $\mathbf{x}_h \in V_{t,h}$, $v \in W_{t,h}$, $\Pi_{\Omega t} \in \mathbb{R}$, and $\delta p_t \in \mathbb{R}$ such that

$$\begin{cases} \displaystyle\int_{\Omega_t} \dot{\mathbf{x}}_h \cdot \boldsymbol{\varphi}_h \, d\Omega_t - \int_{\Omega_t} v_h \mathbf{n}_{\Omega_t} \cdot \boldsymbol{\varphi}_h \, d\Omega_t = 0 & \forall \boldsymbol{\varphi}_h \in V_{t,h}, \\ \displaystyle\int_{\Omega_t} v_h \psi_h \, d\Omega_t + k_c \int_{\Omega_t} (\Delta_{\Omega_t}\mathbf{x}_h \cdot \mathbf{n}_{\Omega_t}) \Delta_{\Omega_t} \psi_h \, d\Omega_t - 2k_c \int_{\Omega_t} H_0 K_h \psi_h \, d\Omega_t & \\ \displaystyle\qquad + k_c \int_{\Omega_t} (\Delta_{\Omega_t}\mathbf{x}_h \cdot \mathbf{n}_{\Omega_t}) \left(\frac{1}{2} H_h^2 - \frac{1}{2} H_0^2 - 2K_h \right) \psi_h \, d\Omega_t & \\ \displaystyle\qquad + \Pi_{\Omega t} \int_{\Omega_t} H_h \psi_h \, d\Omega_t + \delta p_t \int_{\Omega_t} \psi_h \, d\Omega_t = 0 & \forall \psi_h \in W_{t,h}, \\ \displaystyle\int_{\Omega_t} H_h \mathbf{x}_h \cdot \mathbf{n}_{\Omega_t} \, d\Omega_t = A_0, \qquad \int_{\Omega_t} \mathbf{x}_h \cdot \mathbf{n}_{\Omega_t} \, d\Omega_t = V_0, & \\ \mathbf{x}_h(0) = \mathbf{x}_{0,h}. & \end{cases} \tag{6.2.1}$$

We remark that, as problem (6.2.1) derives from the Willmore flow problem and involves second order surface differential operators, we require the function spaces $V_{t,h}$ and $W_{t,h}$ to host basis functions which are at least C^1-continuous a.e. on Ω_t.

6.2.2 Time discretization

As done in Chapter 5, for discretizing in time problem (6.2.1) we consider the k-th order BDF schemes with extrapolation of the geometry, as described in Section 5.2. In particular, by using the same notation, the fully discrete Canham–Helfrich problem reads:

for $n = k, \dots, N-1$ find $\mathbf{v}_h^{n+1} \in V_h^*$, $v_h^{n+1} \in W_h^*$, $\Pi_\Omega^{n+1} \in \mathbb{R}$, and $\delta p^{n+1} \in \mathbb{R}$ such that

$$\begin{cases} \displaystyle\int_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \boldsymbol{\varphi}_h \, d\Omega_* - \int_{\Omega_*} v_h^{n+1} \mathbf{n}_{\Omega_*} \cdot \boldsymbol{\varphi}_h \, d\Omega_* = 0 & \forall \boldsymbol{\varphi}_h \in V_h^*, \\ \displaystyle\int_{\Omega_*} v_h^{n+1} \psi_h \, d\Omega_* + k_c \frac{\Delta t}{\alpha_0} \int_{\Omega_*} (\Delta_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \mathbf{n}_{\Omega_*}) \Delta_{\Omega_*} \psi_h \, d\Omega_* & \\ \displaystyle\qquad + k_c \frac{\Delta t}{\alpha_0} \int_{\Omega_*} \left[\frac{1}{2} (H_h^*)^2 - \frac{1}{2} H_0^2 - 2K_h^* \right] (\Delta_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \mathbf{n}_{\Omega_*}) \psi_h \, d\Omega_* & \\ \displaystyle\qquad + \Pi_\Omega^{n+1} \int_{\Omega_*} H_h^* \psi_h \, d\Omega_* + \delta p^{n+1} \int_{\Omega_*} \psi_h \, d\Omega_* & \\ \displaystyle\qquad = 2k_c \int_{\Omega_*} H_0 K_h^* \psi_h \, d\Omega_* - k_c \int_{\Omega_*} \left(\Delta_{\Omega_*} \mathbf{x}_h^{bdf,n} \cdot \mathbf{n}_{\Omega_*} \right) \Delta_{\Omega_*} \psi_h \, d\Omega_* & \\ \displaystyle\qquad - k_c \int_{\Omega_*} \left[\frac{1}{2} (H_h^*)^2 - \frac{1}{2} H_0^2 - 2K_h^* \right] \left(\Delta_{\Omega_*} \mathbf{x}_h^{bdf,n} \cdot \mathbf{n}_{\Omega_*} \right) \psi_h \, d\Omega_* & \forall \psi_h \in W_h^*, \\ \displaystyle\int_{\Omega_{n+1}} H_h^{n+1} \mathbf{x}_h^{n+1} \cdot \mathbf{n}_{\Omega_{n+1}} \, d\Omega_{n+1} = A_0, \qquad \int_{\Omega_{n+1}} \mathbf{x}_h^{n+1} \cdot \mathbf{n}_{\Omega_{n+1}} \, d\Omega_{n+1} = V_0, & \\ \mathbf{x}_h^{bdf,0} = \mathbf{x}_{0,h}^{bdf,0}, & \end{cases} \tag{6.2.2}$$

where V_h^* and W_h^* correspond to the function spaces $V_{t,h}$ and $W_{t,h}$ built on Ω_*, respectively. Problem (6.2.2) is still nonlinear since we are enforcing the constraints on the unknown surface Ω_{n+1}. In Section 6.2.3 we will discuss how to recover a linear (semi-implicit) formulation of the problem by appropriate handling of the constraints.

6.2.3 Enforcement of the area and volume constraints

To enforce the area and volume constraints of problem (6.2.2) we propose two approaches. The first one is an adaptation of the iterative scheme proposed in [49, 52] to our context based on IGA and BDF discretizations, which enforces the area and volume constraints potentially to machine precision. Then, we propose a second approach based on the approximation of the constraints; although being not exact, it is however more convenient from a computational point of view.

Constraints enforcement: scheme C-1

We assume for the time being that the Lagrange multipliers $\widetilde{\Pi}_\Omega^{n+1}$ and $\delta\widetilde{p}^{n+1}$ are given. Then, we reformulate problem (6.2.2) as follows:

find, for $n=0,\dots,N-1$, $\mathbf{v}_h^{n+1}\in V_h^*$ and $v_h^{n+1}\in W_h^*$ such that

$$\left\{\begin{aligned}
&\int_{\Omega_*}\mathbf{v}_h^{n+1}\cdot\boldsymbol{\varphi}_h\,d\Omega_*-\int_{\Omega_*}v_h^{n+1}\mathbf{n}_{\Omega_*}\cdot\boldsymbol{\varphi}_h\,d\Omega_*=0 &&\forall\boldsymbol{\varphi}_h\in V_h^*,\\
&\int_{\Omega_*}v_h^{n+1}\psi_h\,d\Omega_*+k_c\frac{\Delta t}{\alpha_0}\int_{\Omega_*}(\Delta_{\Omega_*}\mathbf{v}_h^{n+1}\cdot\mathbf{n}_{\Omega_*})\Delta_{\Omega_*}\psi_h\,d\Omega_*\\
&\qquad+k_c\frac{\Delta t}{\alpha_0}\int_{\Omega_*}\left[\frac{1}{2}(H_h^*)^2-\frac{1}{2}H_0^2-2K_h^*\right](\Delta_{\Omega_*}\mathbf{v}_h^{n+1}\cdot\mathbf{n}_{\Omega_*})\psi_h\,d\Omega_*\\
&\qquad=2k_c\int_{\Omega_*}H_0K_h^*\psi_h\,d\Omega_*-k_c\int_{\Omega_*}\left(\Delta_{\Omega_*}\mathbf{x}_h^{bdf,n}\cdot\mathbf{n}_{\Omega_*}\right)\Delta_{\Omega_*}\psi_h\,d\Omega_*\\
&\qquad-k_c\int_{\Omega_*}\left[\frac{1}{2}(H_h^*)^2-\frac{1}{2}H_0^2-2K_h^*\right]\left(\Delta_{\Omega_*}\mathbf{x}_h^{bdf,n}\cdot\mathbf{n}_{\Omega_*}\right)\psi_h\,d\Omega_*\\
&\qquad-\widetilde{\Pi}_\Omega^{n+1}\int_{\Omega_*}H_h^*\psi_h\,d\Omega_*-\delta\widetilde{p}^{n+1}\int_{\Omega_*}\psi_h\,d\Omega_*, &&\forall\psi_h\in W_h^*,
\end{aligned}\right.\tag{6.2.3}$$

with appropriate initial condition $\mathbf{x}_h^{bdf,0}=\mathbf{x}_{0,h}^{bdf,0}$. In compact form, system (6.2.3) reads, for $n=0,\dots,N-1$:

$$\begin{aligned}\mathscr{L}(\mathbf{v}_h^{n+1},v_h^{n+1};\boldsymbol{\varphi}_h,\psi_h)=\mathscr{F}_W(\boldsymbol{\varphi}_h,\psi_h)+\widetilde{\Pi}_\Omega^{n+1}\mathscr{F}_A(\boldsymbol{\varphi}_h,\psi_h)+\delta\widetilde{p}^{n+1}\mathscr{F}_V(\boldsymbol{\varphi}_h,\psi_h),&\\
\forall\boldsymbol{\varphi}_h\in V_h^*,\ \forall\psi_h\in W_h^*,&\end{aligned}\tag{6.2.4}$$

with obvious choice of notation. Because of the linearity of $\mathscr{L}$ with respect to $\mathbf{v}_h^{n+1}$ and v_h^{n+1}, thanks to the superposition of effects we can write:

$$\mathbf{v}_h^{n+1}=\mathbf{v}_{h,W}^{n+1}+\widetilde{\Pi}_\Omega^{n+1}\mathbf{v}_{h,A}^{n+1}+\delta\widetilde{p}^{n+1}\mathbf{v}_{h,V}^{n+1}\tag{6.2.5}$$

and

$$v_h^{n+1}=v_{h,W}^{n+1}+\widetilde{\Pi}_\Omega^{n+1}v_{h,A}^{n+1}+\delta\widetilde{p}^{n+1}v_{h,V}^{n+1},\tag{6.2.6}$$

where $\mathbf{v}_{h,W}^{n+1}$, $v_{h,W}^{n+1}$, $\mathbf{v}_{h,A}^{n+1}$, $v_{h,A}^{n+1}$, $\mathbf{v}_{h,V}^{n+1}$, and $v_{h,V}^{n+1}$ satisfy the following (independent) problems:

$$\begin{aligned}
&\mathscr{L}(\mathbf{v}_{h,W}^{n+1},v_{h,W}^{n+1};\boldsymbol{\varphi}_h,\psi_h)=\mathscr{F}_W(\boldsymbol{\varphi}_h,\psi_h),\\
&\mathscr{L}(\mathbf{v}_{h,A}^{n+1},v_{h,A}^{n+1};\boldsymbol{\varphi}_h,\psi_h)=\mathscr{F}_A(\boldsymbol{\varphi}_h,\psi_h), \qquad \forall\boldsymbol{\varphi}_h\in V_h^*,\ \forall\psi_h\in W_h^*\\
&\mathscr{L}(\mathbf{v}_{h,V}^{n+1},v_{h,V}^{n+1};\boldsymbol{\varphi}_h,\psi_h)=\mathscr{F}_V(\boldsymbol{\varphi}_h,\psi_h).
\end{aligned}\tag{6.2.7}$$

At this stage, after solving Eqs. (6.2.7) and using Eq. (6.2.5), one needs to recover the values of the (unknown) Lagrange multipliers Π_Ω^{n+1} and δp^{n+1}. With this aim, we enforce the area and volume constraints by looking for the zeros of the vector-valued function $\mathbf{f}_c^n:\mathbb{R}^2\to\mathbb{R}^2$ defined as:

$$\mathbf{f}_c^n(\widetilde{\Pi}_\Omega, \delta\widetilde{p}) := \begin{bmatrix} J_A(\widetilde{\Omega}_{n+1}(\widetilde{\Pi}_\Omega, \delta\widetilde{p})) - J_A(\Omega_n) \\ J_V(\widetilde{\Omega}_{n+1}(\widetilde{\Pi}_\Omega, \delta\widetilde{p})) - J_V(\Omega_n) \end{bmatrix} = \mathbf{0}, \tag{6.2.8}$$

where $\widetilde{\Omega}_{n+1}(\widetilde{\Pi}_\Omega, \delta\widetilde{p})$ is the surface defined by the mapping:

$$\widetilde{\mathbf{x}}_h^{n+1}(\widetilde{\Pi}_\Omega, \delta\widetilde{p}) = \mathbf{x}_h^{bdf,n} + \frac{\Delta t}{\alpha_0}\left(\mathbf{v}_{h,W}^{n+1} + \widetilde{\Pi}_\Omega \mathbf{v}_{h,A}^{n+1} + \delta\widetilde{p}\,\mathbf{v}_{h,V}^{n+1}\right), \tag{6.2.9}$$

dependent on the general Lagrange multipliers $\widetilde{\Pi}_\Omega$ and $\delta\widetilde{p}$. Then, the zeros of $\mathbf{f}_c$ are approximated by using a quasi-Newton method. We follow the method described in [49]; by indicating with $k \in \mathbb{N}$ the iteration index, the following algorithm is considered, for all $n = 0, \ldots, N-1$:

1. Set $k = 0$ and initialize $\widetilde{\Pi}_\Omega^{n+1,0}$ and $\delta\widetilde{p}^{n+1,0}$ as the solutions of the following problem:

$$\begin{aligned} &\begin{bmatrix} \displaystyle\int_{\Omega_*} \nabla_{\Omega_*} \cdot \mathbf{v}_{h,A}^{n+1}\, d\Omega_* & \displaystyle\int_{\Omega_*} \nabla_{\Omega_*} \cdot \mathbf{v}_{h,V}^{n+1}\, d\Omega_* \\ \displaystyle\int_{\Omega_*} \mathbf{n}_{\Omega_*} \cdot \mathbf{v}_{h,A}^{n+1}\, d\Omega_* & \displaystyle\int_{\Omega_*} \mathbf{n}_{\Omega_*} \cdot \mathbf{v}_{h,V}^{n+1}\, d\Omega_* \end{bmatrix} \begin{bmatrix} \widetilde{\Pi}_\Omega^{n+1,0} \\ \delta\widetilde{p}^{n+1,0} \end{bmatrix} = \\ &\qquad\qquad = \begin{bmatrix} \displaystyle -\int_{\Omega_*} \nabla_{\Omega_*} \cdot \mathbf{v}_{h,W}^{n+1}\, d\Omega_* \\ \displaystyle -\int_{\Omega_*} \mathbf{n}_{\Omega_*} \cdot \mathbf{v}_{h,W}^{n+1}\, d\Omega_* \end{bmatrix}. \end{aligned} \tag{6.2.10}$$

2. Build the surface $\widetilde{\Omega}_{n+1}^k$, defined by the mapping:

$$\widetilde{\mathbf{x}}_h^{n+1,k} = \mathbf{x}_h^{bdf,n} + \frac{\Delta t}{\alpha_0}\left(\mathbf{v}_{h,W}^{n+1} + \widetilde{\Pi}_\Omega^{n+1,k} \mathbf{v}_{h,A}^{n+1} + \delta\widetilde{p}^{n+1,k}\mathbf{v}_{h,V}^{n+1}\right). \tag{6.2.11}$$

3. Check if the “guess” surface $\widetilde{\Omega}_{n+1}^k$ is sufficiently accurate, either by the stopping criterium based on the absolute area and volume conservation, as:

$$|J_A(\widetilde{\Omega}_{n+1}^k) - J_A(\Omega_n)| \leq \tau_A^A \qquad \text{and} \qquad |J_V(\widetilde{\Omega}_{n+1}^k) - J_V(\Omega_n)| \leq \tau_V^A, \tag{6.2.12}$$

   respectively, or the criterium based on the relative area and volume, as:

$$\frac{|J_A(\widetilde{\Omega}_{n+1}^k) - J_A(\Omega_n)|}{J_A(\Omega_n)} \leq \tau_A^R \qquad \text{and} \qquad \frac{|J_V(\widetilde{\Omega}_{n+1}^k) - J_V(\Omega_n)|}{J_V(\Omega_n)} \leq \tau_V^R, \tag{6.2.13}$$

   respectively, where τ_A^A, τ_V^A, τ_A^R, and $\tau_V^R \in \mathbb{R}$ are suitable tolerances. If the stopping criteria (6.2.12) or (6.2.13) are fulfilled, then stop the iterations, set $\Pi_\Omega^{n+1} = \widetilde{\Pi}_\Omega^{n+1,k}$ and $\delta p^{n+1} = \delta\widetilde{p}^{n+1,k}$, for which $\Omega_{n+1} = \widetilde{\Omega}_{n+1}^k$, and proceed to the following time step. Otherwise, continue to point 4.

4. Evaluate the Jacobian of $\mathbf{f}_c^n$ at step k as follows:

$$\mathrm{D}\mathbf{f}_c^{n,k}(\widetilde{\Pi}_\Omega^{n+1,k}, \delta\widetilde{p}^{n+1,k}) = \frac{\Delta t}{\alpha_0}\begin{bmatrix} \displaystyle\int_{\widetilde{\Omega}_{n+1}^k} \nabla_{\widetilde{\Omega}_{n+1}^k} \cdot \mathbf{v}_{h,A}^{n+1}\, d\widetilde{\Omega}_{n+1}^k & \displaystyle\int_{\widetilde{\Omega}_{n+1}^k} \nabla_{\widetilde{\Omega}_{n+1}^k} \cdot \mathbf{v}_{h,V}^{n+1}\, d\widetilde{\Omega}_{n+1}^k \\ \displaystyle\int_{\widetilde{\Omega}_{n+1}^k} \mathbf{n}_{\widetilde{\Omega}_{n+1}^k} \cdot \mathbf{v}_{h,A}^{n+1}\, d\widetilde{\Omega}_{n+1}^k & \displaystyle\int_{\widetilde{\Omega}_{n+1}^k} \mathbf{n}_{\widetilde{\Omega}_{n+1}^k} \cdot \mathbf{v}_{h,V}^{n+1}\, d\widetilde{\Omega}_{n+1}^k \end{bmatrix}. \tag{6.2.14}$$

5. Solve the linear system:

$$\mathrm{D}\mathbf{f}_c^{n,k}(\widetilde{\Pi}_\Omega^{n+1,k}, \delta\widetilde{p}^{n+1,k}) \begin{bmatrix} \Delta\widetilde{\Pi}_\Omega^{n+1,k+1} \\ \Delta\delta\widetilde{p}^{n+1,k+1} \end{bmatrix} = \mathbf{f}_c^n(\widetilde{\Pi}_\Omega^{n+1,k}, \delta\widetilde{p}^{n+1,k}) \tag{6.2.15}$$

and update the Lagrangian multipliers as:

$$\begin{bmatrix} \widetilde{\Pi}_\Omega^{n+1,k+1} \\ \delta\widetilde{p}^{n+1,k+1} \end{bmatrix} = \begin{bmatrix} \widetilde{\Pi}_\Omega^{n+1,k} \\ \delta\widetilde{p}^{n+1,k} \end{bmatrix} - \rho \begin{bmatrix} \Delta\widetilde{\Pi}_\Omega^{n+1,k+1} \\ \Delta\delta\widetilde{p}^{n+1,k+1} \end{bmatrix}, \tag{6.2.16}$$

where $\rho \in \mathbb{R}$ is a relaxation parameter, which in this work we consider to be $\rho = 1$. Then, set $k = k+1$ and continue from point 2 until convergence.

With this iterative method, we obtain the Lagrange multipliers Π_Ω^{n+1} and δp^{n+1} fulfilling the area and volume constraints by the surface Ω_{n+1} up to chosen tolerances. From now on, we will refer to this approach as *scheme C-1*.

Constraints enforcement: scheme C-2

To avoid solving a nonlinear problem at each time step, we can impose the fulfillment of the area and volume constraints on an approximate surface Ω_* obtained by extrapolation. In particular, for all $n = 0, \ldots, N-1$, we force the identity map $\mathbf{x}_h^{n+1}$ of the surface Ω_{n+1} to fulfill the following relations:

$$\int_{\Omega_*} H_h^* \mathbf{x}_h^{n+1} \cdot \mathbf{n}_{\Omega_*}\, d\Omega_* = A_0 \quad \text{and} \quad \int_{\Omega_*} \mathbf{x}_h^{n+1} \cdot \mathbf{n}_{\Omega_*}\, d\Omega_* = V_0. \tag{6.2.17}$$

By considering the area constraint and by using Eq. (5.2.9) we write:

$$\int_{\Omega_*} H_h^* \left(\mathbf{x}_h^{bdf,n} + \frac{\Delta t}{\alpha_0} \mathbf{v}_h^{n+1} \right) \cdot \mathbf{n}_{\Omega_*}\, d\Omega_* = A_0, \tag{6.2.18}$$

which becomes:

$$\int_{\Omega_*} H_h^* \mathbf{v}_h^{n+1} \cdot \mathbf{n}_{\Omega_*}\, d\Omega_* = \frac{\alpha_0}{\Delta t} \left(A_0 - \int_{\Omega_*} H_h^* \mathbf{x}_h^{bdf,n} \cdot \mathbf{n}_{\Omega_*}\, d\Omega_* \right). \tag{6.2.19}$$

Similarly, we rewrite the volume constraint as:

$$\int_{\Omega_*} \mathbf{v}_h^{n+1} \cdot \mathbf{n}_{\Omega_*} \, d\Omega_* = \frac{\alpha_0}{\Delta t} \left(V_0 - \int_{\Omega_*} \mathbf{x}_h^{bdf,n} \cdot \mathbf{n}_{\Omega_*} \, d\Omega_* \right). \tag{6.2.20}$$

Hereafter, with *scheme C-2* we will refer to problem (6.2.2) however with the equations related to the area and volume constraints replaced, for each $n = 0, \ldots, N-1$, by Eqs. (6.2.19) and (6.2.20), respectively.

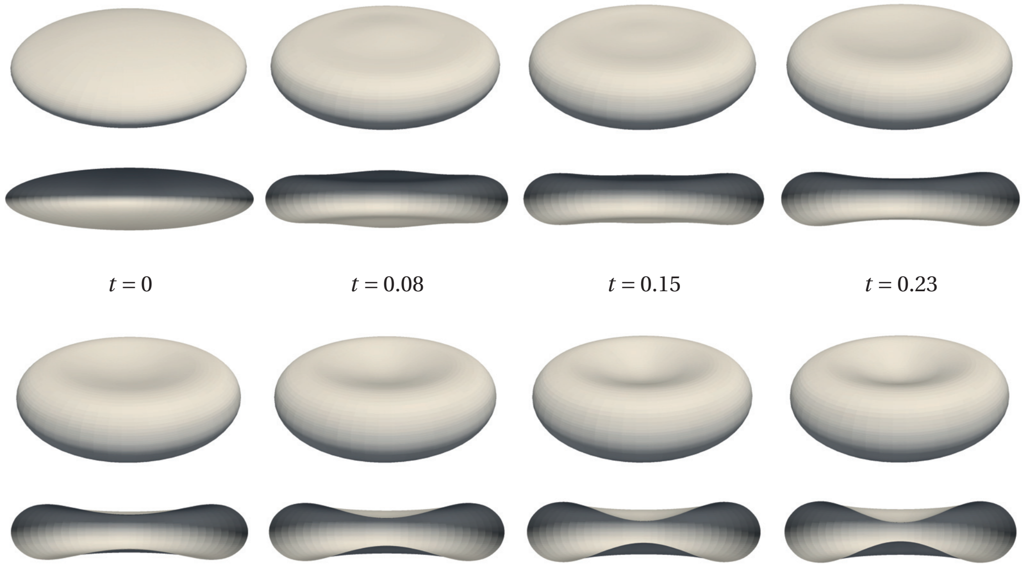
6.3 Numerical Results

In this section, we discuss the numerical results obtained by the approximation of the Canham–Helfrich problem on different initial geometries. Firstly, we test the formulation of Eq. (6.2.2) with two ellipsoids of different aspect ratio, showing the results and analyzing the two constraint enforcement schemes in action. A comparison with known solutions of the spontaneous curvature model follows.

6.3.1 Benchmark cases

As benchmark cases, we consider two biomembranes with initial ellipsoidal geometry $\Omega_0 \subset \mathbb{R}^3$ defined by the relation in Eq. (5.4.3). In the first case, we take $a_0 = 4$, $b_0 = 4$, and $c_0 = 1$ (which we refer to as *ellipsoid 4-4-1*), in the second one, we take $a_0 = 5$, $b_0 = 5$, and $c_0 = 1$ (which we call *ellipsoid 5-5-1*). For each of the two ellipsoids, we consider 4 different meshes: the first two built out of NURBS basis functions of polynomial degree $p = 2$, C^1-continuous a.e., for two refinement levels — yielding 684 elements and 2,048 total DOFs (*ref. 1*) and 2,380 elements and 8,192 total DOFs (*ref. 2*), respectively — and other two meshes built out of NURBS basis functions of degree $p = 3$, C^2-continuous a.e., for two refinement levels yielding 779 elements and 2,048 total DOFs (*ref. 1*) and 2,555 elements and 8,192 total DOFs (*ref. 2*), respectively. Regarding the time discretization, we present numerical results obtained using a BDF scheme of order $k = 2$ since it represents a good compromise between accuracy and computational cost, as shown in Chapter 5. We consider a fixed time step size $\Delta t = 0.01$, and we set the constant $k_c = 1$ and the spontaneous curvature to $H_0 = 0$.

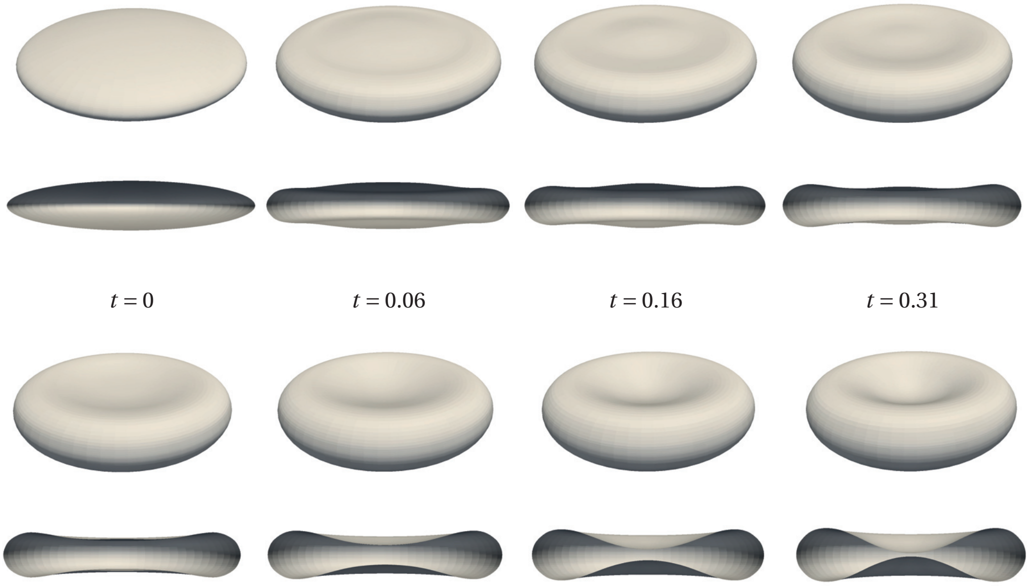
In Figures 6.1 and 6.2 we report the approximated surfaces Ω_n at different time instances computed with the ref. 1 meshes built out of $p = 2$ degree NURBS basis functions and the scheme C-1 for the enforcement of the constraints, for the ellipsoid 4-4-1 and the ellipsoid 5-5-1, respectively. In both the cases, the biomembrane starts with an initial ellipsoid shape and converges to the typical biconcave shape of the red blood cells. The aspect ratio of the initial ellipsoid geometry sets the volume V_0 and area A_0 constraints. Then, also the final shape depends on the initial aspect ratio: considering Eq. (5.4.3), the higher a_0 and b_0 with respect to c_0, the closer the two opposite sides of the biconcave shape. This trend will be shown more in details in Section 6.3.2.



$t = 0$ $t = 0.08$ $t = 0.15$ $t = 0.23$

$t = 0.30$ $t = 0.45$ $t = 0.75$ $t = 1.51$

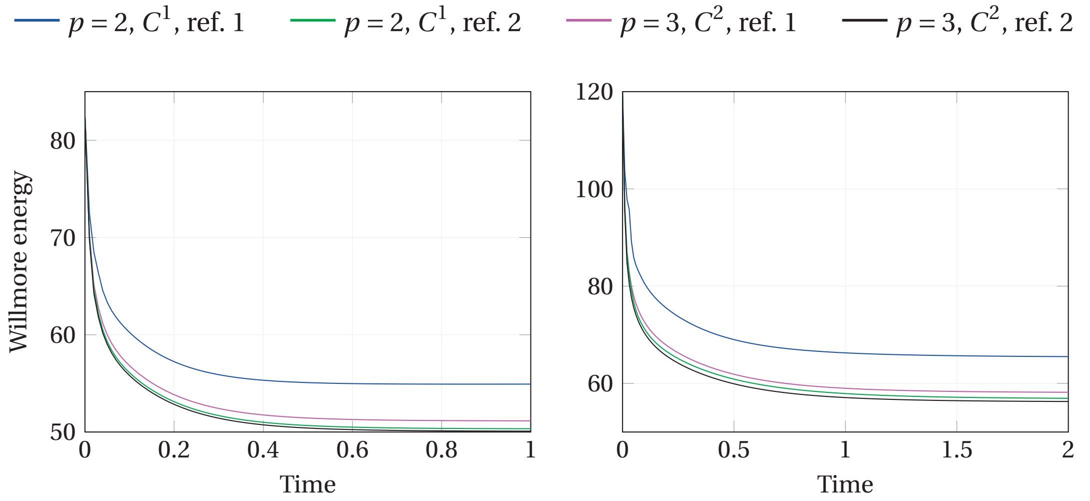
Figure 6.1 – Numerical approximation of the Canham–Helfrich flow of an ellipsoid Ω_0 with aspect ratio 4-4-1. Approximated surface Ω_n at different time instances, computed with the ref. 1 mesh built of NURBS basis functions of degree $p = 2$ and using scheme C-1.



$t = 0$ $t = 0.06$ $t = 0.16$ $t = 0.31$

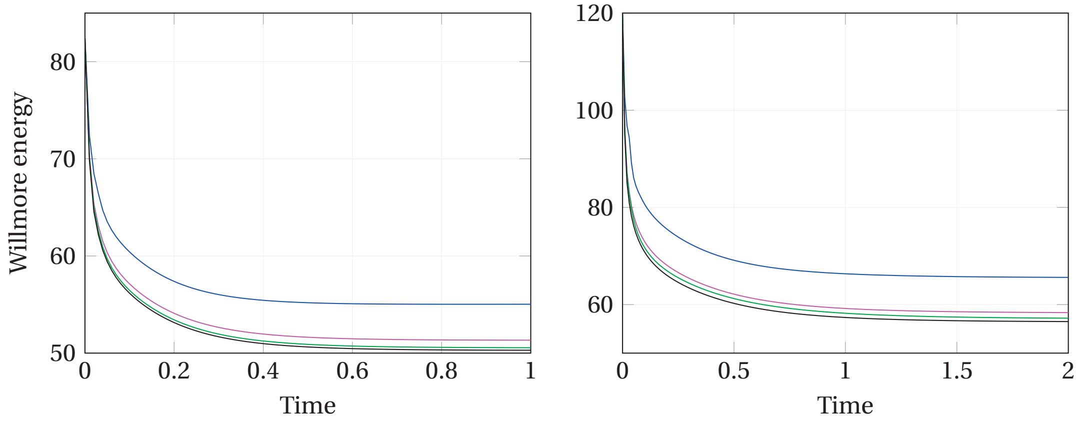
$t = 0.44$ $t = 0.75$ $t = 1.13$ $t = 1.76$

Figure 6.2 – Numerical approximation of the Canham–Helfrich flow of an ellipsoid Ω_0 with aspect ratio 5-5-1. Approximated surface Ω_n at different time instances, computed with the ref. 1 mesh built of NURBS basis functions of degree $p = 2$ and using scheme C-1.



(a) Ellipsoid 4-4-1, scheme C-1

(b) Ellipsoid 5-5-1, scheme C-1

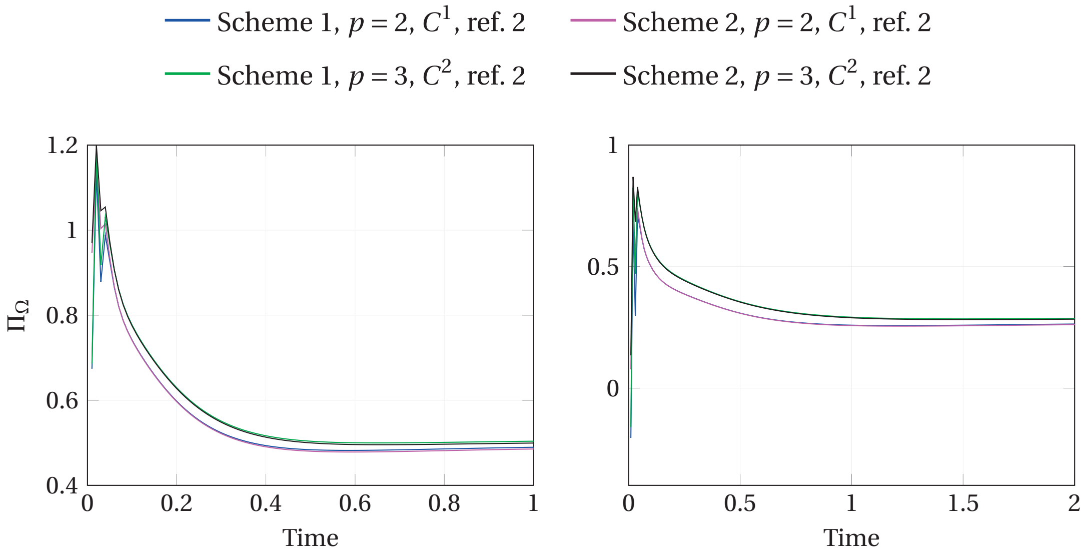


(c) Ellipsoid 4-4-1, scheme C-2

(d) Ellipsoid 5-5-1, scheme C-2

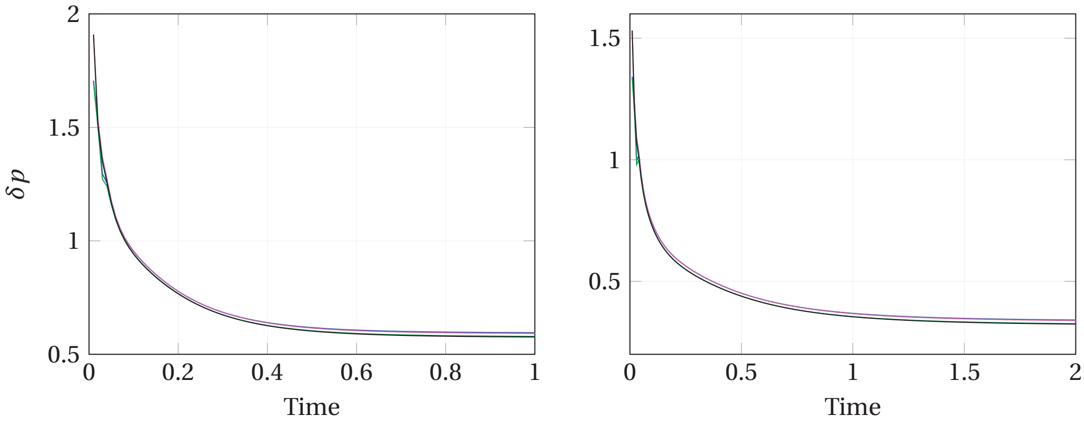
Figure 6.3 – Numerical approximation of the Canham–Helfrich flow on ellipsoids of aspect ratio 4-4-1 (Figs. (a) and (c)) and 5-5-1 (Figs. (b) and (d)). Evolution of the Willmore energy J_W with respect to time, using meshes of two refinement levels built of NURBS of degrees $p = 2$ and $p = 3$, C^1- and C^2-continuous a.e., respectively, for both schemes C-1 (Figs. (a) and (b)) and C-2 (Figs. (c) and (d)).

In Figure 6.3, we report the evolution of the Willmore energy J_W with respect to time for all the meshes considered (ref. 1 and 2 for discretizations with NURBS of both degrees $p = 2$ and $p = 3$) and both the schemes C-1 and C-2. Similarly, we report in Figure 6.4 the evolution of the Lagrange multipliers Π_Ω^n and δp^n with respect to time for the same cases. For all the situations considered, the energy is minimized until it reaches a stable biconcave configuration with a more pronounced pinching in the center of the surface when a lower value of the Willmore energy J_W is reached. The results show a common trend: the Willmore energy is minimized to a smaller and smaller value as the polynomial degree p increases, the mesh is finer and the scheme for the enforcement of the constraints is more accurate.



(a) Ellipsoid 4-4-1, area Lag. mult. Π_Ω

(b) Ellipsoid 5-5-1, area Lag. mult. Π_Ω



(c) Ellipsoid 4-4-1, volume Lag. mult. δp

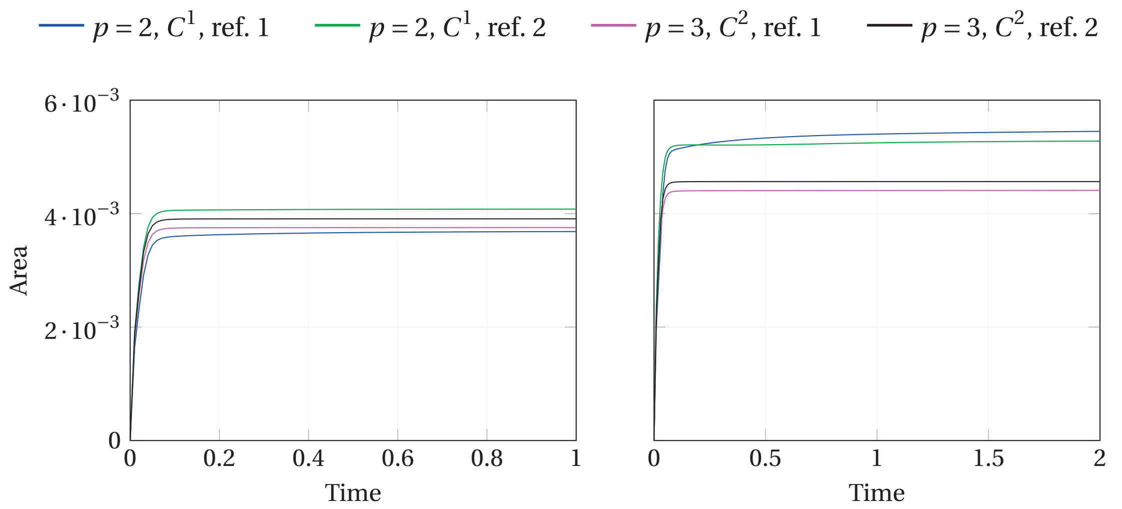
(d) Ellipsoid 5-5-1, volume Lag. mult. δp

Figure 6.4 – Numerical approximation of the Canham–Helfrich flow on ellipsoids of aspect ratio 4-4-1 (Figs. (a) and (c)) and 5-5-1 (Figs. (b) and (d)). Evolution of the Lagrange multipliers Π_Ω (Figs. (a) and (b)) and δp (Figs. (c) and (d)) for meshes of refinement level 2 built of NURBS of degrees $p=2$ and $p=3$, C^1- and C^2-continuous a.e., respectively, for both schemes C-1 and C-2.

Scheme C-1 is able to enforce the area and volume constraints within any prescribed tolerance, i.e. the obtained Ω_n is such that:

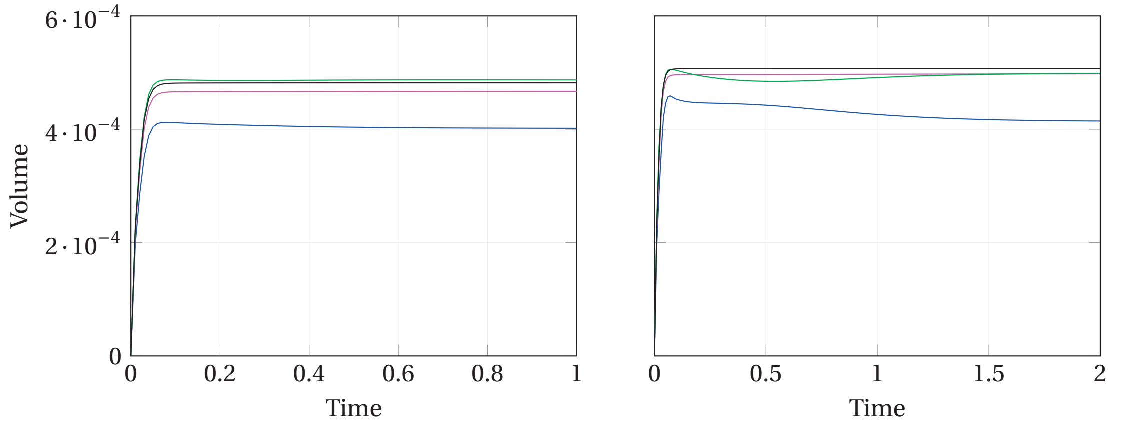
$$|J_A(\Omega_n) - A_0| \leq \varepsilon_A \qquad \text{and} \qquad |J_V(\Omega_n) - V_0| \leq \varepsilon_V, \tag{6.3.1}$$

with given tolerances $\varepsilon_A, \varepsilon_V \in \mathbb{R}$. By considering $\varepsilon_A = \varepsilon_V = 10^{-7}$, the convergence of the quasi-Newton iterations to the final values of the Lagrange multipliers takes a number of iterations usually between 1 and 3. Instead, the scheme C-2 described in Section 6.2.3 enforces the



(a) Ellipsoid 4-4-1, area

(b) Ellipsoid 5-5-1, area



(c) Ellipsoid 4-4-1, volume

(d) Ellipsoid 5-5-1, volume

Figure 6.5 – Numerical approximation of the Canham–Helfrich flow on ellipsoids of aspect ratio 4-4-1 (Figs. (a) and (c)) and 5-5-1 (Figs. (b) and (d)). Errors e_A (Figs. (a) and (b)) and e_V (Figs. (c) and (d)) in area and volume preservation with respect to time, for meshes of two refinement levels built of NURBS of degrees $p = 2$ and $p = 3$, C^1- and C^2-continuous a.e., respectively, employing the scheme C-2.

constraints only approximately. We report the evolution of the errors in area and volume of the approximated surfaces in Figure 6.5, for all the meshes considered, calculated as:

$$e_A(t) = \frac{J_A(\Omega_t) - A_0}{A_0} \qquad \text{and} \qquad e_V(t) = \frac{J_V(\Omega_t) - V_0}{V_0}, \tag{6.3.2}$$

respectively. The errors remain always positive, due to an increment in area and volume of the approximated surfaces with respect to the initial one, mostly concentrated in the initial time steps of the simulations, where the evolution of the surface is faster. Nevertheless, these increments remain in practice "small": with respect to the initial surface, the area of the approximated surface is between 0.368 ÷ 0.408 % larger, for the different NURBS cases of the ellipsoid 4-4-1, and between 0.441 ÷ 0.545 % larger, for the ellipsoid 5-5-1; the volume is instead between 0.041 ÷ 0.049 % larger, for the ellipsoid 4-4-1, and between 0.046 ÷ 0.051 % larger, for the ellipsoid 5-5-1. We consider these errors acceptable, taking also in consideration the lower computational effort needed with scheme C-2 with respect to scheme C-1: indeed, the scheme C-1 involves the solution of 3 linear systems related to the problems of Eq. (6.2.7), which share the same left-hand-side but have different right-hand-sides, followed by the iterative procedure to find the roots of the function of Eq. (6.2.8), whose Jacobian is costly to compute. The scheme C-2, instead, leads to a linear system that stems from the discretized formulation of Eq. (6.2.2) with two additional unknowns, the Lagrange multipliers Π_Ω^n and δp^n, which leads to a bigger (and slightly harder) linear system to solve, but still faster to treat than the whole procedure required for scheme C-1. Basically, by employing scheme C-2 we gain performance in exchange of lower accuracy in the conservation of the area and volume, while with scheme C-1 we obtain the best accuracy we can have (up to machine precision) at the cost of a more costly and involved procedure to follow.

6.3.2 Comparison of equilibrium shapes

Consider a vesicle Ω with surface area $A = J_A(\Omega)$ and volume $V = J_V(\Omega)$. Then let us consider a sphere whose area A_0 is the same as the vesicle area A, i.e. $A_0 = A$. The sphere's radius is therefore:

$$R_0 = \sqrt{\frac{A}{4\pi}}, \tag{6.3.3}$$

and its volume reads:

$$V_0 = \frac{4}{3}\pi R_0^3 = \frac{4}{3}\pi\left(\frac{A}{4\pi}\right)^{\frac{3}{2}}. \tag{6.3.4}$$

Since the Canham–Helfrich energy is invariant with regard to scaling transformations, it is useful to introduce the *reduced volume* V_R as the ratio between the volume of the vesicle and the volume of the sphere with same surface area, i.e. [162]:

$$V_R := \frac{V}{V_0}, \tag{6.3.5}$$

which, in terms of the vesicle area and volume, reads:

$$V_R = \frac{6\sqrt{\pi}V}{A^{3/2}}. \tag{6.3.6}$$

For a sphere, obviously we have $V_R = 1$. Similarly, the *reduced spontaneous curvature* c_0 is derived:

$$c_0 := H_0 R_0, \tag{6.3.7}$$

which, in terms of the vesicle area, reads:

$$c_0 = H_0 \sqrt{\frac{A}{4\pi}}. \tag{6.3.8}$$

We now consider the numerical approximation of the spontaneous curvature model to recover the final equilibrium shapes of vesicles in function of the reduced volume V_R. We consider the case without spontaneous curvature, i.e. $c_0 = 0$. Our aim is to find the final evolved shapes at different values of V_R. Since the area and volume of the vesicles are preserved, the reduced volume remains fixed throughout the evolution. Therefore, as initial solutions we construct ellipsoids with aspect ratio such that their reduced volumes assume the desired values. Our formulation, based on the L^2-gradient flow of the Canham–Helfrich energy, leads the evolution of the geometry towards the nearest local minimum, thus the final equilibrium shape depends on the initial solution. Therefore, for each reduced volume we consider two different initial spheroids: an *oblate* spheroid, for which the polar radius is smaller than the equatorial radius, and a *prolate* spheroid, for which the polar radius is greater than the equatorial radius. By rewriting Eq. (5.4.3) here for convenience, an ellipsoid described by the relation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \qquad \left(x, y, z\right) \in \mathbb{R}^3 \tag{6.3.9}$$

is an oblate spheroid if $b = a$ and $c < a$, while it is a prolate spheroid if $b = a$ and $c > a$ (if $a = b = c$ this reduces to a sphere). Regarding oblate spheroids, their surface area can be obtained as:

$$A_{ob}(a, c) = 2\pi a^2 \left(1 + \frac{1 - e_{ob}^2}{e_{ob}} \tanh^{-1} e_{ob}\right), \tag{6.3.10}$$

where e_{ob} is the ellipticity, reading:

$$e_{ob}(a, c) = \sqrt{1 - \frac{c^2}{a^2}}. \tag{6.3.11}$$

In the case of prolate spheroids, the surface area is calculated as:

$$A_{pr}(a, c) = 2\pi a^2 \left(1 + \frac{c}{a\, e_{pr}} \sin^{-1} e_{pr}\right), \tag{6.3.12}$$

with ellipticity e_{pr} now defined as:

$$e_{pr}(a, c) = \sqrt{1 - \frac{a^2}{c^2}}. \tag{6.3.13}$$

The volume of a generic ellipsoid defined by Eq. (6.3.9) is:

$$V(a,b,c) = \frac{4}{3}\pi abc, \tag{6.3.14}$$

which, in the case of spheroids for which $a = b$, reduces to:

$$V(a,c) = \frac{4}{3}\pi a^2 c. \tag{6.3.15}$$

Then, by using Eqs. (6.3.10) and (6.3.15), in order to calculate the aspect ratio of an oblate spheroid given its reduced volume V_R, one can fix c to a certain constant $\bar{c}$, e.g. $c = \bar{c} = 1$, and then seek a as the root of:

$$f_{ob}(a) = 6\sqrt{\pi}\frac{V(a,\bar{c})}{A_{ob}(a,\bar{c})^{3/2}} - V_R. \tag{6.3.16}$$

Similarly, considering prolate spheroids and using Eqs. (6.3.12) and (6.3.15), the value of a such that a spheroid with $a = b$ and $c = \bar{c}$ fixed has a certain reduced volume V_R can be obtained by finding the root of:

$$f_{pr}(a) = 6\sqrt{\pi}\frac{V(a,\bar{c})}{A_{pr}(a,\bar{c})^{3/2}} - V_R. \tag{6.3.17}$$

For the sake of comparison, we consider values for the reduced volume used in [171], for which the authors have themselves compared their results against the phase diagrams in [162]. An initial NURBS sphere is then scaled in order to assume the desired spheroidal shape by using, for each value of V_R, the parameters obtained with Eqs. 6.3.16 and 6.3.17.

Evolutions of the geometries are reported in Figures 6.7 and 6.9, regarding the oblate and prolate spheroids, respectively. In Figures 6.8 and 6.10 evolution of the Canham–Helfrich energies for each case considered is reported. The computational NURBS mesh is built of NURBS basis functions of degree $p = 3$ and C^2-continuous a.e., for a total of 779 elements, yielding a linear system of 2,048 DOFs to solve at each time step. Discretization in time is performed with a BDF scheme of order 2 and time step size Δt depending empirically on the initial curvature (which itself depends on the initial aspect ratio of the spheroid, in general being $\Delta t = 0.01$ and changed to $0.5 \cdot 10^{-4}$, 10^{-5}, and 10^{-6} when considering prolate ellipsoids of reduced volume 0.7, 0.65, and 0.58, respectively).

In Figure 6.6 we report the obtained normalized Canham–Helfrich energy $\widetilde{J}_{CH}$, defined as:

$$\widetilde{J}_{CH} := \frac{J_{CH}}{8\pi}, \tag{6.3.18}$$

in function of the reduced volume. The resulting diagram shows the same results as Figure 8 in [162]. In particular, for $\widetilde{V}_R < V_R \leq 1$ with $\widetilde{V}_R \simeq 0.65$ the prolate shapes have the lowest energy, evolving towards dumbbell geometries, while for $V_R < \widetilde{V}_R$ it is the oblate shapes, assuming the

V_R	1.0	0.9	0.8	0.7	0.65	0.58
$\widetilde{J}_{CH}$ (oblates)	1.00	1.19	1.39	1.65	1.83	2.20
$\widetilde{J}_{CH}$ (prolates)	1.00	1.22	1.45	1.70	1.83	2.02
Min. $\widetilde{J}_{CH}$	1.00	1.19	1.39	1.65	1.83	2.02
[171]	1.0	1.19	1.40		1.83	2.01

Table 6.1 – Numerical approximation of the Canham–Helfrich flow on different vesicles. Normalized Willmore energy vs reduced volume V_R. Meshes built of NURBS of degree $p = 3$ and C^2-continuous have been employed.

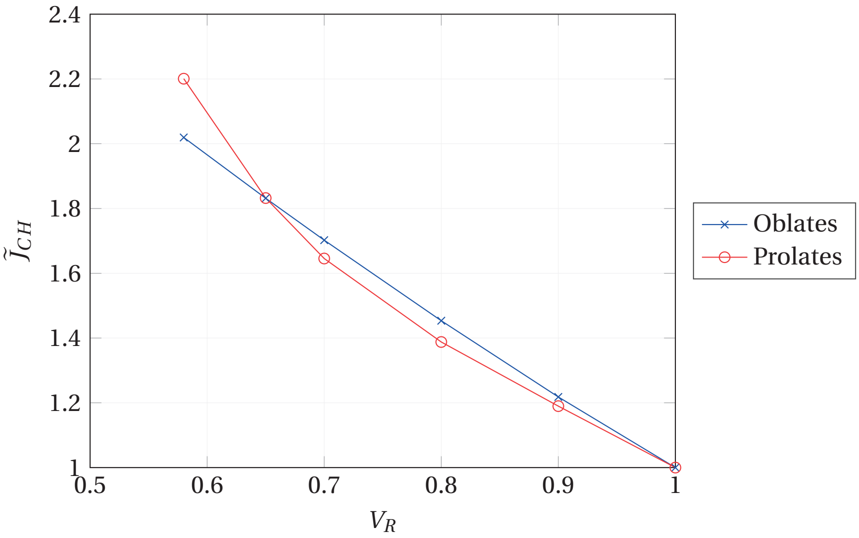
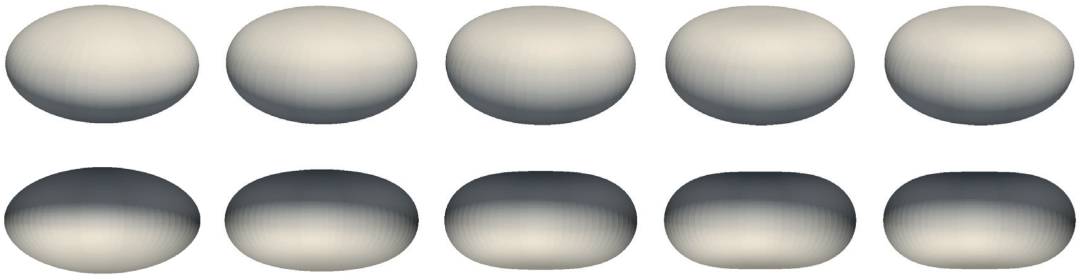


Figure 6.6 – Numerical approximation of the Canham–Helfrich flow on different vesicles. Normalized Canham–Helfrich energy $\widetilde{J}_{CH}$ vs reduced volume V_R. Meshes built of NURBS of degree $p = 3$ and C^2-continuous have been used.

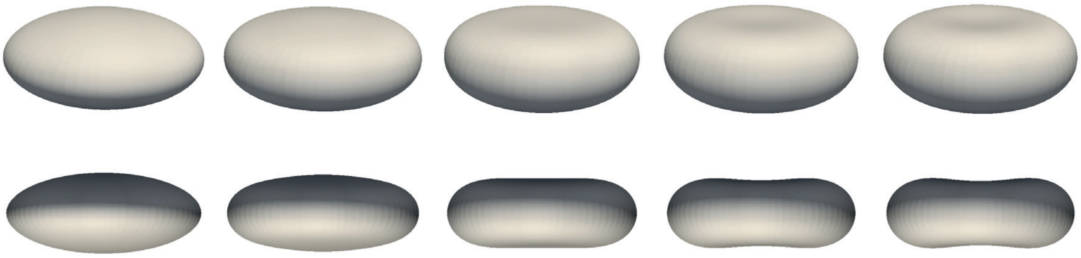
discocyte shape typical of red blood cells. In Table 6.1 we report the normalized energies of the equilibrium shapes obtained by minimizing the Canham–Helfrich energy starting from oblate and prolate spheroids, for each value of V_R considered. The reported values show agreement of the presented results with the reference values reported in [171].

We remark that, with the approach proposed, the initial shape evolves towards a local minimum, a common drawback of a shape energy minimization approach based on L^2-gradient flow; for instance, considering a initial oblate spheroid of $V_R = 0.8$, it will evolve towards a discocyte shape, even if the corresponding dumbbell shape, obtained when evolving from an initial prolate spheroid, would have lower energy.



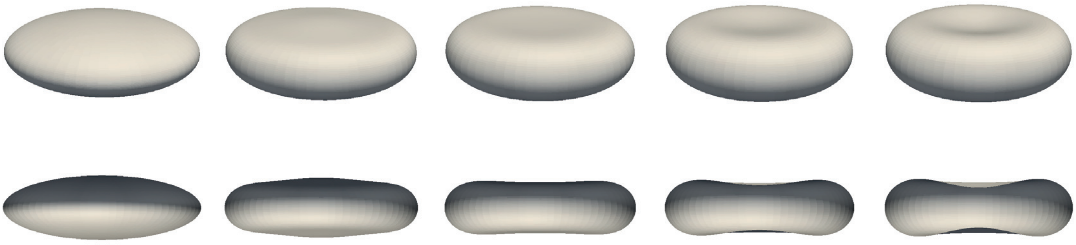
$t = 0$ $t = 0.01$ $t = 0.02$ $t = 0.03$ $t = 0.04$

Oblate ellipsoid with $V_R = 0.9$



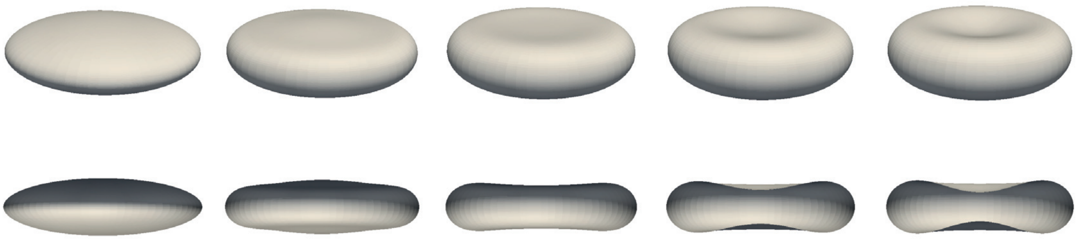
$t = 0$ $t = 0.01$ $t = 0.03$ $t = 0.05$ $t = 0.10$

Oblate ellipsoid with $V_R = 0.8$



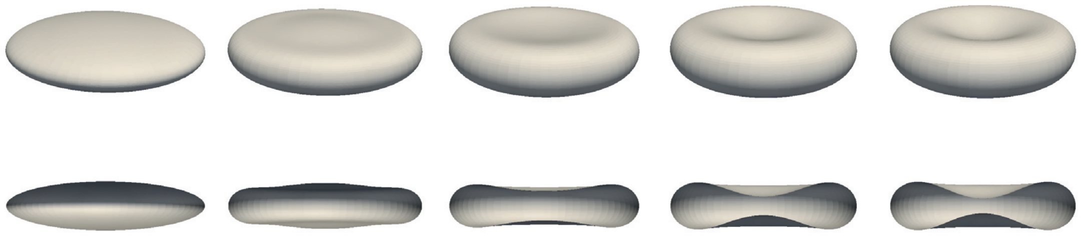
$t = 0$ $t = 0.02$ $t = 0.05$ $t = 0.09$ $t = 0.28$

Oblate ellipsoid with $V_R = 0.7$



$t = 0$ $t = 0.02$ $t = 0.07$ $t = 0.15$ $t = 0.40$

Oblate ellipsoid with $V_R = 0.65$



$t = 0$ $t = 0.03$ $t = 0.15$ $t = 0.35$ $t = 0.80$

Oblate ellipsoid with $V_R = 0.58$

Figure 6.7 – Canham–Helfrich flow applied to oblate ellipsoids of different reduced volume V_R. Approximated geometry at different time instances.

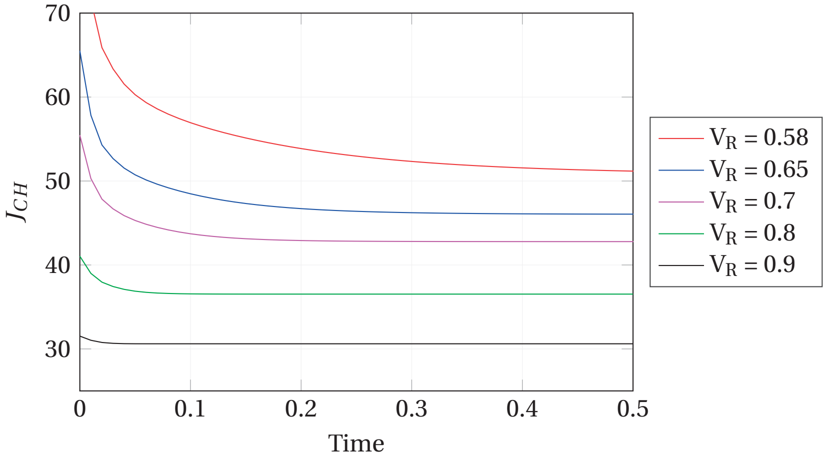
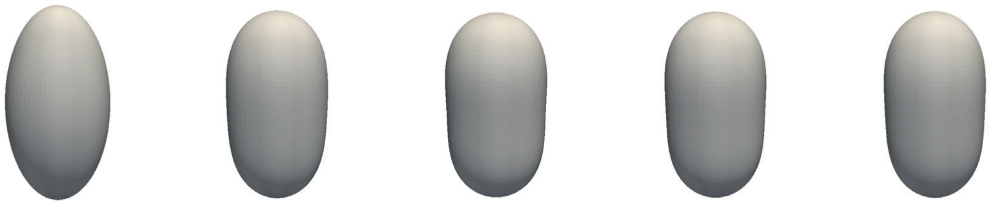
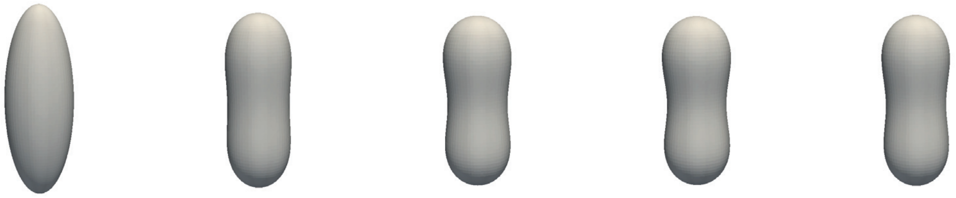


Figure 6.8 – Canham–Helfrich flow applied to oblate ellipsoids of different reduced volume V_R. Evolution of the Canham–Helfrich energies over time.



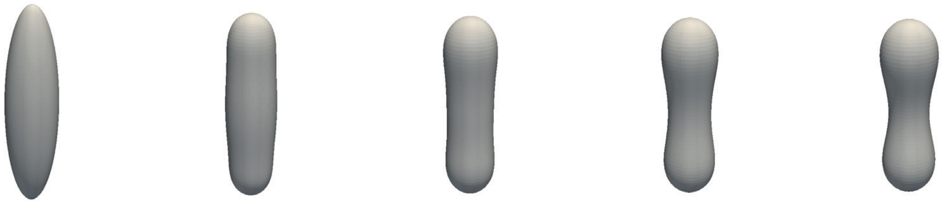
$t = 0$ $t = 0.01$ $t = 0.02$ $t = 0.03$ $t = 0.06$

Prolate ellipsoid with $V_R = 0.9$



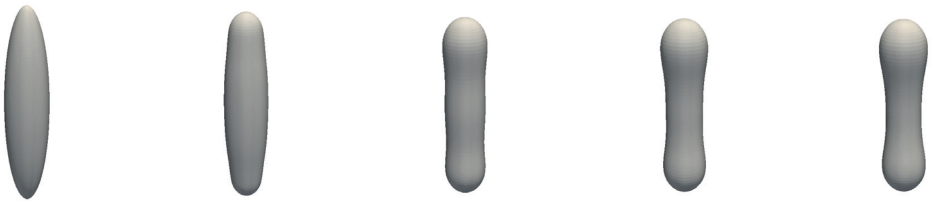
$t = 0$ $t = 0.01$ $t = 0.02$ $t = 0.03$ $t = 0.12$

Prolate ellipsoid with $V_R = 0.8$



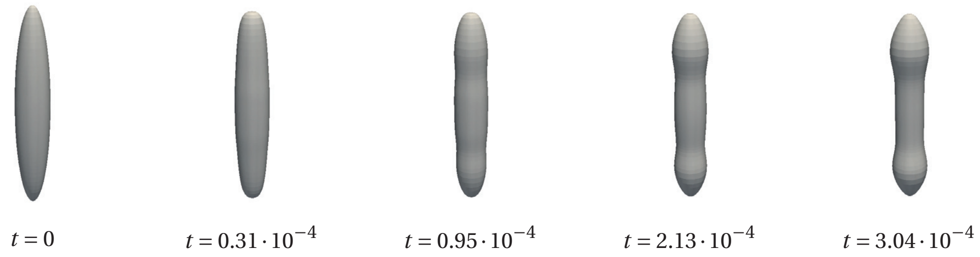
$t = 0$ $t = 0.10 \cdot 10^{-3}$ $t = 0.40 \cdot 10^{-3}$ $t = 0.80 \cdot 10^{-3}$ $t = 2.10 \cdot 10^{-3}$

Prolate ellipsoid with $V_R = 0.7$



$t = 0$ $t = 0.30 \cdot 10^{-4}$ $t = 2.20 \cdot 10^{-4}$ $t = 4.70 \cdot 10^{-4}$ $t = 7.40 \cdot 10^{-4}$

Prolate ellipsoid with $V_R = 0.65$



Prolate ellipsoid with $V_R = 0.58$

Figure 6.9 – Canham–Helfrich flow applied to prolate ellipsoids of different reduced volume V_R. Approximated geometry at different time instances.

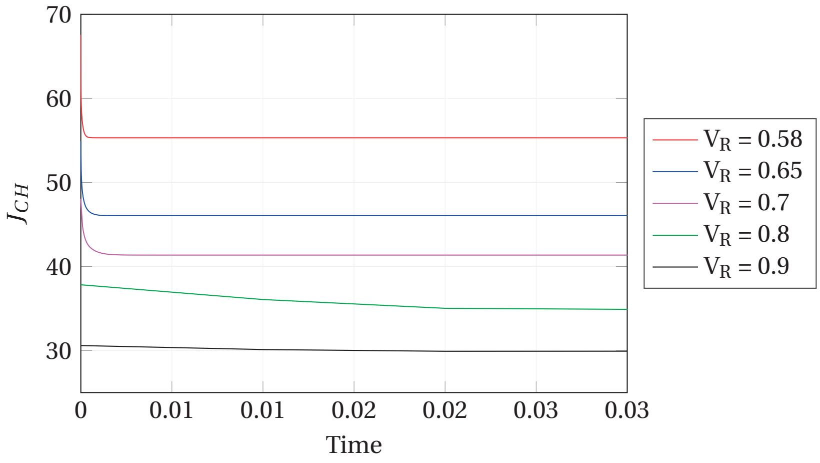


Figure 6.10 – Canham–Helfrich flow applied to prolate ellipsoids of different reduced volume V_R. Evolution of the Canham–Helfrich energies over time.

7 Dynamics of lipid vesicles in fluids

In Chapter 6, we studied a model for determining the equilibrium shapes of lipid vesicles. Aim of this chapter is to extend the considered model by introducing the presence of an external and internal fluid and dealing with the fluid-membrane interaction. This is a common case in biology, a noticeable example being the red blood cells, filled by cytoplasm, and immersed in the plasma. In this chapter, we will consider incompressible Newtonian fluids, hence governed by the incompressible Navier–Stokes equations, an assumption which is usually acceptable at the space and time scales of cellular flow [55]. When considering the dynamics of a vesicle, the external fluid plays a major role, as it exerts forces on the membrane while carrying it with the flow. Nevertheless, also the presence of an internal fluid affects the behavior of the biomembrane [55]. In fact, even when studying the equilibrium configurations, pure geometric curvature-based models, as the one analyzed in Chapter 6, may not be enough to capture the dynamics of the vesicle towards its equilibrium configuration; it may be necessary to take into account the presence of the internal fluid, which influences the inertia and eventual strong pinching of the biomembrane, as studied numerically in [52].

Regarding the numerical approximation, one important aspect is the choice of geometrical representation and space discretization technique for the fluid and the membrane domains. In fact, standard approximation methods employed for fluid-structure interaction problems on boundary fitted meshes [172] may not be well suited for this kind of problems, due to the potentially large displacements and deformations that a vesicle undergoes when immersed into a fluid. Indeed, a boundary fitted representation requires mesh-motion algorithms able to cope with the severe deformations, rotations, and translations of the biomembrane around the fluid computational domain. Conversely, an usual approach consists in empoying a fixed computational mesh for the fluid and an implicit description of the immersed vesicle, e.g. by using level sets [73, 87, 88], or diffuse interfaces, as in [86]. In [79, 89] the evolution of the vesicle is followed by a parametric mesh, which is then updated by the fluid velocity. A different approach, based on the Boundary Integral Method and supporting also vesicle-vesicle interactions, is presented in [81]. Approaches based on the immersed boundary method are employed in [83, 84, 85].

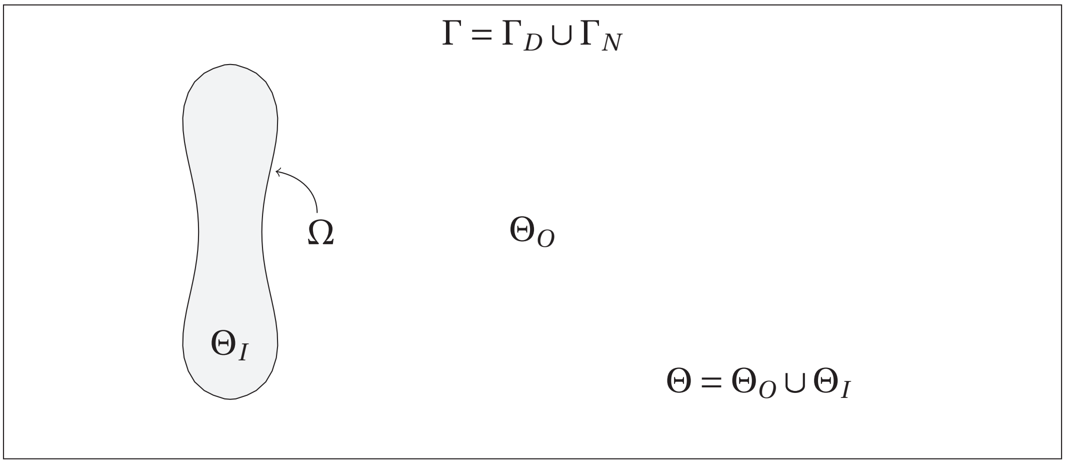

$\Gamma = \Gamma_D \cup \Gamma_N$
Ω
Θ_O
Θ_I
$\Theta = \Theta_O \cup \Theta_I$

Figure 7.1 – Domains for the problems considered in this chapter. The containing domain Θ, with boundary Γ, is split into the domain Θ_I internal to the obstacle and the external domain Θ_O; the boundary of Θ_I, interface between Θ_I and Θ_O, is denoted by Ω.

In this chapter, we consider an approach based on the *Resistive Immersed Surface* method [92], used to model obstacles to the flow and porous interfaces [90, 91] by considering a penalization of the fluid velocity in a specific region of the domain. We will discuss the resistive method in Section 7.2.3; in Section 7.3 we extend the technique to deal with immersed NURBS geometries.

This chapter is divided into four parts. After a brief description of the notation used for the fluid and membrane domains in Section 7.1, we introduce the Navier–Stokes equations and deal with their numerical approximation in Section 7.2. Then, in Section 7.3 we describe our approach for dealing with immersed NURBS geometries. Finally, we introduce the coupled model between the vesicle and the surrounding fluid, discussing about the formulation, discretization, and numerical results in Section 7.4.

7.1 Description of the domains

Let us consider a domain $\Theta \subset \mathbb{R}^d$, with d indicating the dimensionality of the physical space, for $d = 2$ or 3. A vesicle, represented by $\Theta_I \subset \mathbb{R}^d$, is placed in Θ. The subdomain of Θ that is not occupied by the immersed object is denoted with $\Theta_O \subset \mathbb{R}^d$ and it is filled by fluid. The object and fluid domains are such that $\Theta = \Theta_O \cap \Theta_I$. We denote with $\Omega \subset \mathbb{R}^{d-1}$ the interface between the two subdomains. We consider the external boundaries of Θ_O to be partitioned into two subsets Γ_D and Γ_N such that $\Gamma = \Gamma_D \cup \Gamma_N$ and $\mathring{\Gamma}_D \cap \mathring{\Gamma}_N = \emptyset$. The two boundaries will different boundary conditions for the fluid, depending on the case and type of flow considered. The domains, boundaries and the interface are sketched in Figure 7.1. As an example in the context of blood modeling, the domain Θ could represent a vessel, the object Θ_I a red blood cell, and the interface Ω would represent the membrane of the cell; the subdomain Θ_O would then be filled by the plasma.

7.2 The fluid equations

Initially, let us consider only the fluid and treat the immersed object just as an obstacle for the flow. We consider the domains Θ_O and Θ_I to be both filled with incompressible Newtonian fluids, potentially with different parameters (e.g. density or viscosity) between each other. Therefore, we choose to consider the fluids in the whole domain Θ as governed by a single set of equations, with spatially varying parameters such that they represent the properties of one fluid or the other depending if evaluated inside the immersed object or outside. Even if in this section the immersed object is considered to be fixed, we still formulate the problem in the whole domain Θ, in view of the fluid-vesicle interaction that will be introduced in Section 7.4. By considering the interval of time $(0,T)$ and by indicating with $\mathbf{u}(t) : \Theta \to \mathbb{R}^d$ the unknown velocity of the fluid at time t, the equations governing an incompressible fluid in Θ with no external forces read:

$$\text{for a.e. } t \in (0,T), \text{ find } \mathbf{u}(t) : \Theta \to \mathbb{R}^d \text{ such that}$$
$$\begin{cases} \rho \dfrac{\mathrm{D}\mathbf{u}}{\mathrm{dt}} - \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} & \text{in } \Theta,\ t \in (0,T), \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Theta,\ t \in (0,T), \end{cases} \tag{7.2.1}$$

where $\rho \in \mathbb{R}^+$ represents the density of the fluid and $\boldsymbol{\sigma} : \Theta \to \mathbb{R}^{d\times d}$ is the Cauchy stress tensor. The first Eq. in (7.2.1) is the *momentum equation*, since it represents the balance of momentum. The second equation refers instead to the balance of mass and it is responsible for modeling the incompressibility of the fluid; it is commonly called *continuity equation*. In particular, we consider Newtonian incompressible fluids, for which the stress tensor $\boldsymbol{\sigma}$ reads:

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu D(\mathbf{u}) \qquad \text{in } \Theta, \tag{7.2.2}$$

where $\mu : \Theta \to \mathbb{R}^+$ is the dynamic viscosity of the fluid, $p : \Theta \to \mathbb{R}$ is the unknown pressure, and $D(\cdot)$ denotes the symmetric gradient operator, i.e.:

$$D(\boldsymbol{\varphi}) := \frac{1}{2}\left(\nabla\boldsymbol{\varphi} + \left(\nabla\boldsymbol{\varphi}\right)^T\right) \qquad \text{for } \boldsymbol{\varphi} : \Theta \to \mathbb{R}^d. \tag{7.2.3}$$

The notation $\dfrac{\mathrm{D}\mathbf{u}}{\mathrm{dt}}$ refers to the material derivative of $\mathbf{u}$, which reads:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{dt}} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}. \tag{7.2.4}$$

Equations (7.2.1) are supplied with the divergence-free initial condition:

$$\mathbf{u}(0) = \mathbf{u}_0 \qquad \text{in } \Theta, \tag{7.2.5}$$

and with suitable boundary conditions on the boundary of the fluid domain, for example:

$$\begin{aligned} &\mathbf{u} = \mathbf{g} && \text{on } \Gamma_D,\ t \in (0,T),\\ &\boldsymbol{\sigma}\mathbf{n} = -p\mathbf{n} + 2\mu D(\mathbf{u})\mathbf{n} = \mathbf{h} && \text{on } \Gamma_N,\ t \in (0,T), \end{aligned} \tag{7.2.6}$$

with $\mathbf{g} : \Gamma_D \to \mathbb{R}^d$, $\mathbf{h} : \Gamma_N \to \mathbb{R}^d$, and $\mathbf{n}$ being the outward directed unit vector normal to Γ_N. Therefore, for incompressible Newtonian fluids, by using Eqs. (7.2.2) and (7.2.4) in Eq. (7.2.1), the Navier–Stokes equations read:

for a.e. $t \in (0,T)$, find $\mathbf{u}(t) : \Theta \to \mathbb{R}^d$ and $p(t) : \Theta \to \mathbb{R}$ such that

$$\begin{cases} \rho \dfrac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} - 2\nabla \cdot \left(\mu D(\mathbf{u})\right) + \nabla p = \mathbf{0} & \text{in } \Theta,\ t \in (0,T),\\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Theta,\ t \in (0,T),\\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D,\ t \in (0,T),\\ -p\mathbf{n} + 2\mu D(\mathbf{u})\mathbf{n} = \mathbf{h} & \text{on } \Gamma_N,\ t \in (0,T),\\ \mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Theta, \end{cases} \tag{7.2.7}$$

where the viscosity μ is a function defined on Θ such that:

$$\mu(\mathbf{p}) = \begin{cases} \mu_O & \text{if } \mathbf{p} \in \Theta_O,\\ \mu_I & \text{if } \mathbf{p} \in \Theta_I, \end{cases} \qquad \text{for } \mathbf{p} \in \Theta, \tag{7.2.8}$$

where $\mu_O \in \mathbb{R}^+$ is the dynamic viscosity of the part of fluid in Θ_O and $\mu_I \in \mathbb{R}^+$ is the dynamic viscosity of the part of fluid inside Θ_I.

7.2.1 Non-dimensionalization of the Navier–Stokes equations

We proceed with rewriting the Navier–Stokes equations (7.2.7) in dimensionless form. With this aim, the following dimensionless quantities are introduced:

$$\bar{x} = \frac{1}{L}x, \qquad \bar{\mathbf{u}} = \frac{1}{U}\mathbf{u}, \qquad \bar{t} = \frac{1}{T}t = \frac{U}{L}t, \tag{7.2.9}$$

where L, U, and T refers to the characteristic length, velocity, and time, respectively. The differential operators are rewritten as:

$$\frac{\partial}{\partial \bar{t}}\cdot = \frac{1}{T}\frac{\partial}{\partial t}\cdot, \qquad \bar{\nabla}\cdot = \frac{1}{L}\nabla\cdot, \qquad \bar{\nabla}\cdot(\cdot) = \frac{1}{L}\nabla\cdot(\cdot), \qquad \text{and} \quad \bar{D}\cdot = \frac{1}{2}\left(\bar{\nabla}\cdot + \bar{\nabla}\cdot^T\right) = \frac{1}{L}D\cdot. \tag{7.2.10}$$

By using the quantities in Eq. (7.2.9) and the operators in Eq. (7.2.10), the Navier–Stokes equations (7.2.7) are written in dimensionless form as:

for a.e. $t \in (0,T)$, find $\bar{\mathbf{u}}(t) : \Theta \to \mathbb{R}^d$ and $p(t) : \Theta \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\rho U^2}{L}\dfrac{\partial \bar{\mathbf{u}}}{\partial \bar{t}} + \dfrac{\rho U^2}{L}\bar{\mathbf{u}} \cdot \bar{\nabla}\bar{\mathbf{u}} - 2\bar{\nabla} \cdot \left(\dfrac{\mu U}{L^2}\bar{D}(\bar{\mathbf{u}})\right) + \dfrac{1}{L}\bar{\nabla} p = \mathbf{0} & \text{in } \Theta,\ t \in (0,T), \\ \bar{\nabla} \cdot \bar{\mathbf{u}} = 0 & \text{in } \Theta,\ t \in (0,T), \\ U\bar{\mathbf{u}} = \mathbf{g} & \text{on } \Gamma_D,\ t \in (0,T), \\ -p\mathbf{n} + 2\dfrac{\mu U}{L}\bar{D}(\bar{\mathbf{u}})\mathbf{n} = \mathbf{h} & \text{on } \Gamma_N,\ t \in (0,T), \\ U\bar{\mathbf{u}}(0) = \mathbf{u}_0 & \text{in } \Theta. \end{cases} \tag{7.2.11}$$

Let us introduce the Reynolds number Re, defined as:

$$\mathrm{Re} := \frac{\rho U L}{\mu}; \tag{7.2.12}$$

the dimensionless number Re characterizes the flow regime by giving an indication of the prevalence of the inertial forces over the viscous forces. By multiplying the terms in the momentum equation by $L/\rho U^2$, we obtain the following system of equations:

for a.e. $t \in (0,T)$, find $\bar{\mathbf{u}}(t) : \Theta \to \mathbb{R}^d$ and $\bar{p}(t) : \Theta \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial \bar{\mathbf{u}}}{\partial \bar{t}} + \bar{\mathbf{u}} \cdot \bar{\nabla}\bar{\mathbf{u}} - 2\bar{\nabla} \cdot \left(\dfrac{1}{\mathrm{Re}}\bar{D}(\bar{\mathbf{u}})\right) + \bar{\nabla}\bar{p} = \mathbf{0} & \text{in } \Theta,\ t \in (0,T), \\ \bar{\nabla} \cdot \bar{\mathbf{u}} = 0 & \text{in } \Theta,\ t \in (0,T), \\ \bar{\mathbf{u}} = \bar{\mathbf{g}} & \text{on } \Gamma_D,\ t \in (0,T), \\ -\bar{p}\mathbf{n} + \dfrac{2}{\mathrm{Re}}\bar{D}(\bar{\mathbf{u}})\mathbf{n} = \bar{\mathbf{h}} & \text{on } \Gamma_N,\ t \in (0,T), \\ \bar{\mathbf{u}}(0) = \bar{\mathbf{u}}_0 & \text{in } \Theta, \end{cases} \tag{7.2.13}$$

where we have introduced the following dimensionless quantities:

$$\bar{p} = \frac{1}{\rho U^2} p, \qquad \bar{\mathbf{g}} = \frac{1}{U}\mathbf{g}, \qquad \text{and} \quad \bar{\mathbf{h}} = \frac{1}{\rho U^2}\mathbf{h}. \tag{7.2.14}$$

In particular, the dimensionless stress tensor $\bar{\boldsymbol{\sigma}}$ is linked to the standard stress tensor $\boldsymbol{\sigma}$ through the relation:

$$\boldsymbol{\sigma} = -\rho U^2 \bar{p}\mathbf{I} + 2\frac{\rho U}{\rho U}\frac{\mu U}{L}\bar{D}(\bar{\mathbf{u}}) = \rho U^2\left(-\bar{p}\mathbf{I} + \frac{2}{\mathrm{Re}}\bar{D}(\bar{\mathbf{u}})\right) = \rho U^2 \bar{\boldsymbol{\sigma}}. \tag{7.2.15}$$

From now on, the overline indicating the dimensionless quantities will be omitted to simplify the notation; each quantity appearing is always assumed to be dimensionless, unless specified otherwise.

7.2.2 Weak formulation of the fluid equations

In order to write problem (7.2.7) in weak formulation, we introduce the following function spaces for the velocity:

$$V := \left[H^1(\Theta)\right]^d, \qquad V_R := \left\{\boldsymbol{\varphi} \in V : \quad \boldsymbol{\varphi} = \mathbf{g} \quad \text{on } \Gamma_D\right\}, \qquad V_0 := \left\{\boldsymbol{\varphi} \in V : \quad \boldsymbol{\varphi} = \mathbf{0} \quad \text{on } \Gamma_D\right\}, \tag{7.2.16}$$

while for the pressure the following ones:

$$W := L^2(\Theta) \qquad \text{or} \qquad W := \left\{\psi \in L^2(\Theta) : \quad \int_\Theta \psi \, \mathrm{d}\Theta = 0\right\}, \tag{7.2.17}$$

the latter being used if $\Gamma_N \equiv \emptyset$.

The weak formulation of the Navier–Stokes problem then reads:

$$\text{for a.e. } t \in (0,T), \text{ find } \mathbf{u}(t) \in V_R \text{ and } p(t) \in W \text{ such that}$$
$$\begin{cases} \displaystyle\int_\Theta \frac{\partial \mathbf{u}}{\partial t} \cdot \boldsymbol{\varphi} \, \mathrm{d}\Theta + \int_\Theta \mathbf{u} \cdot \nabla \mathbf{u} \cdot \boldsymbol{\varphi} \, \mathrm{d}\Theta + \int_\Theta \frac{2}{\mathrm{Re}} D(\mathbf{u}) : \nabla \boldsymbol{\varphi} \, \mathrm{d}\Theta - \int_\Theta p \nabla \cdot \boldsymbol{\varphi} \, \mathrm{d}\Theta & \\ \displaystyle\qquad = \int_{\Gamma_N} \mathbf{h} \cdot \boldsymbol{\varphi} \, \mathrm{d}\Gamma_N & \forall \boldsymbol{\varphi} \in V_0, \\ \displaystyle\int_\Theta \psi \nabla \cdot \mathbf{u} = 0 & \forall \psi \in W, \end{cases} \tag{7.2.18}$$

with the initial condition:

$$\mathbf{u}(0) = \mathbf{u}_0 \qquad \text{in } \Theta, \tag{7.2.19}$$

where $\mathbf{u}_0 \in V_R$ is a compatible divergence-free initial datum.

7.2.3 The Resistive Immersed Surface method

Let us consider Ω to be the boundary of an obstacle immersed in the fluid domain. We require the following condition to be satisfied in order to have continuity of the velocities at the interface:

$$\mathbf{u} = \mathbf{v} \qquad \text{on } \Omega, \ t \in (0,T), \tag{7.2.20}$$

where $\mathbf{v} : \Omega \to \mathbb{R}^d$ represents the velocity at which Ω is moving. In view of the fluid-vesicle interaction, for which we consider the immersed object free to move and deform inside the fluid domain, we adopt in this chapter an approach based on the weak enforcement of the condition in Eq. (7.2.20). In particular, we consider the *Resistive Immersed Surface* (RIS), a penalization method already employed successfully, for instance, to approximate the flow through porous interfaces [91] or for modeling blood valves [32, 92]. Let us consider the following additional penalization term in the weak formulation of the momentum equation [173]:

$$\mathscr{R}(\mathbf{u},\boldsymbol{\varphi}) = \int_\Omega C_R(\mathbf{u}-\mathbf{v})\cdot\boldsymbol{\varphi}\,d\Omega \tag{7.2.21}$$

where $C_R \in \mathbb{R}^+$ is a penalization constant. The idea is to avoid the integral over the interface Ω by replacing it with an integral on the whole domain Θ:

$$\mathscr{R}(\mathbf{u},\boldsymbol{\varphi}) = \int_\Theta C_R\chi_\Omega(\mathbf{u}-\tilde{\mathbf{v}})\cdot\boldsymbol{\varphi}\,d\Theta \tag{7.2.22}$$

where $\tilde{\mathbf{v}}:\Theta\to\mathbb{R}^3$ is an appropriate extension, in the direction normal to Ω, of the interface velocity $\mathbf{v}$ to the fluid domain, and $\chi_\Omega:\Theta\to\{0,1\}$ is an indicator function such that:

$$\chi_\Omega(\mathbf{p}) := \begin{cases} 1 & \text{if } \mathbf{p}\in\Omega, \\ 0 & \text{if } \mathbf{p}\notin\Omega, \end{cases} \qquad \text{for } \mathbf{p}\in\Theta, \tag{7.2.23}$$

which identifies the location in space where the interface Ω lies. The penalty parameter C_R determines how the condition (7.2.20) is enforced: if $C_R = 0$, the condition is transparent to the equations; for C_R high enough, the continuity of the velocities of the fluid and the immersed body is enforced on Ω [173]. For other values of the penalty factor, the condition acts as a porous interface, as studied in [91].

In order to employ the term in Eq. (7.2.22) to weakly enforce the continuity condition (7.2.20), the indicator function for the immersed interface Ω is required. Let us consider an implicit representation of the interface Ω through a signed distance function, i.e. a function $d:\Theta\to\mathbb{R}$ such that:

$$d(\mathbf{p}) = \begin{cases} -\operatorname{dist}(\mathbf{p},\Omega) & \text{if } \mathbf{p}\in\Theta_I, \\ \operatorname{dist}(\mathbf{p},\Omega) & \text{if } \mathbf{p}\notin\Theta_I, \end{cases} \qquad \text{for } \mathbf{p}\in\Theta, \tag{7.2.24}$$

where $\operatorname{dist}(\mathbf{p},\Omega)$ indicates the Euclidean distance between the point $\mathbf{p}$ and the interface Ω:

$$\operatorname{dist}(\mathbf{p},\Omega) := \inf_{\mathbf{y}\in\Omega}\|\mathbf{y}-\mathbf{p}\| \qquad \text{for } \mathbf{p}\in\Theta, \tag{7.2.25}$$

where $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^d$. The signed distance function is zero on Ω, negative inside the immersed object, and positive outside. With this definition, the indicator function χ_Ω can be rewritten in terms of the signed distance function as:

$$\chi_\Omega(\mathbf{p}) = \delta(d(\mathbf{p})) \qquad \text{for } \mathbf{p}\in\Theta, \tag{7.2.26}$$

where $\delta:\mathbb{R}\to\mathbb{R}$ is the Dirac delta centered in 0, defined as:

$$\delta(x) := \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x \neq 0 \end{cases} \qquad \text{for } x\in\mathbb{R}. \tag{7.2.27}$$

Similarly, the indicator function for the immersed domain $\chi_{\Theta_I} : \Theta \to \mathbb{R}$, defined as:

$$\chi_{\Theta_I}(\mathbf{p}) := \begin{cases} 1 & \text{if } \mathbf{p} \in \Theta_I, \\ 0 & \text{if } \mathbf{p} \notin \Theta_I, \end{cases} \qquad \text{for } \mathbf{p} \in \Theta, \tag{7.2.28}$$

can be written by using the signed distance function as:

$$\chi_{\Theta_I}(\mathbf{p}) = 1 - \mathscr{H}(d(\mathbf{p})) \qquad \text{for } \mathbf{p} \in \Theta, \tag{7.2.29}$$

where $\mathscr{H} : \mathbb{R} \to \mathbb{R}$ is the Heaviside function centered in 0, which reads:

$$\mathscr{H}(x) := \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0, \end{cases} \qquad \text{for } x \in \mathbb{R}. \tag{7.2.30}$$

In practice, in view of the discretization of the resistive term, the indicator functions defined in Eqs. (7.2.26) and (7.2.29) are regularized. This is done by considering smooth approximations of the Delta and Heaviside functions. For an in-depth analysis on these regularizations see [174]; here we just highlight that the delta function should be a compact smoothing of the Dirac delta, with support in a narrow band of controllable width, and should have integral equal to 1.

First of all, let us denote with $\tilde{\delta}_\varepsilon : \mathbb{R} \to [0,1]$ the smoothed delta function. The parameter $\varepsilon \in \mathbb{R}^+$ controls the width of the smoothing interval, i.e.:

$$\operatorname{spt} \tilde{\delta}_\varepsilon \subseteq [-\varepsilon, \varepsilon], \tag{7.2.31}$$

where the operator spt indicates the support. The smooth delta function $\tilde{\delta}_\varepsilon$ should also satisfy:

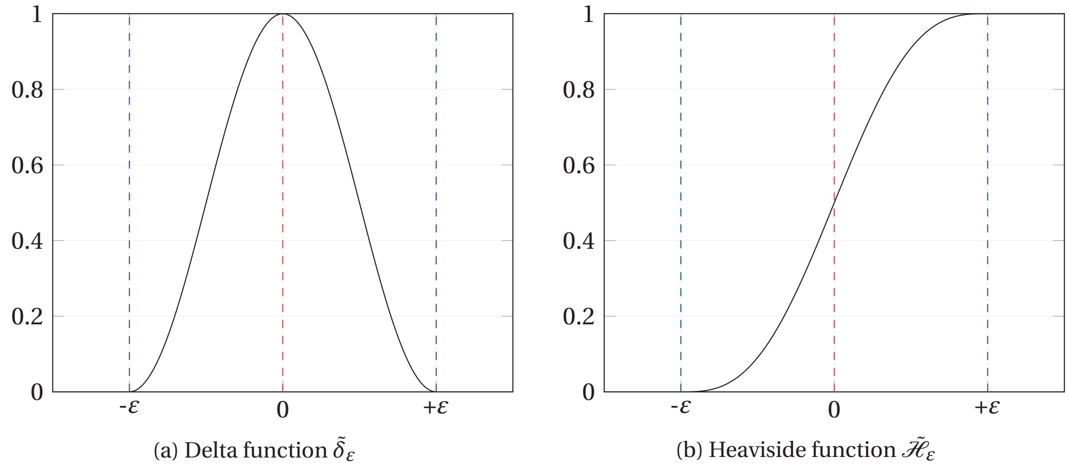
$$\int_{-\infty}^{+\infty} \tilde{\delta}_\varepsilon \, \mathrm{dx} = \int_{-\varepsilon}^{+\varepsilon} \tilde{\delta}_\varepsilon \, \mathrm{dx} = 1. \tag{7.2.32}$$

These properties are both satisfied by the following simple smooth approximation of the Dirac delta ([88, 92, 175]):

$$\tilde{\delta}_\varepsilon(x) = \begin{cases} \dfrac{1}{2\varepsilon}\left(1 + \cos\left(\dfrac{\pi x}{\varepsilon}\right)\right) & \text{if } |x| \leq \varepsilon, \\ 0 & \text{if } |x| > \varepsilon, \end{cases} \qquad \text{for } x \in \mathbb{R}, \tag{7.2.33}$$

which is adopted in this thesis. A corresponding smoothing $\tilde{\mathscr{H}}_\varepsilon : \mathbb{R} \to \mathbb{R}$ of the Heaviside function, such that it changes values from 0 to 1 in a band large 2ε, reads:

$$\tilde{\mathscr{H}}_\varepsilon(x) = \begin{cases} 0 & \text{if } x < -\varepsilon, \\ \dfrac{1}{2}\left(1 + \dfrac{x}{\varepsilon} + \dfrac{1}{\pi}\sin\left(\dfrac{\pi x}{\varepsilon}\right)\right) & \text{if } |x| \leq \varepsilon, \\ 1 & \text{if } x > \varepsilon, \end{cases} \qquad \text{for } x \in \mathbb{R}. \tag{7.2.34}$$



(a) Delta function $\tilde{\delta}_\varepsilon$

(b) Heaviside function $\tilde{\mathscr{H}}_\varepsilon$

Figure 7.2 – Delta and Heaviside functions, in (a) and (b) respectively.

Both $\tilde{\delta}_\varepsilon$ and $\tilde{\mathscr{H}}_\varepsilon$ lead to better approximations of the Dirac delta and Heaviside functions as $\varepsilon \to 0$. A sketch of the two functions is shown in Figure 7.2.

By combining $\tilde{\delta}_\varepsilon$ and $\tilde{\mathscr{H}}_\varepsilon$ with the signed distance function defined in Eq. (7.2.24) we obtain a smooth approximation of the indicator function χ_Ω, which we denote by $\delta_\varepsilon : \Theta \to \mathbb{R}$ and we define as:

$$\delta_\varepsilon(\mathbf{p}) := \tilde{\delta}_\varepsilon\big(d(\mathbf{p})\big) \qquad \text{for } \mathbf{p} \in \Theta. \tag{7.2.35}$$

Similarly, we write $\mathscr{H}_\varepsilon : \Theta \to \mathbb{R}$, the smooth approximation of χ_{Θ_I}, as:

$$\mathscr{H}_\varepsilon(\mathbf{p}) := 1 - \tilde{\mathscr{H}}_\varepsilon\big(d(\mathbf{p})\big) \qquad \text{for } \mathbf{p} \in \Theta. \tag{7.2.36}$$

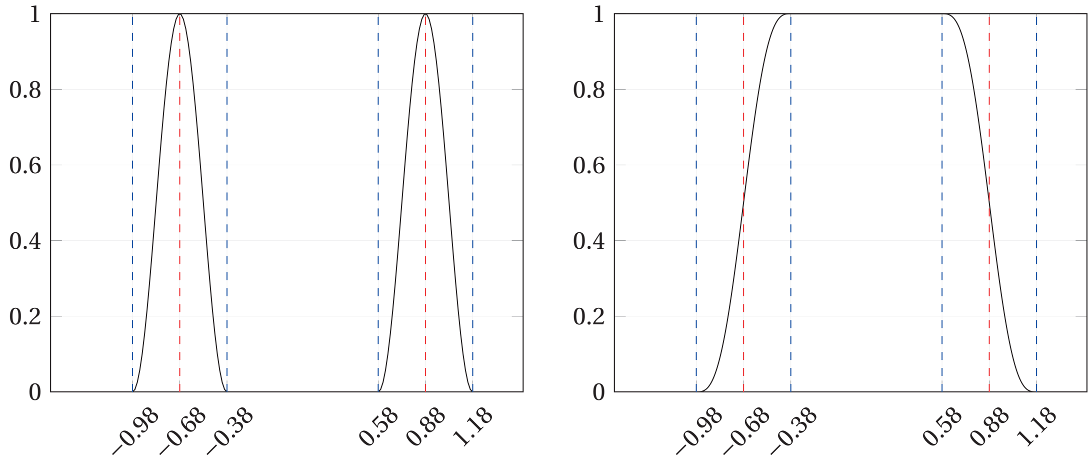
Due to the property in Eq. (7.2.32), the integral of the smooth indicator function δ_ε over the whole domain Θ is equal to the measure of the interface Ω:

$$\int_\Theta \delta_\varepsilon \,\mathrm{d}\Theta = |\Omega|, \tag{7.2.37}$$

while the integral of the smooth indicator function $\mathscr{H}_\varepsilon$ results in the measure of the immersed domain Θ_I:

$$\int_\Theta \mathscr{H}_\varepsilon \,\mathrm{d}\Theta = |\Theta_I|. \tag{7.2.38}$$

The smooth indicator function $\mathscr{H}_\varepsilon$ can also be used to express the smooth variation of quantities defined on Θ which assume different values internally to Θ_I or externally. In fact, the



(a) Delta function $\delta_\varepsilon(d(x))$ (b) Heaviside function $\mathscr{H}_\varepsilon(d(x))$

Figure 7.3 – Delta and Heaviside functions applied to the disk $\bar{D} = \{x \in \mathbb{R} : |x - x_c| \leq r\} \subset \mathbb{R}$, with $r = 0.78$, $x_c = 0.1$, and $\varepsilon = 0.3$.

regularized viscosity μ of the fluid can be written as:

$$\mu(\mathbf{p}) = \mathscr{H}_\varepsilon(\mathbf{p})\mu_I + (1 - \mathscr{H}_\varepsilon(\mathbf{p}))\mu_O, \qquad \text{for } \mathbf{p} \in \Theta. \tag{7.2.39}$$

Example. Let us consider a one-dimensional closed disk $\bar{D} \subset \mathbb{R}$ of radius $r \in \mathbb{R}$ given by:

$$\bar{D} = \{x \in \mathbb{R} : |x - x_c| \leq r\}, \tag{7.2.40}$$

centered in $x_c \in \mathbb{R}$. The boundary of $\bar{D}$ is composed by the two coordinates $x_1, x_2 \in \mathbb{R}$ such that:

$$x_1 = x_c - r \qquad \text{and} \qquad x_2 = x_c + r. \tag{7.2.41}$$

The disk can be represented implicitly by the signed distance function:

$$d(x) = |x - x_c| - r \tag{7.2.42}$$

which assumes negative values inside the disk, positive values outside, and is zero on the boundary. In Figure 7.3 we show the smooth indicator functions δ_ε and $\mathscr{H}_\varepsilon$ applied to the signed distance function in Eq. (7.2.42) for the disk with radius $r = 0.78$ and center in $x_c = 0.1$, by considering a smoothing with $\varepsilon = 0.3$. By decreasing the parameter ε the smoothing bands around the boundary coordinates get narrower; in the limit of $\varepsilon \to 0$ one obtains from δ_ε the sum of two Dirac delta functions centered in x_1 and x_2, and from $\mathscr{H}_\varepsilon$ the indicator function of the disk $\chi_{\bar{D}}$.

Finally, we rewrite problem (7.2.18) by including also the resistive term in Eq. (7.2.22) with smoothed delta function to weakly enforce the condition (7.2.20), obtaining the following problem:

for a.e. $t \in (0,T)$, find $\mathbf{u}(t) \in V_R$ and $p(t) \in W$ such that

$$\begin{cases} \displaystyle\int_\Theta \frac{\partial \mathbf{u}}{\partial t} \cdot \boldsymbol{\varphi} \, \mathrm{d}\Theta + \int_\Theta \mathbf{u} \cdot \nabla \mathbf{u} \cdot \boldsymbol{\varphi} \, \mathrm{d}\Theta + \int_\Theta \frac{2}{\mathrm{Re}} D(\mathbf{u}) : \nabla \boldsymbol{\varphi} \, \mathrm{d}\Theta - \int_\Theta p \nabla \cdot \boldsymbol{\varphi} \, \mathrm{d}\Theta & \\ \displaystyle\qquad + \int_\Theta C_R \delta_\varepsilon (\mathbf{u} - \tilde{\mathbf{v}}) \cdot \boldsymbol{\varphi} \, \mathrm{d}\Theta = \int_{\Gamma_N} \mathbf{h} \cdot \boldsymbol{\varphi} \, \mathrm{d}\Gamma_N & \forall \boldsymbol{\varphi} \in V_0, \\ \displaystyle\int_\Theta \psi \nabla \cdot \mathbf{u} = 0 & \forall \psi \in W. \end{cases} \tag{7.2.43}$$

In general, the signed distance function d depends on the shape of Ω. While for some common geometries it can be expressed analytically, as in Eq. (7.2.42) for the disk, for more general shapes, which cannot be represented via a simple analytical description, its determination becomes a non-trivial task. In this regard, we will discuss our approach for obtaining the signed distance of an immersed NURBS geometry in Section 7.3.

7.2.4 Computation of forces on an immersed object

When a obstacle is immersed in a flowing fluid, it is subject to forces exerted by fluid. In particular, the component of this force aligned to the direction of the flow is called *drag*, while the component of the force normal to the direction of the flow is called *lift*. By considering the setup sketched in Figure 7.1, the dimensionless forces acting on the immersed object Θ_I are computed as:

$$\mathbf{F} = \int_\Omega \left(-p\mathbf{I} + \frac{2}{\mathrm{Re}} D(\mathbf{u}) \right) \mathbf{n} \, \mathrm{d}\Omega, \tag{7.2.44}$$

which in dimensional form read:

$$\mathbf{F}_d = \rho_F U^2 \mathbf{F}. \tag{7.2.45}$$

In particular, the drag and lift are obtained as:

$$F_D = \int_\Omega \left(-p \, n_x + \frac{1}{\mathrm{Re}} \frac{\partial (\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}} n_y \right) \mathrm{d}\Omega \tag{7.2.46}$$

and

$$F_L := \int_\Omega \left(-p \, n_y + \frac{1}{\mathrm{Re}} \frac{\partial (\mathbf{u} \cdot \mathbf{t})}{\partial \mathbf{n}} n_x \right) \mathrm{d}\Omega, \tag{7.2.47}$$

respectively, where **n** is the unit vector normal to the interface Ω, with components (n_x, n_y), and **t** is the unit vector tangent to the interface, with components $(n_y, -n_x)$.

Since we employ an immersed approach to represent the obstacle, we recast the evaluation of the forces by considering volume integrals on the fluid domain Θ [176]. In particular, by considering an indicator function such as χ_{Θ_I} defined in Eq. (7.2.29) and following [177], we notice that the normal vector to the interface can be retrieved by taking the gradient of the indicator as [176, 177]:

$$\tilde{\mathbf{n}} = -\nabla \chi_{\Theta_I} \qquad \text{in } \Theta, \tag{7.2.48}$$

where the normal $\tilde{\mathbf{n}} : \Theta \to \mathbb{R}^d$ is extended to the whole domain Θ and non-zero only at the interface. This allows us to calculate the forces acting on the immersed obstacle by performing an integral on Θ, as:

$$\mathbf{F} = -\int_{\Theta} \left(-p\mathbf{I} + \frac{2}{\text{Re}} D(\mathbf{u}) \right) \nabla \chi_{\Theta_I} \, \mathrm{d}\Theta. \tag{7.2.49}$$

In practice, we always deal with the smooth regularizations of the indicator functions in Eqs. (7.2.35) and (7.2.36) defined with the signed distance describing the immersed object. Therefore, we calculate the approximated forces exert by the fluid on the obstacle as:

$$\mathbf{F} = -\int_{\Theta} \left(-p\mathbf{I} + \frac{2}{\text{Re}} D(\mathbf{u}) \right) \nabla \mathscr{H}_\varepsilon \, \mathrm{d}\Theta. \tag{7.2.50}$$

Then, by using Eq. (7.2.50), the drag and lift forces are derived as:

$$F_D = -\int_{\Omega} \left[-p \frac{\partial \mathscr{H}_\varepsilon}{\partial x} + \frac{1}{\text{Re}} \left(\frac{\partial u_x}{\partial x} \frac{\partial \mathscr{H}_\varepsilon}{\partial x} + \frac{\partial u_x}{\partial y} \frac{\partial \mathscr{H}_\varepsilon}{\partial y} \right) \right] \mathrm{d}\Omega \tag{7.2.51}$$

and

$$F_L = -\int_{\Omega} \left[-p \frac{\partial \mathscr{H}_\varepsilon}{\partial y} + \frac{1}{\text{Re}} \left(\frac{\partial u_y}{\partial x} \frac{\partial \mathscr{H}_\varepsilon}{\partial x} + \frac{\partial u_y}{\partial y} \frac{\partial \mathscr{H}_\varepsilon}{\partial y} \right) \right] \mathrm{d}\Omega, \tag{7.2.52}$$

respectively.

7.2.5 Space discretization of the fluid equations

We consider the fluid domain Θ to be represented by a NURBS volume, if $d = 3$, or a NURBS planar surface, if $d = 2$. We consider the NURBS-based IGA discretization of the Navier–Stokes equations (7.2.43) in the framework of the Galerkin method. To this aim, we introduce the finite dimensional function spaces for the velocity:

$$\begin{aligned} &V_h := V \cap [\mathcal{N}_h]^d, \\ &V_{R,h} := \left\{ \boldsymbol{\varphi}_h \in V_h : \quad \boldsymbol{\varphi}_h = \mathbf{g}_h \quad \text{on } \Gamma_D \right\}, \qquad V_{R,h} := \left\{ \boldsymbol{\varphi}_h \in V_h : \quad \boldsymbol{\varphi}_h = \mathbf{0} \quad \text{on } \Gamma_D \right\}, \end{aligned} \tag{7.2.53}$$

and the pressure:

$$W_h := W \cap \mathcal{N}_h, \tag{7.2.54}$$

where $\mathscr{N}_h$ is the NURBS function space holding the basis functions which define the geometrical mapping of Θ, following the isoparametric paradigm; $\mathbf{g}_h$ is a suitable approximation of $\mathbf{g}$ onto the NURBS space. However, the choice of function spaces in Eqs. (7.2.53) and (7.2.54) for discretizing problem (7.2.43) does not satisfy the Babuška–Brezzi condition [112, 178], since both the velocity and the pressure are expressed in terms of the same basis functions. Therefore, we adopt a stabilized formulation in order to guarantee the well-posedness of the problem. The semidiscrete problem reads:

$$\text{for a.e. } t \in (0,T), \text{ find } \mathbf{u}_h \in V_{R,h} \text{ and } p_h \in W_h \text{ such that}$$
$$\begin{cases} \displaystyle\int_\Theta \frac{\partial \mathbf{u}_h}{\partial t}\cdot\boldsymbol{\varphi}_h \,\mathrm{d}\Theta + \int_\Theta \mathbf{u}_h\cdot\nabla\mathbf{u}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta \frac{2}{\mathrm{Re}} D(\mathbf{u}_h):\nabla\boldsymbol{\varphi}_h\,\mathrm{d}\Theta \\ \qquad \displaystyle - \int_\Theta p_h\nabla\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta C_{R_h}\delta_\varepsilon\left(\mathbf{u}_h-\tilde{\mathbf{v}}_h\right)\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \mathscr{S}_M(\boldsymbol{\varphi}_h;\mathbf{u}_h,p_h) \\ \qquad \displaystyle = \int_{\Gamma_N} \mathbf{h}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Gamma_N & \forall\boldsymbol{\varphi}_h\in V_{0,h}, \\ \displaystyle\int_\Theta \psi_h\nabla\cdot\mathbf{u}_h + \mathscr{S}_C(\psi_h;\mathbf{u}_h,p_h) = 0 & \forall\psi_h\in W_h, \end{cases} \tag{7.2.55}$$

with $\tilde{\mathbf{v}}_h$ representing a suitable extension to Θ of the velocity of Ω. The additional stabilization terms $\mathscr{S}_M$ and $\mathscr{S}_C$ refer to the SUPG stabilization and will be described in Section 7.2.6. Problem (7.2.55) requires also compatible initial conditions:

$$\mathbf{u}_h(0) = \mathbf{u}_{0,h} \qquad \text{in } \Theta, \tag{7.2.56}$$

where $\mathbf{u}_{0,h}$ is the L^2-projection of the initial data $\mathbf{u}_0$ onto the NURBS function space $V_{R,h}$. Regarding the RIS term, the resistive parameter C_{R_h} depends on the mesh element size, and in particular it is proportional to $\frac{1}{h}$, in order to guarantee convergence of the pressure jump [92].

7.2.6 SUPG stabilization

Let $\mathbf{r}_M$ and $\mathbf{r}_C$ indicate the strong residuals of the momentum and continuity equations, reading, element by element:

$$\mathbf{r}_M(\mathbf{u}_h,p_h) := \frac{\partial\mathbf{u}_h}{\partial t} + \mathbf{u}_h\cdot\nabla\mathbf{u}_h - 2\nabla\cdot\left(\frac{1}{\mathrm{Re}}D(\mathbf{u}_h)\right) + \nabla p_h + C_{R_h}\delta_\varepsilon(\mathbf{u}_h-\tilde{\mathbf{v}}_h) \tag{7.2.57}$$

and

$$r_C(\mathbf{u}_h) := \nabla\cdot\mathbf{u}_h, \tag{7.2.58}$$

respectively. The SUPG stabilization technique [93, 94, 95] consists in adding the following term to the semidiscrete momentum equation:

$$\mathscr{S}_M(\boldsymbol{\varphi}_h; \mathbf{u}_h, p_h) := \sum_{K\in\mathscr{K}_h} \left[\int_K \tau_M^K(\mathbf{u}_h)\mathbf{r}_M(\mathbf{u}_h, p_h)\mathbf{u}_h \cdot \nabla\boldsymbol{\varphi}_h dK + \int_K \tau_C^K(\mathbf{u}_h) r_C(\mathbf{u}_h)\nabla\cdot\boldsymbol{\varphi}_h dK \right] \tag{7.2.59}$$

and to the semidiscrete continuity equation:

$$\mathscr{S}_C(\psi_h; \mathbf{u}_h, p_h) := \sum_{K\in\mathscr{K}_h} \int_K \tau_M^K(\mathbf{u}_h)\mathbf{r}_M(\mathbf{u}_h, p_h)\cdot\nabla\psi_h dK. \tag{7.2.60}$$

These stabilization terms are evaluated and added element-wise over the computational mesh $\mathscr{K}_h$. τ_M^K and τ_C^K represent stabilization parameters. In order to evaluate these parameters, we consider the quantities:

$$\bar{G}_{ij} = \sum_{k=1}^{d} \frac{\partial\xi_k}{\partial X_i}\frac{\partial\xi_k}{\partial X_j} \qquad \text{for } i = 1,\dots,d,\ j = 1,\dots,d, \tag{7.2.61}$$

and

$$\bar{g}_i = \sum_{j=1}^{d} \frac{\partial\xi_j}{\partial X_i}, \qquad \text{for } i = 1,\dots,d, \tag{7.2.62}$$

where $\mathbf{X}$ is the NURBS geometrical mapping representing Θ and ξ_i, for $i = 1,\dots,d$, refers to the i-th parametric direction. Then, the parameter τ_M^K is calculated as [95]:

$$\tau_M^K = \left[\frac{C_t^2}{\Delta t^2} + \mathbf{u}_h\cdot\bar{\mathbf{G}}\mathbf{u}_h + C_I\frac{1}{\mathrm{Re}^2}\bar{\mathbf{G}}:\bar{\mathbf{G}} + C_{R_h}^2\delta_\varepsilon^2 \right]^{-\frac{1}{2}}, \tag{7.2.63}$$

where $C_I \in \mathbb{R}$ is a constant independent of the mesh size but dependent on the degree p of the NURBS basis functions, which can be obtained from an element-wise inverse estimate [94, 179]:

$$C_I = 60\cdot 2^{p-2}, \tag{7.2.64}$$

and, in anticipation of the discretization in time that will be introduced in Section 7.2.7, $\Delta t \in \mathbb{R}$ is the time step size and $C_t \in \mathbb{R}$ is equal to the order of the time discretization chosen. We highlight the dependence of the stabilization parameter τ_M^K in Eq. (7.2.63) on the resistive constant and the smoothed indicator function δ_ε, stemming from the RIS approach [92]. Finally, the stabilization parameter τ_C^K reads [95]:

$$\tau_C^K(\mathbf{u}_h) = \frac{1}{\tau_M^K(\mathbf{u}_h)\,\bar{\mathbf{g}}\cdot\bar{\mathbf{g}}}. \tag{7.2.65}$$

We remark that the stabilization parameters in the limit of $\Delta t \to 0$ may become degenerate [95].

7.2.7 Time discretization of the fluid equations

Regarding the time discretization of Eqs. (7.2.55) stabilized with the SUPG terms (7.2.59) and (7.2.60), we employ, as introduced in Chapter 5, a BDF scheme of order k. We adopt a semi-implicit formulation: as in [92, 159] we extrapolate in time the convective term as well as the stabilization parameters of the SUPG stabilization, with an extrapolation compatible with the BDF scheme adopted (see Section 5.2).

Let the interval $(0, T)$ be divided into N time steps of size Δt, with time instances $t_i = i\Delta t$, for $i = 0, \dots, N \in \mathbb{N}$, so that $t_0 = 0$ and $t_N = T$. The fully discrete Navier–Stokes problem, discretized in space with NURBS-based IGA and in time with BDF schemes, then reads:

for $n = k, \dots, N-1$, find $\mathbf{u}_h^{n+1} \in V_{R,h}$ and $p_h^{n+1} \in W_h$ such that

$$\begin{cases} \displaystyle\int_\Theta \frac{\alpha_0}{\Delta t}\mathbf{u}_h^{n+1}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta \mathbf{u}_h^*\cdot\nabla\mathbf{u}_h^{n+1}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta \frac{2}{\mathrm{Re}}D(\mathbf{u}_h^{n+1}):\nabla\boldsymbol{\varphi}_h\,\mathrm{d}\Theta \\ \displaystyle\qquad -\int_\Theta p_h^{n+1}\nabla\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta C_{R_h}\delta_\varepsilon\left(\mathbf{u}_h^{n+1}-\tilde{\mathbf{v}}_h\right)\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta \\ \displaystyle\qquad + \sum_{K\in\mathscr{K}_h}\int_K \tau_M^K(\mathbf{u}_h^*)\tilde{\mathbf{r}}_M(\mathbf{u}_h^{n+1},p_h^{n+1})\cdot\left(\mathbf{u}_h^*\cdot\nabla\boldsymbol{\varphi}_h\right)dK \\ \displaystyle\qquad + \sum_{K\in\mathscr{K}_h}\int_K \tau_C^K(\mathbf{u}_h^*)r_C(\mathbf{u}_h^{n+1})\nabla\cdot\boldsymbol{\varphi}_h dK \\ \displaystyle\qquad = \int_\Theta \frac{\alpha_0}{\Delta t}\mathbf{u}_h^{bdf,n}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_{\Gamma_N}\mathbf{h}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Gamma_N \qquad\qquad \forall\boldsymbol{\varphi}_h\in V_{0,h}, \\ \displaystyle\int_\Theta \psi_h\nabla\cdot\mathbf{u}_h^{n+1} + \sum_{K\in\mathscr{K}_h}\left[\int_K \tau_M^K(\mathbf{u}_h^*)\mathbf{r}_M(\mathbf{u}_h^{n+1},p_h^{n+1})\cdot\nabla\psi_h dK\right] = 0 \qquad \forall\psi_h\in W_h, \end{cases} \tag{7.2.66}$$

where the residual $\tilde{\mathbf{r}}_M$ of the fully discrete momentum equation reads:

$$\tilde{\mathbf{r}}_M := \frac{\alpha_0}{\Delta t}\left(\mathbf{u}_h^{n+1}-\mathbf{u}_h^{bdf,n}\right)+\mathbf{u}_h^*\cdot\nabla\mathbf{u}_h^{n+1}-2\nabla\cdot\left(\frac{1}{\mathrm{Re}}D\left(\mathbf{u}_h^{n+1}\right)\right)+\nabla p_h^{n+1}+C_{R_h}\delta_\varepsilon\left(\mathbf{u}_h^{n+1}-\tilde{\mathbf{v}}_h\right) \tag{7.2.67}$$

The system of Eq. (7.2.66) needs appropriate initial conditions, reflected in the term $\mathbf{u}_{0,h}^{bdf,0}$. since the BDF schemes of order $k > 1$ are multi-step methods, k initial conditions are formally required:

$$\mathbf{u}_h^n = \tilde{\mathbf{u}}_n \qquad \text{in } \Theta, \text{ for } n = 0, \dots, k-1, \tag{7.2.68}$$

where $\tilde{\mathbf{u}}_n$ is the L^2-projection of the initial data $\mathbf{u}_n$ onto the NURBS function space $V_{R,h}$, for $n = 0, \dots, k-1$; then, $\mathbf{u}_{0,h}^{bdf,0}$ is obtained by using Eq. (5.2.7). Finally, we remark that, due to the semi-implicit approach, one linear system has to be solved at each time step.

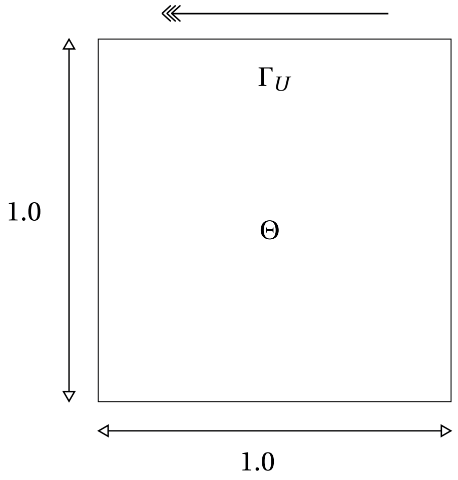


Figure 7.4 – Lid-driven cavity case. Sketch of the setup.

7.2.8 Numerical results: lid-driven cavity

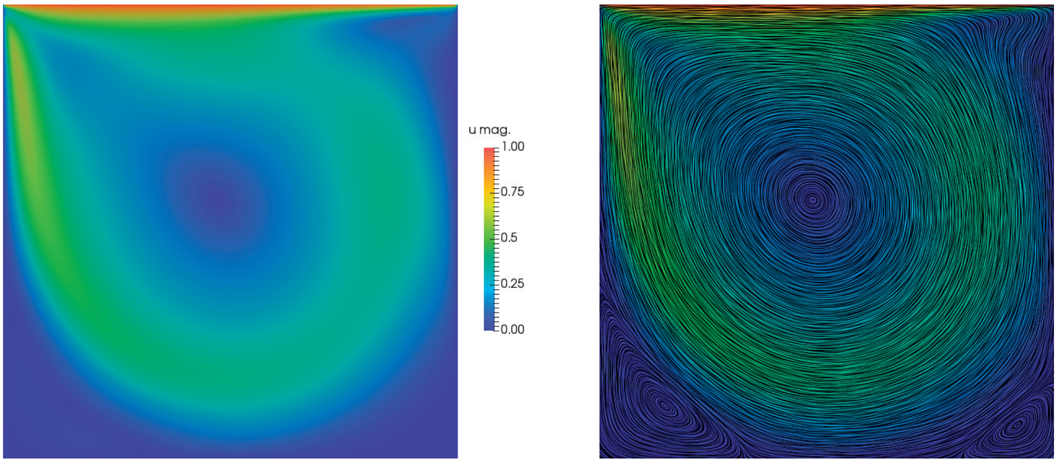
As a test for the numerical approximation of the fluid equations, we consider the lid-driven cavity problem [180]. It is a well known test case used for testing discretization methods and numerical schemes for solving the Navier–Stokes equations. Several authors have reported results obtained with different numerical approaches [180, 181, 182, 183]; for results obtained with IGA-based discretizations, we refer to [40, 184].

The setup is sketched in Figure 7.4. The problem is defined in the two-dimensional quadrangular cavity domain of size $L = 1$, i.e. $\Theta = (0,1)^2$. The topmost wall moves with a prescribed velocity equal to $U = 1$ in the direction highlighted by the arrow, which reflects in a condition on the velocity of the fluid. On the other walls, no-slip conditions are considered. Thus, the following conditions are imposed:

$$\begin{aligned} &\mathbf{u} = -U\mathbf{e}_x && \text{on } \Gamma_U,\ t \in (0,T), \\ &\mathbf{u} = \mathbf{0} && \text{on } \partial\Theta \setminus \Gamma_U,\ t \in (0,T). \end{aligned} \tag{7.2.69}$$

The Reynolds number is set to Re = 1000. We are interested in the results at steady state. Since we approximate the unsteady Navier–Stokes problem, we let the simulation go from an initial condition of $\mathbf{u} = \mathbf{0}$ everywhere to a final time of $T = 100$. The computational domain is represented by means of NURBS of degree $p = 2$ and C^1-continuous in a 128×128 grid of elements, yielding a total of 16,384 elements. The problem is discretized in time by employing a BDF scheme of order $k = 2$ with a fixed time step size of $\Delta t = 0.1$. For each time step, the resulting linear system is composed of 66,567 DOFs and is solved by using the GMRES method with MUMPS preconditioner [185, 186].

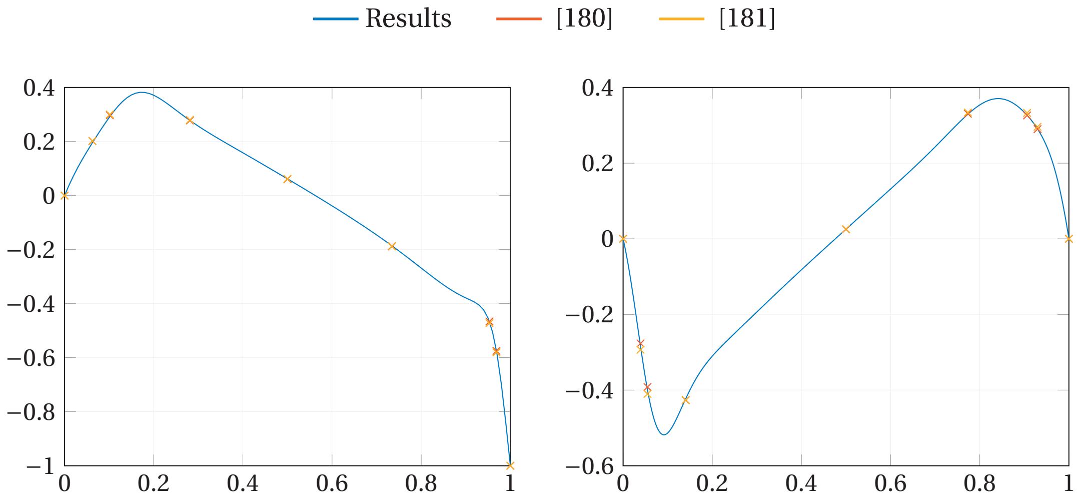
We report in Figure 7.5 the approximated velocity field at $t = 100$. The solution obtained at final time is compared against values reported in the literature in specified points of the domain.



(a) Velocity magnitude

(b) LIC

Figure 7.5 – Lid-driven cavity case. Approximated velocity field in (a) and its line integral convolution (LIC) in (b), at $t = 100$.

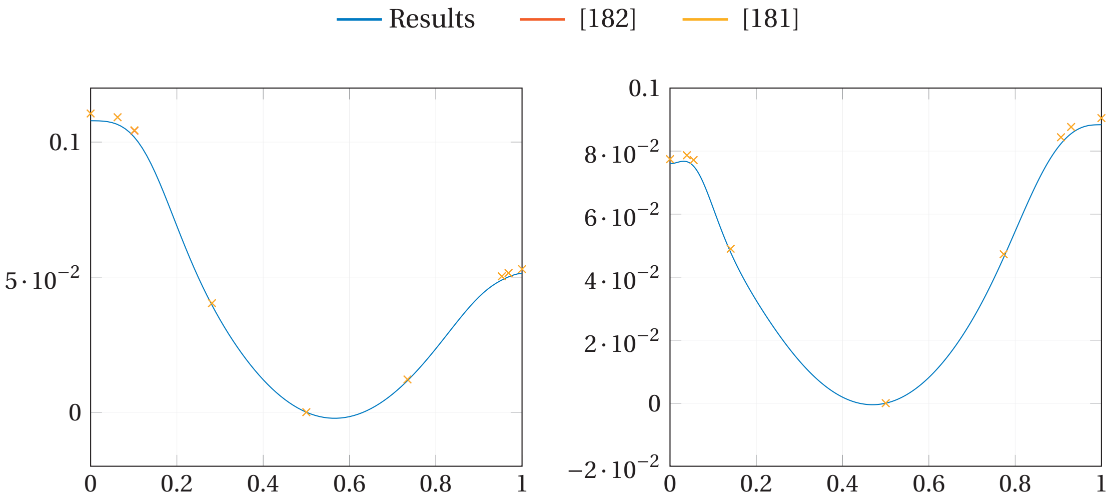


(a) u_x at vert. centerline

(b) u_y at horiz. centerline

Figure 7.6 – Lid-driven cavity case. Horizontal velocity at the vertical centerline (a) and vertical velocity at the horizontal centerline (b), compared against [180] and [181].

As references, we consider the papers [180] and [181], where discretizations based on finite differences and multigrid solvers are employed, and [182], where the authors used a spectral Chebyshev collocation method. In particular, we extract the horizontal velocity u_x and the pressure p in 9 points on the vertical centerline (i.e. on $x = 0.5$) and compare them against the values reported in the references in Table 7.1. We do the same with the vertical velocity u_y and the pressure p in 9 points on the horizontal centerline (i.e. on $y = 0.5$), reporting the values in Table 7.2.



(a) p at vert. center line

(b) p at horiz. center line

Figure 7.7 – Lid-driven cavity case. Pressure at the vertical centerline (a) and at the horizontal centerline (b), compared against [182] and [181].

By comparing the results, we find agreement with the references reported. The horizontal velocity at the vertical center line and vertical velocity at the horizontal center line are shown in Figure 7.6. The pressure, while still being comparable, presents higher differences with the reference results. This could be due to different reasons. First of all, we highlight that the mesh employed is uniform, thus not refined near the boundaries. Moreover, we consider the classical weak formulation in Eq. (7.2.55) with the velocity $\mathbf{u}$ and pressure p as primitive variables approximated using the pair of function spaces (7.2.53) and (7.2.54) which do not satisfy the Babuška–Brezzi condition, as stated in Section (7.2.5), for which we add the SUPG stabilization terms to the formulations. Finally, the imposed Dirichlet boundary conditions on the velocity shows discontinuities in the top corners of the cavity, leading to the vorticity and the pressure being singular in these two points [187]. In this work there is no special treatment for controlling the impact of the singularities on the numerical results; if more accuracy is required, several approaches could be considered, from the simple smoothing of the Dirichlet datum (see e.g. [188]), to the more complex singularity subtraction technique [183]. Nevertheless, with these considerations in mind, we consider the results to be sufficiently accurate for our purposes. Our results are also in agreement with the ones reported for B-splines IGA-based discretizations in the papers [40] and [184] based, respectively, on a scalar streamfunction formulation and the use of divergence-conforming B-splines, which lead to the mass being conserved by construction, for both the cases. Minimum and maximum values of the velocity at the centerlines are reported in Table 7.3; we remark that, in order to perform the comparison, these values refer to the numerical approximation of the lid-driven cavity problem with reversed velocity of the top wall with regard to the setup in Figure 7.4, i.e. with the condition $\mathbf{u} = U\,\mathbf{e}_x$ on Γ_U, with $U = 1$.

y	u_x [180]	u_x [182]	u_x [181]	u_x	p [182]	p [181]	p
1.0000	−1.0000	−1.0000	−1.0000	−1.0000	0.0530	0.0530	0.0513
0.9688	−0.5749	−0.5808	−0.5803	−0.5738	0.0515	0.0515	0.0501
0.9531	−0.4660	−0.4723	−0.4724	−0.4666	0.0503	0.0503	0.0489
0.7344	−0.1872	−0.1887	−0.1886	−0.1860	0.0121	0.0121	0.0117
0.5000	0.0608	0.0621	0.0621	0.0620	0.0000	0.0000	0.0000
0.2813	0.2781	0.2804	0.2804	0.2782	0.0404	0.0404	0.0396
0.1016	0.2973	0.3004	0.3003	0.2925	0.1042	0.1044	0.1018
0.0625	0.2020	0.2023	0.2023	0.1956	0.1092	0.1092	0.1065
0.0000	0.0000	0.0000	0.0000	0.0000	0.1106	0.1106	0.1079

Table 7.1 – Lid-driven cavity case. Horizontal velocity and pressure values in points along the vertical centerline.

x	u_y [180]	u_y [182]	u_y [181]	u_y	p [182]	p [181]	p
0.0000	0.0000	0.0000	0.0000	0.0000	0.0775	0.0774	0.0761
0.0391	−0.2767	−0.2937	−0.2933	−0.2843	0.0787	0.0787	0.0766
0.0547	−0.3919	−0.4104	−0.4102	−0.3994	0.0772	0.0771	0.0752
0.1406	−0.4266	−0.4264	−0.4263	−0.4225	0.0490	0.0490	0.0478
0.5000	0.0253	0.0258	0.0258	0.0255	0.0000	0.0000	0.0000
0.7734	0.3307	0.3340	0.3340	0.3295	0.0473	0.0473	0.0462
0.9062	0.3263	0.3330	0.3329	0.3268	0.0844	0.0844	0.0824
0.9297	0.2901	0.2963	0.2962	0.2904	0.0877	0.0876	0.0855
1.0000	0.0000	0.0000	0.0000	0.0000	0.0905	0.0904	0.0884

Table 7.2 – Lid-driven cavity case. Vertical velocity and pressure values in points along the horizontal centerline.

7.2.9 Numerical results: flow past cylinder

We now consider another benchmark problem: the flow past a cylinder at Re = 100 [189]. The setup of the problem is sketched in Figure 7.8. The domain consists in a rectangular channel of height $H = 0.41$ and length $L = 2.2$, with a cylindrical obstacle Ω of diameter $D = 0.1$ centered in $(x_c, y_c) \in \mathbb{R}^2$, $(x_c, y_c) = (0.2, 0.21)$. The fluid flows through the inlet Γ_I at the left wall towards the outlet Γ_O at the right wall. At the inlet a parabolic velocity profile is prescribed:

$$\mathbf{g} = \frac{4}{0.41^2}\overline{U}y(0.41 - y)\,\mathbf{e}_x \qquad y \in (0, 0.41), \tag{7.2.70}$$

where $\overline{U}$ is the mean inflow velocity. On the bottom and top walls no-slip conditions are prescribed, while on the outlet a homogeneous natural condition is considered. Thus, the

	results	[40]	[184]	[180]
min u_x	−0.3824	−0.3903	−0.3902	−0.3829
min u_y	−0.5183	−0.5287	−0.5288	−0.5155
max u_y	0.3708	0.3787	0.3786	0.3710

Table 7.3 – Lid-driven cavity case. Velocity values computed at the centerlines of the cavity at Re = 1000 and comparison against the results obtained with mesh size $h = 1/128$ in [40], using B-splines of degree $p = 2$, and in [184], using B-splines of degree $p = 1$; the last column reports the classical results of [180].

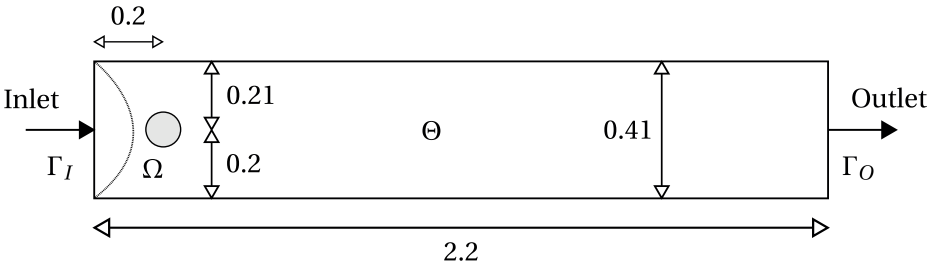


Figure 7.8 – Flow past cylinder. Sketch of the setup.

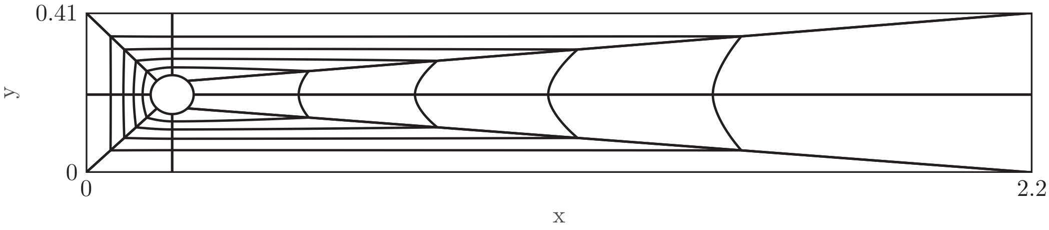
following boundary conditions are enforced on the boundary of the fluid domain Θ:

$$\begin{aligned} \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_I,\ t \in (0,T), \\ \boldsymbol{\sigma}\mathbf{n} &= \mathbf{0} && \text{on } \Gamma_O,\ t \in (0,T), \\ \mathbf{u} &= \mathbf{0} && \text{on } \partial\Theta \setminus (\Gamma_I \cup \Gamma_O),\ t \in (0,T), \end{aligned} \tag{7.2.71}$$

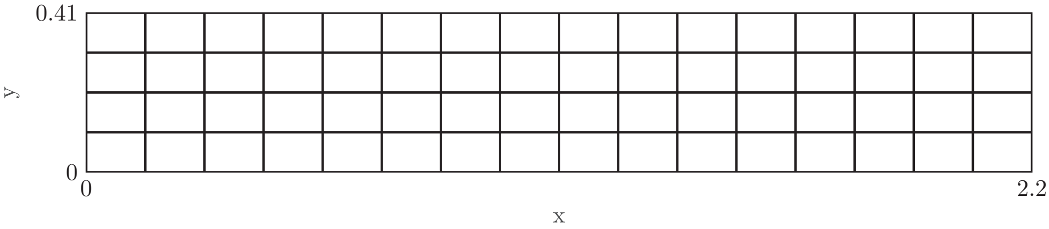
where $\mathbf{n}$ is the outward pointing unit normal vector at the outlet Γ_O. The fluid flowing into the channel has density $\rho = 1$ and kinematic viscosity $\mu = 10^{-3}$. By considering the diameter of the obstacle as characteristic length, the Reynolds number, defined for this problem as Re = $\rho U D/\mu$, is chosen to be equal to Re = 100, with a characteristic velocity of $U = 1$ and mean inflow velocity of $\overline{U} = \frac{3}{2}U = 1.5$.

This problem is widely used for assessing and comparing the performance of discretization techniques and numerical solvers for the Navier–Stokes equations. For example, results obtained with several numerical methods on boundary fitted discretizations are available for comparison in [189, 190, 191]; in [177], the fictitious domain method has instead being used. In the domain of IGA-based discretizations, we cite the work [192], where Taylor–Hood B-spline elements are employed, and [193] for results obtained with NURBS-based IGA.

We consider two approaches for simulating the problem. In the first approach, a NURBS parametrization of the domain is considered such that it represents exactly the "boundary-fitted" geometry with the hole. The computational domain is shown, at a coarse level of refinement, in Figure 7.9b. The geometrical mapping is highly stretched and skewed, since we consider



(a) Boundary fitted mesh (single NURBS patch)



(b) Mesh used with the resistive approach

Figure 7.9 – Flow past cylinder. Computational NURBS meshes considered in the boundary fitted case (a) and with the resistive approach (b).

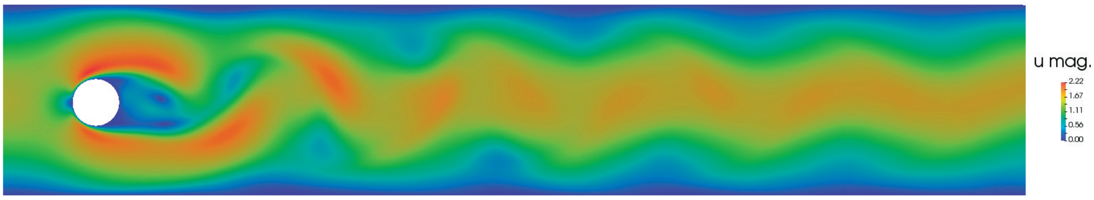
only single-patch NURBS mappings; nevertheless, it represents and fits the boundaries exactly. We remark that, in order to represent exactly the circular hole, it is necessary to employ NURBS basis functions of at least degree $p = 2$. For different parametrizations of this geometry, involving multiple patches, see e.g. [193]. The considered boundary fitted mesh is composed of 33,927 elements and NURBS basis functions of degree $p = 2$ and C^1-continuous everywhere (even across the line from the hole to the outlet corresponding to the "folded" boundary of the parametric domain, by using the same approach described in Section 3.3.1 for building periodic function spaces). The amount of DOFs of the discrete system to solve at each time step is 136,888.

The second approach is based on the RIS method described in Section 7.2.3. The fluid domain consists in the rectangular channel $\Theta = (0, 2.2) \times (0, 0.41)$ parametrized as a regular Cartesian grid, as shown in Figure 7.9b. The circular obstacle is seen by the fluid as a resistive immersed surface, described by the following signed distance function $d : \mathbb{R}^2 \to \mathbb{R}$:

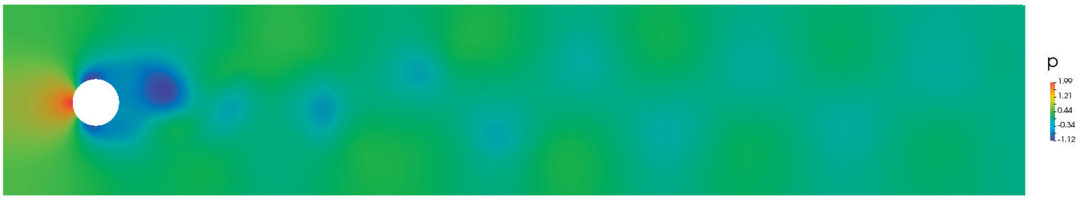
$$d(x, y) = \sqrt{(x - x_c)^2 + (y - y_c)^2} - \frac{D}{2} \qquad \text{for } (x, y) \in \Theta. \tag{7.2.72}$$

In this example, the resistive constant is set to $C_{R_h} = \frac{5000}{h}$ and $\varepsilon = 2h$. The Cartesian mesh is built of 256 × 64 elements, for a total of 16,384 elements and 83,916 DOFs, with NURBS basis functions of degree $p = 2$ and globally C^1-continuous. The PDEs are discretized in time by a BDF scheme of order $k = 2$ with time step size set to $\Delta t = 0.001$. In both the cases, the velocity profile at the inlet is prescribed via L^2-projection of the data onto the NURBS function space.

We report in Figure 7.10 the velocity magnitude together with the pressure field at time $t = 7$

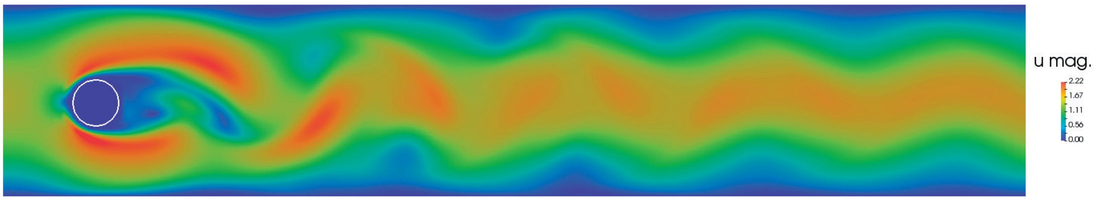


(a) Velocity

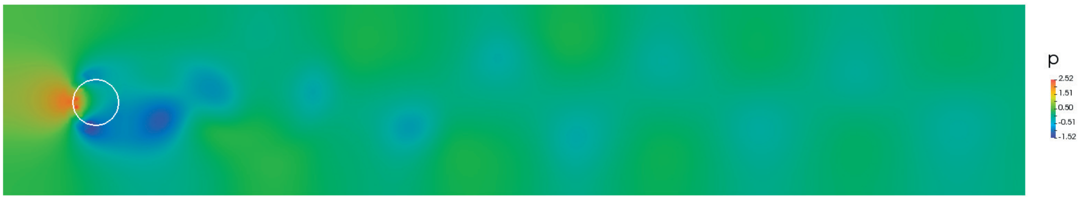


(b) Pressure

Figure 7.10 – Flow past cylinder. Velocity (a) and pressure (b) fields at time $t = 7$ obtained on the boundary fitted mesh.

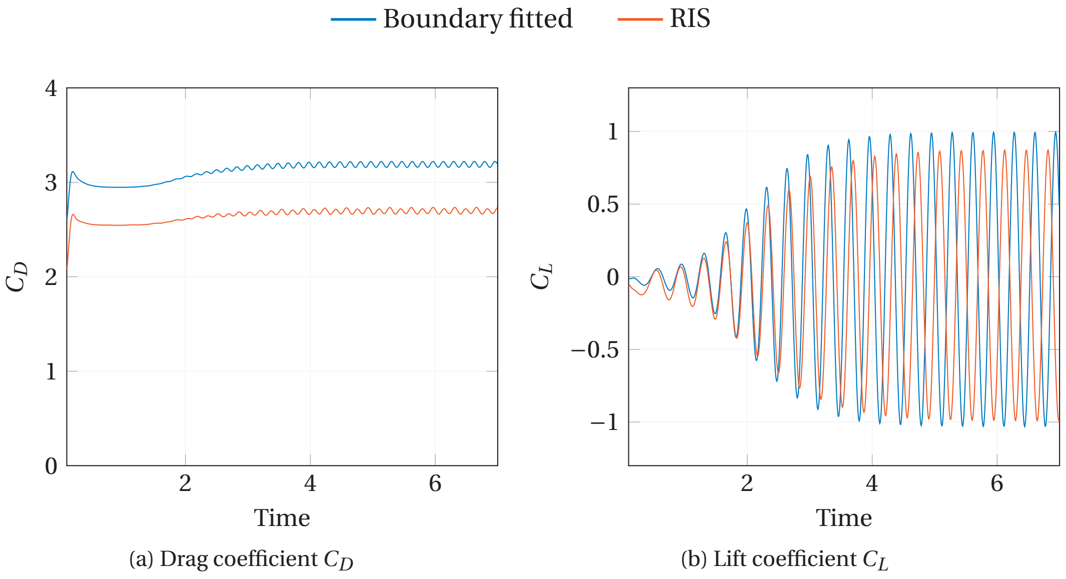


(a) Velocity



(b) Pressure

Figure 7.11 – Flow past cylinder. Velocity (a) and pressure (b) fields at time $t = 7$ obtained with the resistive approach.



(a) Drag coefficient C_D

(b) Lift coefficient C_L

Figure 7.12 – Flow past cylinder. Drag and lift coefficients obtained on the boundary fitted mesh as well as using the resistive approach.

computed on the boundary fitted mesh. In Figure 7.11 the velocity magnitude and pressure field at time $t = 7$ computed with the RIS approach on the Cartesian mesh are shown. The two velocity profiles refer to the same time instance; they differ because the vortex shedding approximated with the two approaches is not synchronized. In Tables 7.4 and 7.5 we report the minimum value, maximum value, mean and amplitude of the oscillations of the drag coefficient C_D and lift coefficient C_L, respectively. We compare the coefficients obtained on the boundary fitted mesh and with the RIS approach on the Cartesian mesh as well as against values reported in the literature, in particular from [192], where IGA with Taylor–Hood B-spline elements has been employed, and from [194], based on a FE discretization with Q_2/P_1^{disc} elements. The numerical approximation using the boundary fitted mesh is in agreement with the sources considered for the comparison. We highlight that, by employing different parametrizations of the geometry based on multi-patch NURBS, better accuracy could be achieved; as stated also in [195] for scalar convection-diffusion equations, the parametrization of the domain plays an important role with regard to the accuracy of the results, also confirmed for the flow past cylinder benchmark problem in [193], where different NURBS-based multi-patch parametrizations have been tested. Nevertheless, the results on the boundary fitted mesh are a valid reference for the comparison against the results obtained with the RIS approach.

In this regard, in Figures 7.12a and 7.12b the evolutions of the drag and lift coefficients computed on the boundary fitted mesh as well as with the RIS method on the Cartesian mesh are shown. It is evident how, with the RIS method, the qualitative behavior of the solution is reproduced. Regarding the drag coefficient, we obtain an error of about 15% on the mean value and 13% on the amplitude of the oscillations, while for the lift coefficient the mean value is particularly offset (of around 90%), but the error in the amplitude of the oscillations is 8%. In contrast, we obtain a better approximation of the Strouhal number, reported in Table 7.6, and being calculated as:

$$\mathrm{St} = \frac{fD}{U}, \tag{7.2.73}$$

where $f \in \mathbb{R}$ is the frequency of the vortex shedding; the error on the Strouhal number is below 0.1%.

In general, the forces and the drag and lift coefficients are very sensitive to the discretization of the immersed interface and to the accuracy in enforcing the no-slip condition on the obstacle. We remark that the computational mesh considered (shown in Figure 7.9b) is not refined around the hole, as it should be when high accuracy is wanted [177, 192, 193]. The choice of not refining the mesh around the obstacle stems from the fact that we aim at assessing the behavior of the method with a coarse discretization. Indeed, when considering a vesicle inside a containing domain, this is free to move and deform; therefore, a priori its location and shape are unknown. This issue can be overcome by employing, for example, adaptive discretizations using hierarchical basis functions [196, 197], or adaptive quadrature rules in the mesh elements intersecting the boundary of the immersed object, as done in the Finite Cell Method [198, 199].

	min. C_D	max. C_D	mean C_D	amp. C_D
RIS	2.6636	2.7345	2.6955	0.0708
bound. fit.	3.1581	3.2207	3.1895	0.0626
[192]	3.1665	3.2300	3.1983	0.0635
[194]	3.1643	3.2274	3.1958	0.0631

Table 7.4 – Flow past cylinder. Minimum value, maximum value, mean, and amplitude of the oscillations of the drag coefficient C_D.

	min C_L	max C_L	mean $\lvert C_L\rvert$	amp. C_L
RIS	-0.9908	0.8721	0.0298	1.8630
bound. fit.	-1.0307	0.9945	0.0157	2.0252
[192]	-1.0242	0.9893	0.0175	2.0135
[194]	-1.0213	0.9866	0.0174	2.0079

Table 7.5 – Flow past cylinder. Minimum value, maximum value, mean, and amplitude of the oscillations of the lift coefficient C_L.

	$1\backslash f$	St
RIS	0.3220	0.3036
bound. fit.	0.3213	0.3033
[192]	0.3300	0.3030
[194]	0.3313	0.3019

Table 7.6 – Flow past cylinder. Computed period of the vortex shedding and Strouhal number.

7.3 Signed distance and resistive method with NURBS

The penalization method introduced in Section 7.2.3 to weakly enforce the condition (7.2.20) on the continuity of the velocities at an immersed interface relies on a signed distance function $d : \Theta \to \mathbb{R}$ to define the immersed shape. For some geometries, defined through simple expressions, the associated signed distance function can be expressed explicitly, e.g. for the disk defined in Eq. (7.2.40). However, for more complex geometries the signed distance functions cannot be written as closed expressions and the construction of such functions is not a straightforward task. The problem of calculating the signed distance field d generated by Ω on Θ consists in:

For all points $\mathbf{p} \in \Theta$:

- Find the minimum distance between $\mathbf{p}$ and the interface Ω by solving the problem:

$$\begin{aligned} &\text{find } \operatorname{dist}(\mathbf{p}) \in \mathbb{R} \text{ such that} \\ &\quad \operatorname{dist}(\mathbf{p}) = \inf_{\mathbf{y}\in\Omega} \left\| \mathbf{y} - \mathbf{p} \right\|, \end{aligned} \tag{7.3.1}$$

  where $\| \cdot \|$ refers to the Euclidean distance in $\mathbb{R}^d$.
- Check if $\mathbf{p}$ is inside or outside the object whose Ω is boundary and change the sign of the obtained distance accordingly:

$$d(\mathbf{p}) = \begin{cases} -\operatorname{dist}(\mathbf{p}) & \text{if inside,} \\ \operatorname{dist}(\mathbf{p}) & \text{if outside.} \end{cases} \tag{7.3.2}$$

Problem (7.3.1), also known as the *minimum distance problem*, is a very ubiquitous and studied topic [200]. For instance, in the domain of computational geometry it appears when considering curve and surface fitting [201]; for physics engines in videogames it represents a core part of the collision detection algorithms [202]; it is also the main problem behind the interactive selection of geometries in CAD software [200] and for ray-tracing [203, 204, 205, 206, 207]. While a common framework for the minimum distance computational problem can be devised [200], the algorithms used to solve problem (7.3.1) are very specific and tailored to the kind of geometry considered. Besides the simple geometrical shapes, usually two classes of geometries are distinguished: polygonal models and parametric models. Polygonal models, for example, represent a common way to describe or approximate complex geometries in realtime computer graphics, in a manner that is numerically easy to handle and process, especially with specialized acceleration hardware as GPUs [208]. Parametric models, as NURBS, are more common in the realm of CAD design, for their capability of accurately representing complex realistic shapes and their malleability and properties related to geometrical modeling [36].

Our focus in this thesis is restricted to parametric geometries, in particular described by NURBS mappings. Let us consider a curve $\mathbf{C} : \widehat{\Omega} \to \mathbb{R}^d$, defined over the parametric domain $\widehat{\Omega} \subset \mathbb{R}$. Given a point $\mathbf{p} \in \mathbb{R}^d$, we seek the projection of the point onto the curve, i.e. we seek the parametric coordinate $\xi \in \widehat{\Omega}$ of the point $\mathbf{q} \in \mathbb{R}^d$ such that $\mathbf{q} = \mathbf{C}(\xi)$, i.e. lying on the curve $\mathbf{C}$, that is perpendicular to the tangent vector of $\mathbf{C}$ in $\mathbf{q}$ [209]. The problem of projecting a given point $\mathbf{p} \in \mathbb{R}^d$ onto the curve $\mathbf{C}$ reads:

$$\begin{aligned} &\text{find } \xi \in \widehat{\Omega} \text{ such that} \\ &\quad f(\xi) = \big(\mathbf{C}(\xi) - \mathbf{p}\big) \cdot \mathbf{C}'(\xi) = 0, \end{aligned} \tag{7.3.3}$$

The function f can have multiple roots, each corresponding to a projection of $\mathbf{p}$ onto the curve. Therefore, to find the minimum distance of $\mathbf{p}$ to the curve, the distance between $\mathbf{p}$ and each of the projected points $\mathbf{C}(\xi)$, with ξ solution of problem (7.3.3), must be calculated and the minimal distance selected. A formulation similar to (7.3.3) can be derived for parametric surfaces [37].

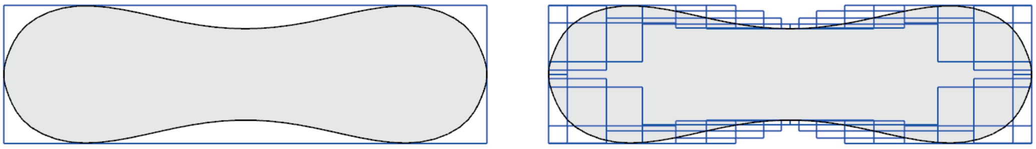
To find the roots of f, the Newton–Raphson method is usually employed [202, 206, 209, 210]. However, this iterative method relies on an appropriate choice of initial value in order to achieve convergence. This is even more crucial when projecting points onto self-intersecting NURBS geometries [211] or when dealing with points near the boundaries of open surfaces [201]. To obtain suitable candidate points, several authors decompose the NURBS geometries into rational Bézier subcurves or patches, to exploit the properties of the resulting control polygons [211], or in a recursive subdivision process [201, 212], with special criteria to accelerate the process by excluding non-necessary parts of the geometry [213, 214]. Alternatively, by using flattening procedures, based on adaptive subdivision or curvature-based refinement [203, 204], it is possible to extract a small part of the geometry, in which the projected point lies, which is locally approximately flat; then, it can be approximated as a polygon and standard ray-against-polygon methods can be employed [203].

Our approach in solving the projection problem (7.3.3) with NURBS geometries is based on the Newton–Raphson method as in [37], with selection of the candidate points accelerated by using a bounding volume hierarchy. We describe our approach in selecting the candidate points in Section 7.3.1; in Sections 7.3.2 and 7.3.3 the algorithms to project a point onto NURBS curves and surfaces are described; finally, in Sections 7.3.4 and 7.3.5 we apply this approach to the penalization method described in Section 7.2.3.

7.3.1 Selection of candidate points

As already stated, the selection of good initial guesses is crucial for the Newton–Raphson method to converge. Moreover, the iterative algorithm that refines the initial candidate point towards the final projected point involves costly evaluations of the NURBS points and derivatives. Therefore, an important aspect of the method for selecting the candidate points is the efficiency, considering also that the signed distance (7.3.2) is evaluated in the whole domain Θ when used to build the penalization term for the Navier–Stokes equations described in Section 7.2.3.

Before describing our approach, we introduce the concept of bounding volume and bounding



(a) AABB of the whole geometry (b) AABBs of the elements

Figure 7.13 – Axis aligned bounding boxes of an example NURBS closed curve in (a) and of its elements in (b).

volume hierarchy. Given a set of n_p points $\left\{\mathbf{p}^i\right\}$, $\mathbf{p}^i \in \mathbb{R}^d$, a *bounding volume* $\mathcal{V} \subset \mathbb{R}^d$ is a region of space such that $\mathbf{p}^i \in \mathcal{V}$, for $i = 1,\dots,n_p$, i.e. such that it encloses all the points of the set considered. A *bounding volume hierarchy* is a hierarchy of bounding volumes organized in a tree such that each node is a bounding volume which encloses completely the bounding volumes of all its children. Bounding hierarchies are often used as space partitioning methods to speed up geometrical algorithms [204]. Several choices for the kind of bounding volume to adopt at each node of the hierarchy can be considered, the most adopted being, for instance, spheres, axis aligned boxes, oriented boxes, trapezoidal prisms, or convex hulls. This choice is driven by the trade-off between tightness of fit of the set of points and efficiency in the construction of the hierarchy and in the intersection tests of points (or other geometric primitives) against the tree.

We consider the use of *axis aligned bounding boxes* (AABB) [215]. An AABB is characterized by two vectors $\mathbf{b}^{min}, \mathbf{b}^{max} \in \mathbb{R}^d$. Given the set of points $\left\{\mathbf{p}^i\right\}_{i=0}^{n_p-1}$, the AABB is such that:

$$b_j^{min} \leq p_j^i \leq b_j^{max} \qquad \text{for } j = 1,\dots,d, \quad \text{for } i = 1,\dots,n_p, \tag{7.3.4}$$

i.e. the AABB describes the tightest region of $\mathbb{R}^d$ aligned with the Cartesian axes which encloses the points $\left\{\mathbf{p}^i\right\}$. The choice of using AABBs as bounding volumes for the bounding hierarchy stems from the efficiency in performing intersection tests and in building the hierarchy [204].

We remark also the *strong convex hull* property of B-spline and NURBS geometries [37] in Property 1.

Property 1. *Following the notation of Section 3.1, let us consider a NURBS curve* $\mathbf{C} : \widehat{\Omega} \to \mathbb{R}^d$, $\widehat{\Omega} \subset \mathbb{R}$, *of degree* p *defined by the knot vector* $\Xi = \{\xi_i\}_{i=1}^n \in \mathbb{R}^n$, *with* $n \in \mathbb{N}$, $n \geq 2$, *and by the control points* $\mathbf{P}_i \in \mathbb{R}^d$, $i = 1,\dots,n_{bf}$, *with* $n_{bf} = n - p - 1$. *By considering a given* $\xi \in \mathbb{R}$ *such that* $\xi \in [\xi_i, \xi_{i+1})$ *then* $\mathbf{C}(\xi)$ *is contained by the convex hull defined by the control points* $\mathbf{P}_{i-p},\dots,\mathbf{P}_i$.

Now, given a NURBS curve or surface Ω, let us consider the problem of finding, for each point $\mathbf{p} \in \Theta$, the minimum distance between $\mathbf{p}$ and Ω. Our approach is divided into a preprocessing step and then a sequence of operations done for each point considered. In the preprocessing step we do the following:

1. Build the AABB of all the control points of the geometry (see Figure 7.13a).

2. For each element of the NURBS geometry, compute the AABB of the control points which would form the convex hull containing the element by following the strong convex hull property (see Figure 7.13b). These are the leaves of the bounding tree.

3. Build the bounding hierarchy by recursively unifying the AABBs of the nodes.

We remark that we choose to compute the AABB of the control points defining an element instead of using directly their convex hull; while the latter would fit the element better, it is also computationally more expensive to build and to test for intersections.

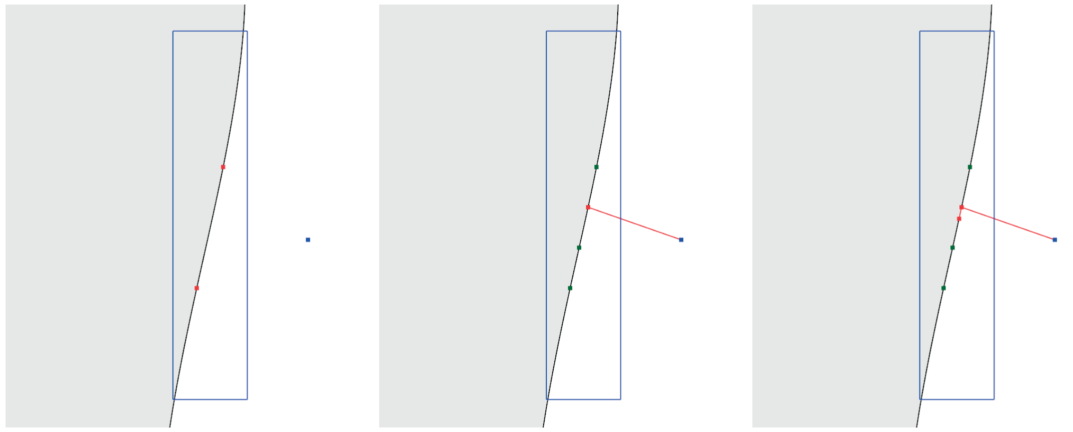
After the preprocessing step, for each point $\mathbf{p} \in \Theta$ in which we need to evaluate the minimum distance with Ω we perform the following:

1. Calculate the distance between $\mathbf{p}$ and the AABB of the whole geometry Ω: if the distance is greater than a threshold, discard the point and consider the distance equal to the threshold. The threshold τ must be chosen such that $\tilde{\delta}_\varepsilon(\tau) = 0$ and $\tilde{\mathscr{H}}_\varepsilon(\tau) = 0$, i.e. such that the point is considered to be outside of Ω and external to the smoothing region of the delta and Heaviside functions around Ω. This preliminary test is done to avoid performing further checks in points distant from the immersed object. If the complete distance field is required, then a large τ can be selected, effectively disabling this preliminary check.

2. The distance between $\mathbf{p}$ and the AABBs of the nodes of the bounding hierarchy is calculated, in a recursive fashion, from the root node towards the leaves. A list of candidate elements is formed by selecting the elements whose AABB includes the point $\mathbf{p}$ or by considering the element with smallest distance between its AABB and $\mathbf{p}$. Since the AABB of the convex hull of an element does not fit tightly the piece of geometry spanned by the element itself (see for example Figure 7.14a), the candidate elements selected through this step can be multiple.

3. For each candidate element, perform a sampling of the subdomain of the parameter space spanned by the knots spans defining the element; for each parametric sampling point, calculate the distance between the NURBS geometry evaluated in that point and the point $\mathbf{p}$ and keep the one with minimum distance (see Figure 7.14b). We consider a uniform sampling of the parametric element, as in [37].

After these steps, one obtains, for each $\mathbf{p} \in \Theta$, a set of candidate points, one for each candidate element. Then, each candidate point is used as initial guess for the iterative scheme described in Section 7.3.2 for NURBS curves and in Section 7.3.3 for NURBS surfaces.



(a) Selected element with its AABB and boundaries (in red)

(b) Sampling points (in green) and selected initial guess (in red)

(c) Projected point after 1 iteration

Figure 7.14 – Steps performed to project a point (in blue) onto an element of a NURBS curve: in (a) the candidate element, delimited by the red points, with its AABB, is shown; in (b), after a uniform sampling of the element (in the points in green), an initial guess (in red) is selected; in (c), after one iteration of Newton–Raphson, the projection is found.

7.3.2 Point projection on NURBS curves

Let us consider a NURBS curve $\mathbf{C} : \widehat{\Omega} \subset \mathbb{R} \to \mathbb{R}^d$. Our aim is to find the projection of a point $\mathbf{p} \in \mathbb{R}^d$ onto $\mathbf{C}$; by recalling Eq. (7.3.3), our problem reads:

$$\begin{aligned} &\text{find } \xi \in \widehat{\Omega} \text{ such that} \\ &\quad f(\xi) = \mathbf{C}'(\xi) \cdot \big(\mathbf{C}(\xi) - \mathbf{p}\big) = 0. \end{aligned} \tag{7.3.5}$$

In order to find the root of f, we consider the Newton–Raphson method, as employed in [37].

Let $n \in \mathbb{N}$ indicate the current iteration. We consider an initial guess $\xi^0 \in \widehat{\Omega}$ obtained by the algorithm described in Section 7.3.1, lying in the element defined by the knot span $[\tilde{\xi}_j, \tilde{\xi}_{j+1}] \subset \widehat{\Omega}$. Then, at each iteration n the following steps are performed:

1. Check if the distance between the current estimated point on the curve and the target point is below a tolerance:

$$\left\| \mathbf{C}(\xi^n) - \mathbf{p} \right\| \leq \varepsilon_1, \tag{7.3.6}$$

with $\varepsilon_1 \in \mathbb{R}$; if this is the case, we consider $\mathbf{C}(\xi^n)$ to be coincident with $\mathbf{p}$, thus we stop the iterations.

2. Check if the segment between the point $\mathbf{p}$ and $\mathbf{C}(\xi^n)$ is approximatively perpendicular

to the tangent of the the curve in ξ^n:

$$\frac{\left|\mathbf{C}'(\xi^n)\cdot\left(\mathbf{C}(\xi^n)-\mathbf{p}\right)\right|}{\|\mathbf{C}'(\xi^n)\|\,\left\|\mathbf{C}(\xi^n)-\mathbf{p}\right\|}\leq\varepsilon_2, \tag{7.3.7}$$

with $\varepsilon_2 \in \mathbb{R}$ being a tolerance. If Eq. (7.3.7) is satisfied then the point $\mathbf{C}(\xi^n)$ is a projection of $\mathbf{p}$ onto $\mathbf{C}$, thus the iterations are stopped.

3. Calculate the parametric coordinate of the new estimation:

$$\xi^{n+1}=\xi^n-\frac{f(\xi^n)}{f'(\xi^n)}=\xi^n-\frac{\mathbf{C}'(\xi^n)\cdot\left(\mathbf{C}(\xi^n)-\mathbf{p}\right)}{\mathbf{C}''(\xi^n)\cdot\left(\mathbf{C}(\xi^n)-\mathbf{p}\right)+\|\mathbf{C}'(\xi^n)\|^2}. \tag{7.3.8}$$

4. Ensure that the new iterate is within the knot span of the current element being considered:

$$\begin{aligned}&\text{if } \xi^{n+1}<\tilde{\xi}_j && \text{then set } \xi^{n+1}=\tilde{\xi}_j,\\ &\text{if } \xi^{n+1}>\tilde{\xi}_{j+1} && \text{then set } \xi^{n+1}=\tilde{\xi}_{j+1}.\end{aligned} \tag{7.3.9}$$

If the current guess is outside the current element, most likely it will lead to parametric points in one near element that would already be considered in the list of candidate elements built with the procedure in Section 7.3.1. Nevertheless, we choose to not stop the iterations and continue, by correcting the estimate ξ^{n+1} to be inside the considered knot span, in case the final projection point actually lies on the boundary of the element; the iterations cannot get stuck on the boundary of the element because of the stopping criterium in point 5.

5. Check if the new estimate is changing or if it is stationary:

$$\left\|\left(\xi^{n+1}-\xi^n\right)\mathbf{C}'(\xi^n)\right\|\leq\varepsilon_1; \tag{7.3.10}$$

if (7.3.10) is satisfied, then the iterations are stopped.

With this algorithm, performed for each candidate element selected in Section 7.3.1 from an initial guess, a set of potential projections of the target point $\mathbf{p}$ onto the curve $\mathbf{C}$ is found. From these points, the nearest point to $\mathbf{p}$ is selected, and the projection of $\mathbf{p}$ onto $\mathbf{C}$ is thus obtained.

7.3.3 Point projection on surfaces

Let us now consider a NURBS surface $\mathbf{S}:\widehat{\Omega}\subset\mathbb{R}^2\to\mathbb{R}^d$. The procedure to find the projection of $\mathbf{p}\in\mathbb{R}^d$ onto the surface $\mathbf{S}$ is similar to the algorithm described in Section 7.3.2. We consider the problem:

find $\boldsymbol{\xi} \in \widehat{\Omega}$ such that

$$\begin{cases} \big(\mathbf{S}(\boldsymbol{\xi}) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_1}(\boldsymbol{\xi}) = 0, \\ \big(\mathbf{S}(\boldsymbol{\xi}) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_2}(\boldsymbol{\xi}) = 0, \end{cases} \tag{7.3.11}$$

where $\mathbf{S}_{\xi_\alpha}(\boldsymbol{\xi})$ refers to the derivative of $\mathbf{S}$ evaluated in $\boldsymbol{\xi}$ along the α-th parametric direction. To find solutions to problem (7.3.11), an approach based on the Newton–Raphson method is considered, following [37].

With $n \in \mathbb{N}$ indicating the current iteration, $\boldsymbol{\xi}^0 \in \widehat{\Omega}$ the initial guess lying in the parametric element $\left[\tilde{\xi}_{1,j}, \tilde{\xi}_{1,j+1}\right] \times \left[\tilde{\xi}_{2,k}, \tilde{\xi}_{2,k+1}\right] \subset \widehat{\Omega}$, then at each iteration n the following steps are performed:

- Check the distance between $\mathbf{p}$ and the current approximation $\mathbf{S}(\boldsymbol{\xi}^n)$:

  $$\left\| \mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p} \right\| \le \varepsilon_1 \tag{7.3.12}$$

  and stop the iterations if the points coincide within the given tolerance $\varepsilon_1 \in \mathbb{R}$.

- Stop the iterations if the segment between $\mathbf{p}$ and $\mathbf{S}(\boldsymbol{\xi}^n)$ is perpendicular to the tangent plane in $\boldsymbol{\xi}^n$, within a tolerance $\varepsilon_2 \in \mathbb{R}$:

  $$\frac{\left| \mathbf{S}_{\xi_1}(\boldsymbol{\xi}^n) \cdot \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \right|}{\left\| \mathbf{S}_{\xi_1}(\boldsymbol{\xi}^n) \right\| \left\| \mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p} \right\|} \le \varepsilon_2 \qquad \text{and} \qquad \frac{\left| \mathbf{S}_{\xi_2}(\boldsymbol{\xi}^n) \cdot \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \right|}{\left\| \mathbf{S}_{\xi_2}(\boldsymbol{\xi}^n) \right\| \left\| \mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p} \right\|} \le \varepsilon_2. \tag{7.3.13}$$

- Solve the following linear system for the increment $\delta \boldsymbol{\xi}^{n+1}$:

  $$\mathrm{J}_i \, \delta \boldsymbol{\xi}^{n+1} = - \begin{bmatrix} \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_1}(\boldsymbol{\xi}^n) \\ \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_2}(\boldsymbol{\xi}^n) \end{bmatrix}, \tag{7.3.14}$$

  where

  $$\mathrm{J}_i = \begin{bmatrix} \left\| \mathbf{S}_{\xi_1}(\boldsymbol{\xi}^n) \right\|^2 + \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_1 \xi_1}(\boldsymbol{\xi}^n) & \mathbf{S}_{\xi_1}(\boldsymbol{\xi}^n) \cdot \mathbf{S}_{\xi_2}(\boldsymbol{\xi}^n) + \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_1 \xi_2}(\boldsymbol{\xi}^n) \\ \mathbf{S}_{\xi_1}(\boldsymbol{\xi}^n) \cdot \mathbf{S}_{\xi_2}(\boldsymbol{\xi}^n) + \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_2 \xi_1}(\boldsymbol{\xi}^n) & \left\| \mathbf{S}_{\xi_2}(\boldsymbol{\xi}^n) \right\|^2 + \big(\mathbf{S}(\boldsymbol{\xi}^n) - \mathbf{p}\big) \cdot \mathbf{S}_{\xi_2 \xi_2}(\boldsymbol{\xi}^n) \end{bmatrix}. \tag{7.3.15}$$

  Then, update the current guess:

  $$\boldsymbol{\xi}^{n+1} = \boldsymbol{\xi}^n + \delta \boldsymbol{\xi}^{n+1}. \tag{7.3.16}$$

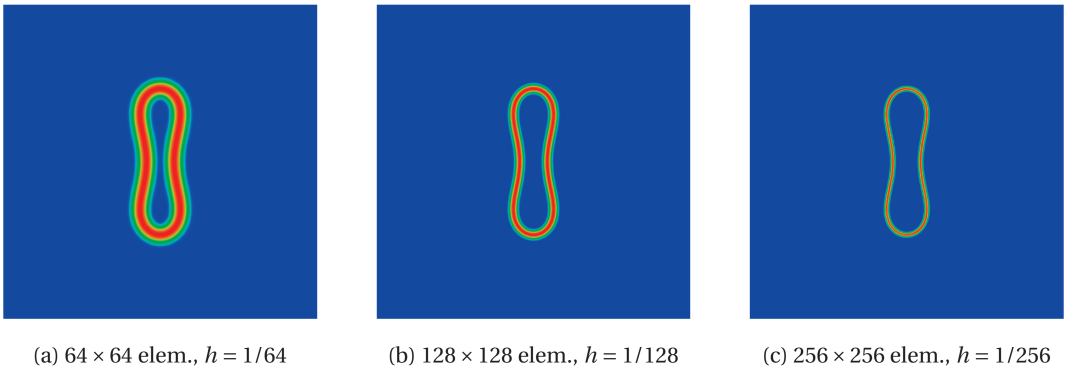

(a) 64 × 64 elem., $h = 1/64$ (b) 128 × 128 elem., $h = 1/128$ (c) 256 × 256 elem., $h = 1/256$

Figure 7.15 – Smoothed delta function applied to a NURBS curve evaluated on a NURBS planar quadrangular surface built of 64^2 elements (a), 128^2 elements (b), and 256^2 elements (c).

- Ensure that the new iterate is within the parametric element being considered:

$$\begin{aligned} &\text{if } \xi_1^{n+1} < \tilde{\xi}_{1,j} && \text{then set } \xi_1^{n+1} = \tilde{\xi}_{1,j}, \\ &\text{if } \xi_1^{n+1} > \tilde{\xi}_{1,j+1} && \text{then set } \xi_1^{n+1} = \tilde{\xi}_{1,j+1}, \\ &\text{if } \xi_2^{n+1} < \tilde{\xi}_{2,j} && \text{then set } \xi_2^{n+1} = \tilde{\xi}_{2,j}, \\ &\text{if } \xi_2^{n+1} > \tilde{\xi}_{2,j+1} && \text{then set } \xi_2^{n+1} = \tilde{\xi}_{2,j+1}. \end{aligned} \tag{7.3.17}$$

- Stop the iterations if the new estimate does not change, within a given tolerance, from the old estimate:

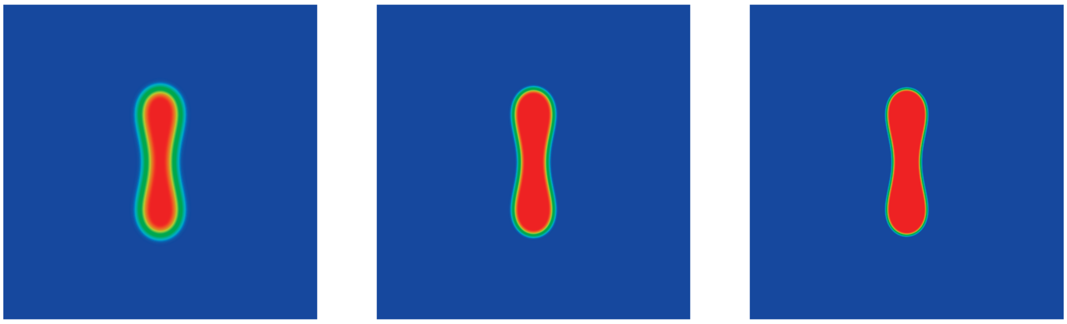
$$\left\| \left(\xi_1^{n+1} - \xi_1^n\right) \mathbf{S}_{\xi_1}(\boldsymbol{\xi}^n) + \left(\xi_2^{n+1} - \xi_2^n\right) \mathbf{S}_{\xi_2}(\boldsymbol{\xi}^n) \right\| \le \varepsilon_1. \tag{7.3.18}$$

After having obtained the set of points on **S** which are projections of the point **p** onto the surface **S**, the point with minimum distance to **p** is selected.

7.3.4 Resistive NURBS immersed object

With the algorithms outlined in Sections 7.3.1, 7.3.2, and 7.3.3 it is possible to obtain, for an arbitrary point **p**, the projected point **q** on a NURBS curve or surface which has minimum distance to **p** among all the projections of **p** onto the considered geometry. In order to build a smooth indicator function of the subspace enclosed by Ω, the signed distance between the arbitrary point **p** and Ω is required. Let $\mathbf{n}_q \in \mathbb{R}^d$ be the outward-pointing unit vector normal to Ω in **q**, which is evaluated exactly on the NURBS geometry since we retrieve, by using the described projection algorithms, the parametric coordinate corresponding to **q** on Ω. Then, the signed distance between the arbitrary point **p** and Ω is obtained as:

$$d(\mathbf{p}, \Omega) = \begin{cases} -\left\|\mathbf{p} - \mathbf{q}\right\| & \text{if } \left(\mathbf{p} - \mathbf{q}\right) \cdot \mathbf{n}_q \le 0, \\ \left\|\mathbf{p} - \mathbf{q}\right\| & \text{otherwise.} \end{cases} \tag{7.3.19}$$



(a) 64×64 elem., $h = 1/64$ (b) 128×128 elem., $h = 1/128$ (c) 256×256 elem., $h = 1/256$

Figure 7.16 – Smoothed Heaviside function applied to a NURBS curve evaluated on a NURBS planar quadrangular surface built of 64^2 elements (a), 128^2 elements (b), and 256^2 elements (c).

In view of the resistive method described in Section 7.2.3, the signed distance function in Eq. (7.3.19) is then combined with the smooth delta and Heaviside functions, as in Eqs. (7.2.35) and (7.2.36), obtaining the smoothed indicator functions of Ω and of the enclosed subdomain Θ_I.

7.3.5 Numerical example: NURBS curves immersed in a 2D domain

As an example, we consider $\Omega \subset \mathbb{R}^2$ to be a curve resembling the biconcave shape of a bidimensional vesicle membrane at equilibrium. The geometry Ω is immersed in a quadrangular domain $\Theta \subset \mathbb{R}^2$. Both Θ and Ω are represented by NURBS. Ω is a closed NURBS curve built of basis functions of degree $p = 3$ and globally C^2-continuous, for a total of 137 elements. The domain Θ is a NURBS planar surface built of $p = 2$ basis functions and C^1-continuous, for which we consider several refinement levels with $2^{n_r} \times 2^{n_r}$ elements, $n_r \in \mathbb{N}$. In order to test the construction of the indicator functions, we consider the L^2-projection of the functions δ_ε and $\mathscr{H}_\varepsilon$ onto the discrete NURBS function space defined by the basis functions which build Θ. The resulting projected delta and Heaviside functions, for three levels of refinement, are shown in Figures 7.15 and 7.16, respectively. The projected signed distance function, used to obtain the delta and Heaviside functions on the mesh of 256×256 elements is shown in Figure 7.17.

We then compare the area enclosed by Ω, calculated as:

$$A = \int_\Omega \mathbf{x} \cdot \mathbf{n}_\Omega \, d\Omega, \tag{7.3.20}$$

with $\mathbf{x}$ being the identity function defined in Eq. (1.1.27) and $\mathbf{n}_\Omega$ the outward-pointing unit vector normal to Ω, against the area calculated by integrating the projected Heaviside function on Θ, i.e.:

$$A_R = \int_{\Theta} \mathrm{P}_h \mathcal{H}_{\varepsilon} \, \mathrm{d}\Theta, \tag{7.3.21}$$

where the operator P_h refers to the L^2-projection operator onto the NURBS function space on Θ, for its different refinement levels. By considering the area obtained with Eq. (7.3.20) to be the reference value, we compare A against A_R by varying the refinement of Θ by calculating the error $e_A = |A - A_R|$. We set the smoothing parameter $\varepsilon = 4h$. In Figure 7.18 the error e_A against the element size h is shown. The trend is quadratic, due to the smoothing of $\mathcal{H}_{\varepsilon}$ proportional to h, and shows how the signed distance function, calculated by using the algorithms presented in Sections 7.3.1, 7.3.2, and 7.3.3, can effectively be used as an implicit representation of an immersed NURBS object.

Finally, we briefly show an example of the RIS method employed with an immersed NURBS geometry. We consider the benchmark case described in Section 7.2.9, for which the Navier–Stokes equations are approximated to simulate the flow past an obstacle; in particular, we consider, as fixed obstacle, the NURBS geometry Ω. The fluid properties and the discretization of the channel are the same as described in Section 7.2.9, with the only difference being the Reynolds number set to $\mathrm{Re} = 20$. In Figure 7.19 the approximated velocity field at time $t = 20$ is shown. We highlight that, if the extraction of the drag and lift on the obstacle is required, the approach outlined in Section 7.2.4 remains valid, being independent from the procedure used to build the signed distance function of the considered obstacle.

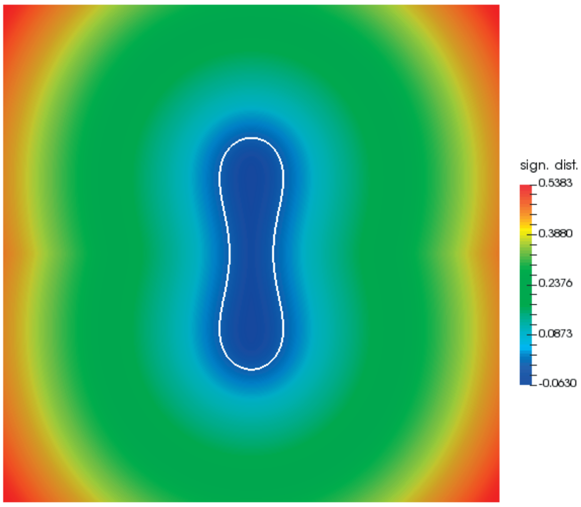


Figure 7.17 – Signed distance field obtained from an example NURBS curve.

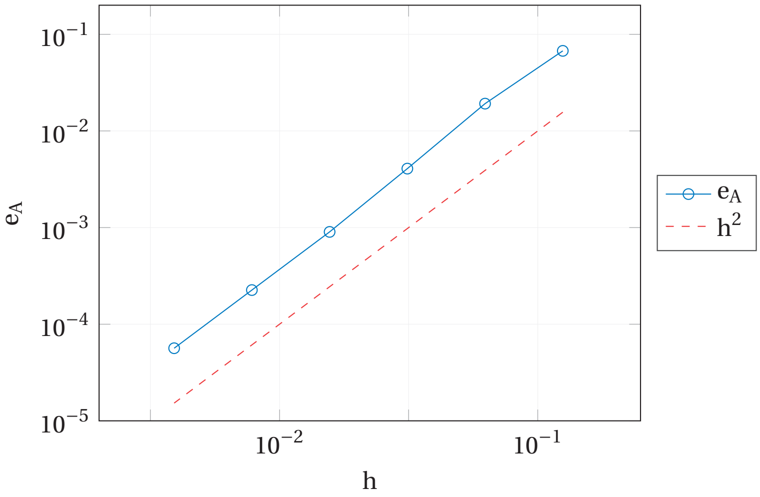


Figure 7.18 – Error in the approximation of the area by integrating $\mathscr{H}_\varepsilon$.

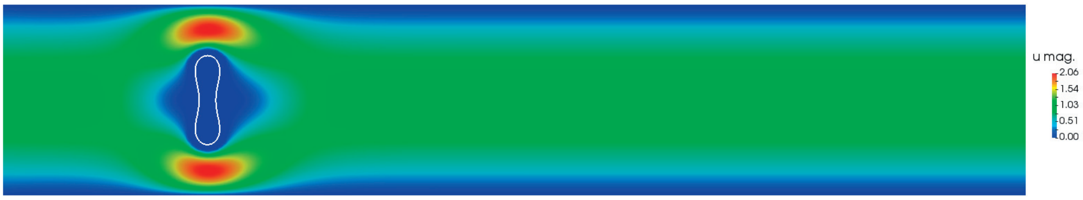


Figure 7.19 – Flow past NURBS ostacle at Re = 20. Velocity at time $t = 20$.

7.4 Fluid-membrane interaction

In Sections 7.2 and 7.3 we have considered the vesicle Θ_I just as being a fixed obstacle. Now, we take into consideration a further equation governing the evolution of the biomembrane as well. As the biomembrane evolves, we now refer to it with Ω_t, as in Chapter 6; similarly, we adopt the notation $\Theta_{I,t}$ to indicate the subset of Θ bounded by Ω_t.

The vesicle is subject to forces from the fluid, as well as to internal forces due to the bending energy and the inextensibility of the membrane. The internal forces are obtained from the shape derivative of the Canham–Helfrich energy (6.1.6), already considered in the formulation by gradient flow for recovering the equilibrium shapes of vesicles, and read:

$$\mathbf{f}_I(\mathbf{x},\Pi_\Omega,\delta p) = -\left\{k_c\left[\Delta_{\Omega_t}H + H\left(\frac{1}{2}H^2 - \frac{1}{2}H_0^2 - 2K\right) - 2H_0K\right] + \Pi_\Omega H + \delta p\right\}\mathbf{n}_\Omega \qquad \text{in } \Omega_t,\ t \in (0,T). \tag{7.4.1}$$

We have considered the same notation as in Chapter 6, including the presence of the Lagrange multipliers Π_Ω and δp. We remark that $\mathbf{x}$ refers to the identity map of the interface Ω_t at a given time $t \in (0,T)$, on which the curvatures and normals depend (in the notation, the dependency is dropped). These forces are balanced by the forces applied by the fluid from inside and outside of the biomembrane [52,216], which, given the fluid velocity $\mathbf{u}$ and pressure p fields, read:

$$\mathbf{f}_E(\mathbf{x},\mathbf{u},p) = [[\boldsymbol{\sigma}(\mathbf{u},p)]]\,\mathbf{n}_{\Omega_t} \qquad \text{in } \Omega_t,\ t \in (0,T), \tag{7.4.2}$$

where $\boldsymbol{\sigma}$ refers to the (dimensional) fluid stress tensor defined in Eq. (7.2.2) and with the notation $[[\cdot]]$ we refer to the jump across the membrane Ω_t. The dependency of $\mathbf{f}_E$ on $\mathbf{x}$ is due to the dependency of $\mathbf{n}_{\Omega_t}$ on Ω_t. Therefore, for a given (dimensional) fluid velocity and pressure fields (respectively $\mathbf{u}$ and p), the following problem governing the dynamics of the vesicle is considered, given the initial shape $\Omega_0 \subset \mathbb{R}^d$ described by $\mathbf{x}_0$:

for a.e. $t \in (0,T)$, find $\Omega_t \subset \mathbb{R}^d$, $\Pi_{\Omega t} \in \mathbb{R}$, and $\delta p_t \in \mathbb{R}$ such that

$$\begin{cases} \rho_\Omega\ddot{\mathbf{x}} - \mu_\Omega\nabla_{\Omega_t}\cdot(\nabla_{\Omega_t}\dot{\mathbf{x}}) - \mathbf{f}_I(\mathbf{x},\Pi_{\Omega t},\delta p_t) = \mathbf{f}_E(\mathbf{x},\mathbf{u},p) & \text{in } \Omega_t,\ t \in (0,T),\\ J_A(\Omega_t) = J_A(\Omega_0), & t \in (0,T),\\ J_V(\Omega_t) = J_V(\Omega_0), & t \in (0,T),\\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0,\\ \dot{\mathbf{x}}(0) = \mathbf{v}_0 & \text{in } \Omega_0, \end{cases} \tag{7.4.3}$$

with $\rho_\Omega \in \mathbb{R}^+$ being the density of the membrane, $\mathbf{v}_0 : \Omega \to \mathbb{R}^d$ the initial velocity of Ω_0, and $\mathbf{f}_I$ and $\mathbf{f}_E$ given in Eqs. (7.4.1) and (7.4.2), respectively.

With respect to the spontaneous curvature model employed in Chapter 6, we consider an additional diffusive term for the velocity of the biomembrane [217], with diffusion coefficient

$\mu_\Omega \in \mathbb{R}^+$; this term resembles the diffusion term of the fluidic model for lipidic membranes [53, 79, 218], for which the lipid molecules of the membrane behave like a viscous superficial fluid, an assumption experimentally verified [219, 220].

We proceed by writing problem (7.4.6) in dimensionless form. We use the same quantities and notation introduced in Section 7.2.1 for the Navier–Stokes equations. Similarly to [88, 89], we introduce the following parameters:

$$\tilde{\rho} := \frac{1}{L}\frac{\rho_\Omega}{\rho_O}, \qquad \tilde{\mu} := \frac{1}{L}\frac{\mu_\Omega}{\mu_O}, \qquad \mathrm{Ca} := \frac{\mu_O L^2 U}{k_c}, \qquad \text{and} \quad \tilde{k}_c := \frac{1}{\mathrm{Ca}\,\mathrm{Re}}; \tag{7.4.4}$$

in particular, Ca refers to the dimensionless capillarity number, characterizing the strength of the flow with respect to the bending resistance of the biomembrane [88]. We then rewrite problem (7.4.3) in dimensionless form:

for a.e. $t \in (0, T)$, find $\Omega_t \subset \mathbb{R}^d$, $\Pi_{\Omega t} \in \mathbb{R}$, and $\delta p_t \in \mathbb{R}$ such that

$$\begin{cases} \tilde{\rho}\,\ddot{\mathbf{x}} - \dfrac{\tilde{\mu}}{\mathrm{Re}} \nabla_{\Omega_t} \cdot (\nabla_{\Omega_t} \dot{\mathbf{x}}) + \tilde{k}_c \left[\Delta_{\Omega_t} H + H \left(\dfrac{1}{2} H^2 - \dfrac{1}{2} H_0^2 - 2K \right) - 2 H_0 K \right] \mathbf{n}_{\Omega_t} & \\ \qquad\qquad + \Pi_{\Omega t} H \mathbf{n}_{\Omega_t} + \delta p_t \mathbf{n}_{\Omega_t} = \left[\left[\boldsymbol{\sigma}(\mathbf{u}, p) \right]\right] \mathbf{n}_{\Omega_t} & \text{in } \Omega_t,\ t \in (0, T), \\ J_A(\Omega_t) = J_A(\Omega_0), & t \in (0, T), \\ J_V(\Omega_t) = J_V(\Omega_0), & t \in (0, T), \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0, \\ \dot{\mathbf{x}}(0) = \mathbf{v}_0 & \text{in } \Omega_0, \end{cases} \tag{7.4.5}$$

where only dimensionless quantities are now considered.

The interaction between the vesicle and the fluid is governed by two coupling conditions: the continuity of the velocities of the fluid and the membrane, by the adherence condition of Eq. (7.2.20), and the balances of forces between the ones due to the fluid and the internal forces. To enforce the continuity of the velocities from the fluid problem at Ω, we consider the RIS method described in Section 7.2.3. This leads to the following coupled problem governing the fluid-membrane interaction:

for a.e. $t \in (0,T)$, find $\mathbf{u} : \Theta \to \mathbb{R}^d$, $p : \Theta \to \mathbb{R}$, $\Omega_t \subset \mathbb{R}^d$, $\Pi_{\Omega t} \in \mathbb{R}$, and $\delta p_t \in \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - 2\nabla \cdot \left(\dfrac{1}{\mathrm{Re}} D(\mathbf{u}) \right) + \nabla p + C_R \delta_\varepsilon (\mathbf{u} - \dot{\mathbf{x}}) = \mathbf{0} & \text{in } \Theta,\ t \in (0,T), \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Theta,\ t \in (0,T), \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D,\ t \in (0,T), \\ -p\mathbf{n} + \dfrac{2}{\mathrm{Re}} D(\mathbf{u})\mathbf{n} = \mathbf{h} & \text{on } \Gamma_N,\ t \in (0,T), \\ \tilde{\rho}\ddot{\mathbf{x}} - \dfrac{\tilde{\mu}}{\mathrm{Re}} \nabla_{\Omega_t} \cdot (\nabla_{\Omega_t} \dot{\mathbf{x}}) + \tilde{k}_c \left[\Delta_{\Omega_t} H + H \left(\dfrac{1}{2} H^2 - \dfrac{1}{2} H_0^2 - 2K \right) - 2H_0 K \right] \mathbf{n}_{\Omega_t} & \\ \qquad\qquad + \Pi_{\Omega t} H \mathbf{n}_{\Omega_t} + \delta p_t \mathbf{n}_{\Omega_t} = [[\boldsymbol{\sigma}(\mathbf{u}, p)]]_\varepsilon \mathbf{n}_{\Omega_t} & \text{in } \Omega_t,\ t \in (0,T), \\ J_A(\Omega_t) = J_A(\Omega_0), & t \in (0,T), \\ J_V(\Omega_t) = J_V(\Omega_0), & t \in (0,T), \\ \mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Theta, \\ \mathbf{x}(0) = \mathbf{x}_0 & \text{in } \Omega_0, \\ \dot{\mathbf{x}}(0) = \mathbf{v}_0 & \text{in } \Omega_0, \end{cases} \tag{7.4.6}$$

where $\mathbf{u}_0 : \Theta \to \mathbb{R}^d$ is a suitable initial condition for the fluid velocity. In Eq. (7.4.6) the volume constraint on the biomembrane can optionally be dropped [52, 216], since one could rely on the incompressibility of the fluid to keep the biomembrane volume conserved. We highlight that we deal with the inextensibility of the membrane by imposing a Lagrange multiplier on the total area, as we did in Chapter 6; alternatively, a more accurate treatment of the inextensibility would be the introduction of an additional unknown, modeling the local surface tension (see e.g. [216]). Moreover, at the length scale of the vesicles, the viscosity of the fluid is the major factor of its rheology, leading to low values of the Reynolds number. This behavior would justify removing the convection term from the fluid equations, leading thus to the fluid being described by the Stokes equations, as considered for example in [87, 89]. However, in this context we choose to keep the convection term, in order to maintain a more general formulation.

In problem (7.4.11) the external forces exerted by the fluid on the biomembrane are obtained by approximating the jump of forces between the inner and outer sides of the membrane. This approximation, indicated with the notation $[[\cdot]]_\varepsilon$, relies on the fact that the smoothed delta function, applied to the signed distance function generated by the vesicle, has support in a narrow band around the membrane, of width 2ε; therefore, to fully evaluate the forces acting on the vesicle we need to consider the stresses in the whole band, in agreement with our discussion in Section 7.2.4. In order to do so, in each point $\mathbf{q} \in \Omega_t$ we approximate the jump of the forces by collecting the contributions to the forces lying on the segment of length

2ε, centered in $\mathbf{q}$, and directed along the normal to Ω_t evaluated in $\mathbf{q}$, i.e.:

$$\left[\!\left[\boldsymbol{\sigma}(\mathbf{u},p)\big|_{\mathbf{q}}\right]\!\right]\mathbf{n}_{\Omega_t}(\mathbf{q}) \approx \left[\!\left[\boldsymbol{\sigma}(\mathbf{u},p)\big|_{\mathbf{q}}\right]\!\right]_{\varepsilon}\mathbf{n}_{\Omega_t}(\mathbf{q}) := \int_{-\varepsilon}^{+\varepsilon} \boldsymbol{\sigma}(\mathbf{u},p)\nabla\delta_{\varepsilon}\big|_{\mathbf{q}+s\mathbf{n}_{\Omega_t}(\mathbf{q})}\,\mathrm{ds} \qquad \text{for } \mathbf{q}\in\Omega_t. \tag{7.4.7}$$

Then, we follow the approaches outlined in Sections 6.1.1 and 7.2.2, regarding the problems of the membrane and the fluid, respectively, to rewrite problem (7.4.6) in weak formulation, obtaining the following coupled problem:

for a.e. $t\in(0,T)$, find $\mathbf{u}(t)\in V_R$, $p(t)\in W$, $\mathbf{x}(t)\in P_t$, $f(t)\in Q_t$, $\Pi_\Omega(t)\in\mathbb{R}$,
and $\delta p(t)\in\mathbb{R}$ such that

$$\begin{cases}
\displaystyle\int_\Theta \frac{\partial\mathbf{u}}{\partial t}\cdot\boldsymbol{\varphi}\,\mathrm{d}\Theta + \int_\Theta \mathbf{u}\cdot\nabla\mathbf{u}\cdot\boldsymbol{\varphi}\,\mathrm{d}\Theta + \int_\Theta \frac{2}{\mathrm{Re}}D(\mathbf{u}):\nabla\boldsymbol{\varphi}\,\mathrm{d}\Theta - \int_\Theta p\nabla\cdot\boldsymbol{\varphi}\,\mathrm{d}\Theta \\
\qquad\displaystyle + \int_\Theta C_R\delta_\varepsilon(\mathbf{u}-\dot{\mathbf{x}})\cdot\boldsymbol{\varphi}\,\mathrm{d}\Theta = \int_{\Gamma_N}\mathbf{h}\cdot\boldsymbol{\varphi}\,\mathrm{d}\Gamma_N & \forall\boldsymbol{\varphi}\in V_0, \\
\displaystyle\int_\Theta \psi\nabla\cdot\mathbf{u} = 0 & \forall\psi\in W, \\
\displaystyle\int_{\Omega_t}\tilde{\rho}\ddot{\mathbf{x}}\cdot\boldsymbol{\eta}\,d\Omega_t + \frac{\tilde{\mu}}{\mathrm{Re}}\int_{\Omega_t}\nabla_{\Omega_t}\dot{\mathbf{x}}:\nabla_{\Omega_t}\boldsymbol{\eta}\,d\Omega_t \\
\qquad\displaystyle - \int_{\Omega_t} f\,\mathbf{n}_{\Omega_t}\cdot\boldsymbol{\eta}\,d\Omega_t = \int_{\Omega_t}[[\boldsymbol{\sigma}(\mathbf{u},p)]]_\varepsilon\mathbf{n}_{\Omega_t}\cdot\boldsymbol{\eta}\,d\Omega_t & \forall\boldsymbol{\eta}\in P_t, \\
\displaystyle\int_{\Omega_t} f\phi\,d\Omega_t - \tilde{k}_c\int_{\Omega_t}(\Delta_{\Omega_t}\mathbf{x}\cdot\mathbf{n}_{\Omega_t})\Delta_{\Omega_t}\phi\,d\Omega_t + 2\tilde{k}_c\int_{\Omega_t}H_0K\phi\,d\Omega_t \\
\qquad\displaystyle - \tilde{k}_c\int_{\Omega_t}(\Delta_{\Omega_t}\mathbf{x}\cdot\mathbf{n}_{\Omega_t})\left(\frac{1}{2}H^2-\frac{1}{2}H_0^2-2K\right)\phi\,d\Omega_t \\
\qquad\displaystyle - \Pi_\Omega\int_{\Omega_t}H\phi\,d\Omega_t - \delta p\int_{\Omega_t}\phi\,d\Omega_t = 0 & \forall\phi\in Q_t, \\
\displaystyle\int_{\Omega_t}H\mathbf{x}\cdot\mathbf{n}_{\Omega_t}\,d\Omega_t = A_0, \\
\displaystyle\int_{\Omega_t}\mathbf{x}\cdot\mathbf{n}_{\Omega_t}\,d\Omega_t = V_0,
\end{cases} \tag{7.4.8}$$

where, by introducing the unknown f, we have considered an approach similar to the Willmore flow in Section 5.4 and the Canham–Helfrich flow in Section 6.1.1, where the normal velocity of the surface is introduced as additional unknown. In problem (7.4.8) we consider the following function spaces:

$$\begin{aligned}
&V := \left[H^1(\Theta)\right]^d, \quad V_R := \left\{\boldsymbol{\varphi} \in V : \boldsymbol{\varphi} = \mathbf{g} \text{ on } \Gamma_D\right\}, \quad V_0 := \left\{\boldsymbol{\varphi} \in V : \boldsymbol{\varphi} = \mathbf{0} \text{ on } \Gamma_D\right\},\\
&W := L^2(\Theta),\\
&P_t := \left[H^2(\Omega_t)\right]^d, \quad Q_t := H^2(\Omega_t),
\end{aligned} \tag{7.4.9}$$

and the initial conditions:

$$\begin{aligned}
\mathbf{u}(0) &= \mathbf{u}_0 && \text{in } \Theta,\\
\mathbf{x}(0) &= \mathbf{x}_0 && \text{in } \Omega_0,\\
\dot{\mathbf{x}}(0) &= \mathbf{v}_0 && \text{in } \Omega_0,
\end{aligned} \tag{7.4.10}$$

with $\mathbf{u}_0 \in V_R$ being divergence-free.

7.4.1 Space discretization of the coupled problem

Following Sections 6.2.1 and 7.2.5, we proceed by considering a Galerkin method using NURBS-based IGA subspaces for the spatial discretization of problem (7.4.8). The fluid domain Θ is represented by a NURBS volume, if $d = 3$, or surface, if $d = 2$, while the biomembrane is represented by the family $\{\Omega_t\}_{t\in(0,T)}$ of closed NURBS surfaces, if $d = 3$, or curves, if $d = 2$. The two geometrical representations, of the fluid domain and the biomembrane domain, are independent. We thus consider two different NURBS function spaces $\mathcal{N}_h$ and $\mathcal{M}_h$, the first associated with the fluid domain and the second with the biomembrane domain. Therefore, function spaces related to the approximation of the fluid equations are $V_h := V \cap [\mathcal{N}_h]^d$, $V_{R,h} := \left\{\boldsymbol{\varphi}_h \in V_h : \boldsymbol{\varphi}_h = \mathbf{g}_h \text{ on } \Gamma_D\right\}$, $V_{R,h} := \left\{\boldsymbol{\varphi}_h \in V_h : \boldsymbol{\varphi}_h = \mathbf{0} \text{ on } \Gamma_D\right\}$, and $W_h := W \cap \mathcal{N}_h$; the function spaces related to the biomembrane are instead $P_{t,h} := P_t \cap [\mathcal{M}_h]^d$ and $Q_{t,h} := Q_t \cap \mathcal{M}_h$.

Then, we discretize in space problem (7.4.8), obtaining the following semi-discrete problem:

for a.e. $t \in (0,T)$, find $\mathbf{u}_h \in V_{R,h}$, $p_h \in W_h$, $\mathbf{x}_h \in P_{t,h}$, $f_h \in Q_{t,h}$, $\Pi_{\Omega t} \in \mathbb{R}$, and $\delta p_t \in \mathbb{R}$ such that

$$
\left\{
\begin{aligned}
&\int_\Theta \frac{\partial \mathbf{u}_h}{\partial t}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta \mathbf{u}_h\cdot\nabla\mathbf{u}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta \frac{2}{\mathrm{Re}} D(\mathbf{u}_h):\nabla\boldsymbol{\varphi}_h\,\mathrm{d}\Theta \\
&\qquad - \int_\Theta p_h \nabla\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \int_\Theta C_{R_h}\delta_\varepsilon(\mathbf{u}_h-\dot{\mathbf{x}}_h)\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta + \mathscr{S}_M(\boldsymbol{\varphi}_h;\mathbf{u}_h,p_h) \\
&\qquad = \int_{\Gamma_N} \mathbf{h}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Gamma_N && \forall \boldsymbol{\varphi}_h \in V_{0,h}, \\
&\int_\Theta \psi_h \nabla\cdot\mathbf{u}_h + \mathscr{S}_C(\psi_h;\mathbf{u}_h,p_h) = 0 && \forall \psi_h \in W_h, \\
&\int_{\Omega_t} \tilde{\rho}\ddot{\mathbf{x}}_h\cdot\boldsymbol{\eta}_h\,d\Omega_t + \frac{\tilde{\mu}}{\mathrm{Re}}\int_{\Omega_t} \nabla_{\Omega_t}\dot{\mathbf{x}}_h : \nabla_{\Omega_t}\boldsymbol{\eta}_h\,d\Omega_t \\
&\qquad - \int_{\Omega_t} f_h\mathbf{n}_{\Omega_t}\cdot\boldsymbol{\eta}_h\,d\Omega_t = \int_{\Omega_t} [[\boldsymbol{\sigma}(\mathbf{u}_h,p_h)]]_\varepsilon\,\mathbf{n}_{\Omega_t}\cdot\boldsymbol{\eta}_h\,d\Omega_t && \forall \boldsymbol{\eta}_h \in P_t, \\
&\int_{\Omega_t} f_h\phi_h\,d\Omega_t - \tilde{k}_c\int_{\Omega_t} (\Delta_{\Omega_t}\mathbf{x}_h\cdot\mathbf{n}_{\Omega_t})\Delta_{\Omega_t}\phi_h\,d\Omega_t + 2\tilde{k}_c\int_{\Omega_t} H_0 K_h\phi_h\,d\Omega_t \\
&\qquad - \tilde{k}_c\int_{\Omega_t} (\Delta_{\Omega_t}\mathbf{x}_h\cdot\mathbf{n}_{\Omega_t})\left(\frac{1}{2}H_h^2 - \frac{1}{2}H_0^2 - 2K_h\right)\phi_h\,d\Omega_t \\
&\qquad - \Pi_{\Omega t}\int_{\Omega_t} H_h\phi_h\,d\Omega_t - \delta p_t\int_{\Omega_t}\phi_h\,d\Omega_t = 0 && \forall \phi_h \in Q_t, \\
&\int_{\Omega_t} H_h\mathbf{x}_h\cdot\mathbf{n}_{\Omega_t}\,d\Omega_t = A_0, \\
&\int_{\Omega_t} \mathbf{x}_h\cdot\mathbf{n}_{\Omega_t}\,d\Omega_t = V_0,
\end{aligned}
\right.
\tag{7.4.11}
$$

with initial conditions:

$$
\begin{aligned}
\mathbf{u}_h(0) &= \mathbf{u}_{0,h} && \text{in } \Theta, \\
\mathbf{x}_h(0) &= \mathbf{x}_{0,h} && \text{in } \Omega_0, \\
\dot{\mathbf{x}}_h(0) &= \mathbf{v}_{0,h} && \text{in } \Omega_0,
\end{aligned}
\tag{7.4.12}
$$

where $\mathbf{u}_{0,h}$ is obtained as the L^2-projection of the initial data $\mathbf{u}_0$ onto the NURBS function space $V_{R,h}$ and $\mathbf{x}_{0,h}$ and $\dot{\mathbf{x}}_{0,h}$ represent the identity map and velocity associated to the initial configuration Ω_0 of the biomembrane. The terms $\mathscr{S}_M$ and $\mathscr{S}_C$ refer to the SUPG stabilization for the Navier–Stokes equations and have been introduced in Section 7.2.6.

7.4.2 Time discretization of the coupled problem

To solve the coupled problem, we consider a staggered approach, for which at each time step the fluid and the biomembrane subproblems are solved in separate stages [96, 221]. Due to the small thickness of the membrane and the type of coupling considered, we adopt a strongly coupled strategy [98, 222] to cope with the effects of the added mass [223, 223]. The coupling in problem (7.4.8) can be interpreted in analogy with the Dirichlet–Neumann coupling strategy ubiquitous in the domain of fluid-structure interaction problems [221, 224, 225]; in fact, the RIS term in the fluid equations plays the role of the Dirichlet condition (continuity of the velocities at the interface), while the balance of forces on the membrane represents the analog of the Neumann condition (continuity of stresses at the interface). To force a strong coupling between the fluid and the biomembrane, at each time step an iterative scheme in which the fluid and biomembrane subproblems are solved in sequence is considered, until a convergence criterium is satisfied. Both subproblems are discretized in time by using a BDF scheme of order k, as introduced in Chapter 5 and employed in Sections 6.2.2 and 7.2.7 for the time discretization of the biomembrane equilibrium problem and the Navier–Stokes equations, respectively.

Let us consider the time interval $(0, T)$, divided into N time steps of size Δt, corresponding to the time instances $t_i = i\Delta t$, for $i = 0, \ldots, N \in \mathbb{N}$, so that $t_0 = 0$ and $t_N = T$. We indicate with the index $j \in \mathbb{N}$, $j \geq 0$, the subiterations. Let us now consider a fixed time step index $n \in \mathbb{N}$, $0 \leq n \leq N-1$, at a given subiteration j.

Fluid subproblem

With $\mathscr{P}_F(\overline{\mathbf{x}}_h, \overline{\mathbf{v}}_h)$ we refer to the discrete fluid subproblem, to be solved at time step index n and subiteration j. The fluid subproblem depends on the identity map $\overline{\mathbf{x}}_h$ and velocity $\overline{\mathbf{v}}_h$ of the biomembrane. During the solution of the fluid subproblem, these terms are fixed and considered as given. The RIS term depends on $\overline{\mathbf{x}}_h$ due to the smooth delta and Heaviside functions δ_ε and $\mathscr{H}_\varepsilon$ being constructed with the NURBS geometry defined by $\overline{\mathbf{x}}_h$; the construction of the signed distance function, needed to evaluate the smooth delta and Heaviside functions, follows the algorithms in Section 7.3. The velocity $\overline{\mathbf{v}}_h$ imposed with the RIS term consists in the extension, to the fluid domain Θ, of the velocity of the membrane along the normal direction. Therefore, given $\overline{\mathbf{x}}_h$ and $\overline{\mathbf{v}}_h$, at each time instance t_n and subiteration j the fluid subproblem reads:

find $\mathbf{u}_h^{n+1,j} \in V_{R,h}$ and $p_h^{n+1,j} \in W_h$ such that

$$\left\{\begin{aligned}
&\int_\Theta \frac{\alpha_0}{\Delta t}\mathbf{u}_h^{n+1,j}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta+\int_\Theta \mathbf{u}_h^{*}\cdot\nabla\mathbf{u}_h^{n+1,j}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta+\int_\Theta\frac{2}{\mathrm{Re}}D(\mathbf{u}_h^{n+1,j}):\nabla\boldsymbol{\varphi}_h\,\mathrm{d}\Theta\\
&\qquad-\int_\Theta p_h^{n+1,j}\nabla\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta+\int_\Theta C_{R_h}\delta_\varepsilon\left(\mathbf{u}_h^{n+1,j}-\bar{\mathbf{v}}_h\right)\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta\\
&\qquad+\sum_{K\in\mathcal{K}_h}\int_K \tau_M^K(\mathbf{u}_h^*)\tilde{\mathbf{r}}_M(\mathbf{u}_h^{n+1,j},p_h^{n+1,j})\cdot\left(\mathbf{u}_h^*\cdot\nabla\boldsymbol{\varphi}_h\right)dK\\
&\qquad+\sum_{K\in\mathcal{K}_h}\int_K \tau_C^K(\mathbf{u}_h^*)r_C(\mathbf{u}_h^{n+1,j})\nabla\cdot\boldsymbol{\varphi}_h dK\\
&\qquad=\int_\Theta\frac{\alpha_0}{\Delta t}\mathbf{u}_h^{bdf,n}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Theta+\int_{\Gamma_N}\mathbf{h}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\Gamma_N \qquad\qquad \forall\boldsymbol{\varphi}_h\in V_{0,h},\\
&\int_\Theta \psi_h\nabla\cdot\mathbf{u}_h^{n+1,j}+\sum_{K\in\mathcal{K}_h}\left[\int_K \tau_M^K(\mathbf{u}_h^*)\mathbf{r}_M(\mathbf{u}_h^{n+1,j},p_h^{n+1,j})\cdot\nabla\psi_h dK\right]=0 \qquad \forall\psi_h\in W_h;
\end{aligned}\right. \tag{7.4.13}$$

we refer to Sections 7.2.6 and 7.2.7 for details about the SUPG stabilization terms for the Navier–Stokes equations and the time discretization with BDF schemes and extrapolation.

Membrane subproblem

Regarding the membrane subproblem, we discretize in time both the velocity $\dot{\mathbf{x}}_h$ and the acceleration $\ddot{\mathbf{x}}_h$ by using the same BDF scheme and of the same order k as employed for the fluid subproblem. We follow the same notation and procedure outlined in Section 5.2; in particular, we highlight that we approximate the second time derivative in Eq. (7.4.11) at time instance t_{n+1} and subiteration j, for $n \geq k-1$, as:

$$\ddot{\mathbf{x}}_h(t_{n+1}) \simeq \alpha_0\frac{\mathbf{v}_h^{n+1,j}-\mathbf{v}_h^{bdf,n}}{\Delta t}, \tag{7.4.14}$$

where $\mathbf{v}_h^{bdf,n} : \Omega_* \to \mathbb{R}^d$ is defined as:

$$\mathbf{v}_h^{bdf,n} := \sum_{i+1}^{k}\frac{\alpha_i}{\alpha_0}\left(\mathbf{v}_h^{n+1-i}\circ\mathbf{X}_h^{n+1-i}\right)\circ(\mathbf{X}_h^*)^{-1} \tag{7.4.15}$$

and we recall that:

$$\mathbf{v}_h^{n+1,j} := \alpha_0\frac{\mathbf{x}_h^{n+1,j}-\mathbf{x}_h^{bdf,n}}{\Delta t}. \tag{7.4.16}$$

We consider, at each time instance t_{n+1} and subiteration j, the velocity $\mathbf{v}_h^{n+1,j}$ as unknown of the problem; the new configuration Ω_{n+1}^j of the biomembrane is then represented by:

$$\mathbf{x}_h^{n+1,j} = \mathbf{x}_h^{bdf,n} + \frac{\Delta t}{\alpha_0}\mathbf{v}_h^{n+1,j}, \tag{7.4.17}$$

as in Eq. (5.2.9).

The biomembrane subproblem depends on the fluid velocity $\overline{\mathbf{u}}_h$ and pressure $\overline{p}_h$, employed for evaluating the forces acting on the membrane due to the fluid. These quantities, during the solution of the biomembrane subproblem, are considered fixed. Therefore, at fixed time instance t_n and subiteration j, considering a given fluid velocity $\overline{\mathbf{u}}_h$ and pressure $\overline{p}_h$, the discrete membrane subproblem $\mathscr{P}_M(\overline{\mathbf{u}}_h, \overline{p}_h)$ reads:

find $\mathbf{v}_h^{n+1,j} \in P_h^*$, $f_h^{n+1,j} \in Q_h^*$, $\Pi_\Omega^{n+1,j} \in \mathbb{R}$, and $\delta p^{n+1,j} \in \mathbb{R}$ such that

$$\begin{cases}
\displaystyle\int_{\Omega_*} \tilde{\rho}\frac{\alpha_0}{\Delta t}\mathbf{v}_h^{n+1,j}\cdot\boldsymbol{\eta}_h\, d\Omega_* + \frac{\tilde{\mu}}{\mathrm{Re}}\int_{\Omega_*} \nabla_{\Omega_*}\mathbf{v}_h^{n+1,j} : \nabla_{\Omega_*}\boldsymbol{\eta}_h\, d\Omega_* \\
\qquad\displaystyle - \int_{\Omega_*} f_h^{n+1,j}\,\mathbf{n}_{\Omega_*}\cdot\boldsymbol{\eta}_h\, d\Omega_* \\
\qquad\displaystyle = \int_{\Omega_*} \tilde{\rho}\frac{\alpha_0}{\Delta t}\mathbf{v}_h^{bdf,n}\cdot\boldsymbol{\eta}_h + \int_{\Omega_*} [[\boldsymbol{\sigma}(\overline{\mathbf{u}}_h, \overline{p}_h)]]_\varepsilon\,\mathbf{n}_{\Omega_*}\cdot\boldsymbol{\eta}_h\, d\Omega_* \qquad \forall\boldsymbol{\eta}_h \in P_h^*, \\
\displaystyle\int_{\Omega_*} f_h^{n+1,j}\psi_h\, d\Omega_* - \tilde{k}_c\frac{\Delta t}{\alpha_0}\int_{\Omega_*} (\Delta_{\Omega_*}\mathbf{v}_h^{n+1,j}\cdot\mathbf{n}_{\Omega_*})\Delta_{\Omega_*}\psi_h\, d\Omega_* \\
\qquad\displaystyle - \tilde{k}_c\frac{\Delta t}{\alpha_0}\int_{\Omega_*}\left[\frac{1}{2}(H_h^*)^2 - \frac{1}{2}H_0^2 - 2K_h^*\right](\Delta_{\Omega_*}\mathbf{v}_h^{n+1,j}\cdot\mathbf{n}_{\Omega_*})\psi_h\, d\Omega_* \\
\qquad\displaystyle - \Pi_\Omega^{n+1,j}\int_{\Omega_*} H_h^*\psi_h\, d\Omega_* - \delta p^{n+1,j}\int_{\Omega_*}\psi_h\, d\Omega_* \\
\qquad\displaystyle = -2\tilde{k}_c\int_{\Omega_*} H_0 K_h^*\psi_h\, d\Omega_* + \tilde{k}_c\int_{\Omega_*}\left(\Delta_{\Omega_*}\mathbf{x}_h^{bdf,n}\cdot\mathbf{n}_{\Omega_*}\right)\Delta_{\Omega_*}\psi_h\, d\Omega_* \\
\qquad\displaystyle + \tilde{k}_c\int_{\Omega_*}\left[\frac{1}{2}(H_h^*)^2 - \frac{1}{2}H_0^2 - 2K_h^*\right]\left(\Delta_{\Omega_*}\mathbf{x}_h^{bdf,n}\cdot\mathbf{n}_{\Omega_*}\right)\psi_h\, d\Omega_* \qquad \forall\psi_h \in Q_h^*, \\
\displaystyle\int_{\Omega_{n+1}^j} H_h^{n+1}\mathbf{x}_h^{n+1}\cdot\mathbf{n}_{\Omega_{n+1}^j}\, d\Omega_{n+1}^j = A_0, \\
\displaystyle\int_{\Omega_{n+1}^j} \mathbf{x}_h^{n+1}\cdot\mathbf{n}_{\Omega_{n+1}^j}\, d\Omega_{n+1}^j = V_0,
\end{cases} \tag{7.4.18}$$

where P_h^* and Q_h^* correspond to the function spaces $P_{t,h}$ and $Q_{t,h}$ built on Ω_*, respectively. To enforce the constraints, the two schemes described in Section 6.2.3 can be employed equivalently.

Coupling strategy

We now describe the strong coupling strategy adopted in this chapter. With this aim, we first introduce the following additional quantities:

- the relaxation parameter $\gamma_j \in \mathbb{R}^+$,
- the displacements of the biomembrane $\mathbf{d}_h^{n,j} : \Omega_* \to \mathbb{R}^d$,
- the relaxed displacements of the biomembrane $\tilde{\mathbf{d}}_h^{n,j} : \Omega_* \to \mathbb{R}^d$,
- the relaxed velocity of the biomembrane $\tilde{\mathbf{v}}_h^{n+1,j} : \Omega_* \to \mathbb{R}^d$,
- the relaxed biomembrane geometry $\widetilde{\Omega}_n^j \subset \mathbb{R}^d$, defined by the map $\tilde{\mathbf{x}}_h^{n+1,j} : \Omega_* \to \mathbb{R}^d$,

with j and n being the subiteration and time step indexes, respectively. Then, at each time instance t_n, with $n = k, \dots, N-1$, k being the order of the BDF scheme employed for the time discretization of the two subproblems, we consider the following procedure:

1. We initialize the subiterations by setting $j = 0$ and:

$$\begin{aligned}
\mathbf{u}_h^{n+1,0} &= \mathbf{u}_h^{n,j_{max,n-1}},\\
\tilde{\mathbf{d}}_h^{n+1,-1} &= \tilde{\mathbf{d}}_h^{n+1,-2} = \mathbf{d}_h^{n,j_{max,n-1}},\\
\tilde{\mathbf{x}}_h^{n+1,0} &= \mathbf{x}_h^{n,j_{max,n-1}},\\
\tilde{\mathbf{v}}_h^{n+1,0} &= \mathbf{v}_h^{n,j_{max,n-1}},
\end{aligned} \tag{7.4.19}$$

   where $j_{max,n-1}$ indicates number of the subiterations performed at time instance t_{n-1}. The initial value of the relaxation parameter can be either set to the value calculated in the last subiteration of the previous time step, or reset to a given fixed initial value.

2. Given $\tilde{\mathbf{x}}_h^{n+1,j}$ and $\tilde{\mathbf{v}}_h^{n+1,j}$, we solve the fluid subproblem $\mathscr{P}_F(\tilde{\mathbf{x}}_h^{n+1,j}, \tilde{\mathbf{v}}_h^{n+1,j})$, obtaining the new approximations of $\mathbf{u}_h^{n+1,j}$ and $p_h^{n+1,j}$; the distance function, needed by the resistive term, is computed on $\widetilde{\Omega}_{n+1}^j$.

3. Given $\mathbf{u}_h^{n+1,j}$ and $p_h^{n+1,j}$, we solve the membrane subproblem $\mathscr{P}_M(\mathbf{u}_h^{n+1,j}, p_h^{n+1,j})$, obtaining the new approximations of $\mathbf{v}_h^{n+1,j}$, $f_h^{n+1,j}$, $\Pi_\Omega^{n+1,j}$, and $\delta p^{n+1,j}$.

4. We perform a relaxation of the displacements of the biomembrane by employing the Aitken acceleration [98, 225]. In particular, the parameter γ_j is updated as:

$$\gamma_j = \gamma_{j-1} + (\gamma_{j-1} - 1)\frac{\left(\delta\mathbf{d}_h^{n+1,j-1} - \delta\mathbf{d}_h^{n+1,j}\right)\cdot\delta\mathbf{d}_h^{n+1,j}}{\left\|\delta\mathbf{d}_h^{n+1,j-1} - \delta\mathbf{d}_h^{n+1,j}\right\|^2}, \tag{7.4.20}$$

where:

$$\begin{aligned}
\delta \mathbf{d}_h^{n+1,j} &= \tilde{\mathbf{d}}_h^{n+1,j-1} - \mathbf{d}_h^{n+1,j}, \\
\delta \mathbf{d}_h^{n+1,j-1} &= \tilde{\mathbf{d}}_h^{n+1,j-2} - \mathbf{d}_h^{n+1,j-1}.
\end{aligned} \tag{7.4.21}$$

Then, the new relaxed displacements $\tilde{\mathbf{d}}_h^{n+1,j}$ are obtained as:

$$\tilde{\mathbf{d}}_h^{n+1,j} = (1-\gamma_j)\mathbf{d}_h^{n+1,j} + \gamma_j \tilde{\mathbf{d}}_h^{n+1,j-1}. \tag{7.4.22}$$

The relaxed displacements are used to build the relaxed biomembrane $\widetilde{\Omega}_{n+1}^{j+1}$ defined through the relaxed map $\tilde{\mathbf{x}}_h^{n+1,j+1}$; the biomembrane relaxed velocity $\tilde{\mathbf{v}}_h^{n+1,j+1}$ is obtained by approximation with the considered BDF formula applied to the relaxed displacements $\tilde{\mathbf{d}}_h^{n+1,j}$ together with the approximated displacements at the previous time steps, in line with Eq. (5.2.5).

5. The relaxed displacements are also used as stopping criterium for the subiterations. If the following condition is satisfied:

$$\left\| \mathbf{d}_h^{n+1,j} - \tilde{\mathbf{d}}_h^{n+1,j-1} \right\|^2 \leq \varepsilon_d, \tag{7.4.23}$$

   with a given tolerance $\varepsilon_d \in \mathbb{R}^+$, then convergence of the subiterations is reached and we proceed with the next time step. Otherwise, j is incremented and we continue from point 2.

7.4.3 Numerical results

We apply the proposed strategy to solve the fluid-membrane interaction problem in three cases: a vesicle under parabolic flow, under parabolic flow with an obstruction, and under shear flow. All problems are setup in the two-dimensional space, i.e. $d = 2$. Nevertheless, the approach proposed is formulated to be compatible both with the three-dimensional case, with the biomembrane being represented as a surface and the fluid domain as a volume, as well as the two-dimensional case, with the biomembrane being a curve and the fluid domain a 2D domain. While the formulations described in Chapter 5 and 6 are developed for surface geometric PDEs, they can be generalized to curves without any modification. The schemes for enforcing the geometric constraints outlined in Section 6.2.3 remain in place when dealing with curves as well; the constraint on the surface *area* becomes a constraint on the curve *perimeter*, while the constraint on the *volume* internal to the surface becomes a constraint on the curve inner *area*. Moreover, the two-dimensional equivalent of the reduced volume V_R, defined in Eq. (6.3.6) for three-dimensional vesicles, is the *reduced area*, defined as ([88]):

$$A_R := \frac{4\pi V}{A^2}, \tag{7.4.24}$$

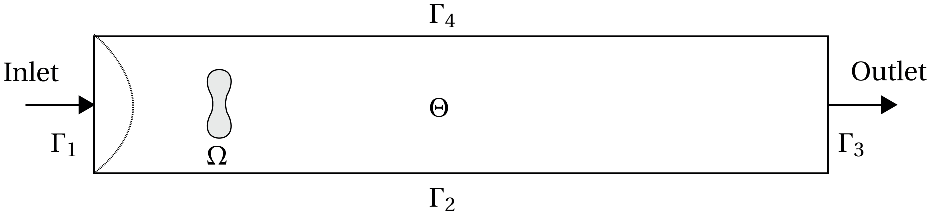


Figure 7.20 – Vesicle in parabolic flow. Sketch of the setup.

where V and A, in two-dimensions, indicate the area and perimeter of the vesicle, respectively.

Vesicle in parabolic flow

We consider a vesicle immersed in a rectangular channel with fluid flowing with a parabolic velocity profile. The case is sketched in Figure 7.20. The fluid domain Θ consists in the rectangle $(0,3) \times (0,1) \subset \mathbb{R}^2$, with boundaries represented by Γ_1, Γ_2, Γ_3, and Γ_4. A vesicle, represented by the curve $\Omega \in \mathbb{R}^2$, lies initially with center in $(0.75, 0.5) \in \mathbb{R}^2$. At the inlet Γ_1 a parabolic velocity profile is prescribed:

$$\mathbf{g}^1 = 4y(1-y)\,\mathbf{e}_x \qquad y \in [0,1]. \tag{7.4.25}$$

This profile is applied gradually over an initial period of time. In general, starting from time t_0 with the velocity profile $\mathbf{g}^0$, we change the velocities towards the wanted profile $\mathbf{g}^1$ with a smooth ramp until time t_1 is reached:

$$\mathbf{g} = \begin{cases} \mathbf{g}^0 & \text{if } t < t_0, \\ \mathbf{g}^0 + \theta\left(\dfrac{t-t_0}{t_1-t_0}\right)(\mathbf{g}^1 - \mathbf{g}^0) & \text{if } t_0 \le t \le t_1, \qquad \text{on } \Gamma_1,\ t \in [0,T), \\ \mathbf{g}^1 & \text{if } t > t_1, \end{cases} \tag{7.4.26}$$

where $\theta : [0,1] \to [0,1]$ is a smooth ramp function from 0 to 1, e.g.:

$$\theta(t) = 3t^2 - 2t^3 \qquad t \in [0,1]. \tag{7.4.27}$$

For this problem, we consider $\mathbf{g}^0 = \mathbf{0}$, $t_0 = 0$, and $t_1 = 0.1$; therefore, the fluid is initially at rest. On Γ_2 and Γ_4 we apply no-slip conditions, while Γ_3 represents the outlet from which the flow exits the domain. Therefore, we apply the following boundary conditions on the fluid subproblem:

$$\begin{aligned} \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_1,\ t \in (0,T), \\ \boldsymbol{\sigma}\mathbf{n} &= \mathbf{0} && \text{on } \Gamma_3,\ t \in (0,T), \\ \mathbf{u} &= \mathbf{0} && \text{on } \Gamma_2 \cup \Gamma_4,\ t \in (0,T), \end{aligned} \tag{7.4.28}$$

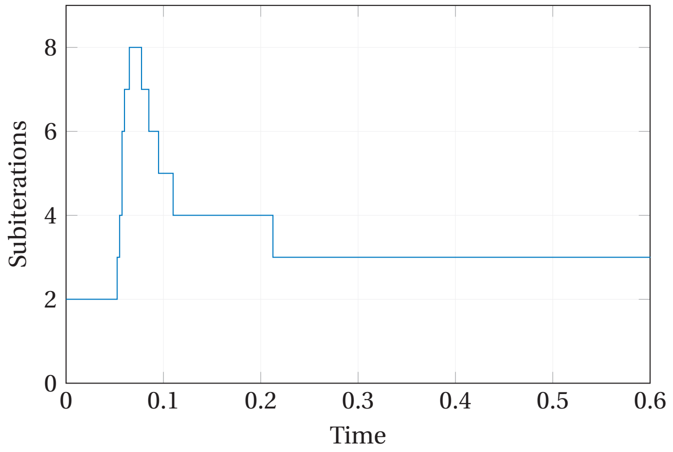
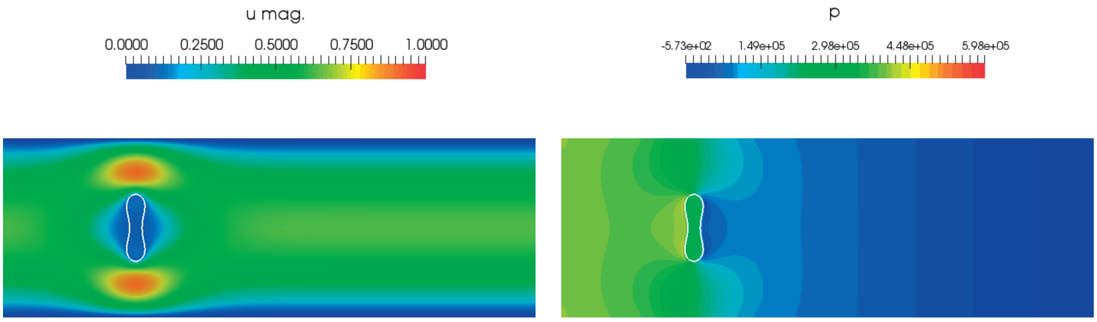


Figure 7.21 – Vesicle in parabolic flow. Numer of subiterations performed by the coupled solver (zoom for $t \in (0, 0.6)$).

where $\mathbf{n}$ is the outward pointing unit normal vector at the outlet Γ_O. By referring to the height of the channel as characteristic length, we impose a flow characterized by $\mathrm{Re} = 10^{-4}$; we consider the same fluid properties for the vesicle internal and external fluid. The biomembrane has perimeter equal to 0.86010, area 0.03142, resulting in a reduced area equal to 0.53380; moreover, we consider $\tilde{\rho} = 10^2$, $\tilde{\mu} = 10^{-3}$, and $\tilde{k}_c = 10^{-4}$. The biomembrane is initialized with ellipsoidal shape. We let it evolve by Canham–Helfrich flow, without interaction with the fluid, towards the biconcave equilibrium configuration. The equilibrium shape is then used as initial condition for the solution of the coupled problem. The initial velocity of the membrane is set to zero.

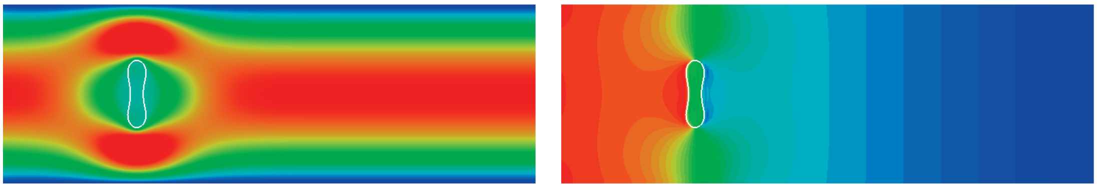
The fluid domain is represented by a NURBS planar surface made of 240 × 80 elements and basis functions of degree $p = 2$ and globally C^1-continuous, for a total of 19,200 elements. The biomembrane is represented by a NURBS curve of 137 elements with basis functions of degree $p = 3$ and globally C^2-continuous. The resulting linear system associated to the fluid subproblem is composed by 78,168 DOFs, while the membrane subproblem results in 560 DOFs. The equations are discretized in time with a BDF scheme of order $k = 2$ and time step size $\Delta t = 0.0025$. The iterative scheme C-1, described in Section 6.2.3, is employed for enforcing the geometric constraints on the membrane. The resistive constant C_{R_h} is set to $10^{-6}/h$. The meshes are divided using METIS [226] into 4 subdomains; the linear systems are assembled in parallel and solved by using the GMRES method with additive-Schwarz preconditioner, provided by IFPACK [227], and MUMPS [185] as local solver.

In Figure 7.22 we show the approximated fluid velocity and pressure at several time instances. The biomembrane is initially at rest. When the fluid velocity starts to rise, an high fluid pressure around the vesicle is developed, resulting in forces aligned with the flow acting on the biomembrane. The vesicle then starts moving, increasing its velocity towards the velocity of the fluid, until it reaches the same velocity. Then, the outlet touches the vesicle at approximately $t = 2.6975$. We remark that the iterative scheme C-1, adopted for these test



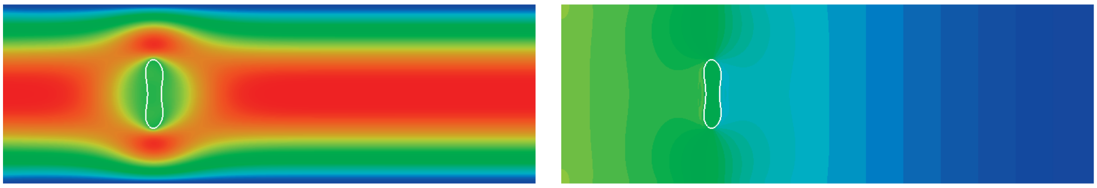
Fluid velocity at $t = 0.063$

Fluid pressure at $t = 0.063$



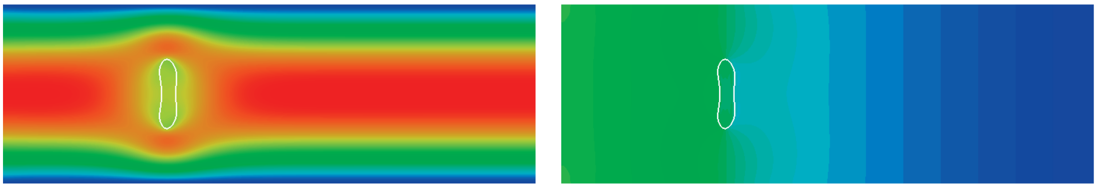
Fluid velocity at $t = 0.125$

Fluid pressure at $t = 0.125$



Fluid velocity at $t = 0.375$

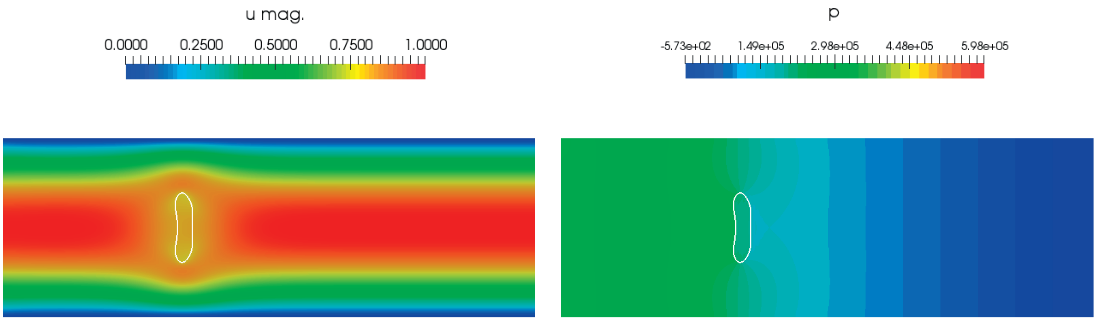
Fluid pressure at $t = 0.375$



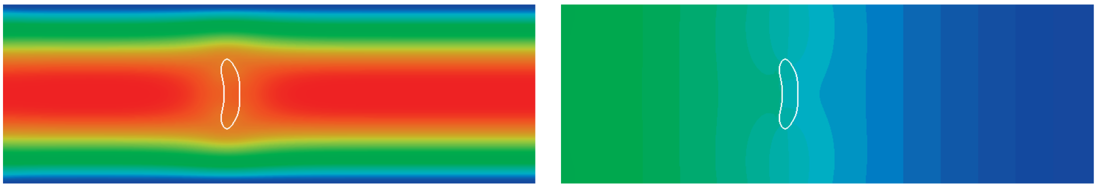
Fluid velocity at $t = 0.500$

Fluid pressure at $t = 0.500$

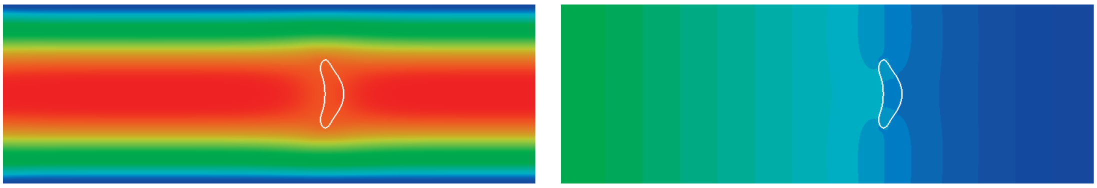
Figure 7.22 – Vesicle in parabolic flow. Fluid velocity and pressure at several time instances.



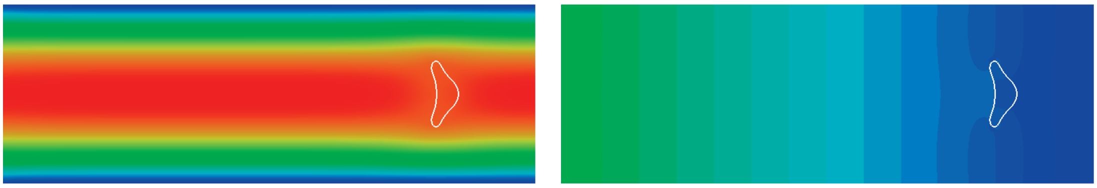
Fluid velocity at $t = 0.625$ Fluid pressure at $t = 0.625$



Fluid velocity at $t = 0.938$ Fluid pressure at $t = 0.938$



Fluid velocity at $t = 1.563$ Fluid pressure at $t = 1.563$



Fluid velocity at $t = 2.250$ Fluid pressure at $t = 2.250$

Figure 7.22 – Vesicle in parabolic flow. Fluid velocity and pressure at several time instances.

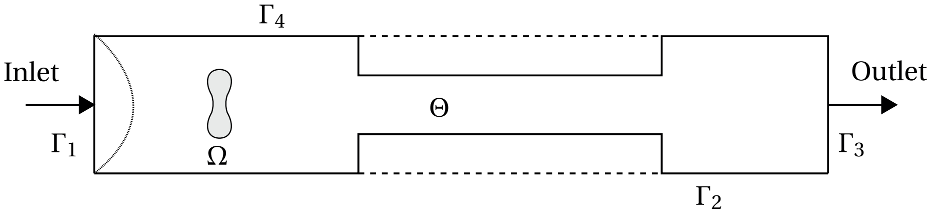


Figure 7.23 – Vesicle in parabolic flow with obstruction. Sketch of the setup.

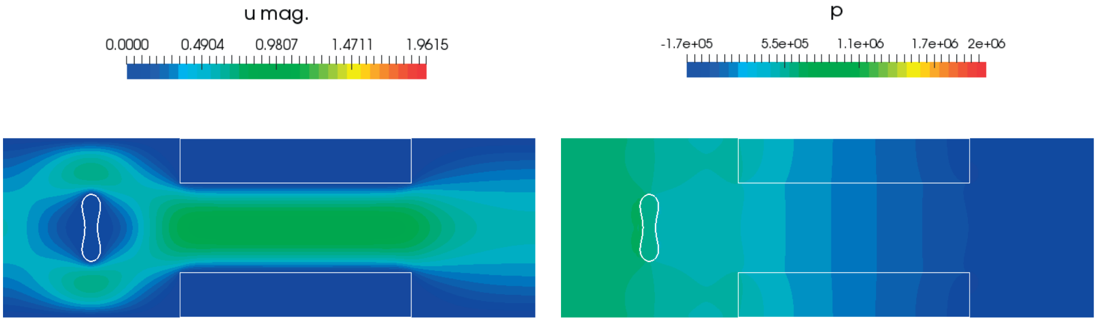
cases, enforces the geometric constraints within a selectable tolerance; therefore, conservation, within the tolerance, of the perimeter and area of the vesicle is observed throughout the whole simulation.

We used a tolerance of $\varepsilon_d = 5 \cdot 10^{-4}$ for the stopping criterium of the fluid-membrane subiterations, with a minimum of 2 subiterations to perform (since we observed an increased stability). In Figure 7.21 we report the number of subiterations performed at each time instance for $t \in (0, 0.6)$. Starting at 2 subiterations per time step, when the biomembrane is at rest, it then reaches a maximum of 8 subiterations per time step when the pressure gradient is strong and the biomembrane starts moving. While the velocities of the fluid and the membrane slowly become constant, the amount of subiterations performed is lowered, until a stable value of 3 is reached and kept constant till the end of the simulation.

The behavior shown by the vesicle is qualitatively in agreement with experimental data [228, 229]. The biomembrane is deformed to cope with the parabolic flow exerted by the fluid: in particular, it assumes a parachute-like shape, typical of vesicles flowing in small capillaries [229]. However, the end of the parachute shape is not straight, but slightly presents tail flaps [228]; this is due to the small value of reduced area of the biomembrane, which is thus subject to larger deformations if compared to vesicles with higher reduced area [230].

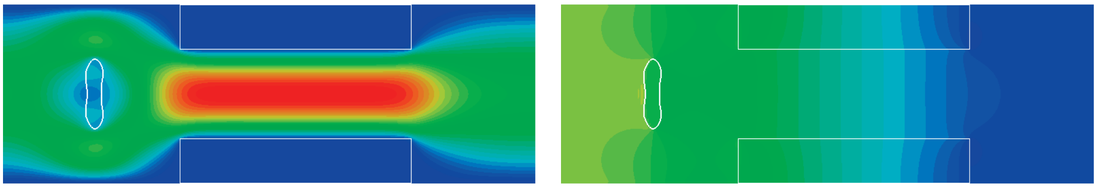
Vesicle in parabolic flow with obstruction

We now consider the dynamics of a vesicle immersed in a channel with an obstruction. In particular, we take the same setup of the case with a vesicle under parabolic flow, shown in Figure 7.20, and in addition we consider a zone where the channel is narrowed. The modified setup is sketched in Figure 7.23. In particular, we consider two different obstructions: in the first case, the obstruction has size equal to $S = 0.5$, while in the second case it is smaller and equal to $S = 0.4$. All the parameters of the problem are the same as for the case of parabolic flow without obstruction, with the exception of $\tilde{k}_c = 10^{-3}$. The meshes and discretization parameters are the same. The obstructions are handled by considering two additional resistive immersed rectangular objects, one near the top wall and the other near the bottom wall, in which a null velocity is imposed weakly. In practice, the three resistive terms (stemming from the two rectangles and the vesicle) are treated together and result in a single resistive term



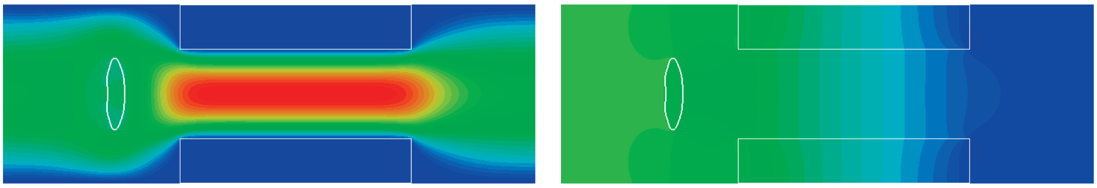
Fluid velocity at $t = 0.05$

Fluid pressure at $t = 0.05$



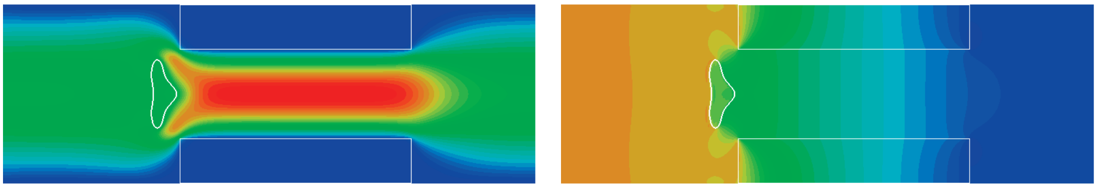
Fluid velocity at $t = 0.20$

Fluid pressure at $t = 0.20$



Fluid velocity at $t = 0.45$

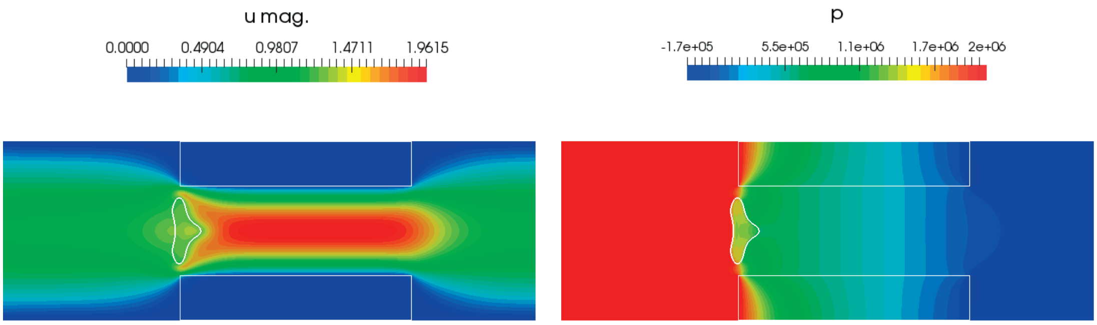
Fluid pressure at $t = 0.45$



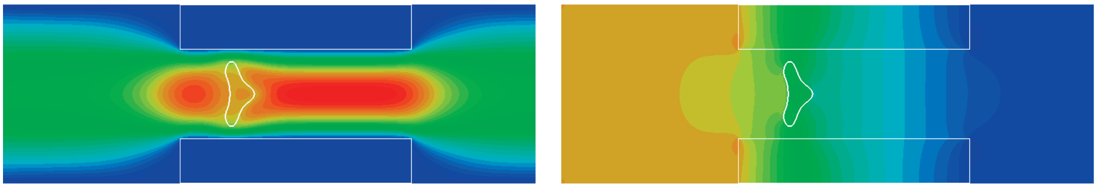
Fluid velocity at $t = 0.80$

Fluid pressure at $t = 0.80$

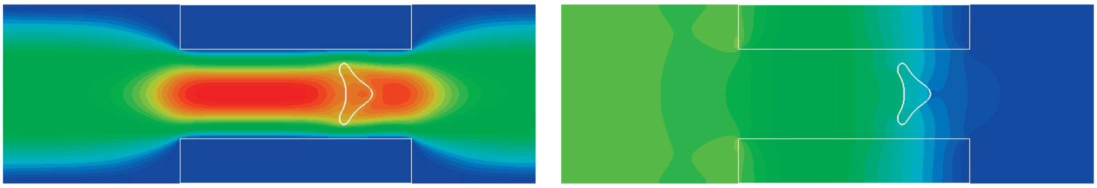
Figure 7.24 – Vesicle in parabolic flow with obstruction $S = 0.5$. Fluid velocity and pressure at several time instances.



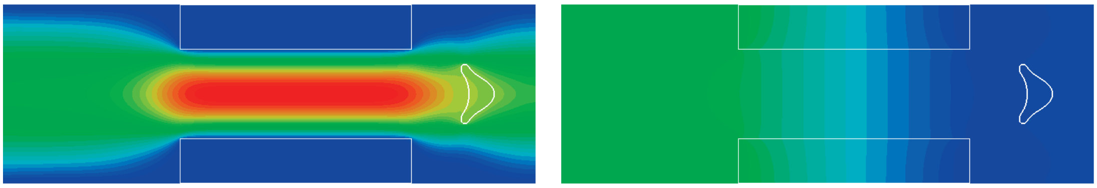
Fluid velocity at $t = 0.95$ Fluid pressure at $t = 0.95$



Fluid velocity at $t = 1.20$ Fluid pressure at $t = 1.20$

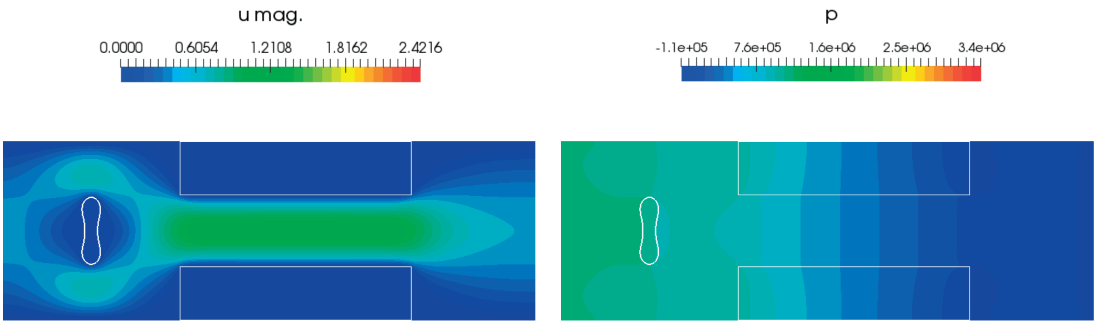


Fluid velocity at $t = 1.65$ Fluid pressure at $t = 1.65$



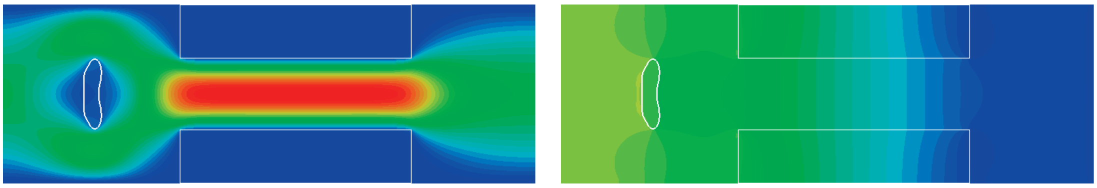
Fluid velocity at $t = 2.10$ Fluid pressure at $t = 2.10$

Figure 7.24 – Vesicle in parabolic flow with obstruction $S = 0.5$. Fluid velocity and pressure at several time instances.



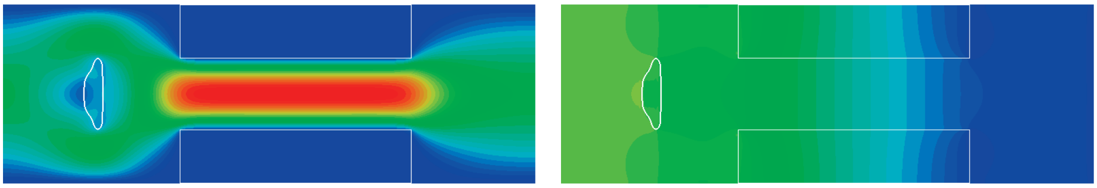
Fluid velocity at $t = 0.05$

Fluid pressure at $t = 0.05$



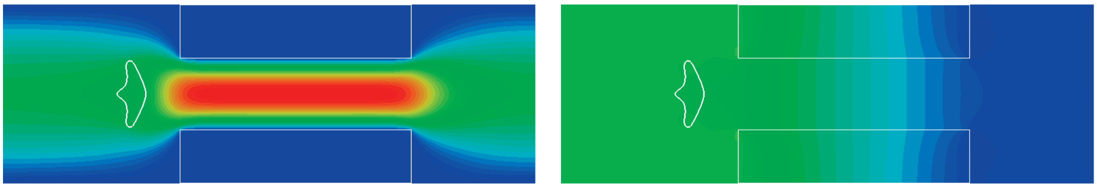
Fluid velocity at $t = 0.20$

Fluid pressure at $t = 0.20$



Fluid velocity at $t = 0.30$

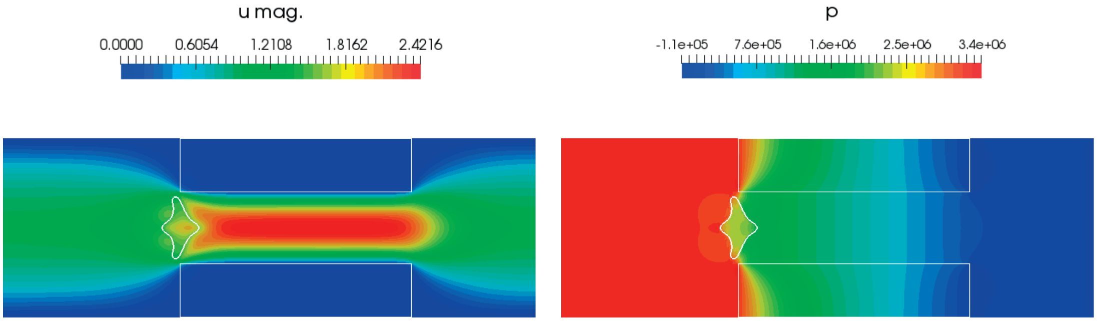
Fluid pressure at $t = 0.30$



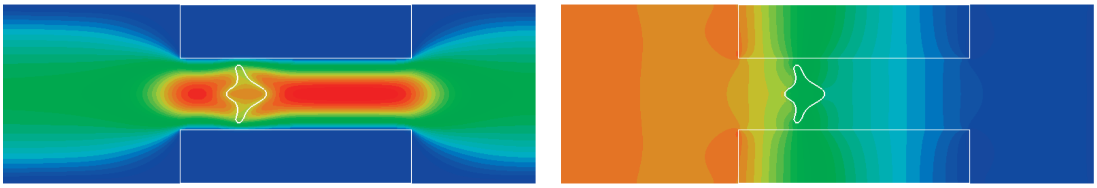
Fluid velocity at $t = 0.55$

Fluid pressure at $t = 0.55$

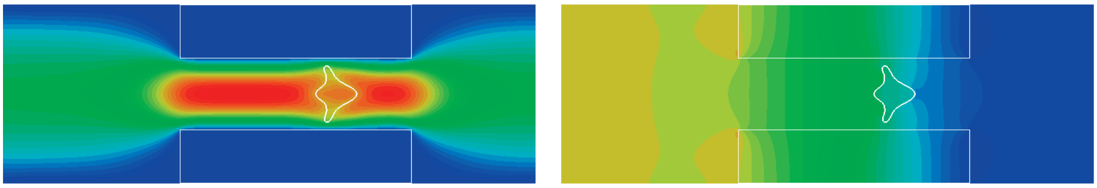
Figure 7.25 – Vesicle in parabolic flow with obstruction $S = 0.4$. Fluid velocity and pressure at several time instances.



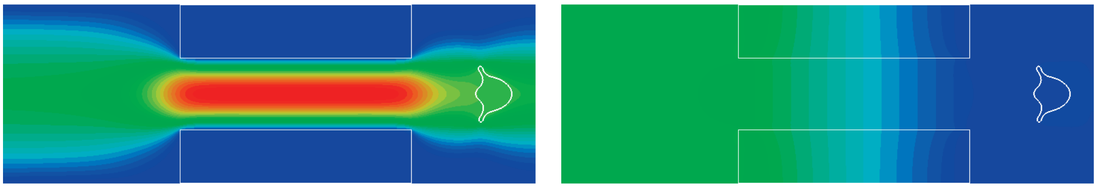
Fluid velocity at $t = 0.80$ Fluid pressure at $t = 0.80$



Fluid velocity at $t = 1.05$ Fluid pressure at $t = 1.05$



Fluid velocity at $t = 1.35$ Fluid pressure at $t = 1.35$



Fluid velocity at $t = 1.85$ Fluid pressure at $t = 1.85$

Figure 7.25 – Vesicle in parabolic flow with obstruction $S = 0.4$. Fluid velocity and pressure at several time instances.

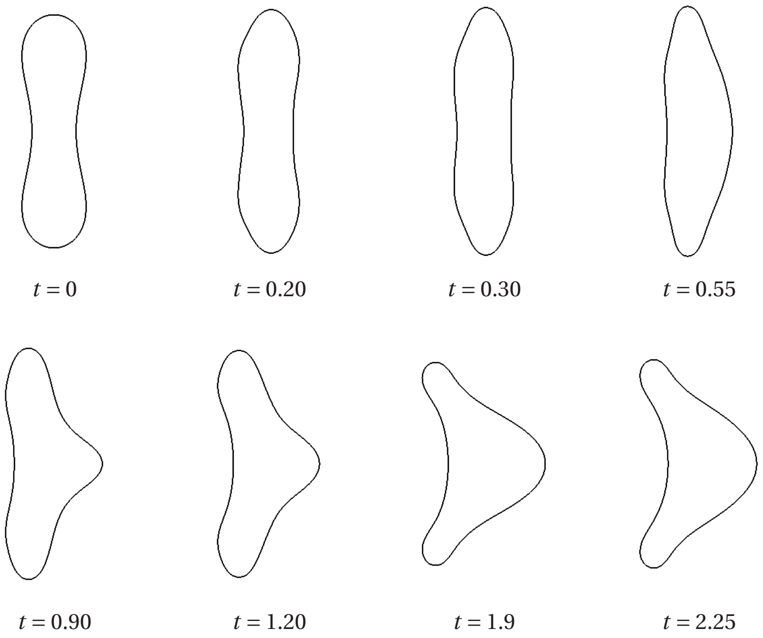


Figure 7.26 – Vesicle in parabolic flow with obstruction $S = 0.5$. Shape of the biomembrane at several time instances.

considered in the equations (since the penalizations cannot overlap). This represents an effective way for shaping the domain Θ “in a weak sense”.

We show in Figure 7.24 the approximated fluid velocity and pressure obtained simulating the coupled problem in the case with obstruction $S = 0.5$. Results regarding the case with obstruction $S = 0.4$ are shown in Figure 7.25. In both the situations, the vesicle at the beginning deforms in the opposite way with respect to the case without obstructions: the upper and lower parts tends slightly to deform under the action of the fluid flowing around the vesicle at rest, while the center is kept behind due to the presence of the narrowing. Soon after, the whole biomembrane starts moving, subject to the forces exerted by the fluid, and tend to assume the parachute shape.

In the case with obstruction $S = 0.5$, with respect to the case without narrowing, the vesicle presents more accentuated tail flaps, due also to the higher fluid velocity caused by the presence of the obstruction [229]. Details about the evolution of the biomembrane shape are shown in Figure 7.26.

In the case with obstruction $S = 0.4$, the deformations of the vesicle at the initial time instances are higher, since the smaller passage obstructs the vesicle from moving until a smaller profile is achieved; in the meanwhile, the fluid bends the extremities of the biomembrane, which then start pushing the frontal part of the vesicle toward the obstruction, leaving behind a little deformation in the rear part which is carried along with the biomembrane. In fact, when the fluid slows down after the narrowing, the vesicle tends to return to the normal parachute shape.

We remark that, having used the same parameters and tolerances also for the fluid-membrane interaction solver, the profile of the number of subiterations performed at each time step is similar between the cases of vesicle under parabolic flow with and without obstruction: the peak of number of subiterations corresponds to the high pressure gradients and the first movements of the vesicle, then it is lowered till a stable value of 3 iterations per time step is reached and kept till the vesicle reaches the end of the channel.

Vesicle in shear flow

We now consider the case of a vesicle immersed in shear flow. The fluid domain Θ is represented by the quadrangular planar surface $(0,1)\times(0,1)\subset\mathbb{R}^2$, sketched in Figure 7.27. A vesicle, represented by the curve $\Omega\in\mathbb{R}^2$, lies in the center of the domain. On the walls Γ_2 and Γ_4 velocities with opposite sign are imposed on the fluid. In particular, we consider the following boundary conditions for the fluid subproblem:

$$\begin{aligned} &\mathbf{u}=\mathbf{e}_x && \text{on } \Gamma_4,\ t\in(0,T),\\ &\mathbf{u}=-\mathbf{e}_x && \text{on } \Gamma_2,\ t\in(0,T),\\ &\boldsymbol{\sigma}\mathbf{n}=\mathbf{0} && \text{on } \Gamma_1\cup\Gamma_3,\ t\in(0,T); \end{aligned} \tag{7.4.29}$$

the conditions on Γ_2 and Γ_4 are applied gradually, by employing the smooth ramping function in Eq. (7.4.26) for $t_0=0$ and $t_1=0.1$. The fluid is initially at rest. By referring to the side of the channel as characteristic length, we consider a flow in the exterior part of the membrane such that $\mathrm{Re}=10^{-4}$; however, the viscosity inside Ω is set to $\frac{1}{10}$ of the external viscosity. The vesicle has perimeter equal to 1.07513 and area 0.04910, with a reduced area equal to 0.53380. As for the other cases considered, the equilibrium shape of the vesicle is approximated initially with the schemes considered in Chapter 6 and then used as initial configuration, at rest, for the approximation of the coupled problem. Additionally, we set $\tilde{\rho}=10^2$, $\tilde{\mu}=10^{-3}$, and $\tilde{k}_c=10^{-4}$.

The fluid domain is represented by a NURBS planar surface built of 128×128 elements, for a total of 16,384 elements, and basis functions of degree $p=2$ and globally C^1-continuous. The biomembrane is represented by a NURBS curve with basis functions of degree $p=3$ and globally C^2-continuous, for a total of 137 elements. The linear system associated with the fluid subproblem is composed by 66,824 DOFs, while the linear system which stems from the membrane subproblem is composed by 560 DOFs. A BDF scheme of order $k=2$ is employed for the time discretization of both subproblems, with time step size $\Delta t=0.001$. To enforce the geometric constraints on the membrane the scheme C-1/ is employed. The resistive term is imposed with constant set to $C_{R_h}=10^{-6}/h$. The same partitioning approach and linear solvers employed for the parabolic flow case are considered.

In Figure 7.29 we show the approximated fluid velocity and pressure fields at several time

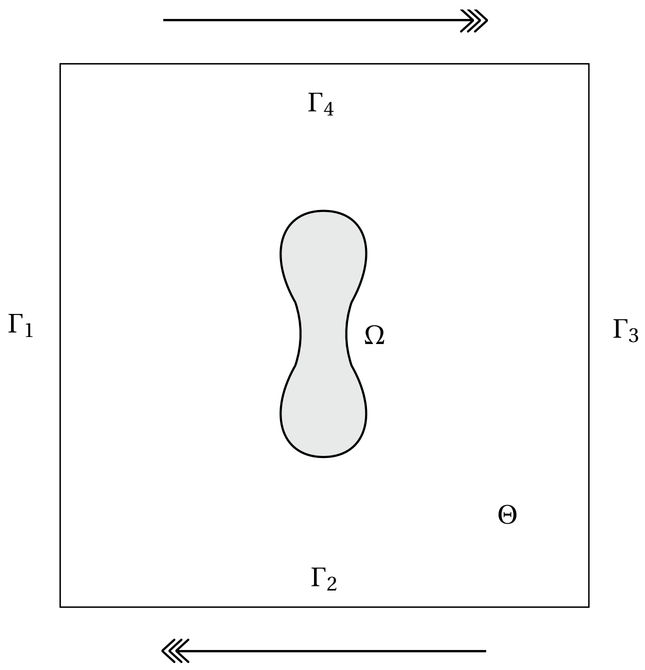


Figure 7.27 – Vesicle in shear flow. Sketch of the setup.

instances, together with the approximated velocity of the membrane. The biomembrane, as soon as the fluid start flowing, is subject to high pressure gradients which impose a rotational movement to the vesicle. This, up to the time interval considered, is in agreement with the behaviors reported in the literature (see e.g. [87, 88]). However, we highlight that the linear solver for the fluid problem starts to underperform in the parallel setting. This behavior is due to a drawback of the resistive method together with the partitioning considered for the fluid domain: if the profile of the smoothed delta function associated with the immersed biomembrane, employed in the resistive term, is such that it overlaps the boundary of a partition, especially when parallel to the boundary, then the local condition number increases drastically; in this cases, the preconditioner becomes less effective and the GMRES struggles to solve the linear system. This aspect merits further investigation, regardless of the application at hand.

Finally, in Figure 7.28 the number of subiterations per time step are reported. These amounts are higher if compared to the number of subiterations employed in the cases with parabolic flow, reported in Figure 7.21. This is mainly due both to the fact that we consider in this case different parameters for the internal and external fluid, which induces a harder convergence at the interface, as well as the more complex dynamics stemming from the shear forces exerted by the fluid, with respect to the pushing forces in the case of parabolic flow.

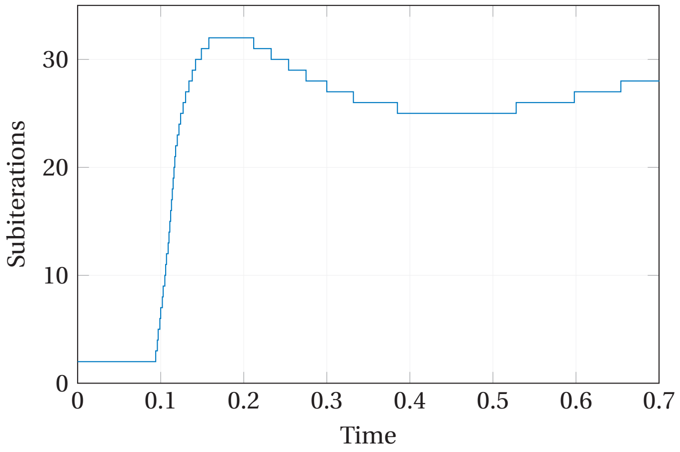
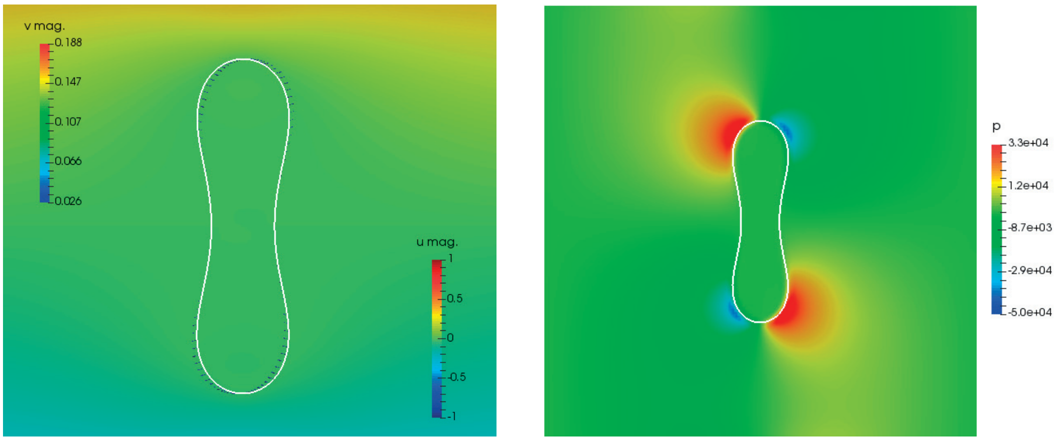
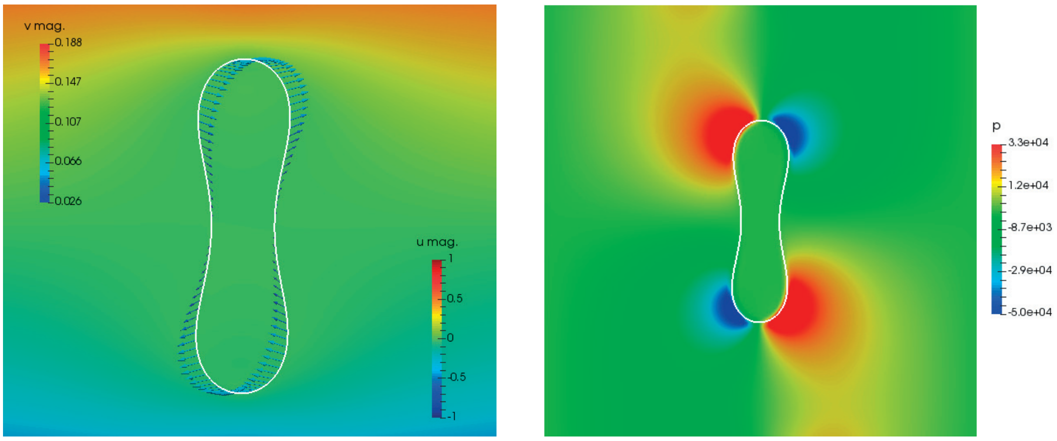


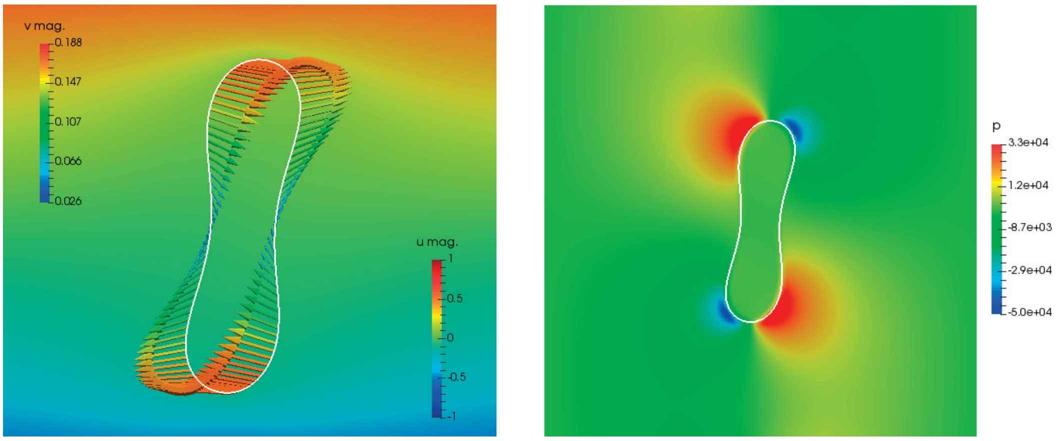
Figure 7.28 – Vesicle in shear flow. Numer of subiterations performed by the coupled solver (zoom for $t \in (0, 0.7)$).



Fluid velocity at $t = 0.058$ Fluid pressure at $t = 0.058$

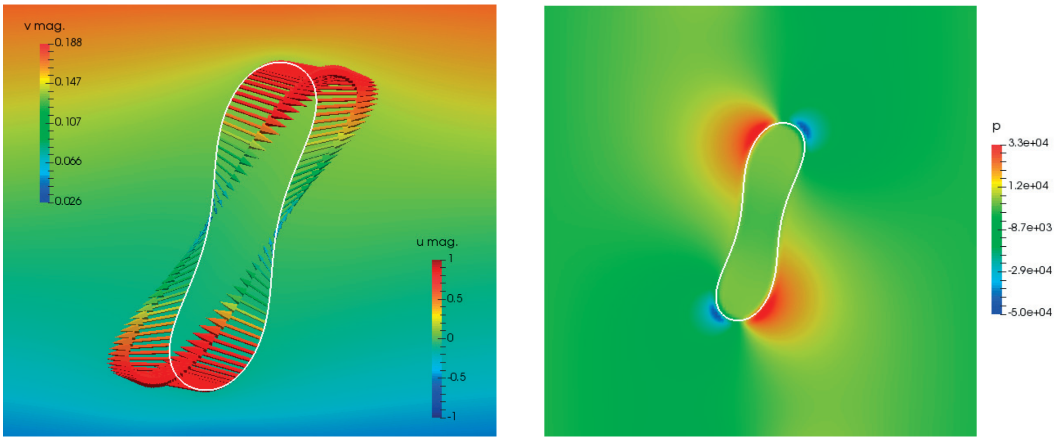


Fluid velocity at $t = 0.098$ Fluid pressure at $t = 0.098$



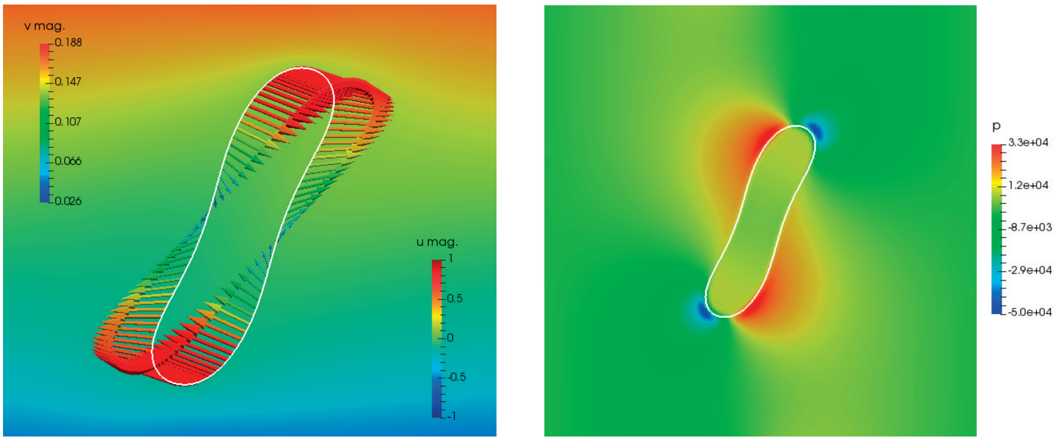
Fluid velocity at $t = 0.238$ Fluid pressure at $t = 0.238$

Figure 7.29 – Vesicle under shear flow. Approximated fluid velocity and pressure at several time instances.



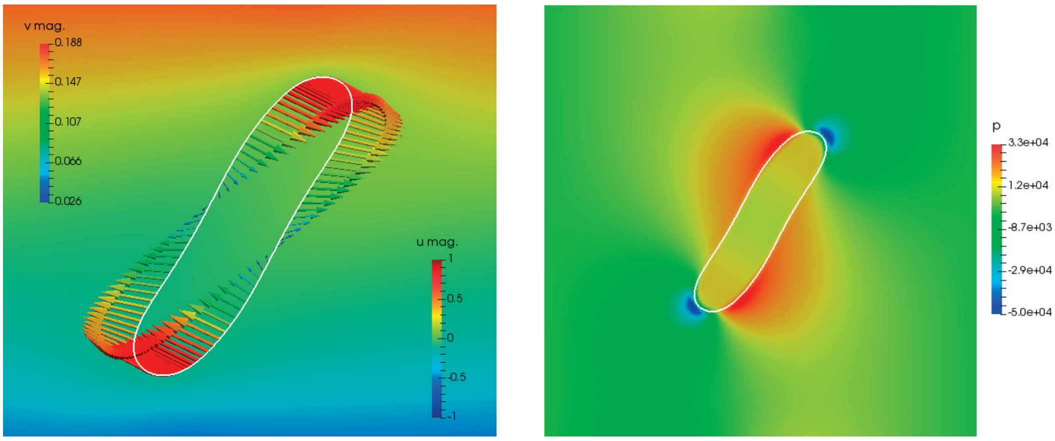
Fluid velocity at $t = 0.378$

Fluid pressure at $t = 0.378$



Fluid velocity at $t = 0.518$

Fluid pressure at $t = 0.518$



Fluid velocity at $t = 0.678$

Fluid pressure at $t = 0.678$

Figure 7.29 – Vesicle under shear flow. Approximated fluid velocity, pressure, and membrane velocity at several time instances.

Conclusions

In this thesis, we have considered the numerical approximation of high order and geometric PDEs defined on surfaces, by means of NURBS-based IGA in the framework of the Galerkin method.

We introduced the mathematical formulation of high order PDEs on surfaces, highlighting the dependence of the differential operator on the underlying features of the manifold. In order to avoid approximating the geometry of the domain, and thus introducing an error in the approximation of the solution due to the geometry, we considered domains represented by NURBS surfaces, in particular single-patch mappings. The choice of NURBS is beneficial both for the exactness of the geometrical representation of geometries of interest, as conic sections, even with a low amount of DOFs, and for the flexibility in tuning the continuity of the basis functions across the computational domain. In this regard, we highlighted how periodic NURBS function spaces can be constructed such that they hold basis functions with high degree of global continuity on closed surfaces. Then, we described a formulation for the spatial discretization of surface high order PDEs using NURBS-based IGA. We highlighted the benefits of employing the same basis functions for both representing the geometry and for approximating the solution, according to the isogeometric concept behind IGA. We considered the numerical approximation of the fourth and sixth order Laplace–Beltrami problems, both on open and closed surfaces, showing numerical results in accordance with the expected theoretical error convergence rates under h-refinement, for different polynomial degrees of the NURBS basis functions, even with singular points in the geometrical mappings. A comparison against the results obtained with the standard isoparametric FEM in mixed formulation has been performed, showing a clear advantage of NURBS-based IGA in terms of number of DOFs required for obtaining a given level of accuracy of the approximation, for this kind of problems. This was highlighted in a further comparison between the NURBS-based IGA and isoparametric FEM discretizations of the Laplace–Beltrami eigenvalue problems of the fourth and sixth orders on the sphere. In this case, the global continuity of the basis functions, together with the exactness of the geometrical representation, played a fundamental role in the accuracy of the approximation. The numerical approximation of two time-dependent phase field models was then considered: firstly the Cahn–Hilliard equation, a nonlinear, fourth order problem modeling the spinodal decomposition of a binary fluid; then the phase field crystal equation, a nonlinear, sixth order model used for studying, for example, the growth of crystals in pure supercooled liquid. Both problems, defined on surfaces, involve high order Laplace–Beltrami operators. The approximation with NURBS-based IGA permitted the direct discretization of the PDEs using globally high order continuous NURBS function spaces, without resorting to mixed formulations. Results have been provided both on open and closed surfaces, showing a dependency of the solution on the underlying geometry. In general, the reported results, regarding the approximation of high order PDEs defined on surfaces, highlighted how NURBS-based IGA is an optimal tool and significantly more efficient than the standard isoparametric FEM for these kind of problems.

Then we considered the numerical approximation of geometric PDEs defined on surfaces, arising from the minimization of shape energy functionals through L^2-gradient flows. In particular, we considered two problems: the mean curvature flow, minimizing the area functional, and the Willmore flow, minimizing the Willmore energy. Both lead to nonlinear geometric PDEs, the first of the second order and the latter of the fourth order. These problems involve high order surface differential operators and the evaluation of geometrical quantities, such as the normal and the curvature. Therefore, the geometrical representation employed to describe the domain is crucial. In this regard, we considered surface domains described by NURBS mappings and spatially discretized the geometric PDEs by means of NURBS-based IGA. This brought several advantages in comparison to standard approaches based on the FEM, due to the exact representation of the geometry, which allowed a formulation where the geometric quantities are directly evaluated, and the possibility to employ NURBS function spaces with basis function featuring the degree of global continuity required for the direct treatment of the high order surface differential operators. In order to fully benefit from the accuracy of the spatial discretization, we proposed a time discretization based on high order BDF schemes. A semi-implicit formulation was adopted, by employing an extrapolation in time, compatible with the BDF scheme, of the geometry and the derived quantities; this was feasible again due to the accurate representation of the geometry. Then we considered the numerical approximation of the mean curvature flow of a sphere, for which the analytical solution is available. We analyzed the effects of the numerical integration and the choice of polynomial degree of the basis functions on the accuracy of the solution: we showed that incrementing the number of quadrature nodes results in a smoothing effect on the behavior of the error, which in general remains small. Moreover, we reported the errors obtained with different orders of the BDF time discretization, showing the expected convergence rates driven by the choice of BDF scheme adopted. The errors corresponding to the spatial discretization were significantly smaller than the errors introduced by the time discretization: this supports the choice of adopting high order time discretization schemes to fully exploit the high spatial accuracy. Results regarding the approximation of the mean curvature flow on tori and cylinders have also been reported and critically discussed, with the final approximated shapes being dependent on the aspect ratio of the initial geometries, as expected. Then we showed results on the numerical approximation of the Willmore flow on an ellipsoid and a torus, for which the final shape at convergence is known; we highlighted again the role of the polynomial degree and continuity of the NURBS basis functions employed for the approximation on the accuracy of the solution. Moreover, an analysis of the linear systems associated with the considered discretizations showed that the condition numbers increase with the polynomial degree of the NURBS basis functions and the amount of DOFs involved, especially when the geometrical mapping presents singularities or the surface degenerates; nevertheless, the accuracy of the solutions is already high enough when coarse discretizations are considered, for which the condition number remains limited. Overall, these numerical tests support the choice of NURBS-based IGA as an accurate and efficient framework for the spatial discretization of geometric PDEs. In this context, the exactness of the geometrical representation and the accurate evaluation of the geometric properties turn out to be advantageous, both from the computational point of view, as a small amounts of DOFs are employed, and for the high accuracy of the approximated solutions.

The approaches proposed for the numerical treatment of the mean curvature and Willmore flow had then been extended in the context of approximating the shapes of lipid biomembranes at equilibrium. In particular, we considered the spontaneous curvature model and formulated the problem as the L^2-gradient flow of the Canham–Helfrich energy, subject to the conservation of total area and volume. We proposed a formulation inline with the Willmore flow with the addition of the two geometric constraints, treated by means of Lagrange multipliers, and then discretized the PDEs by means of NURBS-based IGA and BDF schemes. Regarding the enforcement of the constraints, we proposed two approaches: one based on the iterative scheme to enforce the constraints at discrete level described in [49] and adapted to our discretization approach; the other one relying instead on the enforcement of the constraints on the extrapolated surface. We showed numerical results obtained on initial ellipsoids of different aspect ratio, highlighting the dependence of the numerical solution on the choice of NURBS function space employed for the approximation. The two constraint enforcement schemes were then compared. The iterative scheme ensured, as expected, the conservation of the area and volume within a selected tolerance. The other approach, while being computationally less demanding, introduced a conservation error, since the constraints are applied on the extrapolated surface; nevertheless, thanks to the accuracy of the geometrical description guaranteed by the adoption of NURBS, these errors remain relatively small. A comparison with the existing literature showed that, with the proposed formulations, the equilibrium shapes of lipid biomembranes of different area and volume are accurately achieved as expected by the Canham–Helfrich model, even using a small amount of DOFs.

We considered then a vesicle immersed in a fluid, as e.g. a red blood cell immersed in the plasma, and the corresponding fluid-membrane coupling. In this regard, an incompressible Newtonian fluid, governed by the Navier–Stokes equations, has been considered. We described the resistive method employed for modeling the presence of the vesicle inside the fluid domain. We tested the IGA-based discretization of the Navier–Stokes equations with resistive term and SUPG stabilization in two benchmark cases: considering a lid-driven cavity, we showed agreement in the obtained approximation of the velocity and pressure against reference values; then, in a simulation of the flow past a cylindrical obstacle, we compared the drag and lift obtained with the resistive method against an approximation on a boundary fitted mesh, showing qualitative and quantitative agreement of the forces, with errors on the drag being of about 15% and on the Strouhal number of about 0.1%. Then we proposed algorithms for calculating the signed distance field generated by a NURBS geometry inside a containing domain; this allowed the extension of the resistive method to handle immersed NURBS geometries. Finally, we introduced the coupling of the fluid equations with an equation governing the biomembrane, driven by the forces exerted by the fluid and the internal forces coming from the Canham–Helfrich bending energy. Both the fluid and membrane equations were discretized in space by NURBS-based IGA and in time with BDF schemes, with the coupled system solved by means of a strongly coupled staggered approach. We reported numerical results regarding a vesicle immersed in a fluid, in different flow conditions; the behaviors showed by the biomembrane were in agreement with results reported in the literature.

Overall, this thesis describes a framework for the numerical approximation of high order surface PDEs and geometric PDEs which, relying extensively on the NURBS representation of the geometry and IGA-based spatial discretization, with simple formulations delivers approximations of high accuracy with limited computational costs.

This work can be extended in several ways. First of all, extension to multi-patch NURBS mappings would allow the treatment of more complex geometries, even if particular care would be needed for the construction of multi-patch periodic NURBS function spaces suitable for the discretization of high order operators. Moreover, regarding the approximation of high order geometric PDEs, time step adaptation techniques would be beneficial, as we observed that the speed with which the approximated solution evolves towards the equilibrium may vary significantly over time and could be improved. In the context of the modeling of biomembranes, a development would be the systematic approximation of the equilibrium shapes of vesicles with several values of spontaneous curvature: this is feasible by directly employing the provided formulation, but was not carried on for the inclusion in this thesis. Regarding the coupling with the fluid, an obvious development would be extending the analysis to three-dimensional cases; while the provided results are in 2D, the formulation and the algorithms are already compatible with the extension to the three dimensions. The provided coupled formulation could also be improved both from the point of view of the model, e.g. by introducing the local surface tension or considering a fully fluidic model for the biomembrane, as well as by employing different coupling schemes, or different treatment of the immersed vesicle from the fluid side. Moreover, an in-depth analysis of the behavior of the biomembrane, performed by applying different physical parameters and varying the flow conditions, should be interesting to carry on. Extension to multi-vesicle simulations, within the current framework, could also be a potentially interesting development.

Bibliography

[1] M. Berger, A Panoramic View of Riemannian Geometry, Springer-Verlag, Berlin and Heidelberg, 2003.

[2] S. Timoshenko, S. Woinowsky-Krieger, Theory of Plates and Shells, McGraw–Hill, New York, 1959.

[3] M. Taylor, J. Tribbia, M. Iskandarani, The spectral element method for the shallow water equations on the sphere, Journal of Computational Physics 130 (1) (1997) 92–108.

[4] X. Ding, C. C. Lim, First-order phase transition and high energy cyclonic spots in a shallow water model on a rapidly rotating sphere, Physics of Fluids 21 (4) (2009) 045102.

[5] J. Y.-K. Cho, L. M. Polvani, The emergence of jets and vortices in freely evolving, shallow-water turbulence on a sphere, Physics of Fluids 8 (6) (1996) 1531–1552.

[6] C. D. Cantwell, S. Yakovlev, R. M. Kirby, N. S. Peters, S. J. Sherwin, High-order spectral/hp element discretisation for reaction-diffusion problems on surfaces: Application to cardiac electrophysiology, Journal of Computational Physics 257 (2014) 813–829.

[7] A. S. Patelli, L. Dedè, T. Lassila, A. Bartezzaghi, A. Quarteroni, Isogeometric approximation of cardiac electrophysiology models on surfaces: An accuracy study with application to the human left atrium, Computer Methods in Applied Mechanics and Engineering 317 (2017) 248–273.

[8] R. P. Fedkiw, G. Sapiro, C.-W. Shu, Shock capturing, level sets, and PDE based methods in computer vision and image processing: a review of osher's contributions, Journal of Computational Physics 185 (2) (2003) 309–341.

[9] R. Lai, T. F. Chan, A framework for intrinsic image processing on surfaces, Computer Vision and Image Understanding 115 (12) (2011) 1647–1661.

[10] K. Deckelnick, G. Dziuk, C. M. Elliott, Computation of geometric partial differential equations and mean curvature flow, Acta Numerica 14 (2005) 139–232.

[11] J. W. Barrett, H. Garcke, R. Nürnberg, On stable parametric finite element methods for the Stefan problem and the Mullins–Sekerka problem with applications to dendritic growth, Journal of Computational Physics 229 (18) (2010) 6270–6299.

[12] J. W. Barrett, H. Garcke, R. Nürnberg, Numerical computations of faceted pattern formation in snow crystal growth, Physical Review E 86 (1) (2012) 011604.

[13] H. Garcke, Curvature driven interface evolution, Jahresbericht der Deutschen Mathematiker-Vereinigung 115 (2) (2013) 63–100.

[14] G. Aubert, P. Kornprobst, Mathematical problems in image processing (2002).

[15] T. Chan, L. Vese, Active contour and segmentation models using geometric PDEs for medical imaging, in: R. Malladi (Ed.), Geometric Methods in Bio-Medical Image Processing, Mathematics and Visualization, Springer Berlin Heidelberg, 2002, pp. 63–75.

[16] G. Sapiro, Geometric partial differential equations and image analysis, Cambridge University Press, 2006.

[17] H.-K. Zhao, S. Osher, R. Fedkiw, Fast surface reconstruction using the level set method, in: Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on, IEEE, 2001, pp. 194–201.

[18] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, R. Rusu, A finite element method for surface restoration with smooth boundary conditions, Computer Aided Geometric Design 21 (5) (2004) 427–445.

[19] Y. Duan, L. Yang, H. Qin, D. Samaras, Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces, in: Computer Vision-ECCV 2004, Springer, 2004, pp. 238–251.

[20] D. Liu, G. Xu, A general sixth order geometric partial differential equation and its application in surface modeling, Journal of Information and Computational Science 4 (1) (2007) 129–140.

[21] Q. Zhang, G. Xu, J. Sun, A general sixth order geometric flow and its applications in surface processing, in: Cyberworlds, 2007. CW'07. International Conference on, IEEE, 2007, pp. 447–456.

[22] G. Dziuk, Finite Elements for the Beltrami operator on arbitrary surfaces, in: Partial Differential Equations and Calculus of Variations, Vol. 1357 of Lecture Notes in Mathematics, Springer-Verlag, Berlin and Heidelberg, 1988, pp. 142–155.

[23] G. Dziuk, C. M. Elliott, Surface finite elements for parabolic equations, Journal of Computational Mathematics 25 (4) (2007) 385–407.

[24] M. A. Olshanskii, A. Reusken, X. Xu, A stabilized finite element method for advection-diffusion equations on surfaces, IMA Journal of Numerical Analysis (2013) drt016.

[25] K. Mekchay, P. Morin, R. Nochetto, AFEM for the Laplace–Beltrami operator on graphs: design and conditional contraction property, Mathematics of Computation 80 (274) (2011) 625–648.

[26] M. S. Pauletti, Parametric AFEM for geometric evolution equation and coupled fluid-membrane interaction, Ph.D. thesis, University of Maryland (2008).

[27] P. Morin, R. H. Nochetto, M. S. Pauletti, M. Verani, Adaptive finite element method for shape optimization, ESAIM: Control, Optimisation and Calculus of Variations 18 (4) (2012) 1122–1149.

[28] A. Bonito, R. H. Nochetto, M. S. Pauletti, Geometrically consistent mesh modification, SIAM Journal on Numerical Analysis 48 (5) (2010) 1877–1899.

[29] M. Bertalmío, L.-T. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, Journal of Computational Physics 174 (2) (2001) 759–780.

[30] G. Dziuk, C. M. Elliott, Eulerian finite element method for parabolic PDEs on implicit surfaces, Interfaces and Free Boundaries 10 (119–138) (2008) 464.

[31] A. Rätz, A. Voigt, PDE's on surfaces – a diffuse interface approach, Communications in Mathematical Sciences 4 (3) (2006) 575–590.

[32] M. Astorino, J. Hamers, S. C. Shadden, J.-F. Gerbeau, A robust and efficient valve model based on resistive immersed surfaces, International Journal for Numerical Methods in Biomedical Engineering 28 (9) (2012) 937–959.

[33] M. Reuter, F.-E. Wolter, M. Shenton, M. Niethammer, Laplace–Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis, Computer-Aided Design 41 (10) (2009) 739–755.

[34] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric Analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (39) (2005) 4135–4195.

[35] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009.

[36] L. Piegl, On NURBS: a survey, IEEE Computer Graphics and Applications 11 (1) (1991) 55–71.

[37] L. A. Piegl, W. Tiller, The NURBS Book, Springer-Verlag, Berlin, 1997.

[38] J. A. Cottrell, T. J. R. Hughes, A. Reali, Studies of refinement and continuity in isogeometric structural analysis, Computer Methods in Applied Mechanics and Engineering 196 (41) (2007) 4160–4183.

[39] T. J. R. Hughes, A. Reali, G. Sangalli, Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS, Computer Methods in Applied Mechanics and Engineering 197 (49) (2008) 4104–4124.

[40] A. Tagliabue, L. Dedè, A. Quarteroni, Isogeometric Analysis and error estimates for high order partial differential equations in fluid dynamics, Computers & Fluids 102 (2014) 277–303.

[41] L. Dedè, C. Jäggli, A. Quarteroni, Isogeometric numerical dispersion analysis for two-dimensional elastic wave propagation, Computer Methods in Applied Mechanics and Engineering 284 (2015) 320–348.

[42] T. J. R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, Mineola, 2000.

[43] L. Dedè, A. Quarteroni, Isogeometric Analysis for second order Partial Differential Equations on surfaces, Computer Methods in Applied Mechanics and Engineering 284 (2015) 807–834.

[44] J. Liu, L. Dedè, J. A. Evans, M. J. Borden, T. J. R. Hughes, Isogeometric Analysis of the advective Cahn–Hilliard equation: spinodal decomposition under shear flow, Journal of Computational Physics 242 (2013) 321–350.

[45] H. Gómez, V. M. Calo, Y. Bazilevs, T. J. R. Hughes, Isogeometric Analysis of the Cahn–Hilliard phase-field model, Computer Methods in Applied Mechanics and Engineering 197 (49) (2008) 4333–4352.

[46] H. Gómez, X. Nogueira, An unconditionally energy-stable method for the phase field crystal equation, Computer Methods in Applied Mechanics and Engineering 249 (2012) 52–61.

[47] K. R. Elder, M. Grant, Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals, Physical Review E 70 (5) (2004) 051605.

[48] G. Dziuk, Computational parametric Willmore flow, Numerische Mathematik 111 (1) (2008) 55–80.

[49] A. Bonito, R. H. Nochetto, M. S. Pauletti, Parametric FEM for geometric biomembranes, Journal of Computational Physics 229 (9) (2010) 3171–3188.

[50] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer-Verlag, Berlin and Heidelberg, 2007.

[51] P. Gervasio, F. Saleri, A. Veneziani, Algebraic fractional-step schemes with spectral methods for the incompressible Navier–Stokes equations, Journal of Computational Physics 214 (1) (2006) 347–365.

[52] A. Bonito, R. H. Nochetto, M. S. Pauletti, Dynamics of biomembranes: effect of the bulk fluid, Mathematical Modelling of Natural Phenomena 6 (05) (2011) 25–43.

[53] D. S. Rodrigues, R. F. Ausas, F. Mut, G. C. Buscaglia, A semi-implicit finite element method for viscous lipid membranes, Journal of Computational Physics 298 (2015) 565–584.

[54] J. W. Barrett, H. Garcke, R. Nürnberg, Computational parametric Willmore flow with spontaneous curvature and area difference elasticity effects, SIAM Journal on Numerical Analysis 54 (3) (2016) 1732–1762.

[55] U. Seifert, Configurations of fluid membranes and vesicles, Advances in Physics 46 (1) (1997) 13–137.

[56] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, J. D. Watson, Molecular biology of the cell, 3rd ed., Garland, New York, 1994.

[57] J. T. Jenkins, Static equilibrium configurations of a model red blood cell, Journal of Mathematical Biology 4 (2) (1977) 149–169.

[58] P. B. Canham, The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell, Journal of Theoretical Biology 26 (1) (1970) 61–81.

[59] R. Skalak, A. Tozeren, R. P. Zarda, S. Chien, Strain energy function of red blood cell membranes, Biophysical Journal 13 (3) (1973) 245–264.

[60] W. Helfrich, Elastic properties of lipid bilayers: theory and possible experiments, Zeitschrift für Naturforschung. Teil C 28 (11) (1973) 693–703.

[61] Y. Fung, Biomechanics: Mechanical properties of living tissues, Springer-Verlag, Berlin and Heidelberg, 1993.

[62] M. L. Turgeon, Clinical hematology: theory and procedures, Lippincott Williams & Wilkins, 2005.

[63] S. K. Boey, D. H. Boal, D. E. Discher, Simulations of the erythrocyte cytoskeleton at large deformation. I. Microscopic models, Biophysical Journal 75 (3) (1998) 1573–1583.

[64] D. E. Discher, D. H. Boal, S. K. Boey, Simulations of the erythrocyte cytoskeleton at large deformation. II. Micropipette aspiration, Biophysical Journal 75 (3) (1998) 1584–1597.

[65] W. Pan, D. A. Fedosov, B. Caswell, G. E. Karniadakis, Predicting dynamics and rheology of blood flow: a comparative study of multiscale and low-dimensional models of red blood cells, Microvascular Research 82 (2) (2011) 163–170.

[66] J. Li, M. Dao, C. T. Lim, S. Suresh, Spectrin-level modeling of the cytoskeleton and optical tweezers stretching of the erythrocyte, Biophysical Journal 88 (5) (2005) 3707–3719.

[67] D. Hartmann, A multiscale model for red blood cell mechanics, Biomechanics and Modeling in Mechanobiology 9 (1) (2010) 1–17.

[68] D. A. Fedosov, B. Caswell, G. E. Karniadakis, A multiscale red blood cell model with accurate mechanics, rheology, and dynamics, Biophysical Journal 98 (10) (2010) 2215–2225.

[69] A. Peters, S. Melchionna, E. Kaxiras, J. Lätt, J. Sircar, M. Bernaschi, M. Bison, S. Succi, Multiscale simulation of cardiovascular flows on the IBM Bluegene/P: Full heart-circulation system at red-blood cell resolution, in: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE Computer Society, 2010, pp. 1–10.

[70] D. A. Fedosov, H. Lei, B. Caswell, S. Suresh, G. E. Karniadakis, Multiscale modeling of red blood cell mechanics and blood flow in malaria, PLOS Computational Biology 7 (12) (2011) 1–13.

[71] D. J. Steigmann, Fluid films with curvature elasticity, Archive for Rational Mechanics and Analysis 150 (2) (1999) 127–152.

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

[73] A. Laadhari, P. Saramito, C. Misbah, Vesicle tumbling inhibited by inertia, Physics of Fluids (1994-present) 24 (3) (2012) 031901.

[74] Q. Du, C. Liu, X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, Journal of Computational Physics 198 (2) (2004) 450–468.

[75] Q. Du, L. Zhu, Analysis of a mixed finite element method for a phase field bending elasticity model of vesicle membrane deformation, Journal of Computational Mathematics 24 (3) (2006) 265–280.

[76] X. Wang, Q. Du, Modelling and simulations of multi-component lipid membranes and open membranes via diffuse interface approaches, Journal of Mathematical Biology 56 (3) (2008) 347–371.

[77] R. Fåhræus, T. Lindqvist, The viscosity of the blood in narrow capillary tubes, American Journal of Physiology–Legacy Content 96 (3) (1931) 562–568.

[78] J. B. Freund, Numerical simulation of flowing blood cells, Annual Review of Fluid Mechanics 46 (2014) 67–95.

[79] J. Barrett, H. Garcke, R. Nürnberg, Finite element approximation for the dynamics of asymmetric fluidic biomembranes, Mathematics of Computation 86 (305) (2017) 1037–1069.

[80] H. Noguchi, G. Gompper, Shape transitions of fluid vesicles and red blood cells in capillary flows, Proceedings of the National Academy of Sciences of the United States of America 102 (40) (2005) 14159–14164.

[81] S. K. Veerapaneni, D. Gueyffier, D. Zorin, G. Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, Journal of Computational Physics 228 (7) (2009) 2334–2353.

[82] H. Casquero, C. Bona-Casas, H. Gomez, NURBS-based numerical proxies for red blood cells and circulating tumor cells in microscale blood flow, Computer Methods in Applied Mechanics and Engineering 316 (2017) 646–667.

[83] Y. Kim, M.-C. Lai, Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method, Journal of Computational Physics 229 (12) (2010) 4840–4853.

[84] B. Kaoui, J. Harting, C. Misbah, Two-dimensional vesicle dynamics under shear flow: Effect of confinement, Physical Review E 83 (6) (2011) 066319.

[85] W.-F. Hu, Y. Kim, M.-C. Lai, An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in navier–stokes flows, Journal of Computational Physics 257 (2014) 670–686.

[86] Q. Du, C. Liu, R. Ryham, X. Wang, Energetic variational approaches in modeling vesicle and fluid interactions, Physica D: Nonlinear Phenomena 238 (9) (2009) 923–930.

[87] V. Doyeux, Y. Guyot, V. Chabannes, C. Prud'Homme, M. Ismail, Simulation of two-fluid flows using a finite element/level set method. Application to bubbles and vesicle dynamics, Journal of Computational and Applied Mathematics 246 (2013) 251–259.

[88] A. Laadhari, P. Saramito, C. Misbah, Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods, Journal of Computational Physics 263 (2014) 328–352.

[89] J. W. Barrett, H. Garcke, R. Nürnberg, Numerical computations of the dynamics of fluidic membranes and vesicles, Physical Review E 92 (5) (2015) 052704.

[90] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numerische Mathematik 81 (4) (1999) 497–520.

[91] M. A. Fernández, J.-F. Gerbeau, V. Martin, Numerical simulation of blood flows through a porous interface, ESAIM: Mathematical Modelling and Numerical Analysis 42 (6) (2008) 961–990.

[92] M. Fedele, E. Faggiano, L. Dedè, A. Quarteroni, A patient-specific aortic valve model based on moving resistive immersed implicit surfaces, MATHICSE Report 20.

[93] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Computer Methods in Applied Mechanics and Engineering 197 (1) (2007) 173–201.

[94] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, Y. Zhang, Isogeometric fluid-structure interaction: theory, algorithms, and computations, Computational Mechanics 43 (1) (2008) 3–37.

[95] A. Tagliabue, Isogeometric analysis for reduced fluid-structure interaction models in haemodynamic applications, Ph.D. thesis, Università degli Studi dell'Insubria (2012).

[96] M. A. Fernández, Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit, SeMA Journal 55 (1) (2011) 59–108.

[97] G. Hou, J. Wang, A. Layton, Numerical methods for fluid-structure interaction – a review, Communications in Computational Physics 12 (02) (2012) 337–377.

[98] S. Deparis, Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation, Ph.D. thesis, École polytechnique fédérale de Lausanne (2004).

[99] L. Dedè, H. Garcke, K. F. Lam, A Hele-Shaw-Cahn-Hilliard model for incompressible two-phase flows with different densities, arXiv preprint arXiv:1701.05070.

[100] A. Tagliabue, L. Dedè, A. Quarteroni, Nitsche's method for parabolic partial differential equations with mixed time varying boundary conditions, ESAIM: Mathematical Modelling and Numerical Analysis 50 (2) (2016) 541–563.

[101] A. Tagliabue, L. Dede, A. Quarteroni, Fluid dynamics of an idealized left ventricle: the extended Nitsche's method for the treatment of heart valves as mixed time varying boundary conditions, International Journal for Numerical Methods in Fluids.

[102] C. De Boor, Spline toolbox for use with MATLAB: user's guide, version 3, MathWorks, 2005.

[103] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, et al., An overview of the Trilinos project, ACM Transactions on Mathematical Software (TOMS) 31 (3) (2005) 397–423.

[104] J. Ahrens, B. Geveci, C. Law, C. Hansen, C. Johnson, Paraview: An end-user tool for large-data visualization, The Visualization Handbook (2005) 717.

[105] A. Bartezzaghi, L. Dedè, A. Quarteroni, Isogeometric Analysis of high order partial differential equations on surfaces, Computer Methods in Applied Mechanics and Engineering 295 (2015) 446–469.

[106] A. Bartezzaghi, L. Dedè, A. Quarteroni, Isogeometric Analysis of geometric Partial Differential Equations, Computer Methods in Applied Mechanics and Engineering 311 (2016) 625–647.

[107] A. Bartezzaghi, L. Dedè, A. Quarteroni, Biomembrane modeling with Isogeometric Analysis, SIAM Journal on Scientific Computing.

[108] J. B. Greer, A. L. Bertozzi, G. Sapiro, Fourth order Partial Differential Equations on general geometries, Journal of Computational Physics 216 (1) (2006) 216–246.

[109] M. P. Do Carmo, Differential geometry of curves and surfaces, Vol. 2, Prentice-Hall, Englewood Cliffs (NJ), 1976.

[110] M. C. Delfour, J. P. Zolésio, Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization, SIAM, Philadelphia (PA), 2011.

[111] R. A. Adams, J. J. F. Fournier, Sobolev Spaces, Vol. 140, Academic Press, 2003.

[112] A. Quarteroni, Numerical Models for Differential Problems, Springer, Milan, 2014.

[113] P. C. Fife, Models for phase separation and their Mathematics, Electronic Journal of Differential Equations 2000 (48) (2000) 1–26.

[114] J. W. Cahn, On spinodal decomposition, Acta Metallurgica 9 (9) (1961) 795–801.

[115] J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, The Journal of Chemical Physics 28 (2) (1958) 258–267.

[116] G. N. Wells, E. Kuhl, K. Garikipati, A discontinuous Galerkin method for the Cahn–Hilliard equation, Journal of Computational Physics 218 (2) (2006) 860–877.

[117] O. Wodo, B. Ganapathysubramanian, Computationally efficient solution to the Cahn–Hilliard equation: adaptive implicit time schemes, mesh sensitivity analysis and the 3D isoperimetric problem, Journal of Computational Physics 230 (15) (2011) 6037–6060.

[118] A. N. Kolmogorov, S. V. Fomin, Elements of the Theory of Functions and Functional Analysis, Vol. 2, Graylock Press, Moscow, 1961.

[119] K. R. Elder, M. Katakowski, M. Haataja, M. Grant, Modeling elasticity in crystal growth, Physical Review Letters 88 (24) (2002) 245701.

[120] N. Provatas, J. A. Dantzig, B. Athreya, P. Chan, P. Stefanovic, N. Goldenfeld, K. R. Elder, Using the phase-field crystal method in the multi-scale modeling of microstructure evolution, Journal of the Minerals, Metals and Materials Society 59 (7) (2007) 83–90.

[121] K. A. Brakke, The Motion of a Surface by its Mean Curvature, Princeton University Press, New Jersey, 1978.

[122] G. Huisken, Asymptotic behaviour for singularities of the mean curvature flow, Journal of Differential Geometry 31 (1) (1990) 285–299.

[123] G. Huisken, Flow by mean curvature of convex surfaces into spheres, Journal of Differential Geometry 20 (1984) 237–266.

[124] G. Huisken, Non-parametric mean-curvature evolution with boundary-conditions, Journal of Differential Equations 77 (2) (1989) 369–378.

[125] T. J. Willmore, Riemannian Geometry, Oxford University Press, 1996.

[126] W. Welch, A. Witkin, Free-form shape design using triangulated surfaces, in: Proceedings of the 21st annual conference on Computer Graphics and Interactive Techniques, ACM, 1994, pp. 247–256.

[127] E. Kuwert, R. Schatzle, Gradient flow for the Willmore functional, Communications in Analysis and Geometry 10 (2) (2002) 307–339.

[128] E. Kuwert, R. Schätzle, The Willmore flow with small initial energy, Journal of Differential Geometry 57 (3) (2001) 409–441.

[129] E. Kuwert, R. Schätzle, Removability of point singularities of Willmore surfaces, Annals of Mathematics 160 (2004) 315–357.

[130] G. Simonett, The Willmore flow near spheres, Differential and Integral Equations 14 (8) (2001) 1005–1014.

[131] L. Ambrosio, N. Gigli, G. Savare, Gradient Flows in Metric Spaces and in the Space of Probability Measures, Birkhäuser Basel, 2008.

[132] M. Gage, R. S. Hamilton, The heat equation shrinking convex plane curves, Journal of Differential Geometry 23 (1) (1986) 69–96.

[133] M. A. Grayson, The heat equation shrinks embedded plane curves to round points, Journal of Differential geometry 26 (2) (1987) 285–314.

[134] S. Angenent, Parabolic equations for curves on surfaces, part II: Intersections, blow-up and generalized solutions, Annals of Mathematics (1991) 171–215.

[135] S. J. Altschuler, M. A. Grayson, Shortening space curves and flow through singularities, Journal of Differential Geometry 35 (2) (1992) 283–298.

[136] K. Deckelnick, Weak solutions of the curve shortening flow, Calculus of Variations and Partial Differential Equations 5 (6) (1997) 489–510.

[137] A. Polden, Curves and surfaces of least total curvature and fourth-order flows, Ph.D. thesis, Mathematisches Institut, Universität Tübingen (1996).

[138] G. Dziuk, E. Kuwert, R. Schatzle, Evolution of elastic curves in $\mathbb{R}^n$: Existence and computation, SIAM Journal on Mathematical Analysis 33 (5) (2002) 1228–1245.

[139] O.-Y. Zhong-Can, W. Helfrich, Bending energy of vesicle membranes: General expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders, Physical Review A 39 (10) (1989) 5280.

[140] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, T. W. Sederberg, Isogeometric Analysis using T–splines, Computer Methods in Applied Mechanics and Engineering 199 (5) (2010) 229–263.

[141] M. A. Scott, X. Li, T. W. Sederberg, T. J. R. Hughes, Local refinement of analysis-suitable T–splines, Computer Methods in Applied Mechanics and Engineering 213 (2012) 206–222.

[142] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T–splines and T–NURCCs, in: ACM transactions on graphics (TOG), Vol. 22:3, ACM, 2003, pp. 477–484.

[143] J. Chung, G. M. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method, Journal of Applied Mechanics 60 (2) (1993) 371–375.

[144] K. E. Jansen, C. H. Whiting, G. M. Hulbert, A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method, Computer Methods in Applied Mechanics and Engineering 190 (3) (2000) 305–319.

[145] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (3) (1986) 856–869.

[146] L. Cueto-Felgueroso, J. Peraire, A time-adaptive finite volume method for the Cahn-Hilliard and Kuramoto–Sivashinsky equations, Journal of Computational Physics 227 (24) (2008) 9985–10017.

[147] G. Dziuk, An algorithm for evolutionary surfaces, Numerische Mathematik 58 (1) (1990) 603–611.

[148] J. W. Barrett, H. Garcke, R. Nürnberg, On the parametric finite element approximation of evolving hypersurfaces in $\mathbb{R}^3$, Journal of Computational Physics 227 (9) (2008) 4281–4307.

[149] L. C. Evans, J. Spruck, Motion of level sets by mean curvature I, Journal of Differential Geometry 33 (3) (1991) 635–681.

[150] D. L. Chopp, Computing minimal surfaces via level set curvature flow, Journal of Computational Physics 106 (1) (1993) 77–91.

[151] G. Francis, J. M. Sullivan, R. B. Kusner, K. A. Brakke, C. Hartman, G. Chappell, The minimax sphere eversion, in: Visualization and Mathematics, Springer, 1997, pp. 3–20.

[152] U. F. Mayer, G. Simonett, A numerical scheme for axisymmetric solutions of curvature-driven free boundary problems, with applications to the Willmore flow, Interfaces and Free Boundaries 4 (1) (2002) 89–109.

[153] R. E. Rusu, An algorithm for the elastic flow of surfaces, Interfaces and Free Boundaries 7 (3) (2005) 229–239.

[154] J. W. Barrett, H. Garcke, R. Nürnberg, Parametric approximation of Willmore flow and related geometric evolution equations, SIAM Journal on Scientific Computing 31 (1) (2008) 225–253.

[155] M. Droske, M. Rumpf, A level set formulation for Willmore flow, Interfaces and Free Boundaries 6 (3) (2004) 361–378.

[156] J. W. Barrett, H. Garcke, R. Nürnberg, A parametric finite element method for fourth order geometric evolution equations, Journal of Computational Physics 222 (1) (2007) 441–467.

[157] S. Esedoglu, S. J. Ruuth, R. Tsai, Threshold dynamics for high order geometric motions, Interfaces and Free Boundaries 10 (3) (2008) 263–282.

[158] G. Dziuk, C. M. Elliott, Finite element methods for surface PDEs, Acta Numerica 22 (2013) 289–396.

[159] D. Forti, L. Dedè, Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS–LES modeling in a High Performance Computing framework, Computers & Fluids 117 (2015) 168–182.

[160] G. S. Rao, Numerical Analysis, New Age International Publishers, New Delhi, 2009.

[161] E. W. Weisstein, Minimal surface of revolution. From MathWorld—A Wolfram Web Resource.
URL http://mathworld.wolfram.com/MinimalSurfaceofRevolution.html

[162] U. Seifert, K. Berndl, R. Lipowsky, Shape transformations of vesicles: Phase diagram for spontaneous-curvature and bilayer-coupling models, Physical Review A 44 (2) (1991) 1182.

[163] S. Svetina, A. Ottova-Leitmannová, R. Glaser, Membrane bending energy in relation to bilayer couples concept of red blood cell shape transformations, Journal of Theoretical Biology 94 (1) (1982) 13–23.

[164] S. Svetina, B. Zeks, Bilayer couple hypothesis of red cell shape transformations and osmotic hemolysis., Biomedica Biochimica Acta 42 (11-12) (1983) S86.

[165] S. Svetina, B. Žekš, Membrane bending energy and shape determination of phospholipid vesicles and red blood cells, European Biophysics Journal 17 (2) (1989) 101–111.

[166] W. Helfrich, Blocked lipid exchange in bilayers and its possible influence on the shape of vesicles, Zeitschrift für Naturforschung. Teil C 29 (9-10) (1974) 510–515.

[167] H. J. Deuling, W. Helfrich, Red blood cell shapes as explained on the basis of curvature elasticity, Biophysical Journal 16 (8) (1976) 861.

[168] B. Seguin, E. Fried, Microphysical derivation of the Canham–Helfrich free-energy density, Journal of Mathematical Biology 68 (3) (2014) 647–665.

[169] E. A. Evans, R. Skalak, Mechanics and thermodynamics of biomembranes: part 1., CRC Critical Reviews in Bioengineering 3 (3) (1979) 181–330.

[170] D. Steigmann, E. Baesu, R. E. Rudd, J. Belak, M. McElfresh, On the variational theory of cell-membrane equilibria, Interfaces and Free Boundaries 5 (4) (2003) 357–366.

[171] F. Feng, W. S. Klug, Finite element modeling of lipid bilayer membranes, Journal of Computational Physics 220 (1) (2006) 394–408.

[172] Y. Bazilevs, K. Takizawa, T. E. Tezduyar, Computational fluid-structure interaction: methods and applications, John Wiley & Sons, 2013.

[173] J.-P. Caltagirone, Sur l'intéraction fluide-milieu poreux; application au calcul des efforts exercés sur un obstacle par un fluide visqueux, Comptes rendus de l'Académie des sciences. Série II, Mécanique, physique, chimie, astronomie 318 (5) (1994) 571–577.

[174] T. T. Bringley, Analysis of the immersed boundary method for Stokes flow, Ph.D. thesis, New York University (2008).

[175] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[176] C. Duchanoy, T. R. Jongen, Efficient simulation of liquid–solid flows with high solids fraction in complex geometries, Computers & fluids 32 (10) (2003) 1453–1471.

[177] D. Wan, S. Turek, L. S. Rivkind, An efficient multigrid FEM solution technique for incompressible flow with moving rigid bodies, in: Numerical Mathematics and Advanced Applications, Springer, 2004, pp. 844–853.

[178] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique 8 (2) (1974) 129–151.

[179] C. H. Whiting, Stabilized finite element methods for fluid dynamics using a hierarchical basis, Ph.D. thesis, Rensselaer Polytechnic Institute (1999).

[180] U. Ghia, K. N. Ghia, C. T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, Journal of Computational Physics 48 (1982) 387–411.

[181] C.-H. Bruneau, M. Saad, The 2D lid-driven cavity problem revisited, Computers & Fluids 35 (3) (2006) 326–348.

[182] O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow, Computers & Fluids 27 (4) (1998) 421–433.

[183] F. Auteri, N. Parolini, L. Quartapelle, Numerical investigation on the stability of singular driven cavity flow, Journal of Computational Physics 183 (1) (2002) 1–25.

[184] J. A. Evans, T. J. R. Hughes, Isogeometric divergence-conforming B-splines for the steady Navier–Stokes equations, Mathematical Models and Methods in Applied Sciences 23 (08) (2013) 1421–1478.

[185] P. R. Amestoy, I. S. Duff, J. Koster, J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM Journal on Matrix Analysis and Applications 23 (1) (2001) 15–41.

[186] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel Computing 32 (2) (2006) 136–156.

[187] M. M. Gupta, R. P. Manohar, B. Noble, Nature of viscous flows near sharp corners, Computers & Fluids 9 (4) (1981) 379–388.

[188] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element method for incompressible flows with high Reynolds number, Journal of Computational Physics 229 (23) (2010) 8643–8665.

[189] M. Schäfer, S. Turek, Benchmark computations of laminar flow around a cylinder, in: E. H. Hirschel (Ed.), Flow Simulation with High-Performance Computers II, Springer, 1996, pp. 547–566.

[190] V. John, G. Matthies, Higher-order finite element discretizations in a benchmark problem for incompressible flows, International Journal for Numerical Methods in Fluids 37 (8) (2001) 885–903.

[191] V. John, Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations, International Journal for Numerical Methods in Fluids 40 (6) (2002) 775–798.

[192] B. S. Hosseini, M. Möller, S. Turek, Isogeometric analysis of the Navier–Stokes equations with Taylor–Hood B-spline elements, Applied Mathematics and Computation 267 (2015) 264–281.

[193] J. Bulling, V. John, P. Knobloch, Isogeometric analysis for flows around a cylinder, Applied Mathematics Letters 63 (2017) 65–70.

[194] DFG flow around cylinder benchmark 2D-2, time-periodic case Re = 100, http://www.featflow.de/en/benchmarks/cfdbenchmarking/flow/dfg_benchmark2_re100.html, accessed: 2017-06-04.

[195] V. John, L. Schumacher, A study of isogeometric analysis for scalar convection–diffusion equations, Applied Mathematics Letters 27 (2014) 43–48.

[196] A. Buffa, C. Giannelli, Adaptive isogeometric methods with hierarchical splines: error estimator and convergence, Mathematical Models and Methods in Applied Sciences 26 (01) (2016) 1–25.

[197] D. Schillinger, L. Dede, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, T. J. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces, Computer Methods in Applied Mechanics and Engineering 249 (2012) 116–150.

[198] J. Parvizian, A. Düster, E. Rank, Finite cell method, Computational Mechanics 41 (1) (2007) 121–133.

[199] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, E. Rank, Small and large deformation analysis with the p-and B-spline versions of the finite cell method, Computational Mechanics 50 (4) (2012) 445–478.

[200] D. E. Johnson, E. Cohen, A framework for efficient minimum distance computations, in: Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, Vol. 4, IEEE, 1998, pp. 3678–3684.

[201] L. A. Piegl, W. Tiller, Parametrization for surface fitting in reverse engineering, Computer Aided Geometric Design 33 (8) (2001) 593–603.

[202] M. C. Lin, D. Manocha, Fast interference detection between geometric models, The Visual Computer 11 (10) (1995) 542–561.

[203] P. R. Stay, The definition and ray-tracing of B-spline objects in a combinatorial solid geometric modeling system, Tech. rep., DTIC Document (2013).

[204] W. Martin, E. Cohen, R. Fish, P. Shirley, Practical ray tracing of trimmed NURBS surfaces, Journal of Graphics Tools 5 (1) (2000) 27–52.

[205] O. Abert, M. Geimer, S. Muller, Direct and fast ray tracing of NURBS surfaces, in: Interactive Ray Tracing 2006, IEEE Symposium on, IEEE, 2006, pp. 161–168.

[206] K. Qin, M. Gong, Y. Guan, W. Wang, A new method for speeding up ray tracing NURBS surfaces, Computers & Graphics 21 (5) (1997) 577–586.

[207] A. Efremov, V. Havran, H.-P. Seidel, Robust and numerically stable Bézier clipping method for ray tracing NURBS surfaces, in: Proceedings of the 21st Spring Conference on Computer Graphics, ACM, 2005, pp. 127–135.

[208] C. Ericson, Real-time collision detection, CRC Press, 2004.

[209] M. E. Mortenson, Geometric Modeling, John Wiley & Sons, Inc., New York, NY, USA, 1985.

[210] A. Limaiem, F. Trochu, Geometric algorithms for the intersection of curves and surfaces, Computers & graphics 19 (3) (1995) 391–403.

[211] Y. L. Ma, W. T. Hewitt, Point inversion and projection for NURBS curve and surface: control polygon approach, Computer Aided Geometric Design 20 (2) (2003) 79–99.

[212] E. Dyllong, W. Luther, Distance calculation between a point and a NURBS surface, Tech. rep., DTIC Document (2000).

[213] I. Selimovic, Improved algorithms for the projection of points on NURBS curves and surfaces, Computer Aided Geometric Design 23 (5) (2006) 439–445.

[214] X.-D. Chen, J.-H. Yong, G. Wang, J.-C. Paul, G. Xu, Computing the minimum distance between a point and a NURBS curve, Computer Aided Geometric Design 40 (10) (2008) 1051–1054.

[215] J. Arvo, Graphics gems, in: A. S. Glassner (Ed.), Graphics Gems, Academic Press Professional, Inc., San Diego, CA, USA, 1990, Ch. Transforming Axis-aligned Bounding Boxes, pp. 548–550.
URL http://dl.acm.org/citation.cfm?id=90767.90922

[216] P. M. Vlahovska, T. Podgorski, C. Misbah, Vesicles and red blood cells in flow: From individual dynamics to rheology, Comptes Rendus Physique 10 (8) (2009) 775–789.

[217] T. Jankuhn, M. A. Olshanskii, A. Reusken, Incompressible fluid problems on embedded surfaces: Modeling and variational formulations, arXiv preprint arXiv:1702.02989.

[218] M. Arroyo, A. DeSimone, Relaxation dynamics of fluid membranes, Physical Review E 79 (3) (2009) 031915.

[219] P. Cicuta, S. L. Keller, S. L. Veatch, Diffusion of liquid domains in lipid bilayer membranes, The Journal of Physical Chemistry B 111 (13) (2007) 3328–3331.

[220] R. Dimova, S. Aranda, N. Bezlyepkina, V. Nikolov, K. A. Riske, R. Lipowsky, A practical guide to giant vesicles. Probing the membrane nanoregime via optical microscopy, Journal of Physics: Condensed Matter 18 (28) (2006) S1151.

[221] F. Nobile, C. Vergara, Partitioned algorithms for fluid-structure interaction problems in haemodynamics, Milan Journal of Mathematics (2012) 1–25.

[222] S. Badia, F. Nobile, C. Vergara, Fluid-structure partitioned procedures based on Robin transmission conditions, Journal of Computational Physics 227 (14) (2008) 7027–7051.

[223] C. Förster, W. A. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, Computer Methods in Applied Mechanics and Engineering 196 (7) (2007) 1278–1293.

[224] H. G. Matthies, R. Niekamp, J. Steindorf, Algorithms for strong coupling procedures, Computer Methods in Applied Mechanics and Engineering 195 (17) (2006) 2028–2049.

[225] U. Küttler, W. A. Wall, Fixed-point fluid-structure interaction solvers with dynamic relaxation, Computational Mechanics 43 (1) (2008) 61–72.

[226] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM Journal on Scientific Computing 20 (1) (1998) 359–392.

[227] M. Sala, M. Heroux, Robust algebraic preconditioners with IFPACK 3.0, Tech. Rep. SAND-0662, Sandia National Laboratories (2005).

[228] R. Skalak, P. I. Branemark, Deformation of red blood cells in capillaries, Science 164 (3880) (1969) 717–719.

[229] S. Guido, G. Tomaiuolo, Microconfined flow behavior of red blood cells in vitro, Comptes Rendus Physique 10 (8) (2009) 751–763.

[230] G. Coupier, A. Farutin, C. Minetti, T. Podgorski, C. Misbah, Shape diagram of vesicles in Poiseuille flow, Physical Review Letters 108 (17) (2012) 178106.

ANDREA BARTEZZAGHI

Place and date of birth: Milan, 13th November 1988
Nationality: Italian
Address: Place du Vallon 18, CH-1005 Lausanne, Switzerland
Mobile: +41 76-2327694, +39 334-9064318
E-mail: andrea.bartezzaghi@epfl.ch

CURRENT POSITION

Ph.D. student in **Mathematics** *Since September 2013*
École Polytechnique Fédérale de Lausanne, Switzerland
Chair of Modelling and Scientific Computing (CMCS), Mathematics Department.
Title of the Ph.D. thesis: "*Isogeometric Analysis for high order geometric Partial Differential Equations with applications*".
Supervisor: Prof. A. Quarteroni. Coadvisor: Dr. L. Dedè.

Research interests:
- Scientific calculus, numerical analysis, high performance computing
- Approximation of partial differential equations (PDEs) with the Finite Element Method and NURBS-based Isogeometric Analysis
- High order surface PDEs, geometric evolution PDEs, application to biomembrane modeling

EDUCATION

Master degree in **Mathematical Engineering** *2010 – 2013*
Politecnico di Milano, Italy
Minor in Scientific Calculus
Thesis: "*GPU implementation of a shell element structural solver aimed at fluid–structure interaction problems*". Supervisor: Prof. A. Quarteroni. Final mark: 107/110.

Erasmus exchange program *Fall semester, 2011*
École Polytechnique Fédérale de Lausanne, Switzerland

Bachelor degree in **Mathematical Engineering** *2007 – 2010*
Politecnico di Milano, Italy
Thesis: "*Cardiovascular mathematics: Mathematical models of mass transport across vascular walls*". Supervisor: Prof. P. Zunino. Final mark: 107/110.

High school scientific diploma *2007*
at Liceo Scientifico Statale A. Volta, Milano, Italy

WORK EXPERIENCE

Part-time software development 2009 – 2011
Last Second Ticketing *London, UK*

Development of the administrative portal (in PHP) and the interface with the database (PostgreSQL).

Web site development 2008 – 2011

- Web site design and programming (mainly in PHP, with MySQL database and CSS styles).
- Development of custom content management systems for companies and privates.

LANGUAGE SKILLS

Italian	Mother tongue.
English	Excellent written and oral skills. TOEIC took in 2011, with final mark 975/990. First Certificate in English in 2007.
French	Basic reading, writing and speaking skills.

COMPUTER SKILLS

Operating systems

Excellent knowledge of Linux, Microsoft Windows and Mac OS.

Programming languages

Excellent knowledge of C/C++, Matlab, PHP.
Mobile applications development on Android with Java and JNI.
Parallel programming with MPI and OpenMP, GPU programming with CUDA and OpenCL.
Development of highly optimized parallel applications for scientific calculus. Game development and graphics engine programming with OpenGL and GPU shaders. Programming of low–level operating system components and device drivers. Good knowledge of network protocols, with focus on security. Web development with backend programming in PHP and SQL databases.

Other software

Optimal knowledge of Microsoft Office suite, Microsoft Visual Studio and GNU GCC development environments, MySQL and PostgreSQL databases, Matlab and OpenFOAM scientific calculus software.

Electronics

Low-level hardware programming with PIC, SX, AVR and ARM microprocessors. Good knowledge and experience with the Arduino, ST Nucleo and other platforms. Good experience in RFID technologies and communication protocols. FPGA digital logic programming with VHDL.

TEACHING

- Teaching assistant for the bachelor/master course "*Introduction to the Finite Element Method*" at EPFL, held by Prof. A. Quarteroni (autumn semesters 2013–2014, 2014–2015, 2015–2016, and 2016–2017).

- Teaching assistant for the bachelor course "*Analyse numérique*" at EPFL, held by Prof. A. Quarteroni at EPFL (spring semesters 2013–2014, 2014–2015 and 2015–2016).

PUBLICATIONS

- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Biomembrane modeling with Isogeometric Analysis*". SIAM Journal on Scientific Computing, submitted.
- A. Patelli, L. Dedè, T. Lassila, A. Bartezzaghi, and A. Quarteroni. "*Isogeometric approximation of cardiac electrophysiology models of surfaces: An accuracy study with application to the human left atrium*". Computer Methods in Applied Mechanics and Engineering, 317 (2017): 248–273.
- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Isogeometric Analysis of geometric Partial Differential Equations*". Computer Methods in Applied Mechanics and Engineering, 311 (2016): 625–647.
- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Isogeometric Analysis of high order Partial Differential Equations on surfaces*". Computer Methods in Applied Mechanics and Engineering, 295 (2015): 446–469.
- A. Bartezzaghi, M. Cremonesi, N. Parolini, and U. Perego. "*An explicit dynamics GPU structural solver for thin shell finite elements*". Computers & Structures, 154 (2015): 29–40.

CONFERENCES AND SEMINARS

- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Isogeometric Modeling of Red Blood Cells and their Coupling with Plasma through Immersed Methods*". FEF 2017, Rome, Italy, April 5–7, 2017.
- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Isogeometric Analysis of Geometric Partial Differential Equations: modeling of Red Blood Cells*". IGA 2016, La Jolla CA, USA, October 10–12, 2016.
- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Isogeometric Analysis for the Modeling of Red Blood Cells*". SIMAI 2016, Politecnico di Milano, Milano, Italy, September 13–16, 2016.
- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Isogeometric Analysis for Geometric Partial Differential Equations and Biomembrane Modeling*". IGA 2015, Trondheim, Norway, June 1–3, 2015.
- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*IsoGlib: an Isogeometric Analysis library for the solution of high order Partial Differential Equations on surfaces*". PDESoft 2014, Heidelberg, Germany, July 14–16, 2014.
- A. Bartezzaghi, L. Dedè, and A. Quarteroni. "*Isogeometric Analysis of High-Order Partial Differential Equations on Surfaces*". Swiss Numerics Colloquium 2014, Poster session, Universität Zürich, Zürich, Switzerland, April 25, 2014.
- A. Bartezzaghi and J. Corno. "*Isogeometric Analysis in LifeV*". LifeV Meeting 2014, Politecnico di Milano, Milano, Italy, January 8, 2014.

AWARDS

- First place at national scientific contest "*I giovani e le scienze 2006*", FAST association.
  Representative of Italy at "18^{th} European Union contest for young scientists", Stockholm, Sweden.