Numerical Simulation of Red Blood Cell Electrodeformation

Aurelio Edoardo Spadotto

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'UNIVERSITÉ DE MONTPELLIER

En mathématiques et modélisation

École doctorale: Information, Structures, Systèmes

Unité de recherche: Institut Montpéllierain Alexander Grothendieck

Numerical Simulation of Red Blood Cell Electrodeformation

Présentée par Aurelio Spadotto

Sous la direction de Simon Mendez et Daniele Di Pietro

Le 01-07-2025

Devant le jury composé de

[Lisl WEYNANS, Professeur des Universités, Université de Bordeaux]	[Rapportrice]
[Alexei LOZINSKI, Professeur des Universités, Université de Franche-Comté]	[Rapporteur]
[Vincent MOUREAU, Directeur de Recherche, CNRS]	[Président]
[Jérôme DRONIOU, Directeur de Recherche, CNRS]	[Examinateur]
[Simon MENDEZ, Chargé de Recherche, CNRS]	[Directeur]
[Daniele DI PIETRO, Professeur des Universités, Université de Montpellier]	[Co-directeur]

UNIVERSITÉ DE MONTPELLIER

Remerciements

J'aimerais d'abord remercier les membres du jury, à partir des rapporteurs Pr Weynans et Pr Lozinski pour le temps qu'ils ont dédié à la lecture de ce travail et pour les utiles remarques. Merci aussi à Vincent et Jérôme pour vos conseils au cours de ces années.

Les travaux qui sont exposés dans ce mémoire n'auraient pas vu la lumière si je n'avais pas reçu le soutien et l'assistance de beaucoup de gens que j'ai eu le plaisir de rencontrer á l'IMAG.

Le premier grand merci est destiné à mes directeurs. Merci Simon pour m'avoir initié au monde des simulations des membranes. Merci surtout pour la grande patience avec laquelle tu m'as guidé et encouragé surtout dans les moments plus durs, quand parmi simulations ratées, bugs inextricables et résultats perdus le moral était à terre. Merci Daniele pour tes conseils efficaces qui à plusieurs reprises ont permis de débloquer le travail.

Parmi les chercheurs qui ont pu me conseiller j'aimerais aussi remercier en particulier Franck et Matthieu.

À tous les potes du labo: regarder en arrière et rappeler toutes les belles rencontres faites dans ce laboratoire donne les vertiges. Avec beaucoup de vous tout est commencé en se croisant devant la machine à café un jour quelconque, et souvent on a fini par passer plein de beaux moments ensemble en dehors. Avec certains on se connaît depuis que je ne parlais même pas un mot de français, et ça remonte vu que maintenant je capte la plus part des refs de "Astérix: Mission Cleopatra" et "OSS 117". Beaucoup de vous sont déjà partis de Montpellier il y a un moment, avec d'autres on vient de se connaître. Souvent même avec un mot gentil ou un anecdote rigolo à la pause café vous avez fait la différence dans mes journées sans peut-être le savoir. En vous listant je dessécherais le budget reprographie du labo. En tout cas merci pour toutes les sorties en plage, les randonnées dans les Cévennes, les weekend ski, les soirées pâtes à la maison, les cinémas, le carnaval des matheux et tous les autres moments joyeux passés ensemble. Un mot d'excuse va á tous mes partenaires de coinche: vous méritiez mieux.

Grazie mamma e papà. Inutile dire che senza di voi non sarei arrivato da nessuna parte. Grazie per esserci stati sempre e per continuare a esserci.

Grazie Giuliana per tutto l'affetto che mi hai regalato in questi anni.

Merci Chloé pour tous les beaux moments passés ensemble, pour ton amitié et pour avoir partagé les câlins de la bête féroce qui se balade sur ton canapé.

Merci Tanguy, tu n'es plus à l'IMAG, mais t'as été la première personne que j'y ai rencontré; merci pour toutes les aventures ensemble, les discussions et les choses que tu m'as fait découvrir.

Merci Mathieu; ça ne fait pas longtemps qu'on s'est rencontrés, mais ton soutien dans les derniers mois a été fondamentale. Je suis trop chanceux de t'avoir rencontré.

Préface

Cette thèse porte sur la simulation numérique de systèmes physiques pouvant être décrits comme des capsules entourées par une membrane élastique. De tels objets constituent un modèle souvent adopté en science des matériaux et en bio-ingénierie pour reproduire les processus impliquant des particules colloïdales et des cellules biologiques. Un exemple important est la modélisation de l'écoulement des globules rouges.

Plus particulièrement, l'accent est mis sur l'effet mécanique d'un champ électrique sur la membrane d'une interface souple entre des phases fluides aux propriétés électriques contrastées. Cette étude trouve des applications dans le contexte des cytomètres basés sur des mesures d'impédance ou pour caractériser la rhéologie des suspensions de capsules.

La modélisation mathématique de ce système physique et l'objectif de réaliser une simulation numérique conduisent à un problème d'intérêt général en analyse numérique : l'approximation d'équations aux dérivées partielles comportant une interface interne à travers laquelle les données sont discontinues. Les problèmes d'interface de ce type sont omniprésents dans les modèles physiques et d'ingénierie. Représenter les conditions de saut et l'évolution des interfaces est une tâche fondamentale pour une vaste catégorie de modèles. Ces problèmes apparaissent notamment dans l'étude des écoulements diphasiques, des transitions de phase et de la propagation des fissures. Un autre domaine où les interfaces jouent un rôle central est la biomécanique, en particulier lorsqu'il s'agit d'étudier la dynamique des membranes biologiques.

Outre leur pertinence pour de nombreux modèles physiques, les problèmes d'interface posent également des défis mathématiques importants. Une interface est souvent la région du domaine physique où l'information la plus précieuse est stockée, mais c'est aussi la région où les erreurs numériques les plus importantes sont introduites. Par conséquent, ces problèmes ont suscité un effort considérable de la part de la communauté d'analyse numérique afin de trouver des stratégies de résolution précises mais pratiques.

Lors de la discrétisation des problèmes d'interface, un choix important est de déterminer si le maillage computationnel doit épouser précisément le profil de l'interface ou non. Les méthodes où le maillage suit le profil de l'interface sont dites ajustées (fitted). Ces méthodes ne nécessitent généralement pas de traitement particulier pour les éléments touchant l'interface. En revanche, dans les problèmes instationnaires où le profil de l'interface évolue au cours du temps, un remaillage adaptatif peut s'avérer coûteux.

Pour cette raison, les méthodes non ajustées (unfitted) ont suscité un intérêt croissant. Ces méthodes permettent l'existence d'éléments traversés par l'interface, réduisant ainsi la complexité de la génération du maillage. Toutefois, cela nécessite la mise en place de traitements spécifiques pour les éléments coupés.

Dans ce travail, nous présentons deux approches alternatives pour l'estimation d'un champ électrique discontinu autour d'une membrane biologique. La première approche appartient à la catégorie des méthodes non ajustées. Elle est développée dans le cadre d'un environnement de calcul destiné à la simulation de la dynamique des érythrocytes, appelé YALES2BIO. Il s'agît d'une plateforme de calcul en volumes finis à grande échelle dédiée à la simulation des écoulements biologiques entièrement résolus. La simulation de l'interaction fluide/membrane repose sur la méthode de la frontière immergée (Immersed Boundary Method), l'une des méthodes non ajustées les plus

éprouvées et particulièrement populaire pour la simulation des membranes biologiques. Motivés par la polyvalence de ce solveur, nous proposons une implémentation permettant la représentation d'un champ électrique discontinu basée sur la méthode des fluides fantômes (Ghost Fluid Method).

Bien que l'approche non ajustée soit attrayante pour les raisons mentionnées, un inconvénient majeur est la nécessité de prendre en compte des interfaces qui coupent le maillage de manière défavorable, ce qui peut entraîner un mauvais conditionnement et des instabilités numériques. Pour cette raison, nous proposons également une approche ajustée alternative, basée sur un maillage découpé (cut mesh) constitué d'éléments polygonaux.

D'une part, l'utilisation d'éléments polygonaux réduit la complexité de la génération du maillage et permet la représentation d'interfaces courbes. D'autre part, les solveurs conformes les plus répandus sont limités aux éléments triangulaires/tétraédriques. Cela motive l'introduction d'un schéma hybride, tel que la méthode de De Rham discrète (Discrete de Rham Method).

Dans la seconde partie de cette thèse, nous présentons et analysons un schéma de De Rham discret pour le traitement des problèmes elliptiques avec conditions d'interface, garantissant une robustesse vis-à-vis des contrastes de conductivité et une convergence optimale.

Preface

This dissertation is concerned with the numerical simulation of physical systems that can be described as capsules enclosed by an elastic membrane. Such objects constitute a model that is often adopted in material sciences and bioengineering to emulate the processes involving colloidal particles and biological cells. One important example is the modelisation of the flow of red blood cells. More in particular, a focus is put on the mechanical effect of electric field on the membrane of a soft interface between fluid phases with contrasting electric properties. This enquiry finds applications in the context or cytometers based on impedance measurements or to characterize the rheology of capsule suspensions.

The mathematical modelisation of this physical system and the goal of realizing a numerical simulation leads to a problem of general interest in numerical analysis: the approximation of PDEs featuring internal interface across which data are discontinuous. Interface problems of this type are ubiquitous in models for physics and engineering. Representing jump conditions and evolution of interfaces is a fundamental task for a vast category of models. Interface problems arise when considering multiple phase flows, phase transitions and fissure propagation for example. Another domain in which interfaces play an important role is biomechanics. In particular this is true when considering the dynamics of biological membranes. The leading motivation for this work deals with the simulation of the behavior of human red blood cells subject to electric fields. The interaction between human erythrocytes and electric fields is an important aspect when considering the characterization of blood analysis devices.

Apart from being of the utmost relevance for many different physical models, interface problems also reserve important mathematical challenges. An interface is often the region of the physical domain where the most valuable information is stored, yet it is the region where the most important numerical errors are introduced. Therefore, interface problems have inspired an important effort from the numerical analysis community to find accurate but practical solution strategies.

When dealing with the discretization of interface problems, one important choice is whether the computational mesh should resolve accurately the profile of the interface or not. Methods where the profile of the interface is followed by the mesh are told *fitted*. Fitted methods in general don't need a special treatment for mesh elements touching the interface. On the other hand, especially when dealing with time-dependent problems where the profile of the interface evolves in time, adaptive remeshing can become expensive. For this reason, *unfitted* methods have gained more and more attention. Unfitted methods allow for elements cut by the interface, alleviating the burden of mesh generation. This comes at the cost of designing special treatments for cut elements.

In this work, we present two alternative approaches to the estimation of a discontinuous electric field around a biological membrane. The first one belongs to the category of unfitted methods. It is developed in the framework of a computational environment for the simulation of erythrocyte dynamics called YALES2BIO (Y2B). Y2B is a large-scale finite-volume platform for the simulation of fully resolved biological flows. The simulation of the Fluid/Membrane interaction is based on the Immersed Boundary Method, one of the most well-established unfitted methods, particularly popular for the simulation of biological membranes. Motivated by the versatility of the solver, we propose an implementation for the representation of a discontinuous electric field based on a Ghost Fluid Method.

Even though the unfitted approach is appealing for the aforementioned reasons, one drawback is the need to account for interfaces that cut badly the background mesh. This can be cause of poor

conditioning and numerical instability. For this reason, we also propose an alternative fitted approach, based on a cut mesh featuring polygonal elements. On the one hand, the support of polygonal elements alleviates the burden of mesh generation and allows the representation of curved interfaces. On the other hand, the most popular conforming solvers are limited to triangular/tetrahedral elements. This motivates the introduction of an hybrid scheme, such as the Discrete De Rham Method. In the second part of the thesis, we present and analyze a Discrete De Rham scheme for the treatment of elliptic problems with interface conditions, robust with respect to conductivity contrast and with optimal convergence rate.

CONTENTS

CHAPTER

ONE

INTRODUCTION

Over the last decades, the increased availability of computational power and the rapid development of modern numerical techniques have opened the way to the use of numerical simulation on a large scale as an enquiry tool to study the dynamics of complex physical systems. Capturing the essential features of a system through a model based on partial differential equations and solve them numerically has become a diffused methodology. Among other fields, biology and medicine have been an important source of scenarios to approach through numerical simulation, with many open questions for mathematical modeling and challenges for simulation design.

In particular, hematology is a domain in which numerical simulation can play a decisive role. The subject of study of hematology is blood. Blood is a fluid suspension of which red blood cells, or erythrocytes, constitute 40% of the volume. At larger scales, blood is treated as a liquid flowing in the vessels of the circulatory system. However, when blood flows at the level of vessels with a diameter comparable to that of red blood cells themselves, the dynamics of single erythrocytes and the extent of their deformation becomes important (see Figure (1.1)). Understanding the flow of erythrocytes in microscopic networks is important to model key physiological processes that interest the network of capillaries [Pries and Secomb 2008]. On the other hand, microcirculation of blood in artificial microscopic vessels is an important tool to characterize the mechanical properties of erythrocytes, and therefore it can be exploited for diagnostic purposes [Jiang and Lim 2019].

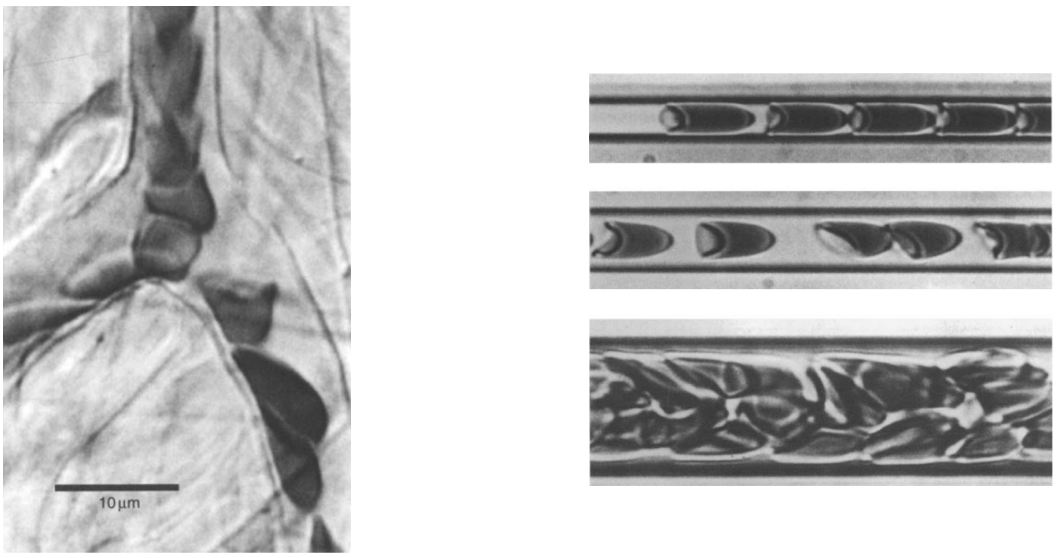


Figure 1.1: After [Pries and Secomb 2008]. Examples of natural and artificial microcirculation. On the left, the flow of red blood cells at the bifurcation of a capillary. On the right, erythrocytes flowing in a tight artificial channel.

The interest in microcirculation has motivated investigations based on mechanical models of erythrocytes that take into account the deformability of the cellular membrane. From this point of view, the dynamics of an erythrocyte is a complex physical process that considers the interaction between the membrane and the flow (for a review, we refer to [Viallat and Abkarian 2019]). As models get more and more sophisticated, predictions of the dynamics have increasingly relied on numerical simulation. The advancements in this field have been tremendous [Bagchi 2007],[Fedosov, Caswell, and Karniadakis 2010] [Ye, Phan-Thien, and Lim 2016], [Kuchel et al. 2021]. At the state of the art, it is possible to realize high-fidelity 3D mappings of real blood vessels networks and simulate the flow of red cells which take into account deformation and interaction with vessel walls.

Despite the remarkable advancements achieved with physical models of erythrocyte flows, several challenges are still open. One important issue is the numerical representation of the interaction between erythrocytes and electric fields. An electric field can have an important effect in deforming the membranes of red blood cells (see Figure (1.2)), and in some flows this effect cannot be neglected by the model. Yet, the electrophisiology of erythrocytes is still little explored by numerical simulation.

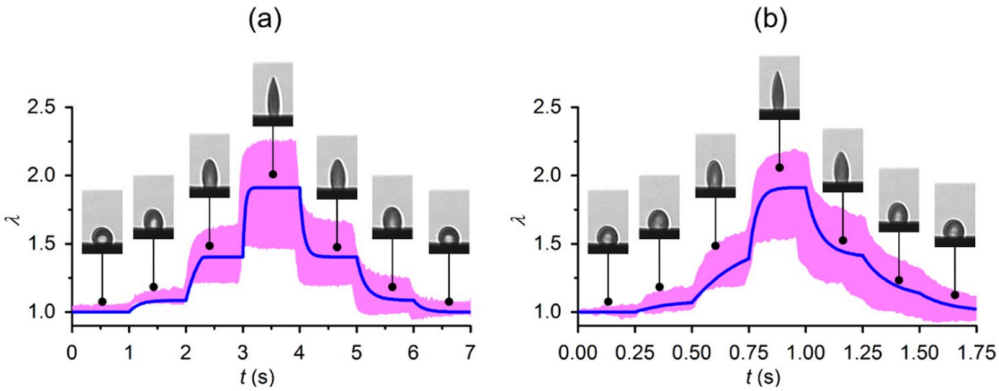


Figure 1.2: After [Qiang et al. 2018] . Response to an electric field of a red blood cell.

A scenario where this is the case regards the flow of erythrocytes in blood analysis devices. A common hemocytometry system is the Coulter counter, whose prototype appears in [Coulter 1949].

The goal of this device is to characterize the blood cells present in a blood sample by detecting electric impulses. The working principle of the cytometer is illustrated in Figure (1.3). A blood sample is injected in a reservoir filled with a fluid solution. Red blood cells suspended in the reservoir are induced to flow singularly towards a second chamber through a thin channel, with a diameter approximately 10 times larger than a typical red cell radius. The channel acts as detection zone. The electric tension between the entrance and the exit of the channel is measured. The passage of a cell provokes a temporary perturbation of the tension that can be processed automatically to characterize the sample. The mechanism is relatively cheap to implement and it constitutes a common system to estimate rapidly the number of cells contained in a sample. In [Kachel, Metzger, and Ruhenstroth-Bauer 1970] an empiric formula is proposed to predict the volume of a cell from the amplitude of the tension peak. Kachel's pioneering work is based on very restrictive hypothesis. Erythrocytes are considered as rigid bodies flowing with negligible rotation. However, the deformation of the cell and the possibility of tumbling trajectories are an influential factor that can pose severe limitation to the interpretability of the signal. One way to solve this problem relies on techniques to prevent uninterpretable cell dynamics. An example is hydrofocalisation, consisting in forcing the trajectory of the cell to the center of the channel. Such techniques can help getting clean signals, however they are complicated to design and increase the cost of implementation. A complementary approach relies on the numerical simulation of an unrestricted cell trajectory. It can be used in combination with machine learning to train models aimed at predicting mechanical parameters of the cells from their electric footprint. Inferring physical properties of erythrocytes from the electric signals is an ambitious task, but appealing for diagnostic testing. Indeed, some pathological conditions such as anemia [Yip et al. 1983] can cause a modification in shape and stiffness of cell membrane.

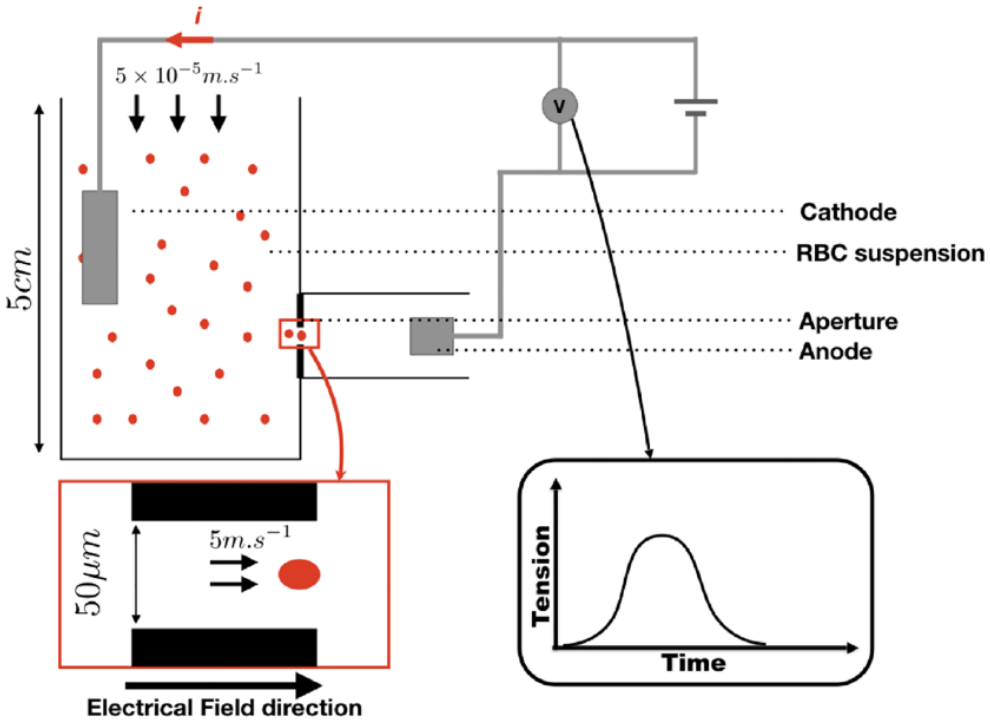


Figure 1.3: After[Taraconat 2020]. The working principle of a Coulter counter.

So far, the effect of an electric field on the trajectory of a cell flowing in cytometers has been neglected in models and simulation. This may have an impact on the signal detected by cytometers, where cells are revealed using of an electric field. Simulating the electrodeformation of red blood cells is important to take into account systematic artifacts introduced by the detection system. This

work goes in the direction of providing a numerical tool to include electrodeformation in erythrocyte simulations. The starting point from a well-established numerical environment for the simulation of red blood cells, the solver YALES2BIO. YALES2BIO is a computing platform developed at IMAG (UMR 5149). It is a general-purpose CFD solver designed to simulate biological flows, in vivo and in-vitro, both at the microscopic scale and at the macroscopic one. Applications of YALES2BIO span several contexts: simulation of cardiac valves [Chnafa et al. 2015], [Ling et al. 2024]; models of platelet adhesion [Valet et al. 2022]; applications to in-silico IRM [MENDEZ et al. 2023] and the study of respiratory flows in transmission of respiratory disease [Mendez, Garcia, and Nicolas 2023].

YALES2BIO is based on the YALES2 solver developed at CORIA (UMR 6614). YALES2 is a massively parallel finite-volume research code for the solution of the Navier-Stokes equations on unstructured meshes.

In the last fifteen years, the YALES2BIO project has been involved with the development of a solver for the microscopic flow of red blood cells with a particular emphasis on flows in cytometers. The results in [Gibaud et al. 2013] show the first application of the red blood cell solver applied to the flow in a channel. In [Afonso, Mendez, and Nicoud 2014] YALES2BIO fluid structure/solver is validated considering benchmark configurations from membrane mechanics literature. The modern version of the solver is presented in [Mendez, Gibaud, and Nicoud 2014] and [Sigüenza et al. 2016]. In [Mauer et al. 2018] and [Mendez and Abkarian 2018], YALES2BIO is adopted to simulate the behaviour of erythrocytes in shear flows. The solver get full maturity with the simulation of complex microfluidic flows of tightly packed red blood cells. In [Mendez et al. 2019], the solver is employed to reproduce the self-organization of groups of erythrocytes in confined channels. Other relevant publications related to YALES2BIO and applications to erythrocyte flows include [Sigüenza, Mendez, and Nicoud 2014], [Loiseau et al. 2015], [Matteoli, Nicoud, and Mendez 2021].

More recently, the solver has already been employed in connection with electricity calculations in cytometry. In [Taraconat et al. 2021], a work in collaboration with HORIBA Medical (Montpellier), YALES2BIO is applied to simulate the electric footprint of a red blood cell and devise a cryterion to filter signals resulting from distorted trajectories. Recent publications on the subject include [Taraconat et al. 2023a]. This pioneering electricity mode has followed the classic approach presented in [Grover et al. 1969] and [Hurley 1970], where the particle flowing in a cytometer is considered as an isolating volume. It is useful to estimate the electric field in the channel, but the mechanical effect on the membrane is not taken into account. This is where this work starts from.

The next step for the simulation of erythrocytes in YALES2BIO consists in adopting a more insightful electric model predicting the appearance of forces that drive a deformation of the membrane. This question addresses a general physical problem: the study of interfaces between media with contrasting electric properties. Many different physical systems can be described as drops of conducting fluid separated by an external flow by a fluid-fluid interface, or a soft membrane. Examples of this category are drops, bubbles and capsules. Biological particles like red blood cells are often treated as members of this broad category of physical objects. These systems share similar characteristics in terms of their response to electric fields. In this work the case of erythrocytes is treated as a subcase of this class.

The electrodynamics of drops, capsules and vesicles has been on its own the object of an important scientific production since the last century. The first developments were motivated by the study of the deformation and bursting of water droplets in electric fields, in order to find a hint for explaining the formation of thunderstorms [Taylor 1964],[Taylor 1966].

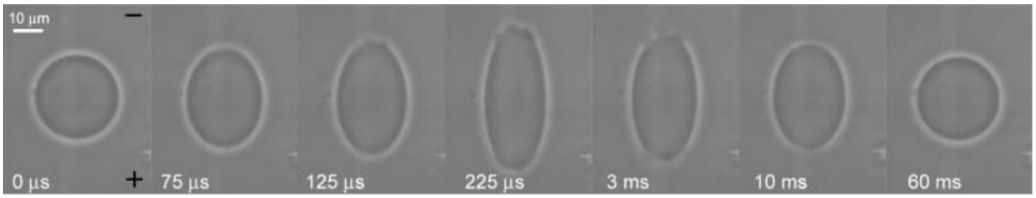


Figure 1.4: After [Riske and Dimova 2005] . Vesicle submitted to electric pulse in the context of a study the electroporation of cell membranes.

Capsules are also studied independently of their role as models for cell behavior. They can be models for encapsulating agents, with applications to drug delivery and other biotechnological applications [Kataoka, Harada, and Nagasaki 2012]. Electric fields also constitute a versatile means to control small-scale fluid and single particle dynamics. An example is the atomization of drops and vesicles to produce micro- and nano-particles [Collins et al. 2008].

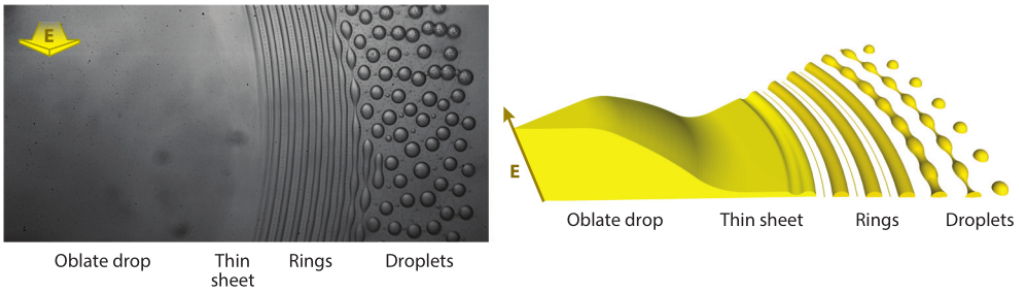


Figure 1.5: After [Vlahovska 2019a]. Atomization induced by application of an electric field in a drop.

The study of single capsule electrodynamics also forms the basis to discuss the effect of collective motions of capsules, therefore providing a model for the rheology of suspensions. In particular, electrohydrodynamics is of interest for the electrorheology of colloidal suspensions, a domain of material physics studying the effect of electric fields on their viscosity [Sheng and Wen 2012].

The kernel of difficulty when simulating electrodeformation is related to the estimation of fields that are discontinous across a moving interface. The contribution of this work consists in two complementary approaches to face these issues. The first is a new module of `YALES2BIO` for electric calculations implemented as a plugin for the erythrocyte solver already existing. The second is based on an original approach hinging on hybrid meshes featuring polytopal elements. The strong point of the first approach consists in the fact that it is tailored on a numerical platform offering advanced state-of-the-art simulations of cell dynamics. However, this is also a source for constraints on the implementation of the electric module. The effort is motivated by the fact that an accurate estimation of electrodeformation on versatile solver like `YALES2BIO` would represent an important breakthrough and would unlock a wide variety of simulation scenarios. The advancements made in this direction are encouraging, but also open questions related to the robustness of the method. An approach based on polytopal elements represents a possible answer. So far, it has been implemented to estimate accurately the evolution of the electric field around a fixed interface. For the future of this project it would be desirable to reach an harmonization of the two approaches that could exploit the robustness of the polytopal approach and the power of the `YALES2BIO` framework.

1.1 Outline of the Thesis

Chapters 2 and 3 present in detail the principles and implementation behind the simulations realized with YALES2BIO. The material is split in two separate chapters in order to highlight the distinction between a physical model of capsule electrodeformation and its numerical implementation on a specific platform.

Chapter 2 contains a description of the state of the art regarding the physics of red blood cells and biological capsules. The simulation of a cell is a multiphysics problem which can be seen as a sum of connected and independent modules. Each of these modules is presented in a dedicated section. In particular, the physical aspects related to the novelties of the thesis are presented in the last section, where the theoretical framework to treat electrodeformation is introduced in the scope of capsule electrohydrodynamics.

Chapter 3 presents in detail the algorithm of the solver YALES2BIO. The chapter is organized in order to create a parallel with the previous one. Therefore, the numerical implementation of each module of the system is presented in a dedicated section. In particular, the numerics of electrodeformation are presented in the final section 3.4.

Chapter 4 presents the performance of the solver on a suite of benchmark configurations. Each section of the chapter is dedicated to a different configuration. Test cases are inspired to well-established scenarios from the literature on capsule electrohydrodynamics. A discussion of the results of the tests and the general performance of the solver is provided at the end of the chapter.

The second part of the thesis is dedicated to some developments related to electrodeformation which have been treated with numerical tools independent of YALES2BIO. Chapter 5 presents a method for the numerical solution of one of the fundamental problems arising from the model: the representation of an electrostatic potential discontinuous across a moving membrane. The method hinges on hybrid meshes with polytopal elements arising along the interface is introduced. The technique is presented as a possible competitive strategy for the treatment of moving interfaces. The stability of the method,and an *a priori* convergence estimate are proved and validated presenting numerical tests.

Chapter (6) presents one possible future direction of the solver: the extension of the techniques introduced in Chapter 5 to design a flow-structure interaction solver based on hybrid meshes.

CHAPTER

TWO

MODELS OF ERYTHROCYTE AND CAPSULE FLOWS

This Chapter presents the mechanical model of red blood cells and more in general biological capsules considered in this work. In Section 2.1, the constitutive relation for the cellular membrane of a capsule is described, with a special emphasis on the mechanical model of erythrocytes. Section 2.3 presents a model of the fluid-structure interaction at the base of the simulations realized with `YALES2BIO`. The first two Sections describe the physical model underlying red blood cell simulations in `YALES2BIO`. Section 2.4 is dedicated to the description of the new phenomena considered in this work, that is the mechanical effect of an electric field acting on a cellular membrane. The model presented is at the base of the electrodeformation computations implemented in the new electrostatic module of `YALES2BIO`.

2.1 Mechanics of Biological Membranes

Erythrocytes are blood cells with a key role in oxygen transportation for the organism. They can be essentially described as drops of fluid enclosed by a cellular membrane. The membrane is flexible and its response to mechanical solicitations is determined by its complex structure (see Figure 2.1). Its surface is made of phospholipids. These molecules are characterized by a hydrophobic tail facing the internal part of the membrane and a hydrophilic head facing the bulk, forming a double layer. The internal side of the membrane is scaffolded by a network of spectrin molecules constituting the cytoskeleton of the cell. The thickness of an erythrocytes membrane is approximately 10 nm. Since the thickness of the membrane is far smaller than the typical 3 μm radius, it can be treated as a 2D surface.

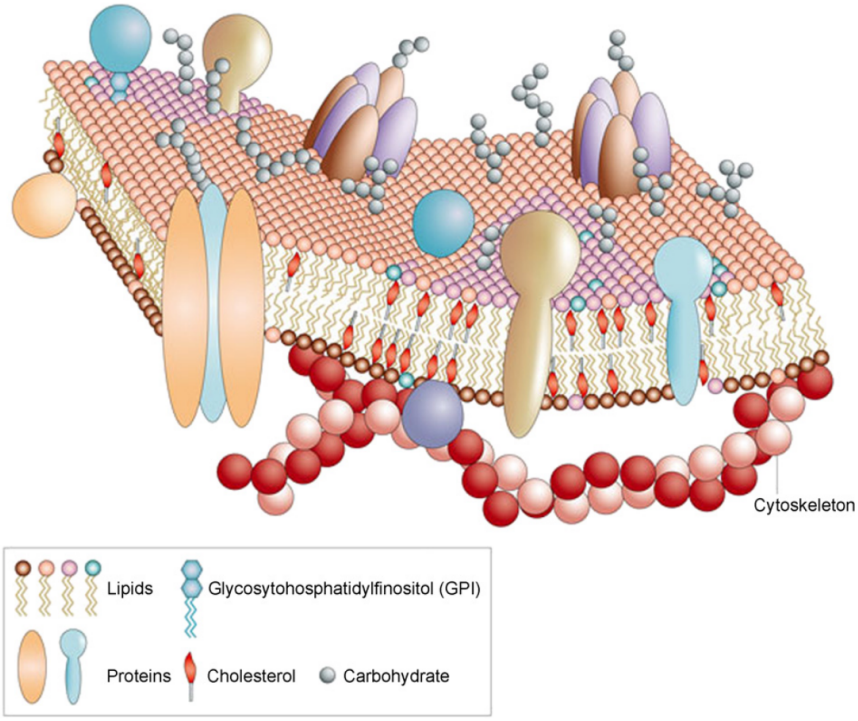


Figure 2.1: Detail of an erythrocyte's cellular membrane. A double sheet of lipids is internally scaffolded by a protein cytoskeleton (after [Vlahovska, Podgorski, and Misbah 2009a]).

This work is based on a continuum model where the membrane is described as a material surface characterized by a constitutive relation reflecting its nanoscopic structure. The definition of a continuum membrane model has been the object of a long-lasting research along the decades. One of the pioneering models was introduced in [Canham 1970], where the rest shape of an erythrocyte is predicted modeling its response to bending. Later, the research for a formulation of membrane energy able to predic the behavior of erythrocytes was deeply influenced by the results presented in [Helfrich 1973a], [Skalak et al. 1973] and [Seifert 1997]. In modern times, numerical simulations of red blood cells can be based on a number of alternative models of membrane energy. For an exhaustive review on the state of the art, we refer the reader to [Viallat and Abkarian 2019]. Modeling of red blood cells is strictly connected to the description of other biological particles like capsules and vesicles. Pioneering works in this field are represented by [Barthes-Biesel 1980], [Barthes-Biesel, Diaz, and Dhenin 2002] and [Barthès-Biesel 2009].

The rest of the Section provides the description of a biological membrane in the context of a continuum model. The general description of a flexible surface is presented, and several alternative mechanical models are introduced. These models reflect simulation options available in `YALES2BIO`, and can be adopted to describe erythrocytes as well as other kind of capsules. Differences between 2D and 3D models are also highlighted.

2.2 Continuum Membrane Model

A membrane can be geometrically represented as closed surface $\Sigma \in \mathbb{R}^d$, with $d \in \{2, 3\}$. The movement of the surface Σ is described in terms of a reference configuration Σ_0, which can represent the configuration of the cell at rest. The time-dependent position $\boldsymbol{x}$ of a material point $\boldsymbol{X} \in \Sigma_0$ is described by a map $\chi : \Sigma_0 \times [0, T] \rightarrow \mathbb{R}^3$, such that:

$$\boldsymbol{x} = \chi(\boldsymbol{X}, t).$$

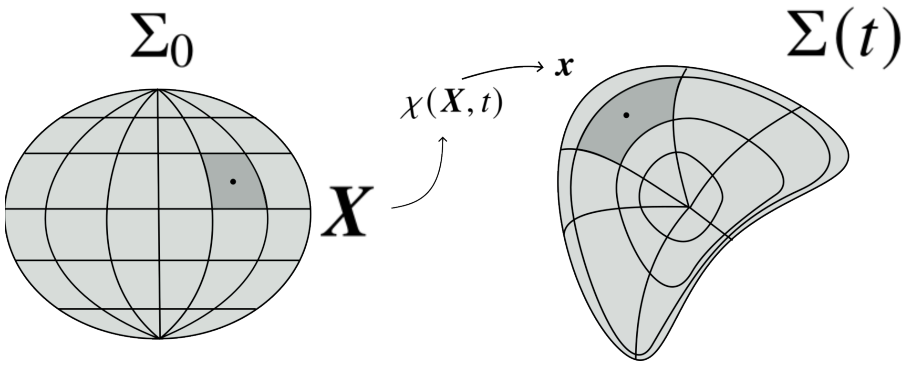


Figure 2.2: Dynamical process of a membrane, reference configuration and actual configuration

With $\boldsymbol{X} \in \Sigma_0$, introduce a local coordinate system (X_1, X_2) and consider the element of surface $dX_1 dX_2$ in the reference configuration. The deformation of the membrane element can be described in terms of two tensors, respectively accounting for in-plane deformations (strain) and out-of-plane deformations (bending). In-plane deformation can be described by the Green-Lagrange strain tensor $\bar{\bar{E}}$ with components:

$$E_{i,j} = \frac{1}{2}(\frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j} - \delta_{i,j}),$$

where $\delta_{i,j}$ are the components of the Identity tensor. Out-of-plane deformation is described by the Hessian H with components:

$$H_{i,j,k} = \frac{\partial^2 x_i}{\partial X_j \partial X_k}.$$

The mechanical response of the membrane can be analysed in terms of fundamental deformation modes. The way any mode of deformation determines the appearance of surface stresses defines the constitutive relation of the membrane.

2.2.1 In-plane deformation

For any deformation, it is possible to find 2 mutually orthogonal directions in the tangent plane of the membrane such that the membrane undergoes a pure stretch. The associated strains are called the principal strains of the deformation [Barthes-Biesel, Diaz, and Dhenin 2002]. The principal strains λ_1, λ_2, such that $E_{ii} = \frac{1}{2}(\lambda_i^2 - 1)$, are called strain invariants. Stresses are typically expressed as a function of the principal strains. An equivalent set of parameters are:

$$\alpha = \lambda_1 \lambda_2 - 1$$
$$\beta = \frac{1}{2}\Big[(\frac{\lambda_1}{\lambda_2} - 1) + (\frac{\lambda_2}{\lambda_1} - 1)\Big].$$

Deformation happening on the plane of the surface element has two fundamental components: isotropic strain and area-preserving shear. When $\lambda_1 = \lambda_2$ the deformation is a purely isotropic strain and $\beta = 0$. When $\lambda_1 = 1/\lambda_2$ the deformation is an area-preserving shear and $\alpha = 0$ (see Figure 2.3). In-plane interactions happen in the plane of the deformed element through forces per unit length or tractions pulling the element along its border. Tractions are described by the tensor $T = T(\lambda_1, \lambda_2)$.

The response of the membrane to isotropic strain is defined by the modulus K_α, whereas the response to shear depends on the shear modulus $G_s = k_{str}/\eta$, where k_{str} is the Young modulus of the membrane and η the membrane thickness.

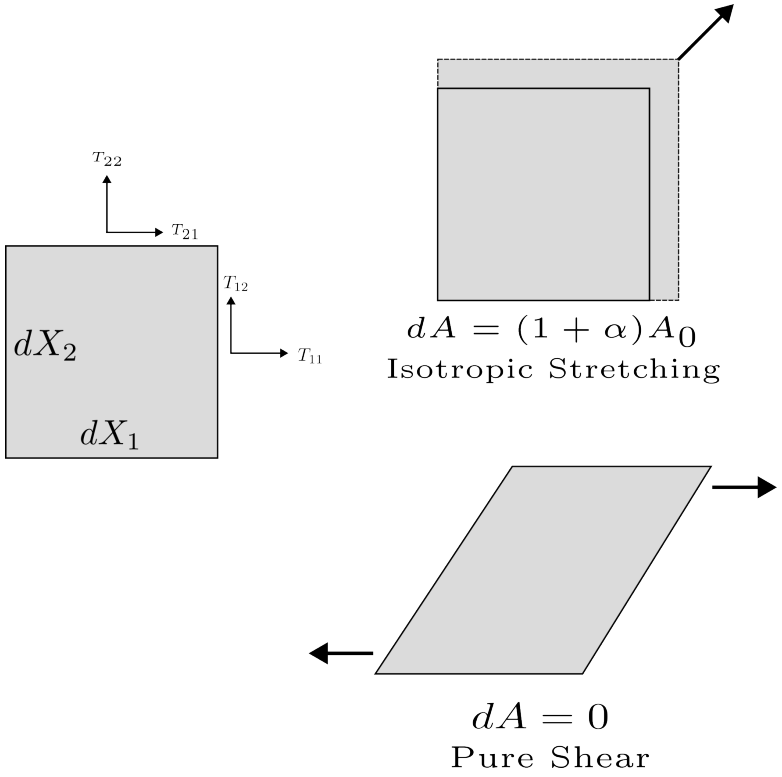


Figure 2.3: Two modes of in-plane deformation for an element of surface.

An hyperelastic model of the membrane is based on the definition of a strain energy density function $W = W(\lambda_1, \lambda_2)$ such that the total energy $\mathcal{E}$ stored in the membrane reads:

$$\mathcal{E} = \int_\Sigma W d\sigma.$$

For a red blood cell membrane, the lipid bilayer gives to the membrane its resistance to change in area and to bending, while the cytoskeleton resists shearing [Mendez and Abkarian 2019]. One important formulation for erythrocytes is the one presented in [Skalak et al. 1973],[Hardeman, Dobbe, and Ince 2001]. This surface model reflects a strain-hardening behaviour and a strong dilation stiffness with respect to shear. With $C = K_\alpha/G_s >> 1$, the Skalak energy density function reads:

$$W_{SK} = \frac{G_s}{4}[(\lambda_1^2 + \lambda_2^2 - 2)^2 + 2(\lambda_1^2 + \lambda_2^2 - \lambda_1\lambda_2 - 1) + C(\lambda_1^2\lambda_2^2 - 1)^2]. \tag{2.1}$$

The corresponding principal tractions read:

$$\begin{aligned} T_{11} &= \frac{G_s}{\lambda_1\lambda_2}\{\lambda_1^2(\lambda_2^2 - 1) + C(\lambda_1\lambda_2)^2[(\lambda_1\lambda_2)^2 - 1] \\ T_{22} &= \frac{G_s}{\lambda_1\lambda_2}\{\lambda_2^2(\lambda_2^2 - 1) + C(\lambda_1\lambda_2)^2[(\lambda_1\lambda_2)^2 - 1]. \end{aligned} \tag{2.2}$$

Skalak's energy model is the reference red blood cell model `YALES2BIO`. Alternative models can also be introduced in order to simulate generic capsules. For the sake of completeness, a list the available strain energy density functions is provided. For a deeper discussion on the effectivenesss of different hyperelastic models in predicting red blood cell behaviour, we refere the reader to [Pepona, Gounley, and Randles 2023].

- Neo-Hookean model [Barthes-Biesel, Diaz, and Dhenin 2002] :

$$W_{NH} = \frac{G_s}{2}(\lambda_1^2 + \lambda_2^2 + \lambda_1^{-2}\lambda_2^{-2} - 3);$$

- Yeoh model

$$W_{YEOH} = W_{NH} + C_3(\lambda_1^2 + \lambda_2^2 + \lambda_1^{-2}\lambda_2^{-2} - 3)^3,$$

where C_3 is a non-linear modulus.

2.2.2 Out of plane deformation

The membrane responds to bending solicitations with forces acting out of plane. Bending energy density can be expressed in terms of membrane curvature. Considering the local family of tangent circles, the principal curvatures C_1 and C_2 are the inverse of the radius of the minimal and maximal circle. The principal moments acting over the bent edges are related to the principal curvatures introducing the bending modulus k_{bnd}:

$$M_{11} = k_{bnd}(C_1 - C_0), \qquad M_{22} = k_{bnd}(C_2 - C_0),$$

where C_0 represents the spontaneous curvature of the surface. Spontaneous curvature can be explained by the difference in surface of the two sides of the lipid bilayer, given to a different shape of the lipids of the two layers or to a difference in chemical environment from inside to outside. Response to bending can be represented by introducing a bending energy F_b. Helfrich-Canham-Evans's formulation of bending energy reads [Helfrich 1973a]:

$$\mathcal{E}_b = \int_\Sigma W_b d\sigma = \frac{k_{bnd}}{2}\int_\Sigma (2H - C_0)^2 d\Sigma, \tag{2.3}$$

where $H = \frac{1}{2}(C_1 + C_2)$ is the mean curvature. Bending traction $\boldsymbol{f}_b$ can be expressed as the first variation of bending energy density W_b with respect to the membrane position Σ [Mendez and Abkarian 2019]:

$$\boldsymbol{f}_b = \frac{\delta W_b}{\delta \boldsymbol{\Sigma}} = k_{bnd}[(2H - C_0)(2H^2 - 2K + C_0 H) + 2\Delta_{LB}H]\boldsymbol{n},$$

where $K = C_1 C_2$ is the Gaussian curvature, Δ_Σ is the Laplace-Beltrami operator and $\boldsymbol{n}$ is the outcoming normal vector.

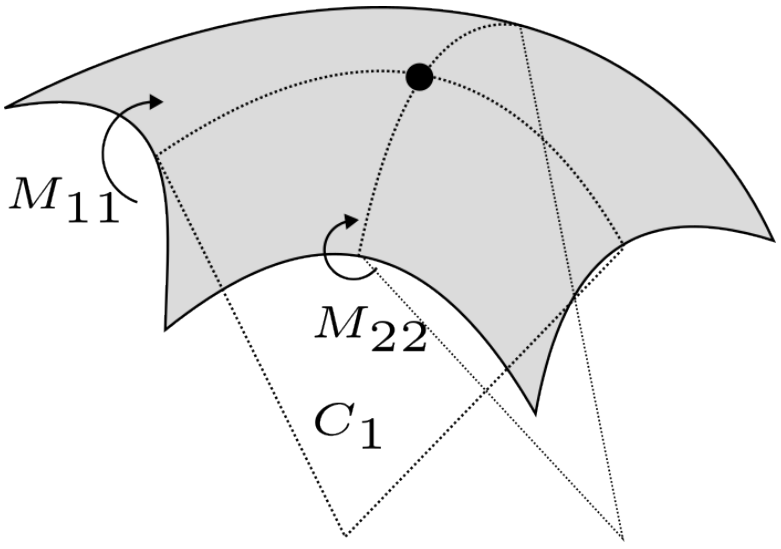


Figure 2.4: Principal curvatures and associated moments.

2.2.3 2D Model

Some of the simulations presented in Section 4 consider a 2D model where the membrane of the cell is represented by a closed curve in the plane. The model is slightly different when reducing dimension. Forces on the membrane act through tensions and in place of an hyperelastic model, an elastic law is considered. An element of length dl_0 in the undeformed configuration transformed into an element dl is subject to the tension:

$$t_{str} = k_{str}\frac{dl - dl_0}{dl_0},$$

where k_{str} is the elastic modulus. The expression of bending tension is reduced to:

$$\boldsymbol{t}_{bnd} \coloneqq k_{bnd}(\frac{1}{2}k^3 + \Delta_{LB}k)\boldsymbol{n},$$

where k and $\boldsymbol{n}$ represent the curvature and normal vector of the curve.

2.3 Fluid Structure Interaction

In order to model the flow of a cell in a suspension, an essential feature of the model is the way in which the elastic membrane interacts with the surrounding environment. In the most typical scenario the cell contains a fluid and is immersed in a fluid. The cell is advected by the flow, and deformation of the membrane implies the appearance of tractions that have an effect on the dynamics of the membrane and of the flow itself.

The representation of moving objects in flows is a large branch of computational mechanics. The Immersed Boundary Method (IBM) is one of the most popular, especially for biological systems. The first appearance of the method can be retraced to the seminal paper [Peskin 1972]. Later works worthy of mention include [Peskin 1977] about the numerical properties of the method and [Kim and Moin 1985], where an implementation similar to the one version in `YALES2BIO` is provided. A thorough discussion of the foundations of the method can be found in [Peskin 2002]. This Section is devoted to exposing the theoretical model underlying the method.

The fluid of the external suspension is a water solution assumed incompressible and characterised by a density ρ_{ext} and a dynamic viscosity μ_{ext}. Likewise, the core of the cell is characterized by the couple of parameters ρ_{int} and μ_{int}. The domain Ω where the flow takes place is partitioned by the surface Σ into the regions Ω_{int} and Ω_{ext}. The flow is decribed introducing a velocity vector field $\boldsymbol{v} : \Omega \to \mathbb{R}^3$ and a pressure scalar field $p : \Omega \setminus \Sigma \to \mathbb{R}$. The membrane is considered as an interface between two fluid phases with different hydrodynamical properties. Introduce $\mu : \Omega \setminus \Sigma \to \mathbb{R}$ and $\rho : \Omega \setminus \Sigma \to \mathbb{R}$ such that:

$$\mu = (\mu_{\text{int}} - \mu_{\text{ext}})\mathbb{I}_{\Omega_{\text{int}}} + \mu_{\text{ext}}, \tag{2.4}$$
$$\rho = (\rho_{\text{int}} - \rho_{\text{ext}})\mathbb{I}_{\Omega_{\text{int}}} + \rho_{\text{ext}}. \tag{2.5}$$

In the Immersed Boundary Method, $\boldsymbol{v}$ and p are considered as extended to the entire domain Ω. The action of the membrane on the flow is represented through a volume force distribution $\boldsymbol{f}_\Sigma : \Omega \to \mathbb{R}^d$. Velocity $\boldsymbol{v}$ and pressure p satisfy Navier-Stokes equations:

$$\begin{cases} \rho\dfrac{D\boldsymbol{v}}{Dt} + \nabla p - \mu\Delta\boldsymbol{v} = \boldsymbol{f}_\Omega + \boldsymbol{f}_\Sigma & \text{(2.6a)} \\ \nabla \cdot \boldsymbol{v} = 0, & \text{(2.6b)} \end{cases}$$

where $\boldsymbol{f}_\Omega : \Omega \to \mathbb{R}^3$ represents an external volume force distribution, taking into account, for example, gravity.

The Immersed Boundary Method establishes a link between surface and volume variables by means of a convolution with Dirac's delta operator δ. The convolution is used to express a double coupling, from the membrane to the flow and viceversa.

The first coupling condition is realised expressing the mechanical surface stress exerted by the membrane onto the flow as an equivalent volume force.
On the membrane, the total force exerted on the surface element $d\Sigma$ reads:

$$d\boldsymbol{\Phi} = \boldsymbol{F}_\Sigma d\Sigma.$$

The transmission condition is imposed by stating that any volume of fluid $\mathcal{P} \in \Omega$ such that $d\Sigma \in \mathcal{P}$ receives from the membrane the force $d\boldsymbol{\Phi}$. Considering the indicator function $\mathbb{I}_\mathcal{P}$ of $\mathcal{P}$, the action:

$$< \delta, \mathbb{I}_\mathcal{P} >= \int_\mathcal{P} \delta(\boldsymbol{x} - \boldsymbol{X}) dV$$

where δ is Dirac's operator centered at $\boldsymbol{X}$ is equal to 1 if and only if $d\Sigma \in \mathcal{P}$. Please, notice that point-wise evaluation of δ is not well-defined, but we present this notation to be consistent with the notations introduced by Peskin.
Therefore, the force $d\boldsymbol{\Phi}$ that is exterted by $d\Sigma$ onto $\mathcal{P}$ can be expressed as:

$$d\boldsymbol{\Phi} = \int_\mathcal{P} \delta(\boldsymbol{x} - \boldsymbol{X}) \big[\boldsymbol{F}_\Sigma(\boldsymbol{X}) d\Sigma\big] dV.$$

The total force $\boldsymbol{\Phi}$ exerted by the membrane onto the volume $\mathcal{P}$ is got by integrating the contributions $d\Sigma$ over the entire surface:

$$\boldsymbol{\Phi} = \int_\Sigma d\boldsymbol{\Phi}(\boldsymbol{X}).$$

The total force $\boldsymbol{\Phi}$ can also be expressed as the integral of the force density $\boldsymbol{f}_\Sigma : \Omega \to \mathbb{R}$ such that:

$$d\boldsymbol{\Phi} = \boldsymbol{f}_\Sigma dV.$$

The balance conditions reads:

$$\int_\mathcal{P} \boldsymbol{f}_\Sigma dV = \int_\mathcal{P} \left(\int_\Sigma \delta(\boldsymbol{x} - \boldsymbol{X}) \boldsymbol{F}_\Sigma(\boldsymbol{X}) d\Sigma \right) dV,$$

which is assumed to hold for an arbitrary portion of volume $\mathcal{P}$. Therefore, the contents of the integrals must coincide:

$$\boldsymbol{f}_\Sigma = \int_\Sigma \delta(\boldsymbol{x} - \boldsymbol{X}) \boldsymbol{F}_\Sigma(\boldsymbol{X}) d\Sigma. \tag{2.7}$$

The coupling (2.7) is called *diffusion* condition, and the surface-to-volume operator:

$$\int_\Sigma \delta(\boldsymbol{x} - \boldsymbol{X})(\cdot) d\Sigma$$

is called diffusion operator.
The introduction of the diffusion operator allows to integrate the action of the membrane without

modifiying the structure of Equations (3.1).

The model is completed with a second kinematic coupling condition. The membrane is supposed to be advected by the flow, such that a microscopic portion of membrane is displaced at the same speed of the surrounding fluid. This no-slip condition can be implemented adopting mapping from the volume to the membrane. Let $\boldsymbol{V} : \Sigma \to \mathbb{R}^d$ be the velocity field over the membrane. Let $P \in \Sigma$ be a material point of the membrane occupying the position $\boldsymbol{x}(P) \in \Omega$. It must hold:

$$\boldsymbol{V}(P) = \boldsymbol{v}(\boldsymbol{X}(P)).$$

Alternatively:

$$\boldsymbol{V}(\boldsymbol{X}) = \int_\Omega \delta(\boldsymbol{X} - \boldsymbol{x})\boldsymbol{v}(\boldsymbol{x})dV. \tag{2.8}$$

The coupling (2.8) constitutes the *interpolation* condition of the Immersed Boundary Method, and the volume-to-surface operator:

$$\int_\Omega \delta(\boldsymbol{X} - \boldsymbol{x})(\cdot)dV,$$

is called interpolation operator.

To resume, the fluid-structure model of the Immersed Boundary method summarizes as:

$$\begin{cases} \rho\dfrac{D\boldsymbol{v}}{Dt} = -\nabla p + \mu\Delta\boldsymbol{v} + \boldsymbol{f}_\Omega + \boldsymbol{f}_\Sigma & \text{Momentum Conservation} \quad (2.9a) \\ \nabla \cdot \boldsymbol{v} = 0 & \text{Incompressibility} \quad (2.9b) \\ \boldsymbol{f}_\Sigma = \displaystyle\int_\Sigma \delta(\boldsymbol{x} - \boldsymbol{X})\boldsymbol{F}_\Sigma(\boldsymbol{X})d\Sigma. & \text{Force Diffusion} \quad (2.9c) \\ \boldsymbol{V}(\boldsymbol{X}) = \displaystyle\int_\Omega \delta(\boldsymbol{X} - \boldsymbol{x})\boldsymbol{v}(\boldsymbol{x})dV & \text{Velocity Interpolation} \quad (2.9d) \end{cases}$$

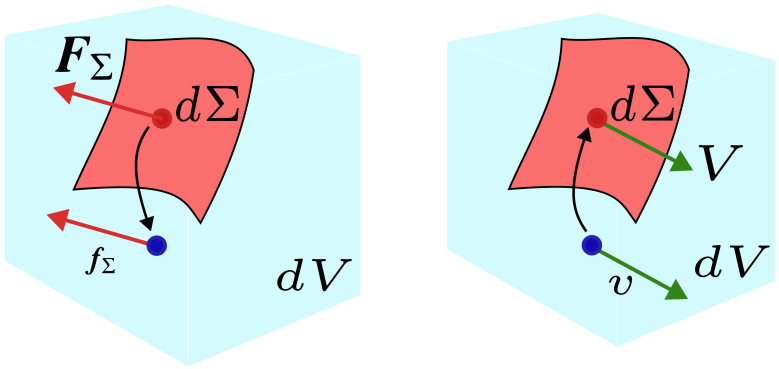


Figure 2.5: Scheme representing the fluid-structure coupling principle. Membrane tractions are transmitted to the fluid in the form of volume forces; flow velocity is transmitted to the membrane. The coupling is relazed by convoluting with Dirac's δ as kernel.

2.4 Electrohydrodynamic model

The response of the membrane with respect to an external electric field is modeled in the framework of electrohydronynamics (EHD). In general, EHD can be considered as a model of the electro-magnetic

field effect in a continuum mechanics. For a general introduction to the subject, with a we refer the reader to [Taylor 1966] and [Saville 1997] .

The electro-magnetic field is described by the celebrated Maxwell equations [Maxwell 1873]. In an electrokinetic theory, charged species are described by space-velocity distributions, whose evolution is modeles through Boltzmann'equation [Pitaevskii and Lifshitz 2012]. The first-principle electro-kinetic theory introduces a complex system of variables varying in space, time, and velocity spectrum, regulated by partial differential equations that are demanding to approach analytically or numerically. A simplified model is rather sought, leveraging on dimensional considerations and homogeneization techniques. In the regime of interest for capsule dynamics [Saville 1997], magnetic effects can be generally disregarded, and the fluctuation of electric charge in the bulk is also negligible. The electric field does not exert a force on the flow but at the interface between media with different electric properties, that is along the membrane.

The description of such interfacial forces is expressed by the Leaky Dielectric Model (LDM), developed by J.R.Melcher and G.I. Taylor [Taylor 1966] . The first apperances of the model arise in the context of meteorology, to give account of drop bursting in strong electric fields as an explaining mechanism for the formation of thunderstorm [Wilson and Taylor 1925] , [O'Konski and Gunther 1955] , [Taylor 1964] and [Torza, Cox, and Mason 1971] . The LDM has then become the standard to describe more complex interfaces, where contrasting fluids are separated by deformable membranes, supporting hyperelastic and bending energies. In particular, it has been coupled with mechanical membrane models to describe the behaviour of capsules and vesicles. In [Das and Thaokar 2018] , the behaviour of a spherical capsule in a DC field is described with a numerical treatment of the LDM for different mechanical constitutive laws, among which the Skalak/Helfrich model. In [McConnell, Miksis, and Vlahovska 2013], [Kolahdouz and Salac 2015] and [Vlahovska 2019a] a prediction of the dynamics is given in the context of vesicles.

In this work, LDM is adopted as a starting model for the description of the electric behaviour of red blood cells.

Both the internal and external media are electric conductors, supporting a distribution of free electric charge. The motility of free charge in the conductors is expressed by the medium conductivity (σ [Sm m^{-1}]). Free charge is put in motion in conductors by the presence of an electric field. Linear conductors are electric media for which there exists a linear relationship between the intensity of the electric field $\boldsymbol{E}$ and the generated current density $\boldsymbol{J}$. Linear conductors are described by Ohm's Law:

$$\boldsymbol{J} = \sigma \boldsymbol{E}.$$

In an electrohydrodynamic model, the effect of the electric field on a volume of fluid is expressed in terms of a surface force, taking into account the distribution of free charge and electric dipoles in the bulk. In particular, electric stress is described by Maxwell tensor:

$$\bar{\bar{T}}_{MW} = \epsilon(\boldsymbol{E}\boldsymbol{E} - \frac{1}{2}\boldsymbol{E}\cdot\boldsymbol{E}\bar{\bar{I}}).$$

where ϵ ([Farad m^{-1}]) is the permittivity of the medium, and $\bar{\bar{I}}$ is the identity. The stress exerted on an oriented parcel of fluid surface $d\boldsymbol{\Sigma}$ is expressed as:

$$d\boldsymbol{f}_{MW} = \bar{\bar{T}}_{MW}\cdot d\boldsymbol{\Sigma}.$$

Following [Taylor 1966] , the charge relaxation time $t_{cr} = \frac{\epsilon}{\sigma}$ expresses the time scale after which a distribution of charge decays in a conducting medium. If compared to the membrane elastic relaxation

time, t_c is negligible. The internal and external media can be considered as electrically neutral. Introducing the scalar electric potential ϕ such that $\boldsymbol{E} = \nabla\phi$, the description of the electro-magnetic field reduces to the condition:

$$\nabla \cdot (\sigma_\bullet \nabla\phi) = 0 \qquad \bullet \in \{\text{int}, \text{ext}\}, \tag{2.10}$$

holding in the respective medium.

Since the media support no volume charge, $\nabla \cdot \boldsymbol{E} = 0$ in the bulk. Moreover, the LDM supposes ϵ as medium-wise constant. As a consequence:

$$\bar{\bar{T}}_{MW} = 0$$

in the bulk, so that a fluid parcel not in contact with the interface is not subject to an electric force. On the other hand, a surface element on the interface with normal $\boldsymbol{n}$ is subject to a traction related to the jump of the normal component of $\bar{\bar{T}}_{MW}$. If the interface is supposed to have no inertia, the conservation of linear momentum along the interface reads:

$$[\bar{\bar{T}}_{MW}]_E \cdot \boldsymbol{n} [\bar{\bar{T}}_{HD}]_E \cdot \boldsymbol{n} = \boldsymbol{f}_\Sigma,$$

where $\bar{\bar{T}}_{HD}$ is the hydrodynamic stress tensor and $\boldsymbol{f}_\Sigma$ is a characteristic interface traction, defined according to the constitutive relation of the interface. In the case of the interface between two immiscible fluids, $\boldsymbol{f}_\Sigma$ is represented by surface tension, but more complex models can be taken into account, as detailed is Section 2.1.

The kind of interaction between the interface and free charge flowing between the two media determines interface conditions that complete equations 2.10 to determine the electric field in the domain. Therefore, an electric constitutive relation of the interface is provided. Different interface models can be considered according to the ability of the permeability of the membrane to charges. For example, fluid-fluid interfaces like drops are completely permeable to electric charge, but physical membranes like lipid bilayers can prevent the flow of charge. For this tractation, we consider a membrane model based on [Seiwert and Vlahovska 2013]. The permeability of the membrane is modeled introducing its conductance G (conductivity per unit surface). Since the conductivity of the membrane is several orders of magnitude lower of the conductivity in the bulk, a surface distribution of charge determines along it. Therefore, the membrane acts as a capacitor, characterized by a capacitance C. The surface charge distribution determines a jump J_Σ of electric potential:

$$[\phi]_E = J_\Sigma.$$

The current flowing through the membrane element ds with normal $\boldsymbol{n}$ from the interior sides reads $di_{\text{int}} = \sigma^{\text{int}} \boldsymbol{E}^{\text{int}} \cdot \boldsymbol{n}$ and it must match the current $di_{\text{ext}} = \sigma^{\text{ext}} \boldsymbol{E}^{\text{ext}} \cdot \boldsymbol{n}$ from the external side, such that:

$$[\sigma \nabla\phi]_E \cdot \boldsymbol{n} = 0.$$

The evolution in time of the potential jump J_Σ is decribed in terms of the conductance and capacitance of the membrane. For biological membranes [Schwalbe, Vlahovska, and Miksis 2011] [DeBruin and Krassowska 1999] , it is described introducing a charge balance condition:

$$C\partial J_{\Sigma,t} + G_\Sigma J_\Sigma = \sigma_{\text{int}} \boldsymbol{E}_{\text{int}} \cdot \boldsymbol{n} = \sigma_{\text{ext}} \boldsymbol{E}_{\text{ext}} \cdot \boldsymbol{n}. \tag{2.11}$$

The first term of the left hand side represents the growth rate of the amount of charge of both signs cumulated on either side of the interface (CJ_Σ). The second term expresses the current flowing across the interface itself. On the right hand side, the current flowing from the bulk towards the interface is expressed in terms of the electric field.

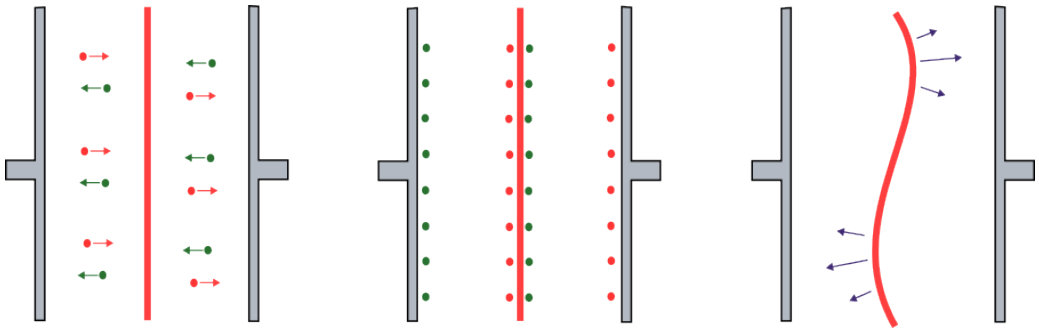


Figure 2.6: Schematic representation of a charge-impermeable interface between conducting media subject to electric field. In electrohydrodynamics, the action of the electric field over a distribution of charges and dipoles in a continuous medium is expressed by the Maxwell tensor. Under the hypotheses of the Leaky Dielectric Model, these stresses act only across interfaces between media with unmatching electric properties.

CHAPTER

THREE

ELECTRODEFORMATION WITH YALES2BIO

In this chapter we present the implementation of electrodeformation calculations in YALES2BIO. As pointed out in the introduction, YALES2BIO is a large numerical platform where the simulation of erythrocyte microflows is only one of several possible applications. In this context, the name YALES2BIO will be used to refer specifically to the Fluid-Structure Interaction solver constituting the kernel of red blood cell calculations. In the vast panorama of Fluid-Structure solvers, YALES2BIO can be categorized as a low-order solver where both the flow and the surface are represented by meshes. Fluid dynamics computations are based on a Finite Volume discretization and the coupling between the membrane and the flow is implemented in the spirit of the Immersed Boundary Method.

The calculation of electric forces is implemented in a separate module called ESS, such that the electrodeformation simulations are realized with an integrated solver YALES2BIO+ESS. In ESS, the membrane is treated as an interface between different electric media. In order to comply with the constraints imposed by YALES2BIO, ESS implements an *unfitted* approach where the interface is not embedded in the structure of the volume mesh.

The first two sections of this chapter present the starting framework into which ESS is plugged, YALES2BIO's kernel. Section 3.1 describes the representation of the structural mechanics of the membrane and the calculation of hyperelastic and bending forces. Section 3.2 is dedicated to CFD aspects and it presents the fluid-structure interaction in the solver.

The electric module ESS is presented in section 3.4. The chapter is concluded with a scheme summarising the entire numerical pipeline.

3.0.1 Notation Convention

The choices for the notation in this chapter are made so to reflect with as much fidelity as possible the actual implementation in the code.

As a starting point, YALES2BIO can be seen as a simulator where two separate geometric and physical entities interact. On the one hand there is the membrane of the cell, treated as a closed void surface. On the other hand there is the control volume where the membrane moves, which hosts a flow and the electric field.

Membranes are represented in YALES2BIO by a class SURFACE. A SURFACE-type object *de facto* implements the membrane as a mesh of the $(d-1)$-dimensional surface representing the position of

the membrane. The membrane configuration is described by a family of connected facets/edges along with markers for the vertices. Given a surface Σ, the SURFACE Σ_h in the most simplified way is intended as a couple $(\mathcal{T}, \mathcal{V})$, where $\mathcal{T}$ is the set of surface elements (or faces) and $\mathcal{V}$ is the collection of vertices. In YALES2BIO connectivities are mostly implemented top-to-bottom. Each surface element is encoded as the list of indices of its vertices. SURFACES are *Lagrangian* meshes. The vertices are associated to material points, whose position can be tracked during the duration of the simulation. Therefore, the position of vertices is updated at each time iteration, implementing the trajectory of the membrane. The movement of a SURFACE is not rigid. The positions of vertices are independent, but the connectivity of vertices is preserved and it is not recalculated. Therefore, every face $\mathcal{T}$ can undergo a deformation associated to a storage of (hyper)-elastic energy.

An object of class SURFACE constitutes the infrastructure to represent numerically membrane variables. In Y2B, membrane fields such as the position of the markers or membrane tensions are described as objects of the type SURFACE_DATA. Once an instance of type SURFACE is declared, it can be equipped with a SURFACE_DATA object by specifying its rank (CONSTANT, SCALAR, VECTOR or TENSOR) and associated object (FACE, EDGE, NODE). Any data is represented by a single (possibly multi-component) value over each mesh entity.

The second geometric structure of YALES2BIO is the control volume. The domain $\Omega \in \mathbb{R}^d$ is represented by the mesh $\Omega_h = (\mathcal{T}_\Omega, \mathcal{V}_\Omega)$, where $\mathcal{T}_\Omega$ is a collection of volume elements, not necessarily tetrahedral, with vertices belonging to $\mathcal{V}_\Omega$. The mesh domain Ω_h is implemented by the class GRID. GRIDS are *Eulerian* meshes. The position of the vertices is fixed during the simulation.

Considering a GRID, fields defined over the fluid domain are represented by the class GRID_DATA. Similarly to SURFACE_DATA, GRID_DATA represent collection of scalar or vector values over the corresponding geometric entities. The gamma of data type in descending order of dimension of the geometric entity is: ELEM, FACE, PAIR and NODE.

In the descripton of the several indexed quantities are introduced, where the index can assume several orders of meaning. To avoid introducing cumbersome lists of indices for every variable considered, we adopt a vectorized notation reflecting the distinction between datas associated to the SURFACE or the GRID. Each variable is associated to a latin or greek letter. Upper-case letters indicate variables whose domain is a surface, and lower-case letters variables defined over the volume domain. The style of the letter defines the rank of the variable. Vector quantities are written with boldcase font and tensor quantities with a double overline. GRID_DATA and SURFACE_DATA are written with an underline. Arrays and matrices are noted between brackets. From time to time, arrays with vector components are introduced. Dependance from time, when not omitted, is expressed by a superscript denoting the iteration number.

Example: with Ω and Σ respectively a volume domain and surface, consider a discretization by assigning a SURFACE $\Sigma_h = (\mathcal{T}_\Sigma, \mathcal{V}_\Sigma)$ and a GRID $\Omega_h = (\mathcal{T}_\Omega, \mathcal{V}_\Omega)$. The velocity of the surface is a field $\boldsymbol{V} : \Sigma \to \mathbb{R}^d$. Its discrete counterpart is the SURFACE_DATA of type VECTOR $\underline{\boldsymbol{V}} = \{\boldsymbol{V}(V) : V \in \mathcal{V}_\Sigma\}$, to be intended as a collection of vector values over each vertex of the SURFACE. On the other hand, the eulerian velocity of the flow is a field $\boldsymbol{v} : \Omega \to \mathbb{R}^d$. Its discrete counterpart is the GRID_DATA of type VECTOR $\underline{\boldsymbol{v}} = \{\boldsymbol{v}(V) : V \in \mathcal{V}_\Omega\}$. Another example is the electric potential $\phi : \Omega \setminus \Sigma \to \mathbb{R}^d$, discretized as the GRID_DATA of type SCALAR $\{\phi(V) : V \in \mathcal{V}_\Omega\}$. A tensor $\bar{\bar{S}} : \Omega \to \mathbb{R}^{d,d}$ is represented with its cartesian components by the matrix $\{S\} = \{S_{ij} : i, j \leq d\}$. The position of the vertices of a surface element $T \in \mathcal{T}_\Sigma$ make up the vector-valued array $\{\boldsymbol{X}\} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3\}$.

3.1 Membrane Forces

The first step in YALES2BIO consists in computing the force fields acting over the membrane. Membrane force $\underline{\boldsymbol{F}} = \{\boldsymbol{F}(V) : V \in \mathcal{V}_\Sigma\}$ is a SURFACE_DATA of type VECTOR allocated at nodes. In YALES2BIO, membrane force is the sum of two components:

$$\underline{\boldsymbol{F}} = \underline{\boldsymbol{F}}_{EL} + \underline{\boldsymbol{F}}_{BND},$$

where $\underline{\boldsymbol{F}}_{EL}$ is the (hyper)-elastic component, and $\underline{\boldsymbol{F}}_{BND}$ is the bending component, associated to Helfrich-type energy.

The implementations for the 2D and the 3D case differ. This difference is not only numeric, but also modelistic. In the solver YALES2BIO, bidimensional membranes are described by a linear elastic law, and the elastic component can be calculated directly with respect to the strain of edges. Tridimensional surfaces can support a range of different hyperelastic laws (see 2.2.1). In this case the determination of forces is based on the Principle of Virtual Works, and related to the variation of hyperelastic energy over each element.

3.1.1 2D Membranes

In dimension $d = 2$, the membrane is represented by a closed curve Γ. In YALES2BIO Γ is represented as a 1-dimensional SURFACE, that is a polygon $\Gamma_h = \{\mathcal{E}, \mathcal{V}\}$, where $\mathcal{E}$ is the collections of edges and $\mathcal{V}$ the set of vertices (see Figure 3.1). Let $\mathcal{E}_V$ be the set of edges sharing the vertex V. In order to determine the elastic forces acting over the vertices, the membrane is treated as a system of springs connected end-to-end. Consider the edge $E \in \mathcal{E}$ with reference length $|E|_0$. With $|E|$ being the length of the edge after a transformation, the tension acting at the extremities of the E reads:

$$T_E = k_{el}\frac{|E| - |E|_0}{|E|_0},$$

where k_{el} is the elastic modulus of the membrane. Consider a vertex $V \in \mathcal{V}$. Let $\mathcal{E}_V = (E^-, E^+)$, with $\mathcal{V}_{E^-} = (V^-, V)$ and $\mathcal{V}_{E^+} = (V, V^+)$. Consider the positions $\boldsymbol{X}_V, \boldsymbol{X}^-, \boldsymbol{X}^+$ of the three vertices. Vertex V is subject to a total recall force:

$$\boldsymbol{F}_{EL,V} = T_{E^-}\frac{\boldsymbol{X}^- - \boldsymbol{X}_V}{|\boldsymbol{X}^- - \boldsymbol{X}_V|} + T_{E^+}\frac{\boldsymbol{X}^+ - \boldsymbol{X}_V}{|\boldsymbol{X}^+ - \boldsymbol{X}_V|}.$$

Bending force acts along the normal to the surface, and depend on the curvature. The discrete curvature $\underline{C}$ is a SURFACE_DATA allocated at nodes, estimated by applying a 3-point approximation. Consider $V \in \mathcal{V}$ and let $\boldsymbol{E}^- = (\boldsymbol{X}_V - \boldsymbol{X}^-)$ and $\boldsymbol{E}^+ = (\boldsymbol{X}^+ - \boldsymbol{X}_V)$. The turning angle θ_V is estimated as:

$$\theta_V = \frac{\boldsymbol{E}^- \cdot \boldsymbol{E}^+}{|\boldsymbol{E}^-||\boldsymbol{E}^+|}.$$

Curvature is then found as:

$$C_V = \frac{\theta_V}{\frac{1}{2}(|\boldsymbol{E}^-| + |\boldsymbol{E}^+|)}.$$

Curvature force is given by:

$$\boldsymbol{F}_{BND,V} = k_b(\frac{1}{2}C_V^3 + \Delta_{LB,h}(C)_V)\boldsymbol{N}_\Gamma,$$

where k_b is the bending modulus of the membrane, $\boldsymbol{N}_\Gamma$ is the normal to the surface calculated as:

$$\boldsymbol{N}_V = \frac{1}{2}\mathrm{rot}_{\pi/2}(\boldsymbol{E}^- + \boldsymbol{E}^+),$$

and $\mathrm{rot}_\vartheta : \mathbb{R}^2 \to \mathbb{R}^2$ is a clock-wise rotation of angle ϑ. The discrete Laplace-Beltrami operator $\Delta_{LB,h}$, that is the Laplace operator over the curve, is calculated with a centered finite difference scheme:

$$\Delta_{LB,h}(C)_V = \frac{2}{|\boldsymbol{E}^-| + |\boldsymbol{E}^+|}\left(\frac{C^+ - C_V}{|\boldsymbol{E}^+|} - \frac{C_V - C^-}{|\boldsymbol{E}^-|}\right).$$

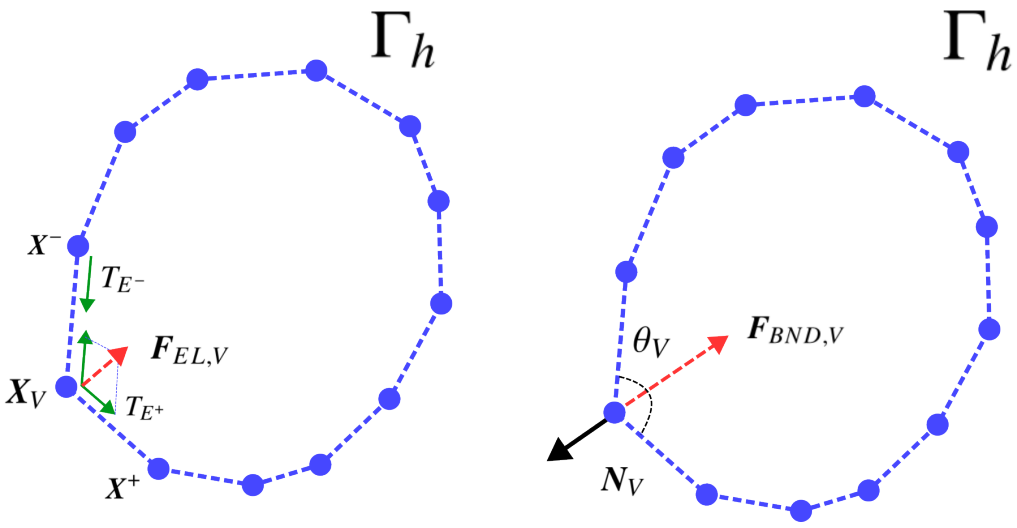


Figure 3.1: Components of the membrane force acting on the discrete curve Γ_h.

3.1.2 3D Membranes

Let $\Sigma_0 \subset \mathbb{R}^3$ be a closed surface representing the reference state of the membrane. In `YALES2BIO`, a tridimensional `SURFACE` is discretized as polyhedron $\Sigma_{0,h}$ with triangular faces (see Figure 3.2).

Let $\Sigma_{0,h} = \{\mathcal{T}, \mathcal{V}\}$, where $\mathcal{T}$ is the collection of triangular faces with cardinality $N_\mathcal{T}$, and $\mathcal{V}$ is the set of vertices, with cardinality $N_\mathcal{V}$. Let $\mathcal{V}_T$ be the set of vertices of the element T.

Since the discrete surface is piece-wise planar, its configuration is determined by tracking the position of its vertices. In particular, the reference configuration of the membrane is encoded by the `SURFACE_DATA` $\underline{\boldsymbol{X}^0} = \{\boldsymbol{X}(V) : V \in \mathcal{V}\}$. Given a membrane transformation $\chi : \Sigma_0 \to \mathbb{R}^3$, a discrete transformation is defined by considering a new `SURFACE` Σ_h where the positions of the vertices are updated as $\underline{\boldsymbol{X}} = \{\chi(V) : V \in \mathcal{V}\}$ and the connectivity of the original `SURFACE` is conserved, such that each triangular element T with vertices $\{V_1, V_2, V_3\}$ is transformed into a new triangle, whose vertices are $\{\chi(\mathcal{V}_1), \chi(V_2), \chi(V_3)\}$.

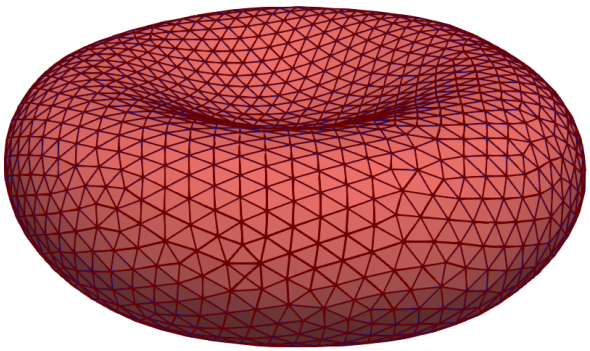


Figure 3.2: The discrete surface $\Sigma_{0,h}$, representing the reference configuration of a red blood cell in Y2B.

To `SURFACE_DATA` that are defined vertex-wise, it is possible to associate polynomials of degree 1 defined on the elemebnts in the spirit of the Conforming Finite Element Method. Consider a triangular element $T = \{\boldsymbol{X}_k\}_{k\leq 3} \in \mathcal{F}$, and a field $\phi \in \mathcal{P}^1(T)$ where $\mathcal{P}^r(T)$ denotes the space of polynomials of degree up to r over T. Let $\boldsymbol{S} \in T$ be a point in the triangle, with vertices located at $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3\}$. Introduce the barycentric coordinates or shape functions $(\gamma_1, \gamma_2, \gamma_3)$ of $\boldsymbol{S}$, such that:

$$\begin{aligned}\boldsymbol{S} &= \gamma_1 \boldsymbol{X}_1 + \gamma_2 \boldsymbol{X}_2 + \gamma_3 \boldsymbol{X}_3\\ 1 &= \gamma_1 + \gamma_2 + \gamma_3.\end{aligned}$$

The value of the field in the internal point $\boldsymbol{S}$ can be expressed as a linear combination of the values assumed at the vertices as:

$$\phi(\boldsymbol{S}) = \gamma_1 \phi(\boldsymbol{X}_1) + \gamma_2 \phi(\boldsymbol{X}_2) + \gamma_3 \phi(\boldsymbol{X}_3).$$

The barycentric coordinates can be calculated with respect to coordinates of $\boldsymbol{S}$ and the coordinates of the vertices as:

$$\begin{aligned}\gamma_1 &= \frac{(\boldsymbol{X}_2 - \boldsymbol{S}) \times (\boldsymbol{X}_3 - \boldsymbol{S})}{(\boldsymbol{X}_2 - \boldsymbol{X}_1) \times (\boldsymbol{X}_3 \boldsymbol{X}_1)} = \frac{\text{area}(\boldsymbol{S}, \boldsymbol{X}_2, \boldsymbol{X}_3)}{\text{area}(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3)}\\ \gamma_2 &= \frac{(\boldsymbol{X}_3 - \boldsymbol{S}) \times (\boldsymbol{X}_1 - \boldsymbol{S})}{(\boldsymbol{X}_2 - \boldsymbol{X}_1) \times (\boldsymbol{X}_3 \boldsymbol{X}_1)} = \frac{\text{area}(\boldsymbol{S}, \boldsymbol{X}_3, \boldsymbol{X}_1)}{\text{area}(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3)}\\ \gamma_3 &= 1 - \gamma_1 - \gamma_2.\end{aligned}$$

Hyperelastic Force

The algorithm to calculate hyperelastic forces is essentially based on the ideas from [Charrier, Shrivastava, and Wu 1989] and adapted to `YALES2BIO` as in [Taraconat 2020]. It is based on an application of the Principle of Virtual Works at a discrete level. For each element, the variation of hyperelastic energy associated to a virtual displacement is considered a a function of the displacement of each vertex by decomposing the Green-Lagrange deformation tensor with respect to the barycentric coordinates. Forces at nodes are associated so that the total virtual work balances the virtual variation of energy. This provides an expression for the forces applied from an element to each of its vertices which can be calculated starting as a function of of nodal displacement.

Consider a transformation $\chi : \Sigma_0 \to \mathbb{R}^3$ and the corresponding discrete configuration $\underline{\boldsymbol{X}}$. Mechanical models of hyperelastic membranes express laws where an energy functional depends on the strains

in the principal deformation direction. The state of deformation is represented by the Green-Lagrange tensor $\bar{\bar{E}}$.

Consider an element $T \in \mathcal{T}$ such that the vertices at rest are located at $\{\boldsymbol{X}_1^0, \boldsymbol{X}_1^0, \boldsymbol{X}_3^0\}$, and introduce the local displacement $\boldsymbol{d} \in \mathcal{P}^1(T)^2$, such that:

$$\boldsymbol{d}(\boldsymbol{X}_k^0) = \chi(\boldsymbol{X}_k^0) - \boldsymbol{X}_k^0 \quad \forall k \in \{1, 2, 3\}.$$

In the plane of T, introduce a local system of orthogonal coordinates (s_1, s_2). With respect to the local system, let (d_1, d_2) be the components of the displacement. The components of the Green-Lagrange tensor read:

$$\{E\}_{ij} = (\delta_{ij} + \frac{\partial d_i}{\partial s_j})(\delta_{ji} + \frac{\partial d_j}{\partial s_i}) \quad i, j \in \{1, 2\},$$

where δ_{ij} represent the components of the identity matrix.

Since $\boldsymbol{d} \in \mathcal{P}^1(T, 2)$, it can be expanded with respect to the shape functions as:

$$\boldsymbol{d}(s_1, s_2) = \sum_{k=1}^{3} \gamma_k(s_1, s_2)\boldsymbol{d}(\boldsymbol{X}_k^0),$$

To expand the components of $\bar{\bar{E}}$ with respect to the shape functions, introduce the arrays $\{\eta\}$ and $\{\boldsymbol{D}\}$ with components:

$$\{\eta\}_{k,i} = \frac{\partial \gamma_k}{\partial s_i}, \quad \{\boldsymbol{D}\}_k = \boldsymbol{d}(\boldsymbol{X}_k^0), \qquad k \in \{1, 2, 3\},\ i \in \{1, 2\}.$$

The components of the Green-Lagrange tensor $\bar{\bar{E}}$ read:

$$\{E\} = (\{I\} + \{\eta\}\{\boldsymbol{D}\})^T(\{I\} + \{\eta\}\{\boldsymbol{D}\}).$$

The components of $\{\eta\}$ are independent of the specific T and of the transformation χ. as they are calculated with respect to the reference element. Therefore to evaluation of the Green-Lagrange tensor is determined by the actual configuration of the vertices $\boldsymbol{X}$.

Hyperelastic force acts at nodes with reference to a strain energy functional assigned by an hyperelastic model. Introducing Skalak's law (reference to 2.2.1), energy density stored in an element of surface is expressed as a function of the principal strains $(\lambda_1, \lambda_2$ as:

$$W_{SK} = \frac{G_{SK}}{4}[(\lambda_1^2 + \lambda_2^2 - 2)^2 + 2(\lambda_1^2 + \lambda_2^2 - \lambda_1\lambda_2 - 1) + C(\lambda_1^2\lambda_2^2 - 1)^2],$$

where G_S is the shear modulus, $G = \frac{G_A}{G_S}$ with G_A the area dilatation modulus, and:

$$\lambda_1^2 = \frac{1}{2}\left[E_{11} + E_{22} + \sqrt{(E_{11} - E_{22})^2 - 4E_{12}^2}\right]$$
$$\lambda_2^2 = \frac{1}{2}\left[E_{11} + E_{22} - \sqrt{(E_{11} - E_{22})^2 - 4E_{12}^2}\right].$$

The total hyperelastic energy of the element is given by:

$$\mathcal{E}_T = W_{SK}|T|,$$

where $|T|$ is the surface of the element.

The position of each vertex represents a degree of freedom for the element T. It is assumed that the element is subject to a system of forces $\{\boldsymbol{F}_T^1, \boldsymbol{F}_T^2, \boldsymbol{F}_T^3\}$ applied at the vertices. By the Pinciple of Virtual Works, the forces are determined by stating that energy variation must match the total work acted by the forces for each infitesimal movement of the vertices compatible with the constraints, or virtual displacement.

A set of virtual displacements $\{\delta\boldsymbol{D}_1\, \delta\boldsymbol{D}_2, \delta\boldsymbol{D}_3\}$ is associated to a variation of energy:

$$\delta\mathcal{E} = |T| \sum_{V\in\mathcal{T}_V} \boldsymbol{F}_T^V \cdot \delta\boldsymbol{D}_k.$$

On the other hand, exploiting the representation of $\bar{\bar{E}}$ in terms of shape functions, the energy variation can be expressed, by applying a chain rule, as:

$$\delta\mathcal{E} = |T|\delta W_{SK} = |T| \sum_{l=1}^{2}\sum_{m=1}^{2}\sum_{k=1}^{3} \frac{\partial W_{sk}}{\partial\lambda_l}\frac{\partial\lambda_l}{\partial\{E\}_{mn}}\frac{\partial\{E\}_{mn}}{\partial\boldsymbol{D}_k} \cdot \delta\boldsymbol{D}_k.$$

Since the balance must be respected for an arbitrary choice of $\{\delta\boldsymbol{D}_k\}$, the Principle is equivalent to the set of conditions:

$$\boldsymbol{F}_T^V = |T| \sum_{l=1}^{2}\sum_{m=1}^{2} \frac{\partial W_{sk}}{\partial\lambda_l}\frac{\partial\lambda_l}{\partial e_{mn}}\frac{\partial e_{mn}}{\partial\boldsymbol{D}_k} \qquad k \in \{1,2,3\},$$

which provides the expression of the forces with respect to the displacement of the vertices.

Finally, the `SURFACE_DATA` $\underline{\boldsymbol{F}}_{EL} = \{\boldsymbol{F}_{EL}(V) : V \in \mathcal{V}\}$ is given by summing the contributions from all the elements. Consider $V \in \mathcal{V}$ and let $\mathcal{T}_V$ be the set of faces sharing the vertex V. The total force acting on the vertex reads:

$$\boldsymbol{F}_{EL} = \sum_{T\in\mathcal{T}_V} \boldsymbol{F}_V^T.$$

Curvature Force

The second component of the membrane force is linked to curvature. Curvature force $\underline{\boldsymbol{F}}_{BND}$ is calculated starting from the mean curvature $\boldsymbol{H}$ and the Gaussian curvature $\boldsymbol{K}$ as:

$$\boldsymbol{F}_{BND,V} = k_b\left[(2H - C_0)(2H^2 - 2K + C_0H) + \Delta_{LB,h}H\right] \cdot \boldsymbol{N}_{\Sigma,V}|\tilde{\boldsymbol{N}}_{\Sigma,h}|,$$

where $\boldsymbol{C_0}$ is a `SCALAR SURFACE_DATA` located at nodes representing spontaneous curvature, $\underline{\boldsymbol{N}}_\Sigma$ is the normal to the surface, located also at nodes, $\underline{\tilde{\boldsymbol{N}}}_{\Sigma,h}$ is a first-approximation normal whose magnitude corresponds to the area of the patch $\mathcal{T}_V$ and $\Delta_{LB,h}$ is a discrete Laplace-Beltrami operator.

The calculation of mean and gaussian curvature, as well as the Laplace-Beltrami operator is based on a local representation of the surface in the neighbouring zone of vertex V as a quadratic surface. The original idea for the algorithm is presented in [Farutin, Biben, and Misbah 2014], and another presentation in the specific context of `YALES2BIO` can be found in [Taraconat 2020] .

Consider a vertex $V \in \mathcal{V}$ and the set of its neighbour vertices $\mathcal{V}_V$, as well as the set $\mathcal{T}_V$ of surface elements sharing V, both with cardinality N_V. For a triangle $T \in \mathcal{T}_V$, consider the set of vertices positions $\{\boldsymbol{X}_V, \boldsymbol{X}_T^-, \boldsymbol{X}_T^+\}$ ordered such that $(\boldsymbol{X}_T^- - \boldsymbol{X}_V) \times (\boldsymbol{X}_T^+ - \boldsymbol{X}_V)$ points out of the surface. The first-approximation normal to the surface is found as the average of the normal of triangles in $\mathcal{T}_V$:

$$\tilde{\boldsymbol{N}}_{\Sigma,V} = \sum_{T\in\mathcal{T}_V} (\boldsymbol{X}_T^- - \boldsymbol{X}_V) \times (\boldsymbol{X}_T^+ - \boldsymbol{X}_V).$$

Notice that the normal is not normalised, so that its magnitude corresponds to the area of the patch $\mathcal{T}_V$. This allows get the force $\boldsymbol{F}_{BND}$ with the correct scaling of a force instead of a force per unit area.

The surface patch is approximated as a quadratic surface. Introduce a local orthognal coordinate system $(\hat{\eta}, \hat{\xi}, \hat{z})$, centered at $\boldsymbol{X}_V$, such that the $\hat{z}$ axis is oriented as the normal $\boldsymbol{N}_{\Sigma,h}$. For a vertex $W \in \mathcal{V}_V$, let (η_W, ξ_W, z_W) be the componenents of $(\boldsymbol{X}_W - \boldsymbol{X}_V)$ int the local reference system.

In the local reference system the surface equation reads:

$$z_{SQ}(\eta, \xi) = z^0 + \partial_\eta z \eta + \partial_\xi z \xi + \frac{1}{2}(\partial_{\eta,\eta} z \eta^2 + \partial_{\eta,\xi} z \eta\xi + \partial_{\xi,\xi} z \xi^2).$$

The coefficients of the surface are determined such to minimize the square error function:

$$\chi = \sum_{W \in \mathcal{V}_V} (z_W - z_{SQ}(\eta_W, \xi_W)).$$

Once the coefficients of the surface are known, the geometric quantities necessary to define $\boldsymbol{F}_{BND}$ can be calculated.

The tangent vectors are given by:

$$\boldsymbol{\eta} = \hat{\eta} + \partial_\eta z \hat{z}, \qquad \boldsymbol{\xi} = \hat{\xi} + \partial_\xi z \hat{z}.$$

The normal vector, the metric tensor, and the second fundamental form are given by:

$$\boldsymbol{N}_{\Sigma,V} = \frac{\boldsymbol{\eta} \times \boldsymbol{\xi}}{\boldsymbol{\eta} \times \boldsymbol{\xi}}, \qquad G_{\alpha,\beta} = \boldsymbol{\alpha} \cdot \boldsymbol{\beta}, \qquad C_{\alpha,\beta} = \frac{1}{\sqrt{1 + \partial_\eta z + \partial_\xi z}} \partial_{\alpha,\beta} z, \qquad \alpha, \beta \in \{\eta, \xi\}.$$

Then, the mean and Gaussian curvature read:

$$H = \frac{1}{2}\mathrm{Tr}(G^{-1}C), \qquad K = \det(G^{-1}C).$$

Finally, the Laplace-Beltrami operator of the mean curvature is calculated as:

$$\Delta_{LB,h}(H) = \sum_{\alpha \in \{\eta,\xi\}} \sum_{\beta \in \{\eta,\xi\}} \frac{1}{\sqrt{|\det G|}} \partial_\alpha \left(\sqrt{|\det G|} G^{-1}_{\alpha,\beta} \partial_\beta H \right),$$

where the derivatives up to order 2 of H are estimated with a least square minimization, in the same fashion of what is done for the coefficients of the surface, by minimizing:

$$\chi_H = \sum_{W \in \mathcal{V}_V} (H_W - H_{SQ}(\eta_W, \xi_W)).$$

3.2 Flow Discretization

The membrane develops its trajectory in a domain $\Omega \subset \mathbb{R}^d$ which is interely occupied by a fluid. The internal phase of the cell is composed by a solutions of proteins, and the external region is occupied by a fluid suspension, which is typically a water solution. Therefore, when considering the transition of a single cell inside the domain, it is assumed that the flow is described by a couple of fields velocity $\boldsymbol{u} : \Omega \times [0, \infty) \to \mathbb{R}^d$ and pressure $p : \Omega \times [0, \infty) \to \mathbb{R}$. The flow is characterized by a density ρ and a dynamic viscosity μ, whose ratio $\nu = \mu/\rho$ defines the kinematic viscosity. Consider also a distribution of force per unite volume $\boldsymbol{f} : \Omega \to \mathbb{R}^d$. The couple $(\boldsymbol{u}, p)$ satisfies the system of Navier-Stokes equations in the entire domain:

$$\begin{cases} \rho\big(\dfrac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u})\big) = -\nabla p + \nabla \cdot [\mu(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u})] + \boldsymbol{f} & \text{in } \Omega \times [0, \infty) \quad (3.1a) \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega \times [0, \infty). \quad (3.1b) \end{cases}$$

Let $\text{int}(\Sigma)$ be the region of space enclosed by the membrane Σ. We consider a scenario where the cell does not leave the domain, nor touches its boundary, namely:

$$\text{int}(\Sigma) \subset \Omega \qquad t \in [0, \infty).$$

As a consequence, it is possible to partition the domain with the subdomains $\Omega_{\text{int}} = \text{int}(\Sigma)$ representing the interior of the cell and $\Omega_{\text{ext}} = \Omega \setminus \overline{\Omega}_{\text{int}}$, such that:

$$\Omega = \overline{\Omega_{\text{int}}} \cup \overline{\Omega_{\text{ext}}}, \qquad \Sigma = \overline{\Omega_{\text{int}}} \cap \overline{\Omega_{\text{ext}}}.$$

The data $\mu : \Omega \to \mathbb{R}$ and $\rho : \Omega \to \mathbb{R}$ are typically assumed as constants in each subdomain:

$$\mu = (\mu_{\text{int}} - \mu_{\text{ext}})\mathbb{I}_{\Omega_{\text{int}}} + \mu_{\text{ext}}$$
$$\rho = (\rho_{\text{int}} - \rho_{\text{ext}})\mathbb{I}_{\Omega_{\text{int}}} + \rho_{\text{ext}}.$$

The volume force distribution $\boldsymbol{f}$ can take into account gravity or apparent forces, but the system is assumed to be inertial and the cell buoyancy neutral. The only component of $\boldsymbol{f}$ is related to the interaction of the flow with the membrane. The representation of the cell action in terms of a force per unit volume acting on the bulk is the keypoint of the Immersed Boundary Method. The distribution of membrane tractions $\boldsymbol{F}_{\boldsymbol{\Sigma}}(t) : \Sigma \to \mathbb{R}^d$ is determined by the configuration $\Sigma(t)$ and by the presence of of an electric field. Membrane tractions are taken into account by introducing a distribution of volume force $\boldsymbol{f}_\Sigma : \Omega \to \mathbb{R}^d$ as detailed in Section 2.3.

3.2.1 Flow Time Advancement

Equations 3.1 are discretized in time with a fractional step scheme based on the Projection method originally introduced in [Chorin 1997] and widely diffused in Computational Fluid Dynamics. The method allows a fast implementation of the time-advancement of Navier-Stokes equations, where the advancement of velocity and pressure happens at different steps of the algorithm.

The Projection is implemented by splitting the time advancement into three steps:

- Velocity Prediction: explicit, implicit or semi-implicit;
- Pressure Advancement: solution of an elliptic problem;
- Velocity Correction: enforcement of divergence-free constraint.

Consider a time step τ. For $n \in \mathbb{N}$ introduce $\boldsymbol{u}^n : \Omega \to \mathbb{R}^d$, $p^n : \Omega \to \mathbb{R}$ and $\boldsymbol{f}^n : \Omega \to \mathbb{R}^d$ such that $\boldsymbol{u}^n(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}, n\tau)$, $p^n(\boldsymbol{x}) = p(\boldsymbol{x}, n\tau)$ and $\boldsymbol{f}^n(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}, n\tau)$. Consider an advancement on the interval $[n\tau, (n+1)\tau]$. With an explicit version of the Projection Method, Equation (3.1a) is discretized as:

$$\frac{1}{\tau}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^n) + \nabla \cdot (\boldsymbol{u}^n \otimes \boldsymbol{u}^n) = -\frac{1}{\rho}\nabla p^n + \nabla \cdot [\nu(\nabla \boldsymbol{u}^n + \nabla^T \boldsymbol{u}^n)] + \frac{\boldsymbol{f}^n}{\rho}. \quad (3.2)$$

Introduce a predictional velocity $\boldsymbol{u}^* : \Omega \to \mathbb{R}^d$, such that:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\tau} = -\nabla \cdot (\boldsymbol{u}^n \otimes \boldsymbol{u}^n) + \nabla \cdot [\nu(\nabla \boldsymbol{u}^n + \nabla^T \boldsymbol{u}^n)] + \frac{\boldsymbol{f}^n}{\rho}. \tag{3.3}$$

Subtracting Equations (3.2) and (3.3), results in:

$$\frac{1}{\tau}(\boldsymbol{u}^{n+1} - \boldsymbol{u}^*) = -\frac{1}{\rho}\nabla p^{n+1}. \tag{3.4}$$

Taking the divergence of both sides, and enforcing $\nabla \cdot (\boldsymbol{u}^{n+1}) = 0$, results in:

$$\frac{\nabla \cdot \boldsymbol{u}^*}{\tau} = \frac{1}{\rho}\Delta p^{n+1},$$

A time advancement algorithm can be then put in place in three steps. First the predictional velocity is calculated; then a Poisson equation is solved to get the advanced pressure, and eventually the predictional velocity is corrected to obtain the velocity at the next time-step:

$$1) \quad \boldsymbol{u}^* = \boldsymbol{u}^n + \tau\{-\nabla \cdot (\boldsymbol{u}^n \otimes \boldsymbol{u}^n) + \nabla \cdot [\nu(\nabla \boldsymbol{u}^n + \nabla^T \boldsymbol{u}^n)] + \boldsymbol{f}^n\} \tag{3.5a}$$

$$2) \quad \Delta p^{n+1} = \frac{\rho}{\tau}(\nabla \cdot \boldsymbol{u}^*) \tag{3.5b}$$

$$3) \quad \boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \frac{\tau}{\rho}\nabla p^{n+1}. \tag{3.5c}$$

The original formulation of step (3.5a) produces a method which is order 1 accurate in time. In order to improve accuracy, a popular alternative is a multistep advancement. In `YALES2BIO`, a 4-th order Runge-Kutta `RK4` time advancement scheme is implemented.

Rewriting more compactly step (3.5a) as:

$$\boldsymbol{u}^* = \mathcal{F}(\boldsymbol{u}^n, n\tau),$$

the time-advancement scheme can be recast as:

$$\boldsymbol{\alpha}^1 = \mathcal{F}(\boldsymbol{u}^n, n\tau) \tag{3.6}$$

$$\boldsymbol{\alpha}^2 = \mathcal{F}(\boldsymbol{u}^n + \frac{\tau}{2}\boldsymbol{\alpha}^1, (n + \frac{1}{2})\tau) \tag{3.7}$$

$$\boldsymbol{\alpha}^3 = \mathcal{F}(\boldsymbol{u}^n + \frac{\tau}{2}\boldsymbol{\alpha}^2, (n + \frac{1}{2})\tau) \tag{3.8}$$

$$\boldsymbol{\alpha}^4 = \mathcal{F}(\boldsymbol{u}^n + \tau\boldsymbol{\alpha}^3, (n + 1)\tau) \tag{3.9}$$

$$\boldsymbol{u}^* = \boldsymbol{u}^n + \frac{\tau}{6}(\boldsymbol{\alpha}^1 + 2\boldsymbol{\alpha}^2 + 2\boldsymbol{\alpha}^2 + \boldsymbol{\alpha}^4). \tag{3.10}$$

3.2.2 Space Discretization

In `YALES2BIO`, the discretization in space of Equations 3.1 is realized by a Finite Volume Method. If the geometry of the boundary of Ω is suitable, a Cartesian grid can be employed to introduce a subdivision in Finite Volumes. More in general, unstructured tetrahedral (triangular when $d = 2$) meshes can be used as a base to construct a dual grid of Finite Volumes (see Figure 3.3). For the sake of simplicity, this exposition is limited to the case with $d = 2$ and an unstructured grid. For a detailed introduction on the case $d = 3$, we refer the reader to [Vantieghem 2011].

Let $\Omega_h = \{\mathcal{T}, \mathcal{V}\}$ be an unstructured tetrahedral mesh of the domain, where $\mathcal{T}$ is the collection of tetrahedral elements and $\mathcal{V}$ the set of vertices. Consider a vertex V and the set $\mathcal{T}_V$ of elements sharing

V. A finite volume Ω_V centered at V is constructed by connecting the barycenters of the elements in $\mathcal{T}_V$ to the midpoints of the edges VW with $W \in \mathcal{V}_V$. To each edge VW, the associated portion S_W of the boundary is the broken segment connecting the midpoint of VW with the barycenters $\boldsymbol{x}_T^-$, $\boldsymbol{x}_T^+$ of the neighbouring elements.

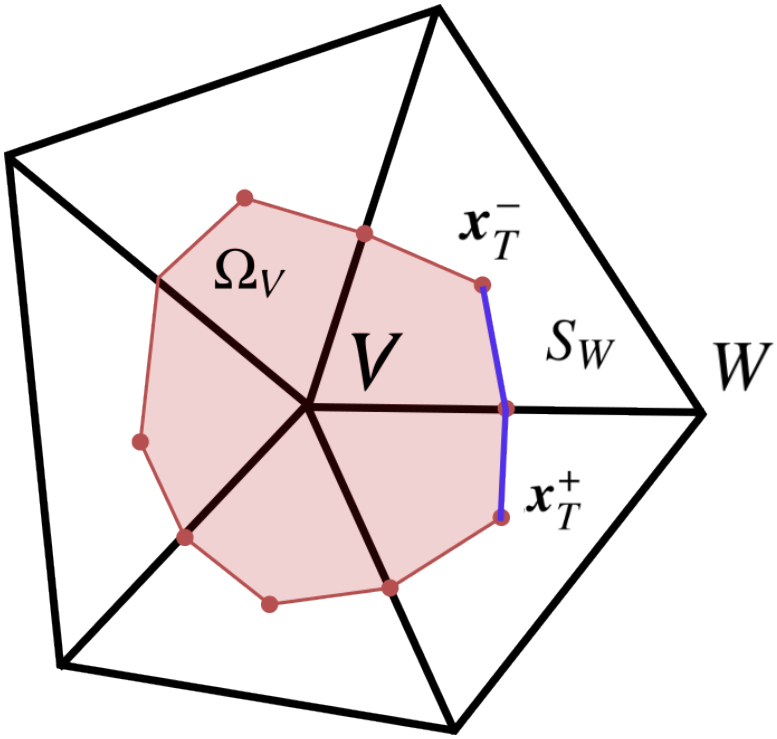


Figure 3.3: Unstructured grid: finite volume constructed as a dual cell around a central node $V \in \mathcal{V}$, with detail of the face associated to edge VW.

Prediction Step

Consider the Prediction Step (3.5a), and a vertex V. An integration by parts over Ω_V provides:

$$\frac{1}{\tau}\int_{\Omega_V}(\boldsymbol{u}^* - \boldsymbol{u}^n) = -\sum_{W\in\mathcal{V}_V}\int_{S_W}(\boldsymbol{u}^n \otimes \boldsymbol{u}^n)\cdot\hat{\boldsymbol{n}} + \sum_{W\in\mathcal{V}}\int_{S_W}\nu\nabla\boldsymbol{u}\cdot\hat{\boldsymbol{n}}, + \sum_{W\in\mathcal{V}}\int_{S_W}\nu\nabla^T\boldsymbol{u}\cdot\hat{\boldsymbol{n}}, + \int_{\Omega_V}\frac{\boldsymbol{f}^n}{\rho} \tag{3.11}$$

where $\hat{\boldsymbol{n}}$ is the normal vector at the boundary $\partial\Omega_V$ pointing outwards.

Integrals are discretized introducing cell-centered data stocked on vertices. $\boldsymbol{u}^n : \Omega \to \mathbb{R}^d$ is replaced by a `GRID_DATA` $\underline{\boldsymbol{u}}^n = \{\boldsymbol{u}_V^n : V \in \mathcal{V}\}$ and $p^n : \Omega \to \mathbb{R}$ by $\underline{p}^n = \{p_V^n : V \in \mathcal{V}\}$. Equivalently, cell-centered viscosity $\underline{\nu}$, density $\underline{\rho}$ and force distribution $\underline{\boldsymbol{f}}$ are introduced. In `YALES2BIO`, variable viscosity is taken into account by implementing a version of the Front-Tracking algorithm originally presented in [Unverdi and Tryggvason 1992].

Volume integrals are approximated as:

$$\int_{\Omega_V}\boldsymbol{w} \longmapsto |\Omega_V|\boldsymbol{w}_V,$$

where $|\Omega_V|$ is the measure of the finite volume. Consider the oriented surface:

$$\boldsymbol{S}_W = \frac{1}{2}(|S_W^-|\boldsymbol{n}^- + |S_W^+|\boldsymbol{n}^+),$$

where S_W^- and S_W^+ are the two straight edges composing S_W and $\boldsymbol{n}^-$ and $\boldsymbol{n}^+$ the respective normals (see Figure 3.4). The flux of a field ϕ across S_W is approximated as:

$$\int_{S_W} \phi \hat{\boldsymbol{n}}_V = \phi_{VW} \boldsymbol{S}_W.$$

where $\phi_{VW} = \phi(\frac{\boldsymbol{x}_V + \boldsymbol{x}_W}{2}) = \phi(\boldsymbol{x}_{VW})$. The midpoint value ϕ_{VW} and the gradient $\nabla\phi|_{VW}$ are approximated via a finite difference scheme. Introducing the Taylor expansions of ϕ around $\boldsymbol{x}_{VW}$:

$$\phi_V = \phi_{VW} + \nabla\phi|_{VW} \cdot (\boldsymbol{x}_V - \boldsymbol{x}_{VW}) + O(\|\boldsymbol{x}_V - \boldsymbol{x}_{VW}\|^2) \tag{3.12}$$

$$\phi_W = \phi_{VW} + \nabla\phi|_{VW} \cdot (\boldsymbol{x}_W - \boldsymbol{x}_{VW}) + O(\|\boldsymbol{x}_V - \boldsymbol{x}_{VW}\|^2), \tag{3.13}$$

the approximation:

$$\phi_{VW} \mapsto \frac{\phi_V + \phi_W}{2}$$

is introduced. Adding term by term in 3.12 and neglecting small terms provides:

$$\phi_W - \phi_V = \nabla\phi|_{VW} \cdot (\boldsymbol{x}_W - \boldsymbol{x}_V),$$

resulting in the approximation of the midpoint gradient:

$$\nabla\phi|_{VW} \mapsto (\phi_W - \phi_V)\frac{\boldsymbol{x}_W - \boldsymbol{x}_V}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|^2}.$$

Please remark that this approximation of the gradient is not consistent unless the edge VW is normal to the boundary portion S_W. It is case if the triangles are all equilateral. The numerical scheme presented in Section 3.4.1 is based on this approximation scheme. However, high order discretizations of the gradient are available in YALES2 [Vantieghem 2011].
Adopting this set of approximations, the surface integral operators:

$$C_V(\boldsymbol{w}, \boldsymbol{w}) \coloneqq \sum_{W \in \mathcal{V}_V} \int_{S_W} (\boldsymbol{w}^n \otimes \boldsymbol{w}^n) \cdot \hat{\boldsymbol{n}}$$

$$\mathcal{F}_V(\boldsymbol{w}, \nu) \coloneqq \sum_{W \in \mathcal{V}} \int_{S_W} \nu(\nabla\boldsymbol{w} + \nabla^T\boldsymbol{w}) \cdot \hat{\boldsymbol{n}},$$

are replaced by their discrete counterparts:

$$C_V^h(\underline{\boldsymbol{w}}, \underline{\boldsymbol{w}}) \coloneqq \sum_{W \in \mathcal{V}_V} (\boldsymbol{w}_{VW} \otimes \boldsymbol{w}_{VW}) \cdot \boldsymbol{S}_W$$

$$\mathcal{F}_V^h(\underline{\boldsymbol{w}}, \underline{\nu}) \coloneqq \sum_{W \in \mathcal{V}_V} \nu_{VW}(\nabla\boldsymbol{w}|_{VW} + \nabla^T\boldsymbol{w}|_{VW}) \cdot \boldsymbol{S}_W.$$

The prediction step is then discretized as:

$$\boldsymbol{u}_V^* = \boldsymbol{u}_V^n + \tau\left[\frac{1}{|\Omega_V|}\big(-C_V^h(\underline{\boldsymbol{u}}^n, \underline{\boldsymbol{u}}^n) + \mathcal{F}_V^h(\underline{\boldsymbol{v}}^n, \underline{\nu}^n)\big) + \frac{\boldsymbol{f}_V^n}{\rho_V}\right] \qquad \forall V \in \mathcal{V}.$$

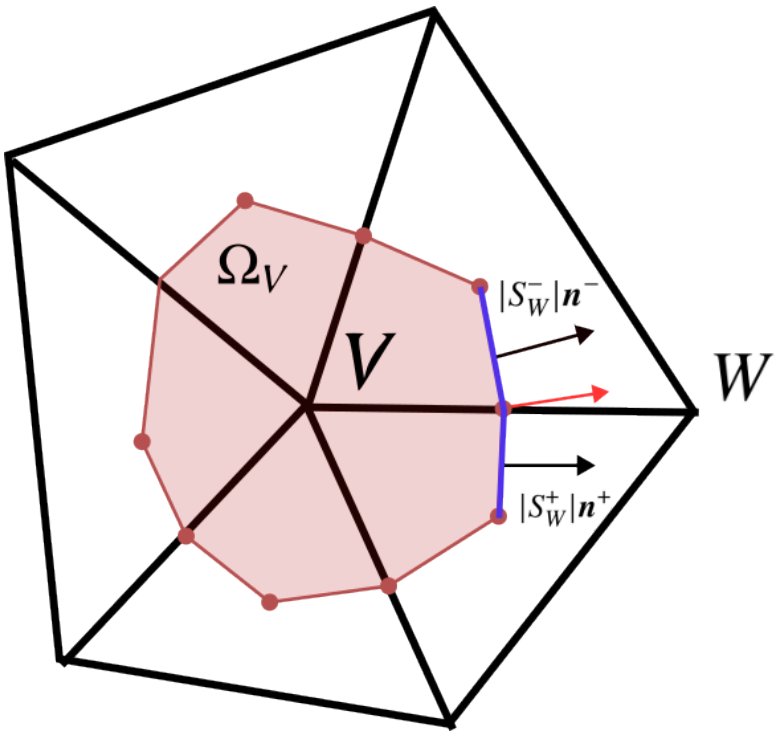


Figure 3.4: Approximation of the oriented surface $\boldsymbol{S}_W$.

Pressure Advancement

The Pressure Advancement step (3.5b), integrated over Ω_V reads:

$$\sum_{W\in\mathcal{V}_V}\int_{S_W}\nabla p^{n+1}\cdot\hat{\boldsymbol{n}}=\frac{\rho}{\tau}\sum_{W\in\mathcal{V}_V}\int_{S_W}\boldsymbol{u}^*\cdot\hat{\boldsymbol{n}}. \tag{3.14}$$

Introducing the discretizations:

$$\mathcal{D}_V(\boldsymbol{w}):=\sum_{W\in\mathcal{V}_V}\int_{S_W}\boldsymbol{w}\cdot\hat{\boldsymbol{n}}\longmapsto\mathcal{D}_V^h(\underline{\boldsymbol{w}}):=\sum_{W\in\mathcal{V}_V}\boldsymbol{w}_{VW}\cdot\boldsymbol{S}_W$$

and:

$$\mathcal{L}_V(q):=\sum_{W\in\mathcal{V}_V}\int_{S_W}\nabla q\cdot\hat{\boldsymbol{n}}\longmapsto\mathcal{L}_V^h(\underline{q}):=\sum_{W\in\mathcal{V}_V}\int_{S_W}\nabla q|_{VW}\cdot\boldsymbol{S}_W,$$

the advanced pressure $\underline{p}^{n+1}$ is found as the solution of the linear system:

$$\mathcal{L}_V^h(\underline{p}^{n+1})=\frac{\rho}{\tau}\mathcal{D}_V^h(\underline{\boldsymbol{u}}^*)\qquad\forall V\in\mathcal{V}.$$

Velocity Correction

The integrated Correction step reads:

$$\frac{1}{\tau}\int_{\Omega_V}(\boldsymbol{u}^{n+1}-\boldsymbol{u}^*)=-\frac{1}{\rho}\sum_{W\in\mathcal{V}_V}\int_{S_W}p^{n+1}\hat{\boldsymbol{n}}.$$

Introducing the approximation of the surface integral:

$$\mathcal{G}_V(q) \coloneqq \sum_{W\in\mathcal{V}_V}\int_{S_W} q\hat{\boldsymbol{n}} \longmapsto \mathcal{G}_V^h(\underline{q}) \coloneqq \sum_{W\in\mathcal{V}_V} q_{VW}\boldsymbol{S}_W,$$

the discretization of velocity correction reads:

$$\boldsymbol{u}_V^{n+1} = \boldsymbol{u}_V^* - \frac{\tau}{\rho|\Omega_V|}\mathcal{G}_V^h(\underline{p}^{n+1}) \qquad \forall\, V \in \mathcal{V}.$$

3.2.3 Implementation of Discrete Operators

All the discrete operators that have been introduced can be represented as linear mappings of `GRID_DATA` into `GRID_DATA`. In particular, the discrete Laplace operator is a linear transformation:

$$\mathcal{L} : \mathbb{R}^N \to \mathbb{R}^N,$$

where $N = \mathrm{card}(\mathcal{V})$. Therefore, in `YALES2BIO` it is represented as a matrix $\{L\}$, such that:

$$\mathcal{V}(\underline{p})_l = \sum_{k=1}^{N}\{L\}_{l,k}p_k, \qquad l \le N.$$

The matrix $\{L\}$ is sparse. The line corresponding to vertex V has non-zero entries only at the location of the vertices of the stencil. Therefore, the representation in memory of the operator can be optimized by exploting the link between the sparsity of the matrix and the structure of the mesh.

In `YALES2BIO`, many linear systems are solved numerically by iterative solvers, like the Conjugate Gradient algorithm (CG). It is worth to cite that in terms of preconditioning strategies, YALES2BIO offers an efficient implementation of standard multi-grid techniques such as the deflation-based preconditioning [Nicolaides 1987]. These techniques are by default applied to solve numerically the Laplace equation for pressure in Navier-Stokes equations. In the implementation of these algorithms, the only operations involving the matrix are matrix-vector products. To implement the product a matrix $\{A\} \in \mathbb{R}^{N,N}$ can be represented in a more efficient way than an $N \times N$ array.

Let $\mathcal{E}$ denote the set of edges of the `GRID` Ω_h, intended as collections of pairs $(V, W) \in \mathcal{V} \times \mathcal{V}$. Lat $M = \mathrm{card}(\mathcal{E})$, and define $\mathcal{E}_V$ as the set of edges connected to the vertex V. The product of the matrix $\{A\}$ with a vector $\{b\} \in \mathbb{R}^N$ can be expressed as:

$$\{A\}\{b\}_l = d_l b_l + \sum_{E\in\mathcal{E}_l} s_E(b_{k(E,V)} - b_l) + t_E(b_{k(E,V)} + b_l), \tag{3.15}$$

where the arrays $\{d\} \in \mathbb{R}^N$ and $\{s\}, \{t\} \in \mathbb{R}^M$ have values associated respectively to nodes and edges of the mesh and $k(E, V)$ is the map associating to the edge E the vertex opposite to V.

It can be shown that the representation (3.15) is always possible, and in particular that the coefficients s and t correspond to the coefficents of the symmetric and skew-symmetric part of $\{A\}$. The product can be implemented by a loop over the vertices followed by a loop over the edges as in Algorithm 1:

3.3 Fluid Structure Interaction

This Section describes the numerical implementation of the Flow-Structure Interaction through the Immersed Boundary Method described in Section 2.3. The coupling is realized by defining a diffusion

Algorithm 1 Matrix-Vector Product

INPUT: Matrix Coefficients (d, s, t), array b, connectivity of Ω_h
$p \leftarrow 0$
for l in Nodes **do**
  $p_l \leftarrow p_l + d_l * b_l$
end for
for k in Edges **do**
  $i, j \leftarrow$ vertices of edge k
  $S_E \leftarrow s_E * (b_j - b_i)$
  $T_E \leftarrow s_E * (b_j + b_i)$
  $p_i \leftarrow p_i + (S_E + T_E)$
  $p_j \leftarrow p_j - (S_E + T_E)$
end for
OUTPUT: p

algorithm to transmit membrane tractions to the flow in the form of a distribution of volume force in the domain, and an interpolation algorithm (`IBM-INT`) to transmit the velocity of the flow to the membrane.

Let $\Sigma_h = \{\mathcal{T}_\Sigma, \mathcal{V}_\Sigma\}$ be the `SURFACE` representing the membrane, and $\Omega_h = \{\mathcal{T}_\Omega, \mathcal{V}_\Omega\}$ the `GRID` of the domain. As detailed in section 3.1, membrane force is represented in `YALES2BIO` by the `SURFACE_DATA` $\underline{\boldsymbol{F}} = \{\boldsymbol{F}(V) : V \in \mathcal{V}_\Sigma\}$. Within the Immersed Boundary Method, the action of the membrane is represented by a force per unit volume $\boldsymbol{f}_\Sigma : \Omega \rightarrow \mathbb{R}^d$, whose discrete counterpart is a `GRID_DATA` $\underline{\boldsymbol{f}}_\Sigma = \{\boldsymbol{f}_\Sigma(V) : V \in \mathcal{V}_\Omega\}$. The passage from $\underline{\boldsymbol{F}}$ to $\underline{\boldsymbol{f}}_\Sigma$ is realized with a diffusion operation. On the other hand, the velocity of the flow is encoded in a `GRID_DATA` $\underline{\boldsymbol{v}} = \{\boldsymbol{v}(V) : V \in \mathcal{V}_\Omega\}$ which is interpolated to obtain the surface velocity $\underline{\boldsymbol{V}} = \{\boldsymbol{V}(V) : V \in \mathcal{V}_\Sigma\}$.

To implement the diffusion and interpolation operators introduced in section 2.3, Dirac's delta δ is replaced by a discrete counterpart δ_h. The latter is proposed so to mimic the property of δ:

$$f(\boldsymbol{x}_0) = \int_\Omega \delta(\boldsymbol{x} - \boldsymbol{x}_0) f(\boldsymbol{x}) d\boldsymbol{x} \qquad \forall f : \Omega \rightarrow \mathbb{R},\ \forall \boldsymbol{x}_0 \in \Omega. \tag{3.16}$$

Considering the expansion of f around $\boldsymbol{x}_0$ up to order p:

$$f(\boldsymbol{x}) = \sum_{k=0}^{p} \sum_{l+m+n=k} \frac{\partial f}{\partial x^l \partial y^m \partial z^n}(x - x_0)^l (y - y_0)^m (z - z_0)^n,$$

the replacement of δ by δ_h in 3.16 leads to:

$$\begin{aligned} f(\boldsymbol{x}_0) &= \sum_{k=0}^{p} \sum_{l+m+n=k} \frac{\partial f}{\partial x^l \partial y^m \partial z^n}|_{\boldsymbol{x}_0} \int_\Omega \delta_h(\boldsymbol{x}_0 - \boldsymbol{x})(x - x_0)^l (y - y_0)^m (z - z_0)^n dxdydz \\ &= \sum_{k=0}^{p} \sum_{l+m+n=k} \frac{\partial f}{\partial x^l \partial y^m \partial z^n}|_{\boldsymbol{x}_0} M_{l,m,n} \qquad \forall f : \Omega \rightarrow \mathbb{R}, \end{aligned} \tag{3.17}$$

where the quantities $M_{l,m,n} = \int_\Omega \delta_h(\boldsymbol{x}_0 - \boldsymbol{x})(x - x_0)^l (y - y_0)^m (z - z_0)^n dxdydz$ are called momenta of δ_h. In order for 3.17 to be valid for any f, momenta must respect the set of condition:

$$\begin{cases} M_{0,0,0} = 1 \\ M_{l,m,n} = 0 \quad (l, m, n) : 1 \leq l + m + n \leq p. \end{cases} \tag{3.18}$$

The original formulation of the Immersed Boundary method on cartesian meshes proposed in [Peskin 1972] is based a radial window function respecting (3.18) defined as:

$$\delta_h(\boldsymbol{x}) = \begin{cases} \frac{1}{2}(1+\cos(\frac{\pi\|\boldsymbol{x}\|}{R})) & \|\boldsymbol{x}\| \leq R \\ 0 & \|\boldsymbol{x}\| > R, \end{cases} \tag{3.19}$$

where h is the cell size of the grid and $R > 0$ is the radius of the support of δ_h. δ_h respects the equation of momenta (3.18) up to order $k = 1$.

Formulation (3.19) must be replaced when unstructured grids are considered. In such a case, the Immersed Boundary Method can be adapted by introducing a window function $\delta_{h,V}$ calibrated for each $V \in \mathcal{V}$ according to the Reproducing Kernel Particle Method [Liu, Jun, and Zhang 1995]. The calibrated window function centered at node $\boldsymbol{x}_V = (x_V, y_V, z_V)$ is defined as:

$$\delta_{h,V}(\boldsymbol{x}-\boldsymbol{x}_V) = \delta_h(\boldsymbol{x}-\boldsymbol{x}_V)\sum_{k=0}^{p}\sum_{l+m+n=k} b_{l,m,n}(x-x_V)^l(y-y_V)^m(z-z_V)^n,$$

where $b_{l,m,n}$ are the calibration coefficients. The latter are determined by solving a linear system obtained imposing condition (3.17) as:

$$\begin{cases} \displaystyle\sum_{k=0}^{p}\sum_{l+m+n=k} b_{l,m,n}\int_\Omega (x-x_V)^l(y-y_V)^m(z-z_V)^n\delta_h(\|\boldsymbol{x}\|) = 1 \\ \displaystyle\sum_{k=0}^{p}\sum_{l+m+n=k} b_{l,m,n}\int_\Omega (x-x_V)^{l+i}(y-y_V)^{m+j}(z-z_V)^{n+k}\delta_h(\|\boldsymbol{x}\|) = 0 \quad \forall\,(i,j,k) : 1 \leq i+j+k \leq p. \end{cases}$$

Given a window function ($\delta_{h,V}$), diffusion of force (2.7) and interpolation of velocity 2.8 take the discrete form:

$$\begin{cases} \boldsymbol{V}_V = \displaystyle\sum_{W\in\mathcal{V}_\Omega} \delta_{h,V}(\boldsymbol{x}_W-\boldsymbol{x}_V)\boldsymbol{u}_W & \forall V \in \mathcal{V}_\Sigma \quad (3.21a) \\ \boldsymbol{f}_{\Sigma,V} = \displaystyle\sum_{W\in\mathcal{V}_\Sigma} \delta_{h,V}(\boldsymbol{x}_W-\boldsymbol{x}_V)\boldsymbol{F}_W & \forall V \in \mathcal{V}_\Omega. \quad (3.21b) \end{cases}$$

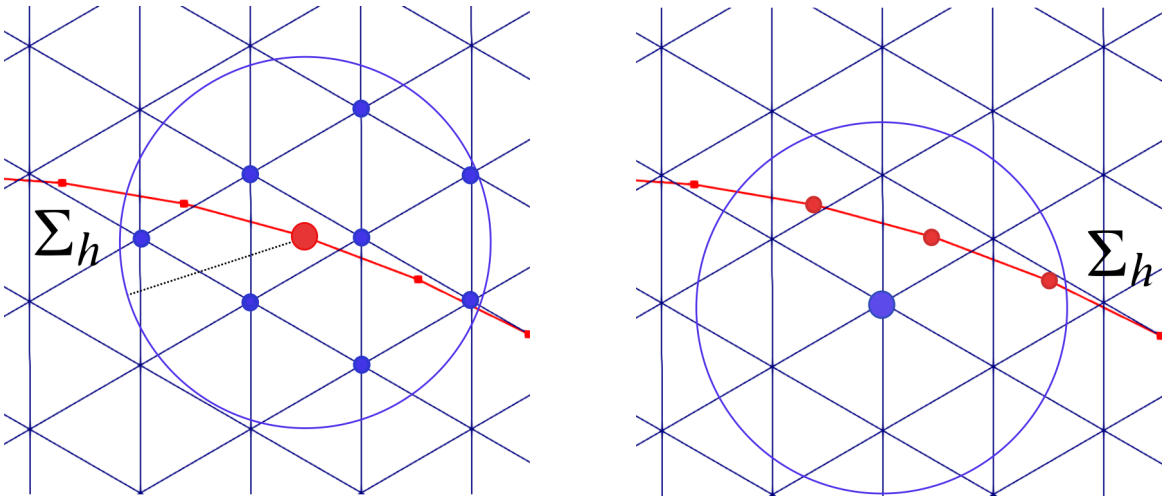


Figure 3.5: Scheme representing the numerical implementation of the Immersed Boundary method. The background mesh of the fluid domain is represented in blue. The surface mesh Σ_h is displayed in red. The boundary of the support of the window function δ_h is represented as a blue disk. On the left, the velocity on a vertex of Σ_h is determined as a weighted average of the values of the velocity on the nodes of the fluid domain falling in the support of δ_h (interpolation). On the right, a volume force density at node of the fluid domain is determined similarly, by averaging over nodes of Σ_h in proximity. The radius of the support is typically chosen as twice the mesh diameter h_Σ. To keep the simulation in the validity regime, an operational cryterion is having at most 1 surface vertex per element on the background mesh.

3.4 Electric Force Calculation

The numerical framework presented so far describes the infrastructure of the solver `YALES2BIO`, where the dynamics of the membrane results from a bidirectional coupling where the membrane tension $\boldsymbol{F} : \Sigma \to \mathbb{R}^d$ is transmitted to the flow and the flow velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ is transmitted to the membrane.

As described in Section 2.4, to take into account the dynamical effect of an external electric field, the Leaky Dielectric Model introduces a surface force $\boldsymbol{F}_{MW} : \Sigma \to \mathbb{R}^d$, such that the total tension applied to the membrane Σ reads:

$$\boldsymbol{F}_{TOT} = \boldsymbol{F} + \boldsymbol{F}_{MW}.$$

In this Section, we describe the module `ESS` of the solver (cf. diagram of Figure (3.10)), whose output is the `SURFACE_DATA` $\underline{\boldsymbol{F}}_{MW}$. In `YALES2BIO`, once $\underline{\boldsymbol{F}}_{MW}$ is calculated, it can be added to the `SURFACE_DATA` $\underline{\boldsymbol{F}}_{TOT}$ describing the total forces acting on the membrane, letting unchanged the rest of the solver.

The calculation of Maxwell tension over the membrane is slightly more involved with respect to the other components, related to hyperelastic and bending energy. The estimation, of the latter is based exclusively on the actual configuration of the membrane Σ. On the other hand, $\boldsymbol{F}_{MW}$ is related to a volume quantity, the Maxwell tensor $\bar{\bar{T}}_{MW} : \Omega \setminus \Sigma \to \mathbb{R}^{d\times d}$, such that:

$$\boldsymbol{F}_{MW} = [\bar{\bar{T}}_{MW}]_\Sigma.$$

Therefore, an estimation of $\bar{\bar{T}}_{MW}$, or at least of its jump across the interface has to be provided in order to estimate the electric tension. This requires to implement a technique to interpolate volume variables onto the surface such that jumps are captured. The IBM interpolation operator (2.8) is

not adapted for the task, because it does not take into account quantities with a jump across the interface.

Maxwell tensor is related to the scalar electric potential $\phi : \Omega \setminus \Sigma \to \mathbb{R}$ as:

$$\bar{\bar{T}}_{MW} = \epsilon\big(\nabla\phi \otimes \nabla\phi - \frac{1}{2}\|\nabla\phi\|^2 \bar{\bar{I}}\big),$$

where $\boldsymbol{I}$ is the Identity, and ϵ is the electric permittivity of the medium (usually constant across the membrane).

Finally, at any time, ϕ is related to the potential jump J_Σ by an elliptic equation with interface conditions:

$$\begin{cases} -\nabla \cdot (\sigma_\bullet \nabla\phi) = 0 & \text{in } \Omega_\bullet,\ \bullet \in \{\text{int}, \text{ext}\} \quad (3.22a) \\ [\phi] = J_\Sigma & \text{on } \Sigma \quad (3.22b) \\ [\sigma_\bullet \nabla\phi_\bullet] \cdot \boldsymbol{n}_\Sigma = 0 & \text{on } \Sigma \quad (3.22c) \end{cases}$$

depending on the phase dependent conductivity σ.

The potential jump $J_\Sigma : \Sigma_0 \times [0, \infty) \to \mathbb{R}$ acts as a state variable describing the charging state of the membrane. It varies in time according to the evolution equation:

$$C\partial_t J_\Sigma = \sigma_{\text{int}} \nabla\phi_{\text{int}} = \sigma_{\text{ext}} \nabla\phi_{\text{ext}}. \quad (3.23)$$

The `SURFACE_DATA` $\underline{J_\Sigma}$ represents a state variable describing the state of charge of the membrane. It is the input of the solver `ESS`, together with the membrane configuration $\underline{\boldsymbol{X}}$. The first step of the algorithm consists in the calculation of the discontinuous potential $\underline{\phi}$, by solving Equation (3.22). The following step consists in postprocessing the potential to get the electric field $\overline{\nabla\phi}$, and the Maxwell tensor $\boldsymbol{T}_{MW}$. The electric force $\boldsymbol{F}_{MW}$ is then calculated by interpolating Maxwell tensor from both sides. The potential jump is finally considering the interpolation of the normal component of the jump of electric field $\nabla\phi$ onto the membrane.

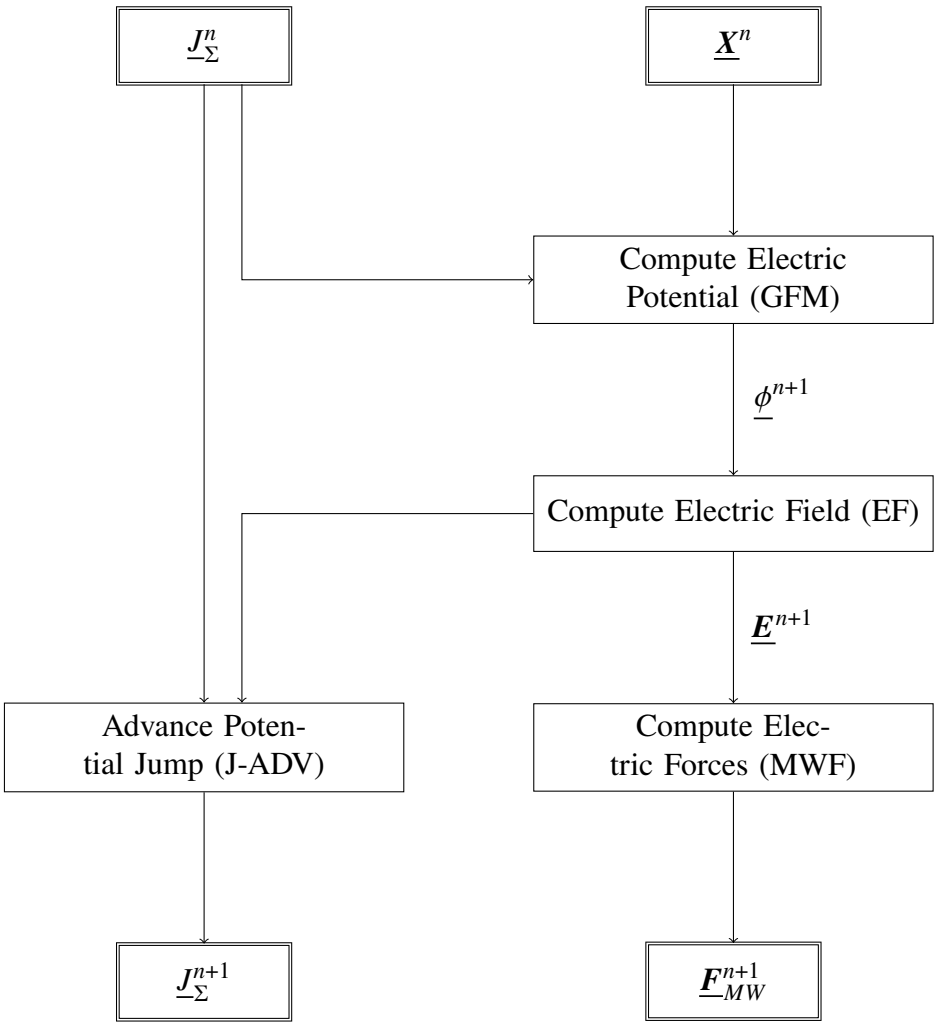


Figure 3.6: Flow diagram of the Electrostatic Solver

3.4.1 Electric Potential Calculation

A Finite Volume discretization is introduced to solve (3.22). Internal interface conditions arise in many physics and engineering problems, from geoscience, to multiphase flow dynamics to electromagnetism. Several approaches for their numerical treatment have emerged, both for Finite Difference/Finite Volume discretizations and Galerkin methods. A thourough review on the subject regarding Galerkin methods is provided at the beginning of Chapter 5.

As for the implementation in `YALES2BIO`, which is based on a Finite Volume scheme, the interface condition is implemented via the Ghost Fluid Method (GFM). The method, presented for the first time in [Fedkiw et al. 1999] , allows to represent variables that are discontinuous across an interface which is not fitted to the mesh. The original applications are related to the treatment of contact discontinuities for the inviscid Euler equations and the capture of shocks, deflagrations, detonations and compressible viscous flows. In this contexts it is conceived as an implicit imposition of Rankine-Hugoniot conditions enforced by introducing an artificial prolongation of the flow beyond the interface (ghost fluid).

The `RBC` solver already incorporates a treatment of internal boundaries, via the Immersed Boundary Method. However, IBM introduces numerical smearing [Liu, Fedkiw, and Kang 2000] and diffuses the jump at the interface. This makes IBM not adapted for estimating the jumps on which the electric forces depend. One of the main advantage of GFM is the sharp representation of discontinuities.

Another method preserving jumps worth to cite is the Immersed Interface Method (IIM), which is second-order accurate [LeVeque and Li 1994]. Despite its accuracy, the linear system associated to IIM is not symmetric, preventing the use of fast linear solvers. Instead, the GFM preserves the symmetry of the linear solver.

Initially proposed as a correction scheme for Finite Difference discretizations, the method has been generalized to Finite Volume discretizations [Vukčević, Jasak, and Gatin 2017]. The implementation in `YALES2BIO` is inspired to the version of the GFM presented in [Desjardins, Moureau, and Pitsch 2008]. The originality of the present approach consists in adapting GFM to the framework of an Immersed Boundary treatment of the membrane. This constitutes, up to our knowledge, the first implementation of an IBM+GFM solver.

A Finite Volume discretization is considered a starting point. On the grid Ω_h, consider a `GRID_DATA` ϕ with values located at vertices. Taking back the notations of subsection 3.2.2, 3.22a integrated over the finite volume Ω_V reads:

$$\sum_{W \in \mathcal{V}_V} \int_{S_W} \sigma \nabla \phi \cdot \hat{\boldsymbol{n}} = 0. \tag{3.24}$$

The method establishes a technique to approximate the flux along an edge VW which is intersected by the interface Σ_h as shown in Figure 3.9.

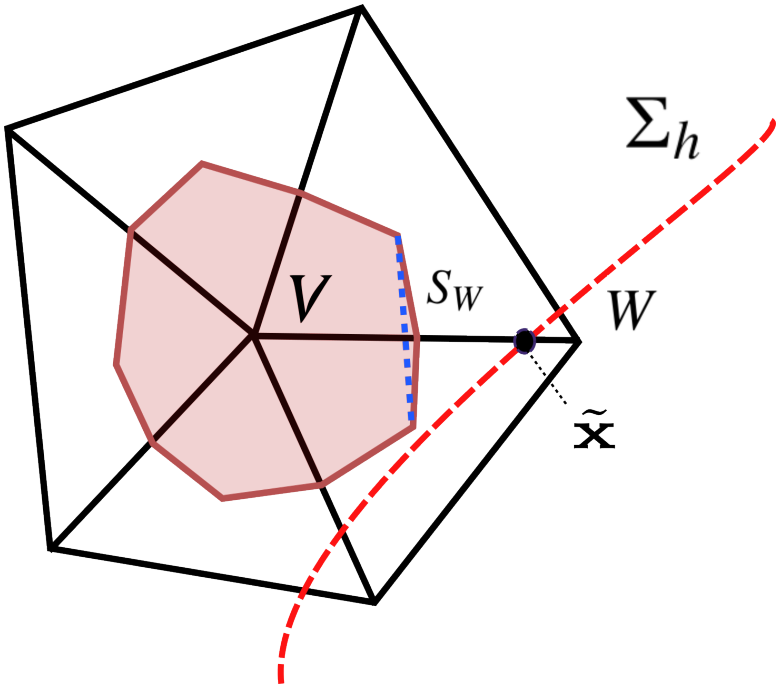


Figure 3.7: A node $V \in \mathcal{V}$ whose stencil is intersected by the interface Σ_h, which crosses the edge VW at point $\tilde{\boldsymbol{x}}$.

The Ghost Fluid Method can be illustrated by considering two virtual fields: $\phi_{\text{int}} : \Omega \to \mathbb{R}$ and $\phi_{\text{ext}} : \Omega \to \mathbb{R}$, both continuous on the entire domain and such that:

$$\phi_V = \begin{cases} \phi_{\text{int},V} & \text{if } V \in \Omega_{\text{int}} \\ \phi_{\text{ext},V} & \text{if } V \in \Omega_{\text{ext}}. \end{cases}$$

The restricitions $\phi_{\text{int}}|_{\Omega_{\text{ext}}}$ and $\phi_{\text{ext}}|_{\Omega_{\text{int}}}$ represent the continuous prolongations of $\phi|_{\Omega_{\text{int}}}$ and $\phi|_{\Omega_{\text{ext}}}$ beyond the interface. The difference $[\phi] : \Omega \to \mathbb{R}$ such that $[\phi](\boldsymbol{x}) = \phi_{\text{ext}}(\boldsymbol{x}) - \phi_{\text{int}}(\boldsymbol{x})$ represents the extension to the bulk of the potential jump along the interface. In particular, condition (3.22c) can be reformulated as:

$$\sigma_{\text{ext}} \nabla[\phi] \cdot \boldsymbol{n}_\Sigma + [\sigma] \nabla\phi_{\text{int}} \cdot \boldsymbol{n}_\Sigma = 0, \tag{3.25}$$

or equivalently:

$$[\nabla\phi] \cdot \boldsymbol{n}_\Sigma = \frac{-[\sigma]}{\sigma_{\text{ext}}} \nabla\phi_{\text{int}} \cdot \boldsymbol{n}_\Sigma. \tag{3.26}$$

To fix the ideas, consider V lying in the internal region Ω_{int}, and W in Ω_{ext}. The flux across the face S_W is replaced by the flux of ϕ_{int}. Since ϕ_{int} is continuous across the interface, it can be estimated using a regular finite difference scheme, for example the one introduced in subsection 3.2.2:

$$\int_{S_W} \sigma \nabla\phi \cdot \hat{\boldsymbol{n}} \mapsto \sigma_{\text{int}} \nabla\phi_{\text{int}}|_{VW} \cdot \boldsymbol{S}_W \tag{3.27}$$

$$= \frac{\sigma_{\text{int}}(\boldsymbol{x}_W - \boldsymbol{x}_V) \cdot \boldsymbol{S}_W}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|^2} (\phi_{\text{int},W} - \phi_V) \tag{3.28}$$

$$= \frac{\sigma_{\text{int}}(\boldsymbol{x}_W - \boldsymbol{x}_V) \cdot \boldsymbol{S}_W}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|^2} (\phi_W - [\phi]|_W - \phi_V). \tag{3.29}$$

In the passage to the third line, the value $\phi_{\text{int},W}$ is expressed in terms of $[\phi]_W$. The value of the jump outside the interface in not known. It is extrapolated by considering a Taylor expansion around the intersection $\tilde{\boldsymbol{x}}$ between the edge VW and the interface:

$$[\phi]_W \mapsto [\phi]_{\tilde{\boldsymbol{x}}} + \|\boldsymbol{x}_W - \tilde{\boldsymbol{x}}\| \nabla[\phi]|_{\tilde{\boldsymbol{x}}} \cdot \boldsymbol{n}_\Sigma \tag{3.30}$$

$$= [\phi]_{\tilde{\boldsymbol{x}}} - \|\boldsymbol{x}_W - \tilde{\boldsymbol{x}}\| \frac{[\sigma]}{\sigma_{\text{ext}}} \nabla\phi_{\text{int}}|_{\tilde{\boldsymbol{x}}} \cdot \boldsymbol{n}_\Sigma \tag{3.31}$$

$$= [\phi]_{\tilde{\boldsymbol{x}}} - \frac{\|\boldsymbol{x}_W - \tilde{\boldsymbol{x}}\|}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|} \frac{[\sigma]}{\sigma_{\text{ext}}} (\phi_{\text{int},W} - \phi_{\text{int},V}) \tag{3.32}$$

$$= [\phi]_{\tilde{\boldsymbol{x}}} - \frac{\|\boldsymbol{x}_W - \tilde{\boldsymbol{x}}\|}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|} \frac{[\sigma]}{\sigma_{\text{ext}}} (\phi_W - [\phi]_W - \phi_V), \tag{3.33}$$

where in the first and second line interface conditions (3.22b) and (3.26) are enforced and in the passage to the third line, a two-point approximation of the flux $\nabla\phi_{\text{int}}|_{\tilde{\boldsymbol{x}}} \boldsymbol{n}_\Sigma$ is introduced. Inverting this relation the extrapolated jump $[\phi]_W$ can then be epressed as:

$$[\phi]_W = \frac{\sigma_{\text{ext}}}{\sigma^*} J_\Sigma(\tilde{\boldsymbol{x}}) + (1 - \frac{\sigma_{\text{ext}}}{\sigma^*})(\phi_W - \phi_V), \tag{3.34}$$

where:

$$\sigma^* = \theta\sigma_{\text{int}} + (1 - \theta)\sigma_{\text{ext}},$$

with:

$$\theta = \frac{\|\boldsymbol{x}_W - \tilde{\boldsymbol{x}}\|}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|}.$$

Plugging (3.34) into (3.26), the approximation of the flux along the cut edge reads:

$$\int_{S_W} \sigma \nabla\phi \cdot \hat{\boldsymbol{n}} \mapsto \frac{(\boldsymbol{x}_W - \boldsymbol{x}_V) \cdot \boldsymbol{S}_W}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|^2} \frac{\sigma_{\text{int}}\sigma_{\text{ext}}}{\sigma^*} (\phi_W - \phi_V) - \frac{(\boldsymbol{x}_W - \boldsymbol{x}_V) \cdot \boldsymbol{S}_W}{\|\boldsymbol{x}_W - \boldsymbol{x}_V\|^2} \frac{\sigma_{\text{int}}\sigma_{\text{ext}}}{\sigma^*} J_\Sigma(\tilde{\boldsymbol{x}}).$$

The GFM approximation of the flux so obtained is replaced in the linear system 3.24 at the lines corresponding to V and W with opposite sign, so that the correction preserves the symmetry of the discrete operator. To resume, implementation of GFM requires locating the position of the intersection $\tilde{\boldsymbol{x}}$ for every edge cut by the interface, and read the value of the jump J_Σ stored on the surface Σ. Therefore, preliminary to the assembly of the system, an interface tracking algorithm is executed to retrieve the position of the `SURFACE` Σ_h relative to the `GRID` Ω_h. The algorithm is summarized in 2.

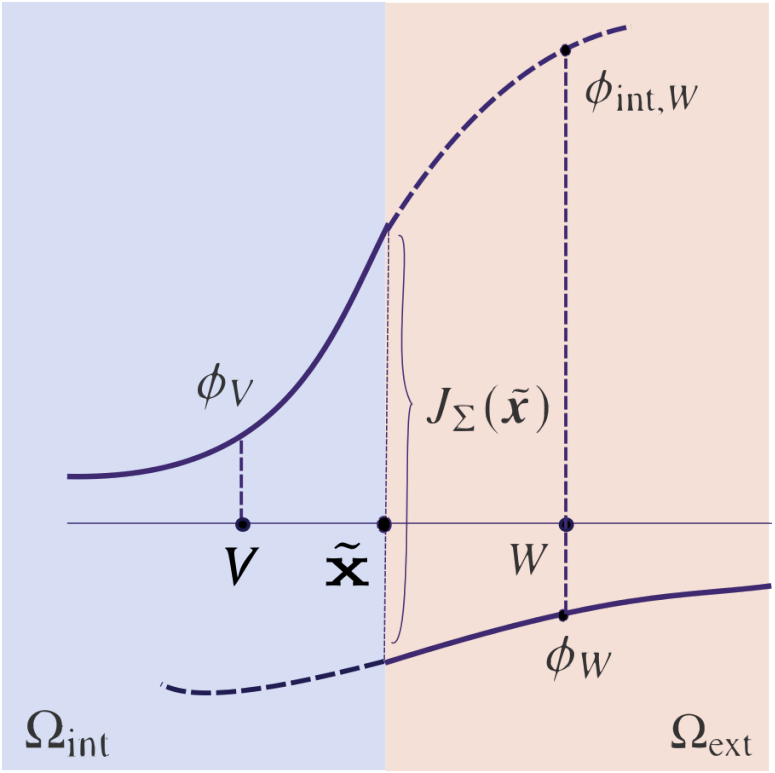


Figure 3.8: Potential ϕ and discontinuity along an edge VW cut by the interface.

Algorithm 2 Interface tracking algorithm

```
for i ← 1,no_facet_Σ_h do
    for j ← 1,no_edge_Ω_h do
        if (check_intersection (i, j) == true) then
            Mark ends of j (0 int, 1 ext);
            Record intersection position for edge j;
        end if
    end for
end for
```

3.4.2 Electric Field and Force Calculation

The electric field $\nabla\phi$ is estimated by the nodal `GRID_DATA` $\underline{\boldsymbol{E}}$, which is obtained by applying to $\underline{\phi}$ the discrete gradient operator $\mathcal{G}$, introduced in Subsection 3.2.2. If V is such that the edge VW is not crossed by the interface Σ_h for any $W \in \mathcal{V}_V$, the nodal value of the gradient is approximated as:

$$\boldsymbol{E}_V = \mathcal{G}_V(\phi) = \sum_{W \in \mathcal{V}_V} \phi_{VW} \cdot \boldsymbol{S}_W.$$

The gradient approximation has to be corrected for nodes close to the interface. In YALES2BIO, the gradient close to the interface is extrapolated from the bulk. If $V \in \Omega_\bullet$ with $\bullet \in \{\text{int}, \text{ext}\}$, the gradient is extrapolated as a weighted average of the gradient at neighbours W falling on the same side of V:

$$\boldsymbol{E}_V = \frac{\sum_{W \in \mathcal{V}_V \cap \Omega_\bullet} \boldsymbol{E}_W |S_W|}{\sum_{W \in \mathcal{V}_V \cap \Omega_\bullet} |S_W|}. \tag{3.35}$$

The discontinous electric field is then interpolated to estimate the electrice force acting on the surface. Let $\Omega_h = \{\mathcal{V}_\Omega, \mathcal{T}_\Omega\}$. Considering $\Sigma_h = \{\mathcal{V}_\Sigma, \mathcal{T}_\Sigma\}$, let the force be represented as a SURFACE_DATA $\underline{\boldsymbol{F}}_{MW}$ defined at nodes. With $V \in \mathcal{V}_\Sigma$, call $T_V \in \mathcal{T}_\Omega$ the element of Ω_h containing V. Let $\mathcal{V}_\mathcal{T}$ be the set of vertices of T and let $\mathcal{V}_V^\bullet := \{W \in \mathcal{V}_T : W \in \Omega_\bullet\}$, with $\bullet \in \{\text{int}, \text{ext}\}$.

Force is calculated as the jump of the normal trace of Maxwell tensor:

$$\begin{aligned}\boldsymbol{F}_{MW,V} =& \epsilon_{\text{ext}} \boldsymbol{E}_{EXT,V} [\boldsymbol{E}_{EXT,V} \cdot \boldsymbol{n}_\Sigma - \frac{1}{2} \boldsymbol{E}_{EXT,V} \cdot \boldsymbol{E}_{EXT,V}] \\ & - \epsilon_{\text{int}} \boldsymbol{E}_{INT,V} [\boldsymbol{E}_{INT,V} \cdot \boldsymbol{n}_\Sigma - \frac{1}{2} \boldsymbol{E}_{INT,V} \cdot \boldsymbol{E}_{INT,V}],\end{aligned}$$

where:

$$\boldsymbol{E}_{\bullet,V} = \frac{\sum_{W \in \mathcal{V}_V^\bullet} \boldsymbol{E}_W}{card(\mathcal{V}_V^\bullet)}.$$

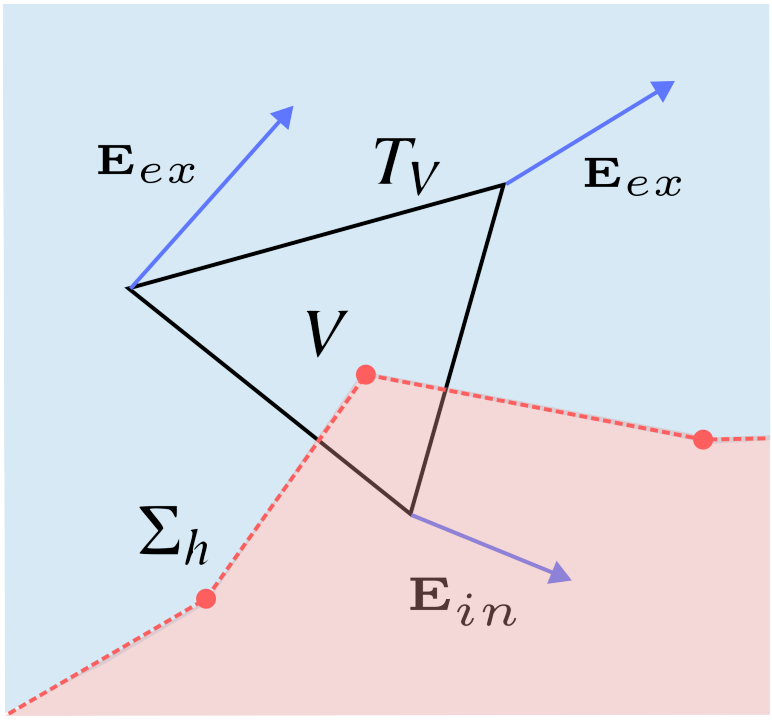


Figure 3.9: A node V on Σ_h and the element $T_V \in \Omega_h$, whose vertices the electric field is interpolated from.

3.4.3 Advancement of Potential Jump

The interpolation of electric field operated to estimate the electric force is also employed to estimate the source term of Equation (3.23), describing the advancement of surface potential J_Σ. The latter is

a SURFACE_DATA with values located on the elements the surface ($\mathcal{T}_\Sigma$). The interpolated electric field defined by (3.35) is transported to elements by defining a element-based SURFACE_DATA $\underline{\boldsymbol{E}}$ such that:

$$\boldsymbol{E}_T^\bullet = \frac{\sum_{V \in \mathcal{V}_T} \boldsymbol{E}_V^\bullet}{card(\mathcal{V}_T)}, \qquad \bullet \in \{\text{int}, \text{ext}\}.$$

The evolution equation is discretized in time adopting an explicit Euler scheme:

$$J_{\Sigma,T}^{n+1} = J_{\Sigma,T}^n + \frac{\tau}{C}(\theta\sigma_{\text{int}}\boldsymbol{E}_{\text{int},T} + (1-\theta)\sigma_{\text{ext}}\boldsymbol{E}_{\text{ext},T}) \cdot \boldsymbol{n}_T,$$

where $\boldsymbol{n}_T$ is the outward normal on element T, and a weighted average of the flux $\sigma\nabla\phi \cdot \boldsymbol{n}_\Sigma$ is considered, depending on a user-dependent weight θ.

3.5 Summary

To conclude the chapter, we provide an overview of the solver YALES2BIO+ESS. The diagram in Figure 3.10 presents a scheme of one time iteration. The list of the main data structures introduced in the solver is provided in Table 3.1.

At the beginning of an iteration, the total forces ($\boldsymbol{F}$) acting on the membrane are computed. The components of the forces linked to strain and curvature can be computed given the current state of the membrane (step MF). On the other hand, the computation of electric foces requires the determination of the electric field in the domain ($\boldsymbol{e}$) and the advancement of the potential jump at the interface (J_Σ). This step is performed by the electrostatic module (ESS).

The following step consists in updating the velocity of the flow. In the framework of the Immersed Boundary Method, membrane forces act on the fluid under the form of an equivalent distribution of volume forces in the domain ($\boldsymbol{f}$). The diffusion algorith (IBM-diff) computes the distribution of volume forces.

The advancement of the velocity is then realized by a Finite Volume solver for the Incompressible Navier-Stokes equations with a fractional step algorithm (NS).

The last step consists in interpolating the updated velocity onto the surface ($\boldsymbol{V}$) (IBM-int). The position of the vertices is then updated and a correction *a posteriori* is applied to enforce conservation of the volume (ADV).

Data	Description	Type
$\underline{\boldsymbol{X}}$	Membrane Node Position	SURFACE, vertices
$\underline{\boldsymbol{V}}$	Membrane Velocity	SURFACE, vertices
$\underline{\boldsymbol{F}}$	Membrane Forces (curvature, elastic)	SURFACE, vertices
$\underline{J_\Sigma}$	Potential Jump	SURFACE, faces
$\underline{\boldsymbol{F}}$	Membrane Forces(electric)	SURFACE, vertices
$\underline{\boldsymbol{v}}$	Fluid Velocity	GRID, vertices
$\underline{\boldsymbol{f}}$	Fluid Volume Force	GRID, vertices

Table 3.1: Main data of the solver.

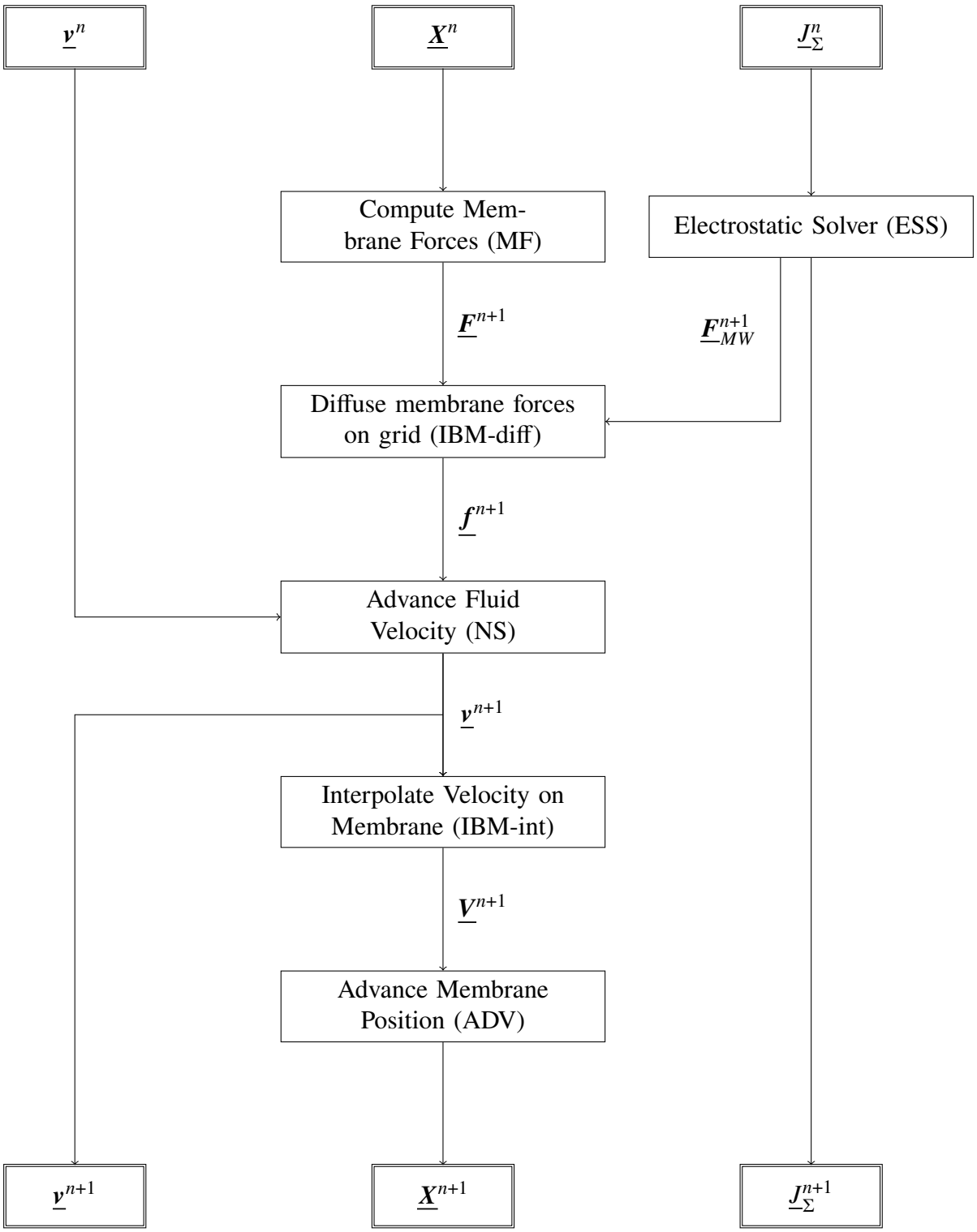


Figure 3.10: Flow diagram of the time-advancement algorithm of the solver. The central column displays the steps performed to advance the position of the membrane, performing the coupling via the Immersed Boundary Method. On the right, the electrostatic solver. The output of the electricity algorithm ESS are the electric membrane forces that are summed to the membrane forces before the diffusion step and the advanced potential jump.

CHAPTER

FOUR

YALES2BIO BENCHMARKS

In this chapter we present a suite of validation benchmarks for the solver YALES2BIO+ESS. Three different scenarios are considered. Since electrodeformation is an experimental feature in YALES2BIO, test cases are chosen among the most documented configurations from the literature on capsule and erythrocyte electrohydrodynamics. These demos catch in a complete way the salient features of the physical phenomenon targeted, making them a reasonable benchmark to validate the solver.

In Section 4.1, the focus is on the charging process of a spherical capacitor that does not move or deform. The capacitor models the electric behaviour of a biological membrane, with the advantage of removing the complexity introduced by deformation. Therefore, the simulation of the capacitor constitutes a preliminary benchmark for the electric ESS before considering more complex scenarios where ESS interacts with other modules of the solver. For this test case, it is possible to compare the output of the simulation with an analytical solution. An exhaustive gallery of convergence plots is available in Appendix A.

The next two test cases are concerned with moving membranes. Section 4.2 presents the transition of a spherical capsule subject to a DC field. The solver is validated by comparing the deformation of the capsule along time with established results from the literature. Finally, section 4.3 presents the simulation of the delaying effect of an electric field on the tumbling motion of a vesicle in a shear flow.

The end of the chapter is devoted to a thorough discussion on the accuracy and robustness of the solver, highlighting its features and limitations. Overall, YALES2BIO's performances encourage exploring further the numeric approach at the base of ESS. Despite the success in predicting some expected phenomena, the robustness of the solver is still an open issue. The challenges emerged testing ESS are a reason to question more in depth the challenges linked to numerical treatment of interface discontinuities. These questions motivate the developments presented in the second part of this thesis.

4.1 Charge build-up on a spherical capacitor

The first test considers a membrane that does not move. The membrane is treated as an interface between media with different electric conductivity, acting as a capacitor (see figure 4.1). When the system, initially neutral, is subject to an external field, electric current flows from the bulk towards the interface. Since the interface is impermeable to electric charges, the latter starts accumulating from both sides with opposite sign. This determines the appearance of a sharp interface jump of electric potential across the interface. The system reaches an electrostatic equilibrum characterized

by vanishing electric field inside the capacitive shell.

In subsection 4.1.1, the benchmark along with its main parameters and dimensional analysis is presented. The results of the simulations with $d = 2$ and $d = 3$ are respectively presented in 4.1.2 and 4.1.3. The numerical solution can be compared with a reference analytical solution to perform an error analysis of ESS. To avoid making this chapter heavy with a large number of convergence plot, the presentation of error analysis has been moved to Appendix A.

4.1.1 Benchmark presentation

Let $d \in \{2, 3\}$. Consider a spherical shell Σ in $\mathbb{R}^d$, with radius R and outward normal $\boldsymbol{n}_\Sigma$. Call Ω_{int} the region of space bounded by Σ, and Ω_{ext} the external region. Let C be the capacitance of the shell, and σ_{int}, σ_{ext} be respectively the electric conductivity of each subregion. Initially, the shell supports no interfacial electric charge. The system is characterized by a time-dependent electrostatic potential $\phi : \Omega \setminus \Sigma \times [0, \infty) \to \mathbb{R}$ and an interface potential jump $J_\Sigma : \Sigma \times [0, \infty) \to \mathbb{R}$ that satisfy the following set of conditions:

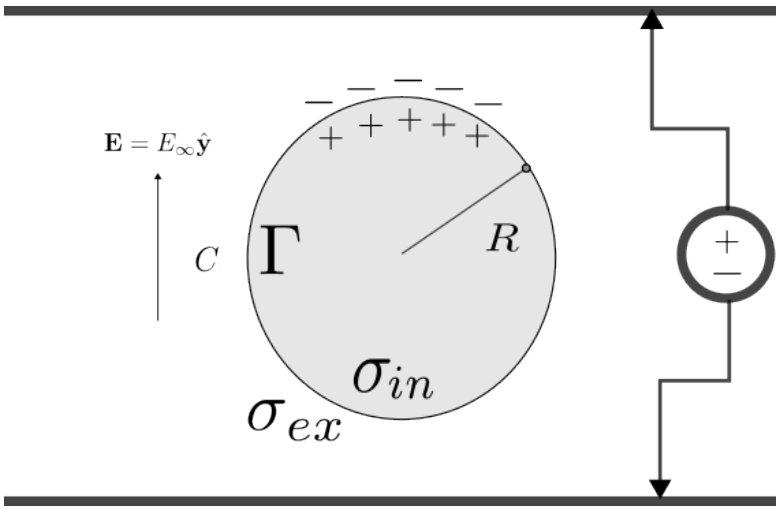


Figure 4.1: Physical phenomenon simulated by ESS. Upon imposition of an external electric field $\boldsymbol{E}$, electric charge cumulates over the two sides of the interface Σ with capacitance C, giving place to a jump of electric potential.

$$
\begin{cases}
-\nabla \cdot (\sigma_\bullet \nabla \phi) = 0 & \text{in } \Omega_\bullet,\ \bullet \in \{\text{int}, \text{ext}\} \\
[\phi] = J_\Sigma & \text{on } \Sigma \\
[\sigma_\bullet \nabla \phi_\bullet \cdot \boldsymbol{n}_\Sigma] = 0 & \text{on } \Sigma \\
C \partial_t J_\Sigma = \sigma_\bullet \nabla \phi_\bullet \cdot \boldsymbol{n}_\Sigma & \text{on } \Sigma,\ \bullet \in \{\text{int}, \text{ext}\} \\
\nabla \phi(\boldsymbol{x}, t) = \boldsymbol{E}_\infty & \|\boldsymbol{x}\| \to \infty,\ t > 0 \\
J_\Sigma(\cdot, 0) = 0 & \text{on } \Sigma.
\end{cases}
\tag{4.1}
$$

An analytical solution for the boundary value problem can be obtained by separation of variables. Consider a system of cylindrical coordinates (r, θ, z), where r represents the radial coordinate, θ the azimuthal coordinate and z the axial coordinate. Let the axis of the coordinate system be oriented in the direction of $\boldsymbol{E}_\infty$. The solution (ϕ, J_Σ) reads:

$$
\phi(\boldsymbol{x}, t) = e^{-\frac{t}{\tau_C}} (\phi^0(\boldsymbol{x}) - \phi^\infty(\boldsymbol{x})) + \phi_\infty(\boldsymbol{x}), \qquad J_\Sigma(\theta, t) = \frac{3}{2}(1 - e^{-\frac{t}{\tau_C}}) E_\infty R \cos\theta,
$$

where the characteristic charging time τ_C reads:

$$\tau_c = CR(\frac{1}{\sigma_{\text{int}}} + \frac{1}{(d-1)\sigma_{\text{ext}}}),$$

and $\phi^0(\boldsymbol{x}) = \phi(\boldsymbol{x}, 0)$, $\phi^\infty(\boldsymbol{x}) = \lim_{t\to\infty} \phi(\boldsymbol{x}, t)$ represent the spatial distribution of the electric field respectively at the beginning of the simulation and at electrostatic equilibrium. It can be remarked that the equilibrium potential ϕ_∞ is independent of the conductivities $\{\sigma_{\text{int}}, \sigma_{\text{ext}}\}$. The space distributions of ϕ^0, ϕ^∞ depend on the space dimension d and are explicited in the paragraphs discussing the solutions in different dimensions.

The data of the testcase and the numeric values adopted in the simulation are summarized in Table 4.1 along with the corresponding physical dimension. In order to determine the dimensionless parameters governing the system, we introduce a recasting of Problem 4.1. Considering the following scalings:

$$\boldsymbol{x} = R\hat{\boldsymbol{x}}, \quad t = \tau_C t, \quad \phi = E_\infty R\hat{\phi}, \quad J_\Sigma = E_\infty R\hat{J}_\Sigma.$$

the scaled system of equations reads (after dropping all the superscripts):

$$\begin{cases} -\nabla \cdot (\tilde{\sigma}_\bullet \nabla\phi) = 0 & \text{in } \Omega_\bullet,\ \bullet \in \{\text{int}, \text{ext}\} \\ [\phi] = J_\Sigma & \text{on } \Sigma \\ [\tilde{\sigma}_\bullet \nabla\phi_\bullet \cdot \boldsymbol{n}_\Sigma] = 0 & \text{on } \Sigma \\ \tilde{C}\partial_t J_\Sigma = \tilde{\sigma}_\bullet \nabla\phi_\bullet \cdot \boldsymbol{n}_\Sigma & \text{on } \Sigma,\ \bullet \in \{\text{int}, \text{ext}\} \\ \nabla\phi(\boldsymbol{x}, t) = \hat{\boldsymbol{z}} & \|\boldsymbol{x}\| \to \infty,\ t > 0 \\ J_\Sigma(\cdot, 0) = 0 & \text{on } \Sigma, \end{cases} \tag{4.2}$$

where $\hat{\boldsymbol{z}}$ is the unit vector directed as the exteral electric field and:

$$\tilde{\sigma}_{\text{int}} = \sigma_r = \frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}}, \quad \tilde{\sigma}_{\text{ext}} = 1, \quad \tilde{C} = \frac{CRE_\infty}{\tau_C}.$$

The regime of the system depends on 2 parameters: the dimensionless capacitance $\tilde{C}$ and the conductivity ratio σ_r.

Parameter	Value	Dimension
R	10^{-5}	m
E_∞	$4.47 \cdot 10^3$	Volt m
σ_{ext}	10^{-4}	Sm m^{-1}
σ_{int}	$\sigma_r\sigma_{\text{ext}}$	Sm m^{-1}
C	10^{-4}	Farad m^{-2}

Table 4.1: Physical parameters of the Testcase.

4.1.2 Case $d = 2$

With $d = 2$, the initial and equilibrium potential distributions read:

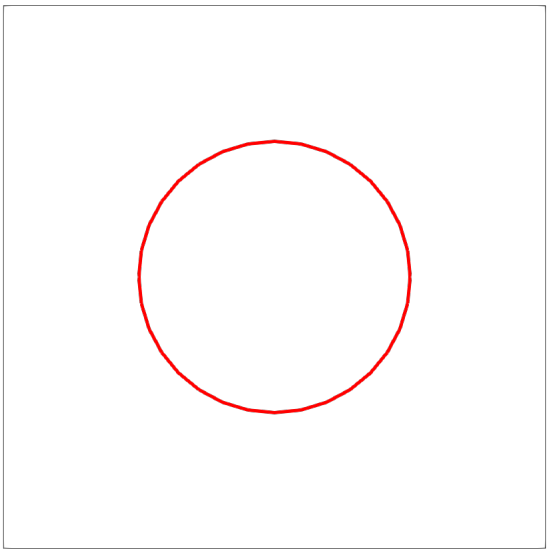
Parameter	Value
Domain Diameter D	$2R$
h_Ω	$0.2R$
h_Σ	$0.19R$
Time Step τ	$10^{-3}\tau_C$

Table 4.2: Numerical Parameters for the case $d = 2$

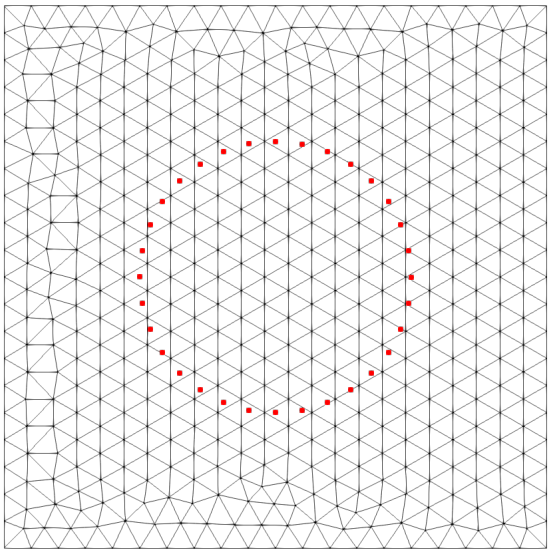
$$\phi_0 = \begin{cases} 2E_\infty \frac{2\sigma_{\text{ext}}}{\sigma_{\text{ext}}+\sigma_{\text{int}}} r\cos\theta & r < R \\ E_\infty (r + (\frac{\sigma_{\text{ext}}-\sigma_{\text{int}}}{\sigma_{\text{ext}}+\sigma_{\text{int}}})\frac{R^2}{r})\cos\theta & r > R, \end{cases}$$
$$\phi_\infty = \begin{cases} E_\infty r(1+\frac{R^2}{r^2})\cos\theta & r < R \\ 0 & r > R. \end{cases} \tag{4.3}$$

The simulation is realized by taking as control volume the square $\Omega = [-2R, 2R]^2$. With the goal of a convergence analysis, each configuration is tested on a sequence of meshes with increasing refinement. Let $h_\Omega = D/10$ represent the coarsest refinement level. We consider a family of unstructured triangular meshes $\{\Omega_h\}_{h\in\mathcal{H}_\Omega}$, with mesh diameters in $\mathcal{H}_\Omega = \{h_\Omega/2^k,\ 1 \le k \le 4\}$. We consider a corresponding family of interface meshes $\{\Sigma_h\}_{h\in\mathcal{H}_\Sigma}$ with $\mathcal{H}_\Sigma = \{h_\Sigma/2^k,\ 1 \le k \le 4\}$ and $h_\Sigma = \pi R/16$. The discretization level of the interface is imposed so to have at most 1 surface vertex per cell of the background mesh. This cryterion reflects the one normally adopted in `YALES2BIO` in the context of the Immersed Boundary Method. The simulation is realized imposing Dirichlet-type boundary condition on the entire boundary $\partial\Omega$.

The rest of the section is dedicated to presenting results of the simulation. Each subsection presents results for a given regime of conductivity ratio. All other parameters are kept fixed and are as can be referred to in Tables 4.1 and 4.2. By default, every visualization in this section is provided taking the run of the test realized with the finest refinement level. Each subsection corresponds to a different regime of conductivity ratio.



(a) Domain Ω and interface Σ. In 2D the interface is replaced by a polygonal discretization.

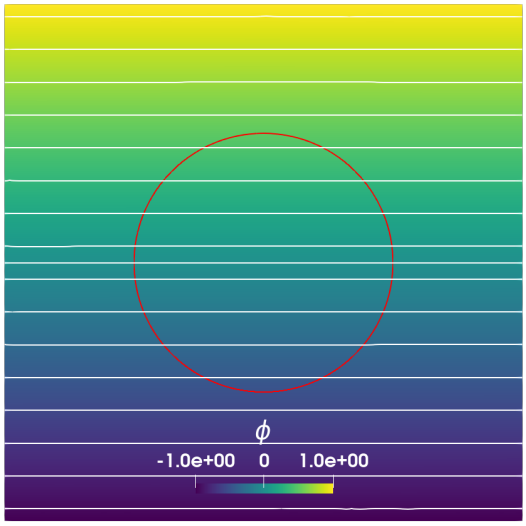


(b) Space discretization at coarsest refinement level (20 elements per side). Only the position of the vertices of the interface mesh Σ_h is represented to higihlight the distribution over the background mesh Ω_h. The discretization of the interface relative to the background is chosen so to get no more than one interface vertex per background element. The finest refinement level of the mesh familiy corresponds to 160 elements per side.

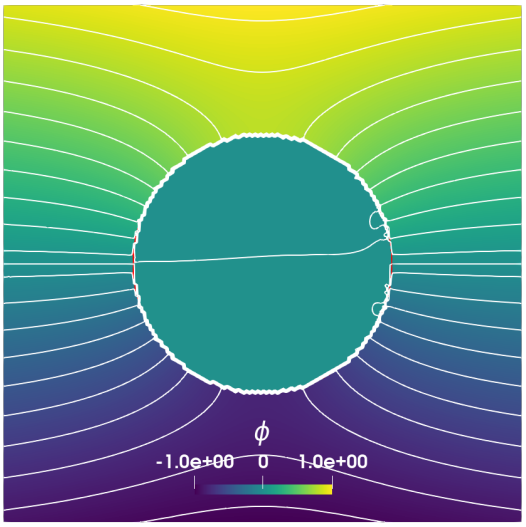
Figure 4.2: Geometry and discrete setting for the 2D simulation.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 1$

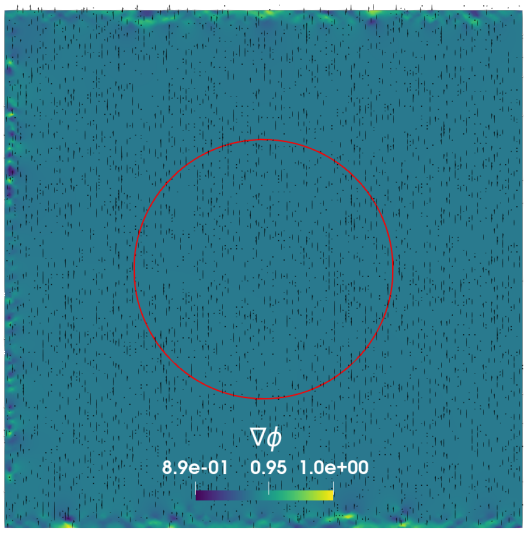
With $\sigma_{\text{int}} = \sigma_{\text{ext}}$, at $t = 0$ the electric potential ϕ, the electric field $\nabla\phi$ and Mawxell tensor $\bar{\bar{T}}_{MW}$ are continuous across Σ. Therefore electric force is initially absent. As long as electric charge starts building up at the interface, a compressional electric force appears at the equator (with respect to the axis determined by the external field $\boldsymbol{E}_\infty$). Force intensity grows until electrostatic equilibrium is reached, keeping a direction normal to Σ towards the center of the cell (see Figure 4.4). For $t \to \infty$, the system reaches an equilibrum characterized by vanishing electric field in Ω_{int} and an external electric field $\nabla\phi_{\text{ext}}$ with vanishing normal component at Σ (see Figure **??**). The interface appears as insulating.



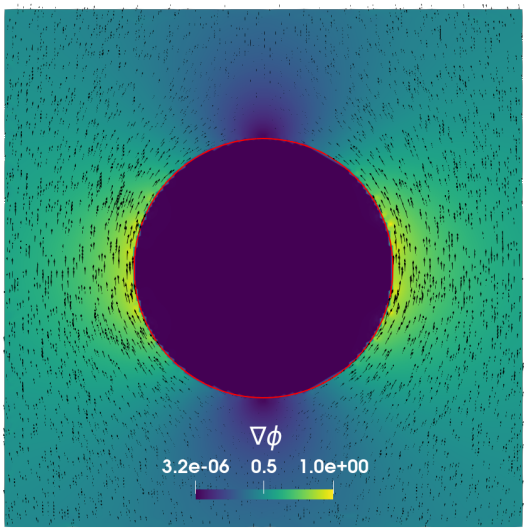
(a) $\phi_h|_{t=0}$



(b) $\phi_h|_{t\to\infty}$



(c) $\nabla\phi_h|_{t=0}$



(d) $\nabla\phi_h|_{t\to\infty}$

Figure 4.3: Numerical solution for electric potential distribution ϕ (top) and electric field $\nabla\phi$ (bottom), at initial time (left) and equilibrium (right). The scalar field ϕ is represented by a colormap. Morevore, isolines showing 20 potential levels are displayed. The vector field $\nabla\phi$ is represented by glyphs and a colormap to represent the magnitude. For $t \to \infty$, ϕ vanishes in Ω_{int} as expected. Moreover, the equilibrium electric field is tangent to the interface Σ.

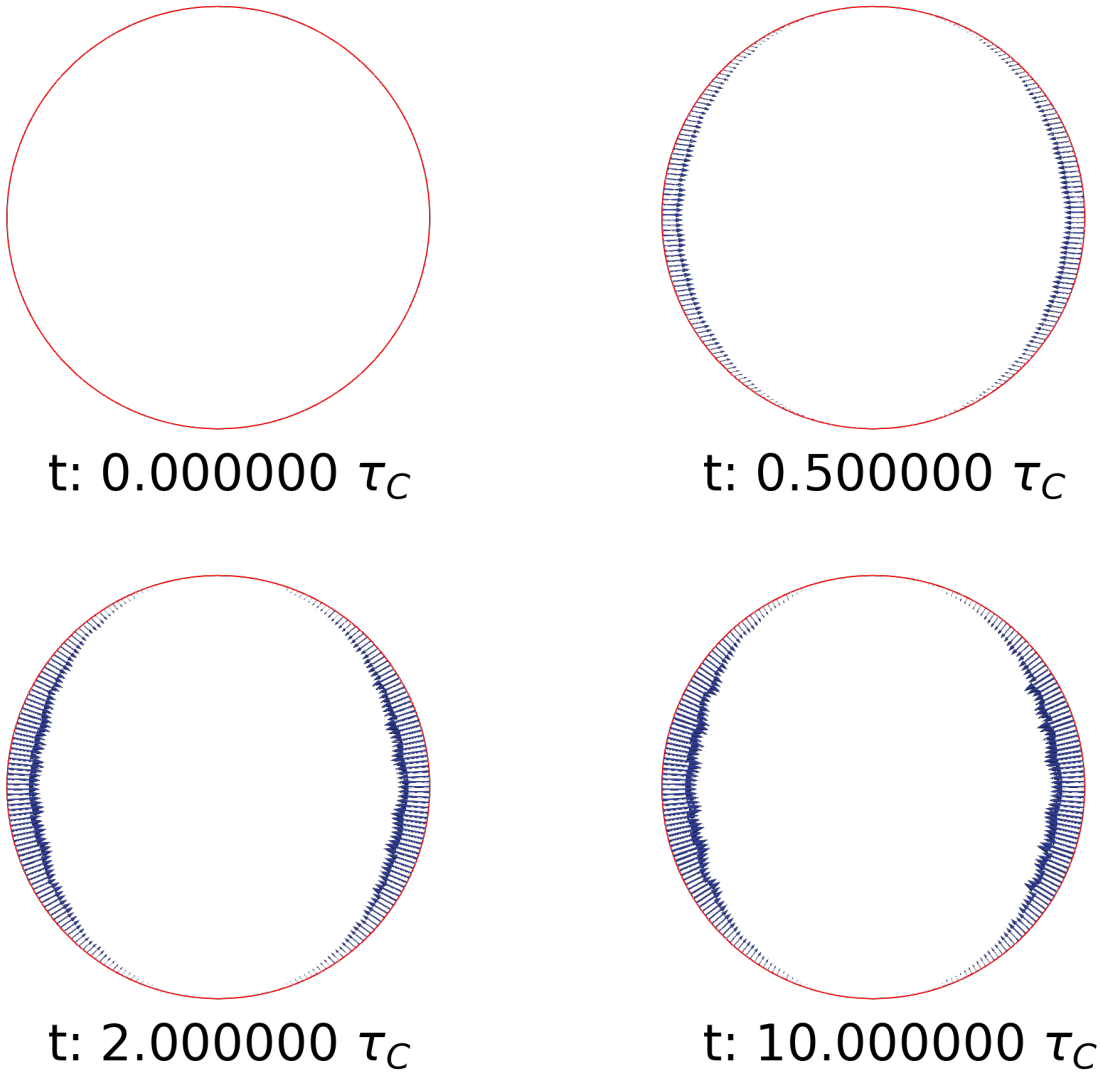
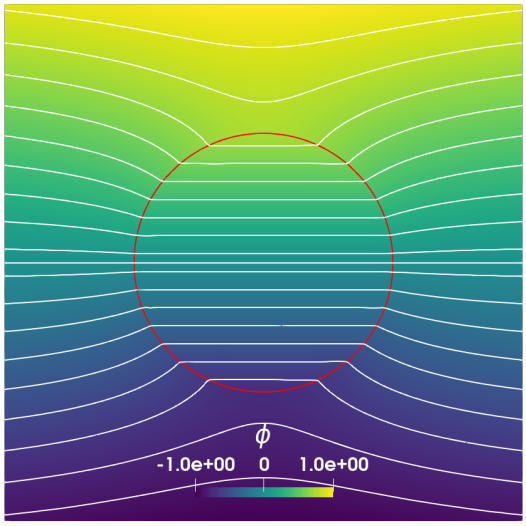


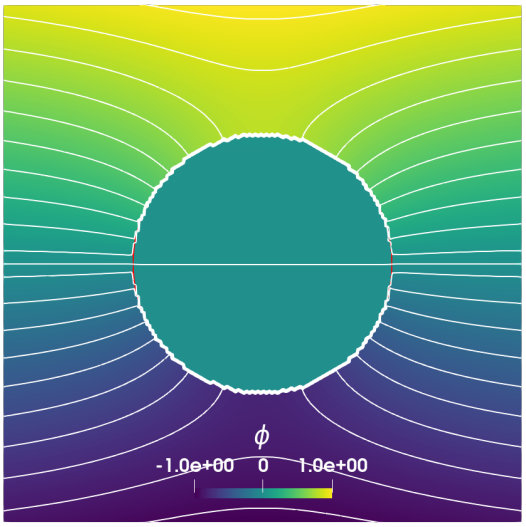
Figure 4.4: Evolution of the electric force $\boldsymbol{F}_{MW}$ along the simulation. With $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 1$, compressional forces at the build up along the equator.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 0.1$

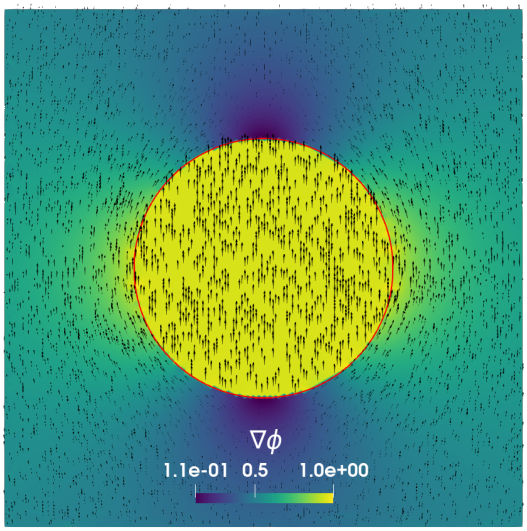
With $\sigma_{\text{int}} = 0.1\sigma_{\text{ext}}$, at $t = 0$ the internal electric field $\nabla\phi$ is stronger and uniform in the direction of the external field $\boldsymbol{E}_0$ (figure **??**), and the resulting electric force tends to compress the cell at the poles. After transitory, the profile of force is identical to the one obtained for $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 1$ (see Figure 4.6).



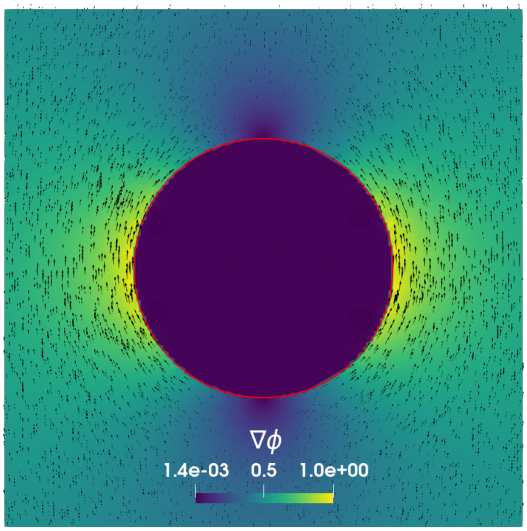
(a) $\phi_h|_{t=0}$



(b) $\phi_h|_{t\to\infty}$



(c) $\nabla\phi_h|_{t=0}$



(d) $\nabla\phi_h|_{t\to\infty}$

Figure 4.5: Numerical solution for electric potential distribution ϕ (top) and electric field $\nabla\phi$ (bottom), at initial time (left) and equilibrium (right). The scalar field ϕ is represented by a colormap. Morevore, isolines showing 20 potential levels are displayed. The vector field $\nabla\phi$ is represented by glyphs and a colormap to represent the magnitude.

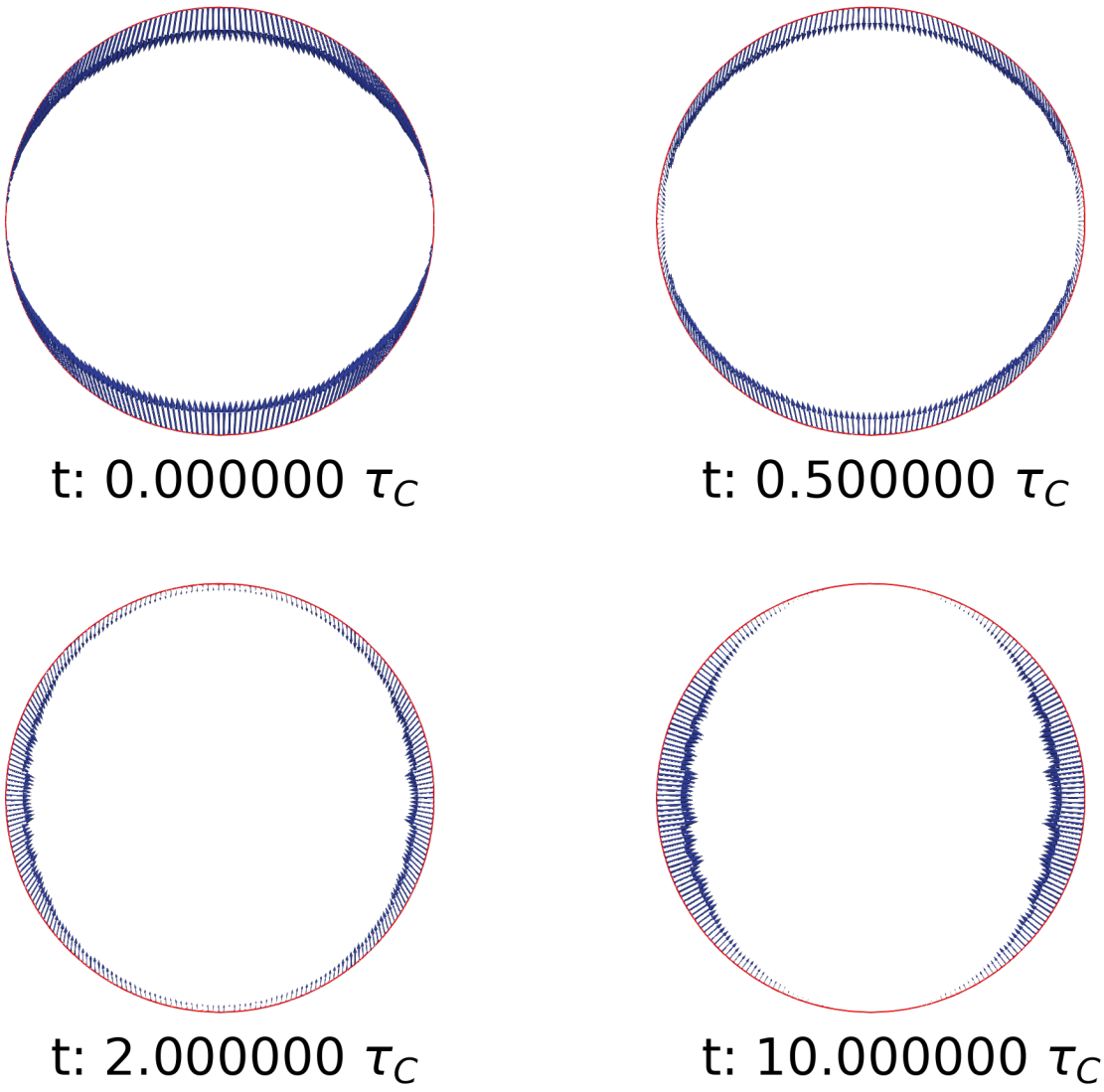
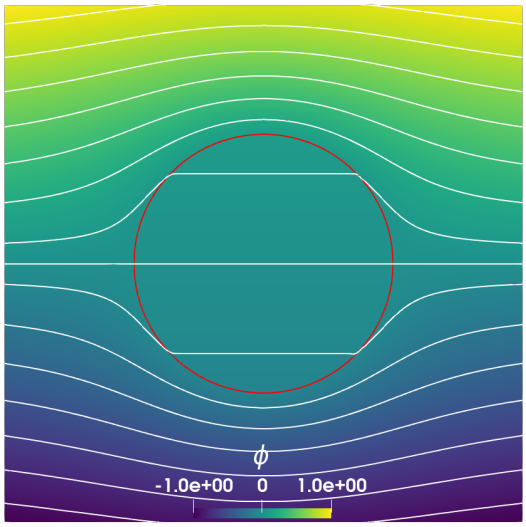


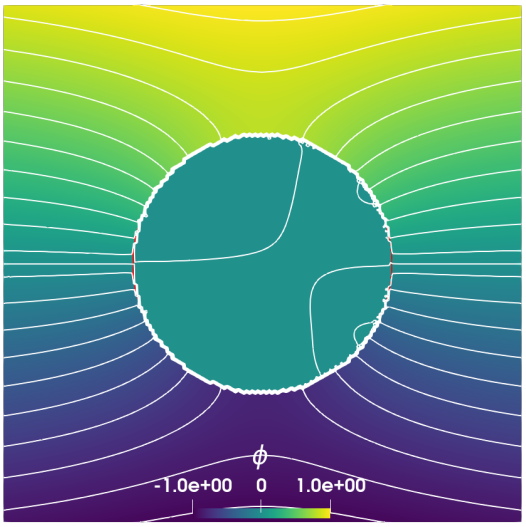
Figure 4.6: Evolution of the electric force $\boldsymbol{F}_{MW}$ along the simulation. With $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 0.1$, Maxwell forces tend to compress the capsule at poles. For $t \to \infty$, forces tend to compress the capsule along the equator.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 10$

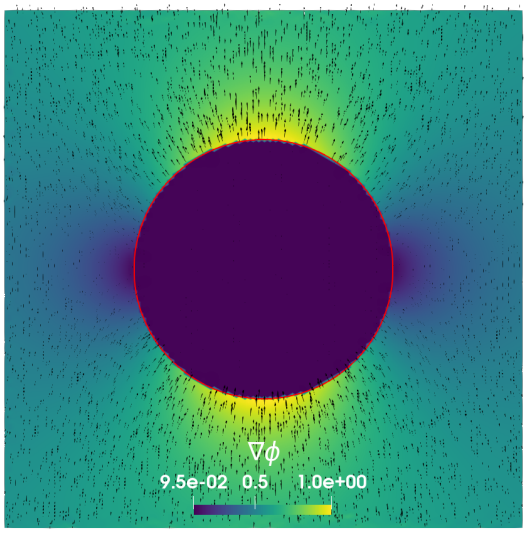
With $\sigma_{\text{int}} = 10\sigma_{\text{ext}}$, the initial external electric field $\nabla\phi_{\text{ext}}$ is tangent to the surface Σ (figure **??**). The initial profile of forces is characterized by an extensional trend at the pole. Force gets compressional at the equator at electrostatic equilibrium (see Figure 4.8).



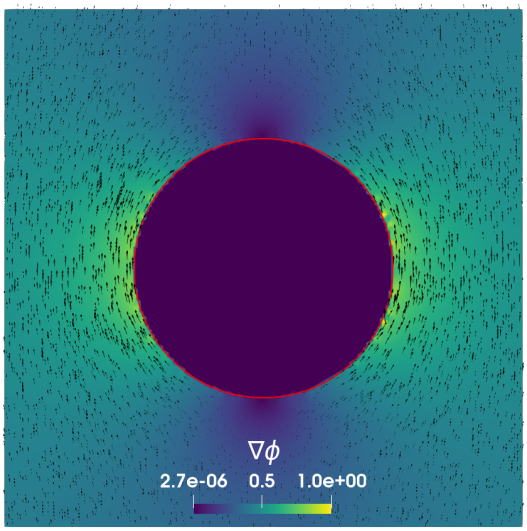
(a) $\phi_h|_{t=0}$



(b) $\phi_h|_{t\to\infty}$



(c) $\nabla\phi_h|_{t=0}$



(d) $\nabla\phi_h|_{t\to\infty}$

Figure 4.7: Numerical solution for electric potential distribution ϕ (top) and electric field $\nabla\phi$ (bottom), at initial time (left) and equilibrium (right). The scalar field ϕ is represented by a colormap. Morevore, isolines showing 20 potential levels are displayed. The vector field $\nabla\phi$ is represented by glyphs and a colormap to represent the magnitude.

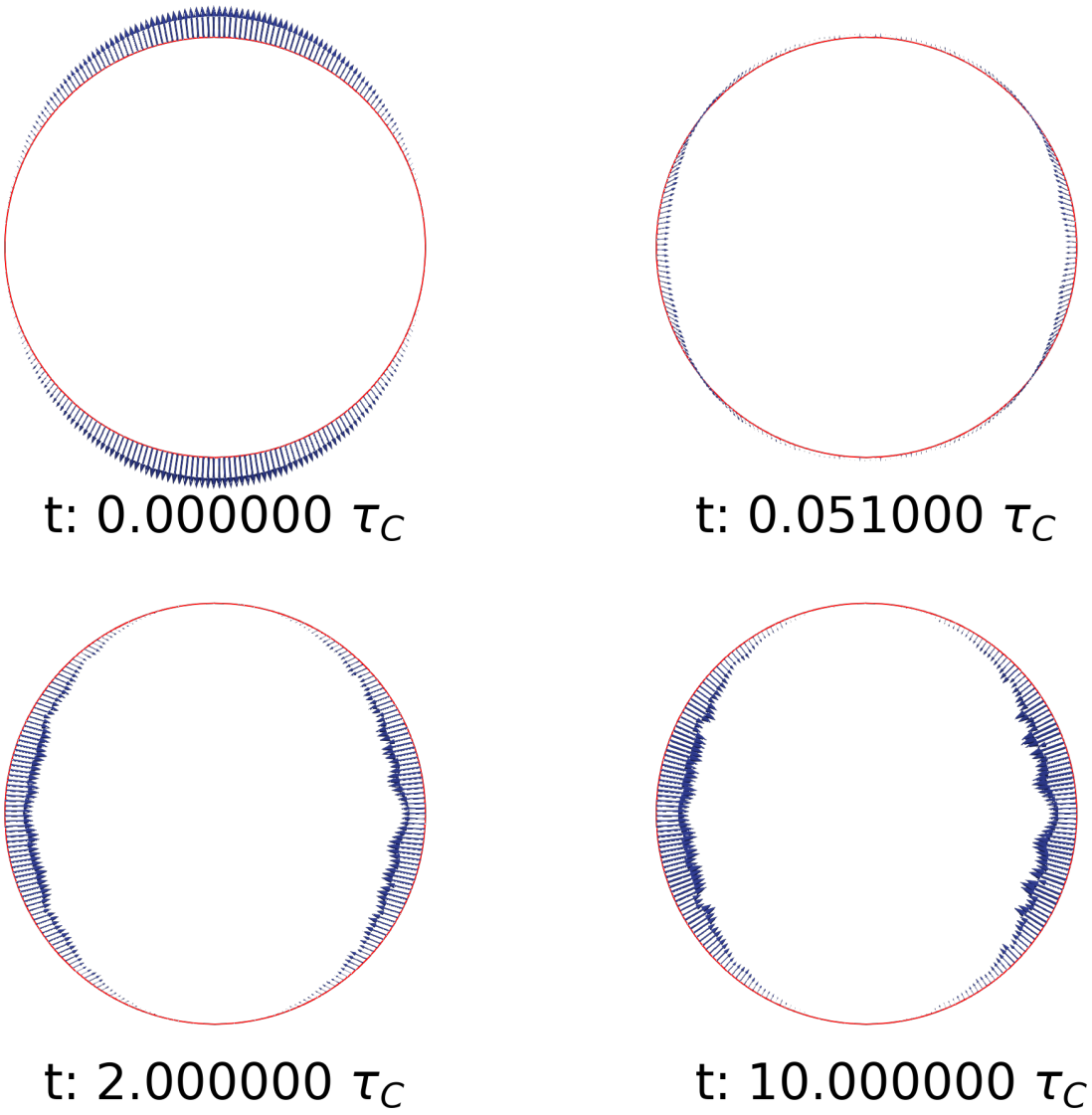


Figure 4.8: Evolution of the electric force $\boldsymbol{F}_{MW}$ along the simulation. With $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 10$, Maxwell forces act to extend the capsule at poles. For $t \to \infty$, forces tend to compress the capsule along the equator.

4.1.3 Case $d = 3$

With $d = 3$, the spatial distribution of ϕ read:

$$\phi_0 = \begin{cases} E_\infty \frac{3\sigma_{\text{ext}}}{2\sigma_{\text{ext}}+\sigma_{\text{int}}} r\cos\theta & r < R \\ E_\infty (r + (\frac{\sigma_{\text{ext}}-\sigma_{\text{int}}}{2\sigma_{\text{ext}}+\sigma_{\text{int}}})\frac{R^3}{r^2})\cos\theta & r > R, \end{cases}$$
$$\phi^\infty = \begin{cases} E_\infty r(1 + \frac{1}{2}(\frac{R}{r})^3)\cos\theta & r < R \\ 0 & r > R. \end{cases} \tag{4.4}$$

Simulations are realized by taking as control volume a cubic box with side $4R$ (see Figure 4.9). Dirichlet-type boundary conditions are imposed on the entire boundary $\partial\Omega$.

An error analysis is realized solving Problem 4.1 with conductivity ratio $\sigma_r \in \{0.1, 1, 10\}$. Simulations are repeated on a family of unstructured tetrehedral meshes $\{\Omega_h\}_{h\in\mathcal{H}_\Omega}$ with element diameters in

$\mathcal{H}_\Omega = \{h_\Omega/2^k,\ 1 \le k \le 4\}$, where $h_\Omega = 4R/5$ is the element diameter of the coarsest mesh. We consider a corresponding family of triangulations of the sphere $\{\Sigma_h\}_{h\in\mathcal{H}_\Sigma}$ with $\mathcal{H}_\Sigma = \{h_\Sigma/2^k,\ 1 \le k \le 4\}$, where $h_\Sigma = 0.15R$ is the face diameter of the coarsest surface mesh. The refinement of the sphere is imposed so to have at most 1 surface vertex per cell of the background mesh. The error analysis is presented in Appendix A.

Figures 4.10 and 4.11 show the results of a simulation with $\sigma_r = 0.1$, highlighting the evolution of the discontinuity of potential ϕ along the interface and the profile of electric force. The initial profile of forces is also showed for $\sigma_r = 10$ in Figure 4.12.

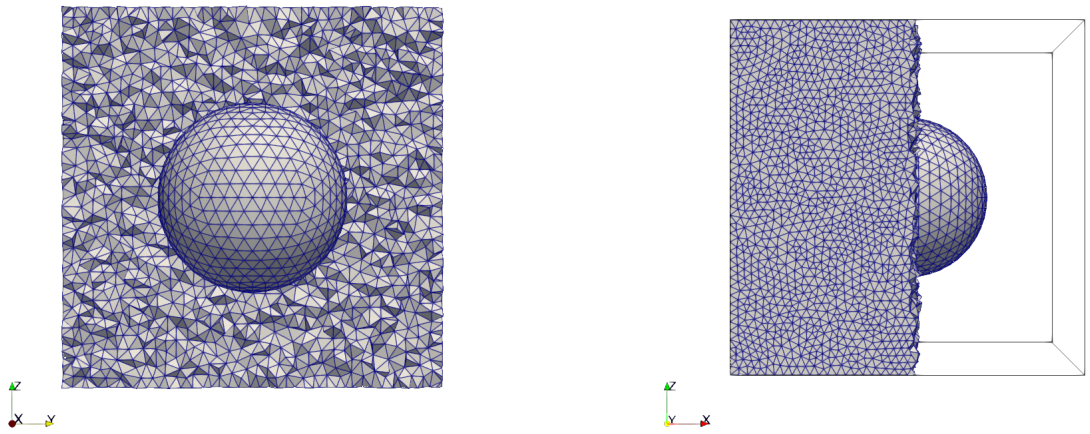
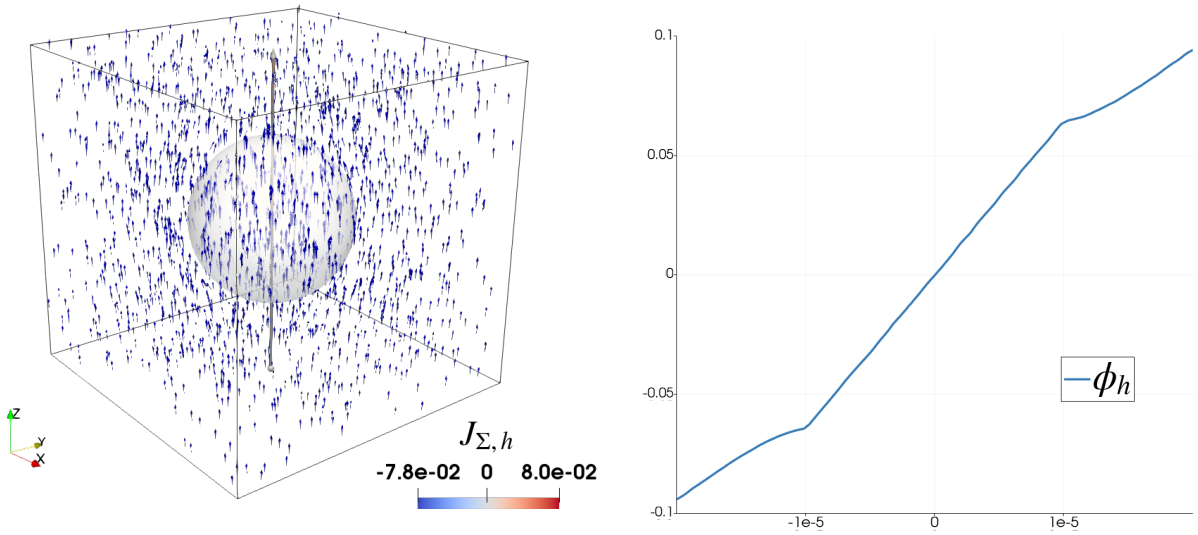
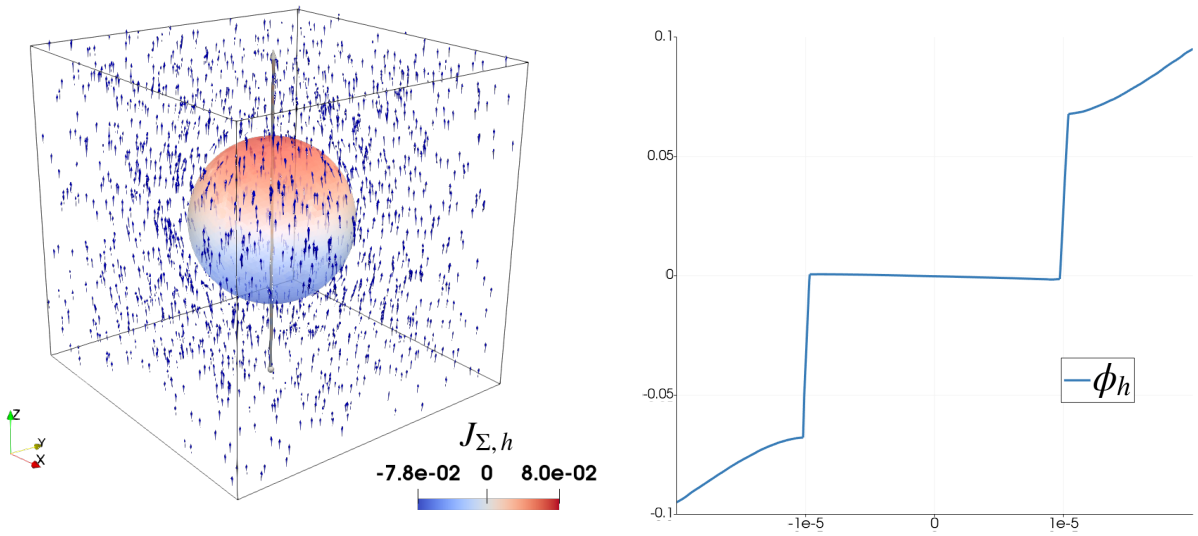


Figure 4.9: Geometry of the domain. Discrete setting with mesh sizes $4 * R/40, 0.15R$ (40 elements per side of the box, refinement level 3 of 4).



(a) $t = 0$



(b) $t \longrightarrow \infty$

Figure 4.10: Simulation with $\sigma_r = \frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 0.1$, comparison of initial and equilibrium state. Refinement level 3 out of 4. On the left, potential jump $J_{\Sigma,h}$ represented with a colormap on the spherical surface. Electric field $\nabla\phi_h$ represented by blue glyphs on the domain. On the right, potential ϕ_h along the symmetry axis $\hat{z}$ (grey rod on the left). At equilibrium, the internal electric field $\nabla\phi_{h,\text{int}}$ vanishes.

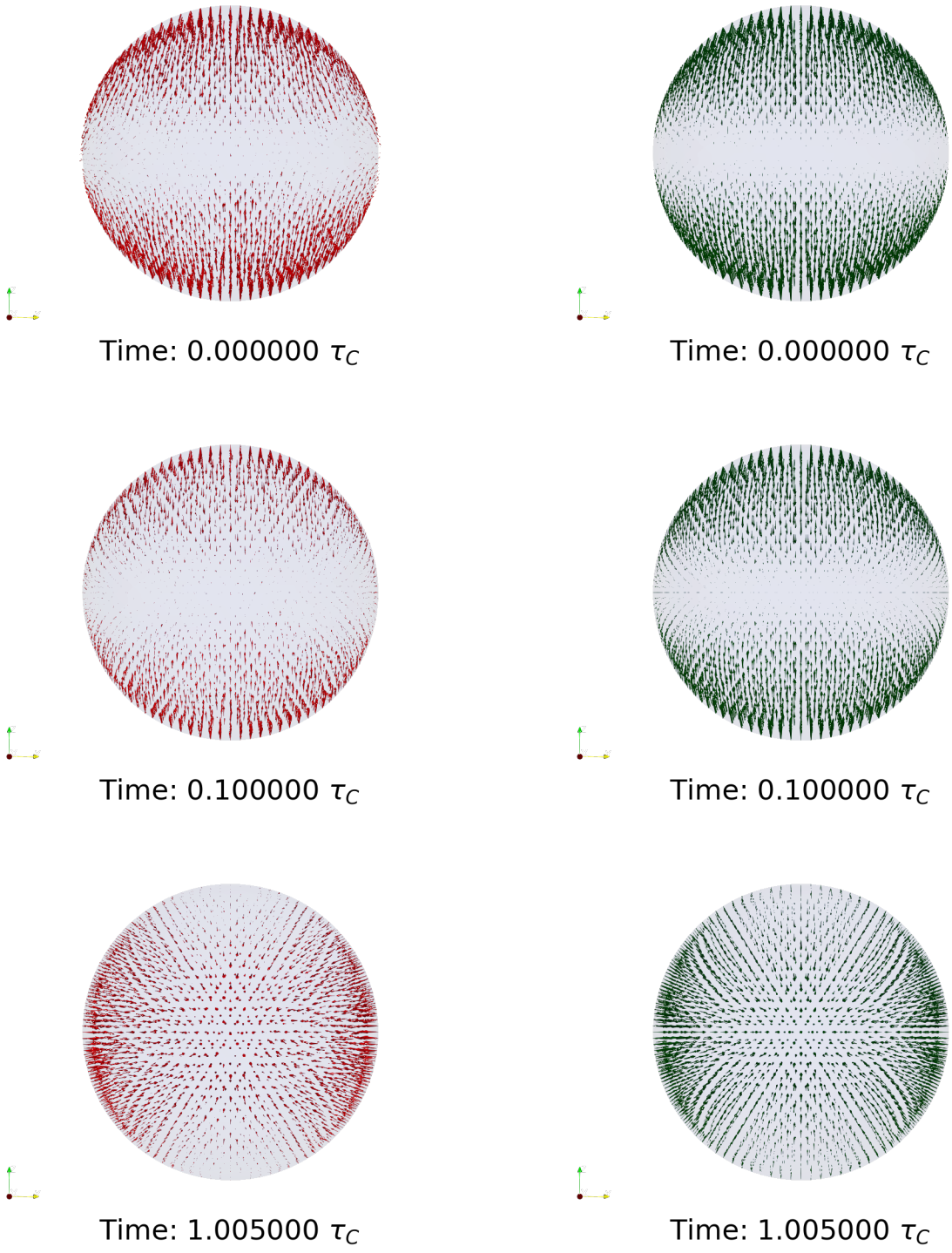
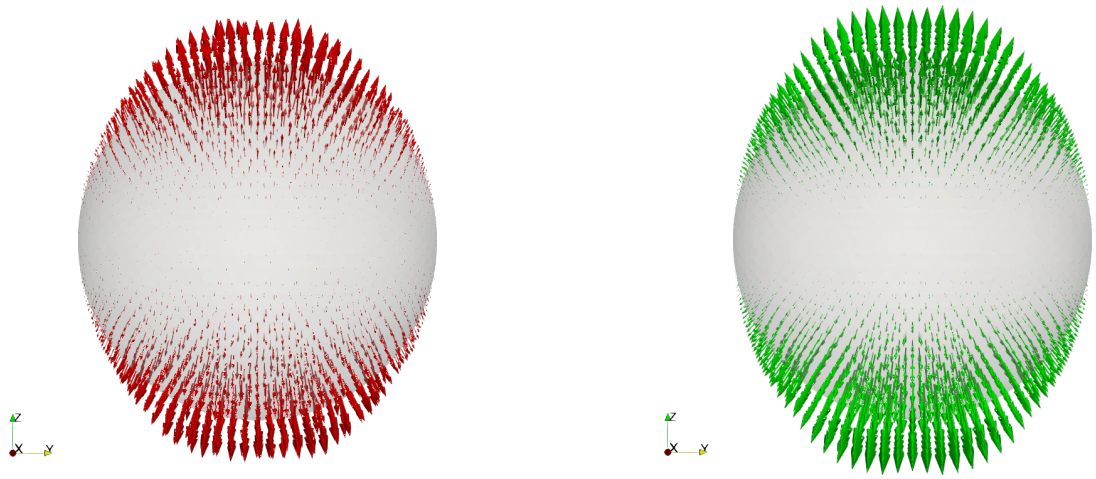


Figure 4.11: On the left, profile of estimated electric force $\boldsymbol{F}_{MW,h}$ along the surface at several time iterations. On the right, comparison with exact electric force $\hat{\boldsymbol{F}}_{MW}$. $\sigma_r = 0.1$. Discretization level 4 out of 4.



(a) $\boldsymbol{F}_{MW,h}$ (b) $\hat{\boldsymbol{F}}_{MW}$

Figure 4.12: On the left, profile of estimated electric force $\boldsymbol{F}_{MW,h}$ along the surface at initial iteration $t = 0$. On the right, comparison with exact electric force $\hat{\boldsymbol{F}}_{MW}$. $\sigma_r = 10$. Discretization level 4 out of 4.

4.2 Transition of a capsule at rest under DC field

In this section, we consider the shape evolution of a spherical capsule subject to a uniform electric field. This prototype benchmark is one of the most simple in which the electricity computations (module `ESS`) interact with the fluid-structure interaction. On the other hand, although simple, this configuration features many key properties of interest when considering the electrohydrodynamics of membrane physics. This physical system presents a complex behaviour, depending on factors such as the conductivity ratio and the intensity of the electric field. In view of the vast range of dynamics that a capsule can undergo, this system is one of the most studied in theoretical and computational electrohydrodynamics. For an exhaustive review on the subject, the reader is referred to [Vlahovska 2019b]. In subsection 4.2.1, the physical phenomenon is introduced along with a discussion on the main dimensionless parameters influencing the dynamics of the capsule and the expected behaviour of the capsule. The results of the simulations obtained with `YALES2BIO` are detailed in subsection 4.2.2 for the case $d = 2$ and 4.2.3 for the case $d = 3$.

4.2.1 Physical Model

Let $d \in \{2, 3\}$ be the dimension of the system. Consider a spherical interface with radius R, separating an internal fluid core from an external fluid environment. Let ρ and μ be the density and viscosity of both phases and $\sigma_\bullet$,$\epsilon_\bullet$ with $\bullet \in \{\text{int}, \text{ext}\}$ represent respectively the electric conductivity and permittivity of the internal and external phase. The constitutive relation of the interface can be described introducing a mechanical model of the interface. The specific formulation of the model depends on the dimension d, therefore it is postponed to the next subsections. In general, the consitutive relation of the membrane is determined by two parameters, the elastic modulus k_{str} [N m^{-1}] and the bending modulus k_{bnd} [J m^{3-d}]. In the literature regarding cellular membranes and affine systems, capsules are characterized by weak bending rigidity with respect to elastic rigidity. The electric behaviour of the membrane is described by the model introduced in section 2.4. The membrane supports interfacial

electric charge, and is characterized by a capacitance C.

The flow is described by hydrodynamics variables pressure p and velocity $\boldsymbol{u}$, respecting Navier-Stokes Equations 3.1. The fluid-structure interaction is modeled in the fashion of the Immersed Boundary Model as detailed in 2.3. The electric field inside and around the cell is described in terms of the electrostatic potential ϕ, subject the set of conditions 3.22. The state of the membranes in terms of electric charge is described by the interface potential jump J_Σ, subject to the evolution equation 3.23.

The system evolves from an initially static condition, with the capsule resting in its equilibrium spherical shape. At the beginning of the observation, $J_\Sigma = 0$. Far from the capsule membrane, the electric field is constant, with intensity E.

The behaviour of the system is determined by the contrasts:

$$\sigma_r = \frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}}, \qquad \epsilon_r = \frac{\epsilon_{\text{int}}}{\epsilon_{\text{ext}}}, \qquad \mu_r = \frac{\mu_{\text{int}}}{\mu_{\text{ext}}}.$$

The global dynamics is dictated by a number of different timescales. Considering the characteristic electric traction $t_{el} = \epsilon E^2$, the characteristic time of the electrohydrodynamic flow is determined as:

$$\tau_{el} = \frac{\mu_{\text{ext}}}{\tau_{el}} = \frac{\mu}{\epsilon E^2}.$$

The charge relaxation times:

$$\tau_{cr,\text{int}} = \frac{\epsilon_{\text{int}}}{\sigma_{\text{int}}}, \qquad \tau_{cr,\text{ext}} = \frac{\epsilon_{\text{ext}}}{\sigma_{\text{ext}}},$$

describe the characterstic times at which a distribution of electric charge decays in the respective phase [Taylor 1966]. The elastic relaxation time over which the capsule restores its rest shape following a perturbation is:

$$\tau_{str} = \frac{\mu_{\text{ext}} R^{d-2}}{k_{str}}.$$

The bending relaxation time reads:

$$\tau_{bnd} = \frac{\mu_{\text{ext}} R^3}{k_{bnd}}.$$

The membrane charging process happens on a timescale:

$$\tau_C = CR\frac{1}{\sigma_{\text{int}}} + \frac{1}{(d-1)\sigma_{\text{ext}}},$$

as described in Section 4.1.

The dynamics of the system is determined by the ratios of the various timescales. In particular, the main dimensionless numbers used to describe the electrohydrodynamics of capsules are:

$$\frac{\sigma_r}{\epsilon_r} = \frac{t_{cr,\text{ext}}}{t_{cr,\text{int}}}, \qquad Ca_{str} = \frac{t_{str}}{t_{el}} = \frac{\epsilon E^2 R^{d-2}}{k_{str}}, \tag{4.5}$$

$$Ca_{bnd} = \frac{t_{bnd}}{t_{el}} = \frac{\epsilon E^2 R^3}{k_{bnd}}, \qquad Re_{el} = \frac{t_{cr,\text{ext}}}{t_{el}} = \frac{\epsilon E^2}{\sigma_{\text{ext}}\mu_{\text{ext}}}. \tag{4.6}$$

The stretching and bending capillary numbers Ca_{str} and Ca_{bnd} fournish a measure of the rigidity of the membrane with respect to deformations induced by the electric field. Dealing with capsules,

the mechanical response of the membrane is essentially driven by elasticity, therefore the regime $Ca_{bnd} \gg Ca_{str}$ is typically considered. Under this assumption, a global capillary number Ca is considered such that $Ca = Ca_{str}$. High values of Ca correspond to higher deformations of the capsule. The parameter σ_r/ϵ_r, which reduces to σ_r in the common scenario where $\epsilon_{\text{int}} = \epsilon_{\text{ext}}$, determines the category of transitory dynamics. The electric Reynolds number Re_{el} is a measure of the rapidity of the initial electrohydrodynamic flow with respect to the reference timescale $t_{cr,\text{ext}}$.

The expected behaviour of a capsule undergoing a uniform electric field is schematically represented in Figure 4.13. When the electric field is switched on, a capsule polarizes in measure of the mismatch of conductivity σ_r and permittivity ϵ_r of the two fluid phases. Electric charge migrates towards the interface. Charge of opposite sign builds up along the two sides of the membrane, which stays overall neutral. If $\sigma_r/\epsilon_r < 1$, charge relaxation inside the capsule is slower. Therefore the interface charge distribution is dominated by charges brought from the external medium. The dipole moment of the capsule is oriented opposite to the applied electric field direction. In this case electric stress drives a motion from the pole to the equator towards an oblate shape. The intensity of the deformation is driven by the capillary number Ca. In the limit $Ca \ll 1$, the capsule stays quasi-spherical. On the other hand, when $\sigma_r/\epsilon_r > 1$, the capsule momentum is aligned with the electric field, and the initial deformation drives the membrane into a prolate shape (see Figure 4.13). After transitory, the membrane is fully charged, overall neutral, and the electric field inside the capsule is screened out by the transmembrane potential. Independently of the transient polarization, the capsule assumes a final prolate shape.

The state of deformation of the membrane is examined in terms of Taylor's deformation parameter:

$$\lambda_T = \frac{L - B}{L + B},$$

where L and B represent respectively the major and minor axis of the inertia ellipsoid.

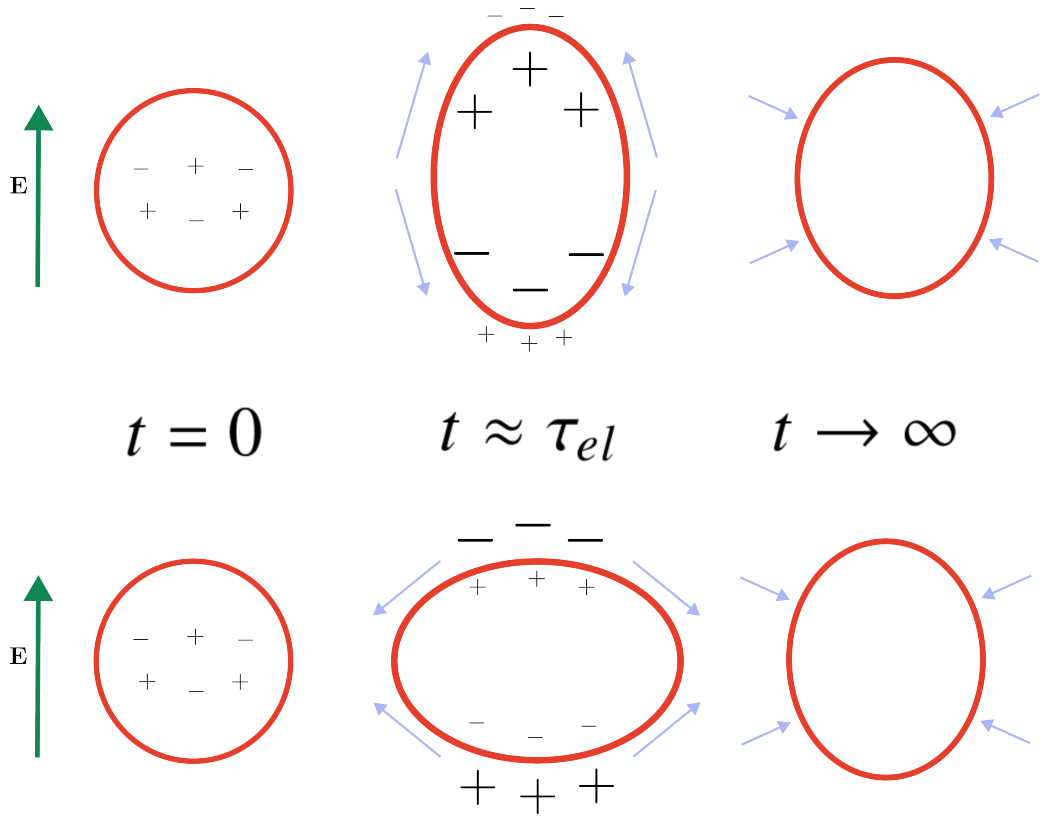


Figure 4.13: Moderate deformation of a spherical capsule according in different regimes of σ_r/ϵ_r. From left to right, a scheme of the three main phases of the transition of the capsule. The green arrow represents the direction of the electric field $\boldsymbol{E}$. When $\sigma_r/\epsilon_r > 1$, the capsule membrane is polarized in the direction of the electric field $\boldsymbol{E}$, and is brought to assume the shape of a prolate ellipsoid (center top). When $\sigma_r/\epsilon_r < 1$, the membrane is polarized in the opposite direction, and is deformed into an oblate ellipsoid (center bottom). The equilibrium shape of the capsule is independent of σ_r/ϵ_r.

Reference physical parameters used to realize the simulations are resumed in Table 4.3. The parameters are then tuned in the tests so tho reproduce specific regimes in terms of dimensionless numbers. The choice of the parameters is modeled upon reference values that can be found in literature for capsule dynamics, see for example [Das and Thaokar 2018].

4.2.2 Case $d = 2$

The 2-dimensional mechanical model of the cell is defined as described in Subsection 2.2.3. Elastic tension along the surface is expressed as:

$$t_{str} = k_{str} \frac{dl - dl_0}{dl_0}.$$

Calling Γ the surface of the membrane, bending energy is modeled as:

$$\mathcal{E}_{bnd} = \frac{k_{bnd}}{2} \int_\Gamma r_\Gamma^{-2} dl,$$

where r_Γ^{-2} is the curvature radius of Γ.

The set of tests is performed considering the numerical setting described in Figure 4.14. The simulation is performed with a control volume with side $2R$. Inlet-type conditions with vanishing

Parameter	Value	Dimension
R	$1 \cdot 10^{-5}$	m
k_{str}	$7.5 \cdot 10^{-6}$	N m^{-1}
k_{bnd}	$7.5\,10^{-19}$	J
ρ	$1 \cdot 10^{3}$	Kg m^{-3}
μ	$7.5 \cdot 10^{-4}$	Kg s^{-1} m^{-1}
E_∞	$1 \cdot 10^{4}$	Volt m
ϵ	$1 \cdot 10^{-10}$	N Volt^{-1} m^{-1}
σ_{ext}	$1 \cdot 10^{-4}$	Sm m^{-1}
σ_{int}	$\sigma_r \sigma_{\text{ext}}$	Sm m^{-1}
C	$5 \cdot 10^{-4}$	Farad m^{-2}
τ_{el}	$7.5 \cdot 10^{2}$	s
τ_{str}	$1 \cdot 10^{-3}$	s
τ_{bnd}	$9.9 \cdot 10^{-1}$	s
$\tau_{cr,\text{ext}}$	$1 \cdot 10^{-3}$	s
τ_C	$6 \cdot 10^{-1}$	s
Ca_{bnd}	$1.33 \cdot 10^{1}$	–
Ca_{str}	$1.33 \cdot 10^{-2}$	–
RE_{el}	$1.33 \cdot 10^{-2}$	–

Table 4.3: Physical parameters of the test with $Ca = Ca_{str} << 1$, $d = 2$.

velocity are assigned along the boundaries orthogonal to the electric field, outlet-type conditions are assigned along boundaries parallel to the field. For the calculation of electrostatic potential, Dirichlet-type conditions are assigned on boundaries orthogonal to the field, so to reproduce the condition $\nabla\phi(\|\boldsymbol{x}\|) \approx \boldsymbol{E}$, $\|\boldsymbol{x}\| \to \infty$. Neumann-type conditions are assumed on the remaining part of the boundary.

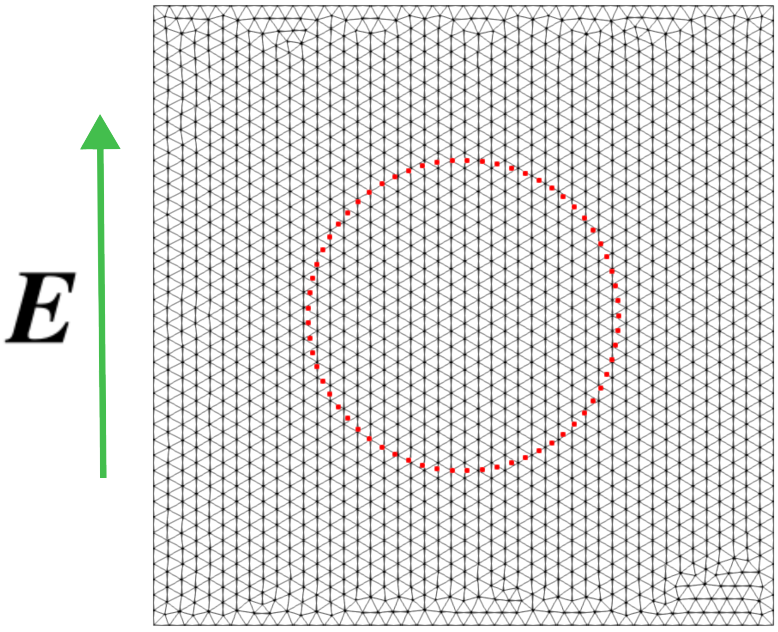


Figure 4.14: Numerical setting employed for the tests described in subsection 4.2.2. The side of the square control volume is twice the radius of the capsule, with 40 elements per side. The reference system is oriented in the direction of the electric field.

A preliminary set of tests (T1 for reference with figures) is conducted in the regime of moderate deformations. The computations are repeated varying the conductivity ratio σ_r with the goal of retrieving the expected transitions presented in Figure 4.13. The transition of the capsule highlighting the distribution of electric forces along the membrane is presented for $\sigma_r/\epsilon_r \in \{0.1, 1, 10\}$ respectively in Figures 4.15, 4.16, 4.17. Results in terms of the deformation parameter λ_T are displayed in Figure 4.18. The results match the expected dynamics in terms of type of deformation and timescales of the transition.

A second test (T2) is realised with a fixed value of Ca, considering $\sigma_r/\epsilon_r = 0.1$, corresponding to a temporary transition to an oblate shape. The simulation is repeated varying the value of Re_{el}. Deformation parameters are plotted in Figure 4.19. The results highlight how Re_{el} impacts the timescale over which the maximal extent of deformation is reached, but not the extent of the deformation itself, nor the time to reach the equilibrum shape.

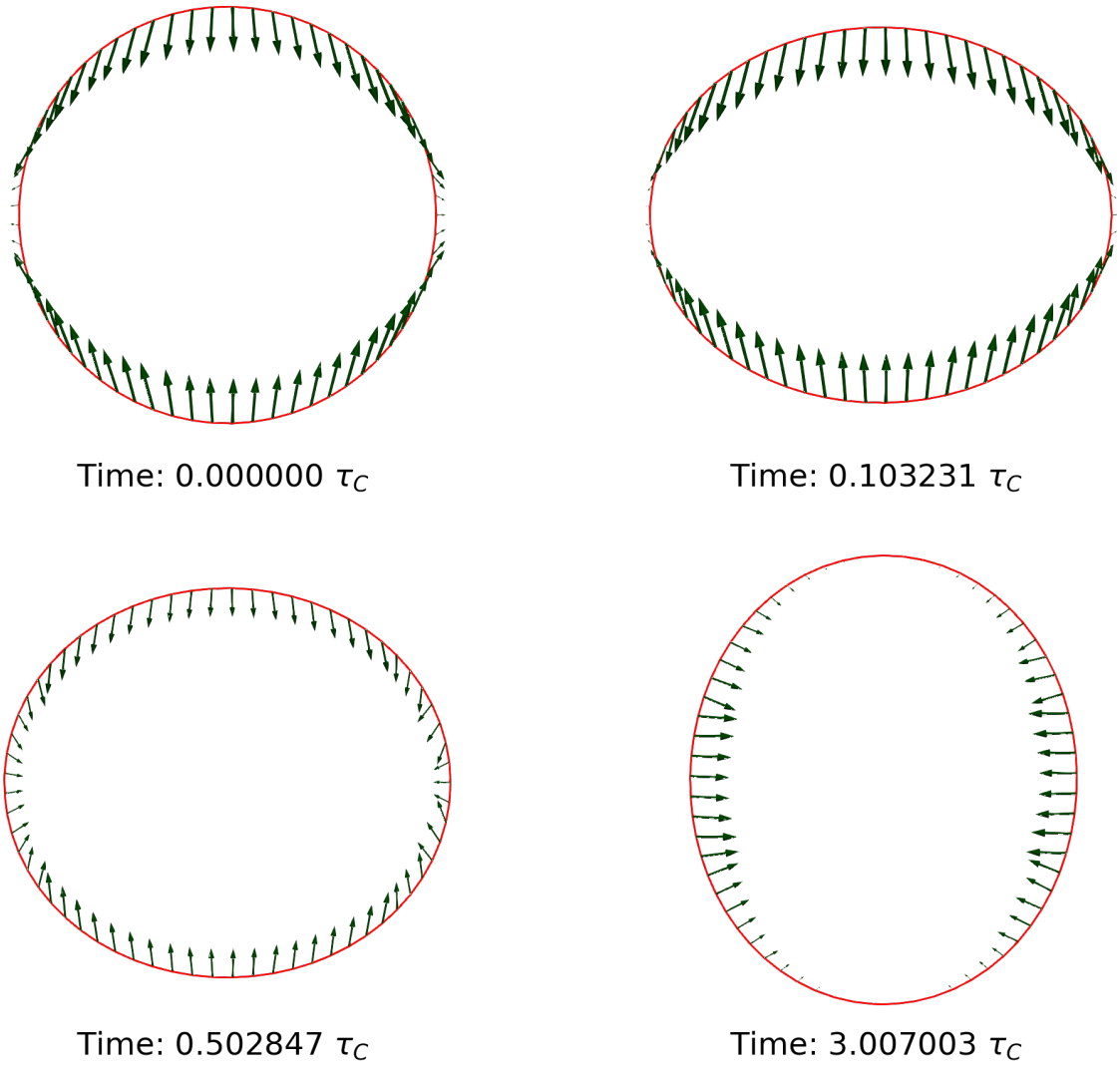


Figure 4.15: Test T1 from subsection 4.2.2. $\sigma_r/\epsilon_r = 0.1$. Oblate-Prolate transition of the capsule. Electric force in $\hat{y}$ direction.

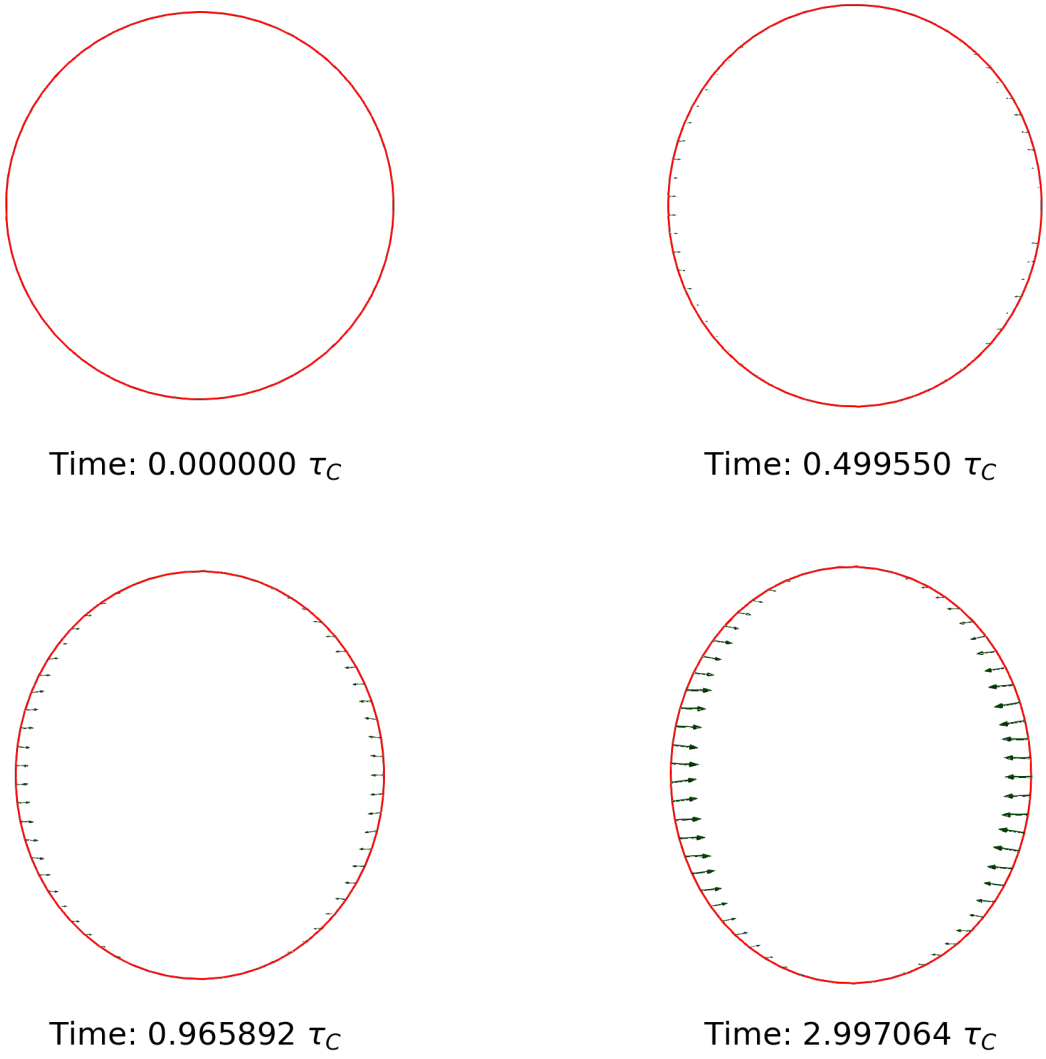


Figure 4.16: Test T1 from subsection 4.2.2. $\sigma_r/\epsilon_r = 1$. Electric force in $\hat{y}$ direction.

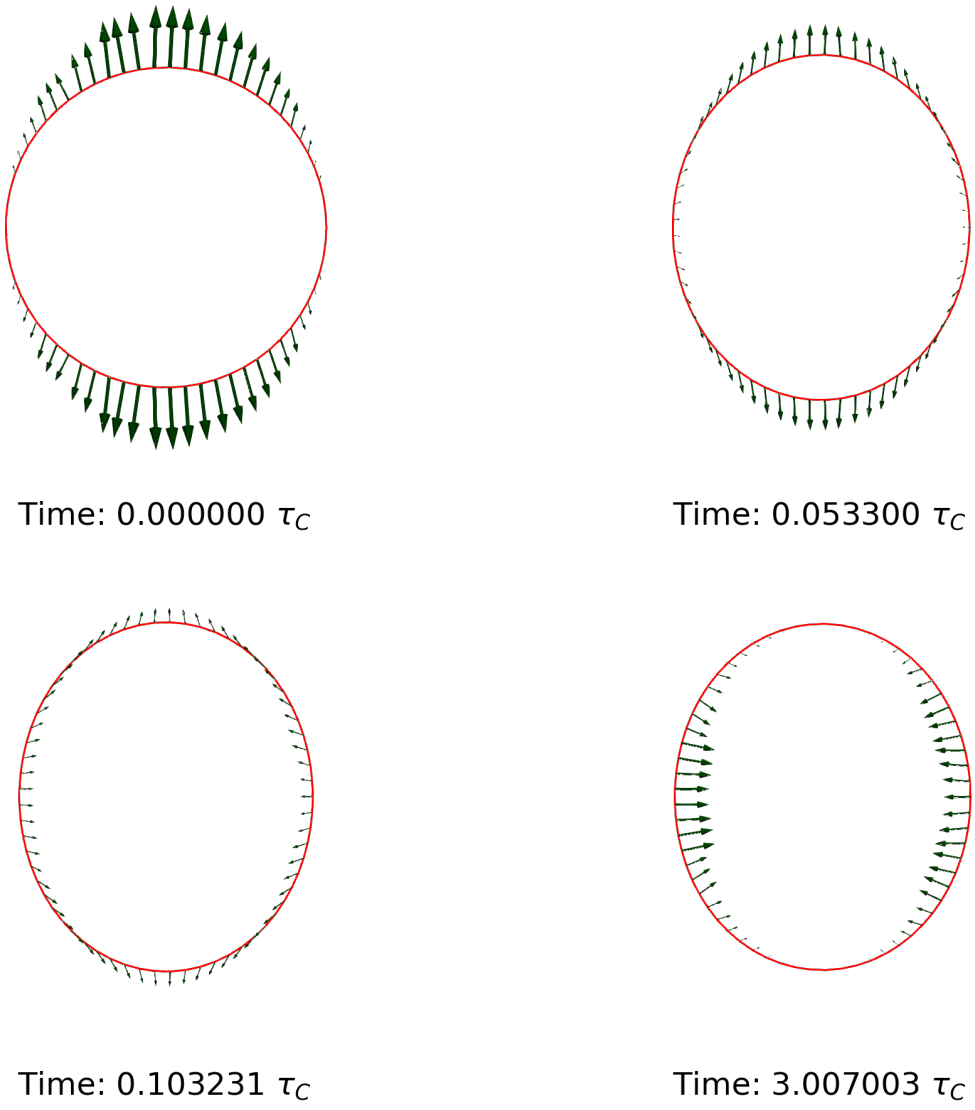


Figure 4.17: Test T1 from subsection 4.2.2. $\sigma_r/\epsilon_r = 1$. Prolate-Prolate transition of the capsule. Electric force in $\hat{y}$ direction.

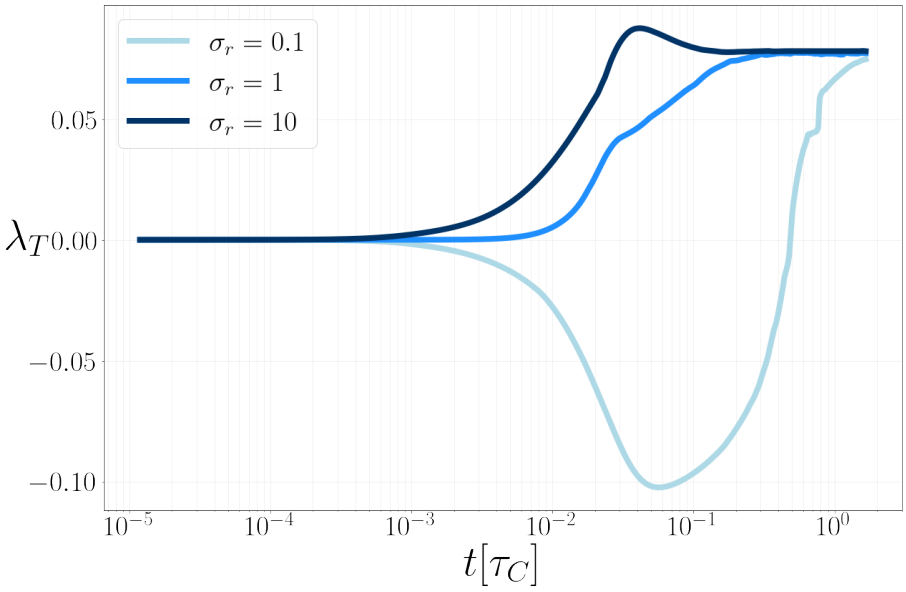


Figure 4.18: Test T1 from subsection 4.2.2. Taylor deformation parameter λ_T along time according to the conductivity ratio regime. Final deformation is independent of σ_r, and happens on a time scale $t_{ch} \approx \tau_C = O(0.1s)$. Independently of conductivity ratio, maximal deformation happens on a timescale $t \approx \tau_{el} = O(10^{-2}s)$.

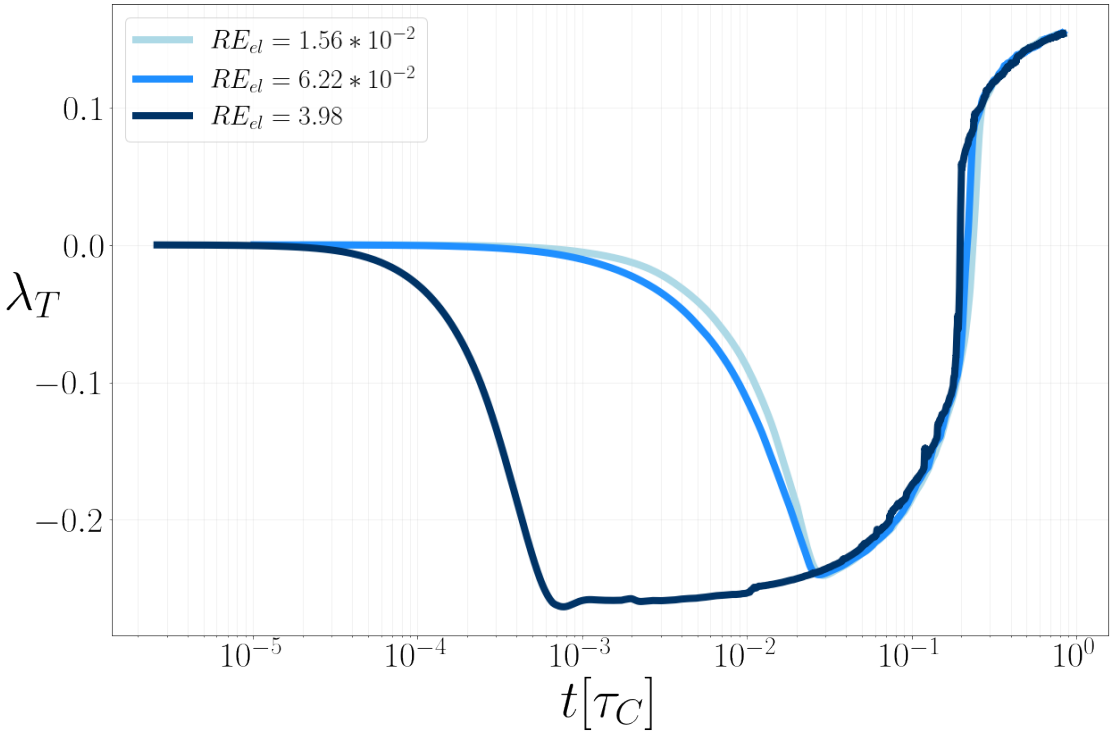


Figure 4.19: Test T2 from subsection 4.2.2. Taylor deformation parameter λ_T for different regimes of RE_{el}. $\sigma_r/\epsilon_r = 0.1$. $Ca = 1.56 \cdot 10^{-2}$. For high values of Re_{el}, the rapid transition of the capsule generates instabilities of the flow that perturb the motion of the capsule, although without symmetry breaking.

4.2.3 Case $d = 3$

With $d = 3$, the elastic behaviour of the membrane is described by Skalak's strain energy density 2.1, and the response to bending by Helfrich's energy density 2.3. A test is set up to compare the simulations obtained with `YALES2BIO` with the numerical results presented in [Das and Thaokar 2018]. The simulations are performed taking as control volume a cubic box with side $4R$. The mesh employed for the test are shown in Figure 4.9. The test consists in simulating the electrodeformation of the capsule varying the conductivity ratio $\sigma_r \in \{0.1, 1, 10\}$, in the regime $Ca = 0.25$. A visualization of the comparison in terms of Taylor deformation parameter λ_T is provided in Figure 4.20. The deformation of the caspule in the regime $\sigma_r = 0.1$ is displayed in Figure 4.21.

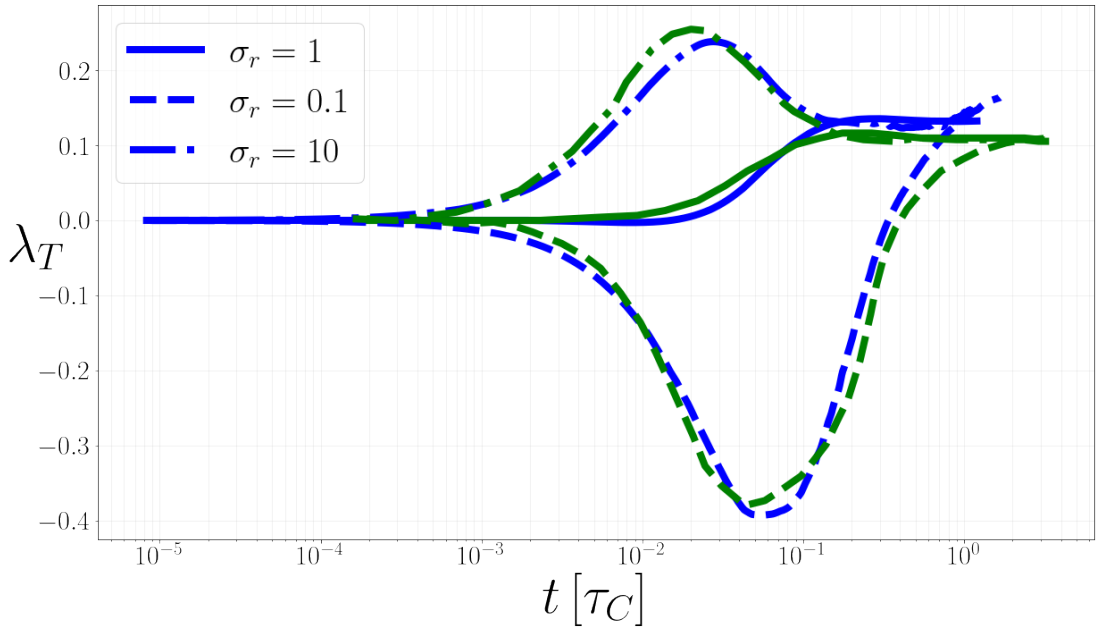


Figure 4.20: Comparison of simulations obtained with YALES2BIO and simulations from [Das and Thaokar 2018]. Deformation parameter λ_T plotted over time for different regimes of conductivity ratio σ_r. $Ca = 0.25$. Green lines correspond to results obtained by [Das and Thaokar 2018], blue plots to simulation produced by YALES2BIO. The two simulations show good agreement in terms of timescale of the phenomenon and maximal deformation. With YALES2BIO, the deformation at equilibrium is overestimated with respect to the reference.

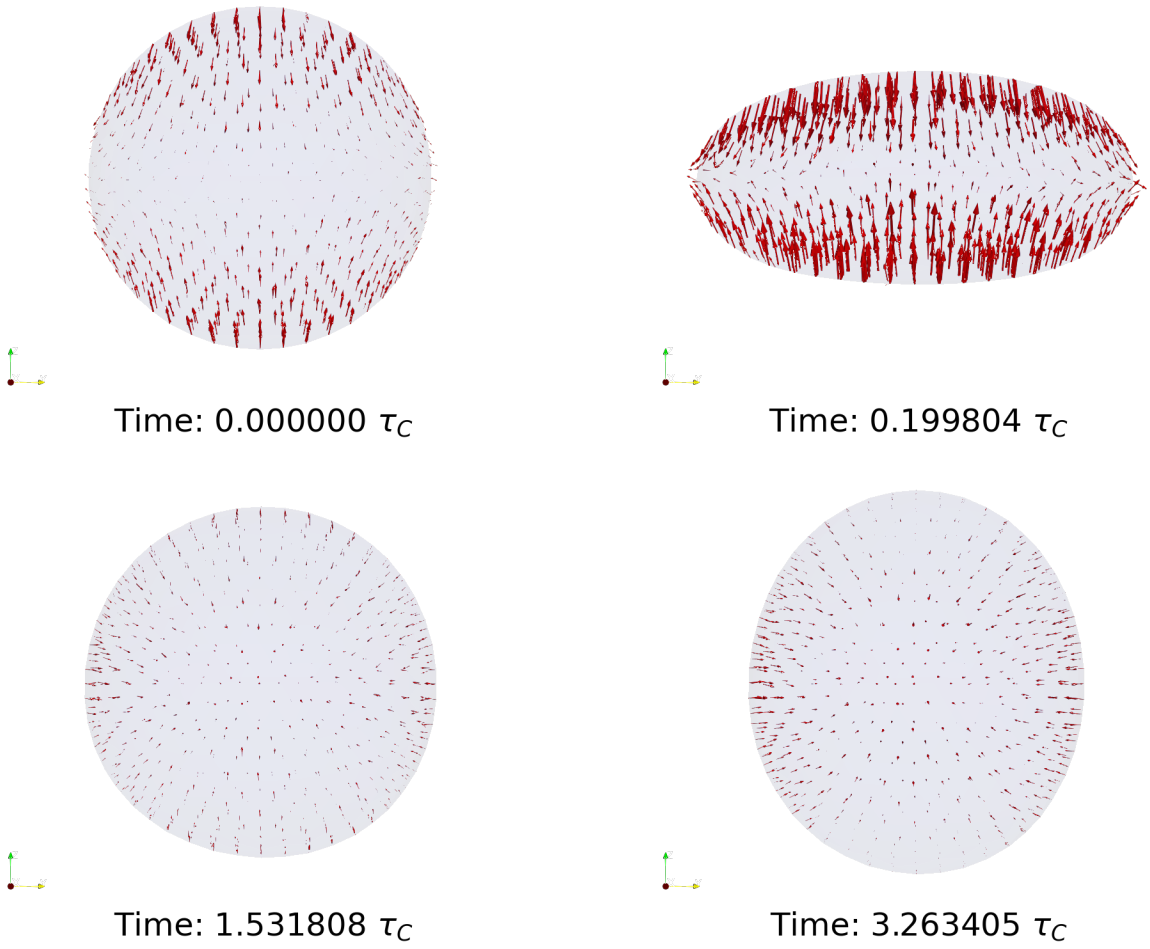


Figure 4.21: Deformation of a capsule with conductivity ratio $\sigma_r = 0.1$, $d = 3$. Maxwell stresses are displayed as red glyphs.

4.3 Shear Flow of a vesicle subject to a DC field

The last test focuses on the dynamics of a vesicle in shear flow. A vesicle is a drop of fluid bounded by an inextensible membrane with a moderate response to bending. Along with capsules, vesicles have been adopted as a model to represent erythrocytes [Vlahovska, Podgorski, and Misbah 2009b]. For this reason the trajectory of vesicles in various flow configurations has been extensively analysed [Vlahovska 2019a]. Among other factors impacting the dynamics of a vesicle, the effect of electric fields has been investigated experimentally and theoretically ([McConnell, Miksis, and Vlahovska 2013], [Vlahovska 2019a]). A benchmark is set up in `YALES2BIO` with the aim of retrieving the predictions from the literature on this subject. In subsection 4.3.1, the physical system is presented along with a discussion of its relevant parameters. The results of the simulation realized with `YALES2BIO` are presented in subsection 4.3.2.

4.3.1 Physical Model

We focus on the case of dimension $d = 2$. Figure 4.22 represents the initial configuration of the physical system. A vesicle with internal area A_0 is bounded by a curve Γ of length L. The perimeter of a vesicle is assumed to be constant, therefore a parameter characterizing its geometry is the reduced area Δ:

$$\Delta = \frac{A}{\pi R^2}, \qquad R = \frac{L}{2\pi},$$

where R represents the radius of a circle with perimeter L. Δ can be interpreted as a measure of deflation of the vesicle. It is minimal and equal to 1 if the vesicle is circular and gets larger for strongly deflated vesicles.

The response to bending of the vesicle is modeled considering the bending energy:

$$\mathcal{E}_{bnd} = \frac{k_{bnd}}{2} \int_\Gamma r_\Gamma^{-2} dl,$$

where r_Γ^{-2} is the curvature radius of Γ and k_{bnd} the membrane bending modulus. Without further assumptions, the resting shape of the vesicle minimizing bending energy would be a circle. In order to enforce a deflated shape at rest, a penalization term of membrane energy is introduced:

$$\mathcal{E}_A = \frac{k_A}{2} \int_\Gamma \frac{(A - A_0)^2}{A_0} dl,$$

where A is the current area of the vesicle and k_A is an area modulus which can be set large enough to impose the reference area.

In most works membrane tension is modeled introducing a Lagrange multiplier enforcing the inextensibility constraint [McConnell, Miksis, and Vlahovska 2013]. In `YALES2BIO`, the membrane is equipped with an elastic energy. With respect to a reference vesicle at rest with boundary Γ_0 and a parametrization $\boldsymbol{s}(l) : [0, L] \to \mathbb{R}^2$ of the current membrane position, elastic energy reads:

$$\mathcal{E}_{str} = \frac{k_{str}}{2} \int_{\Gamma_0} (\|\dot{\boldsymbol{s}}\| - 1)^2 dl,$$

where k_{str} is an elastic modules, which can be set large enough to enforce inextensibility.

The internal fluid core of the vesicle and external suspension are characterized by viscosities μ_{int} and μ_{ext} respectively. The vesicle, with elliptical shape at rest, is set into motion by a shear flow with shear rate γ. The system is subject to an electric field with intensity E_∞ in the direction normal to the flow direction. The two fluid phases have conductivities σ_{int} and σ_{ext} and permittivities ϵ_{int} and ϵ_{ext}. The electric response of the vesicle is modeled as exposed in section 2.4. In particular, the membrane is treated as an insulating interface with capacitance C.

In absence of an electric field, the most influential parameters for the dynamics of the vesicle are the reduced area Δ and the viscosity ratio is $\mu_r = \frac{\mu_{\text{int}}}{\mu_{\text{ext}}}$. In shear flows, largely deflated vesicles or vesicles with high viscosity contrast undergo a rigid-like rotational trajectory (*tumbling* dynamics). Moderately deflated vesicles or vesicles with weak viscosity constrast tend to assume a tilted position in the direction of the flow, with the vesicle membrane engaging in a revolutionary motion around the vesicle itself (*tank treading* dynamics) [McConnell, Miksis, and Vlahovska 2013].

The influence of electric forces is measured by introducing the capillary number:

$$Ca = \frac{t_{bnd}}{t_{el}} = \frac{\epsilon E^2 R^3}{k_{bnd}}$$

where the characteristic timescales of the phenomenon are:

$$t_{el} = \frac{\epsilon E^2}{\mu_{\text{ext}}},$$

associated to electric forces, and:

$$t_{bnd} = \frac{\mu_{\text{ext}} R^3}{k_{bnd}},$$

expressing the time over which the membrane recovers its resting configuration following a perturbation. Experimental investigation [Schwalbe, Vlahovska, and Miksis 2011] and numerical simulation [McConnell, Miksis, and Vlahovska 2013] have showed that forces generated by the electric field act against the shearing force, producing a damping effect on the tumbling motion of the vesicle. The damping is more evident the larger the value of Ca, and there exists a threshold over which tumbling is completely suppressed.

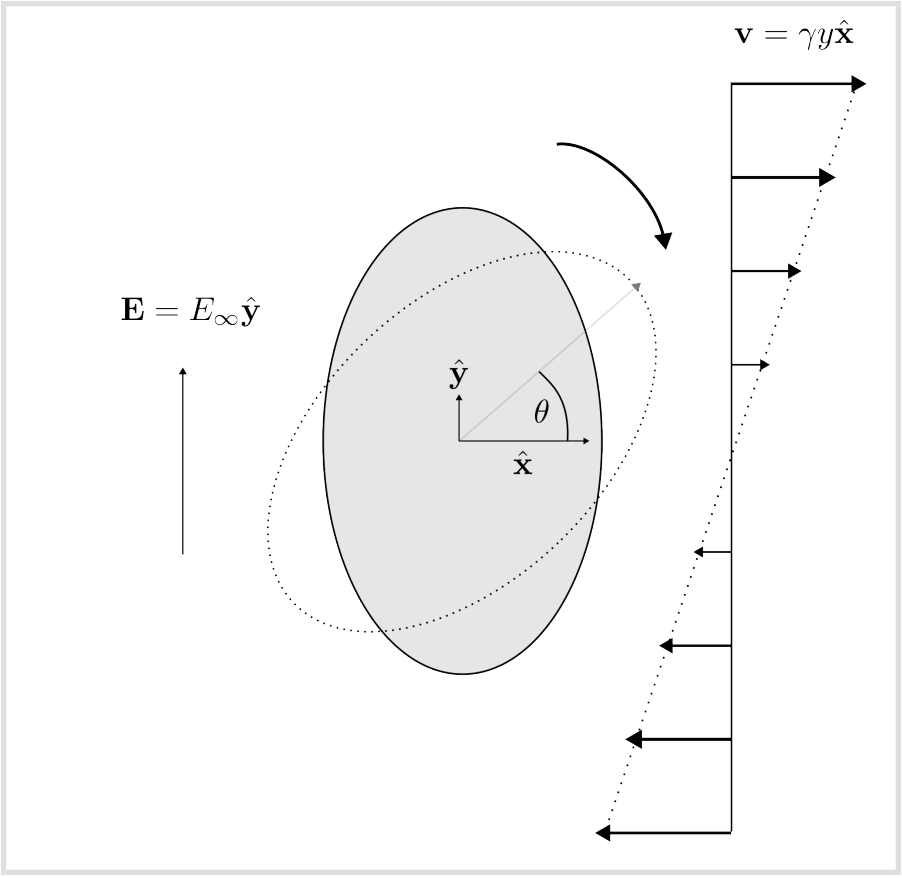


Figure 4.22: Initial configuration of the system. The vesicle is put in motion by a shear flow. The parameter adopted to describe the state of the vesicle is the tilting angle θ.

4.3.2 Simulation Results

With reference to the configuration simulated in [McConnell, Miksis, and Vlahovska 2013], the test is realised considering a vesicle with moderate reduced area ($\Lambda = 0.85$) and moderate viscosity ratio ($\mu_r = 10$). Table 4.4 resumes the settings of physical parameters for the test. Time is scaled with respect to the timescale of the shear flow: $\tau_\gamma = \gamma^{-1}$. The numerical setting is displayed in Figure 4.23

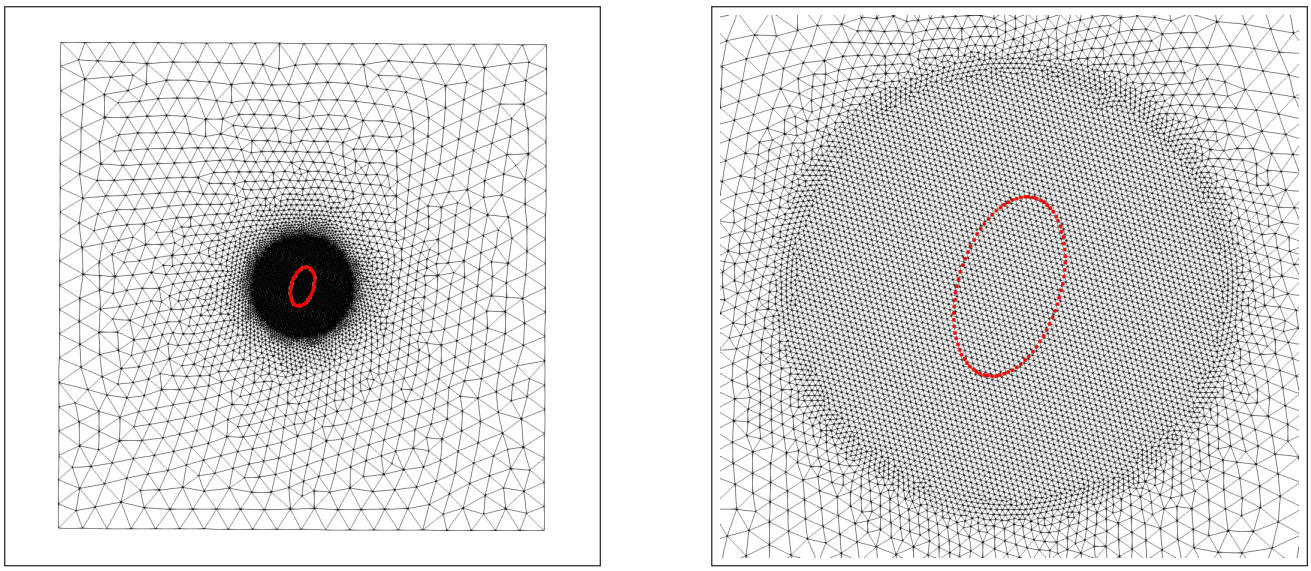


Figure 4.23: Numerical configuration of the test. The domain is a square box with side $30R$. The membrane Γ is discretized with a grid of 80 points. The refinement of the background is set so to respect the cryterion of 1 membrane point per background element, corresponding to a mesh element diameter $h = 7.1 \cdot 10^{-2}R$ in the central region.

First, the flow is studied in absence of electric field ($Ca = 0$). In such case, the vesicle is engaged in a rotational motion with a tumbling period $T \approx 20\tau_C$ (see Figure 4.24).

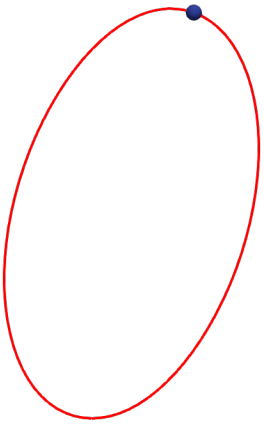
The same setting is tested considering the presence of an electric field. Figure 4.25 shows the dynamics of the vesicle for $Ca = 30$. The movement of the vesicle is still characterized by a periodic tumbling, but the tumpling period is increased to $t \approx 24\tau_C$. The test is also repeated on a refined mesh to assess mesh independence (Figure 4.26).

Vesicle dynamics is simulated for a range of values of Ca. The trend of the tilting angle is plotted against time for several capillary regimes in Figure 4.27. The tumbling period increases with Ca up to a threshold value ($Ca \approx 50$). Above the threshold, the tumbling motion is completely damped. The dynamics of the vesicle at $Ca = 50$ is displayed in Figure 4.29. Once the vesicle is horizontally tilted, the membrane engages in a tank treading motion around the vesicle itself, with period $t \approx 30\tau_C$. Figure 4.30 presents the trend of tilting angle along time corresponding to a series of values of Ca above the critical threshold. No direct relation is highlighted between Ca and the final tilting angle or the time to reach the final tilted configuration.

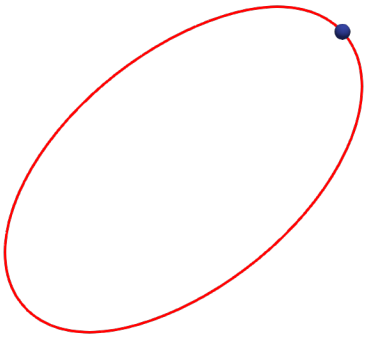
The results of the simulation realized with `YALES2BIO` are in substantial agreement with the numerical simulation presented in [McConnell, Miksis, and Vlahovska 2013], reported for reference in Figure 4.31. The damping effect of the electric field emerges in both cases, as well as the complete stop of the tumbling above a Ca threshold. However, the most important difference is in the value of the threshold, which is found one order of magnitude lower in our work.

Parameter	**Value**	**Dimension**
R	$1 \cdot 10^{-5}$	m
k_{str}	$>> 1$	J m^{-1}
k_{bnd}	$7.5 \cdot 10^{-19}$	J
ρ	$1 \cdot 5 \cdot 10^{-1}$	Kg m^{-2}
μ_{ext}	$5 \cdot 10^{-7}$	Kg s^{-1}
μ_{int}	$10\mu_{\text{ext}}$	Kg s^{-1}
E	variable	Volt m
ϵ	$1 \cdot 10^{-10}$	N Volt^{-1}
σ_{ext}	10^{-4}	Sm m^{-1}
σ_{int}	$0.1\sigma_{\text{ext}}$	Sm m^{-1}
C	$5 \cdot 10^{-1}$	Farad m^{-2}

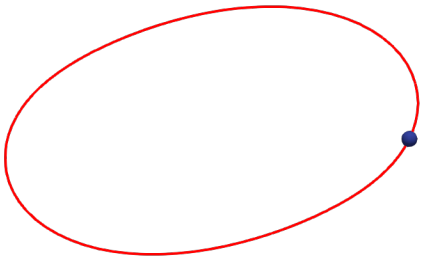
Table 4.4: Physical parameters of the test. The intensity of the electric field is varied to reproduce different regimes of Ca.



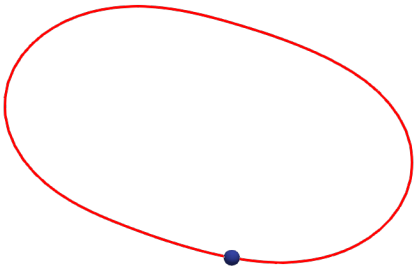
Time: 0.300050 τ_γ



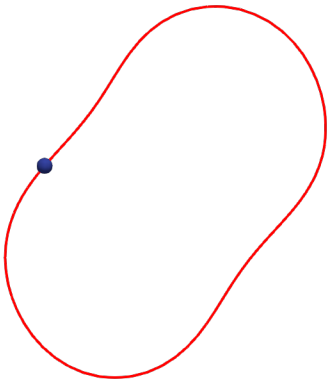
Time: 1.200201 τ_γ



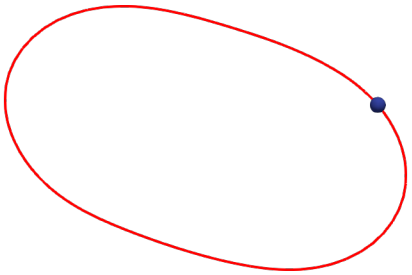
Time: 4.200344 τ_γ



Time: 10.200273 τ_γ



Time: 14.700309 τ_γ



Time: 24.000074 τ_γ

Figure 4.24: Trajectory of the vesicle in absence of electric field ($Ca = 0$). The trajectory of one single lagrangian marker on the membrane is highlighted by a spot.

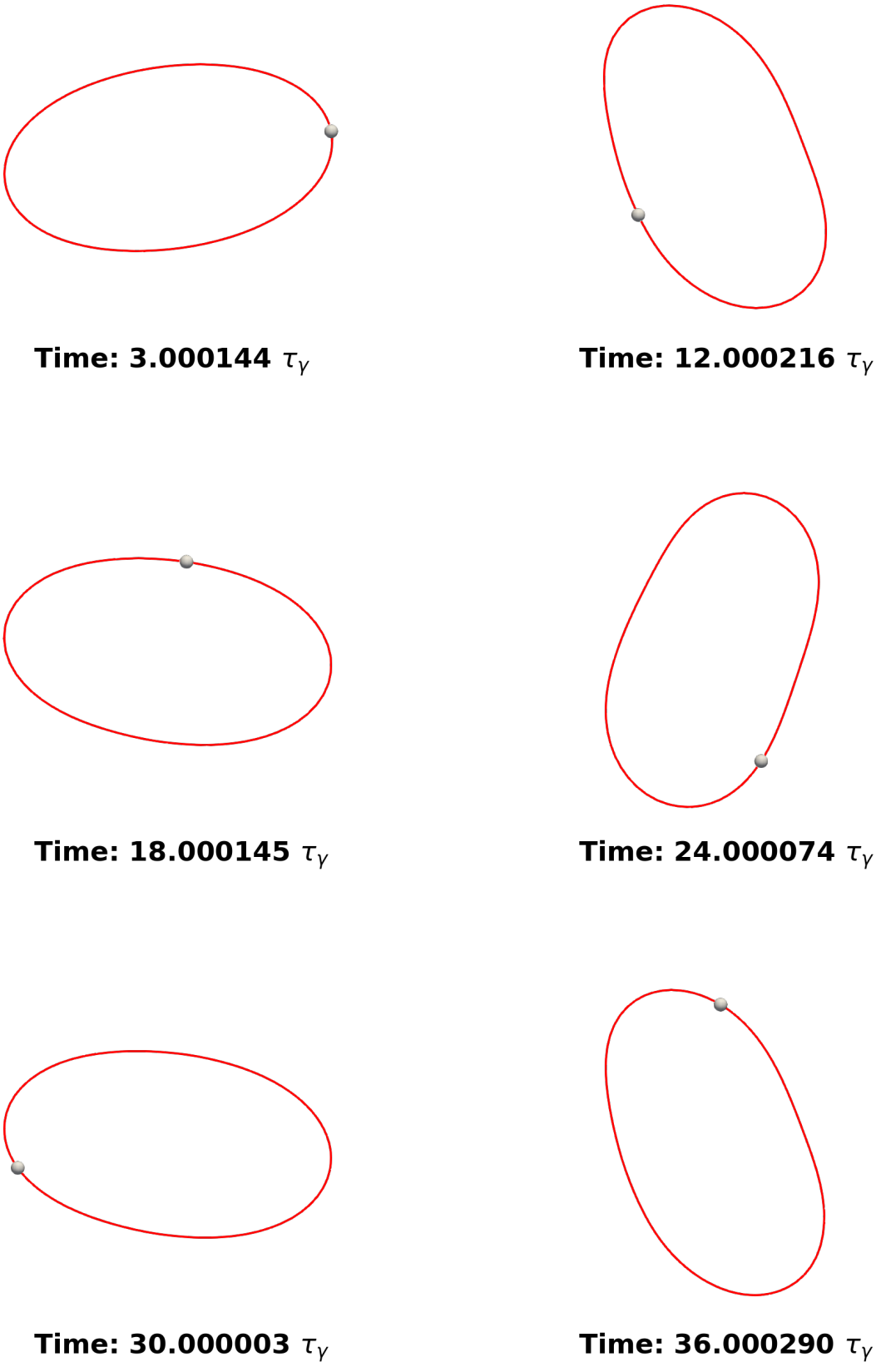


Figure 4.25: Trajectory of a vesicle subject to an electric field, $Ca = 30$.

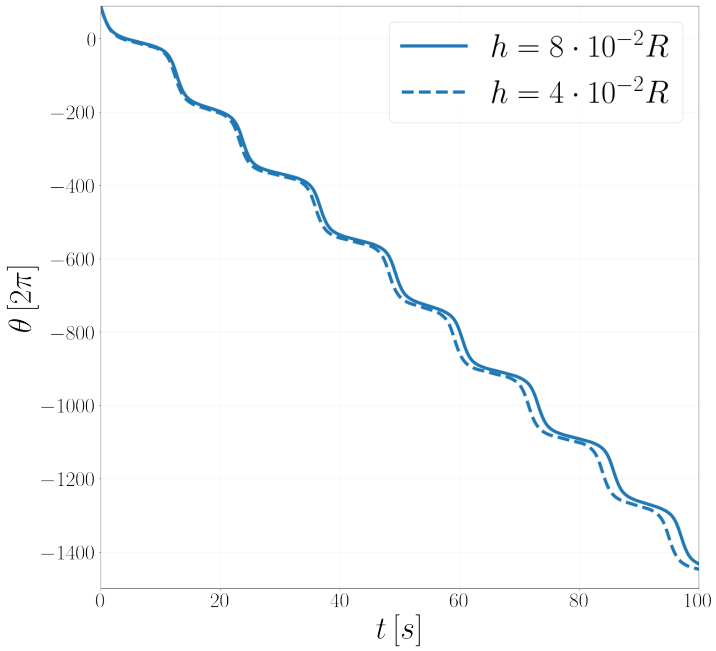


Figure 4.26: Mesh indepedence study. Tumbling angle θ (reference to diagram 4.22) along time for a vesicle in the regime $Ca = 30$. The simulation is repeated increasing the refinement level of the meshes.

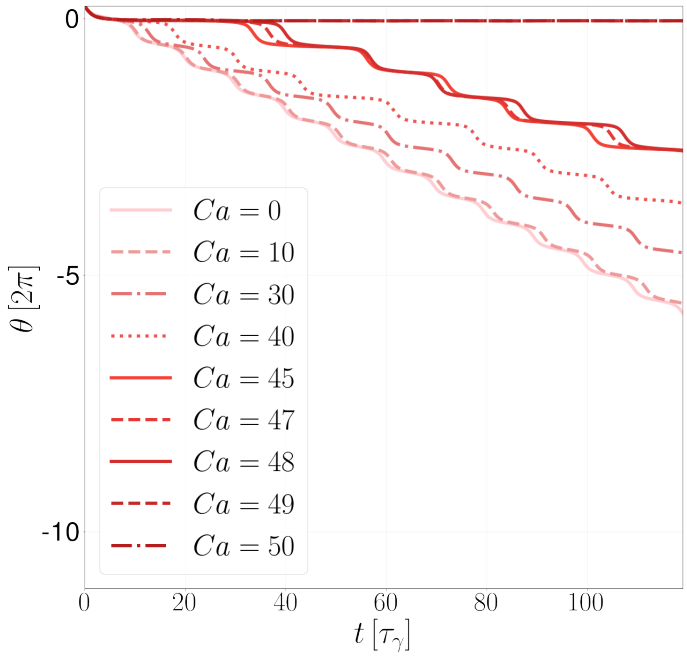


Figure 4.27: Tumbling angle θ along time at different levels of intensity of the electric field.

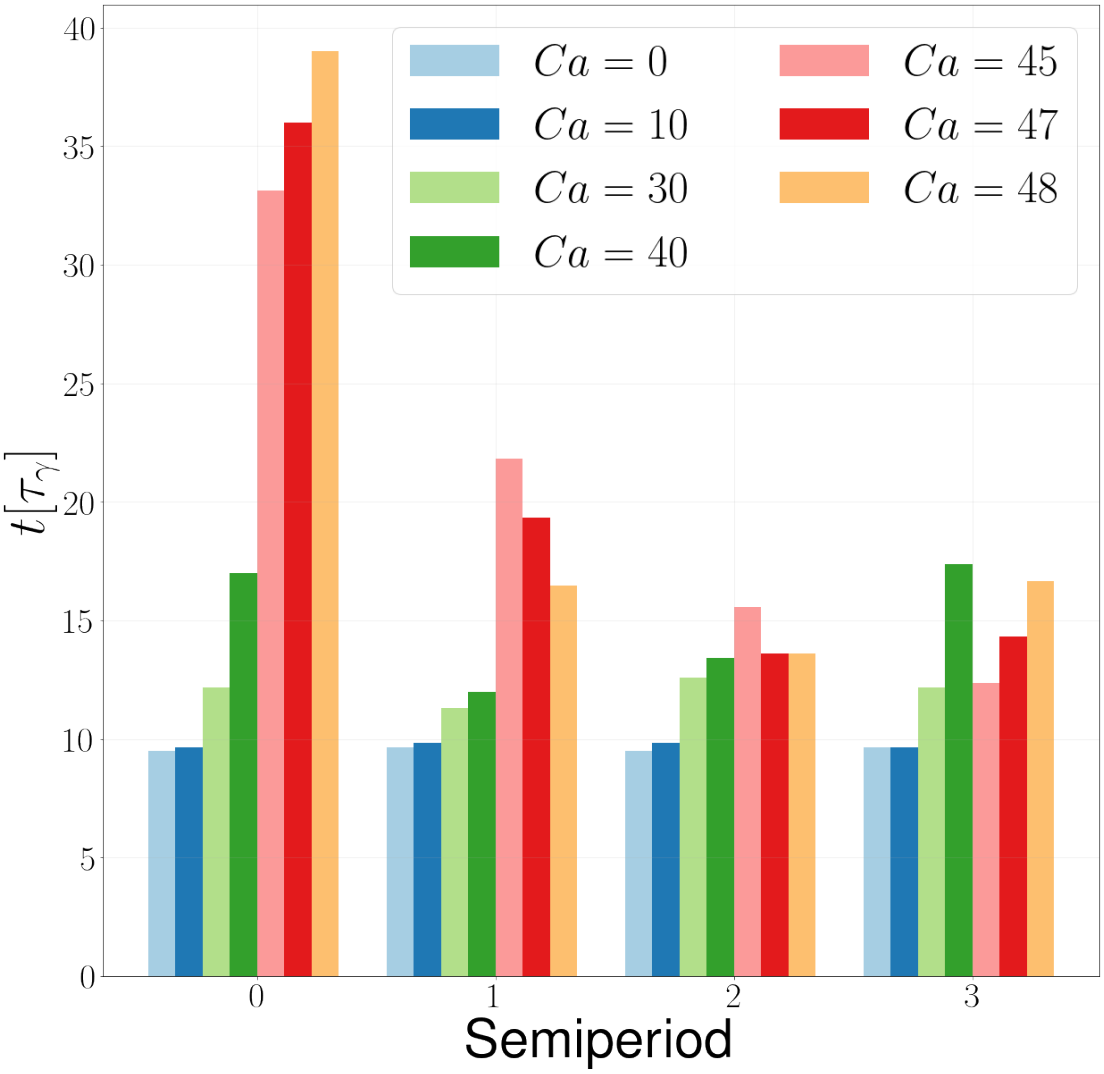
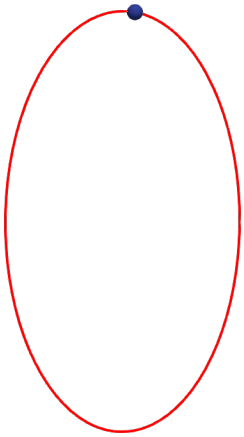
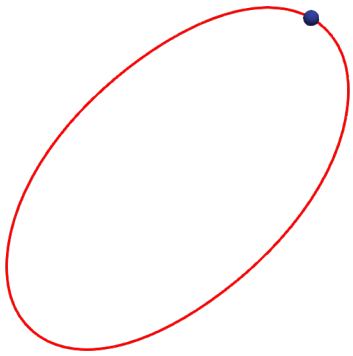


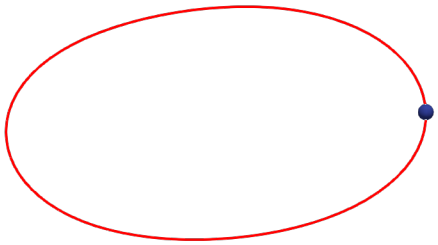
Figure 4.28: Duration of first 4 semiperiods for different regimes of Ca. Semiperiod is the time over which a vesicle realizes a flip of 180^o.



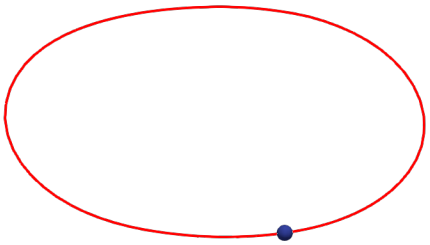
Time: 0.000000 τ_γ



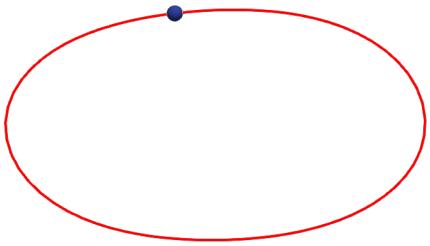
Time: 0.900150 τ_γ



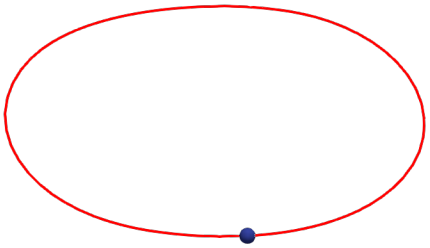
Time: 4.200344 τ_γ



Time: 10.800015 τ_γ



Time: 30.000003 τ_γ



Time: 48.000148 τ_γ

Figure 4.29: Trajectory of the vesicle at $Ca = 50$.

.

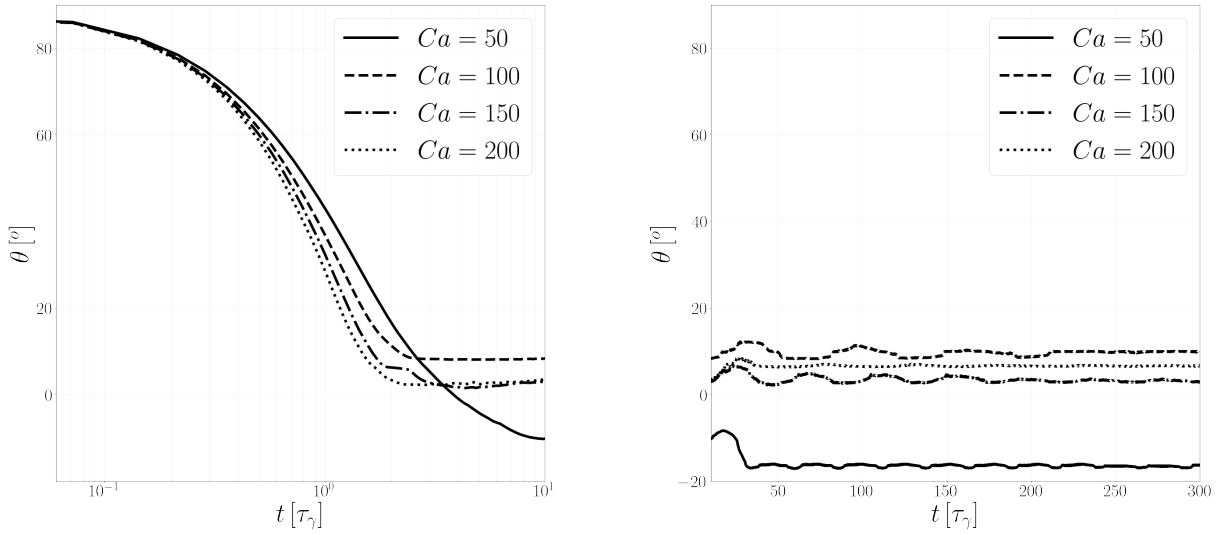


Figure 4.30: Tilting angle θ over time for values of Ca above the threshold. On the left, initial phase of the simulation. On the right, long-term behaviour.

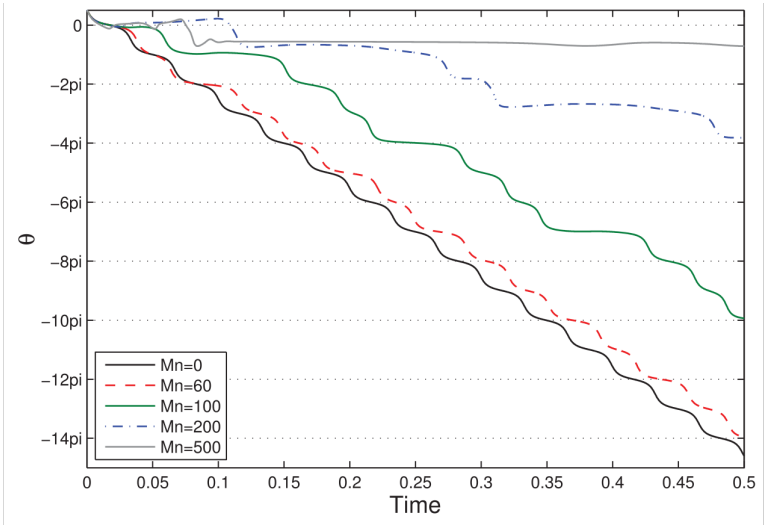


Figure 4.31: From [McConnell, Miksis, and Vlahovska 2013]. Tilting angle θ along time varying te intensity of the electric field. Time scaled with respect to τ_{bnd}. The capillary number Ca is there referred to as Mason number Mn.

4.4 Conclusion

To summarize, the numerical strategy of the module `ESS` of `YALES2BIO` is based on calculating the electric force $\boldsymbol{F}_{MW}$ as the result of a pipeline which starts with the estimation of the electric potential ϕ and continues with the estimation *a posteriori* of the electric field $\nabla\phi$. In some sense, the force is calculated as a post-processing of the potential solution obtained via the Ghost Fluid Method. The algorithm is simple to implement, moreover it is completely *unfitted*, making it suitable to be coupled with the Immersed Boundary Method.

Capsule, vesicles and cells are involved in complex physical processes, highly nonlinear, intrinsically multiscale and the trajectory can follow a rich variety of patterns depending on the considered regime. This make it difficult to isolate a benchmmark where the dynamics is computable *a priori*. A systematic error analysis is possible only for very simple configurations. It is the case for the benchmark presented in 4.1, where an analytical solution is available to realize a comparison.

Results of the error analysis are presented extensively in the Appendix chapter A. Error is traced in a normalised L_∞ and L_2 norm. The error is analysed both at $t = 0$ and along time. A focus on the initial error is considered to isolate the error introduced by the discretization algorithm for the calculation of electric force $\boldsymbol{f}_{MW}$ from the error introduced by the approximation in time used for the advancement of the potential jump J_Σ. In the case $d = 2$, error in L_2 metric lowers when increasing the refinement level of the meshes Ω_h and Σ_h, but the same is not confirmed for the error in L_∞ norm.

On the one hand, such a performance can be a hint on the fact the force calculation algorithm is highly sensitive to the quality of the intersection that the interface has with the backgrond domain, which is not controlled by the refinement level. This is true especially for postprocessed quantities (electric field $\nabla\phi$, electric forces $\boldsymbol{f}_{MW}$). Moreovoer, it is to remark that the interface is at the same time the region of the domain where the larger gradients are expected, and the region where the most valuable information is gathered.

It is necessary to remark that the approach to the calculation of electric force implemented `YALES2BIO` still struggles to produce simulations with $d = 3$ that are sufficiently robust. Despite some encouraging results (see for example Figure 4.20), the solver is still subject to numerical vagaries that discourage from use in complex scenarios. Although programmation errors cannot be completely excluded, the origin of the artifacts may be linked to an intrinsic limit of the interpolation strategy, which is more involved for $d = 3$.

The most probable obstacle preventing error to convergence with respect to mesh size are exceptions introduced by the relative position of the membrane with respect to the background surface. In particular, the method is affected by *blind spots*, snapshots in the simulation where the estimated value of the electric force on a point of the surface is completely erratic with respect to the behaviour of the expected solutions at neighbours. Blind spots appear to be higly dependent on the quality of the intersection between Ω_h and Σ_h, and are probably related to exception in the recognition of the interface position. Often, their lifespan is of 1 iteration. The appearance of blindspots is a serious limit to the solver. However, the simulation can still be significant in presence of sporadic blindspots. Most of the simulated configurations are realized in the limit of small or moderate deformation, where the membrane can be considered rigid with respect to the electric solicitation. In such a scenario, blindspots have little effect on the membrane dynamics, and disappear before introducing significant deformations.

Paradoxally, the larger the deformation, the less impactant blind spots are. The largest instabilities are introduced for cases where the membrane is stationary. In such cases blind spots are a permanent feature in time, and cumulation of error over the many iteration can lead to the divergence of the solution. This is most probably the reason for the strong instabilities that are obtained for the case $d = 3$ of the stationary benchmark test.

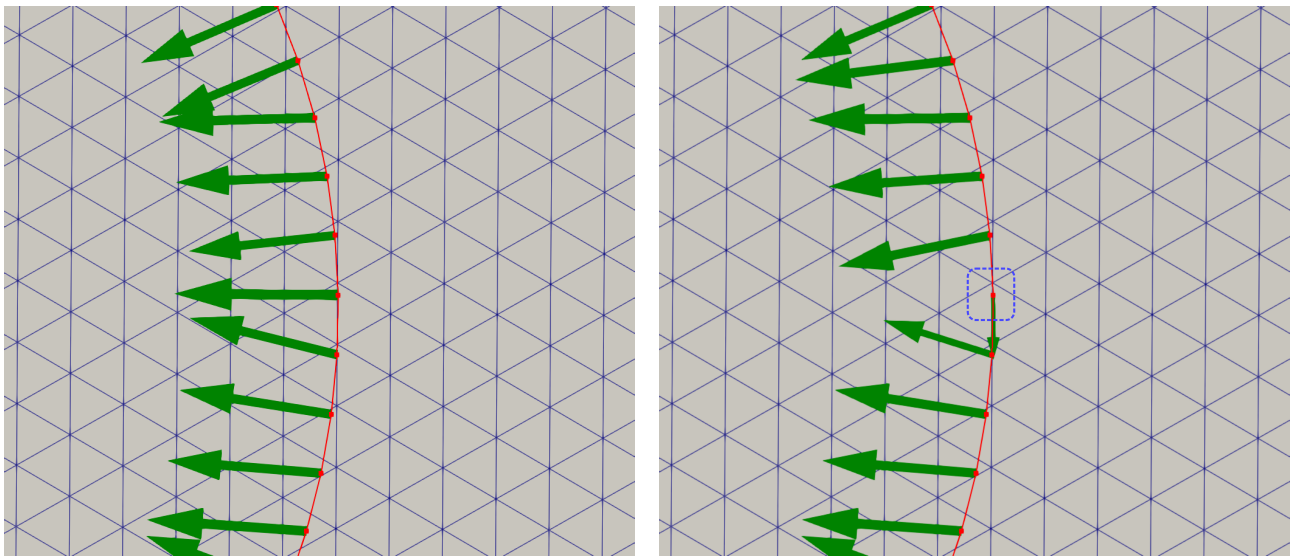


Figure 4.32: Electric force (green) over the membrane (red) over 2 consecutive iterations. Bad intersections between membrane and background mesh can lead to important instability, as shown in the blind spot (blue).

In conclusion, the design of an estimation of electric forces based on the post-processing of a GFM estimation of the electrostatic potential still features a robustness challenge. The challenge is mostly represented by difficulties of designing an interpolation of bulk fields onto the surface Σ which is robust to unfavorable cuts. Although some physical phenomena can be reproduced in a satisfying way for some benchmark configurations, especially for $d = 2$, the determination of blind spots is still an important obstacle.

A possible solution to overcome the difficulties in the estimation of $\boldsymbol{F}_{MW}$ is an approach based on a mesh Ω_h that fits the interface. The advantage of a *fitted* method is that no interpolation procedure is needed. However, a fitted approach introduces an overhead linked to the determination at each time iteration of a bulk mesh that follows the movement of the membrane. This burden can be alleviated if the method is designed to support generic meshes, where the presence of polygonal elements is allowed. The polygonal *fitted* approach to Problem 4.1 is the object of the next Chapter.

CHAPTER

FIVE

A DISCRETE DE RHAM SCHEME FOR AN ELLIPTIC PROBLEM WITH INTERFACES

The literature on numerical methods for interface problems is vast, and we will limit ourselves here to works that bear relations with the present approach. A large number of methods rely on a background mesh which is not compliant with the interface, and are therefore referred to as *unfitted*. In the Generalized/Extended Finite Element method [Sukumar et al. 2000; Strouboulis, Babuška, and Copps 2000], non-polynomial functions with compact support are added to the discrete space in order to capture the behavior at the interface. In the Immersed Finite Element method [Adjerid et al. 2023], the added functions are piecewise polynomials. In the CutFEM method [Burman et al. 2015], interface conditions are taken into account by using discontinuous basis functions inside the elements cut by the interface and relying on Nitsche's techniques for their enforcement. The CutFEM principles have been extended to the Hybrid High-Order (HHO) method [Burman and Ern 2018; Burman and Ern 2019], which supports much more general element geometries than standard Finite Elements; see [Di Pietro, Ern, and Lemaire 2014a; Di Pietro and Ern 2015; Di Pietro and Ern 2017; Di Pietro and Droniou 2020]. Recently, the ideas underlying HHO methods have been applied in [Di Pietro, Droniou, and Rapetti 2020; Di Pietro and Droniou 2023b] to the construction and analysis of discrete de Rham (DDR) sequences in two- and three-space dimensions; see also [Bonaldi et al. 2023] for an extension to arbitrary dimensions using the language of differential forms. In addition to the support of polytopal meshes and arbitrary order, the DDR approach provides a higher level construction that, on certain element types, can result in spaces that are leaner with respect to their Finite Element counterparts (particularly when serendipity techniques are used [Di Pietro and Droniou 2023a] and for advanced complexes [Botti, Di Pietro, and Salah 2023; Di Pietro, Hanot, and Salah 2024]).

In this chapter, we consider a method based on meshes obtained by cutting elements crossed by the interface. No special treatment (such as the addition of ad hoc functions) is required for discretization of the diffusion operator on the elements resulting from the cut. This is because the discretization in the bulk hinges on the H^1-like DDR space of [Di Pietro and Droniou 2023b], which supports general polygonal or polyhedral meshes. As a result, the method enjoys both the assets of unfitted methods (since cuts can occur at arbitrary locations, making the approximation of the interface totally independent of the background mesh) and of fitted methods (since interface elements do not require a special treatment). The support of general polytopal elements additionally provides an effective means to counter the possible degradation of mesh quality resulting from the cuts, since pathological elements can be merged into neighbors in the spirit of [Bassi et al. 2012; Antonietti, Giani, and Houston 2013; Johansson and Larson 2013]; see also [Huang, Wu, and Xiao 2017] for an application of these ideas to conforming finite elements. Additionally, the method proposed here can be seamlessly combined with standard (Lagrange) Finite Elements or Finite Element-Finite Volume methods on non-

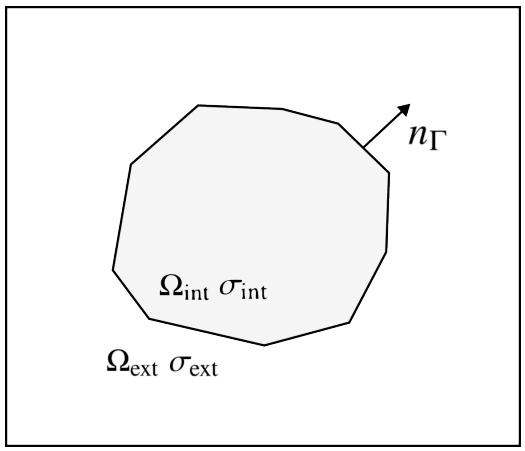


Figure 5.1: Configuration for the continuous problem. The polyline can possibly be an approximation to a smooth interface.

cut elements (such as the ones present in the YALES2BIO platform[1], whose application to Coulter counters motivated the present effort [Taraconat et al. 2019; Taraconat et al. 2023b]). Interface conditions are enforced weakly through subtly defined terms that ensure consistency; see Remark 5 for further insight into this point. The design of such terms, based on the use of trace reconstructions, is indeed one of the main contributions of the present work. Robustness with respect to the jumps of the diffusion coefficient is obtained by using diffusion-dependent weighted averages in the spirit of [Burman and Zunino 2006; Di Pietro, Ern, and Guermond 2008]. To keep the exposition as simple as possible, we focus here on the two-dimensional case with interfaces approximated by closed polygonal chains and consider numerically the case of curved interfaces.

The rest of the chapter is organized as follows. In Section 5.1 we state the continuous problem. The discrete setting (mesh, polynomial spaces) is introduced in Section 5.2. In Section 5.3 we describe the construction leading to the discrete problem and state the scheme as well as the main results of the analysis. A comprehensive set of numerical tests is carried out in Section 5.4, while the application to the Leaky Dielectric Model makes the object of Section 5.5. Finally, the proofs of the stability and error estimates results stated in Section 5.3.6 are provided in Section 5.6.

5.1 Continuous setting

To keep the description of the method as simple as possible, we focus on the two-dimensional case. We emphasize, however, that it is possible to extend the present method to three space dimensions in the spirit of [Di Pietro and Droniou 2023b] , as well as to curved approximations of the interface by adapting the techniques of [Botti and Di Pietro 2018; Veiga, Russo, and Vacca 2019; Yemm 2024] .

Consider an open bounded connected polygonal domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$. Let $\Gamma \subset \Omega$ be a closed non-intersecting polygonal chain such that $\Gamma \cap \partial\Omega = \emptyset$ (see Figure 5.1). The domain is partitioned by Γ into an internal and an external polygonal subdomains, respectively denoted by Ω_{int} and Ω_{ext} in what follows. We denote by n_Γ the unit vector field normal to Γ and pointing out of Ω_{int}.

Given a couple of functions $v = (v_{\text{int}}, v_{\text{ext}})$ with $v_\bullet : \Omega_\bullet \to \mathbb{R}$ for $\bullet \in \{\text{int}, \text{ext}\}$, each smooth enough to admit a trace on Γ, we define the following interface jump operator:

$$[v]_\Gamma \coloneqq v_{\text{int}} - v_{\text{ext}}. \tag{5.1}$$

When applied to couples of vector-valued functions, the jump operator acts component-wise. In what follows, whenever needed, we tacitly identify v with the function $v_{\text{int}} I_{\Omega_{\text{int}}} + v_{\text{ext}} I_{\Omega_{\text{ext}}} : \Omega \setminus \Gamma \to \mathbb{R}$,

[1] See `https://imag.umontpellier.fr/~yales2bio/`.

where $I_{\Omega_\bullet}$ is the characteristic function of $\Omega_\bullet$. Consider a region-wise constant diffusion coefficient $\sigma : \Omega \setminus \Gamma \to \mathbb{R}$ such that $\sigma_{|\Omega_{\text{int}}} \equiv \sigma_{\text{int}} > 0$ and $\sigma_{|\Omega_{\text{ext}}} \equiv \sigma_{\text{ext}} > 0$. Let $f : \Omega \setminus \Gamma \to \mathbb{R}$, $\Phi_\Gamma : \Gamma \to \mathbb{R}^2$, and $J_\Gamma : \Gamma \to \mathbb{R}$ be given functions, which we assume smooth enough for the following discussion to make sense. We consider the problem of finding the couple of scalar-valued functions $u = (u_{\text{int}}, u_{\text{ext}})$ with $u_\bullet : \Omega_\bullet \to \mathbb{R}$, $\bullet \in \{\text{int}, \text{ext}\}$, such that

$$-\nabla \cdot (\sigma_\bullet \nabla u_\bullet) = f \qquad \text{in } \Omega_\bullet,\ \bullet \in \{\text{int}, \text{ext}\}, \tag{5.2a}$$

$$[u]_\Gamma = J_\Gamma \qquad \text{on } \Gamma, \tag{5.2b}$$

$$[\sigma \nabla u]_\Gamma \cdot n_\Gamma = \Phi_\Gamma \qquad \text{on } \Gamma, \tag{5.2c}$$

$$u_{\text{ext}} = 0 \qquad \text{on } \partial\Omega. \tag{5.2d}$$

5.2 Discrete setting

5.2.1 Mesh

We discretize the domain with a polygonal mesh $(\mathcal{T}_h, \mathcal{E}_h)$ in the sense of Di Pietro and Droniou 2020, Definition 1.4, with $\mathcal{T}_h$ collecting the mesh elements and $\mathcal{E}_h$ the mesh edges. We additionally denote by $\mathcal{V}_h$ the set of vertices collecting the edge endpoints. The mesh is assumed to be fitted to the interface, i.e., there exists a subset $\mathcal{E}_{\Gamma,h}$ of $\mathcal{E}_h$ such that $\Gamma = \bigcup_{E \in \mathcal{E}_{\Gamma,h}} \overline{E}$.

Remark 1 (Fitted mesh). It is important to emphasize that fitted polytopal meshes supported by the present method can simply be obtained cutting the elements of a background mesh along the interface, as in the numerical tests of Sections 5.4 and 5.5. This is a major advantage, particularly in the framework of moving interface problems, as it avoids a potentially expensive remeshing step. We also notice that the degradation of mesh quality can also be countered leveraging the support of polytopal elements: whenever an elongated or distorted element results from the cutting, it can be agglomerated into neighboring elements in the spirit of [Bassi et al. 2012; Antonietti, Giani, and Houston 2013; Johansson and Larson 2013] to restore mesh quality.

Remark 2 (Approximation of the interface). The approximation of the interface is totally independent of the background mesh. This is important when considering the more general case of curved interfaces, where a finer discretization can be needed. We refer to Remark 8 for further insight into the situation when the interface is refined "faster" than the bulk mesh.

For $\bullet \in \{\text{int}, \text{ext}\}$, we denote by $\mathcal{T}_h^\bullet$, $\mathcal{E}_h^\bullet$, and $\mathcal{V}_h^\bullet$, respectively, the sets of elements, edges, and vertices contained in $\overline{\Omega}_\bullet$. Notice that edges that lie on Γ belong to both $\mathcal{E}_h^{\text{int}}$ and $\mathcal{E}_h^{\text{ext}}$, and similarly for vertices. For any element $T \in \mathcal{T}_h$, we denote by $\mathcal{E}_T$ the set collecting its edges. Symmetrically, the set of elements sharing one edge $E \in \mathcal{E}_h$ is denoted by $\mathcal{T}_E$. For each edge $E \in \mathcal{E}_h$, we fix once and for all a unit normal vector n_E and, for any $T \in \mathcal{T}_E$, we denote by $\omega_{TE} \in \{-1, 1\}$ the relative orientation of E with respect to T, such that $\omega_{TE} n_E$ points out of T.

In what follows, we assume that the mesh we are working on belongs to a regular sequence in the sense of Di Pietro and Droniou 2020, Definition 1.9. Given a mesh element or edge $X \in \mathcal{T}_h \cup \mathcal{E}_h$, we denote by h_X its diameter, so that $h = \max_{T \in \mathcal{T}_h} h_T$. We will abbreviate by $a \lesssim b$ the inequality $a \le Cb$ with C independent of h, σ and, for local inequalities, the corresponding element or edge. Further details on the dependence of the hidden constants will be provided when appropriate.

5.2.2 Polynomial spaces

Given $X \in \mathcal{T}_h \cup \mathcal{E}_h$ and an integer $m \ge 0$, we denote by $\mathcal{P}^m(X)$ the space spanned by the restriction of polynomials of the space variables to X and by π_X^m the corresponding L^2-orthogonal projector. We

conventionally set $\mathcal{P}^{-1}(X) \coloneqq \{0\}$. For all $T \in \mathcal{T}_h$, we will also need the space

$$\mathcal{R}^{\mathrm{c},m}(T) \coloneqq (x - x_T)\mathcal{P}^{m-1}(T),$$

where x_T is a point inside T at a distance from its boundary comparable to h_T. It can be proved that the divergence from $\mathcal{R}^{\mathrm{c},m}(T)$ to $\mathcal{P}^{m-1}(T)$ is an isomorphism; cf. Arnold 2018, Corollary 7.3.

At the global level, we will need the broken polynomial space

$$\mathcal{P}^m(\mathcal{T}_h) \coloneqq \{v \in L^2(\Omega) \ : \ v_{|T} \in \mathcal{P}^m(T) \text{ for all } T \in \mathcal{T}_h\}.$$

5.3 Discrete problem

5.3.1 Discrete space

For any $k \geq 0$ and $\bullet \in \{\mathrm{int}, \mathrm{ext}\}$, we let

$$\underline{V}^k_{\bullet,h} \coloneqq \Big\{\underline{v}_h = \big((v_T)_{T\in\mathcal{T}_h^\bullet}, (v_E)_{E\in\mathcal{E}_h^\bullet}, (v_V)_{V\in\mathcal{V}_h^\bullet}\big) \ : \begin{aligned} &v_T \in \mathcal{P}^{k-1}(T) \text{ for all } T \in \mathcal{T}_h^\bullet,\\ &v_E \in \mathcal{P}^{k-1}(E) \text{ for all } E \in \mathcal{E}_h^\bullet,\\ &v_V \in \mathbb{R} \text{ for all } V \in \mathcal{V}_h^\bullet\Big\}. \end{aligned}$$

We consider the discrete space

$$\underline{V}^k_h \coloneqq \underline{V}^k_{\mathrm{int},h} \times \underline{V}^k_{\mathrm{ext},h}.$$

as well as its subspace $\underline{V}^k_{h,0}$ with edge and vertex values vanishing on $\partial\Omega$.

The interpolator $\underline{I}^k_h : C^0(\overline{\Omega}_{\mathrm{int}}) \times C^0(\overline{\Omega}_{\mathrm{ext}}) \to \underline{V}^k_h$ is such that, for all $v = (v_{\mathrm{int}}, v_{\mathrm{ext}}) \in C^0(\overline{\Omega}_{\mathrm{int}}) \times C^0(\overline{\Omega}_{\mathrm{ext}})$,

$$\underline{I}^k_h v \coloneqq \big(\underline{I}^k_{\mathrm{int},h} v_{\mathrm{int}}, \underline{I}^k_{\mathrm{ext},h} v_{\mathrm{ext}}\big),$$

where, for $\bullet \in \{\mathrm{int}, \mathrm{ext}\}$,

$$\underline{I}^k_{\bullet,h} v_\bullet \coloneqq \big((\pi^{k-1}_T v_\bullet)_{T\in\mathcal{T}_h^\bullet}, (\pi^{k-1}_E v_\bullet)_{E\in\mathcal{E}_h^\bullet}, (v_\bullet(x_V))_{V\in\mathcal{V}_h^\bullet}\big).$$

For all $T \in \mathcal{T}_h$, we respectively denote the restrictions of $\underline{V}^k_h$, $\underline{v}_h \in \underline{V}^k_h$, and $\underline{I}^k_h$ to T by $\underline{V}^k_T$, $\underline{v}_T \in \underline{V}^k_T$, and $\underline{I}^k_T$. Such restrictions are obtained collecting the polynomial components on T and its boundary. More specifically, since every mesh element T is contained in one and only one subdomain $\mathcal{T}_h^\bullet$, $\bullet \in \{\mathrm{int}, \mathrm{ext}\}$, when restricting a vector of polynomials $\underline{v}_h \in \underline{V}^k_h$ to T, all the components of the local vector $\underline{v}_T = (v_T, (v_E)_{E\in\mathcal{E}_T}, (v_V)_{V\in\mathcal{V}_T})$ (with $\mathcal{V}_T$ denoting the set of vertices of T) are extracted from $\underline{V}^k_{\bullet,h}$. This removes all possible ambiguity for mesh elements such that $\partial T \cap \Gamma \neq \emptyset$.

5.3.2 Element gradient and potential

For any $T \in \mathcal{T}_h$, any $\underline{v}_T \in \underline{V}^k_T$, and any $E \in \mathcal{E}_T$, we let the edge potential v_{TE} be the unique function in $\mathcal{P}^{k+1}(E)$ such that

$$v_{TE}(x_V) = v_V \text{ for any endpoint } V \text{ of } E \text{ and } \pi^{k-1}_E v_{TE} = v_E.$$

Remark 3 (Edge potential). Clearly, $v_{TE} \equiv 0$ whenever $E \subset \partial\Omega$ is a boundary edge and $\underline{v}_h \in \underline{V}^k_{h,0}$. On the other hand, when $E \subset (\partial T_1 \cap \partial T_2) \setminus \Gamma$ is an internal edge that does not lie on the interface, the value of the edge potential does not depend on the element, i.e., $v_{T_1E} = v_{T_2E}$.

We define the discrete gradient $G_T^k : \underline{V}_T^k \to \mathcal{P}^k(T)^2$ and potential $p_T^{k+1} : \underline{V}_T^k \to \mathcal{P}^{k+1}(T)$ such that, for all $\underline{v}_T \in \underline{V}_T^k$,

$$\int_T G_T^k \underline{v}_T \cdot \tau = -\int_T v_T(\nabla \cdot \tau) + \sum_{E\in\mathcal{E}_T} \omega_{TE} \int_E v_{TE}(\tau \cdot n_E) \qquad \forall \tau \in \mathcal{P}^k(T)^2,$$
$$\int_T p_T^{k+1} \underline{v}_T (\nabla \cdot \tau) = -\int_T G_T^k \underline{v}_T \cdot \tau + \sum_{E\in\mathcal{E}_T} \omega_{TE} \int_E v_{TE}(\tau \cdot n_E) \qquad \forall \tau \in \mathcal{R}^{c,k+2}(T). \tag{5.3}$$

Remark 4 (Validity of (5.3)). Following Di Pietro and Droniou 2023b, Remark 17, the relation (5.3) actually holds for all $\tau \in \mathcal{P}^k(T)^2 + \mathcal{R}^{c,k+2}(T)$.

Accounting for the previous remark, we notice that, integrating by parts the left-hand side of (5.3) and rearranging, we have, for all $(\underline{v}_T, \tau) \in \underline{V}_T^k \times (\mathcal{P}^k(T)^2 + \mathcal{R}^{c,k+2}(T))$,

$$\int_T \nabla p_T^{k+1} \underline{v}_T \cdot \tau = \int_T G_T^k \underline{v}_T \cdot \tau + \sum_{E\in\mathcal{E}_T} \omega_{TE} \int_E (p_T^{k+1} \underline{v}_T - v_{TE})(\tau \cdot n_E). \tag{5.4}$$

Selecting $\tau = \nabla p_T^{k+1} \underline{v}_T$ (this is possible since $\nabla p_T^{k+1} \underline{v}_T \in \mathcal{P}^k(T)^d$) in the above expression, using Cauchy–Schwarz, $(2,\infty,2)$-Hölder, and trace inequalities along with $\|n_E\|_{L^\infty(E)^2} \le 1$ in the right-hand side, and simplifying, we get

$$\|\nabla p_T^{k+1} \underline{v}_T\|_{L^2(T)^2} \lesssim \left(\|G_T^k \underline{v}_T\|_{L^2(T)^2}^2 + h_T^{-1} \sum_{E\in\mathcal{E}_T} \|p_T^{k+1} \underline{v}_T - v_{TE}\|_{L^2(E)}^2 \right)^{1/2}. \tag{5.5}$$

Let $v \in H^{r+2}(T)$ for some $r \in \{0,\dots,k\}$ and set $\widehat{\underline{v}}_T := \underline{I}_T^k v$. Using the techniques of [Di Pietro and Droniou 2023b] , where the three-dimensional case is considered, it can be proved that

$$\|\widehat{v}_{TE} - v\|_{L^2(E)} \lesssim h_T^{r+3/2} |v|_{H^{r+2}(T)} \qquad \forall E \in \mathcal{E}_T, \tag{5.6}$$
$$\|G_T^k \widehat{\underline{v}}_T - \nabla v\|_{L^2(T)^2} + h_T^{1/2} \|G_T^k \widehat{\underline{v}}_T - \nabla v\|_{L^2(\partial T)^2} \lesssim h_T^{r+1} |v|_{H^{r+2}(T)}, \tag{5.7}$$
$$\|p_T^{k+1} \widehat{\underline{v}}_T - v\|_{L^2(T)} + h_T^{1/2} \|p_T^{k+1} \widehat{\underline{v}}_T - v\|_{L^2(\partial T)} \lesssim h_T^{r+2} |v|_{H^{r+2}(T)}. \tag{5.8}$$

For future use, we also define the global discrete gradient operator $G_h^k : \underline{V}_h^k \to \mathcal{P}^k(\mathcal{T}_h)^2$ such that, for all $\underline{v}_h \in \underline{V}_h^k$,

$$(G_h^k \underline{v}_h)_{|T} := G_T^k \underline{v}_T \qquad \forall T \in \mathcal{T}_h.$$

5.3.3 Interface trace operators

Let $E \in \mathcal{E}_{\Gamma,h}$ and denote by $T_{\rm int} \in \mathcal{T}_h^{\rm int}$ and $T_{\rm ext} \in \mathcal{T}_h^{\rm ext}$ the unique elements such that $E \subset \partial T_{\rm int} \cap \partial T_{\rm ext}$. Notice that, while such elements clearly depend on E, we do not highlight this dependency in the notation as it will be clear from the context. We define the edge jump $[\cdot]_E : \underline{V}_h^k \to \mathcal{P}^{k+1}(E)$ and skewed average $\{\cdot\}_{\overline{\lambda},E} : \underline{V}_h^k \to \mathcal{P}^{k+1}(E)$ operators such that, for all $\underline{v}_h \in \underline{V}_h^k$,

$$[\underline{v}_h]_E := v_{T_{\rm int}E} - v_{T_{\rm ext}E}, \qquad \{\underline{v}_h\}_{\overline{\lambda},E} := \lambda_{\rm ext} v_{T_{\rm int}E} + \lambda_{\rm int} v_{T_{\rm ext}E}, \tag{5.9}$$

with $\lambda_{\rm int}$ and $\lambda_{\rm ext}$ such that

$$\lambda_{\rm int} := \frac{\sigma_{\rm ext}}{\sigma_{\rm int} + \sigma_{\rm ext}}, \qquad \lambda_{\rm ext} := \frac{\sigma_{\rm int}}{\sigma_{\rm int} + \sigma_{\rm ext}}. \tag{5.10}$$

We additionally let, for any vector-valued field Ψ smooth enough to admit a possibly two-valued trace on E,

$$\{\Psi\}_{\lambda,E} := \lambda_{\rm int} \gamma_{T_{\rm int}E} \Psi + \lambda_{\rm ext} \gamma_{T_{\rm ext}E} \Psi, \tag{5.11}$$

where $\gamma_{T_\bullet E}\Psi$ denotes the trace of $\Psi_{|T_\bullet}$ on E.

5.3.4 Discrete problem

Let

$$\sigma_T \coloneqq \sigma_{|T} \qquad \forall T \in \mathcal{T}_h.$$

Given a user-dependent parameter $\eta > 0$, we define the bilinear form $a_h : \underline{V}_h^k \times \underline{V}_h^k \to \mathbb{R}$ and the linear form $\ell_h : \underline{V}_h^k \to \mathbb{R}$ such that, for all $(\underline{w}_h, \underline{v}_h) \in \underline{V}_h^k \times \underline{V}_h^k$,

$$\begin{aligned} a_h(\underline{w}_h, \underline{v}_h) \coloneqq \sum_{T\in\mathcal{T}_h} \left(\int_T \sigma_T G_T^k \underline{w}_T \cdot G_T^k \underline{v}_T + \frac{\sigma_T}{h_T} \sum_{E\in\mathcal{E}_T} \int_E (p_T^{k+1}\underline{w}_T - w_{TE})(p_T^{k+1}\underline{v}_T - v_{TE}) \right) \\ - \sum_{E\in\mathcal{E}_{\Gamma,h}} \int_E \{\sigma G_h^k \underline{w}_h\}_{\lambda,E} \cdot n_E [\underline{v}_h]_E + \eta \sum_{E\in\mathcal{E}_{\Gamma,h}} \frac{\alpha}{h_E} \int_E [\underline{w}_h]_E [\underline{v}_h]_E, \end{aligned} \tag{5.12}$$

where

$$\alpha \coloneqq \frac{2\sigma_{\mathrm{int}}\sigma_{\mathrm{ext}}}{\sigma_{\mathrm{int}} + \sigma_{\mathrm{ext}}} \tag{5.13}$$

and

$$\ell_h(\underline{v}_h) \coloneqq \sum_{T\in\mathcal{T}_h} \int_T f p_T^{k+1} \underline{v}_T + \sum_{E\in\mathcal{E}_{\Gamma,h}} \int_E \Phi_\Gamma \{\underline{v}_h\}_{\overline{\lambda},E} + \eta \sum_{E\in\mathcal{E}_{\Gamma,h}} \frac{\alpha}{h_E} \int_E J_\Gamma [\underline{v}_h]_E. \tag{5.14}$$

The discrete problem reads: Find $\underline{u}_h \in \underline{V}_{h,0}^k$ such that

$$a_h(\underline{u}_h, \underline{v}_h) = \ell_h(\underline{v}_h) \qquad \forall \underline{v}_h \in \underline{V}_{h,0}^k. \tag{5.15}$$

Remark 5 (Formulation of the interface terms). It is worth noticing that the scheme above is not a simple extension of discontinuous Galerkin techniques to the interface problem of Section 5.1 with the element potential playing the role of the discontinuous solution inside trace operators. On the contrary, such operators use the edge potential in a subtle way, which is required for optimal order consistency; see, in particular, the passages leading to (5.30) in the proof of Lemma 13 below.

5.3.5 Energy norm and interface jump seminorm

In order to state the main results of the theoretical analysis of the method, we define on $\underline{V}_h^k$ the energy norm $\|\cdot\|_{\mathrm{en},h}$ such that, for all $\underline{v}_h \in \underline{V}_h^k$,

$$\|\underline{v}_h\|_{\mathrm{en},h}^2 \coloneqq \sum_{T\in\mathcal{T}_h} \sigma_T \left(\|G_T^k \underline{v}_T\|_{L^2(T)^2}^2 + h_T^{-1} \sum_{E\in\mathcal{E}_T} \|p_T^{k+1}\underline{v}_T - v_{TE}\|_{L^2(E)}^2 \right) + |\underline{v}_h|_{\mathrm{J},h}^2, \tag{5.16}$$

where the interface jump seminorm is such that

$$|\underline{v}_h|_{\mathrm{J},h}^2 \coloneqq \sum_{E\in\mathcal{E}_{\Gamma,h}} \frac{\alpha}{h_E} \|[\underline{v}_h]_E\|_{L^2(E)}^2. \tag{5.17}$$

Proposition 6 (Energy norm)**.** *The map $\|\cdot\|_{\mathrm{en},h}$ defines a norm on $\underline{V}_{h,0}^k$.*

Proof. $\|\cdot\|_{\mathrm{en},h}$ is clearly a seminorm, so we only have to prove that, for all $\underline{v}_h \in \underline{V}_{h,0}^k$, $\|\underline{v}_h\|_{\mathrm{en},h} = 0$ implies $\underline{v}_h = \underline{0}$. The condition $\|\underline{v}_h\|_{\mathrm{en},h} = 0$ implies: (i) for all $T \in \mathcal{T}_h$, $G_T^k \underline{v}_T = 0$ and $(p_T^{k+1}\underline{v}_T)|_E = v_{TE}$ for all $E \in \mathcal{E}_T$ and (ii) for all $E \in \mathcal{E}_{\Gamma,h}$, $[\underline{v}_h]_E = 0$. By (5.5), point (i) implies, in turn, that $p_T^{k+1}\underline{v}_T$ is constant on each $T \in \mathcal{T}_h$. Since $v_{TE} = 0$ whenever $E \in \mathcal{E}_h$ is a boundary edge contained in $\partial\Omega$ and $T \in \mathcal{T}_h$ is the unique mesh element to which it belongs, $p_T^{k+1}\underline{v}_T = 0$. Proceeding from $\partial\Omega$ towards

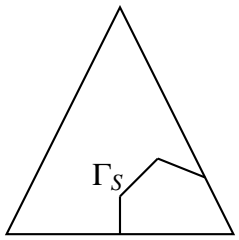


Figure 5.2: An example of triangle $S \in \mathcal{S}_h$ cut by the interface with polyline $\Gamma_S \coloneqq \Gamma \cap S$.

the interior of Ω_{ext}, this gives $p_T^{k+1}\underline{v}_T = 0$ for all $T \in \mathcal{T}_h^{\text{ext}}$ and $v_{TE} = 0$ for all $E \in \mathcal{E}_T$. These two conditions combined show that all element, edge, and vertex components of $\underline{v}_h$ in Ω_{ext} vanish. Since interface jumps vanish as well by point (ii) above, the edge and vertex components of $\underline{v}_h$ on the interface Γ from the side of Ω_{int} are zero. The same reasoning as for Ω_{ext} can therefore be applied (proceeding from the interface Γ towards the interior of Ω_{int}) to show that all elements, edge, and vertex values of $\underline{v}_h$ in Ω_{int} vanish, thus concluding the proof. □

5.3.6 Main results

We recall the following discrete trace inequality valid for any integer $m \geq 0$, any $T \in \mathcal{T}_h$, and any $E \in \mathcal{E}_T$: For all $\varphi \in \mathcal{P}^m(T)$,

$$\|\varphi\|_{L^2(E)} \leq C_{\text{tr}} h_E^{-1/2} \|\varphi\|_{L^2(T)}, \tag{5.18}$$

where the real number $C_{\text{tr}} > 0$ only depends on m and the mesh regularity parameter.

Lemma 7 (Stability)**.** *Let* $N_\partial \coloneqq \max_{T\in\mathcal{T}_h} \operatorname{card}(\mathcal{E}_T \cap \mathcal{E}_{\Gamma,h})$ *and assume that*

$$\eta > \frac{C_{\text{tr}}^2 N_\partial}{4\epsilon} \tag{5.19}$$

for some real number $0 < \epsilon < 1$*. Then, for all* $\underline{v}_h \in \underline{V}_{h,0}^k$*, it holds*

$$C_{\text{stab}} \|\underline{v}_h\|_{\text{en},h}^2 \leq a_h(\underline{v}_h, \underline{v}_h), \tag{5.20}$$

with $C_{\text{stab}} \coloneqq \min\left\{1-\epsilon, \eta - \frac{C_{\text{tr}}^2 N_\partial}{4\epsilon}\right\}$.

Proof. See Section 5.6.1 □

Remark 8 (Interface elements with a large number of edges). In the context of meshes obtained by cutting an underlying standard triangular mesh (such as the ones considered numerically in Sections 5.4 and 5.5), a slight modification of the scheme in the spirit of [Cangiani et al. 2017; Droniou and Yemm 2022] can make it robust in the case where the interface is refined "asymptotically faster" than the bulk mesh. Denote by $\mathcal{S}_h$ the set of triangles of the bulk mesh that are cut by the interface. Let, for all $S \in \mathcal{S}_h$, $\mathcal{E}_{\Gamma,S} \coloneqq \{E \in \mathcal{E}_{\Gamma,h} : E \subset \partial S\}$, denote by $\mathfrak{h}_S \coloneqq \sum_{E\in\mathcal{E}_{\Gamma,S}} h_E$ the length of the polyline $\Gamma_S \coloneqq \Gamma \cap S$, and assume that $h_S \simeq \mathfrak{h}_S$; see Figure 5.2. Consider the bilinear and linear forms obtained replacing the last terms in the right-hand sides of (5.12) and (5.14) respectively by

$$\eta \sum_{S\in\mathcal{S}_h} \frac{\alpha}{h_S} \sum_{E\in\mathcal{E}_{\Gamma,S}} \int_E [\underline{w}_h]_E [\underline{v}_h]_E \text{ and } \eta \sum_{S\in\mathcal{S}_h} \frac{\alpha}{h_S} \sum_{E\in\mathcal{E}_{\Gamma,S}} \int_E J_\Gamma [\underline{v}_h]_E.$$

Then, using the discrete trace inequality $\|\varphi\|_{L^2(\partial T)} \leq \widetilde{C}_{\text{tr}} h_T^{-1/2} \|\varphi\|_{L^2(T)}$ on the whole boundary of the elements of $\mathcal{T}_h$ instead of (5.18) and letting $|\underline{v}_h|_{\text{J},h}^2 \coloneqq \sum_{S\in\mathcal{S}_h} \sum_{E\in\mathcal{E}_{\Gamma,S}} \frac{\alpha}{h_S} \|[\underline{v}_h]_E\|_{L^2(E)}^2$ (compare with

(5.17)), one can replace condition (5.19) with $\eta > \frac{\widetilde{C}_{\mathrm{tr}}^2}{4\epsilon}$, which no longer depends on the (possibly large) integer N_∂.

Theorem 9 (Error estimate)**.** *Let $u = (u_{\mathrm{int}}, u_{\mathrm{ext}})$ be the weak solution to* (5.2) *and $\underline{u}_h$ solve* (5.15)*. Under assumption* (5.19)*, with $\overline{\sigma} := \max\{\sigma_{\mathrm{int}}, \sigma_{\mathrm{ext}}\}$, and further assuming that $u = (u_{\mathrm{int}}, u_{\mathrm{ext}}) \in \left[C^0(\overline{\Omega}_{\mathrm{int}}) \cap H^{r+2}(\mathcal{T}_h^{\mathrm{int}})\right] \times \left[C^0(\overline{\Omega}_{\mathrm{ext}}) \cap H^{r+2}(\mathcal{T}_h^{\mathrm{ext}})\right]$ for some $r \in \{0, \dots, k\}$, it holds*

$$\|\underline{u}_h - \underline{I}_h^k u\|_{\mathrm{en},h} \lesssim \overline{\sigma}^{1/2} h^{r+1} |u|_{H^{r+2}(\mathcal{T}_h)},$$

where $|\cdot|_{H^{r+2}(\mathcal{T}_h)}$ is the broken H^{r+2}-seminorm on the mesh $\mathcal{T}_h$ and the hidden constant depends only on the domain, the stability constant C_{stab} in (5.20)*, the polynomial degree k, and the mesh regularity parameter (but is independent of both the meshsize and σ).*

Proof. See Section 5.6.2. □

Remark 10 (Robustness in σ). Notice that the right-hand side of the above estimate does not depend on the ratio $\frac{\sigma_{\mathrm{int}}}{\sigma_{\mathrm{ext}}}$, making it robust in the case of media with highly contrasting properties. Crucial to obtain this robustness property is the use of weighted averages in the spirit of [Burman and Zunino 2006; Di Pietro, Ern, and Guermond 2008] .

Remark 11 (Extension to non-piecewise constant diffusion coefficients). The arguments for stability and convergence can be extended to the case of diffusion coefficients $\sigma_{\mathrm{int}} : \Omega_{\mathrm{int}} \to \mathbb{R}$ and $\sigma_{\mathrm{ext}} : \Omega_{\mathrm{ext}} \to \mathbb{R}$ that are Lipschitz-continuous on the corresponding subdomain.

5.4 Numerical tests

To numerically assess the theoretical results of Section 5.3.6, we have implemented in Python the lowest-order version of the scheme corresponding to $k = 0$. The implementation is a prototype for the final one in YALES2BIO, and is therefore compliant with the constraints of this software platform.

5.4.1 Square interface

We consider the square domain $\Omega = (-1/2, 1/2)^2$, with a square interface

$$\Gamma = \left\{(x, y) \in [-1/4, 1/4]^2 \ : \ |x| = 1/4 \text{ or } |y| = 1/4\right\}.$$

Since the interface is a polygonal chain, no geometric error is introduced. We consider the following family of solutions $u = (u_{\mathrm{int}}, u_{\mathrm{ext}})$ parametrized by the ratio $\frac{\sigma_{\mathrm{ext}}}{\sigma_{\mathrm{int}}}$ and depicted in Figure 5.3:

$$u_{\mathrm{int}} = \frac{\sigma_{\mathrm{ext}}}{\sigma_{\mathrm{int}}}(x^2 - y^2), \qquad u_{\mathrm{ext}} = x^2 - y^2 \tag{5.21}$$

with forcing term, (non-homogeneous) boundary conditions, and values for J_Γ and Φ_Γ inferred from the expression of $u = (u_{\mathrm{int}}, u_{\mathrm{ext}})$.

We consider two mesh families, both compliant with the interface. The first sequence is composed of Cartesian orthogonal meshes. The second sequence is obtained from the latter by randomly moving vertices that are not located on the interface within a circle of radius $\frac{\ell}{5}$, with ℓ denoting the measure of the sides of the element in the non-deformed mesh; see Figure 5.4.

In order to assess the robustness of the method with respect to the ratio $\frac{\sigma_{\mathrm{int}}}{\sigma_{\mathrm{ext}}}$, we let this quantity vary in $\{10^{-6}, 10^{-3}, 10^{3}, 10^{6}\}$. We monitor two measures of the error: the energy norm defined by (5.16) and the component L^2-norm $\|\cdot\|_{0,h}$ defined by Di Pietro and Droniou 2021, Eq. (4.20), i.e.,

$$\|\underline{v}_h\|_{0,h} := \left(\sum_{T\in\mathcal{T}_h} \|v_T\|_{L^2(T)}^2 + h_T \sum_{E\in\mathcal{E}_T} \|v_E\|_{L^2(E)}^2\right)^{\frac{1}{2}} \qquad \forall \underline{v}_h \in \underline{V}_h^k.$$

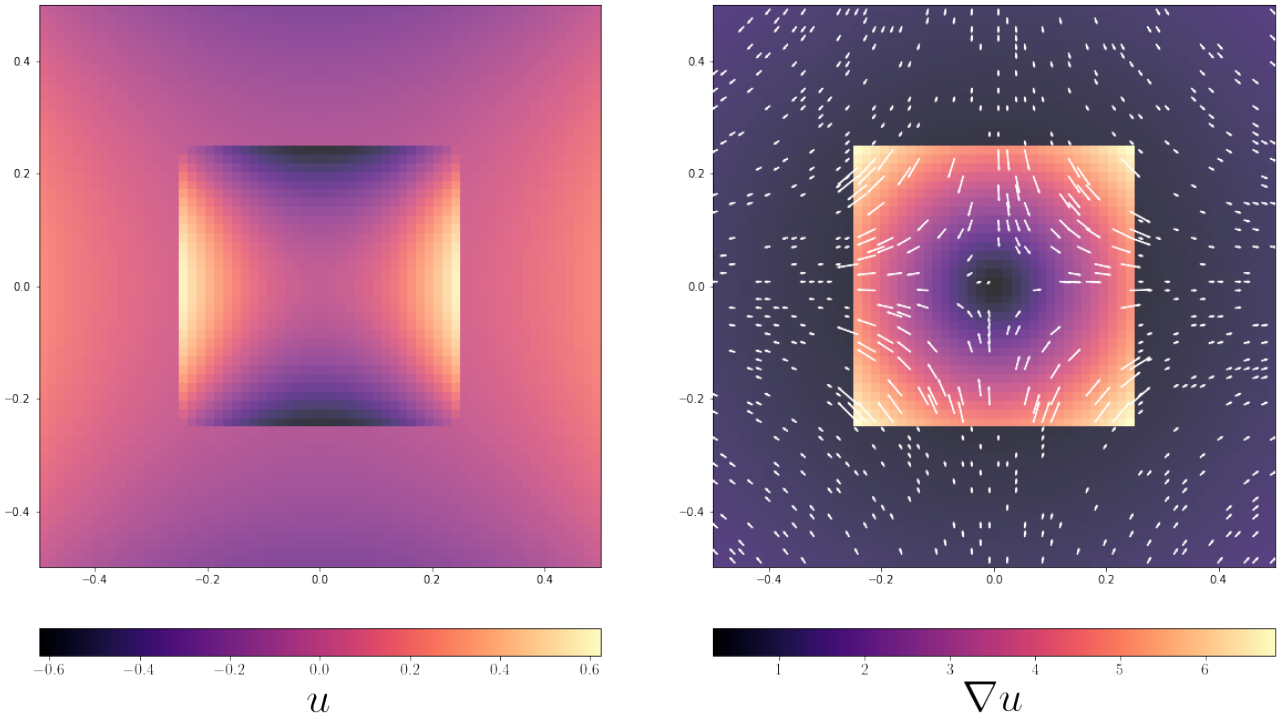


Figure 5.3: The exact solution (5.21) considered in Section 5.4.1 and its gradient for $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 10^{-1}$

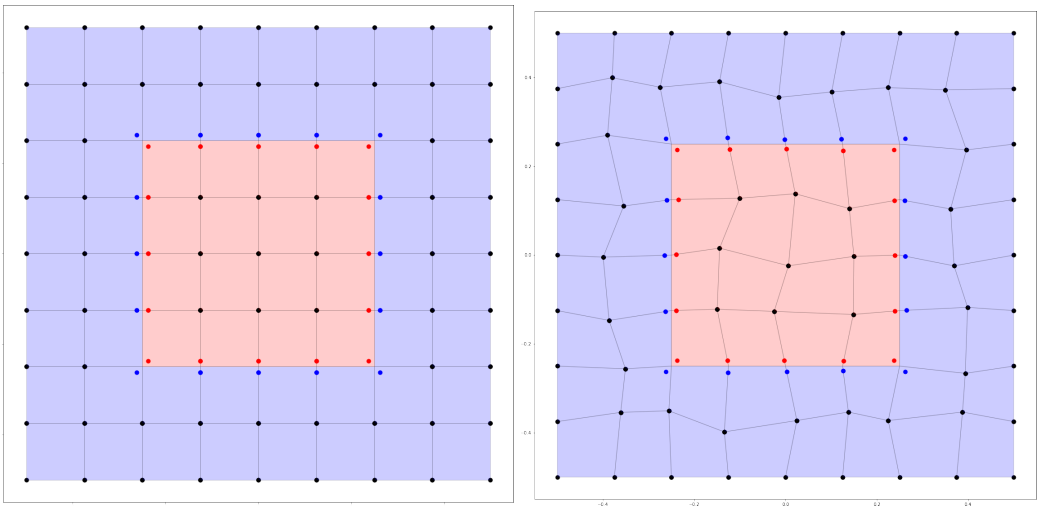


Figure 5.4: Mesh sequences considered in the numerical test of Section 5.4.1.

In all the cases, the error is normalized with respect to the corresponding norm of the discrete solution. The results that are shown are obtained with $\eta = 1$.

The results reported in Figure 5.5 and 5.6 show that the energy norm converges with order 1 (or slightly more), as predicted by Theorem 9 with $r = 0$. We additionally notice that the error is of comparable magnitude irrespectively of the value of $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}}$, which confirms the robustness of the method with respect to the jumps of the diffusion coefficients discussed in Remark 10. As for the error in the L^2-like norm, convergence is close to second order, and its magnitude does not vary significantly with the ratio $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}}$.

5.4.2 Circular interface

The second test introduces an additional difficulty, namely the fact that we deal with a curved interface. More specifically, in the square domain $\Omega = [-1/2, 1/2]^2$, we consider the circular interface $\Gamma = \{(x, y) : x^2 + y^2 = R^2\}$ with $R = 1/4$. The convergence of the method is tested considering the following family of solutions $u = (u_{\text{int}}, u_{\text{ext}})$, represented in Figure 5.7:

$$u_{\text{int}} = \frac{2\sigma_{\text{ext}}}{\sigma_{\text{ext}} + \sigma_{\text{ext}}} x \qquad u_{\text{ext}} = 1 + \left[1 + \left(\frac{\sigma_{\text{ext}} - \sigma_{\text{int}}}{\sigma_{\text{ext}} + \sigma_{\text{int}}}\right) \frac{R^2}{x^2 + y^2}\right] x. \tag{5.22}$$

We consider a sequence of unstructured triangular meshes of Ω with mesh size halved at each refinement step and a family of polygonal discretizations of Γ with segment length divided by 2^M at each refinement step (the integer M therefore represents the refinement ratio of the interface with respect to the background mesh: e.g., if $M = 1$, the background mesh and the interface are refined at the same rate. A fitted mesh is generated by splitting the elements of the original triangular mesh cut by the interface, as represented in Figure 5.8. Please, notice that the number of interface edges of $\mathcal{E}_{\Gamma,h}$ is greater than 2^M, as interface edges that are intersected by edges of the simplicial mesh give place to at least 2 edges of the polygonal mesh originating from the cut. The test is then repeated with $M = 4$ for different values of $\sigma_{\text{int}}/\sigma_{\text{ext}}$ taken in $\{10^{-6}, 10^{-3}, 10^3, 10^6\}$ to assess the convergence and robustness properties of the method. The results that are shown are obtained with $\eta = 1$; see Figure 5.10. As for the test of Section 5.4.1, slightly more than the theoretical convergence rate 1 is obtained for the energy norm. In order to explore the impact of the refinement ratio, in Figure 5.9 we let $\sigma_{\text{int}}/\sigma_{\text{ext}} = 10^{-1}$ and solve for several values of M. The results suggest that $M = 2$ is sufficient to get the theoretical convergence rate 1, showing the ability of the method to capture curved interfaces without increasing the number of interface edges.

5.4.3 Generic interface

In the square domain $\Omega = [-1/2, 1/2]^2$, we consider a last test where the interface is obtained by deforming a circle. The additional difficulty comes from the fact that the curvature is no longer constant. To test the convergence of the method, we consider the family of polynomial solution (5.21) used for the case of a square interface depicted in Figure 5.11. Keeping the refinement ratio $M = 2$, a convergence test showed in Figure 5.12 is realized. The convergence rate over 1 confirms the theoretical prediction of Theorem 9.

5.5 Application to the Leaky Dielectric Model

In this section we discuss a version of problem (5.2) where the interface jump J_Γ is time dependent and obeys an evolution equation depending on the interface gradient of $u = (u_{\text{int}}, u_{\text{ext}})$.

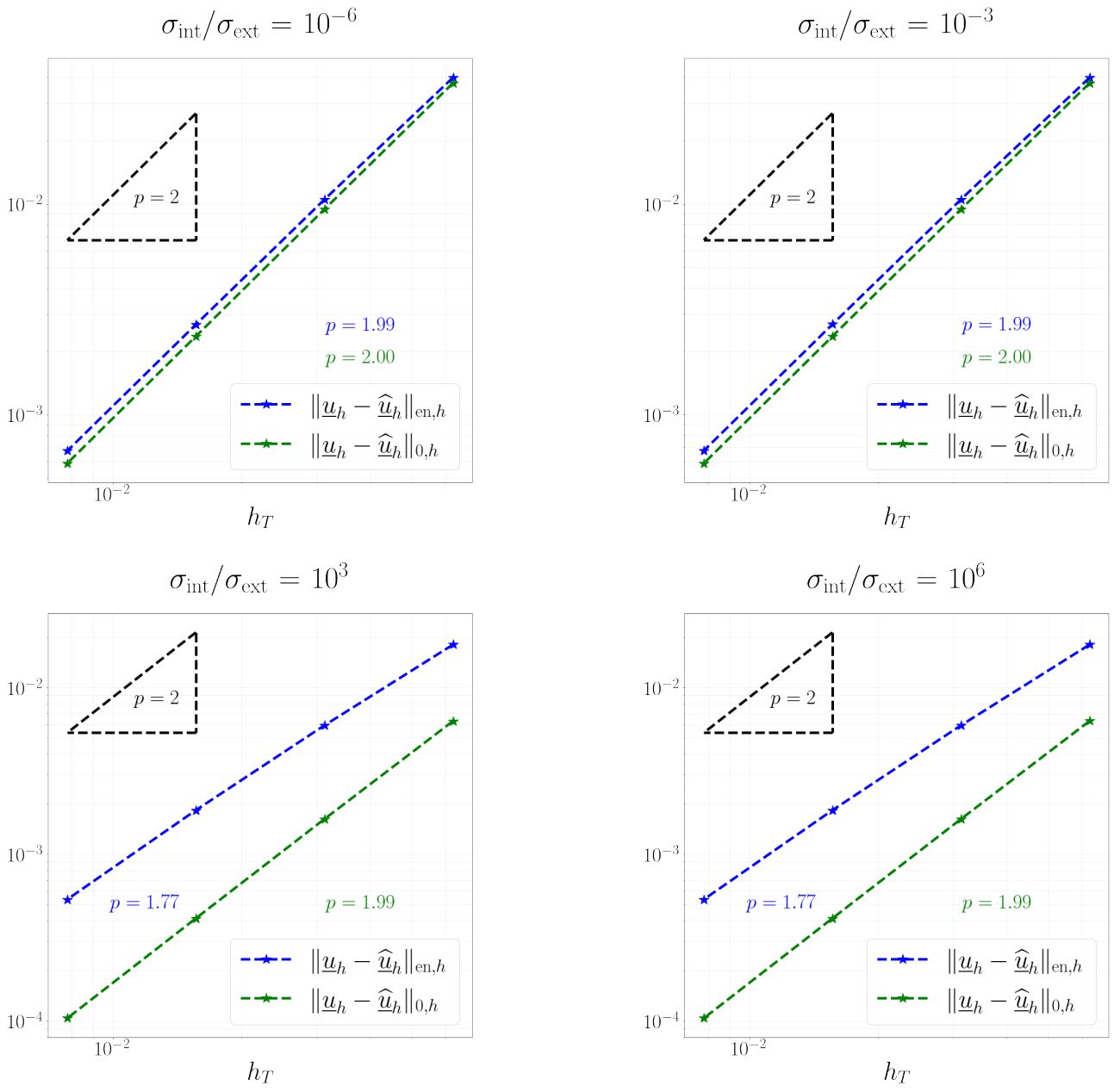


Figure 5.5: Convergence for different values of $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}}$ over a mesh sequence of Cartesian orthogonal meshes, as described in Section 5.4.1. Error is normalized with respect to the norm of the reference solution.

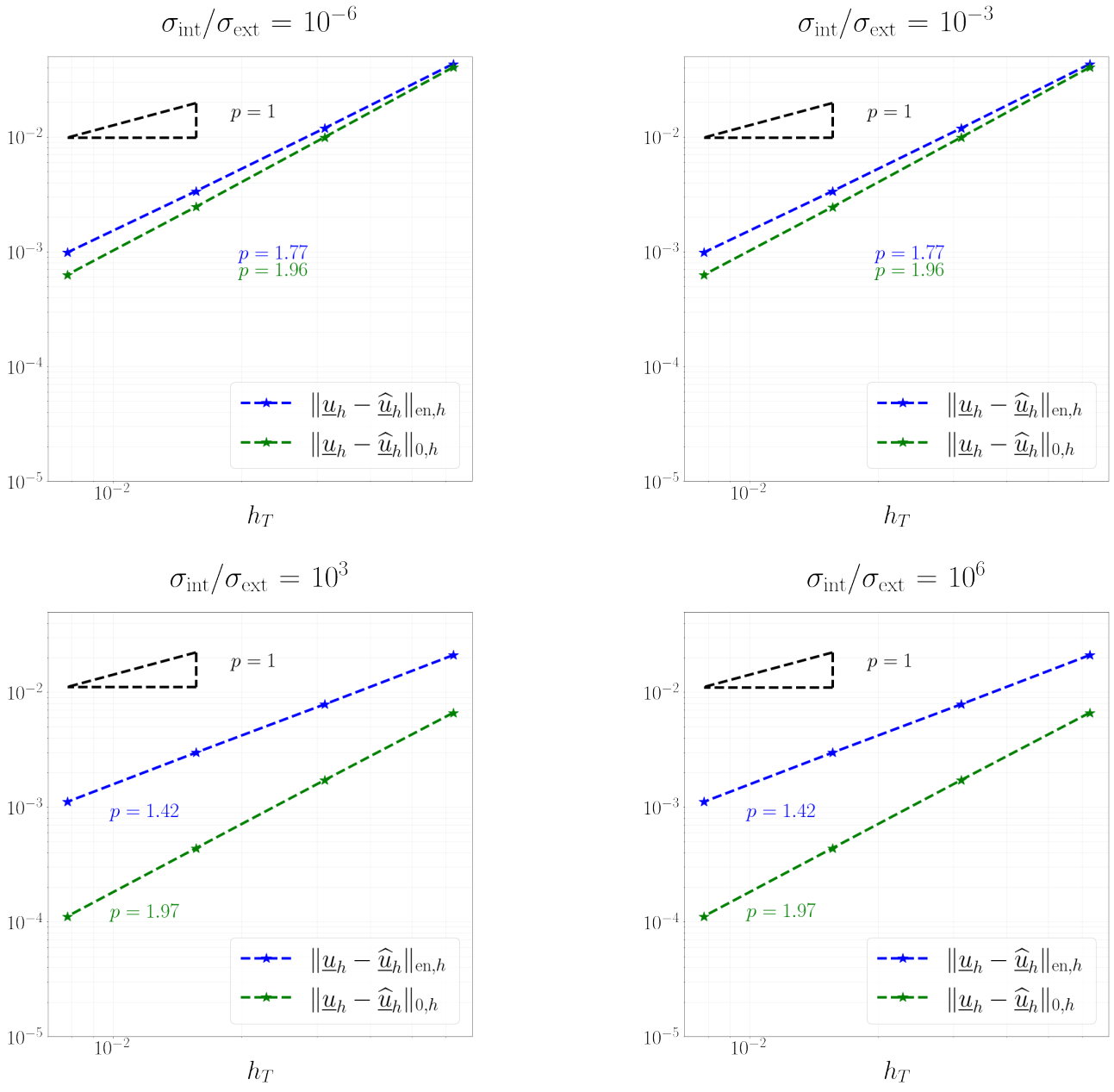


Figure 5.6: Convergence for different values of $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}}$ over a mesh sequence of irregular quadrilaterals as described in Section 5.4.1. Error is normalized with respect to the norm of the reference solution

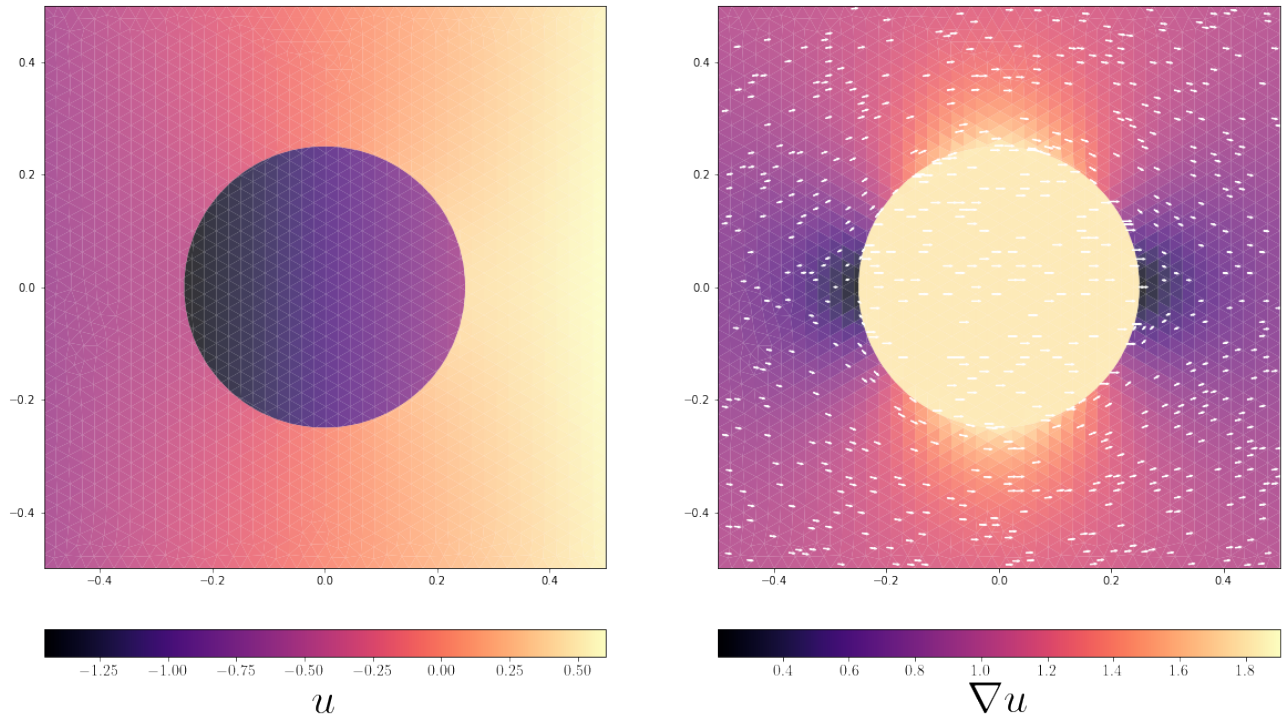


Figure 5.7: The exact solution (5.22) considered in Section 5.4.2 and its gradient with $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 10^{-1}$.

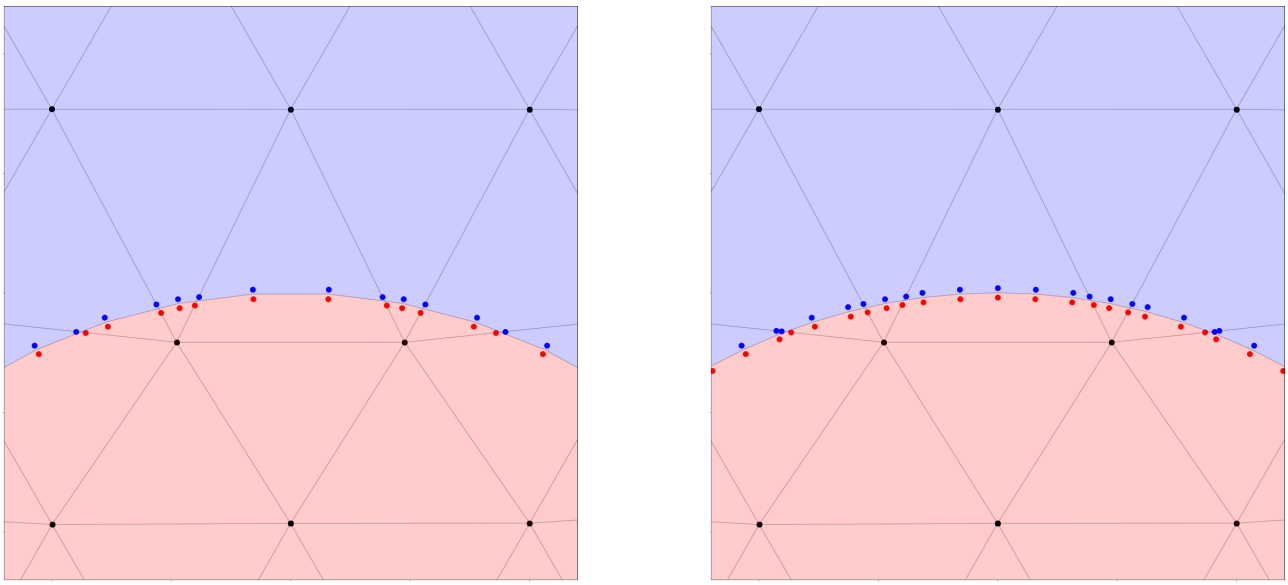


Figure 5.8: Mesh family used in the example form Section 5.4.2. The detail shows new elements generated by cutting a triangular mesh with a polygonal discretization of the interface. Spots represent the distribution of degrees of freedom. The same background triangular mesh can be cut using different refinement levels for the interface. The accuracy of discretization of Γ is therefore arbitrary.

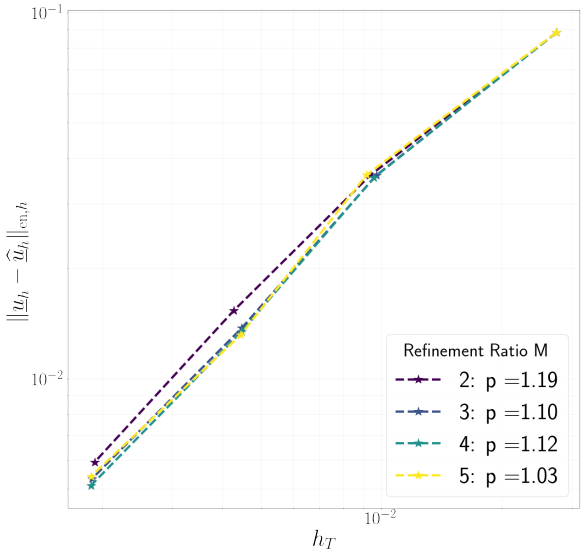


Figure 5.9: Convergence test described in Section 5.4.2, keeping $\sigma_{\mathrm{int}}/\sigma_{\mathrm{ext}} = 0.1$ and varying the refinement ratio M.

5.5.1 Continuous setting

Given a final time $t_{\mathrm{f}} > 0$, a source term $f : (0, t_{\mathrm{f}}] \to \mathbb{R}$, and an initial potential jump J_Γ^0, we consider the problem of finding the time-dependent potential $u = (u_{\mathrm{int}}, u_{\mathrm{ext}})$ with $u_\bullet : \Omega_\bullet \times (0, t_{\mathrm{f}}] \to \mathbb{R}$, $\bullet \in \{\mathrm{int}, \mathrm{ext}\}$ and the interface jump $J_\Gamma : \Gamma \times [0, t_{\mathrm{f}}] \to \mathbb{R}$ such that

$$
\begin{aligned}
-\nabla \cdot (\sigma_\bullet \nabla u_\bullet) &= f && \text{in } \Omega_\bullet \times (0, t_{\mathrm{f}}],\ \bullet \in \{\mathrm{int}, \mathrm{ext}\}, && \text{(5.23a)}\\
[u]_\Gamma &= J_\Gamma && \text{on } \Gamma \times (0, t_{\mathrm{f}}], && \text{(5.23b)}\\
[\sigma \nabla u]_\Gamma \cdot n_\Gamma &= 0 && \text{on } \Gamma \times (0, t_{\mathrm{f}}], && \text{(5.23c)}\\
u_{\mathrm{ext}} &= 0 && \text{on } \partial\Omega \times (0, t_{\mathrm{f}}] && \text{(5.23d)}\\
C \partial_t J_\Gamma &= \sigma_\bullet \nabla u_\bullet \cdot n_\Gamma && \text{on } \Gamma \times (0, t_{\mathrm{f}}],\ \bullet \in \{\mathrm{int}, \mathrm{ext}\} && \text{(5.23e)}\\
J_\Gamma(\cdot, 0) &= J_\Gamma^0 && \text{on } \Gamma, && \text{(5.23f)}
\end{aligned}
$$

with $C > 0$. Problem (5.23) models a situation where two media with electric conductivity respectively equal to σ_{int} and σ_{ext} occupy the regions Ω_{int} and Ω_{ext}. The interface Γ between the two media is characterized by a capacitance C. The variation rate of the charge in the bulk is f and the interface supports no surface charge. The region $\Omega_\bullet$ with $\bullet \in \{\mathrm{int}, \mathrm{ext}\}$ is characterized by an electrostatic potential $u_\bullet$ to determine. The potential is discontinuous at the interface, with a jump J_Γ to determine, and vanishes on the boundary of Ω.

5.5.2 Discrete problem

To adapt the scheme (5.15) to problem (5.23), it is necessary to introduce a time stepping scheme and a suitable discrete space to describe the new variable J_Γ. For the sake of simplicity, we describe the adaption in the case of $k = 0$. Consider $N \geq 1$ time steps with duration $\tau = t_{\mathrm{f}}/N$. For any time-dependent variable w, we introduce the set of time-independent variables $\{w^n\}_{n \leq N}$ such that $w^n(x) = w(x, n\tau)$. An explicit Euler scheme is adopted to replace (5.23e) with:

$$
\frac{C}{\tau}\left(J_\Gamma^{n+1} - J_\Gamma^n\right) = \sigma_{\mathrm{int}} \nabla u_{\mathrm{int}}^n \cdot n_\Gamma = \sigma_{\mathrm{ext}} \nabla u_{\mathrm{ext}}^n \cdot n_\Gamma.
$$

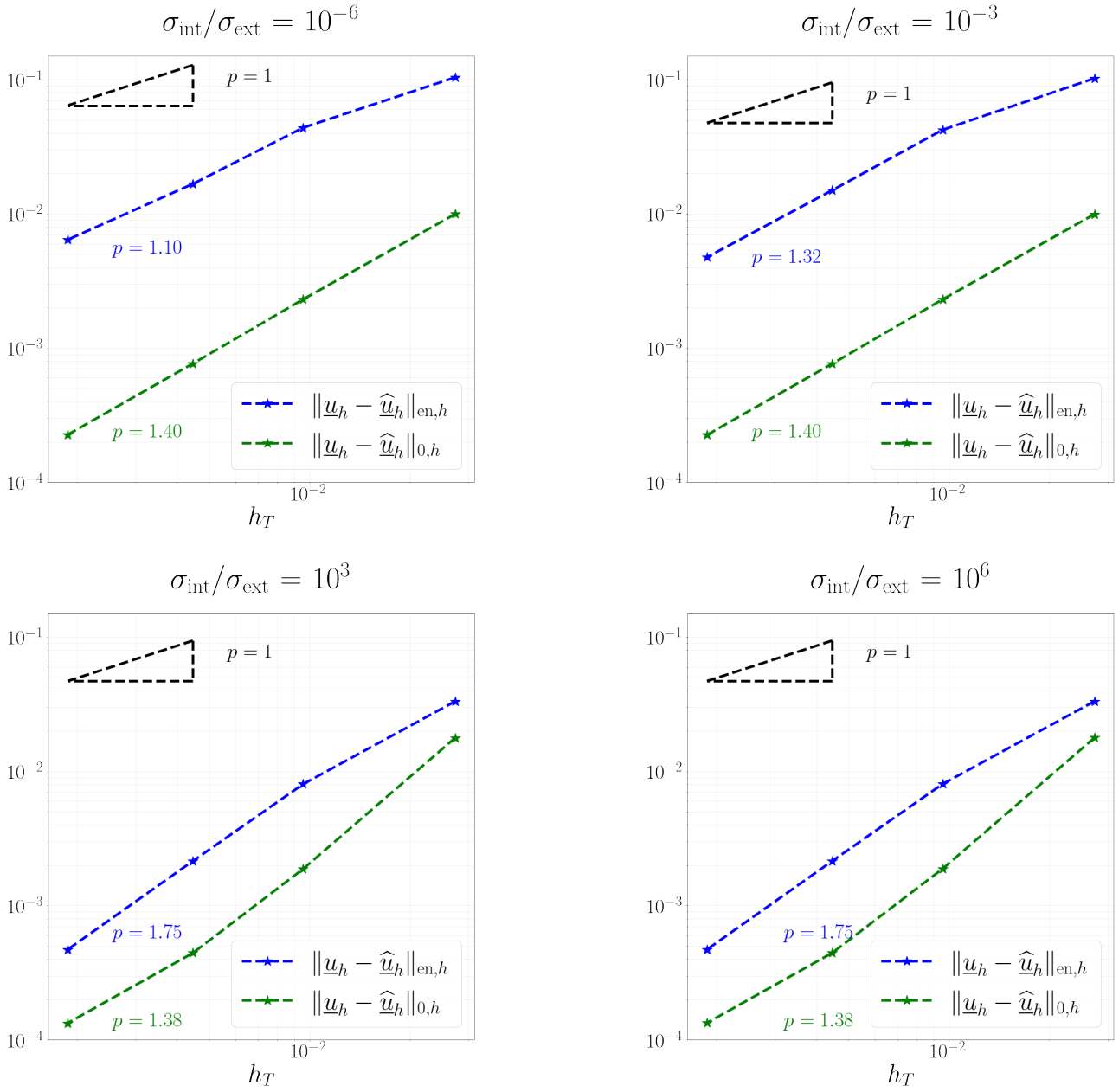


Figure 5.10: Convergence test from Section 5.4.2. With refinement ratio $M = 4$ the test is repeated for several values of $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}}$. Error is normalized with respect to the norm of the reference solution.

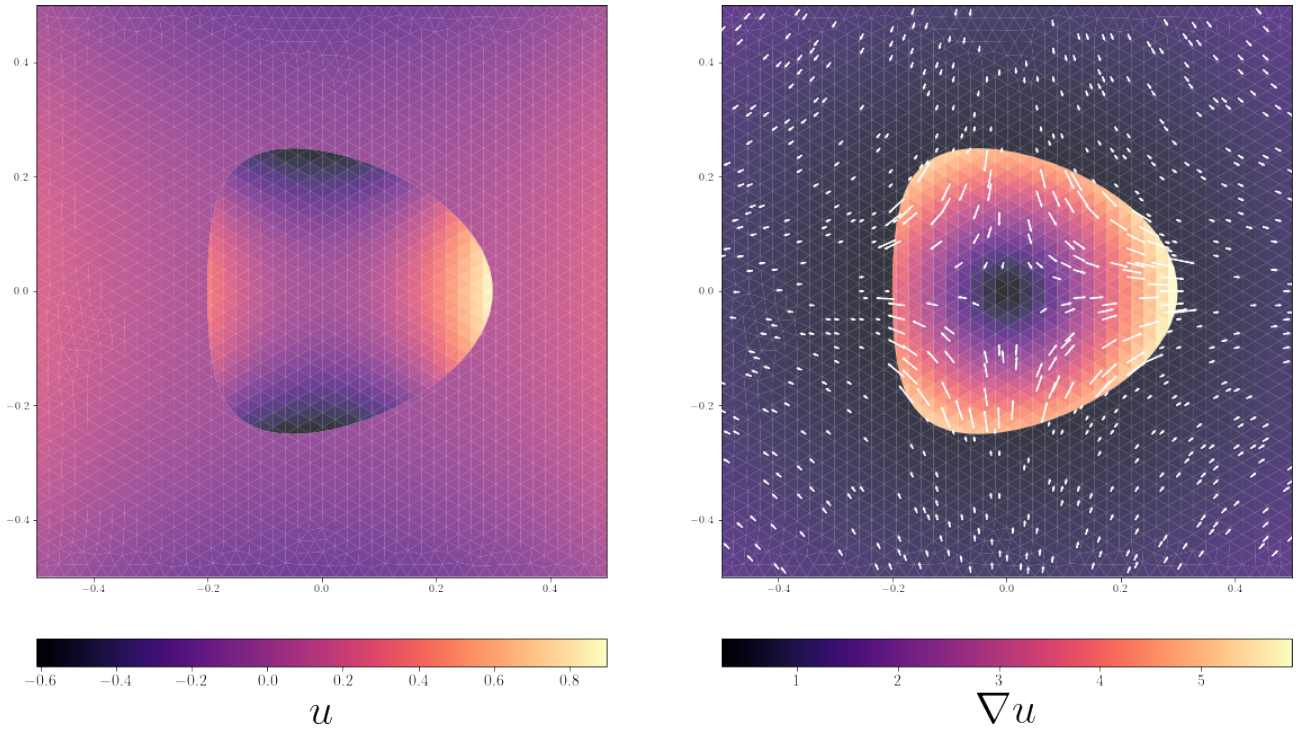


Figure 5.11: The exact solution (5.21) considered in Section 5.4.3 and its gradient with $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 10^{-1}$.

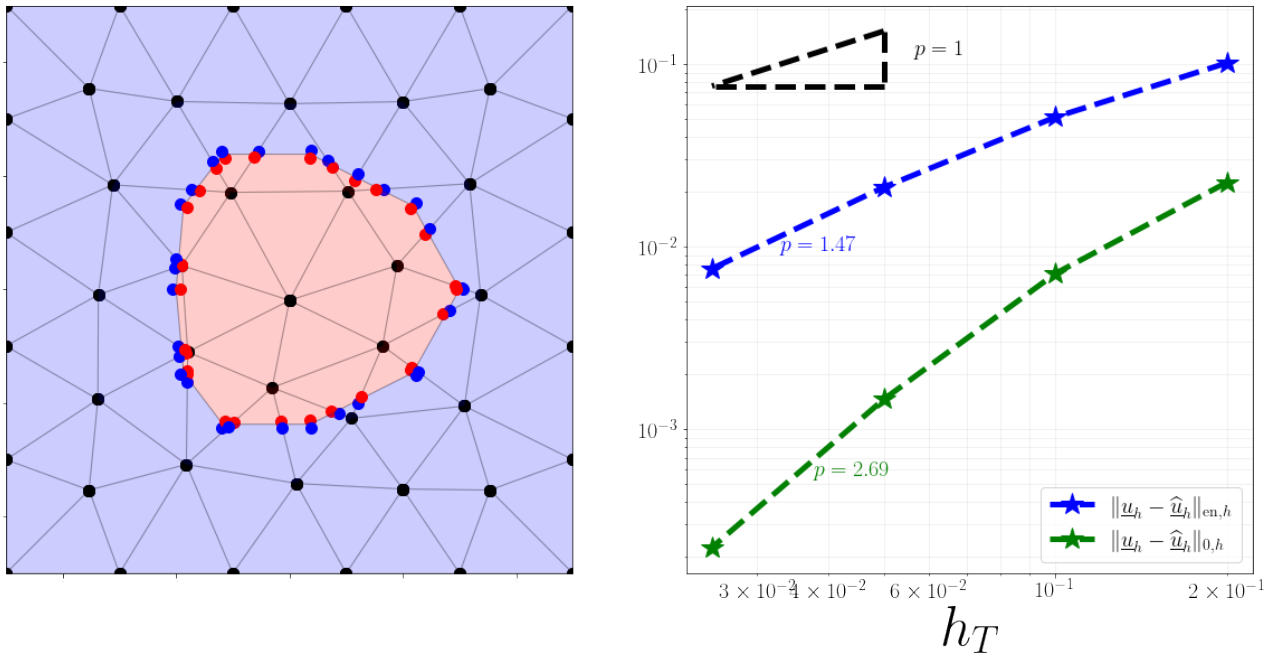


Figure 5.12: Convergence test for the case of a generic interface of Section 5.4.3. On the left, a member of the mesh sequence. The red area represents Ω_{int} and the blue one Ω_{ext}. Dots represent the distribution of degrees of freedom for the first element of the mesh sequence. On nodes belonging to Γ they are doubled. On the right, p indicates the estimated convergence order.

The equation is integrated along Γ after multiplying by a test function $Q^n_\Gamma : \Gamma \to \mathbb{R}$:

$$\frac{C}{\tau}\left(\int_\Gamma J^{n+1}_\Gamma Q^n_\Gamma - \int_\Gamma J^n_\Gamma Q^n_\Gamma\right) = \int_\Gamma (\sigma_{\text{int}} \nabla u^n_{\text{int}} \cdot n_\Gamma) Q^n_\Gamma = \int_\Gamma (\sigma_{\text{ext}} \nabla u^n_{\text{ext}} \cdot n_\Gamma) Q^n_\Gamma.$$

Consider the collection of interface vertices $\mathcal{V}_{\Gamma,h}$ and introduce the space of interface variables:

$$\underline{Z}_{\Gamma,h} \coloneqq \left\{ \underline{p}_{\Gamma,h} = (p_V)_{V \in \mathcal{V}_{\Gamma,h}} \ : \ p_V \in \mathbb{R} \quad \forall V \in \mathcal{V}_{\Gamma,h} \right\}.$$

Given $\underline{p}_{\Gamma,h} \in \underline{Z}_{\Gamma,h}$ and $E \in \mathcal{E}_{\Gamma,h}$, define $p_E \in \mathcal{P}^1(E)$ as the unique affine function that takes the value p_V at every endpoint V of E. Equip $\underline{Z}_{\Gamma,h}$ with the following norm:

$$\|p_{\Gamma,h}\|^2_{0,\Gamma,h} \coloneqq \sum_{E \in \mathcal{E}_{\Gamma,h}} \|p_E\|^2_E.$$

We set $\underline{J}^0_{\Gamma,h} \coloneqq (J^0_\Gamma(x_V))_{V \in \mathcal{V}_{\Gamma,h}}$, with x_V denoting the coordinate vector of the vertex V and, for $n = 0, \dots, N-1$, we advance in time solving the following problem: Find $\underline{J}^{n+1}_{\Gamma,h} \in \underline{Z}_{\Gamma,h}$ such that

$$\frac{C}{\tau}\left(\sum_{E \in \mathcal{E}_{\Gamma,h}} \int_E J^{n+1}_E Q^n_E - \int_E J^n_E Q^n_E\right) = \sum_{E \in \mathcal{E}_{\Gamma,h}} \int_E \{\sigma G_T \underline{u}_T\}_{\lambda,E} \cdot n_\Gamma Q_E \qquad \forall \underline{Q}^n_{\Gamma,h} \in \underline{Z}_{\Gamma,h}. \tag{5.24}$$

Given $\underline{J}^n_{\Gamma,h} \in \underline{Z}^k_{\Gamma,h}$, denote by $\mathcal{M}_h : \underline{Z}^k_{\Gamma,h} \to \underline{V}^0_h$ the operator that associates to a jump $\underline{J}^n_\Gamma$ the solution $\underline{u}_h$ of the stationary problem (5.15). Likewise, call $\mathcal{N}_h : \underline{V}^0_h \times \underline{Z}_{\Gamma,h} \to \underline{Z}_{\Gamma,h}$ the operator that associates to a potential $\underline{u}_h$ and a jump $\underline{J}^n_{\Gamma,h}$ the jump $\underline{J}^{n+1}_{\Gamma,h}$ solution to problem (5.24). Then, the time advancement algorithm for the case of an evolving jump reads: Given $\underline{J}^0_{\Gamma,h}$, for $n = 0, \dots, N-1$, set, in this order,

$$\begin{aligned} \underline{u}^n_h &= \mathcal{M}_h(\underline{J}^n_\Gamma), \\ \underline{J}^{n+1}_{\Gamma,n} &= \mathcal{N}_h(\underline{u}_h, \underline{J}^n_{\Gamma,h}). \end{aligned}$$

5.5.3 Numerical tests

To numerically assess the performance method, we consider a test case with $f = 0$, $\Phi_\Gamma = 0$, $J^0_\Gamma = 0$. This set of conditions is encountered in the description of the electric potential in the context of the Leaky Dielectric Model, and represents a situation where neither the bulk nor the surface support electric charge. Consider a circular interface of radius $R > 0$ immersed in a uniform far field $\epsilon \in \mathbb{R}^2$, such that $\lim_{\|x\| \to \infty} \nabla u = \epsilon$, with $\|\cdot\|$ denoting the Euclidean norm in $\mathbb{R}^2$. With $\bullet \in \{\text{int}, \text{ext}\}$, and centering the reference system at the origin of the circle, the solution reads:

$$\widehat{u}_\bullet(x,t) = \exp\left(-\frac{t}{t_c}\right)(u^0_\bullet(x) - u^\infty_\bullet(x)) + u^\infty_\bullet(x), \tag{5.25}$$

with

$$\begin{aligned} u^0_{\text{int}} &= \left(\frac{2\sigma_{\text{ext}}}{\sigma_{\text{ext}} + \sigma_{\text{int}}}\right) x \cdot \epsilon \qquad & u^0_{\text{ext}} &= \left[1 + \left(\frac{\sigma_{\text{ext}} - \sigma_{\text{int}}}{\sigma_{\text{ext}} + \sigma_{\text{int}}}\right)\frac{R^2}{\|x\|^2}\right] x \cdot \epsilon \\ u^\infty_{\text{int}} &= 0 & u^\infty_{\text{ext}} &= \left(1 + \frac{R^2}{\|x\|^2}\right) x \cdot \epsilon \end{aligned}$$

and $t_c = CR\left(\frac{1}{\sigma_{\text{int}}} + \frac{1}{\sigma_{\text{ext}}}\right)$ (see Figure 5.13). The system evolves from an initial condition with no potential jump at the interface to a condition of electrostatic equilibrium, where the current flow

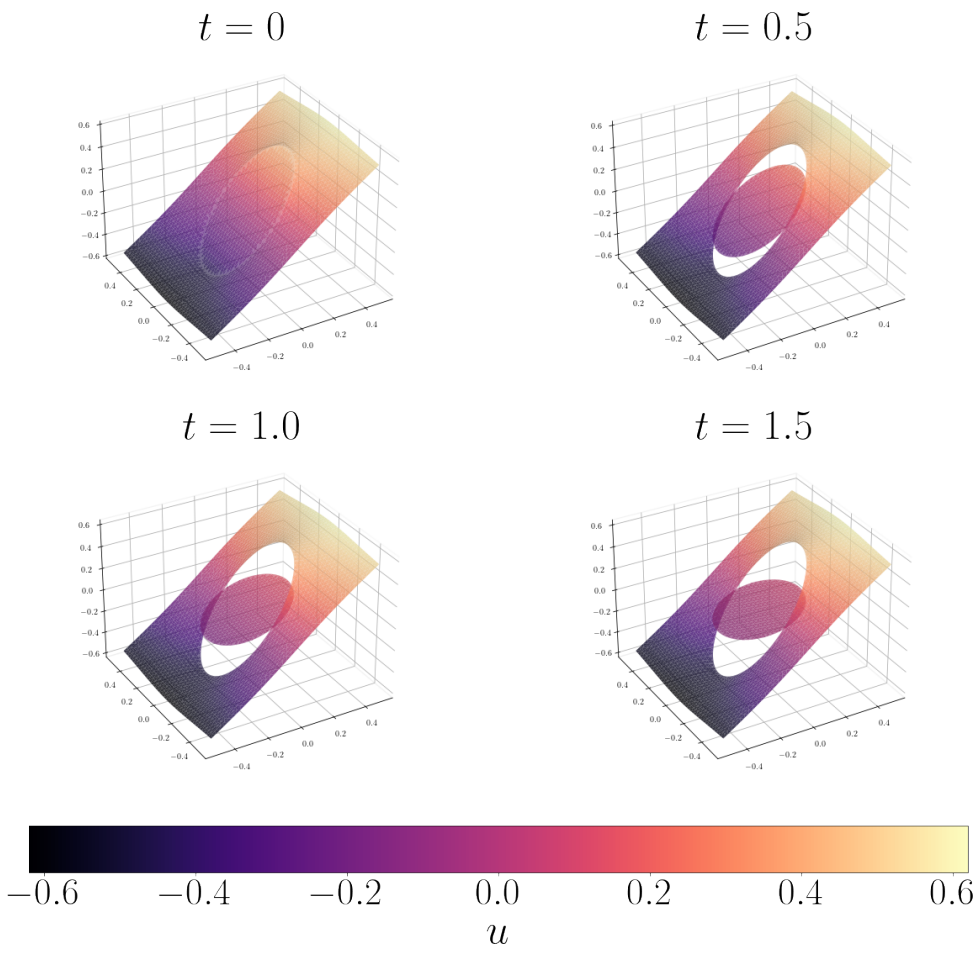


Figure 5.13: Solution 5.25 presented in Section 5.5. Test performed with $R = 1/4$, $\|\epsilon\| = 1$, $\sigma_{\text{int}}/\sigma_{\text{ext}} = 10^{-1}$, and C set such that $t_c = 1$.

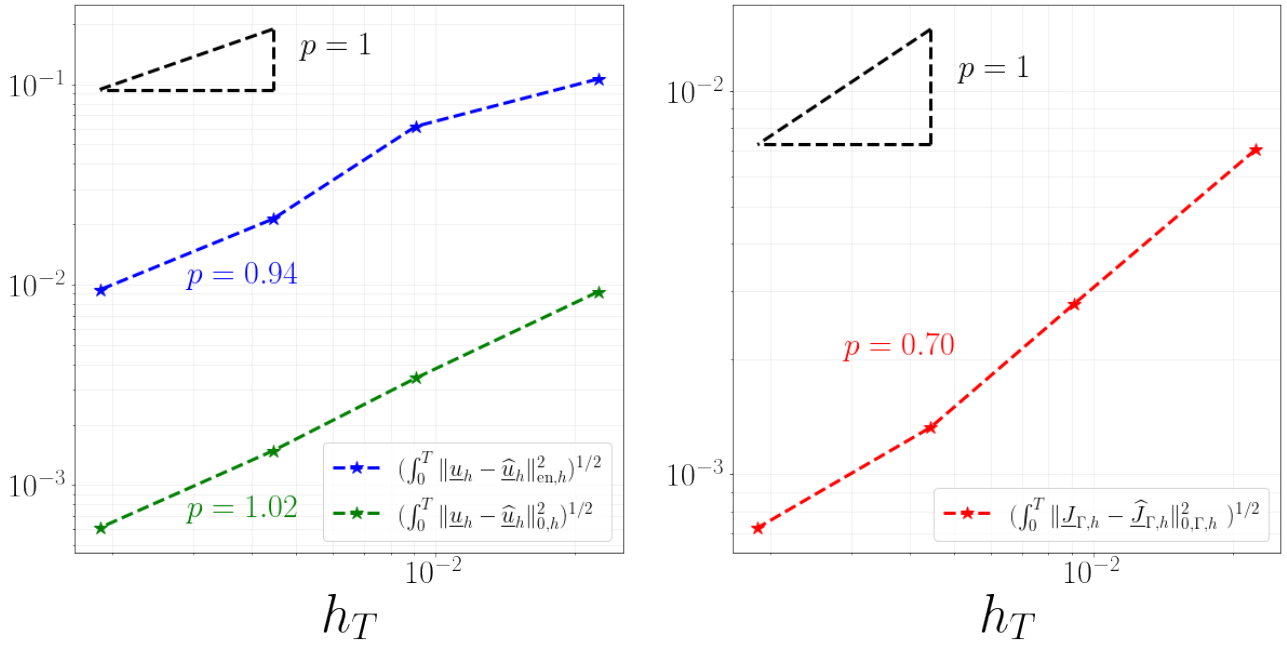


Figure 5.14: Convergence test for the time-dependent scheme used to reproduce Solution 5.25 presented in Section 5.5. The simulation is run with $t_{\mathrm{f}} = 2t_{\mathrm{c}} = 2$.

through the interface $\nabla u_\bullet \cdot n_\Gamma$ with $\bullet \in \{\mathrm{int}, \mathrm{ext}\}$ vanishes from either side as well as the internal electric field.

To monitor the convergence of the scheme, we consider a mesh sequence realized with the same family of triangular background meshes of Section 5.4.2. The interface is refined with a refinement ratio $M = 2$, and a sequence of time steps decreasing by a factor 4 is considered. The results that are shown are obtained with $\eta = 1$. The profile of the error for both the potential $u = (u_{\mathrm{int}}, u_{\mathrm{ext}})$ and the jump J_Γ is displayed in Figure 5.14. Results show that the L^2-temporal norm of the energy error decreases with an order slightly lower than 1. An convergence slightly above 0.5 is observed for the time-space L^2-norm of the interface jump.

5.6 Proofs of the main results

This section collects the proofs of Lemma 7 and Theorem 9.

5.6.1 Proof of Lemma 7

Lemma 12 (Estimate of the consistency interface term)**.** *Then, for all* $(\underline{w}_h, \underline{v}_h) \in \underline{V}_h^k \times \underline{V}_h^k$ *and any real number* $\epsilon > 0$*, it holds, with* N_∂ *as in Lemma 7,*

$$\sum_{E \in \mathcal{E}_{\Gamma,h}} \int_E \{\sigma G_h^k \underline{w}_h\}_{\lambda,E} \cdot n_E [\underline{v}_h]_E \le \epsilon \|\sigma^{1/2} G_h^k \underline{w}_h\|^2_{L^2(\Omega)^2} + \frac{C_{\mathrm{tr}}^2 N_\partial}{4\epsilon} |\underline{v}_h|^2_{\mathrm{J},h}. \tag{5.26}$$

Proof. Let $E \in \mathcal{E}_{\Gamma,h}$. Using a $(2, \infty, 2)$-Hölder inequality, we can write

$$\begin{aligned}
&\int_E \{\sigma G_h^k \underline{w}_h\}_{\lambda,E} \cdot n_E [\underline{v}_h]_E \\
&\quad \le \|\{\sigma G_h^k \underline{w}_h\}_{\lambda,E}\|_{L^2(E)^2} \|n_E\|_{L^\infty(E)^2} \|[\underline{v}_h]_E\|_{L^2(E)} \\
&\quad \overset{(5.11)}{\le} \alpha^{-1/2} \left(\lambda_{\mathrm{int}} \sigma_{\mathrm{int}}^{1/2} \|\sigma_{\mathrm{int}}^{1/2} G^k_{T_{\mathrm{int}}} \underline{w}_{T_{\mathrm{int}}}\|_{L^2(E)^2} + \lambda_{\mathrm{ext}} \sigma_{\mathrm{ext}}^{1/2} \|\sigma_{\mathrm{ext}}^{1/2} G^k_{T_{\mathrm{ext}}} \underline{w}_{T_{\mathrm{ext}}}\|_{L^2(E)^2} \right) \alpha^{1/2} \|[\underline{v}_h]_E\|_{L^2(E)},
\end{aligned}$$

where, in the second equality, we have additionally used the fact that $\|n_E\|_{L^\infty(E)^2} \le 1$. Noticing that, by definition (5.10) of $\lambda_\bullet$ and (5.13) of α, and since $\lambda_\bullet < 1$,

$$\frac{\lambda_\bullet^2 \sigma_\bullet}{\alpha} < \frac{1}{2} \qquad \bullet \in \{\mathrm{int}, \mathrm{ext}\}, \tag{5.27}$$

we can go on writing

$$\begin{aligned}
&\int_E \{\sigma G_h^k \underline{w}_h\}_{\lambda,E} \cdot n_E [\underline{v}_h]_E \\
&\quad \le \frac{1}{\sqrt{2}} \left(\|\sigma_{\mathrm{int}}^{1/2} G_{T_{\mathrm{int}}}^k \underline{w}_{T_{\mathrm{int}}}\|_{L^2(E)^2} + \|\sigma_{\mathrm{ext}}^{1/2} G_{T_{\mathrm{ext}}}^k \underline{w}_{T_{\mathrm{ext}}}\|_{L^2(E)^2} \right) \alpha^{1/2} \|[\underline{v}_h]_E\|_{L^2(E)} \\
&\quad \overset{(5.18),(5.17)}{\le} \frac{C_{\mathrm{tr}}}{\sqrt{2}} \left(\|\sigma_{\mathrm{int}}^{1/2} G_{T_{\mathrm{int}}}^k \underline{w}_{T_{\mathrm{int}}}\|_{L^2(T_{\mathrm{int}})^2} + \|\sigma_{\mathrm{ext}}^{1/2} G_{T_{\mathrm{ext}}}^k \underline{w}_{T_{\mathrm{ext}}}\|_{L^2(T_{\mathrm{ext}})^2} \right) \left(\frac{\alpha}{h_E} \right)^{1/2} \|[\underline{v}_h]_E\|_{L^2(E)}.
\end{aligned}$$

Summing the above inequality over $E \in \mathcal{E}_{\Gamma,h}$, using a Cauchy–Schwarz inequality along with the fact that $(a+b)^2 \le 2(a^2+b^2)$, and recalling the definition (5.17) of $|\cdot|_{\mathrm{J},h}$, we obtain

$$\begin{aligned}
&\sum_{E \in \mathcal{E}_{\Gamma,h}} \int_E \{\sigma G_h^k \underline{w}_h\}_{\lambda,E} \cdot n_E [\underline{v}_h]_E \\
&\quad \le C_{\mathrm{tr}} \left[\sum_{E \in \mathcal{E}_{\Gamma,h}} \left(\|\sigma_{\mathrm{int}}^{1/2} G_{T_{\mathrm{int}}}^k \underline{w}_{T_{\mathrm{int}}}\|_{L^2(T_{\mathrm{int}})^2}^2 + \|\sigma_{\mathrm{ext}}^{1/2} G_{T_{\mathrm{ext}}}^k \underline{w}_{T_{\mathrm{ext}}}\|_{L^2(T_{\mathrm{ext}})^2}^2 \right) \right]^{1/2} |\underline{v}_h|_{\mathrm{J},h} \\
&\quad \le C_{\mathrm{tr}} N_\partial^{1/2} \|\sigma^{1/2} G_h^k \underline{w}_h\|_{L^2(\Omega)^2} |\underline{v}_h|_{\mathrm{J},h} \\
&\quad \le \epsilon \|\sigma^{1/2} G_h^k \underline{w}_h\|_{L^2(\Omega)^2}^2 + \frac{C_{\mathrm{tr}}^2 N_\partial}{4\epsilon} |\underline{v}_h|_{\mathrm{J},h}^2,
\end{aligned}$$

the conclusion being a consequence of the generalized Young's inequality. □

Proof of Lemma 7. Recalling the expression (5.12) of a_h and using (5.26), we obtain, for all $\underline{v}_h \in \underline{V}_{h,0}^k$,

$$\begin{aligned}
a_h(\underline{v}_h, \underline{v}_h) \ge (1-\epsilon) \|\sigma^{1/2} G_h^k \underline{v}_h\|_{L^2(\Omega)^2}^2 \\
+ \sum_{T \in \mathcal{T}_h} \frac{\sigma_T}{h_T} \sum_{E \in \mathcal{E}_T} \|p_T^{k+1} \underline{v}_T - v_{TE}\|_{L^2(E)^2}^2 + \left(\eta - \frac{C_{\mathrm{tr}}^2 N_\partial}{4\epsilon} \right) |\underline{v}_h|_{\mathrm{J},h}^2,
\end{aligned}$$

from which the conclusion readily follows recalling the definition (5.16) of the energy norm. □

5.6.2 Proof of Theorem 9

Lemma 13 (Estimate of the consistency error)**.** *Assume $u \in C^0(\overline{\Omega}_{\mathrm{int}}) \times C^0(\overline{\Omega}_{\mathrm{ext}})$ and define the consistency error linear form $\mathcal{E}_h : \underline{V}_{h,0}^k \to \mathbb{R}$ such that, for all $\underline{v}_h \in \underline{V}_{h,0}^k$,*

$$\mathcal{E}_h(\underline{v}_h) := \ell_h(\underline{v}_h) - a_h(\underline{I}_h^k u, \underline{v}_h).$$

Then, provided that $u \in H^{r+2}(\mathcal{T}_h^{\mathrm{int}}) \times H^{r+2}(\mathcal{T}_h^{\mathrm{ext}})$ for some $r \in \{0, \dots, k\}$, it holds

$$\sup_{\underline{v}_h \in \underline{V}_{h,0}^k \setminus \{\underline{0}\}} \frac{\mathcal{E}_h(\underline{v}_h)}{\|\underline{v}_h\|_{\mathrm{en},h}} \lesssim \overline{\sigma} h^{r+1} |u|_{H^{r+2}(\mathcal{T}_h)}, \tag{5.28}$$

where the hidden constant depends only on the domain, the stability constant C_{stab} in (5.20)*, the polynomial degree k, and the mesh regularity parameter (but is independent of both the meshsize and σ).*

Proof. Let $\underline{v}_h \in \underline{V}_{h,0}^k$. We reformulate the components of the consistency error $\mathcal{E}_h(\underline{v}_h)$ in order to make them comparable. Throughout the proof we let, for the sake of brevity, $\widehat{\underline{u}}_h \coloneqq \underline{I}_h^k u$.

1. *Reformulation of $\ell_h(\underline{v}_h)$.* Recalling (5.2a), $f = -\nabla\cdot(\sigma_\bullet\nabla u_\bullet)$ almost everywhere in $\Omega_\bullet$, $\bullet \in \{\mathrm{int},\mathrm{ext}\}$. We can therefore write for the first term in the right-hand side of (5.14):

$$\begin{aligned}
\sum_{T\in\mathcal{T}_h}\int_T f p_T^{k+1}\underline{v}_T &= -\sum_{T\in\mathcal{T}_h}\int_T \nabla\cdot(\sigma_T\nabla u)p_T^{k+1}\underline{v}_T\\
&= \sum_{T\in\mathcal{T}_h}\int_T \sigma_T\nabla u\cdot\nabla p_T^{k+1}\underline{v}_T - \sum_{T\in\mathcal{T}_h}\sum_{E\in\mathcal{E}_T}\omega_{TE}\int_E(\sigma_T\nabla u\cdot n_E)p_T^{k+1}\underline{v}_T\\
&= \sum_{T\in\mathcal{T}_h}\int_T \sigma_T\nabla u\cdot\nabla p_T^{k+1}\underline{v}_T - \sum_{T\in\mathcal{T}_h}\sum_{E\in\mathcal{E}_T}\omega_{TE}\int_E(\sigma_T\nabla u\cdot n_E)(p_T^{k+1}\underline{v}_T - v_{TE})\\
&\quad \underbrace{-\sum_{T\in\mathcal{T}_h}\sum_{E\in\mathcal{E}_T}\omega_{TE}\int_E(\sigma_T\nabla u\cdot n_E)v_{TE}}_{\mathfrak{T}},
\end{aligned}\tag{5.29}$$

where we have used an integration by parts inside each element in the second equality and inserted $\pm v_{TE}$ into the boundary term to conclude. Let us focus on the last term. Rearranging the sums, we have

$$\begin{aligned}
\mathfrak{T} &= \sum_{E\in\mathcal{E}_h}\sum_{T\in\mathcal{T}_E}\omega_{TE}\int_E(\sigma_T\nabla u\cdot n_E)v_{TE}\\
&= \sum_{E\in\mathcal{E}_{\Gamma,h}}\int_E\left[(\sigma_{T_\mathrm{int}}\nabla u_\mathrm{int}\cdot n_E)v_{T_\mathrm{int}E} - (\sigma_{T_\mathrm{ext}}\nabla u_\mathrm{ext}\cdot n_E)v_{T_\mathrm{ext}E}\right],
\end{aligned}$$

where we have used the fact that both v_{TE} and the normal trace of $\sigma\nabla u$ are single-valued on mesh edges internal to each subdomain along with the fact that $v_{TE} = 0$ on edges contained in $\partial\Omega$ in the second equality. We next notice that, given four real numbers a_1, a_2, b_1, and b_2, since $\lambda_\mathrm{int}+\lambda_\mathrm{ext} = 1$ (cf. (5.10),

$$a_1b_1 - a_2b_2 = (\lambda_\mathrm{int}a_1 + \lambda_\mathrm{ext}a_2)(b_1-b_2) + (a_1-a_2)(\lambda_\mathrm{ext}b_1 + \lambda_\mathrm{int}b_2).$$

Applying this formula with $(a_1,a_2,b_1,b_2) = (\sigma_\mathrm{int}\nabla u_\mathrm{int}\cdot n_E, \sigma_\mathrm{ext}\nabla u_\mathrm{ext}\cdot n_E, v_{T_\mathrm{int}E}, v_{T_\mathrm{ext}E})$, recalling the definitions of the interface trace operators of Section 5.3.3, and using the fact that, by (5.2c), $[\sigma\nabla u]_E\cdot n_E = \Phi_\Gamma$ almost everywhere on Γ, we infer that

$$\mathfrak{T} = \sum_{E\in\mathcal{E}_{\Gamma,h}}\int_E\{\sigma\nabla u\}_{\lambda,E}\cdot n_E[\underline{v}_h]_E + \sum_{E\in\mathcal{E}_{\Gamma,h}}\int_E\Phi_\Gamma\{\underline{v}_h\}_{\overline{\lambda},E}.$$

Plugging this expression into (5.29) and substituting into the expression (5.14) of ℓ_h, we arrive at

$$\begin{aligned}
\ell_h(\underline{v}_h) &= \sum_{T\in\mathcal{T}_h}\int_T\sigma_T\nabla u\cdot\nabla p_T^{k+1}\underline{v}_T - \sum_{T\in\mathcal{T}_h}\sum_{E\in\mathcal{E}_T}\omega_{TE}\int_E(\sigma_T\nabla u\cdot n_E)(p_T^{k+1}\underline{v}_T - v_{TE})\\
&\quad - \sum_{E\in\mathcal{E}_{\Gamma,h}}\int_E\{\sigma\nabla u\}_{\lambda,E}\cdot n_E[\underline{v}_h]_E + \eta\sum_{E\in\mathcal{E}_{\Gamma,h}}\frac{\alpha}{h_E}\int_E J_\Gamma[\underline{v}_h]_E.
\end{aligned}\tag{5.30}$$

2. *Reformulation of $a_h(\widehat{\underline{u}}_h,\underline{v}_h)$.* Let $T\in\mathcal{T}_h$. Writing (5.4) for $\tau = \sigma_T G_T^k\widehat{\underline{u}}_T$ and rearranging, we obtain

$$\int_T\sigma_T G_T^k\widehat{\underline{u}}_T\cdot G_T^k\underline{v}_T = \int_T\sigma_T G_T^k\widehat{\underline{u}}_T\cdot\nabla p_T^{k+1}\underline{v}_T - \sum_{E\in\mathcal{E}_T}\omega_{TE}\int_E(\sigma_T G_T^k\widehat{\underline{u}}_T\cdot n_E)(p_T^{k+1}\underline{v}_T - v_{TE}).$$

Substituting this expression in the definition (5.12) of a_h written for $\underline{w}_h = \widehat{\underline{u}}_h$, we obtain

$$\begin{aligned} a_h(\widehat{\underline{u}}_h, \underline{v}_h) \coloneqq & \sum_{T\in\mathcal{T}_h}\int_T \sigma_T G_T^k\widehat{\underline{u}}_T\cdot\nabla p_T^{k+1}\underline{v}_T - \sum_{T\in\mathcal{T}_h}\sum_{E\in\mathcal{E}_T}\omega_{TE}\int_E(\sigma_T G_T^k\widehat{\underline{u}}_T\cdot n_E)(p_T^{k+1}\underline{v}_T - v_{TE}) \\ & + \sum_{T\in\mathcal{T}_h}\frac{\sigma_T}{h_T}\sum_{E\in\mathcal{E}_T}\int_E(p_T^{k+1}\widehat{\underline{u}}_T - \widehat{u}_{TE})(p_T^{k+1}\underline{v}_T - v_{TE}) \\ & - \sum_{E\in\mathcal{E}_{\Gamma,h}}\int_E\{\sigma G_h^k\widehat{\underline{u}}_h\}_{\lambda,E}\cdot n_E[\underline{v}_h]_E + \eta\sum_{E\in\mathcal{E}_{\Gamma,h}}\frac{\alpha}{h_E}\int_E[\widehat{\underline{u}}_h]_E[\underline{v}_h]_E. \end{aligned} \tag{5.31}$$

3. *Estimate of the consistency error.* Subtracting (5.31) from (5.30), we arrive at the following decomposition of the consistency error:

$$\mathcal{E}_h(\underline{v}_h) = \mathfrak{T}_1 + \cdots + \mathfrak{T}_5 \tag{5.32}$$

with

$$\begin{aligned} \mathfrak{T}_1 &\coloneqq \sum_{T\in\mathcal{T}_h}\int_T\sigma_T(\nabla u - G_T^k\widehat{\underline{u}}_T)\cdot\nabla p_T^{k+1}\underline{v}_T, \\ \mathfrak{T}_2 &\coloneqq \sum_{T\in\mathcal{T}_h}\sum_{E\in\mathcal{E}_T}\omega_{TE}\int_E\sigma_T(G_T^k\widehat{\underline{u}}_T - \nabla u)\cdot n_E(p_T^{k+1}\underline{v}_T - v_{TE}), \\ \mathfrak{T}_3 &\coloneqq \sum_{T\in\mathcal{T}_h}\frac{\sigma_T}{h_T}\sum_{E\in\mathcal{E}_T}\int_E(p_T^{k+1}\widehat{\underline{u}}_T - \widehat{u}_{TE})(p_T^{k+1}\underline{v}_T - v_{TE}), \\ \mathfrak{T}_4 &\coloneqq \sum_{E\in\mathcal{E}_{\Gamma,h}}\int_E\{\sigma(G_h^k\widehat{\underline{u}}_h - \nabla u)\}_{\lambda,E}\cdot n_E[\underline{v}_h]_E, \\ \mathfrak{T}_5 &\coloneqq \eta\sum_{E\in\mathcal{E}_{\Gamma,h}}\frac{\alpha}{h_E}\int_E(J_\Gamma - [\widehat{\underline{u}}_h]_E)[\underline{v}_h]_E. \end{aligned}$$

We next proceed to estimate the above terms. Using Cauchy–Schwarz inequalities along with the fact that $\sigma_T \le \overline{\sigma}$ for all $T\in\mathcal{T}_h$, we have for the first term

$$\begin{aligned} \mathfrak{T}_1 &\le \overline{\sigma}^{1/2}\left(\sum_{T\in\mathcal{T}_h}\|\nabla u - G_T^k\widehat{\underline{u}}_T\|_{L^2(T)^2}^2\right)^{1/2}\left(\sum_{T\in\mathcal{T}_h}\sigma_T\|\nabla p_T^{k+1}\underline{v}_T\|_{L^2(T)^2}^2\right)^{1/2} \\ &\overset{(5.7),(5.5),(5.16)}{\lesssim} \overline{\sigma}^{1/2}h^{r+1}|u|_{H^{r+2}(\mathcal{T}_h)}\|\underline{v}_h\|_{\mathrm{en},h}. \end{aligned} \tag{5.33}$$

For the second term, we use on each edge a $(2,\infty,2)$-Hölder inequality on the integral, the fact that $\|n_E\|_{L^\infty(E)^2}\le 1$ along with $\sigma_T\le\overline{\sigma}$, and a Cauchy–Schwarz inequality on the sums to write

$$\begin{aligned} \mathfrak{T}_2 &\le \overline{\sigma}^{1/2}\left(\sum_{T\in\mathcal{T}_h}h_T\|G_T^k\widehat{\underline{u}}_T - \nabla u\|_{L^2(\partial T)^2}^2\right)^{1/2}\left(\sum_{T\in\mathcal{T}_h}\frac{\sigma_T}{h_T}\sum_{E\in\mathcal{E}_T}\|p_T^{k+1}\underline{v}_T - v_{TE}\|_{L^2(E)}^2\right)^{1/2} \\ &\overset{(5.7),(5.16)}{\lesssim} \overline{\sigma}^{1/2}h^{r+1}|u|_{H^{r+2}(\mathcal{T}_h)}\|\underline{v}_h\|_{\mathrm{en},h}. \end{aligned} \tag{5.34}$$

Cauchy–Schwarz inequalities along with $\sigma_T \le \overline{\sigma}$ yield for the third term

$$\begin{aligned}\mathfrak{T}_3 &\le \overline{\sigma}^{1/2}\left(\sum_{T\in\mathcal{T}_h} h_T^{-1}\sum_{E\in\mathcal{E}_T}\|p_T^{k+1}\underline{\widehat{u}}_T - \widehat{u}_{TE}\|_{L^2(E)}^2\right)^{1/2}\left(\sum_{T\in\mathcal{T}_h}\frac{\sigma_T}{h_T}\sum_{E\in\mathcal{E}_T}\|p_T^{k+1}\underline{v}_T - v_{TE}\|_{L^2(E)}^2\right)^{1/2}\\ &\overset{(5.16)}{\lesssim} \overline{\sigma}^{1/2}\left[\sum_{T\in\mathcal{T}_h} h_T^{-1}\sum_{E\in\mathcal{E}_T}\left(\|p_T^{k+1}\underline{\widehat{u}}_T - \gamma_{TE}u\|_{L^2(E)}^2 + \|\gamma_{TE}u - \widehat{u}_{TE}\|_{L^2(E)}^2\right)\right]^{1/2}\|\underline{v}_h\|_{\mathrm{en},h}\\ &\overset{(5.8),(5.6)}{\lesssim} \overline{\sigma}^{1/2}h^{r+1}|u|_{H^{r+2}(\mathcal{T}_h)}\|\underline{v}_h\|_{\mathrm{en},h}.\end{aligned} \tag{5.35}$$

where, in the second inequality, we have additionally used the fact that $(a+b)^2 \le 2(a^2+b^2)$ after inserting $\pm\gamma_{TE}u$ (the trace of $u_{|T}$ on E) inside the norm. To estimate the fourth term, we start with $(2,\infty,2)$-Hölder inequalities on the integrals and Cauchy–Schwarz inequalities on the sums and recall the definition (5.17) of $|\cdot|_{\mathrm{J},h}$ to write

$$\mathfrak{T}_4 \le \left(\sum_{E\in\mathcal{E}_{\Gamma,h}}\frac{h_E}{\alpha}\|\{\sigma(G_h^k\underline{\widehat{u}}_h - \nabla u)\}_{\lambda,E}\|_{L^2(E)^2}^2\right)^{1/2}|\underline{v}_h|_{\mathrm{J},h}. \tag{5.36}$$

Let now $E\in\mathcal{E}_{\Gamma,h}$ and, using the inequality $(a+b)^2 \le 2(a^2+b^2)$, write

$$\begin{aligned}&\alpha^{-1}\|\{\sigma(G_h^k\underline{\widehat{u}}_h - \nabla u)\}_{\lambda,E}\|_{L^2(E)^2}^2\\ &\quad\overset{(5.11)}{\le} 2\frac{\lambda_{\mathrm{int}}^2\sigma_{\mathrm{int}}}{\alpha}\|\sigma_{\mathrm{int}}^{1/2}(G_{T_{\mathrm{int}}}^k\underline{\widehat{u}}_T - \nabla u_{\mathrm{int}})\|_{L^2(E)^2}^2 + 2\frac{\lambda_{\mathrm{ext}}^2\sigma_{\mathrm{ext}}}{\alpha}\|\sigma_{\mathrm{ext}}^{1/2}(G_{T_{\mathrm{ext}}}^k\underline{\widehat{u}}_T - \nabla u_{\mathrm{ext}})\|_{L^2(E)^2}^2\\ &\quad\overset{(5.27)}{\le} \|\sigma_{\mathrm{int}}^{1/2}(G_{T_{\mathrm{int}}}^k\underline{\widehat{u}}_T - \nabla u_{\mathrm{int}})\|_{L^2(E)^2}^2 + \|\sigma_{\mathrm{ext}}^{1/2}(G_{T_{\mathrm{ext}}}^k\underline{\widehat{u}}_T - \nabla u_{\mathrm{ext}})\|_{L^2(E)^2}^2\\ &\quad\overset{(5.7)}{\lesssim} \overline{\sigma}^{1/2}h^{r+1}|u|_{H^{r+2}(\mathcal{T}_E)}.\end{aligned}$$

Plugging the above estimate into (5.36) and recalling the definition (5.16) of the energy norm, we conclude that

$$\mathfrak{T}_4 \lesssim \overline{\sigma}^{1/2}h^{r+1}|u|_{H^{r+2}(\mathcal{T}_h)}\|\underline{v}_h\|_{\mathrm{en},h}. \tag{5.37}$$

Moving to the fifth term, we recall that, by (5.2b), $J_\Gamma = [u]_\Gamma$ almost everywhere on Γ and use Cauchy–Schwarz inequalities along with the fact that $\alpha \le \frac{2\sigma_{\mathrm{int}}\sigma_{\mathrm{ext}}}{2\min\{\sigma_{\mathrm{int}},\sigma_{\mathrm{ext}}\}} \le \overline{\sigma}$ and the definition (5.17) of the $|\cdot|_{\mathrm{J},h}$-seminorm to write

$$\begin{aligned}\mathfrak{T}_5 &\le \eta\overline{\sigma}^{1/2}\left(\sum_{E\in\mathcal{E}_{\Gamma,h}} h_E^{-1}\|[u]_\Gamma - [\underline{\widehat{u}}_h]_E\|_{L^2(E)}^2\right)^{1/2}|\underline{v}_h|_{\mathrm{J},h}\\ &\overset{(5.1),(5.9),(5.16)}{\lesssim} \overline{\sigma}^{1/2}\left[\sum_{E\in\mathcal{E}_{\Gamma,h}} h_E^{-1}\left(\|\gamma_{T_{\mathrm{int}}E}u - \widehat{u}_{T_{\mathrm{int}}E}\|_{L^2(E)}^2 + \|\gamma_{T_{\mathrm{ext}}E}u - \widehat{u}_{T_{\mathrm{ext}}E}\|_{L^2(E)}^2\right)\right]^{1/2}\|\underline{v}_h\|_{\mathrm{en},h}\\ &\overset{(5.6)}{\lesssim} \overline{\sigma}^{1/2}h^{r+1}|u|_{H^{r+2}(\mathcal{T}_h)}\|\underline{v}_h\|_{\mathrm{en},h}\end{aligned} \tag{5.38}$$

Plugging the estimates (5.33), (5.34), (5.35), (5.37), and (5.38) into (5.32), (5.28) follows. □

Proof of Theorem 9. Straightforward consequence of the Third Strang Lemma Di Pietro and Droniou 2018, Theorem 10 accounting for Lemmas 7 and 13 above. □

CHAPTER

SIX

CAPSULE DYNAMICS BASED ON HYBRID MESHES

This chapter presents a possible future direction of the ideas presented in chapter 5. This is part of an ongoing effort aimed at integrating electricity calculations based on the Discrete De Rham method into a solver taking into account the movement of the membrane. The main challenge consists in proposing a discretization of the equations describing the flow which can leverage hybrid meshes generated by cutting a background mesh along the surface. The originality of this approach with respect to existing solutions would be the possibility to treat interface conditions seamlessly by conforming the interface to the background mesh, while avoiding the overhead introduced by the adaptation of simplicial meshes in the traditional way.

In Section 6.1, the reference model of capsule dynamics is presented. Importantly, velocity and pressure are represented as the solution of a problem involving Stokes equations coupled with interface conditions enforcing the fluid-structure interaction. A numerical solution of interface Stokes equations based on hybrid meshes is treated in section 6.2. An algorithm for the simulation of capsule dynamics is exposed in Section 6.3. The chapter is closed by section 6.4, presenting an early demo, where the performance of the proposed algorithm is compared with `YALES2BIO`.

6.1 Capsule Model

The simulation is based on a reference bidimensional model of capsule dynamics which can be found, among others, in [Breyiannis and Pozrikidis 2000]. Let a capsule be represented as a region of the plane $\mathbb{R}^2$ bounded by a closed curve Γ, representing the membrane. The movement of the capsule is described with respect to a reference configuration Γ_0, equipped with a parametrization $s \in [0,1] \to \Gamma_0$. The trajectory is represented by a map $x : [0,1] \times [0,\infty] \to \mathbb{R}^2$ such that $x(s,t)$ represents the position of the membrane point parametrized by s at time t.

The membrane of the capsule is subject to a stress $t_\Gamma : \Gamma \to \mathbb{R}^2$. Membrane stress reflects the response of the membrane to bending and strain. The response to bending is modeled introducing Helfrich's bending energy [Helfrich 1973b], while a linear elasticity model is introduced to describe the response to strain. Considering the bending stiffness k_{bnd} and the strain stiffness k_{str}, the total stress t_Γ reads:

$$t_\Gamma = t_{\text{bnd}} + t_{\text{str}}, \tag{6.1}$$

with:

$$\begin{aligned} t_{bnd} &= k_{bnd}(\frac{1}{2}k^3 + \Delta_{LB}k)n_\Gamma, \\ t_{str} &= -\partial_s\big(k_{str}(\|\partial_s x\| - 1)\partial_s x\big), \end{aligned} \tag{6.2}$$

where k is the membrane curvature, Δ_{LB} is the Laplace-Beltrami operator on Γ, and n_Γ the normal vector pointing out of the capsule.

The membrane moves with a velocity $V : \Gamma \to \mathbb{R}^2$, such that $V(x) = \partial_t x(x^{-1})$. The capsule is advected by a fluid, surrounding it from both sides. The velocity of the fluid is represented as the vector field $v : \mathbb{R}^2 \to \mathbb{R}^2$. The membrane is advected by the flow, such that:

$$V(x) = v(x) \qquad \forall x \in \Gamma. \tag{6.3}$$

The fluid is assumed to be incompressible and Newtonian. Consider the fluid pressure $p : \mathbb{R}^2 \setminus \Gamma \to \mathbb{R}$ and let $\mu : \mathbb{R}^2 \setminus \Gamma \to \mathbb{R}^2$ represent the dynamical viscosity of the fluid, possibly discontinuous across the interface. The stress tensor $\sigma : \mathbb{R}^2 \setminus \Gamma \to \mathbb{R}^{2,2}$ is defined in the fluid as:

$$\sigma = \mu(\nabla v + \nabla^T v) - pI, \tag{6.4}$$

where I is the identity tensor. The flow is assumed to respect Stokes equations in the bulk:

$$\begin{cases} \nabla \cdot \sigma = 0 & \text{in } \mathbb{R}^2 \setminus \Gamma \quad (6.5a) \\ \nabla \cdot v = 0 & \text{in } \mathbb{R}^2 \setminus \Gamma. \quad (6.5b) \end{cases}$$

Flow continuity and balance of forces acting on the membrane are expressed by the interface conditions:

$$\begin{cases} [v] = 0 & \text{on } \Gamma \quad (6.6a) \\ [\sigma] \cdot n_\Gamma = t_\Gamma & \text{on } \Gamma, \quad (6.6b) \end{cases}$$

where $[\cdot]$ denotes a jump across the interface.

6.2 Numerical discretization of an interface Stokes problem

Given the position of the membrane Γ, the flow is determined solving equations 6.5 coupled with interface conditions 6.6. Let $\Omega \in \mathbb{R}^2$ be the domain where the flow is defined, split by Γ into the subdomains Ω_{int} and Ω_{ext}. Supposing that $g : \partial\Omega \to \mathbb{R}^2$ represents the value of velocity known along $\partial\Omega$, the flow variables (p, v) are solution to the boundary value problem:

$$\begin{cases} \nabla \cdot \sigma = 0 & \text{in } \Omega_\bullet,\ \bullet \in \{\text{int}, \text{ext}\} \quad (6.7a) \\ \sigma = [\mu(\nabla v + \nabla^T v) - pI] & \text{in } \Omega_\bullet,\ \bullet \in \{\text{int}, \text{ext}\} \quad (6.7b) \\ \nabla \cdot \boldsymbol{v} = 0 & \text{in } \Omega_\bullet,\ \bullet \in \{\text{int}, \text{ext}\} \quad (6.7c) \\ [v] = 0 & \text{on } \Gamma \quad (6.7d) \\ [\sigma] \cdot n_\Gamma = t_\Gamma & \text{on } \Gamma \quad (6.7e) \\ \int_\Omega p = 0. & \quad (6.7f) \\ v = g & \text{on } \partial\Omega \quad (6.7g) \end{cases}$$

The numerical solution to problem 6.5 is based on a Hybrid High Order method (HHO). The method has been introduced in [Di Pietro, Ern, and Lemaire 2014b] and [Di Pietro and Ern 2015] and provides a powerful framwork to discretize partial differential equations on general meshes featuring element with an arbitrary number of faces. Applications to Stokes equations can be found in [Aghili, Boyaval, and Di Pietro 2015], [Di Pietro et al. 2016] and [Castanon Quiroz and Di Pietro 2020].

To introduce the method, problem 6.7 is recast in a weak formulation. For the sake of exposition, the numerical scheme is introduced considering homogenous Dirichlet boundary conditions for velocity. Consider the space of functions with square integrable derivatives up to first order and values in $\mathbb{R}^2$:

$$U \coloneqq H_0^1(\Omega)^2 \tag{6.8}$$

and the space:

$$P \coloneqq \{q \in L^2(\Omega) : \int_\Omega q = 0\}. \tag{6.9}$$

Introduce the forms $a : U \times U \to \mathbb{R}$ and $b : U \times P \to \mathbb{R}$ such that:

$$a(w, v) \coloneqq \sum_{\bullet \in \{\text{int,ext}\}} \int_{\Omega_\bullet} 2\mu_\bullet \nabla_s(w) : \nabla_s(v), \quad b(w, q) \coloneqq -\int_\Omega q \nabla \cdot w, \tag{6.10}$$

where $\nabla_s(w) \coloneqq \frac{1}{2}(\nabla w + \nabla^T w)$ is the symmetric part of the gradient. With $t_\Gamma \in L^2(\Gamma)^2$, the problem consists in finding $(u, p) \in U \times P$ such that:

$$\begin{aligned} a(u, v) + b(v, p) &= -(t_\Gamma, v)_{L^2} \qquad && \forall v \in U \\ -b(u, q) &= 0 && \forall q \in L^2(\Omega). \end{aligned} \tag{6.11}$$

The method is based on a polygonal discretization Γ_h of Γ, and a polygonal mesh $(\mathcal{T}_h, \mathcal{E}_h)$ of Ω conforming to Γ_h, according the definitions given in subsection 5.2.1. The HHO method is based on spaces of degrees of freedom associated to elements and edges of the mesh. With $k \in \mathbb{N}$, consider the following discrete spaces:

$$\begin{aligned} \underline{U}_h^k &= \left\{ \underline{v}_h = ((v_T)_{T \in \mathcal{T}_h}, (v_E)_{E \in \mathcal{E}_h}) : v_T \in \mathcal{P}^k(T)^d, v_E \in \mathcal{P}^k(E)^d \right\} \\ P_h^k &= \left\{ p \in \mathcal{P}^k(\mathcal{T}_h) : \int_\Omega p = 0 \right\}, \end{aligned}$$

where $\mathcal{P}^k(X)^d$ is the space of polynomials of degree up to k on X with values in $\mathbb{R}^d$, and $\mathcal{P}^k(\mathcal{T}_h)$ is the subspace of $L^2(\Omega)$ with restrictions to elements of that are polynomials of degree up to k. Define the subspace:

$$\underline{U}_{h,0}^k = \left\{ \underline{v}_h \in \underline{U}_h^k : v_E = 0 \, \forall E \subset \partial\Omega \right\}.$$

The method is based on discrete reconstruction operators that are defined locally on each element. Given $T \in \mathcal{T}_h$, the space of local degrees of freedom for velocity reads:

$$\underline{U}_T^k \coloneqq \left\{ \underline{v}_T = (v_T, (v_E)_{E \in \mathcal{E}_T}) : v_T \in \mathcal{P}^k(T)^2, v_F \in \mathcal{P}^k(E)^2 \quad \forall E \in \mathcal{E}_T \right\}. \tag{6.12}$$

A local velocity reconstruction operator $r_T^{k+1} : \underline{U}_T^k \to \mathcal{P}^{k+1}(T)^2$ is defined such that $r_T^{k+1} \underline{v}_T$ satisfies:

$$\begin{aligned} &\int_T \nabla r_T^{k+1} \underline{v}_T : \nabla w = -\int_T v_T \cdot \Delta w + \sum_{E_\epsilon \mathcal{E}_T} \int_E v_F \cdot (\nabla w \, n_{TE}) \quad \forall w \in \mathcal{P}^{k+1}(T)^2, \\ &\int_T r_T^{k+1} \underline{v}_T - v_T = 0. \end{aligned} \tag{6.13}$$

A local discrete divergence operator $D_T^k : \underline{U}_T^k \to \mathcal{P}^k(T)^2$ is defined such that:

$$\int_T D_T^k \underline{v}_T q = -\int_T v_T \cdot \nabla q + \sum_{E \in \mathcal{E}_T} \int_E (v_E \cdot n_{TE}) q \quad \forall q \in \mathcal{P}^k(T)^2. \tag{6.14}$$

Introduce the following local discrete forms:

$$a_T(\underline{u}_T, \underline{v}_T) = \Big(\int_T \mu \nabla_s r^{k+1} \underline{u}_T : \nabla_s r^{k+1} \underline{v}_T \Big) + s_T(\underline{u}_T, \underline{v}_T)$$
$$b_T(\underline{v}_T, q_T) = -\int_T D_T^k \underline{v}_T q_T,$$

where s_T is a stabilization form introduced to guarantee the well-posedness of the discrete problem. There are several definitions of s_T proposed in the literature about HHO. In general it is designed to be symmetric positive and polynomially consistent [Di Pietro and Droniou 2020, subsection 8.3.2]. A definition based on [Veiga et al. 2013] is:

$$s_T(\underline{u}_T, \underline{v}_T) \coloneqq h_T^{-2} \int_T \delta_T^k \underline{u}_T \cdot \delta_T^k \underline{v}_T + \sum_{E \in \mathcal{E}_T} h_E^{-1} \int_E \delta_{TE}^k \underline{u}_T \cdot \delta_{TE}^k \underline{v}_T,$$

where:

$$\delta_T^k \underline{v}_T \coloneqq \pi_T^k (r_T^{k+1} \underline{v}_T - v_T), \qquad \delta_{TE}^k \underline{v}_T \coloneqq \pi_E^k (r_T^{k+1} \underline{v}_T - v_E) \quad \forall E \in \mathcal{T}_E,$$

$\mathcal{T}_E$ being the set of elements sharing edge E. Introduce the global discrete forms $a_h : \underline{U}_h^k \times \underline{U}_h^k \to \mathbb{R}$ and $b_h : \underline{U}_h^k \times \mathcal{P}^k(\mathcal{T}_h) \to \mathbb{R}$ such that:

$$a_h(\underline{u}_h, \underline{v}_h) = \sum_{T \in \mathcal{T}_h} a_T(\underline{u}_T, \underline{v}_T), \qquad {}_h(\underline{v}_h, p_h) = \sum_{T \in \mathcal{T}_h} b_T(\underline{v}_T, p_T).$$

Let $\mathcal{P}^k(\Gamma_h)^2$ be the space of broken polynomials over the edges of Γ_h with value in $\mathbb{R}^2$. Given $t_\Gamma \in \mathcal{P}^k(\Gamma_h)^2$, the discrete problem reads: Find $(\underline{u}_h, p_h) \in \underline{U}_{h,0}^k \times P_h^k$ such that:

$$\begin{aligned} a_h(\underline{u}_h, \underline{v}_h) - b_h(\underline{v}_h, p_h) &= -\sum_{E \in \Gamma_h} \int_E t_\Gamma \cdot v_E \qquad && \forall \underline{v} \in \underline{U}_{h,0}^k, \\ b_h(\underline{u}_h, q_h) &= 0 && \forall q_h \in \mathcal{P}^k(\mathcal{T}_h). \end{aligned} \tag{6.15}$$

The well-posedness of the discrete problem can be proved for $k \geq 1$ [Di Pietro and Droniou 2020, subsection 7.3.4]. In the particular case $k = 0$, stability can be guaranteed by reformulating the problem as: Find $(\underline{u}_h, p_h) \in \underline{U}_{h,0}^0 \times P_h^k$ such that:

$$\begin{aligned} a_h(\underline{u}_h, \underline{v}_h) - b_h(\underline{v}_h, p_h) + j_h(r^{k+1}\underline{u}_h, r^{k+1}\underline{v}_h) &= -\sum_{E \in \Gamma_h} \int_E t_\Gamma \cdot v_E \qquad && \forall \underline{v} \in \underline{U}_{h,0}^k, \\ b_h(\underline{u}_h, q_h) &= 0 && \forall q_h \in \mathcal{P}^k(\mathcal{T}_h), \end{aligned} \tag{6.16}$$

where the form $j_h : H^1(\mathcal{T}_h)^2 \times H^1(\mathcal{T}_h)^2 \to \mathbb{R}$, inspired by [Botti, Di Pietro, and Guglielmana 2019] is defined as:

$$j_h(u, v) = \sum_{E \in \mathcal{E}_h} h_E^{-1} ([u]_E, [v]_E)_E.$$

Given a mesh $(\mathcal{T}_h, \mathcal{E}_h)$, define $(p_h, \underline{v}_h) = \mathfrak{S}(\mathcal{T}_h, \mathcal{E}_h)$ the solution of problem 6.16.

6.3 Simulation of Capsule Flow

A capsule at rest is discretized as a polygon $\Gamma_{h,0}$, with set of edges $\mathcal{E}_\Gamma$ and set of vertices $\mathcal{V}_\Gamma$. Given the current configuration Γ_h of the membrane, let $\mathbb{P}^1(\Gamma_h, 2) = \mathcal{P}^1(\Gamma_h, 2) \cap C^0(\Gamma_h, 2)$ be the space of vector-valued Lagrange finite elements over Γ_h, whose elements are described by degrees of freedom associated to nodes.

The elastic component of membrane stress is represented by $t_{str} \in \mathbb{P}^1(\Gamma_h, 2)$ such that:

$$t_{str,V} = k_{str} \sum_{E \in \mathcal{E}_V} \left(\frac{|E|}{|E_0|} l_{V,E}\right) \qquad \forall V \in \mathcal{V}_\Gamma,$$

where $|E|$ is the length of E, $|E_0|$ is the length of the edge in the reference configuration and $l_{V,E}$ is the unit vector tangent to E and pointing out of V.

The bending component of membrane tension is represented by $t_{bnd} \in \mathbb{P}^1(\Gamma_h, 2)$ such that:

$$t_{bnd,V} = k_{bnd}\left(\frac{1}{2}k_V^3 + \Delta_{LB,h}k_V\right)n_V,$$

where the curvature at nodes k_V, the discrete Laplace-Beltrami operator $\Delta_{LB,h}$ and the normal vector n_V are computed as subsection 3.1.1. The total stress $t_\Gamma \in \mathcal{P}^0(\Gamma_h, 2)$ is given by:

$$t_\Gamma = I_{\mathcal{V}\to\mathcal{E}}(t_{str} + t_{bnd}),$$

where the edge average operator $I_{\mathcal{V}\to\mathcal{E}} : \mathbb{P}^1(\Gamma_h, 2) \to \mathcal{P}^0(\Gamma_h, 2)$ is such that for $\eta \in \mathbb{P}^1(\Gamma_h, 2)$:

$$I_{\mathcal{V}\to\mathcal{E}}(\eta)_E = \frac{1}{2}\sum_{V \in E_V} \eta_V;$$

a vertex average operator $I_{\mathcal{E}\to\mathcal{V}} : \mathcal{P}^0(\Gamma_h, 2) \to \mathbb{P}^1(\Gamma_h, 2)$ is defined in a similar way.

Introduce a time step τ. Given the configuration of the capsule $\Gamma_h^n = \Gamma_h(n\tau)$, the advancement of position is performed with the following algorithm:

- Determine membrane stress $t_\Gamma^n(\Gamma_h^n)$;
- Determine a mesh $(\mathcal{T}_h^n, \mathcal{E}_h^n)$ conforming to Γ_h^n (see Figure 6.1);
- Determine velocity and pressure of the flow as $(\underline{u}_h, p_h) = \mathfrak{S}(\mathcal{T}_h^n, \mathcal{E}_h^n)$;
- Determine the velocity of the membrane $V_h \in \mathbb{P}^1(\Gamma_h^n)$ such that $V_h = I_{\mathcal{E}\to\mathcal{V}}(\underline{v}_{h,\Gamma})$, where $\underline{v}_{h,\Gamma}$ is the restrictions of $\underline{v}_h$ to $\mathcal{P}^0(\Gamma_h^n, 2)$;
- Obtain the updated configuration Γ_h^{n+1} of the capsule by updating the position of the vertices as:

$$W^{n+1} = W^n + \tau V_{h,W} \qquad \forall W \in \mathcal{V}_\Gamma^n.$$

Since the internal fluid. of the capsule is incompressible, the volume bounded by the membrane is expected to be conserved along the trajectory. In order to enforce volume conservation, the position of the vertices of the updated membrane Γ_h^{n+1} is corrected *a posteriori* in the spirit of [D'Elia et al. 2016] and in a similar fashion to the volume conservation algorithm implemented in `YALES2BIO`. The position of the membrane after advection is represented by $x^{n+1} \in \mathbb{P}^1(\Gamma_h, 2)$. Let $\mathfrak{V} : \mathbb{P}^1(\Gamma_h, 2) \to \mathbb{R}$

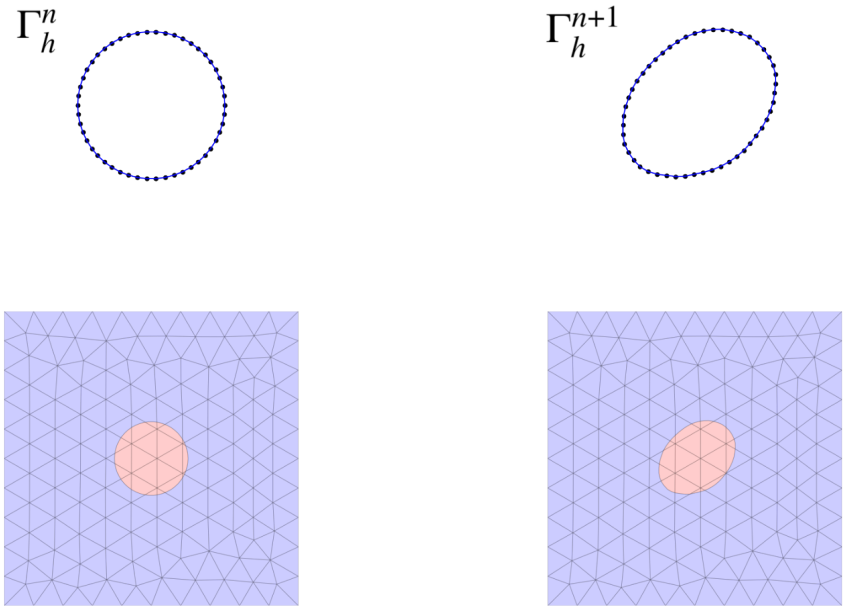


Figure 6.1: On top, discrete interface Γ_h on consecutive time steps. On bottom, conforming polygonal mesh built starting from Γ_h.

represent the operator returning the volume of the membrane associated to a node position, and let V_0 be the volume of Γ_0. The corrected position $x^* \in \mathbb{P}^1(\gamma_h, 2)$ is found as the solution of the problem: Find $y \in \mathbb{P}^1(\Gamma_h, 2)$ minimizing:

$$\min \sum_{V \in \mathcal{V}_\Gamma} \|y_V - x_V^{n+1}\|^2$$

subject to:

$$\mathfrak{V}(y) = V_0.$$

The minimization problem is solved with a Sequential Least Square Programming (SLSQP) algorithm [Boggs and Tolle 1995].

6.4 Preliminary Tests

In this section, the performance of the solver on two benchmarks is considered. The first represents a stationary configuration for which an analytical solution is available, the second is an unsteady test case for which the numeric solution can be compared with results from `YALES2BIO`. Given the implementation constraints of the current implementation, the tests are realized on rather coarse meshes. A systematic error convergence analysis constitutes an essential future step, in program for when an optimized version of the code is available.

6.4.1 Static Extended Capsule

Problem 6.16 is solved considering a configuration in which a circular capsule, with radius R at rest and coefficients $k_{str} = 1\ \mathrm{Nm}^{-1}$, $k_{bnd} = 0$ is expanded isotropically with a factor α. Vanishing velocity at the boundary is considered. The benchmark is taken as a test to check the correcteness in the treatment of the normal stress interface condition.

In such a configuration, the velocity of the flow is expected to be 0 in the entire domain, with a region-wise constant pressure. The pressure jump at the interface found by integrating condition 6.7e multiplied by the normal vector n_Γ reads:

$$p_{\text{ext}} - p_{\text{int}} = -\frac{k_{str}(\alpha - 1)}{R\alpha}, \tag{6.17}$$

reflecting Young-Laplace law relating pressure difference to surface tension. A test is run to check if the expected pressure profile is retrieved and relation 6.17 is respected. The numeric solution is sought for $\alpha = 1.1$. The result is shown in Figure 6.2, along with the discrete setting. The region-wise pressure profile is retrieved. Even though the employed mesh is fairly coarse, the numeric error on pressure difference expressed in terms of the pressure scale (k_{str}/R) is:

$$\Big[(p_{h,\text{ext}} - p_{h,\text{int}}) + \frac{k_{str}(\alpha - 1)}{R\alpha}\Big](\frac{R}{k_{str}}) = 9.262\,10^{-5}.$$

6.4.2 Shear Flow

The algorithm is tested to reproduce the dynamics of a capsule in shear flow. The capsule has a circular shape with radius R at rest. No resistance to bending is assumed ($k_b = 0$). It is set in a shear flow with shear rate γ, and a uniform viscosity μ. The problem is characterized by two timescales, the inverse shear rate $t_\gamma = \gamma^{-1}$ and the elastic relaxation time $t_{str} = \mu/k_{str}$. Their ratio defines the capillary number $Ca = t_{str}/t_{shear}$.

A simulation is performed with $Ca = 1$ (large deformation). The center of the capsule in its resting configuration is taken as center of the reference system (cartesian coordinates), and the boundary condition for velocity is $g = (\gamma x, 0)$. The test is run on the mesh displayed in Figure 6.3 In this regime of Ca, the capsule is expected to deform in the direction of the flow up to reaching a configuration of maximal deformation, after which the membrane starts tank treading around the capsule. A solution based on the HHO solution for the flow is compared with a simulation performed with `YALES2BIO` 6.4.

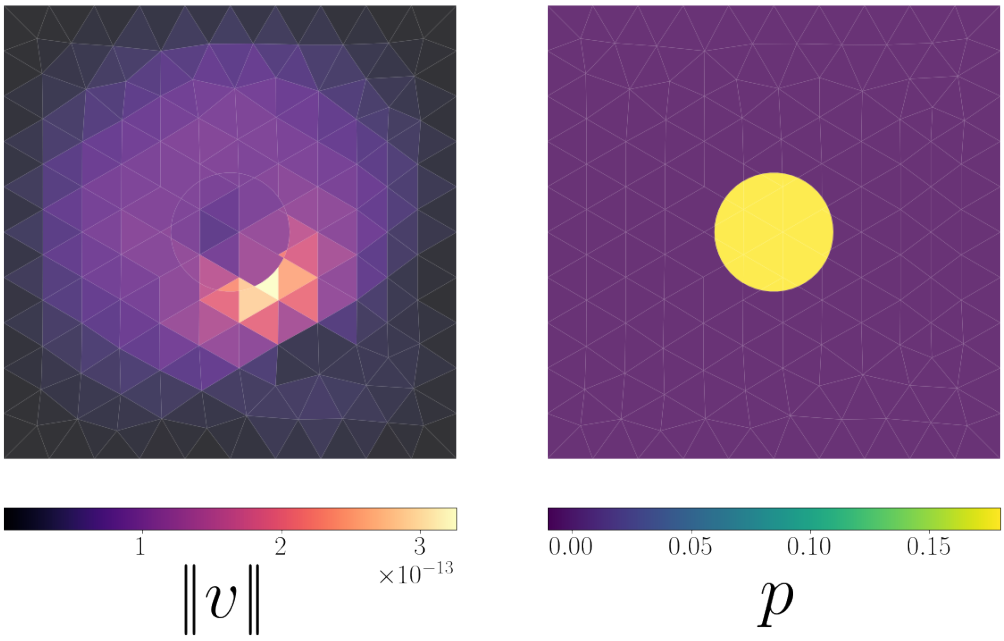


Figure 6.2: Velocity and pressure for the test of subsection 6.4.1. The domain Ω is a square box with side $8R$. The interface is discretized with 50 evenly distributed vertices.

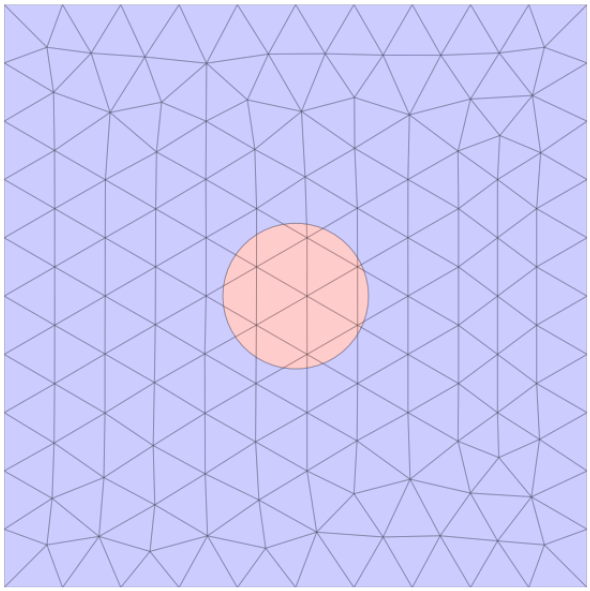


Figure 6.3: Discrete setting of the test from subsection 6.4.2. The domain Ω is a square box with side $8R$. The interface is discretized with 50 evenly distributed vertices.

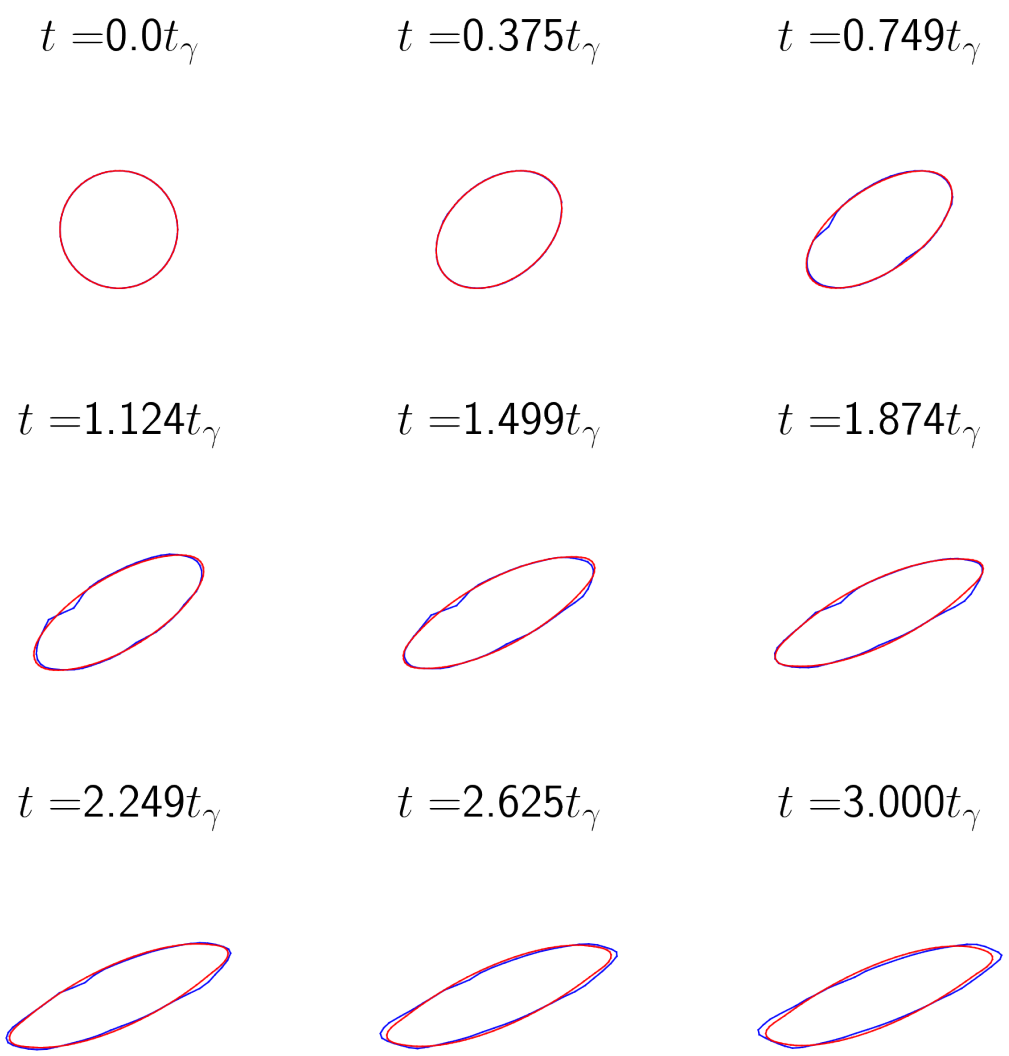


Figure 6.4: Deformation of a circular capsule in shear flow, $Ca = 1$ (subsection 6.4.2). In red, simulation obtained with YALES2BIO. In blue, simulation obtained with HHO-based solver.

CHAPTER

SEVEN

LAST WORDS AND PERSPECTIVES

The work constitutes a small contribution to a large and complex subject, at the crossroad of several disciplines of physics and mathematics. Most of the effort has been adressed to the treatment of jump conditions along moving interfaces, a general problem in computational mechanics and numerical analysis. The great difficulty in this domain consists in conjugating a seamless and cost-effective treatment of the interface with a robust and mathematically consistent numerical approach. As pointed out in section 4.4, the arbitrariness of the interface position with respect to the background mesh can pose serious robustness problems if no adaptation whatsoever is considered. The strategies presented in chapters 5 and 6 appear as a promising solution, well suited to address this kind of problem. The development of polytopal methods for capsule dynamics, however, is only at the beginning. Future work should go in the direction of increasing efficiency. So far, thanks to the great liberty given by polytopal methods in the manipulation of the mesh, some rudimentary forms of nonconforming refinement and element agglomeration have been put in place. These features of the method could be further developed to exploit them at their full potential. In this sense, an important source of inspiration would be the techniques presented in [Burman et al. 2021]. Another important feature of these methods is the possibility to increase the order of accuracy arbitrarily and in a way that does not impact the complexity of the implementation.

The immediate next step would consist in coupling the membrane dynamics solver of 6 to the electrostatic solver of 5 to be able to reproduce the main benchmarks in electrodeformation of capsules. The coupling is straight-forward, and does not need specific adaptations for the implementation.

An even more important goal, however, would be to develop polytopal methods not only as a stand-alone solution, but rather as modules to enhance traditional solvers that already provide efficient frameworks for fluid-structure interaction, such as as `YALES2BIO`. The technical obstacle would regard the management of the mesh and its cut along the interface on an optimized platform with element groups distributed across processors. These difficulties would be even more evident for a full 3D implementation.

To conclude, it is important to cite that the numerical solutions proposed in this work are based on a model of a physical phenomenon that is still the subject of an intense research. Even though the Leaky Dielectric Model provides an important starting point to discuss the numerical treatment, in the future it will be important to treat extensions and alternatives. For a glimpse on current undergoing research on the instabilization of biological membranes under DC fields we refer to works on the modelization of charged Debye layers presented in [Zhao et al. 2025] and [Yu et al. 2025].

APPENDIX

A

ERROR ANALYSIS IN YALES2BIO

In this appendix we consider problem 4.1 from section 4.1, and perform an error analysis by comparing the numerical solution obtained wit `YALES2BIO` with the exact solution.

In order to introduce a measure of the error, consider the mesh of the domain Ω_h with set of vertices $\mathcal{V}_\Omega$. Let Ω_V be the finite volume with measure $|\Omega_V|$ associated to the node V. Call X_h the space of `GRID_DATA` defined on Ω_h with degrees of freedom associated to nodes. With $w : \Omega \setminus \Sigma \to \mathbb{R}$, introduce the interpolation $\hat{w} \in X_h$ such that $\hat{w}_V = w(\boldsymbol{x}_V)$ for $V \in \mathcal{V}$. Introduce a discrete L_2 norm in X_h:

$$\|u\|_2^2 \coloneqq \sum_{V \in \mathcal{V}_\Omega} u_V^2 |\Omega_V|.$$

Consider also a discrete L_∞ norm:

$$\|u\|_\infty \coloneqq \max_{V \in \mathcal{V}_\Omega} |u_V|.$$

Likewise, consider the mesh of the surface Σ_h = with set of vertices $\mathcal{V}_\Sigma$. With $V \in \mathcal{V}$, consider the element of surface Σ_V associated to V. If $d = 2$, Σ_V is given by the broken line connecting the midpoints of the edges sharing V and V itself. With $d = 3$, a similar construction can be considered on a triangulated surface. Let Y_h be the space of scalar `SURFACE_DATA` defined on Σ_h with values defined at vertices. For a field $U : \Sigma \to \mathbb{R}$, introduce the interpolation $\hat{U} \in Y_h$ such that, for $V \in \mathcal{V}_\Sigma$, $\hat{U}_V = U(\boldsymbol{X}_V)$. With $U_h \in Y_h$ define the discrete norms:

$$\|U_h\|_2^2 \coloneqq \sum_{V \in \mathcal{V}_\Sigma} U_{h,V}^2 |\Sigma_V|, \qquad \|U_h\|_\infty \coloneqq \max_{V \in \mathcal{V}_\Sigma} |U_{h,V}|.$$

Let (ϕ, J_Σ) be the solution to Problem 4.1 and let $(\phi_h, J_{\Sigma,h}) \in X_h \times Y_h$ be the solution estimated with `ESS`. Error analysis is performed in terms of the normalised errors:

$$\frac{\|\hat{\phi} - \phi_h\|_\bullet}{\|\hat{\phi}\|_\bullet} \qquad \frac{\|\hat{J}_\Sigma - J_{\Sigma,h}\|_\bullet}{\|\hat{J}_\Sigma\|_\bullet},$$

with $\bullet \in \{2, \infty\}$.

The results of the error analysis are organized in two sections, dedicated respectively to the cases $d = 2$ and $d = 3$. Each section is organized in subsections, each one corresponding to a convergence test realized for a given value of conductivity ratio σ_r. The set of values of conductivty ratio considered is $\{0.1, 1, 10\}$. The convergence tests are realized seeking the numerical solution of Problem 4.1 for each mesh in a sequence of progressively refined meshes. For every refinemet level, the error in L_2

and L_∞ norms along time is plotted for the potential ϕ, the electric field $\nabla\phi$, the potential jump J_Σ and the electric force:

$$\boldsymbol{F}_{MW} = [\nabla\phi \otimes \nabla\phi - \frac{1}{2}|\nabla\phi|^2 I] \cdot \boldsymbol{n}_\Sigma.$$

To estimate the trend of the error with respect to the refinement level of the mesh, errors are plotted against the mesh element diameter at a given time iteration. In particular, the first and the last iteration are considered. Evaluating errors at first iteration is important to isolate errors exclusively dependent of space discretization. Recalling the structure of the electrostatic solver ESS (see Figure 3.6), each iteration takes as input the potential jump J_Σ to compute the potential ϕ_h with the numerical scheme described in Subsection 3.4.1. Electric field $\nabla\phi_h$ is then estimated according to the interpolation method described in Subsection 3.4.2. Therefore, accuracy in estimation of ϕ_h and $\nabla\phi_h$ is influenced by the error on the potential jump J_Σ. Evaluating the error on ϕ_h at first iteration ($t = 0$) allows to assess the accuracy of the space discretization schemes to estimate ϕ_h and $\nabla\phi_h$ independently of the error on J_Σ, since $J_\Sigma(0)$ is known. On the other hand, convergence tests realized at the last iteration allow to establish the accuracy of the solver globally, taking into account errors introduced along the entire simulation by every step of the numerical pipeline.

From the point of view of the expected convergence rate, the Finite Volume scheme employed to estimate the electric potential converges with order 1, and the correction scheme introduced by the Ghost Fluid method is expected to preserve the order. This, however, is true assuming that the data of the elliptic interface problem don't introduce on their own further discretization errors. Since the numerical interface potential jump used as a datum is based on the 0-order interpolation algorithm presented in 3.4.2, this might explain the degradation of the convergence rate in some of the cases that are showed.

A.1 $d = 2$

The convergence tests are realized on a family of unstructured triangular meshes $\{\Omega_h\}_{h\in\mathcal{H}_\Omega}$, with mesh diameters in $\mathcal{H}_\Omega = \{h_\Omega/2^k,\ 1 \le k \le 4\}$, where $h_\Omega = 2R/10$ represents the coarsest refinement level, R being the radius of the surface Σ. A corresponding family of interface meshes $\{\Sigma_h\}_{h\in\mathcal{H}_\Sigma}$ with $\mathcal{H}_\Sigma = \{h_\Sigma/2^k,\ 1 \le k \le 4\}$ and $h_\Sigma = \pi R/16$ is considered.

The convergence slopes are commented in more detail for the case $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 1$. For the remaining two cases similar observations hold.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 1$

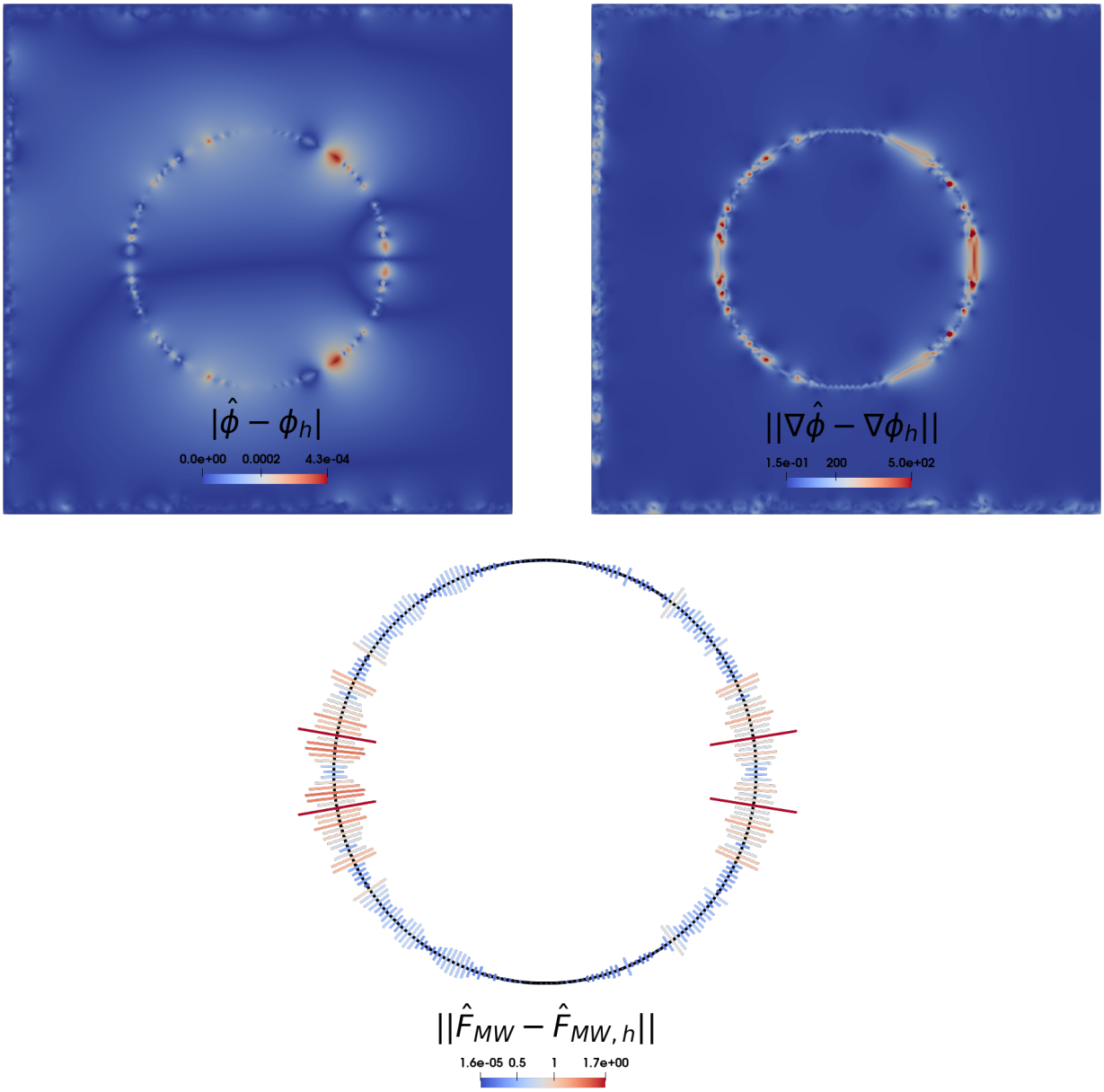


Figure A.1: Distribution of error over Ω and Σ at first iteration ($t = 0$). Error is concentrated in proximity of the interface, and it is largest at equator.

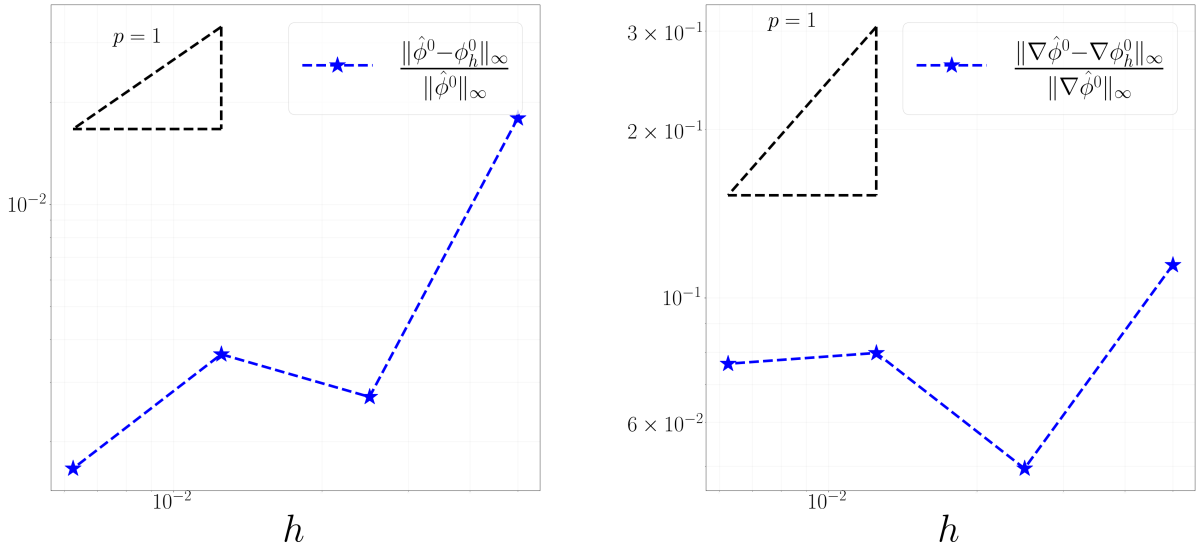
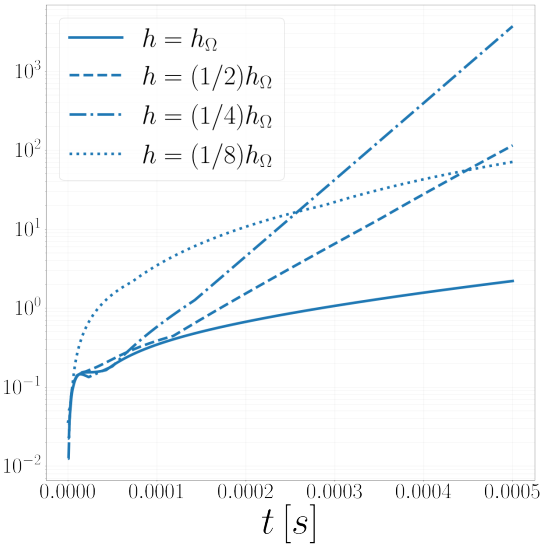
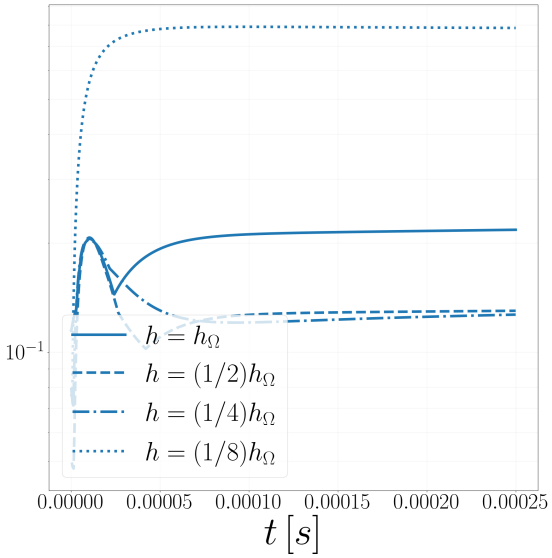


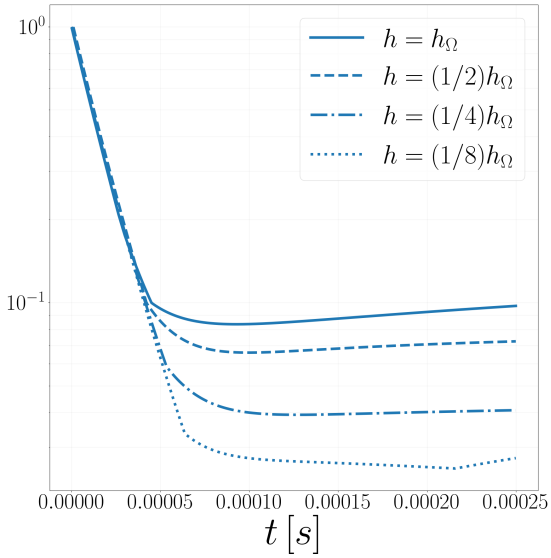
Figure A.2: Error at first iteration ($t = 0$) against mesh size h, norm L_∞. Comparison with a reference trend h^p, with $p = 1$. No convergence attained.



(a) $\frac{\|\hat{\phi}-\phi_h\|_\infty}{\|\hat{\phi}\|_\infty}(t)$



(b) $\frac{\|\nabla\hat{\phi}-\nabla\phi_h\|_\infty}{\|\nabla\hat{\phi}\|_\infty}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_\infty}{\|\hat{J}_\Sigma\|_\infty}(t)$

Figure A.3: Time evolution of error in norm L_∞. No convergence can be established.

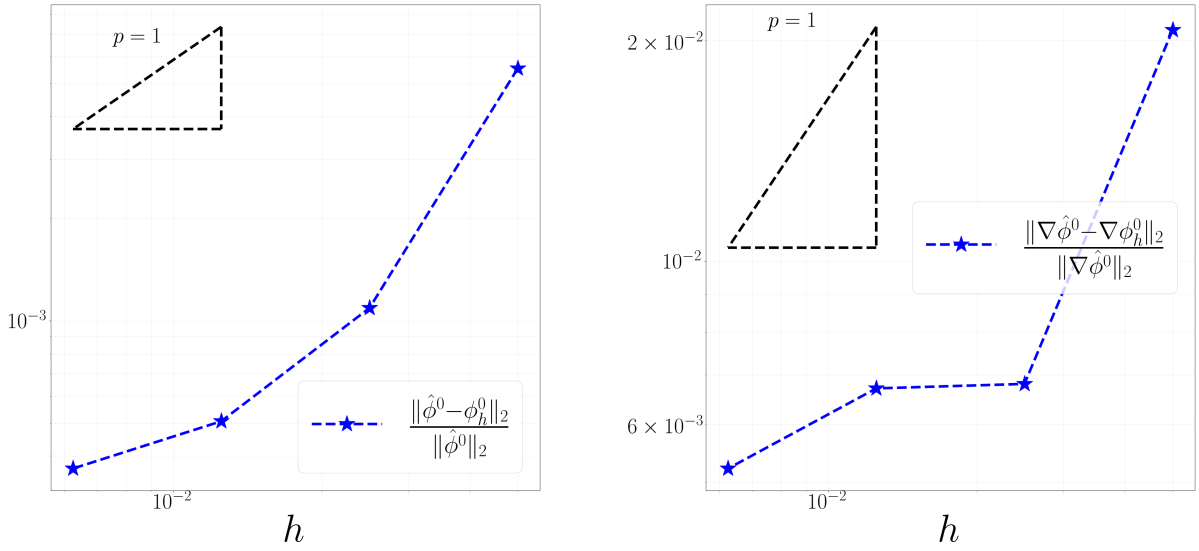
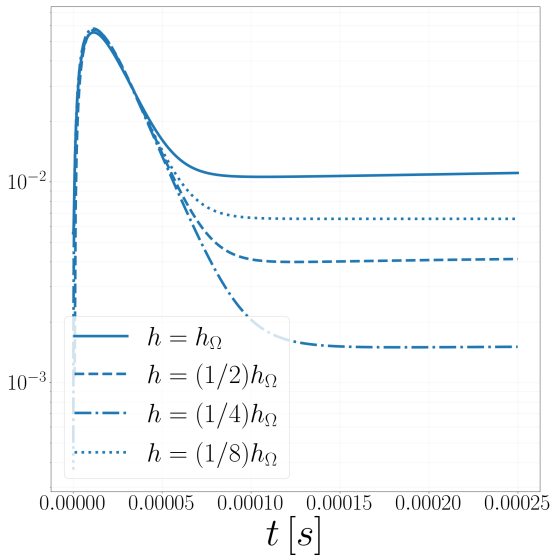
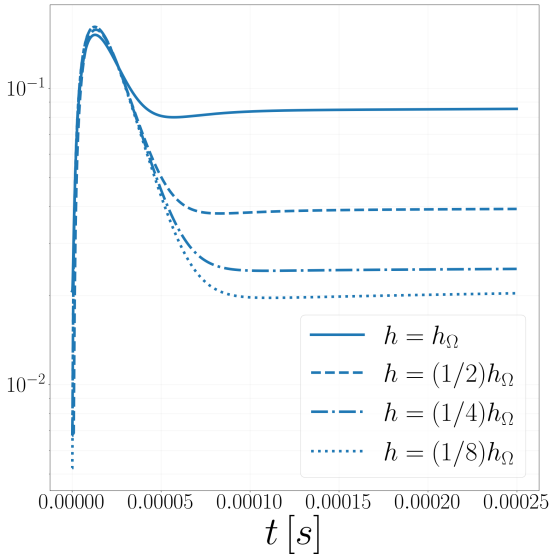


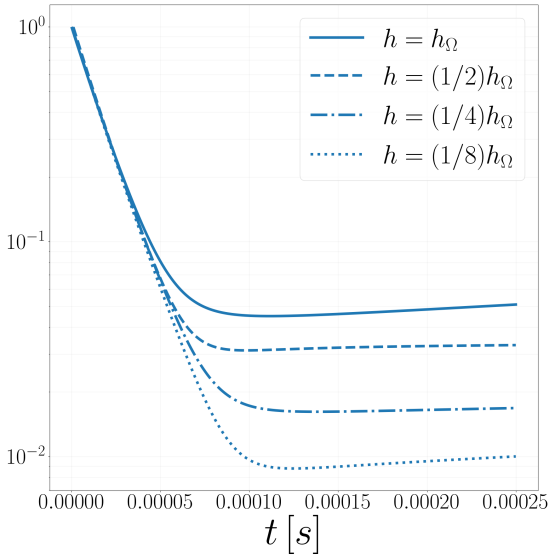
Figure A.4: Error at first iteration ($t = 0$) against mesh size h, norm L_2. Comparison with reference trend h^p, with $p = 1$. Better convergence slope With respect to L_∞ error, but still not conclusive about convergence order.



(a) $\frac{\|\hat{\phi}-\phi_h\|_2}{\|\hat{\phi}\|_2}(t)$



(b) $\frac{\|\nabla\hat{\phi}-\nabla\phi_h\|_2}{\|\nabla\hat{\phi}\|_2}(t)$



(c) $\frac{\|\hat{J}_\Sigma - J_{\Sigma,h}\|_2}{\|\hat{J}_\Sigma\|_2}(t)$

Figure A.5: Time evolution of error in norm L_2.

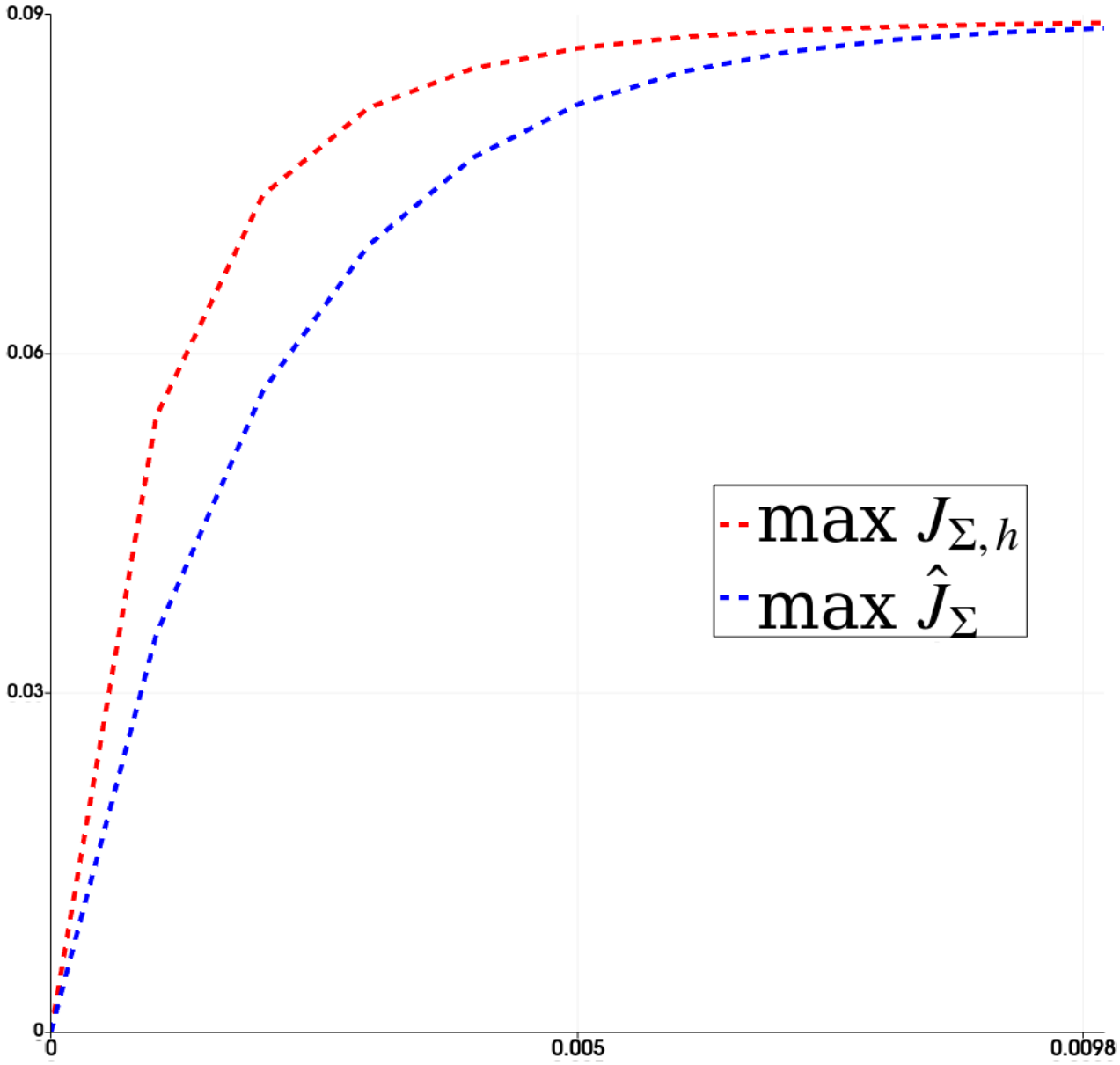


Figure A.6: Maximum potential jump over time (expresssed in s). Comparison of exact and numerical solution. Same comment as for L_∞ version.

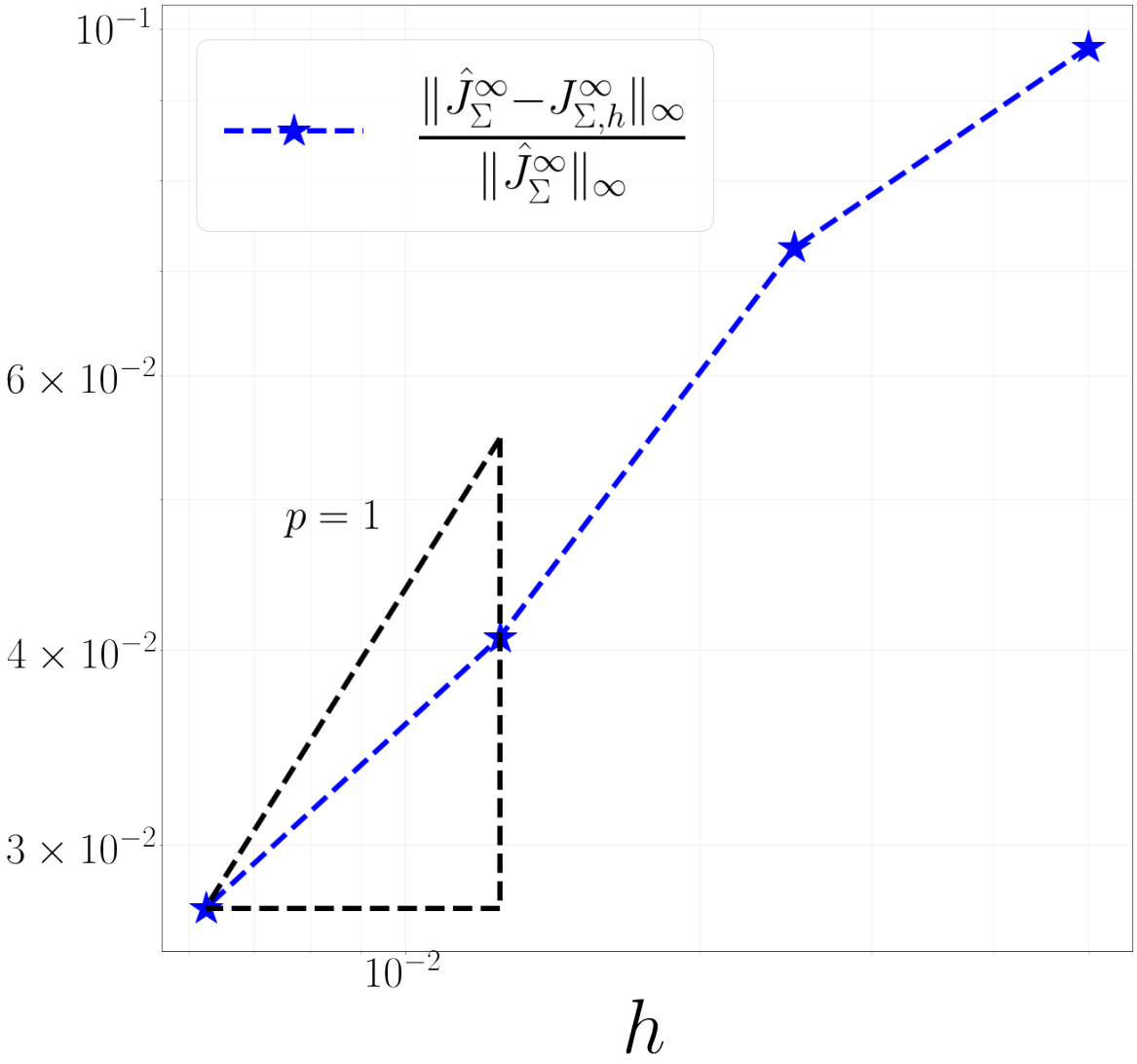


Figure A.7: Error on J_Σ at the end of the simulation against mesh size h, norm L_∞. Error is smaller on finer meshes.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 0.1$

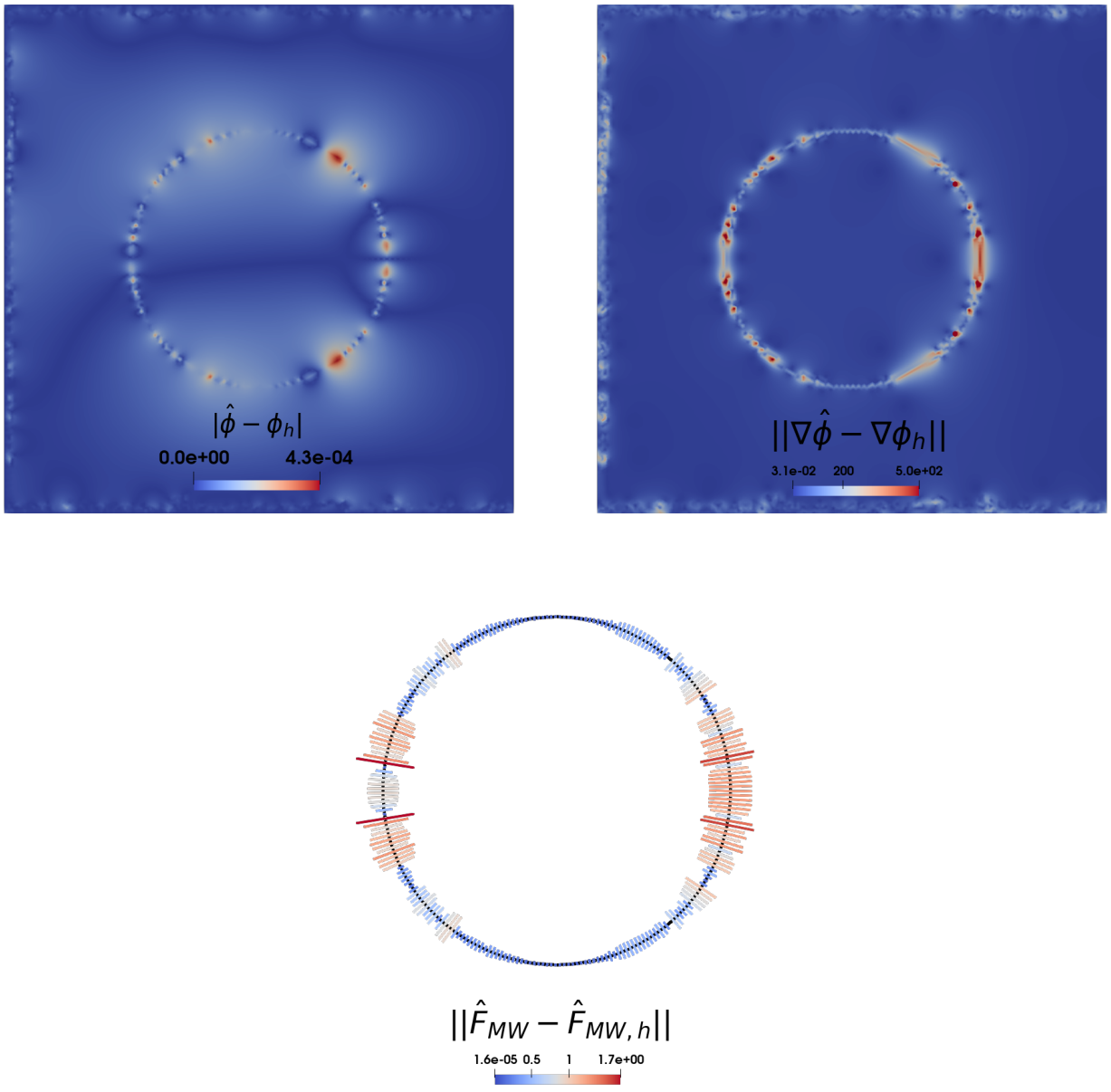


Figure A.8: Distribution of error over Ω and Σ at first iteration ($t = 0$).

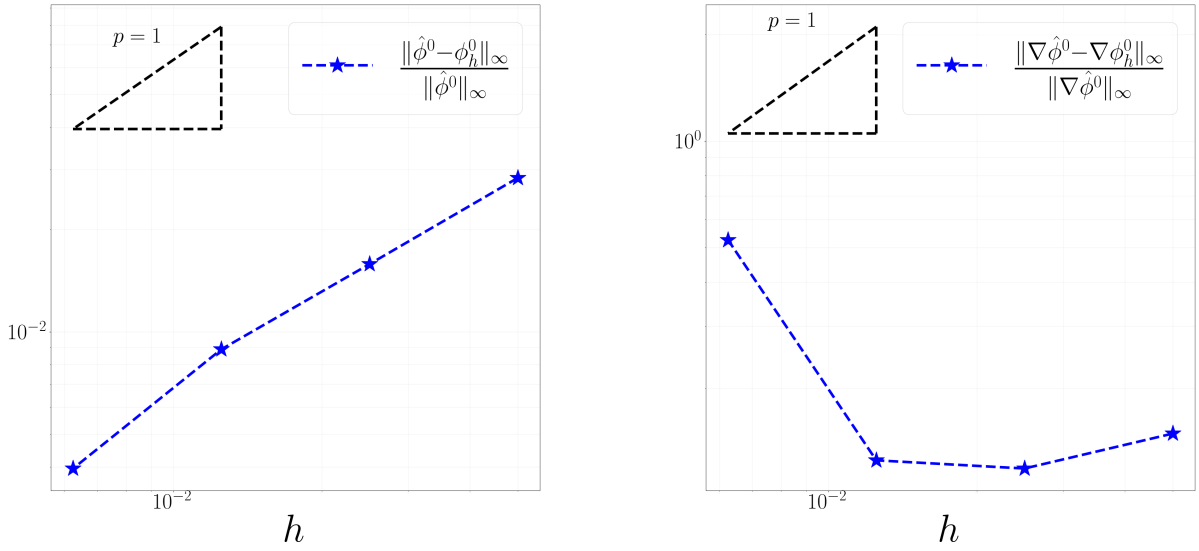
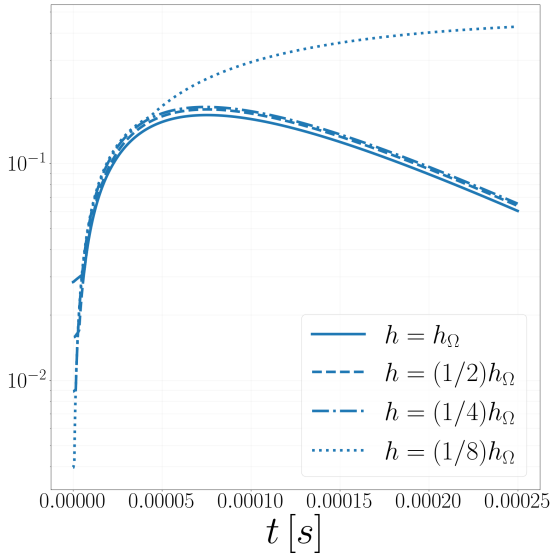
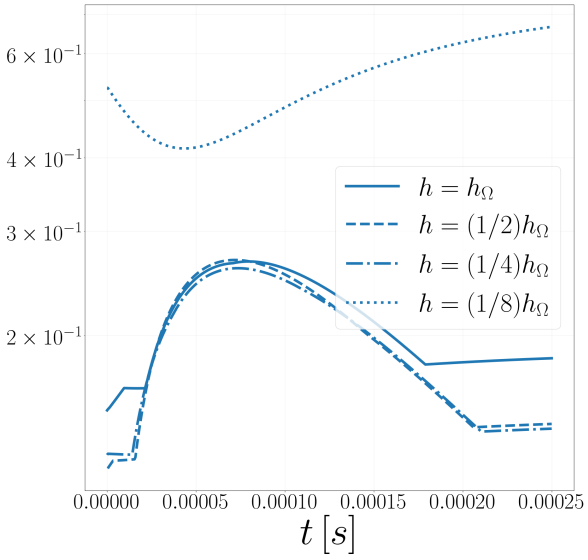


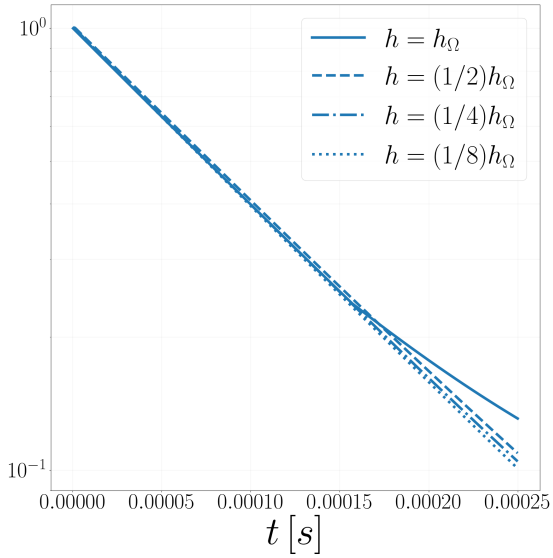
Figure A.9: Error at first iteration ($t = 0$) against mesh size h, norm L_∞. Comparison with a reference trend h^p, with $p = 1$.



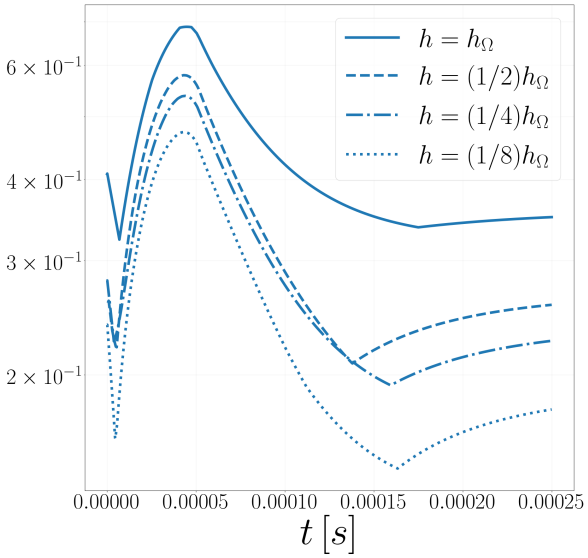
(a) $\frac{\|\hat{\phi}-\phi_h\|_\infty}{\|\hat{\phi}\|_\infty}(t)$



(b) $\frac{\|\hat{\nabla\phi}-\nabla\phi_h\|_\infty}{\|\hat{\nabla\phi}\|_\infty}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_\infty}{\|\hat{J}_\Sigma\|_\infty}(t)$



(d) $\frac{\|\hat{\boldsymbol{F}}_{MW}-\boldsymbol{F}_{MW,h}\|_\infty}{\|\hat{\boldsymbol{F}}_{MW}\|_\infty}(t)$

Figure A.10: Time evolution of error in norm L_∞.

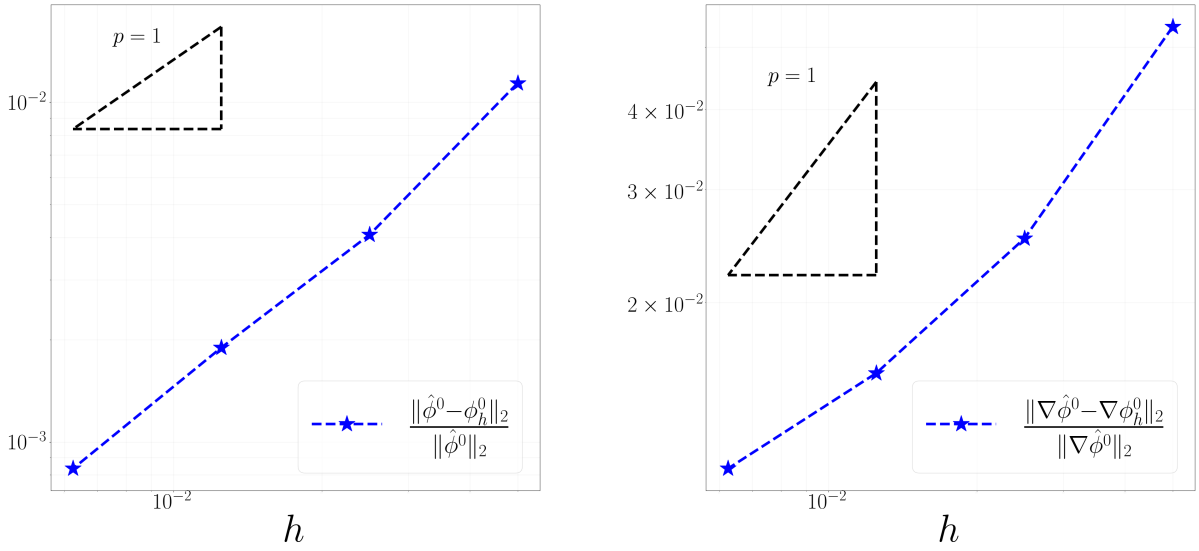
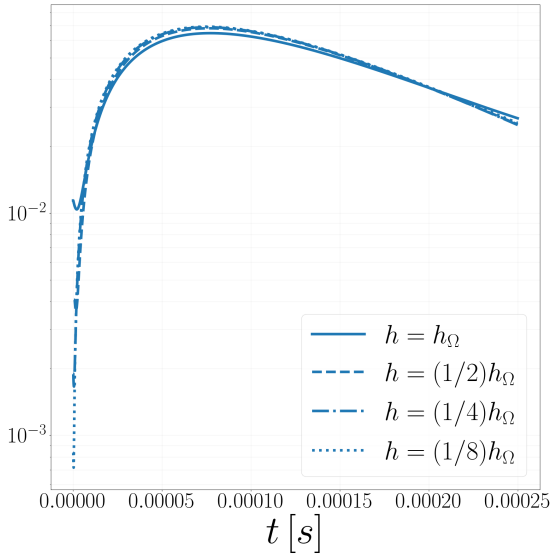
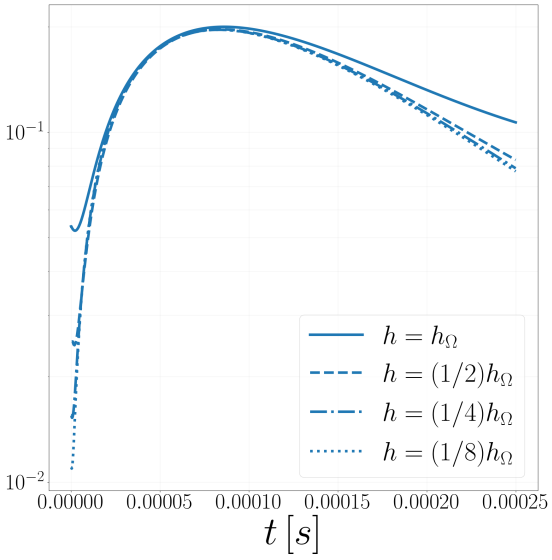


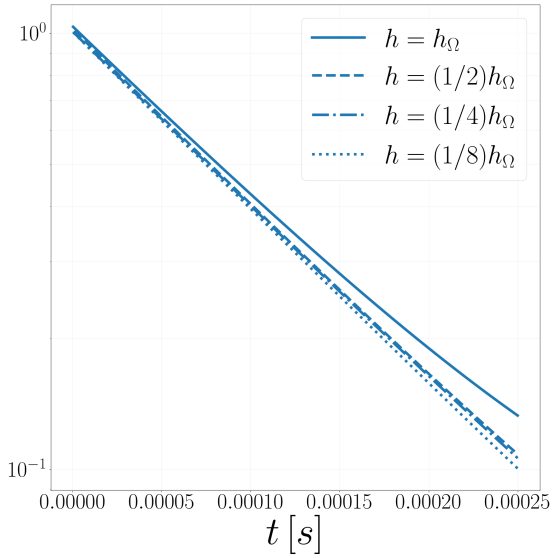
Figure A.11: Error at first iteration ($t = 0$) against mesh size h, norm L_∞. Comparison with reference trend h^p, with $p = 1$.



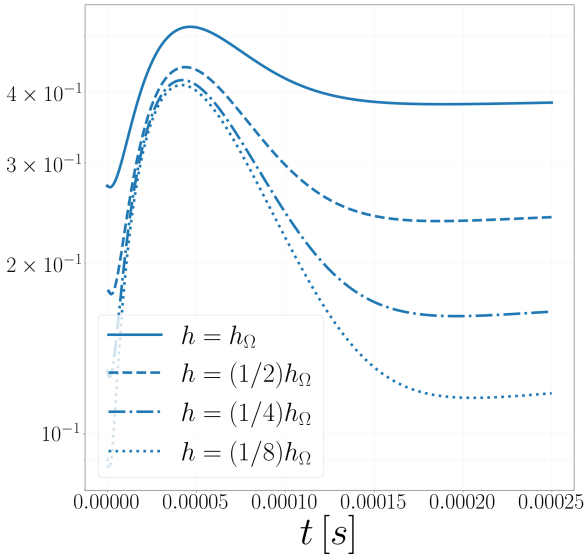
(a) $\frac{\|\hat{\phi}-\phi_h\|_2}{\|\hat{\phi}\|_2}(t)$



(b) $\frac{\|\nabla\hat{\phi}-\nabla\phi_h\|_2}{\|\nabla\hat{\phi}\|_2}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_2}{\|\hat{J}_\Sigma\|_2}(t)$



(d) $\frac{\|\hat{\boldsymbol{F}}_{\boldsymbol{MW}}-\boldsymbol{F}_{\boldsymbol{MW},h}\|_2}{\|\hat{\boldsymbol{F}}_{\boldsymbol{MW}}\|_2}(t)$

Figure A.12: Time evolution of error in norm L_2.

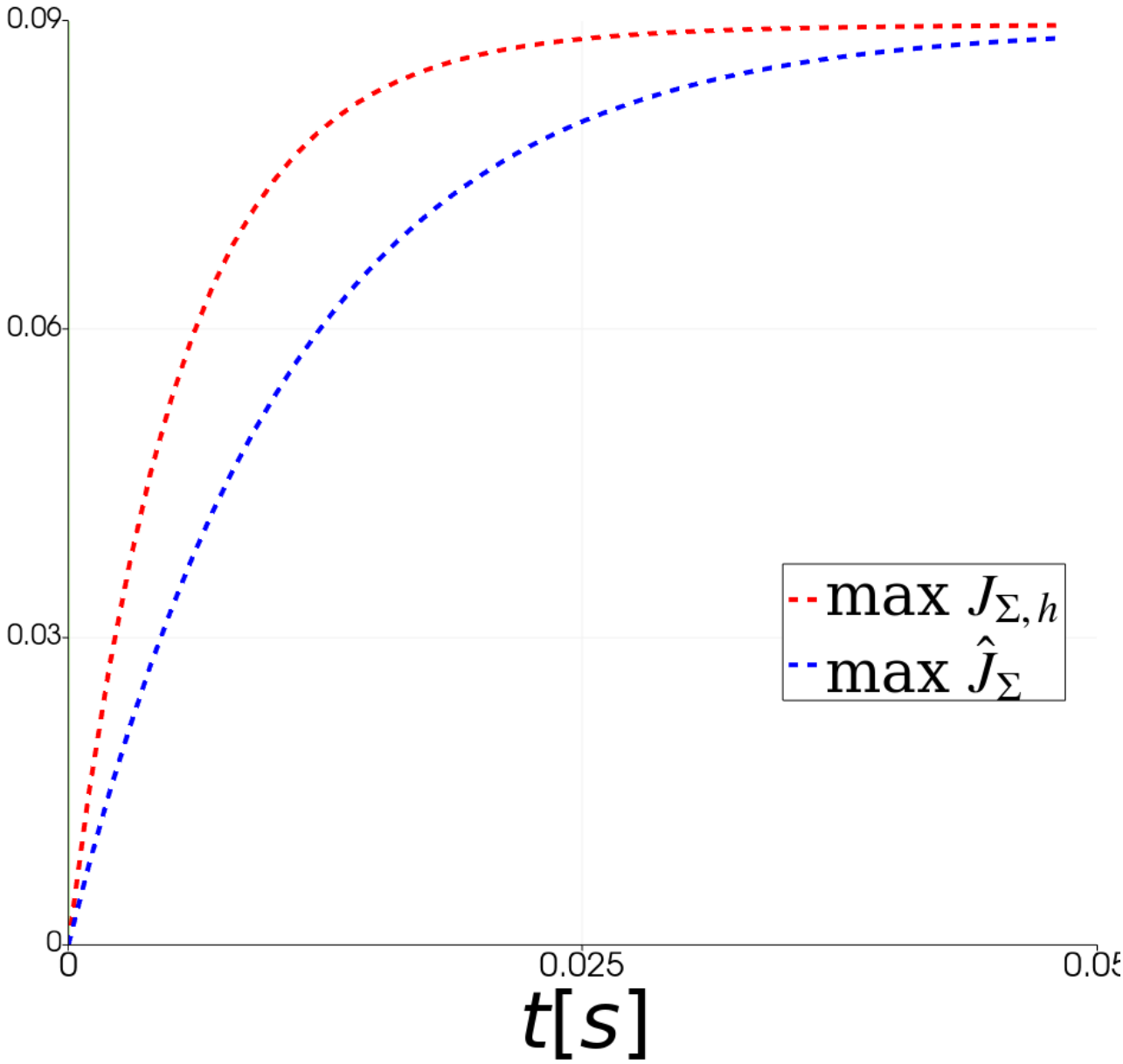


Figure A.13: Maximum potential jump over time. Comparison of exact and numerical solution.

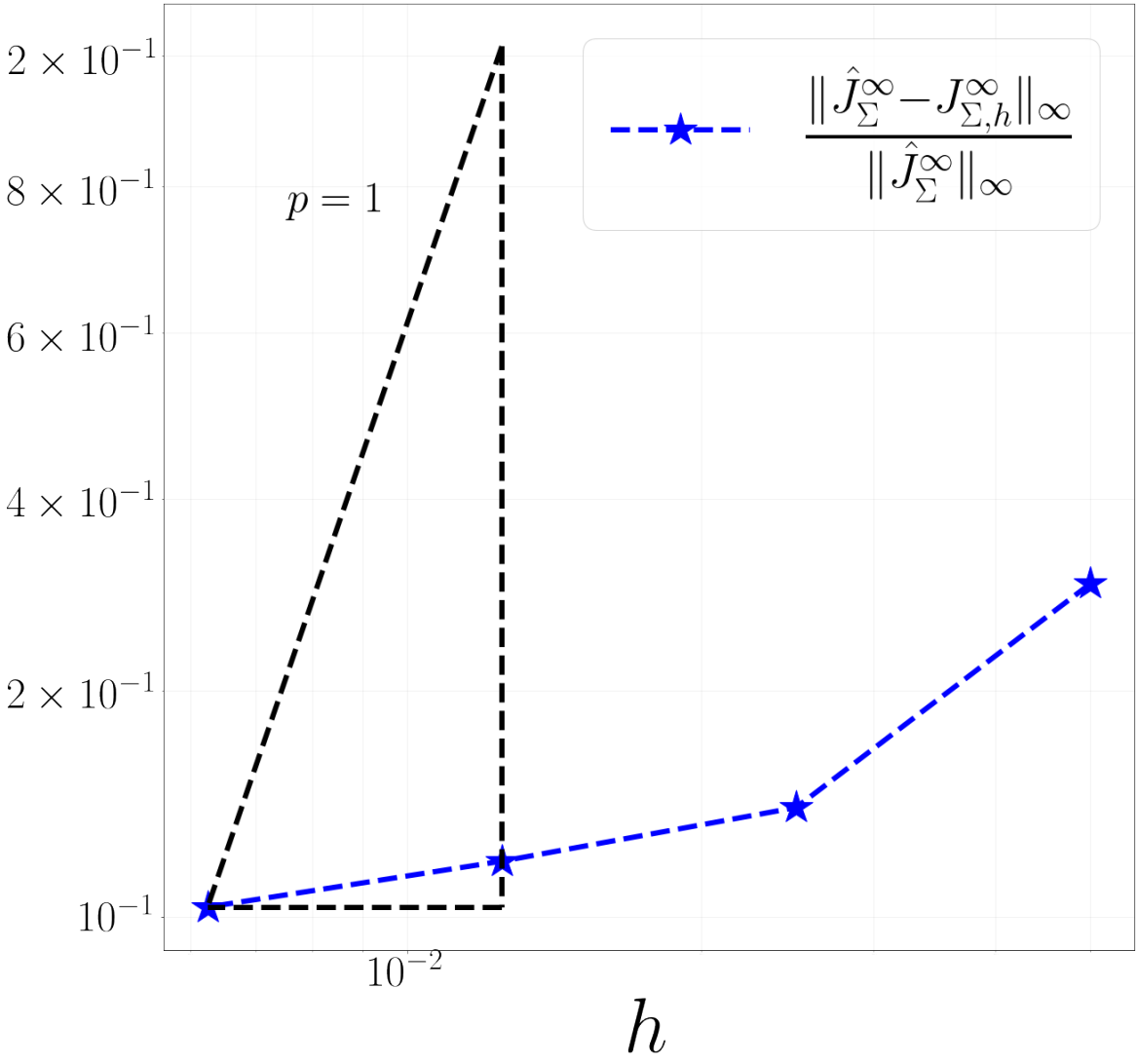


Figure A.14: Error on J_Σ at the end of the simulation against mesh size h, norm L_∞.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 10$

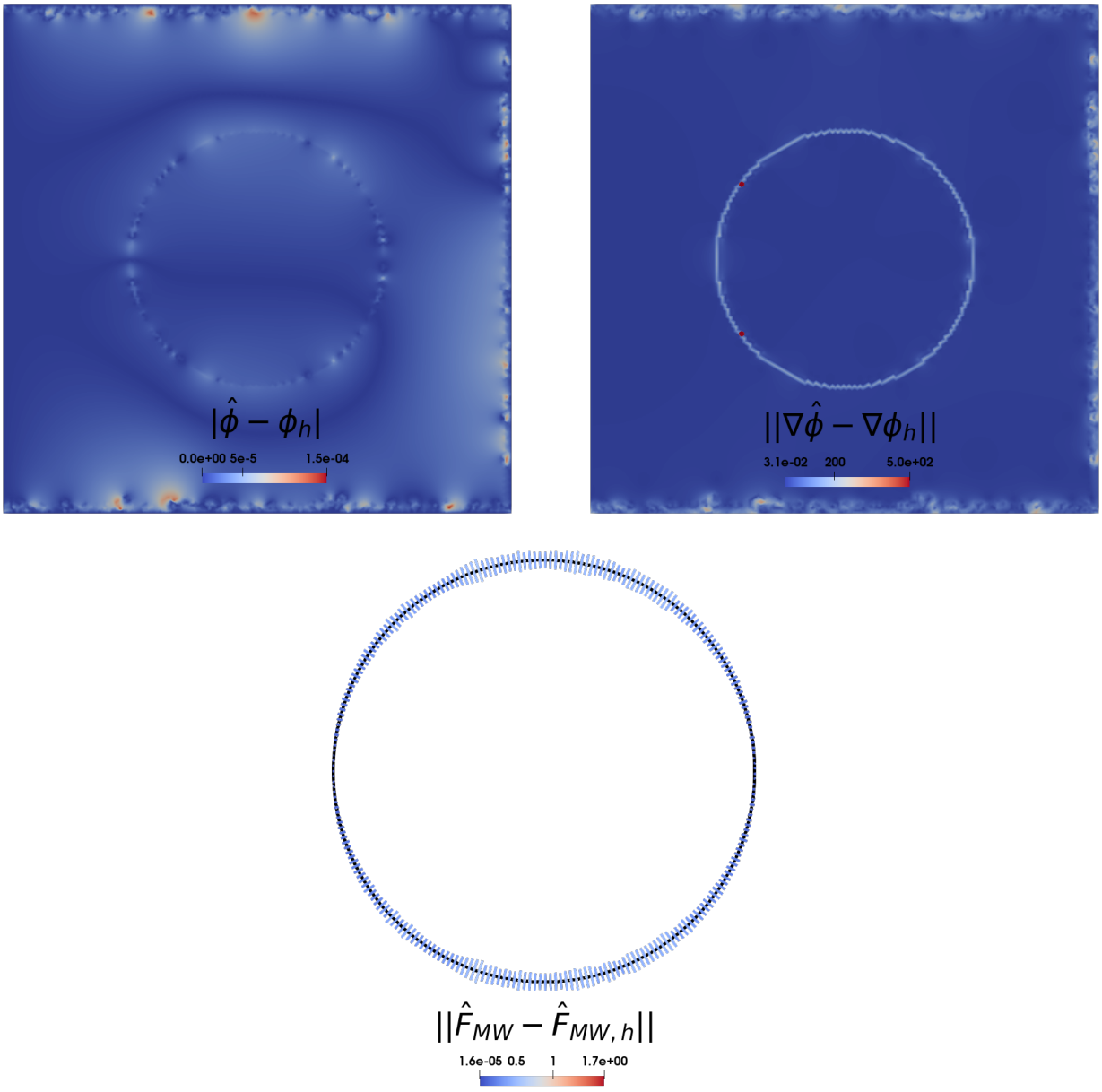


Figure A.15: Distribution of error over Ω and Σ at first iteration ($t = 0$).

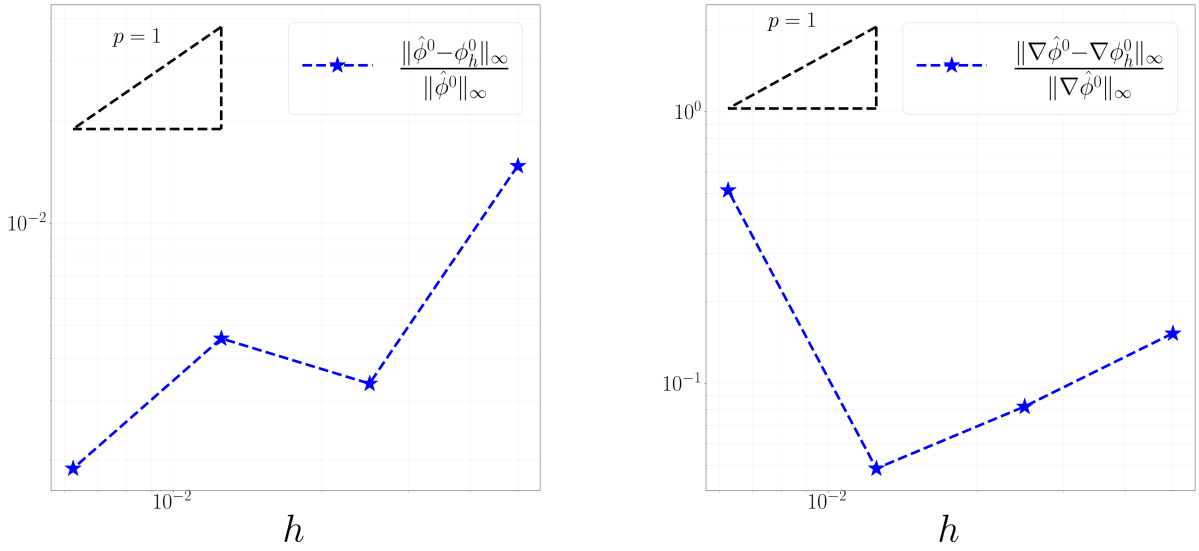
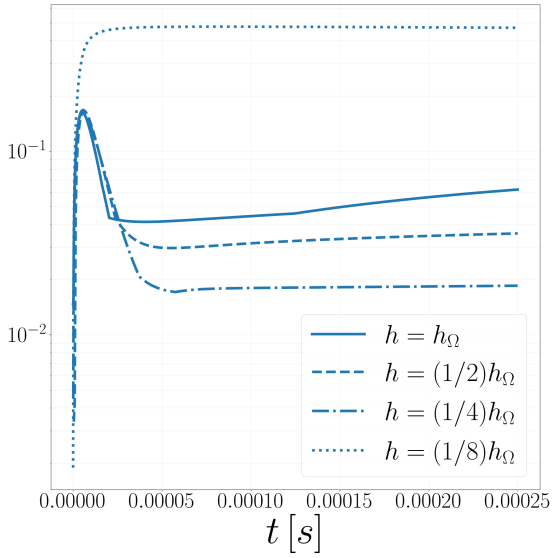
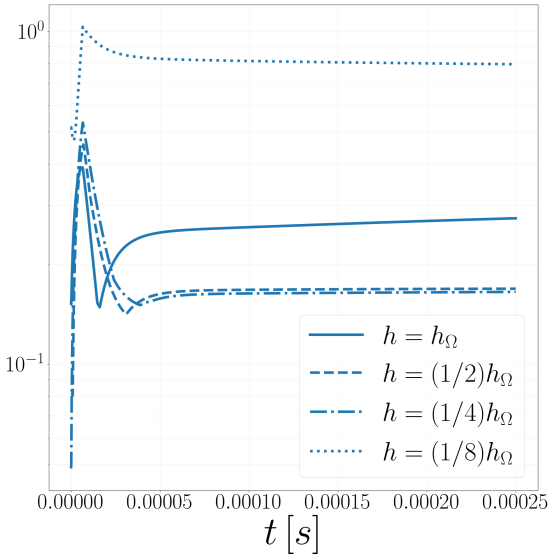


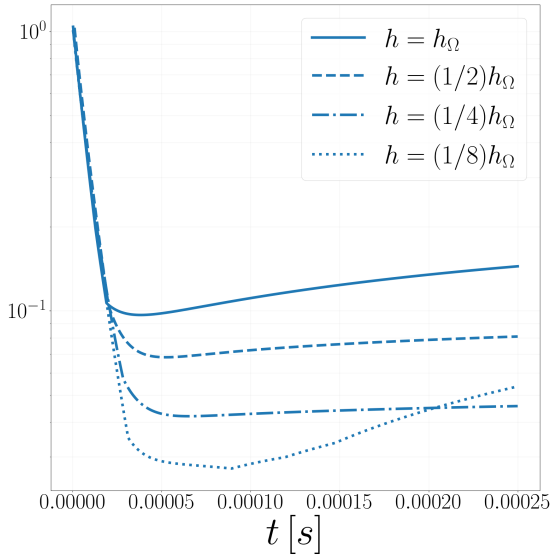
Figure A.16: Error at first iteration ($t = 0$) against mesh size h, norm L_∞. Comparison with a reference trend h^p, with $p = 1$.



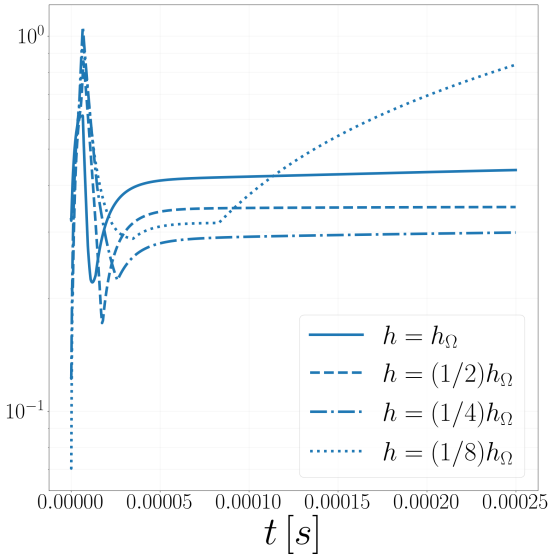
(a) $\frac{\|\hat{\phi}-\phi_h\|_\infty}{\|\hat{\phi}\|_\infty}(t)$



(b) $\frac{\|\nabla\hat{\phi}-\nabla\phi_h\|_\infty}{\|\nabla\hat{\phi}\|_\infty}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_\infty}{\|\hat{J}_\Sigma\|_\infty}(t)$



(d) $\frac{\|\hat{\boldsymbol{F}}_{MW}-\boldsymbol{F}_{MW,h}\|_\infty}{\|\hat{\boldsymbol{F}}_{MW}\|_\infty}(t)$

Figure A.17: Time evolution of error in norm L_∞.

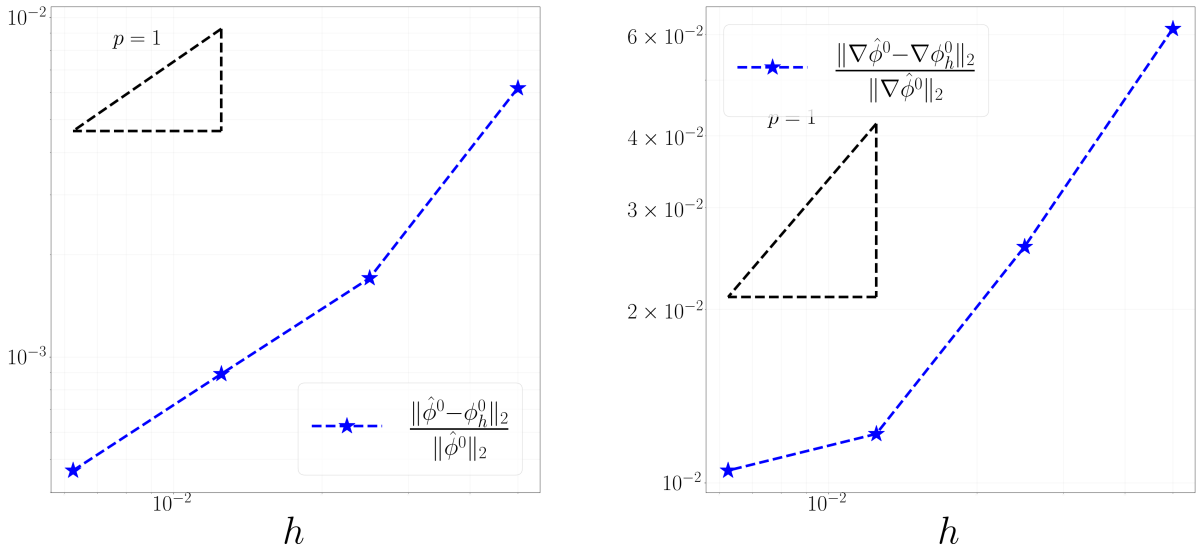
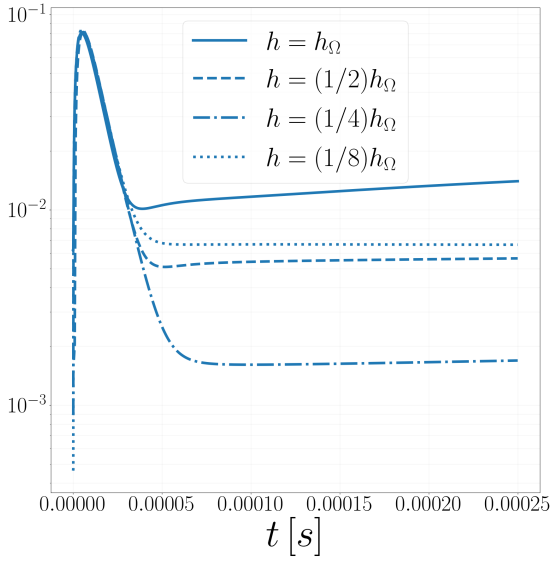
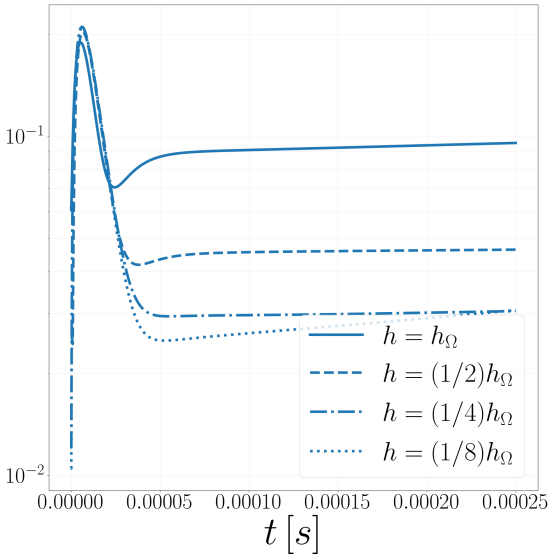


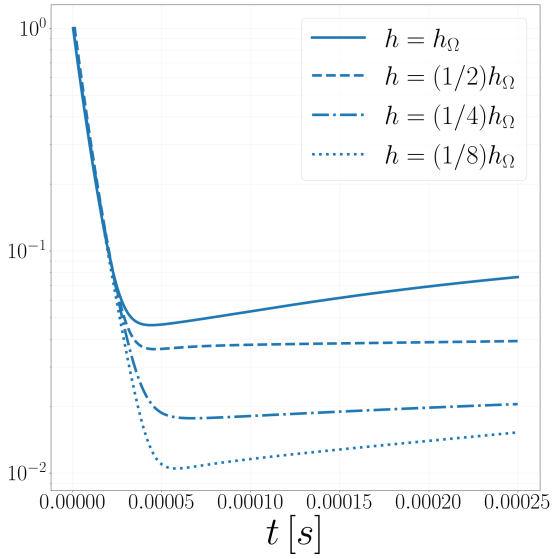
Figure A.18: Error at first iteration ($t = 0$) against mesh size h, norm L_∞. Comparison with a reference trend h^p, with $p = 1$.



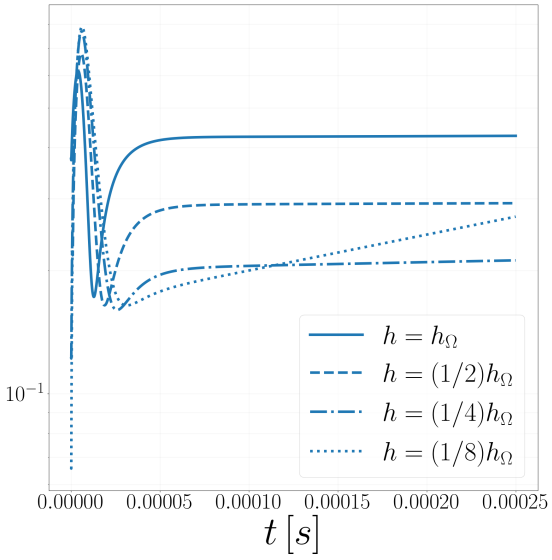
(a) $\frac{\|\hat{\phi}-\phi_h\|_2}{\|\hat{\phi}\|_2}(t)$



(b) $\frac{\|\nabla\hat{\phi}-\nabla\phi_h\|_2}{\|\nabla\hat{\phi}\|_2}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_2}{\|\hat{J}_\Sigma\|_2}(t)$



(d) $\frac{\|\hat{\boldsymbol{F}}_{MW}-\boldsymbol{F}_{MW,h}\|_2}{\|\hat{\boldsymbol{F}}_{MW}\|_2}(t)$

Figure A.19: Time evolution of error in norm L_2.

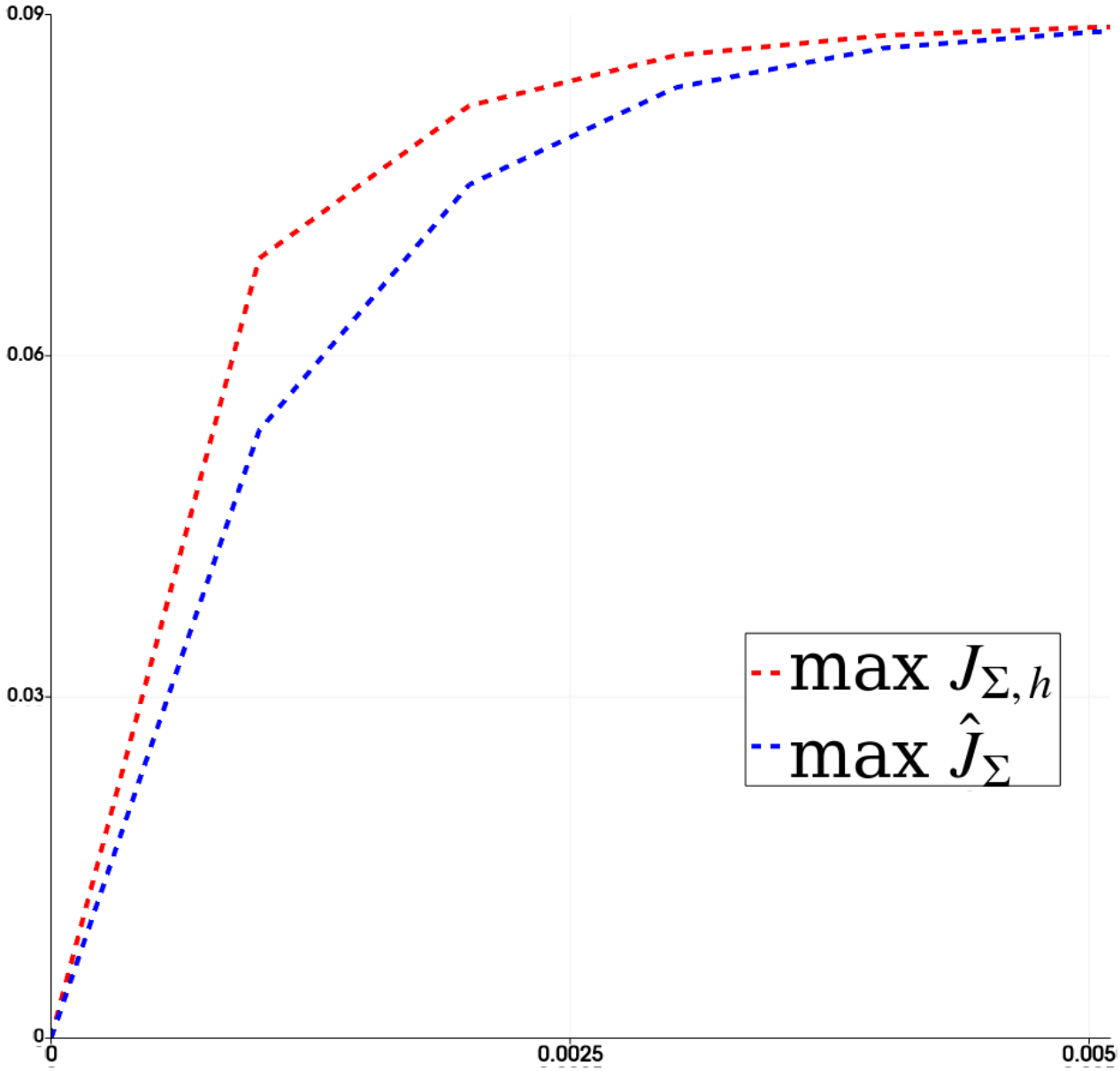


Figure A.20: Maximum potential jump over time. Comparison of exact and numerical solution.

A.2 $d = 3$

Each convergence test is performed considering a family of unstructured tetrehedral meshes $\{\Omega_h\}_{h\in\mathcal{H}_\Omega}$ with element diameters in $\mathcal{H}_\Omega = \{h_\Omega/2^k,\ 1 \leq k \leq 4\}$, where $h_\Omega = 4R/10$ is the element diameter of the coarsest mesh. We consider a corresponding family of triangulations of the sphere $\{\Sigma_h\}_{h\in\mathcal{H}_\Sigma}$ with $\mathcal{H}_\Sigma = \{h_\Sigma/2^k, , 1 \leq\leq 4\}$, where $h_\Sigma =$ is the face diameter of the coarsest surface mesh.

The convergence slopes are commented in more detail for the case $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 1$. For the remaining two cases similar observations hold.

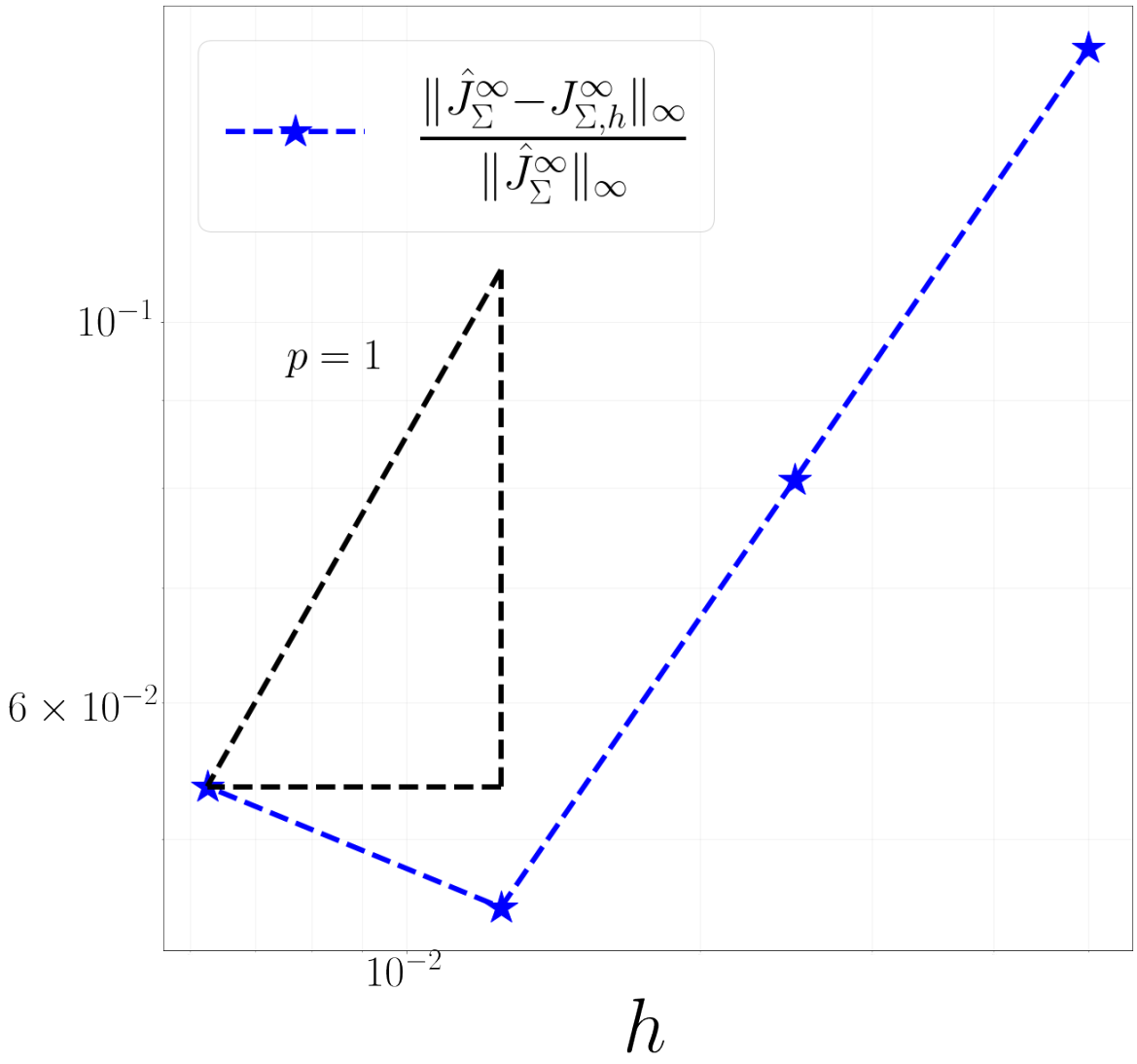


Figure A.21: Error on J_Σ at the end of the simulation against mesh size h, norm L_∞.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 1$

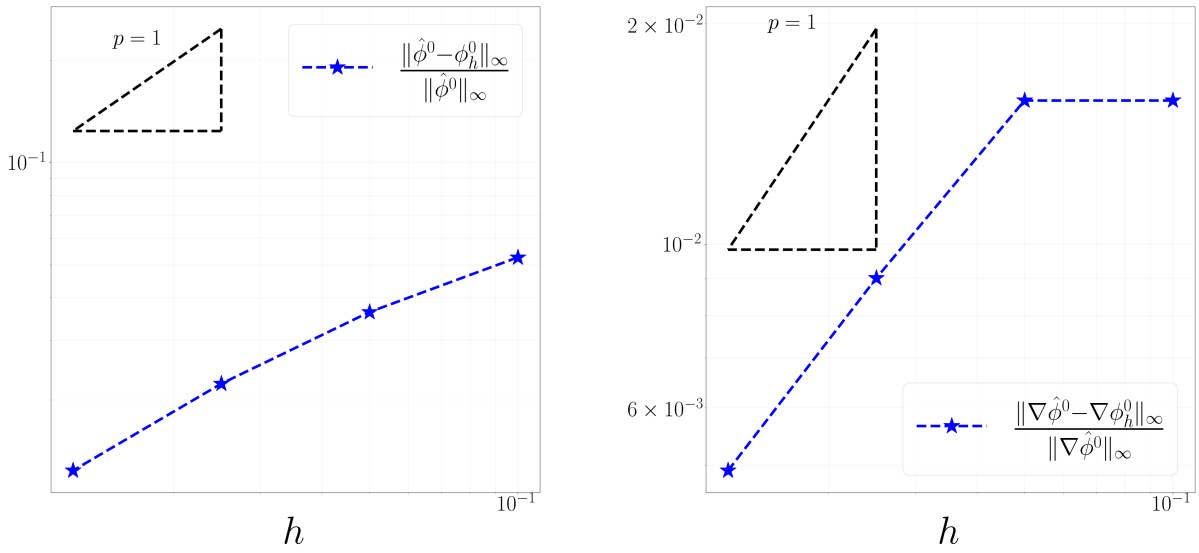
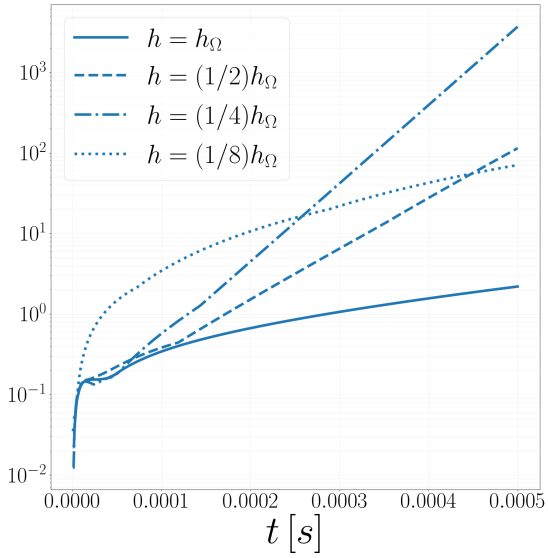
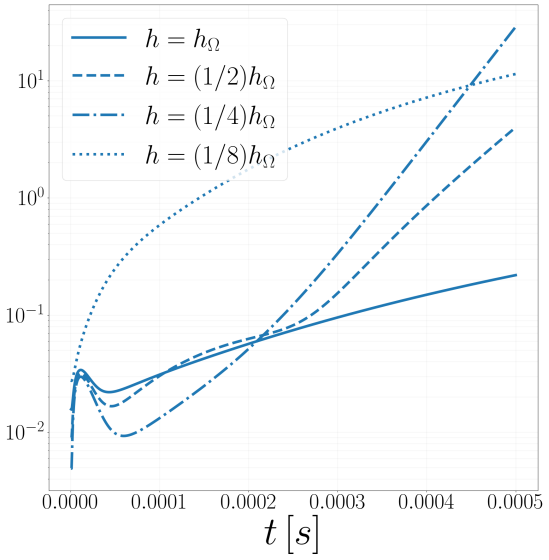


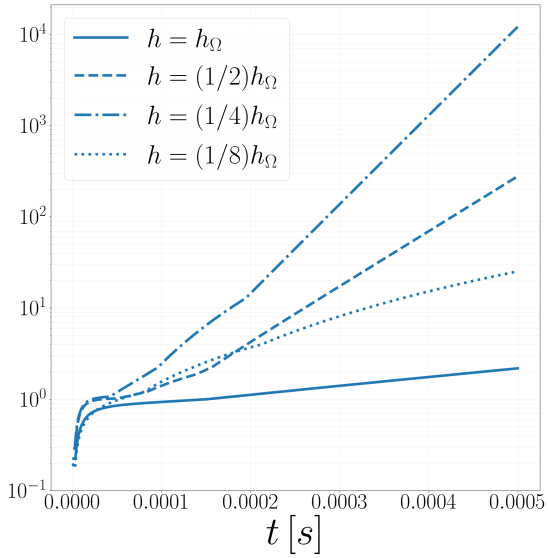
Figure A.22: Initial ($t = 0$) error on potential ϕ, electric field $\nabla\phi$ and electric forces $\boldsymbol{F}_{MW}$ on a mesh family with converging mesh size h for $t = 0$. Comparison with a reference trend h^p, with $p = 1$.



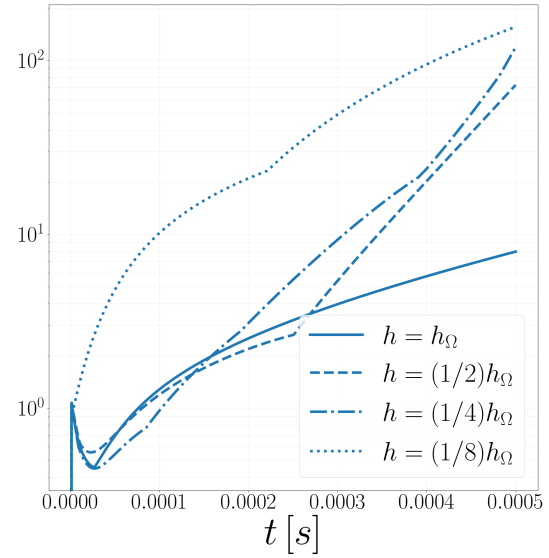
(a) $\frac{\|\hat{\phi}-\phi_h\|_\infty}{\|\hat{\phi}\|_\infty}(t)$



(b) $\frac{\|\hat{\nabla\phi}-\nabla\phi_h\|_\infty}{\|\hat{\nabla\phi}\|_\infty}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_\infty}{\|\hat{J}_\Sigma\|_\infty}(t)$



(d) $\frac{\|\hat{\boldsymbol{F}}_{MW}-\boldsymbol{F}_{MW,h}\|_\infty}{\|\hat{\boldsymbol{F}}_{MW}\|_\infty}(t)$

Figure A.23: Evolution of the L_∞ error for ϕ, $\nabla\phi$, J_Σ and $\boldsymbol{F}_{MW}$ along time for several refinement levels of the mesh. The numerical solution appears to diverge.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 0.1$

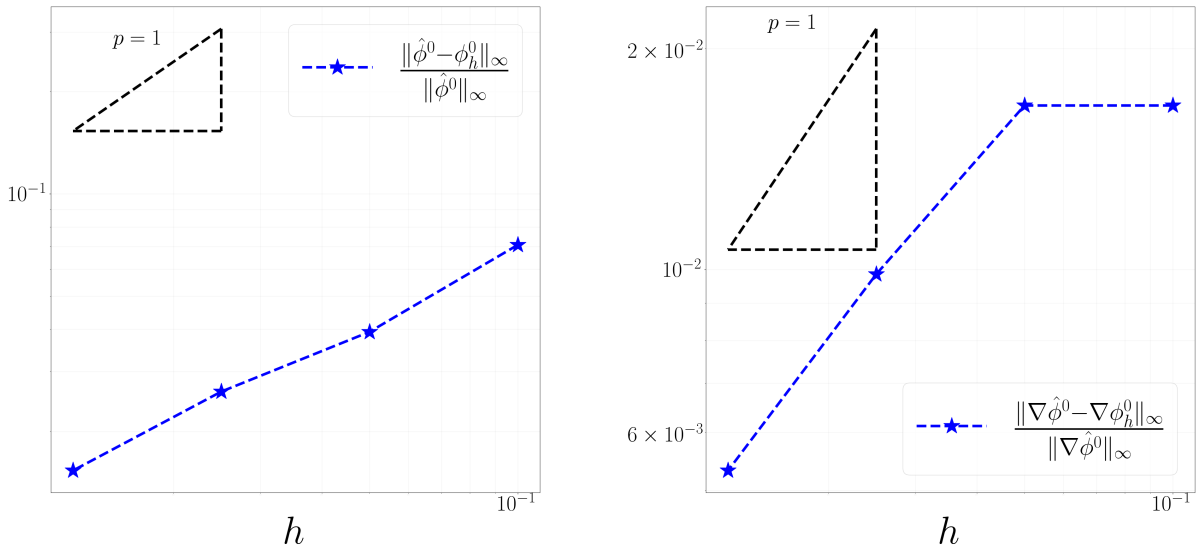
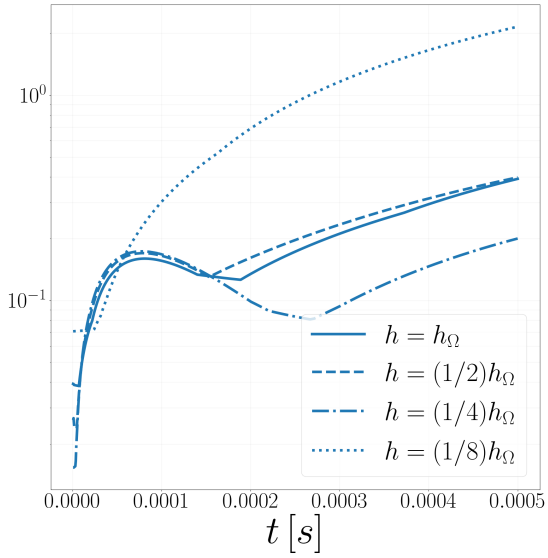
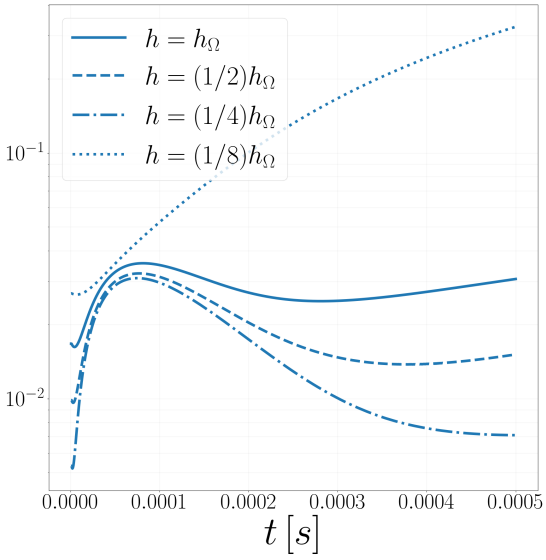


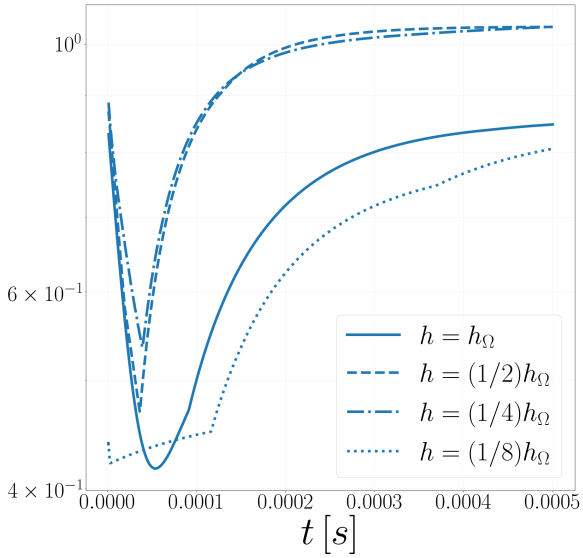
Figure A.24: Initial ($t = 0$) error on potential ϕ, electric field $\nabla\phi$ and electric forces $\boldsymbol{F}_{MW}$ on a mesh family with converging mesh size h for $t = 0$. Comparison with a reference trend h^p, with $p = 1$.



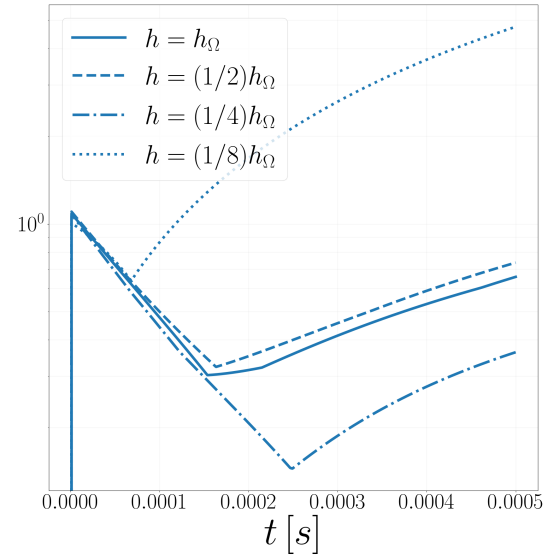
(a) $\frac{\|\hat{\phi}-\phi_h\|_\infty}{\|\hat{\phi}\|_\infty}(t)$



(b) $\frac{\|\hat{\nabla\phi}-\nabla\phi_h\|_\infty}{\|\hat{\nabla\phi}\|_\infty}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_\infty}{\|\hat{J}_\Sigma\|_\infty}(t)$



(d) $\frac{\|\hat{\boldsymbol{F}}_{MW}-\boldsymbol{F}_{MW,h}\|_\infty}{\|\hat{\boldsymbol{F}}_{MW}\|_\infty}(t)$

Figure A.25: Evolution of the L_∞ error for ϕ, $\nabla\phi$, J_Σ and $\boldsymbol{F}_{MW}$ along time for several refinemen,t levels of the mesh.

Conductivity Ratio: $\frac{\sigma_{\text{int}}}{\sigma_{\text{ext}}} = 10$

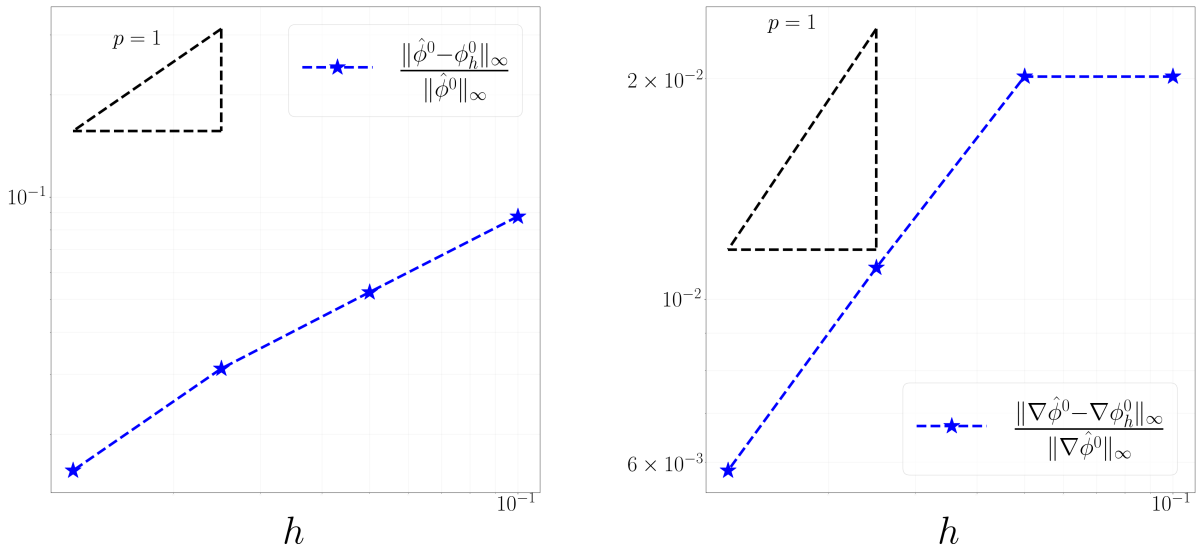
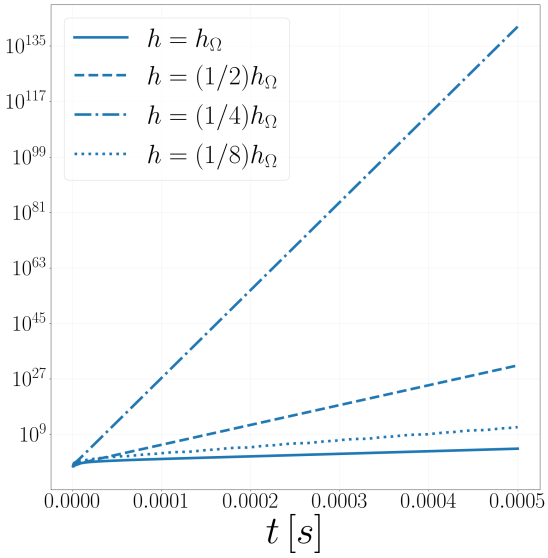
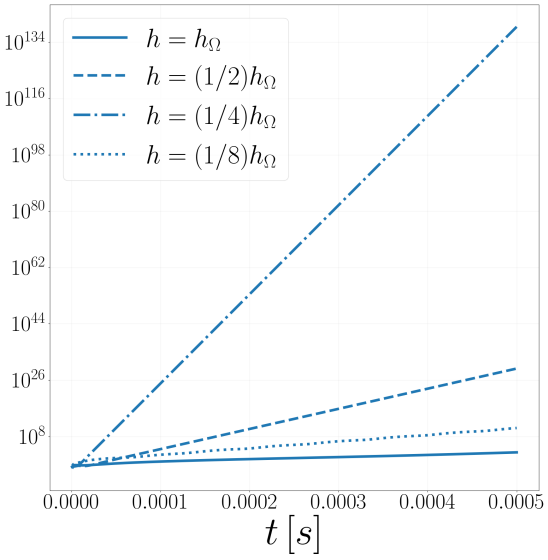


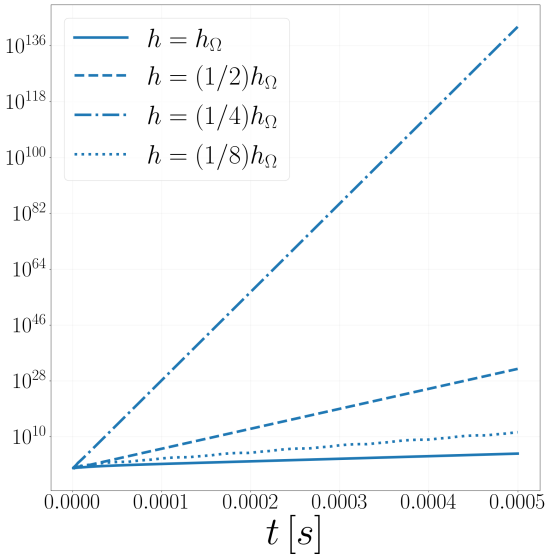
Figure A.26: Initial ($t = 0$) error on potential ϕ, electric field $\nabla\phi$ and electric forces $\boldsymbol{F}_{MW}$ on a mesh family with converging mesh size h for $t = 0$. Comparison with a reference trend h^p, with $p = 1$.



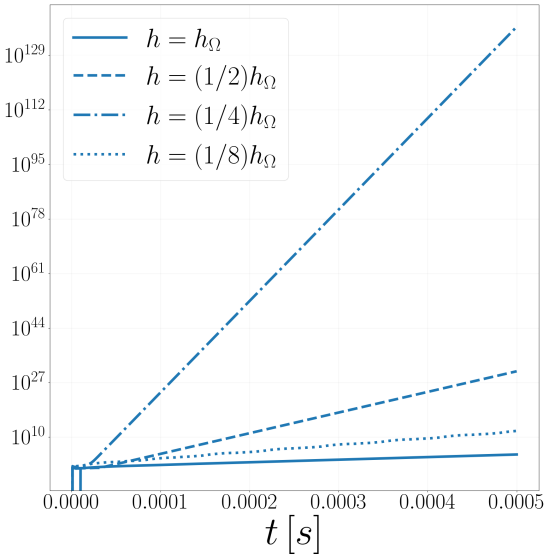
(a) $\frac{\|\hat{\phi}-\phi_h\|_\infty}{\|\hat{\phi}\|_\infty}(t)$



(b) $\frac{\|\hat{\nabla\phi}-\nabla\phi_h\|_\infty}{\|\hat{\nabla\phi}\|_\infty}(t)$



(c) $\frac{\|\hat{J}_\Sigma-J_{\Sigma,h}\|_\infty}{\|\hat{J}_\Sigma\|_\infty}(t)$



(d) $\frac{\|\hat{\boldsymbol{F}}_{MW}-\boldsymbol{F}_{MW,h}\|_\infty}{\|\hat{\boldsymbol{F}}_{MW}\|_\infty}(t)$

Figure A.27: Evolution of the L_∞ error for ϕ, $\nabla\phi$, J_Σ and $\boldsymbol{F}_{MW}$ along time for several refinemen,t levels of the mesh.

BIBLIOGRAPHY

Adjerid, S., I. Babuška, R. Guo, and T. Lin (2023). "An enriched immersed finite element method for interface problems with nonhomogeneous jump conditions". In: *Comput. Meth. Appl. Mech. Engrg.* 404.115770. DOI: 10.1016/j.cma.2022.115770.

Afonso, M. M., S. Mendez, and F. Nicoud (2014). "On the damped oscillations of an elastic quasi-circular membrane in a two-dimensional incompressible fluid". In: *Journal of Fluid Mechanics* 746, pp. 300–331.

Aghili, J., S. Boyaval, and D. A. Di Pietro (2015). "Hybridization of mixed high-order methods on general meshes and application to the Stokes equations". In: *Comput. Meth. Appl. Math.* 15.2, pp. 111–134. DOI: 10.1515/cmam-2015-0004.

Antonietti, P. F., S. Giani, and P. Houston (2013). "hp-version composite discontinuous Galerkin methods for elliptic problems on complicated domains". In: *SIAM J. Sci. Comput.* 35.3, A1417–A1439. DOI: 10.1137/120877246.

Arnold, D. (2018). *Finite Element Exterior Calculus*. SIAM. DOI: 10.1137/1.9781611975543.

Bagchi, P. (2007). "Mesoscale simulation of blood flow in small vessels". In: *Biophysical journal* 92.6, pp. 1858–1877.

Barthes-Biesel, D. (1980). "Motion of a spherical microcapsule freely suspended in a linear shear flow". In: *Journal of Fluid Mechanics* 100.4, pp. 831–853.

Barthès-Biesel, D. (2009). "Capsule motion in flow: Deformation and membrane buckling". In: *Comptes Rendus Physique* 10.8, pp. 764–774.

Barthes-Biesel, D., A. Diaz, and E. Dhenin (2002). "Effect of constitutive laws for two-dimensional membranes on flow-induced capsule deformation". In: *Journal of Fluid Mechanics* 460, pp. 211–222.

Bassi, F., L. Botti, A. Colombo, D. A. Di Pietro, and P. Tesini (2012). "On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations". In: *J. Comput. Phys.* 231.1, pp. 45–65. DOI: 10.1016/j.jcp.2011.08.018.

Boggs, P. T. and J. W. Tolle (1995). "Sequential quadratic programming". In: *Acta numerica* 4, pp. 1–51.

Bonaldi, F., D. A. Di Pietro, J. Droniou, and K. Hu (Mar. 2023). *An exterior calculus framework for polytopal methods*. arXiv: 2303.11093 [math.NA].

Botti, L. and D. A. Di Pietro (2018). "Numerical assessment of Hybrid High-Order methods on curved meshes and comparison with discontinuous Galerkin methods". In: *J. Comput. Phys.* 370, pp. 58–84. DOI: 10.1016/j.jcp.2018.05.017.

Botti, M., D. A. Di Pietro, and A. Guglielmana (2019). "A low-order nonconforming method for linear elasticity on general meshes". In: *Comput. Meth. Appl. Mech. Engrg.* 354, pp. 96–118. DOI: 10.1016/j.cma.2019.05.031.

Botti, M., D. A. Di Pietro, and M. Salah (2023). "A serendipity fully discrete div-div complex on polygonal meshes". In: *Comptes Rendus Mécanique* 351.S1. DOI: 10.5802/crmeca.150. arXiv: 2207.07194 [math.NA].

Breyiannis, G and C Pozrikidis (2000). “Simple shear flow of suspensions of elastic capsules”. In: *Theoretical and Computational Fluid Dynamics* 13.5, pp. 327–347.

Burman, E., S. Claus, P. Hansbo, M. G. Larson, and A. Massing (2015). “CutFEM: Discretizing geometry and partial differential equations”. In: *Int. J. Numer. Meth. Engng* 104, pp. 472–501. DOI: 10.1002/nme.4823.

Burman, E., M. Cicuttin, G. Delay, and A. Ern (2021). “An unfitted hybrid high-order method with cell agglomeration for elliptic interface problems”. In: *SIAM Journal on Scientific Computing* 43.2, A859–A882.

Burman, E. and A. Ern (2018). “An unfitted hybrid high-order method for elliptic interface problems”. In: *SIAM J. Numer. Anal.* 56.3, pp. 1525–1546. DOI: 10.1137/17M1154266.

— (2019). “A Cut Cell Hybrid High-Order Method for Elliptic Problems with Curved Boundaries”. In: *Numerical Mathematics and Advanced Applications ENUMATH 2017*. Ed. by F. A. Radu, K. Kumar, I. Berre, J. M. Nordbotten, and I. S. Pop. Cham: Springer International Publishing, pp. 173–181. ISBN: 978-3-319-96415-7.

Burman, E. and P. Zunino (2006). “A Domain Decomposition Method Based on Weighted Interior Penalties for Advection-Diffusion-Reaction Problems”. In: *SIAM Journal on Numerical Analysis* 44.4, pp. 1612–1638. DOI: 10.1137/050634736.

Cangiani, A., Z. Dong, E. H. Georgoulis, and P. Houston (2017). *hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes*. SpringerBriefs in Mathematics. Springer, Cham, pp. viii+131.

Canham, P. B. (1970). “The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell”. In: *Journal of theoretical biology* 26.1, pp. 61–81.

Castanon Quiroz, D. and D. A. Di Pietro (2020). “A Hybrid High-Order method for the incompressible Navier–Stokes problem robust for large irrotational body forces”. In: *Comput. Math. Appl.* 79.8, pp. 2655–2677. DOI: 10.1016/j.camwa.2019.12.005.

Charrier, J., S Shrivastava, and R Wu (1989). “Free and constrained inflation of elastic membranes in relation to thermoforming—non-axisymmetric problems”. In: *The Journal of Strain Analysis for Engineering Design* 24.2, pp. 55–74.

Chnafa, C., S. Mendez, R. Moreno, and F. Nicoud (2015). “Using image-based CFD to investigate the intracardiac turbulence”. In: *Modeling the Heart and the Circulatory System*, pp. 97–117.

Chorin, A. J. (1997). “A numerical method for solving incompressible viscous flow problems”. In: *Journal of computational physics* 135.2, pp. 118–125.

Collins, R. T., J. J. Jones, M. T. Harris, and O. A. Basaran (2008). “Electrohydrodynamic tip streaming and emission of charged drops from liquid cones”. In: *Nature Physics* 4.2, pp. 149–154.

Coulter, W. H. (1949). *Means for Counting Particles Suspended in a Fluid*.

Das, S. and R. M. Thaokar (2018). “Large-deformation electrohydrodynamics of an elastic capsule in a DC electric field”. In: *Journal of Fluid Mechanics* 841, pp. 489–520.

DeBruin, K. A. and W. Krassowska (1999). “Modeling electroporation in a single cell. I. Effects of field strength and rest potential”. In: *Biophysical journal* 77.3, pp. 1213–1224.

D’Elia, M., D. Ridzal, K. J. Peterson, P. Bochev, and M. Shashkov (2016). “Optimization-based mesh correction with volume and convexity constraints”. In: *Journal of Computational Physics* 313, pp. 455–477.

Desjardins, O., V. Moureau, and H. Pitsch (2008). “An accurate conservative level set/ghost fluid method for simulating turbulent atomization”. In: *Journal of computational physics* 227.18, pp. 8395–8416.

Di Pietro, D. A. and J. Droniou (2018). “A third Strang lemma for schemes in fully discrete formulation”. In: *Calcolo* 55.40. DOI: 10.1007/s10092-018-0282-3.

— (2020). *The Hybrid High-Order method for polytopal meshes. Design, analysis, and applications*. Modeling, Simulation and Application 19. Springer International Publishing. DOI: 10.1007/978-3-030-37203-3.
— (2021). "An arbitrary-order method for magnetostatics on polyhedral meshes based on a discrete de Rham sequence". In: *J. Comput. Phys.* 429.109991. DOI: 10.1016/j.jcp.2020.109991.
— (2023a). "A polytopal method for the Brinkman problem robust in all regimes". In: *Comput. Meth. Appl. Mech. Engrg.* 409.115981. DOI: 10.1016/j.cma.2023.115981.
Di Pietro, D. A., J. Droniou, and F. Rapetti (2020). "Fully discrete polynomial de Rham sequences of arbitrary degree on polygons and polyhedra". In: *Math. Models Methods Appl. Sci.* 30.9, pp. 1809–1855. DOI: 10.1142/S0218202520500372.
Di Pietro, D. A. and A. Ern (2015). "A hybrid high-order locking-free method for linear elasticity on general meshes". In: *Comput. Meth. Appl. Mech. Engrg.* 283, pp. 1–21. DOI: 10.1016/j.cma.2014.09.009.
— (2017). "Arbitrary-order mixed methods for heterogeneous anisotropic diffusion on general meshes". In: *IMA J. Numer. Anal.* 37.1, pp. 40–63. DOI: 10.1093/imanum/drw003.
Di Pietro, D. A., A. Ern, and J.-L. Guermond (2008). "Discontinuous Galerkin methods for anisotropic semidefinite diffusion with advection". In: *SIAM J. Numer. Anal.* 46.2, pp. 805–831. DOI: 10.1137/060676106.
Di Pietro, D. A., A. Ern, and S. Lemaire (2014a). "An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators". In: *Comput. Meth. Appl. Math.* 14.4, pp. 461–472. DOI: 10.1515/cmam-2014-0018.
Di Pietro, D. A., A. Ern, A. Linke, and F. Schieweck (2016). "A discontinuous skeletal method for the viscosity-dependent Stokes problem". In: *Comput. Meth. Appl. Mech. Engrg.* 306, pp. 175–195. DOI: 10.1016/j.cma.2016.03.033.
Di Pietro, D. A., M.-L. Hanot, and M. Salah (July 2024). *Serendipity discrete complexes with enhanced regularity*. arXiv: 2407.12625 [math.NA].
Di Pietro, D. A. and J. Droniou (2023b). "An arbitrary-order discrete de Rham complex on polyhedral meshes: Exactness, Poincaré inequalities, and consistency". In: *Found. Comput. Math.* 23, pp. 85–164. DOI: 10.1007/s10208-021-09542-8.
Di Pietro, D. A., A. Ern, and S. Lemaire (2014b). "An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators". In: *Computational Methods in Applied Mathematics* 14.4, pp. 461–472.
Droniou, J. and L. Yemm (2022). "Robust hybrid high-order method on polytopal meshes with small faces". In: *Comput. Meth. Appl. Math.* 22 (1), pp. 47–71. DOI: 10.1515/cmam-2021-0018.
Farutin, A., T. Biben, and C. Misbah (2014). "3D numerical simulations of vesicle and inextensible capsule dynamics". In: *Journal of Computational Physics* 275, pp. 539–568.
Fedkiw, R. P., T. Aslam, B. Merriman, and S. Osher (1999). "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)". In: *Journal of computational physics* 152.2, pp. 457–492.
Fedosov, D. A., B. Caswell, and G. E. Karniadakis (2010). "A multiscale red blood cell model with accurate mechanics, rheology, and dynamics". In: *Biophysical journal* 98.10, pp. 2215–2225.
Gibaud, E., J. Siguenza, S. Mendez, and F. Nicoud (2013). "Towards numerical prediction of red blood cells dynamics within a cytometer". In: *Computer Methods in Biomechanics and Biomedical Engineering* 16.sup1, pp. 9–10.
Grover, N., J Naaman, S Ben-Sasson, and F Doljanski (1969). "Electrical sizing of particles in suspensions: I. theory". In: *Biophysical journal* 9.11, pp. 1398–1414.
Hardeman, M., J. Dobbe, and C Ince (2001). "The Laser-assisted Optical Rotational Cell Analyzer (LORCA) as red blood cell aggregometer". In: *Clinical hemorheology and microcirculation* 25.1, pp. 1–11.

Helfrich, W. (1973a). “Elastic properties of lipid bilayers: theory and possible experiments”. In: *Zeitschrift für Naturforschung c* 28.11-12, pp. 693–703.

— (1973b). “Elastic properties of lipid bilayers: theory and possible experiments”. In: *Zeitschrift für Naturforschung c* 28.11-12, pp. 693–703.

Huang, P., H. Wu, and Y. Xiao (2017). “An unfitted interface penalty finite element method for elliptic interface problems”. In: *Comput. Methods Appl. Mech. Engrg.* 323, pp. 439–460. ISSN: 0045-7825,1879-2138. DOI: 10.1016/j.cma.2017.06.004.

Hurley, J. (1970). “Sizing particles with a Coulter counter”. In: *Biophysical journal* 10.1, pp. 74–79.

Jiang, K. and C. T. Lim (2019). “Microfluidic biotechnologies for hematology: separation, disease detection and diagnosis”. In: *Dynamics of Blood Cell Suspensions in Microflows*. CRC Press, pp. 343–370.

Johansson, A. and M. G. Larson (2013). “A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary”. In: *Numer. Math.* 123.4, pp. 607–628. ISSN: 0029-599X,0945-3245. DOI: 10.1007/s00211-012-0497-1.

Kachel, V, H Metzger, and G Ruhenstroth-Bauer (1970). “The influence of the particle line on the volume distribution according to the Coulter principle”. In: *Zeitschrift für die gesamte experimentelle Medizin einschließlich experimentelle Chirurgie* 153, pp. 331–347.

Kataoka, K., A. Harada, and Y. Nagasaki (2012). “Block copolymer micelles for drug delivery: design, characterization and biological significance”. In: *Advanced drug delivery reviews* 64, pp. 37–48.

Kim, J. and P. Moin (1985). “Application of a fractional-step method to incompressible Navier-Stokes equations”. In: *Journal of computational physics* 59.2, pp. 308–323.

Kolahdouz, E. M. and D. Salac (2015). “A numerical model for the trans-membrane voltage of vesicles”. In: *Applied Mathematics Letters* 39, pp. 7–12.

Kuchel, P. W., C. D. Cox, D. Daners, D. Shishmarev, and P. Galvosas (2021). “Surface model of the human red blood cell simulating changes in membrane curvature under strain”. In: *Scientific Reports* 11.1, p. 13712.

LeVeque, R. J. and Z. Li (1994). “The immersed interface method for elliptic equations with discontinuous coefficients and singular sources”. In: *SIAM Journal on Numerical Analysis* 31.4, pp. 1019–1044.

Ling, H. J., S. Bru, J. Puig, F. Vixège, S. Mendez, F. Nicoud, P.-Y. Courand, O. Bernard, and D. Garcia (2024). “Physics-Guided Neural Networks for Intraventricular Vector Flow Mapping”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*.

Liu, W. K., S. Jun, and Y. F. Zhang (1995). “Reproducing kernel particle methods”. In: *International journal for numerical methods in fluids* 20.8-9, pp. 1081–1106.

Liu, X.-D., R. P. Fedkiw, and M. Kang (2000). “A boundary condition capturing method for Poisson’s equation on irregular domains”. In: *Journal of computational Physics* 160.1, pp. 151–178.

Loiseau, E., G. Massiera, S. Mendez, P. A. Martinez, and M. Abkarian (2015). “Microfluidic study of enhanced deposition of sickle cells at acute corners”. In: *Biophysical journal* 108.11, pp. 2623–2632.

Matteoli, P, F. Nicoud, and S. Mendez (2021). “Impact of the membrane viscosity on the tank-treading behavior of red blood cells”. In: *Physical Review Fluids* 6.4, p. 043602.

Mauer, J., S. Mendez, L. Lanotte, F. Nicoud, M. Abkarian, G. Gompper, and D. A. Fedosov (2018). “Flow-induced transitions of red blood cell shapes under shear”. In: *Physical review letters* 121.11, p. 118103.

Maxwell, J. C. (1873). *A treatise on electricity and magnetism*. Vol. 1. Clarendon press.

McConnell, L. C., M. J. Miksis, and P. M. Vlahovska (2013). “Vesicle electrohydrodynamics in DC electric fields”. In: *IMA Journal of Applied Mathematics* 78.4, pp. 797–817.

Mendez, S. and M. Abkarian (2018). “In-plane elasticity controls the full dynamics of red blood cells in shear flow”. In: *Physical Review Fluids* 3.10, p. 101101.

— (2019). “Single red blood cell dynamics in shear flow and its role in hemorheology”. In: *Dynamics of Blood Cell Suspensions in Microflows*. CRC Press, pp. 125–182.

MENDEZ, S., A. BÉROD, C. CHNAFA, and F. NICOUD (2023). “YALES2BIO: un solveur dédié aux écoulements sanguins”. In: *Écoulements biologiques dans les grands vaisseaux: Dialogue entre modélisations numériques et études expérimentales in vitro/in vivo*, p. 185.

Mendez, S., W. Garcia, and A. Nicolas (2023). “From Microscopic Droplets to Macroscopic Crowds: Crossing the Scales in Models of Short-Range Respiratory Disease Transmission, with Application to COVID-19”. In: *Advanced Science* 10.19, p. 2205255.

Mendez, S., E. Gibaud, and F. Nicoud (2014). “An unstructured solver for simulations of deformable particles in flows at arbitrary Reynolds numbers”. In: *Journal of computational physics* 256, pp. 465–483.

Mendez, S., C Iss, D Midou, A Moreau, D Held, A. Charrier, A Viallat, and E Helfer (2019). “A joint numerical and experimental study on the self-organization of red blood cells in confined microfluidic channels”. In: *Computer Methods in Biomechanics and Biomedical Engineering* 22.sup1, S336–S337.

O’Konski, C. and R. Gunther (1955). “Verification of the free energy equation for electrically polarized droplets”. In: *Journal of Colloid Science* 10.6, pp. 563–570. ISSN: 0095-8522. DOI: https://doi.org/10.1016/0095-8522(55)90007-6. URL: https://www.sciencedirect.com/science/article/pii/0095852255900076.

Pepona, M., J. Gounley, and A. Randles (2023). “Effect of constitutive law on the erythrocyte membrane response to large strains”. In: *Computers & Mathematics with Applications* 132, pp. 145–160.

Peskin, C. S. (1972). “Flow patterns around heart valves: a numerical method”. In: *Journal of computational physics* 10.2, pp. 252–271.

— (1977). “Numerical analysis of blood flow in the heart”. In: *Journal of computational physics* 25.3, pp. 220–252.

— (2002). “The immersed boundary method”. In: *Acta numerica* 11, pp. 479–517.

Pitaevskii, L. P. and E. Lifshitz (2012). *Physical Kinetics: Volume 10*. Vol. 10. Butterworth-Heinemann.

Pries, A. R. and T. W. Secomb (2008). “Blood flow in microvascular networks”. In: *Microcirculation*. Elsevier, pp. 3–36.

Qiang, Y., J. Liu, F. Yang, D. Dieujuste, and E Du (2018). “Modeling erythrocyte electrodeformation in response to amplitude modulated electric waveforms”. In: *Scientific reports* 8.1, p. 10224.

Riske, K. A. and R. Dimova (2005). “Electro-deformation and poration of giant vesicles viewed with high temporal resolution”. In: *Biophysical journal* 88.2, pp. 1143–1155.

Saville, D. (1997). “Electrohydrodynamics: the Taylor-Melcher leaky dielectric model”. In: *Annual review of fluid mechanics* 29.1, pp. 27–64.

Schwalbe, J. T., P. M. Vlahovska, and M. J. Miksis (2011). “Vesicle electrohydrodynamics”. In: *Physical Review E* 83.4, p. 046309.

Seifert, U. (1997). “Configurations of fluid membranes and vesicles”. In: *Advances in physics* 46.1, pp. 13–137.

Seiwert, J. and P. M. Vlahovska (2013). “Instability of a fluctuating membrane driven by an ac electric field”. In: *Physical Review E* 87.2, p. 022713.

Sheng, P. and W. Wen (2012). “Electrorheological fluids: mechanisms, dynamics, and microfluidics applications”. In: *Annual review of fluid mechanics* 44.1, pp. 143–174.

Sigüenza, J, S Mendez, and F Nicoud (2014). “Characterisation of a dedicated mechanical model for red blood cells: numerical simulations of optical tweezers experiment”. In: *Computer Methods in Biomechanics and Biomedical Engineering* 17.sup1, pp. 28–29.

Sigüenza, J., S. Mendez, D. Ambard, F. Dubois, F. Jourdan, R. Mozul, and F. Nicoud (2016). "Validation of an immersed thick boundary method for simulating fluid–structure interactions of deformable membranes". In: *Journal of Computational Physics* 322, pp. 723–746.

Skalak, R, A Tozeren, R. Zarda, and S Chien (1973). "Strain energy function of red blood cell membranes". In: *Biophysical journal* 13.3, pp. 245–264.

Strouboulis, T., I. Babuška, and K. Copps (2000). "The design and analysis of the generalized finite element method". In: *Comput. Methods Appl. Mech. Engrg.* 181, pp. 43–69. DOI: 10.1016/S0045-7825(99)00072-9.

Sukumar, N., N. Moes, B. Moran, and T. Belytschko (2000). "Extended finite element method for three dimensional crack modelling". In: *Int. J. Numer. Methods Engrg.* 48.11, pp. 1549–1570. DOI: 10.1002/1097-0207.

Taraconat, P. (2020). "Application of numerical simulation for a better characterization of red blood cells by impedance measurement". Thèse de doctorat dirigée par Nicoud, Franck et Mendez, Simon Mathématiques et Modélisation Montpellier 2020. PhD thesis. URL: http://www.theses.fr/2020MONTS006/document.

Taraconat, P., J.-P. Gineys, D. Isebe, F. Nicoud, and S. Mendez (2019). "Numerical simulation of deformable particles in a Coulter counter". In: *International Journal for Numerical Methods in Biomedical Engineering* 35.11, e3243.

Taraconat, P., J.-p. Gineys, D. Isèbe, F. Nicoud, and S. Mendez (2021). "Detecting cells rotations for increasing the robustness of cell sizing by impedance measurements, with or without machine learning". In: *Cytometry Part A* 99.10, pp. 977–986.

Taraconat, P., J.-P. Gineys, D. Isebe, F. Nicoud, and S. Mendez (2023a). "Red blood cell rheology during a complete blood count: A proof of concept". In: *Plos one* 18.1, e0280952.

— (2023b). "Red blood cell rheology during a complete blood count: A proof of concept". In: *Plos one* 18.1, e0280952.

Taylor, G. I. (1964). "Disintegration of water drops in an electric field". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 280.1382, pp. 383–397.

— (1966). "Studies in electrohydrodynamics. I. The circulation produced in a drop by an electric field". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 291.1425, pp. 159–166.

Torza, S, R. Cox, and S. Mason (1971). "Electrohydrodynamic deformation and bursts of liquid drops". In: *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* 269.1198, pp. 295–319.

Unverdi, S. O. and G. Tryggvason (1992). "A front-tracking method for viscous, incompressible, multi-fluid flows". In: *Journal of computational physics* 100.1, pp. 25–37.

Valet, C., M. Magnen, L. Qiu, S. J. Cleary, K. M. Wang, S. Ranucci, E. Grockowiak, R. Boudra, C. Conrad, Y. Seo, et al. (2022). "Sepsis promotes splenic production of a protective platelet pool with high CD40 ligand expression". In: *The Journal of clinical investigation* 132.7.

Vantieghem, S. (2011). "Numerical simulations of quasi-static magnetohydrodynamics using an unstructured finite volume solver: development and applications". In: *Université Libre de Bruxelles. Cité*, p. 71.

Veiga, L. Beirão da, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo (2013). "Basic principles of virtual element methods". In: *Math. Models Methods Appl. Sci. (M3AS)* 199.23, pp. 199–214. DOI: 10.1142/S0218202512500492.

Veiga, L. Beirão da, A. Russo, and G. Vacca (2019). "The Virtual Element Method with curved edges". In: *ESAIM: Math. Model Numer. Anal.* 53.2, pp. 375–404.

Viallat, A. and M. Abkarian (2019). *Dynamics of blood cell suspensions in microflows*. CRC Press.

Vlahovska, P. M. (2019a). "Electrohydrodynamics of drops and vesicles". In: *Annual Review of Fluid Mechanics* 51, pp. 305–330.

— (2019b). “Electrohydrodynamics of drops and vesicles”. In: *Annual Review of Fluid Mechanics* 51, pp. 305–330.

Vlahovska, P. M., T. Podgorski, and C. Misbah (2009a). “Vesicles and red blood cells in flow: From individual dynamics to rheology”. In: *Comptes Rendus Physique* 10.8, pp. 775–789.

— (2009b). “Vesicles and red blood cells in flow: From individual dynamics to rheology”. In: *Comptes Rendus Physique* 10.8, pp. 775–789.

Vukčević, V., H. Jasak, and I. Gatin (2017). “Implementation of the ghost fluid method for free surface flows in polyhedral finite volume framework”. In: *Computers & fluids* 153, pp. 1–19.

Wilson, C. and G. Taylor (1925). “The bursting of soap-bubbles in a uniform electric field”. In: *Mathematical proceedings of the Cambridge philosophical society*. Vol. 22. 5. Cambridge University Press, pp. 728–730.

Ye, T., N. Phan-Thien, and C. T. Lim (2016). “Particle-based simulations of red blood cells—A review”. In: *Journal of biomechanics* 49.11, pp. 2255–2266.

Yemm, L. (2024). “A New Approach to Handle Curved Meshes in the Hybrid High-Order Method”. In: *Found. Comput. Math.* 24, pp. 1049–1076. DOI: 10.1007/s10208-023-09615-w.

Yip, R., N. Mohandas, M. R. Clark, S. Jain, S. B. Shohet, and P. R. Dallman (1983). “Red cell membrane stiffness in iron deficiency”. In.

Yu, Z., S. Zhao, M. J. Miksis, and P. M. Vlahovska (2025). “Instability of a fluctuating biomimetic membrane driven by an applied uniform DC electric field”. In: *arXiv preprint arXiv:2502.12551*.

Zhao, S., B. Balu, Z. Yu, M. J. Miksis, and P. M. Vlahovska (2025). “Diffuse-charge dynamics across a capacitive interface in a DC electric field”. In: *arXiv preprint arXiv:2502.11319*.