Numerical analysis of Poisson-Nernst Planck system of equations to study the propagation of a transient signal in neurons

Paul Paragot

Analyse numérique du système d'équations Poisson-Nernst Planck pour étudier la propagation d'un signal transitoire dans les neurones.

Paul PARAGOT

Laboratoire de Mathématiques J.A. Dieudonné

Présentée en vue de l'obtention du grade de docteur en Mathématiques d'Université Côte d'Azur
Dirigée par : Stella Krell
Codirection : Claire Guerrier
Soutenue le : 11 Juin 2024

Devant le jury, composé de :
Franck Boyer, Professeur des Universités, Université Toulouse 3
Yves Coudière, Professeur des Universités, Institut de Mathématiques de Bordeaux
Claire Guerrier, Chargée de Recherche, Université Côte d'Azur
Stella Krell, Maîtresse de Conférences, Université Côte d'Azur
Pascal Omnes, Directeur de Recherche, Université Paris 13
Romain Veltz, Chargé de Recherche, INRIA Sophia Antipolis

Université Côte d'Azur - UFR Sciences
École Doctorale de Sciences Fondamentales et Appliquées

Thèse

pour obtenir le titre de
DOCTEUR EN MATHÉMATIQUES
de l'Université Côte d'Azur

Discipline : Mathématiques

présentée et soutenue par
Paul Paragot

Analyse numérique du système d'équations Poisson-Nernst Planck pour étudier la propagation d'un signal transitoire dans les neurones.

Département de Mathématiques : Laboratoire J.A. Dieudonné

Thèse dirigée par
Stella Krell
et
Claire Guerrier

soutenue le 11 Juin 2024

devant le jury composé de

Franck Boyer	Professeur	Université Toulouse 3	Rapporteur
Yves Coudière	Professeur	Institut de Mathématiques de Bordeaux	Rapporteur
Claire Guerrier	Chargée de Recherche	Université Côte d'Azur	Directrice de thèse
Stella Krell	Maîtresse de Conférences	Université Côte d'Azur	Directrice de thèse
Pascal Omnes	Directeur de Recherche	Université Paris 13	Examinateur
Romain Veltz	Chargé de Recherche	INRIA Sophia Antipolis	Examinateur

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

This dissertation is my own work that I carried out independently and only with the cited references and professional sources.

In Nice, October 2023

Dedication

Tout d'abord, je souhaite remercier mes directrices de thèse Stella et Claire qui m'ont donné cette opportunité professionnelle dont je garderai un beau souvenir malgré les moments de doute, de frustration et de difficultés.

Merci également aux membres du jury qui ont accepté d'en faire partie.

Durant ces années de thèse, j'ai connu des étapes difficiles qui m'ont mis à l'épreuve. J'ai pu recevoir beaucoup de soutien de personnes proches ou non, qui m'ont permis d'aller de l'avant et de finir ce challenge qu'est une thèse. On ne pense jamais aux difficultés qui peuvent surgir lorsqu'on commence une thèse, et c'est pour cela qu'il est important d'être bien entouré et d'être bienveillant envers les autres car, même ne serait-ce qu'un petit encouragement peut tout changer dans le travail d'un doctorant.

C'est pour cela que je tiens à remercier les personnes du laboratoire qui ont été humainement présentes durant ces années. Je pense surtout à Jean-Marc Lacroix, Roland Ruelle, Martine Smolders, Anna Tykhonenko, Elena Di Bernardino, Roland Diel et Sebastian Minjeaud. Je tiens à remercier Olivier Pantz qui, lorsque j'ai apparu dans son bureau, n'a pas hésité une seule seconde à m'aider dans l'implémentation d'un multi-domaine sous FreeFEM++.

Je pense aussi à Thibault et Christian, deux doctorants du labo sur qui j'ai pu compter pendant cette thèse. Également, Florent et Cambyse, deux post-doctorants qui m'ont apporté bienveillance et conseils.

Que dire de mes amis également, Rémy, Marie, Alexis, Jorge, Driss, Joel, Julie, Cédric, Hubert, Alex, Anto, Clément et Steph. Merci d'être là, merci de votre soutien.

Je remercie également ma tante Marie-Paule qui m'a apporté du réconfort et des conseils lorsque j'en avais le plus besoin.

Je ne remercierai jamais assez mes parents qui, depuis toujours, me soutiennent corps et âme dans ce que j'entreprends. Véronique, François, c'est grâce à vous, merci. Je ne vous exprimerai jamais assez ma gratitude.

Enfin, je ne pouvais pas faire des remerciements sans écrire sur toi, Sara. Que dire que tu ne sais pas déjà ? Quels que soient les moments, on s'est battus tous les deux pour finir nos thèses respectives, on s'est épaulés jusqu'au bout. Si ma thèse a pu voir le jour, c'est en grande partie grâce à ton soutien quotidien et à tes conseils. Merci de faire partie de ma vie, ma compagne.

Merci encore à toutes ces personnes.

Abstract. Neuroscientific questions about dendrites include understanding their structural plasticity in response to learning and how they integrate signals. Researchers aim to unravel these aspects to enhance our understanding of neural function and its complexities. This thesis aims at offering numerical insights concerning voltage and ionic dynamics in dendrites. Our primary focus is on modeling neuronal excitation, particularly in dendritic small compartments. We address ionic dynamics following the influx of nerve signals from synapses, including dendritic spines. To accurately represent their small scale, we solve the well-known Poisson-Nernst-Planck (PNP) system of equations, within this real application. The PNP system is widely recognized as the standard model for characterizing the electrodiffusion phenomenon of ions in electrolytes, including dendritic structures. This non-linear system presents challenges in both modeling and computation due to the presence of stiff boundary layers (BL). We begin by proposing numerical schemes based on the Discrete Duality Finite Volumes method (DDFV) to solve the PNP system. This method enables local mesh refinement at the BL, using general meshes. This approach facilitates solving the system on a 2D domain that represents the geometry of dendritic arborization. Additionally, we employ numerical schemes that preserve the positivity of ionic concentrations. Chapters 1 and 2 present the PNP system and the DDFV method along with its discrete operators. Chapter 2 presents a "linear" coupling of equations and investigate its associated numerical scheme. This coupling poses convergence challenges, where we demonstrate its limitations through numerical results. Chapter 3 introduces a "nonlinear" coupling, which enables accurate numerical resolution of the PNP system. Both of couplings are performed using DDFV method. However, in Chapter 3, we demonstrate the accuracy of the DDFV scheme, achieving second-order accuracy in space. Furthermore, we simulate a test case involving the BL. Finally, we apply the DDFV scheme to the geometry of dendritic spines and discuss our numerical simulations by comparing them with 1D existing simulations in the literature. Our approach considers the complexities of 2D dendritic structures. We also introduce two original configurations of dendrites, providing insights into how dendritic spines influence each other, revealing the extent of their mutual influence. Our simulations show the propagation distance of ionic influx during synaptic connections. In Chapter 4, we solve the PNP system over a 2D multi-domain consisting of a membrane, an internal and external medium. This approach allows the modeling of voltage dynamics in a more realistic way, and further helps checking consistency of the results in Chapter 3. To achieve this, we employ the FreeFem++ software to solve the PNP system within this 2D context. We present simulations that correspond to the results obtained in Chapter 3, demonstrating linear summation in a dendrite bifurcation. Furthermore, we investigate signal summation by adding inputs to the membrane of a dendritic branch. We identify an excitability threshold where the voltage dynamics are significantly influenced by the number of inputs. Finally, we also offer numerical illustrations of the BL within the intracellular medium, observing small fluctuations. These results are preliminary, aiming to provide insights into understanding dendritic dynamics. Chapter 5 presents collaborative work conducted during the Cemracs 2022. We focus on a composite finite volume scheme where we aim to derive the Euler equations with source terms on unstructured meshes.

Key-words. Poisson-Nernst Planck equations, Discrete Duality Finite Volumes, ion transport, FreeFem++.

Résumé. Les questions neuroscientifiques concernant les dendrites incluent la compréhension de leur plasticité structurale en réponse à l'apprentissage et la manière dont elles intègrent les signaux. Les chercheurs visent à élucider ces aspects pour améliorer notre compréhension de la fonction neuronale et de ses complexités. Cette thèse vise à offrir des perspectives numériques concernant la dynamique du voltage et des ions dans les dendrites. Notre objectif est de modéliser l'excitation neuronale dans les dendrites. Nous abordons la dynamique ionique suite à l'afflux de signaux nerveux. Pour les simuler précisémment, nous résolvons le système d'équations Poisson-Nernst-Planck (PNP). Le système PNP est reconnu comme le modèle standard pour caractériser le phénomène d'électrodiffusion des ions dans les électrolytes, y compris les structures dendritiques. Ce système non linéaire présente des défis en modélisation et en calcul en raison de la présence de couches limites rigides (BL). Nous proposons des schémas numériques basés sur la méthode des volumes finis Discrete Duality Finite Volumes (DDFV) pour résoudre le système PNP. Elle permet un raffinement local du maillage au niveau de la BL, en utilisant des maillages généraux. Cette approche facilite la résolution du système sur un domaine 2D représentant la géométrie des dendrites. Nous utilisons des schémas numériques préservant la positivité des concentrations ioniques. Chapitres 1 et 2 présentent le système PNP et la méthode DDFV ainsi que ses opérateurs discrets. Le chapitre 2 présente un couplage "linéaire" des équations et explore son schéma numérique associé. Ce couplage a des problèmes de convergence, où nous illustrons ses limites à travers des résultats numériques. Le chapitre 3 introduit un couplage "non linéaire", permettant une résolution numérique précise du système PNP. Les deux couplages sont effectués avec la méthode DDFV. Dans le chapitre 3, nous illustrons la convergence d'ordre 2 en espace. Nous simulons un cas test impliquant la BL. Nous appliquons le schéma DDFV à la géométrie des épines dendritiques en 2D et discutons nos simulations en les comparant avec des simulations en 1D de la littérature. Nous introduisons également deux configurations originales de dendrites, fournissant des informations sur la manière dont les épines dendritiques s'influencent mutuellement, révélant l'étendue de leur influence mutuelle. Nos simulations montrent la distance de propagation de l'influx ionique lors des connexions synaptiques. Dans le chapitre 4, nous résolvons le système PNP sur un système multi-domaines 2D composé d'une membrane, d'un milieu interne et d'un milieu externe. Cette approche permet la modélisation de la dynamique du voltage de manière plus réaliste, et aide également à vérifier la cohérence des résultats du chapitre 3. Nous utilisons le logiciel FreeFem++ pour résoudre le système PNP dans ce contexte. Nous présentons des simulations correspondant aux résultats du chapitre 3, démontrant la sommation linéaire dans une bifurcation dendritique. Nous étudions la sommation des signaux en ajoutant des entrées à la membrane d'une branche dendritique. Nous identifions un seuil d'excitabilité où la dynamique du voltage est significativement influencée par le nombre d'entrées. Nous offrons également des illustrations numériques de la BL à l'intérieur du milieu intracellulaire, observant de petites fluctuations. Ces résultats sont préliminaires, visant à fournir des informations pour comprendre la dynamique dendritique. Le chapitre 5 présente un travail collaboratif mené lors du Cemracs 2022. Nous nous concentrons sur un schéma de volumes finis composite où nous visons à dériver les équations d'Euler avec des termes sources sur des maillages non structurés.

Mots-clés. Poisson-Nernst Planck équations, Discrete Duality Finite Volumes, transport d'ions, FreeFem++.

Contents

Introduction

In this thesis, we use the so-called Discrete Duality Finite Volume (DDFV) method and the FreeFEM++ software to investigate the voltage propagation and ionic dynamics in neurons. This introduction is dedicated to introducing the biological context, its motivations, along with a comprehensive description of each subsequent chapter.

Short description of neurons

Neurons, the fundamental units of the nervous system, play a crucial role in information processing. A neuron, consisting of a soma, dendrites, an axon, and synapses (see Figure 1 below), orchestrates complex signaling within the nervous system. To be more precise, neurons operate through an excitation process wherein a neuron becomes activated and produces an electrical signal called an action potential. This excitation occurs when the neuron receives signals, often in the form of chemical or electrical messages, from other neurons.

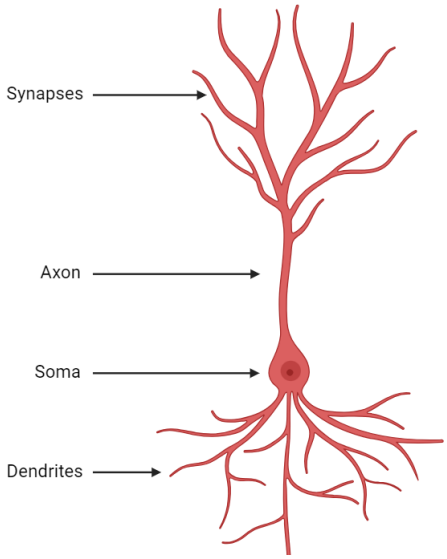

FIGURE 1. Example of a neuron geometry using Biorender https://www.biorender.com/

For the sake of scientific curiosity and clarity, let us dive into the neuron's geometry: first, the **soma**, also known as the cell body or perikaryon, serves as the core center for integrating synaptic inputs. Structurally characterized by a spherical or oval shape, it houses the cell nucleus and essential organelles responsible for protein synthesis. The soma plays a central role in influencing the neuron's excitability and responsiveness to incoming electrochemical signals. Notably, the soma operates at a macroscopic scale, typically ranging from 5 to 100 micrometers in diameter.

As far as **dendrites** are concerned, these branched extensions in a neuron that emanate from the soma, are primary sites for receiving synaptic inputs. On the dendrites, the post-synaptic terminal receiving the neurotransmitter is often a specific neuronal compartment called a **dendritic spine**. This mushroom-like protrusion is characterized by a thin cylindrical neck surrounded by a bulky head.

It serves to compartmentalize and enhance synaptic communication, playing a crucial role in synaptic plasticity and neural connectivity. While dendrites receive signals, dendritic spines facilitate the transmission and processing of those signals at specific synaptic sites. The spatial and temporal summation of signals on dendrites determines whether the neuron generates an action potential, contributing significantly to overall excitability. It is to mention that dendrites operate at a microscopic scale, extending over distances of hundreds of micrometers to several millimeters.

The **axon**, also known as a nerve fiber, facilitates the unidirectional transmission of electrical signals between the soma and synaptic terminals. One important feature of the axon is its ability to enhance nerve impulse propagation, with myelinated axons exhibiting increased conduction speed due to saltatory conduction, while non-myelinated axons also contribute to efficient signal transmission through other mechanisms. In non-myelinated axons, local regeneration of the nerve impulse is facilitated by ion channels distributed along the axon, ensuring continuous and effective signal transmission. Axon terminals release neurotransmitters at synapses, crucial for transmitting information to other neurons or effector cells. Further, the axon acts at varying scales, ranging from a few millimeters to over a meter for humans, depending on the neuron type and anatomical location.

When it comes to **synapses**, these are specialized junctions critical for transmitting information within the nervous system. They facilitate neurotransmitter release and reception. Dynamic and flexible, synapses exhibit plasticity necessary for learning and memory. Besides, at a microscopic scale, the size of the synaptic cleft, the minuscule gap between the presynaptic and postsynaptic membranes, is measured in nanometers. This dimensional specificity ensures precise and rapid transmission of information between neurons.

Focusing on the creation of an action potential, pictured in Figure 2 below, if the incoming signals (or stimulus) are strong enough and reach a certain threshold, they trigger the neuron to fire an action potential. This phenomena implies rapid changes in membrane potential through the opening and closing of voltage-gated ion channels. Recall that, this electrical signal allows neurons to communicate with each other and is fundamental to various functions in the nervous system, including transmitting information and coordinating actions in the body.

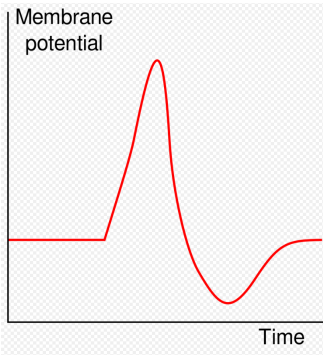

FIGURE 2. Formation of an action potential in a neuron.

Classical model for voltage propagation

Lord Kelvin, in the 19^{th} century, laid the foundation for mathematical models of electrical conduction in neurons. Kelvin introduced the theory of the cable, presenting an analytical framework for understanding the transmission of electrical signals along biological structures. The cable theory, proposed in the 1850s, conceptualized the neuron as a cylindrical cable with electrical resistance and capacitance. Formally, the model described how electrical signals attenuate and propagate along the length of the

neuron, taking into account factors such as membrane resistance and capacitance [58, 62]. The basic equation derived from the cable theory is:

$$\frac{1}{R_l}\frac{\partial^2 V}{\partial x^2} = C_m \frac{\partial V}{\partial t} + \frac{1}{R_m} V. \tag{0.1}$$

Here, R_l represents the longitudinal intracellular resistance, R_m is the membrane resistance, C_m is the membrane capacitance, V is the membrane potential, t is time, and x is the position along the cable.

In the mid-20$^{\text{th}}$ century, the pioneering work of Alan Hodgkin and Andrew Huxley [55] marked a significant advancement in understanding the neuronal electrical dynamics. In 1963, Hodgkin and Huxley were awarded the Nobel Prize in Physiology or Medicine for their development of the Hodgkin-Huxley (HH) model, which was experimentally studied using the squid giant axon (see Figure 3).

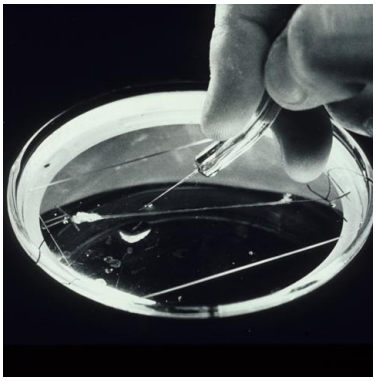

FIGURE 3. The giant axon from a squid was utilized by experimentalists to quantify the voltage drop along the length of the axon (source: *NIH photographers Website*).

This model [55] provided a comprehensive description of the generation and propagation of action potentials in neurons. It consists of a set of differential equations that describe the dynamics of ion channels, membrane potential, and conductances. The above equation (0.1) derived from the cable theory involves an additional key equation, and rewrites as:

$$\frac{1}{R_l}\frac{\partial^2 V}{\partial x^2} = C_m \frac{\partial V}{\partial t} + I_{\text{ion}}, \quad \text{with} \tag{0.2}$$

$$I_{\text{ion}} = \bar{g}_{\text{Na}} m^3 h (V - E_{\text{Na}}) + \bar{g}_{\text{K}} n^4 (V - E_{\text{K}}) + \bar{g}_{\text{L}} (V - E_{\text{L}}).$$

In the above expressions, C_m is the membrane capacitance, V is the membrane potential, I_{ion} is the total ion current, $\bar{g}_{\text{Na}}$ and $\bar{g}_{\text{K}}$ are the maximum sodium and potassium conductances, m, h, and n are gating variables, and E_{Na}, E_{K}, and E_{L} are respectively the equilibrium potentials for sodium, potassium, and leak channels.

The HH model's experimental validation on the squid giant axon demonstrated its accuracy in predicting action potential behavior. Subsequent models, such as compartmental models, have been developed to account for both the variability of channels distribution and the tree-like geometry of dendritic arbor. These models provide more accurate representations of electrical behaviors in complex neuronal structures like dendrites. Despite their success at the macroscopic scale, these models based on (0.2) have limitations when applied to finer neuronal structures like dendrites. Indeed, it does not consider the dynamics of ionic concentrations within the neuronal cytoplasm, which influence is expected to be significant within small neuronal compartments such as dendrites.

In parallel to the development of the model (0.2), another crucial mathematical framework emerged in the form of the Poisson-Nernst-Planck (PNP) system [61]. The PNP system, introduced in the mid-20$^{\text{th}}$ century, represents a set of partial differential equations (PDEs) that describe the dynamics

of ions in an electrolyte solution, accounting for diffusion and electrostatic interactions. Initially formulated for studying ion transport in electrolytes, the PNP system found applications in neuroscience, particularly in modeling ion dynamics within dendrites. The Nernst Planck (NP) equations establish a connection between the distribution of ionic concentrations c_i for some species i, and the electric potential V through the ion conservation equation, by means of an associated drift-diffusion flux $\mathbf{J}_{c_i}$. $\mathbf{J}_{c_i}$ is further decomposed into a diffusion term, $\mathbf{J}_{\text{diff}}$, and a convective term, $\mathbf{J}_{\text{conv}}$. $\mathbf{J}_{\text{diff}}$ represents the Brownian motion of ions according to Fick's law:

$$\mathbf{J}_{\text{diff}} = -D_i \nabla c_i,$$

where D_i is the diffusion coefficient for species i. $\mathbf{J}_{\text{conv}}$ accounts for ion transport induced by the electric field:

$$\mathbf{J}_{\text{conv}} = -\frac{F D_i z_i}{R T_\theta} c_i \nabla V.$$

where F is the Faraday constant, z_i the respective valences for species i, R the gas constant and T_θ the absolute temperature.

This set of equations is then combined with the Poisson equation governing voltage dynamics, thereby forming the comprehensive PNP system of equations:

$$\partial_t c_i = -\nabla \cdot \mathbf{J}_{c_i} + f_{c_i},$$

$$-\nabla \cdot (\varepsilon \varepsilon_0 \nabla V) = \sum_i F z_i c_i + f_V,$$

where ε_0 is the vacuum permittivity, ε the relative permittivity of the solution, f_{c_i} and f_V are the respective source terms. The PNP system is therefore characterized by the coupling of parabolic and elliptic equations. This system gained prominence for its ability to capture the intricate dynamics of ion movement within complex geometries, such as dendritic structures. Dendrites exhibit a high degree of spatial and chemical heterogeneity, making the task of accurate modeling challenging. The PNP system, with its adaptability in microscopic scale, proved suitable for describing the transport of ions within the fine and intricate branches of dendrites, especially the effects of local ionic concentrations and variations on ion mobility. This level of detail is crucial when considering the small dendritic compartments such as dendritic spines, where ionic dynamics play a significant role in signal processing and integration. In summary, the PNP system, with its rich history rooted in electrochemistry, has found a remarkable application in neuroscience, especially in the modeling of ionic dynamics in dendrites. Its flexibility and ability to incorporate detailed geometrical and chemical information make it a valuable tool for unraveling the complexities of electrical signaling in the intricate structures of neuronal dendrites.

Furthermore, given the ever-growing significance of technological progress [87, 32], the use of the PNP system appears indispensable to study the finer aspects of neuronal structure. Indeed, advanced imaging technologies, such as two-photon microscopy (for instance, see Figure 4 below, from [63]), have enhanced our ability to observe ionic and voltage dynamics at the level of dendrites and individual spines.

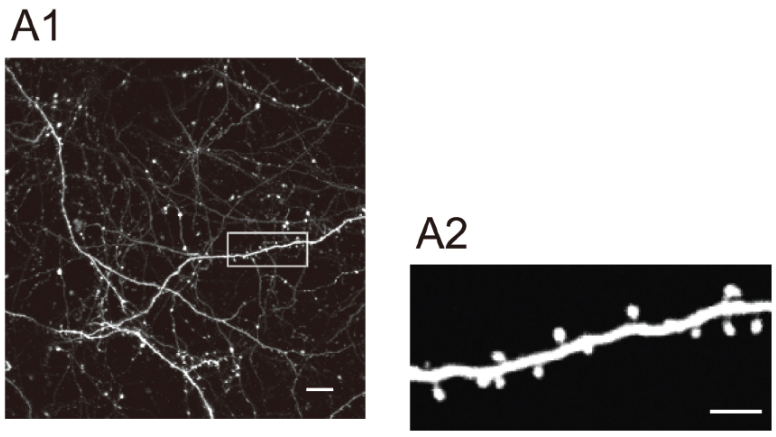

FIGURE 4. A1: Typical morphologies of dendrites in the superficial layer of the neocortex detected by *in vivo* two-photon microscopy. Scale bar, 10 µm . A2: Higher magnification view of the dendritic segment marked by the rectangle in A1. Extracted from [63].

In addition, an examination of dendritic activity reveals that neuronal communication occurs at synapses. In this process, neurotransmitters released from a presynaptic neuron bind to specific receptors on dendrites, thereby influencing the membrane potential of the postsynaptic cell. Therefore, it is imperative to investigate the behavior of dendrites and dendritic spines once again to enhance our understanding of their influence on information propagation.

In the above context of neurons, the main interest of my thesis is studying dendrites and dendritic spines. Indeed, dendritic spines are of key importance in influencing signal integration, synaptic strength, and plasticity (see Figure 5). The investigation of these spines contributes to a deeper understanding of how neuronal communication and information processing occur at the microscopic level within the intricate architecture of dendritic structures.

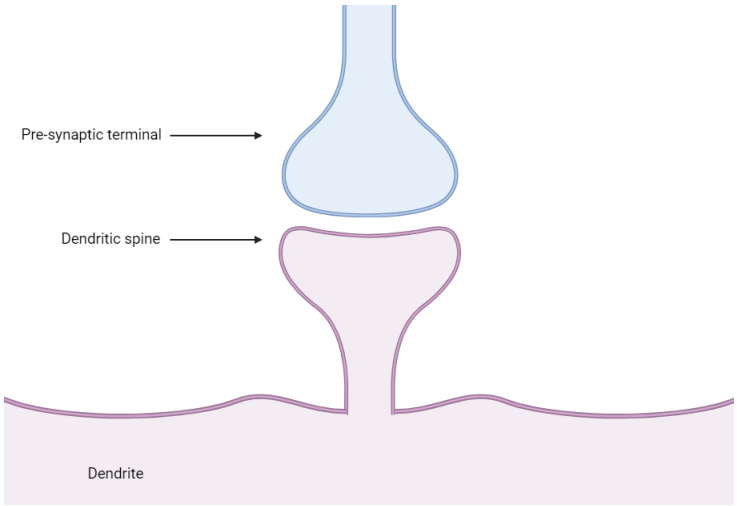

FIGURE 5. Example of a dendritic spine with synapse using *Biorender*.

Academic and scientific research concerning dendrites and dendritic spines, continue to raise numerous intriguing questions. Among the numerous open questions on this subject, my thesis addresses the following inquiries:

- **Synaptic integration in dendrites:** How are signals integrated from multiple influxes situated on dendrites ? In other words, how does the summation of the signal occur when it originates from multiple dendritic branches?

- **Signal propagation in dendrites:** How do electrical signals propagate along dendrites, especially through their complex ramifications and different dendritic zones? What is the impact of dendritic morphology on signal propagation?

- **Communication between dendrites:** How are signals exchanged between different dendrites of the same neuron? What is the impact of this dendritic communication on neuronal function?

Additional open questions that represent an engaging avenue for future research in neuroscience exist, and are beyond the scope of this thesis:

- **Role of dendritic spines:** What is the functional diversity of different forms of dendritic spines? How do these structures influence synaptic transmission and neuronal plasticity?

- **Synaptic plasticity at the dendritic level:** What role do dendritic spines play in synaptic plasticity?

- **Calcium dynamics in dendrites:** How do fluctuations in calcium levels within dendrites influence synaptic regulation and plasticity? What are the underlying mechanisms of calcium dynamics in these cellular compartments?

- **Advanced experimental measures:** How can experimental techniques be improved to precisely study activities at the level of dendrites and dendritic spines? What new technologies are necessary for better visualization and manipulation of these structures at the nanoscale?

These questions illustrate the complexity and diversity of challenges yet to be addressed in understanding the functioning of neurons, particularly regarding dendrites and dendritic spines. Answers to these questions can not only expand our fundamental knowledge of the brain but also have implications for the treatment of neurological diseases and the development of advanced neurological technologies.

In contemporary neuroscience, a synergistic approach combining experimental biology, computational modeling, and robust numerical methods is imperative. This integrative methodology ensures a comprehensive understanding of neuronal function, providing insights into complex phenomena such as synaptic plasticity and network dynamics.

Motivated by these questions, throughout this thesis, we aimed at numerically providing partial answers or validation regarding the behaviors of signals within dendrites. We focused on modeling synaptic inputs, particularly within small compartments such as dendrites and dendritic spines. Indeed, we model the nerve influx originating from the synapse directly on the membrane of the dendrite. To be precise and account for their very small scale, we chose to solve the PNP system using the DDFV method on a single two-dimensional domain. Subsequently, an exploratory study is proposed, considering the PNP equations on a multi-domain representing dendritic branches, using FreeFEM++.

In doing so, we model the dynamics of ions and voltage within the dendrite during synaptic excitation. In chapters 3 and 4, we endeavor to understand how dendritic spines influence each other during excitation and how electrical signals propagate in dendritic branches.

Review of some existing PNP modeling and applications

As mentionned before, the PNP system of equations is a well-established framework for describing the movement of ions and the variations of the electric field in an electrolyte [61]. It is a topic of substantial interest which is widely used in various fields such as semiconductor devices [59], electrophoresis models [60], or biological systems [5, 37]. Solving the PNP system analytically and numerically is challenging due to the nonlinear coupling between the electrostatic potential and the concentrations of ionic species. Also, the system can present, depending on the domain and boundary conditions, a

stiff boundary layer. A standard example is a binary electrolyte between parallel-plate electrodes, a system showing a boundary layer near the electrodes, characterized by significant ionic concentration gradients [5].

Accurate numerical schemes for the PNP equations have garnered significant attention in the recent years. Several numerical methods have been developed using finite element methods [30, 70, 78] and finite volume methods [20, 74, 72, 88].
Other studies specifically focused on preserving the ionic concentration positivity: in 1D, [40] presented a mass-conservative finite difference scheme, while [39] proposed a free energy dissipative finite difference scheme. These approaches required the use of uniform cells, and did not allow for local mesh refinement. For the 2D PNP equations, a mass-conservative finite difference scheme was introduced by [48]. Building upon this work, [49] developed a semi-implicit linearized finite difference scheme with proof of positivity and free energy dissipation. Positivity-preserving finite difference schemes with mass conservation, steady state, energy dissipation, and proof of convergence were presented by [57, 66] and [67]. These methods did not focus on capturing the behavior at the boundary layer. Finally, in [82], the authors presented a numerical scheme integrating both the positivity of ionic concentrations and a mesh refinement at the boundary layer. The method is based on a combination of finite volumes and finite elements methods.

In neuroscience, it is rather recent that models using the PNP system of equations are developed. They follow the latest innovations in experimental techniques now reaching the nano-scale, which create the need for modeling at the same precision [77, 56]. Indeed, as mentioned earlier, the classical model describing voltage propagation in neurons is the cable theory, the one-dimensional macroscopic model that makes an analogy between the neuron and a wire, assuming voltage propagates as in a RC electrical circuit [62]. In particular, this model neglects the variations of ionic concentrations within the neuronal cytoplasm. Such assumption is valid while considering large axons, but is not relevant in smaller neuronal structures [77], where we expect the influence of the ionic dynamics on the electric field to be important. In contrast to this one-dimensional approach, the PNP system of equations describes the dynamics of ionic charges due to both diffusion and electric field, and is highly suitable to capture such dynamics in small neuronal compartments.
Various computational models have been developed and simulated to determine ionic and voltage dynamics in neurons. In [69], a three-dimensional computational model of the PNP equations, combined with a Hodgkin-Huxley formalism [55] for reproducing ionic channels dynamics was introduced. This model was designed to simulate the electrodiffusion of ions within a neuronal node of Ranvier during an action potential. The numerical study focused on an axon model with a diameter of 15µm , utilizing classical finite volume techniques with Delaunay-Voronoi dual meshes. The authors compared their simulations with the one-dimensional cable equations, and their results demonstrated remarkable similarities between the two models. [76] extended this exploration by focusing on studying the boundary layer (BL) formed near the membrane of neuronal axons. This two-dimensional investigation explicitly addressed BL phenomena using finite element methods, with the Hodgkin-Huxley model applied as a Neumann boundary condition on the membrane. These works laid the foundation for understanding BL in axons but did not address the case of smaller neuronal compartments such as dendrites or dendritic spines. Due to their microscopic scale, experimental investigations of voltage and ionic dynamics in spines remain challenging, making it crucial to develop models at the same level of precision [56].

The main objective of this thesis is to investigate the Poisson-Nernst Planck (PNP) system of equations, particularly in the context of the aforementioned challenges encountered in neuroscience. Among these, we pay particular attention to synaptic integration and signal propagation within dendrites, as well as communication between dendrites. In this thesis, my specific areas of interest focused on:

- The coupling of the PNP system using the Discrete Duality Finite Volume method (DDFV).
- The coupling of the PNP system over a multi-domain employing the FreeFEM++ software.

Note that the use of FreeFEM++ is motivated by its suitability for multi-domain exploration within a more lightweight framework compared to DDFV. Also, the need of modeling not only the interior of the domain, but the entire domain at hand, allows for a more natural model avoiding the imposition of a boundary condition on the potential at the membrane. Indeed, such a boundary condition can influence the dynamics and deviate from the expected physical behavior. In this regard, the FreeFEM++ code is used for comparison with results obtained in Chapter 3 based on the DDFV method.

The thesis begins with a concise overview of the PNP system of equations and delves into its relevance in current neuroscience research, particularly in the context of dendrite arborization. Subsequently, we introduce the DDFV method, outlining the reasons behind our choice of this particular approach. Following this, we present two numerical schemes designed for solving the PNP system of equations using the DDFV method over a two dimensional domain. We apply our implementation to numerical and neuroscience-derived cases, providing detailed results and physiological insights in dendrites. In this approach, it is noteworthy that our model imposes a homogeneous Neumann boundary condition on the potential at the membrane wall. Finally, we employ the FreeFEM++ software to solve the PNP system of equations on a multi-domain representing samples of dendrite. This latter approach allows for imposing a continuity condition at the membrane interfaces for the potential instead of the homogeneous Neumann boundary condition, which is a much more natural choice for the dynamics of the physical phenomenon. Ultimately, both approaches enlighten our comprehension of the inherent complexities within neuronal structures.

PNP coupling strategies with DDFV method

In view of the aforementioned neuroscientific issues, the modeling of dendrites and their associated spines requires a comprehensive study of the PNP system. Indeed, we recall that the latter system governs the dynamics occurring within these neuronal components. In this regard, during the thesis, we begin by developing finite volume approaches in order to solve the PNP system of equations, using the DDFV framework, this while preserving the positivity of ionic concentrations. The DDFV method is a finite volume method that ensures local conservativity and the divergence theorem. The flux exiting one cell is equal to the flux entering the neighboring cell. In our study, we specifically focus on the 2D case. Therefore, the DDFV method is particularly attractive for numerically solving the PNP equations over a complex geometry, because of its robustness with respect to mesh distortion. This finite volume method, introduced by [**53**, **35**] in the early 2000s, enables the study of the Laplace equation on various 2D meshes, accommodating nonconformal and deformed meshes. Indeed, DDFV schemes consider discrete unknowns at both the vertices and the centers of initial mesh. In addition, the DDFV method operates on three staggered meshes: the "primal mesh" (denoted with $\mathfrak{M} \cup \partial\mathfrak{M}$), the "dual mesh" (denoted with $\mathfrak{M}^* \cup \partial\mathfrak{M}^*$), centered on the vertices of the primal mesh, and the "diamond mesh" (denoted with $\mathfrak{D}$), which is centered on the edges of the primal mesh. We refer to the illustration in Fig. 6 for clarification. The union of primal and dual mesh will be denoted as $\mathcal{T}$.

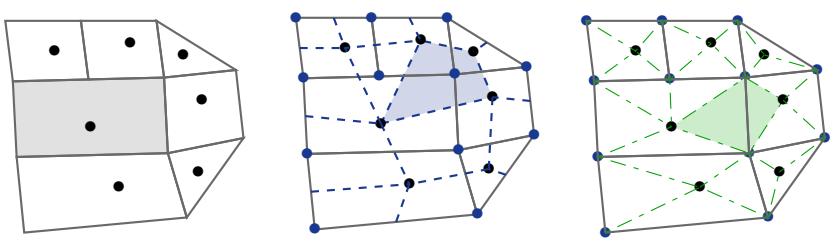

FIGURE 6. DDFV meshes: the primal mesh $\overline{\mathfrak{M}} = \mathfrak{M} \cup \partial\mathfrak{M}$ (gray), the dual mesh $\overline{\mathfrak{M}^*} = \mathfrak{M}^* \cup \partial\mathfrak{M}^*$ (blue) and the diamond mesh $\mathfrak{D}$ (green).

The set of unknowns is then defined within $\mathbb{R}^{\mathcal{T}}$ the linear space of scalar fields that are constant over the cells of $\overline{\mathfrak{M}}$ and $\overline{\mathfrak{M}^*}$, such as:

$$u^{\mathcal{T}} \in \mathbb{R}^{\mathcal{T}} \Longleftrightarrow u^{\mathcal{T}} = \left((u^K)_{K \in \overline{\mathfrak{M}}}, (u^{K^*})_{K^* \in \overline{\mathfrak{M}^*}} \right).$$

It allows for the definition of a full discrete gradient on the diamond mesh. Hence, the method can be used on general meshes that do not necessarily satisfy classical orthogonality conditions, as required by the two-point scheme (known as the classical finite volume method [84]). The DDFV methods construct two discrete operators, the gradient and the divergence, that are in duality in a discrete sense as in the continuous setting [3]. Several studies have demonstrated the numerical advantages of DDFV methods for PDEs, and several methods have been developed for a large number of models. A telling example concerns a DDFV scheme discretizing the Peaceman model - that describes miscible displacement in porous media - on which a convergence result is proved [27]. Numerical analysis of systems similar to PNP in the field of semiconductors was carried out by the authors [9]. Then, a positivity-preserving scheme was introduced in [18], to investigate the existence and large time behavior of solutions in Fokker-Planck equations. Positivity was ensured by using a non-linear reformulation of the equation.

In Chapters 2 and 3 of the present thesis, we propose two numerical schemes, one for the linear and another for the non-linear coupling of the PNP system, inspired by the study within [18].

First, in Chapter 2, we consider the case of one ionic species P with its respective valence $+1$ and concentration c_P. The PNP system consists of a Nernst-Planck equation for $c_P \in L^2(\Omega \times [0, T_f])$ with a Poisson equation for $V \in L^2(\Omega \times [0, T_f])$ the voltage. The latter is then defined over $\Omega \times [O, T_f]$, a connected open bounded domain $\Omega \subset \mathbb{R}^2$ and $T_f > 0$ a finite time:

$$\partial_t c_P = \nabla \cdot (c_P \nabla (\log c_P + V)) + f_{c_P},$$

$$-\nabla \cdot (\nabla V) = c_P,$$

where $f_{c_P} \in L^2(\Omega \times [0, T_f])$ is a non-zero source term.
We provide a time-splitting strategy to couple the equations of the system and suggest a spatial discretization using DDFV. It implies solving the Poisson equation with a concentration c_P at the previous time step, and then, using the obtained potential, calculating a new concentration as in [18]. Also, the nonlinearity of the system is handled by a Newton algorithm. This coupling strategy is called as linear coupling and does not lead to accurate numerical results, unfortunately. We give results of test cases for this coupling in the form of convergence tables. Additionally, we present a numerical study of our implementation, including comparaisons of the Jacobian matrices generated by our code with those from [18]. In conclusion, we discern that the coupling of the nonlinear PNP system using a linear strategy does not yield precise approximations. Hence, it becomes imperative to introduce a new strategy, to achieve more accurate outcomes.

Afterwards, in Chapter 3, we focus our study on the case of two ionic species P and N, with respective valences $+1$ and -1, and concentration $c_P \in L^2(\Omega \times [0, T_f])$ and $c_N \in L^2(\Omega \times [0, T_f])$. The PNP system is then defined over $\Omega \times [O, T_f]$:

$$\partial_t c_P = \nabla \cdot (D_P c_P \nabla (\log c_P + \beta V)) + f_{c_P},$$

$$\partial_t c_N = \nabla \cdot (D_N c_N \nabla (\log c_N - \beta V)) + f_{c_N},$$

$$-\nabla \cdot (\gamma \nabla V) = c_P - c_N + f_V.$$

where $V \in L^2(\Omega \times [0, T_f])$ is the voltage. $D_P \in \mathbb{R}^{+*}$ (resp. $D_N \in \mathbb{R}^{+*}$) is the diffusion coefficient of species P (resp. N). $f_{c_P} \in L^2(\Omega \times [0, T_f])$, $f_{c_N} \in L^2(\Omega \times [0, T_f])$ and $f_V \in L^2(\Omega \times [0, T_f])$ are the respective source terms of c_P, c_N and V. The coefficients $\gamma \in \mathbb{R}^{+*}$ and $\beta \in \mathbb{R}^{+*}$ are defined in Chapter 3.

Within this context, we shift to a new strategy by introducing a non-linear coupling that allows solving the system of equations with DDFV in spatial discretization and an implicit Euler scheme in time. Indeed, the set of unknowns in this strategy is now (c_P, c_N, V), where the idea is to solve all three equations of the system at each time step. In this case, the Jacobian matrix is computed from the entire system. The Newton algorithm is thus generated for the entire nonlinear system at each time step. In order to check consistency of our method, based on simulations, we present numerical results in the form of convergence tables and compare them with existing literature results for similar numerical methods using finite elements [66]. As expected, we numerically obtain a convergence order equal to 2, confirming the method's performance on the PNP system.

Furthermore, we assess the accuracy of our numerical results by comparing them to a test case enabling the visualization of the BL from [81] (see Figure 7). In the latter, we extend the 1D results derived in [81] to a 2D framework by introducing a new modeling of boundary conditions.

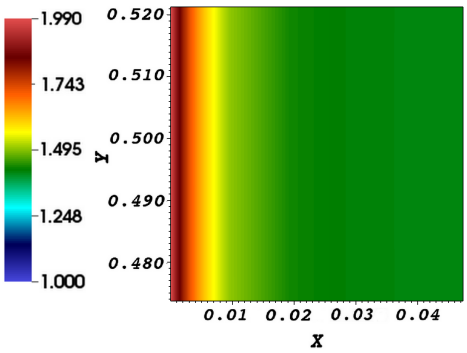

FIGURE 7. Simulation from the DDFV scheme on the test case allowing the visualization of the BL, in a neighborhood of $x = 0$ and at fixed time. The discrete values of c_P show a steep gradient inside the BL which has a thickness of 0.01.

In view of our neurobiological interests addressed earlier, we model real applications in neuroscience, using our DDFV scheme to provide biological insights into dendritic tree behavior. We begin our exploration by examining the propagation of ion influx in a dendritic bifurcation, as seen in Figure 8. To this end, various sizes of dendrites are studied, where we gradually increase the chosen distance from the bifurcation point to what is called an ionic reservoir, i.e. the connexion with a larger branch such as the dendritic shaft, where we expect the ionic concentration to remain constant (corresponding to the boundary at $x = -22$ in Figure 8). In the case of Figure 8, this distance represents the largest configuration.

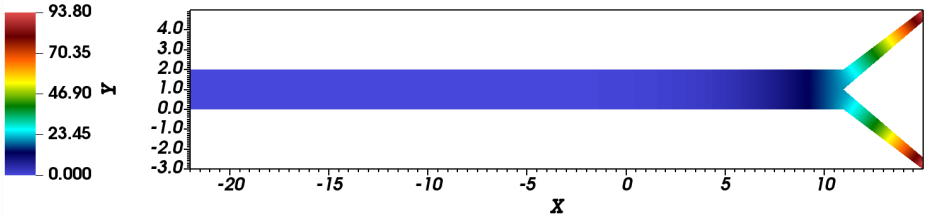

FIGURE 8. Simulation from the DDFV scheme on the dendrite bifurcation. We show the discrete values of c_P at fixed time. We model the ion influx from both thin branches at $x = 15$.

Comparing different distance configurations as mentioned above, we are able to establish a distance of approximately 20 microns from the bifurcation in order to ensure minimal influence of an ionic reservoir. An additional aspect is to examine the propagation of ions from both branches to the rest of the bifurcation. Indeed, we compare the signals when:

- only one branch is "active" (synaptic input at the end of the branch),
- both branches are "active" (synaptic input at the end of both branches).

Therefore, we demonstrate that two signals simultaneously reaching the bifurcation exhibit linear summation.
In further simulations, still within Chapter 3, we shift our focus to dendritic spines, investigating how their distinct geometry affects voltage and ionic concentration dynamics. While a 1D approximation suffices for the neck region, a more intricate 2D approach is imperative for understanding the dynamics within the head, as depicted in Figure 9.

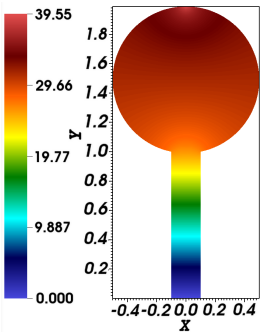

FIGURE 9. Simulation from the DDFV scheme on the dendritic spine. We show the discrete values of c_P at fixed time. We model the ion influx at the top of the head.

Subsequently, we delve into the mutual influence of two neighboring spines, especially when an influx of ions is initiated in one spine. Similarly to the dendrite bifurcation, we vary the lengths of each ionic reservoir (corresponding to the boundaries at $x = -6$ and $x = 8$ in the example of Figure 10) from either side of the spines and also, the distances between both spines.

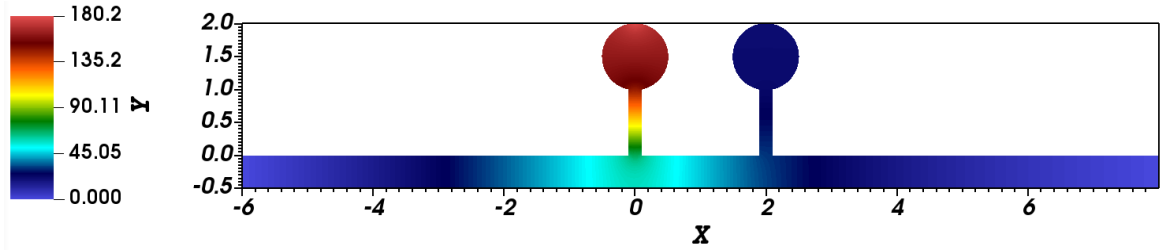

FIGURE 10. Simulation from the DDFV scheme on the two neighboring dendritic spines. We show the discrete values of c_P at fixed time. We model the ion influx at the top of the head of the left dendritic spine.

Our simulations show that the close proximity of an ionic reservoir such as the dendritic shaft or any large compartment, is killing the signal, preventing it from propagating and invading close neuronal structures such as thin branches at a bifurcation or dendritic spines. On the opposite, a signal arriving at the leading edge of the dendritic tree, far from an ionic reservoir and where only small branches are present, will propagate at a larger distance and invade neighboring dendritic spines.
From the spine point of view, these results show that the same spine at different positions in the tree would behave differently: a spine close to the dendritic shaft would more likely act as an autonomous compartment, compared to a collection of spines located in small dendritic protrusions, that would be keener to signal invasion and to influence each other. Hence, in addition to the geometry of the spine, the local geometry of the dendritic tree is shaping spine behavior, in that the same spine at different positions in the tree would behave differently. Consequently, the position of a dendritic spine relative to the entire tree (i.e. close to the soma versus at the distal edge of the tree) shapes its function, making plasticity not at the level of the spine, but at the level of the full dendritic geometry.

Chapter 3 is under review at the *Journal of Computational Physics*. A preprint version is available at https://cnrs.hal.science/hal-04385924

PNP coupling over a multi-domain using FreeFEM++

In this chapter, we once again examine the ionic and voltage dynamics of dendrites. However, we modify our model in several aspects. Indeed, we transition from a single 2D domain to a 2D multi-domain containing the intracellular medium, the extracellular medium and the membrane. The interest is that we can model continuity conditions at the membrane interfaces for the potential, instead of the homogeneous Neumann boundary condition we imposed on the single-domain framework. We believe that this latter approach provides a more realistic study of ion dynamics and voltage, acknowledging that the propagation of electrical signals also occurs within the membrane. In essence, electricity is ubiquitously present around the whole dendrite. We implement this new approach using FreeFEM++.

FreeFEM++ is an open-source software [**51, 50**] for solving PDEs using finite element methods. It has emerged as a powerful tool in the realm of scientific computing. Its development can be traced back to the late 1990s, with continuous enhancements and a growing user community making it a prominent player in numerical simulation. FreeFEM++ employs the finite element method, a powerful numerical technique for approximating solutions to PDEs [**45, 43**]. This method involves discretizing the domain of interest into simple geometric shapes (elements) and constructing a piecewise polynomial solution over these elements. FreeFEM++ excels in handling problems with complex geometries and diverse physics, providing a versatile platform for researchers and engineers.

Also, FreeFEM++ supports the modeling of multi-domain problems [**73**], allowing researchers to study ion dynamics in different regions and their interactions. Thus, this is one opportunity to solve in a multi-domain framework the PNP system modelling the dendrite dynamics. As mentioned earlier, the multi-domain consists of three disjoint sets: the cytosol (the liquid part of the cytoplasm), the membrane and the external space of course. The idea of modeling a multi-domain allows us to avoid the homogeneous Neumann condition on the potential imposed when considering a single domain in the modeling of electrical potential boundary conditions. As the membrane acts as an electrical insulator, we therefore incorporate parameters such as the electric permittivity coefficients present in each medium (cytosol, membrane, external medium). These coefficients play a role in the Poisson's equation and vary according to each sub-domain. The equation is thus solved over the entire domain (the electric potential "exists" everywhere), with only the permittivity coefficients dependent on the different domains.For the NP equations, which are the equations for the ionic concentrations, we solve them within the intra- and extracellular domains, and we model synaptic inputs on certain parts of the membrane wall. Thanks to this new approach, we aim at observing the presence of an EBL. In the previous chapter, the boundary conditions modeled on the potential, such as homogeneous Neumann, did not allow us to numerically illustrate the EBL. In other words, we believe that, numerically, our model "kills" the presence of an EBL. In Chapter 4, we present a multi-domain model, solving the PNP system of equations on 2D dendritic branch geometries that include intracellular, extracellular, and membrane environments. Studying and solving the PNP system on a multi-domain model representing a dendritic branch is a complex task. In [**69**], the authors successfully accomplished this task applied to the axon with Ranvier nodes. To achieve this, they solved the PNP system modelling the HH formalism as membrane boundary conditions. In our study, we solve the PNP system without using the HH model. Consequently, two biological scenarios are considered to model dendrites: a branch bifurcation (Figure 11) and a large rectangular branch (Figure 12). Accordingly, we analyze the dynamics of ionic concentrations and voltage within these scenarios. In the bifurcation scenario (Figure 11), we observe linear summation of signals from two thin branches and compare these results with those presented in Chapter 3.

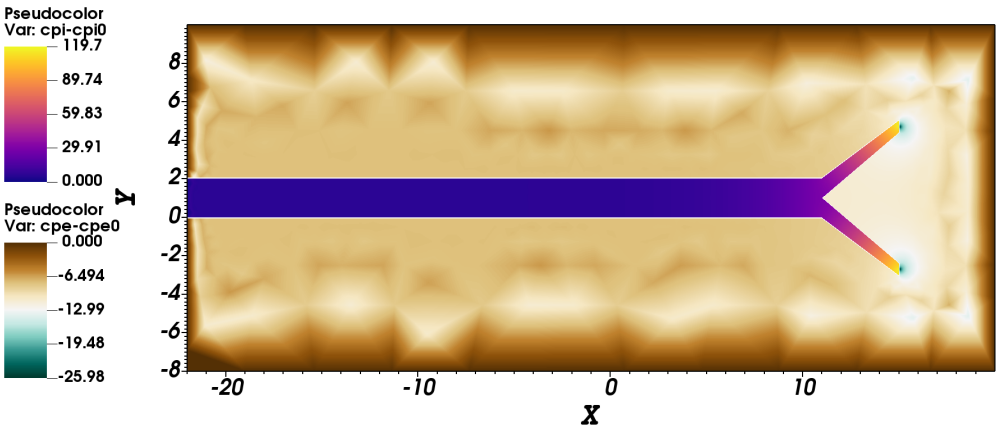

FIGURE 11. Simulation from the FreeFem++ code on the dendrite bifurcation. We show the discrete values of c_P at fixed time. We model the ion influx from both thin branches.

Subsequently, we simulate several influxes on the large rectangular branch, providing insights into the excitability threshold of a membrane potential reached through the summation of synaptic currents, when neighboring synapses generate close influxes on the membrane wall (see Figure 12).

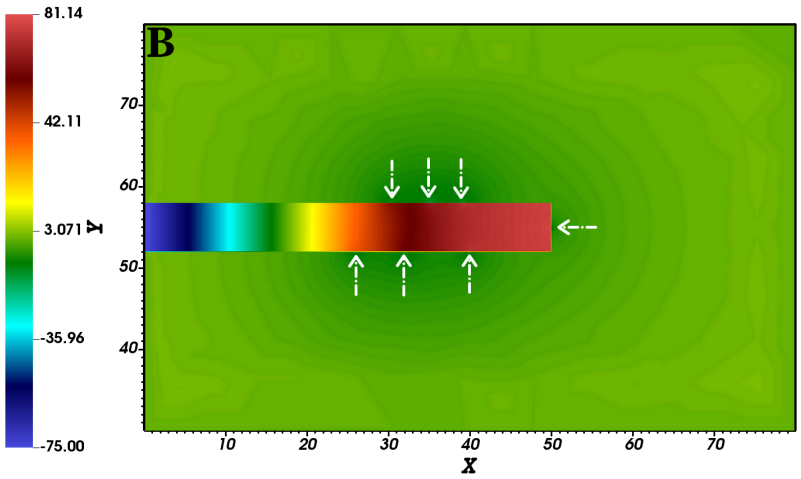

FIGURE 12. Simulation from the FreeFem++ code on the rectangular branch. We show the discrete values of V at fixed time. We model 7 ionic influxes at different positions. The dashed white arrows represent the locations where we model synaptic inputs, i.e the 7 ionic influxes.

Typically, we start with a single influx and progressively add more along the dendrite's walls. It is observed that, adding five influxes yields a significant increase in voltage, with the signal peak reaching a value similar to the voltage threshold during an action potential.

Finally, with this new modeling approach, which avoids numerically "killing off" the EBL along the wall, we are able to numerically highlight the existence of the EBL through our multi-domain modeling. To achieve this, we consider a smaller rectangular branch with smooth angles. Afterwards, we succeed at catching small fluctuations in ionic concentrations and voltage within the interior EBL. Further details can be found in Chapter 4. However, despite this approach, we were unable to capture significant fluctuations and provide results for the exterior side of the membrane. We faced challenges in FreeFEM++ regarding modeling sufficient refinement near the walls, leading us to refine only within the interior medium. One solution to this issue would be to implement the multi-domain approach using a method like DDFV, with an adapted mesh.

DDFV advantages vs FreeFEM++

In the realm of numerical methods for solving partial differential equations (PDEs), the DDFV approach has emerged as a powerful alternative, offering distinct advantages over conventional Finite Volume methods (VF4) and Finite Element methods (FEM). While both VF4 and FEM have been extensively employed, their limitations in capturing discontinuities and dealing with complex spatial variations necessitate exploration of more advanced techniques.

Here, we highlight the unique advantages that DDFV brings to the table in comparison to FreeFEM++. Note that FreeFEM++ is often regarded as a "black box" due to its ease of implementation.

Advantages of DDFV over FreeFEM++:

- Handling discontinuities in diffusion coefficients with flux continuity at the interface between domains, a feature not present in FreeFEM, makes DDFV well-suited for problems with abrupt changes.

- Handling complex mesh structures: the DDFV method demonstrates superior capabilities in dealing with complex mesh structures, ensuring robustness and reliability in simulations. This becomes particularly crucial in scenarios where FreeFEM++ might face challenges in efficiently handling intricate geometries or irregular mesh configurations.

The task of implementing the coupling of the PNP system across multiple domains using the DDFV method is time-consuming. Throughout this thesis, we successfully implemented PNP system coupling with two strategies using the DDFV method on a single domain-namely, a "linear" approach and a "nonlinear" one. However, I couldn't implement the multi-domain with DDFV in the course of this thesis, because of time limitation. Hence, I chose to realize an exploratory work on the multi-domain framework, using the more accessible FreeFEM++ software. To achieve this, I drew inspiration from the "nonlinear" approach with DDFV.

Cemracs Project: A composite finite volume scheme for the Euler equations with source term on unstructured meshes

Lastly, during this thesis, I collaborated in the annual CEMRACS event with M. Boujoudar, E. Franck, P. Hoch, C. Lasuen and Y. Le Hénaff. The project focuses on an adaptation of the method described in [54], to derive source terms in the 2D Euler equations. This adaptation extends conventional 1D solvers (e.g., VFFC, Roe, Rusanov) to the two-dimensional case, specifically on unstructured meshes. The resulting schemes are categorized as composite, as they can be expressed as a convex combination of a purely node-based scheme and a purely edge-based scheme. We combine this extension with the ideas from the work of Alouges, Ghidaglia, and Tajchman, as presented in a preprint [2], primarily focused on the 1D scenario. Our contribution involves proposing two strategies for discretizing the source term in the Euler equations, aiming to enhance the preservation of stationary solutions. To evaluate these discretization procedures, we provide a comparative analysis with the conventional centered discretization approach across various numerical examples. We propose two strategies:

- a local method utilizing sub-volumes of computational cells,

- a global method involving the solution of a Poisson equation across the entire domain.

The local approach demonstrates slightly superior performance at a comparable computational complexity. However, the global approach, while solving a Poisson equation for the entire domain, brings out significantly higher costs without yielding a substantial improvement in performance.

Chapter 5 is under review at the *ESAIM : Proceedings and Surveys*.

Preamble

Presentation of the continuous problem

Introduction. This thesis investigates the Poisson-Nernst-Planck (PNP) system, a crucial model characterizing ion movement in a static solvent via diffusion and electrostatic drift. By combining the Nernst-Planck (NP) equations with the Poisson equation, electrodiffusion and voltage dynamics are effectively captured. This system comprises parabolic and elliptic equations that govern the behavior of ionic species within a polygonal open bounded domain $\Omega \subset \mathbb{R}^2$ over a finite time $T_f > 0$. The PNP system of equations is indeed essential in studying various physical models, including neuronal dynamics.

In this thesis, we first solve numerically the PNP system using the DDFV method, and realize simulations in several geometries representing neuronal microdomains. We then investigate the ionic and voltage dynamics in these domains. In addition, we explore solving the system while preserving the positivity of ion concentrations (see Chapters 2 and 3). This requires the use of a numerical scheme with a non-linear reformulation of the flux, which raises an additional numerical challenge, and is computationally demanding. Hence, one needs to solve the latter system with an appropriate strategy. Additionally, the presence of Boundary Layers (BL) introduces numerical challenges due to their abrupt gradients. In the preamble, we introduce some notations that we will require within both chapters 2 and 3. In a second step, we delve into solving the PNP system on a multi-domain without necessary preserving the positivity of ionic concentrations, using the FreeFEM++ software (Chapter 4). This entails defining an even more complex model, taking into account three distinct environments: the intracellular media, the extracellular media, and the membrane.

The objective is to use our numerical simulations to provide insights on questions coming from neuroscience, such as signal summation in the fine parts of the dendritic tree, or the effects of the colocalization of dendritic spines on votage propagation. Simulating the PNP system in such specific dendritic geometry is a significant challenge. In the first section below, we describe the PNP system and describe the corresponding notations for initial and boundary conditions.

Poisson-Nernst-Planck equations. In the preamble, we present the PNP system of equations, which describes the movement of ions in an electrolyte. The Nernst Planck equation links the distribution of the ionic concentrations c_i of species i to the electric potential V through the ion conservation equation, using a drift-diffusion flux $\mathbf{J}_{c_i}$. $\mathbf{J}_{c_i}$ divides into a diffusion term that models the Brownian motion of ions according to Fick's law, as well as a convective term that describes the ion transport resulting from the electric field. This set of equations is then combined with the Poisson equation for voltage dynamics, creating the PNP system of equations. It is worth noting that the PNP system consists of a combination of parabolic and elliptic equations.
We consider in this paper the case of two ionic species P and N, with respective valences $+1$ and -1, and concentration c_P and c_N - our methods in Chapters 3 and 4 can be extended to any number of ions without difficulties.
Let $\Omega \subset \mathbb{R}^2$ be a connected open bounded domain and $T_f > 0$ a finite time. The PNP system of equations on $\Omega \times [0, T_f]$ writes as:

$$\begin{aligned} \partial_t c_P &= -\nabla \cdot \mathbf{J}_{c_P} + f_{c_P}, \\ \partial_t c_N &= -\nabla \cdot \mathbf{J}_{c_N} + f_{c_N}, \\ -\nabla \cdot (\varepsilon \varepsilon_0 \nabla V) &= F\,(c_P - c_N) + f_V, \end{aligned} \tag{0.3}$$

with:

$$\begin{aligned} \mathbf{J}_{c_P} &= -D_P \nabla c_P - \frac{F D_P}{R T_\theta} c_P \nabla V, \\ \mathbf{J}_{c_N} &= -D_N \nabla c_N + \frac{F D_N}{R T_\theta} c_N \nabla V. \end{aligned} \tag{0.4}$$

In system (0.3), the first (resp. second) equation represents the dynamics of species P (resp. N), with drift-diffusion flux $\mathbf{J}_{c_P}$ (resp. $\mathbf{J}_{c_N}$) and source term f_{c_P} (resp. f_{c_N}). The third equation corresponds to the Poisson equation, where ε_0 is the vacuum permittivity, ε the relative permittivity of the solution, and f_V a source term. Eqs. (0.4) describe the drift-diffusion fluxes, with $D_P > 0$ (resp. $D_N > 0$) the diffusion coefficient of species P (resp. species N), R the gas constant, T_θ the absolute temperature, and F the Faraday constant. In the rest of the paper, we will consider that the source terms f_{c_P}, f_{c_N}, f_V are in $L^2(\Omega \times T_f)$.

We define:

$$\gamma = \frac{\varepsilon \varepsilon_0 R T_\theta}{F^2}, \qquad \beta = \frac{F}{R T_\theta}, \tag{0.5}$$

such that the system rewrites:

$$\begin{aligned} \partial_t c_P &= \nabla \cdot (D_P\,(\nabla c_P + c_P \beta \nabla V)) + f_{c_P}, \\ \partial_t c_N &= \nabla \cdot (D_N\,(\nabla c_N - c_N \beta \nabla V)) + f_{c_N}, \\ -\nabla \cdot (\gamma \beta \nabla V) &= c_P - c_N + f_V. \end{aligned} \tag{0.6}$$

In Chapters 2 and 3, we use a positivity-preserving DDFV scheme as presented in [18] to ensure the positivity of ionic concentrations. The method is based on the reformulation of the fluxes $\mathbf{J}_{c_P}$ and $\mathbf{J}_{c_N}$:

$$\begin{aligned} \mathbf{J}_{c_P} &= -D_P c_P \nabla\,(\log c_P + \beta V)\,, \\ \mathbf{J}_{c_N} &= -D_N c_N \nabla\,(\log c_N - \beta V)\,. \end{aligned} \tag{0.7}$$

Finally, the system (0.3), defined on $\Omega \times [0, T_f]$, rewrites:

$$\begin{aligned} \partial_t c_P &= \nabla \cdot (D_P c_P \nabla\,(\log c_P + \beta V)) + f_{c_P}, \\ \partial_t c_N &= \nabla \cdot (D_N c_N \nabla\,(\log c_N - \beta V)) + f_{c_N}, \\ -\nabla \cdot (\gamma \beta \nabla V) &= c_P - c_N + f_V. \end{aligned} \tag{0.8}$$

Now, we turn our attention to the initial conditions and boundary values. First, we consider initial conditions of the PNP system (0.8) in Ω such that:

$$c_P(\mathbf{x}, t=0) = c_P^0(\mathbf{x}), \quad c_N(\mathbf{x}, t=0) = c_N^0(\mathbf{x}), \quad V(\mathbf{x}, t=0) = V^0(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega. \tag{0.9}$$

with c_P^0, c_N^0, $V^0 \in L^2(\Omega)$.

Then, concerning Chapters 2 and 3, we decompose the domain boundaries as $\partial\Omega = \Gamma_{Dir} \cup \Gamma_{Neu}$, where Γ_{Dir} and Γ_{Neu} represent respectively the subsets of the boundary corresponding to Dirichlet and Neumann conditions. The Dirichlet boundary conditions write as:

$$c_P = c_P^{Dir}, \quad c_N = c_N^{Dir}, \quad V = V^{Dir}, \tag{0.10}$$

with $(c_P^{Dir}, c_N^{Dir}, V^{Dir}) \in L^2(\Gamma_{Dir} \times [0, T_f])$.
Neumann boundary conditions write as:

$$\mathbf{J}_{c_P} \cdot \mathbf{n} = g, \quad \mathbf{J}_{c_N} \cdot \mathbf{n} = 0, \quad \nabla V \cdot \mathbf{n} = 0, \tag{0.11}$$

where $\mathbf{n}$ is the unit outward normal to Ω, and $g \in L^2(\Gamma_{Neu} \times [0, T_f])$.

In Chapter 4, we will delve into the system (0.6) applied over a multi-domain, i.e Ω will consist of several disjoint subsets. Thus, we will explicit the entire model with respective domain boundaries, extending the notations introduced here for the multi-domain representation.

CHAPTER 1

DDFV Method

Introduction

This chapter presents the Discrete Duality Finite Volume (DDFV) method used in Chapters 2 and 3 to couple the Poisson and the Nernst Planck equations.
The DDFV method is a finite volume method which ensures the local conservativity and the divergence theorem. The flux that comes out in one cell is the same as that enters in the neighboring cell. In our study, we specifically focus on the 2D case. The DDFV method was developed in the early 2000s, with the initial introduction and exploration of DDFV schemes found in references [35, 53], aimed at approximating the Laplace equation on a general 2D meshes, including non-conformal and distorted meshes.

The DDFV method considers unknowns at each vertex and center of the mesh and derives its name, "Discrete Duality", from the established duality between the discrete gradient $\nabla^{\mathfrak{D}}$ (see (1.3)) and the discrete divergence $\mathrm{div}^{\mathcal{T}}$ (see (1.4)). DDFV's advantage lies in its ability to discretely reconstruct and mimic the dual properties of these continuous differential operators.

The DDFV method operates on three staggered meshes: the primal mesh (denoted as $\mathfrak{M}\cup\partial\mathfrak{M}$), the dual mesh (denoted as $\mathfrak{M}^*\cup\partial\mathfrak{M}^*$) centered on the vertices of the primal mesh, and the diamond mesh (denoted as $\mathcal{D}$) centered on the edges of the primal mesh (see Fig. 1 and Fig. 2 for an illustration). The union of the primal and dual meshes is denoted as $\mathcal{T}$. This choice of unknowns and meshes facilitates the approximation of the gradient operator while preserving the structure of the continuous problem, including symmetry, which proves beneficial in handling nonlinearities. Hence, this method can readily accommodate general meshes without encountering numerical difficulties.

The chapter is organized as follows: Section 1.1 revisits the description of DDFV meshes, followed by the exploration of approximation spaces. Discrete unknowns and operators are introduced in Section 1.2, and Section 1.3 delves into the definition of discrete projections.

1.1. Meshes and notations

In this section, we present the DDFV scheme and introduce three distinct meshes using the same notations as [3]: the primal mesh, the dual mesh, and the diamond mesh (see Figure 1 and 2). The primal mesh, denoted as $\overline{\mathfrak{M}} = \mathfrak{M}\cup\partial\mathfrak{M}$, is composed of the interior primal mesh $\mathfrak{M}$ and the boundary primal mesh $\partial\mathfrak{M}$. The interior primal mesh $\mathfrak{M}$ is a partition of Ω with polygonal cells. The boundary primal mesh $\partial\mathfrak{M}$ is the set of boundary edges of $\mathfrak{M}$. Notice that these cells are treated as degenerate cells.
For each element $K \in \overline{\mathfrak{M}}$, we associate a point x_K called the center of the cell K. In practice, we choose the barycenter. The set of vertices of the primal mesh is denoted as X^*. We differentiate between interior vertices and boundary vertices: for a given $x_{K^*} \in X^*$ such that x_{K^*} is not on the boundary $\partial\Omega$, we define a dual cell K^* by connecting the centers of all K that share x_{K^*} as a vertex. This set of polygons K^* forms the interior dual mesh $\mathfrak{M}^*$.
For each $x_{K^*} \in X^*\cap\partial\Omega$, i.e. x_{K^*} is on the boundary, we construct a dual cell K^* by joining x_{K^*} and the x_K where $K \in \overline{\mathfrak{M}}$, that share x_{K^*} as a vertex. This set of polygons defines the boundary dual mesh $\partial\mathfrak{M}^*$. We denote the dual mesh as $\overline{\mathfrak{M}^*} = \mathfrak{M}^*\cup\partial\mathfrak{M}^*$. Fig. 1 illustrates two examples of a primal and a dual mesh. The measure of cell K (resp. K^*) is denoted as $|K|$ (resp. $|K^*|$).

For two neighboring primal cells, K and L, we consider the intersection of their boundaries, denoted as $\partial K\cap\partial L$, which corresponds to a line segment and is referred to as an edge of the mesh $\mathfrak{M}$. We denote this edge as $\sigma = K|L$. The set of edges that lie within the interior of the domain is denoted

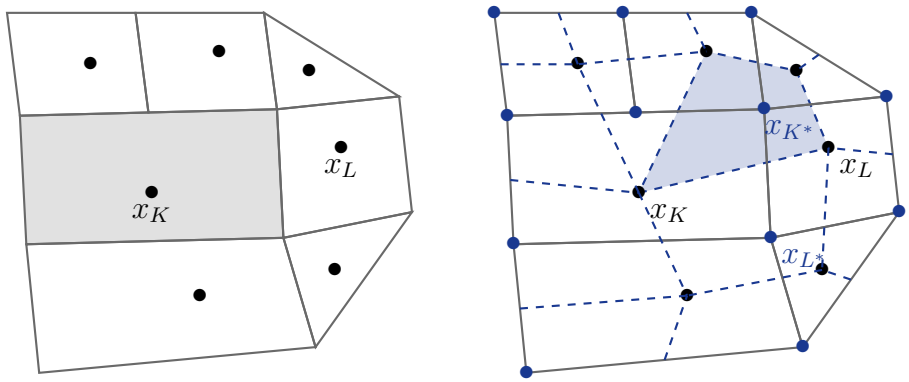

FIGURE 1. Left: Example of a primal mesh (grey). Black dots are the centers of each primal cell K, denoted by x_K. One cell is colored in light grey. Right: the corresponding dual mesh (blue). Blue dots are the centers of each dual cell K^* (resp. L^*) which are denoted by x_{K^*} (resp. x_{L^*}). An example of a dual cell is colored in light blue.

as $\mathcal{E}_{int}$. The set of primal edges on the boundary is denoted $\mathcal{E}_{ext}$. The total set of primal edges is $\mathcal{E} = \mathcal{E}_{int} \cup \mathcal{E}_{ext}$. Similarly, the set of edges in the interior of the dual mesh is denoted as $\mathcal{E}^*_{int}$, the set of edges on the boundary of the dual mesh is $\mathcal{E}^*_{ext}$, and we define $\mathcal{E}^* = \mathcal{E}^*_{int} \cup \mathcal{E}^*_{ext}$, the total set of dual edges. For each pair $(\sigma, \sigma^*) \in \mathcal{E} \times \mathcal{E}^*$, where $\sigma = (x_{K^*}, x_{L^*})$ and $\sigma^* = (x_K, x_L)$, we define a quadrilateral diamond cell, denoted as $\mathcal{D}$, with σ and σ^* as its diagonals. In the case where σ is a boundary edge, the diamond $\mathcal{D}$ degenerates into a triangle. The entire set of diamond cells forms the diamond mesh, denoted as $\mathfrak{D}$, which can be expressed as $\mathfrak{D} = \mathfrak{D}_{ext} \cup \mathfrak{D}_{int}$. Here, $\mathfrak{D}_{ext}$ represents the set of boundary diamonds, where $\sigma \subset \partial\Omega$, and $\mathfrak{D}_{int}$ represents the set of interior diamonds, where $\sigma \not\subset \partial\Omega$.

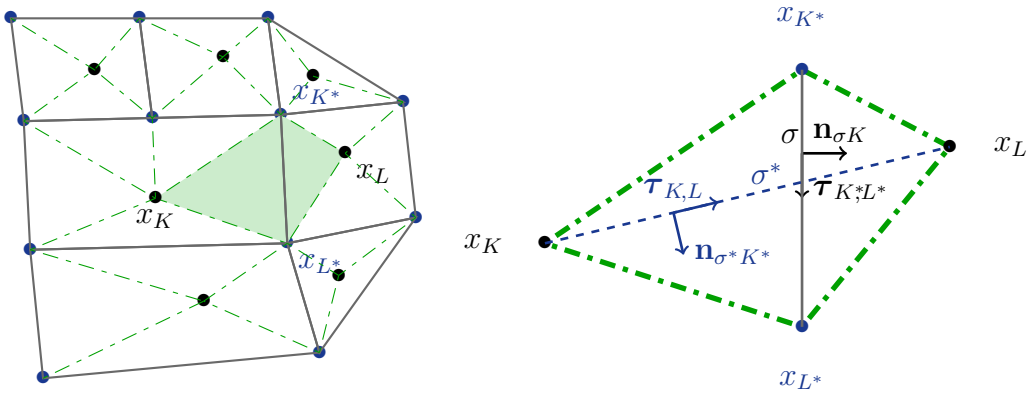

FIGURE 2. Left: Diamond mesh $\mathfrak{D}$ (green). One cell is colored in light green. Right: A diamond $\mathcal{D}$ (green) with direct basis $(\boldsymbol{\tau}_{K^*,L^*}, \mathbf{n}_{\sigma K})$ and $(\mathbf{n}_{\sigma^* K^*}, \boldsymbol{\tau}_{K,L})$. σ (resp. σ^*) is the primal (resp. dual) edge.

In summary, the DDFV mesh $\mathcal{T}$ consists in the union of the primal mesh $\overline{\mathfrak{M}}$ and the dual mesh $\overline{\mathfrak{M}^*}$. For a diamond $\mathcal{D}$ with vertices $(x_{K^*}, x_L, x_{L^*}, x_K)$, we define $x_\mathcal{D}$ as its center, $|\mathcal{D}|$ as its measure, $|\sigma|$ as the length of the primal edge σ, $|\sigma^*|$ as the length of the dual edge σ^*, and $\alpha_\mathcal{D}$ as the angle between the vectors (x_L, x_K) and (x_{L^*}, x_{K^*}). Additionally, we define two mutually orthogonal basis vectors, $(\boldsymbol{\tau}_{K^*,L^*}, \mathbf{n}_{\sigma K})$, and $(\mathbf{n}_{\sigma^* K^*}, \boldsymbol{\tau}_{K,L})$, where $\mathbf{n}_{\sigma K}$ is the unit vector normal to σ oriented from K to L, $\mathbf{n}_{\sigma^* K^*}$ is the unit vector normal to σ^* oriented from K^* to L^*, $\boldsymbol{\tau}_{K^*,L^*}$ is the unit tangent vector to σ oriented from x_{K^*} to x_{L^*}, and $\boldsymbol{\tau}_{K,L}$ is the unit tangent vector to σ^* oriented from x_K to x_L (Fig. 2).

It is worth noting that the area of a diamond cell $\mathcal{D}$ is given by $|\mathcal{D}| = \frac{1}{2}|\sigma||\sigma^*|\sin(\alpha_{\mathcal{D}})$. Furthermore, we define two subsets for the diamond cells as follows:

$$\mathfrak{D}_K = \{\mathcal{D} \in \mathfrak{D}, \sigma \in \mathcal{E}_K\}, \quad \mathfrak{D}_{K^*} = \{\mathcal{D} \in \mathfrak{D}, \sigma^* \in \mathcal{E}_{K^*}\}.$$

where:
- $\mathcal{E}_K$ is the set of edges of $K \in \mathfrak{M} \cup \partial\mathfrak{M}$. Note that for all $K \in \partial\mathfrak{M}$, $\mathcal{E}_K$ is reduced to one edge, i.e. $\sigma = K$.
- $\mathcal{E}_{K^*}$ is the set of edges of $K^* \in \mathfrak{M}^* \cup \partial\mathfrak{M}^*$.

We consider mixed boundary conditions, and thus define four subsets of the boundary mesh. The boundary primal and dual meshes for the Dirichlet boundary condition are:

$$\partial\mathfrak{M}_{Dir} = \{K \in \partial\mathfrak{M} : x_K \in \Gamma_{Dir}\}, \quad \partial\mathfrak{M}^*_{Dir} = \{K^* \in \partial\mathfrak{M}^* : x_{K^*} \in \Gamma_{Dir}\}.$$

The boundary dual mesh for the Neumann boundary condition is:

$$\partial\mathfrak{M}^*_{Neu} = \{K^* \in \partial\mathfrak{M}^* : x_{K^*} \in \Gamma_{Neu} \backslash \Gamma_{Dir}\}.$$

Finally, the boundary diamond mesh for the Neumann boundary condition is:

$$\mathfrak{D}_{ext,Neu} = \{\mathcal{D} \in \mathfrak{D}_{ext} : \sigma \in \mathcal{D} \cap \Gamma_{Neu}\}.$$

Note that per definition, there is a one-to-one relationship between the edges of primal cells and diamonds, and between the edges of dual cells and diamonds.
We finally denote Z^*_S the set of dual cells on a segment S defined as:

$$Z^*_S = \{K^* \in \overline{\mathfrak{M}^*}, \text{ such that } K^* \cap S \neq \emptyset\} \tag{1.1}$$

and K^*_X the unique dual cell containing a point X:

$$K^*_X = \{K^* \in \overline{\mathfrak{M}^*}, \text{ such that } K^* \cap X \neq \emptyset\} \tag{1.2}$$

1.2. Discrete unknowns and operators

The DDFV method enables the construction of two-dimensional discrete gradient and divergence operators that exhibit duality in a discrete sense. A description of the duality framework can be found in [35] and [3]. In this subsection, we introduce the discrete unknowns, the discrete gradient and divergence operators, as well as the reconstruction operator, that maps the unknowns on the primal and dual meshes on the diamond mesh. We first describe the sets of discrete unknowns. We define $\mathbb{R}^{\mathcal{T}}$ the linear space of scalar fields that are constant over the cells of $\overline{\mathfrak{M}}$ and $\overline{\mathfrak{M}^*}$:

$$u^{\mathcal{T}} \in \mathbb{R}^{\mathcal{T}} \Longleftrightarrow u^{\mathcal{T}} = \left((u^K)_{K\in\overline{\mathfrak{M}}}, (u^{K^*})_{K^*\in\overline{\mathfrak{M}^*}}\right).$$

We set $\left(\mathbb{R}^2\right)^{\mathfrak{D}}$ the linear space of vector fields constant on the diamonds:

$$\boldsymbol{\xi}^{\mathfrak{D}} \in \left(\mathbb{R}^2\right)^{\mathfrak{D}} \Longleftrightarrow \boldsymbol{\xi}^{\mathfrak{D}} = \left(\boldsymbol{\xi}^{\mathcal{D}}\right)_{\mathcal{D}\in\mathfrak{D}}.$$

We also denote $\mathbb{R}^{\mathfrak{D}}$ as the set of scalar fields that are constant over the diamonds. The discrete gradient operator is defined as a mapping from $\mathbb{R}^{\mathcal{T}}$ to $\left(\mathbb{R}^2\right)^{\mathfrak{D}}$, denoted by $\nabla^{\mathfrak{D}} u^{\mathcal{T}} = (\nabla^{\mathcal{D}} u^{\mathcal{T}})_{\mathcal{D}\in\mathfrak{D}}$ for all $u^{\mathcal{T}} \in \mathbb{R}^{\mathcal{T}}$. For each diamond $\mathcal{D} \in \mathfrak{D}$:

$$\nabla^{\mathcal{D}} u^{\mathcal{T}} = \frac{1}{2|\mathcal{D}|}[|\sigma|(u^L - u^K)\mathbf{n}_{\sigma K} + |\sigma^*|(u^{L^*} - u^{K^*})\mathbf{n}_{\sigma^* K^*}]. \tag{1.3}$$

The discrete divergence operator is a mapping from $\left(\mathbb{R}^2\right)^{\mathfrak{D}}$ to $\mathbb{R}^{\mathcal{T}}$ given by, for all $\boldsymbol{\xi}^{\mathfrak{D}} \in \left(\mathbb{R}^2\right)^{\mathfrak{D}}$:

$$\mathrm{div}^{\mathcal{T}} \boldsymbol{\xi}^{\mathfrak{D}} = \left(\mathrm{div}^{\mathfrak{M}} \boldsymbol{\xi}^{\mathfrak{D}}, \mathrm{div}^{\partial\mathfrak{M}} \boldsymbol{\xi}^{\mathfrak{D}}, \mathrm{div}^{\mathfrak{M}^*} \boldsymbol{\xi}^{\mathfrak{D}}, \mathrm{div}^{\partial\mathfrak{M}^*} \boldsymbol{\xi}^{\mathfrak{D}}\right). \tag{1.4}$$

The divergence on the primal mesh is $\mathrm{div}^{\mathfrak{M}} \boldsymbol{\xi}^{\mathfrak{D}} = (\mathrm{div}^K \boldsymbol{\xi}^{\mathfrak{D}})_{K\in\mathfrak{M}}$ and $\mathrm{div}^{\partial\mathfrak{M}} \boldsymbol{\xi}^{\mathfrak{D}} = 0$, with:

$$\mathrm{div}^K \boldsymbol{\xi}^{\mathfrak{D}} = \frac{1}{|K|} \sum_{\mathcal{D}\in\mathfrak{D}_K} |\sigma| \boldsymbol{\xi}^{\mathcal{D}} \cdot \mathbf{n}_{\sigma K}, \ \forall K \in \mathfrak{M}.$$

The divergence on the dual mesh is as well $\mathrm{div}^{\mathfrak{M}^*}\boldsymbol{\xi}^{\mathfrak{D}} = (\mathrm{div}^{K^*}\boldsymbol{\xi}^{\mathfrak{D}})_{K^*\in\mathfrak{M}^*}$ and $\mathrm{div}^{\partial\mathfrak{M}^*}\boldsymbol{\xi}^{\mathfrak{D}} = (\mathrm{div}^{K^*}\boldsymbol{\xi}^{\mathfrak{D}})_{K^*\in\partial\mathfrak{M}^*}$ with:

$$\mathrm{div}^{K^*}\boldsymbol{\xi}^{\mathfrak{D}} = \frac{1}{|K^*|}\sum_{\mathcal{D}\in\mathfrak{D}_{K^*}} |\sigma^*|\boldsymbol{\xi}^{\mathcal{D}}\cdot\mathbf{n}_{\sigma^*K^*}, \qquad \forall K^*\in\mathfrak{M}^*,$$

$$\mathrm{div}^{K^*}\boldsymbol{\xi}^{\mathfrak{D}} = \frac{1}{|K^*|}\left(\sum_{\mathcal{D}\in\mathfrak{D}_{K^*}} |\sigma^*|\boldsymbol{\xi}^{\mathcal{D}}\cdot\mathbf{n}_{\sigma^*K^*} + \sum_{\mathcal{D}\in\mathfrak{D}_{K^*}\cap\mathfrak{D}_{ext}} \frac{|\sigma|}{2}\boldsymbol{\xi}^{\mathcal{D}}\cdot\mathbf{n}_{\sigma K}\right), \qquad \forall K^*\in\partial\mathfrak{M}^*.$$

We introduce $r^{\mathfrak{D}}$, a reconstruction operator on diamonds that maps $\mathbb{R}^{\mathcal{T}}$ to $\mathbb{R}^{\mathfrak{D}}$. For any $u^{\mathcal{T}}\in\mathbb{R}^{\mathcal{T}}$, $r^{\mathfrak{D}}[u^{\mathcal{T}}]$ is defined as $(r^{\mathcal{D}}(u^{\mathcal{T}}))_{\mathcal{D}\in\mathfrak{D}}$, where $\mathcal{D}\in\mathfrak{D}$ and its vertices are denoted as $(x_K, x_{K^*}, x_L, x_{L^*})$:

$$r^{\mathcal{D}}(u^{\mathcal{T}}) = \frac{1}{4}\left(u^K + u^L + u^{K^*} + u^{L^*}\right). \tag{1.5}$$

1.3. Discrete notations

In the following, we introduce some useful notations. Let N be a positive integer. We denote $dt = T_f/N$ and $t_n = n\times dt$ for $n\in\{0,...,N\}$. We define the discrete space $\mathbb{R}^{\mathcal{T},dt} := \left(\mathbb{R}^{\mathcal{T}}\right)^{N+1}$ and its associated discrete vector $u^{\mathcal{T},dt}\in\mathbb{R}^{\mathcal{T},dt}$. We also define the discrete space $\mathbb{R}^{\mathcal{T}\times\mathcal{T},dt} := \left(\mathbb{R}^{\mathcal{T}\times\mathcal{T}}\right)^{N+1}$. Furthermore, for $n\in\mathbb{N}$ and a given function u, we define the discrete projection $\mathbb{P}_m^{\mathcal{T},n}u = (\mathbb{P}_m^{\mathfrak{M},n}u, \mathbb{P}_m^{\mathfrak{M}^*,n}u, \mathbb{P}_m^{\partial\Omega,n}u)$ as follows:

$$\mathbb{P}_m^{\mathfrak{M},n}u = \left(\frac{1}{|K|}\int_K u(t_n,\mathbf{x})\mathrm{d}\mathbf{x}\right)_{K\in\mathfrak{M}}, \quad \mathbb{P}_m^{\mathfrak{M}^*,n}u = \left(\frac{1}{|K^*|}\int_{K^*} u(t_n,\mathbf{x})\mathrm{d}\mathbf{x}\right)_{K^*\in\mathfrak{M}^*},$$
$$\mathbb{P}_m^{\partial\Omega,n}u = \left(\left(\frac{1}{|K|}\int_K u(t_n,\mathbf{x})\mathrm{d}\mathbf{x}\right)_{K\in\partial\mathfrak{M}}, \left(\frac{1}{|K^*|}\int_{K^*} u(t_n,\mathbf{x})\mathrm{d}\mathbf{x}\right)_{K^*\in\partial\mathfrak{M}^*}\right).$$

Also, we give Dirichlet discrete projections:

$$\mathbb{P}_m^{\partial\mathfrak{M}_{Dir},n}u = \left(\frac{1}{|K|}\int_K u(t_n,\mathbf{x})\mathrm{d}\mathbf{x}\right)_{K\in\partial\mathfrak{M}_{Dir}}, \quad \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},n}u = \left(\frac{1}{|K^*|}\int_{K^*} u(t_n,\mathbf{x})\mathrm{d}\mathbf{x}\right)_{K^*\in\partial\mathfrak{M}^*_{Dir}}.$$

Thereafter, we define $\mathbb{P}_m^{\mathcal{T},dt}$, $\mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt}$ and $\mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt}$ such that:

$$\mathbb{P}_m^{\mathcal{T},dt}u := (\mathbb{P}_m^{\mathcal{T},n}u)_{n\in\{0,...,N\}},$$
$$\mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt}u := (\mathbb{P}_m^{\partial\mathfrak{M}_{Dir},n}u)_{n\in\{0,...,N\}}, \qquad \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt}u := \left(\mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},n}u\right)_{n\in\{0,...,N\}}.$$

Building upon the previous sections, the source terms are discretized as follows:

$$f_u^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt}f_u,$$

and the initial conditions such as:

$$u^{\mathcal{T},0} = \mathbb{P}_m^{\mathcal{T},0}u^0.$$

Next, we define the discretization for the Dirichlet boundary condition functions:

$$u_{Dir}^{\partial\mathfrak{M}_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt}u^{Dir}, \quad u_{Dir}^{\partial\mathfrak{M}^*_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt}u^{Dir}.$$

The same way, we define the discretization of the Neumann boundary condition functions g:

$$g_{Neu}^{\mathfrak{D}_{ext},dt} = \left((g^{\sigma,n})_{\sigma \in \mathfrak{D}_{ext,Neu}} \right)_{n \in \{0,\dots,N\}} \quad \text{with} \quad g^{\sigma,n} = \int_{\sigma} g(t_n, \mathbf{x}) \mathrm{d}\mathbf{x}.$$

Finally, in Chapters 2 and 3, we present numerical results in the form of convergence tables containing numerical errors and associated convergence orders. For this specific purpose, we define the discrete norm $L^2(\Omega)$:

$$\forall u^{\mathcal{T}} \in \mathbb{R}^{\mathcal{T}}, \quad \|u^{\mathcal{T}}\|_{\mathcal{T},2} = \left(\frac{1}{2} \sum_{K \in \mathfrak{M}} |K| |u^K|^2 + \frac{1}{2} \sum_{K^* \in \overline{\mathfrak{M}^*}} |K^*| |u^{K^*}|^2 \right)^{\frac{1}{2}}. \tag{1.6}$$

Additionally, we define the discrete norm $L^\infty((0,T_f);L^2(\Omega))$ by:

$$\forall u^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T},dt}, \quad \|u^{\mathcal{T},dt}\|_{\mathcal{T},dt,\infty} = \max_{n \in \{0,\dots,N\}} \left(\frac{1}{2} \sum_{K \in \mathfrak{M}} |K| |u^{K,n}|^2 + \frac{1}{2} \sum_{K^* \in \overline{\mathfrak{M}^*}} |K^*| |u^{K^*,n}|^2 \right)^{\frac{1}{2}}. \tag{1.7}$$

CHAPTER 2

Linear coupling of the Poisson-Nernst Planck system of equations using the DDFV method

Introduction

This chapter presents a linear scheme, coupling the Poisson and the Nernst Planck equation within the DDFV framework. We introduce the coupling method on a PNP system consisting of a single Nernst-Planck (NP) equation for the concentration of one ionic species c_P, and the Poisson equation for the potential V. We choose to consider only one species, as we expect that this coupling strategy would not always converge. Also, we set the coefficients D_P, β, and γ equal to 1. Consequently, we realize several numerical tests on this linear coupling and draws the conclusion that the linear coupling is not appropriate for our system of equations. For spatial discretization, we employ the DDFV method, which was introduced in Chapter 1, along with an implicit Euler scheme for time discretization. Furthermore, only Dirichlet boundary conditions are considered.
Let $\Omega \subset \mathbb{R}^2$ be a connected open bounded domain and $T_f > 0$ a finite time. The Poisson equation is defined as follows:

$$\begin{aligned} -\nabla \cdot (\nabla V) &= c_P, && \text{in } \Omega \times [0, T_f], \\ V &= V^{Dir}, && \text{on } \partial\Omega \times [0, T_f], \\ V(\mathbf{x}, t = 0) &= V^0 && \text{in } \Omega. \end{aligned} \tag{2.1}$$

where $V^{Dir} \in L^2(\Gamma_{Dir} \times [0, T_f])$ and $V^0 \in L^2(\Omega)$.
The Nernst-Planck (NP) equation is defined as follows, using the same reformulation as [18] to ensure the positivity of the concentration:

$$\begin{aligned} \partial_t c_P &= \nabla \cdot (c_P \nabla (\log c_P + V)) + f_{c_P}, && \text{in } \Omega \times [0, T_f], \\ c_P &= c_P^{Dir}, && \text{on } \partial\Omega \times [0, T_f], \\ c_P(\mathbf{x}, t = 0) &= c_P^0 && \text{in } \Omega. \end{aligned} \tag{2.2}$$

where $c_P^{Dir} \in L^2(\Gamma_{Dir} \times [0, T_f])$, $c_P^0 \in L^2(\Omega)$ and $f_{c_P} \in L^2(\Omega \times [0, T_f])$ is a non-zero source term.

In this chapter, we will draw comparisons between solving the NP equation (2.2) with a potential V given by an analytical formula (as in [18]), that we call the NP-solver and solving the PNP equations (2.1)-(2.2) with a linear coupling, called the LC-solver. The latter system involves solving Poisson's equation (2.1) with a concentration c_P given at the previous time step. Then, using the obtained potential, a new concentration is calculated as in [18]. It is important to note that the difference between the two approaches is whether the potential V in (2.2) is given by an analytical function or derived from solving the Poisson equation.

Numerical scheme of the NP-solver

We now describe the numerical scheme solving the NP equations alone (NP-solver), as in [18]. We denote $c^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T},dt}$ the numerical solution of the NP equation (2.2) obtained using an analytical potential V_{ex} that is projected such that $V_{\text{ex}}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} V_{\text{ex}}$, with dt the time step. The Dirichlet boundary conditions of the solution c of (2.2) are discretized such as:

$$c_{Dir}^{\partial\mathfrak{M}_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt} c^{Dir}, \quad c_{Dir}^{\partial\mathfrak{M}^*_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt} c^{Dir}.$$

In the same way, the initial conditions of c, the source term f_c and the manufactured potential V_{ex} are also discretized:

$$c^{\mathcal{T},0} = \mathbb{P}_m^{\mathcal{T},0} c^0, \quad f_c^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} f_c, \quad V_{\text{ex}}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} V_{\text{ex}}.$$

The numerical scheme of the NP equation (2.2) is integrated over $\mathfrak{M}$ and $\mathfrak{M}^*$, seeking solutions $c^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T},dt}$ that satisfy the following nonlinear problem for all $n \geq 0$:

$$\begin{aligned}
&\frac{c^{K,n+1} - c^{K,n}}{dt} + \operatorname{div}^K\left(\mathbf{J}_c^{\mathfrak{D},n+1}\right) = f_c^{K,n+1}, && \forall K \in \mathfrak{M},\\
&\frac{c^{K^*,n+1} - c^{K^*,n}}{dt} + \operatorname{div}^{K^*}\left(\mathbf{J}_c^{\mathfrak{D},n+1}\right) = f_c^{K^*,n+1}, && \forall K^* \in \mathfrak{M}^*,\\
&\mathbf{J}_c^{\mathfrak{D},n+1} = -r^{\mathfrak{D}}(c^{\mathcal{T},n+1})\nabla^{\mathfrak{D}}\left(\log c^{\mathcal{T},n+1} + V_{ex}^{\mathcal{T},n+1}\right)
\end{aligned} \tag{2.3}$$

$$\begin{aligned}
&c^{K,n+1} = c_{Dir}^{K,n+1}, && \forall K \in \partial\mathfrak{M}_{Dir},\\
&c^{K^*,n+1} = c_{Dir}^{K^*,n+1}, && \forall K^* \in \partial\mathfrak{M}^*_{Dir}.
\end{aligned} \tag{2.4}$$

The presence of the logarithm introduces nonlinearity, which is handled by a Newton algorithm at each time step:

$$u^{k+1} = u^k - J_k^{-1} F(u^k) \tag{2.5}$$

where $k \geq 0$ is the Newton iteration, u^k the unknown, J_k is the Jacobian matrix and F the function that should vanish. For the NP-solver, the components of the Newton's algorithm are denoted as $\mathbf{M}_{\text{NP}}^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T}\times\mathcal{T},dt}$ for the Jacobian J and $\mathbf{F}_{\text{NP}}^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T},dt}$ for the function F. The latter is defined by:

$$\begin{aligned}
&\mathbf{F}_{\text{NP}}^{K,n+1} = \frac{c^{K,n+1} - c^{K,n}}{dt} + \operatorname{div}^K\left(\mathbf{J}_c^{\mathfrak{D},n+1}\right) - f_c^{K,n+1}, && \forall K \in \mathfrak{M},\\
&\mathbf{F}_{\text{NP}}^{K^*,n+1} = \frac{c^{K^*,n+1} - c^{K^*,n}}{dt} + \operatorname{div}^{K^*}\left(\mathbf{J}_c^{\mathfrak{D},n+1}\right) - f_c^{K^*,n+1}, && \forall K^* \in \mathfrak{M}^*,
\end{aligned} \tag{2.6}$$

$$\begin{aligned}
&\mathbf{F}_{\text{NP}}^{K,n+1} = c^{K,n+1} - c_{Dir}^{K,n+1}, && \forall K \in \partial\mathfrak{M}_{Dir},\\
&\mathbf{F}_{\text{NP}}^{K^*,n+1} = c^{K^*,n+1} - c_{Dir}^{K^*,n+1}, && \forall K^* \in \partial\mathfrak{M}^*_{Dir}.
\end{aligned}$$

Numerical analysis of the reformulated equation (2.3)-(2.4) has been published in [**18**], where the authors determine that it exists a solution for this problem. They also show a second order of convergence in space in the $L^\infty((0,T_f);L^2(\Omega))$ norm.

In this chapter, we build a linear coupling of the PNP system of equation (2.1)-(2.2) in the DDFV framework (the LC-solver), from two existing Fortran90 programs: one solving the Poisson equation (2.1), and the NP-solver described above, coming from [**18**], that addresses the NP equation (2.2) with the nonlinear reformulation of the flux term.
We then run simulations of the LC-solver, and investigate the results of the coupling through detailed numerical tests. Indeed, we begin by comparing the discrete solutions obtained from manufactured test cases using our LC-solver approach and the approximate solutions obtained from the NP-solver. Afterwards, we analyze the accuracy of the Jacobian matrix computed in both codes during the simulations. We observe that, after a several time iterations, the LC-solver does not give a good approximation of the solutions, since the error does not decrease with the decrease of mesh size.
We finally present results from a Scilab code as a benchmark to our coupling approach. More precisely, the Scilab code serves as a numerical debugging tool and solves our PNP system on a cartesian mesh consisting of four equal cells within a unit square domain. The purpose of the latter code is to replicate the LC-solver of the PNP system at a single time iteration. Based on the same test case, we obtain identical solutions as in the LC-solver at each iteration, confirming the correct implementation of the LC-solver on this small domain. However, we cannot assert that the LC-solver works perfectly on all types of meshes.
In the present chapter, we strive to explain the numerical reasons behind this non-convergence. Unfortunately, despite our extensive numerical testing of the LC-solver, yet we have not been able to show a concrete reason for its non-convergence.

The present chapter is organized as follows: in Section 2.1, we present the structure of the LC-solver and introduce the corresponding numerical scheme. Section 2.2 presents our simulation results, as well as the numerical tests. Finally, in Section 2.3, we use the Scilab code, as previously mentioned, to check the mechanism of the LC-solver in a unit square domain consisting of four Cartesian equal cells.

2.1. Presentation of the linear coupling of the PNP system

In this section, we present the numerical scheme for the LC-solver. The PNP system consists in two equations (2.1)-(2.2) with two unknowns (c_P, V). The numerical simulations are conducted using both triangular and cartesian meshes within an unit square domain. Each of the two mesh families consists of four refinement levels with progressively smaller spatial steps. The main goal is to examine the performance of the LC-solver method and to analyze the behavior of the PNP system (2.1)-(2.2) within this context. The discrete unknowns of the system are denoted as $(c_P^{\mathcal{T},dt}, V^{\mathcal{T},dt}) \in \mathbb{R}^{\mathcal{T},dt}$. As previously mentioned, the coupling of the system (2.1)-(2.2) is achieved by injecting the initial condition $c_P^{\mathcal{T},0} \in \mathbb{R}^{\mathcal{T}}$ into the routine solving the Poisson equation, resulting in an approximate solution $V^{\mathcal{T},1} \in \mathbb{R}^{\mathcal{T}}$. Subsequently, $V^{\mathcal{T},1} \in \mathbb{R}^{\mathcal{T}}$ is injected into the routine solving the NP equation to obtain $c_P^{\mathcal{T},1} \in \mathbb{R}^{\mathcal{T}}$. This process is then iterated over time points with respect to some chosen time step, as mapped in Figure 1.

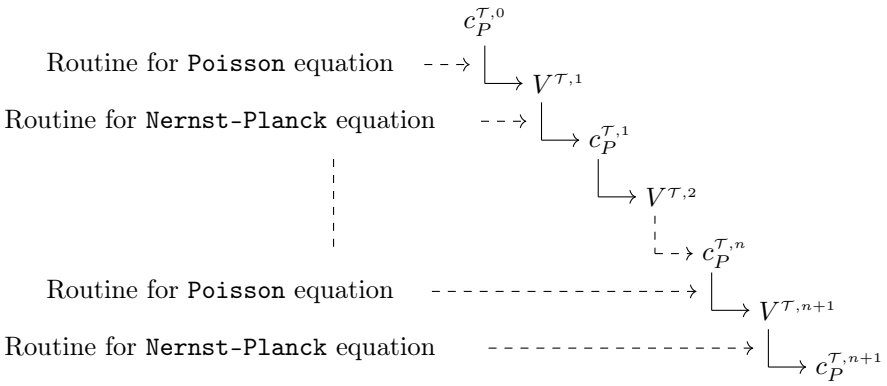

FIGURE 1. Structure of the LC-solver using separate routines for V in Poisson equation and for c_P in NP equation.

Building upon the previous Section 1.3, the source terms and initial conditions are discretized as follows:

$$f_{c_P}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} f_{c_P}, \quad f_V^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} f_V,$$

$$c_P^{\mathcal{T},0} = \mathbb{P}_m^{\mathcal{T},0} c_P^0, \quad V_{\mathcal{T}}^0 = \mathbb{P}_m^{\mathcal{T},0} V^0.$$

In addition, we define the discretizations for the Dirichlet values:

$$c_{P,Dir}^{\partial\mathfrak{M}_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt} c_P^{Dir}, \quad V_{Dir}^{\partial\mathfrak{M}_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt} V^{Dir},$$

$$c_{P,Dir}^{\partial\mathfrak{M}^*_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt} c_P^{Dir}, \quad V_{Dir}^{\partial\mathfrak{M}^*_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt} V^{Dir}.$$

Therefore, we solve the numerical scheme by integrating equations (2.1)-(2.2) over $\mathfrak{M}$ and $\mathfrak{M}^*$, seeking solutions $(c_P^{\mathcal{T},dt}, V^{\mathcal{T},dt}) \in \mathbb{R}^{\mathcal{T},dt}$ that satisfy the following nonlinear problem for all $n \geq 0$:

$$-\operatorname{div}^K(\nabla^{\mathfrak{D}} V^{K,n+1}) = c_P^{K,n}, \qquad \forall K \in \mathfrak{M}, \tag{2.7}$$

$$-\operatorname{div}^{K^*}(\nabla^{\mathfrak{D}} V^{K^*,n+1}) = c_P^{K^*,n}, \qquad \forall K^* \in \mathfrak{M}^*, \tag{2.8}$$

$$\frac{c_P^{K,n+1} - c_P^{K,n}}{dt} + \operatorname{div}^K\left(\mathbf{J}_{c_P}^{\mathfrak{D},n+1}\right) = f_{c_P}^{K,n+1}, \qquad \forall K \in \mathfrak{M}, \tag{2.9}$$

$$\frac{c_P^{K^*,n+1} - c_P^{K^*,n}}{dt} + \operatorname{div}^{K^*}\left(\mathbf{J}_{c_P}^{\mathfrak{D},n+1}\right) = f_{c_P}^{K^*,n+1}, \qquad \forall K^* \in \mathfrak{M}^*, \tag{2.10}$$

$$\mathbf{J}_{c_P}^{\mathfrak{D},n+1} = -r^{\mathfrak{D}}(c_P^{\mathcal{T},n+1}) \nabla^{\mathfrak{D}}\left(\log c_P^{\mathcal{T},n+1} + V^{\mathcal{T},n+1}\right). \tag{2.11}$$

The discrete Dirichlet boundary conditions are expressed as:

$$c_P^{K,n+1} = c_{P,Dir}^{K,n+1}, \qquad \forall K \in \partial\mathfrak{M}_{Dir}, \qquad c_P^{K^*,n+1} = c_{P,Dir}^{K^*,n+1}, \qquad \forall K^* \in \partial\mathfrak{M}^*_{Dir}, \tag{2.12}$$

$$V^{K,n+1} = V_{Dir}^{K,n+1}, \qquad \forall K \in \partial\mathfrak{M}_{Dir}, \qquad V^{K^*,n+1} = V_{Dir}^{K^*,n+1}, \qquad \forall K^* \in \partial\mathfrak{M}^*_{Dir}. \tag{2.13}$$

The existence of a solution (c_P, V) of (2.9)-(2.12) is proved in [18]. The components of the Newton's algorithm for (2.9)-(2.12) are denoted as $\mathbf{M}_{\mathrm{LC}}^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T}\times\mathcal{T},dt}$ for the Jacobian matrix and $\mathbf{F}_{\mathrm{LC}}^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T},dt}$ for the function F under the LC-solver. The latter is defined by:

$$\begin{aligned}
\mathbf{F}_{\mathrm{LC}}^{K,n+1} &= \frac{c_P^{K,n+1} - c_P^{K,n}}{dt} + \mathrm{div}^K\left(\mathbf{J}_{c_P}^{\mathfrak{D},n+1}\right) - f_{c_P}^{K,n+1}, && \forall K \in \mathfrak{M},\\
\mathbf{F}_{\mathrm{LC}}^{K^*,n+1} &= \frac{c_P^{K^*,n+1} - c_P^{K^*,n}}{dt} + \mathrm{div}^{K^*}\left(\mathbf{J}_{c_P}^{\mathfrak{D},n+1}\right) - f_{c_P}^{K^*,n+1}, && \forall K^* \in \mathfrak{M}^*,\\
\mathbf{F}_{\mathrm{LC}}^{K,n+1} &= c_P^{K,n+1} - c_{P,Dir}^{K,n+1}, && \forall K \in \partial\mathfrak{M}_{Dir},\\
\mathbf{F}_{\mathrm{LC}}^{K^*,n+1} &= c_P^{K^*,n+1} - c_{P,Dir}^{K^*,n+1}, && \forall K^* \in \partial\mathfrak{M}^*_{Dir}.
\end{aligned} \tag{2.14}$$

2.2. Numerical tests and analysis

In this section, we present the results obtained from three numerical test cases. Those test cases are constructed using manufactured solutions of the system (2.1)-(2.2). In the current chapter, the computational domain Ω is set to $]0,1[^2$. In our simulation procedure, we use families of increasingly refined meshes generated by the cartesian mesh family and the triangular mesh family. For each mesh in the family, the matching spatial steps h range from 7.071×10^{-1} to 8.839×10^{-2}, and from 4.692×10^{-1} to 6.253×10^{-2} respectively. The time step dt is initially given for the first mesh in the family. For subsequent meshes, the time step is divided by 4. Specifically, the second mesh has a time step of $dt/4$, the third mesh has a time step of $dt/4^2$, and the last mesh has a time step of $dt/4^3$. For each simulation, we provide the involved time related parameters, i.e, the time step and the final time. The first test case for the system (2.1)-(2.2) is denoted **Test 1**. The exact solutions $(c_{P,\mathrm{ex}}, V_{\mathrm{ex}})$ are:

$$\begin{cases} c_{P,\mathrm{ex}}(t,x,y) &= 2t^2 + 6x + 4, \\ V_{\mathrm{ex}}(t,x,y) &= -x^3 - 2y^2 - t^2x^2, \end{cases} \tag{2.15}$$

and the source term is:

$$f_{c_P}(t,x,y) = 54x^2 + (36t^2 + 48)x + 4t^4 + 16t^2 + 4t + 16. \tag{2.16}$$

Then, **Test 2** is defined such as:

$$\begin{cases} c_{P,\mathrm{ex}}(t,x,y) &= 10 + t + 12y^2, \\ V_{\mathrm{ex}}(t,x,y) &= -5x^2 - 0.5tx^2 - y^4, \end{cases} \tag{2.17}$$

with corresponding source term:

$$f_{c_P}(t,x,y) = t^2 + (20 + 24y^2)t + 77 + 240y^2 + 240y^4. \tag{2.18}$$

Finally, we define **Test 3** by:

$$\begin{cases} c_{P,\mathrm{ex}}(t,x,y) &= 10+6y+2t, \\ V_{\mathrm{ex}}(t,x,y) &= -y^3-ty-5x^2, \end{cases} \tag{2.19}$$

where the source term is:

$$f_{c_P}(t,x,y) = 4t^2+(36y+40)t+54y^2+120y+102. \tag{2.20}$$

The exact solutions will be projected onto $\mathbb{R}^{\mathcal{T},dt}$, which we denote $\{c_{P,\mathrm{ex}}^{\mathcal{T},dt}, V_{\mathrm{ex}}^{\mathcal{T},dt}\}$ and define as follows:

$$c_{P,\mathrm{ex}}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} c_{P,\mathrm{ex}}, \quad V_{\mathrm{ex}}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} V_{\mathrm{ex}}.$$

2.2.1. **Comparative analysis of the convergence orders from LC-solver and NP-solver.** In this section, the simulations are conducted with a time step $dt = 5\times10^{-3}$ (which correspond to a number of time steps $N = 100$) and a final time $T_f = 0.5$. We present two convergence tables, each displaying the $L^\infty((0,T_f);L^2(\Omega))$ errors of the discrete solutions of system (2.9)-(2.12) and (2.3)-(2.4) respectively, along with their corresponding convergence orders, based on the refinement levels of the used meshes (both Cartesian and triangular). To compute the errors, we use the discrete norm (1.7) defined in Section 1.3, and define the three errors ErrCp $= \|c_P^{\mathcal{T},dt} - c_{P,\mathrm{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},dt,\infty}$, ErrV $= \|V^{\mathcal{T},dt} - V_{\mathrm{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},dt,\infty}$ from the LC-solver (2.9)-(2.12) and ErrC $= \|c^{\mathcal{T},dt} - c_{\mathrm{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},dt,\infty}$ from the NP-solver (2.3)-(2.4). In Tables 1 and 2, we provide for each mesh family, the value of the three errors. Table 1 and Table 2 differ only in the meshes used, namely Cartesian for Table 1 and triangular for Table 2. Notably, we observe that the convergence orders stemming from the LC-solver exhibit significant deviations, including negative values, which is in contrast to the NP-solver whose orders tend towards 2. Indeed, the $L^\infty((0,T_f);L^2(\Omega))$ errors associated with the LC-solver grow with an increasing number of cells.

Cartesian	Convergence orders from LC-solver				Convergence orders from NP-solver	
h	ErrCp	Order	ErrV	Order	ErrC	Order
7.071E−1	6.769E−2	·	1.794E−1	·	1.150E−1	·
3.536E−1	8.074E−2	-0.25	1.089E−1	0.72	2.755E−2	2.06
1.768E−1	8.901E−2	-0.14	9.609E−2	0.18	6.669E−3	2.05
8.839E−2	9.308E−2	-0.06	9.338E−2	0.04	1.628E−3	2.03

TABLE 1. The error ErrCp, ErrV, ErrC and convergence orders from both codes for **Test 1** on the cartesian mesh family ($h = 7.071\times10^{-1}$ to 8.839×10^{-2}), with $dt = 5\times10^{-3}$ and $T_f = 0.5$.

Triangular	Convergence orders from LC-solver				Convergence orders from NP-solver	
h	ErrCp	Order	ErrV	Order	ErrC	Order
4.692E−1	8.127E−2	·	1.075E−1	·	2.717E−2	·
2.341E−1	8.917E−2	-0.13	9.63E−2	0.16	7.256E−3	1.90
1.212E−1	9.304E−2	-0.06	9.349E−2	0.04	1.842E−3	2.08
6.253E−2	9.497E−2	-0.03	9.272E−2	0.01	4.486E−4	2.13

TABLE 2. The error ErrCp, ErrV, ErrC and convergence orders from both codes for **Test 1** on the triangular mesh family ($h = 4.692\times10^{-1}$ to 6.253×10^{-2}), with $dt = 5\times10^{-3}$ and $T_f = 0.5$.

As we point out non-convergence of the coupling strategy, we provide thereafter the corresponding convergence tables of the LC-solver and the NP-solver, for both **Test 2** and **Test 3** on the triangular meshes. Thus, as previously, we exhibit the $L^\infty((0,T_f);L^2(\Omega))$ errors of the same quantities and the orders from both solvers in Table 3 and Table 4, respectively.

Triangular	Convergence orders from LC-solver				Convergence orders from NP-solver	
h	ErrCp	Order	ErrV	Order	ErrC	Order
4.692E−1	5.897E−2	·	4.696E−2	·	5.659E−3	·
2.341E−1	5.977E−2	-0.02	4.584E−2	0.03	1.625E−3	1.79
1.212E−1	6.051E−2	-0.02	4.546E−2	0.01	4.136E−4	2.08
6.253E−2	6.109E−2	-0.01	4.536E−2	0.00	1.024E−4	2.11

TABLE 3. The error ErrCp and ErrV, ErrC and convergence orders from both codes for **Test 2** on the triangular mesh family ($h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}), with $dt = 5 \times 10^{-3}$ and $T_f = 0.5$.

Triangular	Convergence orders from LC-solver				Convergence orders from NP-solver	
h	ErrCp	Order	ErrV	Order	ErrC	Order
4.692E−1	1.584E−1	·	9.545E−2	·	2.376E−2	·
2.341E−1	1.617E−1	-0.03	9.466E−2	0.01	6.619E−3	1.84
1.212E−1	1.633E−1	-0.01	9.426E−2	0.01	1.69E−3	2.07
6.253E−2	1.644E−1	-0.01	9.415E−2	0.00	4.123E−4	2.13

TABLE 4. The error ErrCp and ErrV, ErrC and convergence orders from both codes for **Test 3** on the triangular mesh family ($h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}), with $dt = 5 \times 10^{-3}$ and $T_f = 0.5$.

Note that for **Test 2** and **Test 3**, the NP-solver provides similar convergence orders as in **Test 1** from Table 2. As observed above, we obtain similar results from all cases for the LC-solver. In light of these non-converging results for the coupling approach, we closely examine results of a single time iteration. In the following, we consider only **Test 1**, as the behavior for the other two tests are very similar. minimal.

2.2.2. **Comparative analysis of the convergence orders from LC-solver and NP-solver at specific iterations.** Hereafter, we provide a convergence table displaying the $L^2(\Omega)$ errors at the first iteration $n = 1$ of the solutions obtained from both solvers: $c_P^{\mathcal{T},1}$ and $V^{\mathcal{T},1}$ for the LC-solver (2.9)-(2.12), and $c^{\mathcal{T},1}$ for the NP-solver (2.3)-(2.4). Again, the errors are computed with respect to the discrete $L^2(\Omega)$ norm (1.6) already defined in Section 1.3.

Triangular	Errors from LC-solver at $n=1$				Errors from NP-solver at $n=1$	
h	ErrCp^1	Order	ErrV^1	Order	ErrC^1	Order
4.692E−1	2.978E−4	·	3.397E−3	·	1.985E−3	·
2.341E−1	1.164E−4	1.35	8.508E−4	1.99	6.344E−4	1.64
1.212E−1	3.610E−5	1.77	2.12E−4	2.11	1.746E−4	1.95
6.253E−2	9.742E−6	1.97	5.282E−5	2.09	4.495E−5	2.05

TABLE 5. The error ErrCp^1, ErrV^1, ErrC^1 and convergence orders from both codes for **Test 1** on the triangular mesh family ($h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}) at $n = 1$, corresponding to $t = 5 \times 10^{-3}$. Simulations are run with the same $dt = 5 \times 10^{-3}$ and $T_f = 0.5$.

In Table 5, we give the value of the different errors at the first iteration denoted $\text{ErrCp}^1 = \|c_P^{\mathcal{T},1} - c_{P,\text{ex}}^{\mathcal{T},1}\|_{\mathcal{T},2}$, $\text{ErrV}^1 = \|V^{\mathcal{T},1} - V_{\text{ex}}^{\mathcal{T},1}\|_{\mathcal{T},2}$ from the LC-solver (2.9)-(2.12) and $\text{ErrC}^1 = \|c^{\mathcal{T},1} - c_{\text{ex}}^{\mathcal{T},1}\|_{\mathcal{T},2}$ from the NP-solver (2.3)-(2.4). Additionally, alongside these errors, we determine the corresponding

convergence orders of the solutions at the first iteration. One observe diminishing errors with increasing mesh refinement levels. At the highest refinement level, the error for $c^{\mathcal{T},1}$ is 10^{-5} for the NP-solver, whereas for the LC-solver, we obtain 10^{-5} for $V^{\mathcal{T},1}$ and 10^{-6} for $c_P^{\mathcal{T},1}$. The errors are very similar between both solvers; we even observe a smaller error on the concentration $c_P^{\mathcal{T},1}$ for the LC-solver. This leads us to believe that the LC-solver appears to be accurate in the first iteration. Also, the convergence results at $n = 1$ are in great accordance with the results from the NP-solver, which depicts an order of 2. In other words, the results from both solvers exhibit close similarity. It appears that, in the initial iteration, the LC-solver provides a good approximation of the solutions, and the non-convergence observed in the preceding section does not arise from the first time-step resolution.

Subsequently, in the LC-solver, we observe that the $L^2(\Omega)$ error of solution c_P start to increase at iteration 2, with the convergence order decreasing. This increase is more important with mesh refinement. We present the relative numerical results of iteration $n = 5$ in Table 6, for the three errors $\text{ErrCp}^5 = \|c_P^{\mathcal{T},5} - c_{P,\text{ex}}^{\mathcal{T},5}\|_{\mathcal{T},2}$, $\text{ErrV}^5 = \|V^{\mathcal{T},5} - V_{\text{ex}}^{\mathcal{T},5}\|_{\mathcal{T},2}$ and $\text{ErrC}^5 = \|c^{\mathcal{T},5} - c_{\text{ex}}^{\mathcal{T},5}\|_{\mathcal{T},2}$. Concerning the potential, the error decreases with the successive mesh refinement, but the order decreases, and reach 0.75 for the most refined mesh, when an order 2 would be expected. In the opposite, the numerical errors for the NP-solver behaves as expected, with a significant decreases, and a numerical convergence order around 2.

Triangular	Errors from LC-solver at $n = 5$				Errors from NP-solver at $n = 5$	
h	ErrCp^5	Order	ErrV^5	Order	ErrC^5	Order
4.692E−1	6.016E−4	·	3.539E−3	·	3.716E−3	·
2.341E−1	1.753E−4	1.77	9.72E−4	1.85	9.38E−4	1.98
1.212E−1	1.104E−4	0.70	3.456E−4	1.57	2.379E−4	2.08
6.253E−2	1.187E−4	-0.10	2.093E−4	0.75	6.448E−5	1.97

TABLE 6. The error ErrCp^5, ErrV^5, ErrC^5 and convergence orders from both codes for **Test 1** on the triangular mesh family ($h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}) at $n = 5$, corresponding to $t = 0.025$. Simulations are conducted with the same $dt = 5 \times 10^{-3}$ and $T_f = 0.5$.

These results indicate that with the current parameter settings, achieving a sufficiently high level of accuracy to match the results obtained from the NP-solver is not feasible. Consequently, our focus shifts toward obtaining more coherent results by changing the time parameters.

2.2.3. **Analysis of convergence orders w.r.t time-related parameters for LC-solver.** In this subsection, we now exclusively examine the results of the LC-solver. We change the time parameters (dt and T_f) in the aim to decrease the error and get a second order for the convergence. We set the final time to $T_f = 0.006$. Consequently, we present the results $\text{ErrCp} = \|c_P^{\mathcal{T},dt} - c_{P,\text{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},dt,\infty}$, $\text{ErrV} = \|V^{\mathcal{T},dt} - V_{\text{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},dt,\infty}$ from the LC-solver (2.9)-(2.12) in Table 7, with $dt = 6 \times 10^{-5}$. These results are carried out by employing the triangular mesh family. The simulations are conducted for **Test 1**.

Triangular	Convergence orders from LC-solver			
h	ErrCp	Order	ErrV	Order
4.692E−1	3.950E−4	·	3.411E−3	·
2.341E−1	1.453E−4	1.44	8.610E−4	1.99
1.212E−1	3.921E−5	1.89	2.217E−4	1.96
6.253E−2	1.061E−5	1.89	6.326E−5	1.81

TABLE 7. The error ErrCp, ErrV and convergence orders for **Test 1** (2.15)-(2.16) on the triangular mesh family ($h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}) with $T_f = 0.006$ and $dt = 6 \times 10^{-5}$.

Fortunately, these simulations yield consistent results. Indeed, the convergence orders of both c_P and the potential V, almost reach the expected value 2. These results demonstrate that the LC-solver (2.9)-(2.12) performs well for small final time and time step. Thereupon, one might naturally attempt replicating these results using the same time step, but with a larger final time, for instance take $T_f = 0.5$. ErrCp and ErrV from the LC-solver (2.9)-(2.12) are presented in Table 8 below:

Triangular	Convergence orders from LC-solver			
h	ErrCp	Order	ErrV	Order
4.692E−1	9.324E−2	·	9.504E−2	·
2.341E−1	9.444E−2	-0.01	9.316E−2	0.03
1.212E−1	9.541E−2	0.00	9.268E−2	0.01
6.253E−2	9.602E−5	0.001	9.192E−2	0.02

TABLE 8. The error ErrCp, ErrV and convergence orders for **Test 1** (2.15)-(2.16) on the triangular mesh family ($h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}) with $T_f = 0.5$ and $dt = 6 \times 10^{-5}$.

Based on the setting of Table 8, one may remark that the newly derived results do not exhibit convergence. The $L^\infty((0,T_f);L^2(\Omega))$ errors in the solution c_P increase as the mesh refinement progresses. A similar behavior is observed for the error on the potential V, ErrV. We then run simulation with an smaller time step, $dt = 10^{-7}$, and show ErrCp and ErrV in Table 9.

Triangular	Convergence orders from LC-solver			
h	ErrCp	Order	ErrV	Order
4.692E−1	9.247E−2	·	9.501E−2	·
2.341E−1	9.36E−2	-0.01	9.302E−2	0.03
1.212E−1	9.534E−2	-0.02	9.219E−2	0.01
6.253E−2	9.623E−2	-0.01	9.328E−2	-0.01

TABLE 9. The error ErrCp, ErrV and convergence orders for **Test 1** (2.15)-(2.16) on the triangular mesh family ($h = 7.071 \times 10^{-1}$ to 8.839×10^{-2}) with $T_f = 0.5$ and $dt = 1 \times 10^{-7}$.

The outcomes obtained with this excessively small time step reaffirm that the solutions non-converge. Consequently, the LC-solver (2.9)-(2.12) fails to accurately reconcile the solutions of c_P and V within the PNP system (2.1)-(2.2).

The numerical results indicate that the LC-solver (2.9)-(2.12) is limited to a very small final time. For instance, it shows consistent performance when the final time is set to $T_f = 0.006$ within the framework of **Test 1**. After these observations, we tried to achieve convergence orders close to 2 with a larger final time $T_f = 0.5$. Regrettably, despite employing a time step ($dt = 6 \times 10^{-5}$) akin to that of the reference effective final time ($T_f = 0.006$) (see Table 7), the results persistently exhibit inconsistency (see Table 8). Analogous conclusions were drawn from both **Test 2** and **Test 3**, both of which exhibit inconsistency as well. Thus, our results do not allow to conclusively check whether the PNP system is properly coupled under the LC-solver, as the solutions non-converge.

2.2.4. **Comparative analysis of the Jacobian matrices from LC-solver and NP-solver at initial iteration.** The present section is dedicated to verifying the implementation of the Jacobian matrix J under the Newton algorithm involved in the LC-solver (2.9)-(2.12) of the PNP system (2.1)-(2.2). Given the nonlinear nature of the system, Jacobian matrices are built up at each time step. The goal is to compare the outcomes of the two Jacobian matrices produced by both solvers at iteration $n = 1$ and $n = 5$ within the simulations of Section 2.2.1. Then, we assess the differences between those matrices, following the setting of **Test 1**. The Jacobian matrices are determined by calculating the derivatives of the functions F associated with each code, specifically $\mathbf{F}_{\mathrm{NP}}^{\mathcal{T},dt}$ and $\mathbf{F}_{\mathrm{LC}}^{\mathcal{T},dt}$, as defined in equations (2.6) and (2.14), respectively. Accordingly, these matrices are represented as $\mathbf{M}_{\mathrm{NP}}^{\mathcal{T},dt}$ and

$\mathbf{M}_{\mathrm{LC}}^{\mathcal{T},dt}$. The components of each respective matrix are stored in a single column vector, wherein each row is concatenated sequentially. For the iteration in time $n = 1$ (resp. $n = 5$), we denote these vectors as $\mathbf{Vec}_{\mathrm{LC}}^{\mathcal{T},1}$ (resp. $\mathbf{Vec}_{\mathrm{LC}}^{\mathcal{T},5}$) for the LC-solver and $\mathbf{Vec}_{\mathrm{NP}}^{\mathcal{T},1}$ (resp. $\mathbf{Vec}_{\mathrm{NP}}^{\mathcal{T},5}$) for the NP-solver, respectively.

We run simulation with $T_f = 0.5$ and $dt = 5 \times 10^{-3}$, with triangular meshes with refinement levels ranging from $h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}. In Table 10, we provide the infinity norm errors $\mathrm{ErrM}^1 = \max |\mathbf{Vec}_{\mathrm{LC}}^{\mathcal{T},1} - \mathbf{Vec}_{\mathrm{NP}}^{\mathcal{T},1}|$ for $n = 1$ and $\mathrm{ErrM}^5 = \max |\mathbf{Vec}_{\mathrm{LC}}^{\mathcal{T},5} - \mathbf{Vec}_{\mathrm{NP}}^{\mathcal{T},5}|$ for $n = 5$.

Triangular	$n=1$		$n=5$	
h	ErrM^1	Order	ErrM^5	Order
4.692E−1	1.62E−2	·	2.4E−2	·
2.341E−1	5.3E−3	1.60	6.1E−3	1.96
1.212E−1	1.5E−3	1.91	1.5E−3	2.09
6.253E−2	4E−4	1.99	3.7E−4	2.12

TABLE 10. The errors ErrM^1, ErrM^5 and their convergence orders for **Test 1** on the triangular mesh family ($h = 4.692 \times 10^{-1}$ to 6.253×10^{-2}) at $n = 1$ and $n = 5$. Simulations are conducted with $dt = 5 \times 10^{-3}$ and $T_f = 0.5$.

The infinity norm errors between the Jacobian matrices from both solvers at both iterations ($n = 1$ and $n = 5$) decrease with increasing refinement levels. Furthermore, we compute the associated convergence orders, observing coherent values that approach 2 as refinement levels increase. These results indicate that the implementation of the Jacobian matrix in the LC-solver (2.9)-(2.12) is correct.

Thereafter, for each solver (LC-solver and NP-solver), we aim at checking the differentiability of the function F involved in Newton's procedure, as one potential clue to the previous non-converging results. Thus, we compare the rate of change of the function F at a given point, say $(F(x+\ell y)-F(x))/\ell$, with the value of the Jacobian matrix times y, denoted $\mathbf{M} \cdot y$. To be more precise, we take randomly selected points x and y within the range of 0 to 1, and compute each term in (2.21) below, for both LC-solver (2.9)-(2.12) and NP-solver (2.3)-(2.4). Note that in order to evaluate the approximated errors associated to the quantity of interest (2.21), we decrease $\ell = 0.1, 0.01, 0.0001$ from both solvers at each time step. Thus, if the Jacobian matrix is well approximated, one hopes to observe that:

$$|(F(x + \ell y) - F(x))/\ell - \mathbf{M} \cdot y| \xrightarrow[\ell \to 0]{} 0. \tag{2.21}$$

The approximated values corresponding to (2.21) offer insights into the difference between both the rate of change of F and its Jacobian matrix.

Hence, in what follows, we compute the estimated maximum errors $\mathrm{ErrF}_{\mathrm{LC}}^n$ for the LC-solver (2.9)-(2.12) and $\mathrm{ErrF}_{\mathrm{NP}}^n$ for the NP-solver (2.3)-(2.4), respectively. These aforementioned maximum errors are defined as:

$$\mathrm{ErrF}_{\mathrm{LC}}^n(x, y, \ell) := \max \left| \frac{\mathbf{F}_{\mathrm{LC}}^{\mathcal{T},n}(x + \ell y) - \mathbf{F}_{\mathrm{LC}}^{\mathcal{T},n}(x)}{\ell} - \mathbf{M}_{\mathrm{LC}}^{\mathcal{T},n} \cdot y \right|,$$

$$\mathrm{ErrF}_{\mathrm{NP}}^n(x, y, \ell) := \max \left| \frac{\mathbf{F}_{\mathrm{NP}}^{\mathcal{T},n}(x + \ell y) - \mathbf{F}_{\mathrm{NP}}^{\mathcal{T},n}(x)}{\ell} - \mathbf{M}_{\mathrm{NP}}^{\mathcal{T},n} \cdot y \right|.$$

Also, we work under the triangular mesh with $h = 4.69 \times 10^{-1}$ and simulations are conducted with the same $dt = 5 \times 10^{-3}$ and $T_f = 0.5$. To illustrate the error evolutions for each ℓ, we select specific iterations in time such as $n = 1, 10, 50, 99$. Also, we present two tables where we make a distinction between primal and dual nodes. Specifically, in Table 11 (resp. Table 12), we only display results for the cell centers (resp. vertices), i.e., when $y = 0$ for the vertices (resp. centers). In other words, we define a vector $y^{\mathfrak{M}} \in \mathbb{R}^{\mathcal{T}}$ (resp. $y^{\mathfrak{M}^*} \in \mathbb{R}^{\mathcal{T}}$) such that we set $y_{K^*} = 0$ for Table 11 (resp. $y_K = 0$ for Table 12).

	$\mathrm{ErrF}^n_{\mathrm{LC}}(x, y^{\mathfrak{M}}, \ell)$ from LC-solver				$\mathrm{ErrF}^n_{\mathrm{NP}}(x, y^{\mathfrak{M}}, \ell)$ from NP-solver			
ℓ	$n=1$	$n=10$	$n=50$	$n=99$	$n=1$	$n=10$	$n=50$	$n=99$
0.1	3.6E−3	4.9E−3	6.1E−3	5.2E−3	9.2E−3	8.7E−3	5.5E−3	5E−3
0.01	6.24E−4	8.26E−4	4.79E−4	7.82E−4	7.1E−4	4E−4	4.3E−4	1E−3
0.0001	9.96E−6	5.6E−6	5.94E−6	7.46E−6	5.6E−6	4.3E−6	6.1E−6	3.5E−6

TABLE 11. Values of $\mathrm{ErrF}^n_{\mathrm{LC}}(x, y^{\mathfrak{M}}, \ell)$ and $\mathrm{ErrF}^n_{\mathrm{NP}}(x, y^{\mathfrak{M}}, \ell)$ for **Test 1** on the triangular mesh ($h = 4.692\mathrm{E}{-1}$) at $n = 1, 10, 50, 99$, for several $\ell = 0.1, 0.01, 0.0001$ and for the cell centers. Simulations are conducted with $dt = 5 \times 10^{-3}$ and $T_f = 0.5$.

	$\mathrm{ErrF}^n_{\mathrm{LC}}(x, y^{\mathfrak{M}^*}, \ell)$ from LC-solver				$\mathrm{ErrF}^n_{\mathrm{NP}}(x, y^{\mathfrak{M}^*}, \ell)$ from NP-solver			
ℓ	$n=1$	$n=10$	$n=50$	$n=99$	$n=1$	$n=10$	$n=50$	$n=99$
0.1	2.8E−3	2.2E−3	3.4E−3	2.4E−3	2.5E−3	2.4E−3	2.7E−3	2.3E−3
0.01	2.79E−4	1.41E−4	3.06E−4	2.25E−4	2.1E−4	3.75E−4	1.3E−4	4.6E−4
0.0001	3.05E−6	2.2E−6	3.98E−6	2.33E−6	1.7E−6	1.8E−6	1.5E−6	2.1E−6

TABLE 12. Values of $\mathrm{ErrF}^n_{\mathrm{LC}}(x, y^{\mathfrak{M}^*}, \ell)$ and $\mathrm{ErrF}^n_{\mathrm{NP}}(x, y^{\mathfrak{M}^*}, \ell)$ for **Test 1** on the triangular mesh ($h = 4.692\mathrm{E}{-1}$) at $n = 1, 10, 50, 99$, for several $\ell = 0.1, 0.01, 0.0001$ and for the cell vertices. Simulations are conducted with $dt = 5 \times 10^{-3}$ and $T_f = 0.5$.

Tables 11 and 12 illustrate that as ℓ decreases, the computed errors $\mathrm{ErrF}^n_{\mathrm{LC}}$ and $\mathrm{ErrF}^n_{\mathrm{NP}}$ converge towards zero. For instance, when $\ell = 0.0001$, values of $\mathrm{ErrF}^n_{\mathrm{LC}}$ are below 10^{-5}, with a minimum value of 5.6×10^{-6} (resp. 2.2×10^{-6}) occurring at $n = 10$ in Table 11 (resp. in Table 12). Moreover, for each iteration n, the $\mathrm{ErrF}^n_{\mathrm{LC}}$ values appear very similar as they have same order (for instance, when $\ell = 0.1$, for each n, the errors are about 10^{-3}). It is worth mentioning that all these maximum errors are attained every time at the centers of the cells, i.e., in Table 11. Additionally, it is notable that we observe highly similar results for the $\mathrm{ErrF}^n_{\mathrm{NP}}$ errors. This outcome affirms that the LC-solver correctly computes the partial derivatives embedded within the Jacobian matrix.

2.3. Comparison with an other implementation on a simple mesh

In this final section, we focus on the coupling mechanism by manually replicating the coupling of the PNP equations (2.1)-(2.2) using the Scilab programming software at a single time iteration. This approach allows to verify whether the LC-solver (2.9)-(2.12) we have implemented in Fortran90 is correct on a simple Cartesian mesh. We want to check if we obtain the same results between the code run in Scilab and the LC-solver at both the first ($n = 1$) and last ($n = 100$) iterations, for the simulation conducted in Section 2.2.1 on **Test 1** with $dt = 5 \times 10^{-3}$ and $T_f = 0.5$. To be precise, the Scilab code simulates the entire numerical process "manually", meaning it reproduces the Newton algorithm by constructing the Jacobian matrix and its associated function F at each Newton iteration. To achieve this, we use the coarsest possible Cartesian grid, i.e., $h = 7.071 \times 10^{-1}$ where we apply the DDFV method. We showcase this grid in Figure 2.

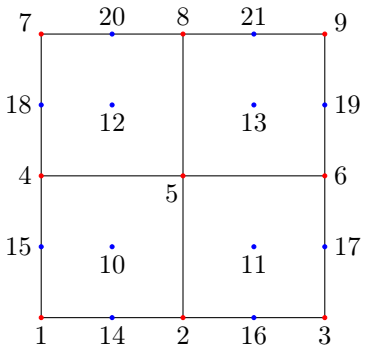

FIGURE 2. Cartesian mesh with $h = 7.071 \times 10^{-1}$. The mesh consists of 21 nodes, including centers in blue and vertices in red.

In Scilab, to apply the DDFV method on this grid, we assign a number to each grid node. For the Newton algorithm, we define $F_S^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T},dt}$ as the F function within the Scilab Code. Additionally, $\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T}\times\mathcal{T},dt}$ represents the Jacobian matrix. We begin by laying out the implementation steps of our Scilab code:

(1) From **Test 1** in the LC-solver, we retrieve $c_P^{\mathcal{T},0}$ (resp. $c_{P,S}^{\mathcal{T},99}$) and $V^{\mathcal{T},1}$ (resp. $V^{\mathcal{T},100}$) as initialisations of the Scilab code. They are defined as column vectors, sized according to the number of nodes on the grid, which consists of the 21 points representing the centers and vertices of the domain Ω (Figure 2).

(2) We define the measures of each cell (Figure 2) and the time parameters of the time-stepping scheme.

(3) We define numerical functions $F_S^{\mathcal{T},dt}$ and $\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}$ for the Newton algorithm on each cell (Figure 2) and their connections with other nodes, based on the DDFV method.

(4) We apply the Newton algorithm and we denote $c_{P,S}^{\mathcal{T},1}$ (resp. $c_{P,S}^{\mathcal{T},100}$) the solution for the first time step (resp. for the last time step).

(5) We compare our Scilab results $c_{P,S}^{\mathcal{T},1}$ for the first time step (resp. $c_{P,S}^{\mathcal{T},100}$ for the last time step) with $c_P^{\mathcal{T},1}$ (resp. $c_P^{\mathcal{T},100}$) the solutions for the LC-solver (2.9)-(2.12) presented in Section 2.2.1.

Thereafter, we will explicit the implementation of step (3) of the Scilab code described earlier. As a result, we consider only nodes 10, 11, 12, and 13 for the centers of interior primal cells, along with node 5 for the vertex of the single interior dual cell (see Figure 2). The functions $F_S^{\mathcal{T},dt}$ and their associated Jacobians $\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}$ have thus been implemented based on these nodes and their corresponding contributing nodes. Here, we only describe the components for the interior cells of the domain. The treatment of the boundary is straightforward since we employ Dirichlet-type conditions. We outline the implemented quantities of interest of node 5, corresponding to the single interior dual cell.

Details for the other nodes, namely 10, 11, 12, and 13, located at the centers of the four interior primal cells, are provided in the Appendix A. We begin by manually implementing the function $F_S^{\mathcal{T},dt}(5)$ while applying the DDFV method, that is, explicitly writing contributions of node 5 with the other nodes 2, 4, 6, 8, 10, 11, 12, 13 in the domain (see Figure 2).

$$
\begin{aligned}
F_S^{\mathcal{T},dt}(5) = \frac{|K^*|}{dt}(c_P^{\mathcal{T},n+1}(5) - c_P^{\mathcal{T},n}(5)) - f_{c_P}^{n+1}(5) \\
+ \frac{1}{4}(c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(13))\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) \\
- \log(c_P^{\mathcal{T},n+1}(6)) - V^{\mathcal{T},n+1}(6)\Big) \\
+ \frac{1}{4}(c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(13))\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) \\
- \log(c_P^{\mathcal{T},n+1}(8)) - V^{\mathcal{T},n+1}(8)\Big) \\
+ \frac{1}{4}(c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(10))\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) \\
- \log(c_P^{\mathcal{T},n+1}(4)) - V^{\mathcal{T},n+1}(4)\Big) \\
+ \frac{1}{4}(c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(10))\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) \\
- \log(c_P^{\mathcal{T},n+1}(2)) - V^{\mathcal{T},n+1}(2)\Big).
\end{aligned}
$$

Node 5 has coordinates $(x, y) = (0.5, 0.5)$, as depicted in Figure 2. Therefore, $f_{c_P}^{n+1}(5)$ corresponds to the source term computed at node 5 for **Test 1** (2.15)-(2.16) in this chapter.

Based on the formula of $F_S^{\mathcal{T},dt}(5)$, we calculate and implement the partial derivatives corresponding to each element of the Jacobian matrix $\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}$ for node 5. We recall that we have Dirichlet boundary conditions throughout the domain. All elements of the matrix in question are defined below. We begin by defining as follows the contributions of primal nodes with node 5, that is, the centers of the domain numbered 10, 11, 12, and 13:

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,11) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(6)) + V^{\mathcal{T},n+1}(6)) + \log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(2)) + V^{\mathcal{T},n+1}(2))\Big)$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,13) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(6)) + V^{\mathcal{T},n+1}(6)) + \log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(8)) + V^{\mathcal{T},n+1}(8))\Big)$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,12) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(8)) + V^{\mathcal{T},n+1}(8)) + \log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(4)) + V^{\mathcal{T},n+1}(4))\Big)$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,10) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(4)) + V^{\mathcal{T},n+1}(4)) + \log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(2)) + V^{\mathcal{T},n+1}(2))\Big).$$

Next, we carry on these implementations by now defining the contributions of dual node 5 with the other dual nodes connected to it, namely nodes 6, 8, 4, and 2:

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,6) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(6)) + V^{\mathcal{T},n+1}(6)) - (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6))\frac{1}{c_P^{\mathcal{T},n+1}(6)}\Big)$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,8) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(8)) + V^{\mathcal{T},n+1}(8)) - (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8))\frac{1}{c_P^{\mathcal{T},n+1}(8)}\Big)$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,4) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(4)) + V^{\mathcal{T},n+1}(4)) - (c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4))\frac{1}{c_P^{\mathcal{T},n+1}(4)}\Big)$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,2) = \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(2)) + V^{\mathcal{T},n+1}(2)) - (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2))\frac{1}{c_P^{\mathcal{T},n+1}(2)}\Big).$$

The final contribution involves implementing contributions from node 5, i.e., on node 5 itself. The element of the Jacobian matrix is derived as follows:

$$
\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(5,5) = \frac{|K^*|}{dt} &+ \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(6)) + V^{\mathcal{T},n+1}(6)) \\
&\qquad + (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6))\frac{1}{c_P^{\mathcal{T},n+1}(5)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - \log(c_P^{\mathcal{T},n+1}(8)) - V^{\mathcal{T},n+1}(8) \\
&\qquad + (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8))\frac{1}{c_P^{\mathcal{T},n+1}(5)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - (\log(c_P^{\mathcal{T},n+1}(4)) + V^{\mathcal{T},n+1}(4)) \\
&\qquad + (c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4))\frac{1}{c_P^{\mathcal{T},n+1}(5)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma^*|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(5)) + V^{\mathcal{T},n+1}(5) - \log(c_P^{\mathcal{T},n+1}(2)) - V^{\mathcal{T},n+1}(2) \\
&\qquad + (c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2))\frac{1}{c_P^{\mathcal{T},n+1}(5)}\Big).
\end{aligned}
$$

Last but not least, once the Scilab code is implemented, we compute the $L^2(\Omega)$ errors between the solutions from the LC-solver and the solutions from the Scilab code, at the first iteration $n = 1$ (resp. the last iteration $n = 100$). In Table 13, we display these errors defined such as $\mathrm{ErrS}^1 = \|c_P^{\mathcal{T},1} - c_{P,\mathrm{S}}^{\mathcal{T},1}\|_{\mathcal{T},2}$ for $n = 1$ and $\mathrm{ErrS}^{100} = \|c_P^{\mathcal{T},100} - c_{P,\mathrm{S}}^{\mathcal{T},100}\|_{\mathcal{T},2}$ for $n = 100$, respectively.

Cartesian	Errors between Scilab code and LC-solver	
h	ErrS^1	ErrS^{100}
4.692E−1	1E−14	1E−14

TABLE 13. The $L^2(\Omega)$ error between $c_{P,\mathrm{S}}^{\mathcal{T},1}$ and $c_P^{\mathcal{T},1}$ (resp. $c_{P,\mathrm{S}}^{\mathcal{T},100}$ and $c_P^{\mathcal{T},100}$) for **Test 1** (2.15)-(2.16) on the cartesian mesh $h = 4.692 \times 10^{-1}$, at $n = 1$ (resp. $n = 100$).

As one may remark, we obtain errors exactly equal to zero (machine error 1E−14) for both iterations. These results demonstrate that the LC-solver (2.9)-(2.12) of the system of equations (2.1)-(2.2) behaves correctly for this particular cartesian mesh.

The Scilab code yields results that align with the implementation of the LC-solver and its associated solutions. However, it is essential to note that this code serves primarily as a debugging tool, intended for performance on the simplest cartesian mesh. On another note, we cannot totally guarantee the efficiency of the LC-solver method across all mesh types. Nevertheless, based on the comprehensive testing performed throughout this chapter, our implementation appears coherent.

At the end, our analysis of these outcomes prompts a relevant question regarding the viability of the so-called "linear" coupling approach when applied to systems of equations characterized by non-linearity.

Conclusion

In conclusion, this chapter has delved deeply into the complexities of the initial coupling approach, referred to as "linear," applied to the challenging Poisson-Nernst-Planck (PNP) system.

Our investigation began by introducing the coupling method involving two key equations: the Nernst-Planck equation for c_P and the Poisson equation for V. This one ionic species system embodied by c_P, was a deliberate choice. Indeed, the inherent complexities and potential issues associated with the coupling process, bring out a substantial risk of numerical non-convergence. We embarked on a comprehensive exploration of this coupling method through various numerical tests conducted post-coupling.

These tests not only provided valuable insights into the coupling's strengths but also uncovered its profound limitations when confronted with the PNP system. Thereafter, our LC-solver implementations encountered a numerical issue: after a certain duration of simulation time, our solutions began to non-converge. We examined this behavior by identifying the moment in time when discrete solutions began to deviate from accurate approximations. Indeed, the initial iteration seemed to progress correctly, achieving a convergence order of 2 for the approximated solutions, as expected with the DDFV method. However, from the fifth iteration, non-convergence became apparent, yielding inconsistent convergence orders. Subsequently, we adjusted the temporal parameters to obtain consistent results with a short final simulation time and to identify the non-convergence issue. We also investigated the implementation of the Jacobian matrix in the Newton resolution. To enrich our analysis, we incorporated results from a Scilab code, which effectively performed as a numerical tool for our coupling methodology. The Scilab code mirrored our coupling process on a simplest Cartesian mesh, revealing that our numerical implementation of the LC-solver effectively couples the equations of the system, as evidenced by the results on this simplified mesh. All these investigations reinforced the coherence of our LC-solver implementation.

In light of these findings, we draw the conclusion that the LC-solver method, while initially promising at first iterations, fails in its suitability for tackling the complexities of the PNP system. This realization sets the stage for the subsequent introduction of a "non-linear" coupling method in the next chapter, motivated by the limitations encountered in the linear approach, with the aim of obtaining highly accurate numerical solutions for the PNP system.

CHAPTER 3

Nonlinear coupling of the Poisson-Nernst Planck system of equations using the DDFV method, application to ionic and voltage dynamics in neuronal compartments

This chapter is under review at *Journal of Computational Physics* (JCP). A preprint version is available at https://cnrs.hal.science/hal-04385924

Introduction

Abstract. The Poisson-Nernst-Planck system of equations is widely recognized as the standard model for characterizing the electrodiffusion of ions in electrolytes, including ionic dynamics in the cellular cytosol. This non-linear system presents challenges for both modeling and simulations, due to the presence of a stiff boundary layer tightly related to the choice of boundary conditions. In this article, we propose a numerical scheme based on the Discrete Duality Finite Volumes method (DDFV) to solve the Poisson-Nernst-Planck system of equations, while preserving the positivity of ionic concentrations. The DDFV method can use general meshes as local mesh refinement, for instance, at the boundary layer. Through several simulations, we illustrate the accuracy of our scheme, achieving second-order accuracy in space. Furthermore, using a specific test case, we show that our method can resolve steep gradients, such as the ones in boundary layer. Finally, we apply our scheme to several geometries inspired by neuronal shapes. We investigate the propagation and attenuation of an ionic influx in small neuronal compartments of the dendritic tree: a branch bifurcation and a dendritic spine - the mushroom-like protrusions that receive neuronal inputs. We show that a two-dimensional model is required to capture ionic and voltage dynamics in the bulbous head of spines. Considering the connection of our neuronal compartments to an ionic reservoir, that could be the dendritic shaft, we observe that depending on the distance to the closest ionic reservoir, a spine can either act as an isolated compartment, or be subject to signal invasion. This result means that the local geometry of the dendritic tree has a major influence on spine behavior, making plasticity being not only at the level of the spine, but at the level of the full dendritic geometry.

Context. In this chapter, we address a new strategy to couple the PNP system of equations. We delve into a non-linear coupling of the PNP system using the DDFV method, along with an Euler implicit scheme. Again, we make use of a Newton algorithm at each time step to deal with the nonlinearity of the system. Thus, we ensure the positivity of ionic concentrations using a positivity-preserving DDFV scheme. In this context, to facilitate readability, we recall the studied system of equations. We therefore consider the case of two ionic species P and N, with respective valences $+1$ and -1, and concentration c_P and c_N. The PNP system of equation on $\Omega \times [0, T_f]$ writes as:

$$\begin{aligned} \partial_t c_P &= \nabla \cdot (D_P c_P \nabla (\log c_P + \beta V)) + f_{c_P}, \\ \partial_t c_N &= \nabla \cdot (D_N c_N \nabla (\log c_N - \beta V)) + f_{c_N}, \\ -\nabla \cdot (\gamma \beta \nabla V) &= c_P - c_N + f_V. \end{aligned} \tag{3.1}$$

The chapter is organized as follows: Section 3.1 provides a detailed description of the numerical scheme corresponding to (3.1) above. Section 3.2 presents the numerical simulations and convergence

results. Section 3.3 focuses on applications in neuroscience. Finally, the discussion and conclusion are presented in the last section.

3.1. DDFV scheme for the PNP system

In this section, we follow the notations and definitions introduced in Preamble and Chapter 1. We provide the numerical counterpart of the theoretical system of equations (0.8)-(0.11). We give the DDFV scheme in which the source terms are discretized as follows:

$$f_{c_P}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} f_{c_P}, \quad f_{c_N}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} f_{c_N}, \quad f_V^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} f_V,$$

and the initial conditions such as:

$$c_P^{\mathcal{T},0} = \mathbb{P}_m^{\mathcal{T},0} c_P^0, \quad c_N^{\mathcal{T},0} = \mathbb{P}_m^{\mathcal{T},0} c_N^0, \quad V^{\mathcal{T},0} = \mathbb{P}_m^{\mathcal{T},0} V^0.$$

Next, we define the discretization for the Dirichlet values:

$$c_{P,Dir}^{\partial\mathfrak{M}_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt} c_P^{Dir}, \quad c_{N,Dir}^{\partial\mathfrak{M}_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt} c_N^{Dir}, \quad V_{Dir}^{\partial\mathfrak{M}_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}_{Dir},dt} V^{Dir},$$
$$c_{P,Dir}^{\partial\mathfrak{M}^*_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt} c_P^{Dir}, \quad c_{N,Dir}^{\partial\mathfrak{M}^*_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt} c_N^{Dir}, \quad V_{Dir}^{\partial\mathfrak{M}^*_{Dir},dt} = \mathbb{P}_m^{\partial\mathfrak{M}^*_{Dir},dt} V^{Dir}.$$

We define the discretization for the Neumann values for all g:

$$g_{Neu}^{\mathfrak{D}_{ext},dt} = \left((g^{\sigma,n})_{\sigma\in\mathfrak{D}_{ext,Neu}} \right)_{n\in\{0,\dots,N\}} \quad \text{with} \quad g^{\sigma,n} = \int_\sigma g(t_n,\mathbf{x})\mathrm{d}\mathbf{x}.$$

We now present the numerical scheme for the PNP system (0.8)-(0.11). In order to ensure stability, we adopt an implicit Euler scheme for time discretization. At each time step, we employ the Newton method to solve the nonlinear system of coupled equations. By integrating equations (0.8)-(0.11) over $\mathfrak{M}$ and $\mathfrak{M}^* \cup \partial\mathfrak{M}^*_{Neu}$, we find the solutions $(c_P^{\mathcal{T},dt}, c_N^{\mathcal{T},dt}, V^{\mathcal{T},dt}) \in \mathbb{R}^{\mathcal{T},dt}$ that satisfy the following nonlinear problem for all $n \geq 0$:

$$\frac{c_P^{K,n+1} - c_P^{K,n}}{dt} + \mathrm{div}^K\left(\mathbf{J}_{c_P}^{\mathfrak{D},n+1}\right) = f_{c_P}^{K,n+1}, \qquad \forall K \in \mathfrak{M}, \tag{3.2}$$

$$\frac{c_P^{K^*,n+1} - c_P^{K^*,n}}{dt} + \mathrm{div}^{K^*}\left(\mathbf{J}_{c_P}^{\mathfrak{D},n+1}\right) = f_{c_P}^{K^*,n+1}, \qquad \forall K^* \in \mathfrak{M}^* \cup \partial\mathfrak{M}^*_{Neu}, \tag{3.3}$$

$$\frac{c_N^{K,n+1} - c_N^{K,n}}{dt} + \mathrm{div}^K\left(\mathbf{J}_{c_N}^{\mathfrak{D},n+1}\right) = f_{c_N}^{K,n+1}, \qquad \forall K \in \mathfrak{M}, \tag{3.4}$$

$$\frac{c_N^{K^*,n+1} - c_N^{K^*,n}}{dt} + \mathrm{div}^{K^*}\left(\mathbf{J}_{c_N}^{\mathfrak{D},n+1}\right) = f_{c_N}^{K^*,n+1}, \qquad \forall K^* \in \mathfrak{M}^* \cup \partial\mathfrak{M}^*_{Neu}, \tag{3.5}$$

$$-\mathrm{div}^K(\gamma\beta\nabla^{\mathfrak{D}} V^{K,n+1}) + c_N^{K,n+1} - c_P^{K,n+1} = f_V^{K,n+1}, \qquad \forall K \in \mathfrak{M}, \tag{3.6}$$

$$-\mathrm{div}^{K^*}(\gamma\beta\nabla^{\mathfrak{D}} V^{K^*,n+1}) + c_N^{K^*,n+1} - c_P^{K^*,n+1} = f_V^{K^*,n+1}, \qquad \forall K^* \in \mathfrak{M}^* \cup \partial\mathfrak{M}^*_{Neu}, \tag{3.7}$$

$$\mathbf{J}_{c_P}^{\mathfrak{D},n+1} = -D_P r^{\mathfrak{D}}(c_P^{\mathcal{T},n+1})\nabla^{\mathfrak{D}}\left(\log c_P^{\mathcal{T},n+1} + \beta V^{\mathcal{T},n+1}\right), \tag{3.8}$$

$$\mathbf{J}_{c_N}^{\mathfrak{D},n+1} = -D_N r^{\mathfrak{D}}(c_N^{\mathcal{T},n+1})\nabla^{\mathfrak{D}}\left(\log c_N^{\mathcal{T},n+1} - \beta V^{\mathcal{T},n+1}\right). \tag{3.9}$$

The discrete mixed boundary conditions can be expressed as:

(3.10)
$$V^{K,n+1} = V_{Dir}^{K,n+1}, \quad c_P^{K,n+1} = c_{P,Dir}^{K,n+1}, \quad c_N^{K,n+1} = c_{N,Dir}^{K,n+1}, \qquad \forall K \in \partial\mathfrak{M}_{Dir}$$

(3.11)
$$V^{K^*,n+1} = V_{Dir}^{K^*,n+1}, \quad c_P^{K^*,n+1} = c_{P,Dir}^{K^*,n+1}, \quad c_N^{K^*,n+1} = c_{N,Dir}^{K^*,n+1}, \qquad \forall K^* \in \partial\mathfrak{M}_{Dir}^*$$

(3.12)
$$|\sigma|\nabla^{\mathcal{D}} V^{\mathcal{T},n+1} \cdot \mathbf{n}_{\sigma K} = 0, \quad |\sigma|\mathbf{J}_{c_P}^{\mathcal{D},n+1} \cdot \mathbf{n}_{\sigma K} = g^{\sigma,n+1}, \quad |\sigma|\mathbf{J}_{c_N}^{\mathcal{D},n+1} \cdot \mathbf{n}_{\sigma K} = 0, \qquad \forall \mathcal{D} \in \mathfrak{D}_{ext,Neu}.$$

The existence of the discrete solution of (3.2)-(3.12) is an oncoming work, that is not the focus of the current paper. It requires a paper in itself, dedicated to the study of existence, positivity of solution and convergence, inspired by [75].

3.2. Performance of the DDFV scheme

In this section, we present numerical simulations to test the performance of our DDFV scheme (3.2)-(3.12). In subsection 3.2.1, we evaluate the accuracy of the DDFV scheme by considering two test cases with known exact solutions of the PNP system (0.8)-(0.11). In the first test case, we compare our numerical results with an analytical solution of the PNP equations. In the second one, we compare our numerical results with existing analytical and numerical results coming from [66]. In subsection 3.2.2, we show the numerical behavior of our DDFV results in the presence of boundary layer, using a test case inspired by the one-dimensional study in [81]. The coefficients β, D_P and D_N are set to 1 in all the section.

3.2.1. **Convergence results.** To evaluate the accuracy of our DDFV scheme (3.2)-(3.12), we compare it with exact solutions at various mesh resolutions, using the discrete norm $L^\infty((0,T_f);L^2(\Omega))$, $\forall u^{\mathcal{T},dt} \in \mathbb{R}^{\mathcal{T},dt}$:

$$\|u^{\mathcal{T},dt}\|_{\mathcal{T},\infty} = \max_{n\in\{0,\dots,N\}} \left(\frac{1}{2} \sum_{K\in\mathfrak{M}} |K||u^{K,n}|^2 + \frac{1}{2} \sum_{K^*\in\overline{\mathfrak{M}^*}} |K^*||u^{K^*,n}|^2 \right)^{\frac{1}{2}}.$$

The computational domain Ω is set to $]0,1[^2$ and the final time $T_f = 1$. We consider only Dirichlet boundary conditions on $\partial\Omega$, i.e., $\Gamma_{Neu} = \emptyset$, and we set $\gamma = 1$. For the first test case, we consider $c_{P,\text{ex}}$, $c_{N,\text{ex}}$ and V_{ex}, the exact solution of (0.6), defined as follows:

(3.13)
$$\begin{cases} c_{P,\text{ex}}(t,x,y) &= 7x + 5 + 3t^2, \\ c_{N,\text{ex}}(t,x,y) &= x + 1 + t^2, \\ V_{\text{ex}}(t,x,y) &= -x^3 - 2y^2 - t^2x^2. \end{cases}$$

with boundary conditions, initial conditions, and source terms being:

$$\begin{cases} f_{c_P}(t,x,y) &= 6t^4 + (46x+22)t^2 + 6t + 63x^2 + 58x + 20, \\ f_{c_N}(t,x,y) &= -2t^4 + (-10x-6)t^2 + 2t - 9x^2 - 10x - 4, \\ f_V(t,x,y) &= 0. \end{cases}$$

The projection on $\mathbb{R}^{\mathcal{T},dt}$ of these exact solutions, $\{c_{P,\text{ex}}^{\mathcal{T},dt}, c_{N,\text{ex}}^{\mathcal{T},dt}, V_{\text{ex}}^{\mathcal{T},dt}\}$ are defined by:

$$c_{P,\text{ex}}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} c_{P,\text{ex}}, \quad c_{N,\text{ex}}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} c_{N,\text{ex}}, \quad V_{\text{ex}}^{\mathcal{T},dt} = \mathbb{P}_m^{\mathcal{T},dt} V_{\text{ex}}.$$

We simulate the system (3.2)-(3.12) on several increasingly refined meshes generated by the quadrangle mesh family. One example mesh is depicted in Figure 1, with a mesh size $h = 2.341\text{E}{-1}$. The respective mesh size h for each mesh ranges from $4.692\text{E}{-1}$ to $6.253\text{E}{-2}$. The time step dt is initially set to

10^{-2} for the mesh with the largest $h = 4.692\text{E}{-1}$. For subsequent meshes, the time step is divided by 4. Recall that at each time step, we run a Newton algorithm to solve our non-linear problem. This Newton algorithm converges, with on average 2 to 1 iterations depending on the mesh refinement. The maximal number of iterations for all the time steps over all the meshes is 3.

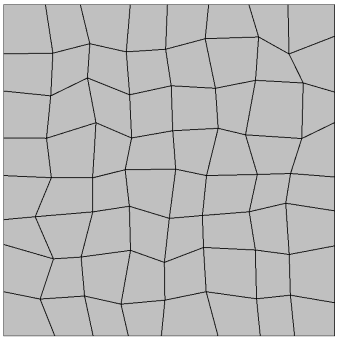

FIGURE 1. Example of a quadrangle mesh with $h = 2.341\text{E}{-1}$.

h	dt	e_{c_P}	Order	e_{c_N}	Order	e_V	Order
4.692E−1	1.0E − 2	2.593E−3	·	1.443E−3	·	2.070E−2	·
2.341E−1	2.5E − 3	8.109E−4	1.67	4.713E−4	1.61	5.137E−3	2.00
1.212E−1	6.25E − 4	2.066E−4	2.08	1.220E−4	2.05	1.312E−3	2.07
6.253E−2	1.56E − 4	4.897E−5	2.18	2.843E−5	2.20	3.182E−4	2.14

TABLE 1. The errors e_{c_P}, e_{c_N} and e_V, as well as the convergence order for the first test case (3.13) on the quadrangle mesh family.

To illustrate the convergence of our method, we define the error according to the norm $L^\infty((0,T_f);L^2(\Omega))$, between a numerical vector y and the corresponding exact solution y_{ex}:

$$e_y = \|y^{\mathcal{T},dt} - y_{\text{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},\infty}. \tag{3.14}$$

The errors e_{c_P}, e_{c_N}, and e_V, for the different meshes, as well as the convergence order obtained numerically are given in Table 1. For both the concentrations c_P and c_N and the electrostatic potential V, we observe a convergence of order two.

We then investigate the convergence and robustness of our method by comparing our results with a test case proposed in [**66**]. In this paper, the authors introduced a finite difference scheme for approximating the PNP solutions in 2D while preserving the positivity of the ionic concentrations. The computational domain Ω is set to $]-1,1[^2$, the final time $T_f = 0.1$ and $\gamma = 1$. The boundary conditions are all Dirichlet (i.e., $\Gamma_{Neu} = \emptyset$). The exact solutions of the second test case are defined as follows:

$$\begin{cases} c_{P,\text{ex}}(t,x,y) &= e^{-t}\cos(2\pi x)\sin(2\pi y) + 2, \\ c_{N,\text{ex}}(t,x,y) &= e^{-t}\sin(2\pi x)\cos(2\pi y) + 2, \\ V_{\text{ex}}(t,x,y) &= e^{-t}\sin(2\pi x)\sin(2\pi y). \end{cases} \tag{3.15}$$

The boundary and the initial conditions, as well as the source terms are:

$$\begin{cases} f_{c_P}(t,x,y) &= -16\cos(2\pi x)e^{-t}\left(e^{-t}\pi^2\left(\cos(2\pi y)^2-\frac{3}{4}\right)\sin(2\pi x)+\sin(2\pi y)\left(-\frac{\pi^2}{2}+\frac{1}{16}\right)\right) \\ &\quad +16\pi^2 e^{-t}\sin(2\pi x)\sin(2\pi y), \\ f_{c_N}(t,x,y) &= \cos(2\pi y)(8\pi^2-1)e^{-t}\sin(2\pi x)+16\left(\cos(2\pi x)^2-\frac{3}{4}\right)\pi^2 e^{-2t}\sin(2\pi y)\cos(2\pi y) \\ &\quad -16\pi^2 e^{-t}\sin(2\pi x)\sin(2\pi y), \\ f_V(t,x,y) &= e^{-t}\left(8\pi^2\sin(2\pi y)+\cos(2\pi y)\right)\sin(2\pi x)-e^{-t}\cos(2\pi x)\sin(2\pi y). \end{cases}$$

We run simulations using the same family of Cartesian meshes and the same time step $dt = h^2$ as in [66]. At each time step, our Newton algorithm converges with a maximum of 3 iterations. The average number of iterations is between 1 and 2, depending on the mesh refinement.

h	dt	e_{c_P}	Order	e_{c_N}	Order	e_V	Order
1E−1	1.0E−2	4.086E−3	·	4.077E−3	·	5.453E−3	·
5E−2	2.5E−3	1.038E−3	1.98	1.036E−3	1.98	1.372E−3	1.99
2.5E−2	6.25E−4	2.606E−4	1.99	2.600E−4	1.99	3.436E−4	2.00
1.25E−2	1.56E−4	6.519E−5	2.00	6.504E−5	2.00	8.593E−5	2.00

TABLE 2. The errors e_{c_P}, e_{c_N} and e_V, as well as the convergence order for the second test case (3.15), using Cartesian meshes.

In Table 2, we give the value of the errors e_{c_P}, e_{c_N}, e_V (eq. (3.14)). We obtain again a convergence of order 2 for c_P, c_N and V. For each mesh size, the $L^\infty((0,T_f);L^2(\Omega))$ errors - between the numerical solutions of (3.2)-(3.12) and the exact solution - obtained from our DDFV scheme is nearly 10 times smaller than the one obtained in [66]. As an example, we obtain a value for e_{c_P} of 6.519E−5 to compare with a value of 3.093E−4 in [66] ($h = 1.25$E−2). Put differently, to obtain an error with the same order of magnitude, our DDFV scheme requires a mesh with half the number of cells than the finite difference scheme proposed in [66], which significantly reduces computation costs.

3.2.2. **Comparative analysis of the DDFV scheme at the boundary layer.** We consider in this part a test case with a boundary layer initially presented in [81]. This test case is defined on the one-dimensional domain $]0,1[$, with the following boundary conditions:

$$c_P(t,0) = 1+t, \quad c_N(t,0) = 1, \quad V(t,0) = 0,$$
$$c_P(t,1) = 1, \quad c_N(t,1) = 1+t, \quad V(t,1) = 0.$$

This system has two boundary layer regions, $]0,0.01[$ and $]0.99,1[$. In [81], the authors propose a finite difference scheme preserving the positivity of the concentration. They adopt a mesh size $\Delta x = 8\times10^{-4}$ in the boundary layer region, and a larger one $\Delta x = 3\times10^{-3}$ in the bulk region $]0.01,0.99[$.

We build our third test case by adapting this 1D test case to our 2D framework. Our computational domain Ω is $]0,1[^2$, and we adapt the Dirichlet boundary conditions so that they also depend on the space variable:

$$c_P^{Dir}(t,x,y) = 1+(1-x)t, \quad c_N^{Dir}(t,x,y) = 1+xt, \quad V^{Dir}(t,x,y) = 0. \tag{3.16}$$

We set the source terms of the system (0.8) to zero. The initial conditions are computed using eq. (3.16) at $t = 0$. The final time $T_f = 1$, with a time step $dt = 10^{-2}$ and $\gamma = 0.01$. We use a fixed non-uniform mesh with a mesh size in the x-direction of $\Delta x = 10^{-3}$ in the boundary layer region, namely $]0,0.01[\times]0,1[$ and $]0.99,1[\times]0,1[$, gradually increasing to $\Delta x = 10^{-2}$ in the bulk region $]0.01,0.99[\times]0,1[$. For all cells, we set the mesh size in the y-direction equal to $\Delta y = 10^{-2}$. The mesh has 11800 quadrilateral cells (rectangles). At each time step, the Newton algorithm converges with a maximum of 2 iterations.

In Figure 2, we plot on the left (**A-C-E**) the dual values of the concentrations c_P and c_N and the electrostatic potential V at final time $T_f = 1$, that is $(c_P^{K^*,100})_{K^*\in\overline{\mathfrak{M}^*}}$, $(c_N^{K^*,100})_{K^*\in\overline{\mathfrak{M}^*}}$ and

$(V^{K^*,100})_{K^* \in \overline{\mathfrak{M}^*}}$. On the right (**B-D-F**), we plot a zoom of c_P, c_N and V at final time $T_f = 1$, on the boundary layer region $[0, 0.05] \times [0.47, 0.52]$, where we observe significant gradients.

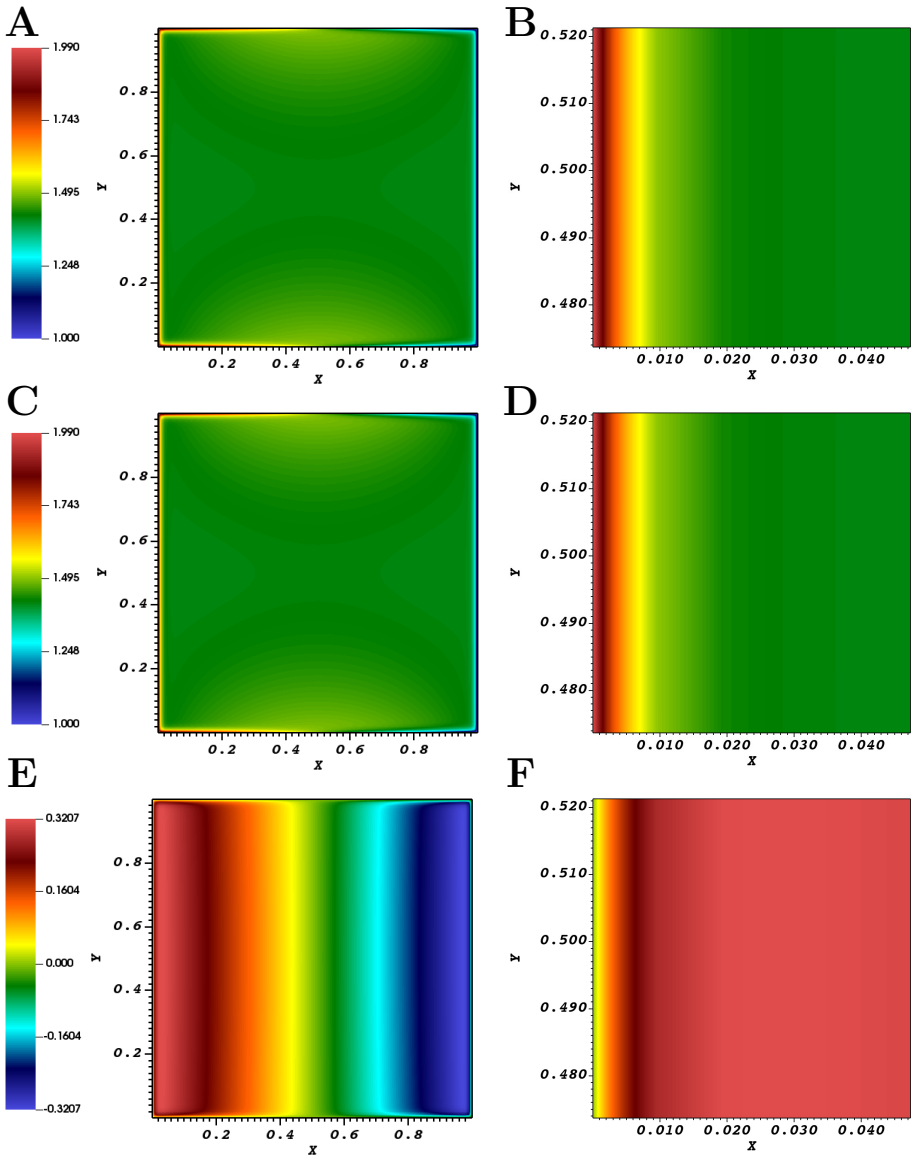

FIGURE 2. Simulation results of the third test case adapted from [81] at $T_f = 1$. **A**: $(c_P^{K^*,100})_{K^* \in \overline{\mathfrak{M}^*}}$. **C**: $(c_N^{K^*,100})_{K^* \in \overline{\mathfrak{M}^*}}$. **E**: $(V^{K^*,100})_{K^* \in \overline{\mathfrak{M}^*}}$. Zoom on the boundary layer region $[0, 0.05] \times [0.47, 0.52]$ for c_P in **B**, c_N in **D** and V in **F**.

We plot in Figure 3 the dual discrete concentrations $(c_P^{K^*,50})_{K^* \in Z^*_{y=0.5}}$ (resp. $(c_P^{K^*,100})_{K^* \in Z^*_{y=0.5}}$) and $(c_N^{K^*,50})_{K^* \in Z^*_{y=0.5}}$ (resp. $(c_N^{K^*,100})_{K^* \in Z^*_{y=0.5}}$) and the dual discrete electrostatic potential $(V^{K^*,50})_{K^* \in Z^*_{y=0.5}}$ (resp. $(V^{K^*,100})_{K^* \in Z^*_{y=0.5}}$) at $t = 0.5$ (resp. $t = 1$) with a linear trend of the potential. We recall that the definition of $Z^*_{y=0.5}$ is given by eq. (1.1). This figure can be compared to Fig. 2 in [81]. We obtain similar behavior and significant gradients near the walls. Additionally, our DDFV scheme takes approximately 30 minutes to solve on a laptop (CPU: Intel(R) Xeon(R) Gold 6254 CPU @ 3.10GHz; MEMORY: DIMM DDR4 Synchronous Registered (Buffered) 2933 MHz (0.3 ns)), compared to 1 hour in [81].

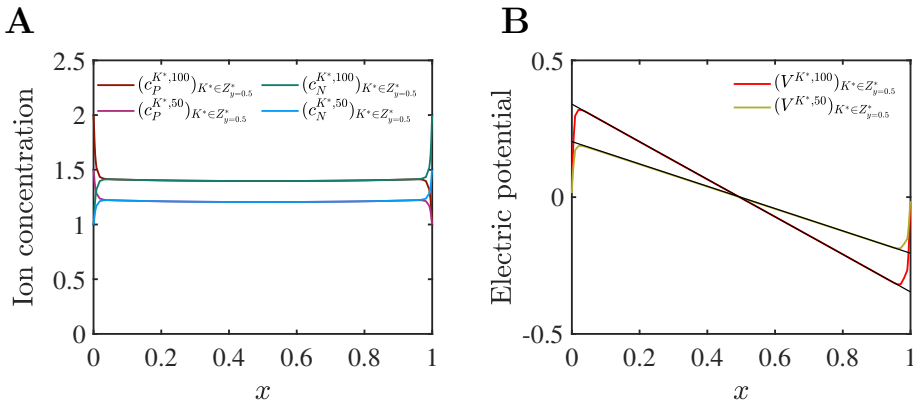

FIGURE 3. **A**: Dual discrete values $(c_P^{K^*,50})_{K^* \in Z^*_{y=0.5}}$, $(c_N^{K^*,50})_{K^* \in Z^*_{y=0.5}}$, $(c_P^{K^*,100})_{K^* \in Z^*_{y=0.5}}$ and $(c_N^{K^*,100})_{K^* \in Z^*_{y=0.5}}$ at $t = 0.5$ and $t = 1$. **B**: Dual discrete values $(V^{K^*,50})_{K^* \in Z^*_{y=0.5}}$, and $(V^{K^*,100})_{K^* \in Z^*_{y=0.5}}$ at $t = 0.5$ and $t = 1$. Black curves represent linear trends of the potential V in the bulk.

We observe that outside of the boundary layer, the ionic concentration c_P and c_N are very close, which corresponds to an electroneutral bulk. We compute the absolute value of this difference at time $t = 0.5$: $q_{K^* \in \overline{\mathfrak{M}^*}} := (|c_P^{K^*,50} - c_N^{K^*,50}|)_{K^* \in \overline{\mathfrak{M}^*}}$. In [81], the authors reported a maximum value for this difference of 3.6×10^{-6} in the interval $[0.25, 0.75]$ at $t = 0.5$. We plot the values for $q_{K^* \in \overline{\mathfrak{M}^*}}$ cut at 3.6×10^{-6}, and color the regions where $q > 3.6 \times 10^{-6}$ in black (Fig. 4**A**). We observe that in our simulations, the x-boundary of the region such that $q_{K^* \in \overline{\mathfrak{M}^*}} \leq 3.6 \times 10^{-6}$ is $[0.075, 0.925]$, which is a larger region than the segment $[0.25, 0.75]$ in [81]. Note that the maximum for $q_{K^* \in Z^*_{[0.25,0.75] \times \{0.5\}}}$ on $[0.25, 0.75] \times \{0.5\}$ is 1.5×10^{-6}, which is half the value 3.6×10^{-6} observed in [81]. In Fig. 4**B**, we plot the values of $q_{K^* \in Z^*_{[0.075,0.925] \times \{0.5\}}}$ (see (1.1)). The error is decreasing from the boundaries ($x = 0$ and $x = 1$) to the center $x = 0.5$. At $x = 0.5$, the error is reaching the machine error 10^{-14}.

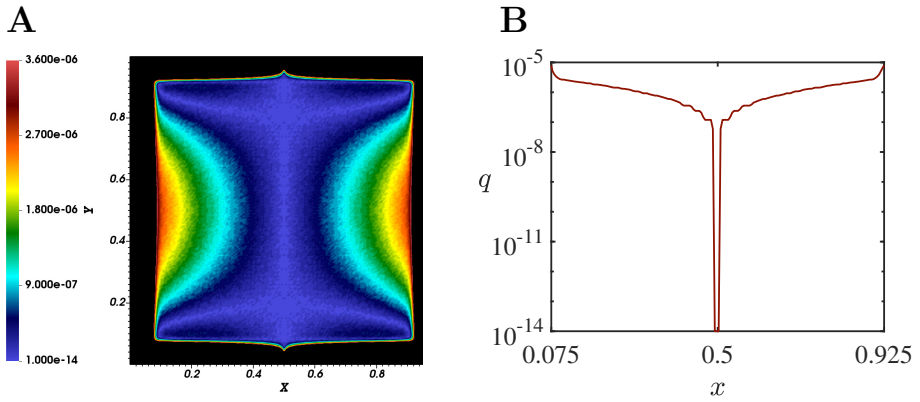

FIGURE 4. **A**: Absolute difference between c_P and c_N at time $t = 0.5$, $q_{K^* \in \overline{\mathfrak{M}^*}} = (|c_P^{K^*,50} - c_N^{K^*,50}|)_{K^* \in \overline{\mathfrak{M}^*}}$. $q_{K^* \in \overline{\mathfrak{M}^*}}$ goes from 10^{-14} to 3.6×10^{-6}. Regions where $q_{K^* \in \overline{\mathfrak{M}^*}} > 3.6 \times 10^{-6}$ are colored in black. **B**: Plot of $q_{K^* \in Z^*_{[0.25,0.75] \times \{0.5\}}}$ on the segment $[0.075, 0.925] \times \{0.5\}$, in logarithmic scale.

We show with this test case that our DDFV scheme can capture well the boundary layer dynamics. Compared to [81], our electroneutral zone is larger, suggesting that the perturbation of the numerical results due to the high gradients inside the boundary layer, is less pronounced with our scheme.

3.3. Applications to neuroscience

In the last section, we apply our DDFV scheme to investigate the dynamics of voltage and ionic concentration in different neuronal geometries. We consider two specific geometries in 2D: a bifurcation

in the dendritic tree, and a dendritic spine. The dendritic tree has a tree-like geometry, and a bifurcation corresponds to the location where one branch divides into two branches (Fig. 5). Dendritic spines are mushroom-like protrusions of a few micrometers, found on the dendritic tree. In section 3.3.1, we consider a dendrite bifurcation with two thin branches connecting to a larger one and simulate the propagation of an influx of ions arriving at the edge of one of the thin branches. We investigate three scenarios: the influence of an ionic reservoir on voltage and ionic concentration dynamics, the summation of two simultaneous influx of ions and the propagation of a single influx in the rest of the domain. Section 3.3.2 focuses on dendritic spines. We first investigate the influence of the spine head geometry on voltage and ionic concentration dynamics. We then consider two neighboring spines and investigate the invasion of an influx of ions arriving in one spine on the other. We recall here the definitions of the initial and boundary conditions given in section 0.9 (values are given in Table 3):

$$c_P(t=0,\mathbf{x}) = c_P^0, \quad c_N(t=0,\mathbf{x}) = c_N^0, \quad V(t=0,\mathbf{x}) = V^0 \quad \forall \mathbf{x} \in \Omega. \tag{3.17}$$

We impose constant Dirichlet boundary condition on Γ_{Dir}:

$$c_P = c_P^0, \quad c_N = c_N^0, \quad V = V^0, \qquad \text{on } \Gamma_{Dir} \times (0, T_f).$$

We consider $\partial\Omega_r$ for homogenous and $\partial\Omega_i$ for non homogeneous Neumann boundary conditions, such that $\Gamma_{Neu} = \partial\Omega_r \cup \partial\Omega_i$. The homogeneous Neumann boundary condition on $\partial\Omega_r$ models the impermeability of the neuronal membrane to ions. The non-homogeneous boundary condition on $\partial\Omega_i$ models the influx of ions I received by the dendrite, at specific locations called synapses:

$$\begin{aligned} &\nabla V \cdot \mathbf{n} = 0, && \text{on } \partial\Omega_i \cup \partial\Omega_r \quad \times(0,T_f),\\ &\nabla c_N \cdot \mathbf{n} = 0, && \text{on } \partial\Omega_i \cup \partial\Omega_r \quad \times(0,T_f),\\ &\nabla c_P \cdot \mathbf{n} = 0, && \text{on } \partial\Omega_r \quad \times(0,T_f),\\ &\nabla c_P \cdot \mathbf{n} = I, && \text{on } \partial\Omega_i \quad \times(0,T_f). \end{aligned} \tag{3.18}$$

$I(t) = \dfrac{I_{stim}(t)}{\pi r_i^2 F D_P}$, where

$$I_{stim}(t) = I_{max} \frac{t}{\tau} \exp\left(-\frac{t}{\tau} + 1\right) \tag{3.19}$$

represents the injected synaptic current, inspired by [21]. Note that in this section, the source terms f_{c_p}, f_{c_n} and f_V are set to zero. We define a time t_0 for each simulation, corresponding to the time at which the concentration c_P is maximal on Ω. Note that it corresponds to a maximum in both space and time. We precise the value of t_0 in each case. We use electrodiffusion parameters as defined in Table 3. Finally, we present in this section the numerical results for c_P, as the results for c_N are qualitatively similar.

F	96485 A.s.mol^{-1}	Faraday constant	
ε	80	Dielectric permittivity	[21]
ε_0	8,8.10^{-12} F.m^{-1}	Permittivity of vacuum	
T_θ	293,15 K	Absolute temperature	
R	8,314 J.K^{-1}.mol^{-1}	Gas constant	
D_P	200 µm .m^2.s^{-1}	Diffusion coefficient for anion	[21]
D_N	200 µm .m^2.s^{-1}	Diffusion coefficient for cation	[21]
γ	1.8431×10^{-4}	Eq. (0.5)	
β	39.5877	Eq. (0.5)	
c_P^0	163 mM	Initial concentration for species P	[21]
c_N^0	163 mM	Initial concentration for species N	[21]
V^0	0 mV	Initial electric potential	[21]
I_{max}	300 pA	Maximum of the injected current, eq. (3.19)	[21]
τ	0.055 s	Decay time constant of the injected current	[21]

TABLE 3. Electrodiffusion parameters.

3.3.1. **Propagation of an influx of ions at a dendritic tree bifurcation.** In this part, we model signal propagation at a bifurcation in the dendritic tree. We consider a domain Ω_B representing the bifurcation, where a large branch divides into two thinner ones (Fig. 5). We model the influx of ions as a current $I(t)$ (eq. (3.19)) injected at the end of the two thin branches ($\partial\Omega_i^{\text{up}}$ for the upper branch, and $\partial\Omega_i^{\text{down}}$ for the lower branch, Fig. 5, $\partial\Omega_i = \partial\Omega_i^{\text{up}} \cup \partial\Omega_i^{\text{down}}$). We impose Dirichlet boundary conditions at the end of the large branch (Γ_{Dir} on Fig. 5), to represent the connection with a larger dendrite. This large dendrite is considered an ionic reservoir due to its large size, i.e. it has fixed ionic concentrations. The different lengths and nodes defining the domain Ω_B are described in Fig. 5, with the length values and node coordinates given in Table 4.

We define two positions near the end of each branch, namely (I) and (J), to compare the voltage and concentration dynamics in the branches. We use a triangular mesh with 6655 cells and a mesh size $h = 0.36$. The simulations are performed with a final time of $T_f = 1.5$ sec, and a time step of $dt = 5 \times 10^{-3}$ sec.

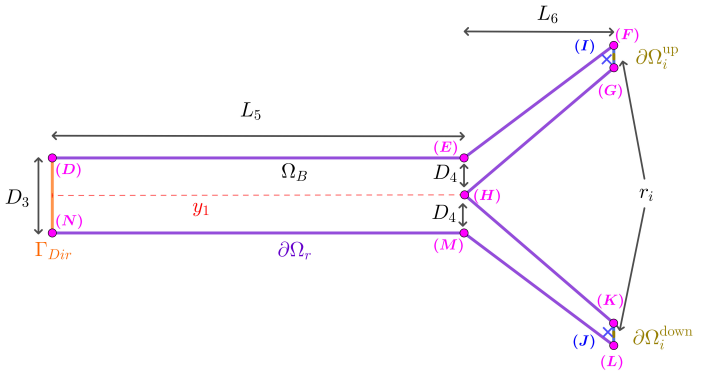

FIGURE 5. Geometry of the bifurcation domain Ω_B, in the configuration $L_5 = 11$ µm . The specific line $y_1 = [0, 11] \times \{1\}$, where we monitor concentration and voltage dynamics is plotted in dashed red. The coordinates of each node are given in Table 4.

L_5	11 µm	Lengths for the sample of the dendrite
L_6	4 µm	Length for both branches
D_3	2 µm	Diameter for the dendrite trunk
D_4	1 µm	Diameter neck for both branches at the junction
r_i	0.12 µm	Radius of $\partial\Omega_i$ in Figure 5 for Ω_B
(D)	(0,2)	Position (x,y) of node D
(E)	(11,2)	Position (x,y) of node E
(F)	(15,5)	Position (x,y) of node F
(G)	(15,4.4)	Position (x,y) of node G
(H)	(11,1)	Position (x,y) of node H
(I)	(14.8,4.6)	Position (x,y) of node I
(J)	(14.8,-2.6)	Position (x,y) of node J
(K)	(15,-2.4)	Position (x,y) of node K
(L)	(15,-3)	Position (x,y) of node L
(M)	(11,0)	Position (x,y) of node M
(N)	(0,0)	Position (x,y) of node N

TABLE 4. Geometric parameters for domain Ω_B, representing a dendritic bifurcation.

3.3.1.1. ***Effect of the distance to an ionic reservoir on voltage and concentration dynamics.*** Our model considers that the end of the large branch is connected with a wider one, called an ionic reservoir, such that the ionic concentrations at Γ_{Dir} are constant. The distance between this ionic reservoir and the bifurcation influences the dynamics of voltage and ionic concentrations everywhere on the domain Ω_B. We expect that the longer the large branch (i.e. the longer L_5, Fig. 5),

the smaller the influence on the dynamics close to the bifurcation point. To measure this influence, we apply a current at the end of the two thin branches ($\partial\Omega_i$ on Fig. 5). We then realize simulations for different values of L_5 ranging from 11 µm to 33 µm (see Fig. 6 for the case $L_5 = 33$ µm). We finally compute the difference between each numerical solution, and consider this difference relative to the peak amplitude. We consider that the influence of the Dirichlet boundary condition on ionic and voltage dynamics is small when this relative difference is below 3.6%.
We realize numerical simulations in domain Ω_B for three different values of L_5: 11 µm , 22 µm and 33 µm . Note that for each value of L_5, the domain is modified, and so is the mesh. We indicate the different coordinates and parameter values that are modified in the three configurations of domain Ω_B in Table 5. To account for the mesh modifications in the discrete solution for the different configurations, we note $c_{P,q}^{\tau,n}$ the discrete value of the concentration c_P in the case $L_5 = q$ µm .

Value of L_5	11 µm	22 µm	33 µm
Number of triangular cells	6655	7279	7863
h	3.6×10^{-1}	3.9×10^{-1}	4×10^{-1}
Position (x,y) of node (D)	(0,2)	(-11,2)	(-22,2)
Position (x,y) of node (N)	(0,0)	(-11,0)	(-22,0)

TABLE 5. Different coordinates and parameter values that are modified in the three configurations of domain Ω_B: $L_5 = 11$ µm , $L_5 = 22$ µm and $L_5 = 33$ µm .

In Figure 6, we plot the dual values $\left(c_{P,33}^{K^*,21} - c_P^0\right)_{K^* \in \overline{\mathfrak{M}^*}}$, corresponding to the concentration $c_P - c_P^0$ at time $t_0 = 0.105$ sec, in the case $L_5 = 33$ µm . We observe that the maximum of the solution, $A = \max_{K^* \in \overline{\mathfrak{M}^*}} \left(c_{P,33}^{K^*,21} - c_P^0\right) = 93.8$ mM, is reached at the injection boundary $\partial\Omega_i$. Along x, the solution decreases to reach the value c_P^0 at the Dirichlet boundary Γ_{Dir}.

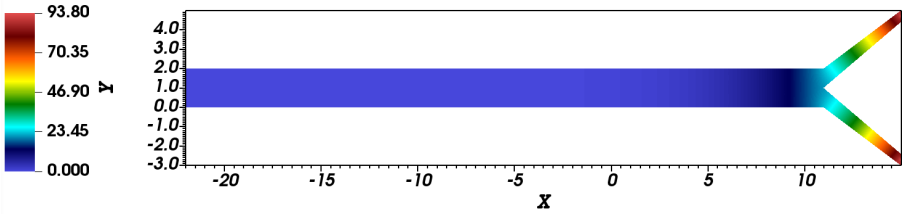

FIGURE 6. Dual values $\left(c_{P,33}^{K^*,21} - c_P^0\right)_{K^* \in \overline{\mathfrak{M}^*}}$ of the concentration $c_P - c_P^0$ at time $t_0 = 0.105$ sec, in the case $L_5 = 33$ µm .

We compare the numerical solutions obtained on different domains, by evaluating their difference on two straight lines $y_1 : [0, 11] \times \{1\}$ (see Fig. 5), and $y_2 : [-11, 11] \times \{1\}$. Note that the line y_2 extends the line y_1 for the configurations $L_5 = 22$ µm and $L_5 = 33$ µm .
We note $\left(e_{p,q}^{K^*,n}\right)_{K^* \in Z_{y_i}^*}$ the absolute value of the difference of the dual values for c_P, on line y_i (see (1.1)), for the domains with $L_5 = p$ and $L_5 = q$, at time iteration n, for $i = 1, 2$:

$$\left(e_{p,q}^{K^*,n}\right)_{K^* \in Z_{y_i}^*} = \left| \left(c_{P,p}^{K^*,n}\right)_{K^* \in Z_{y_i}^*} - \left(c_{P,q}^{K^*,n}\right)_{K^* \in Z_{y_i}^*} \right|. \tag{3.20}$$

Finally, we compute the percentage difference relative to A, between the simulation results on line y_i for $t = 0$ to $t = 0.5$ sec (time steps $n = 0, .., 100$), and plot the result in Fig. 7: $e_1 := \left(\left(\frac{e_{11,22}^{K^*,n}}{A}\right)_{K^* \in Z_{y_1}^*}\right)_{n=0,\dots,100}$ (**A**), $e_2 := \left(\left(\frac{e_{11,33}^{K^*,n}}{A}\right)_{K^* \in Z_{y_1}^*}\right)_{n=0,\dots,100}$ (**B**) and $e_3 := \left(\left(\frac{e_{22,33}^{K^*,n}}{A}\right)_{K^* \in Z_{y_2}^*}\right)_{n=0,\dots,100}$ (**C**).
We observe that the difference between the solutions is always maximal at the Dirichlet boundary

condition, at time $t = 0$, (Fig. 7). The maximal percentage difference between the concentration values c_P for $L_5 = 11$ µm and $L_5 = 22$ µm is 9.1 % (**A**). For $L_5 = 11$ µm and $L_5 = 33$ µm , the maximum percentage difference is 10.4 % (**B**), and finally, for $L_5 = 22$ µm and $L_5 = 33$ µm , to 3.6 % (**C**). Finally, we conclude that as expected, the influence of the Dirichlet boundary condition Γ_{Dir} (i.e. the ionic reservoir) on concentration dynamics c_P, decreases when the length L_5 of the large branch increases. Specifically, for our geometry, the influence is small if the reservoir is at a distance larger than 20 microns.

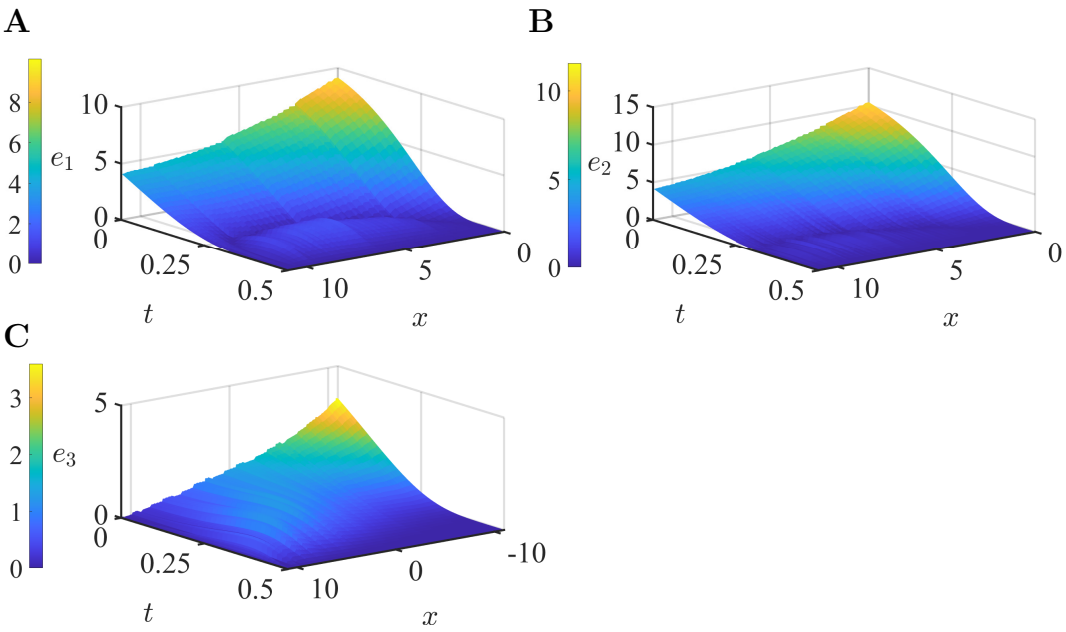

FIGURE 7. Comparison of the numerical solution between the three domains with $L_5 = 11$ µm , $L_5 = 22$ µm and $L_5 = 33$ µm . **A** (resp. **B**, resp. **C**): Plots of e_1 (resp. e_2, resp. e_3) the absolute values of the differences of the dual values for c_P, on line y_1 (resp. y_1, resp. y_2), for the domains with $L_5 = 11$ µm and $L_5 = 22$ µm , (resp. $L_5 = 11$ µm and $L_5 = 33$ µm , resp. $L_5 = 22$ µm and $L_5 = 33$ µm). Note that line y_2 ranges in $[-11, 11]$, so as the x-axis in panel **C**.

3.3.1.2. ***Signal summation.*** We now study signal summation at a bifurcation in the dendritic tree, when the ionic reservoir is far from the bifurcation. We compare ionic concentration and voltage dynamics in two different scenarios: when only the upper branch receives the influx ($\partial\Omega_i^{\text{up}}$ in Fig. 5, scenario **1**, $\partial\Omega_i^{\text{down}}$ is turned into a homogeneous Neumann boundary condition) and when there is an influx of ions at both ends of the two thin branches ($\partial\Omega_i^{\text{up}}$ and $\partial\Omega_i^{\text{down}}$ in Fig. 5, scenario **2**). We will call a branch 'active' if it receives an influx of ions, and 'inactive' if it does not. We run simulations in the domain with $L_5 = 33$ µm (Fig. 5 and Table 5). We consider the line $y_3 = [-22, 11] \times \{1\}$, and the set of dual cells $Z^*_{y_3}$ (see (1.1)). In Fig. 8, we compare the concentration $(c_P^{K^*,21} - c_P^0)^{\mathbf{i}}_{K^* \in Z^*_{y_3}}$ and voltage $(V^{K^*,21})^{\mathbf{i}}_{K^* \in Z^*_{y_3}}$ at time $t_0 = 0.105$ sec, for the two scenarios **1** and **2**. In both graphs, the brown curve represents scenario **1** with one active branch, while the blue curve corresponds to scenario **2** where the two branches are active. We observe that, on line y_3, the concentration and potential values in scenario **2** (blue) are twice the values in scenario **1** (brown).

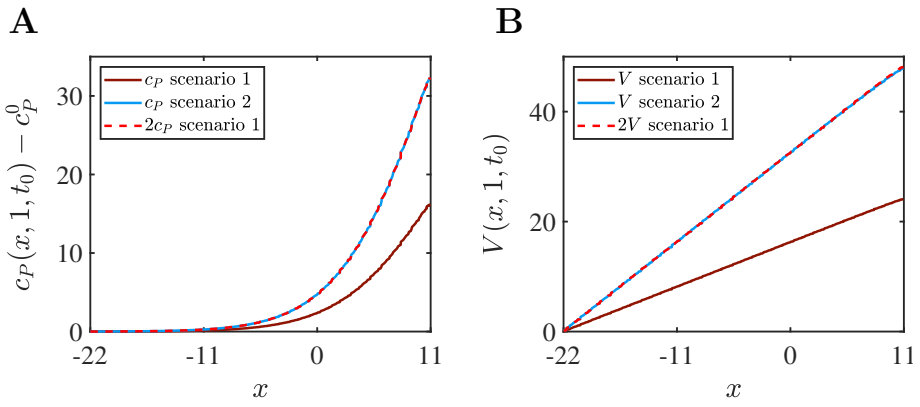

FIGURE 8. Evolution of the concentration and voltage dynamics on line y_3 for the two scenarios **1** and (resp. **2**), with one (resp. two) active branch(es). **A**: Dual values $(c_P^{K^*,21} - c_P^0)^{\mathbf{i}}_{K^* \in Z^*_{y_3}}$, at peak time $t_0 = 0.105$ sec, for $\mathbf{i} = \mathbf{1}$ in brown and $\mathbf{i} = \mathbf{2}$ in blue. In dashed red, we plot $2(c_P^{K^*,21} - c_P^0)^{\mathbf{1}}_{K^* \in Z^*_{y_3}}$. **B**: Dual values $(V^{K^*,21})^{\mathbf{i}}_{K^* \in Z^*_{y_3}}$, at peak time $t_0 = 0.105$ s, for $\mathbf{i} = \mathbf{1}$ in brown and $\mathbf{i} = \mathbf{2}$ in blue. In dashed red, we plot $2(V^{K^*,21})^{\mathbf{1}}_{K^* \in Z^*_{y_3}}$.

In Fig. 9, we plot the time evolution of the concentration $c_P - c_P^0$ and the potential V at position (H), i.e. $(c_P^{K_H^*,n} - c_P^0)^{\mathbf{i}}_{n=0,\cdots,N}$ and $(V^{K_H^*,n})^{\mathbf{i}}_{n=0,\cdots,N}$ for the two scenarios. We recall that the definition of K_H^* is given by (1.2). As in Fig. 8, we observe that the potential and concentration dynamics in scenario **2** are twice the dynamics in scenario **1**.

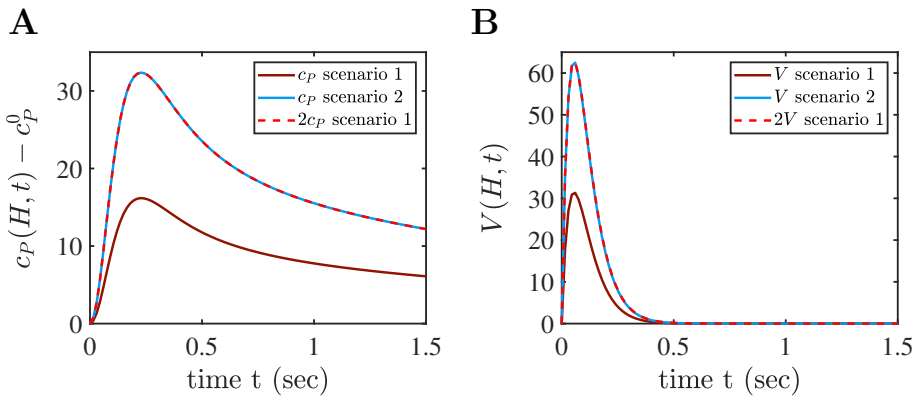

FIGURE 9. Evolution of the concentration and voltage dynamics at point (H) for the two scenarios **1** and **2**. **A**: Dual values $(c_P^{K_H^*,n} - c_P^0)^{\mathbf{i}}_{n=0,\cdots,N}$ for $\mathbf{i} = \mathbf{1}$ in brown and $\mathbf{i} = \mathbf{2}$ in blue. In dashed red, we plot $2(c_P^{K_H^*,n} - c_P^0)^{\mathbf{1}}_{n=0,\cdots,N}$ $\left(\max_{n=0,\cdots,N} \left|(c_P^{K_H^*,n})^{\mathbf{2}} - 2(c_P^{K_H^*,n})^{\mathbf{1}}\right| = 5 \times 10^{-3}\right)$. **B**: Dual values $(V^{K_H^*,n})^{\mathbf{i}}_{n=0,\cdots,N}$ for $\mathbf{i} = \mathbf{1}$ in brown and $\mathbf{i} = \mathbf{2}$ in blue. In dashed red, we plot $2(V^{K_H^*,n})^{\mathbf{1}}_{n=0,\cdots,N}$. We have $\left(\max_{n=0,\cdots,N} \left|(V^{K_H^*,n})^{\mathbf{2}} - 2(V^{K_H^*,n})^{\mathbf{1}}\right| = 0.3\right)$.

We finally compute the absolute difference between the concentration $\left((c_P^{K^*,n} - c_P^0)^{\mathbf{2}}_{K^* \in Z^*_{y_3}}\right)_{n=0,\cdots,N}$ and twice the concentration $\left((c_P^{K^*,n} - c_P^0)^{\mathbf{1}}_{K^* \in Z^*_{y_3}}\right)_{n=0,\cdots,N}$ on line y_3 and for $t \in [0, T_f]$. This difference is zero at the Dirichlet boundary condition (x = - 22 µm , for all t), and stays below 10^{-2} mM up to $x = 8$ µm (resp. 10^{-1} mV up to $x = 8$ µm) for the concentration (resp. the potential). It then steeply increases to 0.33 mM (resp. 0.35 mV) for the concentration (resp. the potential) close

to $x = 11$ µm (bifurcation point (H)). In summary, the signal in the large branch is doubled when two branches are active compared to only one active branch. This result is coherent with a linear summation of the signal in passive dendrites observed experimentally [22].

3.3.1.3. ***Branch invasion.*** We now investigate the invasion of a signal in an inactive branch, by plotting the evolution of the concentration and potential dynamics in the small branches in scenario **1** (only the upper branch is active). We consider the nodes (I) and (J) toward the end of the upper and lower branches respectively, far from the bifurcation point (See Fig. 5 and Table 5). In Fig. 10, we plot the time evolution of the concentrations $(c_P^{K_I^*,n} - c_P^0)^{\mathbf{1}}_{n=0,\cdots,N}$ and $(c_P^{K_J^*,n} - c_P^0)^{\mathbf{1}}_{n=0,\cdots,N}$ and the potential $(V^{K_I^*,n})^{\mathbf{1}}_{n=0,\cdots,N}$ and $(V^{K_J^*,n})^{\mathbf{1}}_{n=0,\cdots,N}$ at the points (I) and (J) (see eq. (1.2)).

We observe that the influx of ions arriving at the upper branch creates a transient rise in the concentration and in the potential that invades the lower branch. We compare the time and amplitude of c_P and the potential at the two positions (I) and (J) and observe that the maximum of the signal arrives in (J) with a delay of 0.21 sec, and an amplitude reduced by 82 %, decreasing from 81.43 mM in node (I) to 14.57 mM in node (J) (**A**). Interestingly, the concentration curves overlap at (I) and (J), for $t > 0.5$ sec. We hypothesize that this is due to the vanishing of the potential at $t > 0.5$ sec, highlighting the role of the convective term in ionic dynamics (ion transport resulting from the electric field). Concerning the potential dynamics, the maximum of the signal arrives in (J) with no delay and a decrease in amplitude of 31 %, decreasing from 44.81 mV to 30.91 mV. In summary, we observe a discrepancy between a rapid and strong invasion of the electrical signal in the inactive branch, and a low invasion of the ionic concentration signal, with a delay of a few hundred ms.

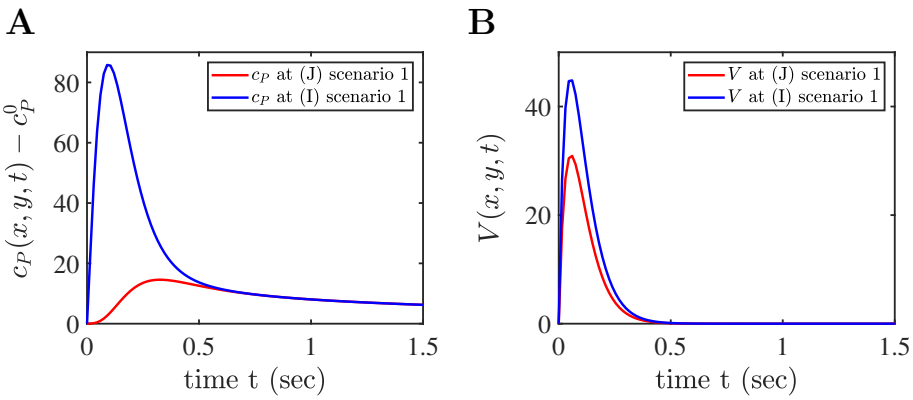

FIGURE 10. Time evolution of the dynamics of c_P and V in the small branches for scenario **1**. **A**: $(c_P^{K_J^*,n} - c_P^0)^{\mathbf{1}}_{n=0,..,N}$, at position (J) in red and $(c_P^{K_I^*,n} - c_P^0)^{\mathbf{1}}_{n=0,..,N}$, at position (I) in blue. **B**: $(V^{K_J^*,n})^{\mathbf{1}}_{n=0,..,N}$, at position (J) in red and $(V^{K_I^*,n})^{\mathbf{1}}_{n=0,..,N}$, at position (I) in blue.

3.3.2. **Modeling and simulation of ionic and voltage dynamics in dendritic spines.** In this section, we investigate the effect of the specific geometry of dendritic spines on voltage and ionic dynamics. Dendritic spines are located on the dendritic tree and serve as sites for receiving synaptic input in the form of an influx of ions. Their variations in size and shape are usually associated with the neuronal coding of learning and memory. Extensive efforts have been made to develop experimental techniques for visualizing and analyzing ionic and electric field dynamics in dendritic spines. However, due to their microscopic scale, such experimental investigations remain challenging.

3.3.2.1. ***Influence of the spine head geometry on ionic and voltage dynamics.*** Dendritic spines have a bulbous head connected to a thin neck, that we model in 2D by a domain Ω_S composed of a circle with radius r (the head) connected to a rectangle of length L_1 and width D_1 (the neck) (Fig. 11). All parameters related to domain Ω_S are defined in Table 6. The influx of ions is modeled, as in the previous section, with a non-homogeneous Neumann boundary condition on $\partial\Omega_i$. We set,

as previously described, homogeneous boundary conditions on $\partial\Omega_r$, as well as Dirichlet boundary conditions on Γ_{Dir} (see Fig. 11).

The mesh used in our simulations is gradually refined along the boundary $\partial\Omega_S$ of domain Ω_S and consists of 6589 triangular cells with a mesh size of $h = 7.33 \times 10^{-2}$. As an example, Figure 11 shows a zoomed-in view of the mesh at the junction between the head and the neck.

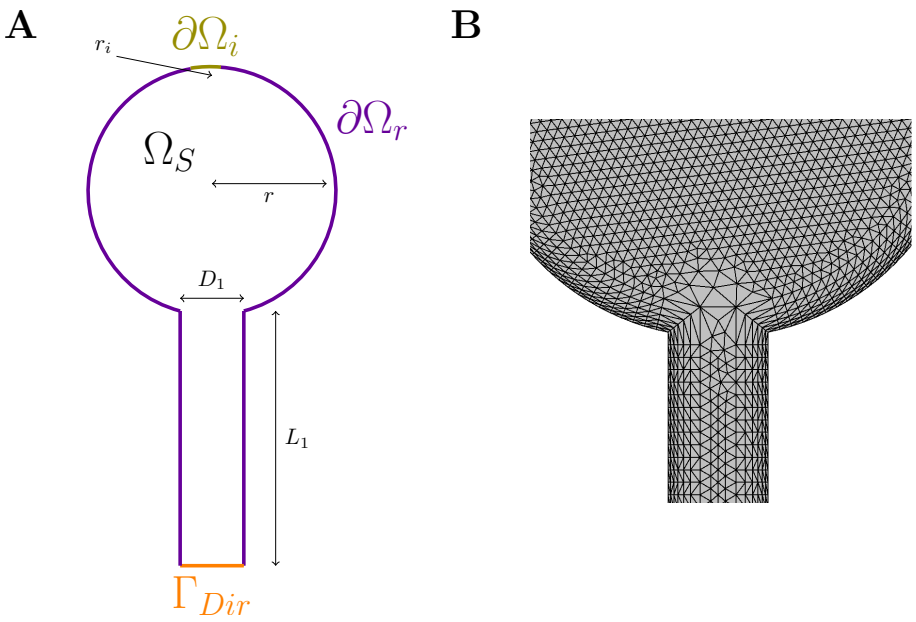

FIGURE 11. **A**: Domain Ω_S representing a dendritic spine. **B**: Mesh for domain Ω_S zoomed-in at the head-neck junction where it is refined close to the boundaries. The mesh size is $h = 7.33 \times 10^{-2}$.

r	0.5 µm	Radius of the head
L_1	1 µm	Length of the neck
D_1	0.2 µm	Diameter of the neck
r_i	0.04 µm	Length of $\partial\Omega_i$ (Fig. 11)

TABLE 6. Geometric parameters for domain Ω_S, representing a dendritic spine

In the following, we present the numerical simulations of the scheme (3.2)-(3.12) on domain Ω_S, with boundary conditions (3.18). The simulation is performed with a final time of $T_f = 0.5$ sec and a time step of $dt = 5 \times 10^{-3}$ sec.
In Fig. 12, we plot the ionic concentration and voltage dynamics at time $t_0 = 0.075$ sec in domain Ω_S. Panel **A** (resp. panel **B**) shows the dual concentration values $(c_P^{K^*,15} - c_P^0)_{K^* \in \overline{\mathfrak{M}^*}}$ (resp. $(V^{K^*,15})_{K^* \in \overline{\mathfrak{M}^*}}$ of the potential V), at time $t_0 = 0.075$ sec.

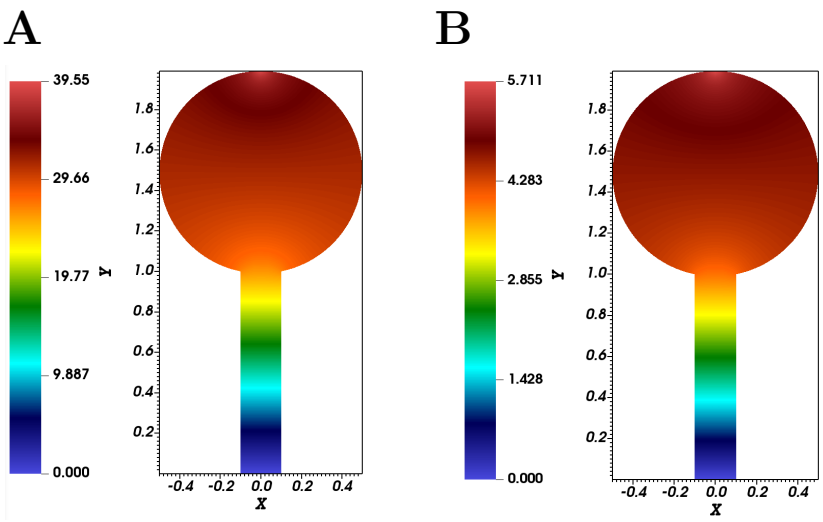

FIGURE 12. **A**: Dual values $\left(c_P^{K^*,15} - c_P^0\right)_{K^* \in \overline{\mathfrak{M}^*}}$ of $c_P - c_P^0$ in domain Ω_S at $t_0 = 0.075$ sec. **B**: Dual values $\left(V^{K^*,15}\right)_{K^* \in \overline{\mathfrak{M}^*}}$ of the potential V at $t_0 = 0.075$ sec.

We now investigate the effects of the specific geometry of the spine on ionic concentration and voltage dynamics, and in particular the spatial variations of voltage and ionic concentration in the spine head. We then compare our 2D-simulation results with the 1D results presented in [21]. In [21], the authors introduced a temporal deconvolution procedure (STAR method) to recover voltage dynamics in dendritic spines, from fluorescence images of a genetically-encoded voltage sensor. This procedure is based on the PNP system of equations. From a domain similar to Ω_S, the authors derived a 1D equivalent model reducing the spine head as a point and simulated the PNP system of equations using COMSOL. Note that we use the same input current at $\partial\Omega_i$ as in [21].

We consider several lines on the domain Ω_S: $\{x = 0\}$ and the lines $\{y = k\}$ for $k \in \{0.5, 1, 1.3, 1.5, 1.7, 1.9, 2\}$ (Fig. 13). We finally define the points P_k, at the intersection between $\{x = 0\}$ and $\{y = k\}$, i.e. the points $(0, k)$.

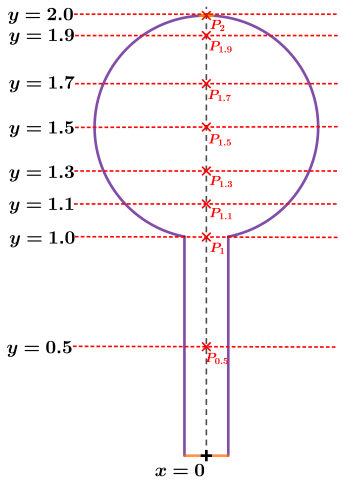

FIGURE 13. Schematic representation of a dendritic spine, illustrating the lines $\{x = 0\}$ (dashed black) and $\{y = k\}$ (dashed red) for $k \in \{0.5, 1, 1.1, 1.3, 1.5, 1.7, 1.9, 2\}$. The points P_k are the intersections between the lines $\{x = 0\}$ and $\{y = k\}$ (red star).

We plot in Fig. 14 the time evolution of the dual concentration value $(c_P^{K^*_{P_k},n} - c_P^0)_{n=0,..,N}$ (**A**) and the dual potential value $(V^{K^*_{P_k},n})_{n=0,..,N}$ (**B**) at points P_k (eq. (1.2)). We observe that the peak amplitude for $c_P - c_P^0$ (resp. V) is equal to 38.59 mM (resp. 6.37 mV) at (P_2) and decreases to 26.16 mM (resp. 4.36 mV) at (P_1), i.e. a decrease of more than 30 % for both the concentration and voltage. We also observe that the decrease is faster in the neck, with the peak amplitude reaching 13.35 mM (resp. 2.31 mV) at $P_{0.5}$ for the concentration (resp. voltage).
In [21] the peak amplitude of $c_P - c_P^0$ (resp. V) at the point representing the head, reaches around 33 mM (resp. around 5 mV) (see Fig. 4**C**, resp. Fig. 2**D-E** in [21]), which is very close to our 2D results at point P_2. Nevertheless, our results indicate that almost a third of the signal is lost within the head, highlighting the necessity to simulate the dynamics inside the head.

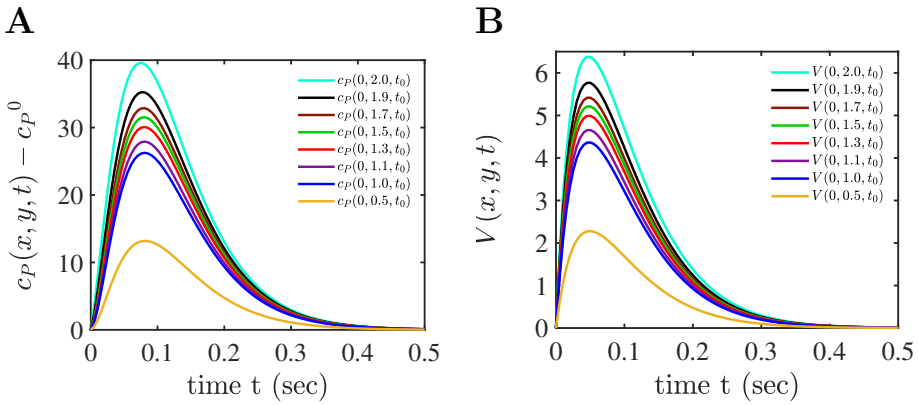

FIGURE 14. Time evolution of the ionic concentration $c_P - c_P^0$ and voltage V in the dendritic spine, at position P_k, for $k \in \{0.5, 1, 1.3, 1.5, 1.7, 1.9, 2\}$. **A**: $(c_P^{K^*_{P_k},n} - c_P^0)_{n=0,..,N}$. **B**: $(V^{K^*_{P_k},n})_{n=0,..,N}$.

We finally investigate the spatial variation of $c_P - c_P^0$ within the spine, at peak time t_0=0.075 sec. In Fig. 15**A**, we plot the dual concentration values $(c_P^{K^*,15} - c_P^0)_{K^* \in Z^*_{x=0}}$ (see eq. (1.1)). We observe that the concentration decreases linearly within the neck, reaching the Dirichlet boundary condition at $(x, y) = (0, 0)$, which suggests that a 1D approximation is relevant in the neck. Nevertheless, in the head, the variation of the concentration profile is more complex, with a higher gradient at the top of the head, near the influx of ions ($\partial\Omega_i$), followed by a lower gradient down to the neck. In Fig. 15**B**, we plot $(c_P^{K^*,15} - c_P^0)_{K^* \in Z^*_{y=k}}$ for $k \in \{0.5, 1, 1.3, 1.5, 1.7, 1.9, 2.0\}$ (eq. (1.1)). The variations along the lines $Y_{0.5}$ and Y_1, located inside the neck, are very small. In contrast, the geometry of the head creates concentration gradients, that are more pronounced close to the boundary receiving the influx of ions, and close to the head-neck junction. On line $Y_{1.9}$, the concentration decreases from 35.46 mM at position $x = 0$, to 33.33 mM at the boundary (black curve on Fig. 15**D**), which corresponds to a decrease of 6% relative to the reference value $c_P^{K_{P^*_{1.9}},15} - c_P^0 = 35.46$ mM. Note that the voltage has qualitatively the same behavior.

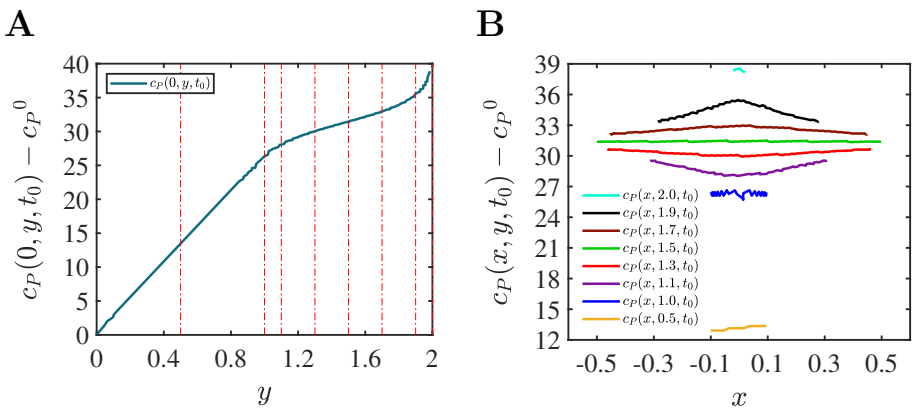

FIGURE 15. **A**: Spatial evolution of $(c_P^{K^*,15} - c_P^0)_{K^* \in Z^*_{x=0}}$ on line $x = 0$ at time $t_0 = 0.075$ sec. **B**: Spatial evolution of $(c_P^{K^*,15} - c_P^0)_{K^* \in Z^*_{y=k}}$ for $k \in \{0.5, 1, 1.3, 1.5, 1.7, 1.9, 2.0\}$ at time $t_0 = 0.075$ sec.

In summary, we observe that the ionic and voltage dynamics in the neck of dendritic spines are well captured by a 1D model. In contrast, to capture the specific dynamics of voltage and concentration inside the head, one needs to take into account its bulbous geometry. In particular, in our 2D simulations, we observe that the peak amplitude is decreased by almost 30 % between the top of the head and the head-neck junction, and the decrease is non-linear. This justifies the need for at least a 2D approach to obtain accurate simulations of ionic concentration dynamics in a spine.

3.3.2.2. ***Influence of the distance between an ionic reservoir and a dendritic spine on the ionic and voltage dynamics.*** In this part, we consider two neighboring dendritic spines and investigate the influence of a signal arriving in one spine, on the ionic and voltage dynamics of the other (Fig. 16). We test the hypothesis that dendritic spines act as autonomous compartments, isolating the material located at their head from the rest of the dendritic tree [**52**], versus the hypothesis of signal invasion, when the voltage and concentration in a spine is substantially increased by an ionic influx arriving in a neighboring one.

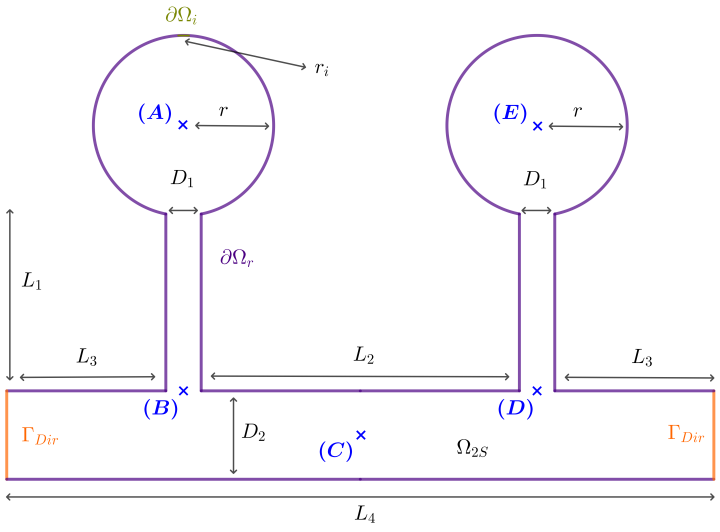

FIGURE 16. Domain Ω_{2S}, representing two neighboring dendritic spines. Each spine has a head (circle with centers (A) and (E)) and a neck (thin vertical rectangle). They are both connected to a dendrite (large horizontal rectangle, with points (B), (C) and (D)).

We consider the domain Ω_{2S} (Fig. 16), with parameters given in Table 7. Within one of the spines, which we call the active spine, we model an influx of ions at boundary $\partial\Omega_i$, using a non-homogeneous Neumann boundary condition, similar to the previous subsections, eq. (3.19). The spine that does not receive an influx of ions is called the inactive spine. We address the questions of the distance at which the influx of ions propagates from the active spine, and to what extent the inactive spine perceives this influx. We use a triangular mesh with 2385 cells and a mesh size $h = 0.16$. The simulations are performed with a final time of $T_f = 5$ sec, and a time step of $dt = 5 \times 10^{-3}$ sec.

Two Spines parameters		
r	0.5 µm	Radius head for both spines
L_1	1 µm m	Length neck for both spines
L_2	1.8 µm	Distance between the two spines
L_3	0.9 µm	Distance between the spine neck and the ionic reservoir.
L_4	4 µm	Total dendrite length
D_1	0.2 µm	Spine neck width.
D_2	0.5 µm	Dendrite width.
r_i	0.1 µm	Length of $\partial\Omega_i$
(A)	(0,1.5)	Coordinates of point (A)
(B)	(0,0)	Coordinates of point (B)
(C)	(1,-0.25)	Coordinates of point (C)
(D)	(2,0)	Coordinates of point (D)
(E)	(2,1.5)	Coordinates of point (E)

TABLE 7. Parameters for the domain Ω_{2S}, representing two neighboring dendritic spines (Fig. 16).

We investigate the effect of the distance L_3, between a spine and the Dirichlet boundary condition Γ_{Dir}, on voltage and ionic dynamics. We consider several domains with L_3 ranging from 0.9 µm to 23.9 µm . In each configuration, the length L_4 is modified such that $L_4 = 2L_3 + 2D_1 + L_2$, and L_2 is set to 1.8 µm (Fig. 16). The different mesh information are given in Table 8.

Value for L_3	0.9 µm	2.9 µm	5.9 µm	11.9 µm	23.9 µm
Value for L_4	4 µm	8 µm	14 µm	26 µm	50 µm
Number of triangular cells	2385	3365	4840	7773	17548
h	1.6×10^{-1}	1.6×10^{-1}	1.6×10^{-1}	1.6×10^{-1}	1.6×10^{-1}

TABLE 8. Parameter values that are modified in the five configurations of domain Ω_{2S}.

Fig. 17 illustrates the influence of the length L_3 on the dynamics of $c_P - c_0$. We plot the dual values $(c_P^{K^*,20} - c_0)_{K^* \in \overline{\mathfrak{M}^*}}$, $(c_P^{K^*,20} - c_0)_{K^* \in \overline{\mathfrak{M}^*}}$ and $(c_P^{K^*,20} - c_0)_{K^* \in \overline{\mathfrak{M}^*}}$ at times $t_0 = 0.1$ sec, for three domains Ω_{2S} with $L_3 = 0.9$ µm , 2.9 µm and 5.9 µm . Note that the minimum value for $c_P - c_0$ is imposed by the Dirichlet boundary condition at Γ_{Dir} ($c_P - c_0 = 0$ mM, with $c_0 = 163$ mM, Table 3). We observe that the size of the domain influences the values of the concentration within the two heads, with a larger L_3 leading to a higher concentration in the active spine, and a lower concentration value in the inactive one. We also observe a slight increase in $c_P - c_0$ peak time when L_3 increases. Hence, the results suggest that the farther the spine is from a large ionic reservoir, the farther the signal propagates.

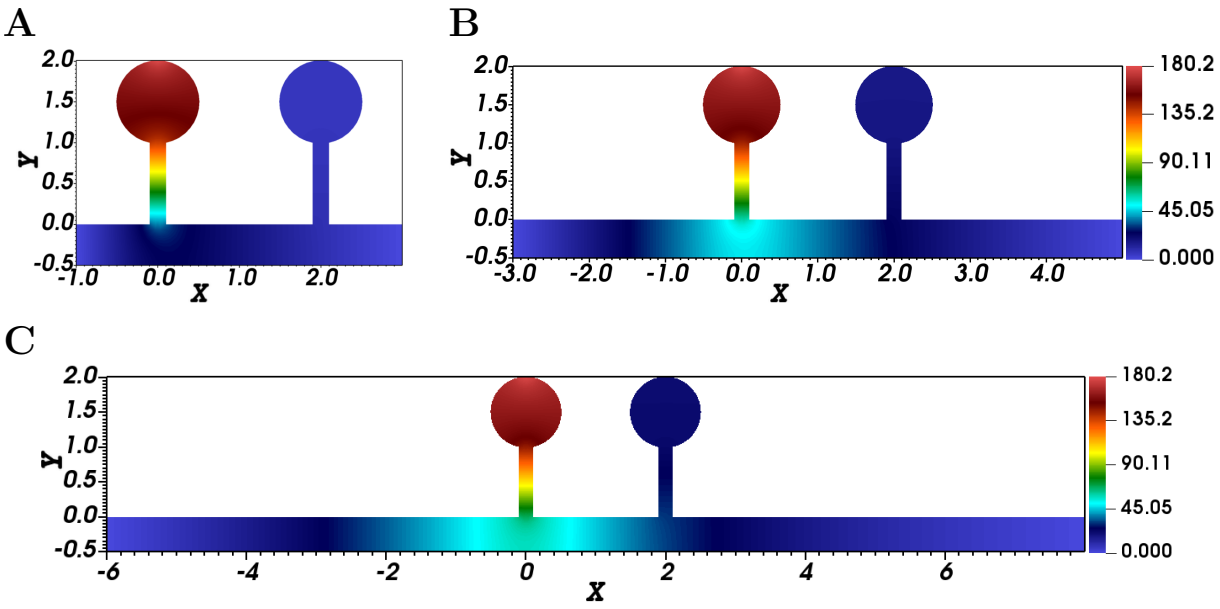

FIGURE 17. Spatial variations of the concentration c_P in three domains with increasing L_3. We plot the dual values $(c_P^{K^*,20} - c_0)_{K^* \in \overline{\mathfrak{M}^*}}$ in domain Ω_{2S}, with $L_3 = 0.9$ µm (**A**), $L_3 = 2.9$ µm (**B**) and $L_3 = 5.9$ µm (**C**) at $t_0 = 0.1$ sec. The range of the colorbar is fixed between plots, with a maximum value $c_P - c_0 = 180.2$ mM, corresponding to the maximum of $c_P - c_0$ in the configuration with $L_3 = 5.9$ µm .

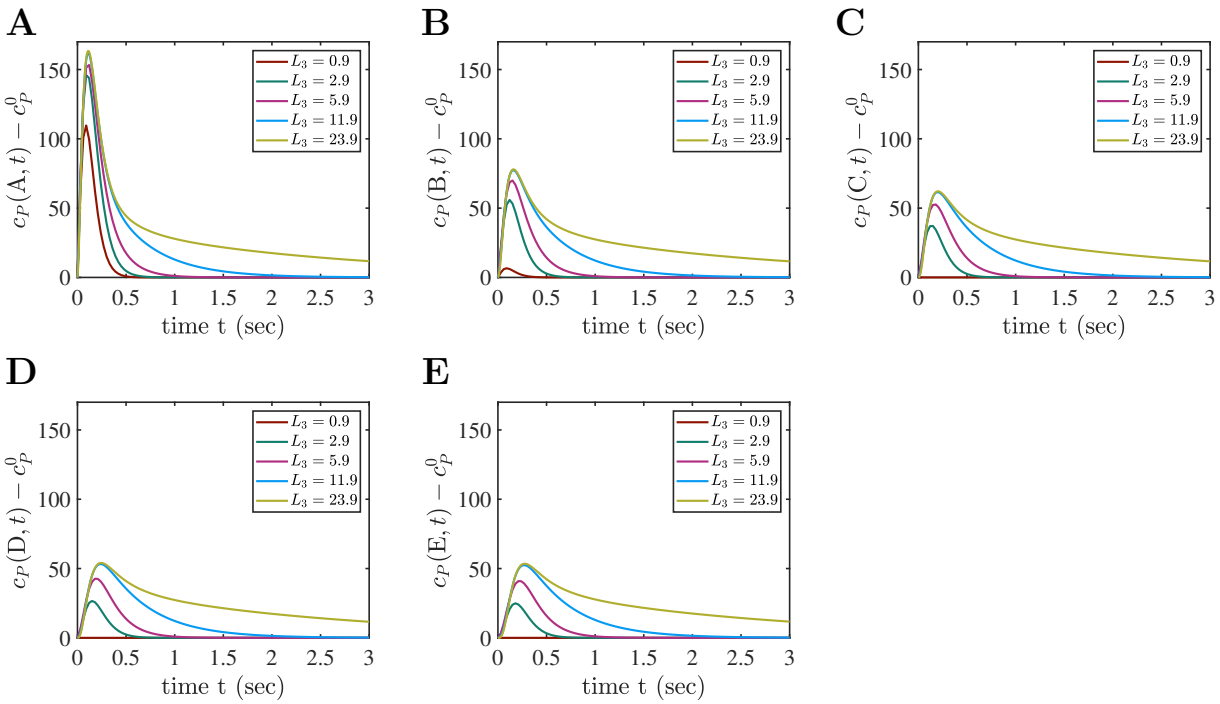

FIGURE 18. Time evolution of $(c_P^{K_X^*,n} - c_P^0)_{n=0,..,N}$, for different values of L_3 ranging from 0.9 to 23.9 µm , at positions (A)-(B)-(C)-(D)-(E). **A**: $(c_P^{K_A^*,n} - c_P^0)_{n=0,..,N}$. **B**: $(c_P^{K_B^*,n} - c_P^0)_{n=0,..,N}$. **C**: $(c_P^{K_C^*,n} - c_P^0)_{n=0,..,N}$. **D**: $(c_P^{K_D^*,n} - c_P^0)_{n=0,..,N}$. **E**: $(c_P^{K_E^*,n} - c_P^0)_{n=0,..,N}$.

We then investigate the propagation of the concentration within the dendrite and inactive spine at several specific points (A) - (E) (Fig. 16, Table 7) in the five different configurations (L_3 ranging from

0.9 µm to 23.9 µm). We plot in Fig. 18 the time evolution of the concentrations $(c_P^{K_X^*,n} - c_P^0)_{n=0,..,N}$ (see (1.2)). Note that the curves plotted in **D** and **E** almost superimpose, with a difference between them lower than 1 mM. We observe the effect of the Dirichlet boundary condition on the time evolution of the concentration. As expected, the shorter L_3, the bigger the impact on signal propagation. Indeed, for $L_3 = 0.9$ µm (red), we observe that the max of $c_P - c_0$ goes from 109.62 mM in (A) to 6.60 mM in (B), and is below 10^{-5} mM at (C), (D) and (E), i.e. the signal does not propagate inside the dendrite and the inactive spine. This means that for short L_3 the active spine behaves as an autonomous compartment, isolating its material from the rest of the dendritic tree.
For $L_3 = 2.9$ µm , the max of $c_P - c_0$ goes from 145.95 mM in (A) to 55.92 mM in (B), and 24.90 mM in (E). Hence, the ratio of signal reaching (E), defined as $r^c_{A-E} = \frac{\max\limits_{n=0,..,N}\left(c_P^{K_E^*,n} - c_P^0\right)}{\max\limits_{n=0,..,N}\left(c_P^{K_A^*,n} - c_P^0\right)}$, is 17 %.
For $L_3 = 11.9$ µm and 23.9 µm , a threshold seems to emerge, with 32 % of the peak concentration in (A) transmitted to (E). We also observe a shift in the peak time between position (A) and (E), denoted by Δ^c_{A-E}, going from 0.03 sec for $L_3 = 0.9$ µm , to 0.09 sec for $L_3 = 2.9$ µm and 0.16 sec for $L_3 = 23.9$ µm . We observe similar behavior for the voltage, where the ratio of the signal reaching (E), $r^V_{A-E} = \frac{\max\limits_{n=0,..,N}\left(V^{K_E^*,n}\right)}{\max\limits_{n=0,..,N}\left(V^{K_A^*,n}\right)}$, is 8% (resp. 63%, resp. 81%) for $L_3 = 0.9$ µm (resp. L_3 =11.9 µm , resp. L_3 =23.9 µm). The shift in the peak time between the electrical signal in (A) and the electrical signal in (E), denoted by Δ^V_{A-E}, is below 0.02 sec in all curves. Table 9 gives, for each configuration, the time shift Δ^c_{A-E}, the peak amplitude $\max\limits_{n=0,..,N}\left(c_P^{K_E^*,n} - c_P^0\right)$ at position (E) and the ratio r^c_{A-E} of signal reaching (E), for the concentration dynamics, as well as the time shift Δ^V_{A-E}, the peak amplitude $\max\limits_{n=0,..,N}\left(V^{K_E^*,n}\right)$ at position (E) and the ratio r^V_{A-E} of signal reaching (E) for the potential dynamics. This indicates that in this condition, a signal arriving in a spine is influencing the ionic concentration and voltage in its inactive neighbors, which we call a signal invasion. The voltage invasion is more important than the ionic concentration invasion. To summarize, our simulations suggest that depending on the distance to the closest ionic reservoir, a spine can either act as an autonomous compartment isolated from its neighbors or be subject to signal invasion.

L_3	Δ^c_{A-E}	$\max\limits_{n=0,..,N}\left(c_P^{K_E^*,n} - c_P^0\right)$	r^c_{A-E}	Δ^V_{A-E}	$\max\limits_{n=0,..,N}\left(V^{K_E^*,n}\right)$	r^V_{A-E}
0.9 µm	0.03 sec	10^{-5} mM	0 %	0.01 sec	1.65 mV	8 %
2.9 µm	0.09 sec	24.9 mM	17 %	0.01 sec	7.62 mV	26 %
5.9 µm	0.09 sec	40.94 mM	27 %	0.02 sec	17.58 mV	44 %
11.9 µm	0.16 sec	52.44 mM	32 %	0.02 sec	38.41 mV	63 %
23.9 µm	0.16 sec	53.45 mM	32 %	0.02 sec	103.46 mV	81 %

TABLE 9. Time shift Δ^c_{A-E} in the concentration signal, peak amplitude $\max\limits_{n=0,..,N}\left(c_P^{K_E^*,n} - c_P^0\right)$ of the concentration at position (E), ratio r^c_{A-E} of the concentration signal in (A) reaching the inactive spine (E), time shift Δ^V_{A-E} of the electrical signal, peak amplitude $\max\limits_{n=0,..,N}\left(V^{K_E^*,n}\right)$ at position (E) and ratio r^V_{A-E} of the electrical signal from (A) reaching (E), for the five different configurations with L_3 going from 0.9 µm to 23.9 µm .

3.3.2.3. ***Influence of the distance between two spines on signal invasion.*** This part focuses on varying the distance L_2 between two spines (Fig. 16). We consider five different values of L_2, ranging from 0.9 to 10.8 µm , and set L_3 to 23.9 µm , to ensure that we reach the plateau in signal invasion observed in subsection 3.3.2.2, for all the configurations. The mesh information are given in Table 10.

Value for L_2	0.9 µm	1.8 µm	3.6 µm	7.2 µm	10.8 µm
Value for L_4	49.1 µm	50 µm	51.8 µm	55.4 µm	59 µm
Number of triangle cells	17598	17548	17548	17546	17548
h	1.6×10^{-1}	1.6×10^{-1}	1.6×10^{-1}	1.6×10^{-1}	1.6×10^{-1}
Position (x, y) of the node (D)	(1.1,0)	(2,0)	(3.8,0)	(7.4,0)	(11,0)
Position (x, y) of the node (E)	(1.1,1.5)	(2,1.5)	(3.8,1.5)	(7.4,1.5)	(11,1.5)

TABLE 10. Parameter values that are modified in the five configurations of domain Ω_{2S}.

Fig. 19 illustrates the time evolution of the concentration $c_P - c_0$, at position (E) (Fig. 16), corresponding to the center of the head of the inactive spine. We plot the dual values $(c_P^{K_E^*,n} - c_P^0)_{n=0,..,N}$ at each time step, for the five configurations of domain Ω_{2S}.

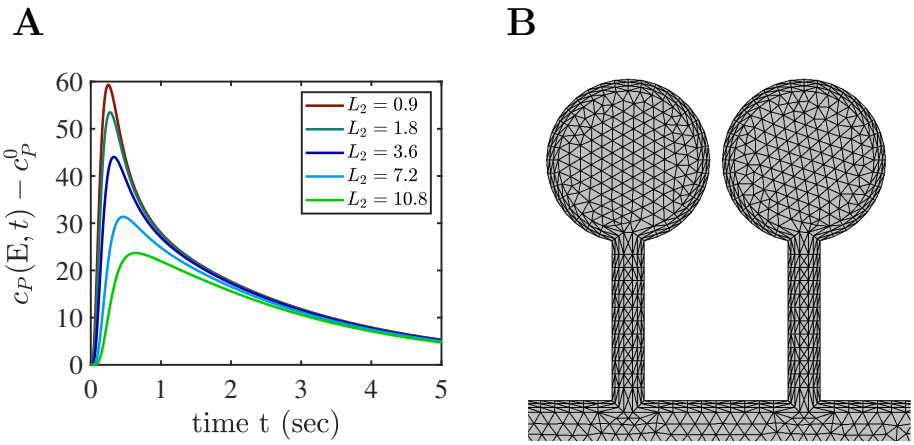

FIGURE 19. **A**: Time evolution of $(c_P^{K_E^*,n} - c_P^0)_{n=0,..,N}$, for different values of L_2 ranging from 0.8 to 10.8 µm , at position (E). **B**: Example of the mesh employed for the simulations, in the case $L_2 = 0.9$ µm , zoomed on the spines.

We observe, as expected, that concerning the variation of concentration, the closer the two spines, the larger the signal received by the inactive one. The shift in the peak time between position (A) and (E), Δ^c_{A-E}, increases when the distance between the spines L_2 increases. We observe a significant shift in the peak concentration time in the inactive spine compared to the active one, going from $\Delta^c_{A-E} = 0.145$ sec in the case of close spines ($L_2 = 0.9$ µm) to $\Delta^c_{A-E} = 0.535$ sec for spines that are farther away ($L_2 = 10.8$ µm). The ratio r^c_{A-E} of signal reaching (E) decreases while increasing L_2, going from 37 % when $L_2 = 0.9$ µm , to 14 % for $L_2 = 10.8$ µm .
Concerning the potential dynamics, the shift in the peak time Δ^V_{A-E} is one order of magnitude smaller, going from 0.02 sec for $L_2 = 0.9$ µm to 0.03 sec for $L_2 = 10.8$ µm . The ratio r^V_{A-E} of the electrical signal reaching (E) goes from 84% for $L_2 = 0.9$ µm , to 62% for $L_2 = 10.8$ µm .
In Table 11 we give, for each configuration, the time shift Δ^c_{A-E}, the peak amplitude $\max_{n=0,..,N} \left(c_P^{K_E^*,n} - c_P^0\right)$ at position (E), and the ratio r^c_{A-E} of signal reaching (E) for the concentration dynamics, as well as the time shift Δ^V_{A-E}, the peak amplitude $\max_{n=0,..,N} \left(V^{K_E^*,n}\right)$ at position (E) and ratio r^V_{A-E} of signal reaching (E) for the potential dynamics. These results suggest that spines receive electrical signals from other spines far away, as a spine at a 10.8 µm distance still receives 63 % of the signal, with tens of ms of delay. The variation in concentration is sensed to a lesser extent, as a spine at a 10.8 µm distance only perceives 14% of the variation of concentration, with a delay superior to 0.5 sec.

L_2	Δ^c_{A-E}	$\max_{n=0,..,N}\left(c_P^{K_E^*,n}-c_P^0\right)$	r^c_{A-E}	Δ^V_{A-E}	$\max_{n=0,..,N}\left(V^{K_E^*,n}\right)$	r^V_{A-E}
0.9 µm	0.145 sec	59.26 mM	37 %	0.02 sec	107.33 mV	84 %
1.8 µm	0.165 sec	53.45 mM	32 %	0.02 sec	103.46 mV	81 %
3.6 µm	0.215 sec	44.01 mM	27 %	0.03 sec	97.06 mV	77 %
7.2 µm	0.355 sec	31.35 mM	19 %	0.03 sec	87.24 mV	69 %
10.8 µm	0.535 sec	23.68 mM	14 %	0.03 sec	77.12 mV	62 %

TABLE 11. Time shift Δ^c_{A-E} in the concentration signal, peak amplitude $\max_{n=0,..,N}\left(c_P^{K_E^*,n}-c_P^0\right)$ of the concentration at position (E), ratio r^c_{A-E} of the concentration signal in (A) reaching the inactive spine (E), time shift Δ^V_{A-E} of the electrical signal, peak amplitude $\max_{n=0,..,N}\left(V^{K_E^*,n}\right)$ at position (E) and ratio r^V_{A-E} of the electrical signal from (A) reaching (E), for the five different configurations with L_2 going from 0.9 to 10.8 µm .

3.4. Discussion and conclusion

In this chapter, we present an algorithm for simulating the Poisson-Nernst Planck system of equations in two dimensions using the DDFV method, while preserving the positivity of ionic concentration. We challenge our algorithm using different test cases and achieve a second-order accuracy in space, which is consistent with existing results coming from literature. We also show that our system can capture a boundary layer when present. The proof of the existence of the discrete solution of (3.2)-(3.12) is both too long and not the goal of the present chapter. Hence, a following work will detail the existence of a solution, as well as positivity and convergence, most probably using similar techniques as in [**75**].

Using our DDFV framework, we then investigate the ionic and voltage dynamics in two-dimensional specific geometries of the neuronal dendritic tree: a branch bifurcation and a dendritic spine. First, our two-dimensional results show that the voltage and ionic dynamics in a dendritic spine are not well approximated by one-dimensional models. Indeed, if a one-dimensional approximation for both voltage and concentration is relevant in the neck, the bulbous geometry of the head requires at least a two-dimensional approach, as the space dynamics is highly non-linear, with steep gradients at the top of the head and around the head-neck junction.

Our numerical results also highlight that dendritic spines can sense electrical signals far away on a thin branch, whereas it is not the case for ionic concentration transients, that reach only close spines. In our configuration, a spine at a distance 10.8 µm of a spine receiving a signal will get 62% of the electrical signal and only 14% of the concentration signal. One direct consequence of this, is that calcium imaging cannot be used as a tool to observe sub-threshold electrical signal propagation in neuronal compartments, especially when voltage-gated calcium channels are not opening. The actual development of voltage sensors could resolve this difficulty.

Our final result concerns the influence of an ionic reservoir on voltage and calcium dynamics. This reservoir can represent any large branch that the neuronal compartment is connected to. The rationale behind this is that the large branch being so large compared to the thin branch, we can consider the ionic concentrations and voltage to be constant at the connection. In the equations, the reservoir is modeled using a Dirichlet boundary condition and behaves as a sink. Our simulations show that the close proximity of an ionic reservoir such as the dendritic shaft or any large compartment, is killing the signal, preventing it from propagating and invading close neuronal structures such as thin branches at a bifurcation or dendritic spines. On the opposite, a signal arriving at the leading edge of the dendritic tree, far from an ionic reservoir and where only small branches are present, will propagate at a larger distance and invade neighboring dendritic spines.

From the spine point of view, these results show that the same spine at different positions in the tree would behave differently: a spine close to the dendritic shaft would more likely act as an autonomous compartment, compared to a collection of spines located in small dendritic protrusions, that would be keener to signal invasion and to influence each other. Hence, in addition to the geometry of the spine,

the local geometry of the dendritic tree is shaping spine behavior, in that the same spine at different positions in the tree would behave differently. Hence, the position of a dendritic spine relative to the entire tree (i.e. close to the soma versus at the distal edge of the tree) shapes its function, making plasticity not at the level of the spine, but at the level of the full dendritic geometry.
These results are in line with several experimental observations showing that synaptic development is spatially regulated inside the dendritic tree [**36**, **68**]. In particular, the electrical compartmentalization of dendritic spines is nowadays a leading question in developmental neuroscience, with various experimental studies showing a variety of results depending on the types of cells or on their developmental stages, and going from signals spreading locally and invading neighboring spines to compartmentalization [**32**, **86**, **64**, **87**]. Our study would suggest considering these differences in relation to the actual size of the dendritic tree, and especially the presence of large dendrites. In the same way, the distance to the nearest large dendritic compartment should be considered while investigating cooperative, homosynaptic and heterosynaptic plasticity [**29**, **80**].

CHAPTER 4

Solving the Poisson-Nernst Planck system of equations over multi-domains using the FreeFEM++ software

Introduction

In this chapter, we delve into the PNP system within a new context. The previous chapters of this thesis have revolved around coupling the system of equations within a single domain. This domain models the internal environment of dendrites and dendritic spines. In this mono-domain approach, it is necessary to define a boundary condition for the potential, which has an impact on voltage and ionic dynamics. In Chapter 3, which we choose to set an homogeneous Neumann boundary condition ($\nabla V \cdot \mathbf{n} = 0$), which means that the membrane is a perfect insulator. Such boundary condition is restrictive, as it prohibits any electric flux from crossing the membrane. This, as observed numerically, lead to the absence of an electric boundary layer near the membrane wall. To get rid of the problem of choosing a boundary condition for the potential, we propose in this chapter to use a multi-domain approach to represent the interior of the dendrite, the membrane, and the exterior media. The condition for the potential then becomes a simple continuity condition across the membrane.
We therefore incorporate to the model the dielectric permittivity coefficients present in each medium (cytosol, membrane, external medium). The electric permittivity plays a role in the Poisson's equation and vary according to the sub-domain. The equation is thus solved over the entire domain (the electric potential "exists" everywhere), with only the permittivity coefficients dependent on the different domains. Moreover, through this new modeling framework, one may gain further insights regarding the boundary layer.
In this chapter, the coupling of equations across this multi-domain is performed using the FreeFEM++ software. This choice of numerical method was made due to time constraints and may serve as an initial result for future numerical comparisons with other existing numerical methods such as DDFV. FreeFEM++ enables to determine complex numerical solutions of nonlinear system problems like the PNP system, based on a variational formulation of the system of equations and its boundary conditions (according to the desired physical problem). The software further handles the Finite Element Method for spatial discretization. We solve the PNP system using an implicit Euler time discretization, akin to the approach tackled in the previous chapters, notably in the nonlinear coupling, Chapter 3. To address the nonlinearity of the system, we apply a Newton algorithm at each time step. Furthermore, it is important to note that our numerical scheme under FreeFEM++ does not inherently ensure the preservation of ion concentration positivity. We subsequently consider the PNP system in eq. (0.6) (i.e without the nonlinear reformulation) and define it over the multi-domain. In the following sections, we give an in-depth exploration of our domain, providing all the essential elements required for its construction and expliciting our modeling choices. We introduce the PNP system as applied over this particular domain together with the appropriate boundary conditions. We then delve into the variational formulation of the system, which we have implemented using FreeFEM++. This section includes specific details regarding our numerical implementation.
In this chapter, we consider three distinct geometries representing neuronal compartments:

(1) a dendrite bifurcation, Figure 1,
(2) a large rectangular portion of dendrite, Figure 2,
(3) a smaller portion of dendrite designed with smooth angles on the membrane walls, Figure 3.

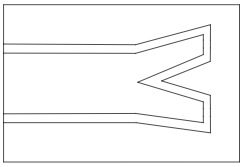

FIGURE 1. Illustration of the bifurcation (Section 4.2.1).

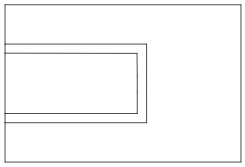

FIGURE 2. Illustration of the large rectangular branch (Section 4.2.2).

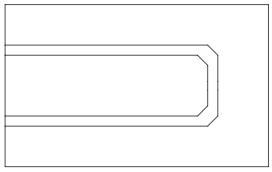

FIGURE 3. Illustration of the smaller portion with smooth angles (Section 4.2.3).

The three domains in Figure 1, 2 and 3, are designed within one main rectangular box, representing the external medium. Inside this box, we consider the intracellular medium and the membrane.

We first study the signal summation within the dendrite bifurcation, Figure 1. Indeed, we observe linear summation of two signals coming from two neighboring branches, similarly to Chapter 3. Then, we compare this latter approach with single-domain models, under FreeFEM++ and DDFV, which consider homogeneous Neumann boundary condition for the potential on the membrane, as the modelling in Chapter 3, Section 3.3.1.2.. We observe analogous results on ionic dynamics and voltage, thereby reinforcing the coherence of our approaches. Finally, we provide outcomes on the voltage summation within the membrane, under the multi-domain only.

Next, under the simpler rectangular modeling (2), we consider several scenarios. We examine the signal invasion over the dendrite when multiple ion influxes occur along the dendrite's walls. We investigate up to how many influxes we observe a drastic change in ion and voltage dynamics. We start with a single influx and progressively add more along the dendrite's walls. It is observed that, adding five influxes yields a significant increase in voltage, with the signal depolarizing. However, regarding concentrations, we do not observe remarkable variations. Nonetheless, we still observe a spatial invasion of ions propagating within the dendrite, which we hardly see when there is only one influx at the branch's end.

Finally, we present results related to the boundary layer within the intra-cellular media only, due to computational constraints. To achieve this, we consider a dendrite branch (3) that is a domain smaller than the rectangular branch (2). The employed mesh allows to take into account nodes close to the interior membrane wall and inside the boundary layer, which has a thickness of few nanometers.

These preliminary results are already demanding in terms of computation time. In future work, it will be necessary to focus on mesh refinement and to delve deeper into the functionalities of FreeFEM++ in order to consider a mesh adapted to the boundary layer. Still, within the boundary layer, we observe a small peak in intra-cellular ionic concentration and voltage. Beyond its thin layer, a relatively constant behavior for ionic concentrations and voltage is observed, which corresponds to the electroneutrality of the media. These results provide preliminary insights into the existence of the boundary layer and demonstrate that our model numerically catches the boundary layer. Still, we were unable to create a more suitable mesh where refinement would be more flexible and optimized for the boundary layer. We believe that a method like DDFV would be more suitable for addressing

the boundary layer.

This chapter serves as an exploration of a multi-domain modeling of dendritic branches, where we aim at providing new understandings of several biological questions within the dendritic tree. We aim at understanding how the electrical signal behaves in dendrites by providing information about its dynamics in the cytosol and within the membrane. We also seek to provide insights into the influence of synaptic inputs during neuronal excitation, for instance, during significant depolarization. Additionally, we aim to comprehend whether ions aggregate near the wall in the Debye layer and if this behavior influences voltage propagation or ionic dynamics.

This chapter is organized as follows: Section 4.1 presents the model over the multi-domain. Section 4.2 focuses on the numerical applications. Discussion and conclusion are provided in the last section.

4.1. Presentation of the model

4.1.1. **Continuous Problem in a Multi-Domain.** In view to accurately model the electrodiffusion processes within the dendrite geometry, it is crucial to establish the continuous problem for the multi-domain approach. In all subsequent sections, we set the interior as Ω_i, the membrane as Ω_m, and the external environment as Ω_e (see Figure 4). Intuitively, each domain requires specific boundary conditions to ensure a fair representation of the physical conditions.

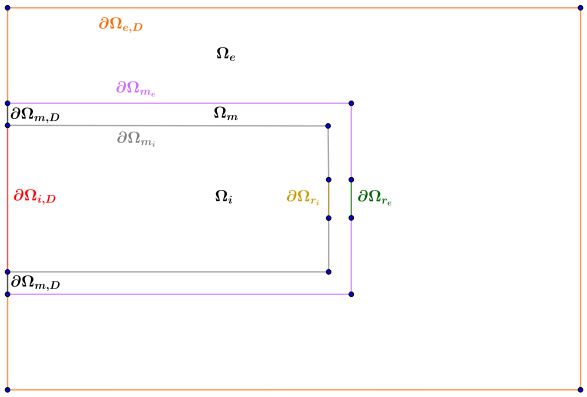

Figure 4. Multi-domain representation showing a large rectangular branch of dendrite with the interior as Ω_i (cytosol), the membrane as Ω_m, and the external environment as Ω_e. The membrane boundaries are defined as $\partial\Omega_{m,D} \cup \partial\Omega_{m_e} \cup \partial\Omega_{m_i} \cup \partial\Omega_{r_i} \cup \partial\Omega_{r_e}$. $\partial\Omega_{e,D}$ is the outer boundary of Ω_e and $\partial\Omega_{i,D}$ is the connection with a larger dendrite within Ω_i.

We now describe the different boundaries. First, $\partial\Omega_{e,D}$ forms the outer boundary of Ω_e, modeling the very distant bulk region. Then, $\partial\Omega_{i,D}$ forms the connection with a larger dendrite. This large dendrite in question is considered as an ionic reservoir due to its large size, i.e. with fixed ionic concentration. In the same way, $\partial\Omega_{m,D}$ characterizes the membrane boundary at the junction with the rest of the media. Next, $\partial\Omega_{m_e}$ designates the outer impermeable membrane wall, and $\partial\Omega_{m_i}$ the equivalent for the inner wall. Ultimately, $\partial\Omega_{r_i}$ and $\partial\Omega_{r_e}$, located on the inner and outer walls of the membrane, portray the influx of ions from a neighboring synapse.

Let us note that each medium within the domain possesses distinct electrodiffusive characteristics. To be more precise, the dielectric permittivity, for instance, varies across the neuron's environment. Consequently, for each medium we introduce a parameter, denoted η_q (akin to γ eq. (0.5) introduced

in the **Preamble**), where q designates the given medium, i.e., i, m, or e. η_q is based on these aforementioned electrodiffusive quantities, such as the dielectric permittivity for each medium (ε_q) exposed in Table 1, with $q = i$, m, or e. The values of the latter coefficients are presented for each medium in Table 1 below.

F	$96485\ A.s.mol^{-1}$	Faraday constant	
ε_i	80	Dielectric permittivity in Ω_i	[21]
ε_e	80	Dielectric permittivity in Ω_e	[21]
ε_m	4	Dielectric permittivity in Ω_m	[21]
ε_0	$8,8.10^{-12}\ F.m^{-1}$	Permittivity of vacuum	
T_θ	$293,15\ K$	Absolute temperature	
R	$8,314\ J.K^{-1}.mol^{-1}$	Gas constant	
β	39.5877	Eq. (0.5)	
D_P	$200\ \mu m^2.s^{-1}$	Diffusion coefficient for anion	[21]
D_N	$200\ \mu m^2.s^{-1}$	Diffusion coefficient for cation	[21]
$c^0_{P,i}$	$170\ mM$	Initial concentration for anion	*URL*
$c^0_{P,e}$	$146\ mM$	Initial concentration for anion	*URL*
$c^0_{N,i}$	$183\ mM$	Initial concentration for cation	*URL*
$c^0_{N,e}$	$131\ mM$	Initial concentration for cation	*URL*
$V^0_{m,i}$	$-75\ mV$	Initial electric potential	*URL*
V^0_e	$0\ mV$	Initial electric potential	*URL*
I_{max}	$300\ pA$	Maximal current	[21]
τ	$0.055\ s$	Decay time constant	[21]

TABLE 1. Electrodiffusion parameters.

The Poisson equation is defined over the whole domain $\Omega = \Omega_i \cup \Omega_m \cup \Omega_e$ and both NP equations are defined in the intra- and extra-cellular space, but not in the membrane medium, as we consider that there are no charges inside the membrane (Fig. 4). We have:

$$
\begin{aligned}
&\partial_t c_P = -\nabla \cdot \left(D_P \left(\nabla c_P + \beta c_P \nabla V\right)\right), && \text{in } \Omega_i \cup \Omega_e,\\
&\partial_t c_N = -\nabla \cdot \left(D_N \left(\nabla c_N - \beta c_N \nabla V\right)\right), && \text{in } \Omega_i \cup \Omega_e,\\
&-\nabla \cdot \left(\eta_q \beta \nabla V\right) = c_P - c_N, && \text{in } \Omega_i \cup \Omega_m \cup \Omega_e,
\end{aligned}
\tag{4.1}
$$

where $\eta_q = \dfrac{\varepsilon_q \varepsilon_0 R T_\theta}{F^2}$ with $q = i, e, m$.
We provide in the following section the initial and boundary conditions associated with the framework above.

4.1.2. **Initial and Boundary Conditions.** In this section, we describe the boundary conditions of the considered multi-domain, highlighting the essential role they play in capturing the behaviors of ionic species and electric potential at the interfaces between the environments. Each interface raises specific challenges, so that it is crucial to select appropriate boundary conditions to replicate physiological conditions effectively.

We introduce below the definitions of the initial and boundary conditions within the domain (see Fig. 4). Notice that we consider initial boundary conditions following the notations and definitions of Table 1, such that:

$$
\begin{aligned}
&c_P(t=0,\mathbf{x}) = c^0_{P,i}, \quad c_N(t=0,\mathbf{x}) = c^0_{N,i}, \quad V(t=0,\mathbf{x}) = V^0_i, && \forall \mathbf{x} \in \Omega_i,\\
&c_P(t=0,\mathbf{x}) = c^0_{P,e}, \quad c_N(t=0,\mathbf{x}) = c^0_{N,e}, \quad V(t=0,\mathbf{x}) = V^0_e, && \forall \mathbf{x} \in \Omega_e,\\
&V(t=0,\mathbf{x}) = V^0_m, && \forall \mathbf{x} \in \Omega_m.
\end{aligned} \tag{4.2}
$$

Then, we impose Dirichlet boundary condition on $\partial\Omega_D = \partial\Omega_{e,D} \cup \partial\Omega_{i,D} \cup \partial\Omega_{m,D}$, since we model an ionic reservoir inside and outside the dendrite. In addition, we consider $\partial\Omega_{m,D}$ as the junction with the rest of the membrane. We define the Dirichlet values to be those given in Table 1:

$$
\begin{aligned}
&c_{P,e} = c^0_{P,e}, \quad c_{N,e} = c^0_{N,e}, \quad V_e = V^0_e, \quad \text{on} \quad \partial\Omega_{e,D} \times (0,T_f),\\
&c_{P,i} = c^0_{P,i}, \quad c_{N,i} = c^0_{N,i}, \quad V_i = V^0_i, \quad \text{on} \quad \partial\Omega_{i,D} \times (0,T_f),\\
&V_m = V^0_m, \quad \text{on} \quad \partial\Omega_{m,D} \times (0,T_f).
\end{aligned} \tag{4.3}
$$

Thereafter, we define V^0 such as:

$$
V^0 = \begin{cases} V^0_e & \text{on } \partial\Omega_{e,D} \\ V^0_i & \text{on } \partial\Omega_{i,D} \\ V^0_m & \text{on } \partial\Omega_{m,D} \end{cases} \tag{4.4}
$$

In this paragraph, we focus on modeling boundary conditions on concentrations only. Indeed, as mentioned earlier, the potential "lives" everywhere and is therefore considered as continuous across the three domains. Hence, the only boundary condition for the potential is the Dirichlet boundary condition on $\partial\Omega_D$. Hence, for the concentration, we set Neumann boundary conditions on $\partial\Omega_{m_e} \cup \partial\Omega_{m_i} \cup \partial\Omega_{r_i} \cup \partial\Omega_{r_e}$. We model the ion influx at two distinct boundaries, denoted as $\partial\Omega_{r_i}$ and $\partial\Omega_{r_e}$, where the ion influx is modelled as an output within Ω_e and as an input within Ω_i. For the remaining part of the membrane, we apply a homogeneous Neumann boundary condition to model impermeability of the membrane for positive and negative ions. Altogether, the system of boundary conditions for the membrane is as follows:

$$
\begin{aligned}
&\nabla c_{N,e} \cdot \mathbf{n} = 0, && \text{on} \quad \partial\Omega_{m_e} \cup \partial\Omega_{r_e} \times (0,T_f),\\
&\nabla c_{N,i} \cdot \mathbf{n} = 0, && \text{on} \quad \partial\Omega_{m_i} \cup \partial\Omega_{r_i} \times (0,T_f),\\
&\nabla c_{P,e} \cdot \mathbf{n} = 0, && \text{on} \quad \partial\Omega_{m_e} \times (0,T_f),\\
&\nabla c_{P,i} \cdot \mathbf{n} = 0, && \text{on} \quad \partial\Omega_{m_i} \times (0,T_f),\\
&\nabla c_{P,e} \cdot \mathbf{n} = -I, && \text{on} \quad \partial\Omega_{r_e} \times (0,T_f),\\
&\nabla c_{P,i} \cdot \mathbf{n} = I, && \text{on} \quad \partial\Omega_{r_i} \times (0,T_f).
\end{aligned} \tag{4.5}
$$

In the above, the ion influx I injected into our model is similar to the one introduced in Chapter 3, taken from [21]. We recall its definition:

$$
I(t) = \frac{I_{stim}(t)}{\pi r^2 F D_P}, \tag{4.6}
$$

where the variable r is defined depending on the studied domain, and where:

$$I_{stim}(t) = I_{max}\frac{t}{\tau}\exp\left(-\frac{t}{\tau}+1\right). \tag{4.7}$$

4.1.3. **Variational Formulation with FreeFEM++.** To solve the PNP system within our multi-domain representation, we employ FreeFEM++ [51], a versatile tool for numerical solutions of complex physical problems. In this chapter, we use classical Finite Elements in the sense of Ciarlet [31]. And more precisely, we choose the polynomial space $\mathbb{P}_1$ of degree 1. We recall that:

Definition 4.1. *For each $k \in \mathbb{N}$, we denote $\mathbb{P}_k$ as the space of polynomials of total degree less than or equal to k. The space $\mathbb{P}_1$ corresponds to the space of affine functions:*

$$\mathbb{P}_1 = \{p; p(x) = a_0 + a_1x_1 + a_2x_2, a_i \in \mathbb{R}\}$$

The Finite Element mesh, denoted $\mathcal{T}_h$, is a union of triangles denoted K such that $\overline{\Omega} = \underset{K\in\mathcal{T}_h}{\cup} K$. The Finite Element space is defined as

$$P_h = \{v_h \in \mathcal{C}^0(\overline{\Omega});\ \forall K \in \mathcal{T}_h \quad v_{h|K} \in \mathbb{P}_1\}$$

and

$$P_{0,h} = \{v_h \in P_h;\ v_h = 0 \text{ on } \partial\Omega_D\}$$

where $\partial\Omega_D$ represents the Dirichlet boundary condition. Since we impose non-homogeneous Dirichlet boundary condition, we use the trace operator denoted γ_0 [1].

We indicate by e or i the domains Ω_e and Ω_i. For instance, for Ω_e, we denote the Finite Element space P_h^e, and the same for i. Thanks to FreeFem, the meshes are conform and compatible in the sense that the mesh of Ω restricted to Ω_e is the same to the mesh of Ω_e. We use an implicit Euler time discretization. The variational formulation is:

$$\begin{cases}
\text{Seek } (c_{h,P}^{n+1,e}, c_{h,N}^{n+1,e}, c_{h,P}^{n+1,i}, c_{h,N}^{n+1,i}, V_h^{n+1}) \in P_h^e \times P_h^e \times P_h^i \times P_h^i \times P_h \text{ such that} \\
F_{c_P}^{n,e}(c_{h,P}^{n+1,e}, V_h^{n+1}, v_h^e) = 0, \quad \forall v_h^e \in P_{0,h}^e, \\
\gamma_0(c_{h,P}^{n+1,e}) = c_{P,e}^0 \text{ on } \partial\Omega_{e,D}, \\
F_{c_N}^{n,e}(c_{h,N}^{n+1,e}, V_h^{n+1}, v_h^e) = 0, \quad \forall v_h^e \in P_{0,h}^e, \\
\gamma_0(c_{h,N}^{n+1,e}) = c_{N,e}^0 \text{ on } \partial\Omega_{e,D}, \\
F_{c_P}^{n,i}(c_{h,P}^{n+1,i}, V_h^{n+1}, v_h^i) = 0, \quad \forall v_h^i \in P_{0,h}^i, \\
\gamma_0(c_{h,P}^{n+1,i}) = c_{P,i}^0 \text{ on } \partial\Omega_{i,D}, \\
F_{c_N}^{n,i}(c_{h,N}^{n+1,i}, V_h^{n+1}, v_h^i) = 0, \quad \forall v_h^i \in P_{0,h}^i, \\
\gamma_0(c_{h,N}^{n+1,i}) = c_{N,i}^0 \text{ on } \partial\Omega_{i,D}, \\
F_V^n(V_h^{n+1}, c_{h,P}^{n+1,e}, c_{h,N}^{n+1,e}, c_{h,P}^{n+1,i}, c_{h,N}^{n+1,i}, v_h) = 0, \quad \forall v_h \in P_{0,h}, \\
\gamma_0(V_h^{n+1}) = V^0 \text{ on } \partial\Omega_D,
\end{cases} \tag{4.8}$$

where

$$F^{n,e}_{c_P}(c^{n+1,e}_{h,P}, V^{n+1}_h, v^e_h) = \int_{\Omega_e} \left(\frac{c^{n+1,e}_{h,P} - c^{n,e}_{h,P}}{dt} v^e_h - D_P \nabla c^{n+1,e}_{h,P} \nabla v^e_h - D_P \beta c^{n+1,e}_{h,P} \nabla V^{n+1}_h \nabla v^e_h \right) dX$$
$$- \int_{\partial\Omega_{r_e}} (I(t) v^e_h)\, ds.$$

$$F^{n,e}_{c_N}(c^{n+1,e}_{h,N}, V^{n+1}_h, v^e_h) = \int_{\Omega_e} \left(\frac{c^{n+1,e}_{h,N} - c^{n,e}_{h,N}}{dt} v^e_h - D_N \nabla c^{n+1,e}_{h,N} \nabla v^e_h + D_N \beta c^{n+1,e}_{h,N} \nabla V^{n+1}_h \nabla v^e_h \right) dX.$$

$$F^{n,i}_{c_P}(c^{n+1,i}_{h,P}, V^{n+1}_h, v^i_h) = \int_{\Omega_i} \left(\frac{c^{n+1,i}_{h,P} - c^{n,i}_{h,P}}{dt} v^i_h - D_P \nabla c^{n+1,i}_{h,P} \nabla v^i_h - D_P \beta c^{n+1,i}_{h,P} \nabla V^{n+1}_h \nabla v^i_h \right) dX$$
$$+ \int_{\partial\Omega_{r_i}} (I(t) v^i_h)\, ds.$$

$$F^{n,i}_{c_N}(c^{n+1,i}_{h,N}, V^{n+1}_h, v^i_h) = \int_{\Omega_i} \left(\frac{c^{n+1,i}_{h,N} - c^{n,i}_{h,N}}{dt} v^i_h - D_N \nabla c^{n+1,i}_{h,N} \nabla v^i_h + D_N \beta c^{n+1,i}_{h,N} \nabla V^{n+1}_h \nabla v^i_h \right) dX.$$

$$F^n_V(V^{n+1}_h, c^{n+1,e}_{h,P}, c^{n+1,e}_{h,N}, c^{n+1,i}_{h,P}, c^{n+1,i}_{h,N}, v_h) = \int_\Omega (-\eta_q \beta \nabla V^{n+1}_h \nabla v_h)\, dX + \int_{\Omega_e} (c^{n+1,e}_{h,P} - c^{n+1,e}_{h,N}) v_h\, dX$$
$$+ \int_{\Omega_i} (c^{n+1,i}_{h,P} - c^{n+1,i}_{h,N}) v_h\, dX.$$

and with

$$c^{0,e}_{h,P} = c^0_{P,e}, \qquad c^{0,e}_{h,N} = c^0_{N,e}, \qquad c^{0,i}_{h,P} = c^0_{P,i}, \qquad c^{0,i}_{h,N} = c^0_{N,i}.$$

We define the function F as

$$F(V^{n+1}_h, c^{n+1,e}_{h,P}, c^{n+1,e}_{h,N}, c^{n+1,i}_{h,P}, c^{n+1,i}_{h,N}, v^e_h, v^i_h, v_h) = \begin{bmatrix} F^n_V(V^{n+1}_h, c^{n+1,e}_{h,P}, c^{n+1,e}_{h,N}, c^{n+1,i}_{h,P}, c^{n+1,i}_{h,N}, v_h) \\ F^{n,e}_{c_P}(c^{n+1,e}_{h,P}, V^{n+1}_h, v^e_h) \\ F^{n,i}_{c_P}(c^{n+1,i}_{h,P}, V^{n+1}_h, v^i_h) \\ F^{n,e}_{c_N}(c^{n+1,e}_{h,N}, V^{n+1}_h, v^e_h) \\ F^{n,i}_{c_N}(c^{n+1,i}_{h,N}, V^{n+1}_h, v^i_h) \end{bmatrix} \tag{4.9}$$

We use the Newton algorithm to solve F equal to zero, and denote J the Jacobian matrix. We first define the differential of the F^n_V

$$J^n_V(V^{n+1}_h, v_h) = -\int_\Omega \eta_q \beta \nabla V^{n+1}_h \nabla v_h\, dX$$

$$J^n_{c^i_P,V}(c^{n+1,i}_{h,P}, v_h) = \int_{\Omega_i} c^{n+1,i}_{h,P} v_h\, dX$$

$$J^n_{c^e_P,V}(c^{n+1,e}_{h,P}, v_h) = \int_{\Omega_e} c^{n+1,e}_{h,P} v_h\, dX \tag{4.10}$$

$$J^n_{c^i_N,V}(c^{n+1,i}_{h,N}, v_h) = -\int_{\Omega_i} c^{n+1,i}_{h,N} v_h\, dX$$

$$J^n_{c^e_N,V}(c^{n+1,e}_{h,N}, v_h) = -\int_{\Omega_e} c^{n+1,e}_{h,N} v_h\, dX$$

Then, we define the differential of the $F^{n,e}_{c_P}$

$$J^{n,e}_{V,c_P}(c^{n+1,e}_{h,P}, V^{n+1}_h, v^e_h) = -\int_{\Omega_e} D_P \beta c^{n+1,e}_{h,P} \nabla V^{n+1}_h \nabla v^e_h \, dX$$

(4.11)
$$J^{n,e}_{c_P}(c^{n+1,e}_{h,P}, v^e_h) = \int_{\Omega_e} \left(\frac{c^{n+1,e}_{h,P}}{dt} v^e_h - D_P \nabla c^{n+1,e}_{h,P} \nabla v^e_h - D_P \beta c^{n+1,e}_{h,P} \nabla V^{n+1}_h \nabla v^e_h \right) dX$$

and the differential of the $F^{n,e}_{c_N}$

$$J^{n,e}_{V,c_N}(c^{n+1,e}_{h,N}, V^{n+1}_h, v^e_h) = \int_{\Omega_e} D_N \beta c^{n+1,e}_{h,N} \nabla V^{n+1}_h \nabla v^e_h \, dX$$

(4.12)
$$J^{n,e}_{c_N}(c^{n+1,e}_{h,N}, v^e_h) = \int_{\Omega_e} \left(\frac{c^{n+1,e}_{h,N}}{dt} v^e_h - D_N \nabla c^{n+1,e}_{h,N} \nabla v^e_h + D_N \beta c^{n+1,e}_{h,N} \nabla V^{n+1}_h \nabla v^e_h \right) dX.$$

Then, we define the differential of the $F^{n,i}_{c_P}$

$$J^{n,i}_{V,c_P}(c^{n+1,i}_{h,P}, V^{n+1}_h, v^i_h) = -\int_{\Omega_i} D_P \beta c^{n+1,i}_{h,P} \nabla V^{n+1}_h \nabla v^i_h \, dX$$

(4.13)
$$J^{n,i}_{c_P}(c^{n+1,i}_{h,P}, v^i_h) = \int_{\Omega_i} \left(\frac{c^{n+1,i}_{h,P}}{dt} v^i_h - D_P \nabla c^{n+1,i}_{h,P} \nabla v^i_h - D_P \beta c^{n+1,i}_{h,P} \nabla V^{n+1}_h \nabla v^i_h \right) dX$$

and the differential of the $F^{n,i}_{c_N}$

$$J^{n,i}_{V,c_N}(c^{n+1,i}_{h,N}, V^{n+1}_h, v^i_h) = \int_{\Omega_i} D_N \beta c^{n+1,i}_{h,N} \nabla V^{n+1}_h \nabla v^i_h \, dX$$

(4.14)
$$J^{n,i}_{c_N}(c^{n+1,i}_{h,N}, v^i_h) = \int_{\Omega_i} \left(\frac{c^{n+1,i}_{h,N}}{dt} v^i_h - D_N \nabla c^{n+1,i}_{h,N} \nabla v^i_h + D_N \beta c^{n+1,i}_{h,N} \nabla V^{n+1}_h \nabla v^i_h \right) dX.$$

Finally, we define the Jacobian matrix J by storing each differentials above, associated with the components of F (also previously defined in (4.9)):

(4.15)
$$J = \begin{bmatrix} J^n_V & J^n_{c^e_P,V} & J^n_{c^i_P,V} & J^n_{c^e_N,V} & J^n_{c^i_N,V} \\ J^{n,e}_{V,c_P} & J^{n,e}_{c_P} & 0 & 0 & 0 \\ J^{n,i}_{V,c_P} & 0 & J^{n,i}_{c_P} & 0 & 0 \\ J^{n,e}_{V,c_N} & 0 & 0 & J^{n,e}_{c_N} & 0 \\ J^{n,i}_{V,c_N} & 0 & 0 & 0 & J^{n,i}_{c_N} \end{bmatrix}$$

We recall that this numerical implementation does not guarantee the preservation of positivity in ionic concentrations, unfortunately. Still, the numerical applications of this chapter do not yield negative concentrations, as hoped for.

4.2. Applications

In the current section, we apply our FreeFEM++ implementation to investigate the dynamics of potential and ionic concentrations in dendrites, dealing with three different multi-domain geometries: a bifurcation in the dendritic tree, a large dendritic branch and a smaller branch with smooth angles. We also consider a mono-domain representing the interior domain of the bifurcation, similar to the domain described in 3, section 3.3.1.2, to compare the dynamics obtained with the DDFV framework, to the dynamics obtained with the FreeFEM++ approach.

In subsection 4.2.1.1, we consider the bifurcation under the multi-domain geometry, where we add the membrane thickness and the exterior media within one main rectangular box (Figure 1). The dendrite bifurcation has two thin branches connected to a larger one. We investigate the scenario of the summation of two simultaneous influxes of ions. We observe as previously a linear summation when the two signals meet, i.e., at the bifurcation point. A final point consists, here, in comparing our results with those coming from the intracellular medium, using FreeFEM++ and DDFV. Consequently, we compare both the DDFV and FreeFEM++ codes only on the intracellular medium (single-domain) with the multi-domain approach. To achieve this, we simulate the propagation of an influx of ions arriving at the edge of both small branches. We observe a slight difference between the numerical results. Additionally, our multi-domain approach allows us to observe the evolution of the potential within the membrane. In subsection 4.2.1.2, we investigate the signal summation within the membrane medium where we observe a linear summation too. Then, section 4.2.2 focuses on a single and large rectangular branch within one large rectangular box (Figure 2). In this study case, we first inspect the signal invasion on this geometry while increasing the number of influxes. We observe that when synaptic inputs are added, the potential increases significantly until reaching a strong excitability threshold. This leads us to believe that the co-localization of synaptic inputs play an important role during neuron depolarization.

In the last section 4.2.3, we examine the ion dynamics and the potential propagation near the membrane, i.e., in the boundary layer. The results reveal some fluctuations near the membrane wall, suggesting the presence of a boundary layer.

According to the previous chapter, we define a time t_0 for each simulation, corresponding to the time at which the concentration reaches its maximum value on the whole domain. Notice that the latter maximum is achieved in both space and time. The value of t_0 will be precised in each case. Besides, the reader may refer to Table 1, for a recall upon the electrodiffusion parameters at hand. Finally, in all the following sections, we present the numerical results for $c_{P,i}$ and $c_{P,e}$, since those for $c_{N,i}$ and $c_{N,e}$ are qualitatively similar.
Finally, throughout this chapter, we define X_h as the set of $\mathbb{P}_1$ vertices of $\mathcal{T}_h$ (see Definition 4.1), K^* a vertex of X_h^* and n as the time iteration in the simulations. Then, in a similar fashion as Chapter 3, we consider the set Z_S^* as the set of vertices on the segment S such that:

$$Z_S^* = \{K^* \in X_h^*, \text{ such that } K^* \cap S \neq \emptyset\} \tag{4.16}$$

and K_X^* as the unique vertex which is the closest of the point X.

In this section, the time step is equal to $dt = 5 \times 10^{-3}$, however the final time is equal to $T_f = 3$ sec for Section 4.2.1 and Section 4.2.2, and it is equal to $T_f = 1$ with $dt = 5 \times 10^{-5}$ sec for the last Section 4.2.3.

4.2.1. **Propagation of an influx of ions at a dendritic tree bifurcation.** In this section, we showcase application of the multi-domain approach with initial conditions from Table 1, aimed at studying the summation on a bifurcation domain, similar to Chapter 3. Then, we compare our results from the multi-domain with single-domain results under FreeFEM++ and DDFV. Finally, we observe electric propagation within the membrane, which is an advantage of the multi-domain approach. Our objective is therefore to model signal propagation at a bifurcation in the dendritic tree, where a large branch divides into two thinner ones (Fig. 5). We consider a multi-domain $\Omega = \Omega_i \cup \Omega_e \cup \Omega_m$ representing the intracellular (Ω_i), extracelllular (Ω_e) and membrane media (Ω_m). We model the influx of ions as a current $I(t)$ (eq. (4.7)) injected at the end of the two thin branches ($\partial\Omega_{r_i}^{\text{up}}$, $\partial\Omega_{r_e}^{\text{up}}$ for the upper branch, and $\partial\Omega_{r_i}^{\text{do}}$, $\partial\Omega_{r_e}^{\text{do}}$ for the lower branch, with $\partial\Omega_{r_i} = \partial\Omega_{r_i}^{\text{up}} \cup \partial\Omega_{r_i}^{\text{do}}$ and $\partial\Omega_{r_e} = \partial\Omega_{r_e}^{\text{up}} \cup \partial\Omega_{r_e}^{\text{do}}$, as defined in Section 4.1.2). We impose Dirichlet boundary conditions at the end of the large branch ($\partial\Omega_{i,D}$ on Fig. 5), to represent the connection with a larger dendrite (ionic reservoir, similarly to Chapter 3). We consider the initial and boundary conditions defined in Section 4.1.2 and Table 1. The different

lengths and nodes defining the domain Ω are described in Fig. 5, with the length values and nodes coordinates given in Table 2.

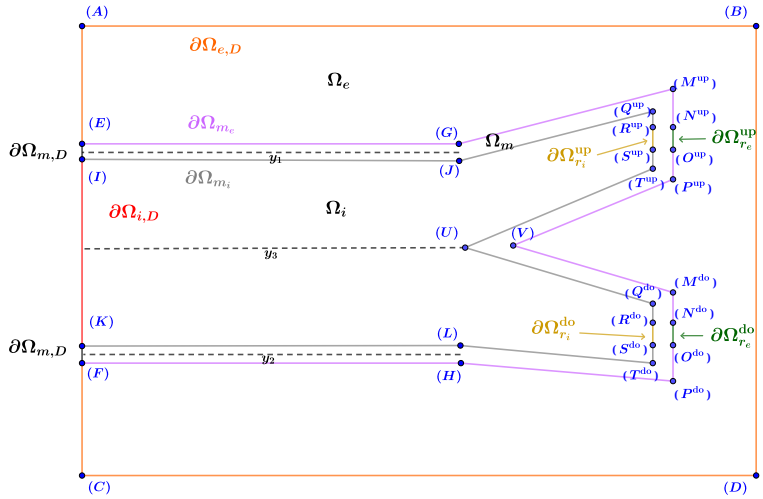

FIGURE 5. Illustration of the bifurcation multi-domain $\Omega = \Omega_i \cup \Omega_m \cup \Omega_e$. Note that the domain is not to scale. The coordinates of each node are given in Table 2 accordingly. We consider the lines $y_1 = [-22, 11] \times \{2.0035\}$, $y_2 = [-22, 11] \times \{-0.0035\}$, $y_3 = [-22, 11] \times \{1\}$.

r	0.12 µm	Radius of $\partial\Omega_{r_i}, \Omega_{r_e}$ in Figure 5 for Ω
(A)	(-22,10)	Position (x,y) of node A
(B)	(20,10)	Position (x,y) of node B
(C)	(-22,-8)	Position (x,y) of node C
(D)	(20,-8)	Position (x,y) of node D
(E)	(-22,2.007)	Position (x,y) of node E
(F)	(-22,-0.007)	Position (x,y) of node F
(G)	(11,2.007)	Position (x,y) of node G
(H)	(11,-0.007)	Position (x,y) of node H
(I)	(-22,2)	Position (x,y) of node I
(J)	(11,2)	Position (x,y) of node J
(K)	(-22,0)	Position (x,y) of node K
(L)	(11,0)	Position (x,y) of node L
(M)	$(15.007, 5.012)^{\mathrm{up}}, (15.007, -2.4)^{\mathrm{do}}$	Position (x,y) of node M
(N)	$(15.007, 4.76)^{\mathrm{up}}, (15.007, -2.64)^{\mathrm{do}}$	Position (x,y) of node N
(O)	$(15.007, 4.64)^{\mathrm{up}}, (15.007, -2.76)^{\mathrm{do}}$	Position (x,y) of node O
(P)	$(15.007, 4.4)^{\mathrm{up}}, (15.007, -3.012)^{\mathrm{do}}$	Position (x,y) of node P
(Q)	$(15, 5)^{\mathrm{up}}, (15, -2.4)^{\mathrm{do}}$	Position (x,y) of node Q
(R)	$(15, 4.76)^{\mathrm{up}}, (15, -2.64)^{\mathrm{do}}$	Position (x,y) of node R
(S)	$(15, 4.64)^{\mathrm{up}}, (15, -2.76)^{\mathrm{do}}$	Position (x,y) of node S
(T)	$(15, 4.4)^{\mathrm{up}}, (15, -3)^{\mathrm{do}}$	Position (x,y) of node T
(U)	(11,1)	Position (x,y) of node U
(V)	(11.007,1)	Position (x,y) of node V

TABLE 2. Geometric parameters for multi-domain Ω, representing a dendritic bifurcation (Figure 5).

As mentioned earlier, our model considers that the end of the large branch is connected with a wider one, called an ionic reservoir, such that the ionic concentrations at $\partial\Omega_{i,D}$ are constant. The distance between this ionic reservoir and the bifurcation influences the dynamics of potential and ionic

concentrations everywhere on Ω_i (see Chapter 3). In the following, we consider the longest branch where the distance between (I) and (J) is equal to 33 µm , as in Chapter 3.

4.2.1.1. ***Bifurcation within the multi-domain.*** We now consider the multi-domain model representing the bifurcation. The objective is to observe how the signal sums up under this model where the potential is modeled with continuity conditions at the membrane interfaces, as explained in Section 4.1. To achieve this, we use a triangular mesh with 461377 cells and a mesh size with $\{h_{\min} = 0.002, h_{\max} = 8.4\}$. The electrodiffusion parameters are taken from Table 1. In particular, the potential is initially -75mV within the interior medium and the membrane. In Figure 6, we plot the discrete values $\left(c^{21,i}_{h,P} - c^0_{P,i}\right)$ and $\left(c^{21,e}_{h,P} - c^0_{P,e}\right)$, corresponding to the concentration $c_{P,i} - c^0_{P,i}$ and $c_{P,e} - c^0_{P,e}$ at time $t_0 = 0.105$ sec. We observe that the maximum (resp. minimum) of the solution, $A_i = \max_{K^* \in \mathcal{T}_h} \left(c^{21,i}_{K^*,P} - c^0_{P,i}\right) = 119.7$ mM (resp. $A_e = \min_{K^* \in \mathcal{T}_h} \left(c^{21,e}_{K^*,P} - c^0_{P,e}\right) = -25.98$ mM), is reached at the injection boundary $\partial\Omega_{r_i}$ (resp. $\partial\Omega_{r_e}$). At the injection boundary, we observe the transfer of ionic concentration from the extracellular medium Ω_e to the intracellular medium Ω_i. As we move from the thinner branches to the larger one, the solution decreases, reaching the value $c^0_{P,i}$ at the Dirichlet boundary $\partial\Omega_{i,D}$.

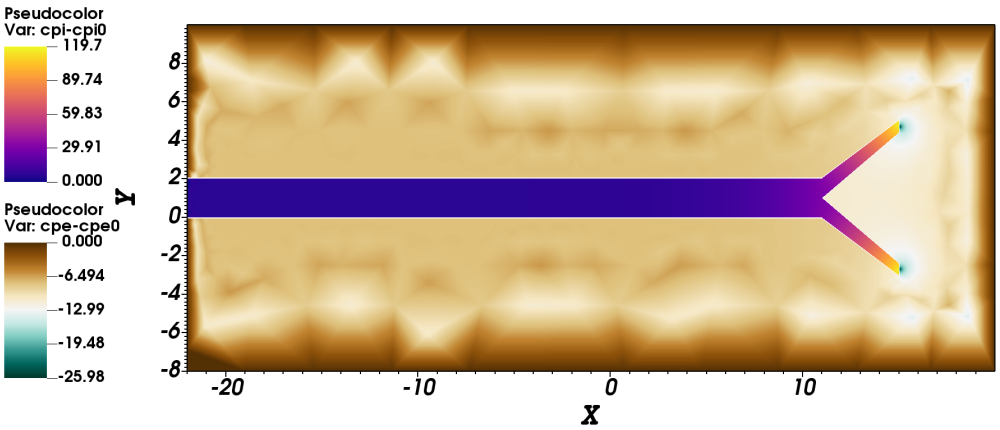

FIGURE 6. Discrete values $\left(c^{21,i}_{h,P} - c^0_{P,i}\right)$ and $\left(c^{21,e}_{h,P} - c^0_{P,e}\right)$ of the concentration $c_{P,i} - c^0_{P,i}$ and $c_{P,e} - c^0_{P,e}$ at time $t_0 = 0.105$ sec.

We compare ionic concentration and potential dynamics in two different scenarios as in Chapter 3. We recall that in scenario **1**, only the upper branch receives the influx ($(\partial\Omega^{\text{up}}_{r_i}, \partial\Omega^{\text{up}}_{r_e})$ in Fig. 5), whereas in scenario **2**, there is an influx of ions at both ends of the two thin branches ($(\partial\Omega^{\text{up}}_{r_i}, \partial\Omega^{\text{up}}_{r_e})$ and $(\partial\Omega^{\text{do}}_{r_i}, \partial\Omega^{\text{do}}_{r_e})$ in Fig. 5). In scenario **1**, $(\partial\Omega^{\text{do}}_{r_e}, \partial\Omega^{\text{do}}_{r_i})$ is turned into an homogeneous Neumann boundary condition. Once again, we will call a branch "active" if it receives an influx of ions, and "inactive" if it does not. We run simulations in domain of Fig. 5. In Fig. 7, we compare the concentration $\left(c^{21,i}_{K^*,P} - c^0_{P,i}\right)^{\mathbf{s}}_{K^* \in Z^*_{y_3}}$ and potential $\left(V^{21}_{K^*}\right)^{\mathbf{s}}_{K^* \in Z^*_{y_3}}$ at time $t_0 = 0.105$ sec, for the two scenarios $\mathbf{s} = \mathbf{1}$ and $\mathbf{s} = \mathbf{2}$.

In both graphs, the blue curve represents the scenario **1** with one active branch, while the red curve corresponds to scenario **2** where the two branches are active. We observe that, on line $y_3 = [-22, 11] \times \{1\}$ (see Fig. 5), the concentration and potential values in the scenario **2** (red) are twice the values in scenario **1** (blue).

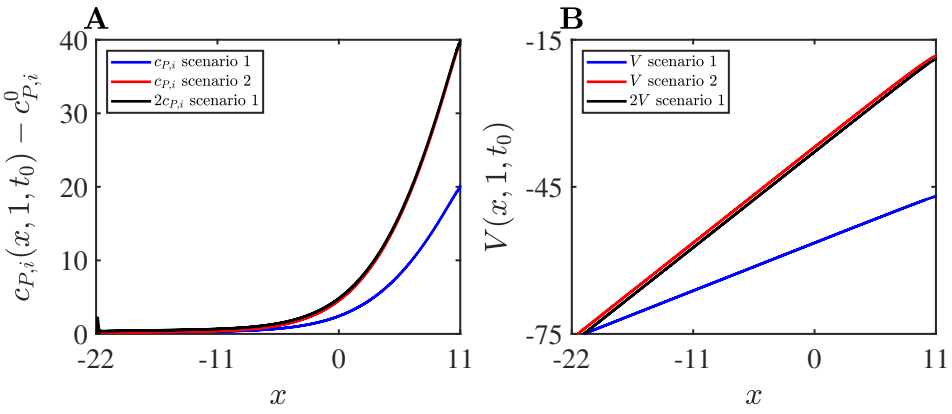

FIGURE 7. Evolution of the concentration and potential dynamics on line y_3 for the two scenarios **1** and (resp. **2**), with one (resp. two) active branch(es). **A**: Discrete values $\left(c^{21,i}_{K^*,P}-c^0_{P,i}\right)^{\mathbf{s}}_{K^*\in Z^*_{y_3}}$, at peak time $t_0=0.105$ sec, for the two scenarios: $\mathbf{s}=\mathbf{1}$ in blue and $\mathbf{s}=\mathbf{2}$ in red. In black, we plot $2\left(c^{21,i}_{K^*,P}-c^0_{P,i}\right)^{\mathbf{1}}_{K^*\in Z^*_{y_3}}$. **B**: Discrete values $\left(V^{21}_{K^*}\right)^{\mathbf{s}}_{K^*\in Z^*_{y_3}}$, at peak time $t_0=0.105$ sec, for $\mathbf{s}=\mathbf{1}$ in blue and $\mathbf{s}=\mathbf{2}$ in red. In black, we plot $2\left(V^{21}_{K^*}\right)^{\mathbf{1}}_{K^*\in Z^*_{y_3}}$.

In Fig. 8, we plot the time evolution of the concentration $c_{P,i}-c^0_{P,i}$ and the potential V at position (U), i.e. $\left(c^{n,i}_{K^*_U,P}-c^0_{P,i}\right)^{\mathbf{s}}_{n=0,\cdots,N}$ and $\left(V^n_{K^*_U}\right)^{\mathbf{s}}_{n=0,\cdots,N}$ for the two scenarios $\mathbf{s}=\mathbf{1}$ and $\mathbf{s}=\mathbf{2}$. We recall that K^*_U is the unique vertex which is the closest of the point (U). We observe that the potential and concentration dynamics in scenario **2** is twice the dynamics in scenario **1**.

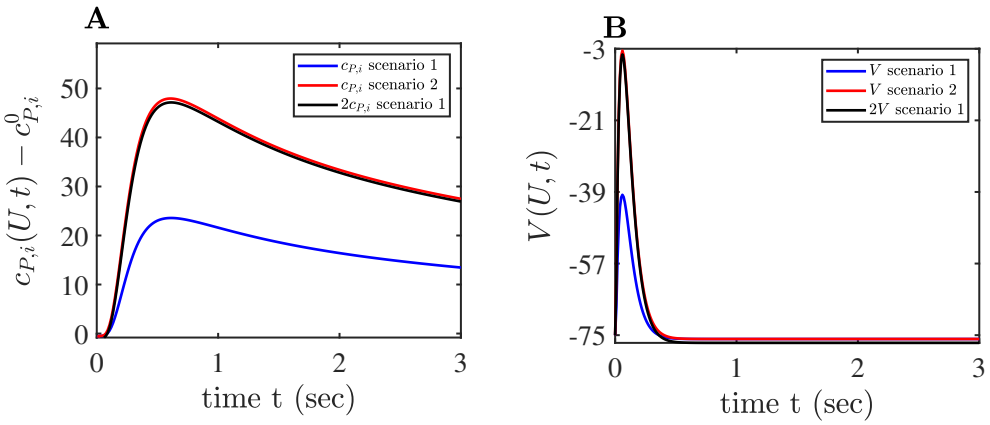

FIGURE 8. Evolution of the concentration and potential dynamics at point (U) for the two scenarios **1** and **2**. **A**: Discrete values $\left(c^{n,i}_{K^*_U,P}-c^0_{P,i}\right)^{\mathbf{s}}_{n=0,\cdots,N}$, for the two scenarios: $\mathbf{s}=\mathbf{1}$ in blue and $\mathbf{s}=\mathbf{2}$ in red. In black, we plot $2\left(c^{n,i}_{K^*_U,P}-c^0_{P,i}\right)^{\mathbf{1}}_{n=0,\cdots,N}$. Notice that $\max_{n=0,\cdots,N}\left|\left(c^{n,i}_{K^*_U,P}-c^0_{P,i}\right)^{\mathbf{2}}-2\left(c^{n,i}_{K^*_U,P}-c^0_{P,i}\right)^{\mathbf{1}}\right|=0.98$. **B**: Discrete values $\left(V^n_{K^*_U}\right)^{\mathbf{s}}_{n=0,\cdots,N}$ for $\mathbf{s}=\mathbf{1}$ in blue and $\mathbf{s}=\mathbf{2}$ in red. In black, we plot $2\left(V^n_{K^*_U}\right)^{\mathbf{1}}_{n=0,\cdots,N}$. Notice that $\max_{n=0,\cdots,N}\left|\left(V^n_{K^*_U}\right)^{\mathbf{2}}-2\left(V^n_{K^*_U}\right)^{\mathbf{1}}\right|=1.1$.

We finally compute the absolute differences between the concentration $\left(\left(c^{n,i}_{K^*,P}-c^0_{P,i}\right)^{\mathbf{2}}_{K^*\in Z^*_{y_3}}\right)_{n=0,\cdots,N}$ and twice the concentration $\left(\left(c^{n,i}_{K^*,P}-c^0_{P,i}\right)^{\mathbf{1}}_{K^*\in Z^*_{y_3}}\right)_{n=0,\cdots,N}$ on the line y_3 and for $t\in[0,T_f]$. In other words, we are interested in the following quantity:

$$e := \left| \left(\left(c^{n,i}_{K^*,P} - c^0_{P,i} \right)^{\mathbf{2}}_{K^* \in Z^*_{y_3}} \right)_{n=0,\cdots,N} - 2 \left(\left(c^{n,i}_{K^*,P} - c^0_{P,i} \right)^{\mathbf{1}}_{K^* \in Z^*_{y_3}} \right)_{n=0,\cdots,N} \right|. \tag{4.17}$$

The difference e remains below 0.3 mM up to $x = 0$ µm and oscillates between 0 mM and 0.6 mM up to $x = 11$ µm (bifurcation at point (U)). Concerning the potential, this difference remains close to 1 mV along y_3. Still, we achieve coherent summation results consistent with experimentally observed linear summation of signals in passive dendrites [22].

Afterwards, our aim is to compare the results for the bifurcation obtained with our FreeFEM++ implementation to the DDFV method for spatial discretization (Chapter 3). To achieve this, we will consider three distinct cases:

- the multi-domain approach under FreeFEM++, denoted by case (1),
- the single-domain approach under FreeFEM++, denoted by case (2),
- the single-domain approach under DDFV method, denoted by case (3).

Recall that here both branches are actives. For the case (2) and (3), we only consider Ω_i, the intracellular domain of the bifurcation and we impose homogeneous Neumann boundary conditions to the potential as describe in Chapter 3. For case (2), we choose a mesh size with $\{h_{\min} = 0.006, h_{\max} = 0.31\}$, containing 7016 triangular cells. For case (3), we use the same mesh as Chapter 3, the mesh has 7863 triangular cells and the mesh size is between $\{h_{\min} = 0.05, h_{\max} = 0.4\}$. For all cases, we took initial conditions from Table 1. Furthermore, in order to clearly distinguish between the discrete values of the three cases, we denote:

- $(c^{dt,i}_{h,P}, V^{dt,i}_h)$ for case (1),
- $(c^{dt}_{h,P}, V^{dt}_h)$ for case (2),
- $(c^{\mathcal{T},dt}_P, V^{\mathcal{T},dt})$ for case (3).

In Figure 9, we plot the FreeFEM discrete values $\left(c^{21}_{h,P} - c^0_{P,i} \right)$, from case (2), corresponding to the concentration $c_P - c^0_{P,i}$ at time $t_0 = 0.105$ sec. We observe that the maximum of the FreeFEM discrete solution for (2), $A_2 = \max_{K^* \in \mathcal{T}_h} \left(c^{21}_{K^*,P} - c^0_{P,i} \right) = 119.7$ mM, is reached at the injection boundary $\partial\Omega_{r_i}$. Along x, the solution decreases to reach the value $c^0_{P,i}$ at the Dirichlet boundary $\partial\Omega_{i,D}$. We obtain the same values as A_i, the maximum from case (1). In case (3), we found a value of $A_3 = 93.8$ mM, which is smaller. We believe this variation is due to the difference in mesh resolution near the inputs, ($h_{\min} = 0.002$ for (1), $h_{\min} = 0.006$ for (2) and $h_{\min} = 0.05$ for (3)). Unfortunately, in FreeFEM++, we were unable to create exactly the same mesh as that used in the DDFV approach (3), especially near the inputs.

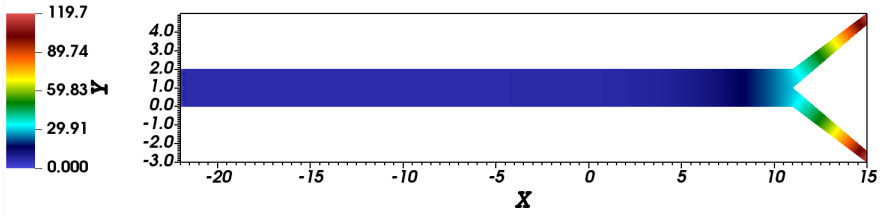

FIGURE 9. FreeFEM discrete values $\left(c^{21}_{h,P} - c^0_{P,i} \right)$ of the concentration $c_{P,i} - c^0_{P,i}$ at time $t_0 = 0.105$ sec, from (2).

Then, we consider the line y_3 and the set of vertices $Z^*_{y_3}$ (see (4.16)). We compare the evolution of concentrations and potential along y_3 at time t_0, for all cases. For scenario $\mathbf{s} = \mathbf{2}$, we retrieve the red curves of discrete values $\left(c^{21,i}_{K^*,P} - c^0_{P,i}\right)^{\mathbf{2}}_{K^*\in Z^*_{y_3}}$ and $\left(V^{21,i}_{K^*}\right)^{\mathbf{2}}_{K^*\in Z^*_{y_3}}$, at peak time $t_0 = 0.105$ sec, from Figure 7, corresponding to case (1). Then, we compare these curves with those obtained from simulations on the single-domain, i.e, discrete values $\left(c^{21}_{K^*,P} - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}}$ and $\left(V^{21}_{K^*}\right)_{K^*\in Z^*_{y_3}}$ for case (2) and $\left(c^{K^*,21}_P - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}}$ and $\left(V^{K^*,21}\right)_{K^*\in Z^*_{y_3}}$ for case (3). We plot these results on the same graphs in Figure 10, with the multi-domain results in red for case (1), the single-domain results in blue for case (2) and in green for case (3). We observe that in the FreeFEM++ cases (1) and (2), the concentration and potential curves overlap. However, in the DDFV setting, which is case (3), the curves are slightly below those of cases (1) and (2).

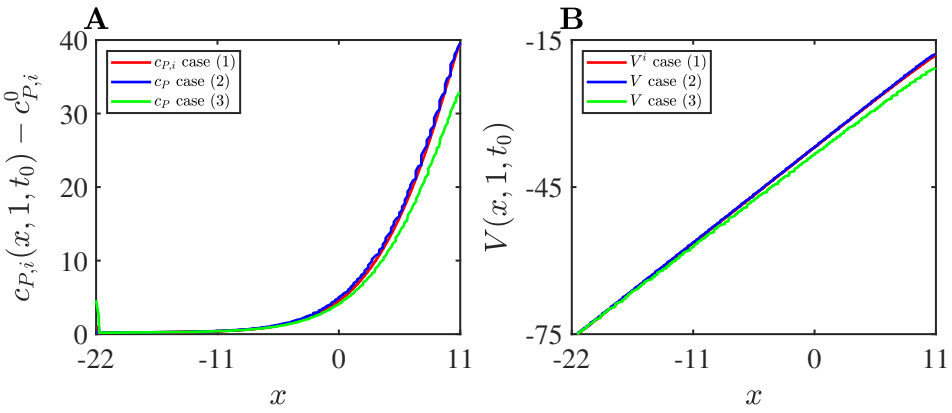

FIGURE 10. Evolution of the concentration and potential dynamics on line y_3 for all cases, at peak time $t_0 = 0.105$ sec. **A**: FreeFEM discrete values $\left(c^{21,i}_{K^*,P} - c^0_{P,i}\right)^{\mathbf{2}}_{K^*\in Z^*_{y_3}}$ in red for case (1). FreeFEM discrete values $\left(c^{21}_{K^*,P} - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}}$ in blue for case (2). DDFV discrete values $\left(c^{K^*,21}_P - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}}$ in green for case (3). **B**: FreeFEM discrete values $\left(V^{21,i}_{K^*}\right)^{\mathbf{2}}_{K^*\in Z^*_{y_3}}$ in red for case (1). FreeFEM discrete values $\left(V^{21}_{K^*}\right)_{K^*\in Z^*_{y_3}}$ in blue for case (2). DDFV discrete values $\left(V^{K^*,21}\right)_{K^*\in Z^*_{y_3}}$ in green for case (3).

As a result, we calculate the differences between the curves of Figure 10 pairwise, as displayed in Figure 11 below. For completeness, the quantities appearing in Figure 11 are formally given by:

$$e_{c_P,12} := \left|\left(c^{21,i}_{K^*,P} - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}} - \left(c^{21}_{K^*,P} - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}}\right|,$$
$$e_{c_P,13} := \left|\left(c^{21,i}_{K^*,P} - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}} - \left(c^{K^*,21}_P - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}}\right|,$$
$$e_{c_P,23} := \left|\left(c^{21}_{K^*,P} - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}} - \left(c^{K^*,21}_P - c^0_{P,i}\right)_{K^*\in Z^*_{y_3}}\right|.$$

In other words, $e_{c_P,12}$ is the error between the c_P-concentrations of cases (1) and (2), while $e_{c_P,13}$ is that between the c_P-concentrations of case (1) and (3), and so on. The above quantities are analogously defined for the potential curves of Figure 10**B**, with according errors denoted by:

$$e_{V,12} := \left|\left(V^{21,i}_{K^*}\right)_{K^*\in Z^*_{y_3}} - \left(V^{21}_{K^*}\right)_{K^*\in Z^*_{y_3}}\right|,$$
$$e_{V,13} := \left|\left(V^{21,i}_{K^*}\right)_{K^*\in Z^*_{y_3}} - \left(V^{K^*,21}\right)_{K^*\in Z^*_{y_3}}\right|,$$
$$e_{V,23} := \left|\left(V^{21}_{K^*}\right)_{K^*\in Z^*_{y_3}} - \left(V^{K^*,21}\right)_{K^*\in Z^*_{y_3}}\right|.$$

From Figure 11**A** below, one may observe that the error $e_{c_P,13}$ for the c_P-concentrations between cases (1) and (3) increases till the bifurcation point, i.e the merging point between the two branches (located at $x = 11$). The same observation holds for $e_{c_P,23}$. In other words, the c_P values computed

based on the multi-domain with FreeFEM++ and the DDFV single-domain differs. This is also the case when considering the single-domain with both FreeFEM++ and DDFV. In contrast, the error $e_{c_P,12}$ is rather flat and approaching zero. The latter means that the multi-domain modeling with FreeFEM++ yields similar values for c_P as the single-domain with FreeFEM++.

From Figure 11**B**, we further remark that the tendency for the potential errors $e_{V,12}$, $e_{V,13}$ and $e_{V,23}$ is analogous to the above behavior of respective c_P-errors.

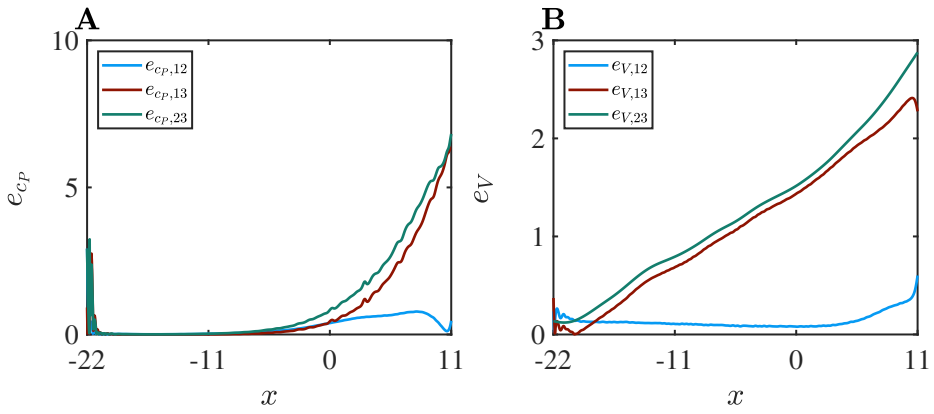

FIGURE 11. Absolute differences between the results from all cases. **A**: $e_{c_{P,12}}$ in blue with $\max(e_{c_{P,12}}) = 2.91$. $e_{c_{P,13}}$ in brown with $\max(e_{c_{P,13}}) = 6.35$. $e_{c_{P,23}}$ in green with $\max(e_{c_{P,23}}) = 6.81$. **B**: $e_{V,12}$ in blue with $\max(e_{V,12}) = 0.61$. $e_{V,13}$ in brown with $\max(e_{V,13}) = 2.41$. $e_{V,23}$ in green with $\max(e_{V,23}) = 2.87$.

We summarize the results of the three approaches in Table 3, where we present the percentages of these differences on y_3.

Error	$e_{c_{P,12}}$	$e_{V,12}$	$e_{c_{P,13}}$	$e_{V,13}$	$e_{c_{P,23}}$	$e_{V,23}$
Maximum	2.91	0.61	6.35	2.41	6.81	2.87
Percentage	8 %	1 %	19 %	4 %	17 %	5 %

TABLE 3. Summary of the errors for all relative cases (1), (2) and (3): Maximum values and according percentages with respect to Fig. 11.

The relatively high differences 19% (resp. 17%) for $e_{c_P,13}$ (resp. $e_{c_P,23}$), can be explained by the difference in meshes used within each framework. Indeed, near the inputs, the FreeFEM++ mesh of both cases (1) and (2), is 10 times finer than the mesh employed in Chapter 3 that is in case (3). For the first two cases with FreeFEM++, we were unable to create a coarser mesh near the wall where we model the inputs, otherwise we would have had too few cells. Still, while we do remark differences between the overall results of Table 3, we qualitatively observe similar behaviors for all pair of comparative cases of study. Regarding the two studies under FreeFEM++, cases (1) and (2), we note differences of 8 % (resp. 1 %) for the concentration c_P (resp. the potential V). For this test case, there is no significant differences in the FreeFEM discrete solution between the multi and mono domain. This suggests that the homogeneous Neumann boundary condition on the potential is not perturbing much the solution.

4.2.1.2. ***Signal summation within the membrane.*** An interesting advantage of the multi-domain approach is the ability to observe events within the membrane. To achieve this, we selected multiple points at different positions within the membrane and studied the potential behavior therein. It is worth noting that, once again, only the potential exists within the membrane. In Figure 12, we depict the potential evolution for both scenarios within the membrane. We plot in **A** the potential on y_1, at the time $t = 0.105$ sec, i.e the discrete values $\left(V^{21}_{K^*}\right)^{\mathbf{s}}_{K^* \in Z^*_{y_1}}$. In addition, we plot in **B** $\left(V^n_{K^*_{\overline{GJ}}}\right)^{\mathbf{s}}_{n=0,\cdots,N}$ at the midpoint between points (G) and (J), denoted as $\overline{GJ}$, for both scenarios.

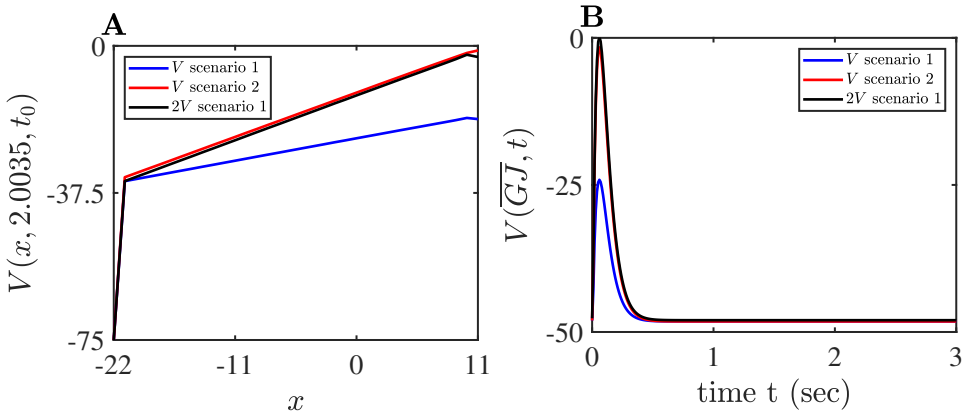

FIGURE 12. Evolution of the potential dynamics for both scenarios **1** and **2**. **A**: Discrete values $\left(V_{K^*}^{21}\right)_{K^*\in Z_{y_1}^*}^{\mathbf{s}}$ on y_1 for $\mathbf{s}=\mathbf{1}$ in blue and $\mathbf{s}=\mathbf{2}$ in red, at the time $t = 0.105$ sec. In black, we plot $2\left(V_{K^*}^{21}\right)_{K^*\in Z_{y_1}^*}^{\mathbf{1}}$. **B**: Discrete values $\left(V_{K^*_{\overline{GJ}}}^{n}\right)_{n=0,\cdots,N}^{\mathbf{s}}$ for $\mathbf{s}=\mathbf{1}$ in blue and $\mathbf{s}=\mathbf{2}$ in red, at point $\overline{GJ}$. In black, we plot $2\left(V_{K^*_{\overline{GJ}}}^{n}\right)_{n=0,\cdots,N}^{\mathbf{1}}$.

We observe that linear summation of potential also occurs within the membrane. Furthermore, an important observation is that, spatially, the potential dynamics are the same, as the values of V on lines y_1 and y_2 are almost equal, even when one branch is active. This result appears to be in line with the fact that the potential propagates along the membrane. We also observe on graph **A** a sudden change (from approximately -37.5 mV to -75 mV) near the Dirichlet boundary ($x = -22$). This abrupt change is due to the membrane being very thin, making it difficult to refine near the Dirichlet boundary, where we imposed -75 mV.
Furthermore, we can conclude that to obtain results such as linear summation, the single-domain approach is sufficient. However, it remains limited since we have no information within the membrane and outside the dendrite. Thus, there is a trade-off between the time cost of multi-domain implementation, its complexity, and obtaining information inside the membrane.

4.2.2. **Effect of influxes on a large rectangular branch of dendrite.** The present setup is dedicated to investigating the signal invasion on the multi-domain by increasing the number of ionic influxes from $i = 1$ to $i = 7$, with i the number of influxes. This choice of the number of ion influxes is determined based on considerations related to numerical and biological aspects, elucidated in detailed explanations, further in this section.

The initial scenario is denoted as $\mathbf{s} = \mathbf{1}$, corresponding to a single influx ($i = 1$) at position 1. Subsequently, with the introduction of each additional influx onto the membrane, we label the scenario as $\mathbf{s} = i$ for i supplementary influxes at distinct positions i. We demonstrate that the potential significantly increases as the number of influxes increases as well, resulting in an important depolarisation. About the ionic dynamics, we observe less significant changes in their quantity during the numerical simulations, in comparison to the potential.

In what follows, we use a triangular mesh containing 251357 cells and a mesh size with $\{h_{\min} = 0.002, h_{\max} = 8.4\}$. We detail the domain dimensions with the coordinates of each point in Table 4 and Table 5.

In this model, similar to the active and inactive branches (section 4.2.1), we define locations on the membrane where ion influxes occur. Depending on the scenario, these ion influxes are either active or inactive. We model seven locations corresponding to the entries of ions, which entails defining: $\partial\Omega_{r_i} = \partial\Omega_{r_i}^1 \cup \partial\Omega_{r_i}^2 \cup \partial\Omega_{r_i}^3 \cup \partial\Omega_{r_i}^4 \cup \partial\Omega_{r_i}^5 \cup \partial\Omega_{r_i}^6 \cup \partial\Omega_{r_i}^7$ and $\partial\Omega_{r_e} = \partial\Omega_{r_e}^1 \cup \partial\Omega_{r_e}^2 \cup \partial\Omega_{r_e}^3 \cup \partial\Omega_{r_e}^4 \cup \partial\Omega_{r_e}^5 \cup \partial\Omega_{r_e}^6 \cup \partial\Omega_{r_e}^7$. We illustrate these boundaries in Figure 13 below. We choose to define variable distances between them

modelling the existing distances between ion influxes within dendrites. These distances vary between 1 et 10 µm in the dendrite arborization, as commonly encountered in the biological literature [83].

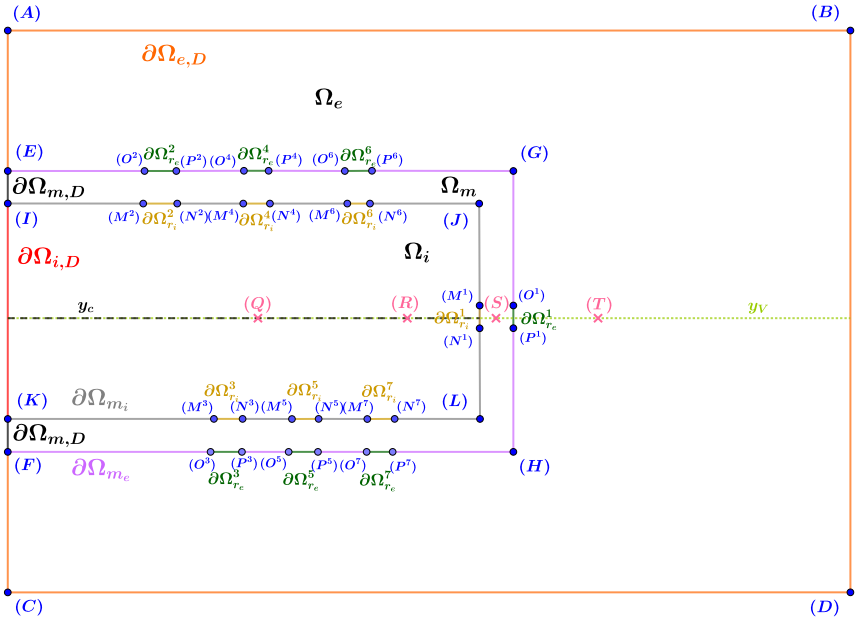

FIGURE 13. Geometry of the multi-domain containing a large rectangular dendrite. The domain is not to scale. The specific lines $y_c = [0, 49.993] \times \{55\}$ and $y_V = [0, 80] \times \{55\}$, where we monitor concentration and potential dynamics are plotted in dashed black and dashed light green, respectively. The dimensions are provided in Table 4 and Table 5.

(A)	(0,80)	Position (x,y) of node A
(B)	(80,80)	Position (x,y) of node B
(C)	(0,30)	Position (x,y) of node C
(D)	(80,30)	Position (x,y) of node D
(E)	(0,58.007)	Position (x,y) of node E
(F)	(0,52)	Position (x,y) of node F
(G)	(50,58.007)	Position (x,y) of node G
(H)	(50,52)	Position (x,y) of node H
(I)	(0,58)	Position (x,y) of node I
(J)	(49.993,68)	Position (x,y) of node J
(K)	(0,52.007)	Position (x,y) of node K
(L)	(49.993,52.007)	Position (x,y) of node L
(M)	$(49.993,55.1)^{\mathbf{1}}$	Position (x,y) of node $M^{\mathbf{1}}$
(N)	$(49.993,54.9)^{\mathbf{1}}$	Position (x,y) of node $N^{\mathbf{1}}$
(O)	$(50,55.1)^{\mathbf{1}}$	Position (x,y) of node $O^{\mathbf{1}}$
(P)	$(50,54.9)^{\mathbf{1}}$	Position (x,y) of node $P^{\mathbf{1}}$
(Q)	(35,55)	Position (x,y) of node Q
(R)	(45,55)	Position (x,y) of node R
(S)	(49.996,55)	Position (x,y) of node S
(T)	(55,55)	Position (x,y) of node T

TABLE 4. Geometric parameters for the multi-domain, representing the large dendrite.

(M)	$(30,58)^{\mathbf{2}}, (25.8,52.007)^{\mathbf{3}}$	Position (x,y) of node $M^{\mathbf{2}}$,$M^{\mathbf{3}}$
(M)	$(35,58)^{\mathbf{4}}, (31.8,52.007)^{\mathbf{5}}$	Position (x,y) of node $M^{\mathbf{4}}$,$M^{\mathbf{5}}$
(M)	$(39,58)^{\mathbf{6}}, (39.8,52.007)^{\mathbf{7}}$	Position (x,y) of node $M^{\mathbf{6}}$,$M^{\mathbf{7}}$
(N)	$(30.2,58)^{\mathbf{2}}, (26,52.007)^{\mathbf{3}}$	Position (x,y) of node $N^{\mathbf{2}}$,$N^{\mathbf{3}}$
(N)	$(35.2,58)^{\mathbf{4}}, (32,52.007)^{\mathbf{5}}$	Position (x,y) of node $N^{\mathbf{4}}$,$N^{\mathbf{5}}$
(N)	$(39.2,58)^{\mathbf{6}}, (40,52.007)^{\mathbf{7}}$	Position (x,y) of node $N^{\mathbf{6}}$,$N^{\mathbf{7}}$
(O)	$(30,58.007)^{\mathbf{2}}, (25.8,52)^{\mathbf{3}}$	Position (x,y) of node $O^{\mathbf{2}}$,$O^{\mathbf{3}}$
(O)	$(35,58.007)^{\mathbf{4}}, (31.8,52)^{\mathbf{5}}$	Position (x,y) of node $O^{\mathbf{4}}$,$O^{\mathbf{5}}$
(O)	$(39,58.007)^{\mathbf{6}}, (39.8,52)^{\mathbf{7}}$	Position (x,y) of node $O^{\mathbf{6}}$,$O^{\mathbf{7}}$
(P)	$(30.2,58.007)^{\mathbf{2}}, (26,52)^{\mathbf{3}}$	Position (x,y) of node $P^{\mathbf{2}}$,$P^{\mathbf{3}}$
(P)	$(35.2,58.007)^{\mathbf{4}}, (32,52)^{\mathbf{5}}$	Position (x,y) of node $P^{\mathbf{4}}$,$P^{\mathbf{5}}$
(P)	$(39.2,58.007)^{\mathbf{6}}, (40,52)^{\mathbf{7}}$	Position (x,y) of node $P^{\mathbf{6}}$,$P^{\mathbf{7}}$

TABLE 5. Geometric parameters of influxes for $\mathbf{s} = \{\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6},\mathbf{7}\}$.

In our context of rectangular dendritic branch with three domains (intracellular, extracellular, and membrane), ion influxes are incrementally added, reaching up to 7 influxes. As stated earlier, we call Scenario **i** the case where we consider i single influx(es), $i = 1,\dots,7$. We run 2D simulations for all scenarios for which we provide a non-exhaustive numerical analysis, but we depict exclusively scenario **1** and scenario **7** for visualization. We focus on the dynamics of potential V in Fig. 14 - for $\mathbf{s} = \mathbf{1}$ in **A** and $\mathbf{s} = \mathbf{7}$ in **B** - and ion concentrations $c_{P,i}$, $c_{P,e}$ in Fig. 15 - for $\mathbf{s} = \mathbf{1}$ in **A** and $\mathbf{s} = \mathbf{7}$ in **B**. We show simulations at respective peak time t_0, which we specify and denote as t_0^c for the concentrations and t_0^V for the potential. We observe a significant difference between the potential in scenario **1** (Fig. 14**A**) and scenario **7** (Fig. 14**B**). The maximum potential for scenario **1** is 1.83 mV, whereas for scenario **7**, it skyrocketed till 81.14 mV. Additionally, we notice that for $\mathbf{s} = \mathbf{1}$, the potential within the dendrite remains overall between -37 mV and -75 mV, which is insufficient to reach a significant depolarization. Moreover, for $\mathbf{s} = \mathbf{7}$, we surpass this value substantially within the dendrite, with a peak at 81.14 mV located near the influx $i = 1$. In Figure 15, we do not observe significant differences between the concentrations in both scenarios **1** and **7**. Indeed, the maximum concentrations within the interior of the dendrite are 44.13 mM and 51.74 mM for $\mathbf{s}$=**1** and $\mathbf{s}$=**7**, respectively, which is less pronounced compared to the potential. However, approximately half of the branch exhibits a different dynamic.

Indeed, the ion concentration $c_{P,i}$, which was elevated only near the influx for scenario **1**, increases spatially between approximately $x = 20$ and $x = 50$ µm in scenario **7**. In the same way, we also note the absence of significant changes in the exterior concentrations.

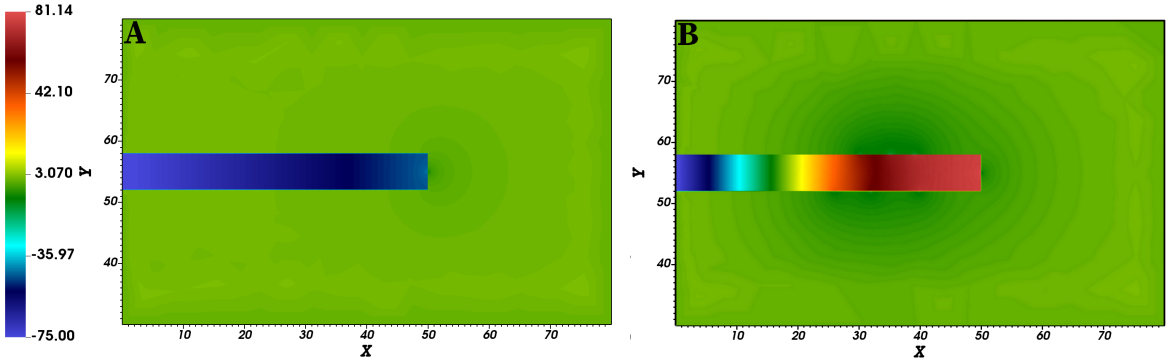

FIGURE 14. Spatial variation of the potential in the domain with scenario $\mathbf{i} = \mathbf{1}, \mathbf{7}$. Discrete values $\left(V_h^{12}\right)^{\mathbf{i}}$ of the potential V at time $t_0^V = 0.06$ sec.

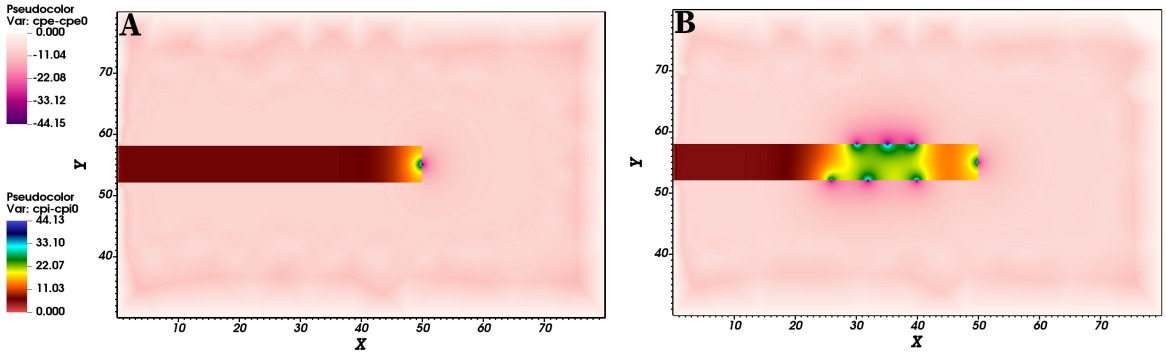

FIGURE 15. Spatial variation of the concentrations in the domain with scenario $\mathbf{i} = \mathbf{1}, \mathbf{7}$. Discrete values $\left(c_{h,P}^{21,i} - c_{P,i}^0\right)^{\mathbf{i}}$ and $\left(c_{h,P}^{21,e} - c_{P,e}^0\right)^{\mathbf{i}}$ of the concentration $c_{P,i} - c_{P,i}^0$ and $c_{P,e} - c_{P,e}^0$ at time $t_0^c = 0.105$ sec.

Following the visualizations above, we compare numerical solutions derived for the different scenarios described earlier, by evaluating their evolution on two straight lines $y_c = [0, 50] \times \{55\}$ for the concentration $c_{P,i}$ at t_0^c and $y_V = [0, 80] \times \{55\}$ for the potential V at t_0^V. We consider the according sets of cells $Z_{y_c}^*$ and $Z_{y_V}^*$. In Figure 16, we compare the concentration $\left(c_{K^*,P}^{21,i} - c_{P,i}^0\right)_{K^* \in Z_{y_c}^*}^{\mathbf{s}}$ at t_0^c and potential $\left(V_{K^*}^{12}\right)_{K^* \in Z_{y_V}^*}^{\mathbf{s}}$ at t_0^V for all scenarios $\mathbf{s} = \mathbf{1}, \ldots, \mathbf{7}$. Notably, the signal invasion along x within the dendrite is observed. Regarding the potential, the creation of significant depolarization becomes evident with at least $i = 5$ nearby ion influxes (scenario $\mathbf{s} = \mathbf{5}$), and a maximum value of potential of 33.33 mV. Additionally, as one moves away from these influxes and gets closer to the ionic reservoir, the potential decreases. Beyond the membrane, outside the dendrite, we observe that the potential tends toward a constant value of 0 mV, as one may expect. Concerning the concentration dynamics, we observe that the maximum is always located at the influx $i = 1$. Furthermore, the addition of influxes creates an increasingly significant fluctuation between approximately $x = 20$ µm and $x = 40$ µm , from 0 mM with $i = 1$ to almost 27 mM with $i = 7$.

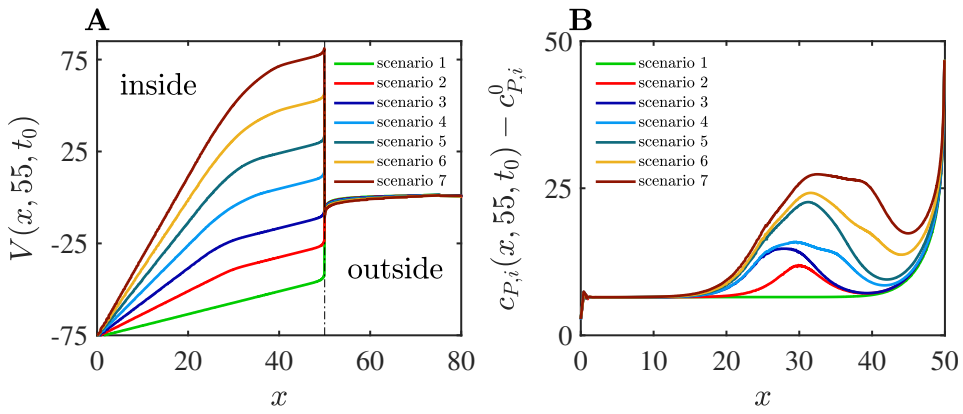

FIGURE 16. Evolution of the potential and concentration dynamics on y_V and y_c for all the scenarios $\mathbf{s} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}\}$. **A**: Discrete values $\left(V^{12}_{K^*}\right)^{\mathbf{s}}_{K^* \in Z^*_{y_V}}$ at peak time $t^V_0 = 0.06$ sec. **B**: Discrete values $\left(c^{21,i}_{K^*,P} - c^0_{P,i}\right)^{\mathbf{s}}_{K^* \in Z^*_{y_c}}$ at peak time $t^c_0 = 0.105$ sec.

We recall that in Chapter 3, we established that when two neighboring dendritic spines are 10.8 µm away from each other, one of them perceives 62 % of the electrical signal from the other. This result suggests that, in our setting of increasing influxes, the electrical signal from an influx will influence the one coming from its neighbor, based on the distance between them both. Thus, we present further investigations involving the temporal evolution of potential and ion concentration dynamics at several specific points (Q)-(R)-(S)-(T) (Fig. 13, Table 4).

We plot in Figure 17 the time evolution of the potential $\left(V^n_{K^*_X}\right)^{\mathbf{s}}_{n=0,\ldots,N}$ for the scenarios $\mathbf{s} = \{\mathbf{1}, \mathbf{4}, \mathbf{5}, \mathbf{7}\}$. Inside and outside the dendritic branch, we clearly observe the effect of adding influxes. Note that the curves in graph **C** of Fig. 17 overlap, i.e., the potential in the membrane remains the same for all scenarios. Outside the membrane (position (T)) in graph **D**, the potential varies overall from 0 to negative values along the time sub-interval $[0, 0.2]$ (sec), then a depolarization is observed until a resting state where the potential tends towards 1 mV. Within the dendrite, i.e., in graph **A** (position (Q)) and graph **B** (position (R)), we observe a large depolarization starting from scenarios **5,6** and **7** only.
These results remarkably suggest that a significant number, at least $i = 5$ influxes, of closely spaced influxes, is needed to generate a sufficient excitability threshold of the potential within a dendritic branch.

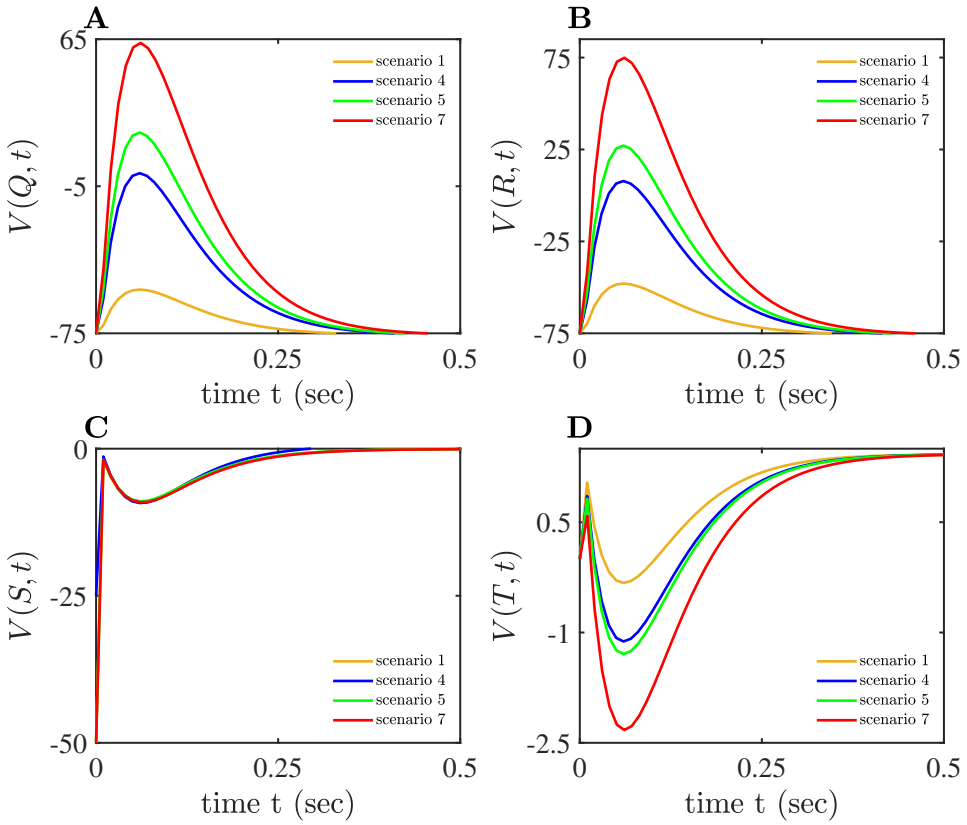

FIGURE 17. Time evolution of $\left(V^n_{K^*_X}\right)^{\mathbf{s}}_{n=0,\dots,N}$ for the scenarios $\mathbf{s} = \{\mathbf{1}, \mathbf{4}, \mathbf{5}, \mathbf{7}\}$ at positions (Q)-(R)-(S)-(T). **A**: $\left(V^n_{K^*_Q}\right)^{\mathbf{s}}_{n=0,\dots,N}$. **B**: $\left(V^n_{K^*_R}\right)^{\mathbf{s}}_{n=0,\dots,N}$. **C**: $\left(V^n_{K^*_S}\right)^{\mathbf{s}}_{n=0,\dots,N}$. **D**: $\left(V^n_{K^*_T}\right)^{\mathbf{s}}_{n=0,\dots,N}$.

Concerning the concentrations inside the dendrite, we examine their dynamics at positions (Q)-(R) for scenarios **1**, **4**, **5**, and **7**. In Fig. 18, we plot the time evolution of concentrations $\left(c^{n,i}_{K^*_X,P} - c^0_{P,i}\right)^{\mathbf{s}}_{n=0,\dots,N}$ for scenarios $\mathbf{s} = \{\mathbf{1}, \mathbf{4}, \mathbf{5}, \mathbf{7}\}$. We observe that, for a single influx ($\mathbf{s} = \mathbf{1}$), the concentration peak is very close to its initial value $c^0_{P,i}$, while the peak within the other scenarios $\mathbf{s} > \mathbf{1}$ increases with the number of influxes.

In graphs **A** and **B** of Fig. 18, the concentration peak of scenario **7** has a value approaching 40 mM, suggesting that ion concentrations add up. The signal invasion is therefore more significant with seven times more influxes contributing to an important depolarization of the potential within the neuron. Also, we observe a sublinear summation of influxes, contrary to the bifurcation case in Section 4.2.1. This result is coherent with the work of [**22**] where the authors find a sublinear summation when inputs are on the same branch.

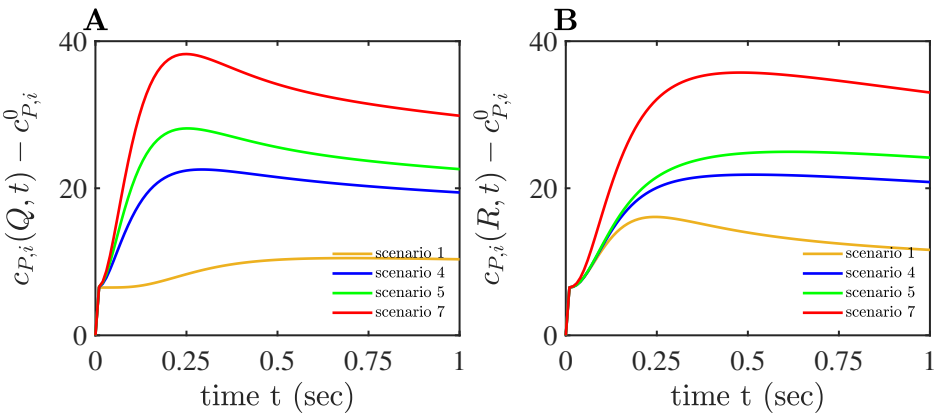

FIGURE 18. Time evolution of the concentration $\left(c^{n,i}_{K^*_X,P} - c^0_{P,i}\right)^{\mathbf{s}}_{n=0,\dots,N}$ for the scenarios $\mathbf{s} = \{\mathbf{1}, \mathbf{4}, \mathbf{7}\}$ at positions (Q)-(R). **A**: $\left(c^{n,i}_{K^*_Q,P} - c^0_{P,i}\right)^{\mathbf{s}}_{n=0,\dots,N}$. **B**: $\left(c^{n,i}_{K^*_R,P} - c^0_{P,i}\right)^{\mathbf{s}}_{n=0,\dots,N}$.

To summarize, the potential invasion is more important than the ionic concentration invasion with neighboring and increasing influxes. This result is coherent with the results found in Chapter 3. As expected, the dendrite requires multiple nearby influxes from neighboring synapses to generate a sufficient excitability threshold. We locally remark that, along a 50 µm long dendrite (see Fig. 15), five influxes are needed to significantly influence information transmission. These influxes are highly frequent in dendrites for signal propagation, validating that dendrites play a crucial role in the functioning and generation of action potentials [**85**].

4.2.3. **Numerical illustration of the boundary layer.** When electrolytes are in contact with solid surfaces, especially charged ones, they give rise to a phenomenon known as the electrical double layer (EDL). This EDL consists of two layers of opposite charge: one layer composed of ions adhering to the surface (the Stern layer) and another layer of diffuse ions extending into the bulk solution. This complex interface, often referred to as the Electric Boundary Layer (EBL), is of immense importance in fields ranging from electrochemistry to biophysics.

Understanding the dynamics of ions within the EBL is critical for elucidating processes such as neural signaling. However, its very small size poses challenges for numerical illustration. The thickness of the Debye length, that is the thickness of the EBL denoted λ_D, can be computed with the formula:

$$\lambda_D = \sqrt{\frac{\varepsilon_i \varepsilon_0 R T_\theta}{F^2 c^0_{P,i} L^2}} \tag{4.18}$$

where L is the typical length of the system. In our study, the EBL thickness λ_D is then approximately 7 nm.

In Chapter 3, the homogeneous Neumann boundary condition on the potential at the membrane wall, kills numerically the EBL. This boundary condition modeling seems coherent when dealing with a single domain representing only the interior of the neuron. Indeed, existing works in the literature have shown that the membrane behaves like an electrical insulator in the absence of ion channels, see [**42**]. In contrast, within our multi-domain modeling, since the Poisson equation lives throughout the domain, only the electrical permittivity coefficients vary among different media. This modeling avoids the need to impose boundary conditions for the potential at the membrane. Furthermore, this setting is consistent with the physiological behavior of the membrane, where its coefficient $\epsilon_m = 4$ (see Table 1) is very small, indicating that the membrane walls barely allow potential propagation. Using our multi-domain approach, we thus expect to capture the EBL in our simulations. Therefore, in this final section, we try to examine the ion dynamics and the potential propagation near the membrane, i.e.,

within the EBL. We show that there is a fluctuation within this layer for the ion concentrations and the potential.

To achieve this, we consider new dimensions of a smaller multi-domain consisting of a dendrite branch (see Figure 19), similar to the previous sections. This domain is smaller to ensure nodes within $\lambda_D = 0.007$ µm , once again the thickness of the EBL (4.18), close to membrane walls. This choice implies an important number of cells, i.e we use 567128 cells and a mesh size with $\{h_{\min} = 0.0002, h_{\max} = 8.4\}$. The simulation time is set to $T_f = 1$ sec with a time step $dt = 5 \times 10^{-5}$. Additionally, we adjusted the right corners of the branch to avoid numerical singularities due to the large number of cells and the Neumann boundary conditions at the membrane walls for the concentrations. Indeed, right angles within a mesh can pose issues when not adequately addressed, as simple finite elements may struggle to accurately represent the solution within these regions. This challenge arises from the localized nature of the shape functions associated with these finite elements, which may struggle to capture rapid or abrupt variations in the solution near corners. Also, when a large number of cells is used, numerical singularities can be increased due to high solution gradients near corners. This can lead to numerical instabilities or poor solution convergence. Hence, we believe that the DDFV method will be better suited to numerically resolve the EBL. The results of this section were challenging to implement via FreeFEM++, i.e., the refinement very close to the wall was challenging, and we could not guarantee mesh points in the EBL on the outside of the membrane. We only guarantee these points inside. Therefore, we encountered numerical issues, and we believe these are, once again, preliminary results. We depict the domain in Figure 19, with its dimensions in Table 6.

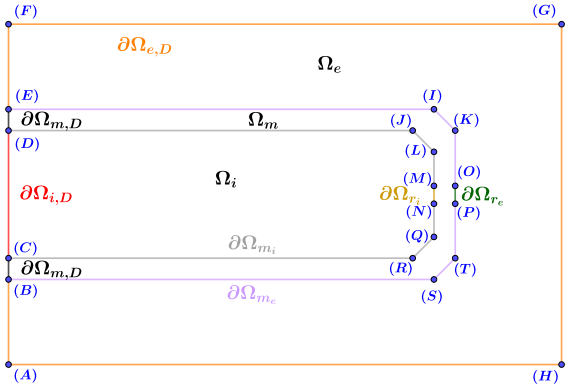

FIGURE 19. Geometry of the multi-domain containing a dendrite branch. The domain is not to scale. The dimensions are provided in Table 6.

In this study, we ensure the presence of 1 to 3 nodes within the "interior" EBL and observe small fluctuations. However, we encounter limitations in further refinement. Therefore, we believe that the standard mesh is insufficient and that a more suitable mesh, closer to the boundary layer, would be necessary for future work, particularly with the DDFV method, where the freedom to choose the mesh would be more straightforward. In addition, we compare our results with those provided in Chapter 3, subsection 3.2.2, where we showed the numerical behavior of our DDFV results in the presence of boundary layer, using a test case inspired by the one-dimensional study conducted in [**81**].

(A)	(0,52)	Position (x,y) of node A
(B)	(0,54.5)	Position (x,y) of node B
(C)	(0,54.507)	Position (x,y) of node C
(D)	(0,55.5)	Position (x,y) of node D
(E)	(0,55.507)	Position (x,y) of node E
(F)	(0,58)	Position (x,y) of node F
(G)	(20,58)	Position (x,y) of node G
(H)	(20,52)	Position (x,y) of node H
(I)	(9.995,55.507)	Position (x,y) of node I
(J)	(9.988,55.5)	Position (x,y) of node J
(K)	(10,55.502)	Position (x,y) of node K
(L)	(9.993,55.495)	Position (x,y) of node L
(M)	(9.993,55.1)	Position (x,y) of node M
(N)	(9.993,54.9)	Position (x,y) of node N
(O)	(10,55.1)	Position (x,y) of node O
(P)	(10,54.9)	Position (x,y) of node P
(Q)	(9.993,54.512)	Position (x,y) of node Q
(R)	(9.988,54.507)	Position (x,y) of node R
(S)	(9.995,54.5)	Position (x,y) of node S
(T)	(10,54.505)	Position (x,y) of node T

TABLE 6. Geometric parameters for the multi-domain, representing the dendrite branch.

Then, we present a zoom of our simulation Figure 20, close to the wall, in order to illustrate the EBL. We show the discrete values $\left(c_{h,P}^{2,i}\right)$ of the concentration $c_{P,i}$ at time $t_1 = 0.001$ sec. The zoom is between $x = [9.981, 9.993]$ and $y = [54.662, 54.676]$. We observe a small fluctuation of the concentrations from the wall $x = 9.993$ to the end of the EBL $x = 9.986$. Then, we observe a constant value of the concentration within the bulk region $x = [9.981, 9.986]$ in Figure 20.

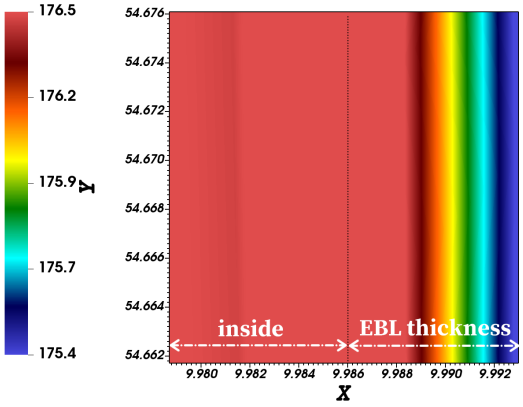

FIGURE 20. Spatial variation of the concentrations in the domain. Discrete values $\left(c_{h,P}^{2,i}\right)$ of the concentration $c_{P,i}$ at time $t_1 = 0.001$sec, between $x = [9.981, 9.993]$ and $y = [54.662, 54.676]$. We depict the bulk region with "inside" $x = [9.981, 9.986]$ and the EBL region with "EBL thickness" $x = [9.986, 9.993]$, corresponding to λ_D. The membrane wall is at $x = 9.993$.

We observe a similar behavior to that shown in Figure 2**B** in Chapter 3, where we investigated the existence of a boundary layer in a 2D test case adapted from the study by [**81**]. Consequently, in Fig. 21, we plot along the line $y_i^{\text{EBL}} = [9.96, 9.993] \times \{54.67\}$, the concentrations $\left(c_{K^*,P}^{2,i}\right)_{K^* \in Z^*_{y_i^{\text{EBL}}}}$ and $\left(c_{K^*,N}^{2,i}\right)_{K^* \in Z^*_{y_i^{\text{EBL}}}}$ in graph **A**, and the potential $\left(V_{K^*}^2\right)_{K^* \in Z^*_{y_i^{\text{EBL}}}}$ in graph **B**.

In graph **A**, we observe that from the bulk at $x = 9.96$, both concentrations are equal with a value

of 176.5 mM, indicating a resting concentration. Indeed, in graph **B**, we similarly observe that the potential appears to tend towards a constant value in the bulk region. Next, for the concentrations in graph **A**, as we reach the EBL, we observe a gradual peak for $c_{P,i}$ (resp. a symmetrically opposite peak for $c_{N,i}$) before decreasing to approximately 175.5 mM (resp., increasing to approximately 177.5 mM for $c_{N,i}$) towards the membrane wall. Regarding the potential in graph **B**, from the bulk to the EBL, we observe a slight decrease upon reaching $x = 9.986$, followed by a gradual increase towards the membrane wall. The fluctuations observed for V are of the order of 0.06 mV, which is very small.

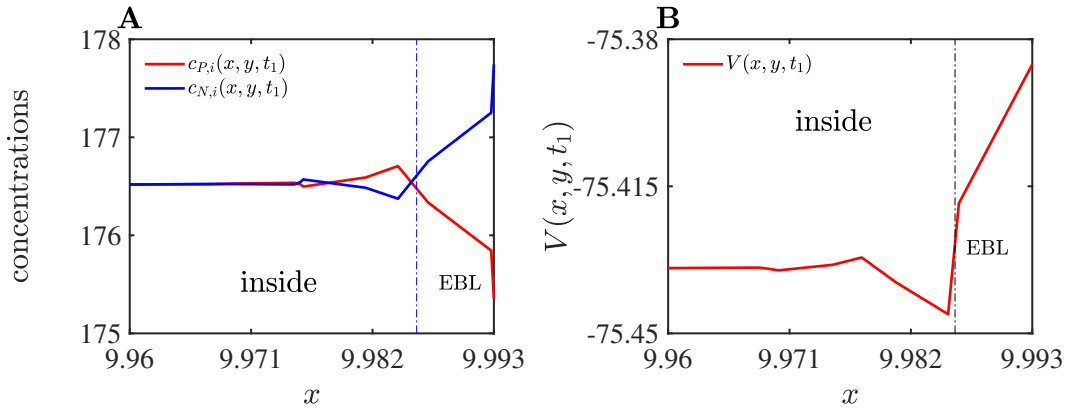

FIGURE 21. Evolution of $c_{P,i}$, $c_{N,i}$ and V, on the line y_i^{EBL}, close to the membrane walls at $t_1 = 0.001$ sec. **A**: Discrete values $\left(c^{2,i}_{K^*,P}\right)_{K^* \in Z^*_{y_i^{\text{EBL}}}}$ in red and $\left(c^{2,i}_{K^*,N}\right)_{K^* \in Z^*_{y_i^{\text{EBL}}}}$ in blue. The thickness of the EBL, λ_D, is shown with dashed blue, i.e between $x = [9.986, 9.993]$. **B**: Discrete values $\left(V^2_{K^*}\right)_{K^* \in Z^*_{y_i^{\text{EBL}}}}$. The thickness of the EBL, λ_D, is shown with dashed black, i.e between $x = [9.986, 9.993]$.

Our results exhibit a behavior similar to those calculated in Figure 3, Chapter 3. Indeed, electroneutrality is attained beyond λ_D. The two concentrations overlap in the bulk region 'inside', Figure 21**A**. However, we observe very small fluctuations whereas we expected strong gradients within the EBL. In addition, we were unable to create a mesh with enough nodes in the domain to illustrate the exterior EBL. Due to time constraints, we were unable to explore all the features of FreeFEM++ to create a highly refined mesh only at the boundary layer. We believe that DDFV should be better at addressing this EBL issue in a multi-domain setting.

Discussion and conclusion

In this chapter, we provide an algorithm to simulate the Poisson-Nernst-Planck system of equations in two dimensions across three distinct domains: intracellular, extracellular, and the membrane, using the FreeFEM++ software. Modeling multiple domains offers the advantage of avoiding the imposition of the homogeneous Neumann condition inherent in the modeling of electrical potential boundary conditions within a single domain. Given that the membrane acts as an electrical insulator, we integrate parameters such as electric permittivity coefficients specific to each medium (cytosol, membrane, external medium). These coefficients are defined in Poisson's equation and exhibit variation across subdomains. Consequently, the equation is solved across the whole domain, where the electric potential is present everywhere, with only the permittivity coefficients varying over the distinct domains. Based on this algorithm, we examine the dynamics of ions and potential within two specific 2D geometries of the dendritic tree: a branching structure and a large rectangular branch. Built on the results obtained in the previous chapter, where the DDFV framework was challenged on numerical applications of the dendritic tree with a single intracellular domain, we came up with consistent outcomes regarding the linear summation of signals from two thin branches modeling ion influx. Furthermore, we compare multi-domain and single-domain approaches, concluding that given the similar results regarding the

signals, the single-domain approach appears to be adequate for such studies. However, it remains limited for investigating the dynamics of the potential within the membrane, where the multi-domain approach is necessary.

Subsequently, we realize that adding ion influxes to a large dendritic branch allows us to obtain a sufficiently high excitability threshold, such that the potential undergoes depolarization with a peak reaching 30 mV and beyond. Even though we do not model the potential-gated channels contributing to action potential generation, we understand that the colocalization of synaptic currents plays a major role in neuron depolarization. We locally observe that, along a 50 µm long dendrite, at least 5 influxes on each side of the dendritic membrane are required for sufficient electrical signal depolarization. Finally, we emphasize that our algorithm enables the visualization of the existence of the electric boundary layer (EBL) at the membrane boundary, even if a weak gradient was observed. These results suggest that all three domains must be considered in order to capture fluctuations in the EBL, requiring a very fine mesh and, consequently, a robust and precise algorithm.

CHAPTER 5

Cemracs Project: A composite finite volume scheme for the Euler equations with source term on unstructured meshes

This chapter presents an independent work issued from a CEMRACS project (in 2022); it is a joint work with M. Boujoudar, E. Franck, P. Hoch, C. Lasuen and Y. Le Hénaff and it has been submitted to ESAIM: Proceedings and Surveys.

This work is a revised version for **ESAIM: Proceeding and Surveys**.

Abstract. In this work we focus on an adaptation of the method described in [54] in order to deal with source term in the 2D Euler equations. This method extends classical 1D solvers (such as VFFC, Roe, Rusanov) to the two-dimensional case on unstructured meshes. The resulting schemes are said to be composite as they can be written as a convex combination of a purely node-based scheme and a purely edge-based scheme. We combine this extension with the ideas developed by Alouges, Ghidaglia and Tajchman in a preprint [2] – focused mainly on the 1D case – and we propose two attempts at discretizing the source term of the Euler equations in order to better preserve stationary solutions. We compare these discretizations with the "usual" centered discretization on several numerical examples.

5.1. Framework

In this work we study a numerical scheme for the 2D fluid Euler system with source term:

$$\partial_t \mathcal{U} + \mathrm{div}\mathcal{F}(\mathcal{U}) = \mathcal{S}(\mathbf{x},\mathcal{U}). \tag{5.1}$$

The unknown is $\mathcal{U} = (\rho, \rho\mathbf{u}, \rho E)^T \in \mathbb{R}^4$. The density of the fluid is $\rho \in \mathbb{R}_+$, its velocity is $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$ and its total energy is $E \in \mathbb{R}_+$. The space variable is denoted by $\mathbf{x} \in \mathbb{R}^2$. The physical flux reads:

$$\mathcal{F}(\mathcal{U}) = (\mathcal{F}_1(\mathcal{U})\ \mathcal{F}_2(\mathcal{U})) = \begin{pmatrix} \rho u_1 & \rho u_2 \\ \rho u_1^2 + P & \rho u_1 u_2 \\ \rho u_1 u_2 & \rho u_2^2 + P \\ (\rho E + P)\, u_1 & (\rho E + P)\, u_2 \end{pmatrix}, \qquad P = (\gamma - 1)\rho \left(E - \frac{1}{2}\|\mathbf{u}\|^2 \right). \tag{5.2}$$

The pressure P is given by the ideal gas law and $\gamma = 1.4$ is the ideal gas constant. The source term $\mathcal{S} = \mathcal{S}(\mathbf{x},\mathcal{U})$ can be any smooth function of its two variables. We focus on the following cases:

- independent from $\mathcal{U}$, $\mathcal{S} = \mathcal{S}(\mathbf{x})$,
- gravity type $\mathcal{S} = (0, \rho\nabla\zeta, \rho\nabla\zeta \cdot \mathbf{u})^T$, where $\zeta = \zeta(\mathbf{x})$ is a smooth function,
- friction type $\mathcal{S} = \left(0, -\lambda\rho\mathbf{u}, -\alpha\lambda\rho\|\mathbf{u}\|^2\right)^T$, where α and λ are constants.

The purpose of this study is to generalize on arbitrary two-dimensional polygonal meshes the ideas developed in [2]. In that work, the authors detail a discretization of the source term that allows to exactly preserve the stationary solutions to (5.1), in the one-dimensional case. We will focus on the two-dimensional framework, and discuss the one-dimensional counterpart in Section 5.2 as an introduction to their method.

The main objective of the method presented in [2] is not different from the motivation of well-balancedness as introduced in the pioneering works [46, 47]: the idea is to discretize Equation (5.1) in such a way that stationary solutions to the continuous problem – or at least a subclass of them –

are also stationary solutions to the discretized problem. Well-balanced methods have been studied for almost three decades now (e.g., [13, 23, 25, 28]), and their efficiency is not to be proved anymore. They have been used in many applications, among others: [65, 79, 15, 4, 26, 19, 38] for shallow waters equations, or [34, 24, 7] for the Burgers' or Euler's equations.

The difference between well-balancedness and Enhanced consistency (a term coined in [2]) lies in the way the source term is discretized. In the former approach, the source term is generally treated in a particular way, different from the discretization of the other quantities in (5.1). In the latter approach, the source term is written as the divergence of some (approximate) function, and the two divergence operators on the LHS and RHS can then be discretized in the same way. We are using the ideas of [2] thoughout this work, and to emphasize the difference of discretization between their method and other well-balanced method, we shall use the term Enhanced consistency for the ideas presented in this work.

The notion of numerical fluxes and their computation in pratice are essential parts of Finite Volume schemes. When arbitrary polygonal meshes are used, one can decide to approximate the numerical fluxes either at the nodes or edges. The first approach is called pure nodal finite volume method because the numerical fluxes are approximated at the nodes of the mesh, and the second approach is called pure edge finite volume method because they are approximated at the edges of the mesh.

Several studies showed that node-based methods are efficient in Lagrangian update hydrodynamics [33, 71] and allow to obtain a local mesh update that is consistent with geometric conservation laws. It is not the case for the edge-based approach [6]. However, the node-based methods sometimes display parasitic behaviors that may be treated using ad-hoc processes [33, 41]. On the other hand, pure edge finite volume schemes perform well and benefit most from the one-dimensional numerical flux design theory. For first order explicit finite volume schemes, the propagation is performed via cell adjacency (i.e, edges), and this may generate an important CFL constraint on certain specified meshes (see for instance [54, Sect. 1]).

This motivates the use of composite schemes, which make a continuous bridge between edge-based and node-based finite volume schemes. This type of schemes was introduced in [54] to solve the homogeneous 2D Euler equations. The numerical results show that these schemes perform well in terms of stability. For instance, in [8, 14, 54] the polygonal pure nodal GLACE scheme from [33] and the composite GLACE conical degenerate extension from [8, 14] were compared on the Sedov test (for the Euler Gas Dynamics) solved with Lagrange update formulation. The authors of these papers showed that the composite GLACE conical degenerate extension provides robust numerical results. However, the pure nodal scheme exhibits numerical instability due to lack of numerical viscosity in the diagonal direction, and furthermore both schemes are first order in time and space.

The purpose of the present work is to develop a composite flux scheme that extends the ideas of [2] in the 2D case. In this unpublished but publicly available work, the authors develop a promising method to discretize the source term in order to numerically preserve stationary solutions. However, upon a closer look, we believe the computations in the section devoted to the 2D case needed more investigations, and this justifies our work.

This paper is organised as follows: in Section 5.2 we give a brief overview of the VFFC scheme in the 1D framework (see also [44]), and we recall the main idea of [2]. In Section 5.3 we define a composite scheme along the lines of [54], which is a generalization of the VFFC scheme in 2D. Section 5.4 is devoted to the treatment of the source term and several methods are detailed. Eventually, some numerical tests are provided in order to compare the different methods.

5.2. 1D Scheme

In this Section, we briefly recall the 1D VFFC scheme from [44], which discretizes the 1D version of (5.1). This scheme is an Eulerian flux-based scheme. We also present the notion of *enhanced consistency* as introduced in [2], which consists in discretizing the source term so as to preserve exactly the stationary solutions of (5.1).

5.2.1. **Flux based scheme.** We consider a space discretization $\{x_{j+1/2}\}_{j+1/2}$ of a given interval $[0, L]$, and define the cells $\Omega_j = [x_{j-1/2}, x_{j+1/2}]$. Their length is denoted by $\Delta x_j = x_{j+1/2} - x_{j-1/2}$ and we also need to define their centroid $\{x_j\}_j$: $x_j = (x_{j+1/2} + x_{j-1/2})/2$.

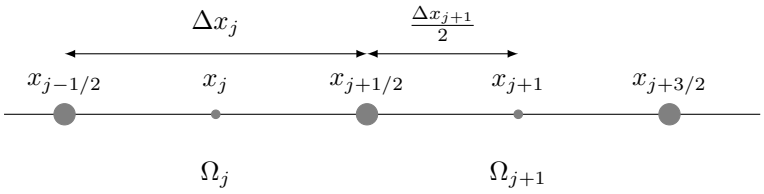

FIGURE 1. 1D discretization

A classical finite volume scheme in time writes :

$$\Delta x_j \frac{\mathcal{U}_j^{n+1} - \mathcal{U}_j}{\Delta t} + \mathcal{G}^{j+1/2} - \mathcal{G}^{j-1/2} = \mathcal{Q}_j, \quad \forall j, \tag{5.3}$$

where the exponent n, denoting current time, has been removed in order to simplify the notations. The numerical source term is denoted by $\mathcal{Q}_j$ and can be computed using several formulas, see Section 5.2.2.

The quantity $\mathcal{G}_{j+1/2}$ is the numerical flux between the cells j and $j+1$. To obtain a flux-based scheme, we define it as:

$$\mathcal{G}^{j+1/2} = \frac{\mathcal{F}_j + \mathcal{F}_{j+1}}{2} + \Lambda_{j+\frac{1}{2}} \frac{\mathcal{F}_j - \mathcal{F}_{j+1}}{2}, \qquad \Lambda_{j+\frac{1}{2}} = \text{sign}(J(\mathcal{U}_{j+1/2})),$$

where the sign of a matrix is defined in Definition 5.1. Here, $J(\mathcal{U}_{j+1/2})$ denotes the Jacobian matrix of the mapping $\mathcal{U} \mapsto \mathcal{F}(\mathcal{U})$ evaluated at $\mathcal{U}_{j+1/2}$. The quantity $\mathcal{U}_{j+1/2}$ is an approximation of the solution at the interface between the cells Ω_j and Ω_{j+1}. One of the simplest approximations is the VFFC average (see [44]), and in this case $\mathcal{U}_{j+1/2}$ is given by $\mathcal{U}_{j+1/2} = (\mathcal{U}_j + \mathcal{U}_{j+1})/2$. We emphasize the fact that $\mathcal{U}_j$ and $\mathcal{U}_{j+1}$ are only the unknowns of the Finite Volume scheme, and no additional knowledge about the solution is required to compute $\mathcal{G}^{j+1/2}$. The matrix $\Lambda_{j+\frac{1}{2}}$ is called an "upwind matrix". We also emphasize that the exponent $j+1/2$ in $\mathcal{G}^{j+1/2}$ has to be understood now as a "global index" in the mesh, and that $\mathcal{G}^{j+1/2}$ is the numerical flux evaluated at the degree of freedom (dof) $j+1/2$. More details about the notion of "global" or "local" indices are given in Section 5.3.

A natural way of discretizing the source term is the following:

$$\mathcal{Q}_j = \mathcal{S}(x_j, \mathcal{U}_j) \Delta x_j. \tag{5.4}$$

However, as explained in [2], this choice of $\mathcal{Q}_j$ does not allow to preserve exactly the stationary solutions. We present in Section 5.2.2 the method designed in [2] aimed at preserving numerically the stationary solutions.

Definition 5.1 (Sign of a matrix)**.** *Let $J \in \mathbb{R}^{m \times m}$ be a diagonalizable matrix of size $m \times m$. Let $\{\lambda_i\}_i$ its eigenvalues, R the matrix composed of the eigenvectors associated to the eigenvalues, and L the matrix associated to the left eigenvectors of J (i.e the eigenvectors of J^T). We have:*

$$J = R\, diag\left(\{\lambda_i\}\right) L, \qquad RL = LR = I_m,$$

and we define:

$$sign(J) = R\, diag\left(\{sign(\lambda_i)\}\right) L. \tag{5.5}$$

In Definition 5.1, unless stated explicitely, the term "eigenvector" refers to the right eigenvectors, that is the vectors v such that $Jv = \lambda v$ for some eigenvalue λ.

5.2.2. **Enhanced consistency.** One of the main disadvantages of the discretization (5.4) is that, even if the scheme were to be initialized with an exact stationary solution, the approximate solution

eventually drifts away from the stationary solution. However, it is desirable to be able to recover the exact stationary solutions at the discrete level.

Criterion 5.2. *If, at some iteration n, we have:*

$$\forall j,\ \mathcal{U}^n(x_j) = \mathcal{U}^*(x_j) \quad \text{with} \quad \frac{d}{dx}\mathcal{F}(\mathcal{U}^*) = \mathcal{S}(x,\mathcal{U}^*(x)),$$

then

$$\forall j,\ \mathcal{U}^{n+1}(x_j) = \mathcal{U}^n(x_j).$$

We want the scheme to be able to "capture" continuous stationary solutions. We define $\Phi = \mathcal{F}(\mathcal{U}^*)$. As explained in [2], a way of satisfying Criterion 5.2 is to define the source term as:

$$\begin{aligned}\mathcal{Q}_j &= \frac{1}{2}\left[(I+\Lambda_{j+1/2})\Phi(x_j) + (I-\Lambda_{j+1/2})\Phi(x_{j+1}) - (I+\Lambda_{j-1/2})\Phi(x_{j-1}) - (I-\Lambda_{j-1/2})\Phi(x_j)\right] \\ &= \frac{1}{2}\left[(I-\Lambda_{j+1/2})(\Phi(x_{j+1})-\Phi(x_j)) + (I+\Lambda_{j-1/2})(\Phi(x_j)-\Phi(x_{j-1}))\right]. \qquad (5.6)\end{aligned}$$

Owing to the well-known formula:

$$\Phi(x_{j+1}) - \Phi(x_j) = \int_{x_j}^{x_{j+1}} \mathcal{S}\left(y,\mathcal{U}^*(t,y)\right) dy,$$

we are able to rewrite (5.6) as

$$\mathcal{Q}_j = \frac{1}{2}\left(\int_{x_{j-1}}^{x_{j+1}} \mathcal{S}\left(y,\mathcal{U}^*(t,y)\right) dy - \Lambda_{j+1/2}\int_{x_j}^{x_{j+1}} \mathcal{S}\left(y,\mathcal{U}^*(t,y)\right) dy + \Lambda_{j-1/2}\int_{x_{j-1}}^{x_j} \mathcal{S}\left(y,\mathcal{U}^*(t,y)\right) dy\right).$$

Since $\mathcal{U}^*$ is not known *apriori*, we define the source term as

$$\mathcal{Q}_j = \frac{1}{2}\left(\int_{x_{j-1}}^{x_{j+1}} \mathcal{S}\left(y,\mathcal{U}(t,y)\right) dy - \Lambda_{j+1/2}\int_{x_j}^{x_{j+1}} \mathcal{S}\left(y,\mathcal{U}(t,y)\right) dy + \Lambda_{j-1/2}\int_{x_{j-1}}^{x_j} \mathcal{S}\left(y,\mathcal{U}(t,y)\right) dy\right), \qquad (5.7)$$

and Criterion 5.2 is still satisfied. As noticed in [2], when $\mathcal{S}$ is piecewise constant ($\mathcal{S} = \mathcal{S}_j = \mathcal{S}(x_j,\mathcal{U}_j)$ on cell j), Equation (5.7) can be reformulated in a simpler way:

$$\mathcal{Q}_j = \frac{1}{2}\left(I+\Lambda_{j-1/2}\right)\left(\frac{\Delta x_{j-1}}{2}\mathcal{S}_{j-1} + \frac{\Delta x_j}{2}\mathcal{S}_j\right) + \frac{1}{2}\left(I-\Lambda_{j+1/2}\right)\left(\frac{\Delta x_j}{2}\mathcal{S}_j + \frac{\Delta x_{j+1}}{2}\mathcal{S}_{j+1}\right). \qquad (5.8)$$

5.3. 2D Scheme

Let $\Omega = [0,1]^2$, and define $\mathcal{T}$ a polygonal mesh over Ω. We denote by Ω_j a polygonal cell of the mesh $\mathcal{T}$, and let $\mathbf{x}_{r-1}$, $\mathbf{x}_r$ and $\mathbf{x}_{r+1}$ three consecutive vertices of the polygon Ω_j. Define

- the middle of the edge $[\mathbf{x}_r, \mathbf{x}_{r+1}]$: $\mathbf{x}_{r+\frac{1}{2}} = \dfrac{\mathbf{x}_r + \mathbf{x}_{r+1}}{2}$,
- the outward normal to the edge directed from $\mathbf{x}_r$ to $\mathbf{x}_{r+1}$, $[\mathbf{x}_r, \mathbf{x}_{r+1}]$: $\mathbf{C}_j^{r+\frac{1}{2}} = -(\mathbf{x}_{r+1} - \mathbf{x}_r)^\perp$,
- the normal to the node r: $\mathbf{C}_j^r = \frac{1}{2}\left(\mathbf{C}_j^{r+\frac{1}{2}} + \mathbf{C}_j^{r-1/2}\right)$,

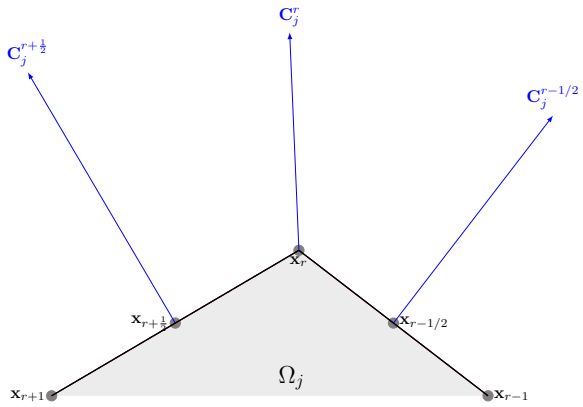

FIGURE 2. Normal vectors at nodes and edges: composite set

where for any vector $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \mathbb{R}^2$ we use the usual notation $\boldsymbol{\xi}^\perp = (-\xi_2, \xi_1)$ which corresponds to a rotation of angle $+\frac{\pi}{2}$. In what follows we present the composite schemes, as a combination of "nodal-based schemes" – which involve $\mathbf{C}_j^r$ – and "edge-based schemes" – which involve $\mathbf{C}_j^{r+\frac{1}{2}}$. This is not new, and very much inspired from [54].

Remark 5.3. *We acknowledge that it normally does not make sense to talk about a "normal vector to a node". However, the definition we use (which has been borrowed from* [54]*) is always well-defined for polygonal meshes. For all of our purposes, it is well-suited and we shall use this definition throughout this work.*

For the sake of clarity, we start by explaining below the different sums that can be performed over the mesh, for an arbitrary function $g = g(x)$. By convention, for any given cell we will use integers r to denote vertices of the cell, also called nodes. They are supposed to be labelled consecutively, so that it makes sense to talk about the edge $[\mathbf{x}_r, \mathbf{x}_{r+1}]$. Moreover, we implicitely suppose $r \in \{1, \dots, V\}$ for a polygonal cell with V vertices (with the loop convention that $\mathbf{x}_{V+1} = \mathbf{x}_1$). We also denote by half-integers $r + \frac{1}{2}$ the middle of the edge $[\mathbf{x}_r, \mathbf{x}_{r+1}]$. Our conventions are illustrated in Figure 2.

For a given cell Ω_j we denote by $dof \in \{r : r \in \Omega_j\} \cup \{r + \frac{1}{2} : r + \frac{1}{2} \in \Omega_j\}$ any degree of freedom of Ω_j (it can be a node or the middle of an edge), and the notation g_j^{dof} denotes the evaluation of the function g at the node or edge dof in cell Ω_j. Sometimes dof will also denote a general degree of freedom in the mesh, i.e it can be any node or middle of an edge in the mesh. The meaning of dof will be clear from the context (think of r and $r + \frac{1}{2}$ as "local indices" within a given cell, and of dof as either a "local" or "global" index, depending on the context).

- $\sum_{r \in \Omega_j} g_j^r$: sum over all the vertices of the cell j,
- $\sum_{r+\frac{1}{2} \in \Omega_j} g_j^{r+\frac{1}{2}}$: sum over all the mid-edge points of the cell j,
- $N_{dof} = \sum_{i : dof \in i} 1$: number of cells that contains the given degree of freedom dof,
- $\sum_{i : dof \in \Omega_i} g_i^{dof}$: sum, for a given degree of freedom in the mesh, over all the cells that contains this degree of freedom,
- $\sum_{j \in \mathcal{T}} g_j$: sum over all cells in the mesh,
- $\sum_{r \in nodes(\mathcal{T})} g^r$: sum over all nodes in the mesh,
- $\sum_{r+\frac{1}{2} \in edges(\mathcal{T})} g^{r+\frac{1}{2}}$: sum over all mid-edge points in the mesh,
- h : the maximum edge length among all edges in the mesh.

The normal vectors $\mathbf{C}_j^r$ satisfy the following equalities:

$$\sum_{r \in \Omega_j} \mathbf{C}_j^r = \sum_{r+\frac{1}{2} \in \Omega_j} \mathbf{C}_j^{r+\frac{1}{2}} = \mathbf{0}, \qquad \forall j \in \mathcal{T}, \tag{5.9}$$

and, for any degree of freedom dof,

$$\sum_{i:dof\in\Omega_i} \mathbf{C}_i^{dof} = \mathbf{0}. \tag{5.10}$$

Moreover, we have the following result, whose proof can be found in [12]:

Proposition 5.4. *Let $g \in \mathcal{C}^2(\mathbb{R}^2;\mathbb{R})$. Then, for all $\theta \in [0,1]$:*

$$\frac{1}{|\Omega_j|}\int_{\partial\Omega_j} g\mathbf{n} = \frac{1}{|\Omega_j|}\left[(1-\theta)\sum_{r\in\Omega_j} g(\mathbf{x}_r)\mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2}\in\Omega_j} g(\mathbf{x}_{r+\frac{1}{2}})\mathbf{C}_j^{r+\frac{1}{2}}\right] + \mathcal{O}(h), \tag{5.11}$$

and the remainder term in (5.11) vanishes if g is an affine function.

We define the sub-volumes (see also Figure 3)

$$|\Omega_j^r| = \frac{1-\theta}{2}(\mathbf{x}_r - \mathbf{x}_j)\cdot\mathbf{C}_j^r, \qquad |\Omega_j^{r+1/2}| = \frac{\theta}{2}(\mathbf{x}_{r+1/2} - \mathbf{x}_j)\cdot\mathbf{C}_j^{r+1/2}, \tag{5.12}$$

so that the volume of the cell Ω_j reads

$$|\Omega_j| \quad = \sum_{r\in\Omega_j} |\Omega_j^r| + \sum_{r+\frac{1}{2}\in\Omega_j} |\Omega_j^{r+\frac{1}{2}}| = \sum_{dof\in\Omega_j} |\Omega_j^{dof}|. \tag{5.13}$$

We end this section by giving a definition that will be useful in the rest of the paper, and which generalizes Definition 5.1 to the two-dimensional case.

Definition 5.5. *We define the Jacobian matrix of the physical flux function $\mathcal{F}$, given by (5.2), along the unit direction vector $\mathbf{n} = (n_1, n_2)$ as*

$$J(\mathcal{U},\mathbf{n}) = n_1\nabla\mathcal{F}_1(\mathcal{U}) + n_2\nabla\mathcal{F}_2(\mathcal{U}).$$

This matrix is diagonalizable because the system (5.1) is hyperbolic. Moreover, we have:

$$J(\mathcal{U},\mathbf{n}) = -J(\mathcal{U},-\mathbf{n}), \tag{5.14}$$

hence, by (5.5),

$$sign\left(J(\mathcal{U},\mathbf{n})\right) = -sign\left(J(\mathcal{U},-\mathbf{n})\right). \tag{5.15}$$

5.3.1. **Edge-based schemes.** We now present a 2D edge-based extension of the 1D VFFC scheme described in Section 5.2. First, integrate Equation (5.1) on a given cell Ω_j, then use Proposition 5.4 with $\theta = 1$, so that the flux is approximated as

$$\int_{\Omega_j} \mathrm{div}\mathcal{F}(\mathcal{U}) = \int_{\partial\Omega_j} \mathcal{F}(\mathcal{U})\cdot\mathbf{n} \approx \sum_{r+\frac{1}{2}\in\Omega_j} \mathcal{G}^{r+\frac{1}{2}}\cdot\mathbf{C}_j^{r+\frac{1}{2}}.$$

Consider an edge labelled $r+\frac{1}{2}$ that does not lie at the boundary of the computational domain (we will deal later with the boundary conditions), and let j,k the labels of its two support cells. That is, $r+\frac{1}{2} \in \Omega_j \cap \Omega_k$. We define:

$$\mathcal{G}^{r+\frac{1}{2}} = \frac{\mathcal{F}_j + \mathcal{F}_k}{2} + \Lambda_j^{r+\frac{1}{2}}\frac{\mathcal{F}_j - \mathcal{F}_k}{2}, \qquad \Lambda_j^{r+\frac{1}{2}} = \mathrm{sign}J\left(\mathcal{U}_{r+\frac{1}{2}}, \mathbf{n}_j^{r+\frac{1}{2}}\right), \tag{5.16}$$

where the sign of a matrix is defined in Definition 5.5, and $\mathbf{n}_j^{r+\frac{1}{2}} = \mathbf{C}_j^{r+\frac{1}{2}} \Big/ \|\mathbf{C}_j^{r+\frac{1}{2}}\|$. Moreover, by Definition 5.5,

$$\Lambda_j^{r+\frac{1}{2}} = -\Lambda_k^{r+\frac{1}{2}}. \tag{5.17}$$

Using (5.17), we note that equation (5.16) writes:

$$\mathcal{G}^{r+\frac{1}{2}} = \frac{1}{N_{r+\frac{1}{2}}}\sum_{i:r+\frac{1}{2}\in\Omega_i}\left(I + \Lambda_i^{r+\frac{1}{2}}\right)\mathcal{F}_i, \tag{5.18}$$

which will be useful in Section 5.3.2. Moreover, equation (5.17) also yields:

$$\sum_{i:r+\frac{1}{2}\in\Omega_i} \Lambda_i^{r+\frac{1}{2}} = 0. \tag{5.19}$$

Finally, the average state $\mathcal{U}_{r+\frac{1}{2}}$ is given by the VFFC average (see [44]):

$$\mathcal{U}_{r+\frac{1}{2}} = \frac{|\Omega_j|\mathcal{U}_j + |\Omega_k|\mathcal{U}_k}{|\Omega_j| + |\Omega_k|}. \tag{5.20}$$

5.3.2. **Generalization to the nodes and composite scheme.** We present here the composite scheme as defined in [54]. Using Proposition 5.4, we have

$$\int_{\partial\Omega_j} \mathcal{F}(\mathcal{U})\cdot\mathbf{n} \approx (1-\theta)\sum_{r\in\Omega_j} \mathcal{G}^r\cdot\mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2}\in\Omega_j} \mathcal{G}^{r+\frac{1}{2}}\cdot\mathbf{C}_j^{r+\frac{1}{2}},$$

and the composite scheme reads:

$$|\Omega_j|\frac{\mathcal{U}_j^{n+1}-\mathcal{U}_j}{\Delta t} + (1-\theta)\sum_{r\in\Omega_j} \mathcal{G}^r\cdot\mathbf{C}_j^r + \theta \sum_{r+1/2\in\Omega_j} \mathcal{G}^{r+\frac{1}{2}}\cdot\mathbf{C}_j^{r+1/2} = \mathcal{Q}_j. \tag{5.21}$$

The edge flux $\mathcal{G}^{r+\frac{1}{2}}$ is defined in Equation (5.18), and we define the node flux $\mathcal{G}^r$ in a similar manner:

$$\mathcal{G}^r = \frac{1}{N_r}\sum_{i:r\in\Omega_i} (I+\Lambda_i^r)\,\mathcal{F}_i. \tag{5.22}$$

A natural choice for defining Λ_i^r would be

$$\Lambda_i^r = \tilde{\Lambda}_i^r, \qquad \tilde{\Lambda}_i^r := \frac{1}{N_r-1}\sum_{k:r\in\Omega_k,\ k\neq i} \mathrm{sign} J(\mathcal{U}_{(i,k)}, \mathbf{n}_i^r).$$

where $\mathcal{U}_{(i,k)}$ is given by the VFFC average between cells i and k (see Equation (5.20)) and $\mathbf{n}_i^r = \mathbf{C}_i^r/\|\mathbf{C}_i^r\|$. However, contrary to equation (5.19), we do not have *apriori*:

$$\sum_{i:r\in\Omega_i} \Lambda_i^r = 0. \tag{5.23}$$

This property is essential to ensure the consistency of the scheme: indeed, this is a key argument in the proof of Proposition 5.7. A way of satisfying equation (5.23) is to define Λ_i^r as:

$$\Lambda_i^r = \tilde{\Lambda}_i^r - \frac{1}{N_r}\sum_{k:r\in\Omega_k} \tilde{\Lambda}_k^r. \tag{5.24}$$

One can recover the edge-based scheme (resp. the node-based scheme) by setting $\theta = 1$ (resp. $\theta = 0$) in the scheme (5.21).

5.3.3. **Dirichlet boundary conditions.** We implement Dirichlet boundary conditions as follows. Let *dof* be a degree of freedom where the solution is imposed to be $\mathcal{U}_{\text{boundary}}$. The numerical flux is given by: $\mathcal{G}^{dof} = \mathcal{F}(\mathcal{U}_{\text{boundary}})$. We make the assumption that the exact solution is known on the boundary of the computation domain.

This is a strong assumption, but we recall that the purpose of this work is to look for new ways of discretizing the source term, and not to deal with boundary conditions. This is why we consider the most simple situation.

5.3.4. **Properties of the composite scheme.**

Proposition 5.6 (Global conservativity)**.** *The scheme* (5.21) *is globally conservative:*

$$\sum_{j\in\mathcal{T}} |\Omega_j|\mathcal{U}_j^{n+1} = \sum_{j\in\mathcal{T}} |\Omega_j|\mathcal{U}_j + \Delta t \sum_{j\in\mathcal{T}} \mathcal{Q}_j,$$

up to some boundary terms.

Proof. We sum (5.21) over $j\in\mathcal{T}$:

$$\sum_{j\in\mathcal{T}}\left(|\Omega_j|\frac{\mathcal{U}_j^{n+1}-\mathcal{U}_j^n}{\Delta t} - \mathcal{Q}_j\right) = -(1-\theta)\sum_{j\in\mathcal{T}}\sum_{r\in\Omega_j} \mathcal{G}^r\cdot\mathbf{C}_j^r - \theta\sum_{j\in\mathcal{T}}\sum_{r+\frac{1}{2}\in\Omega_j} \mathcal{G}^{r+\frac{1}{2}}\cdot\mathbf{C}_j^{r+\frac{1}{2}}.$$

We can then exchange the sum over $\{j \in \mathcal{T}\}$ and the ones over $\{r \in \Omega_j\}$ and $\left\{r + \frac{1}{2} \in \Omega_j\right\}$, and use the relation (5.10) to obtain:

$$\sum_{j\in\mathcal{T}} \sum_{r\in\Omega_j} \mathcal{G}^r \cdot \mathbf{C}_j^r = \sum_{r\in nodes(\mathcal{T})} \sum_{i:r\in\Omega_i} \mathcal{G}^r \cdot \mathbf{C}_i^r = \sum_{r\in nodes(\mathcal{T})} \mathcal{G}^r \cdot \underbrace{\left(\sum_{i:r\in\Omega_i} \mathbf{C}_i^r \right)}_{=\mathbf{0}} = 0,$$

and:

$$\begin{aligned}\sum_{j\in\mathcal{T}} \sum_{r+\frac{1}{2}\in\Omega_j} \mathcal{G}^{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+\frac{1}{2}} &= \sum_{r\in edges(\mathcal{T})} \sum_{i:r+\frac{1}{2}\in\Omega_i} \mathcal{G}^{r+\frac{1}{2}} \cdot \mathbf{C}_i^{r+\frac{1}{2}} \\ &= \sum_{r+\frac{1}{2}\in edges(\mathcal{T})} \mathcal{G}^{r+\frac{1}{2}} \cdot \underbrace{\left(\sum_{i:r+\frac{1}{2}\in\Omega_i} \mathbf{C}_i^{r+\frac{1}{2}} \right)}_{=\mathbf{0}} = 0.\end{aligned}$$

□

Proposition 5.7 (Consistency of the fluxes)**.** *The fluxes* (5.16) *and* (5.22) *are consistent.*

Proof. Let $r + \frac{1}{2}$ be the index of an inner edge and j, k the indices of its support cells. If $\mathcal{U}_j = \mathcal{U}_k = \bar{\mathcal{U}}$, then using (5.16), we have $\mathcal{G} = \mathcal{F}\left(\bar{\mathcal{U}}\right)$. Similarly, let r be the index of an inner node, meaning it does not lie on the boundary of the computational domain. Assuming that $\mathcal{U}_i = \bar{\mathcal{U}}$ for any cell i such that $r \in \Omega_i$, we have, using equations (5.22) and (5.23) :

$$\mathcal{G}^r = \mathcal{F}\left(\bar{\mathcal{U}}\right) + \frac{1}{N_r} \underbrace{\left(\sum_{i:r\in\Omega_i} \Lambda_i^r \right)}_{=0} \mathcal{F}\left(\bar{\mathcal{U}}\right) = \mathcal{F}\left(\bar{\mathcal{U}}\right).$$

□

5.4. Discretization of the source term

In this Section, we study differents ways of computing the numerical source term $\mathcal{Q}_j$ which satisfy a 2D version of Criterion 5.2 (ideally) and are different from the classical discretization: $\mathcal{Q}_j = |\Omega_j| \mathcal{S}(\mathbf{x}_j, \mathcal{U}_j)$.

5.4.1. **Enhanced consistency in 2D.** The reasoning is identical to Section 5.2.2. We want to compute $\mathcal{Q}_j$ so as to satisfy the following property, which a 2D generalization of Criterion 5.2:

Criterion 5.8. *If, at some iteration n, we have:*

$$\forall j,\ \mathcal{U}^n(\mathbf{x}_j) = \mathcal{U}^*(\mathbf{x}_j) \quad \textit{with} \quad \mathrm{div}\mathcal{F}(\mathcal{U}^*) = \mathcal{S}(\mathbf{x}, \mathcal{U}^*(\mathbf{x})),$$

then

$$\forall j,\ \mathcal{U}^{n+1}(\mathbf{x}_j) = \mathcal{U}^n(\mathbf{x}_j).$$

We define $\Phi^* = \mathcal{F}(\mathcal{U}^*)$, and consider the edge (resp. node) flux as defined in (5.18) (resp. (5.22)). In order to satisfy Criterion 5.8, the source term has to be equal to:

$$\mathcal{Q}_j = (1-\theta) \sum_{r\in\Omega_j} \frac{1}{N_r} \left[\sum_{i:r\in\Omega_i} \left(I_4 + \Lambda_i^r\right) \Phi_i^* \right] \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2}\in\Omega_j} \frac{1}{N_{r+\frac{1}{2}}} \left[\sum_{i:r+\frac{1}{2}\in\Omega_i} \left(I_4 + \Lambda_i^{r+\frac{1}{2}}\right) \Phi_i^* \right] \cdot \mathbf{C}_j^{r+\frac{1}{2}}.$$

However, we do not know *apriori* Φ^* nor $\mathcal{U}^*$. Therefore we define $\mathcal{Q}_j$ as:

(5.25)

$$\mathcal{Q}_j = (1-\theta) \sum_{r\in\Omega_j} \frac{1}{N_r} \left[\sum_{i:r\in\Omega_i} \left(I_4 + \Lambda_i^r\right) \Phi_i \right] \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2}\in\Omega_j} \frac{1}{N_{r+\frac{1}{2}}} \left[\sum_{i:r+\frac{1}{2}\in\Omega_i} \left(I_4 + \Lambda_i^{r+\frac{1}{2}}\right) \Phi_i \right] \cdot \mathbf{C}_j^{r+\frac{1}{2}},$$

where Φ is a solution to $\operatorname{div}\Phi = \mathcal{S}(\mathbf{x}, \mathcal{U}^n)$. The upwind matrices are still given by equations (5.16) and (5.24). The scheme (5.21) can then be written as:

$$|\Omega_j| \frac{\mathcal{U}_j^{n+1} - \mathcal{U}_j^n}{\Delta t} + (1-\theta) \sum_{r \in \Omega_j} \tilde{\mathcal{G}}^r \cdot \mathbf{C}_j^r + \theta \sum_{r+1/2 \in \Omega_j} \tilde{\mathcal{G}}^{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+1/2} = 0, \tag{5.26}$$

with:

$$\tilde{\mathcal{G}}^{dof} = \frac{1}{N_{dof}} \sum_{i : dof \in \Omega_i} \left(I_4 + \Lambda_i^{dof} \right) (\mathcal{F}_i - \Phi_i). \tag{5.27}$$

However, in contrast with the 1D case, it is not possible in the general case to find an expression of $\mathcal{Q}_j$ as a function of $\mathcal{S}$ and therefore to end up with a formula that is similar to (5.7). In Section 5.4.2 we present a way of computing Φ by solving a Poisson equation, and in Section 5.4.3, we try to generalize formula (5.8) to the 2D framework, without guarantying that Property 5.8 is still satisfied.

5.4.2. **Anti-derivative using a Poisson equation.** First of all, let us clarify what we are looking for in this section. By "anti-derivative", it is meant a function ψ such that $\nabla\psi = \Phi$. In a one-dimensional setting, the gradient of a function is simply its derivative, hence the anti-derivative is the same as the primitive function. It is then unique up to an additional constant.

This idea comes from [**2**]. In order to solve $\operatorname{div}\Phi = \mathcal{S}$ we solve the following problem:

$$\Delta\psi = \mathcal{S}, \qquad \Phi = \nabla\psi.$$

We impose homogeneous Dirichlet boundary conditions for ψ. The construction of the scheme is detailed in Appendix B. With this method, the unknowns $(\Phi_i)_i$ that appear in (5.27) are given by:

$$|\Omega_j| \Phi_j = \sum_{r \in \Omega_j} |\Omega_j^r| \Phi_r + \sum_{r+\frac{1}{2} \in \Omega_j} |\Omega_j^{r+\frac{1}{2}}| \Phi_{r+\frac{1}{2}}, \qquad \Phi_{r+\frac{1}{2}} = \frac{\Phi_r + \Phi_{r+1}}{2}, \tag{5.28}$$

where the sub-volumes are defined in (5.12) and the anti-derivative at the nodes satisfies

$$\sum_{r \in \Omega_j} \Phi_r \cdot \mathbf{C}_j^r = |\Omega_j| \mathcal{S}_j, \tag{5.29}$$

with

$$\left(\sum_{i | r \in \Omega_i} \mathbf{C}_i^r \otimes (\mathbf{x}_i - \mathbf{x}_r) \right) \Phi_r^T = \sum_{i | r \in \Omega_i} \psi_i \otimes \mathbf{C}_i^r. \tag{5.30}$$

Equations (5.29) and (5.30) define a linear system whose size is the number of cells and whose unknown is ψ. Once this system is solved, we compute the anti-derivative at the nodes with equation (5.30) and the values at the centers of the cells with equation (5.28).

Dirichlet boundary conditions. Let dof be a degree of freedom where the solution is imposed to be $\mathcal{U}_{\text{boundary}}$. The numerical flux (5.27) is given by: $\tilde{\mathcal{G}}^{dof} = \mathcal{F}(\mathcal{U}_{\text{boundary}}) - \Phi_{dof}$, where Φ_{dof} is defined in (5.29) (5.30).

5.4.3. **Sub-volumes upwinding.** We propose here to discretize the source term by upwinding the sub-volumes around r or $r + 1/2$:

$$\mathcal{Q}_j = \sum_{dof \in \Omega_j} \frac{1}{N_{dof}} \sum_{i : dof \in \Omega_i} \left(I_4 + \Lambda_i^{dof} \right) |\Omega_i^{dof}| \mathcal{S}_i \qquad \text{or} \tag{5.31}$$

$$= \sum_{r \in \Omega_j} \frac{1}{N_r} \sum_{i : r \in \Omega_i} (I_4 + \Lambda_i^r) |\Omega_i^r| \mathcal{S}_i + \sum_{r+1/2 \in \Omega_j} \frac{1}{2} \sum_{i : r+1/2 \in \Omega_i} \left(I_4 + \Lambda_i^{r+1/2} \right) |\Omega_i^{r+1/2}| \mathcal{S}_i, \tag{5.32}$$

where $|\Omega_i^r|$ (resp. $|\Omega_i^{r+\frac{1}{2}}|$) is the sub-volume in cell i linked to node r (resp. mid-edge $r + \frac{1}{2}$). See Figure 3 in the case $0 < \theta < 1$.

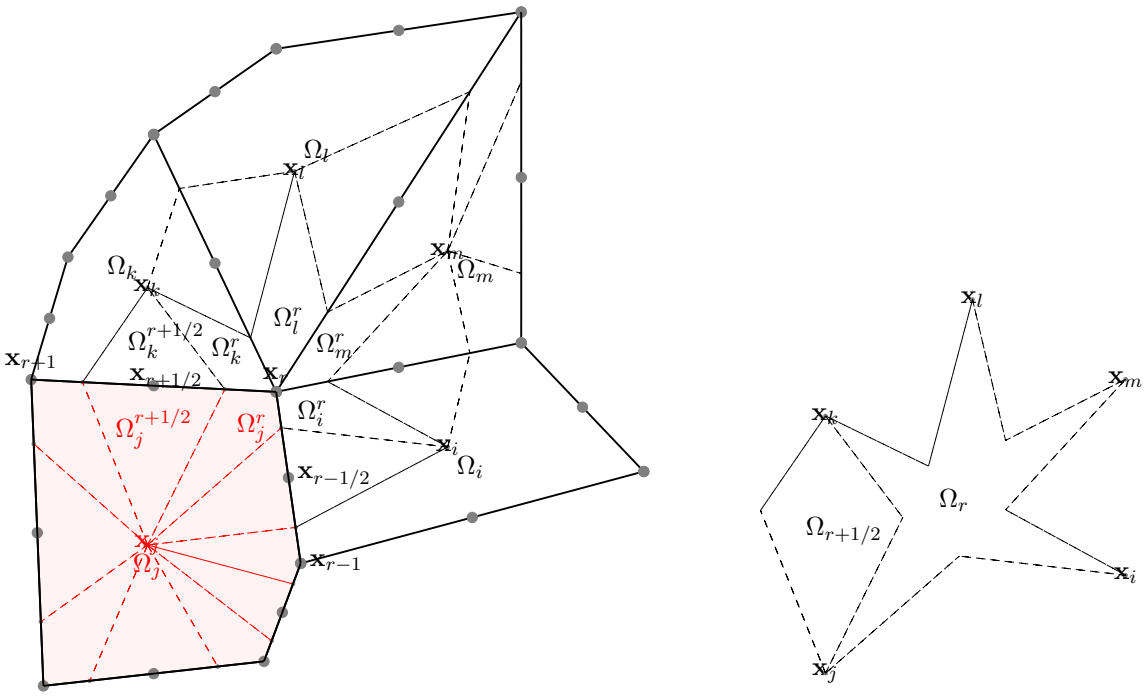

FIGURE 3. Left: Generic cell Ω_j with its sub-cells Ω_j^{dof} defined around nodes r and edges $r+\frac{1}{2}$. Right: Associated composite dual cell around node $\mathbf{x}_r$: $\Omega_r = \cup_{i;r\in i}\Omega_i^r$, and around edge $\mathbf{x}_{r+\frac{1}{2}}$: $\Omega_{r+1/2} = \Omega_j^{r+1/2} \cup \Omega_k^{r+1/2}$.

For example: for $\theta = 1$, noting k the adjacent cell of j with respect to local edge $(r+1/2)$ (Figure 3), the formula (5.31) gives:

$$\sum_{r+1/2\in\Omega_j} \frac{1}{2}\left(\left(I_4 + \Lambda_k^{r+1/2}\right)|\Omega_k^{r+1/2}|\mathcal{S}_k + \left(I_4 + \Lambda_j^{r+1/2}\right)|\Omega_j^{r+1/2}|\mathcal{S}_j\right). \tag{5.33}$$

Knowing that $\Lambda_j^{r+1/2} = -\Lambda_k^{r+1/2}$, this formula gives a fully upwinding discretization at $r+1/2$ in cell j. Indeed, if one consider that all eigenvalues are all strictly positives (resp. all strictly negatives), we recover a local source flux at $r+1/2$ equal to $|\Omega_j^{r+1/2}|\mathcal{S}_j$ (resp. $|\Omega_k^{r+1/2}|\mathcal{S}_k$).

For edge discretization flux at $r+1/2$ ($\theta = 1$), a more natural extension in terms of geometric interpretation of the 1D case with integral form (5.8),

$$\sum_{r+1/2\in\Omega_j} \frac{1}{2}\left(I_4 - \Lambda_j^{r+1/2}\right)\left(|\Omega_k^{r+1/2}|\mathcal{S}_k + |\Omega_j^{r+1/2}|\mathcal{S}_j\right), \tag{5.34}$$

Unfortunately, this formula (5.34) has not yet been tested and like the 1D case, it does not satisfy the full upwinding property. In the following, using the discretizations 5.4.2 and 5.4.3, we now compare them against the usual centered $\mathcal{Q}_j = S_j|\Omega_j|$.

5.5. NUMERICAL RESULTS

We present here some numerical tests to compare the two discretizations of the source term presented in the Section 5.4. The computational domain is $\Omega = [0,1]^2$. Some analytical solutions are known for all the test cases, are stationary, and thus we also use them as initial data. We recall that only Dirichlet boundary conditions are considered.

The timestep is given by $\Delta t = C_{CFL}\Delta x$ with the coefficient C_{CFL} depending on the test case. We perform convergence analysis on Cartesian and random meshes. A random mesh is defined as a uniform Cartesian mesh where the inner nodes are randomly perturbed (see Figure 4). The final time is $T = 0.1$.

For each numerical case, we display the results obtained with $\theta = 0, \frac{1}{2}, \frac{\pi}{4}, 1$. The values $0, \frac{1}{2}, 1$ correspond respectively to a purely node scheme, to a scheme exactly halfway between node and edge scheme, and to a purely edge scheme. The value $\theta = \frac{\pi}{4} \approx 0.78$ correspond to the value obtained when considering a conical mesh with curved edges and letting the curvature tend to zero (see [54]).

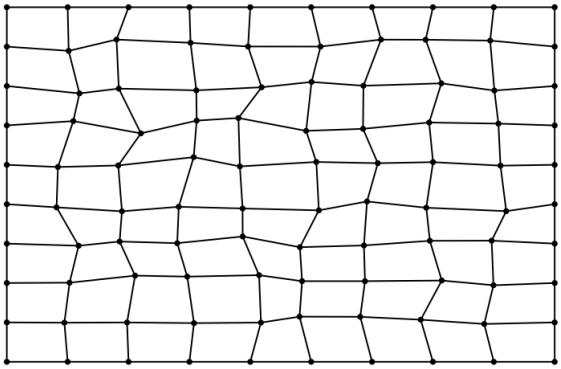

FIGURE 4. Example of a random mesh.

5.5.1. **1D gravity.** We set $\mathbf{u} = \mathbf{0}$. We add a source term of the form $\mathcal{S} = (0, \rho\nabla\zeta, \rho\nabla\zeta \cdot \mathbf{u})^T$ with $\zeta(\mathbf{x}) = \mathbf{g} \cdot \mathbf{x}$ and $\mathbf{g} = (-g, 0)$. In the 1D framework, Equation (5.1) reads as:

$$\frac{dP}{dx} = -\rho g, \qquad P = (\gamma - 1)\rho E. \tag{5.35}$$

We choose P of the form: $P(x) = 2g(x_0 - x) > 0$, $x_0 = 1.5$. The density is $\rho = 2$. We set $C_{CFL} = 0.25$. The results obtained on an uniform mesh with different discretizations of the source term are given in Figures 5, 6, 7 and 8 (resp. for $\theta = 0, \frac{1}{2}, \frac{\pi}{4}, 1$). The results obtained on a random mesh with different discretizations of the source term are given in Figures 9, 10, 11 and 12. (resp. for $\theta = 0, \frac{1}{2}, \frac{\pi}{4}, 1$). We can see that all the methods are first order convergent for $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$. However the purely nodal scheme ($\theta = 0$) is problematic because it seems not to be first order convergent.

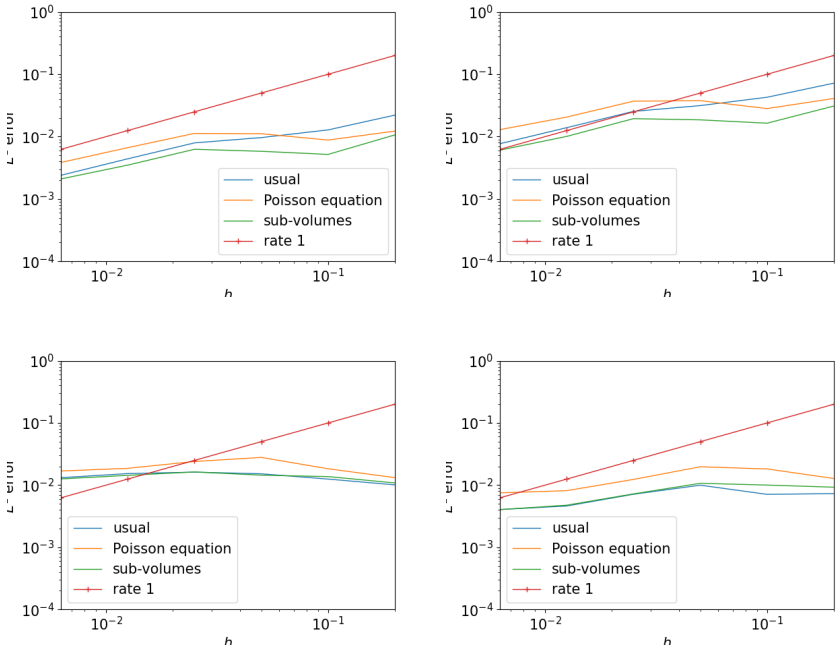

FIGURE 5. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on uniform meshes with $\theta = 0$.

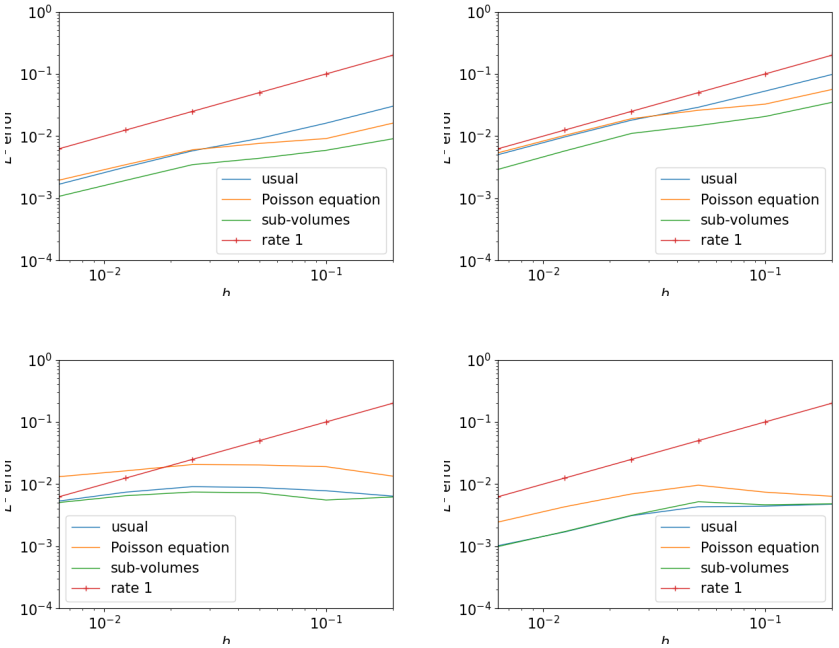

FIGURE 6. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on uniform meshes with $\theta = \frac{1}{2}$.

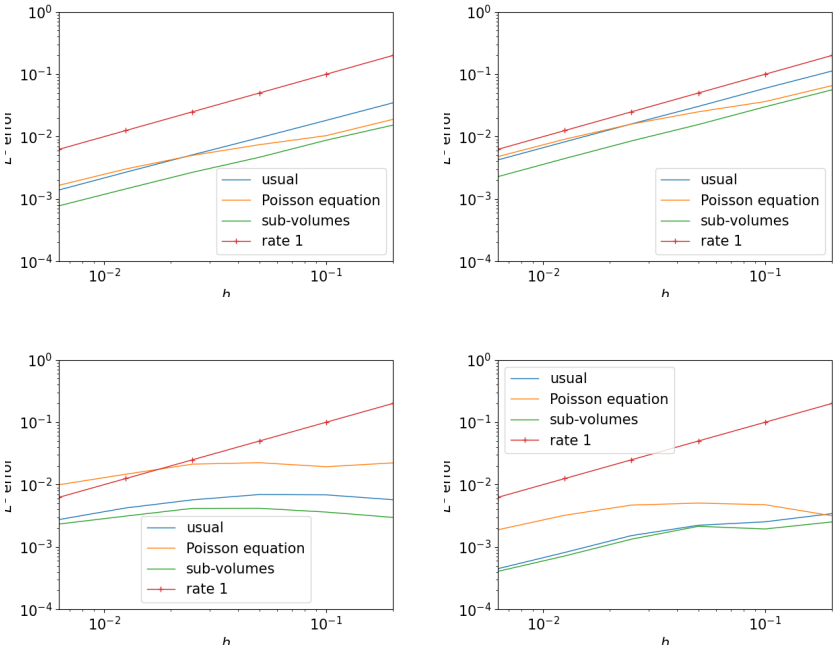

FIGURE 7. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on uniform meshes with $\theta = \frac{\pi}{4}$.

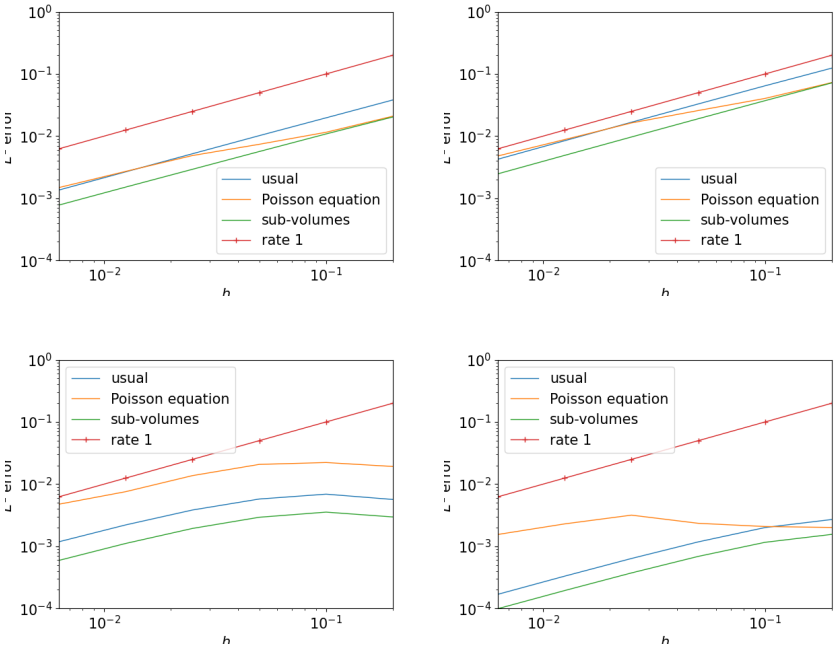

FIGURE 8. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on uniform meshes with $\theta = 1$.

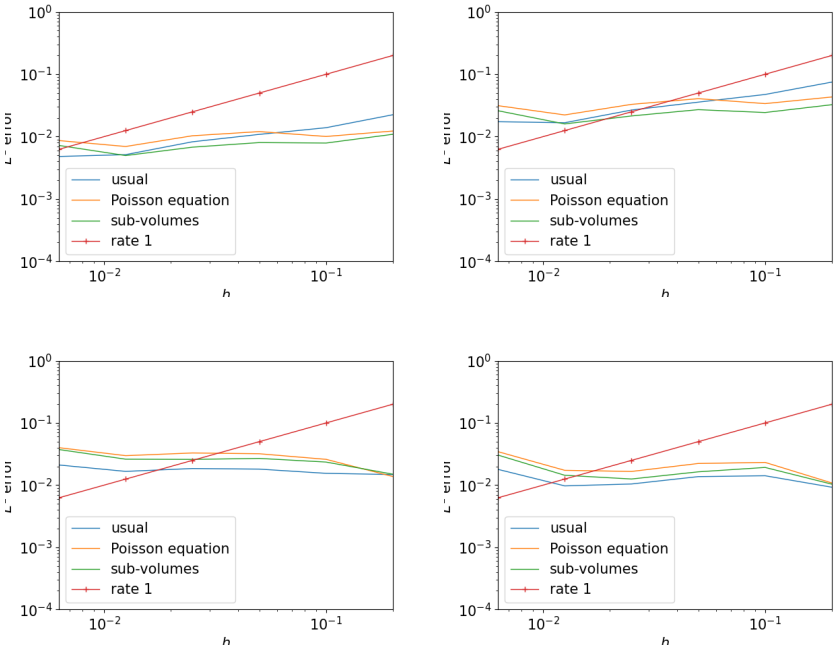

FIGURE 9. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on random meshes with $\theta = 0$.

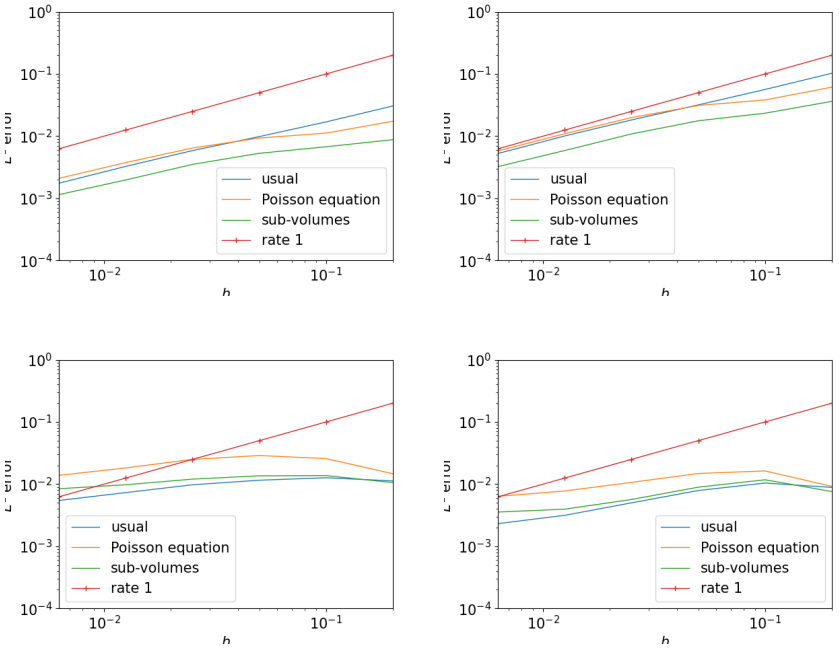

FIGURE 10. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on random meshes with $\theta = \frac{1}{2}$.

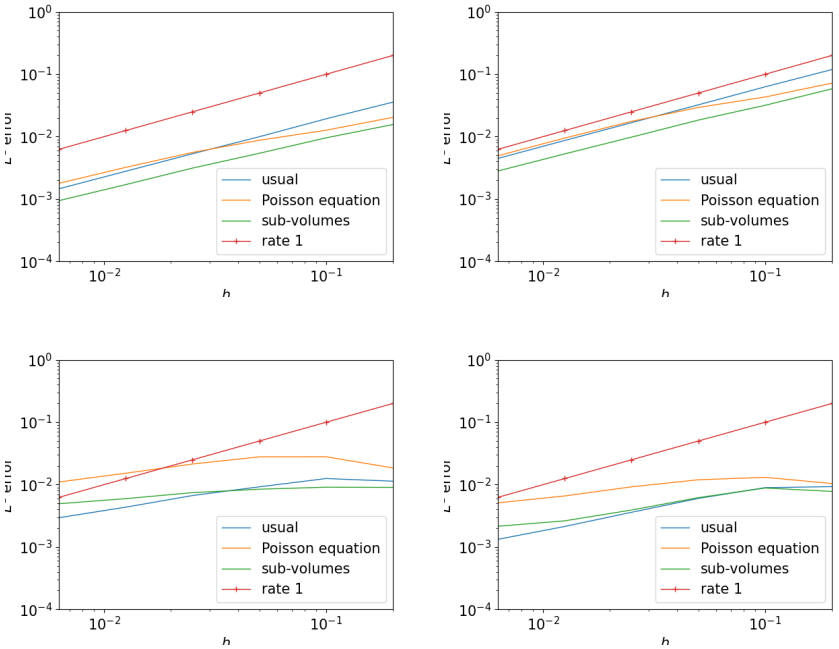

FIGURE 11. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on uniform meshes with $\theta = \frac{\pi}{4}$.

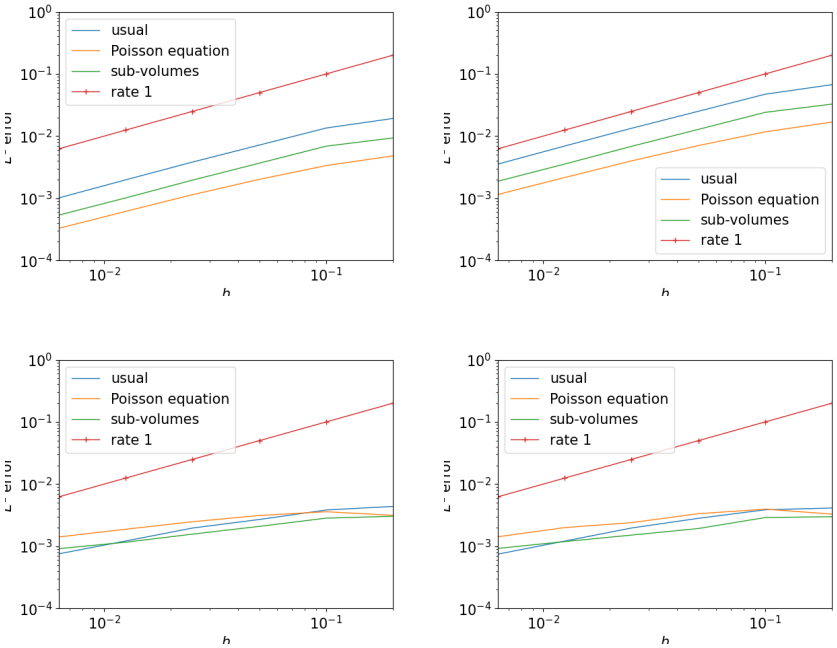

FIGURE 12. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.1 on random meshes with $\theta = 1$.

5.5.2. **2D gravity.** We set $\mathbf{u} = \mathbf{0}$. We add a source term of the form $\mathcal{S} = (0, \rho\nabla\zeta, \rho\nabla\zeta \cdot \mathbf{u})^T$ with $\zeta(\mathbf{x}) = -g\|\mathbf{x} - \mathbf{x}_0\|^2$. Equation (5.1) reads as:

$$\nabla P = \rho\nabla\zeta, \qquad P = (\gamma - 1)\rho E.$$

A solution writes: $P = \rho = e^{\zeta}$ and $E = 1/(\gamma - 1)$. We set $C_{CFL} = 0.25$. The results obtained on an uniform mesh with different discretizations of the source term are given in Figures 13, 14 and 15 (resp. for $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$). The results obtained on a random mesh with different discretizations of the source term are given in Figures 16, 17 and 18 (resp. for $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$). The schemes are all first order convergent for $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$. However we also observed issues with the purely nodal scheme ($\theta = 0$) and we chose not to plot the results in this case.

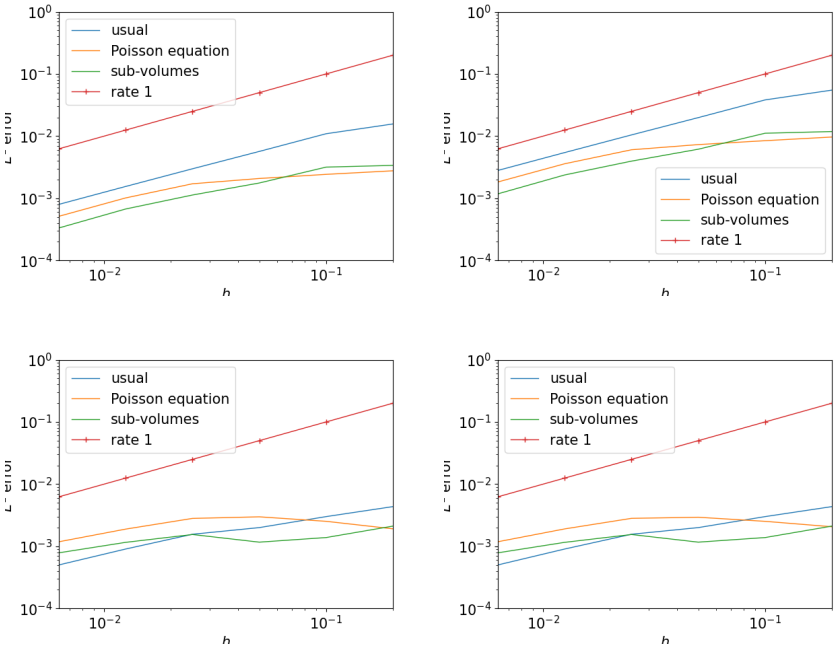

FIGURE 13. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.2 on uniform meshes with $\theta = \frac{1}{2}$.

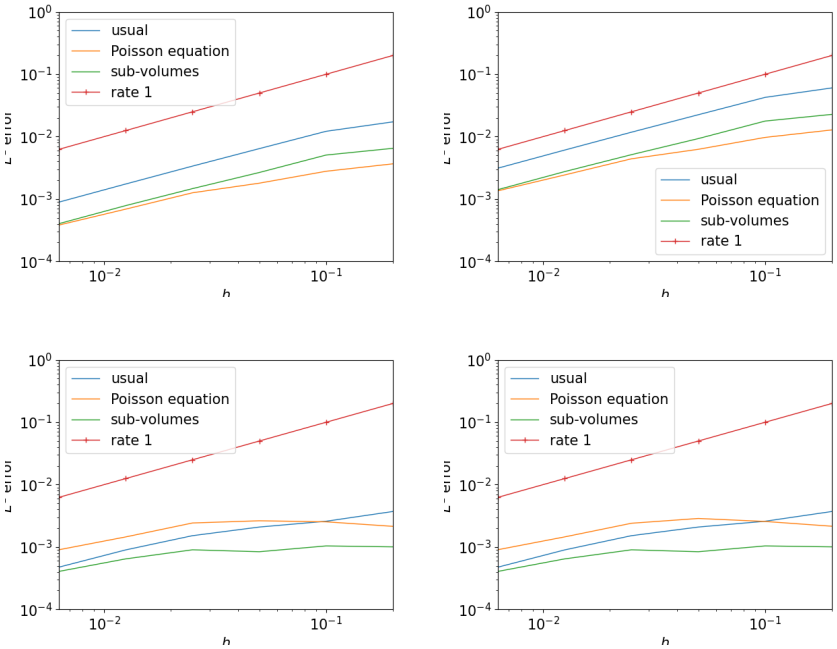

FIGURE 14. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.2 on uniform meshes with $\theta = \frac{\pi}{4}$.

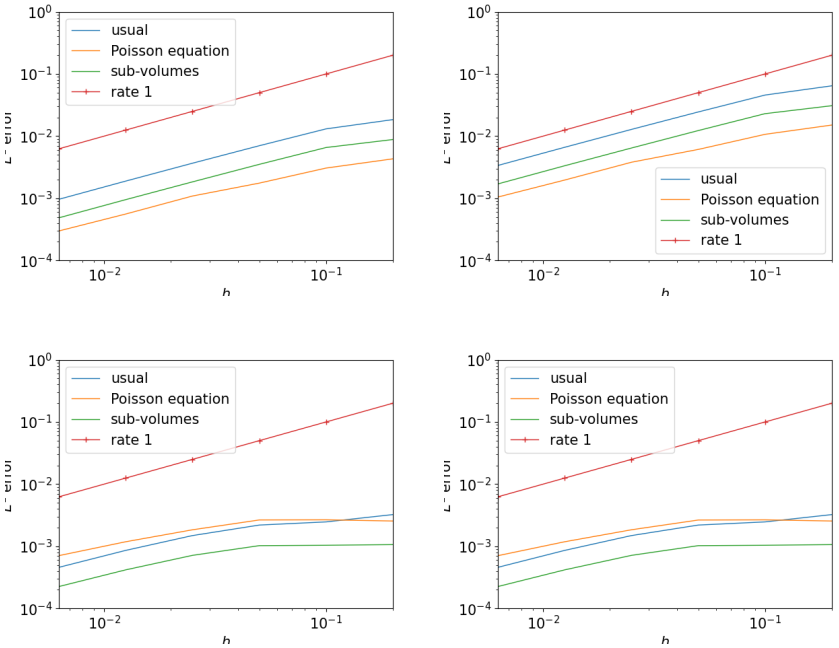

FIGURE 15. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.2 on uniform meshes with $\theta = 1$.

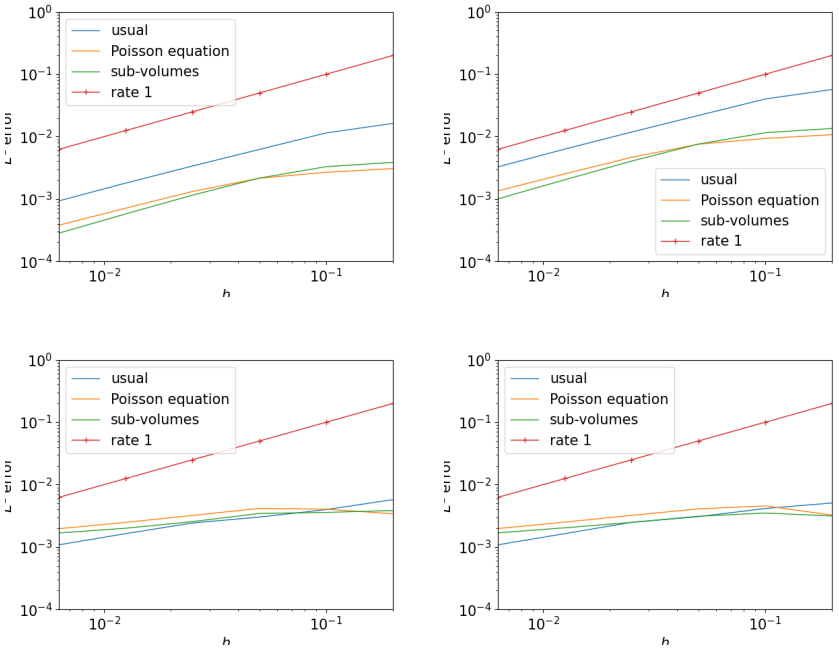

FIGURE 16. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.2 on random meshes with $\theta = \frac{1}{2}$.

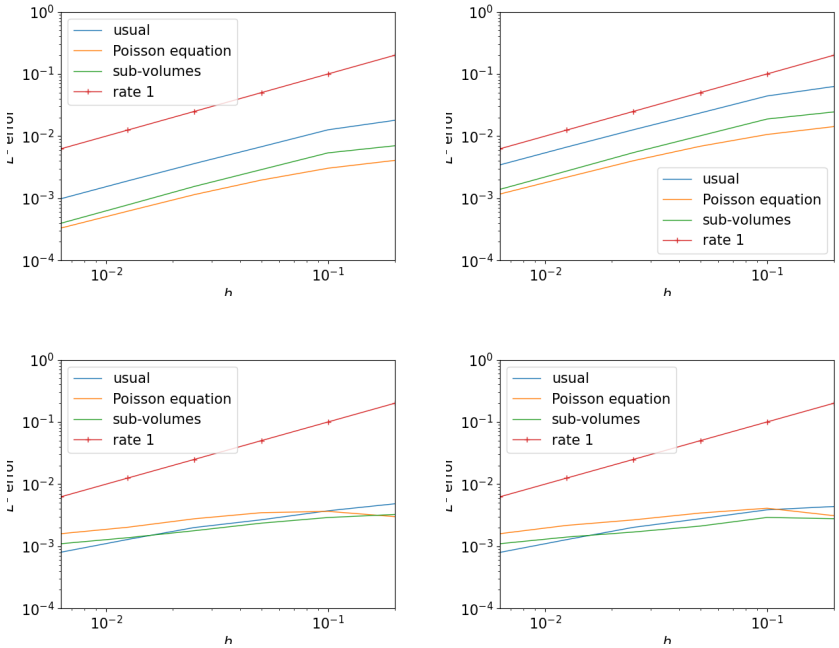

FIGURE 17. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.2 on random meshes with $\theta = \frac{\pi}{4}$.

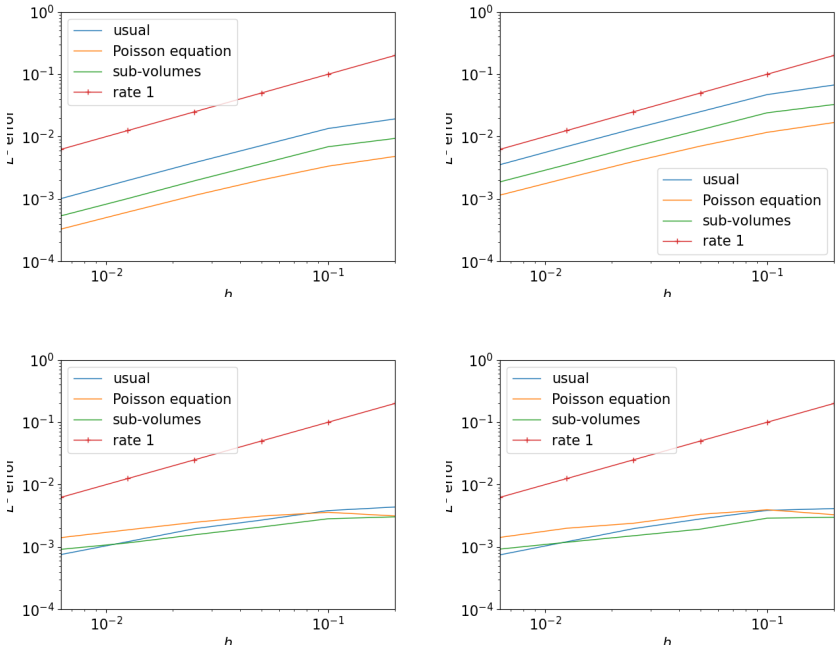

FIGURE 18. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.2 on random meshes with $\theta = 1$.

5.5.3. **Friction source term.** The source term is given by $\mathcal{S} = (0, -\lambda\rho\mathbf{u}, -\alpha\lambda\rho\|\mathbf{u}\|^2)^T$ with $\alpha = \gamma/(\gamma-1)$ and $\lambda = 1$. A solution to the system (5.1) writes:

$$\rho = 1, \qquad \mathbf{u} = \frac{1}{\sqrt{2}}(1,1), \qquad E(\mathbf{x}) = -\frac{\lambda}{\gamma-1}\mathbf{u}\cdot\mathbf{x} + 1 + \frac{2\lambda}{\gamma-1}.$$

One can easily check that $E > \|\mathbf{u}\|^2/2$. We set $C_{CFL} = 0.25$. The results obtained on an uniform mesh with different discretizations of the source term are given in Figures 19, 20 and 21 (resp. for

$\theta = \frac{1}{2}, \frac{\pi}{4}, 1$). The results obtained on a random mesh with different discretizations of the source term are given in Figures 22, 23, and 24 (resp. for $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$). The methods all are first order convergent for $\theta = \frac{1}{2}, \frac{\pi}{4}, 1$. However we also observed issues with the purely nodal scheme ($\theta = 0$) and we chose not to plot the results in this case.

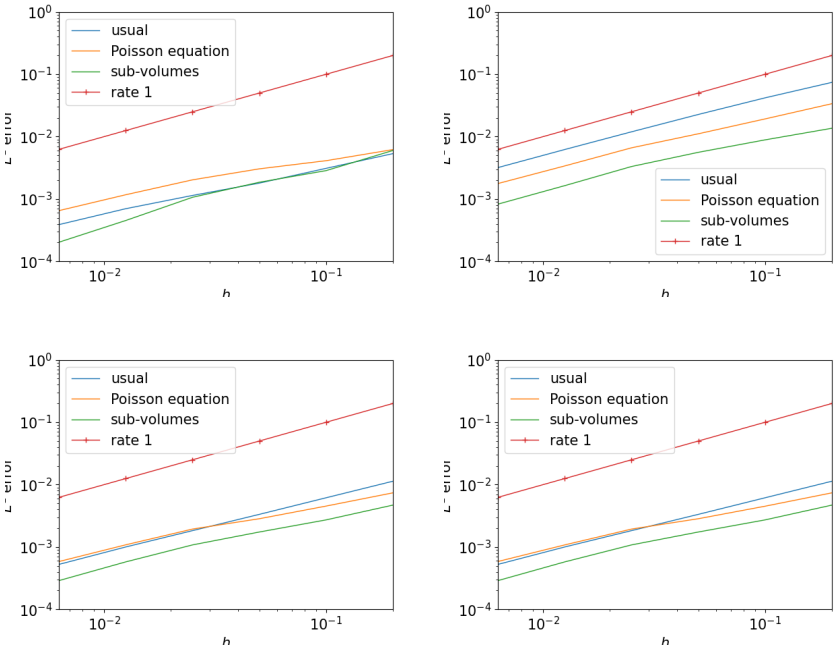

FIGURE 19. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.3 on uniform meshes with $\theta = \frac{1}{2}$.

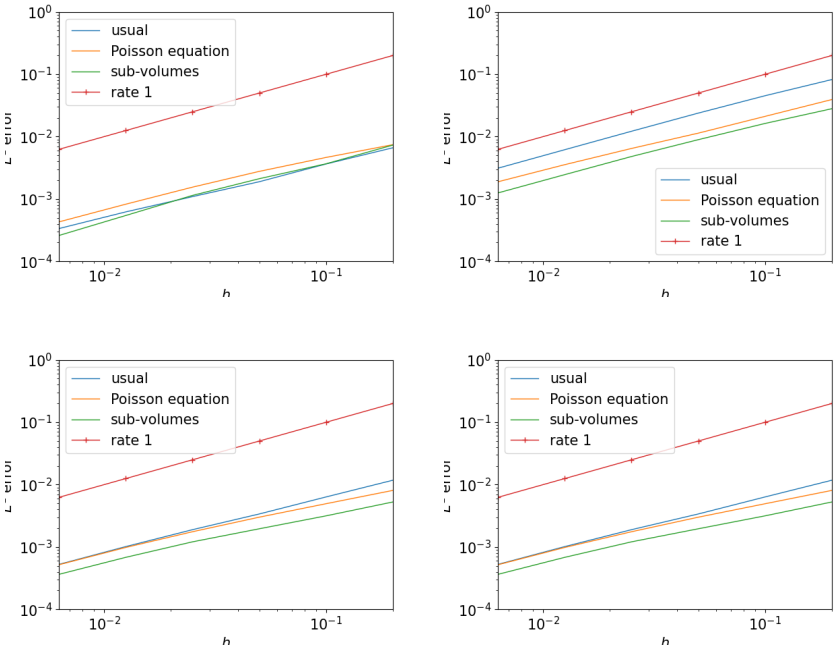

FIGURE 20. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.3 on uniform meshes with $\theta = \frac{\pi}{4}$.

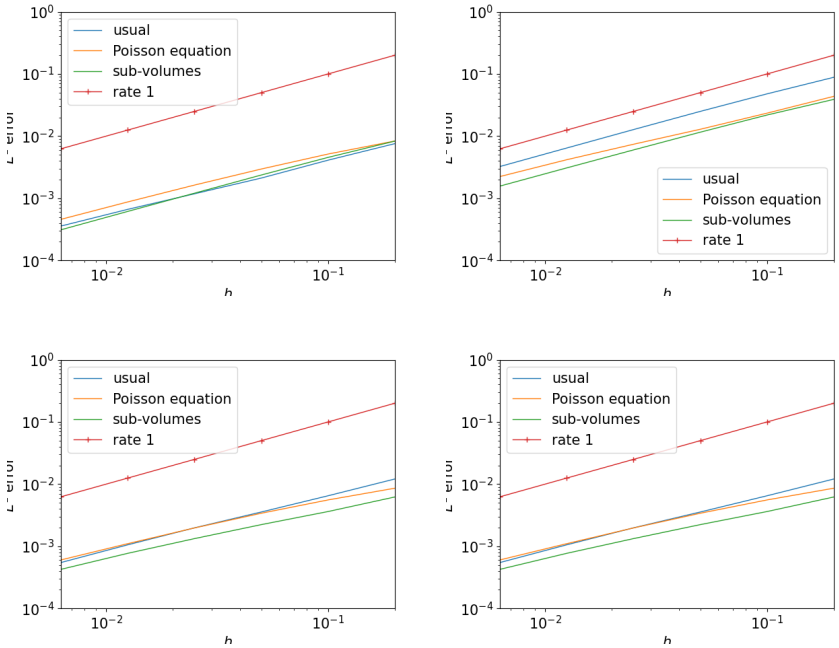

FIGURE 21. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.3 on uniform meshes with $\theta = 1$.

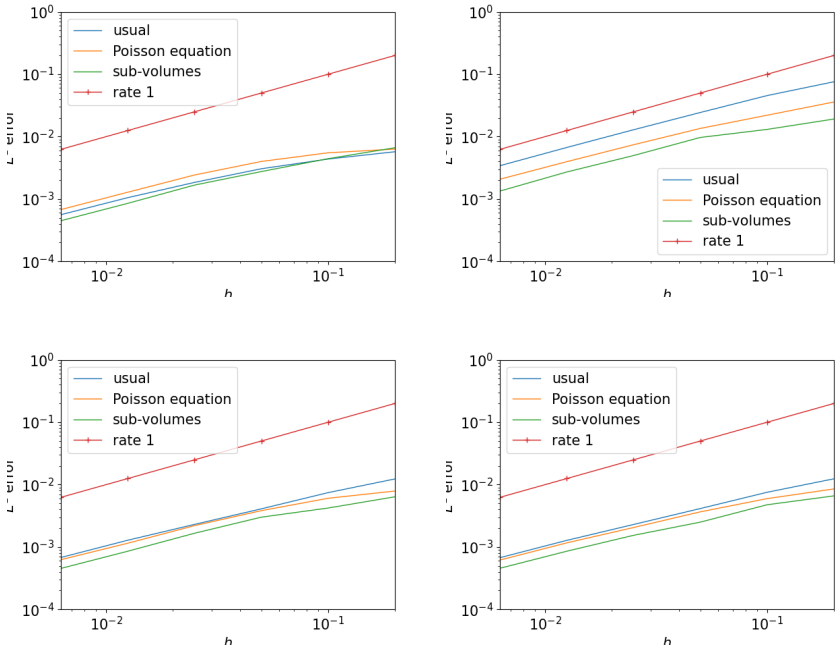

FIGURE 22. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.3 on random meshes with $\theta = \frac{1}{2}$.

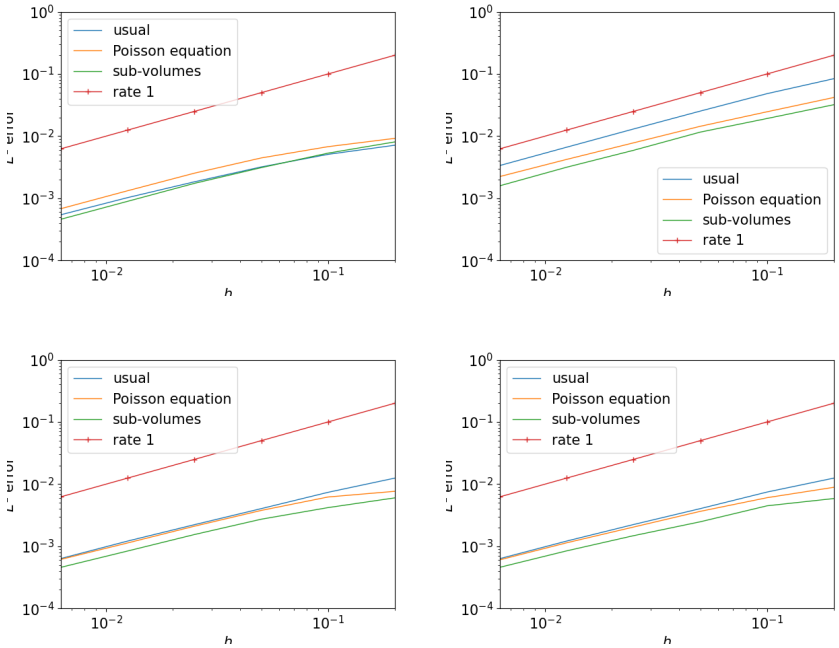

FIGURE 23. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.3 on random meshes with $\theta = \frac{\pi}{4}$.

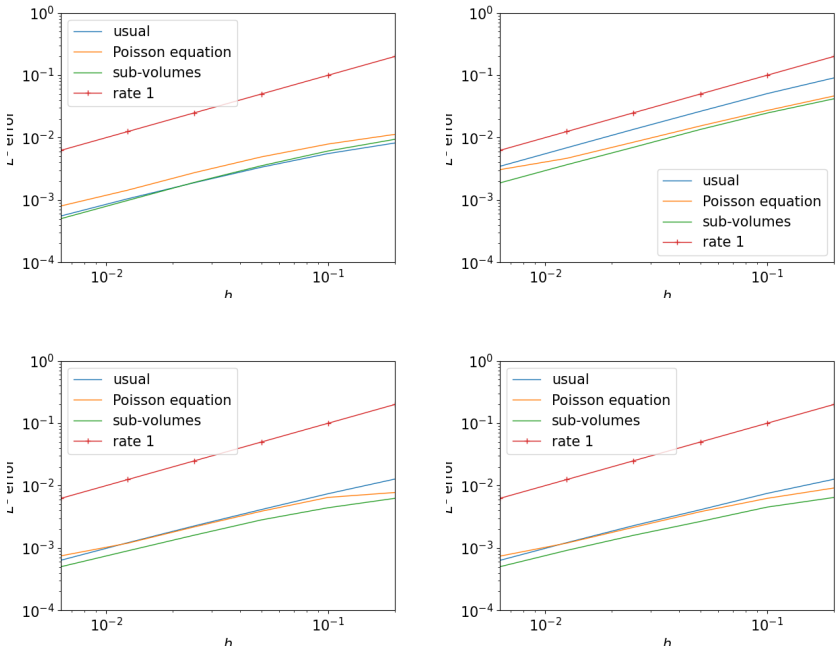

FIGURE 24. L^1 error on ρ (up left), ρE (up right), ρu_1 (down left) ρu_2 (down right) for the test case 5.5.3 on random meshes with $\theta = 1$.

5.6. Discussion and conclusion

In this work we presented several methods to compute the numerical source term of the 2D Euler equations using ideas from [2, 54]. The fluxes are discretized using a composite scheme that generalizes the VFFC scheme (see [44]) to unstructured 2D meshes. We compared the "classical" (or "usual") discretization of the source term (discretized at the center of computational cells) against a local and a global approach. The local approach uses the sub-volumes of computational cells, and performs

slightly better for a computational complexity of the same order of magnitude. The global approach solves a Poisson equation on the whole domain, however it is much more costful and does not perform significantly better.

Bibliography

1. Robert A Adams and John JF Fournier, Sobolev spaces, Elsevier, 2003.
2. Francois Alouges, Jean-Michel Ghidaglia, and Marc Tajchman, On the interaction of upwinding and forcing for nonlinear hyperbolic systems of conservation laws, CMLA (1999), 1–12 (en), Available at https://www.researchgate.net/profile/Francois-Alouges/publication/2354257_On_the_Interaction_of_Upwinding_and_Forcing_for_Nonlinear_Hyperbolic_Systems_of_Conservation_Laws/links/541f083d0cf241a65a1aa84f/On-the-Interaction-of-Upwinding-and-Forcing-for-Nonlinear-Hyperbolic-Systems-of-Conservation-Laws.pdf.
3. Boris Andreianov, Franck Boyer, and Florence Hubert, Discrete duality finite volume schemes for leray- lions- type elliptic problems on general 2d meshes, Numerical Methods for Partial Differential Equations: An International Journal **23** (2007), no. 1, 145–195.
4. Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoit Perthame, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, SIAM Journal on Scientific Computing **25** (2004), no. 6, 2050–2065 (en).
5. Martin Z Bazant, Katsuyo Thornton, and Armand Ajdari, Diffuse-charge dynamics in electrochemical systems, Physical review E **70** (2004), no. 2, 021506.
6. David J Benson, Computational methods in lagrangian and eulerian hydrocodes, Computer methods in Applied mechanics and Engineering **99** (1992), no. 2-3, 235–394.
7. Jonas P. Berberich, Praveen Chandrashekar, and Christian Klingenberg, High order well-balanced finite volume methods for multi-dimensional systems of hyperbolic balance laws, Computers & Fluids **219** (2021), 104858 (en).
8. Aude Bernard-Champmartin, Erwan Deriaz, Philippe Hoch, Gerald Samba, and Michael Schaefer, Extension of centered hydrodynamical schemes to unstructured deforming conical meshes: the case of circles, ESAIM: Proceedings, vol. 38, EDP Sciences, 2012, pp. 135–162.
9. Marianne Bessemoulin-Chatard, Giulia Lissoni, and Hélène Mathis, Numerical analysis of ddfv schemes for semiconductors energy-transport models, Computational and Applied Mathematics **41** (2022), no. 1, 15.
10. X Blanc, P Hoch, and C Lasuen, Proof of uniform convergence for a cell-centered ap discretization of the hyperbolic heat equation on conical meshes, working paper or preprint, April 2022.
11. Xavier Blanc, Vincent Delmas, and Philippe Hoch, Asymptotic preserving schemes on conical unstructured 2d meshes, International Journal for Numerical Methods in Fluids **93** (2021), no. 8, 2763–2802.
12. Xavier Blanc, Philippe Hoch, and Clément Lasuen, Positive composite finite volume schemes for the diffusion equation on unstructured meshes, working paper or preprint, December 2022.
13. François Bouchut, Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balanced schemes for sources, Frontiers in Mathematics, Birkäuser, Basel ; Boston, 2004.
14. Benjamin Boutin, Erwan Deriaz, Philippe Hoch, and Pierre Navaro, Extension of ALE methodology to unstructured conical meshes, ESAIM: Proceedings, vol. 32, EDP Sciences, 2011, pp. 31–55.
15. P. Brufau, M. E. Vázquez-Cendón, and P. García-Navarro, A numerical model for the flooding and drying of irregular domains: 1d and 2d modelling of shallow water flow, International Journal for Numerical Methods in Fluids **39** (2002), no. 3, 247–275 (en).
16. Christophe Buet, Bruno Després, and Emmanuel Franck, Design of asymptotic preserving finite volume schemes for the hyperbolic heat equation on unstructured meshes, Numerische Mathematik **122** (2012), no. 2, 227–278.
17. ———, Design of asymptotic preserving finite volume schemes for the hyperbolic heat equation on unstructured meshes, Numerische Mathematik **122** (2012), 227–278.
18. Clément Cancès, Claire Chainais-Hillairet, and Stella Krell, Numerical analysis of a nonlinear free-energy diminishing discrete duality finite volume scheme for convection diffusion equations, Computational Methods in Applied Mathematics **18** (2018), no. 3, 407–432.
19. Alberto Canestrelli, Annunziato Siviglia, Michael Dumbser, and Eleuterio F. Toro, Well-balanced high-order centred schemes for non-conservative hyperbolic systems. applications to shallow water equations with fixed and mobile bed, Advances in Water Resources **32** (2009), no. 6, 834–844 (en).
20. Xiulei Cao and Huaxiong Huang, An adaptive conservative finite volume method for poisson-nernst-planck equations on a moving mesh, Communications in Computational Physics **26** (2019), 389–412.
21. Jerome Cartailler, Taekyung Kwon, Rafael Yuste, and David Holcman, Deconvolution of voltage sensor time series and electro-diffusion modeling reveal the role of spine geometry in controlling synaptic strength, Neuron **97** (2018), no. 5, 1126–1136.

22. Sydney Cash and Rafael Yuste, Linear summation of excitatory inputs by ca1 pyramidal neurons, Neuron **22** (1999), no. 2, 383–394.
23. Manuel Castro, José M. Gallardo, Juan A. López-GarcÍa, and Carlos Parés, Well-balanced high order extensions of godunov's method for semilinear balance laws, SIAM Journal on Numerical Analysis **46** (2008), no. 2, 1012–1039 (en).
24. Manuel J. Castro and Carlos Parés, Well-balanced high-order finite volume methods for systems of balance laws, Journal of Scientific Computing **82** (2020), no. 2, 48 (en).
25. M.J. Castro, T. Morales de Luna, and C. Parés, Chapter 6 - well-balanced schemes and path-conservative numerical methods, Handbook of numerical analysis, vol. 18, p. 131–175, Elsevier, 2017, Citation Key: CASTRO2017131.
26. M. J. Castro Díaz, T. Chacón Rebollo, E. D. Fernández-Nieto, and Carlos Parés, On well-balanced finite volume methods for nonconservative nonhomogeneous hyperbolic systems, SIAM Journal on Scientific Computing **29** (2007), no. 3, 1093–1126 (en).
27. Claire Chainais-Hillairet, Stella Krell, and Alexandre Mouton, Convergence analysis of a ddfv scheme for a system describing miscible fluid flows in porous media, Numerical Methods for Partial Differential Equations **31** (2015), no. 3, 723–760.
28. Praveen Chandrashekar and Markus Zenk, Well-balanced nodal discontinuous galerkin method for euler equations with gravity, Journal of Scientific Computing **71** (2017), no. 3, 1062–1093 (en).
29. TE Chater and Y Goda, My neighbour hetero—deconstructing the mechanisms underlying heterosynaptic plasticity, Current Opinion in Neurobiology **67** (2021), 106–114.
30. Jehanzeb Hameed Chaudhry, Jeffrey Comer, Aleksei Aksimentiev, and Luke N Olson, A stabilized finite element method for modified poisson-nernst-planck equations to determine ion flow through a nanopore, Communications in computational physics **15** (2014), no. 1, 93–125.
31. P.G. Ciarlet, Basic error estimates for elliptic problems, Finite Element Methods (Part 1), Handbook of Numerical Analysis, vol. 2, Elsevier, 1991, pp. 17–351.
32. VH Cornejo, N Ofer, and R Yuste, Voltage compartmentalization in dendritic spines in vivo, Science **375** (2022), no. 6576, 82–86.
33. Bruno Després and Constant Mazeran, Lagrangian gas dynamics in two dimensions and lagrangian systems, Archive for Rational Mechanics and Analysis **178** (2005), no. 3, 327–372.
34. Vivien Desveaux, Markus Zenk, Christophe Berthon, and Christian Klingenberg, A well-balanced scheme to capture non-explicit steady states in the euler equations with gravity: Well-balanced scheme for the euler equations with gravity, International Journal for Numerical Methods in Fluids **81** (2016), no. 2, 104–127 (en).
35. Komla Domelevo and Pascal Omnes, A finite volume method for the laplace equation on almost arbitrary two-dimensional grids, ESAIM: Mathematical Modelling and Numerical Analysis **39** (2005), no. 6, 1203–1249.
36. S Druckmann, L Feng, B Lee, C Yook, T Zhao, JC Magee, and J Kim, Structured synaptic connectivity between hippocampal regions, Neuron **81** (2014), no. 3, 629–640.
37. Bob Eisenberg and Weishi Liu, Poisson–nernst–planck systems for ion channels with permanent charges, SIAM Journal on Mathematical Analysis **38** (2007), no. 6, 1932–1966.
38. E. Guerrero Fernández, M. J. Castro Díaz, M. Dumbser, and T. Morales De Luna, An arbitrary high order well-balanced ader-dg numerical scheme for the multilayer shallow-water model with variable density, Journal of Scientific Computing **90** (2022), no. 1, 52 (en).
39. Allen Flavell, Julienne Kabre, and Xiaofan Li, An energy-preserving discretization for the poisson–nernst–planck equations, Journal of Computational Electronics **16** (2017), no. 2, 431–441.
40. Allen Flavell, Michael Machen, Bob Eisenberg, Julienne Kabre, Chun Liu, and Xiaofan Li, A conservative finite difference scheme for poisson–nernst–planck equations, Journal of Computational Electronics **13** (2014), no. 1, 235–249.
41. Emmanuel Franck, Design and numerical analysis of asymptotic preserving schemes on unstructured meshes. application to the linear transport and friedrichs systems, HAL **2012** (2012), 1–241.
42. Peter Fromherz, Electrical interfacing of nerve cells and semiconductor chips, ChemPhysChem **3** (2002), no. 3, 276–284.
43. Perle Geoffroy-Donders, Grégoire Allaire, Georgios Michailidis, and Olivier Pantz, Coupled optimization of macroscopic structures and lattice infill, International Journal for Numerical Methods in Engineering **123** (2022), no. 13, 2963–2985.
44. J.-M. Ghidaglia, Anela Kumbaro, and Gérard Le Coq, Une méthode volumes finis à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation, C.R. Acad. Sci. Paris **322** (1996), 981–988.
45. Matteo Giacomini, Olivier Pantz, and Karim Trabelsi, Volumetric expressions of the shape gradient of the compliance in structural shape optimization, arXiv preprint arXiv:1701.05762 (2017), 1–28.
46. Laurent Gosse and Alain-Yves Leroux, Un schema-equilibre adapte aux lois de conservation scalaires non-homogenes, CR Acad. Sci. Paris Sér. I Math (1996), 543–546 (fr).
47. J. M. Greenberg and A. Y. Leroux, A well-balanced scheme for the numerical processing of source terms in hyperbolic equations, SIAM Journal on Numerical Analysis **33** (1996), no. 1, 1–16 (en).
48. Dongdong He and Kejia Pan, An energy preserving finite difference scheme for the poisson–nernst–planck system, Applied Mathematics and Computation **287** (2016), 214–223.

49. Dongdong He, Kejia Pan, and Xiaoqiang Yue, A positivity preserving and free energy dissipative difference scheme for the poisson–nernst–planck system, Journal of Scientific Computing **81** (2019), no. 1, 436–458.
50. Frédéric Hecht, New development in freefem++, Journal of numerical mathematics **20** (2012), no. 3-4, 251–266.
51. Frédéric Hecht, Olivier Pironneau, A Le Hyaric, and K Ohtsuka, Freefem++ manual, Laboratoire Jacques Louis Lions (2005), 1–188.
52. Heike Hering and Morgan Sheng, Dentritic spines : structure, dynamics and regulation, Nature Reviews Neuroscience **2** (2001), no. 12, 880 – 888.
53. Francois Hermeline, A finite volume method for the approximation of diffusion operators on distorted meshes, Journal of computational Physics **160** (2000), no. 2, 481–499.
54. Philippe Hoch, Nodal extension of approximate riemann solvers and nonlinear high order reconstruction for finite volume method on unstructured polygonal and conical meshes: the homogeneous case, HAL (2022), 1–90.
55. Alan L Hodgkin and Andrew F Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, The Journal of physiology **117** (1952), no. 4, 500.
56. David Holcman and Rafael Yuste, The new nanophysiology: regulation of ionic flow in neuronal subcompartments, Nature Reviews Neuroscience **16** (2015), no. 11, 685–692.
57. Jingwei Hu and Xiaodong Huang, A fully discrete positivity-preserving and energy-dissipative finite difference scheme for poisson–nernst–planck equations, Numerische Mathematik **145** (2020), no. 1, 77–115.
58. JJB Jack, D Noble, and RW Tsien, Linear cable theory, Electric current flow in excitable cells (1975), 25–66.
59. Ansgar Jüngel, Quasi-hydrodynamic semiconductor equations, vol. 41, Springer Science & Business Media, 2001.
60. Aditya S Khair and Todd M Squires, Ion steric effects on electrophoresis of a colloidal particle, Journal of fluid mechanics **640** (2009), 343–356.
61. Brian J Kirby, Micro-and nanoscale fluid mechanics: transport in microfluidic devices, Cambridge university press, 2010.
62. Christof Koch, Cable theory in neurons with active, linearized membranes, Biological cybernetics **50** (1984), no. 1, 15–33.
63. Satoru Kondo, Shinichi Kohsaka, and Shigeo Okabe, Long-term changes of spine dynamics and microglia after transient peripheral immune response triggered by lps in vivo, Molecular brain **4** (2011), 27.
64. KFH Lee, C Soares, J-P Thivierge, and J-C Béïque, Correlated synaptic inputs drive dendritic calcium amplification and cooperative plasticity during clustered synapse development, Neuron **89** (2016), no. 4, 784–799.
65. Randall J. LeVeque, Balancing source terms and flux gradients in high-resolution godunov methods: The quasi-steady wave-propagation algorithm, Journal of Computational Physics **146** (1998), no. 1, 346–365 (en).
66. Chun Liu, Cheng Wang, Steven Wise, Xingye Yue, and Shenggao Zhou, A positivity-preserving, energy stable and convergent numerical scheme for the poisson-nernst-planck system, Mathematics of Computation **90** (2021), no. 331, 2071–2106.
67. Hailiang Liu, Zhongming Wang, Peimeng Yin, and Hui Yu, Positivity-preserving third order dg schemes for poisson–nernst–planck equations, Journal of Computational Physics **452** (2022), 110777.
68. M London and M Häusser, Dendritic computation, Annu Rev Neurosci **28** (2005), 503–532.
69. Courtney L Lopreore, Thomas M Bartol, Jay S Coggan, Daniel X Keller, Gina E Sosinsky, Mark H Ellisman, and Terrence J Sejnowski, Computational modeling of three-dimensional electrodiffusion in biological systems: application to the node of ranvier, Biophysical journal **95** (2008), no. 6, 2624–2635.
70. Benzhuo Lu, Michael J Holst, J Andrew McCammon, and YC Zhou, Poisson–nernst–planck equations for simulating biomolecular diffusion–reaction processes i: Finite element solutions, Journal of computational physics **229** (2010), no. 19, 6979–6994.
71. Pierre-Henri Maire, Rémi Abgrall, Jérôme Breil, and Jean Ovadia, A cell-centered lagrangian scheme for two-dimensional compressible flow problems, SIAM Journal on Scientific Computing **29** (2007), no. 4, 1781–1824.
72. Sanjay R Mathur and Jayathi Y Murthy, A multigrid method for the poisson–nernst–planck equations, International Journal of Heat and Mass Transfer **52** (2009), no. 17-18, 4031–4039.
73. John Michopoulos, Panagiota Tsompanopoulou, Elias Houstis, and Anupam Joshi, Agent-based simulation of data-driven fire propagation dynamics, Computational Science-ICCS 2004: 4th International Conference, Kraków, Poland, June 6-9, 2004, Proceedings, Part III 4, Springer, 2004, pp. 732–739.
74. Mohammad Mirzadeh and Frédéric Gibou, A conservative discretization of the poisson–nernst–planck equations on adaptive cartesian grids, Journal of Computational Physics **274** (2014), 633–653.
75. Julien Moatti, A structure preserving hybrid finite volume scheme for semiconductor models with magnetic field on general meshes, ESAIM: Mathematical Modelling and Numerical Analysis **57** (2023), no. 4, 2557–2593.
76. Jurgis Pods, Johannes Schönke, and Peter Bastian, Electrodiffusion models of neurons and extracellular space using the poisson-nernst-planck equations—numerical simulation of the intra-and extracellular potential for an axon model, Biophysical journal **105** (2013), no. 1, 242–254.
77. Leonid P Savtchenko, Mu Ming Poo, and Dmitri A Rusakov, Electrodiffusion phenomena in neuroscience: a neglected companion, Nature reviews Neuroscience **18** (2017), no. 10, 598–612.
78. Ruigang Shen, Shi Shu, Ying Yang, and Benzhuo Lu, A decoupling two-grid method for the time-dependent poisson-nernst-planck equations, Numerical Algorithms **83** (2020), no. 4, 1613–1651.

79. Jin Shi, A steady-state capturing method for hyperbolic systems with geometrical source terms, Mathematical Modelling and Numerical Analysis **35** (2001), no. 4, 631–645 (en).
80. PJ Sjöström and M Häusser, A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons, Neuron **51** (2006), no. 2, 227–238.
81. Zilong Song, Xiulei Cao, and Huaxiong Huang, Electroneutral models for dynamic poisson-nernst-planck systems, Physical Review E **97** (2018), no. 1, 012411.
82. Shuai Su and Huazhong Tang, A positivity-preserving and free energy dissipative hybrid scheme for the poisson-nernst-planck equations on polygonal and polyhedral meshes, Computers & Mathematics with Applications **108** (2022), 33–48.
83. Naoya Takahashi, Kazuo Kitamura, Naoki Matsuo, Mark Mayford, Masanobu Kano, Norio Matsuki, and Yuji Ikegaya, Locally synchronized synaptic inputs, Science **335** (2012), no. 6066, 353–356.
84. Henk Kaarle Versteeg and Weeratunge Malalasekera, An introduction to computational fluid dynamics: the finite volume method, Pearson education, 2007.
85. Stephen R Williams and Greg J Stuart, Role of dendritic synapse location in the control of action potential output, Trends in neurosciences **26** (2003), no. 3, 147–154.
86. R Yuste, Dendritic spines, MIT press, 2023.
87. Rafael Yuste, Dendritic spines and distributed circuits, Neuron **71** (2011), no. 5, 772–781.
88. Qiong Zheng, Duan Chen, and Guo-Wei Wei, Second-order poisson–nernst–planck solver for ion transport, Journal of computational physics **230** (2011), no. 13, 5239–5262.

APPENDIX A

Appendix for Chapter 2

In this Appendix, we provide additional supports for the Chapter 2. We plot in Figure 1 the corresponding convergence orders of the solutions. On the left, the convergence orders for c_P are depicted in red, while on the right, those for V are shown in blue. The quantities Err c_P and Err V represent the $L^\infty(L^2)$ errors defined in the convergence Table 5. We present the graphs on a logarithmic scale, plotting curves with a slope of 2 in dashed lines. This visualization allows us to observe the outcomes of our simulations at the first iteration, where the convergence orders align with the expected value of 2, as established in the work of [18].

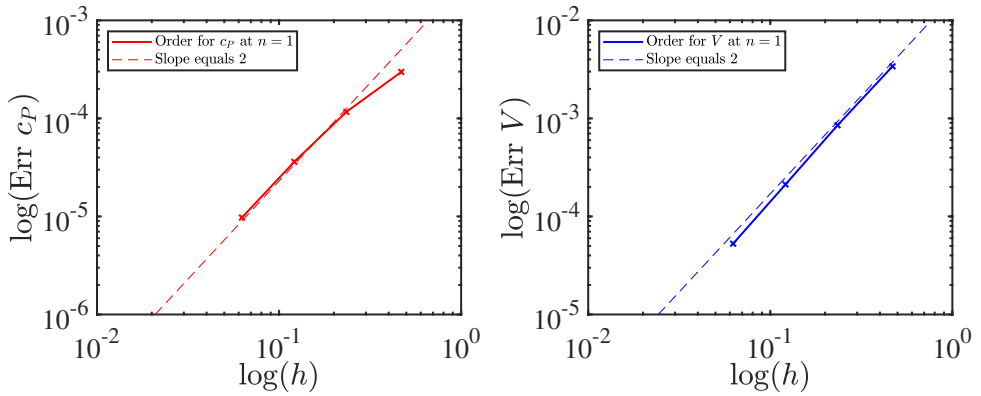

FIGURE 1. Convergence analysis of c_P and V errors at initial iteration $n = 1$. Left: $L^\infty(L^2)$ error of c_P (Err $c_P = \|c_P^{\mathcal{T},dt} - c_{P,\text{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},\infty}$) in red with logarithmic scale. Red dashed lines represent slope equals 2. Right: $L^\infty(L^2)$ error of V (Err $V = \|V^{\mathcal{T},dt} - V_{\text{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},\infty}$) in blue with logarithmic scale. Blue dashed lines represent slope equals 2.

The next figures illustrate the convergence orders at $n = 5$. We proceed to plot the logarithmic scale curves of the quantities Err c_P and Err V in red and blue, respectively. In contrast to the graphs in Figure 1, these curves exhibit a markedly distinct behavior. Specifically, the red curve representing c_P no longer aligns with the dashed line of slope 2 beyond the second data point, corresponding to the logarithmic error ($\log \text{Err } c_P$) of the second mesh refinement level in Table 6. Similarly, for the potential V, the logarithmic error curve changes slope at the third data point, corresponding to the third mesh refinement level.

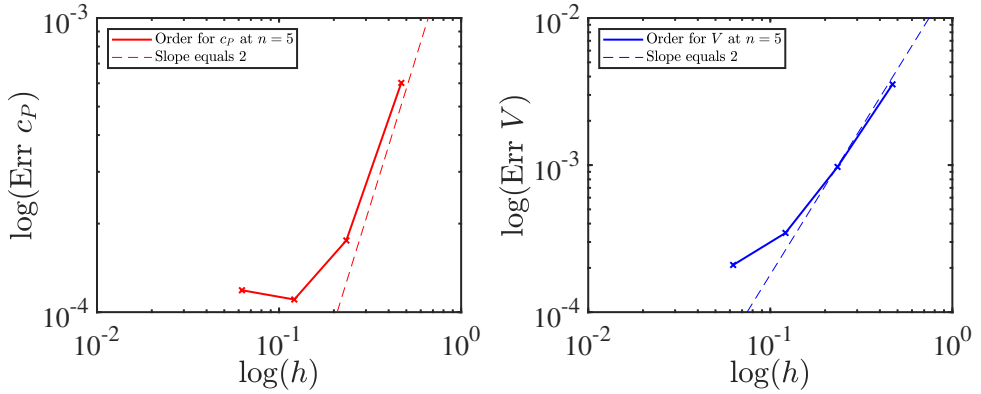

FIGURE 2. Convergence analysis of c_P and V errors at initial iteration $n = 5$. Left: $L^\infty(L^2)$ error of c_P (Err $c_P = \|c_P^{\mathcal{T},dt} - c_{P,\text{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},\infty}$) in red with logarithmic scale. Red dashed lines represent slope equals 2. Right: $L^\infty(L^2)$ error of V (Err $V = \|V^{\mathcal{T},dt} - V_{\text{ex}}^{\mathcal{T},dt}\|_{\mathcal{T},\infty}$) in blue with logarithmic scale. Blue dashed lines represent slope equals 2.

Next, for the Scilab section, we add the component definitions of all nodes. We begin by the F functions of the centers $\{10, 11, 12, 13\}$:

$$
\begin{aligned}
F^{\mathcal{T},dt}(10) = \frac{|K|}{dt}\big(c_P^{\mathcal{T},n+1}(10) - c_P^{\mathcal{T},n}(10)\big) - f_{c_P,10}^{n+1} \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \log(c_P^{\mathcal{T},n+1}(11)) - V_\tau^{n+1}(11)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \log(c_P^{\mathcal{T},n+1}(12)) - V_\tau^{n+1}(12)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(14) + c_P^{\mathcal{T},n+1}(2) + c_P^{\mathcal{T},n+1}(1)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \log(c_P^{\mathcal{T},n+1}(14)) - V_\tau^{n+1}(14)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(15) + c_P^{\mathcal{T},n+1}(4) + c_P^{\mathcal{T},n+1}(1)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \log(c_P^{\mathcal{T},n+1}(15)) - V_\tau^{n+1}(15)\Big).
\end{aligned}
$$

$$
\begin{aligned}
F^{\mathcal{T},dt}(11) = \frac{|K|}{dt}\big(c_P^{\mathcal{T},n+1}(11) - c_P^{\mathcal{T},n}(11)\big) - f_{c_P,11}^{n+1} \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - \log(c_P^{\mathcal{T},n+1}(10)) - V_\tau^{n+1}(10)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - \log(c_P^{\mathcal{T},n+1}(13)) - V_\tau^{n+1}(13)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(17) + c_P^{\mathcal{T},n+1}(6) + c_P^{\mathcal{T},n+1}(3)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - \log(c_P^{\mathcal{T},n+1}(17)) - V_\tau^{n+1}(17)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(16) + c_P^{\mathcal{T},n+1}(2) + c_P^{\mathcal{T},n+1}(3)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - \log(c_P^{\mathcal{T},n+1}(16)) - V_\tau^{n+1}(16)\Big).
\end{aligned}
$$

$$
\begin{aligned}
F^{\mathcal{T},dt}(12) = \frac{|K|}{dt}\big(c_P^{\mathcal{T},n+1}(12) - c_P^{\mathcal{T},n}(12)\big) - f_{c_P,12}^{n+1} \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - \log(c_P^{\mathcal{T},n+1}(10)) - V_\tau^{n+1}(10)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - \log(c_P^{\mathcal{T},n+1}(13)) - V_\tau^{n+1}(13)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(20) + c_P^{\mathcal{T},n+1}(7) + c_P^{\mathcal{T},n+1}(8)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - \log(c_P^{\mathcal{T},n+1}(20)) - V_\tau^{n+1}(20)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(18) + c_P^{\mathcal{T},n+1}(4) + c_P^{\mathcal{T},n+1}(7)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - \log(c_P^{\mathcal{T},n+1}(18)) - V_\tau^{n+1}(18)\Big).
\end{aligned}
$$

$$
\begin{aligned}
F^{\mathcal{T},dt}(13) = \frac{|K|}{dt}\big(c_P^{\mathcal{T},n+1}(13) - c_P^{\mathcal{T},n}(13)\big) - f_{c_P,13}^{n+1} \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - \log(c_P^{\mathcal{T},n+1}(11)) - V_\tau^{n+1}(11)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - \log(c_P^{\mathcal{T},n+1}(12)) - V_\tau^{n+1}(12)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(21) + c_P^{\mathcal{T},n+1}(9) + c_P^{\mathcal{T},n+1}(8)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - \log(c_P^{\mathcal{T},n+1}(21)) - V_\tau^{n+1}(21)\Big) \\
&+ \frac{1}{4}\Big(c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(19) + c_P^{\mathcal{T},n+1}(6) + c_P^{\mathcal{T},n+1}(9)\Big) \\
&\qquad \times \frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - \log(c_P^{\mathcal{T},n+1}(19)) - V_\tau^{n+1}(19)\Big).
\end{aligned}
$$

Then, we turn our attention to the Jacobian implementations. For center 10:

$$\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,10) = \frac{|K|}{dt} &+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11)\big) \\
&\qquad + \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2)\big)\frac{1}{c_P^{\mathcal{T},n+1}(10)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \log(c_P^{\mathcal{T},n+1}(12)) - V_\tau^{n+1}(12) \\
&\qquad + \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4)\big)\frac{1}{c_P^{\mathcal{T},n+1}(10)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(14)) + V_\tau^{n+1}(14)\big) \\
&\qquad + \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(14) + c_P^{\mathcal{T},n+1}(1) + c_P^{\mathcal{T},n+1}(2)\big)\frac{1}{c_P^{\mathcal{T},n+1}(10)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \log(c_P^{\mathcal{T},n+1}(15)) - V_\tau^{n+1}(15) \\
&\qquad + \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(15) + c_P^{\mathcal{T},n+1}(1) + c_P^{\mathcal{T},n+1}(4)\big)\frac{1}{c_P^{\mathcal{T},n+1}(10)}\Big).
\end{aligned}$$

$$\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,1) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(&\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(14)) + V_\tau^{n+1}(14)\big) \\
&+ \log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(15)) + V_\tau^{n+1}(15)\big)\Big).
\end{aligned}$$

$$\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,5) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(&\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11)\big) \\
&+ \log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12)\big)\Big).
\end{aligned}$$

$$\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,2) = &\frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11)\big)\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(14)) + V_\tau^{n+1}(14)\big)\Big).
\end{aligned}$$

$$\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,4) = &\frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12)\big)\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(15)) + V_\tau^{n+1}(15)\big)\Big).
\end{aligned}$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,11) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11)\big) \\ - \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2)\big)\frac{1}{c_P^{\mathcal{T},n+1}(11)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,12) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12)\big) \\ - \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4)\big)\frac{1}{c_P^{\mathcal{T},n+1}(12)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,14) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(14)) + V_\tau^{n+1}(14)\big) \\ - \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(14) + c_P^{\mathcal{T},n+1}(1) + c_P^{\mathcal{T},n+1}(2)\big)\frac{1}{c_P^{\mathcal{T},n+1}(14)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(10,15) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10) - \big(\log(c_P^{\mathcal{T},n+1}(15)) + V_\tau^{n+1}(15)\big) \\ - \big(c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(15) + c_P^{\mathcal{T},n+1}(4) + c_P^{\mathcal{T},n+1}(1)\big)\frac{1}{c_P^{\mathcal{T},n+1}(15)}\Big).$$

For center 11:

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,11) = \frac{|K|}{dt} + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13)) + (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6))\frac{1}{c_P^{\mathcal{T},n+1}(11)}\Big) + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - \log(c_P^{\mathcal{T},n+1}(10)) - V_\tau^{n+1}(10) + (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2))\frac{1}{c_P^{\mathcal{T},n+1}(11)}\Big) + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(17)) + V_\tau^{n+1}(17)) + (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(17) + c_P^{\mathcal{T},n+1}(6) + c_P^{\mathcal{T},n+1}(3))\frac{1}{c_P^{\mathcal{T},n+1}(11)}\Big) + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - \log(c_P^{\mathcal{T},n+1}(16)) - V_\tau^{n+1}(16) + (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(16) + c_P^{\mathcal{T},n+1}(3) + c_P^{\mathcal{T},n+1}(2))\frac{1}{c_P^{\mathcal{T},n+1}(11)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,3) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(17)) + V_\tau^{n+1}(17)) + \log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(16)) + V_\tau^{n+1}(16))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,5) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10)) + \log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,2) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10))\Big) + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(16)) + V_\tau^{n+1}(16))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,6) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13))\Big) + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(17)) + V_\tau^{n+1}(17))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,13) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13)) \\ - (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6))\frac{1}{c_P^{\mathcal{T},n+1}(13)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,10) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10)) \\ - (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(2))\frac{1}{c_P^{\mathcal{T},n+1}(10)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,17) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(17)) + V_\tau^{n+1}(17)) \\ - (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(17) + c_P^{\mathcal{T},n+1}(3) + c_P^{\mathcal{T},n+1}(6))\frac{1}{c_P^{\mathcal{T},n+1}(17)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(11,16) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11) - (\log(c_P^{\mathcal{T},n+1}(16)) + V_\tau^{n+1}(16)) \\ - (c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(16) + c_P^{\mathcal{T},n+1}(2) + c_P^{\mathcal{T},n+1}(3))\frac{1}{c_P^{\mathcal{T},n+1}(16)}\Big).$$

For center 12:

$$
\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,12) = \frac{|K|}{dt} &+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13)) \\
&\qquad + (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8))\frac{1}{c_P^{\mathcal{T},n+1}(12)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - \log(c_P^{\mathcal{T},n+1}(10)) - V_\tau^{n+1}(10) \\
&\qquad + (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4))\frac{1}{c_P^{\mathcal{T},n+1}(12)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(18)) + V_\tau^{n+1}(18)) \\
&\qquad + (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(18) + c_P^{\mathcal{T},n+1}(7) + c_P^{\mathcal{T},n+1}(4))\frac{1}{c_P^{\mathcal{T},n+1}(12)}\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - \log(c_P^{\mathcal{T},n+1}(20)) - V_\tau^{n+1}(20) \\
&\qquad + (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(20) + c_P^{\mathcal{T},n+1}(8) + c_P^{\mathcal{T},n+1}(7))\frac{1}{c_P^{\mathcal{T},n+1}(12)}\Big).
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,7) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(&\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(18)) + V_\tau^{n+1}(18)) \\
&+ \log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(20)) + V_\tau^{n+1}(20))\Big).
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,5) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(&\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10)) \\
&+ \log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13))\Big).
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,4) = &\frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10))\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(18)) + V_\tau^{n+1}(18))\Big).
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,8) = &\frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13))\Big) \\
&+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(20)) + V_\tau^{n+1}(20))\Big).
\end{aligned}
$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,13) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13)) \\ - (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8))\frac{1}{c_P^{\mathcal{T},n+1}(13)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,10) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(10)) + V_\tau^{n+1}(10)) \\ - (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(10) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(4))\frac{1}{c_P^{\mathcal{T},n+1}(10)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,18) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(18)) + V_\tau^{n+1}(18)) \\ - (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(18) + c_P^{\mathcal{T},n+1}(7) + c_P^{\mathcal{T},n+1}(4))\frac{1}{c_P^{\mathcal{T},n+1}(18)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(12,20) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12) - (\log(c_P^{\mathcal{T},n+1}(20)) + V_\tau^{n+1}(20)) \\ - (c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(20) + c_P^{\mathcal{T},n+1}(8) + c_P^{\mathcal{T},n+1}(7))\frac{1}{c_P^{\mathcal{T},n+1}(20)}\Big).$$

For center 13:

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,13) = \frac{|K|}{dt} + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11)) + (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6))\frac{1}{c_P^{\mathcal{T},n+1}(13)}\Big)$$
$$+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - \log(c_P^{\mathcal{T},n+1}(12)) - V_\tau^{n+1}(12) + (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8))\frac{1}{c_P^{\mathcal{T},n+1}(13)}\Big)$$
$$+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(19)) + V_\tau^{n+1}(19)) + (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(19) + c_P^{\mathcal{T},n+1}(6) + c_P^{\mathcal{T},n+1}(9))\frac{1}{c_P^{\mathcal{T},n+1}(13)}\Big)$$
$$+ \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - \log(c_P^{\mathcal{T},n+1}(21)) - V_\tau^{n+1}(21) + (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(21) + c_P^{\mathcal{T},n+1}(8) + c_P^{\mathcal{T},n+1}(9))\frac{1}{c_P^{\mathcal{T},n+1}(13)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,9) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(19)) + V_\tau^{n+1}(19)) + \log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(21)) + V_\tau^{n+1}(21))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,5) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11)) + \log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,6) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11))\Big) + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(19)) + V_\tau^{n+1}(19))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,8) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12))\Big) + \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(21)) + V_\tau^{n+1}(21))\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,11) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(11)) + V_\tau^{n+1}(11)) \\ - (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(11) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(6))\frac{1}{c_P^{\mathcal{T},n+1}(11)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,12) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(12)) + V_\tau^{n+1}(12)) \\ - (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(12) + c_P^{\mathcal{T},n+1}(5) + c_P^{\mathcal{T},n+1}(8))\frac{1}{c_P^{\mathcal{T},n+1}(12)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,19) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(19)) + V_\tau^{n+1}(19)) \\ - (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(19) + c_P^{\mathcal{T},n+1}(6) + c_P^{\mathcal{T},n+1}(9))\frac{1}{c_P^{\mathcal{T},n+1}(19)}\Big).$$

$$\mathbf{M}_{\mathrm{S}}^{\mathcal{T},dt}(13,21) = \frac{1}{4}\frac{|\sigma|^2}{|\mathcal{D}_{ext}|}\Big(\log(c_P^{\mathcal{T},n+1}(13)) + V_\tau^{n+1}(13) - (\log(c_P^{\mathcal{T},n+1}(21)) + V_\tau^{n+1}(21)) \\ - (c_P^{\mathcal{T},n+1}(13) + c_P^{\mathcal{T},n+1}(21) + c_P^{\mathcal{T},n+1}(8) + c_P^{\mathcal{T},n+1}(9))\frac{1}{c_P^{\mathcal{T},n+1}(21)}\Big).$$

APPENDIX B

Appendix for Chapter 5

Construction of the diffusion scheme (5.29)

We follow [**16**, **11**, **12**] and propose here a composite scheme to solve the following Poisson problem:

$$\Delta v = f, \tag{B.1}$$

where v and f are scalar valued functions and with homogeneous Dirichlet boundary conditions. Equation (B.1) is integrated over a cell Ω_j:

$$\int_{\partial\Omega_j} \nabla v \cdot \mathbf{n} = \int_{\Omega_j} f. \tag{B.2}$$

Using Proposition 5.4, we can approximate:

$$\int_{\partial\Omega_j} \nabla v \cdot \mathbf{n} \approx (1-\theta) \sum_{r\in\Omega_j} (\nabla v)(\mathbf{x}_r) \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2}\in\Omega_j} (\nabla v)(\mathbf{x}_{r+\frac{1}{2}}) \cdot \mathbf{C}_j^{r+\frac{1}{2}}. \tag{B.3}$$

The gradient at the nodes is computed as follows. Using a Taylor expansion, one can write:

$$v_j = v_r + (\nabla v)(\mathbf{x}_r) \cdot (\mathbf{x}_j - \mathbf{x}_r) + O(h^2). \tag{B.4}$$

Multiplying (B.4) by $\mathbf{C}_j^r$ and summing over all the cells around r:

$$\sum_{i|r\in\Omega_i} v_i \mathbf{C}_i^r = v_r \sum_{i|r\in\Omega_i} \mathbf{C}_i^r + \left(\sum_{i|r\in\Omega_i} \mathbf{C}_i^r \otimes (\mathbf{x}_i - \mathbf{x}_r) \right) (\nabla v)(\mathbf{x}_r) + O(h^3).$$

The sum $v_r \sum_{i|r\in\Omega_i} \mathbf{C}_i^r$ always vanishes. Indeed, if the node r is inside the domain, we have $\sum_{i|r\in\Omega_i} \mathbf{C}_i^r = \mathbf{0}$. Otherwise, the node r is on the boundary and v_r is given by the Dirichlet boundary condition, which is 0. Therefore a good approximation of the gradient at node r, denoted by $\mathbf{w}_r$, reads as:

$$\left(\sum_{i|r\in\Omega_i} \mathbf{C}_i^r \otimes (\mathbf{x}_i - \mathbf{x}_r) \right) \mathbf{w}_r = \sum_{i|r\in\Omega_i} v_i \mathbf{C}_i^r, \tag{B.5}$$

and the gradient at the edge $r + 1/2$ is defined by:

$$\mathbf{w}_{r+1/2} = \frac{\mathbf{w}_r + \mathbf{w}_{r+1}}{2}. \tag{B.6}$$

Therefore $\mathbf{w}_{dof}$ is well defined and is consistent with $\nabla v(\mathbf{x}_{dof})$. In addition, using (B.6), the diffusion flux (B.3) reads as:

$$(1-\theta) \sum_{r\in\Omega_j} \mathbf{w}_r \cdot \mathbf{C}_j^r + \theta \sum_{r+\frac{1}{2}\in\Omega_j} \mathbf{w}_{r+\frac{1}{2}} \cdot \mathbf{C}_j^{r+\frac{1}{2}} = \sum_{r\in\Omega_j} \mathbf{w}_r \cdot \mathbf{C}_j^r.$$

Eventually, the matrix in (B.5) is proved to be invertible if the mesh is not too deformed (see [**17**] and [**10**]). In practice, it is invertible as soon as the number of cells that contain the node r is different from 1. For the nodes located at the corner of the mesh (only one support cell, denoted as j), we approximate: $\mathbf{w}_r = v_j \mathbf{C}_j^r / \|\mathbf{C}_j^r\|^2$.