SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of

Matematica e Applicazioni

Joint PhD program MATEMATICA

Università degli studi di Milano-Bicocca – Università di Pavia – INdAM

Cycle XXXVI

Spline Upwind: a novel stabilization method for space–time Isogeometric Analysis

TESINI PAOLO

Registration number: 874709

Supervisor: Prof. Giancarlo Sangalli

Coordinator: Prof. Pierluigi Colli

ACADEMIC YEAR 2022-2023

alla mia amata famiglia

"Una famiglia felice non è che un anticipo del Paradiso."
attribuita a George Bernard Shaw

Acknowledgements

I would like to acknowledge my thesis supervisor Prof. Giancarlo Sangalli and Dr. Andrea Bressan, Dr. Gabriele Loli, Dr. Monica Montardini, Prof. Simone Scacchi: our fruitful collaboration has been instrumental in my growth from a scientific and research standpoint.

I express my appreciation to Prof. Paola Antonietti and Prof. Luca Dedè for introducing me to the fascinating topics of cardiac electrophysiology.

I would like to convey my deepest gratitude to Prof. Alfredo Marzocchi for his constant support and precious advice. My sincere recognition also goes to Prof. Maurizio Paolini and Prof. Francesco Ballarin for their significant counsels.

Contents

List of Figures

List of Tables

Introduction

The idea of using finite elements in space–time domains comes from [22, 2]. The space–time numerical paradigm is an approach to solve problems that involve spatial and temporal variations of physical quantities. In this framework, space and time are treated in an integrated manner, with the spatial domain and time discretized simultaneously, rather than separately, as in traditional numerical methods. This approach has been developed for various problems such as heat transfer [9], advection–diffusion [40], and elastodynamics [26], while the mathematical theory of space–time Galerkin methods has been established in recent works [51, 52].

This approach offers several advantages: it allows for a more natural handling of phenomena that vary in both time and space, enabling the accurate capture of transient phenomena and the analysis of their evolution over time and space. Thanks to the computing power of modern computers and the development of new mathematical techniques, space–time discretizations have recently gained considerable interest.

In summary, the space–time paradigm represents an innovative and powerful approach for simulating dynamic problems, providing more accurate and detailed solutions, especially where spatial and temporal variations are significant.

On the other hand, Isogeometric Analysis (IgA) [25, 18] is an evolution of classical finite elements. IgA employs spline functions, or their generalizations (Non-Uniform Rational B-Splines – NURBS), both to represent the computational domain and to approximate the solution of the partial differential equation modeling the problem of interest. Consequently, IgA facilitates the interoperability between Computer-Aided Design (CAD) and numerical simulations. Moreover, IgA benefits from the properties of smooth splines, such as providing higher accuracy compared to C^0 piecewise polynomials (see, e.g., [21, 7]).

In IgA framework, it is interesting to investigate space–time formulations, due to the additional opportunity to exploit the properties of smooth splines in time as well, as proposed in [53], while in [30] a stable isogeometric method for parabolic evolution equations is developed. In [38, 33] the authors propose preconditioners and solvers regarding the heat equation, while [48] investigates a continuous space–time IgA formulation to linear and nonlinear elastodynamics.

Space–time formulations, due to the augmented dimensionality, detrimentally impact on conventional solvers. However, it is noteworthy that space–time formulations hold promise for local mesh refinement [32] and parallelisation [23], attracting interest in the field (see, e.g., [31]).

However, the use of smooth splines with respect to time poses important challenges as well, in particular with regard to the causality principle. In fact, while the sequentiality of Discontinuous Galerkin method in time guarantees causality, this is not the case for Galerkin discretization with smooth spline approximation in time. This fact leads to a lack of causality, resulting in unphysical behaviors such as numerical instabilities and spurious oscillations, which may propagate backward in time.

The results presented in Chapter 1 are based on our recent paper [35], which emerged from our joint work with Prof. Giancarlo Sangalli and Dr. Gabriele Loli (Dipartimento di Matematica "F. Casorati", Università di Pavia). The aim of Chapter 1 is to design the Spline Upwind (SU) formulation for the heat equation, based on stabilization terms that promote causality. The proposed SU formulation generalizes classical upwinding techniques, such as Streamline upwind Petrov–Galerkin (SUPG), proposed in [8], to higher-degree splines. In particular, the SUPG method in time, applied to the heat equation with piecewise linear finite elements, results in a lower block-triangular linear system.

If we add artificial diffusion where the residual is high, as in Shock Capturing [54], stability is further promoted. Indeed, high-order derivatives enable targeted damping of spurious oscillations by selectively applying diffusion near layers while preserving the smooth regions of the solution. In this way, the high-order terms act as high-frequency filters, effectively attenuating higher frequency components, which correspond to spurious oscillations, more heavily than the lower frequency components.

This selective behavior ensures that unwanted oscillations are suppressed near layers without introducing excessive diffusion into smooth regions, thereby maintaining both the stability and accuracy of numerical simulations.

In order to apply these considerations, we add diffusion terms of different high orders to the plain Galerkin formulation, such that the resulting linear system is lower block-triangular. Furthermore, these terms are weighted by the residual in order to preserve the optimal convergence order where the solution is smooth.

We present various numerical tests to evaluate the stability and accuracy of the proposed SU formulation. In particular, in Chapter 1, we conduct experiments with a concentrated source term, and the results highlight that the numerical solutions are free from spurious oscillations.

This stabilization method for space–time IgA has several possible fields of application. Indeed, problems that generate solutions with layers are of interest in various engineering applications and real-world problems, ranging from laser-based additive manufacturing (see, e.g., [28]) to electrophysiology applied to clinical settings and human body simulations.

From this perspective, among the various possible fields of application, cardiac electrophysiology certainly represents an interesting research area.

The heart is a muscular organ that contracts when a concentrated electrical signal from the sinoatrial node, the heart's natural pacemaker, is transmitted through the His–Purkinje system to the cardiac muscle. This signal travels along the membranes of cardiac muscle cells (cardiomyocytes) and passes between them via gap junctions. During an action potential, these cells rapidly depolarize as positively charged ions enter, causing the contraction of sarcomeres, the contractile units of the muscle. After contraction, the ions are pumped out, and the cells repolarize to their resting state, allowing the muscle to relax and be ready for the next signal after a refractory period. For more details on cardiac activation physiology, [24] is an important reference.

Mathematical modeling plays an important role in electrophysiology for diagnostic and predictive purposes, and it holds promise for shaping future clinical decision-making (see, e.g., [15, 44, 46, 45, 56]).

The bidomain models, a mathematical framework that conceptually divides tissue into intracellular and extracellular compartments, are relevant in cardiac

electrophysiology. These models, which abstract the cellular microstructure, result in a system of two reaction–diffusion equations governing the intracellular and extracellular potentials. A further simplification is made by assuming equal anisotropic conductivities in both compartments, leading to the formulation of the monodomain equations. These equations describe the transmembrane potential as the difference between the intracellular and extracellular potentials, reducing the system to a single reaction–diffusion equation. The works [15, 14] provide a comprehensive derivation of these equations and their mathematical approximations.

The bi-/monodomain equations form the foundation of cardiac electrophysiology models but must be coupled with a detailed representation of ionic currents through voltage-sensitive protein structures (the ion channels). Regarding this topic, various sophisticated ionic models for atrial electrophysiology exist, such as the Nygren–Fiset–Firek–Clark–Lindblad–Clark–Giles model [41] and the Courtemanche–Ramirez–Nattel model [19].

In summary, numerous ionic models of varying complexity have been developed to describe cellular excitation, resulting in a system of ordinary differential equations coupled with the bi-/monodomain equations. For comprehensive insights into the derivation and analysis of these membrane models, interested readers can refer to [27].

Moreover, cardiac tissue has a highly anisotropic structure, with cardiomyocytes organized into laminar sheets and aligned mostly in the same direction, embedded in an extracellular matrix of collagen produced by fibroblasts. The atrial walls, though thinner than ventricular walls, exhibit a similar yet more complex anisotropic structure [57]. Often, for this reason, the atria are modeled as two-dimensional surfaces to simplify the bi-/monodomain equations. For reviews on the computational modeling of the atria, see [17].

The numerical discretization of the bi-/monodomain equations and their associated membrane models still present numerous numerical challenges. The traveling pulse solutions are characterized by sharp wave-fronts, particularly when employing more realistic membrane models. Therefore, a low resolution accuracy of these traveling fronts can lead to erroneous propagation velocity and dynamics, resulting in incorrect predictions regarding the cardiac activation.

Among the most recent developments in investigating cardiac function and

pathology, a promising computational framework is developed in [10]: it offers a three-dimensional representation of both the cardiac muscle and the hemodynamics, integrated within a fluid–structure interaction (FSI) model. The authors, to ensure efficiency and flexibility in the numerical solution, use time discretization with a segregated electrophysiology–force generation–FSI method and spatial discretization with finite elements. Simulations concentrating on the left side of the human heart reveal that the computational framework accurately captures the heart's physics and reproduces essential aspects of cardiac function observed in clinical scenarios, including the durations of all phases of the heartbeat. By integrating the heart model with a simplified circulation model, the authors simulate the interactions between the heart and the circulatory system, ensuring the preservation of total blood volume over time and replicating the effects associated with traveling pressure waves.

Furthermore, several strategies have been proposed in the literature to enhance the front approximation without resorting to excessive global refinement. These include changing the quadrature rule for ionic currents [29], implementing mesh adaptivity near the front [6, 13, 50], and employing high-order Spectral Element discretizations [11]. These approaches aim to strike a balance between computational efficiency and solution accuracy.

In this context, the employment of a sufficiently smooth function space for spatial approximation becomes crucial. This fact helps to mitigate the effects of the numerical grid on solution accuracy. The numerical methods of the bi-/monodomain equations with low-order spatial approximations need highly refined uniform meshes to accurately capture the front propagation. However, in electrophysiology simulations of the full human heart, the use of such detailed meshes leads to systems that are prohibitively expensive from a computational cost perspective. Regarding this issue, the works [42, 43] show that IgA emerges as a natural framework for this purpose, as suggested also in previous works on partial differential equations on surfaces [3, 20].

In particular, in [42] the authors explore the use of IgA to spatially approximate models of cardiac electrophysiology, defined on NURBS surfaces. They initially assume that the atria can be modeled as thin-walled structures, represented as two-dimensional surfaces with anisotropy present only in the tangent plane. While this assumption might appear to overlook the trans-mural anisotropy inherent in

the fiber structure of atrial walls, the studies [12, 16] demonstrate that proper homogenization treatment allows for accurate representation of mid-wall activation patterns. In this work, monodomain problems with the Mitchell–Schaeffer [37] and the Aliev–Panfilov [1] models are considered. The authors focus on the comparisons between basis functions of degree $p \geq 1$, which are globally C^k-continuous, with $k = 0$ or $p - 1$, aiming to identify the most accurate approximation of propagating fronts in relation with the number of degrees of freedom. The results indicate that spline basis functions of degree $p \geq 1$ with C^{p-1} continuity, even with moderately refined meshes, effectively allow for a more accurate description of the velocity of transmembrane potential fronts. The authors also apply IgA to the monodomain Mitchell–Schaeffer model of a realistic human left atrial geometry, with physiological characteristics as anisotropic conductivity tensor. Through this application, the numerical scheme maintains its advantageous approximation properties even in realistic contexts.

Similar results are confirmed in [43]: the Rogers–McCulloch ionic model [47], always in IgA framework, is applied to solve the bidomain equations on surfaces. Benchmark numerical simulations reveal that high-degree basis functions with high-order continuity across mesh elements accurately approximate the potential front velocity, with monotonic convergence in relation to mesh element size and the number of degrees of freedom. In this study, the use of spline basis functions improves the accuracy if compared to lower-degree, low-continuity functions with similar computational costs. Realistic simulations on surface atrial geometries are performed, using quadratic NURBS basis functions and two atria are described as separate NURBS patches with interatrial connection hypotheses. By coupling bidomain equations with the Courtemanche–Ramirez–Nattel ionic membrane model [19], the authors achieve realistic action potential approximations: measurements indicate that the electric propagation velocity aligns with physiological values and that the shape of the action potential is preserved on the surface.

Therefore, Chapter 2 of this thesis presents the main results of our collaborative effort with Prof. Paola Antonietti, Prof. Luca Dedè (MOX, Mathematics Department, Politecnico di Milano), Prof. Giancarlo Sangalli, Dr. Gabriele Loli and Dr. Monica Montardini (Dipartimento di Matematica "F. Casorati", Università di Pavia). The aim of Chapter 2 is to investigate the behavior of the SU stabilization

method, always in space–time IgA framework, within the cardiac electrophysiology context. For this reason we modify the SU method to improve effectiveness and computational efficiency, and we apply it to the monodomain equation coupled with the Rogers–McCulloch ionic model [47]. This model is widely used to simulate electrophysiological wave propagation in the heart and is based on differential equations, allowing for the simulation of front propagation phenomena.

To assess the behavior of the SU method in cardiac electrophysiology, we perform several numerical tests and compare the computational costs of both the plain Galerkin method and the stabilized one, analyzing the results also from the perspective of computational cost.

Furthermore, as detailed in Section 2.1.2, preconditioners play a crucial role especially in IgA space–time framework on 3D spatial domains, where the complexity and size of the resulting linear systems can significantly impact on computational performances. The application of preconditioners accelerates the iterative solver convergence of resulting linear system. For this reason, we propose a suitable preconditioner, based on the arrowhead preconditioner [33] and the preconditionier of the mass matrix as in [34].

The outline of the thesis is as follows.

- In Preliminaries we discuss the basics of space–time IgA.
- In Chapter 1, we discuss the proposed SU method, starting with a review of some classical stabilized formulations in one dimension for advection and advection–diffusion equations, and introducing our novel SU formulation (Section 1.1). In Section 1.2, we apply the SU method to the heat equation. We propose numerical tests, assessing the performance of the presented stabilization method, in Section 1.3.
- In Chapter 2, the application of the SU method to cardiac electrophysiology is presented. In Section 2.1, we optimize our stabilization method to better suit this context and we study the behavior of our novel method using the monodomain equation coupled with the Rogers–McCulloch ionic model. Moreover, in Section 2.2, we present various numerical tests in up to 3D spatial domains, including one approximating the geometry of the left ventricle, to

evaluate the performance of the stabilization method, also from a computational cost standpoint.

- The last part (Conclusion) sums up the work and highlights some future research directions.

Preliminaries

In this section, we present the notation adopted in the following and we discuss the basics of splines and space–time IgA. For more details on splines in IgA context, we refer to [18].

Given n and p two positive integers, we consider the open knot vector

$$\widehat{\Xi} := \left\{0 = \widehat{\xi}_1 = \cdots = \widehat{\xi}_{p+1} \leq \cdots \leq \widehat{\xi}_n = \cdots = \widehat{\xi}_{n+p+1} = 1\right\},$$

and the vector $\widehat{Z} := \left\{\widehat{\zeta}_1, \ldots, \widehat{\zeta}_m\right\}$ of knots without repetitions (i.e., breakpoints).

The univariate spline space is defined as:

$$\widehat{\mathcal{S}}_h^p := \text{span}\{\widehat{b}_{i,p}\}_{i=1}^n,$$

where $\widehat{b}_{i,p}$ are the univariate B-splines and h plays the role of family parameter, while $\widehat{h}$ denotes the mesh-size in the parametric domain, i.e.,

$$\widehat{h} := \max\{|\widehat{\xi}_{i+1} - \widehat{\xi}_i| \text{ s.t. } i = 1, \ldots, n+p\}.$$

For each univariate B-spline $\widehat{b}_{i,p}$, we associate a knot average, called Greville abscissa, defined as:

$$\gamma_{i,p} := \frac{\widehat{\xi}_{i+1} + \cdots + \widehat{\xi}_{i+p}}{p} \quad \text{for} \quad i = 1, \ldots, n.$$

Multivariate B-splines are constructed as tensor products of univariate B-splines. In our work, we focus on functions that depend on d spatial variables and the time variable, making multivariate B-splines a natural choice for this setting.

Given positive integers n_l, p_l, for $l = 1, \ldots, d$, and n_t, p_t, we define $d+1$ univariate knot vectors $\widehat{\Xi}_l := \left\{\widehat{\xi}_{l,1} \leq \cdots \leq \widehat{\xi}_{l,n_l+p_l+1}\right\}$, for $l = 1, \ldots, d$, and $\widehat{\Xi}_t := \left\{\widehat{\xi}_{t,1} \leq \cdots \leq \widehat{\xi}_{t,n_t+p_t+1}\right\}$. The $d+1$ breakpoints vectors are $\widehat{Z}_l := \left\{\widehat{\zeta}_{l,1}, \ldots, \widehat{\zeta}_{l,m_l}\right\}$,

for $l = 1, \dots, d$, and $\widehat{Z}_t := \left\{ \widehat{\zeta}_{t,1}, \dots, \widehat{\zeta}_{t,m_t} \right\}$.

We denote by $\widehat{h}_l$ the mesh-size in the parametric domain, associated to the knot vector $\widehat{\Xi}_l$ for $l = 1, \dots, d$, while $\widehat{h}_t$ represents the mesh-size in the parametric domain of the time knot vector $\widehat{\Xi}_t$.

Additionally, let $\boldsymbol{p}$ be the vector containing the degree of each univariate spline space, i.e., $\boldsymbol{p} := (\boldsymbol{p}_s, p_t)$, where $\boldsymbol{p}_s := (p_1, \dots, p_d)$.

The multivariate B-splines are defined as:

$$\widehat{B}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\eta}, \tau) := \widehat{B}_{\boldsymbol{i_s},\boldsymbol{p}_s}(\boldsymbol{\eta})\, \widehat{b}_{i_t,p_t}(\tau),$$

where

$$\widehat{B}_{\boldsymbol{i_s},\boldsymbol{p}_s}(\boldsymbol{\eta}) := \widehat{b}_{i_1,p_1}(\eta_1) \dots \widehat{b}_{i_d,p_d}(\eta_d),$$

with $\boldsymbol{i_s} := (i_1, \dots, i_d)$, $\boldsymbol{i} := (\boldsymbol{i_s}, i_t)$ and $\boldsymbol{\eta} = (\eta_1, \dots, \eta_d)$.
The corresponding spline space is defined as:

$$\widehat{\mathcal{S}}_h^{\boldsymbol{p}} := \operatorname{span} \left\{ \widehat{B}_{\boldsymbol{i},\boldsymbol{p}} \;\middle|\; i_l = 1, \dots, n_l;\ l = 1, \dots, d;\ i_t = 1, \dots, n_t \right\}.$$

We have that $\widehat{\mathcal{S}}_h^{\boldsymbol{p}} = \widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p}_s} \otimes \widehat{\mathcal{S}}_{h_t}^{p_t}$, where

$$\widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p}_s} := \operatorname{span} \left\{ \widehat{B}_{\boldsymbol{i_s},\boldsymbol{p}_s} \;\middle|\; i_l = 1, \dots, n_l;\ l = 1, \dots, d \right\}$$

is the space of tensor-product splines on $\widehat{\Omega} := (0,1)^d$, and

$$\widehat{\mathcal{S}}_{h_t}^{p_t} := \operatorname{span} \left\{ \widehat{b}_{i_t,p_t} \;\middle|\; i_t = 1, \dots, n_t \right\}.$$

We assume that $p_1, \cdots, p_d, p_t \geq 1$ and $\widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p}_s} \subset C^0(\widehat{\Omega})$ while $\widehat{\mathcal{S}}_{h_t}^{p_t} \subset C^{p_t - 1}(0,1)$. We allow variable continuity in space, since it may be useful for geometry representation, while we consider only maximum continuity with respect to time in order to benefit from the approximation properties of smooth splines (see [21, 7]).

We denote by $\Omega \times (0,T)$ the space–time computational domain, where $\Omega \subset \mathbb{R}^d$ and Ω is parametrized by $\boldsymbol{F} : \widehat{\Omega} \to \Omega$, with $\boldsymbol{F} \in \left[\widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p}_s} \right]^d$, and $T > 0$ is the final time. The space–time domain is parametrized by $\boldsymbol{G} : \widehat{\Omega} \times (0,1) \to \Omega \times (0,T)$, such that $\boldsymbol{G}(\boldsymbol{\eta}, \tau) := (\boldsymbol{F}(\boldsymbol{\eta}), T\tau) = (\boldsymbol{x}, t)$.

The spline space with zero initial and Dirichlet conditions, in parametric coordinates, is

$$\widehat{\mathcal{X}}_{h,0} := \left\{ \widehat{v}_h \in \widehat{\mathcal{S}}_h^{\boldsymbol{p}} \;\middle|\; \widehat{v}_h = 0 \text{ on } \partial\widehat{\Omega} \times (0,1) \text{ and } \widehat{v}_h = 0 \text{ on } \widehat{\Omega} \times \{0\} \right\}.$$

We also have that $\widehat{\mathcal{X}}_{h,0} = \widehat{\mathcal{X}}_{s,h_s,0} \otimes \widehat{\mathcal{X}}_{t,h_t}$, where

$$\begin{aligned}
\widehat{\mathcal{X}}_{s,h_s,0} &:= \left\{ \widehat{w}_h \in \widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p}_s} \;\middle|\; \widehat{w}_h = 0 \text{ on } \partial\widehat{\Omega} \right\} \\
&= \text{ span} \left\{ \widehat{b}_{i_1,p_1} \dots \widehat{b}_{i_d,p_d} \;\middle|\; i_l = 2, \dots, n_l - 1;\ l = 1, \dots, d \right\}, \\
\widehat{\mathcal{X}}_{t,h_t} &:= \left\{ \widehat{w}_h \in \widehat{\mathcal{S}}_{h_t}^{p_t} \;\middle|\; \widehat{w}_h(0) = 0 \right\} \;=\; \text{span} \left\{ \widehat{b}_{i_t,p_t} \;\middle|\; i_t = 2, \dots, n_t \right\}.
\end{aligned}$$

With colexicographical re-orderings of the basis functions, we write

$$\begin{aligned}
\widehat{\mathcal{X}}_{s,h_s,0} &= \text{ span} \left\{ \widehat{b}_{i_1,p_1} \dots \widehat{b}_{i_d,p_d} \;\middle|\; i_l = 1, \dots, n_{s,l};\ l = 1, \dots, d \right\} \\
&= \text{span} \left\{ \widehat{B}_{i,\boldsymbol{p}_s} \;\middle|\; i = 1, \dots, N_{s,0} \right\}, \\
\widehat{\mathcal{X}}_{t,h_t} &= \text{ span} \left\{ \widehat{b}_{i,p_t} \;\middle|\; i = 1, \dots, N_t \right\},
\end{aligned}$$

and

$$\widehat{\mathcal{X}}_{h,0} = \text{span} \left\{ \widehat{B}_{i,\boldsymbol{p}} \;\middle|\; i = 1, \dots, N_{\text{dof},0} \right\}, \tag{1}$$

where $N_{\text{dof},0} := N_{s,0} N_t$, and $N_{s,0} := \prod_{l=1}^{d} n_{s,l}$ with $n_{s,l} := n_l - 2$, for $l = 1, \dots, d$, while $N_t := n_t - 1$.

Our isogeometric space is the isoparametric push-forward of (1) through the geometric map $\boldsymbol{G}$, i.e.,

$$\mathcal{X}_{h,0} := \text{span} \left\{ B_{i,\boldsymbol{p}} := \widehat{B}_{i,\boldsymbol{p}} \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, \dots, N_{\text{dof},0} \right\},$$

where again $\mathcal{X}_{h,0} = \mathcal{X}_{s,h_s,0} \otimes \mathcal{X}_{t,h_t}$, with

$$\mathcal{X}_{s,h_s,0} := \text{span} \left\{ B_{i,\boldsymbol{p}_s} := \widehat{B}_{i,\boldsymbol{p}_s} \circ \boldsymbol{F}^{-1} \;\middle|\; i = 1, \dots, N_{s,0} \right\}$$

and

$$\mathcal{X}_{t,h_t} := \text{span} \left\{ b_{i,p_t}(\cdot) := \widehat{b}_{i,p_t} \left(\frac{\cdot}{T} \right) \;\middle|\; i = 1, \dots, N_t \right\}.$$

On the other hand, the spline space with only initial condition, in parametric coordinates, is

$$\widehat{\mathcal{X}}_h := \left\{ \widehat{v}_h \in \widehat{\mathcal{S}}_h^{\boldsymbol{p}} \;\middle|\; \widehat{v}_h = 0 \text{ on } \widehat{\Omega} \times \{0\} \right\}.$$

We also have that $\widehat{\mathcal{X}}_h = \widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p}_s} \otimes \widehat{\mathcal{X}}_{t,h_t}$.

With a colexicographical re-orderings of the basis functions, we write

$$\widehat{\mathcal{X}}_h = \operatorname{span} \left\{ \widehat{B}_{i,\boldsymbol{p}} \;\middle|\; i = 1, \dots, N_{\text{dof}} \right\}, \tag{2}$$

where $N_{\text{dof}} := N_s N_t$, and $N_s := \prod_{l=1}^{d} n_l$, with $l = 1, \dots, d$.

Our isogeometric space is the isoparametric push-forward of (2) through the geometric map $\boldsymbol{G}$, i.e.,

$$\mathcal{X}_h := \operatorname{span} \left\{ B_{i,\boldsymbol{p}} := \widehat{B}_{i,\boldsymbol{p}} \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, \dots, N_{\text{dof}} \right\}, \tag{3}$$

where again $\mathcal{X}_h = \mathcal{X}_{s,h_s} \otimes \mathcal{X}_{t,h_t}$, with

$$\mathcal{X}_{s,h_s} := \operatorname{span} \left\{ B_{i,\boldsymbol{p}_s} := \widehat{B}_{i,\boldsymbol{p}_s} \circ \boldsymbol{F}^{-1} \;\middle|\; i = 1, \dots, N_s \right\}.$$

Moreover we define the breakpoints in the time interval as:

$$\zeta_{t,i} := T \widehat{\zeta}_{t,i} \quad \text{for} \quad i = 1, \cdots, m_t,$$

and the time steps as:

$$h_{t,i} := \zeta_{t,i+1} - \zeta_{t,i} \quad \text{for} \quad i = 1, \cdots, m_t - 1.$$

Finally, following [5], we define the support extension, for $\boldsymbol{i_s} := (i_1, \dots, i_d)$, with $i_l = 1, \dots, n_l$, $l = 1, \dots, d$ and $i_t = 1, \dots, n_t$, as:

$$\widetilde{I}_{\boldsymbol{i_s},i_t} := \left(\widehat{\xi}_{1,i_1-p_1} \, , \, \widehat{\xi}_{1,i_1+p_1+1} \right) \times \cdots \times \left(\widehat{\xi}_{d,i_d-p_d} \, , \, \widehat{\xi}_{d,i_d+p_d+1} \right) \times \left(\widehat{\xi}_{t,i_t-p_t} \, , \, \widehat{\xi}_{t,i_t+p_t+1} \right). \tag{4}$$

Chapter 1

Spline Upwind for space–time Isogeometric Analysis

The aim of the present chapter is to design the Spline Upwind (SU) formulation of the heat equation, incorporating stabilization terms that promote causality. The proposed SU generalizes classical upwinding, such as Streamline Upwind Petrov–Galerkin (SUPG) [8], to higher-degree splines. It is worth recalling that the SUPG method, applied to the heat equation in time and using piecewise linear finite elements, results in a lower block-triangular linear system. Stability is further enhanced by adding artificial diffusion when the residual is larger, as done in Shock Capturing techniques [54].

The use of high-order derivatives allows for targeted damping of spurious oscillations by selectively applying diffusion near layers, due to their ability to act as high-frequency filters, effectively attenuating the higher frequency components associated with spurious oscillations.

These techniques promote causality and stability, thereby enhancing overall computational robustness. The proposed SU formulation extends these concepts to higher-degree splines: we enrich the plain Galerkin formulation by adding diffusion terms of varying orders, ensuring that the resulting linear system remains block-triangular. Furthermore, these terms are weighted by the residual to preserve the optimal convergence order where the solution is smooth.

We present several numerical tests, ranging from one-dimensional problems to the heat equation on 2D spatial domain, to evaluate the expected behavior of the proposed

SU formulation. In particular, we conduct experiments with a concentrated source term, evidencing that the numerical solutions are free from spurious oscillations.

It is important to emphasize that the main focus of this chapter is not on computational costs. As such, we do not discuss or analyze the efficiency of the proposed method; these aspects are addressed in Chapter 2.

1.1 Upwinding in one dimension

Our first model problem is the one-dimensional advection: we look for a function u such that:

$$\begin{cases} u' &= f \quad \text{in } (0,T), \\ u(0) &= 0. \end{cases} \tag{1.1}$$

We assume $f \in L^2(0,T)$ and we consider the following Galerkin method:

find $u_h \in \mathcal{X}_{t,h_t}$ such that:

$$\mathcal{A}(u_h; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{t,h_t},$$

where

$$\mathcal{A}(u_h; v_h) := \int_0^T u_h' \, v_h \, \mathrm{dt} \quad \text{and} \quad \mathcal{F}(v) := \int_0^T f \, v_h \, \mathrm{dt}.$$

1.1.1 Standard Upwind and Shock Capturing

The Streamline Upwind Petrov–Galerkin (SUPG) method, reads:

find $u_h \in \mathcal{X}_{t,h_t}$ such that:

$$\mathcal{A}(u_h; v_h) + \mathcal{S}_{\mathrm{SUPG}}(u_h, f; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{t,h_t}, \tag{1.2}$$

where

$$\mathcal{S}_{\mathrm{SUPG}}(u_h, f; v_h) := \sum_{i=1}^{m_t - 1} \tau_{\mathrm{SUPG},i} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} (u_h' - f) v_h' \, \mathrm{dt},$$

In order to enhance the stability of SUPG, following [54], we can further add a Shock Capturing term leading to

find $u_h \in \mathcal{X}_{t,h_t}$ such that:

$$\mathcal{A}(u_h; v_h) + \mathcal{S}_{\text{SUPG}}(u_h, f; v_h) + \mathcal{S}_{\text{SC}}(u_h; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{t,h_t},$$

where, in accordance with [4],

$$\mathcal{S}_{\text{SC}}(u_h; v_h) := \sum_{i=1}^{m_t-1} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \kappa_{\text{SC},i} u_h' v_h' \, \text{dt} \quad \text{and} \quad \kappa_{\text{SC},i} := \tau_{\text{SC},i} \frac{|u_h' - f|}{u_{\text{ref}}},$$

with $\tau_{\text{SC},i} := \frac{h_{t,i}^2}{4}$, and u_{ref} is a reference magnitude for u_h.

1.1.2 High-order Upwind

With the choice $\tau_{\text{SUPG},i} = \frac{h_{t,i}}{2}$ and when $p_t = 1$, formulation (1.2) leads to a lower triangular linear system. However, for higher degree splines, the matrix does not exhibit a lower triangular structure regardless of the $\tau_{\text{SUPG},i}$ value chosen. This motivates the design of a new stabilizing formulation for splines with maximum continuity C^{p_t-1}, that we call Non Consistent Spline Upwind (NCSU), i.e.,

find $u_h \in \mathcal{X}_{t,h_t}$ such that:

$$\mathcal{A}(u_h; v_h) + \mathcal{S}_{\text{NCSU}}(u_h; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{t,h_t}, \tag{1.3}$$

where the new stabilizing term fulfills:

$$\mathcal{S}_{\text{NCSU}}(u_h; v_h) := \sum_{k=1}^{p_t} \sum_{i=1}^{m_t-1} {h_{t,i}}^{2k-1} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \tau_k(t) \, u_h^{(k)} v_h^{(k)} \, \text{dt},$$

where each $\tau_k(T\,\cdot) \in \widehat{\mathcal{S}}_{h_t}^{p_t-k} \subset C^{p_t-k-1}$ is a spline with maximum continuity that is selected in order to make the linear system matrix lower triangular, that is

$$\int_0^T b_{\ell+i,p_t}' b_{i,p_t} \, \text{dt} + \sum_{k=1}^{p_t} \sum_{j=1}^{m_t-1} {h_{t,j}}^{2k-1} \int_{\zeta_{t,j}}^{\zeta_{t,j+1}} \tau_k(t) \, b_{\ell+i,p_t}^{(k)} b_{i,p_t}^{(k)} \, \text{dt} = 0, \tag{1.4}$$

for $i = 1, \ldots, N_t - 1$ and $\ell = 1, \ldots, r,$ with $r = \min(p_t, N_t - i)$.

In Figure 1.1 we present $\tau_k(t)$ for different degrees ($p_t = 2, 3, 4$) on a uniform mesh ($h_t = 1/50$), while in Figure 1.2 we have plotted $\tau_k(t)$ for $p_t = 3$ on a non-uniform

mesh: in all our numerical tests we have observed that (1.4) is well posed, the τ_k are bounded and positive in most of the domain and where they are negative, they are much smaller in magnitude.

However, the formulation (1.3) lacks optimal convergence. To overcome this limitation we introduce a nonlinear weighting based on residual, denoted Spline Upwind (SU) method, and defined as:

find $u_h \in \mathcal{X}_{t,h_t}$ such that:

$$\mathcal{A}(u_h; v_h) + \mathcal{S}_{\mathrm{SU},1}(u_h, f; v_h) + \mathcal{S}_{\mathrm{SU},2}(u_h; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{t,h_t}, \tag{1.5}$$

with

$$\mathcal{S}_{\mathrm{SU},1}(u_h, f; v_h) := \sum_{i=1}^{m_t-1} h_{t,i} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \tau_1(t) \left(u_h' - (1-\theta(t))\, f\right) v_h' \, \mathrm{dt},$$

and

$$\mathcal{S}_{\mathrm{SU},2}(u_h; v_h) := \sum_{k=2}^{p_t} \sum_{i=1}^{m_t-1} h_{t,i}^{2k-1} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \tau_k(t)\, \theta(t)\, u_h^{(k)} v_h^{(k)} \, \mathrm{dt},$$

where $\theta(t)$ is a piecewise linear interpolation of Θ_i computed in the breakpoints $\zeta_{t,i}$, for $i = 1, \ldots, m_t$, as:

$$\Theta_i := \min(\mathrm{res}_i, 1),$$

with the relative residual res_i defined as:

$$\mathrm{res}_i := \frac{\|u_h' - f\|_{L^\infty([\zeta_{t,\max(1,i-1)}, \zeta_{t,\min(i+1,m_t)}])}}{T^{-1}\, \|u_h\|_{L^\infty([0,T])} + \|u_h'\|_{L^\infty([0,T])}}.$$

In this way, the function $\theta(t)$ ensures that, when the residual is high, such as within layers, $\theta \approx 1$ and the SU stabilization locally reduces to the NCSU formulation (1.3).

For advection–diffusion problem

$$\begin{cases} -\varepsilon u'' + u' &= f \quad \text{in } (0,T), \\ u(0) = u(T) &= 0, \end{cases} \tag{1.6}$$

where $\varepsilon > 0$, the SU stabilization is extended straightforwardly by redefining $\mathcal{S}_{\mathrm{SU},1}$

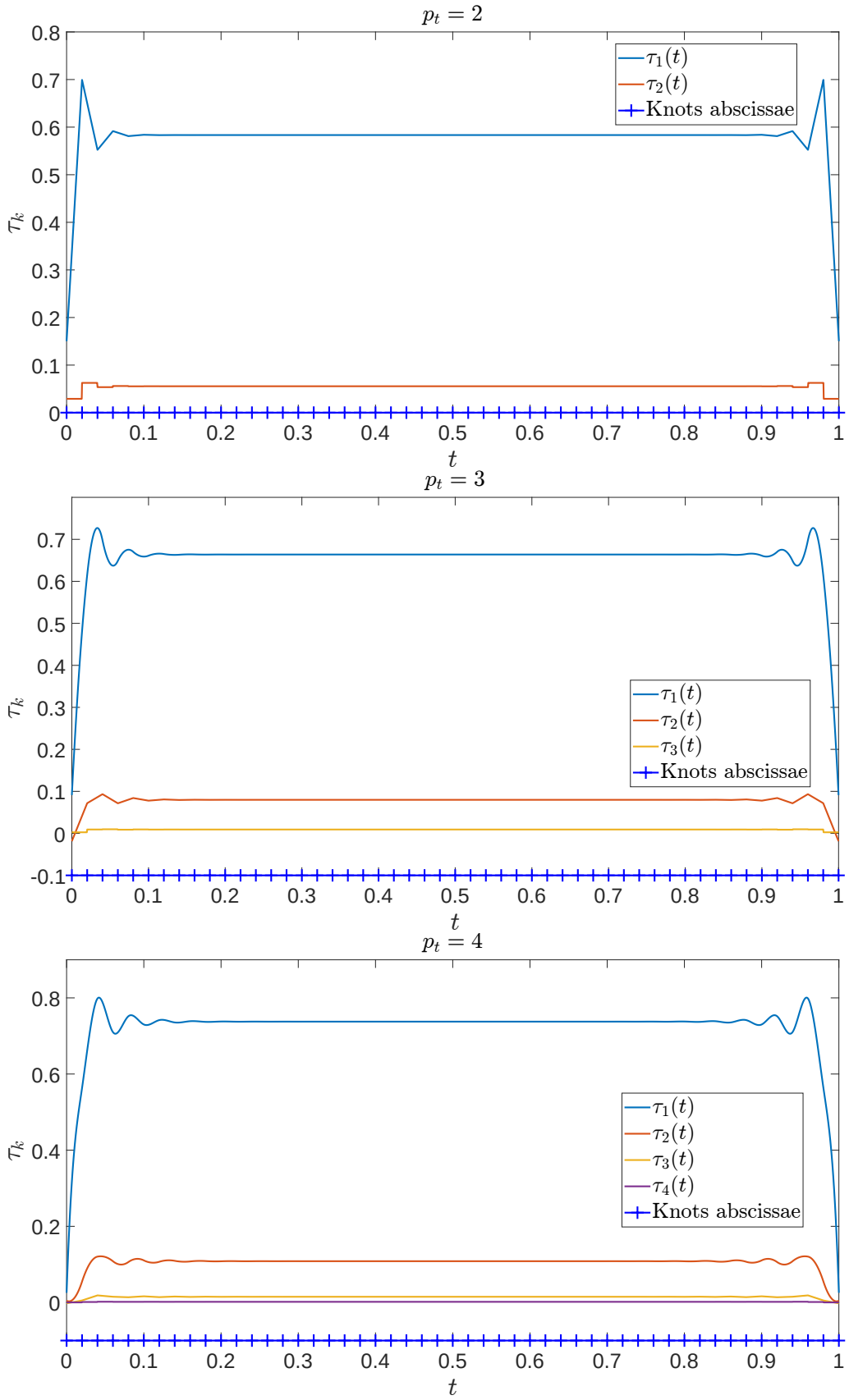

Figure 1.1: Plots of $\tau_k(t)$ of the SU method on a uniform mesh, depicted in blue on horizontal axes.

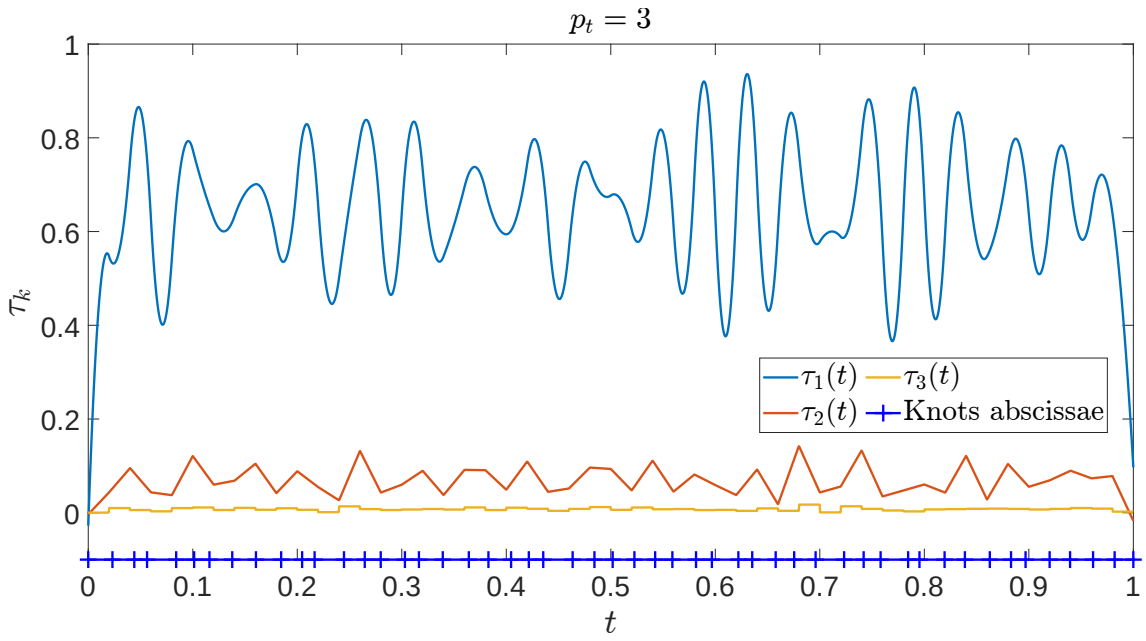

Figure 1.2: Plot of $\tau_k(t)$ of the SU method on a non-uniform mesh, depicted in blue on horizontal axis.

in (1.5) as follows:

$$\mathcal{S}_{\mathrm{SU},1}(u_h, f; v_h) := \sum_{i=1}^{m_t-1} h_{t,i} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \tau_1(t) \left(u_h' - (1-\theta(t))(-\varepsilon u_h'' - f)\right) v_h' \, \mathrm{dt},$$

and

$$\Theta_i := \min(\mathrm{res}_i, 1),$$

with

$$\mathrm{res}_i := \frac{\|-\varepsilon u_h'' + u_h' - f\|_{L^\infty([\zeta_{t,\max(1,i-1)}, \zeta_{t,\min(i+1,m_t)}])}}{T^{-1}\|u_h\|_{L^\infty([0,T])} + \|u_h'\|_{L^\infty([0,T])}}.$$

1.2 Upwinding the heat equation

The heat equation with homogeneous boundary and initial condition reads

$$\begin{cases} \partial_t u - \Delta u &= f \quad \text{in} \quad \Omega \times (0,T), \\ u &= 0 \quad \text{on} \quad \partial\Omega \times [0,T], \\ u &= 0 \quad \text{in} \quad \Omega \times \{0\}, \end{cases} \tag{1.7}$$

and we assume that $f \in L^2\big(\Omega \times (0,T)\big)$.

Introducing the bilinear form $\mathcal{A}(\cdot;\cdot)$ and the linear form $\mathcal{F}(\cdot)$ as:

$$\mathcal{A}(w;v) := \int_0^T \int_\Omega \big(\partial_t w\, v + \nabla w \cdot \nabla v\big)\, \mathrm{d}\Omega\, \mathrm{dt} \quad \text{and} \quad \mathcal{F}(v) := \int_0^T \int_\Omega f\, v\, \mathrm{d}\Omega\, \mathrm{dt},$$

we consider the Galerkin method

find $u_h \in \mathcal{X}_{h,0}$ such that:

$$\mathcal{A}(u_h; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{h,0}.$$

The matrix of the linear system has the following Kronecker product structure:

$$\mathbf{W}_t \otimes \mathbf{M}_s + \mathbf{M}_t \otimes \mathbf{K}_s, \tag{1.8}$$

where, for $i, j = 1, \ldots, N_t$,

$$[\mathbf{W}_t]_{i,j} = \int_0^T b'_{j,p_t}(t)\, b_{i,p_t}(t)\, \mathrm{dt} \quad \text{and} \quad [\mathbf{M}_t]_{i,j} = \int_0^T b_{j,p_t}(t)\, b_{i,p_t}(t)\, \mathrm{dt},$$

while, for $i, j = 1, \ldots, N_{s,0}$,

$$[\mathbf{K}_s]_{i,j} = \int_\Omega \nabla B_{j,\boldsymbol{p}_s}(\boldsymbol{x}) \cdot \nabla B_{i,\boldsymbol{p}_s}(\boldsymbol{x})\ \mathrm{d}\Omega \quad \text{and} \quad [\mathbf{M}_s]_{i,j} = \int_\Omega B_{j,\boldsymbol{p}_s}(\boldsymbol{x})\ B_{i,\boldsymbol{p}_s}(\boldsymbol{x})\ \mathrm{d}\Omega.$$

The SUPG method reads

find $u_h \in \mathcal{X}_{h,0}$ such that:

$$\mathcal{A}(u_h; v_h) + \mathcal{S}_{\mathrm{SUPG}}(u_h, f; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{h,0}, \tag{1.9}$$

where

$$\mathcal{S}_{\mathrm{SUPG}}(u_h, f; v_h) := \sum_{i=1}^{m_t - 1} \tau_{\mathrm{SUPG},i} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \int_\Omega \big(\partial_t u_h - \Delta u_h - f\big) \partial_t v_h\, \mathrm{d}\Omega\, \mathrm{dt}.$$

With the choice $\tau_{\mathrm{SUPG},i} = \frac{h_{t,i}}{2}$ and when $p_t = 1$, formulation (1.9) leads to a lower block-triangular time derivative matrix.

The novel space–time formulation for the heat equation is based, as in the SU one-dimensional formulation, on the idea of modifying (1.8) in order to obtain lower

triangular time matrices. This is accomplished by introducing numerical diffusion in the time direction. The proposed method reads

find $u_h \in \mathcal{X}_{h,0}$ such that:

$$\mathcal{A}(u_h; v_h) + \mathcal{S}_{\mathrm{SU},1}(u_h, f; v_h) + \mathcal{S}_{\mathrm{SU},2}(u_h; v_h) + \mathcal{S}_{\mathrm{SU},3}(u_h; v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_{h,0}, \tag{1.10}$$

where

$$\mathcal{S}_{\mathrm{SU},1}(u_h, f; v_h) := \sum_{i=1}^{m_t-1} h_{t,i} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \tau_1(t) \int_\Omega \big(\partial_t u_h + (1-\theta(\boldsymbol{x},t))(-\Delta u_h - f)\big)\partial_t v_h \,\mathrm{d}\Omega \,\mathrm{dt},$$

and

$$\mathcal{S}_{\mathrm{SU},2}(u_h; v_h) := \sum_{k=2}^{p_t} \sum_{i=1}^{m_t-1} h_{t,i}^{2k-1} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \tau_k(t) \int_\Omega \theta(\boldsymbol{x},t)\, \partial_t^k u_h \, \partial_t^k v_h \,\mathrm{d}\Omega \,\mathrm{dt},$$

while

$$\mathcal{S}_{\mathrm{SU},3}(u_h; v_h) := \sum_{k=1}^{p_t} \sum_{i=1}^{m_t-1} h_{t,i}^{2k} \int_{\zeta_{t,i}}^{\zeta_{t,i+1}} \sigma_k(t) \int_\Omega \theta(\boldsymbol{x},t) \left(\nabla(\partial_t^k u_h) \cdot \nabla(\partial_t^k v_h)\right) \mathrm{d}\Omega \,\mathrm{dt}.$$

As in Section 1.1.2, $\tau_k(T\,\cdot) \in \widehat{\mathcal{S}}_{h_t}^{p_t-k} \subset C^{p_t-k-1}$, are selected such that:

$$\int_0^T b'_{\ell+i,p_t} b_{i,p_t} \,\mathrm{dt} + \sum_{k=1}^{p_t} \sum_{j=1}^{m_t-1} h_{t,j}^{2k-1} \int_{\zeta_{t,j}}^{\zeta_{t,j+1}} \tau_k(t)\, b_{\ell+i,p_t}^{(k)} b_{i,p_t}^{(k)} \,\mathrm{dt} = 0,$$

for $i = 1, \ldots, N_t - 1$ and $\ell = 1, \ldots, r$, with $r = \min(p_t, N_t - i)$.
Moreover, $\sigma_k(T\,\cdot) \in \widehat{\mathcal{S}}_{h_t}^{p_t-k} \subset C^{p_t-k-1}$, are selected in order to make the time mass matrix lower triangular

$$\int_0^T b_{\ell+i,p_t} b_{i,p_t} \,\mathrm{dt} + \sum_{k=1}^{p_t} \sum_{j=1}^{m_t-1} h_{t,j}^{2k} \int_{\zeta_{t,j}}^{\zeta_{t,j+1}} \sigma_k(t)\, b_{\ell+i,p_t}^{(k)} b_{i,p_t}^{(k)} \,\mathrm{dt} = 0,$$

for $i = 1, \ldots, N_t - 1$ and $\ell = 1, \ldots, r$, with $r = \min(p_t, N_t - i)$.

The function $\theta(\boldsymbol{x}, t)$ ranges from 0 to 1. If we set θ as a fixed parameter equal to 1, (1.10) yields a lower block-triangular global system matrix. However, in order to achieve optimal convergence order, similar to the one-dimensional case (see Section 1.1.2). In detail, $\theta(\boldsymbol{x}, t)$ denotes the function defined in the physical domain,

which is associated with the $(d+1)$-linear interpolation in the parametric domain of the values corresponding to the breakpoints, for $i_l = 1, \ldots, m_l$, $l = 1, \ldots, d$, and $j = 1, \ldots, m_t$. The calculation of these values is as follows:

$$\Theta_{\boldsymbol{i},j} := \min(\mathrm{res}_{\boldsymbol{i},j}, 1),$$

with

$$\mathrm{res}_{\boldsymbol{i},j} := \frac{\|\partial_t u_h - \Delta u_h - f\|_{L^\infty(\psi_{s,\boldsymbol{i}} \times \psi_{t,j})}}{T^{-1}\|u_h\|_{L^\infty(\Omega\times[0,T])} + \|\partial_t u_h\|_{L^\infty(\Omega\times[0,T])}},$$

where

$$\psi_{s,\boldsymbol{i}} := [\widehat{\zeta}_{1,\max(1,i_1-1)}, \widehat{\zeta}_{1,\min(i_1+1,m_1)}] \times \ldots \times [\widehat{\zeta}_{d,\max(1,i_d-1)}, \widehat{\zeta}_{d,\min(i_d+1,m_d)}],$$

and

$$\psi_{t,j} := [\widehat{\zeta}_{t,\max(1,j-1)}, \widehat{\zeta}_{t,\min(j+1,m_t)}].$$

1.3 Numerical Results

All numerical tests in this chapter are conducted using MATLAB and GeoPDEs [55].

Just for the sake of simplicity, in all our tests, we consider splines of the same polynomial degree in all parametric directions for space and time. Specifically, we set $p_1 = \cdots = p_d = p_t := p$. Additionally, although the proposed methods are designed for maximum regularity only with respect to time, we adopt splines of global maximum continuity C^{p-1} also with respect to space. Numerical tests with different degrees and regularities (in space) indeed yield results entirely analogous to those reported below.

Nonlinearities in the equations are addressed through fixed-point iterations, and the resulting linear systems are solved using the direct solver provided by MATLAB (backslash operator "\").

In the numerical experiments, $\chi_{[0.3,0.6]}(t)$ refers to the characteristic function, defined as:

$$\chi_{[0.3,0.6]}(t) := \begin{cases} 1 & \text{for } t \in [0.3, 0.6], \\ 0 & \text{otherwise.} \end{cases}$$

1.3.1 Advection equation

We consider the advection equation (1.1) on $(0, T)$ with $T = 1$ and uniform meshes.

Test with smooth solution

We deal with a smooth solution on the whole domain, to analyze the convergence order of our method. We are interested in error estimates with respect to the L^2-norm, under h-refinement. To obtain this mesh refinement it is necessary to insert additional knots in the original knot vector, without changing the multiplicity of each knot.

We consider $u_{\text{ex}} \in H^{p+1}(0, T)$ the exact solution of problem (1.1), with $p \geq 1$ and $\Pi_{\mathcal{X}_{t,h_t}} u_{\text{ex}}$ is the application of a quasi-interpolation operator (defined, e.g., in [5, Section 2.1.5]) to the function u_{ex}, that yields to an approximation of u_{ex} within a discrete space $\mathcal{X}_{t,h_t}$. Therefore, we can write the following error estimate under h–refinement in L^2-norm:

$$\frac{\left\| u_{\text{ex}} - \Pi_{\mathcal{X}_{t,h_t}} u_{\text{ex}} \right\|_{L^2(0,T)}}{\left\| u_{\text{ex}} \right\|_{H^{p+1}(0,T)}} \leq C\, h_t^{p+1},$$

where C is a suitable constant independent of h_t.

To assess the convergence order of the L^2-norm of error, a numerical validation is conducted. This involves a detailed graphical examination to analyze the dependency of error behavior on grid refinement and polynomial degree, thereby facilitating a comprehensive assessment of the method convergence properties.

The methodology begins with the generation of approximate solutions, utilizing varying discretization parameters h_t and polynomial degrees p. The discrepancy between the exact and approximated solutions is subsequently quantified using L^2-norm of error. The analysis entails plotting the relative errors on a base-10 logarithmic scale, with h_t on the x-axis and relative error in L^2-norm on the y-axis. Each curve on this plot represents the error behavior for specific combinations of h_t and p. A meticulous examination of these curves reveals the error convergence trend with respect to h_t and p. If the linear relationship in the log-log plot is characterized by a slope approaching $p + 1$, we have a numerical scheme with optimal convergence order in L^2-norm.

In this test we consider $f = 50\cos(50t)$, that leads to the exact solution $u_{\text{ex}}(t) = \sin(50t)$.

In Figure 1.3, we plot the x- and y-coordinates using logarithmic scale on the x-axis and the y-axis. We present the error plot for the SU formulation on uniform meshes with degree $p = 1, \ldots, 6$. The method exhibits optimal convergence, as evidenced by the slopes of the lines representing the error trends with mesh refinement, which align $p + 1$ slopes.

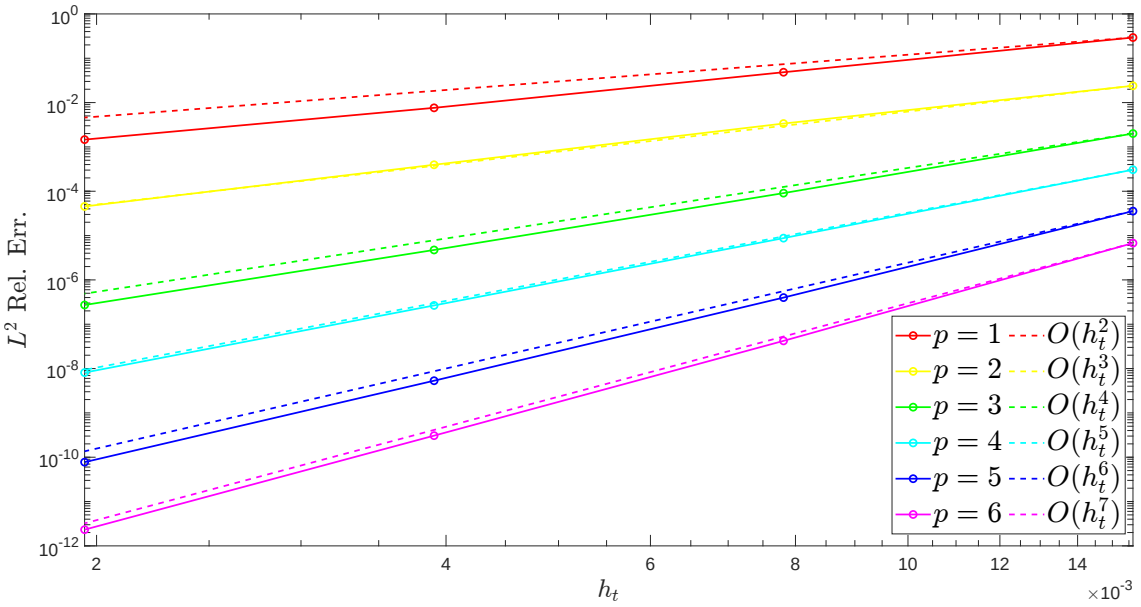

Figure 1.3: Advection equation, SU relative error plots in L^2-norm (test in Section 1.3.1, smooth solution).

Solution with layers

In these tests we chose a reasonably coarse uniform mesh ($h_t = 2^{-6}$) and we take into account solutions characterized by sharp layers. Our aim is therefore to analyze the presence of spurious oscillations due to strong gradients in the solutions and to examine how the stabilization method we proposed improves the solutions from a numerical stability perspective.

We select f such that the exact solution is

$$u_{\text{ex}}(t) = \sin(50t) + 10\frac{1 + \tanh(\frac{t-t_0}{\delta})}{2} - 5\frac{1 + \tanh(\frac{t-t_1}{\delta})}{2} - 5\frac{1 + \tanh(\frac{t-t_2}{\delta})}{2},$$

with $t_0 = 0.3$, $t_1 = 0.5$, $t_2 = 0.7$ and $\delta = 10^{-3}$.

Due to the presence of internal layers, the plain Galerkin method is unstable as we expect: the numerical solution differs considerably from the exact solution, with high numerical instabilities (see Figure 1.4, for $p = 3$).

Figure 1.5 shows that, for $p = 3$, the SUPG method improves the numerical approximation compared to the plain Galerkin approach but still exhibits numerical instabilities near the layers. By tuning the parameter τ_{SUPG}, as presented in Figure 1.6, for $p = 3$, it is evident that spurious oscillations persist in all cases. However, $\tau_{\text{SUPG}} = \frac{h_t}{2}$ can be considered the suitable value not only for $p = 1$ (as discussed in Section 1.1.1) but also for higher-degree splines.

Adding Shock Capturing with $\tau_{\text{SC}} = h_t^2$ (Figure 1.7), spurious oscillations are further reduced if compared to previous methods but some spurious oscillations can still be noticed in the vicinity of the layers. Also in this case a tuning on τ_{SC} (Figure 1.8, for $p = 3$) shows that the numerical instabilities are not completely removed.

With the NCSU method (Figure 1.9, for $p = 3$) spurious oscillations disappear near the layers but the numerical approximation is not satisfactory: significant numerical and phase errors can be noticed in particular where the solution is smooth.

Figures 1.10 and 1.11 present numerical results for $p = 3$ and $p = 4$ (similar results are obtained for other spline degrees), assessing the behavior of the SU formulation on uniform meshes. The solutions exhibit no numerical instabilities near the layers, and from a qualitative perspective, they provide a good numerical approximation of the exact solution in the smooth regions. To deepen the analysis, we examine the convergence order in the L^2-norm where the solution is smooth, i.e., after the three layers, for $t > 0.85$. The relative error plots in Figure 1.12, corresponding to splines of different degrees, indicate optimal convergence of the relative error.

Since our stabilization method is designed for application to non-uniform meshes as well, similar stable and accurate behavior is observed in this case, as shown in Figure 1.13, for $p = 3$. Indeed, numerical instabilities near the layers are completely removed, similarly to the case of using uniform meshes.

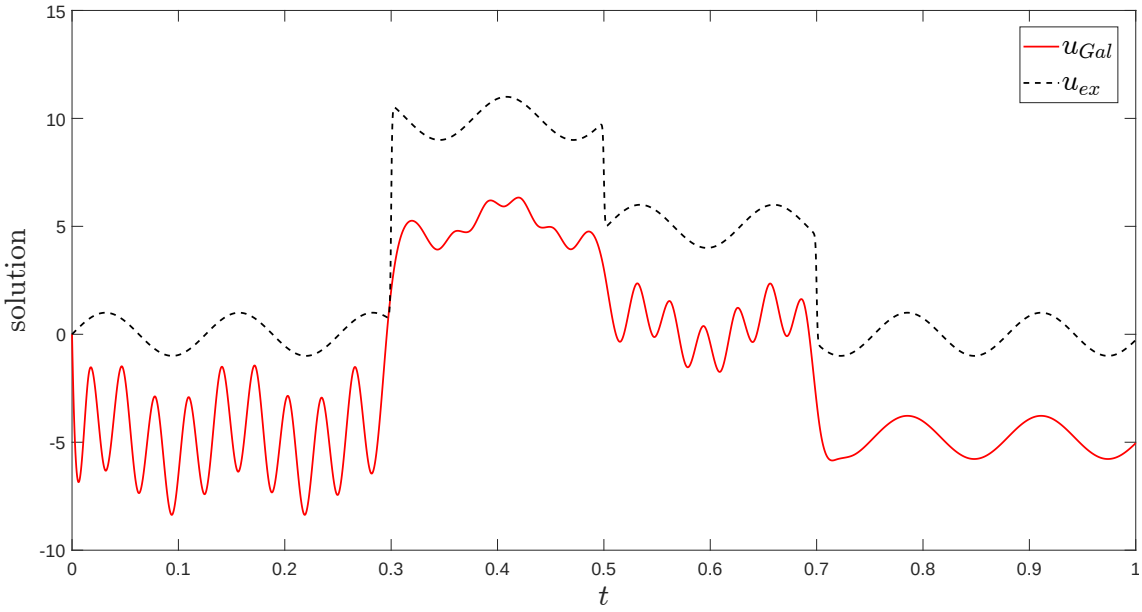

Figure 1.4: Advection equation, exact and plain Galerkin solutions (test in Section 1.3.1, solution with layers).

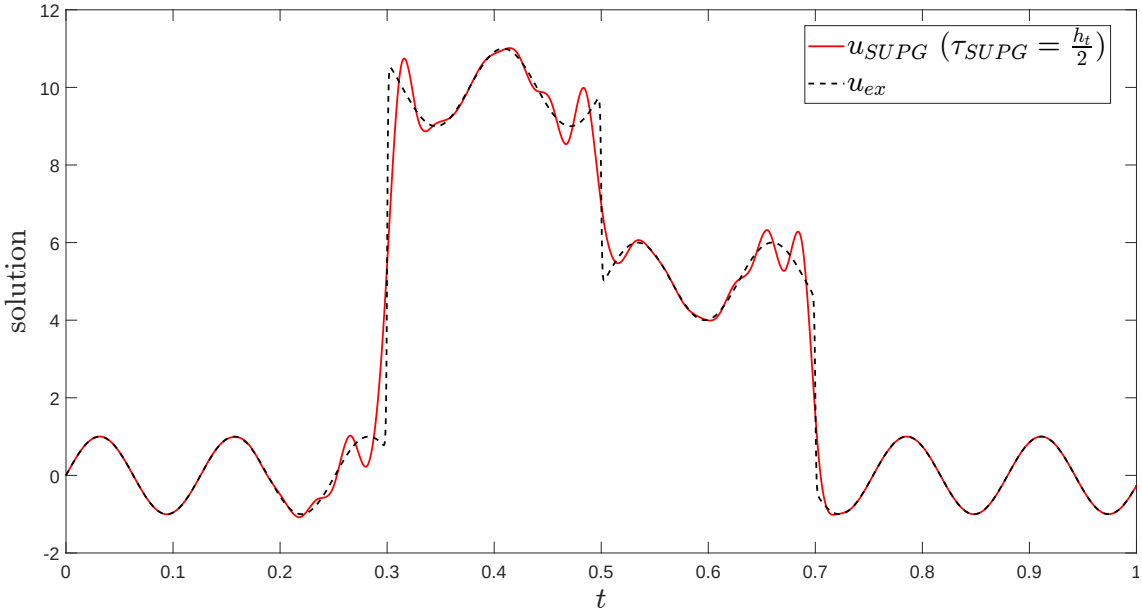

Figure 1.5: Advection equation, exact and SUPG solutions (test in Section 1.3.1, solution with layers).

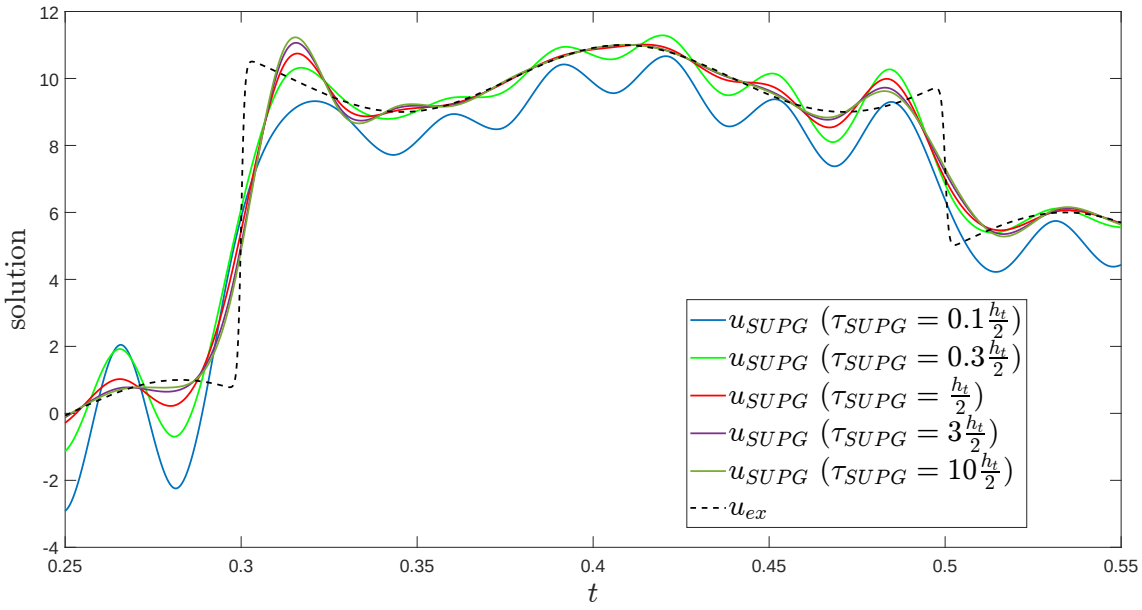

Figure 1.6: Advection equation, exact and SUPG solutions for different choices for τ_{SUPG} (test in Section 1.3.1, solution with layers).

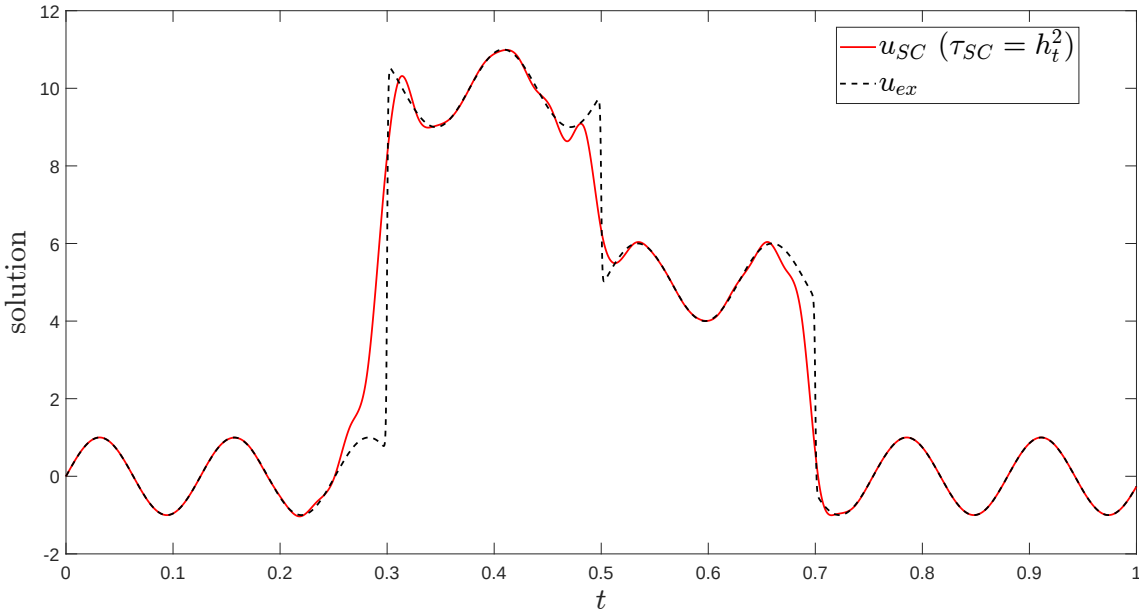

Figure 1.7: Advection equation, exact and Shock Capturing solutions (test in Section 1.3.1, solution with layers).

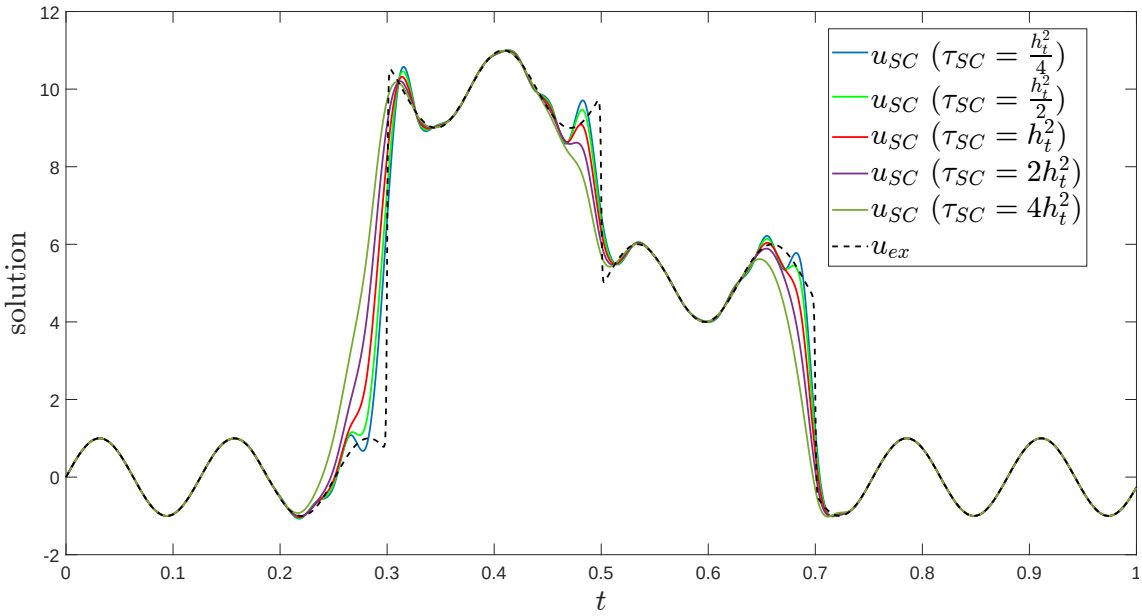

Figure 1.8: Advection equation, exact and Shock Capturing solutions for different choices for τ_{SC} (test in Section 1.3.1, solution with layers).

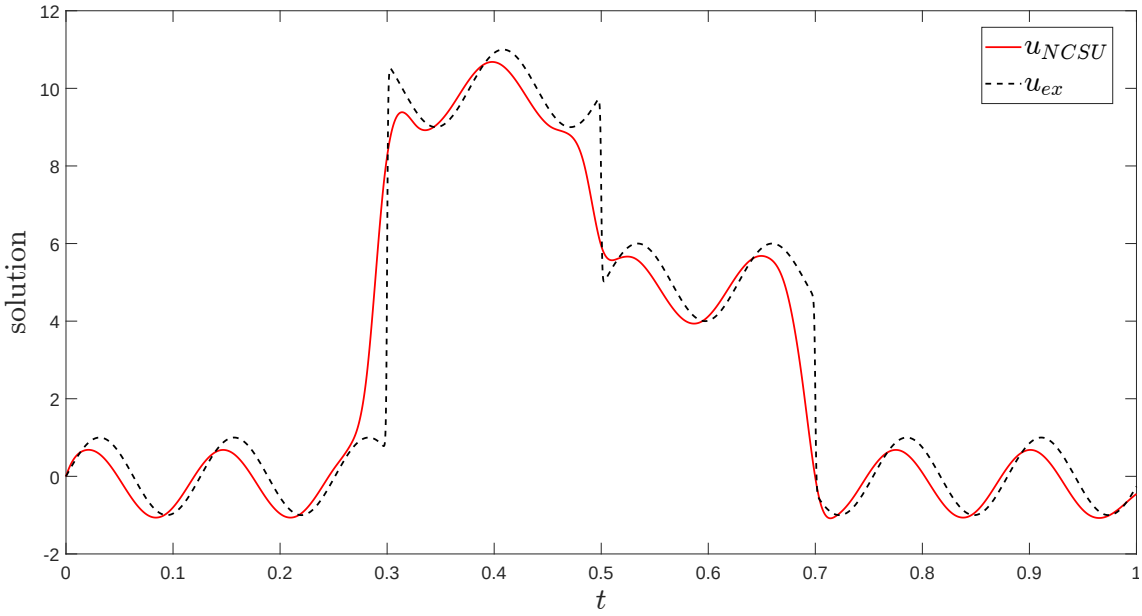

Figure 1.9: Advection equation, exact and NCSU solutions (test in Section 1.3.1, solution with layers).

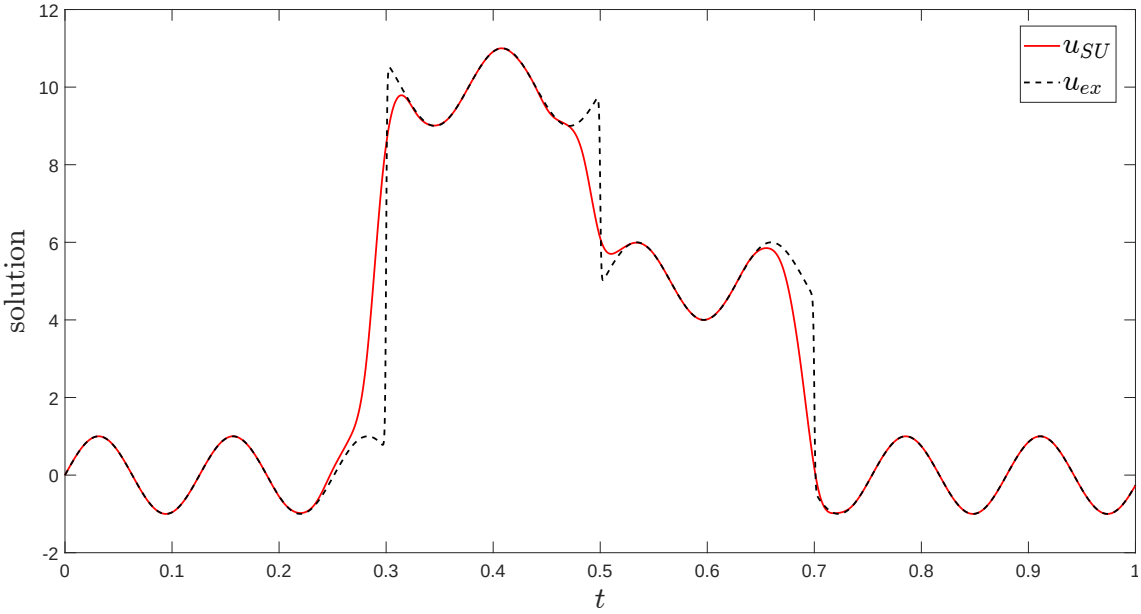

Figure 1.10: Advection equation, exact and SU solutions for $p = 3$ (test in Section 1.3.1, solution with layers).

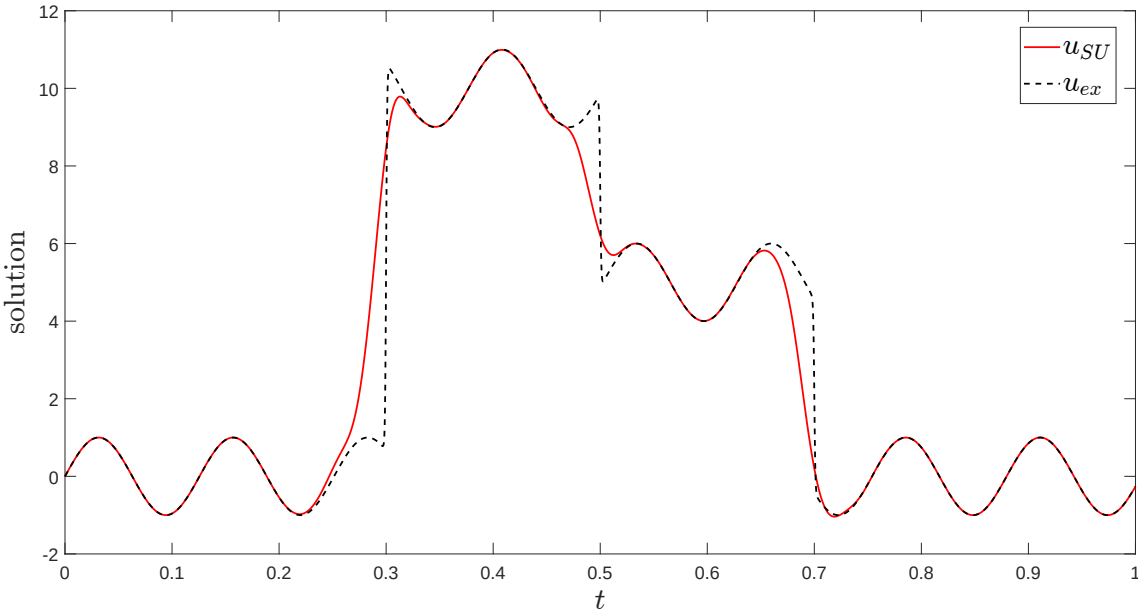

Figure 1.11: Advection equation, exact and SU solutions for $p = 4$ (test in Section 1.3.1, solution with layers).

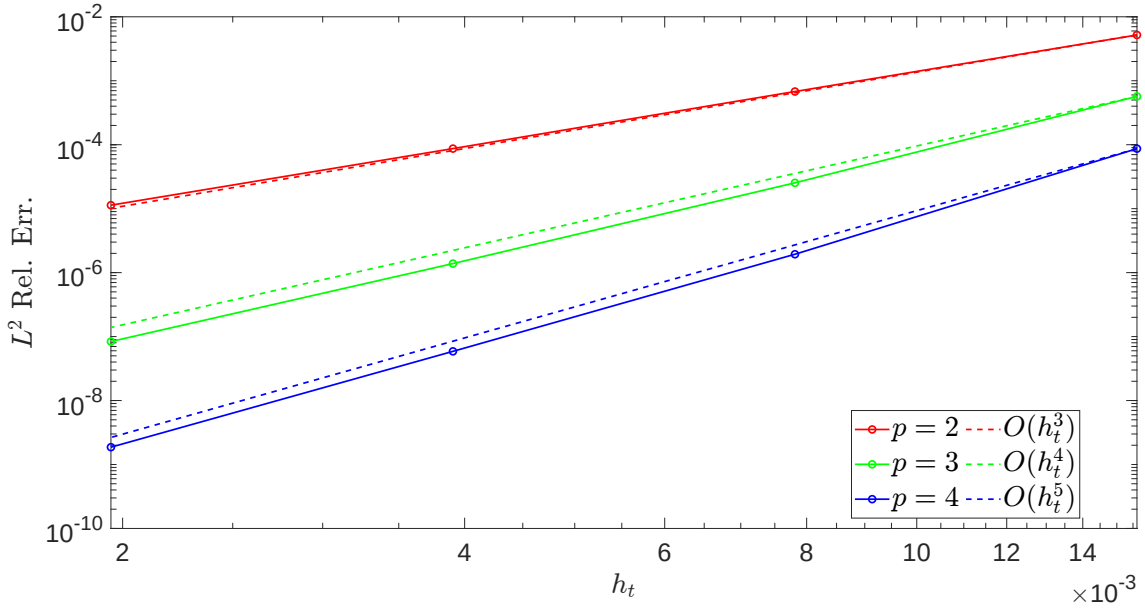

Figure 1.12: Advection equation, SU relative error plots in L^2-norm, computed where the solution is smooth ($t > 0.85$) (test in Section 1.3.1, solution with layers).

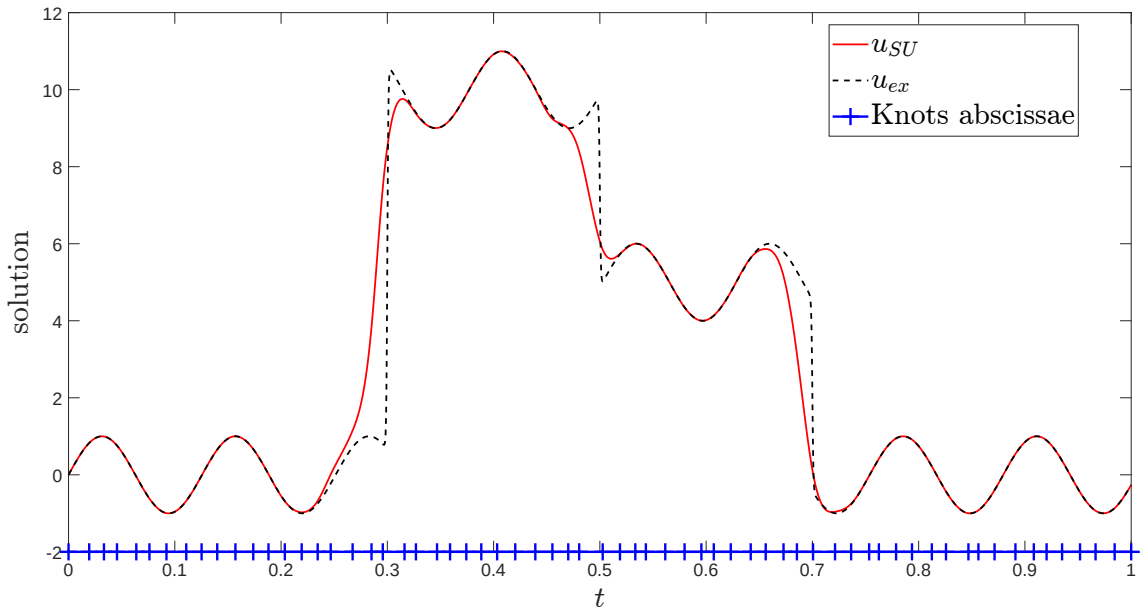

Figure 1.13: Advection equation, exact and SU solutions on a non-uniform mesh, depicted in blue on the horizontal axis (test in Section 1.3.1, solution with layers).

1.3.2 Advection–diffusion equation

We consider the advection–diffusion equation (1.6) on $(0, T)$ with $T = 1$, $f = 1$, $\varepsilon = 10^{-6}$ and uniform mesh with $h_t = 2^{-6}$.

Our objective is to analyze the behavior of the SU stabilization method by comparing its numerical approximation to that of the SUPG method.

Figures 1.14 and 1.15 illustrate the comparison between SUPG and SU solutions for $p = 3$ and $p = 4$, respectively. The SU method shows higher accuracy, though at an increased computational cost. Specifically, SU solutions are sharper than those obtained with SUPG and provide a better approximation of the layer near $T = 1$.

In our implementation, the increased cost arises from the need to evaluate the residual at each fixed-point iteration. However, a thorough exploration of the computational cost and efficient implementation is beyond the scope of the present chapter.

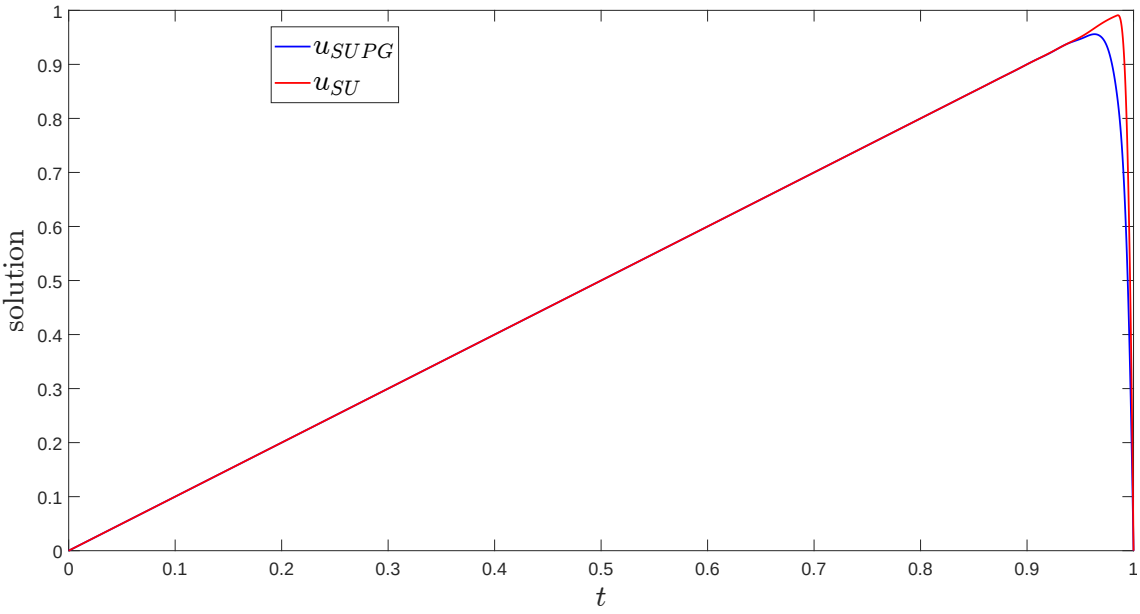

Figure 1.14: Advection–diffusion equation, SUPG and SU solutions for $p = 3$, (test in Section 1.3.2).

1.3.3 Heat equation

We consider the heat equation (1.7) on $\Omega \times (0, T)$ with $T = 1$, within the space–time framework, and different space domains Ω, focusing on solutions characterized by sharp layers arising from concentrated source terms. We want to analyze the presence

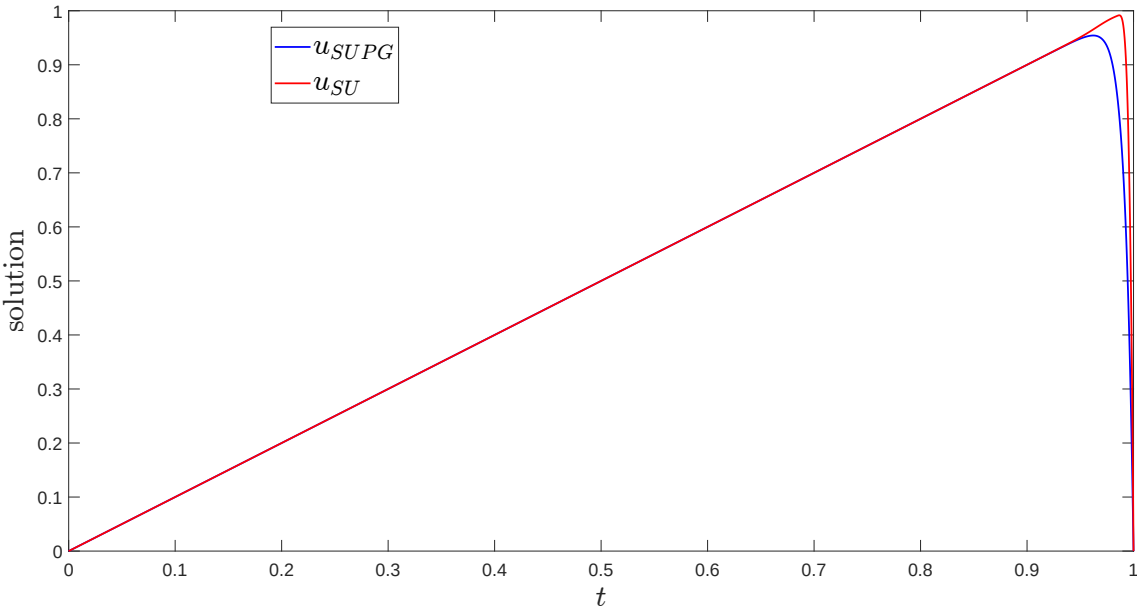

Figure 1.15: Advection–diffusion equation, SUPG and SU solutions for $p = 4$, (test in Section 1.3.2).

of spurious oscillations caused by strong gradients in the solutions and to evaluate how the SU method enhances the solutions from the perspective of numerical stability.

In these tests, we chose relatively coarse uniform meshes, $p = 3$ and we denote with x and y the spatial Cartesian coordinates and with t the time coordinate.

1D spatial domain

For the first test, we take into account $\Omega = (0, 1)$ and we set $h_1 = h_t =: h = 2^{-6}$. The function f is given by

$$f(x,t) = C_{1\mathrm{D}} \exp\left(-\left(\frac{x - \frac{1}{4}\left(\sin\left(10\pi t\right) + 2\right)}{\delta}\right)^2\right) \chi_{[0.3,0.6]}(t),$$

where we set $C_{1\mathrm{D}} = 1/\delta^2$ and $\delta = 10^{-3}$.

In Figures 1.16, 1.17, and 1.18, we present the numerical solutions obtained using the plain Galerkin, the SUPG and the SU methods, respectively. Notably, spurious oscillations are observed, particularly with the plain Galerkin method, where significant oscillations are present even for $t < 0.3$ before the source is activated, where zero solution is expected. Additionally, both the plain Galerkin and the SUPG methods exhibit numerical instabilities at the peak of the solution generated by the source activation, within the time interval $t \in [0.3, 0.6]$.

On the other hand, the numerical results of the SU method (Figure 1.18) show that spurious oscillations are eliminated. The SU stabilization method effectively removes spurious oscillations across the entire domain, both before and during the activation of the source term.

Figure 1.19 presents the function $\theta(x,t)$, which activates the high-order Upwind stabilization near sharp layers. Indeed, the function $\theta(x,t)$ acts as an indicator, guiding the activation of the stabilization technique to address sharp layers in the solution, reaching a value of 1 where the residual is high.

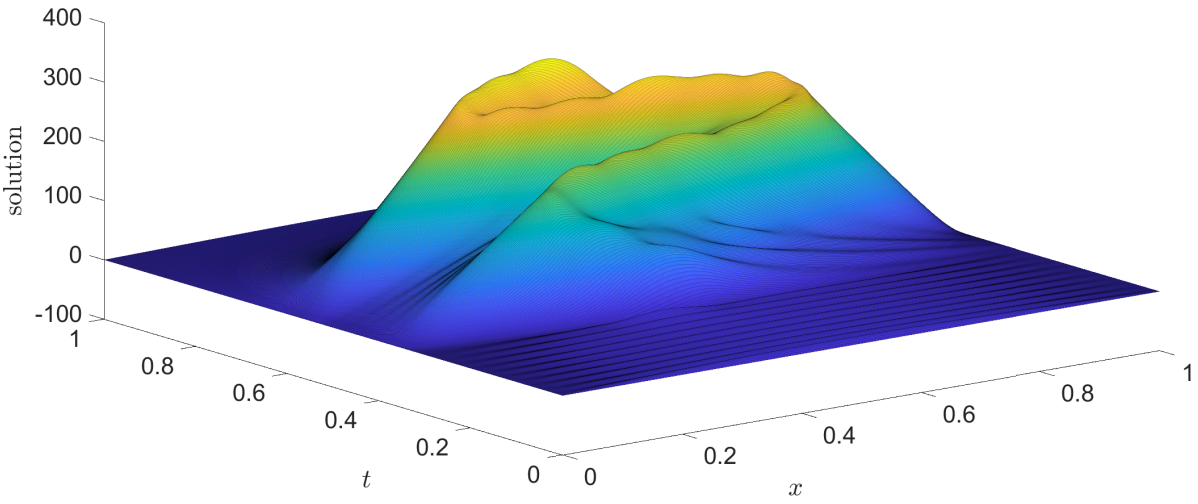

Figure 1.16: Heat equation, plain Galerkin solution (test in Section 1.3.3, $\Omega = (0,1)$).

2D spatial domain

For the second test, Ω is a quarter annulus (Figure 1.20, with depicted also the section line s for solution plots) and we set $\widehat{h}_1 = \widehat{h}_2 = \widehat{h}_t =: \widehat{h} = 2^{-5}$, while

$$f(x,y,t) = C_{2\mathrm{D}} \exp\left(-\frac{1}{2}\left(\left(\frac{x-\frac{3}{2}\cos\left(\frac{\pi}{2}t\right)}{\delta}\right)^2 + \left(\frac{y-\frac{3}{2}\sin\left(\frac{\pi}{2}t\right)}{\delta}\right)^2\right)\right) \chi_{[0.3,0.6]}(t),$$

where $C_{2\mathrm{D}} = 10^3/(2\pi\delta^2)$ and $\delta = 10^{-1}$.

In Figures 1.21 and 1.22, we present the numerical results that assess the behavior of the plain Galerkin approximation and the SUPG method. Spurious oscillations

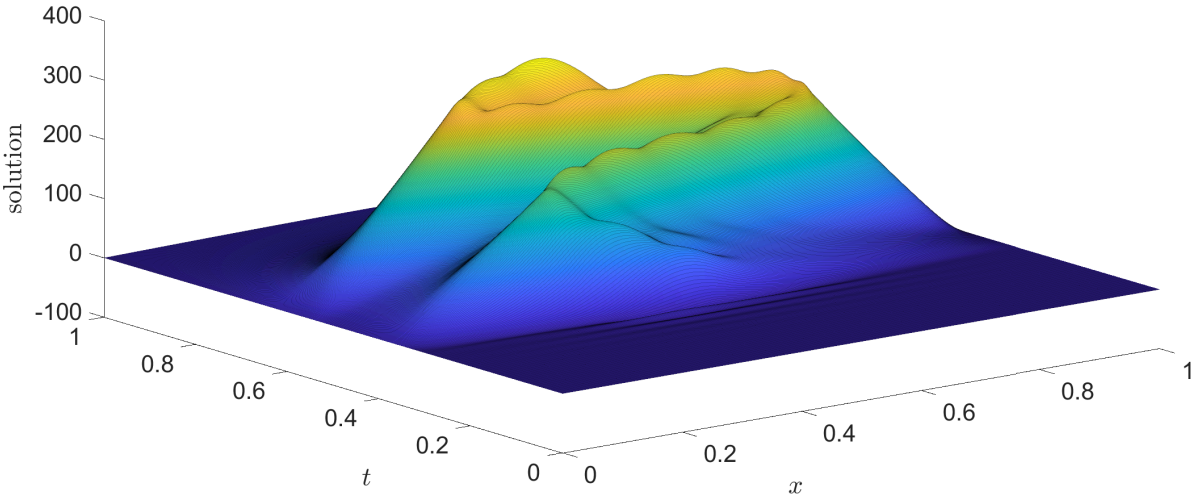

Figure 1.17: Heat equation, SUPG solution (test in Section 1.3.3, $\Omega = (0,1)$).

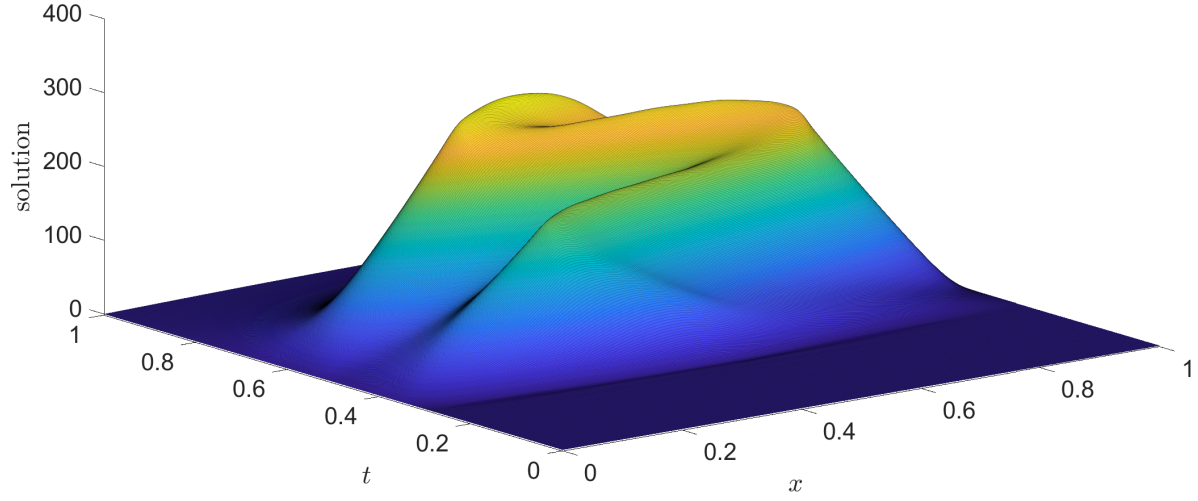

Figure 1.18: Heat equation, SU solution (test in Section 1.3.3, $\Omega = (0,1)$).

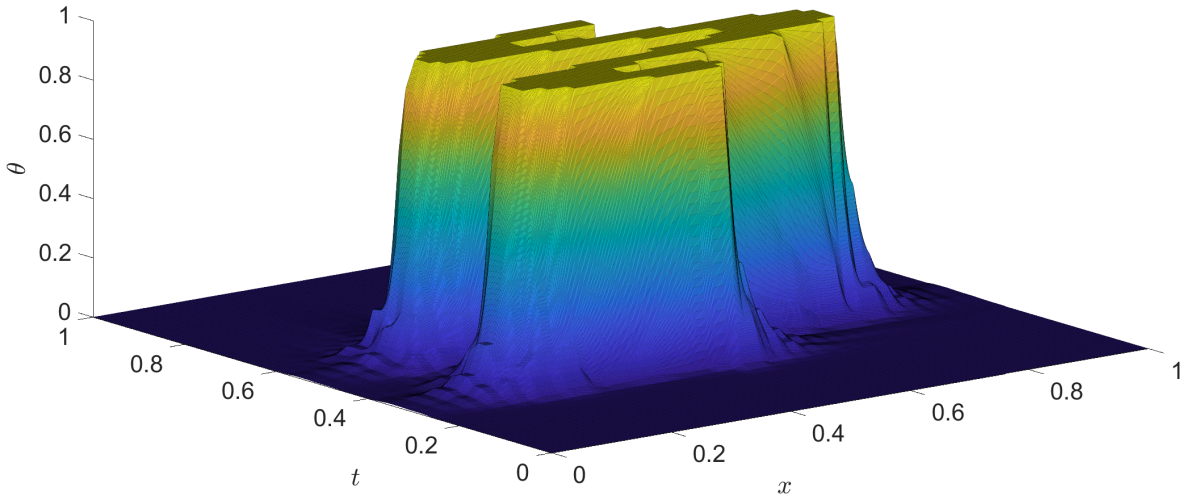

Figure 1.19: Heat equation, function $\theta(x,t)$ in the SU method (test in Section 1.3.3, $\Omega = (0,1)$).

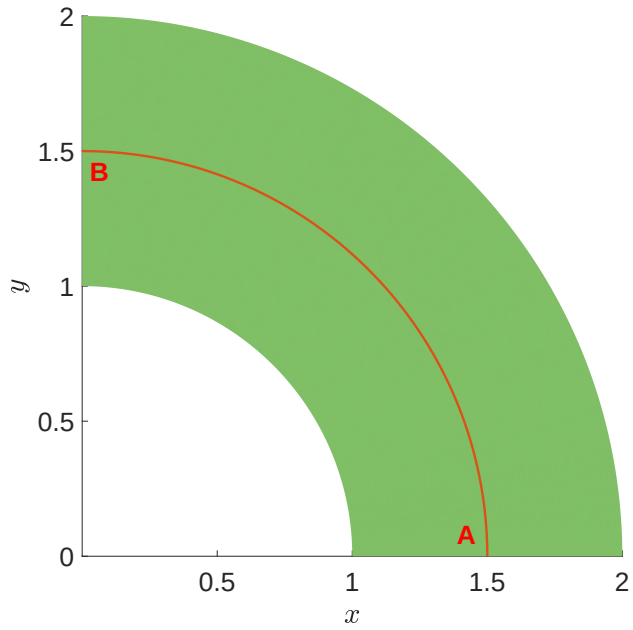

Figure 1.20: Spatial domain Ω, a quarter annulus, with section line s (A–B), for second test in Section 1.3.3.

are observed, particularly in the case of the plain Galerkin method, where significant oscillations are present even for $t < 0.3$, before the source activation. Both methods exhibit numerical instabilities for $t \in [0.3, 0.6]$ at the solution peak generated by the source activation.

In contrast, the numerical results for the SU method, shown in Figure 1.23, reveal that this method effectively removes spurious oscillations across the entire domain.

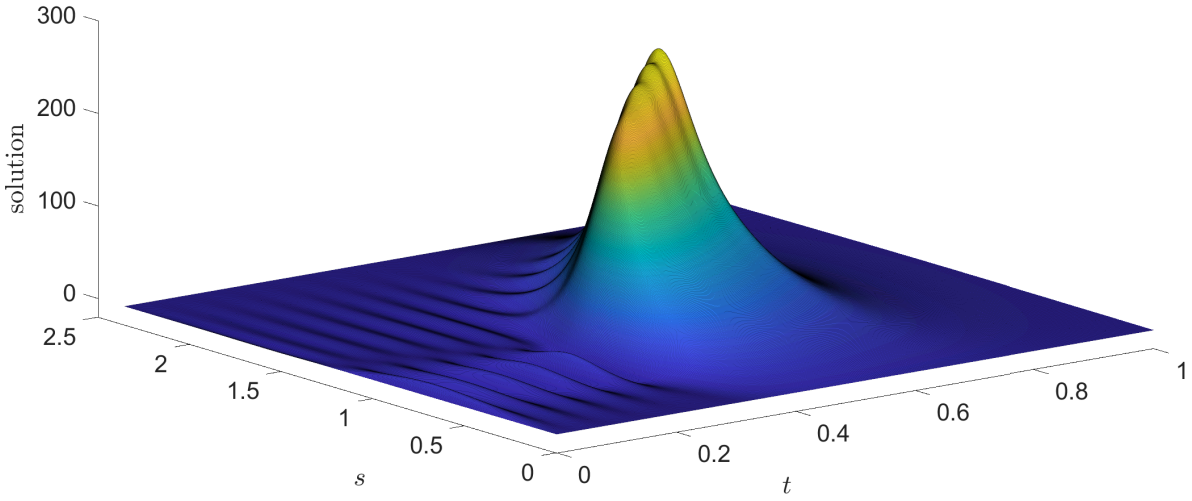

Figure 1.21: Heat equation, plain Galerkin solution along section line s (test in Section 1.3.3, Ω is a quarter annulus).

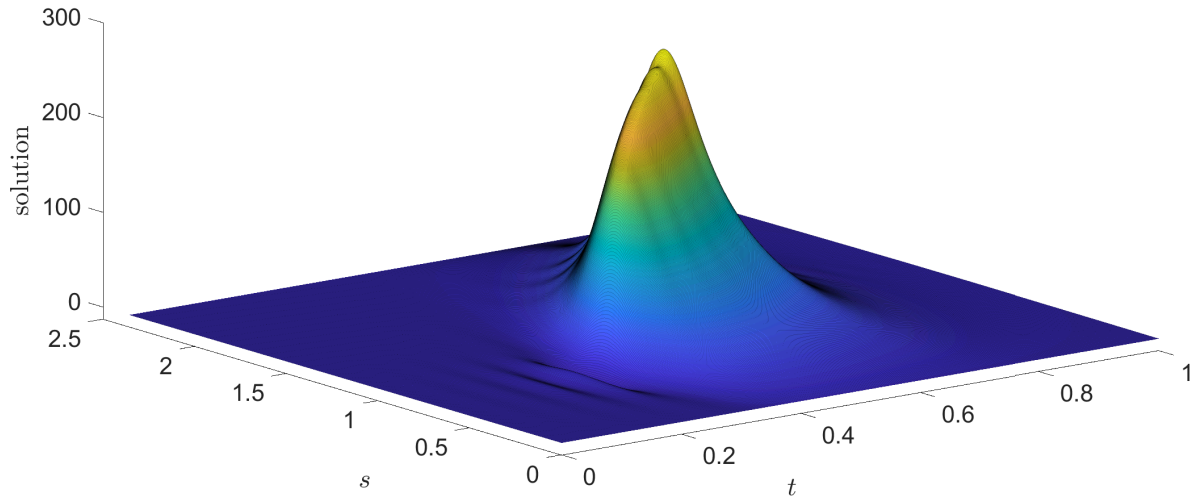

Figure 1.22: Heat equation, SUPG solution along section line s (test in Section 1.3.3, Ω is a quarter annulus).

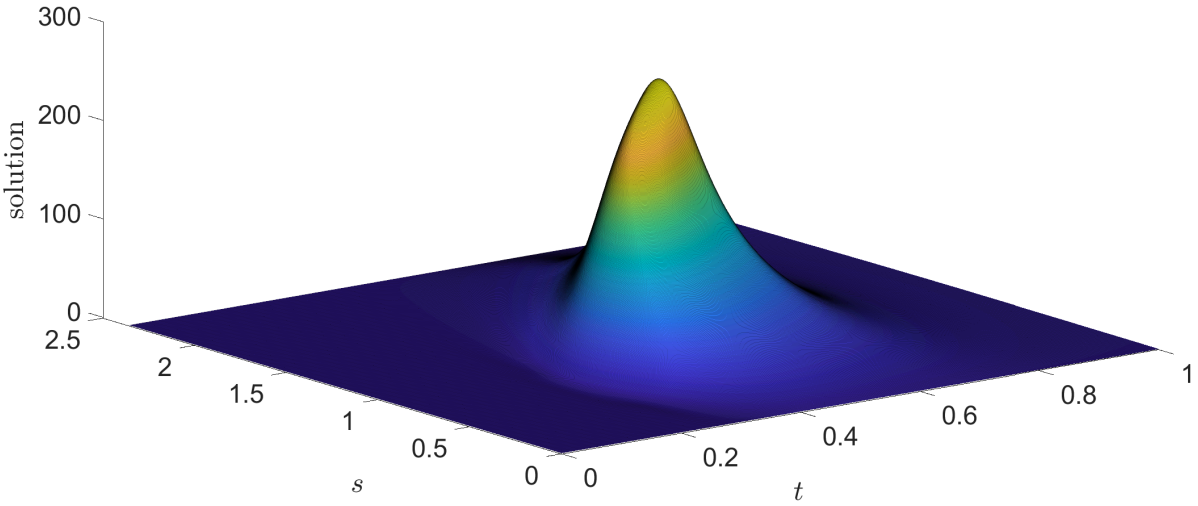

Figure 1.23: Heat equation, SU solution along section line s (test in Section 1.3.3, Ω is a quarter annulus).

Chapter 2

Space–time Isogeometric Analysis of cardiac electrophysiology

The purpose of this chapter is to study the behavior of the SU stabilization method in the context of cardiac electrophysiology, using space–time IgA discretization. We modify the SU method to enhance its effectiveness and computational efficiency, applying it to the monodomain equation coupled with the Rogers–McCulloch ionic model [47], a widely used model for simulating electrophysiological wave propagation in the heart. It is based on a differential problem governed by differential equations, allowing for the simulation of front propagation phenomena.

We perform several numerical tests and we compare the computational costs of both the plain Galerkin method and the SU method.

2.1 Spline Upwind for cardiac electrophysiology

The dimensionless unknowns, in the monodomain equation coupled with the Rogers–McCulloch ionic model, are the transmembrane potential u (the main unknown) and the recovery variable w.

The governing differential problem is

$$\begin{cases} C_m\partial_t u - D\Delta u + c_1 u(u-a)(u-1) + c_2 uw &= f \quad \text{in} \quad \Omega\times(0,T), \\ \partial_t w - b(u - d_e w) &= 0 \quad \text{in} \quad \Omega\times(0,T), \\ \frac{\partial u}{\partial \boldsymbol{n}} &= 0 \quad \text{on} \quad \partial\Omega\times[0,T], \\ u &= 0 \quad \text{in} \quad \Omega\times\{0\}, \\ w &= 0 \quad \text{in} \quad \Omega\times\{0\}, \end{cases}$$

where $\boldsymbol{n}$ is the exterior normal; a, b, c_1, c_2 and d_e are dimensionless parameters, specific of the Rogers–McCulloch model. The dimensionless constants C_m and D are the local membrane capacitance and the conductivity, respectively. In particular, the conductivity is very small with respect to the membrane capacitance (as in [42]). We assume that $f \in L^2\big(\Omega \times (0,T)\big)$.

We consider the plain Galerkin method

find u_h,$w_h \in \mathcal{X}_h$ such that:

$$\begin{cases} \mathcal{A}(u_h, w_h; v_h) &= \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_h, \\ \mathcal{L}(w_h, u_h; z_h) &= 0 \quad \forall z_h \in \mathcal{X}_h, \end{cases} \tag{2.1}$$

where

$$\mathcal{A}(u_h, w_h; v_h) := \int_0^T \int_\Omega \big(C_m\partial_t u_h\, v_h + D\nabla u_h\cdot\nabla v_h + (c_1(u_h-a)(u_h-1) + c_2 w_h)\; u_h\, v_h\big)\,\mathrm{d}\Omega\,\mathrm{dt},$$

$$\mathcal{F}(v_h) := \int_0^T \int_\Omega f\, v_h\,\mathrm{d}\Omega\,\mathrm{dt}, \tag{2.2}$$

and

$$\mathcal{L}(w_h, u_h; z_h) := \int_0^T \int_\Omega \big(\partial_t w_h\, z_h - b(u_h - d_e w_h)\, z_h\big)\,\mathrm{d}\Omega\,\mathrm{dt}.$$

To enhance stability within the Galerkin framework, we consider the SU method, introduced in Chapter 1, and we adapt our stabilization method to be more suitable for this context. In detail, we consider the plain Galerkin formulation where the solution is smooth, and we introduce the stabilization only regarding the time derivative term of $\mathcal{A}(u_h, w_h; v_h)$, where the residual is high. Following these ideas, the SU method for cardiac electrophysiology reads:

find u_h,$w_h \in \mathcal{X}_h$ such that:

$$\begin{cases} \mathcal{A}(u_h, w_h; v_h) + \mathcal{S}_{\mathrm{SU}}(u_h; v_h) &= \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{X}_h, \\ \mathcal{L}(w_h, u_h; z_h) &= 0 \quad \forall z_h \in \mathcal{X}_h, \end{cases} \tag{2.3}$$

where for the stabilization terms $\mathcal{S}_{\mathrm{SU}}(u_h; v_h)$, we extend the definitions presented in Chapter 1. In particular, the term $\mathcal{S}_{\mathrm{SU}}(u_h; v_h)$ reads as follows:

$$\mathcal{S}_{\mathrm{SU}}(u_h; v_h) := C_m \sum_{k=1}^{p} \int_0^T \tau_k(t) \int_\Omega \theta(\boldsymbol{x}, t)\, \partial_t^k u_h\, \partial_t^k v_h \, \mathrm{d}\Omega \, \mathrm{dt}, \tag{2.4}$$

where, for $k = 1, \cdots, p_t$, $\tau_k(T\,\cdot) \in \widehat{\mathcal{S}}_{h_t}^{p_t-k} \subset C^{p_t-k-1}$, are selected such that:

$$\int_0^T b'_{\ell+i,p_t} b_{i,p_t} \, \mathrm{dt} + \sum_{k=1}^{p_t} \int_0^T \tau_k(t)\, b^{(k)}_{\ell+i,p_t} b^{(k)}_{i,p_t} \, \mathrm{dt} = 0, \tag{2.5}$$

for $i = 1, \ldots, N_t - 1$ and $\ell = 1, \ldots, r$, with $r = \min(p_t, N_t - i)$.
For simplicity, in (2.5), we incorporate the temporal scaling term, unlike in Chapter 1, where it is stated explicitly.

Furthermore, $\theta(\boldsymbol{x}, t)$ denotes the function defined in the physical domain, which is associated with the $(d+1)$-linear interpolation in the parametric domain of the values corresponding to the Greville abscissae, for $\boldsymbol{i_s} := (i_1, \ldots, i_d)$, with $i_l = 1, \ldots, n_l$, $l = 1, \ldots, d$ and $i_t = 1, \ldots, N_t$. The calculation of these values is as follows:

$$\Theta_{\boldsymbol{i_s}, i_t} := \min\left(\frac{\|C_m \partial_t u_h - D\Delta u_h + c_1 u_h(u_h - a)(u_h - 1) + c_2 u_h w_h - f\|_{L^\infty(\widetilde{I}_{\boldsymbol{i_s}, i_t})}}{C_m(T^{-1}\,\|u_h\|_{L^\infty(\Omega\times[0,T])} + \|\partial_t u_h\|_{L^\infty(\Omega\times[0,T])})}, 1 \right), \tag{2.6}$$

where $\widetilde{I}_{\boldsymbol{i_s}, i_t}$ is defined in (4).

Finally, in order to speed up the computation of the stabilizing term, we consider the low-rank matrix approximation technique (see, e.g., [36]).

Given a relative tolerance $\varepsilon > 0$, using the algorithm in [58], we find $R \in \mathbb{N}$ such that $0 < R \leq \min(N_t, N_s)$, $\mathbf{U} \in \mathbb{R}^{N_t \times R}$, $\mathbf{V} \in \mathbb{R}^{N_s \times R}$ and $\mathbf{R} \in \mathbb{R}^{R \times R}$ a diagonal matrix such that:

$$\frac{\left\| \widehat{\boldsymbol{\Theta}} - \mathbf{U}\mathbf{R}\mathbf{V}^T \right\|_{\mathrm{F}}}{\left\| \widehat{\boldsymbol{\Theta}} \right\|_{\mathrm{F}}} \leq \varepsilon, \tag{2.7}$$

where $\widehat{\boldsymbol{\Theta}} \in \mathbb{R}^{N_t \times N_s}$ is obtained by reshaping the tensor (2.6) and $\|\cdot\|_{\mathrm{F}}$ is the Frobenius

norm.

In this way, we can write the function $\theta(\boldsymbol{x}, t)$ as:

$$\theta(\boldsymbol{x}, t) \approx \sum_{r=1}^{R} [\mathbf{R}]_{r,r} \theta_{t,r}(t)\, \theta_{s,r}(\boldsymbol{x}),$$

where $\theta_{t,r}(t)$ is a linear interpolation of the r-th column of $\mathbf{U}$, on the Greville abscissae in the time direction and $\theta_{s,r}(\boldsymbol{x})$ is a d-linear interpolation of the reshaping of r-th column of $\mathbf{V}$, on the Greville abscissae in the space directions.

Therefore, we have the following approximation:

$$\mathcal{S}_{\mathrm{SU}}(u_h; v_h) \approx \widetilde{\mathcal{S}}_{\mathrm{SU}}(u_h; v_h) := C_m \sum_{r=1}^{R} [\mathbf{R}]_{r,r} \sum_{k=1}^{p} \int_0^T \tau_k(t)\, \theta_{t,r}(t) \int_\Omega \theta_{s,r}(\boldsymbol{x})\, \partial_t^k u_h\, \partial_t^k v_h \,\mathrm{d}\Omega \,\mathrm{d}t. \tag{2.8}$$

2.1.1 Nonlinear solver

We solve (2.1) and (2.3) by the following fixed-point scheme:

given $u_h^0, w_h^0 \in \mathcal{X}_h$, the $k+1$-th iteration consists in finding $\widetilde{u}_h^{k+1}$, $\widetilde{w}_h^{k+1} \in \mathcal{X}_h$ such that:

$$\begin{cases} \mathcal{A}(\widetilde{u}_h^{k+1}, u_h^k, w_h^k; v_h) + \widetilde{\mathcal{S}}_{\mathrm{SU}}(\widetilde{u}_h^{k+1}; v_h) &= \mathcal{F}(v_h) & \forall v_h \in \mathcal{X}_h, \\ \mathcal{L}(\widetilde{w}_h^{k+1}; z_h) &= \mathcal{G}(u_h^k; z_h) & \forall z_h \in \mathcal{X}_h, \end{cases} \tag{2.9}$$

where, with abuse of notation, in (2.9) we used

$$\mathcal{A}(\widetilde{u}_h^{k+1}, u_h^k, w_h^k; v_h) := \int_0^T \int_\Omega \left(C_m\, \partial_t \widetilde{u}_h^{k+1}\, v_h + D\, \nabla \widetilde{u}_h^{k+1} \cdot \nabla v_h + C_r(u_h^k, w_h^k) \widetilde{u}_h^{k+1}\, v_h \right) \mathrm{d}\Omega \,\mathrm{d}t,$$

with

$$C_r(u_h^k, w_h^k) := c_1(u_h^k - a)(u_h^k - 1) + c_2 w_h^k, \tag{2.10}$$

$$\mathcal{L}(\widetilde{w}_h^{k+1}; z_h) := \int_0^T \int_\Omega \left(\partial_t \widetilde{w}_h^{k+1} + b\, d_e\, \widetilde{w}_h^{k+1} \right) z_h \,\mathrm{d}\Omega \,\mathrm{d}t,$$

and

$$\mathcal{G}(u_h^k; z_h) := \int_0^T \int_\Omega b\, u_h^k\, z_h \,\mathrm{d}\Omega \,\mathrm{d}t,$$

while $\mathcal{F}$ and $\widetilde{\mathcal{S}}_{\mathrm{SU}}$ are defined in (2.2) and (2.8), respectively. System (2.9) consists of

two decoupled equations, each of which can be solved independently.

Moreover, in order to promote the convergence of fixed-point iterations, we introduce relaxations

$$u_h^{k+1} = \alpha \widetilde{u}_h^{k+1} + (1-\alpha) u_h^k \quad \text{and} \quad w_h^{k+1} = \alpha \widetilde{w}_h^{k+1} + (1-\alpha) w_h^k,$$

where we set $\alpha = 0.5$.

Discrete linear system

The linear systems resulting from (2.9) are

$$\mathbf{A}\widetilde{\mathbf{u}}^{k+1} = \mathbf{f}, \tag{2.11}$$

$$\mathbf{L}\widetilde{\mathbf{w}}^{k+1} = \mathbf{g}, \tag{2.12}$$

where

$$[\mathbf{A}]_{i,j} := \mathcal{A}(B_{j,\boldsymbol{p}}, u_h^k, w_h^k; B_{i,\boldsymbol{p}}), \quad [\mathbf{L}]_{i,j} := (B_{j,\boldsymbol{p}}; B_{i,\boldsymbol{p}}),$$

$$[\mathbf{f}]_i := \mathcal{F}(B_{i,\boldsymbol{p}}) \quad \text{and} \quad [\mathbf{g}]_i := \mathcal{G}(u_h^k; B_{i,\boldsymbol{p}}).$$

Exploiting the tensor-product structure of the isogeometric space $\mathcal{X}_h$ defined in (3), we have

$$\mathbf{A} := C_m \, \mathbf{W}_t \otimes \mathbf{M}_s + D \, \mathbf{M}_t \otimes \mathbf{K}_s + \mathbf{M}_{\mathrm{R}} + C_m \sum_{r=1}^{R} [\mathbf{R}]_{r,r} \sum_{k=1}^{p} \mathbf{S}^t_{\mathrm{SU}r,k} \otimes \mathbf{S}^s_{\mathrm{SU}r,k},$$

$$\mathbf{L} := (\mathbf{W}_t + b \, d_e \mathbf{M}_t) \otimes \mathbf{M}_s,$$

where, using the definition (2.10), for $i, j = 1, \ldots, N_{\mathrm{dof}}$, we have

$$[\mathbf{M}_{\mathrm{R}}]_{i,j} = \int_0^T \int_\Omega C_r(u_h^k, w_h^k) \; B_{i,\boldsymbol{p}}(\boldsymbol{x}, t) \; B_{j,\boldsymbol{p}}(\boldsymbol{x}, t) \; \mathrm{d}\Omega \, \mathrm{d}t,$$

while, for $i, j = 1, \ldots, N_t$,

$$[\mathbf{S}^t_{\mathrm{SU}r,k}]_{i,j} = \int_0^T \tau_k(t) \, \theta_{t,r}(t) \, \partial_t^k b_{i,p_t}(t) \, \partial_t^k b_{j,p_t} \; \mathrm{d}t,$$

$$[\mathbf{W}_t]_{i,j} = \int_0^T b'_{j,p_t}(t) \, b_{i,p_t}(t) \, \mathrm{d}t \quad \text{and} \quad [\mathbf{M}_t]_{i,j} = \int_0^T b_{i,p_t}(t) \, b_{j,p_t}(t) \, \mathrm{d}t,$$

and, for $i, j = 1, \ldots, N_s$,

$$[\mathbf{S}^s_{\mathrm{SU}r,k}]_{i,j} = \int_\Omega \theta_{s,r}(\boldsymbol{x})\, B_{i,\boldsymbol{p}_s}(\boldsymbol{x}) B_{j,\boldsymbol{p}_s}(\boldsymbol{x}) \mathrm{d}\Omega,$$

$$[\mathbf{K}_s]_{i,j} = \int_\Omega \nabla B_{i,\boldsymbol{p}_s}(\boldsymbol{x}) \cdot \nabla B_{j,\boldsymbol{p}_s}(\boldsymbol{x}) \quad \text{and} \quad [\mathbf{M}_s]_{i,j} = \int_\Omega B_{i,\boldsymbol{p}_s}(\boldsymbol{x})\; B_{j,\boldsymbol{p}_s}(\boldsymbol{x})\; \mathrm{d}\Omega.$$

2.1.2 Preconditioner

Preconditioners are crucial for improving the efficiency of iterative algorithms used to solve linear systems. They modify the original system into an equivalent form that is easier to solve, thereby speeding up convergence for iterative methods such as the conjugate gradient (CG) and generalized minimal residual (GMRES).

A good preconditioner significantly reduces the number of iterations required by iterative methods, enhancing the conditioning properties of the system. Indeed, an effective preconditioner approximates the inverse of the original matrix, creating a more balanced and manageable system, which makes it possible to solve large-scale problems.

In this section we present the preconditioner applied to our numerical simulations on 3D space domains, presented in Section 2.2.3.

We consider for the linear system (2.11) the following preconditioner:

$$[\widehat{\mathbf{A}}]_{i,j} = \widehat{\mathcal{A}}(\widehat{B}_{j,\boldsymbol{p}}, \widehat{B}_{i,\boldsymbol{p}}),$$

where

$$\widehat{\mathcal{A}}(u; v) := \int_0^T \int_{\widehat{\Omega}} (C_m\, \partial_t u\, v + D\, \nabla u \cdot \nabla v + a\, c_1\, u\, v)\; \mathrm{d}\Omega\, \mathrm{d}t.$$

We have that

$$\widehat{\mathbf{A}} = C_m\, \mathbf{W}_t \otimes \widehat{\mathbf{M}}_s + \mathbf{M}_t \otimes (D\, \widehat{\mathbf{K}}_s + a\, c_1\, \widehat{\mathbf{M}}_s), \tag{2.13}$$

where, for $i, j = 1, \ldots, N_s$,

$$[\widehat{\mathbf{K}}_s]_{i,j} = \int_{\widehat{\Omega}} \nabla \widehat{B}_{i,\boldsymbol{p}_s}(\boldsymbol{x}) \cdot \nabla \widehat{B}_{j,\boldsymbol{p}_s}(\boldsymbol{x}) \quad \text{and} \quad [\widehat{\mathbf{M}}_s]_{i,j} = \int_{\widehat{\Omega}} \widehat{B}_{i,\boldsymbol{p}_s}(\boldsymbol{x})\; \widehat{B}_{j,\boldsymbol{p}_s}(\boldsymbol{x})\; \mathrm{d}\widehat{\Omega}$$

are the stiffness and mass matrices in the parametric domain, respectively. It also

holds

$$\widehat{\mathbf{K}}_s = \sum_{i=1}^{d} \widehat{M}_d \otimes \ldots \widehat{M}_{i+1} \otimes \widehat{K}_i \otimes \widehat{M}_i \otimes \cdots \otimes \widehat{M}_1 \quad \text{and} \quad \widehat{\mathbf{M}}_s = \widehat{M}_d \otimes \cdots \otimes \widehat{M}_1,$$

where $\widehat{K}_l$ and $\widehat{M}_l$ are the univariate stiffness and mass matrices, respectively. To apply the preconditioner (2.13), we generalize the technique proposed in [33]. Thus, following [33], we consider the generalized eigendecomposition of the pencils $(\widehat{\mathbf{K}}_l, \widehat{\mathbf{M}}_l)$ for $l = 1, \ldots, d$, which gives the matrices $\mathbf{U}_l$ for $l = 1, \ldots, d$ for $l = 1, \ldots, d$, that contain in each column the $\widehat{\mathbf{M}}_l$-orthonormal generalized eigenvectors and $\mathbf{\Lambda}_l$ that are diagonal matrices whose entries contain the corresponding generalized eigenvalues. Moreover, we define

$$[\mathbf{w}]_i = [\mathbf{W}_t]_{i,N_t} \quad \text{and} \quad [\mathbf{m}]_i = [\mathbf{M}_t]_{i,N_t} \quad \text{for} \quad i = 1, \ldots, N_t - 1,$$

$$[\overset{\circ}{\mathbf{W}}_t]_{i,j} = [\mathbf{W}_t]_{i,j} \quad \text{and} \quad [\overset{\circ}{\mathbf{M}}_t]_{i,j} = [\mathbf{M}_t]_{i,j} \quad \text{for} \quad i, j = 1, \ldots, N_t - 1,$$

and we consider the matrices $\overset{\circ}{\mathbf{U}}_t$ and $\mathbf{\Lambda}_t$, that are the matrix whose columns contain the $\overset{\circ}{\mathbf{M}}_t$-orthogonal generalized eigenvectors of the pencil $(\overset{\circ}{\mathbf{W}}_t, \overset{\circ}{\mathbf{M}}_t)$ and the matrix of the corresponding eigenvalues, respectively. We then define the matrix $\mathbf{U}_t$ as:

$$\mathbf{U}_t := \begin{bmatrix} \overset{\circ}{\mathbf{U}}_t & \mathbf{t} \\ \mathbf{0}^T & \rho \end{bmatrix},$$

where $\mathbf{0} \in \mathbb{R}^{N_t - 1}$ denotes the null vector, while

$$\begin{bmatrix} \mathbf{t} \\ \rho \end{bmatrix} := \frac{\begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}}{\left([\mathbf{v}^* \ 1] \mathbf{M}_t \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} \right)^{\frac{1}{2}}},$$

and $\mathbf{v} \in \mathbb{C}^{N_t - 1}$ such that:

$$\overset{\circ}{\mathbf{M}}_t \mathbf{v} = -\mathbf{m}.$$

Finally, we set $\mathbf{\Delta}_t := \mathbf{U}_t^* \mathbf{W}_t \mathbf{U}_t$. The matrix $\mathbf{\Delta}_t$ has an arrowhead structure (we refer to [33] for details).

Then, factorizing the common terms, we get that $\widehat{\mathbf{A}}$ can be written as:

$$\widehat{\mathbf{A}} = \left(\mathbf{U}_t^* \otimes \mathbf{U}_s^T\right)^{-1} \left(C_m\, \boldsymbol{\Delta}_t \otimes \mathbb{I}_{N_s} + D\, \mathbb{I}_{N_t} \otimes \boldsymbol{\Lambda}_s + a\, c_1\, \mathbb{I}_{N_t} \otimes \mathbb{I}_{N_s}\right) \left(\mathbf{U}_t \otimes \mathbf{U}_s\right)^{-1}, \tag{2.14}$$

where $\boldsymbol{\Lambda}_s := \sum_{l=1}^{d} \mathbb{I}_{n_d} \otimes \cdots \otimes \mathbb{I}_{n_{l+1}} \otimes \boldsymbol{\Lambda}_l \otimes \mathbb{I}_{n_{l-1}} \otimes \cdots \otimes \mathbb{I}_{n_1}$, $\mathbf{U}_s := \mathbf{U}_d \otimes \cdots \otimes \mathbf{U}_1$, and $\mathbb{I}_n$ denotes the identity of dimension $n \times n$. Note that the second term of (2.14) exhibits a block-arrowhead structure, allowing for a straightforward and computationally efficient LU decomposition (i.e., a method of factorizing a matrix into a product of a lower triangular matrix and an upper triangular matrix)

$$C_m\, \boldsymbol{\Delta}_t \otimes \mathbb{I}_{N_s} + D\, \mathbb{I}_{N_t} \otimes \boldsymbol{\Lambda}_s + a\, c_1\, \mathbb{I}_{N_t} \otimes \mathbb{I}_{N_s} = \begin{bmatrix} \mathbf{H}_1 & & & \mathbf{B}_1 \\ & \ddots & & \vdots \\ & & \mathbf{H}_{N_t-1} & \mathbf{B}_{N_t-1} \\ -\mathbf{B}_1^* & \dots & -\mathbf{B}_{N_t-1}^* & \mathbf{H}_{N_t} \end{bmatrix},$$

where $\mathbf{H}_i$ and $\mathbf{B}_i$ are diagonal matrices defined as:

$$\mathbf{H}_i := C_m\, [\boldsymbol{\Lambda}_t]_{i,i} \mathbb{I}_{N_s} + D\, \boldsymbol{\Lambda}_s + a\, c_1\, \mathbb{I}_{N_s} \quad \text{and} \quad \mathbf{B}_i := C_m\, [\mathbf{g}]_i \mathbb{I}_{N_s} \quad \text{for} \quad i = 1, \dots, N_t - 1,$$

$$\mathbf{H}_{N_t} := C_m\, \sigma \mathbb{I}_{N_s} + D\, \boldsymbol{\Lambda}_s + a\, c_1\, \mathbb{I}_{N_s},$$

while $\mathbf{g} := \overset{\circ}{\mathbf{U}}_t^* \begin{bmatrix} \overset{\circ}{\mathbf{W}}_t\ \mathbf{w} \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \rho \end{bmatrix}$ and $\sigma := [\mathbf{t}^* \rho^*]\, \mathbf{W}_t \begin{bmatrix} \mathbf{t} \\ \rho \end{bmatrix}$. This property is used in the application of the preconditioner (2.14), that we perform with [33, Algorithm 1 Extended FD]. Additionally, parallelization presents a further opportunity for enhancing computational efficiency, but it is not considered in the present work.

We now focus on the solution of (2.12). First, we note that, exploiting the properties of the Kronecker product, the solution can be obtained as:

$$\mathbf{w} = \mathrm{vec}\left(\mathbf{M}_s^{-1} \mathbf{G} \left(\mathbf{W}_t + b\, d_e\, \mathbf{M}_t\right)^{-\top}\right),$$

where the "vec" operator applied to a matrix stacks its columns into a vector, and $\mathbf{G}$ is the $N_s \times N_t$ matrix such that $\mathbf{g} = \mathrm{vec}(\mathbf{G})$. Therefore, $\mathbf{w}$ can be computed by

solving N_s independent systems associated with the $N_t \times N_t$ matrix $(\mathbf{W}_t + b\, d_e\, \mathbf{M}_t)^\top$ and N_t independent systems associated with the $N_s \times N_s$ spatial mass matrix $\mathbf{M}_s$, for which we can efficiently leverage the preconditioner proposed in [34].

2.2 Numerical Results

All numerical tests are performed using MATLAB with GeoPDEs [55] and Tensorlab [49] on Intel(R) Xeon(R) Gold 6130 CPU processor, running at 2.10 GHz and with 128 GB of RAM.

Following [47], we set the dimensionless parameters $a = 0.13$, $b = 0.013$, $c_1 = 0.26$, $c_2 = 0.1$ and $d_e = 1$.

Just for the sake of simplicity, in the following, we consider uniform meshes for each parametric direction and splines of the same polynomial degree in all parametric directions for space and time. Specifically, we set $p_1 = \cdots = p_d = p_t := p$, with global spline continuity C^{p-1}.

Numerical tests with different degrees, regularities (in space) and non-uniform meshes yield results analogous to those reported below.

For the low-rank approximation (2.7), we use `svdsketch` MATLAB function with $\varepsilon = 10^{-1}$ and the fixed-point scheme, presented in Section 2.1.1, is used to solve nonlinearities in the equations, with the stopping criterion $\left\|u_h^{k+1} - u_h^k\right\|_\infty \leq \delta$, where $\delta = 10^{-4}$.

In the numerical experiments, $\chi_{[\overline{\psi}_1,\overline{\psi}_2]}(\psi)$ refers to the characteristic function, defined as:

$$\chi_{[\overline{\psi}_1,\overline{\psi}_2]}(\psi) := \begin{cases} 1 & \text{for } \psi \in [\overline{\psi}_1, \overline{\psi}_2], \\ 0 & \text{otherwise,} \end{cases}$$

and the coordinates x, y and z refer to the spatial Cartesian coordinates, while t is the time coordinate. Finally, with η_1, η_2, η_3 we denote the coordinates in the parametric spatial domain $\widehat{\Omega}$.

2.2.1 Test with smooth solution

We deal with a smooth solution on all the space–time domain $\Omega \times (0, T)$, with $\Omega = (0, 1)$ and $T = 1$, to analyze the convergence order of the SU method. We set

the dimensionless parameters $C_m = 1$ and $D = 10^{-4}$.

In this test with smooth solution, we neglect, in the monodomain equation coupled with the Rogers–McCulloch ionic model, the action of the recovery variable, i.e., we set a zero value for w, and, regarding the unknownwn u, we do not consider the relaxation presented in Section 2.1.1.

Moreover, we compute the function f in order to have the exact solution

$$u_{\text{ex}} = 10\,\sin(\pi x)\sin(\pi t)(1-\exp(-x))(1-\exp(x-1))(1-\exp(t-1)).$$

The resulting linear systems are solved by direct solver provided by MATLAB (backslash operator "\").

Figure 2.1 shows the log-log error plot in L^2-norm for the SU method with uniform meshes in space and time, i.e., $\widehat{h}_1 = \widehat{h}_t =: h$ and spline degrees $p = 2, 3$. The slopes of the lines, representing the error trends with mesh refinement, which align $p+1$ slopes, highlight the optimal convergence order. Indeed, the presence of θ in (2.4) deactivates the stabilization terms if the residual is low: since no sharp layers are present in the domain, the stabilization is correctly not activated.

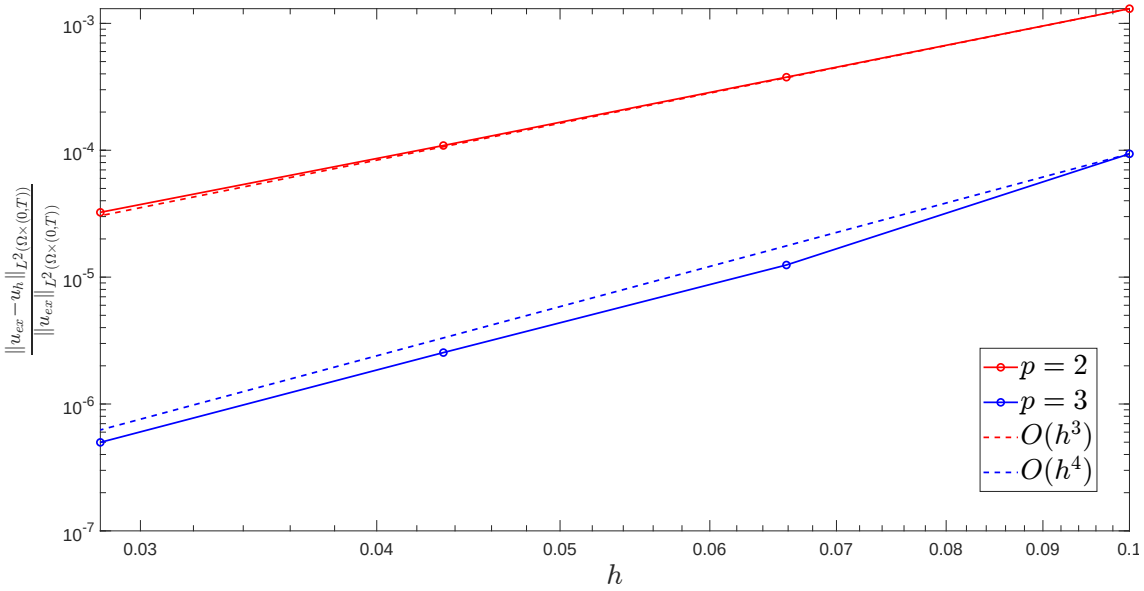

Figure 2.1: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU relative error plots in L^2-norm (test in Section 2.2.1, smooth solution).

2.2.2 2D spatial domains

In this section we want to analyze the numerical behavior of the stabilized formulation, for the monodomain equation coupled with the Rogers–McCulloch ionic model, on different 2D spatial domains Ω. The interval $(0, T)$, with $T = 300$, is the time domain. In particular, our aim is to analyze the presence of spurious oscillations caused by concentrated source terms and to evaluate how the proposed stabilization method enhances the solutions in terms of numerical stability.

In these tests we consider $\widehat{h}_1 = 2^{-6}$, $\widehat{h}_2 = 2^{-3}$, $\widehat{h}_t = 2^{-5}$ and $p = 3$, while we set the dimensionless parameters $C_m = 1$ and $D = 10^{-4}$.

The resulting linear systems are solved by direct solver provided by MATLAB (backslash operator "\").

A rectangle

For this test we consider a rectangular spatial domain $\Omega = (0, L_1) \times (0, L_2)$ with $L_1 = 1.5$, $L_2 = 0.25$. The concentrated source term f is as follows:

$$f(x, y, t) = \frac{1}{4} \exp\left(-\left(5 \cdot 10^2 \left(\left(y - \frac{L_2}{2}\right)^2 + \left(x - \frac{8L_1}{5T}t\right)^2\right)\right)\right) \chi_{[90,100]}(t).$$

Figures 2.2 and 2.3 show the numerical solutions of the plain Galerkin method (i.e., with the stabilization term set to zero) and the SU method, respectively, plotted for a fixed value of y, specifically $y = 0.125$.

In detail, in the numerical solutions of the standard Galerkin method (Figure 2.2), for $t < 90$, spurious oscillations propagating backward in time are generated due to the activation of the concentrated source.

On the other hand, with the SU method (Figure 2.3), these significant spurious oscillations are eliminated throughout almost the entire domain, and no numerical instabilities propagate backward in time. Indeed, for $t < 90$, the numerical solution is consistent with the expected zero transmembrane potential. The function $\theta(x, y, t)$, shown in Figure 2.4, regulates the activation of stabilization in the SU method, which is active only in the vicinity of the layer.

The numerical solutions at specific fixed times (Figure 2.5) show that, beginning with zero potential, the stimulus generated by the concentrated impulse gradually

spreads across the spatial domain.

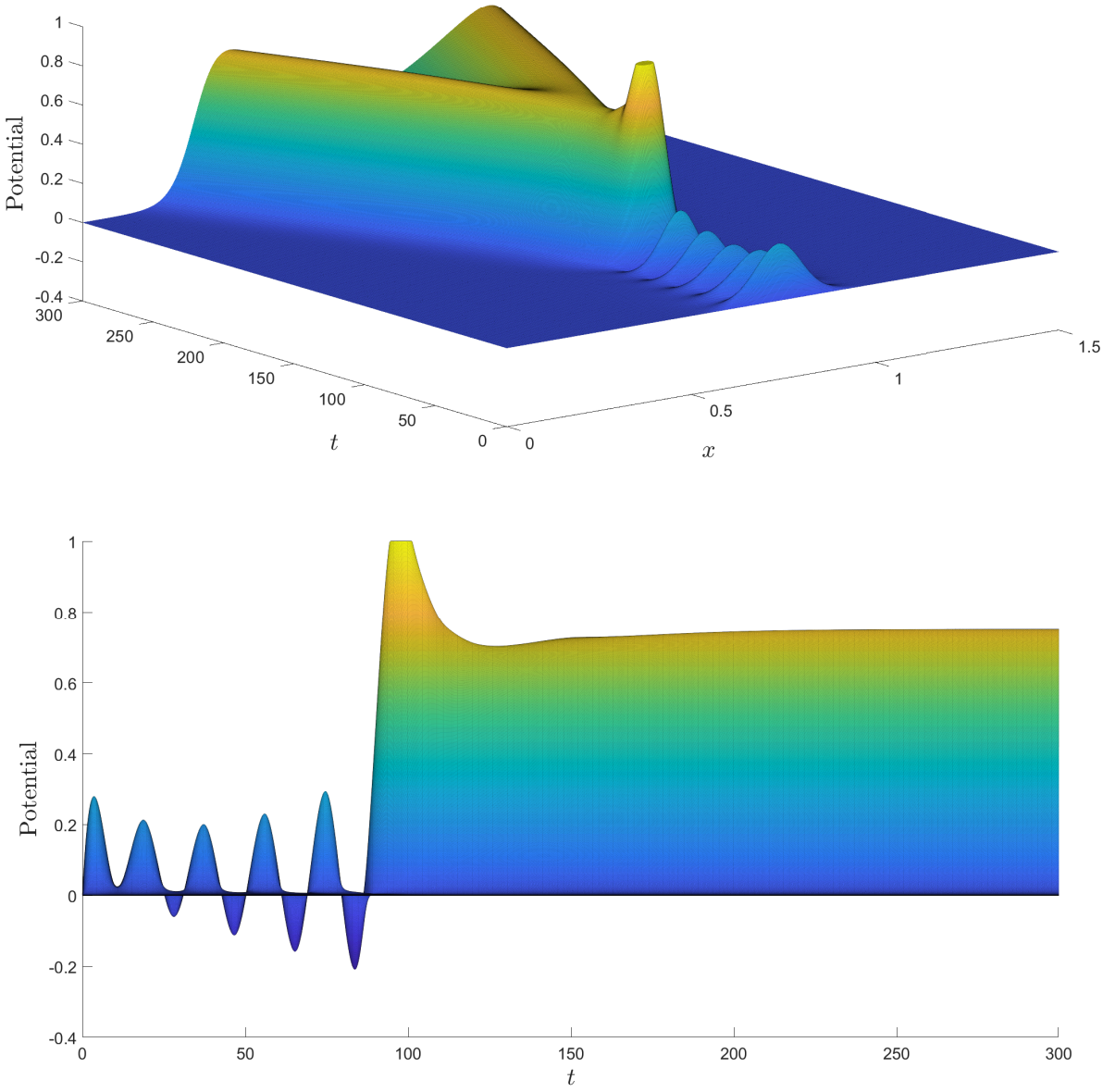

Figure 2.2: Monodomain equation coupled with the Rogers–McCulloch ionic model, plain Galerkin solution for $y = 0.125$ (test in Section 2.2.2, Ω is a rectangle).

Finally, Table 2.1 summarizes the computational cost: the total time needed to solve the problem numerically of the SU method is very close to the time of the plain Galerkin method, with a similar number of fixed-point iterations to achieve the set convergence.

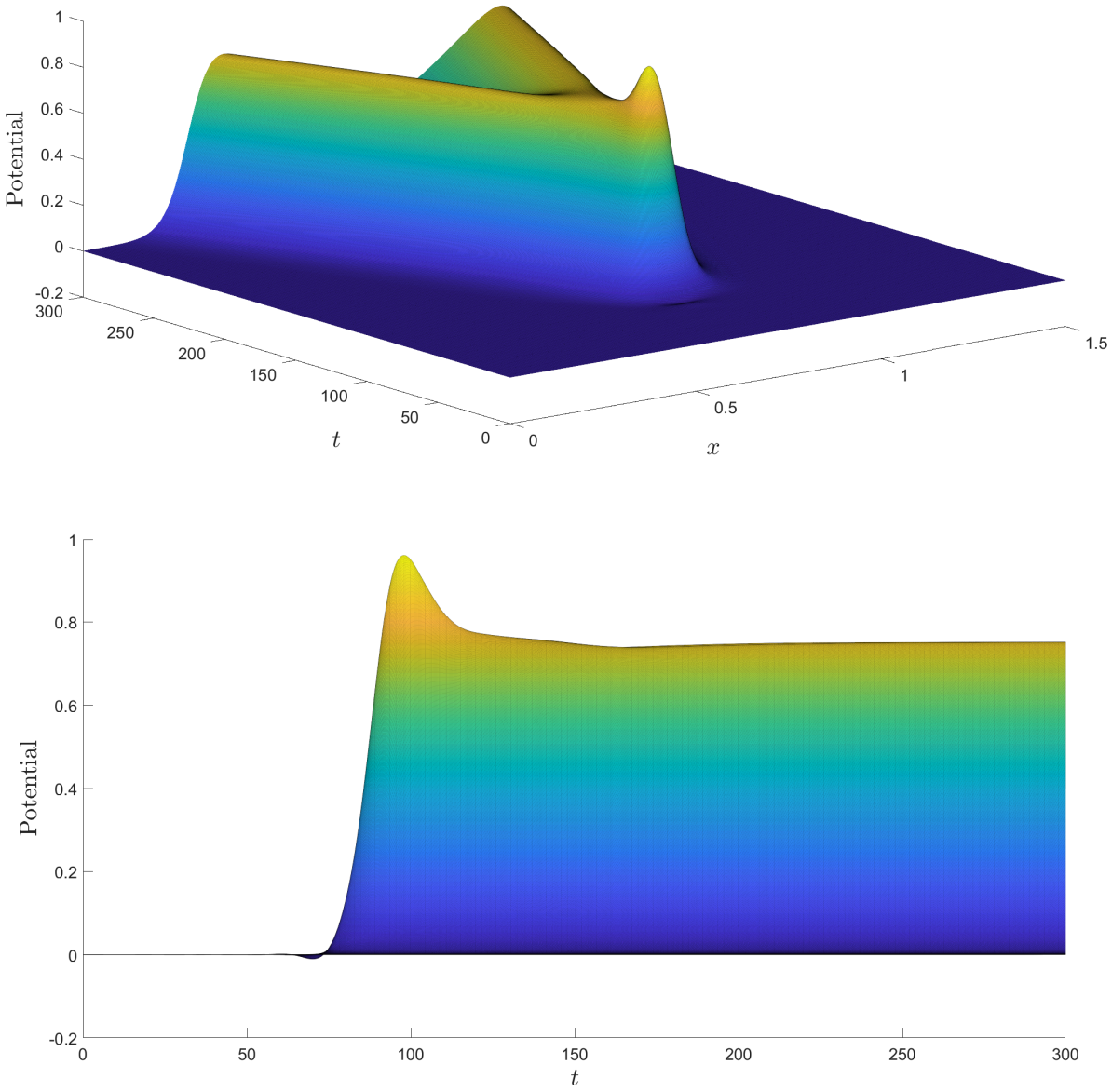

Figure 2.3: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU solution for $y = 0.125$ (test in Section 2.2.2, Ω is a rectangle).

Method	Fixed-point iterations	Time (s)
Plain Galerkin method	33	$2.4 \cdot 10^3$
SU method	36	$2.8 \cdot 10^3$

Table 2.1: Monodomain equation coupled with the Rogers–McCulloch ionic model, computational cost comparison (test in Section 2.2.2, Ω is a rectangle).

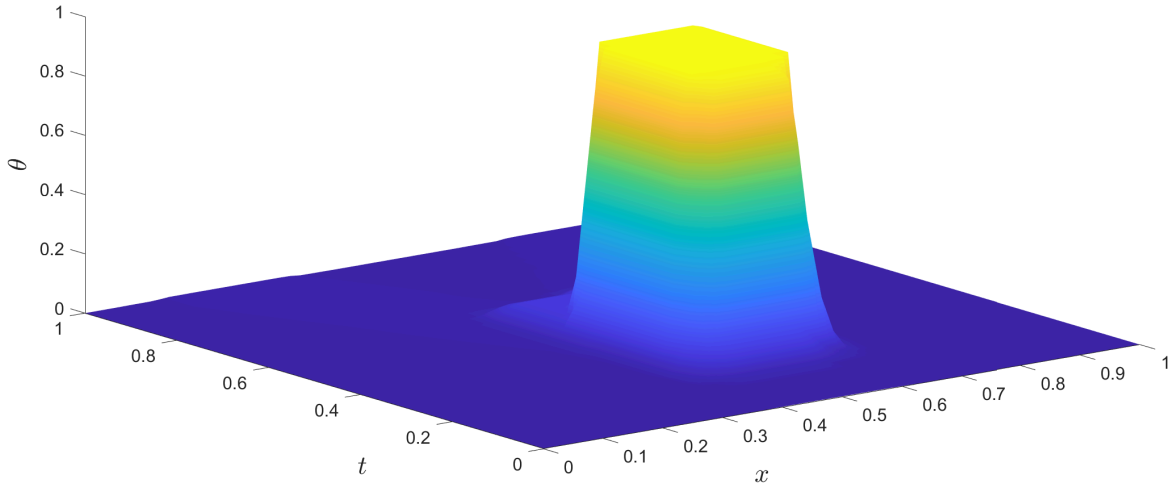

Figure 2.4: Monodomain equation coupled with the Rogers–McCulloch ionic model, function $\theta(x, y, t)$ for $y = 0.125$, in the SU method (test in Section 2.2.2, Ω is a rectangle).

An ellipse with hole

We use an ellipse (semi-major axis: $a_o = 0.75$; semi-minor axis: $b_o = 0.125$) with an elliptic hole (semi-major axis: $a_i = 0.375$; semi-minor axis: $b_i = 0.0625$) as spatial domain Ω (Figure 2.6). The source term f is as follows:

$$f(x, y, t) = \frac{1}{4} \exp\left(-\left(5 \cdot 10^2 \left(\left(y - \frac{L_2}{2}\right)^2 + \left(x - \frac{8L_1}{15T}t\right)^2\right)\right)\right) \chi_{[90,100]}(t).$$

Figures 2.7 and 2.8 present the numerical solutions along the section line s (depicted in Figure 2.6), for the plain Galerkin method and the SU method, respectively. The function $\theta(x, y, t)$, which regulates the activation of high-order stabilization terms in the SU method, is shown in Figure 2.9.

The plain Galerkin method (Figure 2.7) exhibits numerical instabilities that propagate backward in time (i.e., for $t < 90$).

On the other hand, the SU method (Figure 2.8) reveals negligible spurious oscillations throughout the domain. Notably, the numerical instabilities that propagate backward in time, observed in the plain Galerkin method, are absent.

The numerical solutions for specific fixed times (Figure 2.10) indicate that, starting

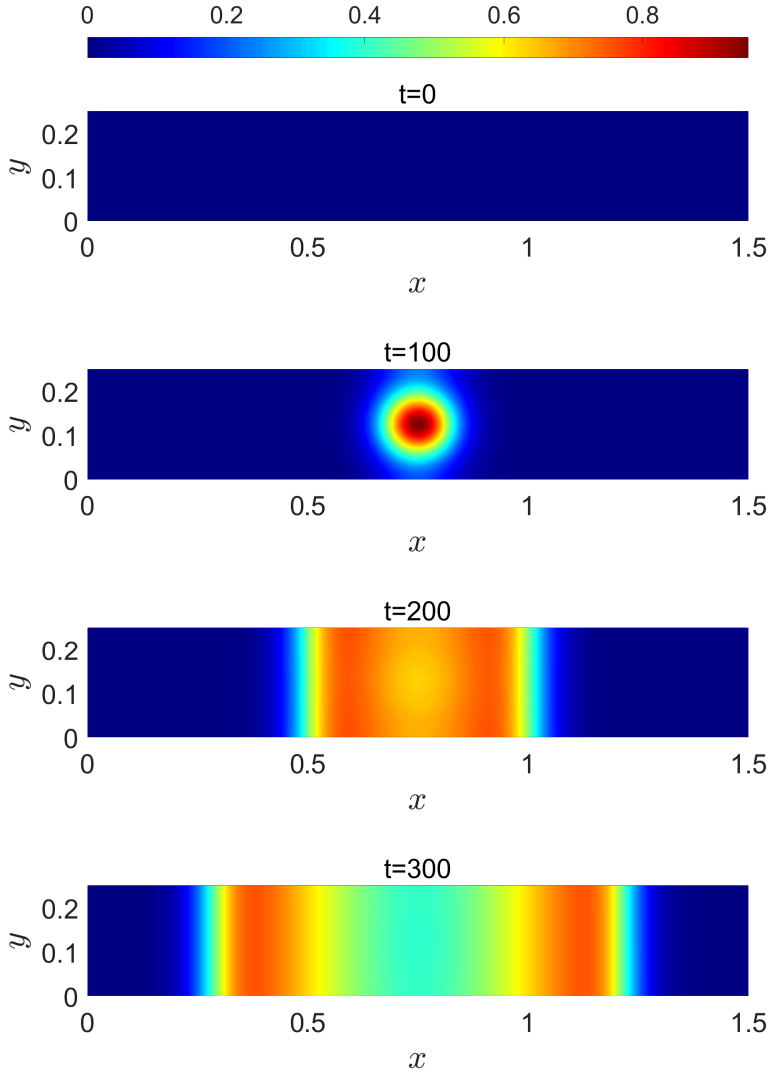

Figure 2.5: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU solutions for various fixed times (test in Section 2.2.2, Ω is a rectangle), the colors represent dimensionless transmembrane potential.

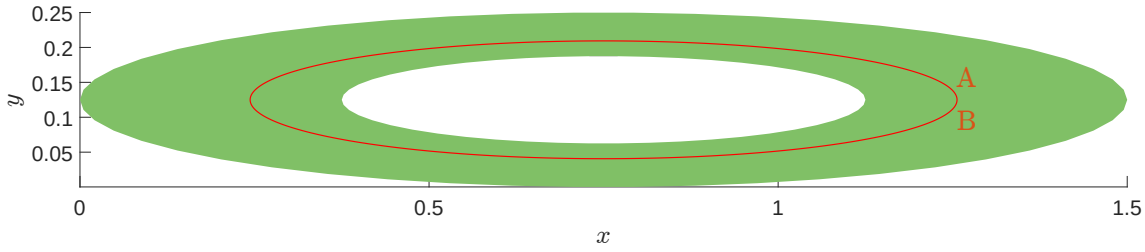

Figure 2.6: Spatial domain Ω, an ellipse with hole, with section line s (A–B), for the second test in Section 2.2.2.

from zero potential, the stimulus generated by a concentrated impulse gradually propagates through the spatial domain.

The total time needed to solve the problem numerically of the plain Galerkin method is considerably higher than the time of the stabilized method (Table 2.2). This interesting result is due to the significant numerical instabilities that affect the number of iterations required by the plain Galerkin method to reach the set convergence in the fixed-point scheme, as highlighted in Table 2.2.

Method	Fixed-point iterations	Time (s)
Plain Galerkin method	145	$1.0 \cdot 10^4$
SU method	31	$2.7 \cdot 10^3$

Table 2.2: Monodomain equation coupled with the Rogers–McCulloch ionic model, computational cost comparison (test in Section 2.2.2, Ω is an ellipse with hole).

2.2.3 3D spatial domains

In this section we deal with different 3D spatial domains Ω, while the interval $(0, T)$, is the time domain. Indeed, we aim to analyze the numerical behavior of the proposed stabilization method with a concentrated source term using these 3D spatial domains. In these tests we consider $p = 2$.

The linear systems (2.11) are solved using the generalized minimal residual method (GMRES) preconditioned as presented in Section 2.1.2. Concerning (2.12), we exploit the technique presented Section 2.1.2 and we solve N_s independent systems associated with the $N_t \times N_t$ matrix with direct solver provided by MATLAB (backslash operator "\"), while N_t independent systems associated with the $N_s \times N_s$ mass matrix are solved by using the preconditioned conjugate gradient method (PCG).

A parallelepiped

For this test we use a parallelepiped as spatial domain $\Omega = (0, L_1) \times (0, L_2) \times (0, L_3)$, with $L_1 = 0.5$, $L_2 = 0.5$, $L_3 = 0.3$ and $\widehat{h}_1 = 2^{-4}$, $\widehat{h}_2 = 2^{-4}$, $\widehat{h}_3 = 2^{-3}$, $\widehat{h}_t = 2^{-4}$, while $T = 150$.

We set the dimensionless parameters $C_m = 1$ and $D = 10^{-4}$.

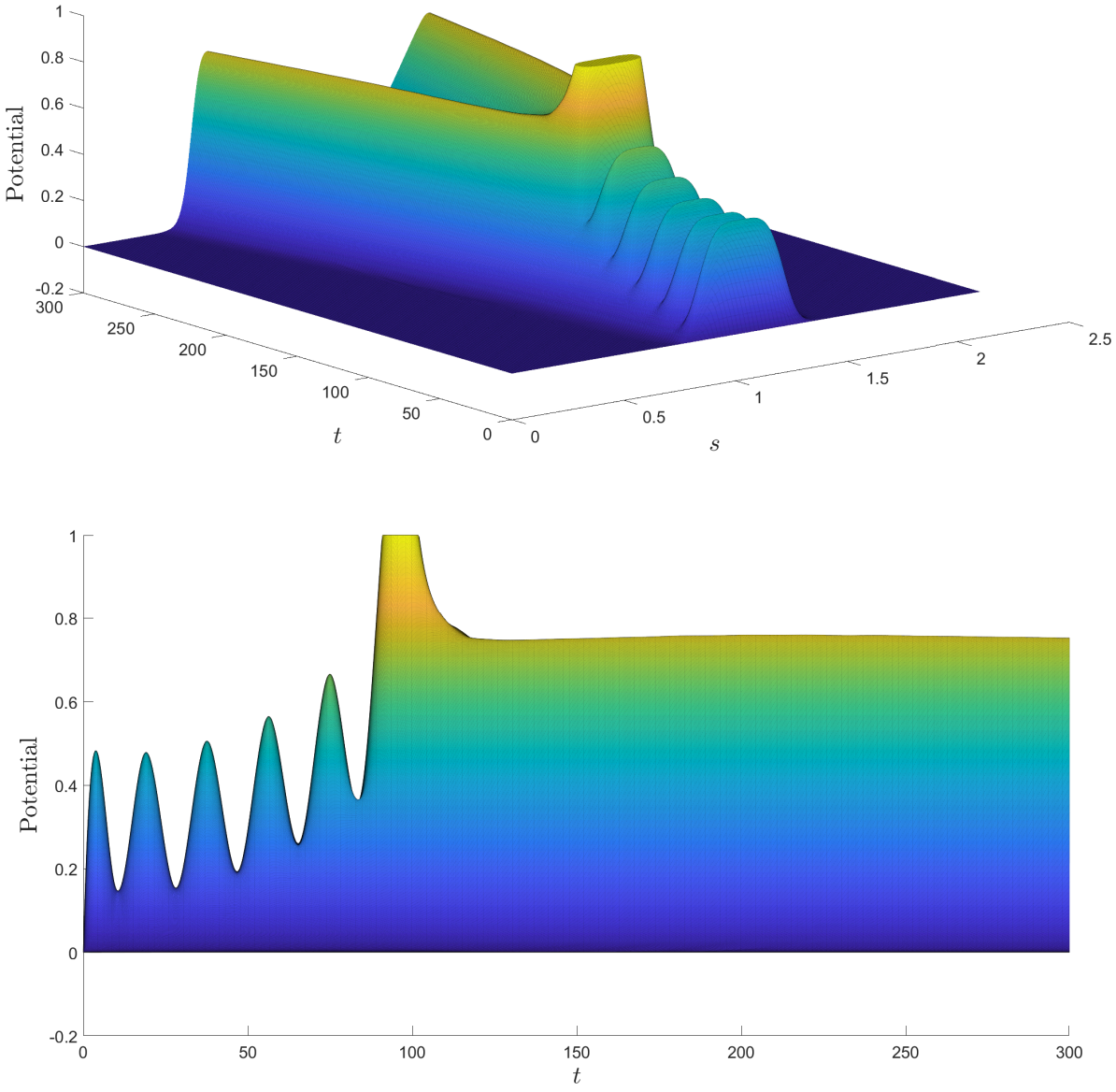

Figure 2.7: Monodomain equation coupled with the Rogers–McCulloch ionic model, plain Galerkin solution along the section line s (test in Section 2.2.2, Ω is an ellipse with hole).

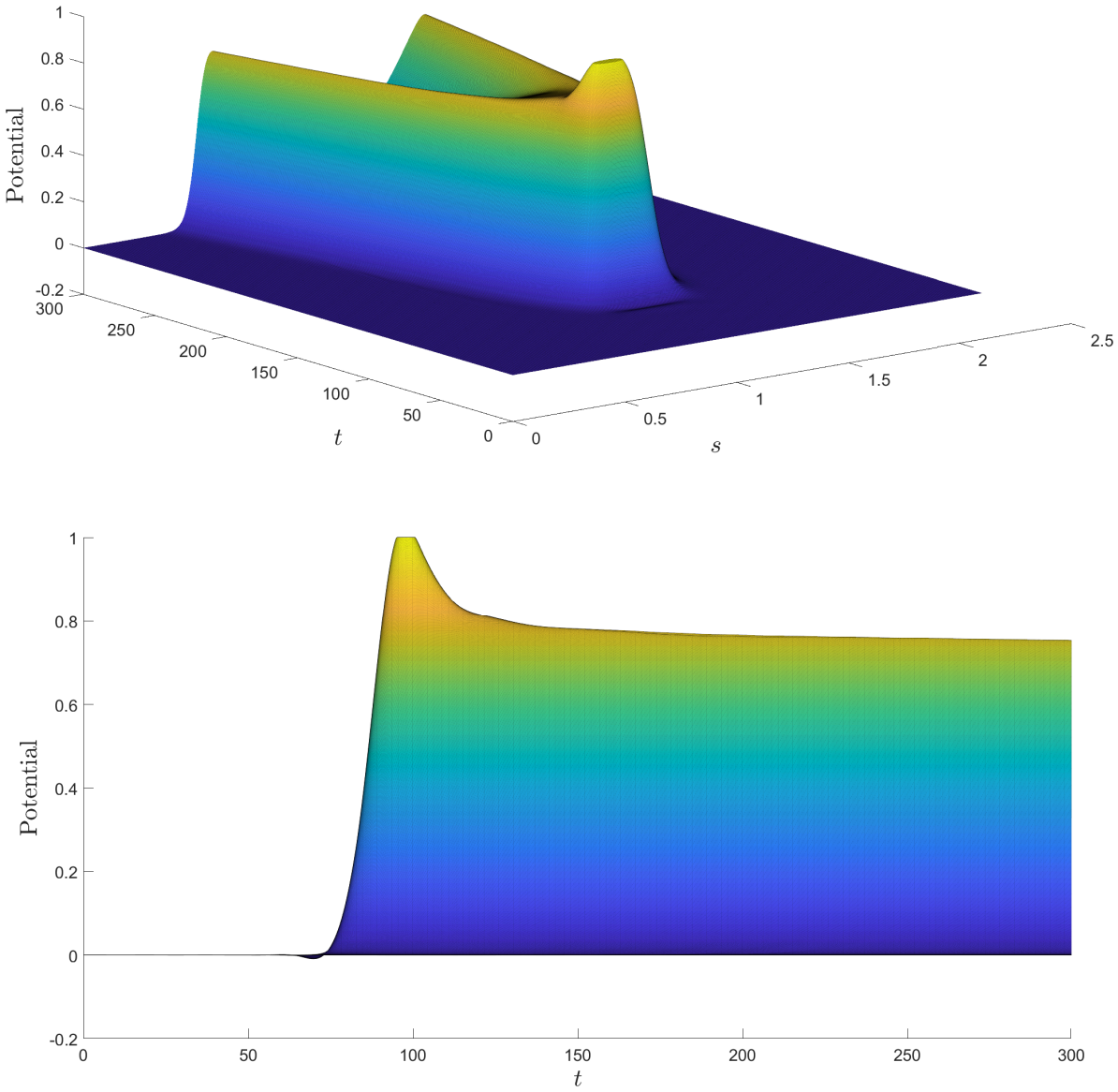

Figure 2.8: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU solution along the section line s (test in Section 2.2.2, Ω is an ellipse with hole)

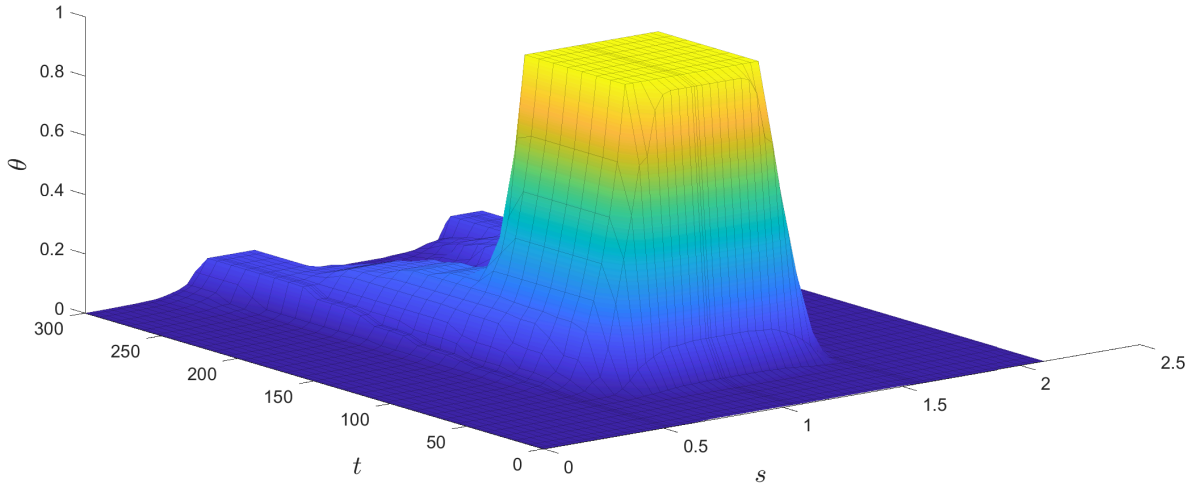

Figure 2.9: Monodomain equation coupled with the Rogers–McCulloch ionic model, function $\theta(x, y, t)$ along the section line s, in the SU method (test in Section 2.2.2, Ω is an ellipse with hole).

The concentrated source term f is as follows:

$$f(x, y, z, t) = \frac{1}{5}\chi_{[0,0.15]}(x)\ \chi_{[0,0.15]}(y)\ \chi_{[0.15,0.3]}(z)\ \ \chi_{[30.0,40.5]}(t).$$

Figures 2.11 and 2.12 show the numerical solutions of the plain Galerkin method and the SU method, respectively, for various fixed times. We observe that the plain Galerkin solutions exhibit non-physical solutions for $t < 30$, even in the absence of the activation of the source term (Figure 2.11). However, these numerical instabilities are not present in the SU method (Figure 2.12), where the solutions correctly remain zero in the spatial domain for $t < 30$, as expected.

The total computational time of the SU method is approximately 1.6 times that of the plain Galerkin method and a reasonably low number of iterations is required by the preconditioned solvers, for each fixed-point iteration (Table 2.3).

Method	GMRES, PCG iterations	Fixed-point iterations	Time (s)
Plain Galerkin method	19, 1	26	$7.4 \cdot 10^2$
SU method	33, 1	27	$1.2 \cdot 10^3$

Table 2.3: Monodomain equation coupled with the Rogers–McCulloch ionic model, computational cost comparison (test in Section 2.2.3, Ω is a parallelepiped); GMRES, PCG iterations are averaged over each fixed-point iteration.

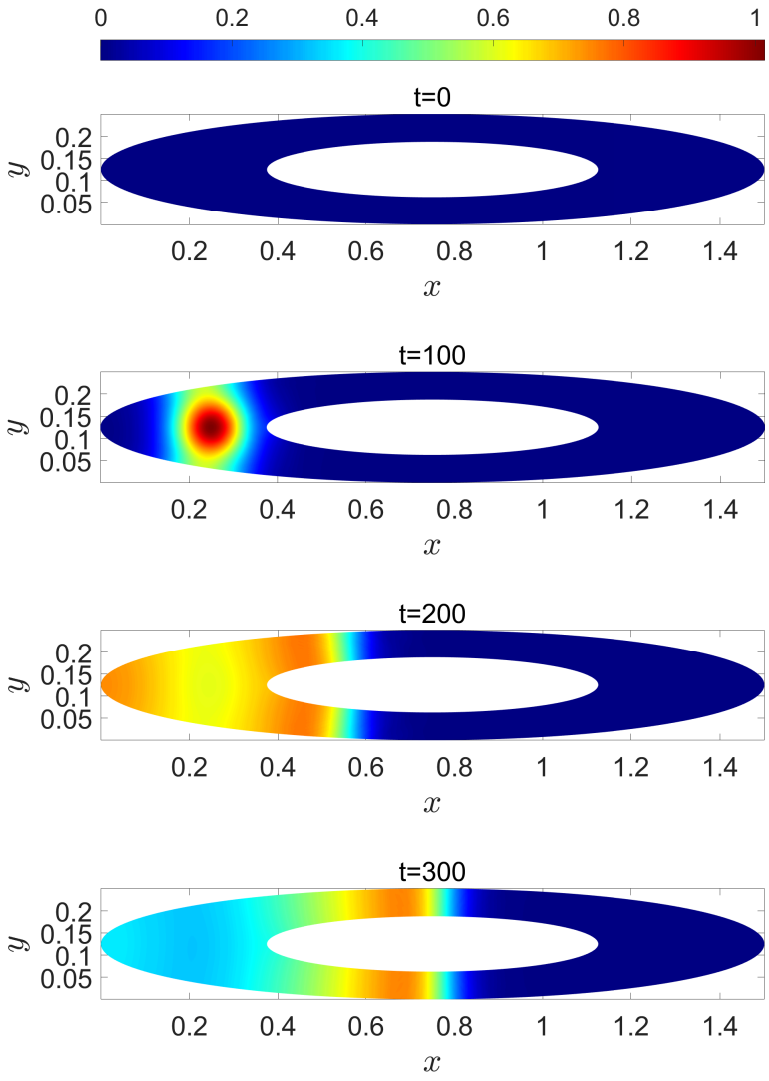

Figure 2.10: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU solutions for various fixed times (test in Section 2.2.2, Ω is an ellipse with hole), the colors represent dimensionless transmembrane potential.

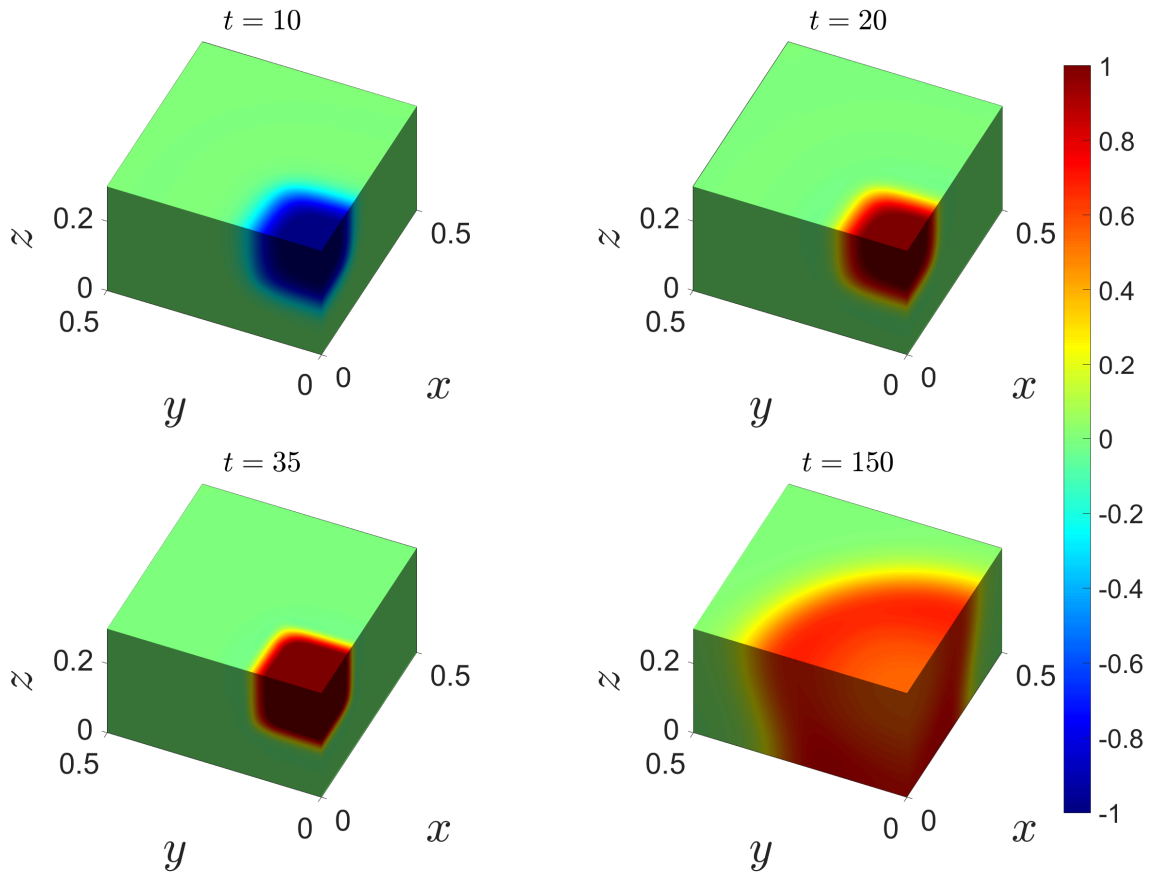

Figure 2.11: Monodomain equation coupled with the Rogers–McCulloch ionic model, plain Galerkin solutions for various fixed times (test in Section 2.2.3, Ω is a parallelepiped), the colors represent dimensionless transmembrane potential.

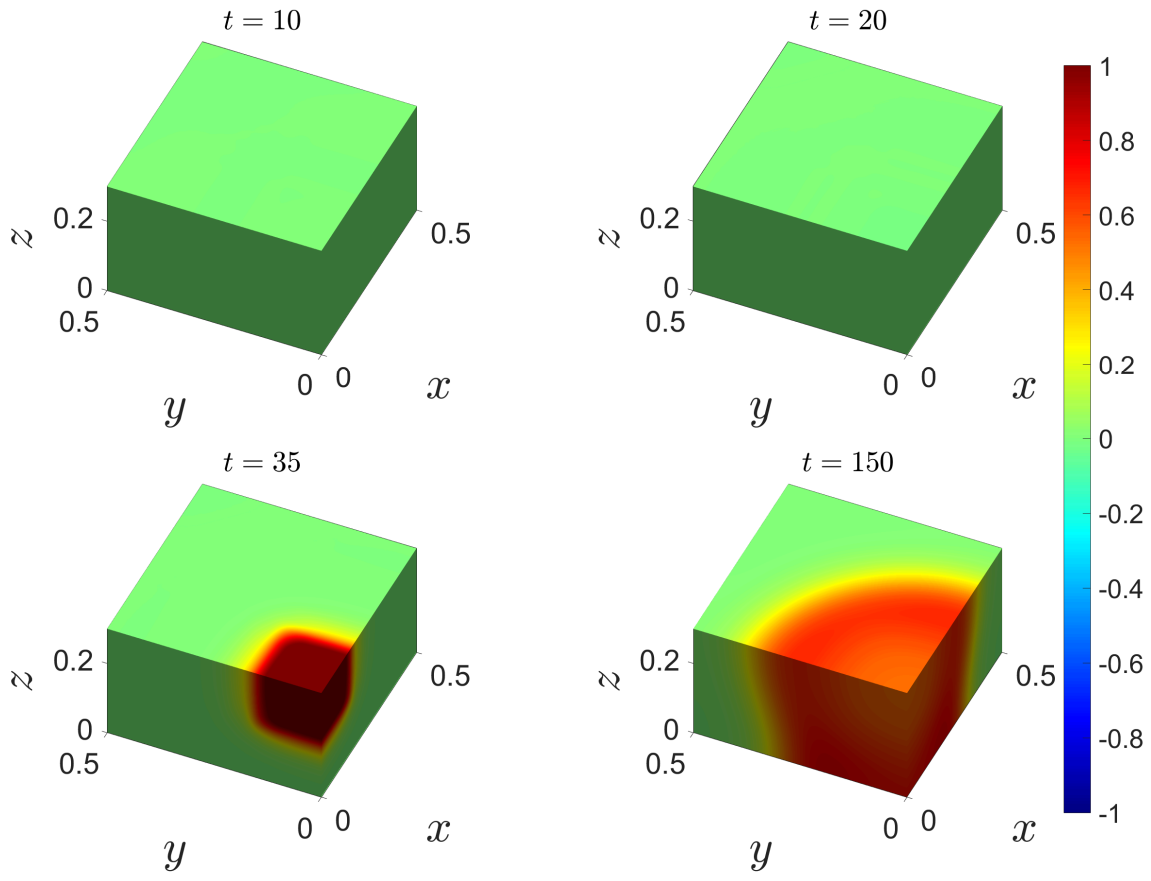

Figure 2.12: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU solutions for various fixed times (test in Section 2.2.3, Ω is a parallelepiped), the colors represent dimensionless transmembrane potential.

A curved spatial domain

For this test we use the spatial domain Ω shown in Figure 2.13, with $\widehat{h}_1 = 2^{-4}$, $\widehat{h}_2 = 2^{-3}$, $\widehat{h}_3 = 2^{-4}$ and $\widehat{h}_t = 2^{-4}$, while $T = 150$.

We set the dimensionless parameters $C_m = 1$ and $D = 10^{-4}$.

The source term f is as follows:

$$f(\eta_1, \eta_2, \eta_3, t) = \frac{1}{5}\chi_{[0,0.3]}(\eta_1)\ \chi_{[0,0.3]}(\eta_2)\ \chi_{[0.3,1]}(\eta_3)\ \ \chi_{[30.0,40.5]}(t).$$

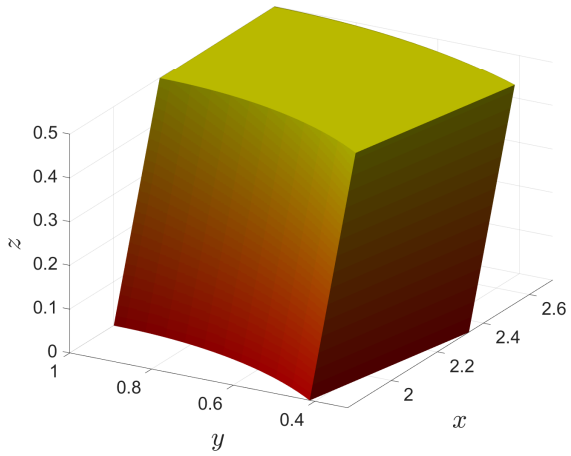

Figure 2.13: Spatial domain Ω, a curved domain, for the second test in Section 2.2.3 .

Figures 2.14 and 2.15 present the numerical solutions obtained using the plain Galerkin method and the SU method for various fixed times, respectively. Before the activation of the source term (i.e., for $t < 30$), Figure 2.14 shows the presence of numerical instabilities in the plain Galerkin solutions. These non-physical values disappear in the solutions of the SU method, as shown in Figure 2.15, where the solutions for $t < 30$ are correctly zero in the spatial domain, as expected.

Table 2.4 illustrates the total computation time needed to solve the problem numerically. The time required for the SU method is less than two and a half times that of the plain Galerkin method, and a reasonable number of iterations is needed by the preconditioned solvers for each fixed-point iteration.

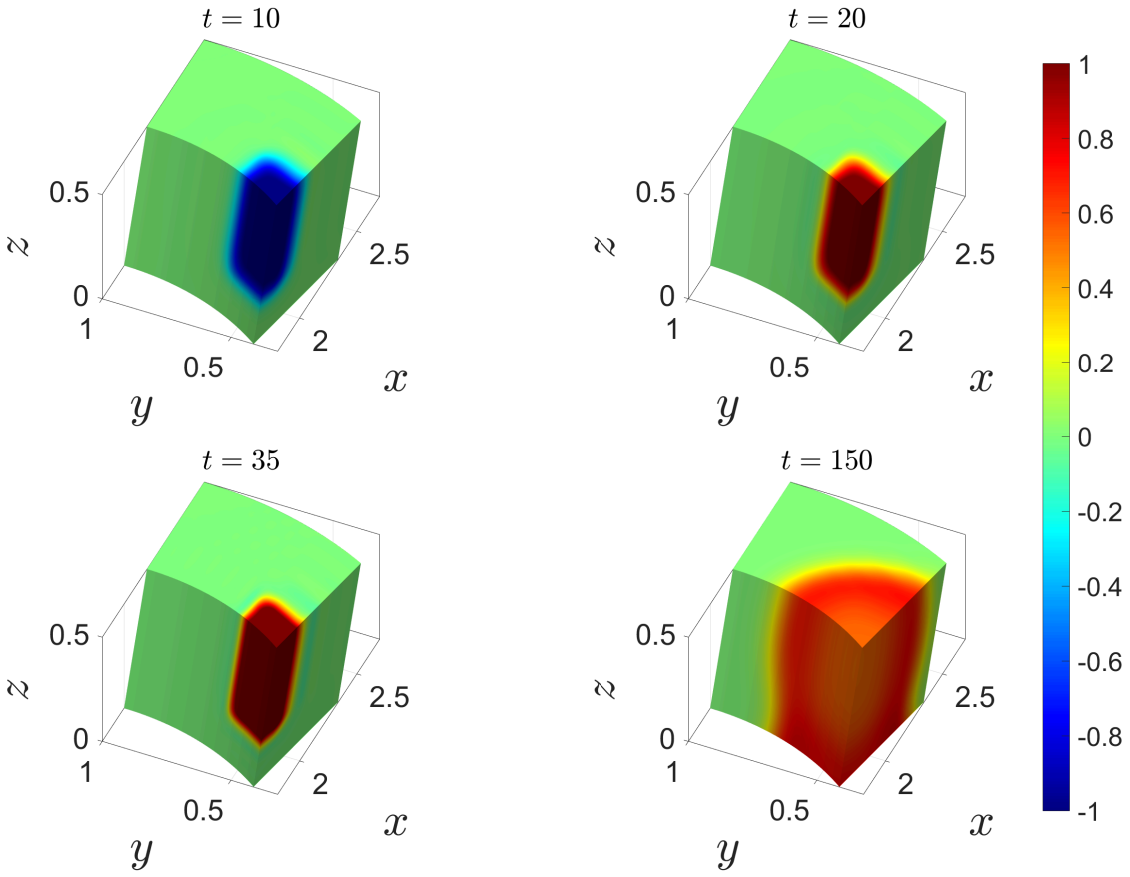

Figure 2.14: Monodomain equation coupled with the Rogers–McCulloch ionic model, plain Galerkin solutions for various fixed times (test in Section 2.2.3, Ω is a curved spatial domain), the colors represent dimensionless transmembrane potential.

Method	GMRES, PCG iterations	Fixed-point iterations	Time (s)
Plain Galerkin method	24, 3	27	$4.7 \cdot 10^2$
SU method	42, 3	34	$1.1 \cdot 10^3$

Table 2.4: Monodomain equation coupled with the Rogers–McCulloch ionic model, computational cost comparison (test in Section 2.2.3, Ω is a curved domain); GMRES, PCG iterations are averaged over each fixed-point iteration.

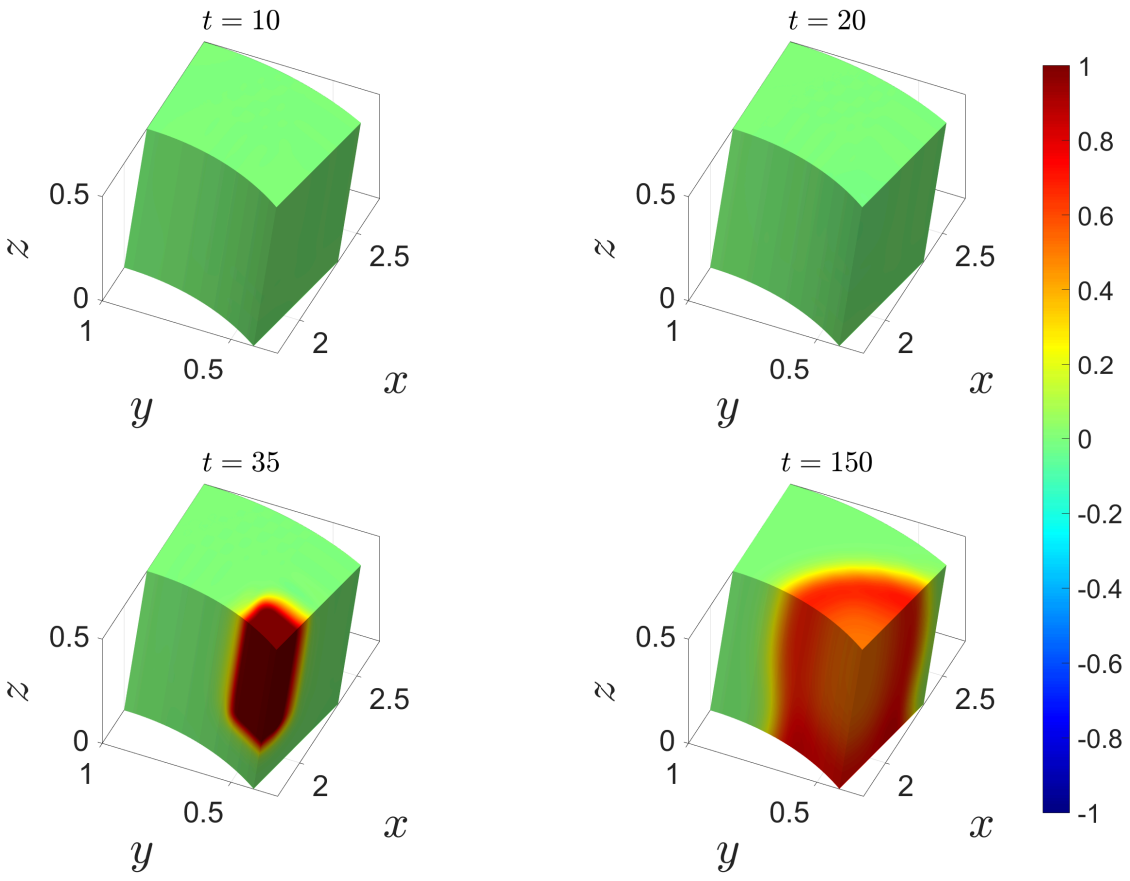

Figure 2.15: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU solutions for various fixed times (test in Section 2.2.3, Ω is a curved spatial domain), the colors represent dimensionless transmembrane potential.

Spatial domain approximating the left ventricular geometry

For this test we use the spatial domain Ω shown in Figure 2.16, with $\widehat{h}_1 = \widehat{h}_2 = \widehat{h}_3 = 2^{-4}$ and $\widehat{h}_t = 2^{-6}$, while $T = 300$.

We set the dimensionless parameters $C_m = 1$ and $D = 10^{-3}$, to simulate the propagation of the impulse under ventricular physiological conditions.

The source term f is as follows:

$$f(\eta_1, \eta_2, \eta_3, t) = \frac{1}{10}\chi_{[0.9,1]}(\eta_3)\ \chi_{[45.0,60.0]}(t).$$

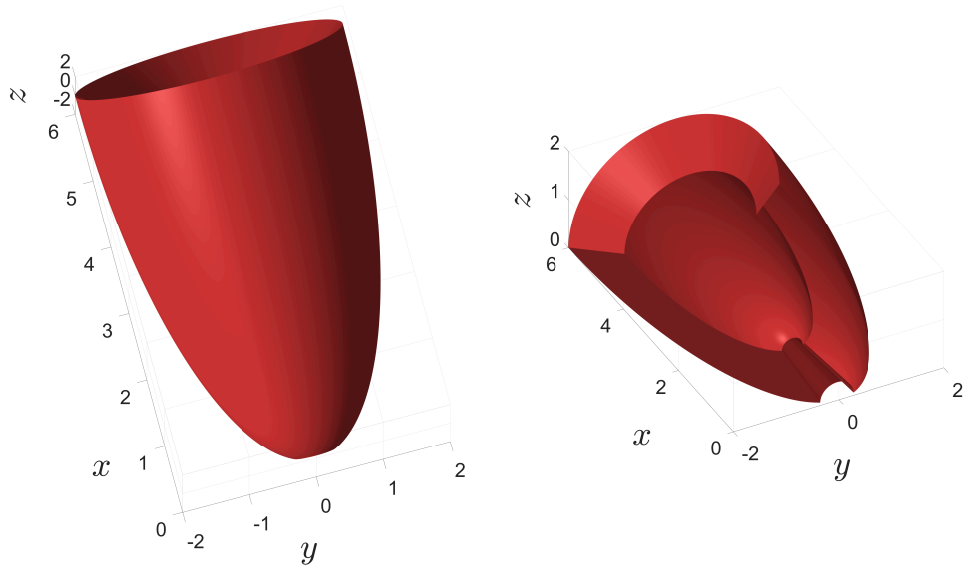

Figure 2.16: Spatial domain Ω, approximating the left ventricular geometry, for the third test in Section 2.2.3 on the left, and its corresponding cross-section for solution plots on the right.

Figures 2.17 and 2.18 compare the numerical solutions obtained with the plain Galerkin method and the SU method at various time steps. Before the activation of the source term (i.e., for $t < 45$), where zero transmembrane potential is expected, Figure 2.17 shows non-physical solutions in the plain Galerkin solutions, indicating numerical instabilities. In contrast, Figure 2.18 suggests that the SU method effectively suppresses these non-physical values, with the solution correctly remaining

zero in the spatial domain. This comparison highlights the robustness of the SU method in providing stable solutions, especially in phases where no source term is present.

Table 2.5 shows the total computation time for numerically solving the problem, including the iterations performed by the preconditioned solvers during each fixed-point iteration. In particular, the SU method requires one and a half times the total computation time of the plain Galerkin method.

Method	GMRES, PCG iterations	Fixed-point iterations	Time (s)
Plain Galerkin method	115, 4	41	$8.7 \cdot 10^3$
SU method	132, 4	39	$1.3 \cdot 10^4$

Table 2.5: Monodomain equation coupled with the Rogers–McCulloch ionic model, computational cost comparison (test in Section 2.2.3, Ω approximates the left ventricular geometry); GMRES, PCG iterations are averaged over each fixed-point iteration.

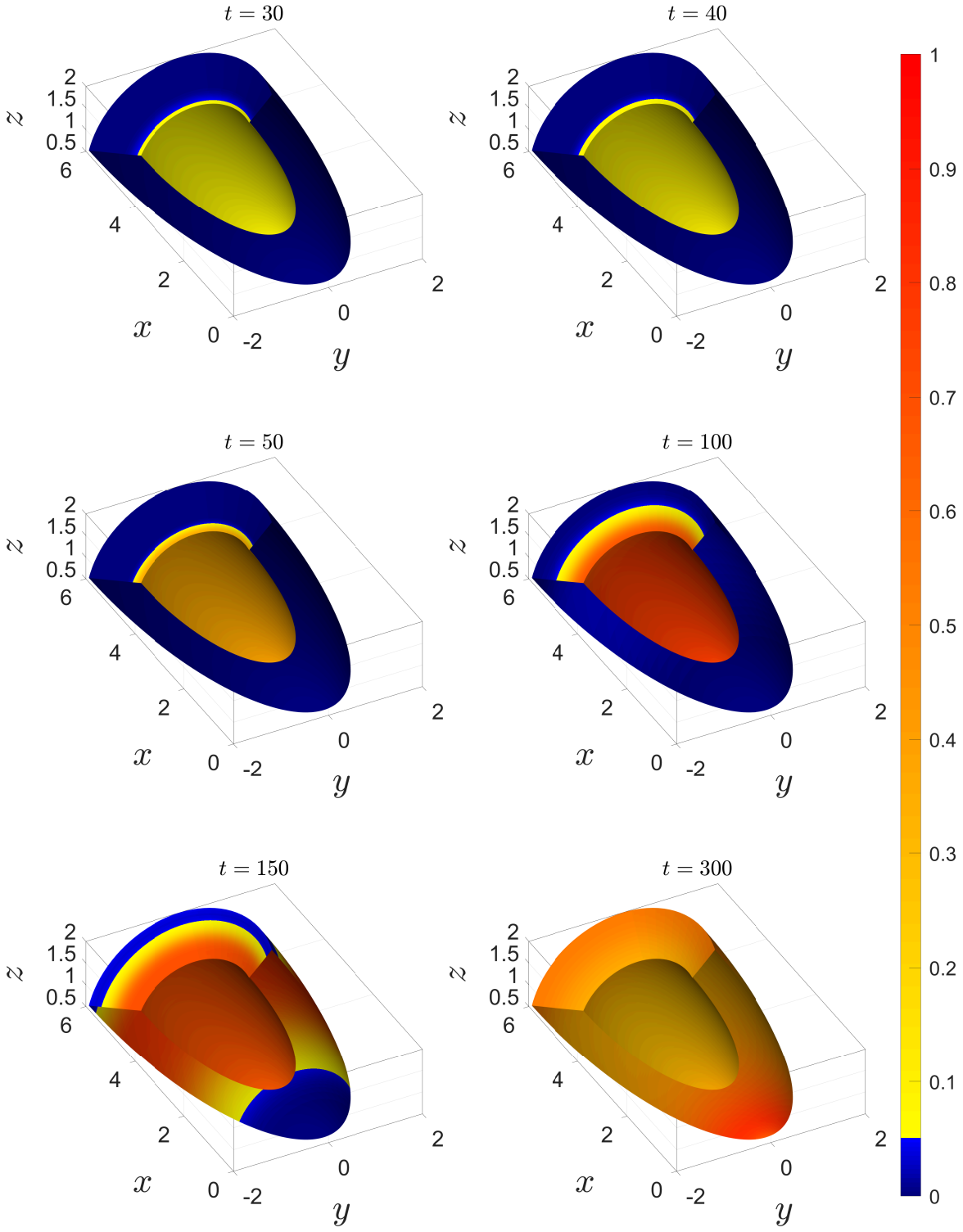

Figure 2.17: Monodomain equation coupled with the Rogers–McCulloch ionic model, plain Galerkin solutions for various fixed times (test in Section 2.2.3, Ω approximates the left ventricular geometry), the colors represent dimensionless transmembrane potential.

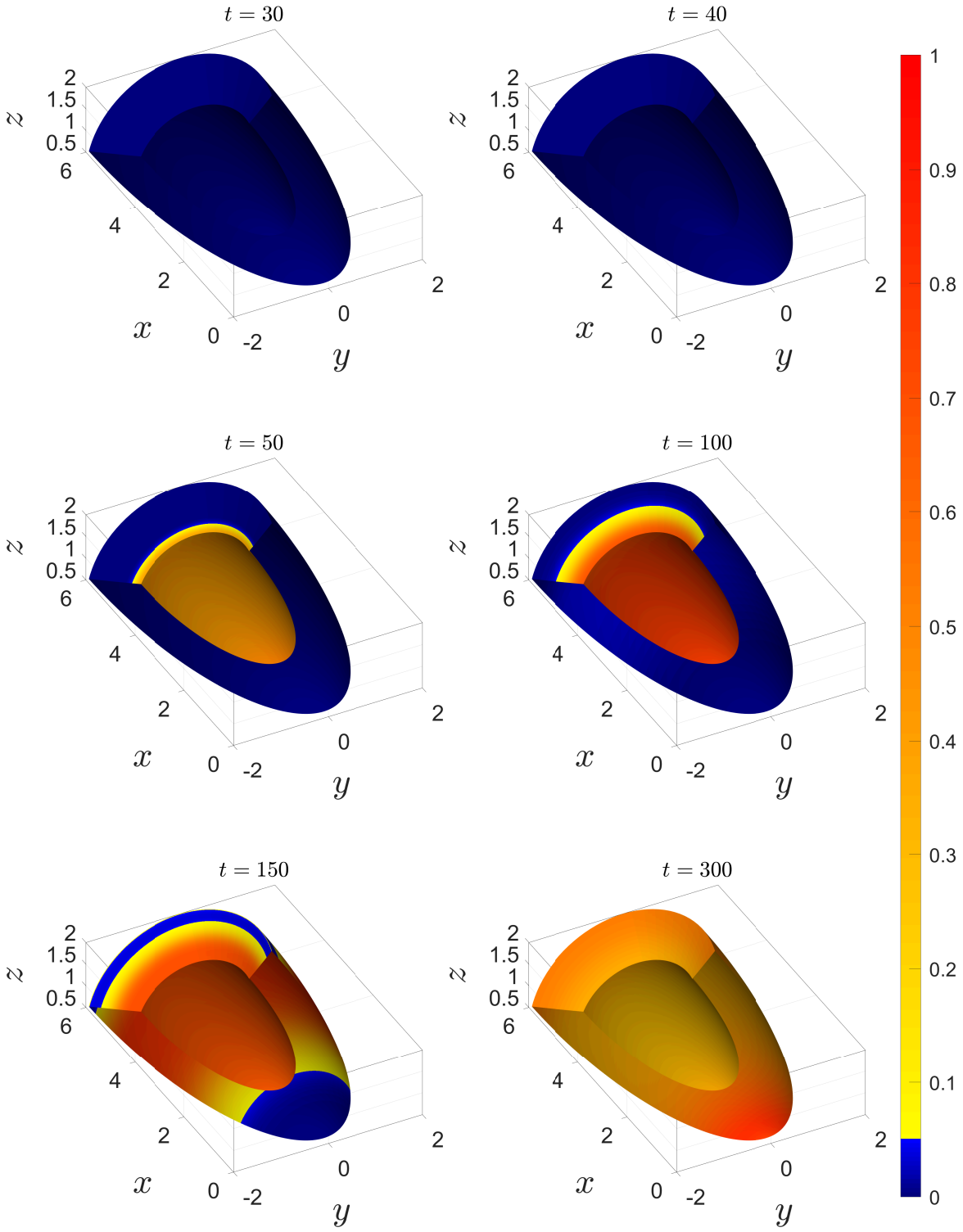

Figure 2.18: Monodomain equation coupled with the Rogers–McCulloch ionic model, SU solutions for various fixed times (test in Section 2.2.3, Ω approximates the left ventricular geometry), the colors represent dimensionless transmembrane potential.

Conclusions

In this thesis, we have first introduced a novel stable space–time method for the heat equation within the Isogeometric Analysis (IgA) framework. We named this method Spline Upwind (SU), and it is based on stabilizing terms that extend the Streamline upwind Petrov–Galerkin (SUPG) stabilization mechanism to high-degree and high-continuity splines, promoting causality with respect to time.

Additionally, we presented a modification of the SU method to make it more suitable for cardiac electrophysiology. In particular, we developed a formulation that is both simple and highly effective, designed to minimize spurious oscillations while ensuring optimal computational efficiency.

The results provide numerical evidence of optimal convergence order of the SU method, when dealing with smooth solutions.

Moreover, the behavior remains stable in the presence of sharp layers, both in one-dimensional cases and for the heat equation, as well as for the monodomain equation coupled with the Rogers–McCulloch ionic model. In particular, for the monodomain equation, we also investigated 3D spatial domains, which lead to 4D space–time domains. These problems present a challenging aspect from a computational cost perspective, particularly when dealing with the ventricular approximation. Therefore, we employed efficient solver based on preconditioner.

To complete the analysis, we quantified the increase in computational cost observed in our tests when transitioning from the standard Galerkin method to the SU method in the context of cardiac electrophysiology. From this perspective, we emphasize that the computational times of the SU method are generally close to those of the Galerkin method, making SU stabilization appealing not only from the standpoint of solution stability but also in terms of computational cost.

Finally, we acknowledge the importance of efficient and fast solvers in

the space–time framework. The higher dimensionality poses computational challenges when investigating three-dimensional spatial domains, with geometrical representations closely approximating reality, such as the entire heart. To address these challenges, it is important to employ advanced techniques, such as local mesh refinement (see, e.g., [32]), in order to reduce the number of degrees of freedom, or low-rank tensor techniques to approximate the unknown solutions (see, e.g., [39]). These research directions offer promising fields for further explorations.

Bibliography

[1] R.R. Aliev and A.V. Panfilov. A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals*, 7(3):293–301, 1996. doi:10.1016/0960-0779(95)00089-5.

[2] J.H. Argyris and D.W. Scharpf. Finite Elements in Time and Space. *The Aeronautical Journal*, 73(708):1041–1044, 1969. doi:10.1017/S0001924000051198.

[3] A. Bartezzaghi, L. Dedè, and A. Quarteroni. Isogeometric Analysis of high order Partial Differential Equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 295:446–469, 2015. doi:10.1016/j.cma.2015.07.018.

[4] Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes. YZβ discontinuity capturing for advection-dominated processes with application to arterial drug delivery. *International Journal for Numerical Methods in Fluids*, 54(6-8):593–608, 2007. doi:10.1002/fld.1484.

[5] L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287, 2014. doi:10.1017/S096249291400004X.

[6] M. Bendahmane, R. Bürger, and R. Ruiz-Baier. A multiresolution space-time adaptive scheme for the bidomain model in electrocardiology. *Numerical Methods for Partial Differential Equations*, 26(6):1377–1404, 2010. doi:10.1002/num.20495.

[7] A. Bressan and E. Sande. Approximation in FEM, DG and IGA: a theoretical comparison. *Numerische Mathematik*, 143(4):923–942, 2019. doi:10.1007/s00211-019-01063-5.

[8] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, 1982. doi:10.1016/0045-7825(82)90071-8.

[9] J.C. Bruch Jr and G. Zyvoloski. Transient two-dimensional heat conduction problems solved by the finite element method. *International Journal for Numerical Methods in Engineering*, 8(3):481–494, 1974. doi:10.1002/nme.1620080304.

[10] M. Bucelli, A. Zingaro, P.C. Africa, I. Fumagalli, L. Dedè, and A. Quarteroni. A mathematical model that integrates cardiac electrophysiology, mechanics, and fluid dynamics: Application to the human left heart. *International Journal for Numerical Methods in Biomedical Engineering*, 39(3):e3678, 2023. doi:doi.org/10.1002/cnm.3678.

[11] C.D. Cantwell, S. Yakovlev, R.M. Kirby, N.S. Peters, and S.J. Sherwin. High-order spectral/hp element discretisation for reaction–diffusion problems on surfaces: Application to cardiac electrophysiology. *Journal of Computational Physics*, 257:813–829, 2014. doi:10.1016/j.jcp.2013.10.019.

[12] D. Chapelle, A. Collin, and J.F. Gerbeau. A surface-based electrophysiology model relying on asymptotic analysis and motivated by cardiac atria modeling. *Mathematical Models and Methods in Applied Sciences*, 23(14):2749–2776, 2013. doi:10.1142/S0218202513500450.

[13] P. Colli Franzone, P. Deuflhard, B. Erdmann, J. Lang, and L.F. Pavarino. Adaptivity in Space and Time for Reaction-Diffusion Systems in Electrocardiology. *SIAM Journal on Scientific Computing*, 28(3):942–962, 2006. doi:10.1137/050634785.

[14] P. Colli Franzone, L.F. Pavarino, and S. Scacchi. Mathematical and numerical methods for reaction-diffusion models in electrocardiology. In D. Ambrosi, A. Quarteroni, and G. Rozza, editors, *Modeling of Physiological Flows*, pages 107–141, Milano, 2012. Springer Milan. doi:10.1007/978-88-470-1935-5_5.

[15] P. Colli Franzone, L.F. Pavarino, and S. Scacchi. *Mathematical cardiac electrophysiology*, volume 13. Springer, 2014. doi:10.1007/978-3-319-04801-7.

[16] A. Collin, J.F. Gerbeau, M. Hocini, M. Haïssaguerre, and D. Chapelle. Surface-Based Electrophysiology Modeling and Assessment of Physiological Simulations in Atria. In S. Ourselin, D. Rueckert, and N. Smith, editors, *Functional Imaging and Modeling of the Heart*, pages 352–359, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-38899-6_42.

[17] M.A. Colman, S.J. Castro, E.A. Perez Alday, J.C. Hancox, C. Garratt, and H. Zhang. Recent progress in multi-scale models of the human atria. *Drug Discovery Today: Disease Models*, 14:23–32, 2014. doi:10.1016/j.ddmod.2014.04.003.

[18] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[19] M. Courtemanche, R.J. Ramirez, and S. Nattel. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *American Journal of Physiology-Heart and Circulatory Physiology*, 275(1):H301–H321, 1998. doi:10.1152/ajpheart.1998.275.1.H301.

[20] L. Dedè and A. Quarteroni. Isogeometric Analysis for second order Partial Differential Equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 284:807–834, 2015. doi:10.1016/j.cma.2014.11.008.

[21] J.A. Evans, Y. Bazilevs, I. Babuška, and T.J.R. Hughes. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometic finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198:1726–1741, 2009. doi:10.1016/j.cma.2009.01.021.

[22] I. Fried. Finite-element analysis of time-dependent phenomena. *AIAA Journal*, 7(6):1170–1173, 1969. doi: 10.2514/3.5299.

[23] M.J. Gander. 50 years of time parallel time integration. In T. Carraro, M. Geiger, S. Körkel, and R Rannacher, editors, *Multiple Shooting and Time Domain*

Decomposition Methods, pages 69–113, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-23321-5.

[24] J.E. Hall and M.E. Hall. *Guyton and Hall Textbook of Medical Physiology, 14th Edition*. Elsevier, 2021.

[25] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005. doi:10.1016/j.cma.2004.10.008.

[26] T.J.R. Hughes and G.M.H Hulbert. Space-time finite element methods for elastodynamics: Formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66(3):339–363, 1988. doi:10.1016/0045-7825(88)90006-0.

[27] J. Keener and J. Sneyd. Calcium Dynamics. In J. Keener and J. Sneyd, editors, *Mathematical Physiology: I: Cellular Physiology*, pages 273–346. Springer New York, 2009. doi:10.1007/978-0-387-75847-3_7.

[28] P. Kopp, V. Calo, E. Rank, and S. Kollmannsberger. Space-time hp-finite elements for heat evolution in laser powder bed fusion additive manufacturing. *Engineering with Computers*, 38(6):4879–4893, 2022. doi:10.1007/s00366-022-01719-1.

[29] S. Krishnamoorthi, M. Sarkar, and W.S. Klug. Numerical quadrature and operator splitting in finite element methods for cardiac electrophysiology. *International Journal for Numerical Methods in Biomedical Engineering*, 29(11):1243–1266, 2013. doi:10.1002/cnm.2573.

[30] U. Langer, S.E. Moore, and M. Neumüller. Space-time isogeometric analysis of parabolic evolution problems. *Computer Methods in Applied Mechanics and Engineering*, 306:342 – 363, 2016. doi:10.1016/j.cma.2016.03.042.

[31] U. Langer and O. Steinbach. *Space-Time Methods: Applications to Partial Differential Equations*, volume 25. Walter de Gruyter GmbH & Co KG, 2019. doi:10.1515/9783110548488.

[32] U. Langer, O. Steinbach, F. Troltzsch, and H. Yang. Unstructured space-time finite element methods for optimal control of parabolic equations. *SIAM Journal on Scientific Computing*, 43(2):A744–A771, 2021. doi:10.1137/20M1330452.

[33] G. Loli, M. Montardini, G. Sangalli, and M. Tani. An efficient solver for space-time isogeometric Galerkin methods for parabolic problems. *Computers and Mathematics with Applications*, 80(11):2586–2603, 2020. doi:10.1016/j.camwa.2020.09.014.

[34] G. Loli, G. Sangalli, and M. Tani. Easy and efficient preconditioning of the isogeometric mass matrix. *Computers & Mathematics with Applications*, 116:245–264, 2022. doi:10.1016/j.camwa.2020.12.009.

[35] G. Loli, G. Sangalli, and P. Tesini. High-order spline upwind for space–time Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 417(11):116408, 2023. doi:10.1016/j.cma.2023.116408.

[36] A. Mantzaflaris, B. Jüttler, B.N. Khoromskij, and U. Langer. Low rank tensor methods in Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:1062–1085, 2017. doi:10.1016/j.cma.2016.11.013.

[37] C.C. Mitchell and D.G. Schaeffer. A two-current model for the dynamics of cardiac membrane. *Bulletin of Mathematical Biology*, 65(5):767–793, September 2003. doi:10.1016/S0092-8240(03)00041-7.

[38] M. Montardini, M. Negri, G. Sangalli, and M. Tani. Space-time least-squares isogeometric method and efficient solver for parabolic problems. *Mathematics of Computation*, 89(323):1193–1227, 2020. doi:10.1090/mcom/3471.

[39] M. Montardini, G. Sangalli, and M. Tani. A low-rank isogeometric solver based on Tucker tensors. *Computer Methods in Applied Mechanics and Engineering*, 417:116472, 2023. doi:10.1016/j.cma.2023.116472.

[40] H. Nguyen and J. Reynen. A space-time least-square finite element scheme for advection-diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 42(3):331–342, 1984. doi:10.1016/0045-7825(84)90012-4.

[41] A. Nygren, C. Fiset, L. Firek, J.W. Clark, D.S. Lindblad, R.B. Clark, and W.R. Giles. Mathematical Model of an Adult Human Atrial Cell. *Circulation Research*, 82(1):63–81, 1998. doi:10.1161/01.RES.82.1.63.

[42] A.S. Patelli, L. Dedè, T. Lassila, A. Bartezzaghi, and A. Quarteroni. Isogeometric approximation of cardiac electrophysiology models on surfaces: An accuracy study with application to the human left atrium. *Computer Methods in Applied Mechanics and Engineering*, 317:248–273, 2017. doi:10.1016/j.cma.2016.12.022.

[43] L. Pegolotti, L. Dedè, and A. Quarteroni. Isogeometric Analysis of the electrophysiology in the human heart: Numerical simulation of the bidomain equations on the atria. *Computer Methods in Applied Mechanics and Engineering*, 343:52–73, 2019. doi:10.1016/j.cma.2018.08.032.

[44] G. Plank, L. Zhou, J.L. Greenstein, S. Cortassa, R.L. Winslow, B. O'Rourke, and N.A. Trayanova. From mitochondrial ion channels to arrhythmias in the heart: computational techniques to bridge the spatio-temporal scales. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1879):3381–3409, 2008. doi:10.1098/rsta.2008.0112.

[45] A. Quarteroni, L. Dedè, A. Manzoni, and C. Vergara. *Mathematical Modelling of the Human Cardiovascular System: Data, Numerical Approximation, Clinical Applications*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2019.

[46] A. Quarteroni, T. Lassila, S. Rossi, and R. Ruiz-Baier. Integrated Heart—Coupling multiscale and multiphysics models for the simulation of the cardiac function. *Computer Methods in Applied Mechanics and Engineering*, 314:345–407, 2017. doi:10.1016/j.cma.2016.05.031.

[47] J.M. Rogers and A.D. McCulloch. A collocation-Galerkin finite element model of cardiac action potential propagation. *IEEE Transactions on Biomedical Engineering*, 41(8):743–757, 1994. doi:10.1109/10.310090.

[48] C. Saadé, S. Lejeunes, D. Eyheramendy, and R. Saad. Space-Time Isogeometric Analysis for linear and non-linear elastodynamics. *Computers & Structures*, 254:106594, 2021. doi:10.1016/j.compstruc.2021.106594.

[49] L. Sorber, M. Van Barel, and L. De Lathauwer. Tensorlab v2.0. *Available online*, 2014.

[50] J. Southern, G.J. Gorman, M.D. Piggott, and P.E. Farrell. Parallel anisotropic mesh adaptivity with dynamic load balancing for cardiac electrophysiology. *Journal of Computational Science*, 3(1):8–16, 2012. doi:10.1016/j.jocs.2011.11.002.

[51] O. Steinbach. Space-Time Finite Element Methods for Parabolic Problems. *Computational Methods in Applied Mathematics*, 15(4):551–566, 2015. doi:10.1515/cmam-2015-0026.

[52] R. Stevenson and J. Westerdiep. Stability of Galerkin discretizations of a mixed space–time variational formulation of parabolic evolution equations. *IMA Journal of Numerical Analysis*, 41(1):28–47, 2021. doi:10.1093/imanum/drz069.

[53] K. Takizawa and T.E. Tezduyar. Space-time computation techniques with continuous representation in time (ST-C). *Computational Mechanics*, 53(1):91–99, 2014. doi:10.1007/s00466-013-0895-y.

[54] T.E. Tezduyar and M. Senga. Stabilization and shock-capturing parameters in SUPG formulation of compressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(13):1621–1632, 2006. doi:10.1016/j.cma.2005.05.032.

[55] R. Vázquez. A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0. *Computers & Mathematics with Applications*, 72(3):523–554, 2016. doi:10.1016/j.camwa.2016.05.010.

[56] E. Vigmond, F. Vadakkumpadan, V. Gurev, H. Arevalo, M. Deo, G. Plank, and N. Trayanova. Towards predictive modelling of the electrophysiology of the heart. *Experimental physiology*, 94(5):563–577, 2009. doi:10.1113/expphysiol.2008.044073.

[57] K. Wang, S.Y. Ho, D.G. Gibson, and R.H. Anderson. Architecture of atrial musculature in humans. *Heart*, 73(6):559–565, 1995.

[58] Y. Wenjian, G. Yu, and L. Yaohang. Efficient Randomized Algorithms for the Fixed-Precision Low-Rank Matrix Approximation. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1339–1359, 2018. doi:10.1137/17M1141977.