



Academic Year 2024–2025

University of Strasbourg

Faculty of Mathematics and Computer Science

Master's in Scientific Computing and Mathematics for Innovation

Mathematical Modeling and Simulation of Organ-on-Chip (OOC)

Author: Adama DIENG
Date: August 21, 2025

Supervisors: Christophe PRUD'HOMME
Micol PENNACCHIO
Silvia BERTOLUZZA
Gabriella BRETTI

Contents

Acknowledgments	3
Abstract	4
1 Introduction: General Context and Modeling	6
1.1 General Context	6
1.2 Vocabulary and Key Concepts	6
1.3 Objectives and Approach of the Internship	7
2 Resolution Strategy with the HDG Method	11
2.1 General Overview of the HDG Method	11
2.1.1 Context and Motivation	11
2.1.2 HDG Method for a Convection-Diffusion Problem	12
2.2 The model problem	12
2.2.1 Notations and Approximations Spaces	13
2.2.2 Characterization of the Exact Solution	14
2.2.3 Weak Formulation on Each Element	15
2.2.4 The HDG Method	16
2.2.4.1 Matrix Formulation	17
2.2.4.2 Boundary Conditions and Global Solver	20
2.2.4.3 Accuracy Analysis	21
2.2.5 HDG implementation in Feel++	21
2.2.5.1 Feel++ general description	21
3 Keller-Segel chemotaxis model	23
3.1 Presentation of the Keller-Segel Model	23
3.2 Weak Formulation Adapted to HDG	24
3.3 Remarks and properties	27
3.3.1 The lowest order case	28
4 Application to Organ-on-Chip OOC	29
4.1 General Mathematical Model of OOC	30
4.1.1 Single-Chamber Model	30

4.1.1.1	Coefficients and Modeling Choices	31
4.1.1.2	Explanation of the Model	32
4.1.2	Two-Chamber Model with Connecting channels	33
4.1.2.1	Description of the Model	34
4.1.2.2	Chemotactic Sensitivity and killing functions	35
4.2	Setup and results on the the numerical approximation of the OOC .	36
4.2.1	Numerical Setup and Results for the Single-Chamber Model .	36
4.2.1.1	Initial and Boundary Conditions	37
4.2.1.2	Quantitative Analysis and Parameter Sensitivity Study	44

Bibliographie	47
----------------------	-----------

Acknowledgments

I want to express my deep gratitude to everyone who helped make this internship a success. A special thanks to my supervisors, Christophe PRUD'HOMME, Micol PENNACCHIO, Silvia BERTOLUZZA, and Gabriella BRETTI, for their thorough guidance, wise advice, and availability throughout this project. Their expertise in mathematical modeling, numerical simulation, and especially organ-on-chip dynamics was a valuable source of inspiration, helping me overcome technical challenges and deepen my understanding of simulated biological systems.

I also thank my professors and tutors, especially Mr. JOUBINE AGHILI, for his careful supervision and constant encouragement during several projects in this master's program. Their dedication and guidance played a key role in developing my skills and keeping me motivated throughout this academic journey.

Finally, I would like to warmly thank my friends for their support and the moments of sharing that helped us stay motivated in the face of the demands of this master's program.

Abstract

This report presents the development of a numerical framework for simulating cellular dynamics in Organ-on-Chip (OOC) systems, with a particular focus on immune-tumor cell interactions in Tumor-on-Chip (TOC) applications. The work combines mathematical modeling based on chemotaxis principles with advanced numerical methods to analyze complex biological processes in microfluidic environments.

The report is structured in four main chapters:

Chapter 1 establishes the general context and motivation for OOC modeling. We introduce the fundamental concepts including the Hybrid Discontinuous Galerkin (HDG) method, chemotaxis, the Keller-Segel model, and the specific challenges posed by OOC geometries. We also outline the internship objectives: developing mathematical models adapted to complex OOC geometries and implementing them using the HDG method in the Feel++ computational framework.

Chapter 2 provides a comprehensive presentation of the HDG method as the core numerical strategy. We begin with the theoretical foundations of HDG for convection-diffusion problems, detailing the weak formulation, matrix formulation, and static condensation procedures. It demonstrates how HDG effectively handles discontinuous solutions across complex geometries while maintaining computational efficiency through element-wise reconstruction and reduced global coupling.

Chapter 3 focuses on the Keller-Segel chemotaxis model, which serves as the mathematical foundation for describing cellular migration dynamics. We present the model equations, introduce auxiliary variables for numerical treatment, and develop the HDG-adapted weak formulation. Particular attention is given to the discretization of chemotactic terms and the stability properties of the resulting numerical scheme.

Chapter 4 applies the developed framework to specific OOC configurations. We present two main scenarios: a single-chamber model capturing basic immune-tumor interactions, and a more complex two-chamber model connected by microchannels that reflects realistic TOC geometries. It details the mathematical formulation for hybrid 2D-1D geometries and discusses the implementation chal-

lenges associated with coupling chambers and channels.

The report demonstrates how the combination of the Keller-Segel model with the HDG method provides a robust computational platform for studying cellular dynamics in OOC systems. The framework successfully handles the complex geometries typical of microfluidic devices while maintaining the physical properties essential for accurate biological modeling, such as mass conservation and solution positivity.

This work contributes to the growing field of computational biology by providing validated numerical tools for analyzing OOC experiments and supporting the development of more effective therapeutic strategies through *in silico* studies.

1 Introduction: General Context and Modeling

1.1 General Context

Recent advances in bioengineering have led to the development of Organ-on-Chip (OOC) devices, microfluidic systems capable of reproducing the physiological microenvironments of human tissues *in vitro*. These systems, composed of two-dimensional chambers interconnected by one-dimensional microchannels, enable the simulation of complex biological processes, such as interactions between immune and tumor cells in the context of Tumor-on-Chip (TOC) studies. Compared to animal experiments (*in vivo*), often limited by interspecies physiological differences and ethical concerns, or traditional cell cultures (*in vitro*), which lack complexity, OOCs offer a valuable alternative. They allow the study of phenomena such as cell migration, immune response, or treatment efficacy under controlled conditions that closely mimic physiological reality.

However, the wealth of data generated by OOCs requires mathematical modeling and numerical simulation tools to interpret observations and predict cellular behaviors. Macroscopic models, based on partial differential equations (PDEs), are particularly suited to describe chemotactic dynamics, *i.e.*, cell movements directed by chemical gradients. These models, combined with *in silico* simulations, allow testing biological hypotheses and optimizing experimental conditions. This internship is part of this approach, developing a numerical framework to analyze cellular dynamics in OOCs, with a focus on immune-tumor interactions and chemotactic migration.

1.2 Vocabulary and Key Concepts

To lay the groundwork for this work, several fundamental concepts are introduced:

- **Hybrid Discontinuous Galerkin (HDG) Method:** A numerical method combining the advantages of discontinuous Galerkin methods (flexibility on complex meshes) and hybridization (reduction of degrees of freedom) to solve PDEs.

- **Chemotaxis:** directed cell migration in response to chemical signals coming from external sources or secreted by the organisms themselves [14]
- **Keller-Segel Model** [18]: A system of coupled parabolic PDEs describing chemotactic migration through two main variables: cell density and chemoattractant concentration.
- **Organ-on-Chip (OOC):** Microfluidic systems containing miniature, artificial, or natural tissues cultured inside chips, composed of 2D chambers (where cellular interactions occur) and 1D microchannels (for transport).
- **Blow-up:** A phenomenon where cell density rapidly accumulates at a point, often used as a test case to validate numerical schemes.
- **Feel++:** A C++ library for the numerical resolution of PDEs, used here to implement the HDG method, (see [1] for more details).

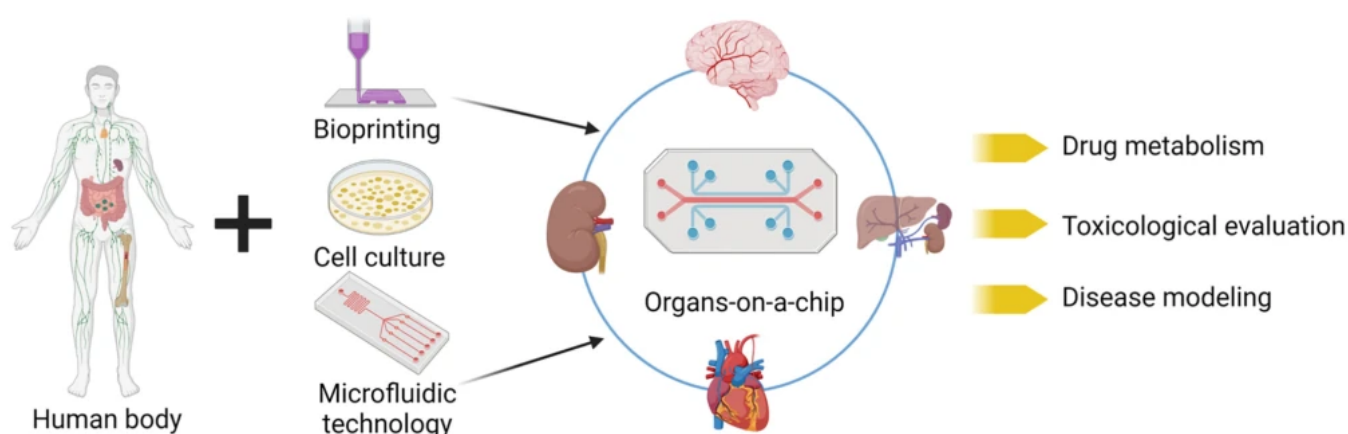


Figure 1.1: Human-On-Chip: A composition of multiple Organ-on-Chips to replicate human microarchitecture and functions.

The complexity of OOC devices is illustrated in Figure 1.2, which presents a comprehensive view of a microfluidic chip designed for studying cellular interactions. This particular configuration demonstrates the multi-scale nature of OOC systems, featuring interconnected culture chambers linked by microchannels that facilitate controlled cellular migration and communication.

1.3 Objectives and Approach of the Internship

This internship aims to develop a numerical tool to simulate cellular dynamics in OOCs, focusing on immune-tumor cell interactions in a TOC context. The specific objectives are:

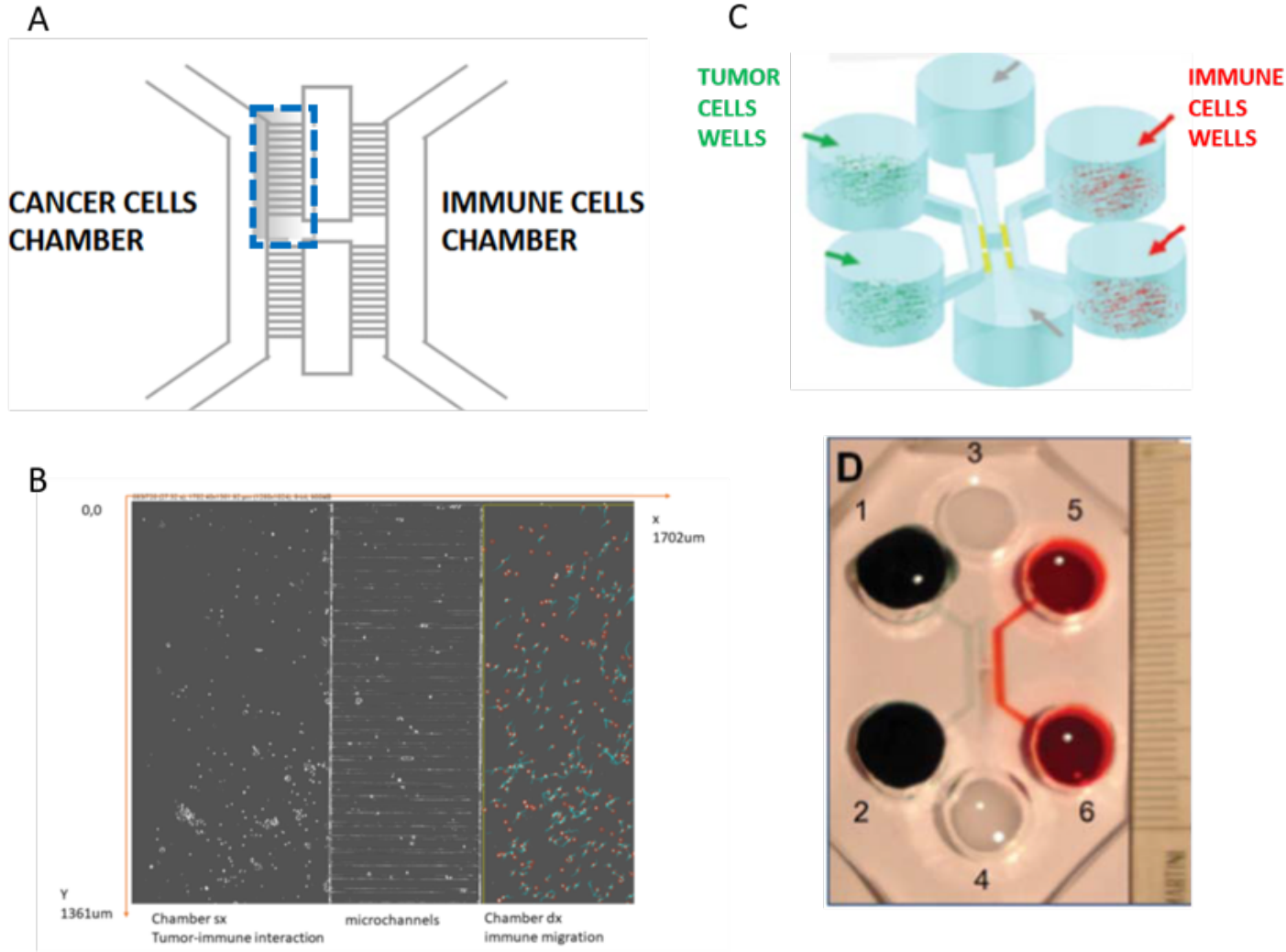


Figure 1.2: Microfluidic chip environment showing: (A) Schematic 2D layout with three main culture chambers connected by microchannel arrays, with the monitored region highlighted; (B) Chip area captured in experimental video recordings; (C) Three-dimensional configuration of the complete chip architecture; (D) Actual photograph of the fabricated microfluidic device.

1. Build a mathematical model based on chemotaxis, adapted to the complex geometries of OOCs (2D chambers and 1D microchannels). This objective was partially achieved by developing the Keller-Segel model for a single-chamber configuration, with preliminary extensions to a two-chamber model connected by a channel. The hybrid 2D-1D geometry was theoretically defined, and a detailed mesh was created using Gmsh to support future simulations.
2. Use Hybrid Discontinuous Galerkin (HDG) method and feelpp toolboxes to solve this model, including the study of stabilization to ensure convergence and accuracy. This goal was advanced through the implementation of the coefficientformpdes[2] and the HDG method[3] in the Feel++ library for the

single-chamber model, with successful simulations validating its stability. However, a bug in Feel++ affecting the HDG implementation for the two-chamber model limited further progress, despite efforts to address it.

3. Test and validate the method on simple cases with known analytical solutions, then on realistic OOC configurations. The method was tested and validated on the single-chamber model, with numerical results and parameter sensitivity analyses presented in Section 4.2. Due to time constraints and the unresolved Feel++ bug, validation on the two-chamber configuration could not be completed, though the theoretical framework is ready for future implementation.
4. Explore potential improvements, such as the use of adaptive meshes or the integration of experimental data to calibrate the model. This objective was partially addressed through the creation of a detailed Gmsh mesh for the complete chip geometry, laying the groundwork for adaptive meshing. However, the integration of experimental data and further refinements were beyond the scope of this internship due to time limitations.

The approach adopted relies on a combination of mathematical modeling and numerical simulation. The Keller-Segel model [18] is chosen as the basis to describe chemotactic dynamics, due to its simplicity and its ability to capture the phenomena of interest. The HDG method, implemented in Feel++ [1], is used to numerically solve this model, with a particular focus on stability and accuracy in hybrid geometries, see section 2.

The report is structured to follow a logical progression: first, the section 2 elaborates in detail on the numerical method, explaining the variational formulation, the handling of convection-diffusion terms, stabilization techniques, and verification through simple test cases with known analytical solutions. Next, the section 3 focuses on the mathematical model, presenting its equations and variational formulation adapted to HDG, while relying on the previously established vocabulary to avoid redundancies. Then, section 4 discusses the adaptation of the model and the HDG method to the specific geometries of OOCs, illustrating this with simulation examples in the context of Tumor-on-Chip (TOC). Finally, a general conclusion will synthesize the results obtained, highlight the challenges encountered, and propose perspectives for future work.

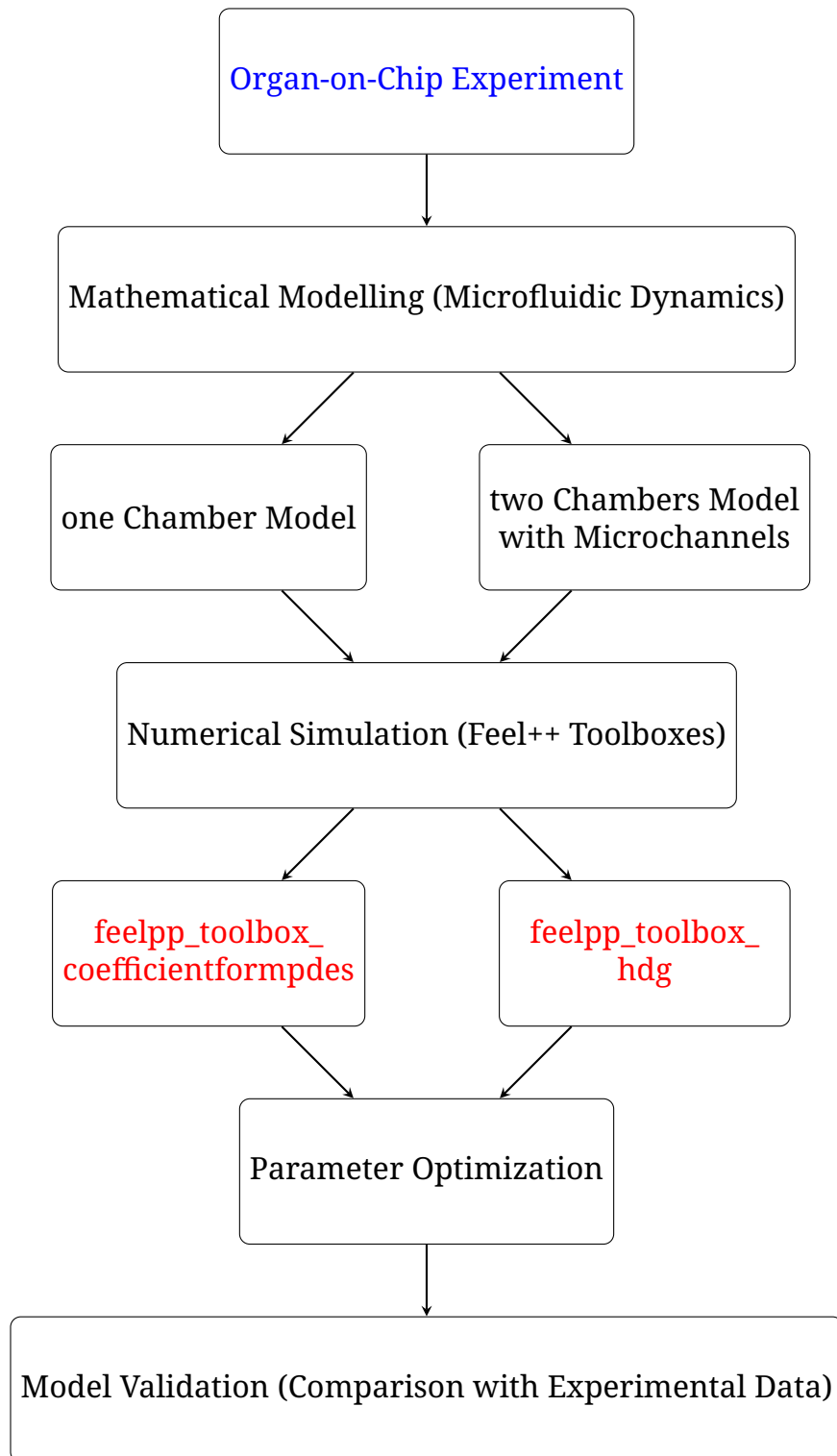


Figure 1.3: Workflow diagram for modeling and simulation of organ-on-chip systems.

2 Resolution Strategy with the HDG Method

2.1 General Overview of the HDG Method

2.1.1 Context and Motivation

In recent years, Discontinuous Galerkin (DG) finite element methods have emerged as a competitive alternative for solving nonlinear hyperbolic systems of conservation laws. The advantages of DG methods over classical finite difference and finite volume methods are well-documented in the literature: DG methods work efficiently on arbitrary meshes, allow stable and high-order accurate discretizations of convective and diffusive operators, facilitate simple and unambiguous imposition of boundary conditions, and are highly suitable for parallelization and adaptivity. Despite all these advantages, DG methods have not yet had a significant impact in practical applications. This is largely due to the high computational cost associated with them compared to finite difference or finite volume schemes.

Hybridizable Discontinuous Galerkin (HDG) methods have been recently introduced to address this issue (see details in [5, 6]). The HDG method has several attractive features:

- It provides optimal approximation of both primal and flux variables.
- It requires less globally coupled degrees of freedom than DG methods of comparable accuracy.
- It allows local element-by-element postprocessing to obtain new approximations with enhanced accuracy and conservation properties.

The HDG implementation has been carried out using Feel++(see [7]). Feel++ is a C++ library for solving partial differential equations in \mathbb{R}^d , $d = 1, 2, 3$ and on manifolds using a wide range of Galerkin methods, from standard finite element methods, both continuous and discontinuous, to reduced basis methods. Endowed with a domain specific language, whose grammar and vocabulary are tailored to the description of variational formulations and their discretization, Feel++ provides a scalable, versatile and expressive computational framework

that applied mathematicians, engineers and scientists can use to access and test a wide range of methods and problems, both classical and newly developed.

2.1.2 HDG Method for a Convection-Diffusion Problem

A new HDG method has been proposed for the convection-diffusion problem on general polyhedral meshes. This new HDG method is a generalization of HDG methods for linear elasticity, [12], to problems involving a convection term. They were initially developed for elliptic problems [10, 11] and have already been extended and demonstrated for linear and nonlinear convection-diffusion problems [8, 9].

For arbitrary polyhedral elements, we use polynomials of degree $k + 1$ and $k \geq 0$ to approximate the scalar variable (the potential) and its gradient (the flux), respectively. However, we only use polynomials of degree k to approximate the numerical trace of the scalar variable on the faces, which allows for a very efficient implementation of the method, as the numerical trace of the scalar variable is the only globally coupled unknown. The global L^2 norm of the error of the scalar variable converges with an order of $k + 2$, while that of its gradient converges with an order of $k + 1$. From the perspective of the degrees of freedom of the globally coupled unknown, namely the numerical trace, this method achieves superconvergence for the scalar variable without requiring post-processing.

2.2 The model problem

This section introduces the Hybridizable Discontinuous Galerkin (HDG) method for solving diffusion-reaction problems, providing a foundational framework for the numerical methods used in this report. The content is adapted from the works of [10, 11], reformulated to align with the objectives of this study.

We consider a second-order elliptic model problem defined on a polyhedral domain $\Omega \subset \mathbb{R}^n$ (where $n \geq 2$), with the boundary $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N$. The governing equations are:

$$\Lambda \mathbf{u} + \nabla p - \Lambda \alpha p = \mathbf{0}, \quad \text{in } \Omega, \quad (2.1a)$$

$$\nabla \cdot \mathbf{u} + dp = f, \quad \text{in } \Omega, \quad (2.1b)$$

$$p = h_D, \quad \text{on } \Gamma_D, \quad (2.1c)$$

$$\mathbf{u} \cdot \mathbf{n} = h_N, \quad \text{on } \Gamma_N, \quad (2.1d)$$

where:

- \mathbf{u} : the flux, representing the flow or transport of a quantity,
- p : the scalar variable, such as a potential or concentration,
- $\Lambda(\mathbf{x})$: is a matrix valued function that is symmetric and uniformly positive definite on Ω ,
- α : a vector-valued function representing the convection velocity,
- $d(\mathbf{x})$: a non-negative scalar function associated with reaction terms,
- $f \in L^2(\Omega)$: a source term describing external contributions,
- h_D and h_N : the prescribed Dirichlet and the Neumann boundary conditions, respectively.

2.2.1 Notations and Approximations Spaces

Let \mathcal{T}_h be a partition of the domain Ω into disjoint elements (not necessarily conforming), where the shape of the elements is flexible within this general framework. An interior face F of \mathcal{T}_h is defined as a set of positive $(n - 1)$ -dimensional Lebesgue measure, given by $F = \partial K^+ \cap \partial K^-$, where K^+ and K^- are two distinct elements in \mathcal{T}_h . A boundary face F is defined as $F = \partial K \cap \Gamma$ for some element $K \in \mathcal{T}_h$, with a non-zero $(n - 1)$ -dimensional measure.

We denote the set of interior faces by \mathcal{E}_h^o , the set of boundary faces by \mathcal{E}_h^∂ , and so:

$$\mathcal{E}_h = \mathcal{E}_h^o \cup \mathcal{E}_h^\partial$$

The finite element spaces for the approximated flux \mathbf{u}_h and scalar solution p_h are defined as:

$$\begin{aligned} \mathbf{V}_h &= \{\mathbf{v} : \Omega \rightarrow \mathbb{R}^n : \mathbf{v}|_K \in \mathbf{V}(K) \quad \forall K \in \mathcal{T}_h\}, \\ W_h &= \{w : \Omega \rightarrow \mathbb{R} : w|_K \in W(K) \quad \forall K \in \mathcal{T}_h\}, \end{aligned}$$

where $\mathbf{V}(K)$ and $W(K)$ are local polynomial spaces on element K , to be specified based on the chosen HDG variant. We also define the space for the numerical trace on the mesh skeleton:

$$M_h = \{\mu : \mathcal{E}_h \rightarrow \mathbb{R} : \mu|_F \in M(F) \quad \forall F \in \mathcal{E}_h\},$$

along with auxiliary subspaces:

$$\begin{aligned}
M_h^o &= \{\mu \in M_h : \mu|_\Gamma = 0\}, \\
M_h^D &= \{\mu \in M_h : \mu|_F = 0 \quad \forall F \in \mathcal{E}_h \setminus \Gamma_D\}, \\
M_h^N &= \{\mu \in M_h : \mu|_F = 0 \quad \forall F \in \mathcal{E}_h \setminus \Gamma_N\},
\end{aligned}$$

where $M(F)$ is a local polynomial space on face F . Different choices of $\mathbf{V}(K)$, $W(K)$, and $M(F)$ lead to distinct HDG formulations.

For discontinuous functions $u \in W_h$ or $\mathbf{v} \in \mathbf{V}_h$, the trace on an interior face $F = \partial K^+ \cap \partial K^-$ is double-valued, with branches $u|_{K^+}$ and $u|_{K^-}$. For a vector-valued function \mathbf{v} , the jump of its normal component across an interior face is defined as:

$$[[\mathbf{v}]]_F = \mathbf{v}_{K^+} \cdot \mathbf{n}_{K^+} + \mathbf{v}_{K^-} \cdot \mathbf{n}_{K^-},$$

where \mathbf{n}_K is the outward unit normal to element K . On a boundary face F , the jump is:

$$[[\mathbf{v}]]_F = \mathbf{v}_K \cdot \mathbf{n}_K.$$

A similar definition applies to scalar functions. For functions $u, v \in L^2(D)$, we use the notations

$$\begin{aligned}
(u, v)_D &= \int_D uv \quad \forall \quad D \subset \mathbb{R}^n, \\
\langle u, v \rangle_D &= \int_D uv \quad \forall \quad D \subset \mathbb{R}^{n-1}.
\end{aligned}$$

And so,

$$\begin{aligned}
(v, w)_{\mathcal{T}_h} &= \sum_{K \in \mathcal{T}_h} (v, w)_K, \\
\langle \mu, \lambda \rangle_{\partial \mathcal{T}_h} &= \sum_{K \in \mathcal{T}_h} \langle \mu, \lambda \rangle_{\partial K}.
\end{aligned}$$

2.2.2 Characterization of the Exact Solution

The exact solution (\mathbf{u}, p) to equations (2.1a)–(2.1d) satisfies local problems on each element $K \in \mathcal{T}_h$:

$$\begin{aligned}
\Lambda \mathbf{u} + \nabla p - \Lambda \alpha p &= \mathbf{0}, \quad \text{in } K, \\
\nabla \cdot \mathbf{u} + dp &= f, \quad \text{in } K,
\end{aligned}$$

along with transmission conditions ensuring continuity across element inter-

faces:

$$\begin{aligned} [[\hat{p}]] &= 0, & \text{if } F \in \mathcal{E}_h^o, \\ [[\hat{\mathbf{u}}]] &= 0, & \text{if } F \in \mathcal{E}_h^o, \end{aligned}$$

and boundary conditions:

$$\hat{p} = h_D, \quad \text{if } F \in \Gamma_D, \quad (2.2)$$

$$\hat{\mathbf{u}} \cdot \mathbf{n} = h_N, \quad \text{if } F \in \Gamma_N, \quad (2.3)$$

where \hat{p} and $\hat{\mathbf{u}}$ are the traces of p and \mathbf{u} on the element boundaries.

The transmission conditions enforce continuity of the normal component of $\hat{\mathbf{u}}$ across inter-element faces.

To solve for (\mathbf{u}, p) within each element K , we use a local Dirichlet problem defined by:

$$\Lambda \mathbf{u} + \nabla p - \Lambda \alpha p = \mathbf{0}, \quad \text{in } K, \quad (2.4)$$

$$\nabla \cdot \mathbf{u} + dp = f, \quad \text{in } K, \quad (2.5)$$

$$p = \hat{p}, \quad \text{on } \partial K, \quad (2.6)$$

where \hat{p} on ∂K acts as a boundary condition.

The global problem then determines \hat{p} on each face $F \in \mathcal{E}_h$ by solving:

$$[[\hat{\mathbf{u}}]]_F = 0, \quad \text{if } F \in \mathcal{E}_h^o, \quad (2.7)$$

$$[[\hat{\mathbf{u}}]]_F = h_N, \quad \text{if } F \in \Gamma_N, \quad (2.8)$$

$$\hat{p}|_F = h_D, \quad \text{if } F \in \Gamma_D. \quad (2.9)$$

The HDG method constructs a discrete version of equations (2.4)–(2.9), where the globally coupled degrees of freedom are associated with the numerical trace \hat{p} .

2.2.3 Weak Formulation on Each Element

For each element $K \in \mathcal{T}_h$, we define the approximate solution $(\mathbf{u}_h, p_h) \in \mathbf{V}(K) \times W(K)$ in terms of \hat{p}_h and f , satisfying the weak formulation:

$$(\Lambda \mathbf{u}_h, \mathbf{v})_K - (p_h, \nabla \cdot \mathbf{v})_K - (\Lambda \alpha p_h, \mathbf{v})_K + \langle \hat{p}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial K} = 0, \quad (2.10)$$

$$-(\mathbf{u}_h, \nabla w)_K + \langle \hat{\mathbf{u}}_h \cdot \mathbf{n}, w \rangle_{\partial K} + (dp_h, w)_K = (f, w)_K, \quad (2.11)$$

for all test functions $(\mathbf{v}, w) \in \mathbf{V}(K) \times \mathbf{W}(K)$, where $\hat{\mathbf{u}}_h$ is the numerical flux.

The numerical trace $\hat{p}_h \in M_h$ is determined by solving the global system:

$$\langle \mu, [[\hat{\mathbf{u}}_h]] \rangle_F = 0, \quad \forall \mu \in M(F), \quad \text{if } F \in \mathcal{E}_h^o, \quad (2.12)$$

$$\langle \mu, [[\hat{\mathbf{u}}_h]] \rangle_F = \langle \mu, h_N \rangle_F, \quad \forall \mu \in M(F), \quad \text{if } F \in \Gamma_N, \quad (2.13)$$

$$\langle \mu, \hat{p}_h \rangle_F = \langle \mu, h_D \rangle_F, \quad \forall \mu \in M(F), \quad \text{if } F \in \Gamma_D. \quad (2.14)$$

By solving (2.10) and (2.11) for (\mathbf{u}_h, p_h) on each element in terms of \hat{p}_h and f , and substituting into (2.12)–(2.14), we obtain a reduced system where the only globally coupled unknowns are the degrees of freedom of \hat{p}_h . This process corresponds to static condensation of the full discrete system.

Problem Summary:

The approximate solution $(\mathbf{u}_h, p_h, \hat{p}_h) \in \mathbf{V}_h \times W_h \times M_h$ satisfies the following system of equations:

$$(\Lambda \mathbf{u}_h, \mathbf{v})_{\mathcal{T}_h} - (p_h, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} - (\Lambda \alpha p_h, \mathbf{v})_{\mathcal{T}_h} + \langle \hat{p}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} = 0, \quad \forall \mathbf{v} \in \mathbf{V}_h, \quad (2.15)$$

$$-(\mathbf{u}_h, \nabla w)_{\mathcal{T}_h} + \langle \hat{\mathbf{u}}_h \cdot \mathbf{n}, w \rangle_{\partial \mathcal{T}_h} + (dp_h, w)_{\mathcal{T}_h} = (f, w)_{\mathcal{T}_h}, \quad \forall w \in W_h, \quad (2.16)$$

$$\langle \mu, \hat{\mathbf{u}}_h \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h \setminus \Gamma} = 0, \quad \forall \mu \in M_h^o, \quad (2.17)$$

$$\langle \mu, \hat{\mathbf{u}}_h \cdot \mathbf{n} \rangle_{\Gamma_N} = \langle \mu, h_N \rangle_{\Gamma_N} \quad \forall \mu \in M_h^N \quad (2.18)$$

$$\langle \mu, \hat{p}_h \rangle_{\Gamma_D} = \langle \mu, h_D \rangle_{\Gamma_D} \quad \forall \mu \in M_h^D \quad (2.19)$$

where the local spaces $\mathbf{V}(K)$, $W(K)$, $M(F)$, and the numerical flux $\hat{\mathbf{u}}_h$ must be specified to fully define the HDG method.

2.2.4 The HDG Method

Having established the general structure of the HDG method, we now focus on its specific implementation for solving diffusion-reaction problems. While other hybridizable methods, such as the Raviart-Thomas Hybrid (RT-H) and Brezzi-Douglas-Marini Hybrid (BDM-H) methods, also fit within this framework (see [3])

and references therein for details), we concentrate on the HDG method due to its balanced polynomial spaces and effective stabilization techniques. To define the HDG method, we need to specify three key components:

- The numerical trace of the flux $\hat{\mathbf{u}}_h$,
- The local spaces $\mathbf{V}(K)$ and $W(K)$,
- The space of approximate traces M_h .

This method leverages local discontinuous Galerkin (DG) solvers to construct its approximation. Its defining components are:

1. For each element $K \in \mathcal{T}_h$, the numerical flux is defined as:

$$\hat{\mathbf{u}}_h = \mathbf{u}_h + \tau_K(p_h - \hat{p}_h)\mathbf{n} \quad \text{on } \partial K,$$

where τ_K is a non-negative stabilization function that may vary across ∂K , with $\tau_K > 0$ on at least one face of ∂K . On an interior face $F \in \mathcal{E}_h^o$ shared by elements K^+ and K^- , τ can be double-valued with branches $\tau^- = \tau_{K^-}$ and $\tau^+ = \tau_{K^+}$.

2. The local approximation spaces are:

$$\mathbf{V}(K) = [P_k(K)]^n, \quad W(K) = P_k(K), \quad k \geq 0,$$

where $P_k(K)$ denotes the space of polynomials of degree at most k on element K .

3. The space of numerical traces is:

$$M_h = \left\{ \mu \in L^2(\mathcal{E}_h) : \mu|_F \in P_k(F) \quad \forall F \in \mathcal{E}_h \right\}.$$

The numerical flux $\hat{\mathbf{u}}_h$ is conservative (i.e., $\hat{\mathbf{u}}_h \in H(\text{div}, \Omega)$) due to the continuity imposed by the global problem, although the underlying flux \mathbf{u}_h may not be conservative since $\tau_K \neq 0$. The accuracy and convergence properties of the HDG method are detailed in Table 2.1.

2.2.4.1 Matrix Formulation

The primary goal of hybridization in the HDG method is to reduce the system (2.15)–(2.19) to a linear system involving only the numerical trace \hat{p}_h , a process known as *static condensation*. The variables \mathbf{u}_h and p_h are then reconstructed element-wise after solving for \hat{p}_h . This approach is efficient because equations

(2.15) and (2.16) are local, and the spaces \mathbf{V}_h and W_h are fully discontinuous. Here, we outline the static condensation procedure for the HDG method.

For that method method, the system (2.15)–(2.19) is rewritten by incorporating the numerical flux $\hat{\mathbf{u}}_h = \mathbf{u}_h + \tau_K(p_h - \hat{p}_h)\mathbf{n}$, yielding:

$$(\Lambda \mathbf{u}_h, \mathbf{v})_{\mathcal{T}_h} - (p_h, \nabla \cdot \mathbf{v})_{\mathcal{T}_h} - (\Lambda \alpha p_h, \mathbf{v})_{\mathcal{T}_h} + \langle \hat{p}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial \mathcal{T}_h} = 0, \quad \forall \mathbf{v} \in \mathbf{V}_h, \quad (2.20)$$

$$(\nabla \cdot \mathbf{u}_h, w)_{\mathcal{T}_h} + \langle \tau p_h, w \rangle_{\partial \mathcal{T}_h} + (dp_h, w)_{\mathcal{T}_h} - \langle \tau \hat{p}_h, w \rangle_{\partial \mathcal{T}_h} = (f, w)_{\mathcal{T}_h}, \quad \forall w \in W_h, \quad (2.21)$$

$$\langle \mathbf{u}_h \cdot \mathbf{n}, \mu_1 \rangle_{\partial \mathcal{T}_h \setminus \Gamma} + \langle \tau p_h, \mu_1 \rangle_{\partial \mathcal{T}_h \setminus \Gamma} - \langle \tau \hat{p}_h, \mu_1 \rangle_{\partial \mathcal{T}_h \setminus \Gamma} = 0, \quad \forall \mu_1 \in M_h^o, \quad (2.22)$$

$$\langle \mathbf{u}_h \cdot \mathbf{n}, \mu_2 \rangle_{\Gamma_N} + \langle \tau p_h, \mu_2 \rangle_{\Gamma_N} - \langle \tau \hat{p}_h, \mu_2 \rangle_{\Gamma_N} = \langle h_N, \mu_2 \rangle_{\Gamma_N} \quad \forall \mu_2 \in M_h^N \quad (2.23)$$

$$\langle \hat{p}_h, \mu_3 \rangle_{\Gamma_D} = \langle h_D, \mu_3 \rangle_{\Gamma_D} \quad \forall \mu_3 \in M_h^D \quad (2.24)$$

for all test functions $(\mathbf{v}, w, \mu_1, \mu_2, \mu_3) \in \mathbf{V}_h \times W_h \times M_h^o \times M_h^N \times M_h^D$.

The static condensation process eliminates \mathbf{u}_h and p_h from this system, resulting in a reduced linear system for \hat{p}_h . After solving for \hat{p}_h , the local variables \mathbf{u}_h and p_h can be recovered element-wise using (2.20) and (2.21). This element-by-element reconstruction leverages the discontinuous nature of the approximation spaces, making the HDG method computationally efficient for large-scale problems.

Now let us define matrices corresponding to the bilinear forms in (2.20) and (2.21):

$$\begin{aligned} A_{11}^K &\leftrightarrow (\Lambda \mathbf{u}_h, \mathbf{v})_K, & A_{12}^K &\leftrightarrow -(p_h, \nabla \cdot \mathbf{v})_K - (\Lambda \alpha p_h, \mathbf{v})_K, & A_{13}^K &\leftrightarrow \langle \hat{p}_h, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial K}, \\ A_{21}^K &\leftrightarrow (\nabla \cdot \mathbf{u}_h, w)_K, & A_{22}^K &\leftrightarrow \langle \tau p_h, w \rangle_{\partial K} + (dp_h, w)_K, & A_{23}^K &\leftrightarrow -\langle \tau \hat{p}_h, w \rangle_{\partial K}, \\ A_{31}^K &\leftrightarrow \langle \mathbf{u}_h \cdot \mathbf{n}, \mu \rangle_{\partial K}, & A_{32}^K &\leftrightarrow \langle \tau p_h, \mu \rangle_{\partial K}, & A_{33}^K &\leftrightarrow \langle \tau \hat{p}_h, \mu \rangle_{\partial K}, \\ & & A_f^K &\leftrightarrow (f, w)_K. \end{aligned}$$

and represent $\mathbf{u}_h|_K$, $p_h|_K$, and $\hat{p}_h|_{\partial K}$ by vectors \mathbf{u}_K , \mathbf{p}_K , and $\mathbf{p}_{\partial K}$, respectively, using appropriate finite element bases. We define the block matrices and vectors:

$$A^K = \begin{bmatrix} A_{11}^K & A_{12}^K \\ A_{21}^K & A_{22}^K \end{bmatrix}, \quad B^K = \begin{bmatrix} A_{13}^K \\ A_{23}^K \end{bmatrix}, \quad F^K = \begin{bmatrix} \mathbf{0} \\ A_f^K \end{bmatrix}, \quad C^K = \begin{bmatrix} A_{31}^K & A_{32}^K \end{bmatrix}.$$

Then the local system (2.20)–(2.21) can then be written as:

$$A^K \begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} + B^K \mathbf{p}_{\partial K} = F^K.$$

Solving for \mathbf{u}_K and \mathbf{p}_K , assuming A^K is invertible, gives:

$$\begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} = -(A^K)^{-1} B^K \mathbf{p}_{\partial K} + (A^K)^{-1} F^K. \quad (2.25)$$

Next, we compute the numerical flux $\mathbf{u}_h \cdot \mathbf{n} + \tau(p_h - \hat{p}_h)$ on ∂K , which appears in (2.22) and (2.23). The associated bilinear form is:

$$\langle \mathbf{u}_h \cdot \mathbf{n} + \tau(p_h - \hat{p}_h), \mu \rangle_{\partial K} = \langle \mathbf{u}_h \cdot \mathbf{n} + \tau p_h, \mu \rangle_{\partial K} - \langle \tau \hat{p}_h, \mu \rangle_{\partial K},$$

for all $\mu \in M(\partial K)$.

Using C^K , the matrix representation of this form is:

$$C^K \begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} - A_{33}^K \mathbf{p}_{\partial K} = 0,$$

Substituting (2.25) into this expression yields:

$$\begin{aligned} C^K \begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} - A_{33}^K \mathbf{p}_{\partial K} &= C^K \left(-(A^K)^{-1} B^K \mathbf{p}_{\partial K} + (A^K)^{-1} F^K \right) - A_{33}^K \mathbf{p}_{\partial K} \\ &= -C^K (A^K)^{-1} B^K \mathbf{p}_{\partial K} + C^K (A^K)^{-1} F^K - A_{33}^K \mathbf{p}_{\partial K}. \end{aligned}$$

Let's define

$$D^K = C^K (A^K)^{-1} B^K + A_{33}^K \quad \text{and} \quad D_f^K = C^K (A^K)^{-1} F^K,$$

so the numerical flux term becomes:

$$C^K \begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} - A_{33}^K \mathbf{p}_{\partial K} = D_f^K - D^K \mathbf{p}_{\partial K}. \quad (2.26)$$

This expression represents the numerical flux on ∂K , which is used to assemble the global system for \hat{p}_h by enforcing continuity conditions across elements.

2.2.4.2 Boundary Conditions and Global Solver

To obtain the final solution, we assemble a global system for $\hat{p}_h \in M_h$, incorporating the boundary conditions specified in (2.23) and (2.24).

Dirichlet Boundary Conditions The Dirichlet condition (2.24) requires that \hat{p}_h matches the prescribed data h_D on Γ_D . This is enforced by computing the L^2 -projection of h_D on to $M_h|_{\Gamma_D}$, the restriction of M_h to Dirichlet boundary faces. Let \hat{p}_D denote the vector of coefficients representing this projection, obtained by solving:

$$\langle \hat{p}_h, \mu \rangle_{\Gamma_D} = \langle h_D, \mu \rangle_{\Gamma_D} \quad \forall \mu \in M_h|_{\Gamma_D}.$$

These coefficients will be used to set the values of \hat{p}_h on Dirichlet faces and adjust the global system accordingly.

Neumann Boundary Conditions The Neumann condition (2.23) involves the numerical flux on Γ_N . Using (2.26), this condition contributes to the right-hand side of the global system. We define a vector \mathbf{G}_N , which contains the values $\langle h_N, \mu \rangle_{\Gamma_N}$ for basis functions $\mu \in M_h^N$ (i.e., supported on Neumann faces) and zeros elsewhere.

Assembling the Global Solver The global system is assembled by enforcing continuity of the numerical flux across interior faces (via (2.22)) and applying the Neumann condition (via (2.23)). For each element K , the local contribution to the flux is given by (2.26). The global matrix \mathbb{M} is constructed by assembling the local matrices D^K from all elements, ensuring that on interior faces $F \in \mathcal{E}_h^o$, the fluxes from adjacent elements K^+ and K^- are summed with opposite signs (due to the outward normals pointing in opposite directions), enforcing continuity. Similarly, the global vector \mathbf{F} is assembled from the local vectors D_f^K , summing contributions from all elements. The global system before handling Dirichlet conditions is:

$$\mathbb{M}\hat{\mathbf{p}} = \mathbf{F} + \mathbf{G}_N, \tag{2.27}$$

where $\hat{\mathbf{p}}$ is the global vector of coefficients for $\hat{p}_h \in M_h$.

To incorporate the Dirichlet condition, the degrees of freedom of $\hat{\mathbf{p}}$ corresponding to faces on Γ_D are set to the values in \hat{p}_D . These known values are moved to the right-hand side of the system, and the rows of \mathbb{M} corresponding to Dirichlet degrees of freedom are removed, resulting in a reduced system for the

remaining degrees of freedom. Solving this system yields \hat{p}_h , after which \mathbf{u}_h and p_h can be recovered element-wise using (2.25).

2.2.4.3 Accuracy Analysis

The following table summarizes the effect of the local spaces and the stabilization parameter τ on the accuracy of the HDG method considered. The superconvergence of \bar{p}_h , where $\bar{p}_h|_K$ denotes the average of p_h on $K \in \mathbf{T}_h$, is obtained by postprocessing.

Method	τ	\mathbf{u}_h	p_h	\bar{p}_h	k
HDG	$O(h)$	$k + 1$	k	$k + 2$	≥ 1
HDG	$O(1)$	$k + 1$	$k + 1$	$k + 2$	≥ 1
HDG	$O(1)$	1	1	1	$= 0$
HDG	$O(1/h)$	k	$k + 1$	$k + 1$	≥ 1

Table 2.1: Orders of accuracy for HDG

This following subsection is adapted from [7].

2.2.5 HDG implementation in Feel++

2.2.5.1 Feel++ general description

The computational framework described hereafter is based on the Finite Element Embedded Library in C++ (Feel++). Feel++ allows using a very wide range of Galerkin methods, as well as other numerical methods such as domain decomposition methods including mortar and three fields methods, fictitious domain methods, or certified reduced basis. The ingredients of the software include a very expressive embedded language, seamless interpolation, mesh adaption, and seamless parallelization. Feel++ has been used in various contexts including the development and numerical verification of new mathematical methods or the development of large multi-physics applications. The range of users spans from mechanical engineers in industry, physicists in complex fluids, computer scientists in biomedical applications, to applied mathematicians thanks to the shared common mathematical embedded language hiding linear algebra and computer science technical issues.

Feel++ provides a mathematical kernel for solving partial differential equations using arbitrary order Galerkin methods (fem, sem, cg, dg, crb) in 1d, 2d, 3d, and on manifolds using simplices and hypercubes meshes [1]. It is based on:

- a polynomial library allowing for a wide range of polynomial expansions including Hdiv and Hcurl elements;
- a light interface to Boost.UBlas, Eigen3, and PETSc/SLEPc as well as a scalable in-house solution strategy;
- a language for Galerkin methods starting with fundamental concepts such as function spaces, linear and bilinear forms, operators, functionals, and integrals;
- a framework that allows user codes to scale seamlessly from single core computation to thousands of cores and enables hybrid computing.
- Feel++ takes advantage of the newest C++ standard (C++17) such as type inference and the Boost C++ Libraries such as Boost.Parameter, Boost.Fusion, Boost.Hana, Boost.MPL, and many more. These language enhancements and libraries favor the development of very concise, robust, and expressive C++ codes.

See the details of the HDG implementation in Feel++ in [7], Page 17 of 33.

3 Keller-Segel chemotaxis model

The Keller-Segel model, introduced as early as 1970 [16], is a cornerstone for describing the chemotactic dynamics of cellular populations. Despite its relatively simple formulation, it captures complex biological phenomena such as directed cell movement and pattern formation. This model couples a parabolic equation for cell density with an elliptic or parabolic equation for chemoattractant concentration, thereby representing cell migration toward regions of high chemical concentration, which can lead to marked aggregation, sometimes in the form of a blow-up phenomenon. However, the strong non-linearity of the chemotactic terms makes this system analytically challenging to solve, and solutions can either remain bounded with localized peaks or become unbounded in finite time (see [17]). In this section, we focus on a bounded regime, suitable for our biological application where blow-up is absent, while numerically exploring the blow-up regime to validate the method. The numerical resolution of the Keller-Segel model is challenging due to its sensitivity to parameters and the singular structure of the solutions. An effective numerical method must therefore preserve physical properties, such as the non-negativity of solutions, mass conservation under zero-flux conditions, and avoid non-physical oscillations in regions of high density. The following section will be inspired by [14].

3.1 Presentation of the Keller-Segel Model

This section describes the Keller-Segel model, which models the chemotactic dynamics of cell populations. We define the main variables as follows:

- u : cell density,
- φ : chemoattractant concentration,
- χ : chemical sensitivity constant.

The system of equations governing these dynamics is given by:

$$\frac{\partial u}{\partial t} - \nabla \cdot (\mu \nabla u - \chi u \nabla \varphi) = f_u, \quad (3.1a)$$

$$\frac{\partial \varphi}{\partial t} - \nabla \cdot (\nu \nabla \varphi) + a\varphi - bu = f_\varphi, \quad (3.1b)$$

where f_u and f_φ are source terms.

The system is complemented by natural boundary conditions for u and φ :

$$\nu \nabla \varphi \cdot n = 0, \quad \mu (\nabla u - \chi u \nabla \varphi) \cdot n = 0, \quad \text{on } \Gamma_N,$$

$$0\nu \nabla \varphi \cdot n + \frac{1}{\varepsilon_\varphi} \varphi = g_\varphi, \quad \mu (\nabla u - \chi u \nabla \varphi) \cdot n + \frac{1}{\varepsilon_u} u = g_u, \quad \text{on } \Gamma_R,$$

where g_φ and g_u are boundary functions, and $\varepsilon_u, \varepsilon_\varphi$ are regularization parameters.

To facilitate the numerical resolution of the coupled equations, we introduce two new auxiliary variables:

- j : cell flux,
- ψ : chemoattractant gradient.

These variables are defined by:

$$j + \mu \nabla u + \frac{\mu}{\nu} \chi u \psi = 0, \quad (3.2)$$

$$\psi + \nu \nabla \varphi = 0. \quad (3.3)$$

and so we can rewrite the Keller-Segel system 3.1a and 3.1b as:

$$\frac{\partial u}{\partial t} + \nabla \cdot j = f_u, \quad (3.4a)$$

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot \psi + a\varphi - bu = f_\varphi, \quad (3.4b)$$

The natural and Robin boundary conditions are reformulated as follows:

$$\psi \cdot n = 0, \quad \text{on } \partial\Omega, \quad j \cdot n = 0, \quad \text{on } \Gamma_N.$$

$$\psi \cdot n + \frac{1}{\varepsilon_\varphi} \varphi = g_\varphi^R, \quad j \cdot n + \frac{1}{\varepsilon_u} u = g_u^R, \quad \text{on } \Gamma_R.$$

3.2 Weak Formulation Adapted to HDG

To numerically solve the Keller-Segel system introduced in the previous section, we employ a Hybrid Discontinuous Galerkin (HDG) approach as described in the

section 2. Drawing inspiration from [14] and references therein, we select polynomial approximations of order $k \geq$ for all variables except the flux j , which is approximated with order $k - 1$.

Let us define $P_k(\mathcal{T}_h)$ and $P_k(\mathcal{E}_h)$ as the spaces of discontinuous polynomials of degree at most k , defined respectively on the elements of the mesh \mathcal{T}_h and on its skeleton \mathcal{E}_h :

$$P_k(\mathcal{T}_h) = \{p \in L^2(\Omega) : p|_K \in P_k \quad \forall K \in \mathcal{T}_h\},$$

$$P_k(\mathcal{E}_h) = \{p \in L^2(\mathcal{E}_h) : p|_e \in P_k \quad \forall e \in \mathcal{E}_h\}.$$

For all $q \in (P_k(\mathcal{T}_h))^2$ we have the following inverse inequality

$$h \|q \cdot \mathbf{n}\|_{\mathcal{E}_h}^2 \leq \|q\|_{\Omega}^2. \quad (3.5)$$

In order to write down the HDG formulation of the Keller-Segel system, we introduce the traces of u and φ on the skeleton \mathcal{E}_h as additional independent unknowns, which we denote by \hat{u} and $\hat{\varphi}$.

We begin by multiplying, respectively, the flux equations 3.2 and 3.3 by test functions $q \in (P_{k-1}(\mathcal{T}_h))^2$ and $\theta \in (P_k(\mathcal{T}_h))^2$ and substitute u and φ on $\partial\Omega$ by their traces $\hat{u} = u|_{\partial\mathcal{E}_h}$ and $\hat{\varphi} = \varphi|_{\partial\mathcal{E}_h}$.

$$(\alpha j + \beta \chi u \psi, q)_{\mathcal{T}_h} - (u, \nabla \cdot q)_{\mathcal{T}_h} + (\hat{u}, q \cdot n)_{\mathcal{E}_h} = 0, \quad (3.6)$$

$$(\beta \psi, \theta)_{\mathcal{T}_h} - (\varphi, \nabla \cdot \theta)_{\mathcal{T}_h} + (\hat{\varphi}, \theta \cdot n)_{\mathcal{E}_h} = 0, \quad (3.7)$$

where

$$\alpha = \mu^{-1}, \quad \beta = \nu^{-1}$$

So the equations 3.4a and 3.4b can be rewritten respectively as:

$$\begin{aligned} (f_u, w)_{\Omega} &= \left(\frac{\partial u}{\partial t}, w \right)_{\Omega} + (\nabla \cdot j, w)_{\mathcal{T}_h} \\ &= \left(\frac{\partial u}{\partial t}, w \right)_{\Omega} - (j, \nabla w)_{\mathcal{T}_h} + (j \cdot n, w)_{\mathcal{E}_h} \end{aligned} \quad (3.8)$$

$$\begin{aligned}
(f_\varphi, \tau)_\Omega &= \left(\frac{\partial \varphi}{\partial t} + \nabla \cdot \psi + a\varphi - bu, \tau \right)_{\mathcal{T}_h} \\
&= \left(\frac{\partial \varphi}{\partial t}, \tau \right)_\Omega + (\nabla \cdot \psi, \tau)_{\mathcal{T}_h} + (a\varphi, \tau)_\Omega - (bu, \tau)_\Omega \\
&= \left(\frac{\partial \varphi}{\partial t}, \tau \right)_\Omega - (\psi, \nabla \tau)_{\mathcal{T}_h} + (\psi \cdot n, \tau)_{\mathcal{E}_h} + (a\varphi, \tau)_\Omega - (bu, \tau)_\Omega \quad (3.9)
\end{aligned}$$

The flux continuity, together with the imposed boundary conditions, leads to the following:

$$\begin{aligned}
-\langle \psi \cdot n, \hat{\tau} \rangle_{\mathcal{E}_h} + \varepsilon_\varphi^{-1} \langle \hat{\varphi}, \hat{\tau} \rangle_{\Gamma_R} &= \langle g_R^\varphi, \hat{\tau} \rangle_{\Gamma_R}, \\
-\langle j \cdot n, \hat{w} \rangle_{\mathcal{E}_h} + \varepsilon_u^{-1} \langle \hat{u}, \hat{w} \rangle_{\Gamma_R} &= \langle g_R^u, \hat{w} \rangle_{\Gamma_R},
\end{aligned}$$

Let's define the following composite discrete spaces:

$$\begin{aligned}
V_h &= P_k(\mathcal{T}_h) \times P_k(\mathcal{E}_h) \times (P_{k-1}(\mathcal{T}_h))^2, \\
S_h &= P_k(\mathcal{T}_h) \times P_k(\mathcal{E}_h) \times (P_k(\mathcal{T}_h))^2.
\end{aligned}$$

Following the approach in [15], the numerical fluxes on the mesh skeleton \mathcal{T}_h are expressed as:

$$\begin{aligned}
\hat{j}^K \cdot n^K &= j^K \cdot n^K + \gamma_u(u^K - \hat{u}^K), \\
\hat{\psi}^K \cdot n^K &= \psi^K \cdot n^K + \gamma_\varphi(\varphi^K - \hat{\varphi}^K),
\end{aligned}$$

where γ_u and γ_φ are stabilization parameters and are appropriately chosen. Following [13] p.348, we set:

$$\gamma_u = \frac{\tau_D}{h} + \tau_C$$

with $\tau_C = \max(\psi \cdot n, 0)$ and $\tau_D = 1$ or 10 a constant.

Problem: The global discrete problem is to find $U_h = (u_h, \hat{u}_h, j_h) \in V_h$ and $\Phi_h = (\varphi_h, \hat{\varphi}_h, \psi_h) \in S_h$ such that, for all $W = (w, \hat{w}, q) \in V_h$ and $\Theta = (\tau, \hat{\tau}, \theta) \in S_h$, the following equations hold:

$$\left(\frac{\partial u_h}{\partial t}, w\right)_\Omega + (\hat{j}_h \cdot n, w)_{\mathcal{E}_h} - (j_h, \nabla w)_{\mathcal{T}_h} = (f_u, w)_\Omega, \quad (3.10)$$

$$\left(\frac{\partial \varphi_h}{\partial t}, \tau\right)_\Omega + (\hat{\psi}_h \cdot n, \tau)_{\mathcal{E}_h} - (\psi_h, \nabla \tau)_{\mathcal{T}_h} + (a\varphi_h, \tau)_\Omega - (bu_h, \tau)_\Omega = (f_\varphi, \tau)_\Omega, \quad (3.11)$$

$$(\beta\psi_h, \theta)_\Omega - (\varphi_h, \nabla \cdot \theta)_{\mathcal{T}_h} + (\widehat{\varphi}_h, \theta \cdot n)_{\mathcal{E}_h} = 0, \quad (3.12)$$

$$(\alpha j_h + \beta \chi u_h \psi_h, q)_\Omega - (u_h, \nabla \cdot q)_{\mathcal{T}_h} + (\widehat{u}_h, q \cdot n)_{\mathcal{E}_h} = 0, \quad (3.13)$$

$$-\langle \psi_h \cdot n, \widehat{\tau} \rangle_{\mathcal{E}_h} + \varepsilon_\varphi^{-1} \langle \widehat{\varphi}_h, \widehat{\tau} \rangle_{\Gamma_R} = \langle g_R^\varphi, \widehat{\tau} \rangle_{\Gamma_R}, \quad (3.14)$$

$$-\langle j_h \cdot n, \widehat{w} \rangle_{\mathcal{E}_h} + \varepsilon_u^{-1} \langle \widehat{u}_h, \widehat{w} \rangle_{\Gamma_R} = \langle g_R^u, \widehat{w} \rangle_{\Gamma_R}. \quad (3.15)$$

3.3 Remarks and properties

Remark 3.1. For simplicity, we used order k polynomials to approximate the trace unknown \widehat{u} . We could instead use the cheaper order $k-1$ polynomial space $P_{k-1}(\mathcal{E}_h)$, as proposed in [15]. In such a case we need to resort to the Lehrenfeld-Schöberl stabilization, that is, the numerical flux must be defined as

$$\widehat{j} \cdot n = j \cdot n + h^{-1} \Pi_{k-1}^\partial(u - \widehat{u}).$$

This implies that the numerical flux is a polynomial of degree $k-1$ and not k , independently of the order of the discretization for u .

Remark 3.2. To introduce upwind in the scheme, we can modify the definition of the numerical fluxes and traces. When integrating by part in the drift term one can write:

$$\int_T \nabla \cdot (\mathbf{b}u)v = - \int_T \mathbf{b}u \cdot \nabla v + \int_{\partial T} u^{upw} \mathbf{b} \cdot n v$$

where

$$u^{upw} = \begin{cases} \widehat{u} & \mathbf{b} \cdot n \leq 0 \quad (\text{inflow}) \\ u & \mathbf{b} \cdot n > 0 \quad (\text{outflow}). \end{cases}$$

Then (3.6) becomes

$$\sum_K \left[(\alpha j + \beta \chi u \psi, q)_K - (u, \nabla \cdot q)_K + (\widehat{u}, q \cdot n^K)_{\partial K_{in}} + (u, q \cdot n^K)_{\partial K_{out}} \right] = 0 \quad (3.16)$$

Theorem 3.1. Let $\gamma_u = h^{-1}$ and $\gamma_\varphi = 1$. Then, there exists h^* such that for all $h < h^*$

it holds that

$$\begin{aligned} \|u - u_h\|_{L^\infty(0,T;L^2(\Omega))} + h^{-1/2}\|u_h - \hat{u}_h\|_{L^2(0,T;L^2(\mathcal{E}_h))} &\leq h^{k+1-\varepsilon} \\ \|j - j_h\|_{L^2(0,T;L^2(\Omega))} &\leq h^k \\ \|\varphi - \varphi_h\|_{L^\infty(0,T;L^2(\Omega))} + \beta\|\psi - \psi_h\|_{L^2(0,T;L^2(\Omega))} + \|\varphi_h - \hat{\varphi}_h\|_{L^2(0,T;L^2(\mathcal{E}_h))} &\leq h^{k+1-\varepsilon} \end{aligned}$$

for $\varepsilon > 0$ arbitrarily small, the constant in the inequality depending on ε .

The error estimate can be derived closely following the proof strategy presented in [15], with some modifications to account for the time derivative in equation (3.1b), the presence of mixed Neumann / Robin boundary conditions, and the specific definition of the flux variable j . Since the proof is rather technical and involves several subtle points, a detailed version is provided in [14] to clarify how these differences influence the final result.

3.3.1 The lowest order case

Let us examine the particular case $k = 1$, which corresponds to the lowest-order HDG discretization. Here, the flux variable j is approximated by piecewise constants. In this setting, the discrete equation (3.10) simplifies to

$$\left(\frac{\partial u_h}{\partial t}, w \right)_\Omega + \gamma_u(u_h - \hat{u}_h, w)_{\mathcal{E}_h} = (f_u, w)_\Omega, \quad (3.17)$$

where u_h is determined in terms of its trace \hat{u}_h .

The error estimates provided by Theorem 3.1 in this lowest order case yield

$$\begin{aligned} \|u - u_h\|_{L^\infty(0,T;L^2(\Omega))}^2 + \|j - j_h\|_{L^2(0,T;L^2(\Omega))}^2 + h^{-1}\|u_h - \hat{u}_h\|_{L^2(0,T;L^2(\mathcal{E}_h))}^2 &\leq h^4, \\ \|\varphi - \varphi_h\|_{L^\infty(0,T;L^2(\Omega))}^2 + \beta\|\psi - \psi_h\|_{L^2(0,T;L^2(\Omega))}^2 + \|\varphi_h - \hat{\varphi}_h\|_{L^2(0,T;L^2(\mathcal{E}_h))}^2 &\leq h^4. \end{aligned}$$

These results confirm that, even at the lowest polynomial degree, the HDG method achieves optimal convergence rates for both the solution and its fluxes.

4 Application to Organ-on-Chip OOC

This chapter adapts the mathematical and numerical framework developed in Chapters 2 and 3 to the specific context of Organ-on-Chip (OOC) systems, focusing on cancer-on-chip applications such as Tumor-on-Chip (TOC). OOCs are microfluidic platforms designed to mimic physiological microenvironments, consisting of chambers where cellular interactions occur and microchannels that facilitate the transport of cells and chemicals, see figures 4.1, 4.2 and 4.3.

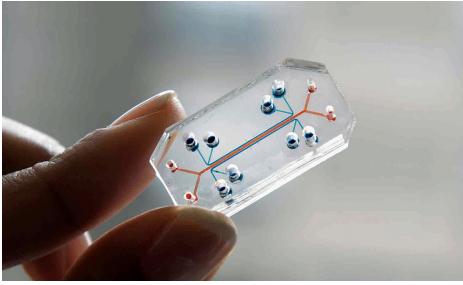


Figure 4.1: Microfluidic chip

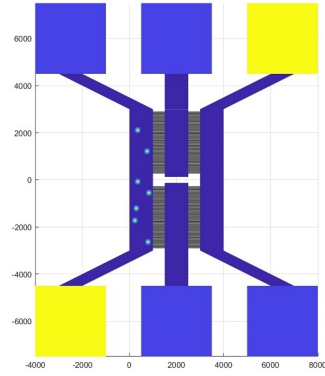


Figure 4.2: Microfluidic chip: experimental view.

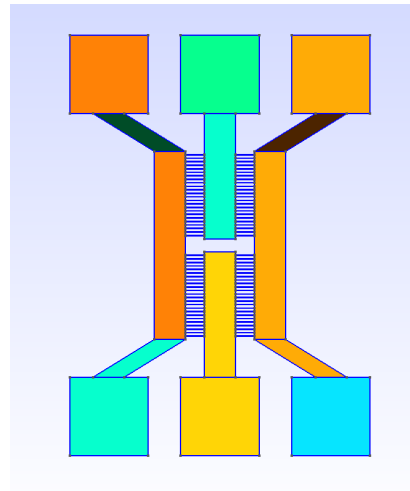
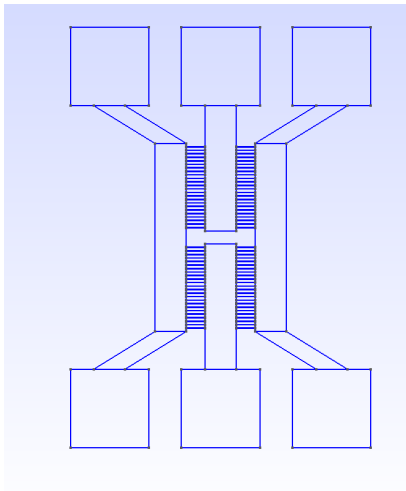


Figure 4.3: Geometry and mesh representation of the microfluidic chip used in Organ-on-Chip experiments.

We explore two configurations: a single-chamber model and a two-chamber model connected by a channel, building on the Keller-Segel chemotaxis framework to account for immune-tumor dynamics. The Hybrid Discontinuous Galerkin (HDG) method is employed to solve these models, particularly the two-chamber model, which presents unique challenges due to its hybrid 2D-1D geometry. The HDG method is particularly well-suited for this application, as it allows for the efficient handling of discontinuities in the solution across the chambers and channels, while also providing a robust framework for addressing the complex boundary conditions and interactions between the different components of the OOC system. The implementation of the HDG method in the Feel++ library is detailed (see [4]), showcasing its capabilities in solving the OOC models.

4.1 General Mathematical Model of OOC

4.1.1 Single-Chamber Model

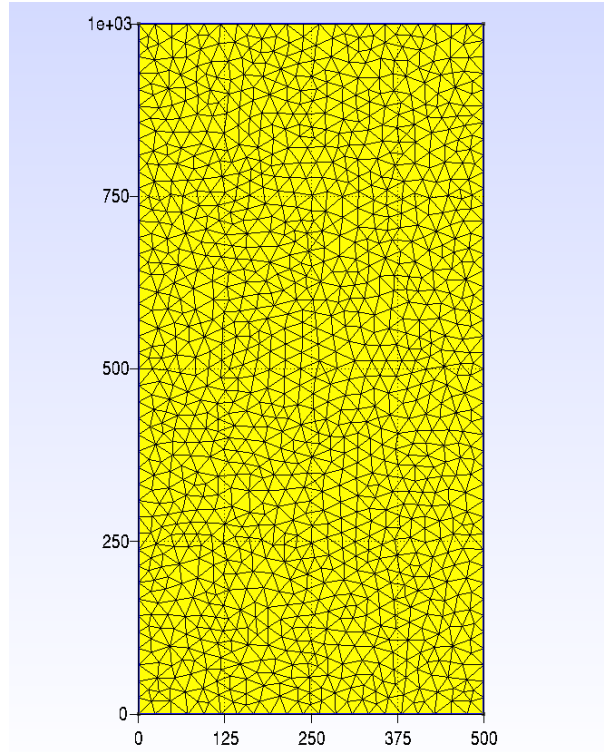


Figure 4.4: Schematic representation of a single-chamber Organ-on-Chip (OOC) device. The chamber is modeled as a rectangular domain where immune and tumor cells interact, and chemoattractants are produced and diffuse.

We begin with a simplified representation of a cancer-on-chip device, focusing

on a single chamber denoted as a rectangular domain, figure 4.4:

$$\Omega = [0, L_x] \times [0, L_y],$$

. This toy model captures the interactions between immune and tumor cells, as well as the chemoattractants they produce.

The key variables in this model are:

- $u(x, y, t)$: density of immune system cells,
- $v(x, y, t)$: density of tumor cells,
- $\varphi(x, y, t)$: chemoattractant concentration secreted by immune cells,
- $\psi(x, y, t)$: chemoattractant concentration secreted by tumor cells.

The governing equations within Ω are formulated as a system of coupled reaction–diffusion–chemotaxis equations:

$$\partial_t v - \nabla \cdot (\nu_v \nabla v) + \lambda(\varphi)v = 0, \quad \text{in } \Omega, \quad (4.1)$$

$$\partial_t u - \nabla \cdot (\nu_u \nabla u - \chi u \nabla \psi) = 0, \quad \text{in } \Omega, \quad (4.2)$$

$$\partial_t \varphi - \nabla \cdot (\nu_\varphi \nabla \varphi) + \beta \varphi - \alpha u = 0, \quad \text{in } \Omega, \quad (4.3)$$

$$\partial_t \psi - \nabla \cdot (\nu_\psi \nabla \psi) + a_\psi \psi - b_\psi v = 0, \quad \text{in } \Omega, \quad (4.4)$$

with homogeneous Neumann boundary conditions on $\partial\Omega$. Here,

$$\lambda(\phi) = \frac{k_{\varphi_1} \phi}{k_{\varphi_2} + \phi}$$

represents the killing rate of tumour cells v caused by φ , and chemotaxis enters Eq. (4.2) via $\chi u \nabla \psi$.

4.1.1.1 Coefficients and Modeling Choices

In the one chamber present model, the chemotactic sensitivity χ is taken as a positive constant (see table 4.1), meaning that immune cells respond linearly to gradients of the chemoattractant ψ produced by tumor cells.

In more complex biological contexts, χ may depend nonlinearly on the chemoattractant concentration ψ , reflecting saturation effects or other regulatory mechanisms, we will develop this form in the section 4.1.2.

Symbol	Description	Value	Units	Refs
D_v	Tumor-cell diffusivity	5.6×10^1	$\mu\text{m}^2/\text{s}$	[20]
D_u	Immune-cell diffusivity	2.0×10^2	$\mu\text{m}^2/\text{s}$	[20]
D_φ	Diffusivity of φ	9.0×10^2	$\mu\text{m}^2/\text{s}$	[20]
D_ψ	Diffusivity of ψ	9.0×10^2	$\mu\text{m}^2/\text{s}$	[20]
χ	Chemotactic sensitivity	$10^6\text{-}10^{10}$	$M\mu\text{m}^2/(\text{s} \cdot \text{cell})$	[21]
α	Production rate of φ per immune cell	1.0×10^{-1}	$\text{s}^{-1} \cdot \text{cell}^{-1}$	[21]
β	Decay rate of φ	1.0×10^{-4}	s^{-1}	[21]
a_ψ	Decay rate of ψ	1.0×10^{-4}	s^{-1}	[21]
b_ψ	Production rate of ψ per tumor cell	1.0×10^{-1}	$\text{s}^{-1} \cdot \text{cell}^{-1}$	[21]
$k_{\varphi_1}, k_{\varphi_2}$	Killing parameters	1.0, 1.0	—	[20]
k_1	cellular drift velocity	1.0	$M \cdot \mu\text{m}^2/\text{s}$	[20]
k_2	receptor dissociation constant	1.0	M	[20]
k_3	drift velocity	1.56×10^{10}	$M \cdot \mu\text{m}^2/\text{s}$	[20]
L_x, L_y	chambers dimensions	500, 1000	μm	[19]
c_L	Channel length	600	μm	[19]
c_w	Channel width	12	μm	[19]
c_d	Width between channels	100	μm	[19]

Table 4.1: Model parameters used in Experiment 1.

4.1.1.2 Explanation of the Model

This mathematical model describes the interactions between tumor and immune cells, as well as the influence of chemical messengers (chemoattractants) they emit within a microfluidic environment.

- **Cellular dynamics and Dispersion:** Both tumor cells (v) and immune cells (u) undergo diffusion, representing random movement within the chamber. Additionally, immune cells exhibit *chemotaxis*, migrating preferentially toward higher concentrations of the chemoattractant (ψ) produced by tumor cells, which guides them toward tumor sites.
- **Chemical Signaling:** Each cell population secretes its own chemoattractant: immune cells release φ , while tumor cells release ψ . These substances diffuse and degrade over time, enabling long-range communication: tumor cells signal their presence via ψ , and immune cells respond by producing φ .
- **Cellular Interactions and Cytotoxicity:** When immune cells reach tumor-dense regions, they increase the secretion of φ . The local concentration of φ enhances the tumor cell death rate, as modeled by the nonlinear function $\lambda(\varphi)$ in the tumor equation. This mechanism captures the intensified immune response in areas where both cell types coexist.

- **Production and Degradation Dynamics** The rates at which chemoattractants are produced and decay (parameters $\alpha, b_\psi, \beta, a_\psi$) determine how quickly chemical gradients form and dissipate, influencing the timing and extent of immune-tumor interactions.

The model assumes impermeable boundaries (homogeneous Neumann conditions), preventing any exchange of cells or chemicals with the environment. Initially, tumor and immune cells are introduced as localized populations, with both φ and ψ set to zero throughout the domain.

This framework simulates the detection and migration of immune cells toward tumors, the mutual influence of cell populations through chemical signaling, and the resulting impact on tumor control. It provides a computational platform for exploring biological mechanisms and evaluating therapeutic strategies prior to experimental studies.

4.1.2 Two-Chamber Model with Connecting channels

This section extends the single-chamber OOC model to a two-chamber configuration connected by a channel, reflecting more complex biological interactions such as those in Tumor-on-Chip (TOC) systems. The domain consists of two two-dimensional chambers, Ω^1 and Ω^2 , linked by a one-dimensional channel ω , with interfaces $\gamma^1 \subset \partial\Omega^1$ and $\gamma^2 \subset \partial\Omega^2$ at the connection points \mathbf{x}^1 and \mathbf{x}^2 (see Figure 4.5). This hybrid 2D-1D geometry allows for the transport of cells and chemicals between chambers, mimicking physiological microenvironments.

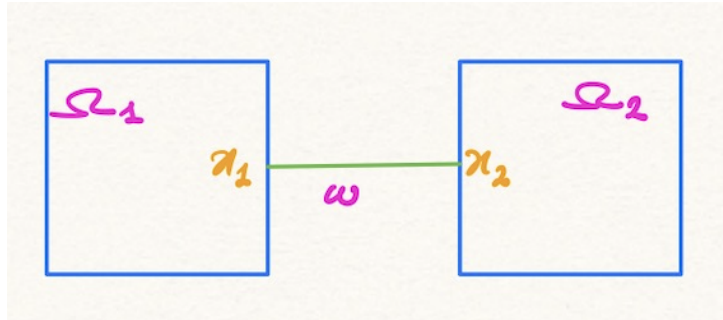


Figure 4.5: Schematic of the two-chamber OOC model with a connecting channel c .

The model variables include the densities of immune cells (u^1, u^2, u^c) and tumor cells (v^1, v^2, v^c), as well as the concentrations of immune chemoattractants ($\varphi^1, \varphi^2, \varphi^c$) and tumor chemoattractants (ψ^1, ψ^2, ψ^c) in the respective domains.

The governing equations in the chambers ($\Omega^*, * = 1, 2$) are:

$$\partial_t v^* - \nabla \cdot (\nu_v^* \nabla v^*) + \lambda^*(\varphi^*)v^* = 0, \quad \text{in } \Omega^*, \quad (4.5)$$

$$\partial_t u^* - \nabla \cdot (\nu_u^* \nabla u^* - \chi u^* \nabla \psi^*) = 0, \quad \text{in } \Omega^*, \quad (4.6)$$

$$\partial_t \varphi^* - \nabla \cdot (\nu_\varphi^* \nabla \varphi^*) + \beta \varphi^* - \alpha u^* = 0, \quad \text{in } \Omega^*, \quad (4.7)$$

$$\partial_t \psi^* - \nabla \cdot (\nu_\psi^* \nabla \psi^*) + a_\psi^* \psi^* - b_\psi^* v^* = 0, \quad \text{in } \Omega^*, \quad (4.8)$$

and in the channel (ω):

$$\partial_t v^c - \partial_s (\nu_v^c \partial_s v^c) + \lambda^c(\varphi^c)v^c = 0, \quad \text{in } \omega, \quad (4.9)$$

$$\partial_t u^c - \partial_s (\nu_u^c \partial_s u^c - \chi \partial_s \psi^c u^c) = 0, \quad \text{in } \omega, \quad (4.10)$$

$$\partial_t \varphi^c - \partial_s (\nu_\varphi^c \partial_s \varphi^c) + \beta \varphi^c - \alpha u^c = 0, \quad \text{in } \omega, \quad (4.11)$$

$$\partial_t \psi^c - \partial_s (\nu_\psi^c \partial_s \psi^c) + a_\psi^c \psi^c - b_\psi^c v^c = 0, \quad \text{in } \omega, \quad (4.12)$$

where ν^* , λ^* , a_ψ^* , b_ψ^* are domain-specific coefficients.

Neumann boundary conditions are imposed on $\partial\Omega^* \setminus \gamma^*$:

$$\begin{cases} \nu_v^* \nabla v^* \cdot \mathbf{n}^* = 0, \\ (\nu_u^* \nabla u^* - \chi u^* \nabla \psi^*) \cdot \mathbf{n}^* = 0, \\ \nu_\varphi^* \nabla \varphi^* \cdot \mathbf{n}^* = 0, \\ \nu_\psi^* \nabla \psi^* \cdot \mathbf{n}^* = 0. \end{cases} \quad (4.13)$$

The model accounts for diffusion, chemotaxis, and reaction terms, with the channel facilitating the exchange of cells and chemicals. Interface conditions at γ^1 and γ^2 (e.g., continuity of fluxes or concentrations) are critical but require further specification for numerical implementation, as discussed in the following steps.

4.1.2.1 Description of the Model

- **Spatial Coupling:** The model captures the transport of immune and tumor cells, as well as their chemoattractants, between the two chambers via the channel. This spatial coupling is crucial for understanding how localized interactions in one chamber can influence dynamics in another.
- **Channel Dynamics:** The channel is modeled as a one-dimensional domain, allowing for efficient transport of cells and chemicals. The equations in the channel account for diffusion and advection, reflecting how substances move through the connecting structure.

- **Chemical Signaling Across Chambers:** Chemoattractants produced in one chamber can diffuse into the other, influencing cell densities and migration patterns. This interaction is modeled through the coupled equations in both chambers and the channel.
- **Boundary Conditions:** The model incorporates appropriate boundary conditions at the interfaces between chambers and the channel, ensuring continuity of cell densities and chemical concentrations across these boundaries.
- **Parameterization:** The model parameters, such as diffusivities and production rates, can be adjusted to reflect specific biological contexts, allowing for flexibility in modeling different types of OOC systems, see Table 4.1.

4.1.2.2 Chemotactic Sensitivity and killing functions

In its simplest and most common form, the chemotactic sensitivity function is defined as $\chi = k$ for a constant $k \in \mathbb{R}$. This choice models only the directed migration along the gradient of the chemoattractant φ , without accounting for other possible biological mechanisms.

In the following, we outline several forms of chemotactic terms that model cell migration under different biological assumptions, including self-generated gradients, attractive and repulsive cell interactions, and crowding effects. Cell movement is typically described as a combination of random diffusion and a directed chemotactic response, where cells move up or down the gradient of a chemical signal φ . These functions can be dependent on the chemoattractant or the density of the cells themselves:

- **Basic model:** directional movement up a spatial gradient of chemoattractant

$$\chi := k_3$$

with drift velocity k_3 .

- **Receptor saturation:** dependence on concentration of chemoattractant in a cell's local environment...

$$\chi(\varphi) := k_1 \frac{1}{(k_2 + \varphi)^2}$$

with k_1 k_2 representing the the cellular drift velocity and receptor dissociation constant respectively.

- **Overcrowding:** ability to move freely reduces at high densities as cell density approaches maximum value

$$\chi(u, \varphi) := k_1 \frac{1}{(k_2 + \varphi)^2} \left(1 - \frac{u}{u_{\max}} \right)$$

where the cell's ability to move freely reduces at high cell densities as they approach the maximum cell density u_{\max} .

- **Killing function:** for the tumor cells, the killing function is defined as

$$\lambda(\varphi) := \frac{k_{\varphi_1} \varphi}{k_{\varphi_2} + \varphi}$$

where k_{φ_1} and k_{φ_2} are parameters that control the killing rate of tumor cells by immune chemoattractants.

4.2 Setup and results on the the numerical approximation of the OOC

4.2.1 Numerical Setup and Results for the Single-Chamber Model

The numerical experiments for the single-chamber model are conducted using the `feelpp` toolbox `coefficientformpdes` application, which is suitable for solving PDEs in coefficient form.

The generic form of a PDE considered in *feelpp*[2] is:

$$d \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f \quad \text{in } \Omega.$$

So, for the OOC model with a single chamber(Important for the configuration file):

OOC 1	feelpp
$\partial_t v - \nabla \cdot \nu_v \nabla v + \lambda(\varphi)v = 0$	$v + \nabla \cdot (-c \nabla v) + av = 0$ $d = 1, c = \nu_v \quad a = \lambda(\varphi)$
$\partial_t u - \nabla \cdot (\nu_u \nabla u - \chi u \nabla \varphi) = 0,$	$\partial_t u + \nabla \cdot (-c \nabla \varphi - \alpha u) = 0$ $d = 1, c = \nu_u \quad \alpha = -\chi \nabla \varphi$
$\partial_t \psi - \nabla \cdot \nu_\psi \nabla \psi + \beta_\psi \psi - \alpha_\psi u = 0$	$\partial_t \psi + \nabla \cdot (-c \nabla \psi) + a\psi = 0$ $d = 1, c = \nu_\psi \quad a = \beta_\psi \quad f = \alpha_\psi u$
$\partial_t \varphi - \nabla \cdot \nu_\varphi \nabla \varphi + a_\varphi \varphi - b_\varphi u = 0$	$\partial_t \varphi + \nabla \cdot (-c \nabla \varphi) + a\varphi = 0$ $d = 1, c = \nu_\varphi \quad a = a_\varphi \quad f = b_\varphi v$

We also define the parameters in the configuration file, which are used to set the coefficients of the PDEs. The values are defined in table 4.1. For the domain Ω , we used a rectangular domain discretized with a mesh of size $h = 5 \mu m$.

4.2.1.1 Initial and Boundary Conditions

The initial conditions used in the simulations are gaussians and defined as follows (matching the configuration file) $\forall (x, y) \in \Omega$:

$$\begin{aligned}
v(x, y, 0) &= 0.001 \exp\left(-\frac{x^2 + (y - 500)^2}{1000}\right) + 0.001 \exp\left(-\frac{x^2 + (y - 1000)^2}{1000}\right) \\
&\quad + 0.001 \exp\left(-\frac{x^2 + y^2}{1000}\right), \\
u(x, y, 0) &= 0.005 \exp\left(-\frac{(x - 400)^2 + (y - 500)^2}{1000}\right) + 0.004 \exp\left(-\frac{(x - 400)^2 + (y - 900)^2}{1000}\right) \\
&\quad + 0.002 \exp\left(-\frac{(x - 400)^2 + y^2}{1000}\right), \\
\varphi(x, y, 0) &= 0, \\
\psi(x, y, 0) &= 0.
\end{aligned}$$

Homogeneous Neumann boundary conditions are imposed on all boundaries for all variables v , u , φ , and ψ .

Initial Distributions: The initial distributions of the variables in the single-chamber OOC model are shown in Figure 4.6. To better observe the dynamics of cellular interactions, the tumor cell density v is initialized with three Gaussian distributions, representing tumor clusters at different locations in the chamber. Similarly, the immune cell density u is initialized with three Gaussian distributions, representing immune cell populations initially present in the chamber. The chemoattractant concentrations φ and ψ are initialized to zero, indicating that no chemoattractants are present at the start of the simulation.

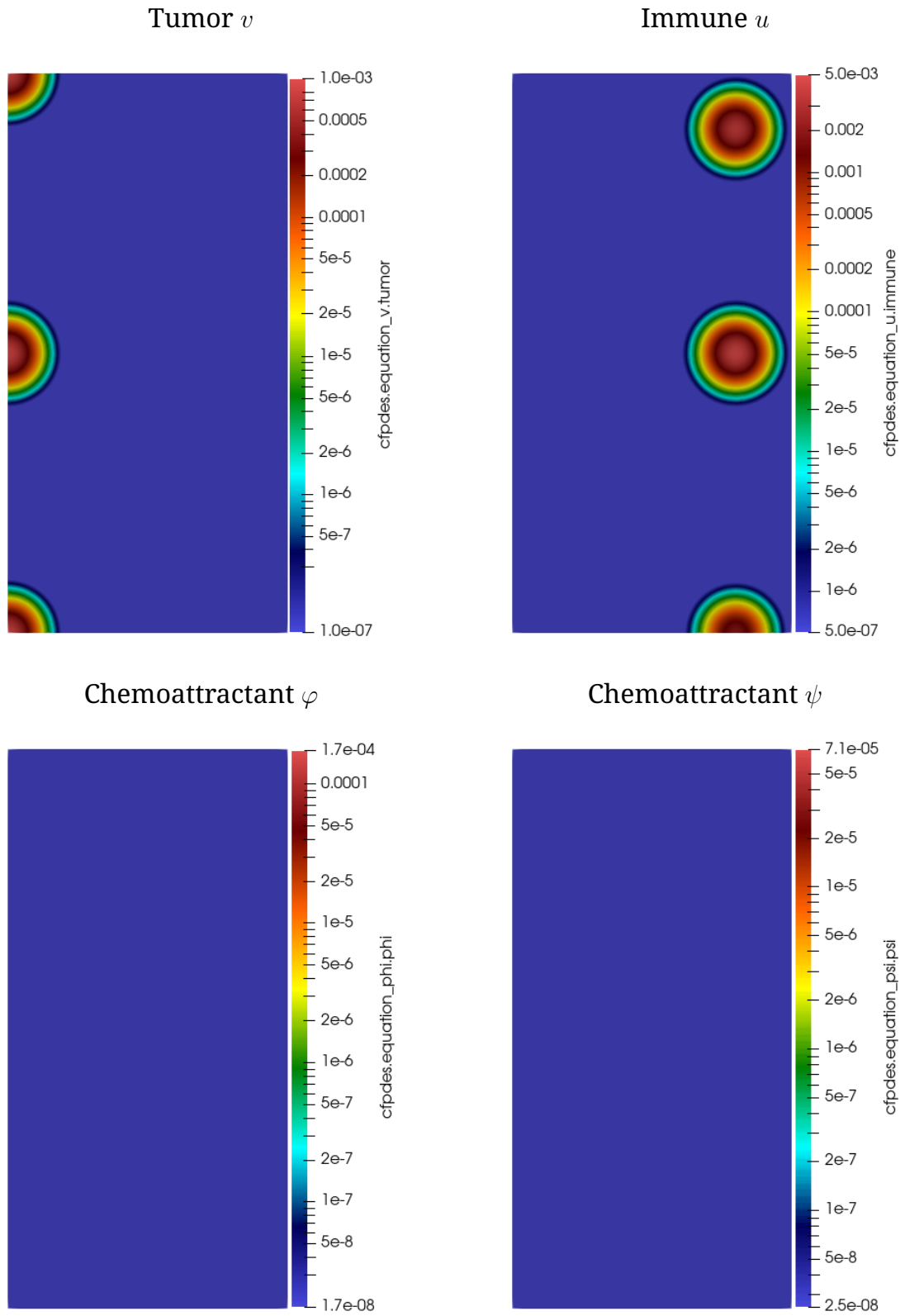


Figure 4.6: Initial distributions of the variables in the single-chamber OOC model.

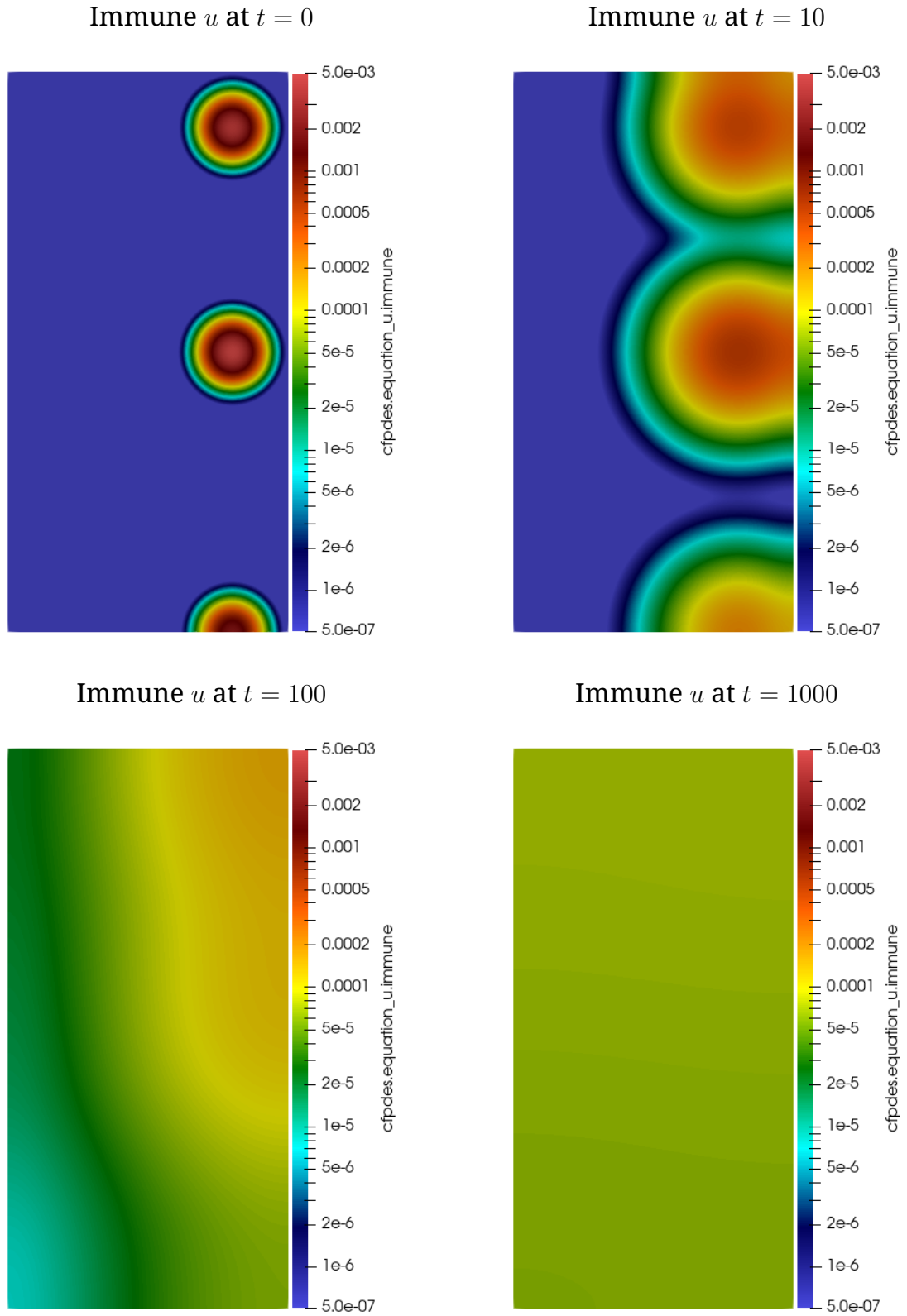


Figure 4.7: Evolution of the immune cell density u in the single-chamber OOC model at different time steps.

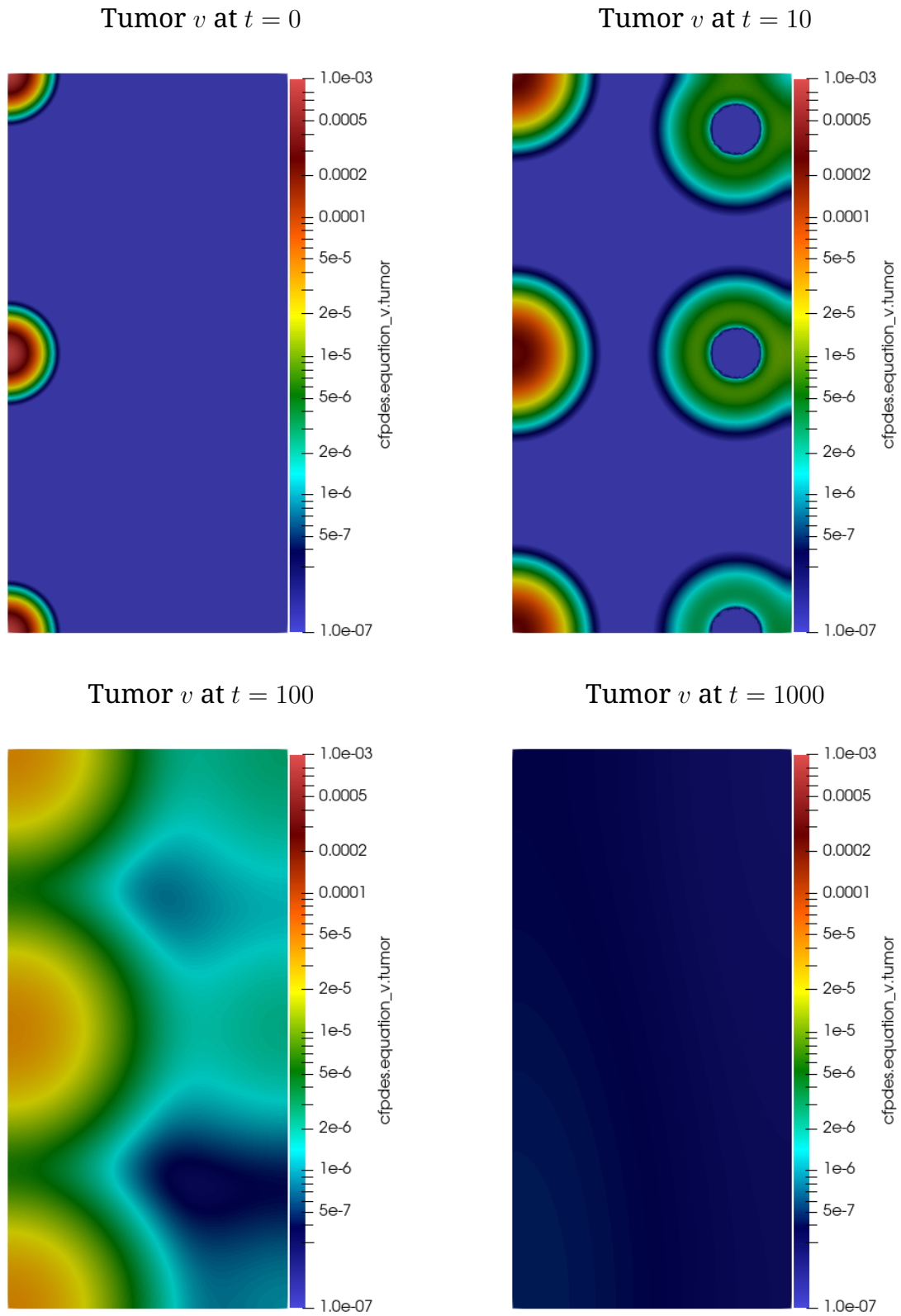


Figure 4.8: Evolution of the tumor cell density v in the single-chamber OOC model at different time steps.

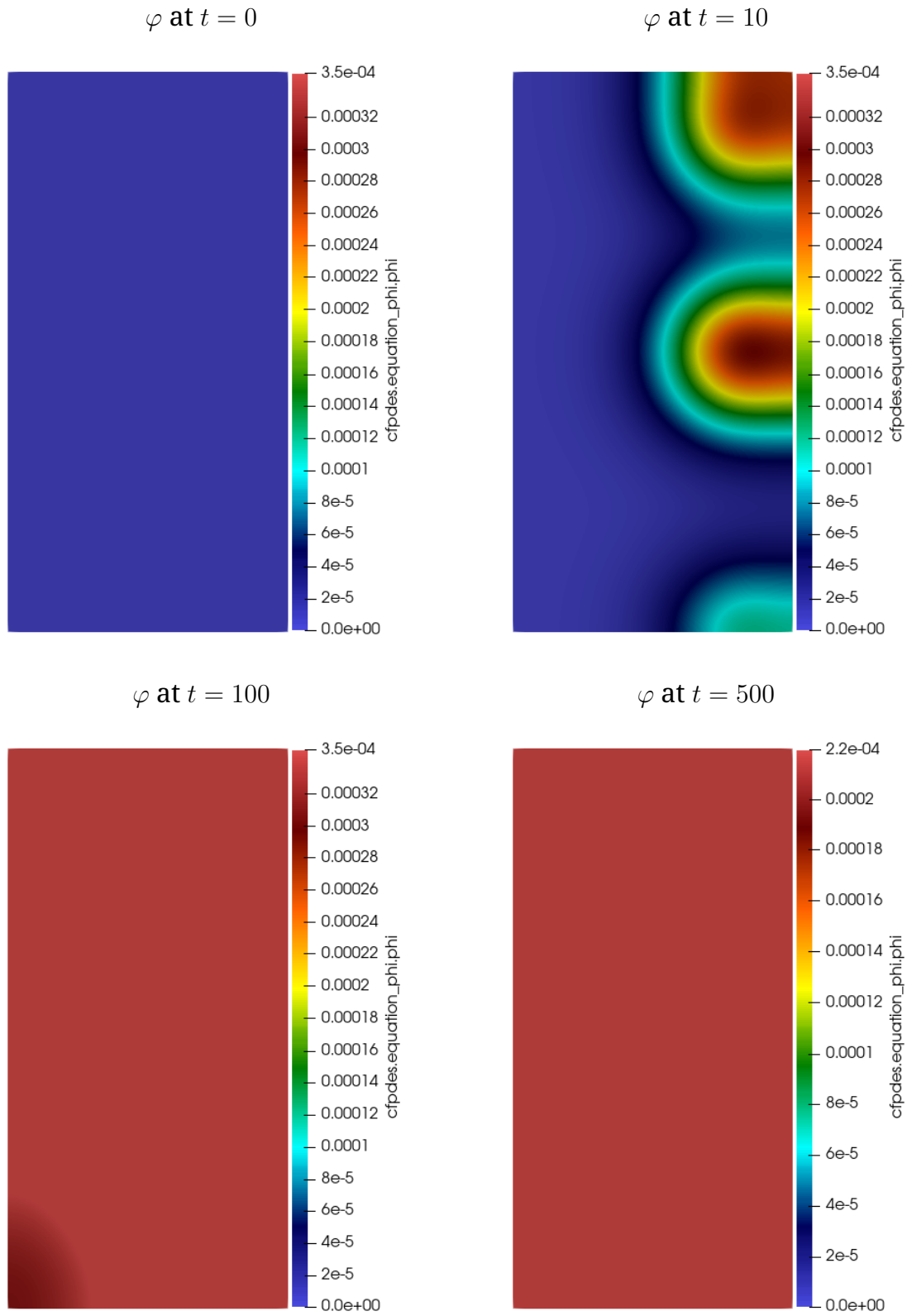


Figure 4.9: Evolution of the chemoattractant density ϕ in the single-chamber OOC model at different time steps.

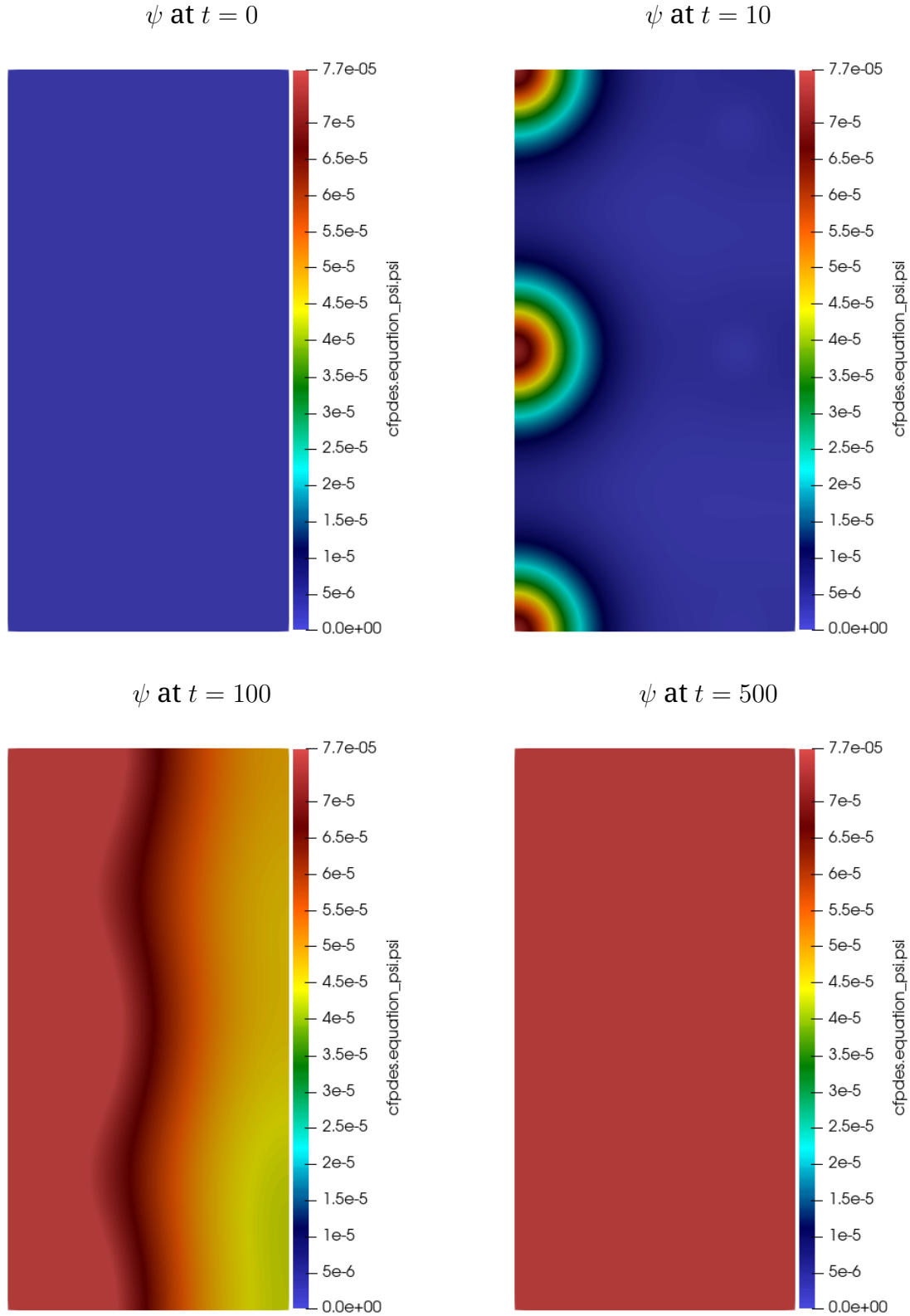


Figure 4.10: Evolution of the chemoattractant density ψ in the single-chamber OOC model at different time steps.

Comments on the evolution of the variables: Figure 4.8 shows how the tumor cell density evolves in the microfluidic chamber. At the initial time ($t = 0$),

there are three distinct clusters with relatively high concentrations (around 10^{-3}). Early on ($t = 10$), these clusters begin to expand radially while maintaining their original shape, creating visible diffusion zones.

By $t = 100$, the shapes become more complex. The interaction between diffusion and the killing function $\lambda(\varphi)$ becomes noticeable, especially near regions with high immune cell concentration. At the final stage ($t = 1000$), the initial clusters have spread out and formed a more uniform, lower-concentration distribution, suggesting that the immune response (modeled by $\lambda(\varphi)$) has successfully controlled the tumor.

Figure 4.7 illustrates how immune cell density changes over time. At $t = 0$, immune cells are uniformly distributed with low concentrations (around 10^{-5}). By $t = 10$, they start migrating toward tumor areas in response to the chemical signal ψ .

At $t = 100$, immune cells accumulate strongly around tumor clusters, showing an effective chemotactic response. Finally, at $t = 1000$, they remain concentrated in those regions, suggesting that they have established long-term immune surveillance and that the immune response is still active.

The figures 4.9 and 4.10 show the evolution of the chemoattractant concentrations φ and ψ . Initially, both are zero, but as the simulation progresses, they develop spatial patterns influenced by the distributions of tumor and immune cells. The chemoattractant φ produced by immune cells increases in regions with high immune cell density, while ψ produced by tumor cells increases near tumor clusters. They respectively reach maximum values around 10^{-4} at $t = 500$.

4.2.1.2 Quantitative Analysis and Parameter Sensitivity Study

To better understand the influence of model parameters on the system dynamics, we conducted systematic parameter sweeps focusing on two key parameters: the chemotactic sensitivity χ and the parameter k_1 in the killing function λ . These analyses provide quantitative insights into how parameter variations affect the temporal evolution of tumor and immune cell populations, complementing the spatial distribution analysis presented above.

The parameter sweep study examines:

- **Chemotactic sensitivity χ :** Values of 10^4 , 10^6 , and 10^8 were tested to assess how immune cell chemotactic response strength affects tumor control efficiency.
- **Killing parameter k_1 :** Values of 0.1, 1.0, and 10.0 were examined to understand the impact of tumor cells killing dynamics on the overall system behavior.

For each parameter combination, we analyze the temporal evolution of mean concentrations, spatial variability (min/max bounds), and integrated quantities over the simulation domain. The integrated quantities represent the total amount of each species in the system and provide a measure of the overall population dynamics.

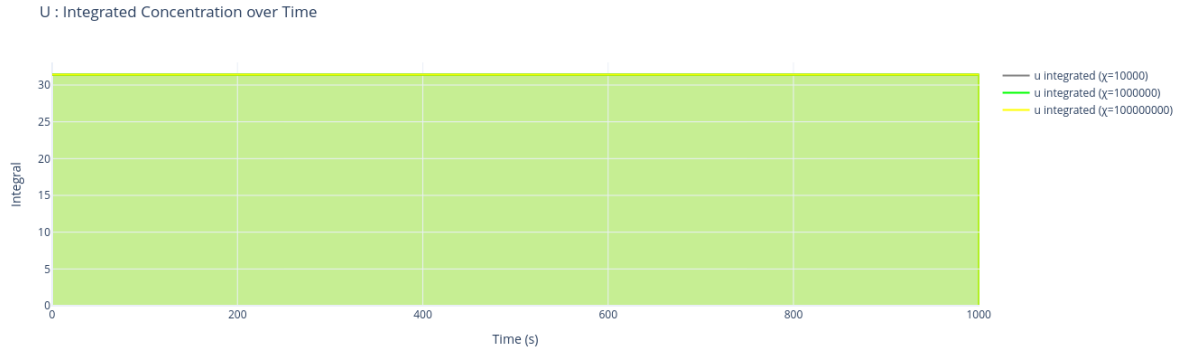


Figure 4.11: Temporal evolution of immune cell concentrations for different chemotactic sensitivity values χ .

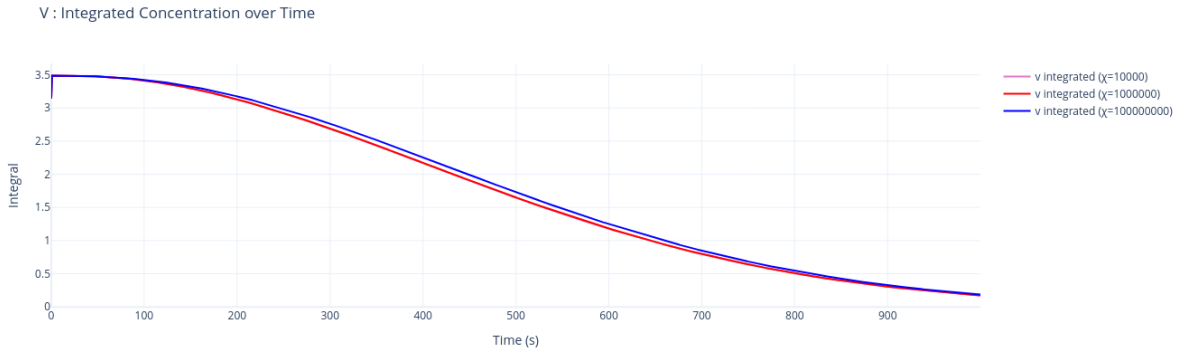


Figure 4.12: Evolution of tumor cell concentrations over time for different chemotactic sensitivity values χ .



Figure 4.13: Temporal evolution of chemoattractant concentrations φ and ψ for different chemotactic sensitivity values χ .

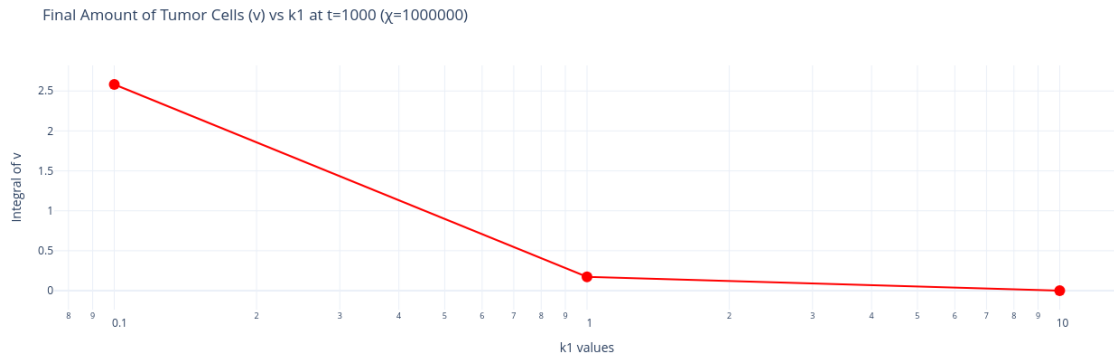


Figure 4.14: Final tumor cell amount (integrated concentration) at $t = 1000$ s, $\chi = 10^6$ for different killing rates k_1 .

The parameter sensitivity analysis reveals key insights into the dynamics of tumor-immune interactions: The figure 4.12 shows that the tumor cell population decreases over time for all χ values while the figure 4.11 shows that the immune cell population stay constant over time.

The Figure 4.14 shows that the final tumor cell amount decreases with high killing parameter k_1 , indicating that stronger immune responses lead to more effective tumor control.

Bibliographie

Bibliography

- [1] Feel++, *Documentation*, available at: <https://docs.feelpp.org/home/index.html>
- [2] Feel++, *Coefficient Form PDEs (cfpdes) Toolbox Manual*, available at: <https://docs.feelpp.org/toolboxes/latest/cfpdes/manual.html>
- [3] *Hybrid Discontinuous Galerkin (HDG) Method Documentation*, available at: <https://docs.feelpp.org/math/hdg/diffusion-reaction/index.html>
- [4] Feel++, *Organ-on-Chip Documentation*, available at: <https://feelpp.github.io/organ-on-chip/organ-on-chip/index.html>
- [5] Chen, Y., Cockburn, B., *Analysis of variable-degree HDG methods for convection-diffusion equations. Part I: general nonconforming meshes*, IMA J. Numer. Anal. 32(4), 1267–1293 (2012)
- [6] Chen, Y., Cockburn, B., *Analysis of variable-degree HDG methods for convection-diffusion equations. Part II: semimatching nonconforming meshes*, Math. Comput. 83, 87–111 (2014)
- [7] Silvia Bertoluzza, Giovanna Guidoboni, Romain Hild, Daniele Prada, Christophe Prud’Homme, et al., *A HDG Method for Elliptic Problems with Integral Boundary Condition: Theory and Applications*, Journal of Scientific Computing, 2023, 95 (1), pp.6. <https://doi.org/10.1007/s10915-023-02109-5>. <https://hal.inrae.fr/hal-04073316v1>
- [8] N. C. Nguyen, J. Peraire, and B. Cockburn, *An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations*, J. Comput. Phys., 228:3232–3254, 2009.
- [9] N. C. Nguyen, J. Peraire, and B. Cockburn, *An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations*, J. Comput. Phys., 228:8841–8855, 2009.

- [10] B. Cockburn, J. Guzmán, S.-C. Soon, and H. K. Stolarski, *An analysis of the embedded discontinuous Galerkin method for second-order elliptic problems*, SIAM J. Numer. Anal., 47(4):2686–2707, 2009.
- [11] B. Cockburn, J. Guzmán, and H. Wang, *Superconvergent discontinuous Galerkin methods for second-order elliptic problems*, Math. Comp., 78:1-24, 2009.
- [12] Qiu, W., Shi, K., *An HDG method for linear elasticity with strong symmetric stresses*, arXiv:1312.1407, submitted (2013).
- [13] Qiu, Weifeng, Shi, Ke, *An HDG Method for Convection Diffusion Equation*, Journal of Scientific Computing, 66, 2015. <https://doi.org/10.1007/s10915-015-0024-5>.
- [14] Bertoluzza, S., Bretti, G., Pennacchio, M., and Prud’homme, C., *HDG Discretization for the Keller–Segel Equation*, Journal of Computational Physics, 2023.
- [15] Gang Chen, Peter Monk, and Yangwen Zhang, *An HDG Method for the time-dependent drift–diffusion model of semiconductor devices*, Journal of Scientific Computing, 80(1):420–443, 2019. Publisher: Springer.
- [16] Keller, Evelyn F., and Segel, Lee A., *Initiation of slime mold aggregation viewed as an instability*, Journal of Theoretical Biology, 26(3):399–415, 1970.
- [17] Dejak, S.I., Egli, D., Lushnikov, P.M., and Sigal, I.M., *On blow up dynamics in the Keller-Segel model of chemotaxis*, St. Petersburg Mathematical Journal, 25, 2014. <https://www.ams.org/journals/spmj/2014-25-04/S1061-0022-2014-01306-4/home.html>
- [18] Yekaterina Epshteyn, *Discontinuous Galerkin methods for the chemotaxis and haptotaxis models*, Journal of Computational and Applied Mathematics, 224(1):168–181, 2009
- [19] Braun Elishan, Christian Thesis *Organs-On-Chips: mathematical modelling and parameter estimation*, PhD thesis, Università degli Studi Roma Tre, 2021.
- [20] Murray, J.D., *II. Spatial models and biomedical applications*, Springer, 2003.
- [21] Curk, T., Marenduzzo, D., Dobnikar, J., *Chemotactic Sensing towards Ambient and Secreted Attractant Drives Collective Behaviour of E. coli*, PLoS ONE 8(10): e74878, 2013. <https://doi.org/10.1371/journal.pone.0074878>

- [22] Leung, C.M., de Haan, P., Ronaldson-Bouchard, K. et al., *A guide to the organ-on-a-chip*, Nat Rev Methods Primers 2, 33 (2022). <https://doi.org/10.1038/s43586-022-00118-6>.