UNIVERSITÀ DEGLI STUDI ROMA TRE

DIPARTIMENTO DI MATEMATICA E FISICA

Organs-On-Chips: mathematical modelling and parameter estimation

Dottorato di ricerca in Matematica
XXXIII ciclo

Dottorando:
Elishan Christian Braun

Relatore di tesi:
Roberto Natalini
Gabriella Bretti
Coordinatore:
Angelo Felice Lopez

Contents

Part

Introduction

Introduction

During the last decades, mathematical modelling has become a powerful asset to explore biological phenomena and to detect obvious but also hidden mechanisms that might have not been known previously solely based on the experiment observations.
In a figuratively sense mathematics has become biology's next microscope, able to reveal otherwise hidden understandings.
Mathematical models have already been used centuries ago by Euler to model the age structure of stable populations, the logistic equation for population growth by Verhulst and the Lotka-Volterra predator-prey model that is even now frequently used to describe the dynamics of two interacting species [104, 105].
Most mathematical models' goal is to replicate the observation that has been made in experiments and to translate them to a more general setting to predict observational outcomes that otherwise would be impossible to obtain in reality.

The work carried out in the present thesis is motivated by the comprehension of the complex mechanisms behind the cell dynamics and interactions between immune and tumour cells in microfluidic chips. In recent years there was the development of Organ on Chip technology (OOC) [1, 15, 31, 36, 55] and the related availability of data coming from laboratory experiments. Here, in the framework of OOC we are focused on Tumour-on-Chip (TOC) experiment [31, 97, 141], designed to assess the effects of therapeutic drugs on cancer cells and on the possible reactions of the immune system.
The aim of this thesis are manifold: the construction of a macroscopic chemotaxis model to reproduce the main features of the observed phenomena (cell migration driven by a chemical stimulus, short and long-range interactions between immune cells); the development of accurate and stable numerical approximation schemes in order to obtain a simulation algorithm, based on the mathematical model, able to reproduce the qualitative behaviour of cells observed in OOC laboratory experiments; the development of robust techniques for the estimation of model parameters in order to perform the validation of the mathematical model and the discovery of new and unknown features by predictions.
These can be predictions about the future behaviour in an existing experimental setting, and simulations with a new set of experimental settings such as different cell species, different chemicals etc.
When new results are obtained with such simulations and perhaps even new features discovered, one can conduct the experiment.
If the experimental observations and results coincide with the results of sim-

ulation, the new discoveries have been verified. Even when the results of the experiment does not coincide with the simulation, the data obtained from the experiment can be used for a training procedure against the simulating model to modify and further improve the mathematical model. This cycle is depicted in Figure 1.1.

Mathematical Model for Chemotaxis

In mathematics the migration of cells has been a widely studied field and described at different scales: microscopic by describing the movement of individual cells [54], but also in a macroscopic way that describes the cell density as a whole through partial differential equations. A variety of mathematical models for biological phenomena, including cell dynamics and cancer can be found in [5, 42, 47, 71, 99, 118].
The most popular mathematical model to describe chemotactic cell migration was proposed by Keller and Segel [83] in the 1970s. Their model is given by a coupled convection-diffusion equation system with reaction term for the cell density $u(\boldsymbol{x}, t)$ and the concentration of a chemical $\phi(\boldsymbol{x}, t)$ that plays the role of the chemoattractant

$$\begin{cases} \partial_t u &= \nabla \cdot (-D_1 \nabla \phi + D_2 \nabla u), \\ \partial_t \phi &= \nabla \cdot (D_\phi \nabla \phi) + u f(\phi) - \phi k_\phi. \end{cases} \tag{1.1}$$

The Keller-Segel model (1.1) describes the movement of cells u through a parabolic partial differential equation which consists of diffusion, according to Fick's Law, and chemotaxis, driven by the gradient of the chemical concentration field.
The chemoattractant ϕ in (1.1) is also described as a parabolic partial differential equation in which the dispersion of chemicals is also determined through the diffusive term. The reaction term consists of the degradation and production of chemoattractant by other cells or external sources.
However, the Keller-Segel model (1.1) has some disadvantages such as the infinite speed of propagation. Then, it does not seems to be sufficiently precise to describe the migration of cells for short time scales [43]. For this reason we also consider, at least in one dimension, a hyperbolic model that is obtained by using Cattaneo's Law:

$$\begin{cases} \partial_t u + \partial_x v &= g, \\ \partial_t v + \lambda_c^2 \partial_x u &= \frac{1}{\tau}(-v + f), \\ \partial_t \phi &= D_\phi \partial_{xx} \phi + a u - b \phi, \end{cases} \tag{1.2}$$

with cell density u, chemoattractant ϕ, flux v, chemotactic function $f := \widehat{f}u$ and source term $g := \widehat{g}u$ where $\widehat{f}, \widehat{g}$ can depend on u and ϕ.
Such model takes into account a certain delay $\tau > 0$, also called relaxation time, which describes the time needed for the flux to adapt to the gradient of chemoattractant [54].
This system was analytically studied on the whole interval and on bounded intervals in [60], and a numerical method, the Asymptotic High Order (AHO) scheme, was introduced in [109] for the approximation of the hyperbolic equations in (1.2). This new class of schemes increases the approximation order when the solution becomes close to a stationary solution, making it increasingly accurate for large times. In particular, these schemes are based on standard finite difference methods, modified by a suitable treatment of the source terms, and take into account the behaviour of the solutions near non-constant stationary states.

Mathematical Model of Organ on Chip(OOC)

Based on the parabolic Keller-Segel model and the hyperbolic Cattaneo model, and motivated by the laboratory experiment made on the microfluidic chip [31, 141], we introduce a mathematical model mimicking the interactions between two cell populations, namely immune and cancer cells, see also the paper [25], extracted from the work of this thesis and already published.
Although Keller-Segel models and Cattaneo models have been already studied for many decades, their application on Organ-on-Chips is a novelty.
Here in particular, we focus on Tumour-on-Chip (TOC) experiments, particularly designed for testing the effects of chemotherapy drugs on the immunocompetent behaviour.
In the two-dimensional chambers of the OOC we consider the following doubly-parabolic model

$$\begin{cases} \partial_t T &= D_T \Delta T - \lambda_T(\omega) T - k_T(t) T, \\ \partial_t M &= D_M \Delta M - \operatorname{div}(f(M, \phi)) - k_M(t) M, \\ \partial_t \phi &= D_\phi \Delta \phi + \alpha_\phi T - \beta_\phi \phi, \\ \partial_t \omega &= D_\omega \Delta \omega + \alpha_\omega M - \beta_\omega \omega, \end{cases} \tag{1.3}$$

with tumour cell density T, immune cell density M, chemoattractant concentration ϕ and cytokine ω, whereas in the microchannels connecting the two dimensional chambers, we consider two approaches: a one-dimensional version of the doubly-parabolic model used in the chambers

$$\begin{cases} \partial_t T_c &= D_T \partial_{xx} T_c - \lambda_{T_c}(\omega_c) T_c - k_{T_c}(t) T_c, \\ \partial_t M_c &= D_M \partial_{xx} M_c - \partial_x f_c - k_{M_c}(t) M_c, \\ \partial_t \phi_c &= D_\phi \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c, \\ \partial_t \omega_c &= D_\omega \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_\omega \omega_c \end{cases} \tag{1.4}$$

and a one-dimensional hyperbolic Cattaneo-like model:

$$\begin{cases} \partial_t T_c + \partial_x v_c^T = -\lambda_{T_c}(\omega_c) T_c - k_{T_c}(t) T_c, \\ \partial_t v_c^T + \frac{D_T}{\tau_T} \partial_x T_c = \frac{-v_c^T}{\tau_T}, \\ \partial_t \omega_c = D_\omega \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_c \omega_c, \\ \partial_t M_c + \partial_x v_c^M = -k_{M_c}(t) M_c, \\ \partial_t v_c^M + \frac{D_M}{\tau_M} \partial_x M_c = \left(f_c - v_c^M \right) \frac{1}{\tau_M}, \\ \partial_t \phi_c = D_\phi \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c, \end{cases} \tag{1.5}$$

where we indicate the one-dimensional quantities and functions with a c subscript.
The model parameters are the diffusion coefficients $D_T, D_M, D_\phi, D_\omega > 0$, growth and consumption rate of the chemicals $\alpha, \beta > 0$, the chemotactic term $f := \widehat{f} M$ that describes the directed movement of cells M in response to a chemical gradient, the tumour suppression function λ_T, representing the killing rate of tumour cells T caused by the cytokine ω, drug administration rate $k_M(t)$ and $k_T(t)$, and for the hyperbolic model the flux of immune and tumour cells v_M, v_T and relaxation times $\tau_T, \tau_M \in \mathbb{R}_{\geq 0}$.
Since in the laboratory experiments on the OOC [31, 141] no growth of immune and tumour cells outside the reservoir has been observed within the time range of the experiments, we did not include growth terms to the mathematical model (1.3)-(1.5) but it can be easily extended by adding an additional source term that describes the cell growth.

Transmission Condition at Interfaces

An important property of this multi-domain problem is the introduction of mass-preserving and positivity-preserving conditions involving the balancing of incoming and outgoing fluxes passing through interfaces between the 2D and 1D domains of the chip in absence of source terms.

This is realized with the permeability Kedem-Katchalsky (KK) conditions, describing the crossing of the flux through a node [82].
The approximation of double-parabolic chemotaxis models for the 1D-Keller-Segel model on networks was already considered in [19], however the transmission conditions were only between 1D-1D interfaces and on each domain the same parabolic model was considered.
Also transmission conditions between 1D-1D interfaces for the hyperbolic model (1.2) have been already considered in [26, 27]. On the contrary, transmission conditions between 2D-1D interfaces and with different types of equations have not been considered in the literature before. Then, this presents an original result of this thesis, published already in [25]. It is the first numerical work where this new technique of switching size of the domains and type of equations is introduced (parabolic-hyperbolic), in order to develop mass-preserving and positivity-preserving schemes, see [25]. The following equations are the KK-transmission conditions between 2D-parabolic-1D-parabolic interface

$$\begin{cases} D_u \partial_x u\left(L_x, y, t\right) - f^x\left(L_x, y, t\right) &= K\left(u_c\left(0,t\right) - u\left(L_x, y, t\right)\right), \\ D_{u_c} \partial_x u_c\left(0,t\right) - f_c\left(0,t\right) &= K\left[\left(b-a\right) u_c\left(0,t\right) - \displaystyle\int_a^b u\left(L_x, y, t\right) dy\right] \end{cases} \tag{1.6}$$

for $y \in [a,b]$ and the KK-transmission conditions between 2D-parabolic-1D-hyperbolic interface

$$\begin{cases} D_u \partial_x u\left(L_x, y, t\right) - f^x\left(L_x, y, t\right) &= K\left(u_c\left(0,t\right) - u\left(L_x, y, t\right)\right), \\ v_c\left(0,t\right) &= K\left[-\left(b-a\right) u_c\left(0,t\right) + \displaystyle\int_a^b u\left(L_x, y, t\right) dy\right] \end{cases} \tag{1.7}$$

for $y \in [a,b]$, with Kedem-Katchalsky constant $K \in \mathbb{R}_{>0}$, and interval $[a,b]$ indicating the entry region of the micro-channels.

Numerical Discretization

For the proper discretization of (1.3) for both two-dimensional chambers, we use a finite difference approximation, consisting in the Crank-Nicolson method with additional artificial viscosity term to take into account the otherwise very restrictive stability constants due to the dominant convection [134].
The same discretization method is used for the parabolic model (1.4) for the one-dimensional microchannels, whereas for the hyperbolic model (1.5), we apply the AHO scheme [109], whereas for the hyperbolic-parabolic model

(1.5) we apply second order AHO scheme, called in the following $(AHO)^2$, for the approximation of the hyperbolic equations in the system.
Special care has been taken for an appropriate discretization of the KK-transmission conditions at the interfaces in order to ensure mass and positivity preservation. This has been achieved by including a ghost cell value taken from the neighbouring domain and discretizing (1.6) and (1.7) properly to ensure the conservation of mass.
From a stability point of view, the stability conditions for the modified Crank-Nicolson method with $f := \widehat{f}u$ and $g := \widehat{g}u$ are

$$\textbf{modified Crank-Nicolson} \begin{cases} \textbf{for 1D} & \triangle t \leq \frac{\triangle x^2}{D+\triangle x \max\limits_i |\widehat{f}_i^n|} \\ \textbf{for 2D} & \triangle t \leq \frac{1}{\frac{D}{\triangle x^2}+\frac{D}{\triangle y^2}+\left(\frac{\max\limits_{i,j}|\widehat{f}_{i,j}^{x,n}|}{\triangle x}+\frac{\max\limits_{i,j}|\widehat{f}_{i,j}^{y,n}|}{\triangle y}\right)} \end{cases} \tag{1.8}$$

and for the $(AHO)^2$ scheme [109]

$$(AHO)^2 \begin{cases} \textbf{Stability} \begin{cases} \triangle x & \leq & 4\lambda_c\tau \\ \triangle t & \leq & \frac{4\triangle x\tau}{\triangle x+4\lambda_c\tau} \end{cases} \\ \textbf{Monotonicity} \left|\frac{\widehat{f}_c^n}{\lambda_c}\right| - \widehat{g}_c^n \leq 1 \end{cases} \tag{1.9}$$

with relaxation time $\tau > 0$ and $\lambda_c = \sqrt{\frac{D}{\tau}}$.
The introduction of the transmission conditions induces also stability conditions at the interfaces between the domains [25]. Although transmission conditions for 1D-parabolic-1D-parabolic and 1D-parabolic-1D-hyperbolic have been derived, no stability conditions have been characterized before. For the 2D-parabolic-1D-parabolic interface we have worked out the following stability conditions

$$\triangle t \leq \begin{cases} \frac{1}{D\left(\frac{1}{\triangle x^2}+\frac{1}{\triangle y^2}\right)+\frac{2|\widehat{f}_{N_x+1,j}^{x,n}|}{\triangle x}+\frac{|\widehat{f}_{N_x+1,j}^{y,n}|}{\triangle y}+\frac{K}{\triangle x}} & \text{for } \widehat{f}_{N_x+1,j}^{x,n} < 0. \\ \frac{1}{D\left(\frac{1}{\triangle x^2}+\frac{1}{\triangle y^2}\right)+\frac{|\widehat{f}_{N_x+1,j}^{y,n}|}{\triangle y}+\frac{K}{\triangle x}} & \text{else.} \\ \frac{\triangle x^2}{D+\triangle x\sigma K} & \text{for } \widehat{f}_0^n > 0 \\ \frac{\triangle x^2}{D+2\triangle x|\widehat{f}_0^n|+\triangle x\sigma K} & \text{else,} \end{cases} \tag{1.10}$$

and for the 2D-parabolic-1D-hyperbolic interface the stability is given by

$$\triangle t \leq \begin{cases} \dfrac{1}{D\left(\frac{1}{\triangle x^2}+\frac{1}{\triangle y^2}\right)+\frac{2|\widehat{f}^{x,n}_{N_x+1,j}|}{\triangle x}+\frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y}+\frac{K}{\triangle x}} & \text{for } \widehat{f}^{x,n}_{N_x+1,j} < 0. \\ \dfrac{1}{D\left(\frac{1}{\triangle x^2}+\frac{1}{\triangle y^2}\right)+\frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y}+\frac{K}{\triangle x}} & \text{else,} \\ \dfrac{1+\rho}{\frac{2\lambda_c}{\triangle x}(\rho-1)-\left(\frac{\triangle t}{2\tau}-\lambda_c\frac{\triangle t}{\triangle x}\right)(\rho-1)+K_k\frac{\sigma(1+\rho)}{\triangle x}+\frac{\widehat{f}^n_0}{2\lambda_c\tau}}. & \end{cases} \tag{1.11}$$

with $\rho := \frac{\lambda_c-\sigma K}{\lambda_c+\sigma K}$.

The stability constraints due to the transmission conditions only depend on the width σ of the channels and the constant $K \in \mathbb{R}_{>0}$ and has been shown to be not as restrictive as the general stability conditions of the schemes.
With the complete discretization of the OOC-model we are able to simulate the chemotactic movement and the interactions between the two different cell species (immune and cancer cells) living in the microfluidic chip environment. The model have been tested on several chemotactic functions, each describing different underlying mechanisms [49, 71, 140]:

- A proportional directional movement up a spatial gradient of chemoattractant ϕ.
- A receptor saturation, such that chemotaxis of cells is reduced in areas of high chemoattractant concentrations.
- Contact inhibition of locomotion.
- Interaction among cells to move up a spatial gradient of their own density.

The numerical results indicate that each model qualitatively reproduces the actual laboratory experiment well, with only slight differences in the dynamics on a smaller time scale.

Parameter Estimation of Convection-Diffusion Equation

The second part of this thesis is devoted to the recovery of the model parameters of the OOC model.
The parameter estimation problem is also known as inverse problem [10, 48], which in general is often very ill-posed and violates one of the three properties proposed by Hadamard [64]. Usually the parameter estimation

problem suffers from strong sensitivity to noise and errors and need to be re-formulated for numerical treatment [48].
Typically the parameter estimation problem is reformulated into a non-linear least square minimization problem [20]

$$\arg\min \parallel F(\Theta) - u_{data} \parallel_2^2, \tag{1.12}$$

where F is a non-linear operator that computes the solution of the model in dependence of the unknown model parameters Θ, and u_{data} as the observation data that are being used to fit the model [10].
The main challenges involved in the parameter estimation process are:

- Large computational costs due to the optimization methods, which involve a large number of unknown parameters.
- Presence of many local minima, making it difficult for local optimization method to find appropriate parameters.
- Sensitivity towards errors and noise.

A common numerical technique to overcome, or at least improve stability, is represented by Tikhonov-regularization techniques [48]

$$\arg\min \parallel F(\Theta) - u_{data} \parallel_2^2 + \lambda \parallel \Theta \parallel_2^2, \tag{1.13}$$

which involve an additional regularization term inducing smoothness and bounds to the parameter set Θ. The regularization parameter $\lambda \in \mathbb{R}_{\geq 0}$ is being determined by the L-Curve criteria [12, 68].
In our work we mostly make use of such a regularization for all the parameter estimations but we investigate also different norms for the regularization term, namely an L^1-norm $\parallel \cdot \parallel_1$ and a combination of L^1 and L^2 norm, also called elastic net [152]. The main differences of the L^1 and L^2-norm are:

- **Robustness**: L^2-norm increases the cost of outliers exponentially, whereas the L^1-norm only takes the absolute value, so it considers the unknown parameter values linearly and is more robust towards outliers.
- **Stability**: L^2-norm is more stable than L^1-norm in regards to the parameter estimation [85].
- **Computational difficulty**: L^1-norm makes the cost function non differentiable at certain points which requires special treatment [85].
- **Sparsity**: L^1-norm tends to set parameter values to zero.

We show numerically that the parameter estimation results in better parameter recovery with smaller errors for the L^2 norm but L^1 provides a reduction of parameters whenever no data are available to fit.
There have been many studies in the literature for the parameter estimation of convection-diffusion equations [8, 80, 91, 110, 130].
Most problems that are being considered however only deal with the recovery of the diffusion coefficient and not much work has been done for the recovery of the convection coefficients due to difficulty because of convection dominance. In this thesis we start from already existing parameter estimation methods designed to recover the diffusion coefficient of diffusion-convection equations and extend them to recover the convective parameters as well when the velocity field depends on time.

In [91] a multigrid approach is presented which makes use of the benefits that come from using a coarser grid resolution for the discretization of the convection-diffusion equation:

- Reduction of local minima.
- Reduced computational costs.

Taking advantage of this multigrid approach and inspired by [91], we derive a new multigrid parameter estimation method for the convection-diffusion equation where both diffusion coefficient D as a scalar and convection term $\boldsymbol{v}$ are estimated.

We then proceed to extend the parameter estimation methodology for the case of time-varying parameters and present two original approaches of this thesis how to realize it:

- Interpolation of in-between values between two time steps n and $n+1$.
- Temporal multigrid, which makes use of the coarser temporal grid to reduce the parameter number.

Benefits of this extension are the reduced number of parameters that needs to be estimated for time-varying functions and thus reduces the number of local minima [91].
Lastly we propose a post-smoothing step convection-diffusion problems, an original contribution to this work, where the convection term is not arbitrary but is proportional to the gradient of the chemoattractant ϕ which is defined as a parabolic partial differential equation.
We show that within the parameter estimation, we can apply a second estimate

$$\arg\min \parallel H(\Theta_v) - v \parallel_2^2 \tag{1.14}$$

with forward operator H which contains the condition under which the convection term is constrained and the convection field parameter estimation $\boldsymbol{v}$. This estimation is improved through the additional information provided by the forward operator H, which in the tested case is the parabolic partial differential equation of the chemoattractant.

Particle-Density Transformation

Subsequent to the mathematical model of the OOC (1.3), (1.4), (1.5) and a robust algorithm for the parameter estimation, we prepare the application of parameter estimation of the OOC on real data. In the laboratory experiment [31] the only data available at this point are the trajectories of individual cells, obtained through automated cell tracking software.
In order to use such microscopic data on our macroscopic model, we give a brief introduction to kernel density estimates [147], which allow a non-parametric transformation of the cell trajectories into density without a-priori knowledge about the underlying density function.
As an original contribution of the present thesis, a new parameter estimation methodology, the kernel density estimate parameter estimation methods (KDE-PE), is introduced that combine the kernel density estimate with our extended parameter estimation methods. Such estimation algorithm enables the calibration of macroscopic mathematical model directly with microscopic data.
In order to test the KDE-PE method on microscopic data, we make use of the acceptance-rejection method [73, 133] to produce microscopic data out of the macroscopic data and then retransform them back into density by using the kernel density estimation.

Parameter Estimation of OOC

After verifying that our KDE-PE does perform robustly under the kernel density estimate, we then transform the laboratory data, i.e the cell positions, into density fields and conduct the parameter estimation on a simplified model

$$\partial_t M \quad = \quad D_M \Delta M - \operatorname{div}(\boldsymbol{v} M) \tag{1.15}$$

with immune cell density M, diffusion coefficient $D_M > 0$ and velocity field function $\boldsymbol{v}$ which is defined as

$$\boldsymbol{v} := \widehat{f} = \chi \nabla \phi \tag{1.16}$$

and is connected to the chemoattractant equation

$$\partial_t \phi = D_\phi \Delta \phi + \alpha_\phi T - \beta_\phi \phi. \tag{1.17}$$

The simplification are based on the available informations about the laboratory experiment [31] and the data available for the parameter estimation procedure.
Conducting the parameter estimate on several regions in the chambers of the microfluidic chip, we are able to recover model parameters for the different models.
We then provide some insights about the chemoattractant field and attempt, based on assumptions made about the distribution, to reconstruct the chemoattractant distribution through a parameter estimation method and obtain conclusions since no real information are available about the chemoattractant in the experiment. In order to determine a model that describes the data and at the same time reduces the complexity of the model, i.e. balances between model complexity and good fit, we apply the Akaine information criterion (AIC) [2] and the Bayesian information criterion (BIC) [124] which are methods to rank and compare models with each other.

The total framework from the laboratory experiment on the OOC to the final result which is an appropriate mathematical model that replicates the laboratory experiment is shown in the flow chart in Figure 1.1.

Main Results and Original Contribution

The main results and original contribution of this thesis are:

- A novel macroscopic mathematical model for the Organ on Chip that is able to describe the different cell dynamics and interactions between them within the microfluidic chip under chemoattractants and cytokines.
- A numerical approximation scheme for the OOC-model with mass-preserving and positivity preserving transmission condition of multi-domain models between two-dimensional and one-dimensional domains with different types of partial differential equations (hyperbolic and parabolic).
- A robust parameter estimation method for general convection-diffusion equations for the recovery of diffusion coefficients and velocity fields, which can be dependent on space and on time.
- Extension of the parameter recovery for time-varying velocity fields in convection-diffusion equations (Temporal Multigrid approach).
- Development of the KDE-parameter estimation method (KDE-PE) for microscopic data.
- The recovery of model parameters involved in the OOC-model based on the laboratory experiment and the available data.

- The reconstruction of the chemoattractant distribution which dictates the dynamics of immune cells, but for which no data are available.

Plan of the Thesis

The thesis is structured into four parts.
In Part I, we introduce the Organ on Chip, a microfluidic chip lined with living human cells for drug development and disease modelling. We explain briefly the ability of microfluidic chips to host and combine a variety of different cells. We also present the laboratory experiments made on these microfluidic chips of the interaction between immune cells and cancer cells [15, 31, 55] which motivated this work and already resulted in a published paper [25]. We will then state the key results that have been obtained in [31, 36] which are essential in understanding the key mechanisms that are present in their experiments in order to reproduce them in silico through a mathematical representation, which is constructed in the following sections of Part I of the thesis.
The first section is devoted to the analytical mathematical modelling of the chemotactic movement and interactions between cells by presenting and deriving the classical macroscopic model, the Patlak-Keller-Segel model, a parabolic partial differential equation system but also consider a hyperbolic model such as the Cattaneo models, which are able to capture short time range properties better.
With these models as foundation we proceed to the following section. There we define the so called Kedem-Katchalsky transmission conditions which are mass and flux conserving interface boundary conditions. These are important properties to appropriately define chemotactic models on multiple domains of different spatial dimensions which has not been studies priorly and ensure that the total mass of cells is conserved.
The novel mathematical model of the OOC with the KK-transmission condition allows us to properly define the laboratory experiment conducted in [31] and to attempt the replication of the laboratory experiment in silico.
The background necessary for the numerical solution of the model is provided in the following sections. We present the finite difference schemes and we define important but known numerical properties such as stability, consistency and convergence and derive the proper discretization of mass-preserving boundary conditions which is an important property of our mathematical model.
Furthermore we also present the AHO schemes [6, 109], for the one-dimensional hyperbolic model defined in the microchannels.
The last section of Part I is devoted to the discretization of the mathematical model of the laboratory experiment. We conduct numerical experiments

with a set of model parameters which are appropriately chosen or taken experimentally from other works [104, 105]. We investigate the differences between a variety of different chemotactic models and present the numerical results qualitatively and quantitatively as an original contribution to this novel mathematical model for OOC.
We continue with Part II, which is devoted to the estimation of model parameters in chemotactic models.
The estimation of model parameters represents a typical inverse problem, here defined as a minimization problem in a given metric. For this reason we firstly present the numerical background about non-linear optimization methods, for which numerical analytical theory of non-linear and linear equation system is needed, and then we proceed with the formulation of the parameter estimation problem as an ill-posed inverse problem for which special care must be taken in form of proper regularization which is frequently used in practise in order to formulate a minimization problem that can be solved with satisfying results.
The penultimate section of Part II presents a multi-grid approach, inspired by the work of [91] to solve the minimization problem. First proposed by [91, 110] for multiphase porous media flow in order to estimate a space varying permeability coefficient, we modify their approach for chemotactic models that includes the velocity field parameters of the convection term and extend their work by several known regularization techniques with the aim of improving the results of parameter recovery for convection dominant models. Furthermore, additional post smoothing steps for an improved performance have been developed.

The newly improved algorithm for the recovery of model parameters is then tested on different examples and their results and quality compared and evaluated.
In last part of this thesis, Part III, we do the next step by introducing a calibration procedure for parameters of the mathematical model of the OOC against experimental data.
In the first section of this part we describe how the data have been obtained from the laboratory experiment, and in the following next section we focus on how such data, which are only available in microscopic form as cell positions, can be properly transformed with a kernel density estimate which is a non-parametric density estimation method frequently used in the field of statistics.
We present how the transformation from macroscopic data, in form of density, to microscopic data, in form of cell positions can be achieved through non-uniform random variate generation, and vice versa and investigate the errors involved in such transformation.

We conclude by extending our non-linear multigrid parameter estimation method for diffusion and convection parameters with the kernel density estimate, the so called KDE-PE method, a novelty in the context of parameter fitting of microscopic data for macroscopic models and evaluate the quality of parameter estimates on several test equations.
Lastly, as a first attempt in the field of parameter estimation for mathematical models for OOC, we use all the previously presented data preparation techniques and original numerical methods to apply the parameter estimation on the mathematical model of the laboratory experiment on the real data. This Chapter 8 combines all the previously discussed sections of this thesis from the mathematical modelling, and the numerical approximation of solutions, to the non-linear optimization, parameter estimation and data preparation. Several assumptions have been made in order to simplify the calculations as much as possible and increase robustness.
The results are visually shown in a particle representation that compares the real data with the artificially created particle based on the results of the parameter estimation.
These results are being used in an attempt to reconstruct a possible chemoattractant field, for which no data are available. The KDE-PE method is then applied to the mathematical model of the laboratory experiment for different chemotactic functions and the results compared through so called information criteria such as Akaike Information Criteria (AIC) [145] and the Bayesian Information Criteria (BIC) [124]. These are model selections criteria that measure the quality of each model relatively with each other by penalizing overfitting. Based on these criteria, a choice that balances the goodness of fit and simplicity of the model helps to determine which intrinsic cell movement mechanics are present in the laboratory experiment.
In Part IV we summarize the results obtained so far and discuss the challenges that has been appearing and we conclude this work with a prospective look for future research and possible improvements.
The results in this thesis are partially continued in one published paper [25], and two other papers in preparation.

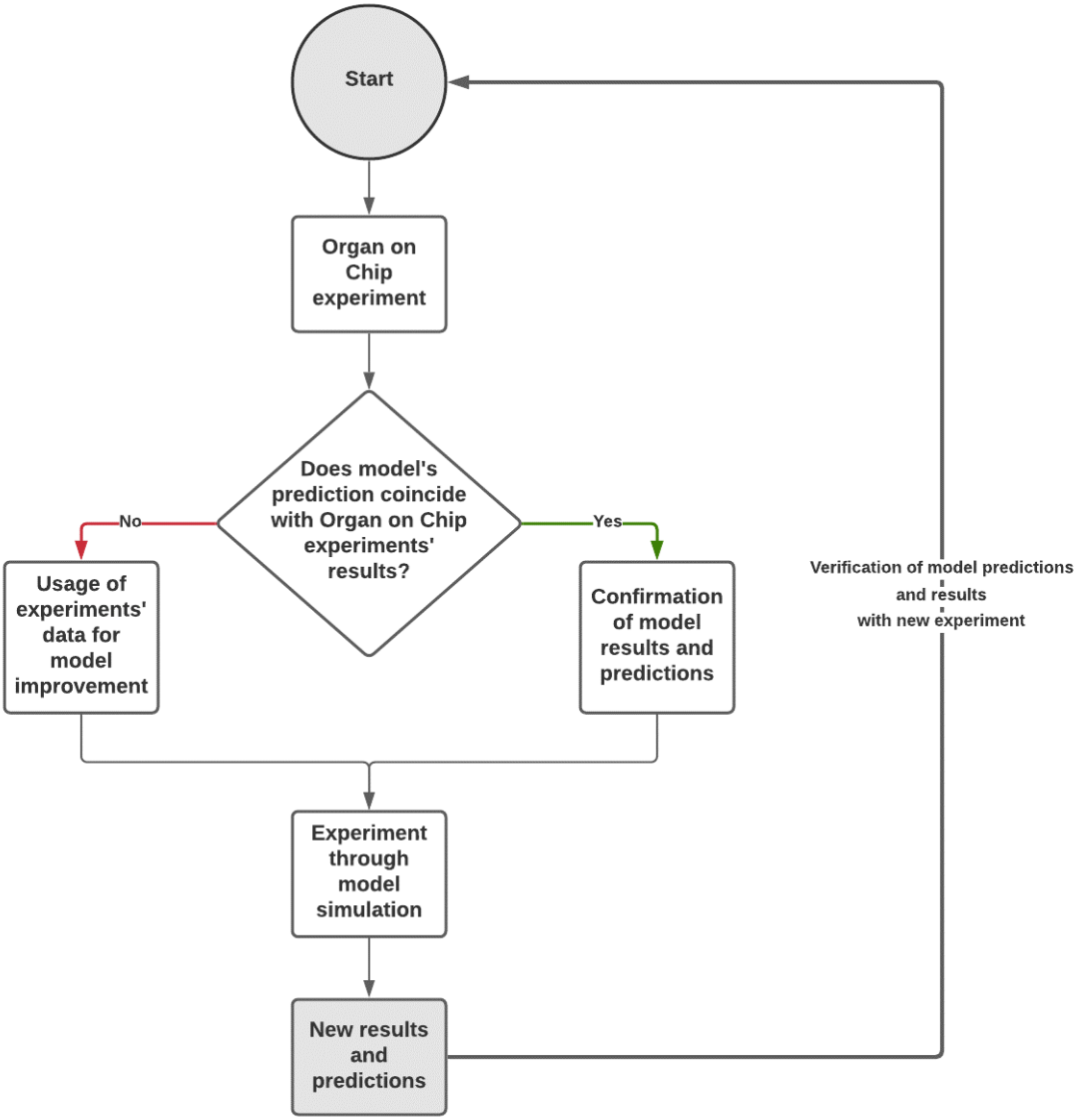


Figure 1.1: Representation of the mutual interdependency between mathematical modelling/simulation and in vitro experiment. Results from laboratory settings can be used to construct and improve mathematical models, which in turn can make predictions and discoveries that can be examined in new experiment setting proposed by the model, which creates new results to further improve the quality of the mathematical model.

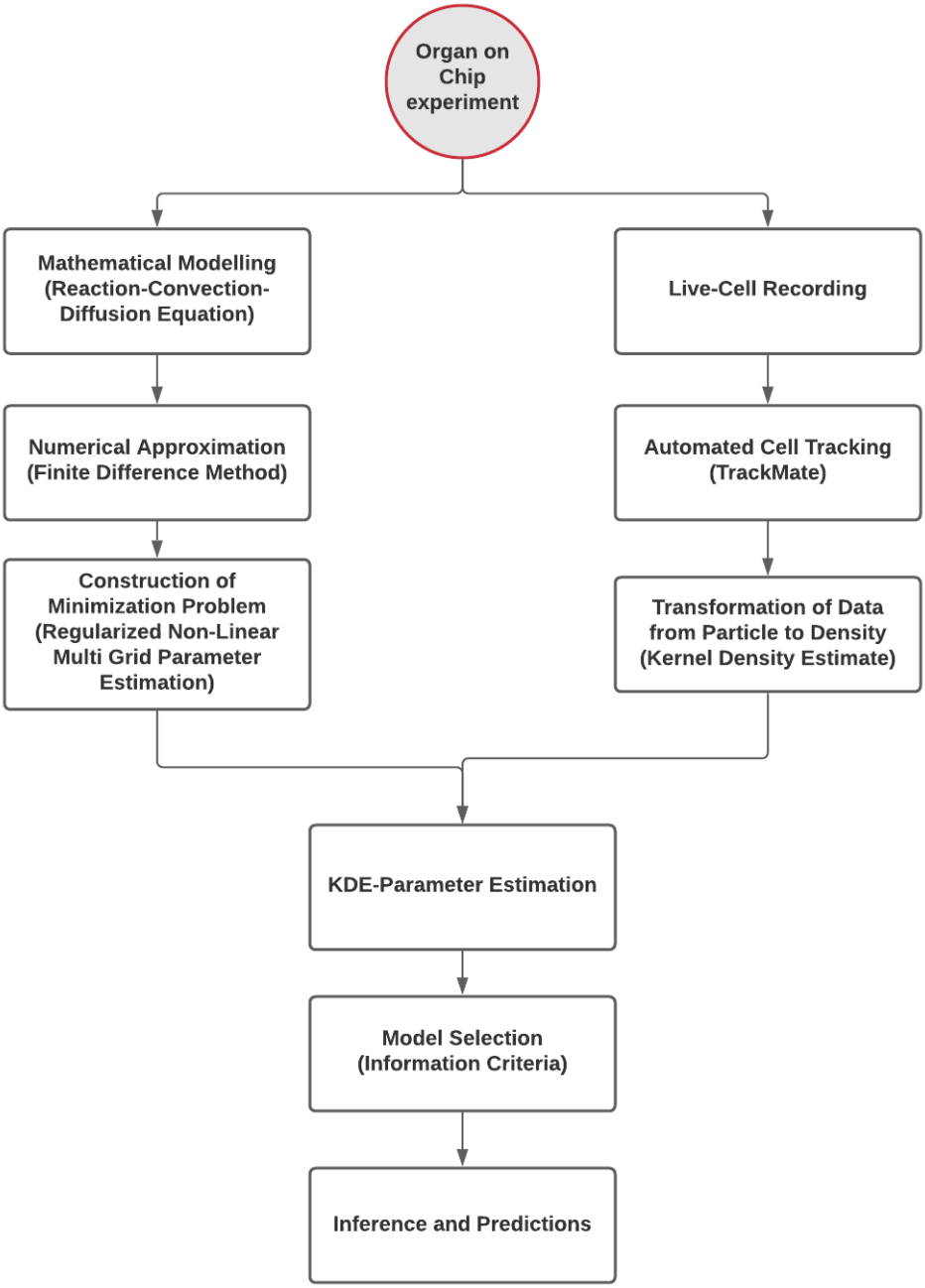


Figure 1.2: Outline of the thesis, which is based on the OOC experiment and divided into mathematical modelling and numerical analysis, and into obtaining data and data preparation, which combined can be used for an appropriate model parameter estimation and thus leads to a model that can be used for predictions and experiment suggestions.

Part I

A Multi-Domain Transmission Model for Organs-on-Chip

1.1 Organs-on-Chip (OOC)

Biological experiments can be roughly classified in three categories: In vivo, in vitro and in silico.
In vivo (Latin for "within the living") experiments, in which biological effects are studied on the whole living organism. Examples are drug testing on animals and the pathogenesis of diseases by comparing the effects of bacterial infection with the ones of purified bacterial toxic.
The other category are in vitro (Latin for "in glass") experiments. These experiments are conducted on test tubes, petri dishes, using only components of an organism. Advantages of such experiments over the in vivo experiments are the reduced complexity which makes it otherwise difficult to identify interactions between biological components. In vitro experiments are convenient for detailed studies which otherwise would remain obscure if not analysed isolated from the organism.
However it can be challenging extrapolating in vitro results to the in vivo because the overall effects of an experiment on a living subject can differ from the results in vitro due to additional interactions which were not present in the in vitro experiment.
Organ on Chips are recently developed microfluidic chips, engineered to mechanics and surrounding physiochemical environment of a living organism are recreated by essential increasing the complexity of in vitro systems. This allows microfluidic chips to recreate in vivo systems on a in vitro system.
The third category is in silico experiments, which are performed via computer simulations which are based on the modelling of biological processes and mechanisms.
In silico experiments and in vitro experiments go hand in hand such that data which are more easily obtained through in vitro experiments, can be used to build and further improve the in silico experiments. In silico experiments on the other hand can gain insights by computer-based simulations and model analysis, providing a possible key of finding new mechanisms and features which then can be tested in vitro for verification and for more data to further improve and modify the in silico experiments.

To identify interactions and reactions, i.e. the so-called "toxicity mechanisms", requires highly complex in vitro test systems. It is crucial that these test systems include defined metabolic performances of individual organs, as well as the potential involvement of the immune system. To capture the complex mechanism of action at the systemic level, taking into account organ interactions, animal models have been used to date [139].
It should be noted that due to evolutionary differences in the adaptation to different habitats and nutritional strategies, animal models show species-

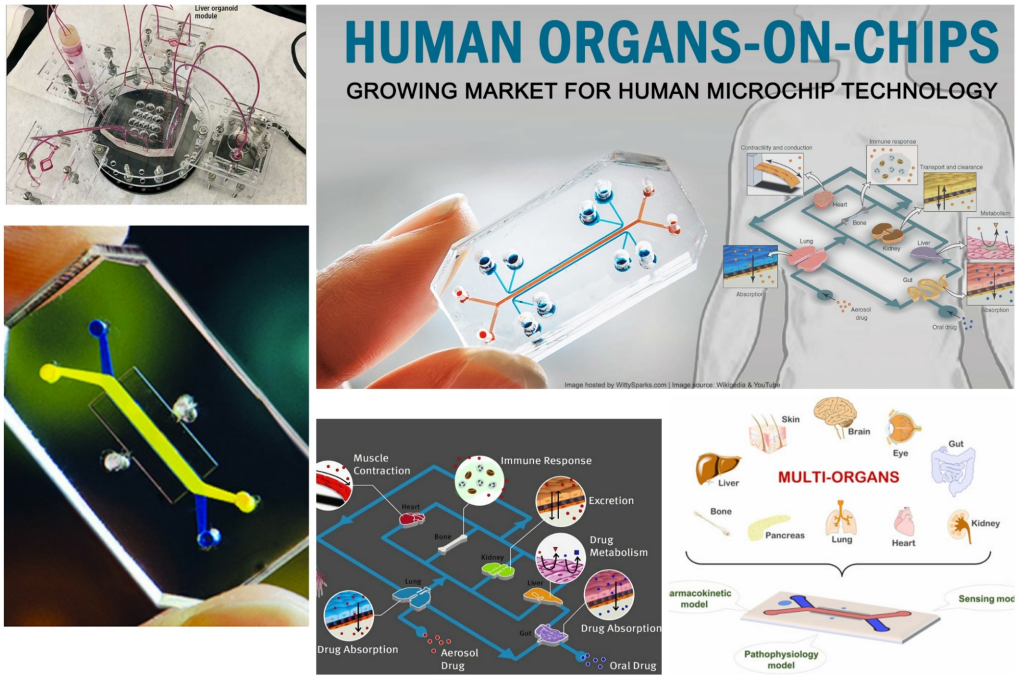


Figure 1.3: Organs-on-Chip

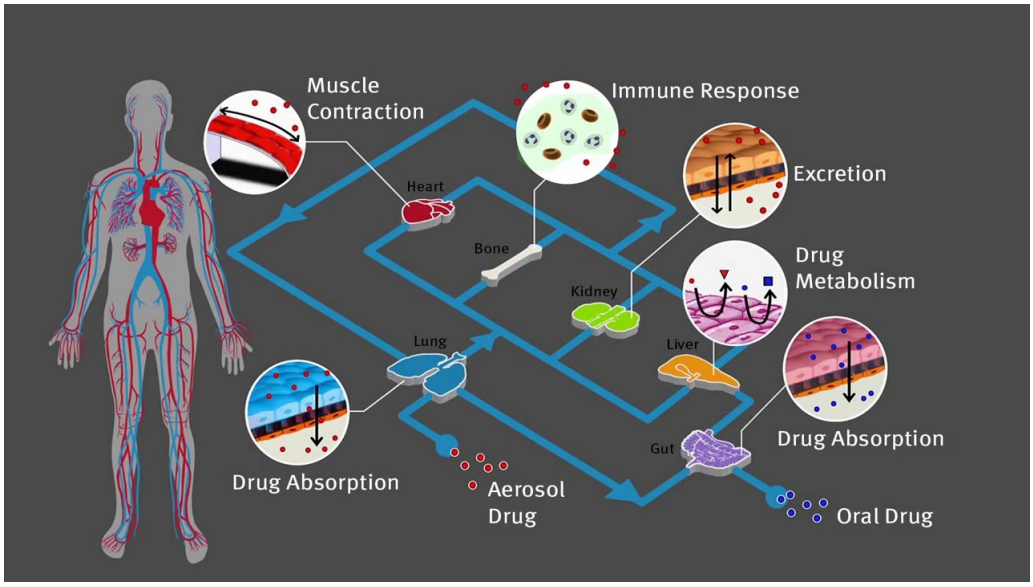


Figure 1.4: Human-On-Chip: A composition of several Organ-on-Chips to recapitulate the microarchitecture and functions of humans.

specific physiological differences compared to humans. Furthermore, the ethical aspect should not be disregarded. In order to be able to offer an ethically justifiable alternative to animal experiments, the advantages of organ-on-a-chip models present themselves here [103].
The aim of organ-on-a-chip models is to reproduce microphysiological conditions as lifelike as possible in vitro, thus avoiding the typical disadvantages of conventional cell cultures as far as possible. A major challenge in the cultivation of complex, three-dimensional organ models under physiologically relevant conditions is the maintenance of their function over longer periods of time. Microfluidics and the microfluidic perfusing biochip systems based on it form the basis for the long-term and standardised supply of such artificial tissues and therefore represent a significant further development of previous cell culture approaches. Potential applications of organs-on-a-chip and multi-organ systems include [94]:

- Drug development
- Toxicity screening
- Disease models
- personalised medicine and precision medicine (Person-on-Chip)

Organs-on-Chip models have a wide range of applications and offer great potential. To give a few examples of Organs-on-Chip systems [94]:

- Liver-on-Chip
- Heart-on-Chip
- Brain-on-Chip
- Lung-on-Chip
- Tumour-on-Chip

We will deal with the latter two in the following:

Lungs on chips

Up to this point, it is unfortunately not possible to mimic a complete lung yet. Nevertheless, Huh et al [11] have managed to reproduce the function of an alveolo-capillary membrane, the smallest functional unit in the lung. This organ is vital when trying out new drugs because it is the physical barrier between the body and the external world. To recreate this membrane, polydimethylsiloxane (PDMS) was used, which is coated with collagen for better

cell adhesion and the epithelial cells, which are in contact with the air, on one side and the endothelial cells, which are filled with a fluid made from nutrients instead of blood are in contact, on the other hand, separates.

Tumour on Chip (TOC)

In order to be able to study the survival and multiplication of malignant cells, the microenvironment in which cancer cells interact is to be simulated physically and chemically. The progress of research into replicating the tumour environment is much faster than with other chips: many articles already deal with this topic [90, 148]. Many tumours on chips were developed to test new treatments with different dosages [90]. Kim et al. [96] developed an automatic, programmable system to determine the optimal concentration. The efficiency of their system has been demonstrated on PC3 cells (prostate cancer cells) with a combination of doxorubicin and mitoxantrone.

1.1.1 Biological Framework

The in vitro experiment on a microfluidic chip, which inspired this work, is the investigation of interaction between cancer and immune cells in the context of immune competence and immunodeficiency to obtain insights of cancer-driven dynamics of immune cells.
We also refer to these papers [1, 15, 31, 141] for a more detailed explanation of the laboratory settings and focus more on the main ideas for the sake of focusing on the future mathematical modelling of the experiment.

OOC-Experiement: Crosstalk between cancer and immune cells

For the experiment in [1, 15, 31, 141] two types of spleen cells of mice are used. The Knock-Out type (KO-type) mice spleen cells which have a defect (a deficient for iterferon regulatory factor 8,IRF-8) that regulates the immunosurveillance of immune cells and B16 melanoma cells interactions.

Immunosurveillance are processes of the host immune system to limit or suppress tumour growth, which is compromised in the case of KO-types.
The other type of immune cells are from the spleen of wild type (WT), where the defect is not present.
In in vivo experiments it was observed that the KO-type immune cells are not highly responsive in the presence of B16 melanoma growth and also are not efficient in tumour suppression.

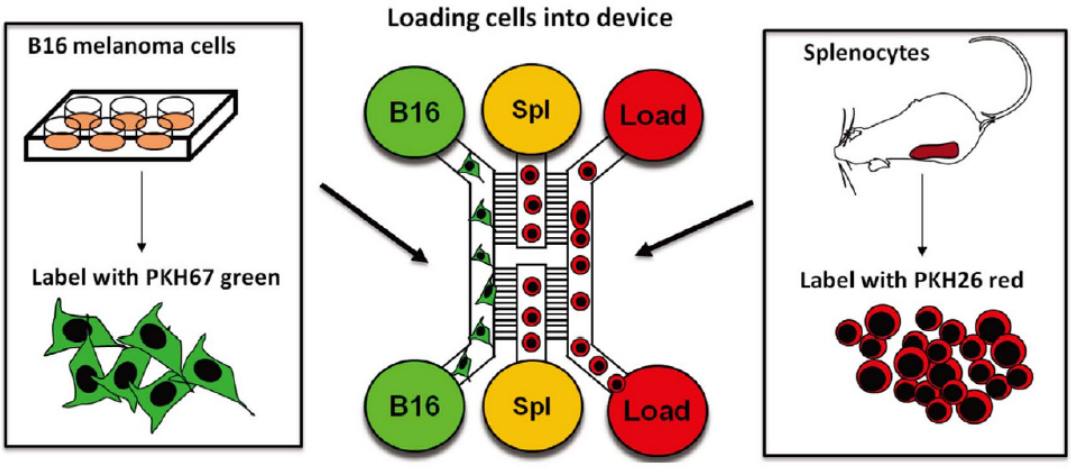


Figure 1.5: Illustration of OOC loaded with melanoma cells B16 and spleen cells [1].

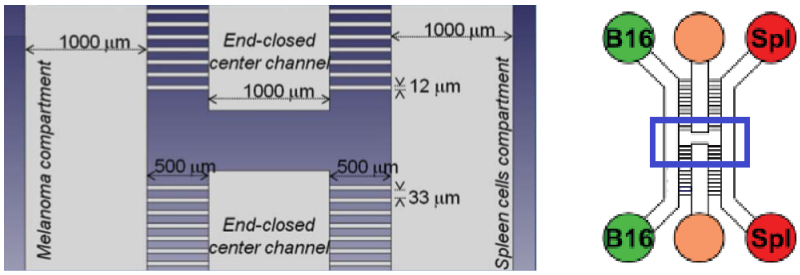


Figure 1.6: Left: Two-dimensional planimetry of the center area of the OOC (see figure on the right). The platform design features two center end-closed channels, adjacent to two cell culture compartments, connected via four sets of micron-size channels. Arrows and dotted lines represent the sizes of the indicated structures expressed in μm. The height of the microchannels is 10 μm. Right: Illustration of OOC. Blue region shows the two-dimensional planimetry [31].

For the in vitro experiment on the Organ on Chip, the cancer cells , B16-melanoma, and the two immune cell types, KO-type and WT-type were extracted from the spleen of mice and co-cultured.
Additionally further substances were extracted as well to create a more reliable in vivo environment on the chip.

The microfluidic chip that is being used in the experiments is fabricated in PDMS, a biocompatible silicone elastomer. The chip contains six reservoirs for cell loading and culture medium replacement and four compartments for cell cultures. Two reservoirs for both immune and tumour cells each will be loaded.
The left chamber that is connected to the reservoir of the tumour cells is connected via microchannels to the two center compartments, which are connected to the right chamber via microchannels as well.
In the experiment the spleen immune cells passively migrate into the two central compartments through the micro-channels and stop until sensing a chemo-attractant signal.
A more detailed schematic view can be seen in Figure 1.5. The time-lapse recordings were performed with a Juli Smart microscope [31] where a defined region (see Figure 1.8) can be recorded by 2 minutes per frame for a total of 48 hours of the experiment.
Additionally to the obtained video recording, an automated cell tracking program, TrackMate© [137], as been employed to analysis the trajectories of the cells.

We will briefly describe the experimental outcomes of the laboratory test made with WT- and KO-type immune cells.
In the experimental settings it was shown that WT-type cells have a much higher motility rate and migrated through their chamber through the microchannels into the right chamber where the tumour cells are present, whereas the KO-type cells showed a much lower motility rate and impaired motility towards the microchannels at which other end the tumour cells are located.
After a certain amount of time many WT-type immune cells can be found in the proximity of the tumour cells whereas this was not the case for the KO-type cells where the interaction between them and the tumour cells were only of short duration.
Additionally to this, it was shown that the tumour cells show slightly higher motility rates in presence of KO-type cells in contrast to slower in presence of WT-type cells.

These observations (here simplified, see [31] for further details) have been in coherence with in vivo results and confirms that the OOC laboratory experi-

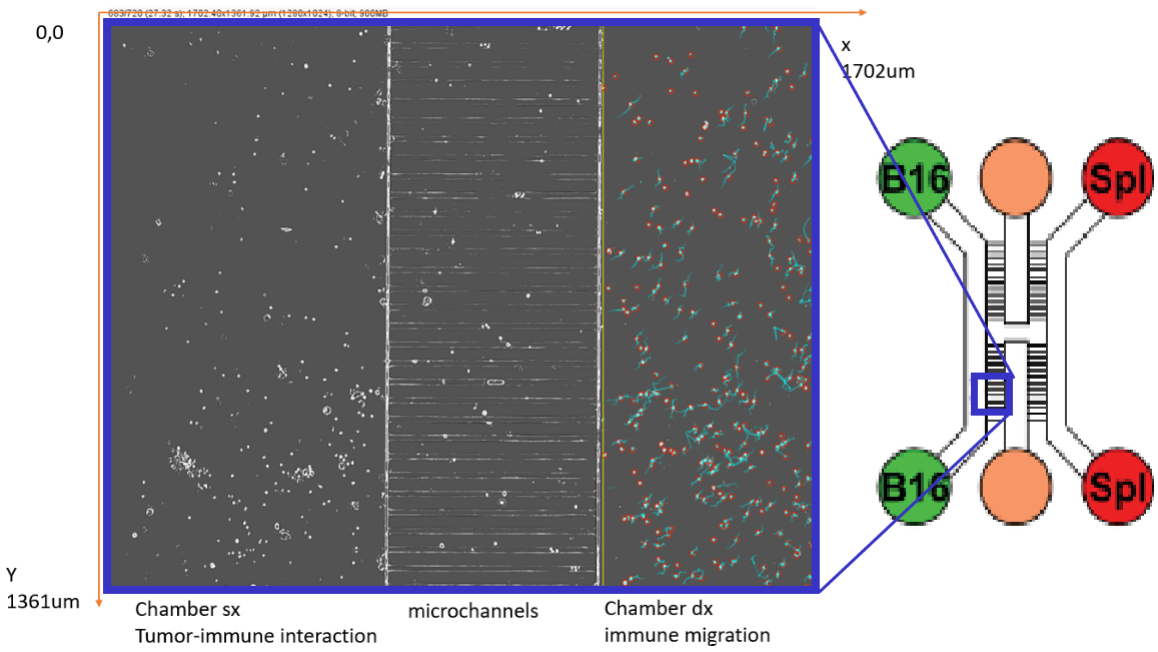


Figure 1.7: Defined region that has been able to be recorded through the microscope [141].

ment (in vitro) was able to reproduce key aspects that has been observed in real mice in vivo.
The experiments conducted and published in [31], with the video footage were used for a more data-driven analysis in [15] to investigate the trajectories of the individual cells based within stochastic process theory.
It was worked out that KO-type cells have shown to perform pure uncorrelated random walks whereas the WT-type cells performed drifted random walks, more coordinated towards the tumour cells.
In Figure 1.8 the trajectories indicating these results can be seen.
In later works [15, 141] it can be deduced that each tumour cell secreted a diffusing chemoattractant which generated an interaction range that can attract WT-type immune cells and can give an explanation to the drifted random walks of these types of cells and that these immune cells are capable of generating cytokines, a chemical that leads to an eradication of tumour cells.

The generalized results in this biological framework can be summarized in the following:

- Two cell cultures are present in each experiment: WT/KO-Immune cells and tumour cells.
- KO-type immune cells move in a uncorrelated random walk, indicating normal diffusive behaviour.

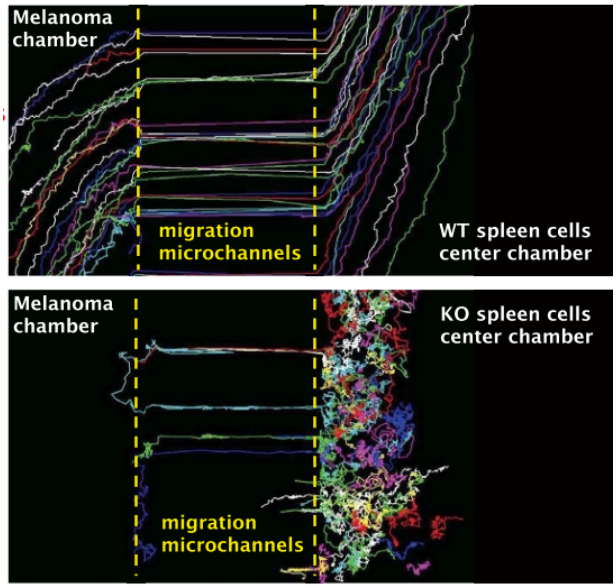


Figure 1.8: Exampled of immune cell trajectories performed by WT-cells and KO-type cells. KO-type cells show a uncorrelated random walk movement and WT-type cells show strongly directed diagonal movement through the right chamber into the left chamber through the micro-channels [1].

- WT-type immune cells migrate in a strongly directed movement towards turmour cells, following a chemoattractant gradient.
- Tumour cells are local sources of chemoattractant.
- WT-type immune cells generate a cytokine, expanding and eradicating tumour cells.
- Chemicals diffuse through the OOC.

Further inference based on the laboratory settings on the OOC are that both cell populations are being loaded in their reservoirs from which they migrate into their compartments, the immune cells on the right compartment and the tumour cells into the left. This means that at the beginning of the experiment the highest concentrations were located in the reservoirs, which in conclusion means, that the chemicals produced by both cells, have their highest concentration in the reservoirs as well, diffusing through the microfluidic chip.
Based on these observations and deductions, we derive a mathematical model to describe the laboratory experiments appropriately.
Finally it can be said that the in vitro experiment in [15, 31, 141] on the organ on chip have been in coherence with the actual in vivo experimental observations of the interactions between immune cells and cancer cells and shows

how the OOC is able to reproduce such results. The advantage of the experiment on the microfluidic chip is that the observation data obtained through it can be further analysed and gives further insight about the mechanics and effects within the interaction which then all can be used to derive an appropriate mathematical model that, in the best case, not only reproduce the in vitro experiment in silico but also makes correct predictions and new discoveries which then can be tried to verified in vitro.

1.1.2 Accession of Data with TrackMate©

Over the last decades, a lot of research has been made in the study of cell migration. Parallel to this also the development of imaging processing has been tremendously improved [137].
Many programs have been designed to study cell movements through live-cell imagining, i.e recordings, tracking them over time.
Many of those tracking tools must either be adapted for a individual live-cell imaging due to different laboratory settings, or entirely built from scratch.
This involves detection tools to extract cells from images, linking tools that required to track cells over the time of the recordings, visualization tools to overlay the raw data and tracking results and lastly analysis tolls that can evaluate the data.
These difficulties are the reason why it is still common to rely on manual tracking tools.
However there are many automated, semi-automated and manual single-cell tracking software available that are highly adjustable and user friendly for the end user thanks to an easy accessible graphical interface.
One of those software is TrackMate©, an open and extensible program for the single-cell tracking.
TrackMate© is a plugin within the Fiji ImageJ distribution, an imagine processing tool, that allows highly adjustable tracking of image recordings. The tracking procedure in TrackMate© consists of several steps that range from choosing the algorithm for the detection and tracking of single cells, to various tools to inspect intermediate results and adjust the settings for improved results.
Each cell can be tracked individually and be marked according to individual choices or to numerical values such as cell diameter, shape, track length, track displacement and so on.
Although automated, the manual inspection of the intermediate results allows for immediate changes in the tracking and the available visualization tools like the overlay of the trajectories of cells eases the readjusting.
An important asset is the configuration of the detection tools that are able to detect linking, gap-closing, cell splitting or merging events which can all be

set manually by the user.
The three classes of particle-linking algorithms in TrackMate© are derived from the Linear Assignment Problem (LAP) framework where the linking costs are calculated through the squared instance between cells, Kalman filter to tackle linear motions and nearest-neighbour search [137].

This software has already been used in protein motility studies, molecular motor tracking, sperm cell tracking, wound healing, cell movements on stiffness-patterned substrate and in zebra fish and many other studies.

Data from OOC experiment

The cell tracking of the laboratory experiments has been accomplished with TrackMate©. The live-cell recordings of the experiments on a section of the OOC (Figure 1.8) consist of 720 frames, each taken between a 2 minutes interval. The tracking software detects the cells in both chamber visible in the recording and tracks their movements through the 720 frames. Gap-closing events take into account the disappearance of cells in missing frames. In Figure 1.8 the cell tracking of the laboratory experiment can be seen step by step.
The data obtained through TrackMate© contain a list of labelled cells with their respective positions and velocities based on their tracked trajectories for each frame between 1 to 720.
This data can be easily imported into other software such as MATLAB© for further use such as in the parameter estimation process.
For our purpose only the detection and localization of cells per frame is being used in the parameter estimation of the OOC model.
The additional information of velocity and trajectory will not be used at this point but may be used in future works.

1.2 Parabolic and Hyperbolic Model for Chemotaxis

The migration of cells due to the influence of chemicals is known as chemotaxis. Many models in form of partial differential equations have been proposed to describe this evolution on a macroscopic level, considering each cell not individually, but as a density.
One of the most considered model is the Patlak-Keller-Segel system [83]. Originally it was meant to model the aggregation behaviour of Dictyostelium discodeum, a cellular slime mold, classified as eukaryotic organisms that can live as single cells independently but also are able to aggregate with each

other to form multicellular reproductive structures. Motivated by the identification of the chemical which acts as acrasin, they aimed to deduce the co-operative behaviour of the cellular slime mold from their individual properties.
The relevant quantities in their model are the density of the cellular slime mold $u(\boldsymbol{x}, t)$, the concentration of the chemical acrasin $\phi(\boldsymbol{x}, t)$, the concentration of the enzyme acrasinase $\eta(\boldsymbol{x}, t)$ that degrades acrasin and lastly a concentration of a complex $c(\boldsymbol{x}, t)$ that forms when acrasin and acrasinase react with each other.
Keller and Segel [83] proposed the following assumptions for their model:

- Acrasin is produced by the amoeba at the rate of $s(\phi)$ per amoeba.
- Acrasinase is produced by the amoeba at a rate of $g(\phi, \eta)$ per amoeba.
- Acrasin and acrinase react for form a complex which dissociates into the free enzyme acrasinase and a degraded product

$$\eta + \phi \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} c \xrightarrow{k_2} \eta + \text{ degraded product}, \tag{1.18}$$

  where k_1, k_{-1} and k_2 are the reaction rates.
- Acrasin, acrasinase and the complex diffuse according to Fick's law.
- The migration of amoebas is towards the direction of the increasing gradient of the chemical concentration ϕ and a random motion analogous to diffusion.

Based on the balance of mass, let V be an arbitrarily fixed domain with boundary ∂V, then

$$\frac{d}{dt}\int_V u dV = -\int_{\partial V} J^u \cdot \boldsymbol{n} dS + \int_V Q^u dV \tag{1.19}$$

with flux vector J^u, outer normal vector $\boldsymbol{n}$ and net mass of amoeba created/destroyed Q^u per unit time and unit volume.
Applying the divergence theorem on the surface integral and having a domain V which is not dependent on time (1.19)

$$\int_V \partial_t u + \nabla \cdot J^u - Q^u dV = 0. \tag{1.20}$$

Since this equation is true for all arbitarily domains V, we gain the differential equation

$$\partial_t u = -\nabla \cdot J^u + Q^u. \tag{1.21}$$

Equation (1.21) is true for all the four quantities u, ϕ, η and c and we only need to specify the flux J and Q for each of them.
By assumption, ϕ, η and c diffuse according to Fick's law [54]

$$J^{\phi} = -D_{\phi}\nabla\phi \qquad \text{and respectively for } \eta \text{ and } c \tag{1.22}$$

with constant diffusion coefficient $D_{\phi}, D_{\eta}, D_c \in \mathbb{R}$ respectively.
As for the amoeba u, in accordance to the assumption made by [83], we obtain the flux

$$J^{u} = -D_2\nabla u + D_1\nabla\phi, \tag{1.23}$$

where D_1 and D_2 can depend on ϕ and u.
In all flux terms J, the first term with positive constant be interpreted as a spreading out, whereas the second term $D_1\nabla\phi$, exclusively to the flux of amoeba, models the chemotaxis that describes the migration towards higher concentration of ϕ.
As for the functions Q, the total mass of amoebas is assumed constant, hence $Q^u = 0$ and for the other quantities based on the other assumptions made

$$\begin{aligned} Q^{\phi} &= us(\phi) - k_1\phi\eta + k_{-1}c, \\ Q^{\eta} &= ug(\phi,\eta) - k_1\phi\eta + (k_{-1}+k_2)c, \\ Q^{c} &= k_1\phi\eta - (k_{-1}+k_2)c. \end{aligned} \tag{1.24}$$

Thus, the complete model can be written as

$$\begin{cases} \partial_t u &= \nabla\cdot\left(-D_1\nabla\phi + D_2\nabla u\right), \\ \partial_t \phi &= \nabla\cdot\left(D_{\phi}\nabla\phi\right) + us(\phi) - k_1\phi\eta + k_{-1}c, \\ \partial_t \eta &= \nabla\cdot\left(D_{\eta}\nabla\eta\right) + ug(\phi,\eta) - k_1\phi\eta + (k_{-1}+k_2)c, \\ \partial_t c &= \nabla\cdot\left(D_c\nabla c\right) + k_1\phi\eta - (k_{-1}+k_2)c. \end{cases} \tag{1.25}$$

The model (1.25) can be further simplified and reduced to only two equations for u and ϕ by making further assumptions.
One is to assume that the complex c is in a chemical equilibrium, thus

$$Q^{\phi} = 0. \tag{1.26}$$

The second additional assumption made is that the total concentration of the enzyme η is constant η_0 and indicated by

$$\eta_0 = \eta + c. \tag{1.27}$$

Substituting these two assumptions into (1.25) reduces the model to

$$\begin{cases} \partial_t u &= \nabla\cdot\left(-D_1\nabla\phi + D_2\nabla u\right), \\ \partial_t \phi &= \nabla\cdot\left(D_{\phi}\nabla\phi\right) + us\left(\phi\right) - \phi k_{\phi} \end{cases} \tag{1.28}$$

with

$$\begin{aligned} k_\phi &= \eta_0 k_2 \tfrac{K}{1+K\phi}, \\ K &= \tfrac{k_1}{k_{-1}+k_2}. \end{aligned} \tag{1.29}$$

In other works found in the literature [47, 99, 105] similar models have been derived where the chemotactic function and reaction terms differ from the one used here.
The Keller-Segel model in its most general form [71] can be written as

$$\begin{cases} \partial_t u &= \nabla \cdot (D(u)\nabla u - A(u)B(\phi)C(\nabla\phi)) + F(u), \\ \partial_t \phi &= D_\phi \Delta\phi + uG(\phi) - b\phi, \end{cases} \tag{1.30}$$

but we will consider a "simplified" Keller-Segel model

$$\begin{cases} \partial_t u &= D_u \Delta u - \operatorname{div} f + g(x,y,t,u), \\ \partial_t \phi &= D_\phi \Delta\phi + \alpha u - \beta\phi \end{cases} \tag{1.31}$$

with chemotactic function $f = u\widehat{f}$ for some analytical results and for the construction of numerical schemes to solve (1.31) approximately.

Despite their ability to capture key phenomena, intuitive nature and relative tractability, the Patlak-Keller-Segel models are parabolic equations and as such not sufficiently precise to describe the migration of cells for short times due to the fast dissipation, since the diffusion implies an infinite speed of propagation of cells which is highly unrealistic and not physical.
For these reasons models based on hyperbolic equations have been considered since the propagation speed in this formulation is finite, and show that these models can be interpreted as a description of chemotaxis at a mesoscopic level [43, 45, 50, 54, 117].
A model for chemosensitive movement was introduced by Hillen and Dolak [45] and showed that their model was coherent with the observation made on Dictyostelium discoideum and E. coli.
In the following we will derive this model in a general way.

The Cattaneo model [34, 78] is based on a modification of Fourier law of heat conduction, which is the thermal equivalent of Fick's law to describe the propagation with finite speed.
Whereas in the Fick's law, the relation between cause (a non homogeneous density) and effect (a flux) is linear, the Cattaneo's law is more reasonable in the sense that there should be a time interval between cause and effect.
Let $\Theta(t,\boldsymbol{x})$ be the temperature of a homogeneous medium $\Omega \subset \mathbb{R}^n$ and $q(\boldsymbol{x},t)$ the heat flux, then Cattaneo's law is

$$q(\boldsymbol{x}, t+\tau) = -D\nabla\Theta(\boldsymbol{x},t) \tag{1.32}$$

with time delay $\tau > 0$ describing the time of the flux q needed to adapt to the negative gradient of the temperature.
Expanding the flux $q(\boldsymbol{x}, t+\tau)$ up to the first order in τ one has

$$\tau \partial_t q(\boldsymbol{x}, t) + q(\boldsymbol{x}, t) = -D\nabla\Theta(\boldsymbol{x}, t). \tag{1.33}$$

Combining Cattaneo's law with an energy conservation equation leads to the so called Cattaneo system.
If we generalize the context and consider $u(\boldsymbol{x}, t)$ as a cell density and $v(\boldsymbol{x}, t)$ to be the cell flux, then the Cattaneo model is written as

$$\begin{cases} \partial_t u + \nabla v &= 0, \\ \tau v_t + v &= -D\nabla u. \end{cases} \tag{1.34}$$

We can use this approach to derive the corresponding Cattaneo model for chemosensitive movement applied to the flux

$$\begin{aligned} w(\boldsymbol{x}, t+\tau) &= -D_u \nabla u(\boldsymbol{x}, t) + V(u, \phi)\nabla\phi(\boldsymbol{x}, t) \\ \Rightarrow \quad \tau\partial_t w + w &= -D_u \nabla u(\boldsymbol{x}, t) + V(u, \phi)\nabla\phi(\boldsymbol{x}, t) \end{aligned} \tag{1.35}$$

with cross-diffusion coefficient $V(u, \phi)$.
The Cattaneo model for chemosensitive movement can be written as

$$\begin{cases} \partial_t u + \nabla q &= 0, \\ \tau\partial_t q + q &= -D_u \nabla u + V(u, \phi)\nabla\phi. \end{cases} \tag{1.36}$$

For $\tau \to 0$ (1.36) reduces to parabolic Keller-Segel model.

For long time ranges the same behaviour as for the Keller-Segel model can be expected for the Cattaneo model. For shorter time ranges however, the Cattaneo model is expected to describe the dynamics better due to the finite characteristic speed. This was confirmed experimentally in [45] that both models fit the data using realistic parameter values as given in [52] but differences can be seen in time ranges up to about 40 seconds.

Of course there are a variety of other different ways to derive this class of models, see [54], and at this point we want to briefly mention another derivation of the model through the anisotropic random walk with reaction. Using this way we obtain the Greenberg-Alt model [58]

$$\begin{cases} \partial_t u^+ + \lambda\partial_x u^+ &= -\mu^+(\phi, \partial_x\phi)\, u^+ + \mu^-(\phi, \partial_x\phi)\, u^-, \\ \partial_t u^- - \lambda\partial_x u^- &= \mu^+(\phi, \partial_x\phi)\, u^+ - \mu^-(\phi, \partial_x\phi)\, u^-, \\ \partial_t\phi - D_\phi\partial_{xx}\phi &= \alpha(u^+ + u^-) - \beta\phi, \end{cases} \tag{1.37}$$

where $u^{\pm}$ denotes the density of the right and left moving part of the total density u, ϕ the chemoattractant and f a linear function.
The parameter $D_\phi > 0$ is the diffusion coefficient and $\lambda > 0$ the characteristic speed of propagation of $u^{\pm}$.
The terms $\mu^{\pm}$ are the turning rates, which control the transition event from u^+ to u^- and vice versa.
By defining the total flux $v = \lambda\left(u^+ - u^-\right)$ and total density $u = u^+ + u^-$, we recover the Cattaneo model for chemosensitive movement (1.36).
The formulation of (1.37) will later be useful in derivation of numerical schemes such as the asymptotic high order scheme [109] and for stability analysis using the monotonicity condition. For most of the following hyperbolic numerical approximation we will focus on models for chemosensitive movement, which is equivalent to (1.37), with

$$\begin{cases} \partial_t u + \partial_x v &= g(t,x), \\ \partial_t v + \lambda^2 \partial_x u &= \lambda\left(\mu^- - \mu^+\right) u - \left(\mu^+ + \mu^-\right) v, \\ \partial_t \phi &= D_\phi \partial_{xx}\phi + \alpha u - \beta\phi. \end{cases} \tag{1.38}$$

and source term function $g(t,x)$.

1.3 Boundary Conditions

An interesting property of both models, the parabolic Keller-Segel model (1.31) and hyperbolic Cattaneo model (1.38) when there is no source term $g = 0$, is the mass-preservation under homogeneous Neumann boundary conditions, also known as no-flux boundary conditions [101]. These results were obtained in [61] and several analytical results such as existence of global solution.
For the Keller-Segel model (1.31) defined on the two-dimensional domain Ω we have

$$\begin{aligned} \frac{d}{dt}\int_\Omega u\left(x,y,t\right) d\Omega &= \int_\Omega D_u \triangle u\left(x,y,t\right) - \operatorname{div} f\left(x,y,t\right) d\Omega \\ &= \oint_{\delta\Omega} \left(D_u \nabla u\left(x,y,t\right) - f\left(x,y,t\right)\right) \cdot \boldsymbol{n} dS \end{aligned} \tag{1.39}$$

$$\implies \frac{d}{dt}\int_\Omega u\left(x,y,t\right) d\Omega = 0, \qquad \text{for } \left(D_u \nabla u - f\left(x,y,t\right)\right) \cdot \boldsymbol{n}\big|_{\delta\Omega} = 0,$$

resulting in mass-preserving boundary conditions.
For the one-dimensional Cattaneo model (1.38) under the same assumptions

on the one-dimensional domain $\Omega := [0, L]$ we obtain

$$\begin{aligned} \frac{d}{dt}\int_\Omega u\,(x,t)\,dx &= \int_\Omega -\partial_x v(x,t)dx \\ \implies \frac{d}{dt}\int_\Omega u\,(x,y,t)\,d\Omega &= 0 \qquad \text{for } v(0,t) = v(L,t). \end{aligned} \tag{1.40}$$

1.4 Permeability Kedem-Katchalsky Interface Conditions

In view of the mathematical model we aim to derive, we have to deal with a multi-domain problem where we have a transmission between the domains Ω_l and Ω_r at an adjacent interface.
An important property for such transmission condition is the mass preservation.
The conservation of flux that has been described by Kedem and Katchalsky [82] and named permeability Kedem-Katchalsky conditions (KK-condition in short) describe such conservation of flux and have been already considered in [119] in the approximation of multi-domain linear partial differential equation problems with finite elements methods and also studied in [127] for reaction diffusion problems.
Transmission conditions between 1D-1D interfaces are also considered in [26, 27] for a network where on each arc the same parabolic model was defined on. The numerical treatment of interface conditions are explained in Chapter 3 in more detail.
However, transmission conditions at 2D-1D interfaces with different types of partial differential equations, namely parabolic and hyperbolic, have not been studied before and are an original contribution of the thesis which has been already published in [25].

1.4.1 Interface between 1D-2D models

In the following we consider the two dimensional domain $\Omega_l = [0, L_x] \times [0, L_y]$ and the one-dimensional domain $\Omega_c = [0, L]$. These two domains are connected at the interface $\delta\Omega_l^{interface} = \{L_x\} \times [a, b]$ and $\delta\Omega_c^{interface} = \{0\}$. Furthermore we assume no-flux boundary condition at the outer boundaries of $\delta\Omega_l \diagdown \delta\Omega_l^{interface}$ and $\delta\Omega_c \diagdown \delta\Omega_c^{interace} = \{L\}$. The schematization of the multi-domain is indicated in Figure 1.9.
The two-dimensional parabolic Keller-Segel model (1.31) without source term is defined on the domain Ω_l.

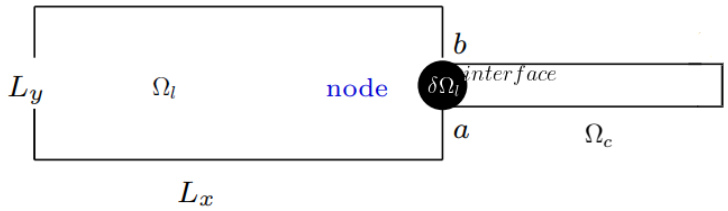


Figure 1.9: Schematization of the multi-domain with the two-dimensional domain $\Omega_l := [0, L_x] \times [0, L_y]$ connected through a one-dimensional channels $\Omega_c := [0, L]$ at the node.

Now let us consider first the one-dimensional parabolic Keller-Segel model (1.31) which is defined at the domain Ω_c, then if there is no flux at $\delta\Omega_c \setminus \delta\Omega_c^{interface}$ we have at the interface

$$\begin{aligned}
&\tfrac{d}{dt}\int_{\Omega_l} u_l(x,y,t)\,d\Omega + \frac{d}{dt}\int_{\Omega_c} u_c(x,t)\,dx &&= 0\\
\Leftrightarrow\quad &\int_{\Omega_l} D_{u_l}\triangle u_l(x,y,t) - \mathrm{div} f_l(x,y,t)\,d\Omega &&= -\int_0^L D_{u_c}\partial_{xx}u_c(x,t) - \partial_x f_c(x,t)\,d\Omega\\
\Leftrightarrow\quad &\int_{\delta\Omega_l} (D_{u_l}\nabla u_l(x,y,t) - f_l(x,y,t))\cdot \boldsymbol{n} d\Omega &&= -\left[D_{u_c}\partial_x u_c(x,t) - f_c(x,t)\right]|_0^L\\
\underbrace{\Rightarrow}_{\text{no-flux BC}}\quad &\int_a^b (D_{u_l}\partial_x u_l(L_x,y,t) - f_l^x(L_x,y,t))\,dy &&= D_{u_c}\partial_x u_c(0,t) - f_c(0,t),
\end{aligned} \tag{1.41}$$

which represents the continuity of flux at the 2D-1D interface.

Next we impose the Kedem-Katchalsky conditions [82] describing the conservation of the flux through a node which leads with (1.41) to the following KK-2D (parabolic)-1D (parabolic)-transmission condition

$$\begin{cases}
D_{u_l}\partial_x u_l(L_x,y,t) - f_l^x(L_x,y,t) &= K\left(u_c(0,t) - u_l(L_x,y,t)\right), \quad \text{for } y\in[a,b],\\
D_{u_c}\partial_x u_c(0,t) - f_c(0,t) &= K\left[(b-a)\,u_c(0,t) - \int_a^b u_l(L_x,y,t)\,dy\right].
\end{cases} \tag{1.42}$$

If we now define the one-dimensional hyperbolic Cattaneo-model (1.38) instead at domain Ω_c we can derive under the condition of mass preservation

$$\begin{aligned}
&\tfrac{d}{dt}\int_{\Omega_l} u_l(x,y,t)\,d\Omega + \frac{d}{dt}\int_{\Omega_c} u_c(x,t)\,dx &&= 0\\
\Longleftrightarrow\quad &\int_{\Omega_l} D_{u_l}\triangle u_l(x,y,t) - \mathrm{div} f_l(x,y,t)\,d\Omega &&= \int_0^L \partial_x v_c(x,t)\,dx\\
\underbrace{\Longrightarrow}_{\text{no-flux BC}}\quad &\oint_a^b (D_{u_l}\partial_x u_l(L_x,y,t) - f_l^x(L_x,y,t))\,dy &&= -v_c(0,t)
\end{aligned} \tag{1.43}$$

and obtain the following KK-2D (parabolic)-1D (hyperbolic) transmission conditions

$$\begin{cases} D_u \partial_x u_l\left(L_x, y, t\right) - f_l^x\left(L_x, y, t\right) &= K\left(u_c\left(0,t\right) - u_l\left(L_x, y, t\right)\right), \quad \text{for } y \in [a,b] \\ v_c\left(0,t\right) &= K\left[-\left(b-a\right) u_c\left(0,t\right) + \int_a^b u_l\left(L_x, y, t\right) dy\right]. \end{cases} \tag{1.44}$$

The numerical treatment of the KK-interface condition (1.42) and (1.44) are being worked out in Chapter 3 but can also be found in [25] which contains result of this work.

1.5 Mathematical Modelling of OOC Experiment

Having described the laboratory experiment made on the microfluidic chip, built the framework of the present biological mechanisms behind the laboratory experiment and derived two prototypical mathematical models, the parabolic Patlak-Keller-Segel model and hyperbolic Cattaneo model for chemotaxis, with which the evolution of the density of cells can be described, with a proper definition of mass-preserving boundary conditions and mass-preserving interface conditions, we will now continue to develop a mathematical model devoted to reproduce the chemotactic movement and interactions between the immune and cancer cells living in the microfluidic chip environment according to the laboratory experiment. This is the first attempt of deriving a mathematical model for OOC that has been made.
We begin with the proper definition of the domain by making several assumptions and inferences and proceed with the mathematical modelling in a similar way as for the Keller-Segel model.

The microfluidic chip, as shown in Figure 1.5 and formally described in a biological framework, is composed of six reservoirs for cell loading and culture medium replacement and four chambers for cell culture. However it is not necessary for the mathematical model to be defined over the whole chip for two reasons. Firstly, the cell cultures passively move into the two central chambers where they stop until sensing the chemoattractant from the treated tumour cells.
Thus in the areas other than the chambers, the cell dynamics are expected to be solely driven by diffusion.
Secondly, the cell cultures on the microfluidic chip have been observed and recorded for only a defined region (see Figure 1.8) and from the cell tracking only observational data from said region are available.
This leads to the justified assumption to only consider the defined region of

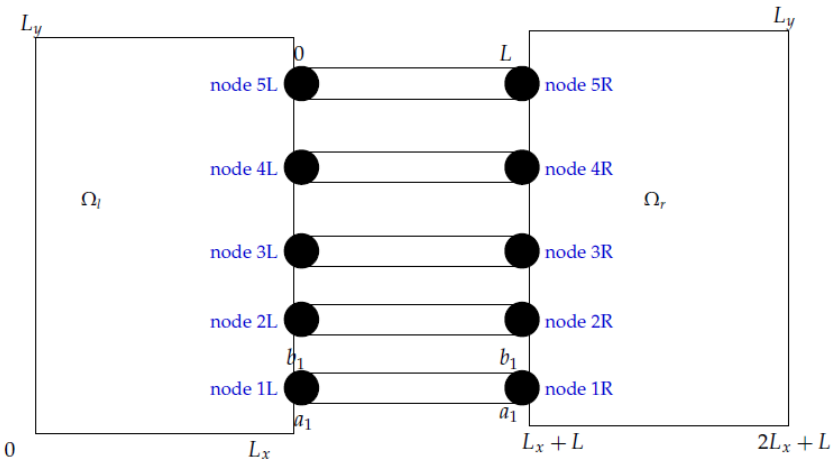


Figure 1.10: Simplified schematization of the chip geometry [25].

the video recording as the computational domain for the mathematical model. The two main culture chamber (a tumour and an immune cell compartment) are connected through narrow capillary migration micro-channels. The geometry of the micro-channels are: $10\mu m$ in height, $500\mu m$ in length and $12\mu m$ in width.
The geometry of the chambers are: $1mm$ in width and $100\mu m$ in height.
Because the video footage of the experiment is recorded at a fixed height, the third spatial dimension will be neglected in our model.
Furthermore, we assume that the micro-channels are considered as one-dimensional intervals with zero thickness. The reasoning behind this assumption is the fact that the size of the immune cells are about $8-10\mu m$ of diameter which is comparable to the width of the micro-channels with $12\mu m$, allowing us to assume a one-dimensional movement within the channels.
We summarize the assumptions for the computational domain as follows, which is also depicted in the simplified schematization of the chip geometry in Figure 1.10.

- Coherently with the video recording, we consider only the defined region (see Figure 1.8) as computational domain with a left chamber Ω_l and right chamber Ω_r connected through several microchannels Ω_{c_k}.
- The left and right chamber $\Omega_l := [0, L_x] \times [0, L_y]$ and $\Omega_r := [L_x + L, 2L_x + L] \times [0, L_y]$ are considered two-dimensional.
- The micro-channels $\Omega_{c_k} := [L_x, L_x + L]$ are considered one-dimensional.
- The interface between the two chambers and the micro-channels are located at $\delta\Omega_l^{interface} = \{L_x\} \times [a_k, b_k]$, $\delta\Omega_{c_k}^{interface} = \{L_x, L_x + L\}$ and

$\delta\Omega_r^{interface} = \{L_x + L\} \times [a_k, b_k]$ with $\sigma = b_k - a_k$ being the width of the micro-channels at the nodes.

Model

Coherently with the experiment settings, we identify the relevant quantities as:

- M is the density of the population of immune cells (macrophage).
- T is the density of the population of cancer cells.
- ϕ is the concentration of a chemical produced by the tumour cells T.
- ω is the concentration of a cytokine, produced by the immune cells M.

The assumptions made on the model are based on the inferences about the intrinsic biological mechanisms:

1. The chemical ϕ is produced by the tumour cells T at a constant rate $\alpha_\phi > 0$ per tumour cell T and decays at the constant rate $\beta_\phi > 0$.
2. The cytokine ω is produced by the immune cells M at a constant rate $\alpha_\omega > 0$ per immune cell M and decays at the constant rate $\beta_\omega > 0$.
3. T, ϕ and ω diffuse according to Fick's Law with constant diffusion coefficients $D_T, D_\phi, D_\omega > 0$.
4. The immune cells M diffuse according to Fick's Law with constant diffusion coefficient $D_M > 0$ and move in direction of a chemotactic function $f(M, \phi)$, which is specified later in Section 1.6.
5. The cytokine ω is responsible for killing of tumour cells T, with killing rate $\lambda_T(\omega)$ per tumour T.
6. Possible drug administration with an exponential decay rate in time that can kill immune cells M and tumour cells T at the decay rate $k_T(t)$ per tumour cell T (resp. $k_M(t)$ per immune cell M).

Taking assumptions 1-6 into our modelling, we can follow the same derivation path used for the Patlak-Keller-Segel model (1.31) to obtain the following

parabolic model in the two-dimensional chambers Ω_l and Ω_r

$$\begin{cases} \partial_t T &= D_T \Delta T - \lambda_T(\omega) T - k_T(t) T, \\ \partial_t M &= D_M \Delta M - \operatorname{div}(f(M,\phi)) - k_M(t) M, \\ \partial_t \phi &= D_\phi \Delta \phi + \alpha_\phi T - \beta_\phi \phi, \\ \partial_t \omega &= D_\omega \Delta \omega + \alpha_\omega M - \beta_\omega \omega. \end{cases} \tag{1.45}$$

The same parabolic Keller-Segel like model can be also used to describe the dynamical behaviour in the one-dimensional micro-channels as follows

$$\begin{cases} \partial_t T_c &= D_T \partial_{xx} T_c - \lambda_{T_c}(\omega_c) T_c - k_{T_c}(t) T_c, \\ \partial_t M_c &= D_M \partial_{xx} M_c - \partial_x f_c - k_{M_c}(t) M_c, \\ \partial_t \phi_c &= D_\phi \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c, \\ \partial_t \omega_c &= D_\omega \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_\omega \omega_c. \end{cases} \tag{1.46}$$

A different approach for the modelling of the one-dimensional channels is to consider a hyperbolic model, using a Cattaneo-like model as in (1.38) to obtain

$$\begin{cases} \partial_t T_c + \partial_x v_c^T = -\lambda_{T_c}(\omega_c) T_c - k_{T_c}(t) T_c, \\ \partial_t v_c^T + \frac{D_T}{\tau_T} \partial_x T_c = \frac{-v_c^T}{\tau_T}, \\ \partial_t \omega_c = D_{\omega_c} \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_c \omega_c, \\ \partial_t M_c + \partial_x v_c^M = -k_{M_c}(t) M_c, \\ \partial_t v_c^M + \frac{D_M}{\tau_M} \partial_x M_c = \frac{1}{\tau_M} \left(f_c - v_c^M \right), \\ \partial_t \phi_c = D_{\phi_c} \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c. \end{cases} \tag{1.47}$$

Such a hyperbolic model for the one-dimensional channels seems to be more realistic due to their finite propagation speed which is the dominant property at this scale.
The chemotactic function f that characterizes the intrinsic mechanics of the directional movement of immune cells M influenced by the spatial gradient of chemoattractant ϕ will be specified in Section 1.6.

The tumour suppression function $\lambda_T(\omega)$ that describes the action of the cytokine ω produced by immune cells M, which determines the death of tumour cells T, is defined as

$$\lambda_T(\omega) := \frac{k_{\omega_1}}{k_{\omega_2} + \omega} \tag{1.48}$$

with immune cells killing efficiency constant k_{ω_1} and dissociation constant k_{ω_2} as described in [105].
Although the tumour suppression is added into the numerical simulation in Chapter 3 in order to include this effect in the model qualitatively, in the laboratory experiments [1, 15, 31, 141] no information about the real killing rate in the microchip environment induced by the cytokine ω is available and no tumour cell deaths have been identified during the duration of the experiments.
The drug administration functions are defined as

$$\begin{aligned} k_T(t) : &= K_T e^{-\alpha_T t} \\ k_M(t) : &= K_M e^{-\alpha_M t} \end{aligned} \tag{1.49}$$

with the cell decay rate K_T, K_M and the drug decay rate α_T and α_M. We remark that the drug administration does not occur in the current laboratory experiments nor in the numerical simulations. However, for the sake of generality, they are included to address this phenomenon in future experiments with drug testing.
Growth terms for both immune cells M and tumour cells T are not included into our model, due to the fact that no growth has been registered during the laboratory experiment outside the reservoir [31, 141].
The model is completed with appropriate initial conditions for each quantity M,T,ϕ and ω and appropriate boundary conditions.
For the interface boundary conditions we apply (1.42) for the 2D-parabolic-1D-parabolic model whereas for the 2D-parabolic-1D-hyperbolic model (1.44). As for the outer boundaries, in the numerical simulations of Chapter 3 we applied no-flux boundary conditions in order to have mass conservation in absence of source terms. However, this is not realistic in the laboratory experiment since there is an inflow of cells from the outer boundaries.
This case will be only considered in Chapter 8 for the parameter estimation simulations. However the interface boundary conditions are applied in both cases.

1.6 Chemotactic Functions

We consider the following form of the Keller-Segel equations, which have been frequently used to describe chemotaxis models, for the cell density u

$$\partial_t u = D_u \Delta u - \operatorname{div}\left(\widehat{f} u\right). \tag{1.50}$$

Most work regarding this equation are focused on the case, in which the chemotactic term defined as $\widehat{f} = \chi \nabla \phi$ with chemotactic sensitivity function χ which can depend on both, cell density u and chemical concentration ϕ.
Chemotaxis describes the movement of cells up or down the concentration of the chemical ϕ, more precisely chemoattractant if cells migrate towards the chemical concentration and chemorepellent if otherwise.
The gradient of the chemical concentration ϕ can be the result from diffusion from an external source that releases the chemical, the depletion of chemical caused by cells themselves or a local source such as cells that produce the chemical themselves

$$\partial_t \phi = D_\phi \Delta \phi + \alpha_\phi u - \beta_\phi \phi, \tag{1.51}$$

which assumes cell-dependent chemical production with production rate $\alpha_\phi > 0$ and linear self-degradation with decay rate $\beta_\phi > 0$.
The classical Keller-Segel model is composed of the two equations (1.50) and (1.51). In [149] a theoretical mechanism in breast tumour is presented where the tumour cells release an attractant for macrophages towards which these immune cells migrate and release in turn a chemical that repels the tumour cells away from the primary tumour. Other possible influences of chemotactic movement of cells are density-dependent effects and contact inhibition of locomotion (CIL). The former effect occurs when cells are in a compact environment and restrict each others movements. The latter, as the name indicates, occurs when cells force directional changes when they collide with each other which has been investigated in [98].
In its most basic and most common form, the chemotactic sensitivity function is defined as $\chi = k$ for a constant $k \in \mathbb{R}$, which models only the migration towards/away from the gradient of chemical ϕ, but does not consider any other of the possible biological mechanism.
This basic model in combination with the Keller-Segel equations and the chemical equation (1.51), which makes the model a linear convection-diffusion equation system, has been studied in literature where many properties including global existing solutions and blow-up at finite time have been worked out [16].

In the following we will present various chemotactic terms $f = \widehat{f} u = u \chi \nabla \phi$ that describes the cell movements based on various biological hypotheses

such as self-generating gradients, repulsive and attractive interactions between cells and cell crowding effects.
The migration of cells is composed by random components, the diffusion, and a determined directional component, the chemotactic movement, describing the movement towards or away from the gradient of a chemical concentration ϕ.

Most chemotactic sensitivity functions χ can be categorized to the following classes:

- **Signal-dependent sensitivity**
- **Density-dependent sensitivity**

and we will investigate the following:

I **Basic model** (directional movement up a spatial gradient of chemoattractant)

$$\chi := k_1 \tag{1.52}$$

with cellular drift velocity $k_1 \in \mathbb{R}$.

II **Receptor saturation** [140]

$$\chi(\phi) := \frac{k_1}{(k_2 + \phi)^2} \tag{1.53}$$

with $k_2 \in \mathbb{R}_{>0}$ representing the receptor dissociation constant.

III **Overcrowding** [71], [98])

$$\chi(u, \phi) := \frac{k_1}{(k_2 + \phi)^2} \left(1 - \frac{u}{u_{max}}\right), \tag{1.54}$$

where cell's ability to move freely reduces at high cell densities as they approach the maximum cell density $u_{max} \in \mathbb{R}$. This is also known as contact inhibition of locomotion (CIL).

IV **Interaction** [49] (migration of cells in response to gradients of their own density and of chemoattractant)

$$\widehat{f} := \frac{k_1}{(k_2 + \phi)^2} \nabla\phi + \frac{\eta_1}{1 + \eta_2 u} \nabla u, \tag{1.55}$$

where $\eta_1 \in \mathbb{R}$ denotes the repulsion strength between cells and $\eta_2 \in \mathbb{R}_{>0}$ the strength of the interaction which is reduced at high cell densities.

V **Interaction** [49] (chemoattractant free)

$$\widehat{f} := \frac{\eta_1}{(\eta_2 + u)^\gamma} \nabla u. \tag{1.56}$$

Basic model

The basic model has been motivated by the cell movements as biased random walks of each cells. In the one-dimensional case, let h be the constant jump length of a cell to the left $x-h$ or right $x+h$. We also assume that cells do not interact with each other. Then the discrete evolution of the particle density $u(x,t)$ at position x can be defined as

$$\partial_t u(x,t) = p^+_{x-h} u(x-h,t) + p^-_{x+h} u(x+h,t) - \left(p^+_x + p^-_x\right) u(x,t) \tag{1.57}$$

with probability

$$p^\pm_x = \left(a + b\left(\phi\left(x \pm h, t\right) - \phi\left(x,t\right)\right)\right), \tag{1.58}$$

where $p^\pm_x$ is the probability of a cell at x to jump to $x \pm h$. With these probabilities we can model a variety of chemotaxis models.

We propose that the cell jump is biased according to a local gradient of the chemical ϕ.
Plugging the probabilities (1.58) into (1.57) and using a Taylor expansion we obtain

$$\partial_t u(x,t) = \lambda h^2 \partial_x \left(a u \partial_x u(x,t) - 2b u(x,t) \partial_x \phi(x,t)\right) + \mathcal{O}(h^4). \tag{1.59}$$

If we further assume the existence of the limits

$$\begin{aligned} \lim_{h\to 0, \lambda\to\infty} a\lambda h^2 &= D_u, \\ \lim_{h\to 0, \lambda\to\infty} 2b\lambda h^2 &= k \end{aligned} \tag{1.60}$$

with time scaling $\tau = \lambda t$, we re-obtain the partial differential equation (1.50)

$$\partial_t u(x,t) = \partial_x \left(D_u \partial_x u(x,t)\right) - \partial_x \left(k u \partial_x \phi\left(x,t\right)\right). \tag{1.61}$$

For the multidimensional case, the derivation is straightforward and follows the same steps such that

$$\partial_t u(\boldsymbol{x},t) = \nabla \cdot \left(D_u \nabla u(\boldsymbol{x},t) - k u(\boldsymbol{x},t) \nabla \phi\left(x,t\right)\right). \tag{1.62}$$

We remark that the diffusion D is not necessarily a constant but can also be a time and space dependant function. An example for such diffusion can be found in multiphase porous media flow models [91].

Signal-dependent sensitivity models

It has been shown in Tweedy [140] that the ability of cells to externally detect the chemical ϕ (i.e signal) through the binding of ϕ to receptors located at the cell's surfaces.
The reception of signals then leads to a movement response which we have expressed in the basic model before as

$$\chi = k \tag{1.63}$$

with rate of convection parameter $k \in \mathbb{R}$.
This model gets extended by including a signal-saturation effect.
At high concentrations of ϕ, the receptors might become fully occupied and no further detection of the chemical gradient $\nabla\phi$ is possible.
According to [116] let C be a single molecule of the chemical ϕ and R_{free} a free cell receptor and, resp. R_{full} an occupied cell receptor with $R_{total} = R_{full} + R_{free} = const$. The chemical reaction equation is then describes as

$$R_{free} + V \underset{r_{-1}}{\overset{r_1}{\leftrightharpoons}} R_{full} \tag{1.64}$$

with reaction rates $r_1, r_{-1} > 0$. The chemical reaction equation can then be solved which leads to the receptor-saturation model with

$$\chi(\phi) = \frac{k_1}{(k_2 + \phi)^2}, \tag{1.65}$$

where k_2 depends on both the reaction rates r_1, r_2 and the concentration of receptors.
The receptor-saturation model have found application in many chemotaxis models [52, 87, 126] and more detailed versions can be found in the literature that include other forms of signal-sensitivity dependencies [18, 129].

Density-dependent sensitivity model

The overcrowding model (1.54) is actually a combination of the receptor saturation model (1.53) with an additional factor contributed to a density-dependent effect, introduced in [71].
Assuming that cells are nonzero space-filling objects, they occupy the area they are positioned at, limiting the movement of other cells, we can use the random walk equation (1.57) and extend the jump probability $p_x^{\pm}$ (1.58) with the function $q(u)$ to

$$p_x^{\pm} = q(u(x \pm h))\,(a + b\,(\phi(x \pm h, t) - \phi(x, t)))\,. \tag{1.66}$$

Here the function $q\left(u\left(x \pm h\right)\right)$ describes the probability of finding space which depends on the density of cells in that position.
Using Taylor expansion as in the derivation of the basic models, leads to the following convection-diffusion equation for cell density u

$$\partial_t u = \nabla \cdot \left(D\left(q(u) - u\partial_u q(u)\right)\nabla u - kuq(u)\nabla\phi\right). \tag{1.67}$$

A prototypical function for q would be

$$q\left(u\right) = 1 - \frac{u}{u_{max}} \tag{1.68}$$

with u_{max} as the maximum cell density, which leads to the general overcrowding model (without receptor saturation)

$$\chi\left(u\right) = k\left(1 - \frac{u}{u_{max}}\right), \tag{1.69}$$

which can be easily extended to the overcrowding model (1.54) by replacing k with the receptor-saturation (1.65).

These models have been studied in [70] where global existence of solutions and other properties have been shown.

Most chemotaxis models in the literature assume the diffusion coefficient D as constant, but nonconstant diffusion coefficient can be found in the literature such as [72] for example where the effect of cell-cell adhesion is investigated which belong to the class of non-linear diffusion models.

The interaction model (1.55) as described in [49] is the extension of the receptor-saturation model (1.53) with an additional term that takes into account that the migration of cells might be affected by attraction or repulsion among themselves through other chemicals released by the cells

$$\frac{\eta_1}{1 + \eta_2 u}\nabla u \tag{1.70}$$

with parameter $\eta_1 \in \mathbb{R}$ and $\eta_2 > 0$.
This interaction model motivates the chemoattractant free interaction model (1.56), a modification of the proposed model in [49], where the cells directly migrate towards their own concentration gradient.
This model makes the chemical ϕ obsolete since it does not influence the migration of cells u anymore.

Any of the presented models can be extended and further modified to include more assumptions and possible mechanics of the chemotactic movement of

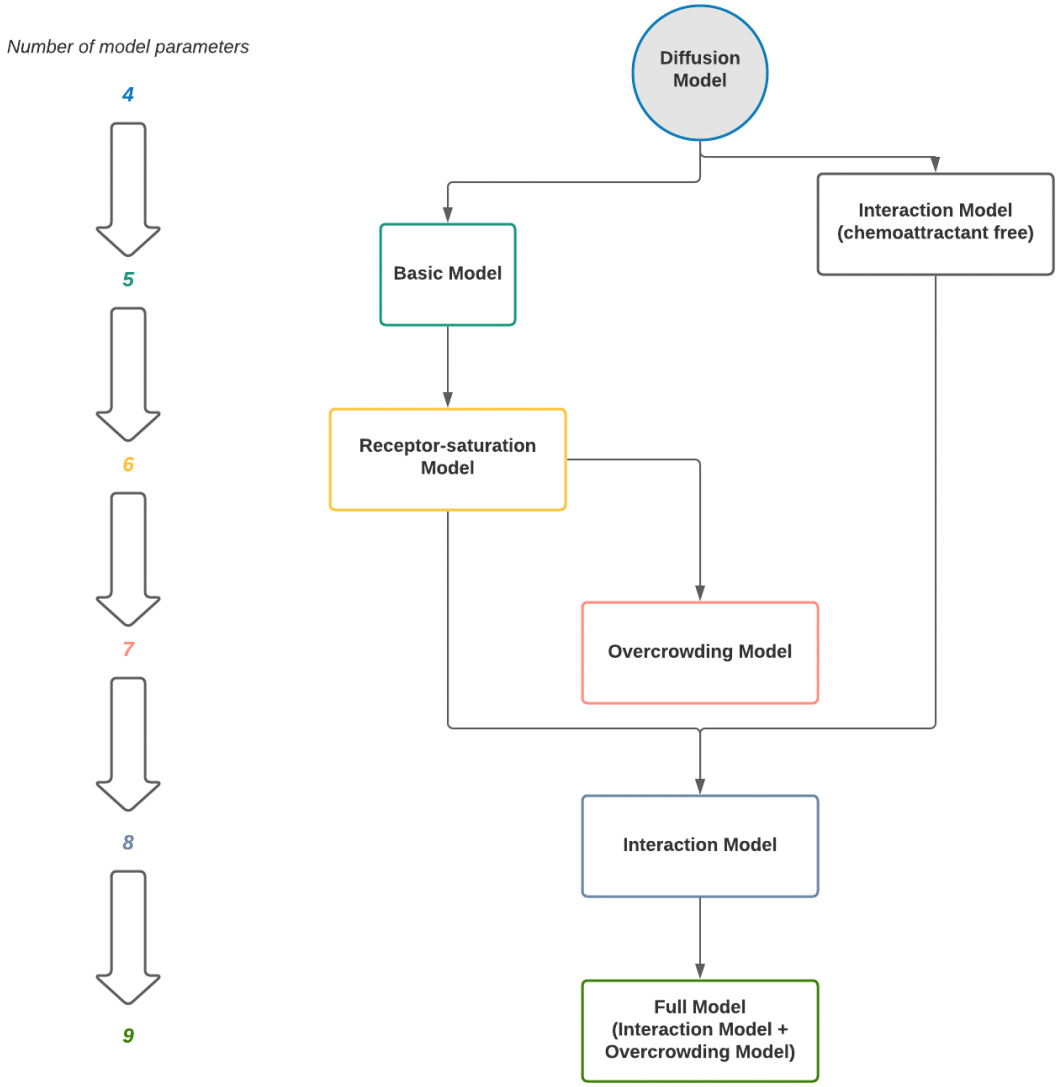


Figure 1.11: Flow chart of chemotaxis model (1.45) for different chemotactic terms with the number of model parameters for cell density u and chemoattractant concentration ϕ. Start of the model is the pure diffusion equation for both quantities without chemotactic term. Following the flow chart increases the complexity of the chemotactic term and hence increases the number of model parameters. A model with redundant terms (redundant model parameters) can lead to overfitting of the model and a too simplified model to an underfitting.

cells but we restrict ourselves with the presented ones.

With this we conclude the modelling part where we discussed the various models of partial differential equations and boundary and transmission conditions. Although local in time solutions and under certain conditions global solutions exist for these well-posed problems, an analytical solution can be very difficult to find [117]. For this reason we must find approximations to the solutions. The theory to accomplish this will be discussed in the next chapter.

Chapter 2

Numerical Backgrounds

In this chapter we want to provide numerical background about the numerical approximation of hyperbolic-parabolic partial differential equations such as the Keller-Segel model which we described in the previous chapter and is used later for the proper numerical approximation of the mathematical model of the Organ-on-Chip with transmission conditions.
We begin by introducing the concept of finite difference schemes and define important concepts such as consistency, stability, order and convergence and use these to present some finite difference schemes for prototypical equations.
The main references of this chapter are taken from [88, 102, 134].

2.1 Introduction to Finite Difference Schemes

We focus our studies on the following homogeneous partial differential equations of different types:

- **Advection equation** (hyperbolic)

$$\partial_t u + c\partial_x u = 0 \tag{2.1}$$

- **Diffusion equation** (parabolic)

$$\partial_t u - D\partial_{xx} u = 0 \tag{2.2}$$

- **Convection-Diffusion equation** (parabolic)

$$\partial_t u + c\partial_x u - D\partial_{xx} u = 0. \tag{2.3}$$

We start with a grid in the (t, x)-plane.
For the domain $\Omega \subset \mathbb{R}$ and for the time domain $0 \leq t \leq T$ we define equispaced points $x_i := i\triangle x$, $t_n := n\triangle t$ with $\triangle x, \triangle t > 0$ and $J_\Omega := \{i\}$ with

$i = 0, \ldots, N+1$ and $\mathcal{T} := \{n\}$ with $n = 0, \ldots, M+1$. For a continuous function $u(t_n, x_i)$ on the grid we define its discretization vector as $v = \{v_i^n\}$ with $u(t_n, x_i) \approx v_i^n$.

For later use we also define two of the most common used discrete norms which are the l^∞-norm

$$\| v^n \|_{\triangle x,\infty} = \max_{i \in J_\Omega} \{|v_i^n|\} \tag{2.4}$$

and the l^2-norm

$$\| v^n \|_{\triangle x,2} = \left(\triangle x \sum_{i \in J_\Omega} |v_i^n|^2 \right)^{\frac{1}{2}}. \tag{2.5}$$

The essential idea behind finite difference schemes is to approximate the derivatives of a partial differential equation by finite differences such like

- **Forward difference**

$$\delta_1 u(t_n, x_i) = \frac{u_{i+1}^n - u_i^n}{\triangle x} + \mathcal{O}(\triangle x) \tag{2.6}$$

- **Backward difference**

$$\delta_{-1} u(t_n, x_i) = \frac{u_i^n - u_{i-1}^n}{\triangle x} + \mathcal{O}(\triangle x) \tag{2.7}$$

- **Central difference**

$$\delta_0 u(t_n, x_i) = \frac{u_{i+1}^n - u_{i-1}^n}{2\triangle x} + \mathcal{O}\left(\triangle x^2\right) \tag{2.8}$$

for first derivatives, and higher order differences such as

- **Central difference**

$$\delta_0^2 u(t_n, x_i) = \frac{u_{i+1} - 2u_i + u_{i-1}}{\triangle x^2} + \mathcal{O}\left(\triangle x^2\right), \tag{2.9}$$

which all can be derived by means of Taylor expansion but also in many other ways as well (see [53]).
Before we proceed we consider now a more general linear partial differential equation of the form

$$\begin{cases} P(\partial_t, \partial_x) u(t,x) &= f(t,x), & \text{for } x \in \mathbb{R},\ t \leq 0 \\ u(0,x) &= u_0(x), & \text{for } x \in \mathbb{R} \end{cases} \tag{2.10}$$

with P being a linear operator and first order in the derivative with respect to time t. The previously introduced partial differential equations (2.1), (2.2)

and (2.3) are such examples.
To create a finite difference scheme for any of those equations, we can replace the derivatives with one of the introduced finite differences. These schemes can be generalized by the algebraic system

$$P_{\triangle t,\triangle x} v_i^n = 0, \tag{2.11}$$

where $P_{\triangle t,\triangle x}$ is the difference operator.
A general form of a one-step finite difference method can be written as

$$L(S_+, S_-)v^{n+1} = Q(S_+, S_-)v^n \tag{2.12}$$

with polynomial L and Q in S_+, S_- with $S_\pm v_i = v_{i\pm1}$.
This can be also written in

$$\sum_{i\in I_\Omega} d_i v_i^{n+1} = \sum_{i\in I_\Omega} c_i v_i^n, \tag{2.13}$$

with coefficents $c_i, d_i \in \mathbb{R}$.
Examples of such finite difference schemes for the advection equation (2.1) are

- **Upwind** for $c > 0$

$$v_i^{n+1} = v_i^n - c\frac{\triangle t}{\triangle x}\left(v_i^n - v_{i-1}^n\right) \tag{2.14}$$

- **Upwind** for $c < 0$

$$v_i^{n+1} = v_i^n - c\frac{\triangle t}{\triangle x}\left(v_{i+1}^n - v_i^n\right) \tag{2.15}$$

- **FTCS** (forward in time, central in space)

$$v_i^{n+1} = v_i^n - c\frac{\triangle t}{2\triangle x}\left(v_{i+1}^n - v_{i-1}^n\right) \tag{2.16}$$

- **Lax-Friedrich**

$$v_i^{n+1} = \frac{v_{i+1}^n + v_{i-1}^n}{2} - c\frac{\triangle t}{2\triangle x}\left(v_{i+1}^n - v_{i-1}^n\right) \tag{2.17}$$

- **BTCS** (backward in time, central in space)

$$v_i^{n+1} = v_i^n - c\frac{\triangle t}{2\triangle x}\left(v_{i+1}^{n+1} - v_{i-1}^{n+1}\right) \tag{2.18}$$

- **Leapfrog** (multistep method)

$$v_i^{n+1} = v_i^{n-1} - c\frac{\triangle t}{\triangle x}\left(v_{i+1}^n - v_{i-1}^n\right). \tag{2.19}$$

These finite difference schemes (2.14) to (2.16), except for the last two, can be summarized to the class of **explicit three point schemes** of the form

$$v_i^{n+1} = \alpha v_{i+1}^n + \beta v_i^n + \gamma v_{i-1}^n \tag{2.20}$$

with $\alpha, \beta, \gamma \in \mathbb{R}$, whereas the *BCTS* scheme (2.18) is an implicit scheme which involves solving a linear equation system for each time iteration $n+1$. If the partial differential equation has non-linear terms, then solving a non-linear equation would be required.
In Section 4 we will explain in more detail how to solve them.
The leapfrog method is a so called multistep method which, besides t_n, also requires previous time iterations t_{n-1} in order to calculate the $n+1$ iteration for t_{n+1}. Multistep methods involve an initialization phase as well, because in order to calculate the first iteration for t_2, not only the initial data for t_0 are needed but also an additional at t_1. Often the initialization is done by a one step scheme with the same order as the multistep scheme to obtain values for t_1. In this work we restrict ourselves with one step methods only and refer to [102, 134] for more information.

For the parabolic diffusion equation (2.2) we have the following standard finite difference schemes

- **FTCS** (Diffusion)

$$v_i^{n+1} = v_i^n + D\frac{\triangle t}{\triangle x^2}\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) \tag{2.21}$$

- **BTCS** (Diffusion)

$$v_i^{n+1} = v_i^n + D\frac{\triangle t}{\triangle x^2}\left(v_{i+1}^{n+1} - 2v_i^{n+1} + v_{i-1}^{n+1}\right) \tag{2.22}$$

- **Crank-Nicolson scheme**

$$v_i^{n+1} = v_i^n + D\frac{\triangle t}{2\triangle x^2}\left(\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) + \left(v_{i+1}^{n+1} - 2v_i^{n+1} + v_{i-1}^{n+1}\right)\right), \tag{2.23}$$

which can also be generalized to the θ**-scheme**

$$v_i^{n+1} = v_i^n + D\frac{\triangle t}{\triangle x^2}\left[(1-\theta)\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) + \theta\left(v_{i+1}^{n+1} - 2v_i^{n+1} + v_{i-1}^{n+1}\right)\right) \tag{2.24}$$

for $\theta \in [0,1]$.

2.2 Consistency and Convergence

In order to establish the connection between the partial differential equation (2.10) and the finite difference scheme (2.11) we can use the following definition.

Definition 2.2.1 (**Convergence**). *A one-step finite difference scheme (2.11) approximating a partial differential equation (2.10) is convergent if for any solution to the partial differential equation, $u(t,x)$, with appropriate initial and boundary value, and solutions to the finite difference scheme, v_i^n, such that v_i^0 converges to $u_0(x)$ as $i\triangle x$ converges to x, then v_i^n converges to $u(t,x)$ as $(n\triangle t, i\triangle x)$ converges to (t,x) as $\triangle t, \triangle x \longrightarrow 0$.*

A scheme that fulfils definition 2.2.1 is called a convergent scheme and such schemes are being used in numerical computations.
An equivalent definition of convergence is that we call a scheme convergent in the discrete norm $\|\cdot\|_{\triangle x,\cdot}$ if

$$\| v^n - u^n \|_{\cdot,\triangle x} \to 0, \quad \text{for } \triangle t \to 0 \text{ and } n\triangle t \to t \in (0,T), \tag{2.25}$$

for all initial value $u(0,x) = u^0$ such that the corresponding partial differential equation is well-posed.
Another important concept of finite difference schemes is consistency.

Definition 2.2.2 (**Consistency**). *Given a partial differential equation, $Pu = f$ and a finite difference scheme $P_{\triangle t,\triangle x} v = f$, we say that the finite difference scheme is consistent with the partial differential equation if for any smooth function $\phi\,(t,x)$*

$$P\phi - P_{\triangle t,\triangle x}\phi \to 0 \text{ as } \triangle t, \triangle x \to 0 \tag{2.26}$$

the convergences being pointwise at each point (t,x).

We can apply this definition in the proof of the following proposition.

Proposition 2.2.3. *Let $\frac{\triangle t}{\triangle x}$ be fixed as $\lambda_x := \frac{\triangle t}{\triangle x}$ with constant $\lambda_x > 0$. Then the three-point finite difference scheme (2.20) is consistent with the advection equation (2.1) iff $\alpha + \beta + \gamma = 1$ and $\gamma - \alpha = \lambda_x a$.*

Proof: For the three point scheme (2.20)

$$v_i^{n+1} = \alpha v_{i+1}^n + \beta v_i^n + \gamma v_{i-1}^n \tag{2.27}$$

we make a Taylor expansion to obtain at (t_n, x_i)

$$\begin{aligned}
u\,(t_{n+1}, x_i) &= u(t_n,x_i) + \triangle t \partial_t u(t_n,x_i) + \tfrac{\triangle t^2}{2}\partial_{tt} u(t_n,x_i) + \mathcal{O}(\triangle t^3) \\
u\,(t_n, x_{i+1}) &= u(t_n,x_i) + \triangle x \partial_x u(t_n,x_i) + \tfrac{\triangle x^2}{2}\partial_{xx} u(t_n,x_i) + \mathcal{O}(\triangle x^3) \\
u\,(t_n, x_{i-1}) &= u(t_n,x_i) - \triangle x \partial_x u(t_n,x_i) + \tfrac{\triangle x^2}{2}\partial_{xx} u(t_n,x_i) + \mathcal{O}(\triangle x^3).
\end{aligned} \tag{2.28}$$

Replacing the values in (2.20) with (2.28), we get

$$\begin{aligned} u + \triangle t\partial_t u + \tfrac{\triangle t^2}{2}\partial_{tt} u &= (\alpha+\beta+\gamma)\, u + \triangle x\,(\alpha-\gamma)\,\partial_x u \\ &\quad + \tfrac{\triangle x^2}{2}\,(\alpha+\gamma)\,\partial_{xx} u \\ &\quad + \mathcal{O}(\triangle t^3) + \mathcal{O}(\triangle x^3) \\ \iff \qquad \partial_t u &= (\alpha+\beta+\gamma-1)\, u - \tfrac{1}{\lambda_x}\,(\gamma-\alpha)\,\partial_x u \\ &\quad + \tfrac{\triangle x^2}{2\triangle t}\,(\alpha+\gamma)\,\partial_{xx} u - \tfrac{\triangle t}{2}\partial_{tt} u \\ &\quad + \mathcal{O}(\triangle t^2) + \mathcal{O}(\triangle x^2). \end{aligned} \tag{2.29}$$

□

Since consistency requires two linear constraints for the parameters α, β, γ of the three point scheme (2.20), we only have one degree of freedom. Setting $q = \alpha + \gamma$ we can write the explicit three point schemes in the so called **viscous form**

$$v_i^{n+1} = v_i^n - \frac{c\lambda_x}{2}\left(v_{i+1}^n - v_{i-1}^n\right) + \frac{q}{2}\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) \tag{2.30}$$

with artificial viscosity $\frac{q}{2}$. The smaller q, the smaller the dissipation of the numerical solution is. In table 2.1 the finite difference schemes are listed with their respective coefficients.

Consistency implies that the solution of the partial differential equation is an approximate solution of the finite difference scheme, if it is smooth. In the same manner, convergence means that a solution of the finite difference scheme approximates a solution of the partial differential equation. However consistency is necessary but not sufficient for convergence.

For that we need another condition which is the **stability** of a finite difference scheme.

2.3 Stability

Before giving the definition of stability, we need to define the stability region. For most of the finite difference schemes, there are certain restrictions for $\triangle t$ and $\triangle x$ such that the scheme is stable. Hence a stability region must contain a sequence $(\triangle t_p, \triangle x_p)$ such that it converges to the origin as $p \to \infty$.

Definition 2.3.1. ***Stability*** *A one-step finite difference scheme $P_{\triangle t,\triangle x} u_i^n = 0$ for the first order equation (2.10) is stable in the norm $\|\cdot\|_{\cdot,\triangle x}$ in a stability region Ω if for every $T > 0$ there exists a constant C_T such that, for each u^0,*

$$\| u^n \|_{\cdot,\triangle x} \leq C_T \| u^0 \|_{\cdot,\triangle x}, \tag{2.31}$$

for $0 \leq n\triangle t \leq T$.

scheme	(α, β, γ)	q	Convergence	Order
upwind (c<0)	$(-c\lambda_x, 1+c\lambda_x, 0)$	$-c\lambda_x$	$-1 \leq c\lambda_x \leq 0$	(1,1)
upwind (c>0)	$(0, 1-c\lambda_x, c\lambda_x)$	$\lambda_x c$	$0 \leq c\lambda_x \leq 1$	(1,1)
FTCS	$(\frac{-c\lambda_x}{2}, 1, \frac{c\lambda_x}{2})$	0	unstable	(-,-)
Lax-Friedrich	$(\frac{1-c\lambda_x}{2}, 0, \frac{1+c\lambda_x}{2})$	1	$\lvert c\lambda_x\rvert \leq 1$	(1,1)
Lax-Wendroff	$(\frac{c\lambda_x+c^2\lambda_x^2}{2}, 1-c^2\lambda_x^2, (\frac{-c\lambda_x+c^2\lambda_x^2}{2})$	$c^2\lambda_x^2$	$\lvert c\lambda_x\rvert \leq 1 (l^2-stable)$	(2,2)

Table 2.1: Three point schemes (2.20) for the advection equation (2.1).

scheme	θ	Convergence	Order
FTCS	0	$D\mu_x \leq \frac{1}{2}$	(1,2)
BTCS	1	unconditonally stable	(1,2)
Crank-Nicolson	$\frac{1}{2}$	unconditionally stable in l^2-norm $D\mu_x \leq 1$ in l^∞-norm	(2,2)

Table 2.2: θ-scheme (2.24) for the diffusion equation (2.2) for $\mu_x = \frac{\triangle t}{\triangle x^2}$.

Now we have defined the concept of stability, consistency and convergence. The connection of these three concepts can be seen in the Lax-Richtmeyer equivalence theorem, which is a fundamental theorem in the theory of finite difference schemes for initial value problems.

Theorem 2.3.2. *(**Lax-Richtmeyer**) A consistent finite difference scheme for a linear partial differential equation for which the initial value problem is well-posed is convergent iff it is stable.*

Proof: See [134].

This theorem proves to be very useful in determining whether a finite difference scheme is convergent by avoiding the difficult proof of convergent itself, but proving consistency and stability instead, which is much easier to do.
In the next section we will introduce procedures how to prove stability in an easier way.

2.3.1 $\mathcal{L}^\infty$ Stability and Monotonicity

One way to prove stability is by finding estimates in the maximum norm, which can be done by verifying that the monotone comparison property holds [102]. This means that if two discretized initial data $v_i^0, \tilde{v}_i^0$ are such that $v_i^0 \leq \tilde{v}_i^0$, then for all $n > 0$ the inequality $v_i^n \leq \tilde{v}_i^n$ holds.

Definition 2.3.3. *A one step scheme of the form (2.13) is monotone if for all coefficients*

$$c_i \geq 0. \tag{2.32}$$

Theorem 2.3.4. *Let a monotone scheme in the form (2.13) be given. Then, setting*

$$v_{min} := \min\{v_i^0\}, \quad v_{max} := \max\{v_i^0\} \tag{2.33}$$

we have

$$v_{min} \leq v_i^n \leq v_{max}, \qquad \text{for all } i \text{ and } n. \tag{2.34}$$

Proposition 2.3.5. *The explicit three-point scheme in viscous form (2.30), which is consistent to the advection equation (2.1) is monotone iff*

$$c|\lambda_x| \leq q \leq 1. \tag{2.35}$$

Proof: We sort the coefficients c_i of (2.30) accordingly and require $c_i \leq 0$.

$$v_i^{n+1} = (1-q)\, v_i^n + \left(-\frac{c\lambda_x}{2} + \frac{q}{2}\right) v_{i+1}^n + \left(\frac{c\lambda_x}{2} + \frac{q}{2}\right) v_{i-1}^n. \tag{2.36}$$

By checking the inequality of each coefficient we get the monotone conditions $q \leq 1$ and $q \geq c|\lambda_x|$.

□

In the same way we can derive monotone conditions for the θ-scheme (2.24) for the diffusion equation (2.2) with $\mu_x = \frac{\triangle t}{\triangle x^2}$

$$\begin{aligned}(1+2D\mu_x\theta)v_i^{n+1} - D\mu_x\theta v_{i-1}^{n+1} - D\mu_x\theta v_{i+1}^{n+1} &= (1-2D\mu_x(1-\theta))\, v_i^n \\ &\quad + D\mu_x(1-\theta)v_{i+1}^n + D\mu_x(1-\theta)v_{i-1}^n\end{aligned} \tag{2.37}$$

and by requiring the coefficients of $v_i^n, v_{i+1}^n, v_{i-1}^n$ to be positive we get as a condition for monotonicity

$$D\mu_x(1-\theta) \leq \frac{1}{2}. \tag{2.38}$$

The stability conditions can be found in table 2.2.
Schemes that are monotone and consistent are, according to the Lax-Richtmeyer theorem 2.3.2, convergent.
In the next section we will investigate stability in the l^2-norm with the help of Fourier analysis.

2.4 Von Neumann Stability Analysis

With the help of Fourier analysis we will introduce a very powerful tool to study and analyse partial differential equations and finite difference schemes with which the Von Neumann analysis of stability is derived to give necessary and sufficient stability conditions for the stability of finite difference schemes in a generally more applicable way.
We will refer to [102, 134] for a more detailed explanation on Fourier analysis and will just mention the Parseval's relations theorem on which the Von Neumann analysis is based on.

Theorem 2.4.1 (Parseval's relations). *Let $u(x) \in C^1$ and be $\widehat{u}(\omega)$ its Fourier transform, then with the L^2-norm the equality*

$$\| u \|_2 = \| \widehat{u} \|_2 \tag{2.39}$$

and for grid function v and its discrete Fourier transform the equality

$$\| v \|_{\triangle x}^2 = \| \widehat{v} \|_{\triangle x}^2 \tag{2.40}$$

hold.

With Parseval's relations we can replace the equation of the stability definition 2.3.1 with the equivalent inequality in its Fourier transform

$$\| \widehat{v}^n \|_{\triangle x} \leq C_T^* \sum_{j=0}^{J} \| \widehat{v}^j \|_{\triangle x} \, . \tag{2.41}$$

It is worth mentioning that the Parseval relation is not valid in the l^∞-norm.
With this powerful tool we can now introduce the von Neumann analysis of stability of finite difference schemes which is an important application of the Fourier analysis.

We will generically show the method on the upwind scheme (2.14) for the advection equation (2.1) with $\lambda_x = \frac{\triangle t}{\triangle x}$ and $c > 0$.
For the Upwind scheme

$$v_m^{n+1} = (1 - c\lambda_x)\, v_m^n + c\lambda_x v_{m-1}^n \tag{2.42}$$

we substitute $v_m^{n+1}, v_{m+1}^n, v_{m-1}^n$ with their discrete Fourier inversion [134] and obtain

$$\frac{1}{2\pi} \int_{-\frac{\pi}{\triangle x}}^{\frac{\pi}{\triangle x}} e^{im\triangle x\xi} \widehat{v}^{n+1}(\xi) d\xi = \frac{1}{2\pi} \int_{-\frac{\pi}{\triangle x}}^{\frac{\pi}{\triangle x}} e^{im\triangle x\xi} \left[(1 - c\lambda_x) + c\lambda_x e^{-i\triangle x\xi} \right] \widehat{v}^n(\xi) d\xi. \tag{2.43}$$

By comparing both terms in the integral (2.43) pointwise, we get

$$\widehat{v}^{n+1}(\xi) = g(\triangle x\xi)\widehat{v}^{n}(\xi) \tag{2.44}$$

with amplification factor $g(\triangle x\xi) = (1 - c\lambda_x) + c\lambda_x e^{-i\triangle x\xi}$ and thus

$$\widehat{v}^{n+1} = g^{n+1}(\triangle x\xi)\widehat{v}^{0}, \tag{2.45}$$

which shows that advancing the solution by $n+1$ time steps is equivalent with multiplying the Fourier transform of the solution by the amplification factor. If we use (2.44) on the definition of stability 2.3.1 with the Parseval's relations (2.4.1) we reveal

$$\begin{aligned} \| \, v^n \, \|^2_{\triangle x,2} &= \int_{-\frac{\pi}{\triangle x}}^{\frac{\pi}{\triangle x}} |\widehat{v}^n|^2(\xi) d\xi \\ &= \int_{-\frac{\pi}{\triangle x}}^{\frac{\pi}{\triangle x}} |g(\triangle x\xi)|^{2n} |\widehat{v}^0(\xi)|^2 d\xi \end{aligned} \tag{2.46}$$

and the stability inequality (2.31) holds if $|g(\triangle x\xi)|^{2n}$ is suitably bounded. For the amplification factor $g(\theta)$ with $\theta = \triangle x\xi$ of the upwind scheme we have

$$\begin{aligned} |g(\theta)|^2 &= |\,(1 - c\lambda_x) + c\lambda_x e^{-i\theta}|^2 \\ &= \left(1 - 2c\lambda_x \sin^2(\tfrac{\theta}{2})\right)^2 + 4c^2\lambda_x^2 \sin^2(\tfrac{\theta}{2}) \cos^2(\tfrac{\theta}{2}) \\ &= 1 - 4c\lambda_x \left(1 - 4c\lambda_x\right) \sin^2(\tfrac{\theta}{2}) \\ \Rightarrow \quad |g(\theta)|^2 &\leq 1 \quad \text{for } 0 \leq c\lambda_x \leq 1. \end{aligned} \tag{2.47}$$

A simpler and equivalent procedure is to substitute v_m^n in the finite difference scheme with $g^n e^{im\theta} = v_m^n$ to obtain the amplification factor.

Although we derived a stability criteria for just the upwind scheme, we can generalize the results in the following theorem.

Theorem 2.4.2. *A one-step finite difference scheme with constant coefficients is stable in a stability region Λ iff there is a constant K, independent of $\theta, \triangle t$ and $\triangle x$, such that*

$$|g\,(\theta, \triangle t, \triangle x)\,| \leq 1 + K\triangle t \tag{2.48}$$

with $(\triangle t, \triangle x) \in \Lambda$. If $g\,(\theta, \triangle t, \triangle x)$ is independent of $\triangle t$ and $\triangle x$, the stability criteria (2.48) can be replaced with the restricted stability condition

$$|g(\theta)| \leq 1. \tag{2.49}$$

Theorem 2.4.2 gives us a simple but powerful tool to analyze stability in l^2-norm of one step finite difference methods by only investigating the respective amplification factor $g(\theta)$.

Stability conditions derived with the Von Neumann analysis for other finite difference schemes are indicated in table 2.1 and 2.2.

Remark 2.4.3. *We want to point out that stability derived from the verification of the monotonicity condition 2.3.3 is often more restrictive than in l^2-norm which can be seen for the Crank-Nicolson-scheme (2.23), consistent to the diffusion equation (2.2). Applying the Von Neumann stability analysis tells us that the scheme is unconditionally stable in l^2-norm, whereas with the monotonicity condition we have the stability restriction $D\mu_x \leq 1$.*

The question whether something changes when viewing schemes for inhomogeneous linear partial differential equations can be answered with the following corollary.

Corollary 2.4.3.1. *If a scheme is modified so that the modifications result only in the addition to the amplification factor of terms that are $\mathcal{O}(\triangle t)$ uniformly in ξ, then the modified scheme is stable iff the original scheme is stable.*

The corollary states that for stability the lower order terms such as function f does not influence the stability condition and hence only the finite difference scheme for homogeneous hyperbolic partial differential equations are needed to be analysed.

Remark 2.4.4. *The stability condition in theorem 2.4.2 does not apply directly to linear partial differential equations with variable coefficients, but we can still use the stability condition by considering each of the frozen coefficient problems. These are constant coefficient problems obtained by fixing the coefficients at their values at each point of the domain of the computation. If each of those frozen coefficient problems is stable, then the variable coefficient problem is stable as well [134].*

Although the Von Neumann stability analysis can give us necessary and sufficient stability conditions for initial value problems of linear partial differential equations. It is strictly only valid for the scheme excluding boundary conditions due to the nature of Fourier Analysis. However these conditions are often only necessary conditions for non-linearities and complicated boundary

conditions. Here the Von Neumann stability analysis may only be applied locally on linearised equations.

We have now established several strategy to derive necessary and sufficient stability conditions with the monotonicity condition in 2.3.3 and with the Von Neumann stability analysis.
We refer to Section 2.8 where we briefly discuss the theories to check explicitly the stability of boundary conditions and the matrix method which is an alternative approach to derive stability conditions for finite difference schemes with boundary conditions.

2.5 Order of Accuracy

In the previous sections we have introduced the concepts of consistency, convergence and stability, but left out the concept of order of accuracy which gives us information about the accuracy of finite difference schemes. We will focus about it in this section.

Definition 2.5.1 (Order). *A scheme $P_{\triangle t,\triangle x}v = R_{\triangle t,\triangle x}f$ that is consistent with the partial differential equation $Pu = f$ is accurate of order p in time and order q in space, if for any smooth function $\phi(t,x)$,*

$$P_{\triangle t,\triangle x}\phi - R_{\triangle t,\triangle x}P\phi = \mathcal{O}(\triangle t^p) + \mathcal{O}(\triangle x^q). \tag{2.50}$$

We say that such a scheme is accurate of order (p,q).
*The quantity $T(t,x) := P_{\triangle t,\triangle x}\phi(t,x) - R_{\triangle t,\triangle x}P\phi(t,x)$ is called **truncation error** of the scheme.*

Because in not every scheme $\triangle t$ and $\triangle x$ are independent from each other, we give a more general definition of order which takes into account the time step as a smooth function of the space step $\triangle t = \Lambda(\triangle x)$.

Definition 2.5.2 (Order). *A scheme $P_{\triangle t,\triangle x}v = R_{\triangle t,\triangle x}f$ with $\triangle t = \Lambda(\triangle x)$ that is consistent with he differential equation $Pu = f$ is accurate of order r, if for any smooth function $\phi(t,x)$,*

$$P_{\triangle t,\triangle x}\phi - R_{\triangle t,\triangle x}P\phi = \mathcal{O}(\triangle t^r). \tag{2.51}$$

It is important to note that in the definition of consistency 2.2.2 the truncation error only requires to be $T(x,t) = o(1)$ whereas the definition of order 2.51 the more specific definition of the truncation error with $T(x,t) = \mathcal{O}(\triangle t^p) + \mathcal{O}(\triangle x^q)$.

As an example, we will inspect the order of the three point scheme (2.20) for the advection equation (2.1).
We can use the truncation error and the Taylor expansion from (2.28) with the conditions of proposition (2.2.3) to obtain

$$T(t_n, x_i) = \frac{\triangle x^2}{2\triangle t} q \partial_{xx} u - \frac{\triangle t}{2} \partial_{tt} u + \mathcal{O}(\triangle t^2) + \mathcal{O}(\triangle x^2). \tag{2.52}$$

Now we can use the advection equation (2.1) itself to derive the second time derivative of u

$$\begin{aligned} \partial_{tt} u &= -c \partial_t \partial_x u \\ \partial_{tt} u &= c^2 \partial_{xx} u \end{aligned} \tag{2.53}$$

and substituting (2.53) into the truncation error (2.52) leads to

$$T(t_n, x_i) = \left(\frac{\triangle x^2}{2\triangle t} q - \frac{c^2 \triangle t}{2} \right) \partial_{xx} u + \mathcal{O}(\triangle t^2) + \mathcal{O}(\triangle x^2). \tag{2.54}$$

If we fix $\lambda_x := \frac{\triangle t}{\triangle x}$ we notice that the three point schemes are of order 1 in time and space and only for $q = (\lambda_x c)^2$ the scheme is order 2 in time and space. This scheme is the Lax-Wendroff scheme.

Also the choice of the linear operator $R_{\triangle t, \triangle x}$ for the discretization of the function f affects the order of the scheme as well.
As an example we take the **Lax-Wendroff method**

$$v_i^{n+1} = v_i^n - \frac{c\lambda_x}{2} \left(v_{i+1}^n - v_i^n \right) + \frac{c^2 \lambda_x^2}{2} \left(v_{i+1}^n - 2v_i^n + v_{i-1}^n \right) + \triangle t R_{\triangle t, \triangle x} f_i^n \tag{2.55}$$

applied to the inhomogeneous advection equation

$$\partial_t u + c \partial_x u = f \tag{2.56}$$

with a consistent linear operator $R_{\triangle t, \triangle x}$ such that $R_{\triangle t, \triangle x} f_i^n \to f_i^n$ for $\triangle t, \triangle x \to 0$.
We will demonstrate this on the following two consistent linear operators:

(i) $R_{\triangle t, \triangle x} f_i^n = \frac{1}{2} \left(f_i^{n+1} + f_i^n \right) - \frac{c\lambda_x}{4} \left(f_{i+1}^n - f_{i-1}^n \right)$.

(ii) $R_{\triangle t, \triangle x} f_i^n = f_i^n$.

We use the Taylor expansion on

$$P_{\triangle t, \triangle x} \phi = \frac{\phi_i^{n+1} - \phi_i^n}{\triangle t} + c \frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\triangle x} - \frac{c^2 \lambda_x^2}{2} \frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{\triangle x^2} \tag{2.57}$$

evaluated at (t_n, x_i)

$$P_{\triangle t, \triangle x} \phi = \partial_t \phi + \frac{\triangle t}{2} \partial_{tt} \phi + c \partial_x \phi - \frac{c^2 \lambda_x^2}{2} \partial_{xx} \phi + \mathcal{O}(\triangle t^2) + \mathcal{O}(\triangle x^2) \tag{2.58}$$

and for (i) with $f = \partial_t \phi + c\partial_x \phi = P\phi$

$$R_{\triangle t,\triangle x} f = f + \frac{\triangle t}{2}\partial_t f - \frac{c\triangle x}{2}\partial_x f + \mathcal{O}(\triangle t^2) + \mathcal{O}(\triangle x^2) \tag{2.59}$$

$$R_{\triangle t,\triangle x} P\phi \;=\; \partial_t \phi + c\partial_x \phi + \tfrac{\triangle t}{2}\partial_{tt}\phi - \tfrac{c^2\triangle t^2}{2}\partial_{xx}\phi + \mathcal{O}(\triangle t^2) + \mathcal{O}(\triangle x^2). \tag{2.60}$$

This means for the truncation error $T(t,x) = P_{\triangle t,\triangle x}\phi - R_{\triangle t,\triangle x} P\phi = \mathcal{O}(\triangle t^2) + \mathcal{O}(\triangle x^2)$ and hence the Lax-Wendroff scheme is of order $(2,2)$.
If we choose for $R_{\triangle t,\triangle x} f$ (ii) instead, we would get

$$R_{\triangle t,\triangle x} f = f = \partial_t \phi + c\partial_x \phi \tag{2.61}$$

and for the truncation error $T(t,x) = P_{\triangle t,\triangle x}\phi - R_{\triangle t,\triangle x} P\phi = \mathcal{O}(\triangle t) + \mathcal{O}(\triangle x^2)$. Hence the method would be of order $(1,2)$.

We want to mention also that for non-smooth initial conditions additional criteria must be fulfilled in order for the scheme to preserve its order of accuracy, otherwise an order reduction can occur. We refer to [134] for further details on this topic.

2.6 Convection-Diffusion Equation

In the previous sections we have introduced the concepts of consistency, stability, order of accuracy and convergence and applied these on finite difference schemes for the diffusion and advection equation.
This section is devoted to give a through insight into the convection-diffusion equations

$$\partial_t u + c\partial_x u = D\partial_{xx} u.$$

These types of partial differential equations are a combination of a hyperbolic term (convection) and a parabolic term (diffusion) and are the foundation of the mathematical model we use to simulate chemotaxis, such as the Keller-Segel model (1.31).

It is straightforward to use finite difference scheme (2.30) for the convection part of the equation and a central discretization for the second order derivative in space.
We set $\mu_x := \frac{\triangle t}{\triangle x^2}$ and $\lambda_x := \frac{\triangle t}{\triangle x}$ and have the finite difference scheme

$$v_i^{n+1} = v_i + \left[D\mu_x + \frac{q}{2}\right]\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) - \frac{c\lambda_x}{2}\left(v_{i+1}^n - v_{i-1}^n\right) \tag{2.62}$$

with artificial viscosity parameter $q \in \mathbb{R}$.
Although we could use the Von Neumann stability analysis, we take advantage of the monotonicity property of the convection-diffusion equations and use the criteria for monotonicity in 2.3.3 to show stability in the l^∞-norm.

We recall that the main property of parabolic partial differential equation is that for the solution $u(x,t)$

$$\sup_{x\in\Omega} |u(x,t)| \leq \sup_{x\in\Omega} |u(x,t')| \text{ for } t > t'. \tag{2.63}$$

If we want the numerical solutions of the finite difference scheme to have similar properties, we want the condition

$$\max_{i\in J_\Omega} |v_i^{n+1}| \leq \max_{i\in J_\Omega} |v_i^n| \tag{2.64}$$

to be fulfilled.
With the help of the monotonicity conditions we organize the factors of $v_i^n, v_{i+1}^n, v_{i-1}^n$ and acquire the conditions

$$\begin{aligned} 1 - 2D\mu_x - q &\geq 0, \\ D\mu_x + \tfrac{q}{2} \pm \tfrac{c\lambda_x}{2} &\geq 0, \end{aligned} \tag{2.65}$$

which leads to the conditions

$$\begin{aligned} \mu_x &\leq \tfrac{1-q}{2D}, \\ \triangle x &\leq \tfrac{2D}{c} + \tfrac{q}{c\lambda_x}. \end{aligned} \tag{2.66}$$

As expected, the first condition of (2.66) gives us the stability criterion for that scheme that restricts the time step size $\triangle t$ in dependence on the mesh grid size $\triangle x$ with the diffusion coefficient D.
The second condition of (2.66) on the other hand is not a stability condition since stability only deals with the limit when $\triangle t$ and $\triangle x$ tend to zero. It is only restricting the mesh grid size $\triangle x$ and is always fulfilled for small enough $\triangle x$.
If this condition is violated, oscillations will occur which are only the result of inappropriate mesh grid resolution.
In table 2.3 we have listed several explicit finite difference schemes with their respective stability condition.

Remark 2.6.1. *The quantity $\frac{cL}{D}$ with characteristic length L corresponds to the Reynold number in fluid dynamics or the Peclet number in heat flow.*

scheme	q	Stability condition	Order
upwind(c>0)	$\lambda_x c$	$\mu_x \leq \frac{1}{2D+c\triangle x}$	(1,1)
FTCS	0	$\mu_x \leq \frac{1}{2D}$ $\triangle x \leq \frac{2D}{c}$	(1,2)
Lax-Friedrich	1	unstable	(-,-)
Crank-Nicolson-CS(IMEX)	0	$\mu_x \leq \frac{1}{D}$ $\triangle x \leq \frac{2D}{c}$	(1,2)

Table 2.3: Explicit finite difference schemes (2.62) for the convection-diffusion equation (2.3).

We can see from table 2.3 that the stability restriction are rather harsh for convection dominant equations. For the FTCS-scheme a high Peclet number would enforce very small mesh grid size to avoid oscillations and therefore even smaller time step size to ensure the stability of the scheme.
A remedy can be achieved by discretizing the first order term $c\partial_x u$ with an upwind method which stability restriction becomes less restrictive for larger Peclet numbers compared to the FTCS-schemes.
We can also observe that the introduction of artificial viscosity can greatly loosen the restrictions on the time step $\triangle t$ and mesh grid size $\triangle x$.
This will enable us to have an efficient and less costly implementation of the chemotaxis model (1.45) where we deal with highly convection dominant equations.

2.7 IMEX-Schemes

The previously discussed finite difference schemes are divided into groups of explicit methods, where only the current vector $\mathbf{v^n}$ is needed for the calculation of the next time iteration $\mathbf{v^{n+1}}$. These methods are never unconditionally stable and certain stability restrictions need to be enforced [102].
The other group of methods are implicit methods, where beside the current vector $\mathbf{v^n}$ also $\mathbf{v^{n+1}}$ are needed for the calculation of $\mathbf{v^{n+1}}$. This requires solving a linear equation system if all terms are linear but for non-linear terms, a nonlinear equation system must be solved. How to solve those systems will be discussed later in Chapter 4.
The advantage of avoiding very restrictive stability conditions is tied to the disadvantage of high computational cost of solving such equation systems. The idea is to use a different discretization methods on different terms of the partial differential equations. This gives rise to Implicit-Explicit schemes (IMEX) [9]. An example is a convection diffusion equation with non-linear

convection term $\partial_x (f(u))$ where an implicit discretization is used on the linear diffusion term and an explicit discretization on the non-linear convection term.
This would allow us to improve the stability restriction by solving a linear equation system in each time iteration step but avoid solving a non-linear equation system.

In order to appropriately define IMEX-methods, we extend the definition of finite difference schemes and introduce the Method of Lines [84, 123], which is the discretization in all but one dimension.
The method of lines dates back to at least the early 60's. Many papers discussing the accuracy and stability of the method of lines for various types of partial differential equations have appeared since [33, 81, 89].
If we take the general first order in time partial differential equation

$$\begin{cases} \partial_t u(t,x) &= P(\partial_x) u(t,x), & \text{for } x \in \Omega,\, t \geq 0, \\ u(0,x) &= u_0(x), & \text{for } x \in \Omega \end{cases} \tag{2.67}$$

and only discretize the spatial dimension, we get the ordinary differential equation system

$$\begin{cases} \frac{d}{dt} u_i(t) &= P_{\triangle x} u_i(t), & \text{for } t > 0, \\ u(0)_i &= u_0(x_i) \end{cases} \tag{2.68}$$

with $i \in J_\Omega$ and appropriate linear operator $P_{\triangle x}$ for the spatial discretization, on which we can apply methods to numerically solve such equations (Runge-Kutta-Methods, BDF-Methods and others) [32, 121, 135].
The MOL-methods do require that the partial differential equation is a well-posed initial value problem because the numerical integrators for ordinary differential equation are initial value problem solvers.
The finite difference methods in the preceding sections can be identified as MOL methods. For an initial value ordinary differential equation

$$\begin{cases} \frac{du(t)}{dt} &= F(t, u(t)), \\ u(0) &= u_0 \end{cases} \tag{2.69}$$

the following examples are one-step methods for the approximated solution

- **Explicit Euler method**

$$u^{n+1} = u^n + \triangle t F(t_n, u(t_n)) \tag{2.70}$$

- **Implicit Euler method**

$$u^{n+1} = u^n + \triangle t F(t_{n+1}, u(t_{n+1}) \tag{2.71}$$

- **Classical Runge-Kutta** (explicit)

$$\begin{aligned} u^{n+1} &= u^n + \tfrac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \\ k_1 &= F(t_n, u^n), \\ k_2 &= F(t_n + \tfrac{\triangle t}{2}, u^n + \triangle t \tfrac{k_1}{2}), \\ k_3 &= F(t_n + \tfrac{\triangle t}{2}, u^n + \triangle t \tfrac{k_2}{2}), \\ k_4 &= f(t_n + \triangle t, u^n + \triangle t k_3) \end{aligned} \tag{2.72}$$

- **Trapezoidal rule** (implicit)

$$u^{n+1} = u^n + \frac{\triangle t}{2}\left(F(t_n, u^n) + F(t_{n+1}, u^{n+1})\right). \tag{2.73}$$

There is a rich literature presenting concepts such as A-,B- and L-stability, order of accuracy and the derivation of other one step and multi step methods which can be found in [32, 39, 66, 121, 135, 144].
Applying one step methods on the ordinary differential equation (2.68), we recover the previously derived finite difference schemes.
We can confirm that for the forward in time finite difference methods the first order explicit Euler method (2.70) have been used, whereas for the backward in time finite difference schemes the first order implicit Euler method (2.71) has been applied.
Even the Crank-Nicolson method for the diffusion equation (2.23) can be derived by using the second order trapezoidal rule (2.73).
The main idea behind IMEX-schemes is to divide the ordinary differential equation (2.69) into two parts

$$\frac{d}{dt}u_i(t) = \underbrace{F(u_i(t))}_{\text{non stiff, non-linear}} + \underbrace{G(u_i(t)),}_{\text{stiff and linear}} \tag{2.74}$$

on which we use an explicit one step method on function F and an implicit one step method on G.
Usually the function F contains the non-linear terms of the partial differential equations which would be very difficult to solve numerically, especially when non-linear, whereas the function G contains stiff linear terms of the partial differential equation, which require solving a much easier linear equation.
With the θ-scheme (2.24) we already have seen an IMEX-method with

$$\frac{du(t)}{dt} = \underbrace{\theta P_{\triangle x} u}_{\text{explicit Euler method}} + \underbrace{(1-\theta)\, P_{\triangle x} u}_{\text{implicit Euler}} \tag{2.75}$$

and the advantage achieved was by choosing $\theta = \frac{1}{2}$ which increased the approximation order in time to 2 and benefit from unconditional stability for the diffusion equation (2.2).
Also in Section 2.6 in table 2.3 we already listed an IMEX method for the convection-diffusion equation (2.3). For

$$\partial_t u(t,x) = \underbrace{D\partial_{xx}u(t,x)}_{\text{spatial discretization with } F} - \underbrace{c\partial_c u(t,x)}_{\text{spatial discretization with } G} \tag{2.76}$$

we attain

$$\frac{du}{dt} = \underbrace{\frac{1}{2}F(u)}_{\text{explicit Euler}} + \underbrace{\frac{1}{2}F(u)}_{\text{implicit Euler}} + \underbrace{G(u)}_{\text{explicit Euler}} . \tag{2.77}$$

If we compare the order of accuracy and stability condition in table 2.3 we do not notice an increase in the order in time and just a marginally better stability condition which raises the question why an implicit method is applied in the first place.
The answer lies within the spatial discretization of the diffusion term, which leads to a so called stiff term.
Stiff terms dictate the time step size due to stability constraints which requires the usage of implicit methods.
Using the Crank-Nicolson scheme also on the non-linear convection term $\partial_x (f(u))$ would increase the order in time to 2, but makes it necessary to not solve just a linear equation system which can be calculated efficiently, but also to solve a non-linear equation system which can be very costly and computationally complex.
We refer to [39, 144] for the proper definition and further information of stiffness.

2.8 Boundary Conditions

Until now we have considered finite difference schemes in order to approximate solutions to partial differential equations. The implementation of the initial value $u(0,x) = u_0(x)$ has been used as the initialization $v^0 = u_0$ for the finite difference schemes. These initial value problems were only considered on the real line $(-\infty, \infty)$ where we did not encompass the boundary values.
In order to solve initial-boundary value problems, we must use the boundary conditions required by the partial differential equations in order to determine uniquely solutions.
As we presented in the previous chapter, we have different kind of boundary conditions which are needed for a well-posed initial-value problem.
For a partial differential equation on $\Omega \times \mathbb{R}^+$ with boundary $\delta\Omega$ and smooth function $g \in C^1$ we have

- **Dirichlet-Boundary Condition**

$$u(t,x) = g(t,x), \tag{2.78}$$

for $x \in \delta\Omega$ and $t > 0$.

- **Neumann-Boundary Condition**

$$\frac{\partial u(t,x)}{\partial \boldsymbol{n}} = g(t,x), \tag{2.79}$$

for $x \in \delta\Omega$, outer normal $\boldsymbol{n}$ and $t > 0$.

- **Robin-Boundary Condition**

$$\alpha u(t,x) + \beta \frac{\partial u(t,x)}{\partial \mathbf{n}} = g(t,x), \tag{2.80}$$

for $x \in \delta\Omega$, outer normal $\mathbf{n}$, $\alpha, \beta \in \mathbb{R}_{\geq 0}$ and $t > 0$.

We will illustrate the inclusion of these boundary conditions for the advection equation (2.1) on the Lax-Wendroff scheme for $c > 0$ (2.55):

$$v_i^{n+1} = v_i^n - \frac{c\lambda_x}{2}\left(v_{i+1}^n - v_{i-1}^n\right) + (\frac{c\lambda_x}{2})^2\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right). \tag{2.81}$$

As we explained in Chapter 1.2, the well-posed advection equation (2.1) requires only one boundary condition, depending on the sign of the velocity c. In the following we choose $c > 0$ and hence the boundary condition on the left side of the domain at $x = 0$.
For $i = 1$ the Lax-Wendroff scheme reads

$$v_1^{n+1} = v_1^n - \frac{c\lambda_x}{2}\left(v_2^n - v_1^n\right) + (\frac{c\lambda_x}{2})^2\left(v_2^n - 2v_1^n + v_0^n\right). \tag{2.82}$$

As we can see, v_0^n is necessary in order to compute v_1^{n+1}.
If we deal with Dirichlet-Boundary Conditions (2.78) with $u(t,0) = g(t)$, we can directly apply $v_0^n = g(t_n)$ for all $t > 0$.
With Neumann-Boundary Conditions (2.79) $\frac{\partial u(t,0)}{\partial \boldsymbol{n}} = g(t)$ we only have the outer derivative at $x = 0$ given, which we cannot apply directly to the upwind scheme.
For this reason there are several ways.
A very easy and common solution is to use finite differences on the outer normal derivative to obtain for example a one-sided finite difference

$$\begin{aligned} \tfrac{u_1^n - u_0^n}{\triangle x} &= g(t_n) \\ \Leftrightarrow \quad u_0^n &= u_1^n - \triangle t g(t_n), \qquad \text{for all } n = \mathbb{N}_0. \end{aligned} \tag{2.83}$$

and obtain a formula for the boundary value v_0^{n+1}.
As expected these are, unlike for Dirichlet boundary conditions, only approximation for v_0^{n+1} as we used a first order difference scheme to approximate the outer derivative.
Another possibility is to combine the scheme at the boundary with $i = 0$ such that

$$v_0^{n+1} = v_0^n - \frac{c\lambda_x}{2}\left(v_1^n - v_{-1}^n\right) + (\frac{c\lambda_x}{2})^2\left(v_1^n - 2v_0^n + v_{-1}^n\right). \tag{2.84}$$

Although we have now the formula to determine v_0^{n+1} with the scheme (2.84), we have an unknown value v_{-1}^n outside the domain Ω, which is also called a **ghost value**.
We can determine such a value by approximating the Neumann condition with a central difference

$$\begin{aligned} & \tfrac{v_1^n - v_{-1}^n}{2\triangle x} &&= g(t_n) \\ \Leftrightarrow \quad & v_{-1}^n &&= v_1^n - 2\triangle x g(t_n) \end{aligned} \tag{2.85}$$

and apply the ghost value (2.85) to the Lax-Wendroff scheme (2.84) at $i = 0$ to obtain

$$v_0^{n+1} = v_0^n - c\lambda_x \triangle x g(t_n) + (\frac{c\lambda_x}{2})^2\left(2v_1^n - 2v_0^n - 2\triangle x g(t_n)\right). \tag{2.86}$$

Not only do schemes sometimes need to be modified so that the boundary conditions can be applied, but also some schemes do need additional values which are not given by the partial differential equation.
If we look at the domain $\Omega = [0, x_{end}]$ with $x_i = i\triangle x$ and $x_{end} = (M+1)\triangle x$, we see that with the Lax-Wendroff method in order to compute v_M^{n+1} and v_{M+1}^{n+1} additional boundary conditions are needed, so called **numerical boundary condition**.
Those numerical boundary conditions should be some form of extrapolation that determines the solution on the boundary in regards to the solution in the interior domain. Typical examples are

- $v_{M+1}^{n+1} = v_M^{n+1}$
- $v_{M+1}^{n+1} = 2v_M^{n+1} - v_{M-1}^{n+1}$
- $v_{M+1}^{n+1} = v_M^n$
- $v_{M+1}^{n+1} = 2v_M^n - v_{M-1}^n$

where the first two examples are simple extrapolations and the latter two quasi-characteristic extrapolation which are extrapolations derived near the characteristics.
The choice depends not only on the desired accuracy and also on the ad-hoc physical properties some boundary conditions have (such as mass preservation, see Section 1.3) but also on stability.
The stability of boundary conditions not only depend on the boundary condition itself, but also on the finite difference scheme that is being used.
If an unstable boundary condition is used on a stable scheme, then oscillations can still arise, originating at the boundary and propagating through the domain.
As we can see, a fully stable finite difference schemes consists of a stable finite difference scheme on an unbounded domain, and of boundary conditions, which are stable when applied to the stable finite difference scheme.
With the stability analysis given in Section 2.3 we presented the necessary tools to determine stability.
Analyzing the stability of boundary conditions can be difficult and involves complex algebraic manipulations.
A general theory to analyze stability of boundary conditions for finite difference schemes is the Gustafsson-Kreiss-Sundstrom-Osher theory (GKSO theory), developed by Kreiss, Sundstrom and Osher [62, 114, 115] to which we refer here as we will not include it in this work.
Another more general analysis of stability is the so called **Matrix Method**, which considers both the finite difference scheme with the boundary conditions for its stability analysis.
To demonstrate its application, we consider a general explicit finite difference scheme with boundary conditions

$$v^{n+1} = Cv^n + b^n \tag{2.87}$$

with matrix $C \in \mathbb{R}^{(M+1)\times(M+1)}$ and vector $b^n \in \mathbb{R}^{M+1}$.
This can be rewritten in

$$v^{n+1} = C^{n+1}v^0 + \sum_{j=0}^{n+1} C^{n+1-j}b^j. \tag{2.88}$$

If the norms of $\| C^j \|$ are bounded uniformly for $0 \leq n\triangle t \leq T$, we get

$$\triangle x \| v^n \| \leq C_T \left(\triangle x \| v^0 \| + \triangle x \sum_{j=0}^{n+1} \| v_0^j \| \right) \tag{2.89}$$

with $\| v_0^j \|$ the boundary data at $i = 0$.
This inequality is equivalent to the stability definition given in 2.4.2 with the

addition of the boundary value.
Thus the boundedness of matrix C is giving us necessary and sufficient conditions for the stability of (2.87).
Despite it is generality in analyzing stability, the matrix method has two major disadvantages.
Firstly, it is quite difficult in proving the estimates of the matrices C^j, because the size of the matrix increases with decreasing $\triangle x$ but for stability, this estimation must be independent of $\triangle x$.
Secondly, since the matrix method incorporates both the finite difference scheme with the boundary conditions, it is difficult to determine whether an instability originates from an unstable finite difference scheme and/or from unstable boundary conditions.

2.8.1 Mass Preservation

Before we conclude this section about boundary conditions, we want to demonstrate another method to derive boundary conditions which hold specific properties which preserves attributes of the original partial differential equation.
As mentioned in Chapter 1.2, certain initial-boundary value problems possess certain properties such as mass or energy preservation [30].
For the one-dimensional diffusion equation (2.2) on the domain $\Omega := [0, L]$ with no-flux boundary condition, we showed that the total mass of u is conserved in time

$$I(t) := \int_\Omega u(t,x)dx = \int_\Omega u(0,x)dx = const, \tag{2.90}$$

which is equivalent to

$$\begin{aligned} \tfrac{dI(t)}{dt} &= \tfrac{d}{dt}\int_\Omega u(t,x)dx = \int_\Omega \partial_t u(t,x)dx = \int_\Omega D\partial_{xx}u(t,x)dx \\ &= \oint_{\delta\Omega} D\partial_x u(t,x)dx = D\left[\partial_x u(t,L) - \partial_x u(t,0)\right] = 0. \end{aligned} \tag{2.91}$$

It is beneficial to use finite difference schemes that preserve (2.91) in a discrete manner.
However we need to define the basic ideas behind the numerical integration first.
We will refer to [24, 29] for the proper definitions, theorems and notation of so called quadrature formulas and especially of the class of closed Newton-Cotes formulas which are being used for the mass-preserving boundary con-

ditions.

For equidistant nodes $x_i = a + i\triangle x$ with $\triangle x = \frac{b-a}{n}$ we define the so called closed Newton-Cotes formulas to compute $\int_a^b f(x)dx$.

Definition 2.8.1 (closed Newton-Cotes formulas). *A quadrature Q_{n+1} is called a closed Newton-Cotes formula if it is in the form*

$$Q_{n+1}[f] := \triangle x \sum_{i=0}^{n} \alpha_i f(x_i) \tag{2.92}$$

with the weights defined as

$$\alpha_i := \int_0^n \prod_{j=0, j\neq i}^{n} \frac{s-j}{i-j} ds. \tag{2.93}$$

If s is chosen such that $\sigma_i := s\alpha_i$, $i = 0, \ldots, n$ with $\sigma_i \in \mathbb{Z}$, the closed Newton-Cotes formulas can be written as

$$Q_{n+1}[f] = \frac{b-a}{ns} \sum_{i=0}^{n} \sigma_i f(x_i). \tag{2.94}$$

A collection of some closed Newton-Cotes formulas can be found in table 2.4.

n	σ_i	ns	$\lvert R_{n+1}[f]\rvert$	Name
1	1 1	2	$\frac{\triangle x^3}{12} f''(\xi)$	Trapezoidal rule
2	1 4 1	6	$\frac{\triangle x^5}{90} f^4(\xi)$	Simpson rule
3	1 3 3 1	8	$\frac{3\triangle x^5}{80} f^6(\xi)$	8/3 rule
4	7 32 12 32 7	90	$\frac{8\triangle x^7}{945} f^6(\xi)$	Milne rule
5	19 75 50 50 75 19	288	$\frac{275\triangle x^7}{12096} f^6(\xi)$	
6	41 216 27 272 27 216 41	840	$\frac{9\triangle x^9}{1400} f^8(\xi)$	Weddle rule

Table 2.4: Closed Newton-Cotes formulas for different number of nodes n for $\xi \in [a, b]$.

We want to give to examples of closed Newton-Cotes formulas for the interval $[a, b]$ with $a = x_0 < x_1 < \cdots < x_m = b$ and $\triangle x = x_{i+1} - x_i \; \forall i = 0, \ldots, m$.

- **Trapezoidal rule**

$$Q_2^{Tr}[f] := \frac{b-a}{2} [f(a) + f(b)] \tag{2.95}$$

- **Simpson rule**

$$Q_3^{Sim}[f] := \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \tag{2.96}$$

- **Composite Trapezoidal rule**

$$\begin{aligned} Q_{2,m}^{compTr}[f] &:= \sum_{i=0}^{m-1} Q_2^{Tr(i)}[f] \\ &= \frac{\triangle x}{2} \sum_{i=0}^{m-1} \left(f(x_i) + f(x_{i+1})\right) \end{aligned} \tag{2.97}$$

- **Composite Simpson rule**

$$\begin{aligned} Q_{3,m}^{compSim}[f] &:= \sum_{i=0}^{m-1} Q_3^{Sim(i)}[f] \\ &= \frac{\triangle x}{3} \left[f(x_0) + 2 \sum_{i=1}^{\frac{m}{2}-1} f(x_{2i}) + 4 \sum_{i=1}^{\frac{m}{2}} f(x_{2i-1}) + f(x_m) \right]. \end{aligned} \tag{2.98}$$

Mass-preserving boundary conditions

We will now derive mass preserving boundary conditions for the θ-method (2.24) for the diffusion equation (2.2) on the domain $[0, L]$ with no-flux boundary conditions and $x_i = i\triangle x$ for $i = 0, \ldots, N+1$.
We will use as discrete integral the composite trapezoidal rule (2.97) and have

$$\begin{aligned} I_2^{n+1} - I_2^n &= \frac{\triangle x}{2} \sum_{i=0}^{N+1} \left(u_i^{n+1} + u_{i+1}^{n+1} - u_i^n - u_{i+1}^n\right) \\ &= \triangle x \left[\sum_{i=1}^{N} \left[(1-\theta) \frac{D\mu_x}{2} \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \right.\right. \\ &\quad \left. + \theta \frac{D\mu_x}{2} \left(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right) \right] \\ &\quad \left. + \frac{u_0^{n+1} - u_0^n}{2} + \frac{u_{N+1}^{n+1} - u_{N+1}^n}{2} \right] \\ &= \triangle x \left[(1-\theta) \frac{D\mu_x}{2} \left(u_{N+1}^n - u_N^n + u_0^n - u_1^n\right) \right. \\ &\quad + \theta \frac{D\mu_x}{2} \left(u_{N+1}^{n+1} - u_N^{n+1} + u_0^{n+1} - u_1^{n+1}\right) \\ &\quad \left. + \frac{u_0^{n+1} - u_0^n}{2} + \frac{u_{N+1}^{n+1} - u_{N+1}^n}{2} \right]. \end{aligned} \tag{2.99}$$

Rearranging the terms in (2.99) and choosing u_0^{n+1} and u_{N+1}^{n+1} such that the expression vanishes, gives us the following implicit discrete boundary condi-

tions

$$\begin{aligned} u_0^{n+1} &= u_0^n + (1-\theta)D\mu_x\left(u_1^n - u_0^n\right) + \theta D\mu_x\left(u_1^{n+1} - u_0^{n+1}\right), \\ u_{N+1}^{n+1} &= u_{N+1}^n + (1-\theta)D\mu_x\left(u_N^n - u_{N+1}^n\right) + \theta D\mu_x\left(u_N^{n+1} - u_{N+1}^{n+1}\right). \end{aligned} \tag{2.100}$$

If we take a closer look and compare the formula for the boundary values (2.100) with the actual θ-method, we notice their strong resemblance.
This is due to the fact that discretizing the Neumann boundary condition with a central difference

$$\begin{aligned} \partial_x u(t_n, 0) &\approx \tfrac{u_1^n - u_{-1}^n}{2\triangle x}, \\ \partial_x u(t_n, L) &\approx \tfrac{u_{N+2}^n - u_N^n}{2\triangle x} \end{aligned} \tag{2.101}$$

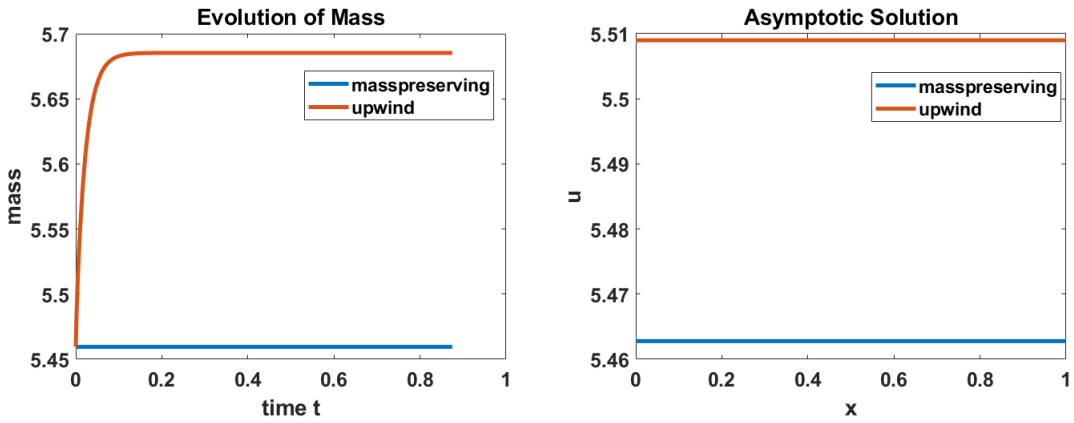
we obtain the ghost value u_{-1}^n (resp. u_{N+2}^n) which can be used for the θ-scheme for $i = 0$ (resp. $i = N+1$) and results in the same boundary conditions. This application was also shown in (2.85).
Further inspection of the coefficients with application of definition of 2.4.2 for monotone schemes, we can confirm that the θ-scheme with trapezoidal mass-preserving boundary conditions (2.100) are monotone.

Proposition 2.8.2. *The θ-scheme (2.24) consistent with diffusion equation (2.2) with the trapezoidal boundary conditions (2.100) constructed with the use of the discrete integral method $I_2^{n+1} - I_2^n = 0$ is mass-preserving. Moreover, the scheme, obtained with the integral method above, is second order in space up to the boundaries since it can be equivalently obtained using the following second-order approximation of the first derivative including a ghost cell (2.101).*
Finally the scheme (2.24) is also monotonicity preserving under the stability condition (2.38).

If instead of the mass-preserving trapezoidal boundary conditions, we used a first order one sided approximation of the homogeneous Neumann boundary condition

$$\begin{aligned} \partial_x u(t_n, 0) &\approx \tfrac{u_1^n - u_0^n}{\triangle x}, \\ \partial_x u(t_n, L) &\approx \tfrac{u_{N+1}^n - u_N^n}{\triangle x} \end{aligned} \tag{2.102}$$

and plug it into the trapezoidal integral method (2.99), we see that $I_2^{n+1} - I_2^n = \mathcal{O}(\triangle x)$.
For some schemes we have stability restrictions that lead to fix
$\lambda_x := \frac{\triangle t}{\triangle x}$. This would mean for the integral method, that the error is $\mathcal{O}(\triangle x) = \mathcal{O}(\triangle t)$ such that for $0 < N\triangle t \leq T$ we obtain the error $\mathcal{O}(1)$ and the mass cannot be preserved.
In Figure 2.1 the evolution of mass over time is shown with the asymptotic solution for both mass-preserving boundary conditions and upwind boundary



(a) Evolution of mass (b) Constant asymptotic solution

Figure 2.1: The evolution of mass and the asymptotic solution for the one-dimensional parabolic equation, computed with the finite difference scheme for both mass-preserving boundary conditions and upwind boundary conditions. The mass is exactly preserved with mass-preserving boundary conditions compared to the other boundary condition.

conditions. The question arises whether it is possible to derive higher order mass-preserving boundary conditions by using an higher order discrete integral method.

We can illustrate the answer by using the same method as for the mass-preserving trapezoidal boundary conditions but instead we will use the composite Simpson rule, applied to the θ-scheme. For the sake of simplicity, we set $\theta = 0$ which corresponds to an explicit scheme

$$u_i^{n+1} = u_i^n + D\mu_x \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right). \tag{2.103}$$

We will use as discrete integral the composite simpson rule (2.98) and have

$$\begin{aligned} I_3^{n+1} - I_3^n &= \tfrac{\triangle x}{3}\Big[\quad 2\sum_{i=1}^{\frac{N-1}{2}} u_{2i}^{n+1} - u_{2i}^n + 4\sum_{i=1}^{\frac{N+1}{2}} u_{2i-1}^{n+1} - u_{2i-1}^n \\ &\qquad +u_0^{n+1} - u_0^n + u_{N+1}^{n+1} - u_{N+1}^n\Big] \\ &= \tfrac{\triangle x}{3}\Big[\quad 2D\mu_x \sum_{i=1}^{\frac{N-1}{2}} \left[u_{2i+1}^n - 2u_{2i}^n + u_{2i-1}^n\right] + 4D\mu_x \sum_{i=1}^{\frac{N+1}{2}} \left[u_{2i}^n - 2u_{2i-1}^n + u_{2i-2}^n\right] \\ &\qquad +u_0^{n+1} - u_0^n + u_{N+1}^{n+1} - u_{N+1}^n\Big] \\ &= \tfrac{\triangle x}{3}\Big[\quad -4D\mu_x \sum_{i=1}^{\frac{N-1}{2}} u_{2i}^n - 4D\mu_x \sum_{i=2}^{\frac{N-1}{2}} u_{2i-1}^n + 4u_0^n - 6u_1^n - 6u_N^n + 4u_{N+1}^n \\ &\qquad +u_0^{n+1} - u_0^n + u_{N+1}^{n+1} - u_{N+1}^n\Big]. \end{aligned} \tag{2.104}$$

As can be seen, the equation does not cancel the summation out, thus making the composite Simpson rule incompatible to find mass-preserving boundary conditions for the θ-scheme. However, we want to remark at this point that with other finite difference schemes, such as multistep schemes, it can be possible to derive mass-preserving boundary conditions which are compatible with the composite Simpson rule.

For the presented methods we assume an equidistant spatial discretization. Of course this method can also be applied on adaptive spatial discretization for Newton-Cotes formulas. Also so called Gauß-Quadratures can be used, which is more accurate compared to Newton-Cotes formulas with the same number of function evaluation.
Since the function evaluations we are using are from finite difference schemes with an equidistant discretization of the domain Ω, we will not indulge in them further and refer to [24, 29], where the numerical integration in higher dimensions can be found as well.

2.9 Asymptotic High Order (AHO) Schemes

This section is dedicated to a special class of finite difference schemes for numerical solutions of dissipative hyperbolic partial differential equations with source term, such as the equations presented in Chapter 1.2 based on [109]. Many difficulties arise when dealing with equations of this kind, such as stiffness of the source term, instability of the solutions and inaccurate approxi-

mation of stationary solutions.
Many attempts were made in the last decades to address these problems like well balanced schemes [56, 59, 75], IMEX-schemes [9] which we briefly presented and discussed in Section 2.7, upwinding the source term [7, 11, 13, 21, 120] and asymptotic preserving schemes [76, 77].
The idea is to use the knowledge of the analytical behaviour of the solution and to plug this information into the schemes to obtain additional properties.
A general type of schemes that has arisen from this are **Asymptotic High order schemes (AHO)** [109] to deal with the inaccuracies that come from the approximation of non constant asymptotic states for large times.
AHO schemes are based on standard finite differences schemes we introduced before, which uses a modified treatment of the source term and consider the behaviour of solutions near non constant stationary states.
These AHO schemes increase their order of accuracy when converging towards the stationary solutions.

We present the AHO schemes on the hyperbolic initial-value-boundary problem (dissipative one-dimensional wave equation with external force) introduced in [109]

$$\begin{cases} \partial_t u(t,x) + \partial_x v(t,x) &= 0, \\ \partial_t v(t,x) + \lambda^2 \partial_x u(t,x) &= \frac{f(t,x)-v}{\tau} \end{cases} \tag{2.105}$$

with initial conditions at $t = 0$

$$\begin{cases} u(0,x) &= u_0(x), \\ v(0,x) &= v_0(x) \end{cases} \tag{2.106}$$

and no-flux Neumann boundary conditions

$$\begin{cases} \frac{\partial u(t,x)}{\partial x} &= 0, \quad \text{for } x \in \delta\Omega, \\ \frac{\partial f(t,x)}{\partial x} &= 0, \quad \text{for } x \in \delta\Omega, \end{cases} \tag{2.107}$$

for $u, f, v : \mathbb{R}^+ \times \Omega \to \mathbb{R}$ and constants $\lambda, \tau, a, b > 0$ on the domain $\Omega := [0, L]$.

We need to remark that the hyperbolic equation (2.105) used in [109] fixes the relaxation time τ in (1.36) with $\tau = 1$.
Equation (2.105) is a general form of the biological model introduced in (1.36), where we chose $f = u\nabla\phi$ with a chemoattractant density ϕ subject to the parabolic diffusion equation

$$\partial_t \phi(t,x) - D\partial_{xx}\phi(t,x) = \alpha u(t,x) - \beta\phi(t,x) \tag{2.108}$$

with source term $\alpha u - \beta\phi$, $\alpha, \beta \geq 0$.
u denotes the density of cells, the function v is their averaged flux, and the

function f the chemotactic-driven movement of the cells.
A standard procedure is to apply an explicit upwind method with the source term being approximated by the Euler scheme.
We have already discussed the analytical behaviour and properties of (2.105) and its stationary solutions in Section 1.2. Using the standard finite difference scheme on (2.105) leads to good approximation of the stationary solution of u, while for the flux v we obtain a rather big error and not $v = 0$ under the no-flux condition.
If we have a look at the scheme

$$u_i^{n+1} = u_i^n + \frac{\triangle t}{2\triangle x}\left(v_{i+1}^n - v_{i-1}^n\right) + \frac{\lambda}{2}\frac{\triangle t}{\triangle x}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right), \tag{2.109}$$

we see that for u near the equilibrium and for large $\partial_{xx}u$, the mesh grid size $\triangle x$ must be chosen very small to make v constant.
For constant asymptotic solutions, this method would lead to a constant v whereas for non-constant asymptotic solutions with large $\partial_{xx}u$, this leads to large errors. Since this problem is not due to stiffness of the source term, implicit methods will not improve this situation.

As we have shown in Section 1.2, we can rewrite (2.105) in a more mathematically tractable way by using its diagonal variables

$$\begin{aligned} w &= \tfrac{1}{2}\left(u - \tfrac{v}{\lambda}\right), \\ z &= \tfrac{1}{2}\left(u + \tfrac{v}{\lambda}\right) \end{aligned} \tag{2.110}$$

and its inversion

$$\begin{aligned} u &= w + z, \\ v &= \lambda(w - z) \end{aligned} \tag{2.111}$$

to obtain the following system of equations

$$\begin{cases} \partial_t w - \lambda\partial_x w &= \frac{1}{2\tau}(z - w) - \frac{1}{2\tau\lambda}f, \\ \partial_t z + \lambda\partial_x z &= \frac{1}{2\tau}(w - z) + \frac{1}{2\tau\lambda}f \end{cases} \tag{2.112}$$

with boundary conditions now being

$$\begin{aligned} w(t,0) &= z(t,0), \\ w(t,L) &= z(t,L), \\ \partial_x f(t,0) &= \partial_x f(t,L) = 0. \end{aligned} \tag{2.113}$$

Before we present the AHO schemes, let us analyse the qualitative properties of the upwind scheme on the hyperbolic-parabolic equation system

$$\begin{cases} \partial_t w - \lambda \partial_x w &= \frac{1}{2\tau}(z-w) - \partial_x \phi \frac{1}{2\tau\lambda}(w+z), \\ \partial_t z + \lambda \partial_x z &= \frac{1}{2\tau}(w-z) + \partial_x \phi \frac{1}{2\tau\lambda}(w+z), \\ \partial_t \phi + D\partial_{xx}\phi &= au - b\phi, \end{cases} \tag{2.114}$$

where we chose $f = u\partial_x\phi$ in (2.112).
Because the hyperbolic equation includes the term $\partial_x\phi$, we need an approximation by using a standard second order central difference scheme for the interior points and a second order forward difference scheme for the boundaries where ϕ satisfy homogenous Neumann boundary conditions, which yields to the discretization vector

$$\Phi_x^n = \frac{1}{2\triangle x} \begin{pmatrix} \frac{4}{3}(\phi_2^n - \phi_1^n) \\ \phi_3^n - \phi_1^n \\ \vdots \\ \phi_M^n - \phi_{M-2}^n \\ \frac{4}{3}\left(\phi_M^n - \phi_{M-1}^n\right) \end{pmatrix} \tag{2.115}$$

with $(\Phi_x^n)_0 = (\Phi_x^n)_{M+1} = 0$. If we choose an explicit upwind method for the hyperbolic part of the equation (2.114) (in regards to the direction of the propagation $\pm\lambda$) and the Crank-Nicolson scheme for the parabolic equation we obtain the following finite difference scheme

$$\begin{aligned} w_i^{n+1} &= w_i + \lambda\tfrac{\triangle t}{\triangle x}\left(w_{i+1}^n - w_i^n\right) - \tfrac{\triangle t}{2\tau\lambda}\left(\Phi_x^n\right)_i (w_i^n + z_i^n) - \tfrac{\triangle t}{2\tau}(w_i^n - z_i^n), \\ z_i^{n+1} &= z_i - \lambda\tfrac{\triangle t}{\triangle x}\left(z_i^n - z_{i-1}^n\right) + \tfrac{\triangle t}{2\tau\lambda}\left(\Phi_x^n\right)_i (w_i^n + z_i^n) + \tfrac{\triangle t}{2\tau}(w_i^n - z_i^n), \\ \phi_i^{n+1} &= \phi_i^n + D\tfrac{\triangle t}{2\triangle x^2}\left(\left(\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n\right) + \left(\phi_{i+1}^{n+1} - 2\phi_i^{n+1} + \phi_{i-1}^{n+1}\right)\right) \\ &\quad - \tfrac{\triangle t}{2}\left(a\left(w_i^n + z_i^n + w_i^{n+1} + z_i^{n+1}\right) - b\left(\phi_i^n + \phi_i^{n+1}\right)\right), \end{aligned} \tag{2.116}$$

which is obviously a consistent finite difference scheme for (2.114) and stable for $\lambda\frac{\triangle t}{\triangle x} \leq 1$ which can be proved with the concepts and methods discussed in Section 2.3 and 2.5 and extended for partial differential equation systems in 2.10.
We recall some of the analytic behaviour of the partial differential equation (2.105) from Chapter 1.2.

- **Mass preservation**

$$\frac{d}{dt}\int_{[0,L]} u(t,x)dt = 0 \tag{2.117}$$

- **Symmetry preservation**

$$\begin{cases} u(0,L-x) &=& u(0,x) \\ v(0,L-x) &=& -v(0,x) \\ \phi(0,L-x) &=& \phi(0,x) \end{cases} \Rightarrow \begin{cases} u(t,L-x) &=& u(t,x) \\ v(t,L-x) &=& -v(t,x), \quad \text{for } t>0 \\ \phi(t,L-x) &=& \phi(t,x) \end{cases} \tag{2.118}$$

- **Stationary solutions** with infinitely many nonconstant and a class of constant stationary solutions (u^*, v^*, ϕ^*).

$$\begin{cases} u^* &=& U, \\ v^* &=& 0, \\ \phi^* &=& \frac{\alpha}{\beta}U, \end{cases} \quad \text{stable for } 0 < U \leq U_1 = \frac{\lambda^2}{\alpha}\left(D\frac{\pi^2}{L^2}+\beta\right), \text{ else wise unstable.} \tag{2.119}$$

A more detailed explanation about the analytical properties can be found in the previous Section 1.2.

In [109] it is shown that the coupled finite difference scheme (2.116) does preserve the properties (2.117), (2.118) and (2.119).
In view of the construction of the AHO schemes, we regard the general linear hyperbolic system

$$\partial_t U + A\partial_x U = BU + \mathcal{F} \tag{2.120}$$

with solution U and define $U^{n+1} = H(U^n, \mathcal{F}^n)$ as a linear numerical scheme consistent with (2.120) and local truncation error T^H.
Furthermore let $\widehat{U}$ be the solution of the generic stationary solution of

$$A\partial_x \widehat{U} = B\widehat{U} + \mathcal{F}. \tag{2.121}$$

With this we can formally define the AHO schemes as follow.

Definition 2.9.1 (AHO-scheme). *Let $U^{n+1} = H(U^n, \mathcal{F}^n)$ be a numerical scheme, consistent with (2.120) with stationary solution $\widehat{U}$. Then we say the scheme is locally AHO of order p, i.e. $(AHO)^p$ if the scheme is of order p on every stationary solution $\widehat{U}$ with $T^H(\widehat{U}) = \mathcal{O}(\triangle x^p)$.*

If we assume $\widehat{U}$ to be a stationary asymptotic state for U, $U - \widehat{U} = \tau(\frac{1}{t})$ as $t \to \infty$ in some appropriate functional space topology, where τ is a modulus of continuity. Then

$$T^H(U) = T^H(U-\widehat{U}) + T^H(U) \approx \tilde{\tau}(\frac{1}{t}) + \mathcal{O}(\triangle x^p) \tag{2.122}$$

for some modulus of continuity $\tilde{\tau}$.

For the sake of an easier application for upwinding, we diagonalize (2.120) with the diagonalizing matrix $R = \begin{pmatrix} 1 & 1 \\ -\lambda & \lambda \end{pmatrix}$ and its inverse $R^{-1} = \frac{1}{2}\begin{pmatrix} 1 & -\frac{1}{\lambda} \\ 1 & \frac{1}{\lambda} \end{pmatrix}$ such that $\omega := \begin{pmatrix} w \\ z \end{pmatrix} = R^{-1}\begin{pmatrix} u \\ v \end{pmatrix}$ to get the diagonalized form of (2.120)

$$\partial_t \omega + \Lambda \partial_x w = B\omega + F \tag{2.123}$$

with $\Lambda = \begin{pmatrix} -\lambda & 0 \\ 0 & \lambda \end{pmatrix}, B = \frac{1}{2\tau}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, F = \frac{1}{2\tau\lambda}\begin{pmatrix} -f \\ f \end{pmatrix}$ and consider a general consistent finite difference scheme of (2.123) of the form

$$\frac{\omega_i^{n+1} - \omega_i^n}{\triangle t} + \frac{\Lambda}{2\triangle x}\left(\omega_{i+1}^n - \omega_i^n\right) - \frac{q}{2\triangle x}\left(\omega_{i+1}^n - 2\omega_i^n + \omega_{i-1}^n\right) = \sum_{l=-1,0,1}\left(B^l \omega_{i+l}^n + D^l F_{i+l}^n\right) \tag{2.124}$$

with matrices $B^l, D^l \in \mathbb{R}^{2\times 2}$ and artificial viscosity $q \in \mathbb{R}_{\geq 0}$.
Or by using the diagonalizing matrix R in the original variables u and v to the equivalent finite difference scheme

$$\begin{aligned} u_i^{n+1} &= u_i^n + \tfrac{\triangle t}{2\triangle x}\left(v_{i+1}^n - v_{i-1}^n\right) + \lambda\tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \\ &\quad + \tfrac{\triangle t}{2}\left(\textstyle\sum_{l=-1,0,1}\left[\beta_{u,u}^l u_{i+l}^n + \frac{1}{\lambda}\beta_{u,v}^l v_{i+l}^n + \frac{1}{\lambda}\gamma_{u,i}^l f_{i+l}\right]\right), \\ v_i^{n+1} &= v_i^n - \lambda^2\tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - u_{i-1}^n\right) + \lambda\tfrac{\triangle t}{2\triangle x}\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) \\ &\quad + \tfrac{\triangle t}{2}\left(\lambda\textstyle\sum_{l=-1,0,1}\left[\beta_{v,u}^l u_{i+l}^n + \beta_{v,v}^l v_{i+l} + \gamma_v^l f_{i+l}^n\right]\right) \end{aligned} \tag{2.125}$$

with

$$RB^lR^{-1} = \frac{1}{2}\begin{pmatrix} \beta_{u,u}^l & \frac{\beta_{u,v}^l}{\lambda} \\ \lambda\beta_{v,u}^l & \beta_{v,v} \end{pmatrix}, \quad RD^lR^{-1} = \frac{1}{2}\begin{pmatrix} \eta_u^l & \frac{\gamma_u^l}{\lambda} \\ \eta_v^l & \gamma_v^l \end{pmatrix}. \tag{2.126}$$

We can recognize (2.124) as a 2-dimensional form of the viscous form (2.30). The matrices B^l and D^l are being used to balance the source term to allow us to increase the order of accuracy for large times near steady state solutions. Of course these matrices must follow certain conditions in order for the scheme (2.124) to be consistent and monotone as well.
For consistency, we require the condition

$$\sum_{l=-1,0,1} B^l = B, \sum_{l=-1,0,1} D^l = I_{2,2} \tag{2.127}$$

and $I_{2,2} \in \mathbb{R}^{2\times 2}$ being the unity matrix.
For monotonicity [6] we require the two conditions

$$B^l_{i,j} \geq 0, \qquad \text{for } l = -1, 0, 1 \text{ and } i \neq j \tag{2.128}$$

and

$$1 - \frac{\triangle t}{\triangle x} q + \triangle t B^0_{i,i} \geq 0, \qquad \mp(-1)^i \frac{\triangle t}{\triangle x} \frac{\lambda}{2} + \frac{\triangle t}{2\triangle x} q + \triangle t B^{\pm 1}_{i,i} \geq 0, \qquad \text{for } i = 1, 2. \tag{2.129}$$

If we use the scheme (2.124) to compute the local truncation error T^H for the stationary solutions of (2.121) and use a Taylor expansion, we obtain with $\partial_t \widehat{w} = 0$, $\Lambda \partial_x \widehat{\omega} = B\widehat{\omega} + F$ and $(\Lambda^{-1}B)^2 = 0$

$$\begin{aligned}
T\widehat{\omega}^n_i &= \tfrac{\lambda}{2\triangle x}\left(\widehat{\omega}^n_{i+1} - \widehat{\omega}^n_{i-1}\right) - \tfrac{q}{2\triangle x}\left(\widehat{\omega}^n_{i+1} - 2\widehat{\omega}^n_i + \widehat{\omega}^n_{i-1}\right) \\
&\quad - \textstyle\sum_{l=-1,0,1} B^l \widehat{\omega}^n_{i+l} - \sum_{l=-1,0,1} D^l F(t_n, x_{i+l}) \\
&= \left(\left(B - \left(B^{-1} + B^0 + B^1\right)\right)\widehat{\omega}^n_i + \left(I_{2,2} - \left(D^{-1} + D^0 + D^1\right)\right) F(t_n, x_i)\right. \\
&\quad - \tfrac{\triangle x}{2}\left(2\left(B^1 - B^{-1}\right)\Lambda^{-1} B \widehat{\omega}_i n + \left(q\Lambda^{-1}B + 2\left(B^1 - B^{-1}\right)\right)\Lambda^{-1} F(t_n, x_i)\right. \\
&\quad \left.\left(q\Lambda^{-1} + 2\left(D^1 - D^{-1}\right)\right)\partial_x F(t_n x_i)\right) \\
&\quad - \tfrac{\triangle x^2}{6}\left(3\left(B^1 + B^{-1}\right)\Lambda^{-1} B \Lambda^{-1} F(t_n, x_i) + \left(\left(B^1 + B^{-1}\right) - B\right)\Lambda^{-1}\partial_x F(t_n, x_i)\right. \\
&\quad \left. + \left(3\left(D^1 + D^{-1}\right) - I_{2,2}\right)\partial_{xx} F(t_n, x_i)\right) + \mathcal{O}(\triangle x^3).
\end{aligned} \tag{2.130}$$

To impose second order accuracy on the stationary solution, we impose the following conditions

$$B^1 - B^{-1} = -\frac{q}{2}\Lambda^{-1}B = \frac{q}{2\lambda}\begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}, D^1 - D^{-1} = -\frac{q}{2}\Lambda^{-1} = \frac{q}{2\lambda}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.131}$$

which, with consistency condition (2.127) translates into the equivalent conditions

$$\begin{aligned}
&\begin{pmatrix} \beta^l_{u,u} & \beta^1_{u,v} \\ \beta^1_{v,u} & \beta^1_{v,v} \end{pmatrix} + \begin{pmatrix} \beta^0_{u,u} & \beta^0_{u,v} \\ \beta^0_{v,u} & \beta^0_{v,v} \end{pmatrix} + \begin{pmatrix} \beta^{-1}_{u,u} & \beta^{-1}_{u,v} \\ \beta^{-1}_{v,u} & \beta^{-1}_{v,v} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & -2 \end{pmatrix}, \\
&\begin{pmatrix} \beta^l_{u,u} & \beta^1_{u,v} \\ \beta^1_{v,u} & \beta^1_{v,v} \end{pmatrix} - \begin{pmatrix} \beta^{-l}_{u,u} & \beta^{-1}_{u,v} \\ \beta^{-1}_{v,u} & \beta^{-1}_{v,v} \end{pmatrix} = \tfrac{q}{2}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \\
&\begin{pmatrix} \gamma^1_u \\ \gamma^1_v \end{pmatrix} + \begin{pmatrix} \gamma^0_u \\ \gamma^0_v \end{pmatrix} + \begin{pmatrix} \gamma^{-1}_u \\ \gamma^{-1}_v \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \\
&\begin{pmatrix} \gamma^1_u \\ \gamma^1_v \end{pmatrix} - \begin{pmatrix} \gamma^{-1}_u \\ \gamma^{-1}_v \end{pmatrix} = \tfrac{q}{\lambda}\begin{pmatrix} -1 \\ 0 \end{pmatrix}.
\end{aligned} \tag{2.132}$$

This leaves us with a certain degree of freedom which can be used to add additional properties such as the conversation of symmetry (2.118) that the

system possesses for symmetric initial data.
For

$$\beta^0_{u,v} = 0, \quad \beta^0_{v,u} = 0, \quad \gamma^0_u = 0 \tag{2.133}$$

we achieve the symmetry preservation.

Proposition 2.9.2. *A scheme in the form (2.125) is consistent for the system (2.120) and of order 2 on stationary solutions, hence a $(AHO)^2$-scheme, if conditions (2.132) are verified. Moreover if it also verifies conditions (2.133), it conserves the symmetry of the solutions for symmetric initial data. Monotonicity, for a given f, holds under conditions (2.128) and (2.129).*

With the conditions in proposition 2.9.2 we can derive an $(AHO)^2$ scheme with the coefficients for the diagonalized scheme (2.124) as

$$\begin{gathered} B^0 = \tfrac{1}{4\tau}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, \quad B^1 = \tfrac{1}{4\tau}\begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix}, \quad B^{-1} = \tfrac{1}{4\tau}\begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \\ D^0 = \tfrac{1}{2\tau}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad D^1 = \tfrac{1}{2\tau}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad D^{-1} = \tfrac{1}{2\tau}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{gathered} \tag{2.134}$$

and we obtain the following $(AHO)^2$ scheme if we choose $q = \lambda$ with stability conditions

$$\begin{aligned} \triangle x &\leq 4\lambda\tau, \\ \triangle t &\leq \tfrac{4\triangle x\tau}{\triangle x + 4\tau\lambda}, \end{aligned} \tag{2.135}$$

which are being derived by satisfying the monotonicity conditions in (2.128) and (2.129)

$$\begin{aligned} w_i^{n+1} &= w_i^n + \lambda\tfrac{\triangle t}{\triangle x}\left(w_{i+1}^n - w_i^n\right) + \tfrac{\triangle t}{4\tau}\left(z_i^n - w_i^n + z_{i+1}^n - w_{i+1}^n\right) - \tfrac{\triangle t}{4\tau\lambda}\left(f_i^n + f_{i+1}^n\right), \\ z_i^{n+1} &= z_i^n - \lambda\tfrac{\triangle t}{\triangle x}\left(z_i^n - z_{i-1}^n\right) + \tfrac{\triangle t}{4\tau}\left(w_i^n - z_i^n + w_{i-1}^n - z_{i-1}^n\right) + \tfrac{\triangle t}{4\tau\lambda}\left(f_i^n + f_{i-1}^n\right) \end{aligned} \tag{2.136}$$

or the equivalent scheme

$$\begin{aligned} u_i^{n+1} &= u_i^n + \lambda\tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) - \left(\tfrac{\triangle t}{2\triangle x} - \tfrac{\triangle t}{4\tau\lambda}\right)\left(v_{i+1}^n - v_{i-1}^n\right) \\ &\quad + \tfrac{\triangle t}{4\tau\lambda}\left(f_{i-1}^n - f_{i+1}^n\right), \\ v_i^{n+1} &= v_i^n - \lambda^2\tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - u_{i-1}^n\right) + \lambda\tfrac{\triangle t}{2\triangle x}\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) \\ &\quad - \tfrac{\triangle t}{4\tau}\left(v_{i+1}^n + 2v_i^n + v_{i-1}^n\right) + \tfrac{\triangle t}{4\tau}\left(f_{i+1}^n + 2f_i^n + f_{i-1}^n\right). \end{aligned} \tag{2.137}$$

With the successfully derived $(AHO)^2$ scheme [109] for the interior points, we still need to compute the appropriate boundary conditions. The boundary conditions for the flux v are already given from the initial-boundary-value

problem (2.107) with $v_0^n = v_{M+1}^n = f_0^n = f_{M+1}^n = 0$ but we still need values for u_0^n and u_{M+1}^n.
For the derivation of the remaining numerical boundary condition, we can use the results presented in Section 2.8.1 where we obtained mass preserving boundary conditions by applying a discrete integral formulation and using the property that the partial differential equation under no-flux boundary condition is mass preserving. Since mass-preservation is a property the partial differential equation (2.117), we can apply this methodology.
As we recall from Section 2.8.1, the quantity

$$\frac{dI(t)}{dt} = \frac{d}{dt}\int_\Omega u(t,x)dx = \int_\Omega \partial_t u(t,x)dx = \int_\Omega -\partial_x v dx = \underbrace{v(t,0) - v(t,L)}_{\text{no flux BC condition}} = 0 \tag{2.138}$$

is satisfied by the hyperbolic partial differential equation (2.105).
So we want an equivalent property to be fulfilled in a discrete manner, thus using the AHO scheme on

$$I^{n+1} = I^n, \tag{2.139}$$

for all $n \geq 0$.
If we use the composite trapezoidal rule (2.97) we obtain

$$I^{n+1} - I^n \;=\; \triangle x \left(\frac{\left(u_0^{n+1} - u_0^n\right)}{2} + \sum_{i=1}^{M} \left(u_i^{n+1} - u_i^n\right) + \frac{\left(u_{M+1}^{n+1} - u_{M+1}^n\right)}{2} \right) = 0. \tag{2.140}$$

Using the general scheme (2.125) we obtain

$$\begin{aligned} \triangle x \Big(& \tfrac{u_0^{n+1}}{2} - \tfrac{u_0^n}{2} + \tfrac{u_{M+1}^{n+1}}{2} - \tfrac{u_{M+1}^n}{2} \\ & - \tfrac{\triangle t}{2\triangle x}\left(v_{M+1}^n + v_M^n - v_0^n - v_1^n\right) + \lambda \tfrac{\triangle t}{2\triangle x}\left(u_{M+1}^n - u_M^n + u_0^n - u_1^n\right)\Big) \\ & + \tfrac{\triangle t}{2} \sum_{i=1}^{M} \sum_{l=-1,0,1} \left[\beta_{u,u}^l u_{i+l}^n + \beta_{u,v}^l v_{i+l}^n + \gamma_u^l f_{i+l}^n\right] = 0. \end{aligned} \tag{2.141}$$

Applying the conditions (2.132) for the $(AHO)^2$ scheme (2.137) into (2.141) leads to

$$\begin{aligned} \triangle x \Big(& \tfrac{u_0^{n+1}}{2} - \tfrac{u_0^n}{2} + \tfrac{u_{M+1}^{n+1}}{2} - \tfrac{u_{M+1}^n}{2} \\ & - \tfrac{\triangle t}{2\triangle x}\left(v_{M+1}^n + v_M^n - v_0^n - v_1^n\right) + \lambda \tfrac{\triangle t}{2\triangle x}\left(u_{M+1}^n - u_M^n + u_0^n - u_1^n\right)\Big) \\ & + \tfrac{\triangle t}{2}\left[\tfrac{q}{2\tau\lambda}\left(v_{M+1}^n + v_M^n - v_1^n - v_0^n\right) - \tfrac{q}{2\tau\lambda}\left(f_{M+1}^n + f_M^n - f_1^n - f_0^n\right)\right] = 0 \end{aligned} \tag{2.142}$$

and reorganizing such that the expression equals to zero brings us the following mass-preserving boundary condition for the $(AHO)^2$ scheme (2.137)

$$\begin{aligned} u_0^{n+1} &= u_0^n + \lambda\tfrac{\triangle t}{\triangle x}\left(u_1^n - u_0^n\right) - \left(\tfrac{\triangle t}{\triangle x} - \tfrac{\triangle t}{2\tau\lambda}\right)\left(v_0^n + v_1^n\right) - \tfrac{\triangle t}{2\tau\lambda}\left(f_0^n + f_1^n\right), \\ u_{M+1}^{n+1} &= u_{M+1}^n + \lambda\tfrac{\triangle t}{\triangle x}\left(u_M^n - u_{M+1}^n\right) + \left(\tfrac{\triangle t}{\triangle x} - \tfrac{\triangle t}{2\tau\lambda}\right)\left(v_{M+1}^n + v_M^n\right) + \tfrac{\triangle t}{2\tau\lambda}\left(f_{M+1}^n + f_M^n\right) \end{aligned} \tag{2.143}$$

without applying the boundary conditions $v_0^n = v_{M+1}^n = f_0^n = f_{M+1}^n = 0$ yet.
The general form of the boundary conditions (2.143) is more versatile when we deal with multi-domain transmission models where the system is mass-preserving, but the flux at the boundary are not zero, which happens at the interfaces between the domains, which we will use later (see Chapter 3).
It is also interesting to note, that if we use non-mass-preserving boundary conditions, such as the boundary conditions for the upwind scheme (2.116) where the error of the discrete integral (2.139) is $\mathcal{O}(\triangle x)$, then because of the large time duration, the error will increase with time [109]. Mass-preservation is a crucial property for the scheme to possess if we want the non-constant asymptotic state to be approximated well enough.

For an extension to have an $(AHO)^3$ scheme, we need to generalize the consistency conditions (2.127) into

$$\sum_{l=-1,0,1} B^l = B + \triangle x C, \qquad \sum_{l=-1,0,1} D^l = I_{2,2} + \triangle x E, \tag{2.144}$$

for matrices $C, E \in \mathbb{R}^{2\times 2}$. To gain a third order truncation error we impose the following additonal conditions

$$\begin{aligned} &C = EB, \quad D^1 - D^{-1} = -\tfrac{q}{2}\Lambda^{-1}, \quad D^1 + D^{-1} = \tfrac{1}{3}I_{2,2}, \\ &B^1 - B^{-1} = -E\Lambda - \tfrac{q}{2}\Lambda^{-1}B, \quad B^1 + B^{-1} = \tfrac{1}{3}B. \end{aligned} \tag{2.145}$$

Since the $(AHO)^3$ schemes are not being used in this work, we refer to the paper [109] where the scheme with consistent conditions can be found.

Theorem 2.9.3. *For every AHO scheme, which satisfies conditions (2.144), we have that for all x_i, $1 \le i \le M$ and t_n, $n > 0$,*

$$T\omega = \mathcal{O}(\triangle t) + \mathcal{O}(\triangle x). \tag{2.146}$$

For the proof on stability of the boundary conditions and numerical tests of the AHO schemes we refer to [109].

Before the complete the discussion on AHO schemes, we want to extend the AHO schemes for

$$\begin{aligned} \partial_t u(t,x) + \partial_x v(t,x) &= g(t,x), \\ \partial_t v(t,x) + \lambda^2 \partial_x u(t,x) &= \tfrac{1}{\tau}\left(f(t,x) - v(t,x)\right). \end{aligned} \tag{2.147}$$

This is a more general equation by adding an additional source term $g(t,x)$ to the first equation of (2.105). As we already have thoroughly described in Section 1.2, they are based on chemotactic models on hyperbolic heat equation where the source term f accounts for the chemotactic motion in the flux, whereas the source term g is the actual source term for the quantity u. We can also here diagonalize (2.147) into

$$\partial_t \omega + \Lambda \partial_x \omega = B\omega + \tilde{F} \tag{2.148}$$

just as (2.123) but with an additional term $\tilde{F} = F + \frac{1}{2}\begin{pmatrix} g \\ g \end{pmatrix}$.

The extra term needs to be considered.
We rewrite (2.125) with $g_i^n = g(t_n, x_i)$ into

$$\begin{aligned} u_i^{n+1} &= u_i^n + \tfrac{\triangle t}{2\triangle x}\left(v_{i+1}^n - v_{i-1}^n\right) + \lambda \tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \\ &\quad + \tfrac{\triangle t}{2}\left(\textstyle\sum_{l=-1,0,1}\left[\beta_{u,u}^l u_{i+l}^n + \frac{1}{\lambda}\beta_{u,v}^l v_{i+l}^n + \frac{1}{\lambda}\gamma_{u,i}^l f_{i+l}^n + \eta_u^l g_{i+l}^n\right]\right), \\ v_i^{n+1} &= v_i^n - \lambda^2 \tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - u_{i-1}^n\right) + \lambda \tfrac{\triangle t}{2\triangle x}\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) \\ &\quad + \tfrac{\triangle t}{2}\left(\lambda \textstyle\sum_{l=-1,0,1}\left[\beta_{v,u}^l u_{i+l}^n + \beta_{v,v}^l v_{i+l} + \gamma_v^l f_{i+l}^n + \eta_v^l g_{i+l}^n\right]\right). \end{aligned} \tag{2.149}$$

With the AHO^2 conditions (2.134) we can derive the conditions on the coefficients for the source term g

$$\begin{pmatrix} \eta_u^1 \\ \eta_v^1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ -\frac{q}{2} \end{pmatrix}, \quad \begin{pmatrix} \eta_u^0 \\ \eta_v^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} \eta_u^{-1} \\ \eta_v^{-1} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{q}{2} \end{pmatrix} \tag{2.150}$$

and obtain the following AHO^2 scheme with source term g

$$\begin{aligned} u_i^{n+1} &= u_i^n + \lambda \tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) - \left(\tfrac{\triangle t}{2\triangle x} - \tfrac{\triangle t}{4\tau\lambda}\right)\left(v_{i+1}^n - v_{i-1}^n\right) \\ &\quad + \tfrac{\triangle t}{4\tau\lambda}\left(f_{i-1}^n - f_{i+1}^n\right) + \tfrac{\triangle t}{4}\left(g_{i+1}^n + 2g_i^n + g_{i-1}^n\right), \\ v_i^{n+1} &= v_i^n - \lambda^2 \tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - u_{i-1}^n\right) + \lambda \tfrac{\triangle t}{2\triangle x}\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) \\ &\quad - \tfrac{\triangle t}{4\tau}\left(v_{i+1}^n + 2v_i^n + v_{i-1}^n\right) + \tfrac{\triangle t}{4\tau}\left(f_{i+1}^n + 2f_i^n + f_{i-1}^n\right) \\ &\quad - \lambda \tfrac{\triangle t}{4}\left(g_{i+1}^n - g_{i-1}^n\right). \end{aligned} \tag{2.151}$$

2.10 Extension to Higher Dimensions

We have introduced finite difference schemes and the concepts of consistency, stability, convergence and order for scalar partial differential equations in one dimension. Before we proceed with the numerical simulations of our biological model (1.45)-(1.47), we will briefly discuss the extension of those concepts and results to systems of partial differential equations in higher dimensions.
Firstly, we consider partial differential equation such as (2.2), (2.1) and (2.62) where u is not an scalar, but a vector of dimension d, i.e $u : [0, \infty) \times \mathbb{R} \to \mathbb{R}^d$.
The diffusion coefficient $D \in \mathbb{R}_{\geq 0}$ becomes a symmetric matrix $D \in \mathbb{R}^{d \times d}$ and respectively the coefficient $c \in \mathbb{R}$ of the linear convection equation becomes a matrix $C \in \mathbb{R}^{d \times d}$ as well.
We remark here that for the convection equation system to be hyperbolic, the matrix C must be diagonalizable with real eigenvalues and for the symmetric matrix D of the diffusion equation to be parabolic, the real part of its eigenvalues must be positive.

The finite difference schemes we have used for the scalar partial differential equations can be, in general, applied one-to-one to the linear partial differential equation systems. An example for the diffusion equation would be the $FTCS$-scheme

$$V_i^{n+1} \quad = \quad V_i^n + D\tfrac{\triangle t}{\triangle x^2}\left(V_{i+1}^n - 2V_i^n + V_{i-1}^n\right), \tag{2.152}$$

where with the capital $V_i^m \in \mathbb{R}^d$ we want to indicate that it is a vector and not a scalar.
In the same manner for the scalar case, we can prove consistency and the order of accuracy [102, 134].
The only main difference is in the stability analysis.
By applying the Von-Neumann stability analysis, we now do not have an amplification factor g but an amplification matrix G which can be obtained by making the substitution $G^n e^{im\theta}$ for V_m^n.
We obtain the general stability condition, similar to (2.48) such that for each $T > 0$ there exist a constant C_T such that for $0 \leq nt \leq T$ we have

$$\| G^n \| \leq C_T. \tag{2.153}$$

For strict hyperbolic systems, such as the convection equation system

$$\partial_t u + C\partial_x u = 0 \tag{2.154}$$

the matrix $C \in \mathbb{R}^{d \times d}$ is diagonalizable with diagonalization matrix $R \in \mathbb{R}^{d \times d}$ and $D = diag\{\lambda_1, \ldots, \lambda_d\}$ such that the we can rewrite the equation to

$$\partial_t w_i + \lambda_i \partial_x w_i = 0, \tag{2.155}$$

for $w = Ru$ and eigenvalues λ_i, $i = 1, \ldots, d$. With (2.155) the stability only depends on the scalar equations.
For the Lax-Friedrich scheme for the convection equation, we would obtain the stability criteria

$$\frac{\triangle t}{\triangle x} \leq \frac{1}{|\lambda_i|}, \tag{2.156}$$

for $i = 1, \ldots, d$ or in general for the scheme

$$\frac{\triangle t}{\triangle x} \leq \frac{1}{\max\limits_{i=1,\ldots,d} |\lambda_i|}. \tag{2.157}$$

We also notice that the diagonalization to (2.155) is necessary to use the upwind-scheme properly which needs to consider the direction of the characteristic speeds λ_i.
A necessary but not sufficient condition for general schemes is

$$|g_v| \leq 1 + K\triangle t, \tag{2.158}$$

for each eigenvalue g_v of the amplification matrix G.

Another method for stability analysis is the Matrix method, described in Section 2.8, where we presented a way to determine stability on finite difference schemes of partial differential equation systems.

Lastly, we give a brief overview of finite difference schemes for partial differential equations of spatial dimension higher than one.
For the linear convection-diffusion equation (2.62) in two dimensions we would have

$$\begin{cases} \partial_t u(t,x,y) + \mathbf{c} \cdot \nabla \mathbf{u}(\mathbf{t},\mathbf{x},\mathbf{y}) &= D\Delta u, & \text{for } (x,y) \in \mathbb{R} \times \mathbb{R} \text{ and } t \geq 0, \\ u(0,x,y) &= u_0(x,y), & \text{for } (x,y) \in \mathbb{R} \times \mathbb{R}, \end{cases} \tag{2.159}$$

for $\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$, $D > 0$, $c_1, c_2 \in \mathbb{R}$ and initial value $u_0 : \mathbb{R}^2 \to \mathbb{R}$.

For the discretization of the two dimensional domain $\Omega = [a,b] \times [c,d]$ we now have $x_i = i\triangle x$, $y_j = j\triangle y$, $t_n = n\triangle t$ for $i = 0, \ldots, N_x + 1$, $j = 0, \ldots, N_y + 1$ with $\triangle x, \triangle y, \triangle t > 0$ and discretized vector $(v)_{i,j}^n$.
The discretization of the partial differential equation (2.62) is being made in the same way as for one spatial dimension, just as consistency and order of accuracy follows the same procedure.
For stability the same definitions apply and the matrix method can also be

used to derive stability conditions.
For the von-Neumann stability analysis the ansatz

$$g^n e^{im\theta_x + ij\theta_y} = v^n_{m,j}, \quad \text{for } \theta_x = \triangle x\xi_1 \text{ and } \theta_y = \triangle y\xi_2 \tag{2.160}$$

can be used to derive stability condition.
For the numerical approximation of the two-dimensional convection-diffusion equation (2.159) a suitable finite difference scheme would be the previously mentioned one-dimensional Crank-Nicolson-Central-Space scheme

$$\begin{aligned} v^{n+1}_{i,j} \;=\; v^n_{i,j} \;& + D\tfrac{\triangle t}{2}\left[\tfrac{\left(v^n_{i+1,j}-2v^n_{i,j}+v^n_{i-1,j}\right)}{\triangle x^2} + \tfrac{\left(v^n_{i,j+1}-2v^n_{i,j}+v^n_{i,j-1}\right)}{\triangle y^2}\right] \\ & + D\tfrac{\triangle t}{2}\left[\tfrac{\left(v^{n+1}_{i+1,j}-2v^{n+1}_{i,j}+v^{n+1}_{i-1,j}\right)}{\triangle x^2} + \tfrac{\left(v^{n+1}_{i,j+1}-2v^{n+1}_{i,j}+v^{n+1}_{i,j-1}\right)}{\triangle y^2}\right] \\ & - c_1\tfrac{\triangle t}{2\triangle x}\left(v^n_{i+1,j} - v^n_{i-1,j}\right) - c_2\tfrac{\triangle t}{2\triangle y}\left(v^n_{i,j+1} - v^n_{i,j-1}\right). \end{aligned} \tag{2.161}$$

We can derive the stability condition by using the monotonicity condition from definition 2.3.3 to obtain

$$\begin{aligned} 1 - D\triangle t\left(\tfrac{1}{\triangle x^2} + \tfrac{1}{\triangle y^2}\right) &\geq 0, \\ D\triangle t\left(\tfrac{1}{2\triangle x^2} + \tfrac{1}{2\triangle y^2}\right) \mp c_1\tfrac{\triangle t}{\triangle x} \mp c_2\tfrac{\triangle t}{\triangle y} &\geq 0 \end{aligned} \tag{2.162}$$

which leads to the stability conditions

$$\begin{aligned} D\mu_x + D\mu_y &\leq 1, \\ \triangle x\triangle y &\leq D\tfrac{\triangle y^2}{c_1\triangle y + c_2\triangle x} + D\tfrac{\triangle x^2}{c_1\triangle y + c_2\triangle x} \end{aligned} \tag{2.163}$$

with $\mu_x := \frac{\triangle t}{\triangle x^2}$ and $\mu_y := \frac{\triangle t}{\triangle y^2}$. These conditions can get simplified if we use the same mesh grid size for both x and y with $\triangle x = \triangle y$ and obtain

$$\begin{aligned} D\mu &\leq \tfrac{1}{2}, \\ \triangle x &\leq \tfrac{D}{(c_1+c_2)}. \end{aligned} \tag{2.164}$$

As we can see, the stability criteria (2.164) are more restrictive than the ones found in table 2.3 for one dimensional convection-diffusion equation (2.3).

With this we conclude the numerical background of finite difference schemes. We have provided concepts and methodology to approximately solve partial differential equation.
We are now able to apply this knowledge to thoroughly analyze and derive a numerical scheme that provides us with an approximate solution of the OOC-model (1.45)-(1.47).

Chapter 3

Numerical Approximation of the Multi-Domain Transmission Model for the OOC

This chapter is devoted to provide some numerical approximations of the mathematical models of chemotaxis describing the qualitative behaviour of different cell species living in a confined environment which we have presented in Chapter 1.5.

We recall that in Chapter 1.1.1 we have presented the biological model which is inspired by the laboratory experiment of the chemotactic movement and interactions between immune and tumour cells in a microfluidic chip environment, and created a mathematical model (1.45), (1.46), (1.47) for the underlying biological model, based on coupled reaction-diffusion-transport equations, that describes the qualitative behaviour accurately.
The results presented in this chapter have been published in [25].
Here we will present a positivity- and mass-preserving numerical discretization of the mathematical model (1.45) with homogeneous Neumann boundary conditions at the boundaries of the two-dimensional domain for the parabolic reaction diffusion problem and a scheme for the inner nodes of the network connecting the two dimensional chambers with the one dimensional channels.
Whereas we will use the parabolic reaction-diffusion-convection Keller-Segel-like model (1.45) for the two dimensional chambers, we will study two different models for the one dimensional channels: the one dimensional equivalent to the parabolic reaction-diffusion-transport equation (1.46) and the hyperbolic model (1.47) which is based on the a Cattaneo model for chemosensitive movements [54] and compare the different dynamics with different chemotactic functions.

The numerical simulation of both models will be conducted by respecting the geometry of the micro chip of the laboratory experiment [15, 31] and the parameters are chosen such that they are reasonable for the quantities (cells, chemicals etc.). The parameters can be found in table 3.1.
They will differ for the two experiments conducted on the microfluidic chip, which represents the two different settings of the laboratory experiment described in [1, 15, 31] and summarized in Section 1.1.1.
Also we will make a qualitatively comparison among the different chemotactical functions $\widehat{f}$ which gradually increase the complexity of the chemotactic movement of the immune cells.
Finally, we will use the results to create a visual comparison by transforming the density field into a particle field, representing the movements of the cells such that a comparison between the models and the available videos of the real laboratory experiment can be made.
Firstly, we will restate the necessary informations from Section 1.5 for the discretization.

3.1 General Model of the OOC

The domain consists of the two dimensional left chamber $\Omega_l := [0, L_x] \times [0, L_y]$ and right chamber $\Omega_r := [L_x + L, 2L_x + L] \times [0, L_y]$ connected with m one dimensional channels $\Omega_{c_k} := [L_x, L_x + L] \times \{y_k\}$, where $L_x, L_y, L > 0$ are based on the geometry of the chip (see Section 1.5) and $y_k \in [0, L_y]$ indicates the position of the k-th channel.
Also let $\delta C_l^k = \{L_x\} \times [y_k, y_k + \sigma]$ and $\delta C_r^k = \{L_x + L\} \times [y_k, y_k + \sigma]$ be the border of the interface between the two and one dimensional domains and $\sigma \in \mathbb{R}_{>0}$ the width of the interface.
The outer boundary of the domains not interfacing the channels are indicated as $\delta\Omega_l$, and $\delta\Omega_r$.
The underlying partial differential equation system in the two dimensional domain is

$$\begin{cases} \partial_t T &= D_T \Delta T - \lambda_T(\omega) T, \\ \partial_t M &= D_M \Delta M - \operatorname{div}\left(f\left(M, \phi\right)\right), \\ \partial_t \phi &= D_\phi \Delta \phi + \alpha_\phi T - \beta_\phi \phi, \\ \partial_t \omega &= D_\omega \Delta \omega + \alpha_\omega M - \beta_\omega \omega, \end{cases} \tag{3.1}$$

with diffusion coefficients $D_T, D_M, D_\phi, D_\omega > 0$ and growth rate and decay rate of chemicals $\alpha_\phi, \alpha_\omega, \beta_\phi, \beta_\omega > 0$. The density quantities are tumour cell density T, immune cells (macrophages) M, the chemoattractant density ϕ which is produced by the tumour cells and influencing the movements of the immune cells and the cytokine density ω, produced by the immune cells, acting as a chemical killer of tumour cells.
The functions $f := \widehat{f}M$ with $\widehat{f} = \chi\nabla\phi$ and $\lambda_T(\omega)$ representing, respectively, the chemotactic sensitivity of immune cells and the decay rate of cancer cells under the action of immune cells, with cellular drift velocity $k_1 > 0$, receptor dissociation constant $k_2 > 0$, killing efficiency coefficient $k_{\omega 1} > 0$ and $k_{\omega 2} > 0$.
In Chapter 1.6 we have presented a variety of different functions $\hat{f}$, i.e chemotactic convection terms, each representing an hypothesis for the drivers of cell movement in dependence of the chemoattractant, and each increasing in complexity.
The boundary conditions used to simulate the laboratory experiment on the chip are (1.42), (1.44). We recall that this is due to the fact that the available data for this experiment are only available in form of a video crop with open boundaries (see Figure 1.8) but in this numerical simulation we will restrict ourselves to no-flux Neumann boundary conditions at the outer boundaries of Ω_l and Ω_r:

$$\frac{\partial T}{\partial \boldsymbol{n}}|_{(x,y)\in\delta\Omega\setminus\delta C^k} = \frac{\partial M}{\partial \boldsymbol{n}}|_{(x,y)\in\delta\Omega\setminus\delta C^k} = \frac{\partial \phi}{\partial \boldsymbol{n}}|_{(x,y)\in\delta\Omega\setminus\delta C^k} = \frac{\partial \omega}{\partial \boldsymbol{n}}|_{(x,y)\in\delta\Omega\setminus\delta C^k} = 0. \quad (3.2)$$

At the interfaces δC_l^k and δC_r^k we have Kedem-Katchalsky (KK) interface conditions which we have discussed in Section 1.4.
For the one-dimensional channels Ω_{c_k} we have the parabolic equation (1.46), which is the one-dimensional version of (3.1)

$$\begin{cases} \partial_t T_c &= D_T \partial_{xx} T_c - \lambda_T(\omega_c) T_c, \\ \partial_t M_c &= D_M \partial_{xx} M_c - \partial_x f_c \\ \partial_t \phi_c &= D_\phi \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c, \\ \partial_t \omega_c &= D_\omega \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_\omega \omega_c, \end{cases} \quad (3.3)$$

with appropriate initial conditions.
Because both boundaries of the one-dimensional channels are interface boundary conditions, we require KK-interface boundary conditions compatible with the ones used for the two-dimensional problem.

For the hyperbolic model (1.47) we have

$$\begin{cases} \partial_t T_c + \partial_x v_c^T = -\lambda_T(\omega) T_c, \\ \partial_t v_c^T + \frac{D_T}{\tau_T} \partial_x T_c = -\frac{v_c^T}{\tau_T}, \\ \partial_t \omega_c = D_{\omega_c} \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_c \omega_c, \\ \partial_t M_c + \partial_x v_c^M = 0, \\ \partial_t v_c^M + \frac{D_{M_c}}{\tau_M} \partial_x M_c = \frac{f_c - v_c^M}{\tau_M}, \\ \partial_t \phi_c = D_{\phi_c} \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c, \end{cases} \tag{3.4}$$

where $f_c = \widehat{f_c} M_c$ and with v_c^T and v_c^M, respectively, which are the average flux of the tumour cells T_c and immune cells M_c in the channels with relaxation time $\tau_T, \tau_M \in \mathbb{R}_{\geq 0}$.
Also here KK-interface boundary conditions (1.42), (1.44) are required for the flux v.

3.2 Generalized Model of OOC

For the sake of simplicity, we omit the super/subscript l, c, r for left, central and right respectively, and only use them when necessary. We rewrite the two dimensional model (3.1) as a general 2D-doubly-parabolic system with source term g as:

$$\begin{cases} \partial_t u &= D_u \Delta u - \text{div} f + g(x, y, t, u) \\ \partial_t \phi &= D_\phi \Delta \phi + \alpha u - \beta \phi, \end{cases} \tag{3.5}$$

with u as the density of individuals and ϕ the concentration of chemoattractant. From now on, the two components of the drift term $f := \widehat{f} u$ will be indicated as:

$$f(x, y, t) := \begin{pmatrix} f^x(x, y, t) \\ f^y(x, y, t) \end{pmatrix} = \begin{pmatrix} \widehat{f^x}(x, y, t) \\ \widehat{f^y}(x, y, t) \end{pmatrix} u(x, y, t) \in \mathbb{R}^2. \tag{3.6}$$

The initial conditions are indicated as

$$\begin{cases} u(0, x, y) &= u_0(x, y), \quad \text{for } (x, y) \in \Omega, \\ \phi(0, x, y) &= \phi_0(x, y), \quad \text{for } (x, y) \in \Omega \end{cases} \tag{3.7}$$

with no-flux Neumann boundary conditions

$$\begin{cases} \dfrac{\partial u}{\partial \mathbf{n}}(t,x,y)\Big|_{\delta\Omega\setminus\delta C^k} = 0, \\ \\ f(t,x,y)\cdot\mathbf{n}|_{\delta\Omega\setminus\delta C^k} = 0 \end{cases} \tag{3.8}$$

and outer normal $\mathbf{n} \in \mathbb{R}^{\mathbf{2}}$ of $\delta\Omega \setminus \delta C^k$.
For the Kedem-Katchalsky (KK) interface conditions (1.42), (1.44) at the interface boundaries δC_l^k, δC_r^k we do not need to differentiate between using the one dimensional parabolic equation (3.3) and the hyperbolic model (3.4)

- two-dimensional KK-interface conditions for left chamber δC_l^k

$$D_u \partial_x u_l\left(L_x, y, t\right) - f_l^x\left(L_x, y, t\right) = K\left(u_c\left(0,t\right) - u_l(L_x, y, t)\right), \tag{3.9}$$

  with $y \in [y_k, y_k + \sigma]$ and $K \in \mathbb{R}_{\geq 0}$.

- two-dimensonal KK-interface conditions for right chamber δC_r^k

$$D_u \partial_x u_r\left(L_x + L, y, t\right) - f_r^x\left(L_x + L, y, t\right) = K\left(u_r(L_x + L, y, t) - u_c\left(L,t\right)\right), \tag{3.10}$$

  with $y \in [y_k, y_k + \sigma]$ and $K \in \mathbb{R}_{\geq 0}$,

which can also be used for the quantity ϕ.

In the one-dimensional channel we rewrite the one dimensional-doubly-parabolic system (1.46) in the more general form:

$$\begin{cases} \partial_t u_c &= D_{u_c}\partial_{xx} u_c - \partial_x f_c + g_c, \\ \partial_t \phi_c &= D_\phi \partial_{xx}\phi_c + \alpha_\phi u_c - \beta_\phi \phi_c \end{cases} \tag{3.11}$$

and the one dimensional hyperbolic-parabolic system (1.47) rewrites as:

$$\begin{cases} \partial_t u_c + \partial_x v_c &= g(t,x), \\ \partial_t v_c + \lambda_c^2 \partial_x u_c &= \frac{1}{\tau}\left(-v_c + f_c\right), \\ \partial_t \phi_c &= D_{\phi_c}\partial_{xx}\phi_c + \alpha_c u_c - \beta_c \phi_c, \end{cases} \tag{3.12}$$

with $f_c = \widehat{f_c} u_c$ and diffusion coefficient $D_u = \lambda_c^2 \tau$ with relaxation time $\tau \in \mathbb{R}_{\geq 0}$.
The initial conditions for both models are

$$\begin{cases} u_c(0,x) &= u_{c_0}(x), \quad \text{for } x \in \Omega_c, \\ \phi_c(0,x) &= \phi_{c_0}(x), \quad \text{for } x \in \Omega_c \end{cases} \tag{3.13}$$

with an additional initial condition for the flux $v_c(0,x) = v_{c_0}(x)$ which we set $v_{c_0}(x) = 0$.
For the interface boundary conditions we need to differentiate between the one-dimensional parabolic (3.11) and the hyperbolic equation (3.12)

- (**parabolic**) one-dimensional KK-interface conditions for $\delta\Omega_{c_k}^l = \{0\}$

$$D_u \partial_x u_c(t,0) - f_c(t,0) = K_k \left(u_c(t,0)\sigma - \int_{y_k}^{y_k+\sigma} u_l(t, L_x, y)dy \right). \tag{3.14}$$

- (**parabolic**) one-dimensional KK-interface conditions for $\delta\Omega_{c_k}^r = \{L\}$

$$D_u \partial_x u_c(t,L) - f_c(t,L) = K_k \left(\int_{y_k}^{y_k+\sigma} u_r(t, L_x + L, y)dy - u_c(t,L)\sigma \right). \tag{3.15}$$

- (**hyperbolic**) one-dimensional KK-interface for $\delta\Omega_{c_k}^l = \{0\}$ and $\delta\Omega_{c_k}^r = \{L\}$

$$\begin{aligned} v_c(t,0) &= K_k \left(\int_{y_k}^{y_k+\sigma} u_l(t, L_x, y)dy - u_c(t,0)\sigma \right), \\ v_c(t,L) &= -K_k \left(\int_{y_k}^{y_k+\sigma} u_r(t, L_x + L, y)dy - u_c(t,L)\sigma \right). \end{aligned} \tag{3.16}$$

An important property of the parabolic equation (3.5) is monotonicity.
On the other hand, for hyperbolic equation (3.12) it is shown in [108] that for linear convection term $f_c = au_c$ and linear source term $g_c = bu_c$, the monotonicity is only given under the condition

$$\left|\frac{a}{\lambda_c}\right| - b \leq 1 \tag{3.17}$$

in order for the quantity u to be non-negative. A negative u would lead to non-physical solutions and flawed dynamics.

3.2.1 Finite Difference Scheme of OOC

We start with the proper discretization of the computational domain.
For the sake of simplicity, we shift the interval of the domains such that $\Omega_l = \Omega_r := [0, L_x] \times [0, L_y]$ and $\Omega_{c_k} := [0, L]$ for $L_x, L_y, L > 0$ and discretize by setting $x_i := i\triangle x$, $y_j := j\triangle y$, $t_n := n\triangle t$ with $i = 0, \ldots, N_x + 1$, $j = 0, \ldots, N_y + 1$, $n = 0, \ldots, M$ such that $x_{N_x+1} = L_x$ and $y_{N_y+1} = L_y$.
For the discretization of the k one-dimensional channels Ω_{c_k} we have $x_i := i\triangle x$ with $i = 0, \ldots, N + 1$ such that $x_{N+1} = L$.

The values of $u_{l_{i,j}}$ and $u_{r_{i,j}}$ at the interfaces at $\delta C_l^k = \{L_x\} \times [y_k, y_k + \sigma_k]$ and $\delta C_r^k = \{0\} \times [y_k, y_k + \sigma_k]$ with channel width σ_k are set at $u_{l_{N_x+1,j}}$, $u_{r_{0,j}}$ with $j = j_{a_k}, \ldots, j_{b_k}$, $y_k = j_{a_k} \triangle y$ and $\sigma_k := (j_{b_k} - j_{a_k})$.
For the one-dimensional parabolic equation (3.11), a reaction-diffusion-convection equation, a finite difference scheme can be used, which we have already discussed in Section 2.6 We can directly apply the Crank-Nicolson-CS(IMEX) scheme to obtain

$$\begin{aligned} u_i^{n+1} &= u_i^n + D\frac{\triangle t}{\triangle x^2}\left[\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{2} + \frac{u_{i+1}^{n}-2u_i^{n}+u_{i-1}^{n}}{2}\right] - \frac{\triangle t}{2\triangle x}\left(\widehat{f}_{i+1}^n u_{i+1}^n - \widehat{f}_{i-1}^n u_{i-1}^n\right) \\ &\quad + \frac{\triangle t}{2}\left(g_i^{n+1} + g_i^n\right) \end{aligned} \tag{3.18}$$

and stability condition (see table 2.3)

$$\begin{aligned} \frac{\triangle t}{\triangle x^2} &\leq \frac{1}{D}, \\ \triangle x &\leq \frac{2D}{\max\limits_{i,j} |\widehat{f}_{i,j}^n|}. \end{aligned} \tag{3.19}$$

Similarly we can use the Crank-Nicolson-CS (IMEX) scheme (2.161) on (3.5)

$$\begin{aligned} u_{i,j}^{n+1} = u_{i,j}^n &+ D\frac{\triangle t}{2}\left[\frac{\left(u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n\right)}{\triangle x^2} + \frac{\left(u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n\right)}{\triangle y^2}\right] \\ &+ D\frac{\triangle t}{2}\left[\frac{\left(u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}\right)}{\triangle x^2} + \frac{\left(u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}\right)}{\triangle y^2}\right] \\ &- \frac{\triangle t}{2\triangle x}\left(f_{i+1,j}^{x,n} - f_{i-1,j}^{x,n}\right) - \frac{\triangle t}{2\triangle y}\left(f_{i,j+1}^{y,n} - f_{i,j-1}^{y,n}\right) \\ &+ \frac{\triangle t}{2}\left(g_{i,j}^{n+1} + g_{i,j}^n\right) \end{aligned} \tag{3.20}$$

with stability conditions (2.163)

$$\begin{aligned} D\left(\mu_x + \mu_y\right) &\leq 1, \\ \triangle t &\leq D\frac{\triangle y^2}{\max\limits_{i,j} \widehat{f}_{i,j}^{x,n} \triangle y + \max\limits_{i,j} \widehat{f}_{i,j}^{y,n} \triangle x} + D\frac{\triangle x^2}{\max\limits_{i,j} \widehat{f}_{i,j}^{x,n} \triangle y + \max\limits_{i,j} \widehat{f}_{i,j}^{y,n} \triangle x}. \end{aligned} \tag{3.21}$$

Although these stable schemes are applicable, they are insufficient for chemotactic problems since the convection term is always extremely dominant compared to diffusion, hence we are dealing with very large Peclet numbers Pe. This forces us to use very small time step sizes $\triangle t$ in order to satisfy stability conditions. As we have seen in Section 2.6, we can add artificial viscosity in order to ease the stability restriction. We can take advantage of the special form of the flux function $f = u\widehat{f}$ with $\widehat{f}$ independent of u, and instead of adding the artificial viscosity term $\frac{\triangle x}{2}\partial_{xx}\left(u\widehat{f}\right)$, add $\frac{\triangle x}{2}\partial_{xx}\left(u|\widehat{f}|\right)$ and obtain

the modified Crank-Nicolson-CS(IMEX) equation

$$\begin{aligned} u_i^{n+1} &= u_i^n + D\mu_x \left[\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{2} + \frac{u_{i+1}^{n}-2u_i^{n}+u_{i-1}^{n}}{2}\right] - \frac{\lambda_x}{2}\left(u_{i+1}^n \widehat{f}_{i+1}^n - u_{i-1}^n \widehat{f}_{i-1}^n\right) \\ &\quad + \frac{\lambda_x}{2}\left(|\widehat{f}_{i+1}^n| u_{i+1}^n - 2|\widehat{f}_i^n| u_i^n + |\widehat{f}_{i-1}^n| u_{i-1}^n\right) + \frac{\triangle t}{2}\left(g_i^{n+1} + g_i^n\right), \end{aligned} \tag{3.22}$$

for the one-dimensional parabolic equation with stability condition

$$\mu_x \leq \frac{1}{D + \triangle x \max\limits_i |\widehat{f}_i^n|} \tag{3.23}$$

2D-parabolic scheme

For the two-dimensional parabolic equation with $\mu_x := \frac{\triangle t}{\triangle x^2}$, $\mu_y := \frac{\triangle t}{\triangle y^2}$, $\lambda_x := \frac{\triangle t}{\triangle x}$, $\lambda_y := \frac{\triangle t}{\triangle y}$ we have the finite difference scheme

$$\begin{aligned} u_{i,j}^{n+1} &= u_{i,j}^n \quad + D\mu_x \left[\frac{\left(u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}\right)}{2} + \frac{\left(u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}\right)}{2}\right] \\ &\quad + D\mu_y \left[\frac{\left(u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}\right)}{2} + \frac{\left(u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}\right)}{2}\right] \\ &\quad - \frac{\lambda_x}{2}\left(\widehat{f}_{i+1,j}^{x,n} u_{i+1,j}^n - \widehat{f}_{i-1,j}^{x,n} u_{i-1,j}^n\right) - \frac{\lambda_y}{2}\left(\widehat{f}_{i,j+1}^{y,n} u_{i,j+1}^n - \widehat{f}_{i,j-1}^{y,n} u_{i,j-1}\right) \\ &\quad + \frac{\lambda_x}{2}\left(|\widehat{f}_{i+1,j}^{x,n}| u_{i+1,j}^n - 2|\widehat{f}_{i,j}^{x,n}| u_{i,j}^n + |\widehat{f}_{i-1,j}^{x,n}| u_{i-1,j}^n\right) \\ &\quad + \frac{\lambda_y}{2}\left(|\widehat{f}_{i,j+1}^{y,n}| u_{i,j+1}^n - 2|\widehat{f}_{i,j}^{y,n}| u_{i,j}^n + |\widehat{f}_{i,j-1}^{y,n}| u_{i,j-1}^n\right) \\ &\quad + \frac{\triangle t}{2}\left(g(t_{n+1}, x_i, y_j) + g(t_n, x_i, y_j)\right) \end{aligned} \tag{3.24}$$

with stability condition, derived similarly to equation (2.163) by the monotonicity condition (2.32)

$$\triangle t \leq \frac{1}{\frac{D}{\triangle x^2} + \frac{D}{\triangle y^2} + \left(\frac{\max\limits_{i,j} |\widehat{f}_{i,j}^{x,n}|}{\triangle x} + \frac{\max\limits_{i,j} |\widehat{f}_{i,j}^{y,n}|}{\triangle y}\right)}. \tag{3.25}$$

In the upcoming numerical simulations, we will use the schemes (3.24) and (3.22) for the parabolic equations of the model (3.1) and (3.3).

1D-hyperbolic scheme

For the hyperbolic model (3.4) we will use the $(AHO)^2$ scheme (2.151) on the general hyperbolic equation (3.12) with arbitrary relaxation time $\tau \in \mathbb{R}_{\geq 0}$, which is quite suitable for the numerical simulation since the time scope of

the biological models are quite long compared to simulations in fluid dynamics and we can benefit from its asymptotic higher order. This means

$$\begin{aligned} u_i^{n+1} &= u_i^n + \lambda_c \tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) - \left(\tfrac{\triangle t}{2\triangle x} - \tfrac{\triangle t}{4\lambda_c \tau}\right)\left(v_{i+1}^n - v_{i-1}^n\right) \\ &\quad + \tfrac{\triangle t}{4\lambda_c\tau}\left(f_{i-1}^n - f_{i+1}^n\right) + \tfrac{\triangle t}{4}\left(g_{i+1}^n + 2g_i^n + g_{i-1}^n\right), \\ v_i^{n+1} &= v_i^n - \lambda_c^2 \tfrac{\triangle t}{2\triangle x}\left(u_{i+1}^n - u_{i-1}^n\right) + \lambda_c \tfrac{\triangle t}{2\triangle x}\left(v_{i+1}^n - 2v_i^n + v_{i-1}^n\right) \\ &\quad - \tfrac{\triangle t}{4\tau}\left(v_{i+1}^n + 2v_i^n + v_{i-1}^n\right) + \tfrac{\triangle t}{4\tau}\left(f_{i+1}^n + 2f_i^n + f_{i-1}^n\right) \\ &\quad - \lambda_c \tfrac{\triangle t}{4}\left(g_{i+1}^n - g_{i-1}^n\right) \end{aligned} \tag{3.26}$$

with stability conditions (2.135)

$$\begin{aligned} \triangle x &\leq 4\lambda_c \tau, \\ \triangle t &\leq \tfrac{4\triangle x \tau}{\triangle x + 4\lambda_c \tau}. \end{aligned} \tag{3.27}$$

An important additional condition is the analytical monotonicity condition (3.17) which needs to be fulfilled to guarantee positive solutions for all times. The monotonicity condition with functions $\widehat{f_c}$ and $\widehat{g_c}$ such that $f_c = \widehat{f_c} u$ and $g_c = \widehat{g_c} u$ is

$$\left|\frac{\widehat{f_c}}{\lambda_c}\right| - \widehat{g_c} \leq 1 \tag{3.28}$$

As we will see later, the monotonicity condition is quite restrictive for the choice of the chemotactic functions $\widehat{f_c}$ and source term $\widehat{g_c}$ and must always be considered in the numerical simulation in order to avoid non-physical solutions.

We showed in (1.36) the connection between the hyperbolic (3.4) and parabolic model (3.3) that both models have the same stationary solution and that in the limit of the relaxation time $\tau \to 0$, the parabolic equation can be obtained from the hyperbolic one.
For consistency this requires the model parameter $\lambda_c \in \mathbb{R}_{\geq 0}$ to be $\lambda_c := \sqrt{\frac{D_u}{\tau}}$.

3.2.2 Positivity Mass-Preserving Boundary Conditions

For the non-interface boundary conditions we have no-flux Neumann boundary conditions (3.8). As we have already worked out, the parabolic equations (3.1) and (3.3) with source term $g = 0$ do have the mass preservation property (1.39), (1.40) and in Proposition 2.8.2 we can derive mass preserving boundary conditions, hence for $\delta\Omega \setminus \delta C^k$ we have the mass-preserving boundary

conditions for the left and right boundaries

$$
\begin{aligned}
u_{0,j}^{n+1} \quad = \quad u_{0,j}^{n} \quad & +D\mu_x\left[\left(u_{1,j}^{n}-u_{0,j}^{n}\right)+\left(u_{1,j}^{n+1}-u_{0,j}^{n+1}\right)\right] \\
& +D\mu_y\left[\frac{\left(u_{0,j+1}^{n}-2u_{0,j}^{n}+u_{0,j-1}^{n}\right)}{2}+\frac{\left(u_{0,j+1}^{n+1}-2u_{0,j}^{n+1}+u_{0,j-1}^{n+1}\right)}{2}\right] \\
& -\lambda_x\left(\widehat{f}_{0,j}^{x,n}u_{0,j}^{n}+\widehat{f}_{1,j}^{x,n}u_{1,j}^{n}\right)-\frac{\lambda_y}{2}\left(\widehat{f}_{0,j+1}^{y,n}u_{0,j+1}^{n}-\widehat{f}_{0,j-1}^{y,n}u_{0,j-1}\right) \\
& +\lambda_x\left(|\widehat{f}_{1,j}^{x,n}|u_{1,j}^{n}-|\widehat{f}_{0,j}^{x,n}|u_{0,j}^{n}\right) \\
& +\frac{\lambda_y}{2}\left(|\widehat{f}_{0,j+1}^{y,n}|u_{0,j+1}^{n}-2|\widehat{f}_{0,j}^{y,n}|u_{0,j}^{n}+|\widehat{f}_{0,j-1}^{y,n}|u_{0,j-1}^{n}\right) \\
& +\frac{\triangle t}{2}\left(g_{0,j}^{n+1}+g_{0,j}^{n+1}\right), \\
u_{N_x+1,j}^{n+1} \quad = \quad u_{N_x+1,j}^{n} \quad & +D\mu_x\left[\left(u_{N_x,j}^{n}-u_{N_x+1,j}^{n}\right)+\left(u_{N_x,j}^{n+1}-u_{N_x+1,j}^{n+1}\right)\right] \\
& +D\mu_y\left[\frac{\left(u_{N_x+1,j+1}^{n}-2u_{N_x+1,j}^{n}+u_{N_x+1,j-1}^{n}\right)}{2}+\frac{\left(u_{N_x+1,j+1}^{n+1}-2u_{N_x+1,j}^{n+1}+u_{N_x+1,j-1}^{n+1}\right)}{2}\right] \\
& +\lambda_x\left(\widehat{f}_{N_x,j}^{x,n}u_{N_x,j}^{n}+\widehat{f}_{N_x+1,j}^{x,n}u_{N_x+1,j}^{n}\right) \\
& -\frac{\lambda_y}{2}\left(\widehat{f}_{N_x+1,j+1}^{y,n}u_{N_x+1,j+1}^{n}-\widehat{f}_{N_x+1,j-1}^{y,n}u_{N_x+1,j-1}\right) \\
& +\lambda_x\left(|\widehat{f}_{N_x,j}^{x,n}|u_{N_x,j}^{n}-|\widehat{f}_{N_x+1,j}^{x,n}|u_{N_x+1,j}^{n}\right) \\
& +\frac{\lambda_y}{2}\left(|\widehat{f}_{N_x+1,j+1}^{y,n}|u_{N_x+1,j+1}^{n}-2|\widehat{f}_{N_x+1,j}^{y,n}|u_{N_x+1,j}^{n}+|\widehat{f}_{N_x+1,j-1}^{y,n}|u_{N_x+1,j-1}^{n}\right) \\
& +\frac{\triangle t}{2}\left(g_{N_x+1,j}^{n+1}+g_{N_x+1,j}^{n}\right),
\end{aligned}
\tag{3.29}
$$

for $[0, y_j] \in \delta\Omega \setminus \delta C^k$ and $[L_x, y_j] \in \delta\Omega \setminus \delta C^k$ and for the top and bottom boundaries

$$
\begin{aligned}
u_{i,0}^{n+1} \quad = \quad u_{i,0}^{n} \quad & +D\mu_x \left[\frac{\left(u_{i+1,0}^{n} - 2u_{i,0}^{n} + u_{i-1,0}^{n}\right)}{2} + \frac{\left(u_{i+1,0}^{n+1} - 2u_{i,0}^{n+1} + u_{i-1,0}^{n+1}\right)}{2} \right] \\
& +D\mu_y \left[\left(u_{i,1}^{n} - u_{i,0}^{n}\right) + \left(u_{i,1}^{n+1} - u_{i,0}^{n+1}\right) \right] \\
& -\tfrac{\lambda_x}{2} \left(\widehat{f}_{i+1,0}^{x,n} u_{i+1,0}^{n} - \widehat{f}_{i-1,0}^{x,n} u_{i-1,0}^{n} \right) - \lambda_y \left(\widehat{f}_{i,1}^{y,n} u_{i,1}^{n} + \widehat{f}_{i,0}^{y,n} u_{i,0} \right) \\
& +\tfrac{\lambda_x}{2} \left(|\widehat{f}_{i+1,0}^{x,n}| u_{i+1,0}^{n} - 2|\widehat{f}_{i,0}^{x,n}| u_{i,0}^{n} + |\widehat{f}_{i-1,0}^{x,n}| u_{i-1,0}^{n} \right) \\
& +\lambda_y \left(|\widehat{f}_{i,1}^{y,n}| u_{i,1}^{n} - |\widehat{f}_{i,0}^{y,n}| u_{i,0}^{n} \right) \\
& +\tfrac{\triangle t}{2} \left(g_{i,0}^{n+1} + g_{i,j}^{n} \right), \\
u_{i,N_y+1}^{n+1} \quad = \quad u_{i,N_y+1}^{n} \quad & +D\mu_x \left[\frac{\left(u_{i+1,N_y+1}^{n} - 2u_{i,N_y+1}^{n} + u_{i-1,N_y+1}^{n}\right)}{2} + \frac{\left(u_{i+1,N_y+1}^{n+1} - 2u_{i,N_y+1}^{n+1} + u_{i-1,N_y+1}^{n+1}\right)}{2} \right] \\
& +D\mu_y \left[\left(u_{i,N_y}^{n} - u_{i,N_y+1}^{n}\right) + \left(u_{i,N_y}^{n+1} - u_{i,N_y+1}^{n+1}\right) \right] \\
& -\tfrac{\lambda_x}{2} \left(\widehat{f}_{i+1,N_y+1}^{x,n} u_{i+1,N_y+1}^{n} - \widehat{f}_{i-1,N_y+1}^{x,n} u_{i-1,N_y+1}^{n} \right) \\
& +\lambda_y \left(\widehat{f}_{i,N_y}^{y,n} u_{i,N_y}^{n} + \widehat{f}_{i,N_y+1}^{y,n} u_{i,N_y+1} \right) \\
& +\tfrac{\lambda_x}{2} \left(|\widehat{f}_{i+1,N_y+1}^{x,n}| u_{i+1,N_y+1}^{n} - 2|\widehat{f}_{i,N_y+1}^{x,n}| u_{i,N_y+1}^{n} + |\widehat{f}_{i-1,N_y+1}^{x,n}| u_{i-1,N_y+1}^{n} \right) \\
& +\lambda_y \left(|\widehat{f}_{i,N_y}^{y,n}| u_{i,N_y}^{n} - |\widehat{f}_{i,N_y+1}^{y,n}| u_{i,N_y+1}^{n} \right) \\
& +\tfrac{\triangle t}{2} \left(g_{i,N_y+1}^{n+1} + g_{i,N_y+1}^{n} \right),
\end{aligned}
\tag{3.30}
$$

at $[x_i, 0] \in \delta\Omega \setminus \delta C^k$ and $[x_i, L_y] \in \delta\Omega \setminus \delta C^k$.

For the corners $(0,0),(L_x,0),(0,L_y),(L_x,L_y)$ we have

$$
\begin{aligned}
u_{0,0}^{n+1} &= u_{0,0}^{n} \quad +D\mu_x\left[\left(u_{1,0}^{n}-u_{0,0}^{n}\right)+\left(u_{1,0}^{n+1}-u_{0,0}^{n+1}\right)\right] \\
&\quad +D\mu_y\left[\left(u_{0,1}^{n}-u_{0,0}^{n}\right)+\left(u_{0,1}^{n+1}-u_{0,0}^{n+1}\right)\right] \\
&\quad -\lambda_x\left(\widehat{f}_{1,0}^{x,n}u_{1,0}^{n}+\widehat{f}_{0,0}^{x,n}u_{0,0}^{n}\right)-\lambda_y\left(\widehat{f}_{0,1}^{y,n}u_{0,1}^{n}+\widehat{f}_{0,0}^{y,n}u_{0,0}\right) \\
&\quad +\lambda_x\left(|\widehat{f}_{1,0}^{x,n}|u_{1,0}^{n}-|\widehat{f}_{0,0}^{x,n}|u_{0,0}^{n}\right) \\
&\quad +\lambda_y\left(|\widehat{f}_{0,1}^{y,n}|u_{0,1}^{n}-|\widehat{f}_{0,0}^{y,n}|u_{0,0}^{n}\right) \\
&\quad +\tfrac{\triangle t}{2}\left(g_{0,0}^{n+1}+g_{0,0}^{n}\right), \\
u_{N_x+1,0}^{n+1} &= u_{N_x+1,0}^{n} \quad +D\mu_x\left[\left(u_{N_x,0}^{n}-u_{N_x+1,0}^{n}\right)+\left(u_{N_x,0}^{n+1}-u_{N_x+1,0}^{n+1}\right)\right] \\
&\quad +D\mu_y\left[\left(u_{N_x+1,1}^{n}-u_{N_x+1,0}^{n}\right)+\left(u_{N_x+1,1}^{n+1}-u_{N_x+1,0}^{n+1}\right)\right] \\
&\quad +\lambda_x\left(\widehat{f}_{N_x,0}^{x,n}u_{N_x,0}^{n}+\widehat{f}_{N_x+1,0}^{x,n}u_{N_x+1,0}^{n}\right) \\
&\quad -\lambda_y\left(\widehat{f}_{N_x+1,1}^{y,n}u_{N_x+1,1}^{n}+\widehat{f}_{N_x+1,0}^{y,n}u_{0,0}\right) \\
&\quad +\lambda_x\left(|\widehat{f}_{N_x,0}^{x,n}|u_{N_x,0}^{n}-|\widehat{f}_{N_x+1,0}^{x,n}|u_{N_x+1,0}^{n}\right) \\
&\quad +\lambda_y\left(|\widehat{f}_{N_x+1,1}^{y,n}|u_{N_x+1,1}^{n}-|\widehat{f}_{N_x+1,0}^{y,n}|u_{N_x+1,0}^{n}\right) \\
&\quad +\tfrac{\triangle t}{2}\left(g_{N_x+1,0}^{n+1}+g_{N_x+1,0}^{n}\right), \\
u_{0,N_y+1}^{n+1} &= u_{0,N_y+1}^{n} \quad +D\mu_x\left[\left(u_{1,N_y+1}^{n}-u_{0,N_y+1}^{n}\right)+\left(u_{1,N_y+1}^{n+1}-u_{0,N_y+1}^{n+1}\right)\right] \\
&\quad +D\mu_y\left[\left(u_{0,N_y}^{n}-u_{0,N_y+1}^{n}\right)+\left(u_{0,N_y}^{n+1}-u_{i0,N_y+1}^{n+1}\right)\right] \\
&\quad -\lambda_x\left(\widehat{f}_{1,N_y+1}^{x,n}u_{1,N_y+1}^{n}+\widehat{f}_{0,N_y+1}^{x,n}u_{0,N_y+1}^{n}\right) \\
&\quad +\lambda_y\left(\widehat{f}_{0,N_y}^{y,n}u_{0,N_y}^{n}+\widehat{f}_{0,N_y+1}^{y,n}u_{0,N_y+1}\right) \\
&\quad +\lambda_x\left(|\widehat{f}_{1,N_y+1}^{x,n}|u_{1,N_y+1}^{n}-2|\widehat{f}_{0,N_y+1}^{x,n}|u_{0,N_y+1}^{n}\right) \\
&\quad +\lambda_y\left(|\widehat{f}_{0,N_y}^{y,n}|u_{0,N_y}^{n}-|\widehat{f}_{0,N_y+1}^{y,n}|u_{0,N_y+1}^{n}\right) \\
&\quad +\tfrac{\triangle t}{2}\left(g_{0,N_y+1}^{n+1}+g_{0,N_y+1}^{n}\right),
\end{aligned}
\tag{3.31}
$$

$$
\begin{aligned}
u^{n+1}_{N_x+1,N_y+1} \;=\; u^{n}_{N_x+1,N_y+1} \;& +D\mu_x\left[\left(u^{n}_{N_x,N_y+1}-u^{n}_{N_x+1,N_y+1}\right)+\left(u^{n+1}_{N_x,N_y+1}-u^{n+1}_{N_x+1,N_y+1}\right)\right]\\
&+D\mu_y\left[\left(u^{n}_{N_x+1,N_y}-u^{n}_{N_x+1,N_y+1}\right)+\left(u^{n+1}_{N_x+1,N_y}-u^{n+1}_{N_x+1,N_y+1}\right)\right]\\
&+\lambda_x\left(\widehat{f}^{x,n}_{N_x,N_y+1}u^{n}_{N_x,N_y+1}+\widehat{f}^{x,n}_{N_x+1,N_y+1}u^{n}_{N_x+1,N_y+1}\right)\\
&+\lambda_y\left(\widehat{f}^{y,n}_{N_x+1,N_y}u^{n}_{N_x+1,N_y}+\widehat{f}^{y,n}_{N_x+1,N_y+1}u_{N_x+1,N_y+1}\right)\\
&+\lambda_x\left(|\widehat{f}^{x,n}_{N_x,N_y+1}|u^{n}_{N_x,N_y+1}-|\widehat{f}^{x,n}_{N_x+1,N_y+1}|u^{n}_{N_x+1,N_y+1}\right)\\
&+\lambda_y\left(|\widehat{f}^{y,n}_{N_x+1,N_y}|u^{n}_{N_x+1,N_y}-|\widehat{f}^{y,n}_{N_x+1,N_y+1}|u^{n}_{N_x+1,N_y+1}\right)\\
&+\tfrac{\triangle t}{2}\left(g^{n+1}_{N_x+1,N_y+1}+g^{n}_{N_x+1,N_y+1}\right).
\end{aligned}
\tag{3.32}
$$

3.2.3 Discretized Transmission Conditions at Interfaces

The only boundary conditions for our models that are missing, are the interface boundary conditions (3.9), (3.10), (3.14), (3.15) for the parabolic models and (3.16) for the hyperbolic model, which all need to be discretized.
In order to keep the equidistant mesh grid sizes, we require for the channel width σ to be a multiple of the meshgrid size $\triangle y$ such that $\sigma := m\triangle y$ for $m \in \mathbb{N}$.
Furthermore, for the interfaces δC^k_l and δC^k_r we set the (i,j) indices such that $u_{N_x+1,j}$ with $j = j_{a_k}, j_{a_k}+1, \ldots, j_{b_k}-1, j_{b_k}$ with the y-coordinate of the k-th channel to be at $y_k = j_{a_k}\triangle y$.

2D-parabolic

For the KK-interface boundary condition for the left chamber (3.9) we need to discretize

$$
D_u\partial_x u_l\left(L_x,y,t\right) - f^x_l\left(L_x,y,t\right) = K\left(u_c\left(0,t\right) - u_l(L_x,y,t)\right). \tag{3.33}
$$

As shown in Proposition 2.8.2, we can obtain mass-preserving boundary conditions by using a central difference to get a ghost value, which included in the finite difference scheme, gives us a mass-preserving boundary conditions, hence with (3.33)

$$
u^n_{N_x+2,j} = u^n_{N_x,j} + 2\frac{\triangle x}{D_u}K\left(u^n_0 - u^n_{N_x+1,j}\right) + \frac{2\triangle x}{D_u}f^{x,n}_{N_x+1,j} \tag{3.34}
$$

and inserted into the two-dimensional parabolic scheme (3.24) gives us the interface boundary condition at δC_l^k

$$\begin{aligned}
u_{N_x+1,j}^{n+1} \;=\; u_{N_x+1,j}^{n} \;& +D\mu_x\left[\left(u_{N_x,j}^{n}-u_{N_x+1,j}^{n}\right)+\left(u_{N_x,j}^{n+1}-u_{N_x+1,j}^{n+1}\right)\right] \\
& +D\mu_y\left[\frac{\left(u_{N_x+1,j+1}^{n}-2u_{N_x+1,j}^{n}+u_{N_x+1,j-1}^{n}\right)}{2}+\frac{\left(u_{N_x+1,j+1}^{n+1}-2u_{N_x+1,j}^{n+1}+u_{N_x+1,j-1}^{n+1}\right)}{2}\right] \\
& +\lambda_x\left(\widehat{f}_{N_x,j}^{x,n}u_{N_x,j}^{n}+\widehat{f}_{N_x+1,j}^{x,n}u_{N_x+1,j}^{n}\right) \\
& -\tfrac{\lambda_y}{2}\left(\widehat{f}_{N_x+1,j+1}^{y,n}u_{N_x+1,j+1}^{n}-\widehat{f}_{N_x+1,j-1}^{y,n}u_{N_x+1,j-1}\right) \\
& +\lambda_x\left(|\widehat{f}_{N_x,j}^{x,n}|u_{N_x,j}^{n}-|\widehat{f}_{N_x+1,j}^{x,n}|u_{N_x+1,j}^{n}\right) \\
& +\tfrac{\lambda_y}{2}\left(|\widehat{f}_{N_x+1,j+1}^{y,n}|u_{N_x+1,j+1}^{n}-2|\widehat{f}_{N_x+1,j}^{y,n}|u_{N_x+1,j}^{n}+|\widehat{f}_{N_x+1,j-1}^{y,n}|u_{N_x+1,j-1}^{n}\right) \\
& +\tfrac{\triangle t}{2}\left(g_{N_x+1,j}^{n+1}+g_{N_x+1,j}^{n}\right) \\
& +\underbrace{\frac{\triangle x}{\triangle t}K_k\left(\left(u_0^n-u_{N_x+1,j}^{n}\right)+\left(u_0^{n+1}-u_{N_x+1,j}^{n+1}\right)\right)}_{\text{KK-transmission term}},
\end{aligned} \tag{3.35}$$

for $j=j_{a_k},\ldots,j_{b_k}$

For the right interface boundary conditions we proceed in the same way.

$$D_u\partial_x u_r\left(L_x+L,y,t\right)-f_r^x\left(L_x+L,y,t\right)=K\left(u_r(L_x+L,y,t)-u_c\left(L,t\right)\right), \tag{3.36}$$

to get

$$u_{-1,j}^{n}=u_1^n-2\frac{\triangle x}{D_u}K\left(u_{0,j}^{n}-u_{N+1}^{n}\right)-\frac{2\triangle x}{D_u}f_{0,j}^{x,n} \tag{3.37}$$

and obtain at the interface boundary on the right chamber δC_r^k

$$\begin{aligned}
u_{0,j}^{n+1} \;=\; u_{0,j}^{n} \;& +D\mu_x\left[\left(u_{1,j}^{n}-u_{0,j}^{n}\right)+\left(u_{1,j}^{n+1}-u_{0,j}^{n+1}\right)\right] \\
& +D\mu_y\left[\frac{\left(u_{0,j+1}^{n}-2u_{0,j}^{n}+u_{0,j-1}^{n}\right)}{2}+\frac{\left(u_{0,j+1}^{n+1}-2u_{0,j}^{n+1}+u_{0,j-1}^{n+1}\right)}{2}\right] \\
& -\lambda_x\left(\widehat{f}_{1,j}^{x,n}u_{1,j}^{n}+\widehat{f}_{1,j}^{x,n}u_{1,j}^{n}\right)-\tfrac{\lambda_y}{2}\left(\widehat{f}_{0,j+1}^{y,n}u_{0,j+1}^{n}-\widehat{f}_{0,j-1}^{y,n}u_{0,j-1}\right) \\
& +\lambda_x\left(|\widehat{f}_{1,j}^{x,n}|u_{1,j}^{n}-|\widehat{f}_{0,j}^{x,n}|u_{0,j}^{n}\right) \\
& +\tfrac{\lambda_y}{2}\left(|\widehat{f}_{0,j+1}^{y,n}|u_{0,j+1}^{n}-2|\widehat{f}_{,j}^{y,n}|u_{i,j}^{n}+|\widehat{f}_{0,j-1}^{y,n}|u_{0,j-1}^{n}\right) \\
& +\tfrac{\triangle t}{2}\left(g_{0,j}^{n+1}+g_{0,j}^{n}\right) \\
& +\underbrace{\frac{\triangle x}{\triangle t}K_k\left(\left(u_{N+1}^{n}-u_{0,j}^{n}\right)+\left(u_{N+1}^{n+1}-u_{0,j}^{n+1}\right)\right)}_{\text{KK-transmission term}},
\end{aligned} \tag{3.38}$$

for $j = j_{a_k}, \ldots, j_{b_k}$.

1D-parabolic

In a very similar manner we can derive the KK-interface boundary conditions (3.14), (3.15) for the finite difference scheme (3.22) for the one-dimensional parabolic equation (3.11)

$$\begin{aligned} D_u \partial_x u_c(t,0) - f_c(t,0) &= K_k \left(u_c(t,0)\sigma - \int_{y_k}^{y_k+\sigma} u_l(t, L_x, y) dy \right), \\ D_u \partial_x u_c(t,L) - f_c(t,L) &= K_k \left(\int_{y_k}^{y_k+\sigma} u_r(t, L_x, y) dy - u_c(t,0)\sigma \right). \end{aligned} \tag{3.39}$$

We must remark that the integral expressions in (3.39) need to be discretized such that

$$\int_{y_k}^{y_k+\sigma} u_l(t_n, L_x, y) dy \approx \triangle y \sum_{j_{a_1}}^{j_{b_1}} u^n_{N_x+1,j}. \tag{3.40}$$

Different quadrature formulas are not compatible with the mass-preserving boundary conditions by using central differences and the discrete integral equation at (3.45) and lead to inconsistency at the interfaces.
We then obtain the ghost values

$$\begin{aligned} u^n_{-1} &= u^n_1 - 2\tfrac{\triangle x}{D} K_k \left(\sigma u^n_0 - \triangle y \sum_{j_{a_1}}^{j_{b_1}} u^n_{l_{N_x+1,j}} \right) - \tfrac{2\triangle x}{D} f^n_0, \\ u^n_{N+2} &= u^n_N + 2\tfrac{\triangle x}{D} K_k \left(\triangle y \sum_{j_{a_1}}^{j_{b_1}} u^n_{r_{N_x+1,j}} - \sigma u^n_{N+1} \right) + \tfrac{2\triangle x}{D} f^n_{N+1} \end{aligned} \tag{3.41}$$

and inserted into the scheme (3.22)

$$\begin{aligned} u^{n+1}_0 &= u^n_0 + D\mu_x \left[u^{n+1}_1 - u^{n+1}_0 + u^n_1 - u^n_0 \right] - \lambda_x \left(f^n_1 + f^n_0 \right) \\ &\quad + \tfrac{\triangle t}{2} \left(g^{n+1}_0 + g^n_0 \right) \\ &\quad + \underbrace{\frac{\triangle x}{\triangle t} K_k \left(\triangle y \sum_{j_{a_1}}^{j_{b_1}} \left(u^n_{l_{N_x+1,j}} + u^{n+1}_{l_{N_x+1,j}} \right) - \sigma \left(u^n_0 + u^{n+1}_0 \right) \right)}_{\text{KK-transmission term}} \\ u^{n+1}_{N+1} &= u^n_{N+1} + D\mu_x \left[u^{n+1}_N - u^n_{N+1} + u^n_N - u^n_{N+1} \right] + \lambda_x \left(f^n_N + f^n_{N+1} \right) \\ &\quad + \tfrac{\triangle t}{2} \left(g^{n+1}_{N+1} + g^n_{N+1} \right), \\ &\quad + \underbrace{\frac{\triangle x}{\triangle x} K_k \left(\triangle y \sum_{j_{a_1}}^{j_{b_1}} \left(u^n_{r_{0,j}} + u^{n+1}_{r_{0,j}} \right) - \sigma \left(u^{n+1}_{N+1} + u^n_{N+1} \right) \right)}_{\text{KK-transmission term}}. \end{aligned} \tag{3.42}$$

We can notice that (3.42) contains integral terms. A natural way is to use a numerical quadrature as presented in Section 2.8.1.
We remark that not any numerical quadrature can be used.
In order to preserve the mass-preserving attribute of the finite difference schemes, we have to consistently use the same numerical quadrature method that has been used to derive the KK-interface boundary conditions (3.34), (3.36) and (3.41). Otherwise we would include an additional integration error, which would grow over time.

1D-hyperbolic

For the finite difference scheme for the hyperbolic model (3.26) we firstly derive the KK-interface boundary conditions for the flux v with (3.16)

$$\begin{aligned} v_0^n &= K_k \left(\triangle y \sum_{j_{a_1}}^{j_{b_1}} u^n_{l_{N_x+1,j}} - u_0^n \sigma \right), \\ v_{N+1}^n &= -K_k \left(\triangle y \sum_{j_{a_1}}^{j_{b_1}} u^n_{r_{0,j}} - u_{N+1}^n \sigma \right). \end{aligned} \tag{3.43}$$

As we can see, also (3.43) contains integral terms.
In the case of the one-dimensional parabolic model, the interface boundary values u_0^{n+1}, u_{N+1}^{n+1} in (3.42) were obtained through the KK-interface boundar conditions (3.41). However, in the hyperbolic case, we cannot obtain the missing interface boundary conditions for u_0^{n+1} and u_{N+1}^{n+1} with just the KK-interface boundary conditions (3.43) which are already used to determine the interface boundary conditions for the flux v_0^n, v_{N+1}^n.
For this reason we derive them by using the discrete integral with the condition

$$I^{n+1} - I^n = 0, \tag{3.44}$$

as we have used in (2.99) for mass-preserving boundary conditions and in the derivation of the AHO schemes boundary conditions.
The difference now is that we do not just require mass preservation in one domain but in $\Omega_l \cup \Omega_r \cup \Omega_c$.
For the numerical integration of higher order, we refer to [29], but in the following we will use the two-dimensional trapezoidal rule which is defined

as

$$\begin{aligned}\int_\Omega F(x,y)\mathrm{dydx} \;\approx\; & \tfrac{\triangle x\triangle y}{4}\Big(F(x_0,y_0)+F(x_{N_x+1},y_0)+F(x_0,y_{N_y+1})\\ & +F(x_{N_x+1},y_{N_y+1})+2\sum_{i=1}^{N_x}\big(F(x_i,y_0)+F(x_i,y_{N_y+1})\big)\\ & +2\sum_{j=1}^{N_y}\left(F(x_0,y_j)+F(x_{N_x+1},y_j)\right)+4\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}F(x_i,y_j)\Big).\end{aligned} \tag{3.45}$$

For the mass-preservation of the whole model in all domains we require

$$I_{\Omega_l}^{n+1}-I_{\Omega_l}^{n}+I_{\Omega_r}^{n+1}-I_{\Omega_r}^{n}+I_{\Omega_c}^{n+1}-I_{\Omega_c}^{n}=0. \tag{3.46}$$

If we apply the trapezoidal rule for all discrete integrals and replace the discrete values $u_{i,j}^{n+1}$ and u_i^{n+1} with the values obtained by the finite difference schemes (3.22), (3.26), (3.29) the remaining term between the left and right chamber Ω_l, Ω_r and one channel Ω_c is

$$\begin{aligned}&\tfrac{\triangle x}{2}\left[u_0^{n+1}-u_0^n+\lambda_c\tfrac{\triangle t}{\triangle x}\left(u_0^n-u_1^n\right)-\left(\tfrac{\triangle t}{\triangle x}-\tfrac{\triangle t}{2\lambda_c\tau}\right)\left(-v_0^n-v_1^n\right)+\tfrac{\triangle t}{2\lambda_c\tau}\left(f_0^n+f_1^n\right)\right]\\ &+\tfrac{\triangle x}{2}\left[u_{N+1}^{n+1}-u_{N+1}^n+\lambda_c\tfrac{\triangle t}{\triangle x}\left(u_{N+1}^n-u_N^n\right)-\left(\tfrac{\triangle t}{\triangle x}-\tfrac{\triangle t}{2\lambda_c\tau}\right)\left(v_{N+1}^n+v_N^n\right)\right.\\ &\left.+\tfrac{\triangle t}{2\lambda_c\tau}\left(-f_{N+1}^n-f_N^n\right)\right]\\ &+\tfrac{\triangle x\triangle y}{4}\left[2\sum_{j=j_{a_1}}^{j_{b_1}}\frac{\triangle t}{\triangle x}K\left(u_0^n-u_{l_{N_x+1,j}}^n+u_0^{n+1}-u_{l_{N_x+1,j}}^{n+1}\right)\right]\\ &+\tfrac{\triangle x\triangle y}{4}\left[2\sum_{j=j_{a_1}}^{j_{b_1}}\frac{\triangle t}{\triangle x}K\left(u_{N+1}^n-u_{r_{0,j}}^n+u_{N+1}^{n+1}-u_{r_{0,j}}^{n+1}\right)\right]=0.\end{aligned} \tag{3.47}$$

Choosing u_0^{n+1} and u_{N+1}^{n+1} such that the term (3.47) vanishes, gives us the following mass-preserving KK-interface boundary conditions for the scheme

(3.26)

$$
\begin{aligned}
u_0^{n+1} &= u_0^n + \lambda_c \tfrac{\triangle t}{\triangle x}\left(u_1^n - u_0^n\right) - \left(\tfrac{\triangle t}{\triangle x} - \tfrac{\triangle t}{2\lambda_c \tau}\right)\left(v_0^n + v_1^n\right) - \tfrac{\triangle t}{2\lambda_c \tau}\left(f_0^n + f_1^n\right) \\
&\quad + \tfrac{\triangle t}{2}\left(g_0^n + g_1^n\right) \\
&\quad - K \tfrac{\triangle t}{\triangle x} \triangle y \sum_{j=j_{a_1}}^{j_{b_1}} \left(u_0^n - u_{l_{N_x+1,j}}^n + u_0^{n+1} - u_{l_{N_x+1,j}}^{n+1}\right), \\
u_{N+1}^{n+1} &= u_{N+1}^n + \lambda_c \tfrac{\triangle t}{\triangle x}\left(u_N^n - u_{N+1}^n\right) + \left(\tfrac{\triangle t}{\triangle x} - \tfrac{\triangle t}{2\lambda_c \tau}\right)\left(v_N^n + v_{N+1}^n\right) + \tfrac{\triangle t}{2\lambda_c \tau}\left(f_N^n + f_{N+1}^n\right) \\
&\quad + \tfrac{\triangle t}{2}\left(g_{N+1}^n + g_N^n\right) \\
&\quad - K \tfrac{\triangle t}{\triangle x} \triangle y \sum_{j=j_{a_1}}^{j_{b_1}} \left(u_{N+1}^n - u_{r_{0,j}}^n + u_{N+1}^{n+1} - u_{r_{0,j}}^{n+1}\right).
\end{aligned}
\tag{3.48}
$$

With this we conclude the discretization and proceed with the stability analysis of the interface boundary conditions.

3.3 Main Results on the Numerical Approximation of the OOC-model

Before we begin with the numerical simulation, we will verify the correctness of the discretized permeability conditions (3.35), (3.38), (3.42).
In our stability analysis of our finite difference schemes, we derived sufficient stability criteria by using the maximum principle to ensure stability. Although such stability conditions are sometimes more restrictive than stability conditions derived through other means (Matrix analysis), they ensure non-oscillatory solutions and more importantly, positivity which is crucial in a biological model where the quantities are densities.
However we have only considered the interior points in our stability analysis and not the stability of the numerical boundary conditions. Often these stability conditions coincide with the stability conditions of the overall finite difference schemes but especially by using KK-interface boundary conditions, a closer look must be taken to ensure stability also at the interface boundaries. This study contributes to new results about stability conditions of KK-interface boundary conditions.

3.3.1 Time Step and Mesh Grid Size Restrictions

1D-parabolic-1D-parabolic

We start by considering two one-dimensional domains $\Omega_l := [0, L_1]$, $\Omega_r := [L_1, L_2]$ for the one-dimensional parabolic equation

$$\partial_t u = D\partial_{xx} u - \partial_x \left(u\widehat{f} \right) \tag{3.49}$$

and use the finite difference scheme (3.22) where the discretized vector u_l belongs to the left domain Ω_l and u_r to the right domain Ω_r.
As such, the finite difference schemes are

$$\begin{aligned}
u_{l_i}^{n+1} &= u_{l_i}^n + D\mu_x \left[\tfrac{u_{l_{i+1}}^{n+1} - 2u_{l_i}^{n+1} + u_{l_{i-1}}^{n+1}}{2} + \tfrac{u_{l_{i+1}}^{n} - 2u_{l_i}^{n} + u_{l_{i-1}}^{n}}{2} \right] - \tfrac{\lambda_x}{2} \left(u_{l_{i+1}}^n \widehat{f}_{l_{i+1}}^n - u_{l_{i-1}}^n \widehat{f}_{l_{i-1}}^n \right) \\
&\quad + \tfrac{\lambda_x}{2} \left(|\widehat{f}_{l_{i+1}}^n| u_{l_{i+1}}^n - 2|\widehat{f}_{l_i}^n| v_i^n + |\widehat{f}_{l_{i-1}}^n| u_{l_{i-1}}^n \right), \\
u_{r_i}^{n+1} &= u_{r_i}^n + D\mu_x \left[\tfrac{u_{r_{i+1}}^{n+1} - 2u_{r_i}^{n+1} + u_{r_{i-1}}^{n+1}}{2} + \tfrac{u_{r_{i+1}}^{n+1} - 2u_{r_i}^{n} + u_{r_{i-1}}^{n}}{2} \right] - \tfrac{\lambda_x}{2} \left(u_{r_{i+1}}^n \widehat{f}_{r_{i+1}}^n - u_{r_{i-1}}^n \widehat{f}_{r_{i-1}}^n \right) \\
&\quad + \tfrac{\lambda_x}{2} \left(|\widehat{f}_{r_{i+1}}^n| u_{r_{i+1}}^n - 2|\widehat{f}_{r_i}^n| u_{r_i}^n + |\widehat{f}_{r_{i-1}}^n| u_{r_{i-1}}^n \right),
\end{aligned} \tag{3.50}$$

with homogeneous Neumann boundary condition at the outer boundaries

$$\begin{aligned}
u_{l_0}^{n+1} &= u_{l_0}^n + D_1\mu_x\left(\left[u_{l_1}^n - u_{l_0}^n + u_{l_1}^{n+1} - u_{l_0}^{n+1}\right]\right) \\
&\quad -\lambda_x\left(v_0^n\widehat{f}_{l_0}^n + u_{l_1}^n\widehat{f}_{l_1}^n\right) + \lambda_x\left(u_{l_1}^n|\widehat{f}_{l_1}^n| - u_{l_0}^n|\widehat{f}_{l_0}^n|\right), \\
u_{r_{N_2+1}}^{n+1} &= u_{r_{N_2+1}}^n + D_2\mu_x\left(\left[u_{r_{N_2}}^n - u_{r_{N_2+1}}^n + u_{r_{N_2}}^{n+1} - u_{r_{N_2+1}}^{n+1}\right]\right) \\
&\quad +\lambda_x\left(u_{r_{N_2+1}}^n\widehat{f}_{r_{N_2+1}}^n + u_{r_{N_2}}^n\widehat{f}_{r_{N_2}}^n\right) + \lambda_x\left(u_{r_{N_2}}^n|\widehat{f}_{r_{N_2}}^n| - u_{r_{N_2+1}}^n|\widehat{f}_{r_{N_2+1}}^n|\right)
\end{aligned} \tag{3.51}$$

and KK-interface boundary conditions at the inner boundary

$$\begin{aligned}
u_{l_{N_1+1}}^{n+1} &= u_{l_{N_1+1}}^n + D_1\mu_x\left(u_{l_{N_1}}^n - u_{l_{N_1+1}}^n + u_{l_{N_1}}^{n+1} - u_{l_{N_1+1}}^{n+1}\right) \\
&\quad +\lambda_x\left(u_{l_{N_1+1}}^n\widehat{f}_{l_{N_1+1}}^n + u_{l_{N_1}}^n\widehat{f}_{l_{N_1}}^n\right) + \lambda_x\left(u_{l_{N_1}}^n|\widehat{f}_{l_{N_1}}^n| - u_{l_{N_1+1}}^n|\widehat{f}_{l_{N_1+1}}^n|\right) \\
&\quad +\tfrac{\triangle t}{\triangle x}K\left(u_{r_0}^n - u_{l_{N_1+1}}^n + u_{r_0}^{n+1} - u_{l_{N_1+1}}^{n+1}\right), \\
u_{r_0}^{n+1} &= u_{r_0}^n + D_2\mu_x\left(u_{r_1}^n - u_{r_0}^n + u_{r_1}^{n+1} - u_{r_0}^{n+1}\right) \\
&\quad -\lambda_x\left(u_{r_0}^n\widehat{f}_{r_0}^n + u_{r_1}^n\widehat{f}_{r_1}^n\right) + \lambda_x\left(u_{r_1}^n|\widehat{f}_{r_1}^n| - u_{r_0}^n|\widehat{f}_{r_0}^n|\right) \\
&\quad +\tfrac{\triangle t}{\triangle x}K\left(u_{l_{N_1+1}}^n - u_{r_0}^n + u_{l_{N_1+1}}^{n+1} - u_{r_0}^{n+1}\right).
\end{aligned} \tag{3.52}$$

We recall the stability condition for this method (3.23) is

$$\triangle t \leq \frac{\triangle x^2}{D + \triangle x \max\limits_i |\widehat{f}_i^n|}. \tag{3.53}$$

Applying the monotonicity conditions (2.32) for the outer boundary conditions leads to

$$\begin{aligned}
1 - D\mu_x - \lambda_x\widehat{f}_{l_0}^n - \lambda_x|\widehat{f}_{l_0}^n| &> 0, \\
D\mu_x - \lambda_x\widehat{f}_{l_1}^n + \lambda_x|\widehat{f}_{l_1}^n| &> 0
\end{aligned} \tag{3.54}$$

and

$$\begin{aligned}
1 - D\mu_x + \lambda_x\widehat{f}_{r_{N_2+1}}^n - \lambda_x|\widehat{f}_{r_{N_2+1}}^n| &> 0, \\
D\mu_x + \lambda_x\widehat{f}_{r_{N_2}}^n + \lambda_x|\widehat{f}_{r_{N_2}}^n| &> 0,
\end{aligned} \tag{3.55}$$

which gives us the stability conditions at the outer boundaries

$$\begin{cases}
\triangle t \leq \frac{\triangle x^2}{D + 2\triangle x\widehat{f}_{l_0}^n}, & \text{for } \widehat{f}_{l_0}^n \geq 0 \\
\triangle t \leq \frac{\triangle x^2}{D} & \text{else}
\end{cases} \tag{3.56}$$

and

$$\begin{cases} \triangle t & \leq & \frac{\triangle x^2}{D-2\triangle x \widehat{f}^n_{r_{N_2+1}}}, & \text{for } \widehat{f}^n_{r_{N_2+1}} \leq 0 \\ \triangle t & \leq & \frac{\triangle x^2}{D} & \text{else.} \end{cases} \tag{3.57}$$

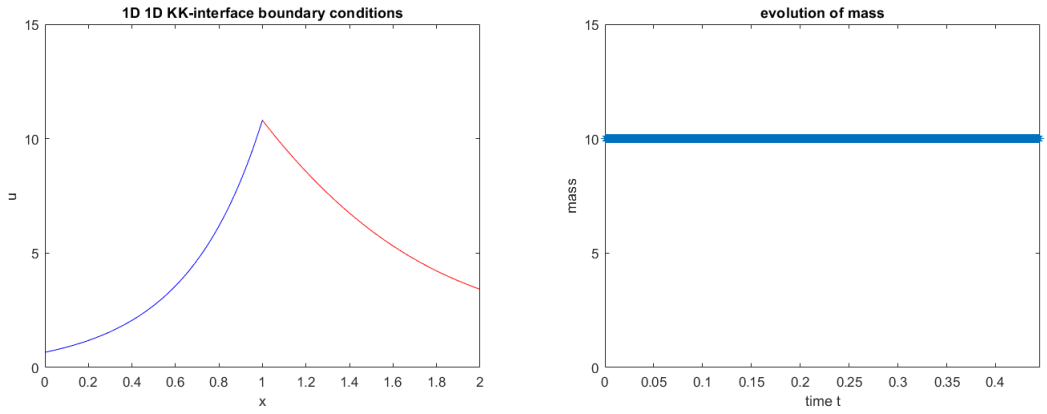
The stability constraints (3.56), (3.57) at the outer boundary are slightly more restrictive compared to the stability constraint of the method at the inner nodes because of the application of the mass-preserving boundary condition. Although we use a central difference scheme for the convection term $\partial_x \left(u\widehat{f}\right)$ at the inner points, the derivation of mass-preservation boundary conditions uses a forward (resp. backward) difference scheme for the convection term. This leads to the slightly more restrictive stability condition.
Now for the KK-interface boundary condition (3.52) we check the monotonicity condition once again.

$$\begin{aligned} 1 - D\mu_x + \lambda_x \widehat{f}^n_{l_{N_1+1}} - \lambda_x |\widehat{f}^n_{l_{N_1+1}}| - \tfrac{\triangle t}{\triangle x} K &\geq 0, \\ D\mu_x + \lambda_x \widehat{f}^n_{l_{N_1}} + \lambda_x |\widehat{f}^n_{l_{N_1}}| &\geq 0 \end{aligned} \tag{3.58}$$

and we obtain the stability condition at the inner boundary

$$\begin{cases} \triangle t & \leq & \frac{\triangle x^2}{D+\triangle x K}, & \text{for } \widehat{f}^n_{l_{N_1+1}} > 0, \\ \triangle t & \leq & \frac{\triangle x^2}{D+2\triangle x \widehat{f}^n_{l_{N_1+1}} + \triangle x K}, & \text{else.} \end{cases} \quad \begin{cases} \triangle t & \leq & \frac{\triangle x^2}{D+\triangle x K}, & \text{for } \widehat{f}^n_{r_0} > 0, \\ \triangle t & \leq & \frac{\triangle x^2}{D-2\triangle x \widehat{f}^n_{r_0} + \triangle x K}, & \text{else.} \end{cases} \tag{3.59}$$

These stability conditions show that the KK-constant $K \in \mathbb{R}_{>0}$ does influence the stability at the interface and must be considered in the choice of an appropriate time step $\triangle t > 0$.
In Figure 3.1 the evolution of mass and steady state solution can be see for the KK-interface boundary conditions connecting two one-dimensional parabolic equations (3.49). To numerically confirm the correctness of the derived stability conditions (3.59) for the KK-interface boundary conditions, we also formulated the finite difference method (3.50) in matrix form $U^{n+1} = MU^n$ and numerically calculated the spectral radius $\rho(M)$ and the maximum norm of matrix M for different time steps $\triangle t$, which can be seen in Figure 3.2 and confirms the stability condition.



(a) Steady state solution (b) Evolution of mass

Figure 3.1: Two parabolic equations (3.49) with $\triangle x = 0.01$, $D = 1$, $\widehat{f}_l = -1$ and $\widehat{f}_r = 3$ and $K = 100$ with KK-interface boundary conditions and time step dt satisfying stability condition (3.59). Mass is preserved at all times. Violation of the stability condition (3.59) and usage of (3.56), (3.57) instead, leads to blow up.

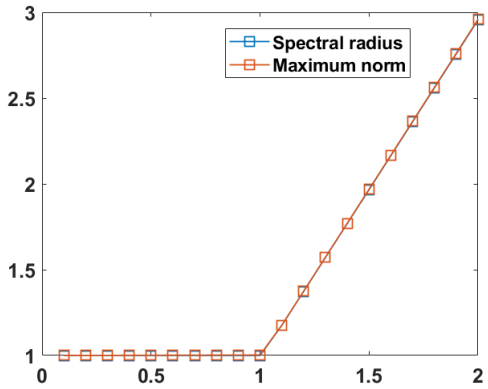


Figure 3.2: Spectral radius $\rho(M)$ and maximums norm of matrix M of the finite difference method (3.50) with $U^{n+1} = MU^n$ is depicted for different time steps $k \cdot \triangle t^{opt}$ with $\triangle t^{opt} = \frac{\triangle x^2}{D - 2\triangle x \max_i \widehat{f}_i^n + \triangle x K}$. This numerically confirms the stability condition (3.59) such that for time step $\triangle t > \triangle t^{opt}$ the stability conditions for the KK-interface boundary conditions is violated and the method is instable.

2D-parabolic-1D-parabolic

If we consider now the interface between a two-dimensional domain $\Omega_l := [0, L_x] \times [0, L_y]$ with an one-dimensional domain $\Omega_r := [0, L]$, for the two-dimensional parabolic equation

$$\partial_t u = D \triangle u - \operatorname{div}\left(u \widehat{f}\right) \tag{3.60}$$

and analyse the monotonicity conditions for the KK-interface boundary conditions

$$
\begin{aligned}
u^{n+1}_{l_{N_x+1,j}} &= u^n_{l_{N_x+1,j}} + D\mu_x\left[\left(u^n_{l_{N_x,j}} - u^n_{l_{N_x+1,j}}\right) + \left(u^{n+1}_{l_{N_x,j}} - u^{n+1}_{l_{N_x+1,j}}\right)\right] \\
&\quad + D\mu_y\left[\frac{\left(u^n_{l_{N_x+1,j+1}} - 2u^n_{l_{N_x+1,j}} + u^n_{l_{N_x+1,j-1}}\right)}{2} + \frac{\left(u^{n+1}_{l_{N_x+1,j+1}} - 2u^{n+1}_{l_{N_x+1,j}} + u^{n+1}_{l_{N_x+1,j-1}}\right)}{2}\right] \\
&\quad + \lambda_x\left(\widehat{f}^{x,n}_{N_x,j}u^n_{l_{N_x,j}} + \widehat{f}^{x,n}_{N_x+1,j}u^n_{l_{N_x+1,j}}\right) \\
&\quad - \tfrac{\lambda_y}{2}\left(\widehat{f}^{y,n}_{N_x+1,j+1}u^n_{l_{N_x+1,j+1}} - \widehat{f}^{y,n}_{N_x+1,j-1}u_{l_{N_x+1,j-1}}\right) \\
&\quad + \lambda_x\left(|\widehat{f}^{x,n}_{N_x,j}|u^n_{l_{N_x,j}} - |\widehat{f}^{x,n}_{N_x+1,j}|u^n_{l_{N_x+1,j}}\right) \\
&\quad + \tfrac{\lambda_y}{2}\left(|\widehat{f}^{y,n}_{N_x+1,j+1}|u^n_{l_{N_x+1,j+1}} - 2|\widehat{f}^{y,n}_{N_x+1,j}|u^n_{l_{N_x+1,j}} + |\widehat{f}^{y,n}_{N_x+1,j-1}|u^n_{l_{N_x+1,j-1}}\right) \\
&\quad + \underbrace{\frac{\triangle t}{\triangle x}K_k\left(\left(u^n_{r_0} - u^n_{l_{N_x+1,j}}\right) + \left(u^{n+1}_{r_0} - u^{n+1}_{l_{N_x+1,j}}\right)\right)}_{\text{KK-transmission term}} \\
u^{n+1}_{r_0} &= u^n_{r_0} + D_2\mu_x\left(u^n_{r_1} - u^n_{r_0} + u^{n+1}_{r_1} - u^{n+1}_{r_0}\right), \\
&\quad + \lambda_x\left(u^n_{r_0}\widehat{f}^n_0 + u^n_{r_1}\widehat{f}^n_1\right) + \lambda_x\left(u_{r_1}|\widehat{f}^n_1| - u_{r_0}|\widehat{f}^n_0|\right) \\
&\quad + \underbrace{\frac{\triangle t}{\triangle x}K_k\left(\triangle y\sum_{j_{a_1}}^{j_{b_1}}\left(u^n_{l_{N_x+1,j}} + u^{n+1}_{l_{N_x+1,j}}\right) - \sigma\left(u^n_{r_0} + u^{n+1}_{r_0}\right)\right)}_{\text{KK-transmission term}},
\end{aligned}
\tag{3.61}
$$

for $j = j_{a_k}, \dots j_{b_k}$. For the interface boundary condition at $u^{n+1}_{l_{N_x+1,j}}$ we see that monotonicity is preserved when

$$
1 - D\mu_x - D\mu_y + \lambda_x\widehat{f}^{x,n}_{N_x+1,j} - \lambda_x|\widehat{f}^{x,n}_{N_x+1,j}| - \lambda_y|\widehat{f}^{y,n}_{N_x+1,j}| - \tfrac{\triangle t}{\triangle x}K \;>\; 0, \tag{3.62}
$$

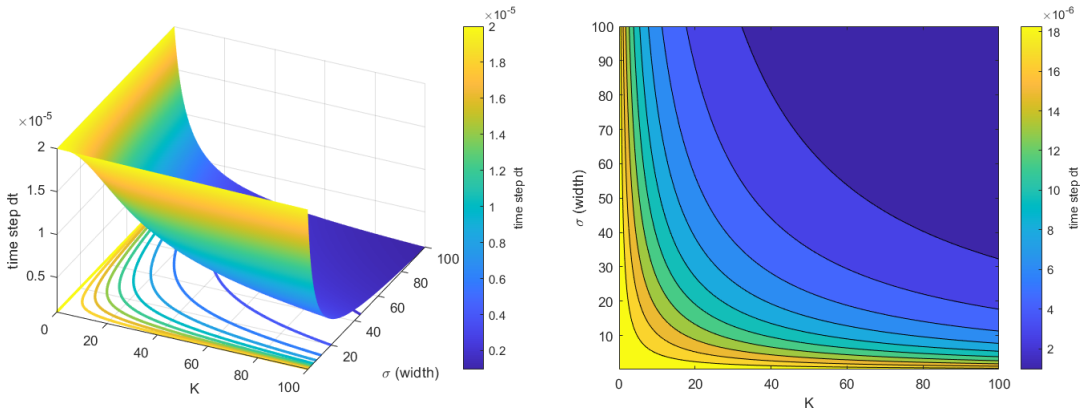
which gives use the stability condition

$$
\begin{cases}
\triangle t \;\leq\; \dfrac{1}{D\left(\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2}\right) + \frac{2|\widehat{f}^{x,n}_{N_x+1,j}|}{\triangle x} + \frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y} + \frac{K}{\triangle x}} & \text{for } \widehat{f}^{x,n}_{N_x+1,j} < 0, \\
\triangle t \;\leq\; \dfrac{1}{D\left(\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2}\right) + \frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y} + \frac{K}{\triangle x}} & \text{else,}
\end{cases}
\tag{3.63}
$$

for the left side of the interface, and for the right side of the interface

$$
\begin{aligned}
&1 - D\mu_x + \lambda_x\widehat{f}^n_0 - \lambda_x|\widehat{f}^n_0| - \tfrac{\triangle t}{\triangle x}K\sigma \\
\Rightarrow\quad &\begin{cases}
\triangle t \;\leq\; \frac{\triangle x^2}{D + \triangle x\sigma K}, & \text{for } \widehat{f}^n_0 > 0, \\
\triangle t \;\leq\; \frac{\triangle x^2}{D + 2\triangle x|\widehat{f}^n_0| + \triangle x\sigma K} & \text{else.}
\end{cases}
\end{aligned}
\tag{3.64}
$$

The stability condition (3.63) is similar to the one-dimensional one (3.59) where the KK-constant K influences the stability, whereas the stability condition (3.64) is not only influenced by the KK-constant K but also by the channel width σ, which must be taken care of accordingly.
In Figure 3.3 a visual representation of the time step restriction (3.64) for varying K and channel width σ is presented. We summarize the results in



(a) (b)

Figure 3.3: Time step restriction (3.64) $\triangle t$ for the one-dimensional parabolic interface boundary condition with $\triangle x = 0.01$, $\triangle y = 0.1$ and $D = 5$ for different K and channel width σ, $\widehat{f} = 0$. As expected the time step $\triangle t$ must be chosen smaller when either K or σ increases.

the following two propositions.

Proposition 3.3.1 (1D-1D Parabolic KK-interface boundary stability conditions)**.** *Let the parabolic convection-diffusion equation (3.49), defined at the one-dimensional domains $\Omega_r = [0, L_1]$, $\Omega_r = [L_1, L_2]$ and homogeneous Neumann boundary conditions at the outer boundaries $\{0\}$ and $\{L_2\}$ be connected at the boundary interface $\{L_1\}$ defined as KK-interface boundary condition with KK-parameter $K \in \mathbb{R}_{>0}$. Furthermore let the finite difference scheme (3.50), consistent to (3.49), be complemented with homogeneous Neumann boundary conditions (3.51) each at the outer boundaries and with the discretized KK-interface boundary conditions at the inner boundary (3.52).*
Then the complete finite difference schemes stability conditions at the inner points (3.49) and outer boundaries (3.56) and (3.57) complemented with the additional stability constraint for the KK-interface boundary conditions are

$$\textbf{\textit{KK-interface boundary condition}} \text{ at } \Omega_l \begin{cases} \triangle t \leq \frac{\triangle x^2}{D + \triangle x K}, & \text{for } \widehat{f}^n_{N_1+1} > 0, \\ \triangle t \leq \frac{\triangle x^2}{D + 2\triangle x \widehat{f}^n_{N_1+1} + \triangle x K}, & \text{else} \end{cases} \tag{3.65}$$

and

$$\textbf{\textit{KK-interface boundary condition}}\ \text{at}\ \Omega_r \begin{cases} \triangle t \ \leq \ \frac{\triangle x^2}{D+\triangle x K}, & \text{for } \widehat{f}_0^n > 0, \\ \triangle t \ \leq \ \frac{\triangle x^2}{D-2\triangle x \widehat{f}_0^n + \triangle x K}, & \text{else.} \end{cases} \tag{3.66}$$

Proposition 3.3.2 (2D-1D Parabolic KK-interface boundary stability conditions)**.** *Let the parabolic convection-diffusion equations,(3.60), defined at the* $\Omega_l = [0, L_x] \times [0, L_y]$*, and (3.60) defined at* $\Omega_r = [L_x, L]$ *with homogeneous Neumann boundary conditions at the outer boundaries* $\delta_{out}\Omega_l = \Omega_l \setminus \delta_{interface}\Omega_l$ *and* $\delta_{out}\Omega_r = \{L\}$ *be connected at the boundary interface* $\delta_{interface}\Omega_l = \{L_x\} \times [C, C+\sigma]$ *and* $\delta_{interface}\Omega_r = \{0\}$ *defined as* KK*-interface boundary condition with KK-parameter* $K \in \mathbb{R}_{>0}$*, and corridor width* $\sigma > 0$ *with* $0 < C < L_y - \sigma$*. Furthermore let the finite difference scheme (3.50), consistent to (3.49), (3.24) consistent to (3.60) be complemented with homongeous Neumann boundary conditions (3.51), (3.29) each at the outer boundaries and with the discretized KK-interface boundary conditions at the inner boundary (3.61).*
Then the complete finite difference schemes stability conditions at the inner points (3.23), (3.25) complemented with the additional stability constraint for the KK*-interface boundary conditions are*

$$\delta_{interface}\Omega_l \begin{cases} \triangle t \ \leq \ \dfrac{1}{D\left(\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2}\right) + \frac{2|\widehat{f}^{x,n}_{N_x+1,j}|}{\triangle x} + \frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y} + \frac{K}{\triangle x}}, & \text{for } \widehat{f}^{x,n}_{N_x+1,j} < 0, \\ \triangle t \ \leq \ \dfrac{1}{D\left(\frac{1}{\triangle x^2}+\frac{1}{\triangle y^2}\right)+\frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y}+\frac{K}{\triangle x}}, & \text{else} \end{cases}$$

and

$$\delta_{interface}\Omega_r \begin{cases} \triangle t \ \leq \ \frac{\triangle x^2}{D+\triangle x \sigma K}, & \text{for } \widehat{f}_0^n > 0, \\ \triangle t \ \leq \ \frac{\triangle x^2}{D+2\triangle x|\widehat{f}_0^n| + \triangle x \sigma K}, & \text{else.} \end{cases} \tag{3.67}$$

2D-parabolic-1D-hyperbolic

Analogously we conduct the derivation of stability conditions for the interface boundary conditions between the one-dimensional domain $\Omega_r = [L_x, L]$, for which the underlying partial differential equation is the hyperbolic equation

$$\begin{cases} \partial_t u_c + \partial_x v_c \ = \ 0, \\ \partial_t v_c + \lambda_c^2 \partial_x u_c \ = \ (f_c - v_c)\frac{1}{\tau} \end{cases} \tag{3.68}$$

and in the left domain $\Omega_l = [0, L_x] \times [0, L_y]$ we have again the parabolic equation.

For the KK-interface boundary conditions we have (3.48), (3.43)

$$\begin{aligned} u_0^{n+1} &= u_0^n + \lambda_c \tfrac{\triangle t}{\triangle x}\left(u_1^n - u_0^n\right) - \left(\tfrac{\triangle t}{\triangle x} - \tfrac{\triangle t}{2\lambda_c \tau}\right)\left(v_0^n + v_1^n\right) - \tfrac{\triangle t}{2\lambda_c \tau}\left(f_0^n + f_1^n\right) \\ &\quad -K \tfrac{\triangle t}{\triangle x} \triangle y \sum_{j=j_{a_1}}^{j_{b_1}} \left(u_0^n - u_{l_{N_x+1,j}}^n + u_0^{n+1} - u_{l_{N_x+1,j}}^{n+1}\right), \\ v_0^n &= K\left(\triangle y \sum_{j_{a_1}}^{j_{b_1}} u_{l_{N_x+1,j}}^{l,n} - u_0^n \sigma\right). \end{aligned} \tag{3.69}$$

In order to use the monotonicity condition, we need to diagonalize the boundary conditions of u_0^{n+1}, v_0^{n+1} into $w_0^{n+1}, z_0^{n+1}, w_{N+1}^{n+1}, z_{N+1}^{n+1}$ with the relation $u = w + z$ and $v = \lambda_c (z - w)$.
Now, we set

$$\begin{aligned} v_0^{n+1} &= \lambda_c \left(z_0^{n+1} - w_0^{n+1}\right) = K\left(\triangle y \sum_{j_{a_1}}^{j_{b_1}} u_{l_{N_x+1,j}}^{n+1} - \sigma\left(z_0^{n+1} + w_0^{n+1}\right)\right) \\ \Leftrightarrow \quad z_0^{n+1} &= \tfrac{\lambda_c - \sigma K}{\lambda_c + \sigma K} w_0^{n+1} + \tfrac{K}{\lambda_c + \sigma K} \triangle y \sum_{j_{a_1}}^{j_{b_1}} u_{l_{N_x+1,j}}^{n+1} \end{aligned} \tag{3.70}$$

and if we set $\rho := \frac{\lambda_c - \sigma K}{\lambda_c + \sigma K}$,$\varsigma^{n+1} := \frac{K}{\lambda_c + \sigma K} \triangle y \sum_{j_{a_1}}^{j_{b_1}} u_{l_{N_x+1,j}}^{n+1}$ we obtain for the inner boundary w_0^{n+1}

$$\begin{aligned} w_0^{n+1} + \rho w_0^{n+1} + \varsigma^{n+1} &= (1+\rho) w_0^n + \varsigma^n + 2\lambda_c \tfrac{\triangle t}{\triangle x}\left(w_1^n - \rho w_0^n + w_0^n - \varsigma^n\right) \\ &\quad + \left(\tfrac{\triangle t}{2\tau} - \lambda_c \tfrac{\triangle t}{\triangle x}\right)\left(\left(\rho - 1\right) w_0^n + \varsigma^n + z_1^n - w_1^n\right) - \tfrac{\triangle t}{2\lambda_c \tau}\left(f_0^n + f_1^n\right) \\ &\quad - \tfrac{\triangle t}{\triangle x} K \left(\sigma\left(1+\rho\right)\left(w_0^n + w_0^{n+1}\right) + \sigma\left(\varsigma^n + \varsigma^{n+1}\right)\right. \\ &\quad \left. - \textstyle\sum_{j=j_{a_k}}^{j_{b_k}} \left(u_{l_{N_x+1,j}}^n + u_{l_{N_x+1,j}}^{n+1}\right)\right) \end{aligned} \tag{3.71}$$

and applying the monotonicity condition leads to the following inequality

$$\begin{aligned} &(1+\rho) - 2\lambda_c \tfrac{\triangle t}{\triangle x}\left(\rho - 1\right) + \left(\tfrac{\triangle t}{2\tau} - \lambda_c \tfrac{\triangle t}{\triangle x}\right)\left(\rho - 1\right) - \tfrac{\triangle t}{\triangle x} K_k \sigma \left(1+\rho\right) - \tfrac{\triangle t}{2\lambda_c \tau} \widehat{f}_0^n > 0 \\ &\Leftrightarrow \triangle t \leq \tfrac{1+\rho}{\frac{2\lambda_c}{\triangle x}(\rho-1) - \left(\frac{\triangle t}{2\tau} - \lambda_c \frac{\triangle t}{\triangle x}\right)(\rho-1) + K_k \frac{\sigma(1+\rho)}{\triangle x} + \frac{\widehat{f}_0^n}{2\lambda_c \tau}}. \end{aligned} \tag{3.72}$$

Proposition 3.3.3 (2D-parabolic-1D-hyperbolic KK-interface boundary condition stability)**.** *Let the parabolic convection-diffusion equations, (3.60), defined at the* $\Omega_l = [0, L_x] \times [0, L_y]$*, and the hyperbolic equation (3.68) defined at* $\Omega_r = [L_x, L]$ *with homogeneous Neumann boundary conditions at the outer boundaries* $\delta_{out}\Omega_l = \Omega_l \setminus \delta_{interface}\Omega_l$ *and* $\delta_{out}\Omega_r = \{L\}$ *be connected at the boundary interface* $\delta_{interface}\Omega_l = \{L_x\} \times [C, C + \sigma]$ *and* $\delta_{interface}\Omega_r = \{L_x\}$ *defined as* KK*-interface boundary condition with KK-parameter* $K \in \mathbb{R}_{>0}$ *and corridor width* $\sigma > 0$ *with* $0 < C < L_y - \sigma$*. Furthermore let the finite difference scheme (3.24), consistent to (3.60), and (3.26) consistent to (3.68) be complemented with homogeneous Neumann boundary conditions (3.29), (2.143) each at the outer boundaries and with the discretized KK-interface boundary conditions at the inner boundary (3.61) and (3.69).*
Then the complete finite difference schemes stability conditions at the inner points,(3.25) and (3.27) complemented with the additional stability constraint for the KK*-interface boundary conditions are*

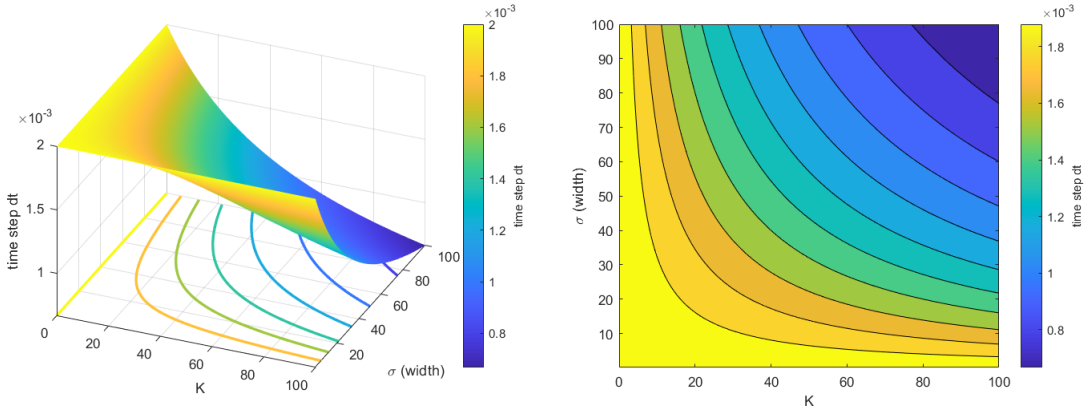
$$\delta_{interface}\Omega_l \begin{cases} \triangle t \leq \dfrac{1}{D\left(\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2}\right) + \frac{2|\widehat{f}^{x,n}_{N_x+1,j}|}{\triangle x} + \frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y} + \frac{K}{\triangle x}}, & \text{for } \widehat{f}^{x,n}_{N_x+1,j} < 0, \\ \triangle t \leq \dfrac{1}{D\left(\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2}\right) + \frac{|\widehat{f}^{y,n}_{N_x+1,j}|}{\triangle y} + \frac{K}{\triangle x}}, & \text{else} \end{cases}$$

and

$$\delta_{interface}\Omega_r \left\{ \triangle t \leq \frac{1+\rho}{\frac{2\lambda_c}{\triangle x}(\rho-1) - \left(\frac{\triangle t}{2\tau} - \lambda_c \frac{\triangle t}{\triangle x}\right)(\rho-1) + K_k \frac{\sigma(1+\rho)}{\triangle x} + \frac{\widehat{f}^n_0}{2\lambda_c \tau}} \right. \tag{3.73}$$

with $\rho := \frac{\lambda_c - \sigma K}{\lambda_c + \sigma K}$.

In Figure 3.4 we have visualized the time step restriction (3.72) for a qualitative understanding of how the Kedem-Katchalsky constant K and channel width σ impose more restrictive stability conditions on the time step $\triangle t$.



(a) (b)

Figure 3.4: Time step restriction (3.72) $\triangle t$ for the hyperbolic interface boundary condition with $\triangle x = 0.01$, $\triangle y = 0.1$ and $\lambda_c = 5$ for different K and channel width σ for the interface boundary between the two-dimensional parabolic equation (3.60) with the one-dimensional hyperbolic equation (3.68) with $\widehat{f} = 0$. As expected the time step $\triangle t$ must be chosen smaller when either K or σ increases. Furthermore, for $K = 0$ we recover the time step restriction of the $(AHO)^2$-scheme $\triangle t \leq \frac{4\triangle x\tau}{\triangle x + 4\lambda_c \tau} = 2 \cdot 10^{-3}$. Since the values of K typically are of similar magnitude as the diffusion coefficients, the additional stability restriction causes by the hyperbolic part of the interface boundary conditions are minimal.

Finally we also point out that the time step restriction for the interface boundary condition for the one-dimensional parabolic equation (3.64) is much more severe than for the one-dimensional hyperbolic equation (3.72) as can also be qualitatively evaluated by the steepness of Figure 3.3 and 3.4.
Comparing all time step restrictions (3.64), (3.59) and (3.72) with each other, the restriction for the two-dimensional parabolic interface boundary condition dominates the full model.

We conclude this section with an example which shows the influence the KK-constant $k \in \mathbb{R}_{\geq 0}$ has on the dynamics of a 1D-1D-parabolic model and a 1D-parabolic-1D-hyperbolic model for different relaxation times τ, connected at two interfaces.

Example 1. *We consider two one-dimensional parabolic equations (3.49) on* $\Omega_l := [0, 1]$ *and* $\Omega_r := [2, 3]$ *with* $\triangle x = 0.01$, $D = 1$, $\widehat{f_l} = \widehat{f_r} = 3$ *with no-flux boundary condition at the outer boundaries* $\sigma_{out}\Omega_l \cup \sigma_{out}\Omega_r = \{0\} \cup \{3\}$ *and KK-interface boundary conditions at* $x = 1$ *and* $x = 2$ *with a one-dimensional hyperbolic equation (3.68) on* $\Omega_c := [1, 2]$ *with* $\widehat{f_c} = 3$ *and* $\lambda_c = \sqrt{\frac{D}{\tau}}$ *for different values for the KK-parameters* $K = 0.1, 1, 10, 100, 1000$ *and different relaxation times* $\tau = 1, 0.1, 0.01$. *The time step* $\triangle t$ *satisfies stability conditions (3.59) and (3.72).*
Additionally, we also consider the case the hyperbolic equation in the center

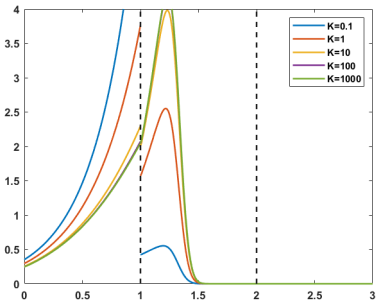
Ω_c is replaced by a parabolic equation.
Initial conditions for the left, central and right domain respectively are

$$\begin{cases} u_{l_0}(x) &= 10e^{-10(x-0.5)^2}, & \text{for } x \in \Omega_l, \\ u_{c_0}(x) &= 0, & \text{for } x \in \Omega_c, \\ v_{c_0}(x) &= 0, & \text{for } x \in \Omega_c, \\ u_{r_0}(x) &= 0, & \text{for } x \in \Omega_r. \end{cases} \tag{3.74}$$

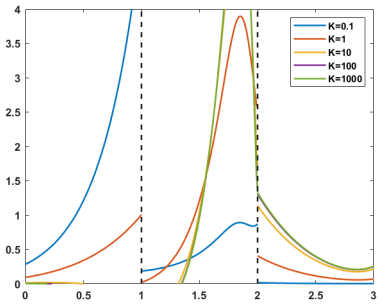
Results: *In Figure 3.5 the numerical solution is shown for $K = 0.1, 1, 10, 100, 1000$ at times $t = 0.37$ and $t = 1$ for different relaxation times $\tau = 1, 0.1$ and 0.01. Due to the constant convection term $\widehat{f} = 3$ in all three domains, the cell density u migrates from the left domain through the central domain into the right domain.*
For small KK-constant K, the permeability is decreased which results in an accumulation at $x = 1$ and also visibly at $x = 2$. The larger the value of K becomes, the smaller the discontinuity between left and right side of the interface becomes.
Furthermore, for larger relaxation times τ the flow of density u from the left domain through the center domain into the right domain is slower, as can been seen for $t = 0.37$.
For larger time $t = 1$ it seems that the dynamics of the parabolic model compared to the hyperbolic for $\tau = 0.1, 0.01$ are the same, whereas for $\tau = 0.01$ there is no significant difference at both times.
Lastly, we want to remark that the initial mass of $\Omega := \Omega_l \cup \Omega_c \cup \Omega_r$

$$I_0 = \int_\Omega u_0(x)\mathrm{dx} \approx 5.46292 \tag{3.75}$$

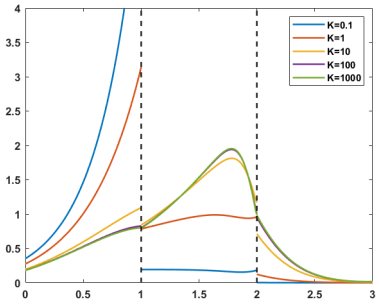
is preserved in all models which shows that the KK-transmission boundary conditions are mass-preserving.



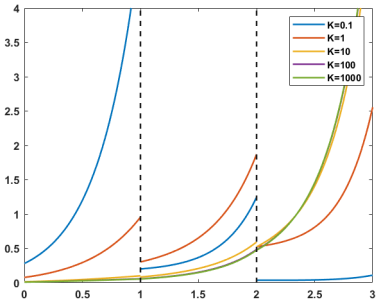
(1): $\tau = 1$ at time $t = 0.37$.



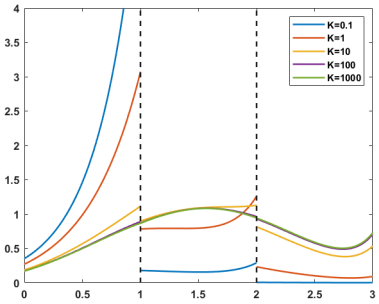
(2): $\tau = 1$ at time $t = 1$.



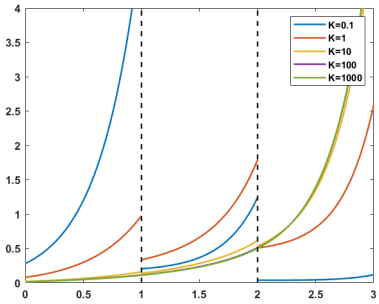
(3): $\tau = 0.1$ at time $t = 0.37$.



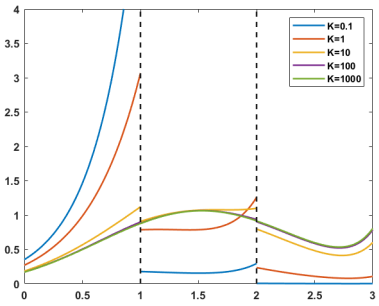
(4): $\tau = 0.1$ at time $t = 1$.



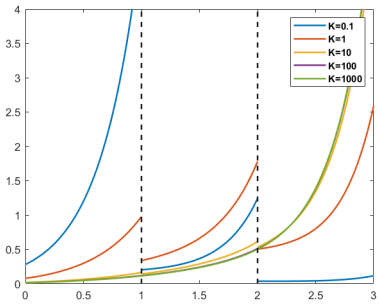
(5): $\tau = 0.01$ at time $t = 0.37$.



(6): $\tau = 0.01$ at time $t = 1$.



(7): Parabolic equation in Ω_c at time $t = 0.37$.



(8): Parabolic equation in Ω_c at time $t = 1$.

Figure 3.5: Influence of KK-interface boundary conditions for $K = 0.1, 1, 10, 100, 1000$ for two interfaces, connecting three domains. Parabolic equations defined in Ω_l and Ω_r. In Ω_c a hyperbolic model in Ω_c for relaxation times $\tau = 1, 0.1, 0.01$ is defined and also for a parabolic model.

3.3.2 Numerical Tests and Results

With the finite difference methods from Section 3.2.1, the derivation of appropriate mass preserving outer boundary conditions and KK-interface boundary conditions from Section 3.2.2 and 3.2.3 and time step and meshgrid size restrictions from 3.3.1 we are finally able to simulate the model (3.1), (3.3) and (3.4) presented in the first part of Chapter 3.

Initial distribution

As initial conditions we use the following for the two-dimensional chambers Ω_l and Ω_r

$$\begin{aligned}
M(0,x,y) &= \begin{cases} 5\cdot 10^{-2}\frac{cell}{\mu m^2}, & \text{for } (x,y)\in\Omega_r \text{ with } x\in[670,700], \\ 0\frac{cell}{\mu m^2}, & \text{for } (x,y)\in\Omega_l, \end{cases} \\
T(0,x,y) &= \begin{cases} 0\frac{cell}{\mu^2 m}, & \text{for } (x,y)\in\Omega_r, \\ 10^{-3}e^{-\frac{x^2+(y-500)^2}{1000}}+10^{-3}e^{-\frac{x^2+(y-1000)^2}{1000}} & \\ \quad +10^{-3}e^{-\frac{x^2+y^2}{1000}}\frac{cell}{\mu m^2}, & \text{for } (x,y)\in\Omega_l, \end{cases} \\
\phi(0,x,y) &= 0,\frac{mol}{\mu m^2}, \quad \text{for } (x,y)\in\Omega_l\cap\Omega_r, \\
\omega(0,x,y) &= 0,\frac{mol}{\mu m^2}, \quad \text{for } (x,y)\in\Omega_l\cap\Omega_r
\end{aligned} \tag{3.76}$$

and for the one-dimensional channels Ω_{c_k}

$$\begin{aligned}
M(0,x)=T(0,x) &= 0,\frac{cell}{\mu m}, \\
\phi(0,x)=\omega(0,x) &= 0\frac{mol}{\mu m}.
\end{aligned} \tag{3.77}$$

For the two-dimensional chambers Ω_l, Ω_r we use the two-dimensional parabolic model (3.1), whereas for the one-dimensional channels Ω_c we mainly use the one-dimensional hyperbolic model (3.4) and only conduct one numerical simulation where we use the one-dimensional parabolic model (3.3) for comparison reasons.
The initial distributions are shown in Figure 3.6.
This choice of initial conditions corresponds with the laboratory experiment where the immune cells M are almost evenly distributed in the right chamber Ω_r. We set the density of immune cells in the left chamber to zero to better see the dynamics of the immune cell migration towards the tumour cells.
As for the tumour cells, we choose a normal distribution at three locations in the left chamber which simulates the case where only a small number of tumour cells is present in the left chamber.
For the chemoattractant ϕ and cytokine ω we set the initial distribution to

zero. Although we expect in the laboratory experiment the presence of chemoattractant and cytokine evenly distributed in channels and chambers, we neglect those since no data of their distribution are currently available in the laboratory experiment and set them to zero.
Additionally, as seen in Figure 1.8, the outer boundaries of the domains are open, yet in this numerical simulation we propose closed boundaries, i.e. homogeneous Neumann boundary conditions.

Chemotactic and source terms

For the chemotactic term $f := M\widehat{f} = M\chi(M,\phi)\nabla\phi$ we are considering several types as presented in Section 1.6.

I **Basic model** (directional movement up a spatial gradient of chemoattractant) (1.52)
$$\chi(M,\phi) := k_3. \tag{3.78}$$

II **Receptor saturation** (dependence on concentration of chemoattractant in a cell's local environment [140]) (1.53)
$$\chi(M,\phi) := \frac{k_1}{(k_2+\phi)^2}. \tag{3.79}$$

III **Overcrowding** (ability to move freely reduces at high densities as cell density approaches maximum value M_{max} [71]) (1.54)
$$\chi(M,\phi) := \frac{k_1}{(k_2+\phi)^2}\left(1-\frac{M}{M_{max}}\right). \tag{3.80}$$

IV **Interaction** (migration of cells in response to gradients of their own density and of chemoattractant [49]) (1.55)
$$\widehat{f} := \frac{k_1}{(k_2+\phi)^2}\nabla\phi + \frac{\eta_1}{1+\eta_2 M}\nabla M. \tag{3.81}$$

V **Interaction** (chemoattractant free [49]) (1.56)
$$\widehat{f} := \frac{\eta_1}{(\eta_2+T)^\gamma}\nabla T. \tag{3.82}$$

For the decay rate of tumour cells we choose
$$\lambda_T(\omega) := \frac{k_{w_1}}{k_{w_2}+\omega}, \tag{3.83}$$

and do not take into account the administration of drugs, hence

$$k_T(t) = k_M(t) = 0$$

since they were not conducted in the laboratory experiments.

As for the values of the parameters, geometry of the chip, and time step and mesh grid size, which are used for the following numerical simulations, they can be found in table 3.1.
For the finite difference methods we have used $\triangle x = 2.5\mu m$ and $\triangle y = 25\mu m$ and choose $\triangle t$ according to satisfy the stability conditions.
For a qualitative comparison between the different dynamics due to the variety of chemotactive terms f, we have chosen coefficients k_1, k_2, k_3, η_1 and η_2 to have similar magnitudes.
As for the geometry, we have chosen $m = 14$ channels Ω_{c_m}. An increase of number of channels left unchanged both the dynamics and the values of densities in both chambers.
For the numerical simulations in Figure 3.7-3.12 we have chosen the priorly presented finite difference schemes.

Graphical representation and explanation

Figure 3.7-3.12 show the evolutions of T,M,ϕ and ω over time for different chemotactic terms f.
In Figure 3.11 the parabolic equation for the one-dimensional channels with the receptor saturation model (3.79) has been used.
Figure 3.12 is a special case where we have chosen different parameters compared to the ones in Table 3.1, where the diffusion coefficient D_T is reduced one hundred fold to simulate more fixated tumour cells, D_M reduced ten fold but the diffusion of the chemoattractant D_ϕ and D_ω increased ten fold, in order to have a more chemotactic dominant dynamic.
In all figures the value of density is represented by a color map with its corresponding color bar.

We need to remark that the color bar indicates the value of concentration per area in the two-dimensional chamber and per length in the one-dimensional channels.
This allows us to visually investigate the dynamical behaviour in the chambers and channels but does not allow to compare the concentration quantities in the one-dimensional channels with the two-dimensional chambers only based on their color to each other.
Furthermore, even when a density distribution in a chamber looks constant according to its color and corresponding color value, this can be misleading

if only small changes of concentration are present which might be not noticeable due to a marginal change in color.
For this reason we included monochromatic contour lines ranging from the color black to white to better visualize the motion and shape of the densities in the chambers and around the interfaces between two-dimensional chambers and one-dimensional channels. The contours have to be interpreted relatively to the whole two-dimensional domain. If the color value indicates a constant distribution, the contours are able to show slight accumulation and de-cumulation of density.
This visualization helps to compare the dynamics of each model better.

Figure 3.13 shows a direct comparison of the immune cell dynamics M between the different models with the chemoattractant distribution for $t = 250$.

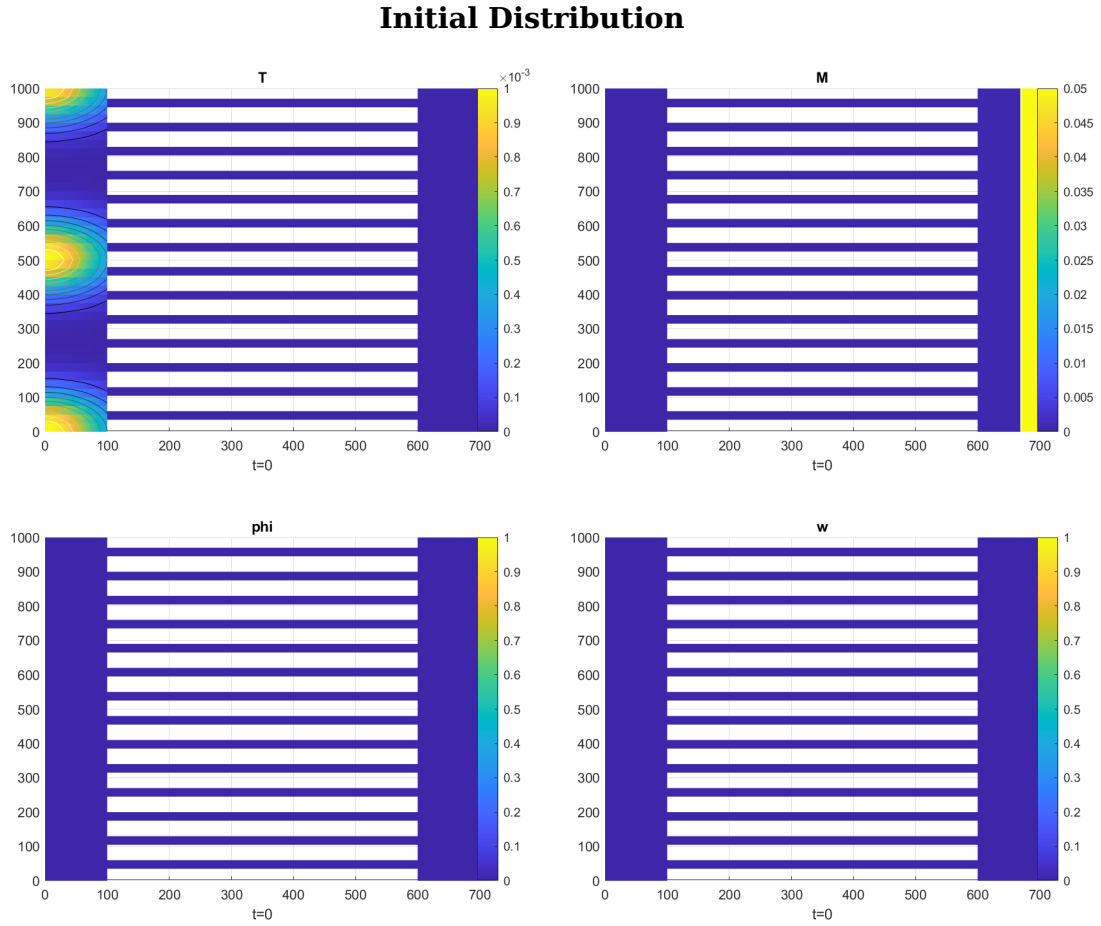


Figure 3.6: Initial distribution of T_0, M_0, ϕ_0 and ω_0 (3.76). Immune cells are distributed evenly in the right chamber Ω_r, tumour cells concentrated in three locations in the left chamber.

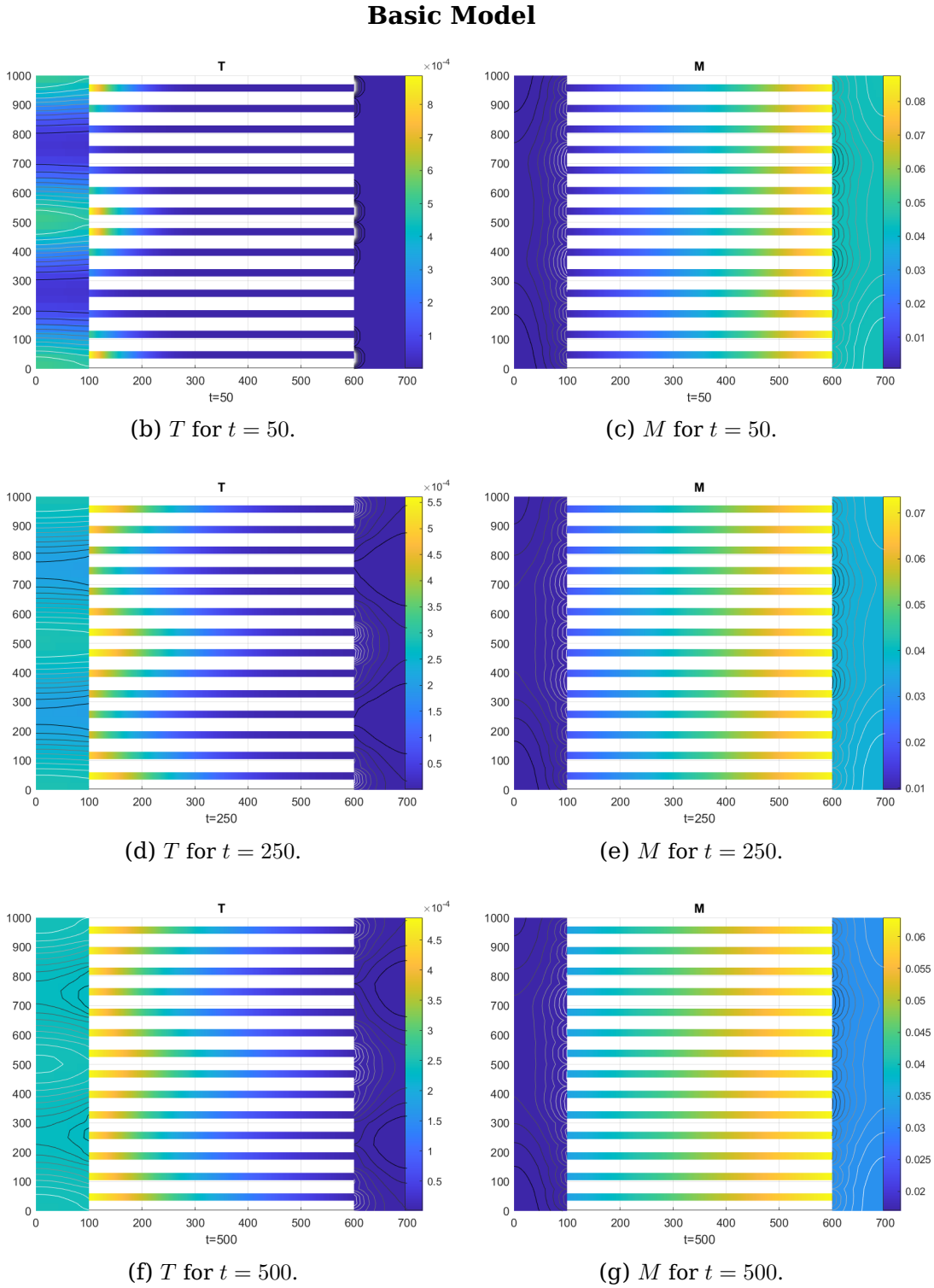
Influence of chemotactic terms on immune cell dynamics

General similarities

When comparing the different models with equal diffusion coefficient D_T the dynamics of T is similar due to the fact that the models differ only in the chemotactic term $\widehat{f}$ which influences the dynamic of immune cells M and consequently the released cytokine ω. All models show a reduction in tumour density T towards an almost constant value of $2.5 \cdot 10^{-4}$ for $t = 500$.
Furthermore, we can see in all models that the tumour cells are accumulating around the entry part of the one-dimensional channels. However a small quantity of tumour cells have been able to enter the right chamber Ω_r, mainly focused on the top, middle and bottom channel exits.
Consequently, the evolution of the chemoattractant distribution ϕ over time

Parameter	Description	Units	Value	Ref.
D_M	diffusivity of cells	$\mu m^2/s$	9×10^2	[105]
D_T	diffusivity of cells	$\mu m^2/s$	5.6×10^1	[105]
D_ϕ, D_ω	diffusivity of chemoattractants	$\mu m^2/s$	2×10^2	[105]
α_T	decay rate of drug release	s^{-1}	0	-
α_M	decay rate of drug release	s^{-1}	0	-
K_T	decay rate of T caused by drug	s^{-1}	0	-
K_M	decay rate of M caused by drug	s^{-1}	0	-
α_ϕ	growth rate of ϕ	$s^{-1}/cell$	10^{-1}	[37]
β_ϕ	consumption rate of ϕ	s^{-1}	10^{-4}	[37]
α_ω	growth rate of ω	$s^{-1}/cell$	10^{-1}	[37]
β_ω	consumption rate of ω	s^{-1}	10^{-4}	[37]
k_1	cellular drift velocity	Mcm^2s^{-1}	$3.9 \cdot 10^{-9}$	[105]
k_2	receptor dissociation constant	M	$5 \cdot 10^{-6}$	[105]
k_3	drift velocity	Mcm^2s^{-1}	156	[105]
η_1	cellular drift velocity	Mcm^2s^{-1}	$3.9 \cdot 10^{-9}$	[105]
η_2	receptor dissociation constant	M	$5 \cdot 10^{-6}$	[105]
k_{ω_1}	killing efficiency of immune cells	$cell^{-1}\mu m/s$	1	-
k_{ω_2}	dissociation constant	$cell/\mu m^2$	1	-
γ	exponent in chemotactic response χ		2	[105]
M_{max}	maximum cell density	$cell/\mu m^2$	M_0	[71]
τ_M	relaxation time of M	s	1	-
τ_T	relaxation time of T	s	1	-
L	length of the channel	μm	500	[31]
σ	width of the channel	μm	12	[31]
$\tilde{\sigma}$	width between channels	μm	33	[31]
L_x	horizontal size of the chamber	μm	100	[31]
L_y	vertical size of the chamber	μm	1000	[31]

Table 3.1: Parameters of the problem.



(b) T for $t = 50$.

(c) M for $t = 50$.

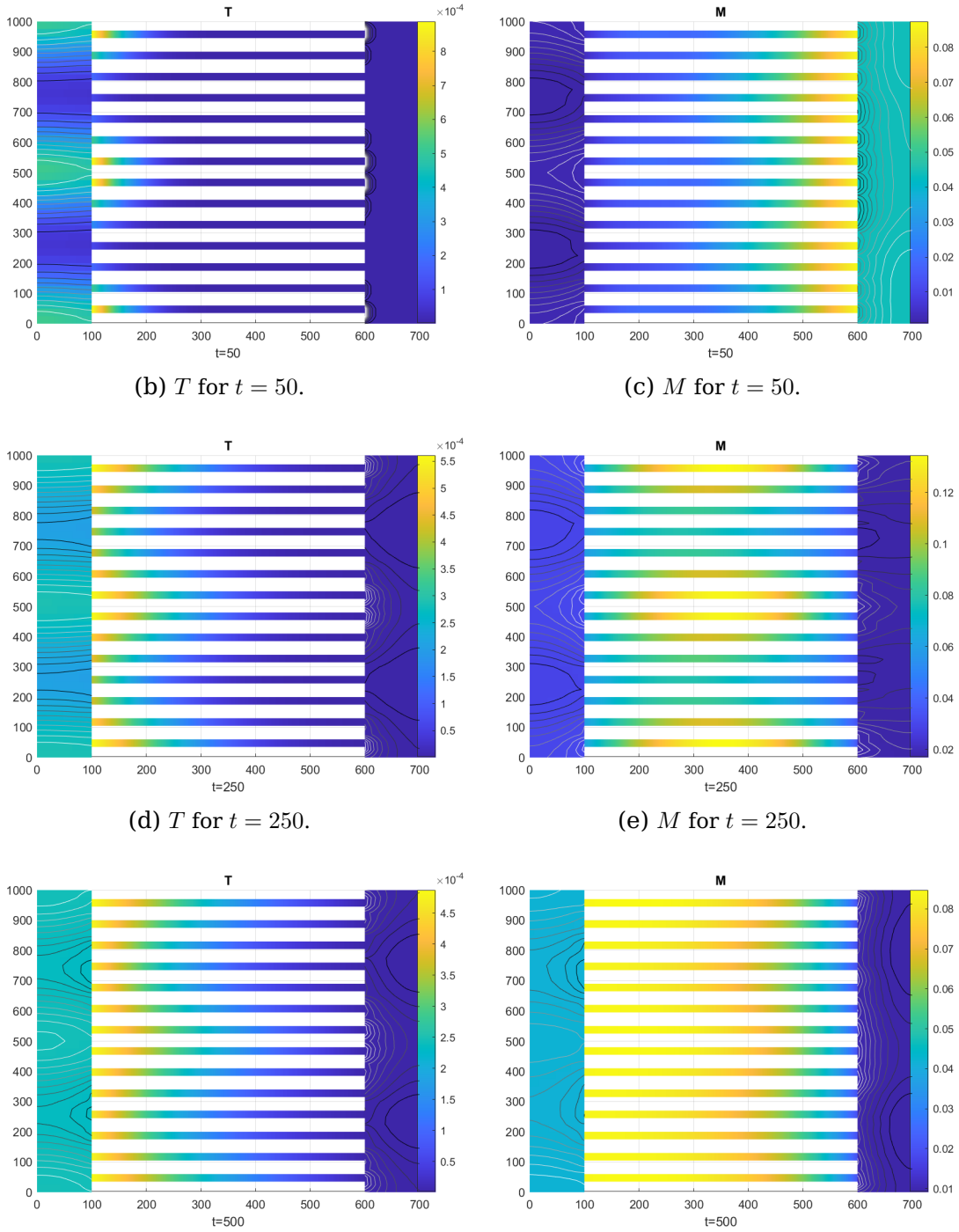
(d) T for $t = 250$.

(e) M for $t = 250$.

(f) T for $t = 500$.

(g) M for $t = 500$.

Figure 3.7: Evolution of T (left)and M (right) for the basic model (3.78). Immune cells migrating through all channels into the left chamber where the tumour cells are present.



(b) T for $t = 50$.

(c) M for $t = 50$.

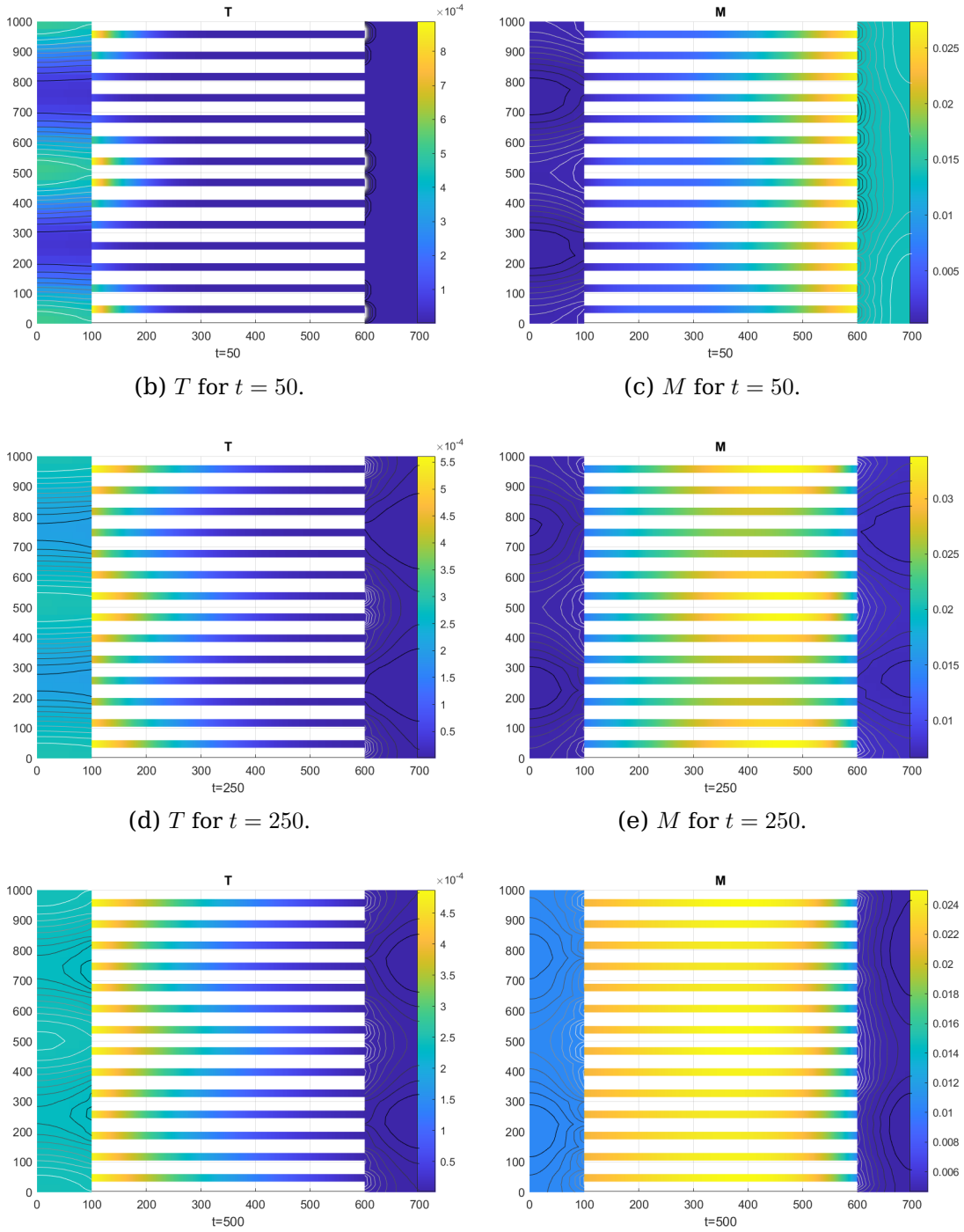
(d) T for $t = 250$.

(e) M for $t = 250$.

(f) T for $t = 500$.

(g) M for $t = 500$.

Figure 3.8: Evolution of T (left) and M (right) for the receptor saturation model (3.79). Immune cells favour bottom,top and central channels for migration initially and most noticeable at $t = 50 - 250$.

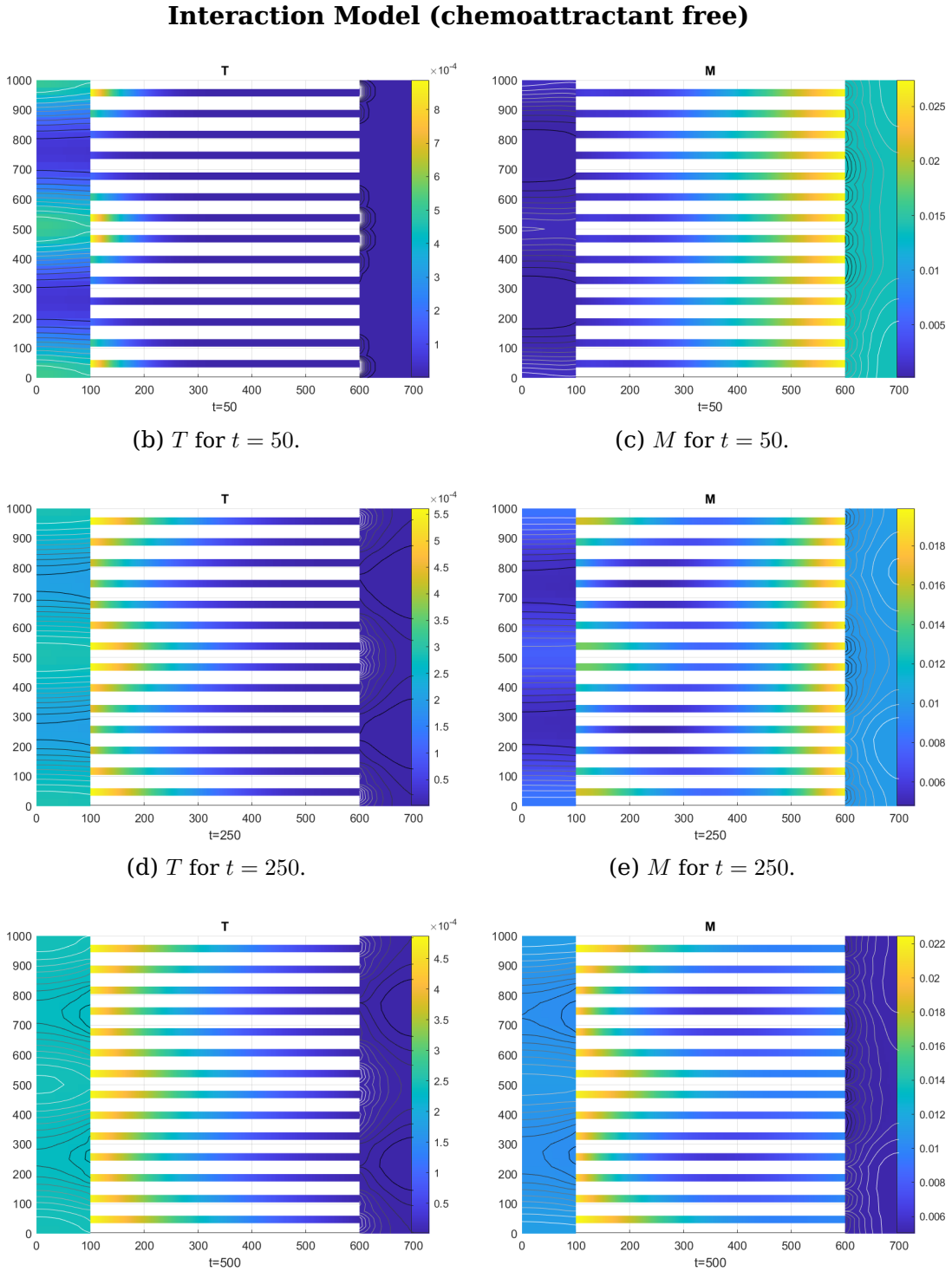


(b) T for $t = 50$. (c) M for $t = 50$.

(d) T for $t = 250$. (e) M for $t = 250$.

(f) T for $t = 500$. (g) M for $t = 500$.

Figure 3.9: Evolution of T (left) and M (right) for the overcrowding model (3.80). Similar dynamics as for the receptor saturation model but the white contours indicate a stronger migation of immune cells through the central channels.

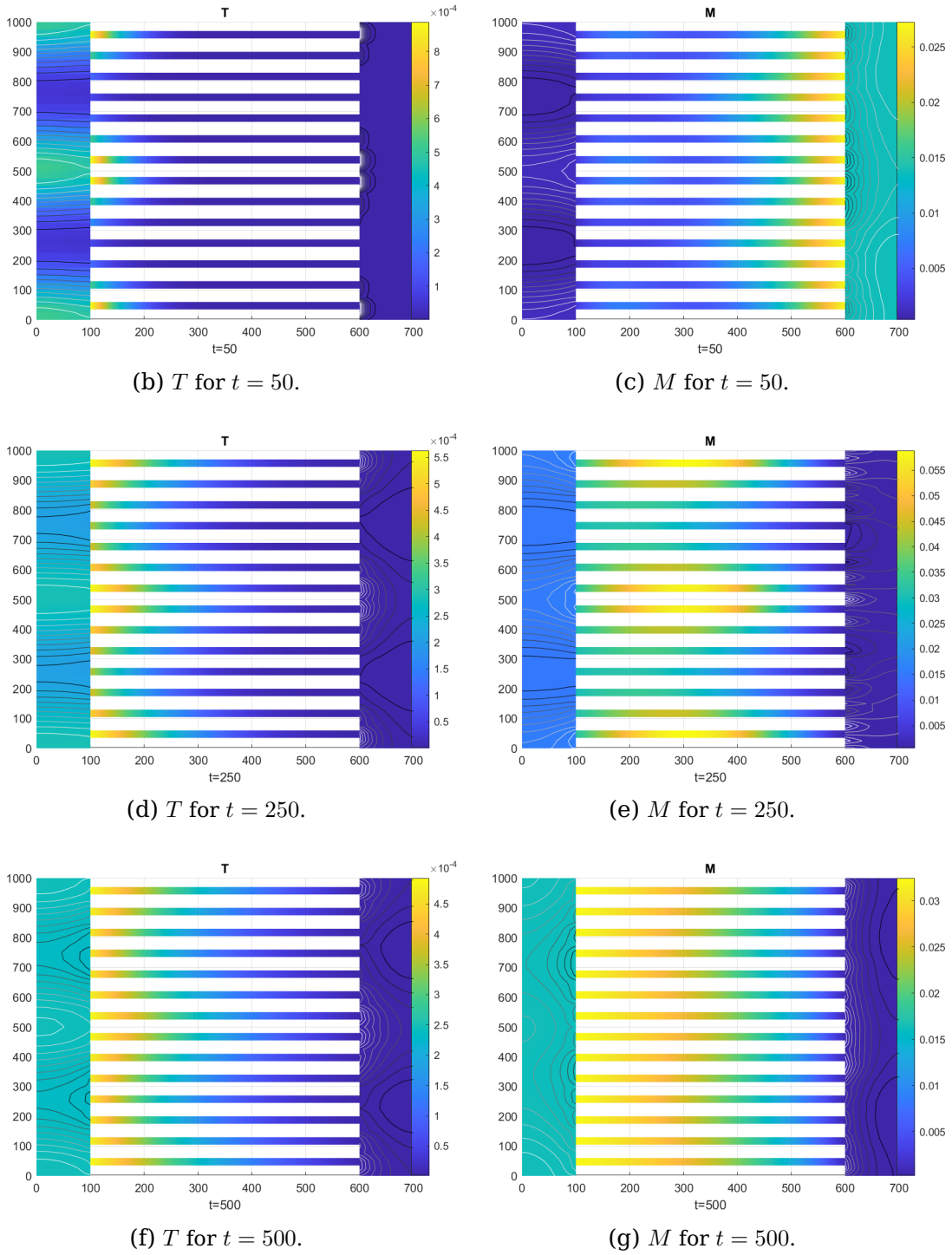


(b) T for $t = 50$. (c) M for $t = 50$.

(d) T for $t = 250$. (e) M for $t = 250$.

(f) T for $t = 500$. (g) M for $t = 500$.

Figure 3.10: Evolution of T (left) and M (right) for the interaction (chemoattractant free) model (3.82). The results are very similar to the receptor-saturation model.



(b) T for $t = 50$.

(c) M for $t = 50$.

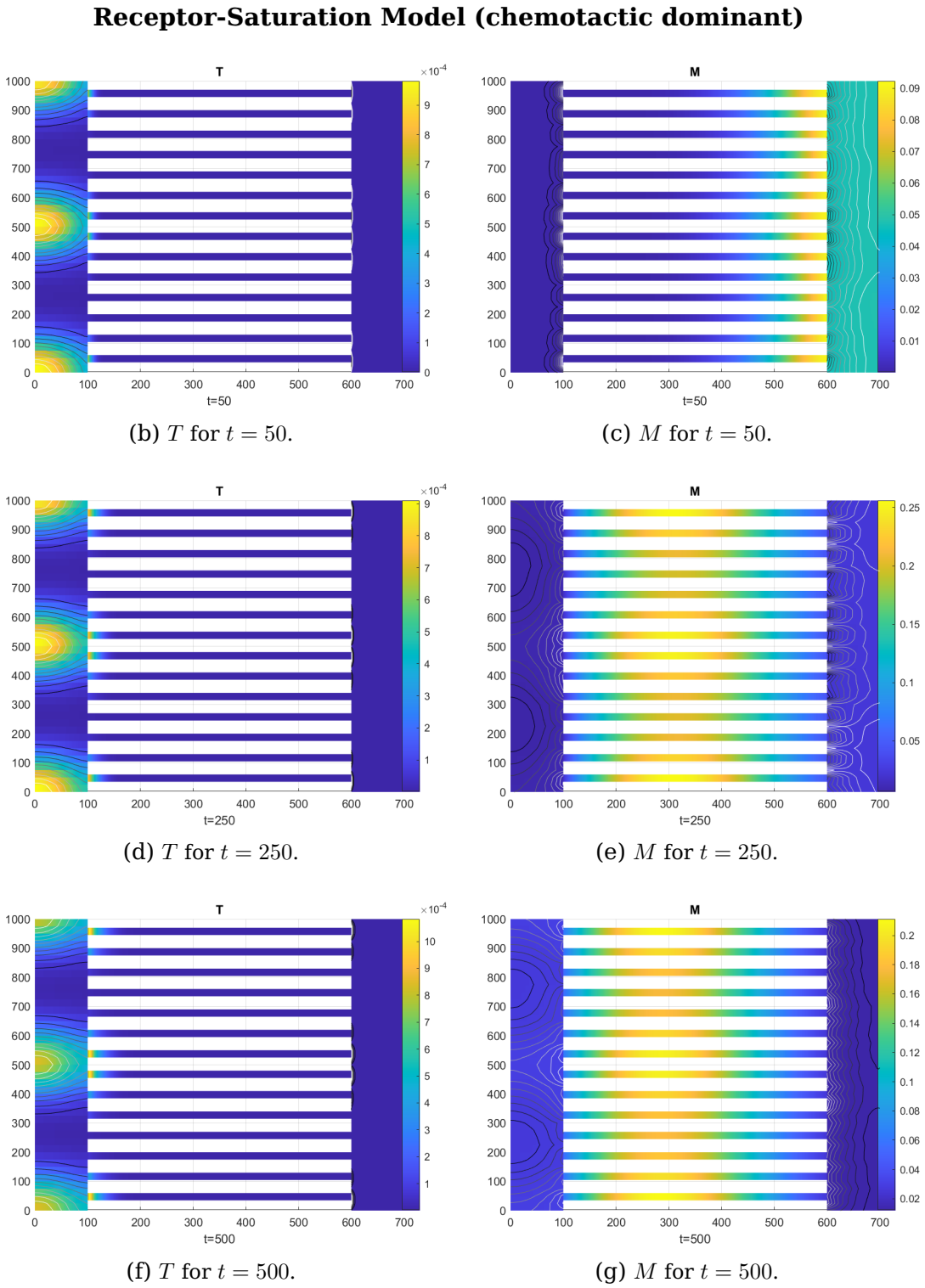
(d) T for $t = 250$.

(e) M for $t = 250$.

(f) T for $t = 500$.

(g) M for $t = 500$.

Figure 3.11: Evolution of T (left) and M (right) for the receptor saturation model (3.79) and the one-dimensional channels modelled by the one-dimensional parabolic equation (3.3). Similar dynamics compared to the hyperbolic equation in channels but overall slower migration of immune cells into the right chamber.



(b) T for $t = 50$.

(c) M for $t = 50$.

(d) T for $t = 250$.

(e) M for $t = 250$.

(f) T for $t = 500$.

(g) M for $t = 500$.

Figure 3.12: Evolution of T (left) and M (right) for the receptor saturation model (3.79) with lower diffusion of cells to ensure sharper and stronger chemotaxis. The dynamics are similar to the model with larger diffusion, but the immune cells are more concentrated around the micro-channels.

can be represented by the dynamics of the tumour cells T which are producing ϕ.

Comments on results depicted in Figures 3.7-3.12

The vigorous difference of each model is the dynamics of the immune cells M and thus of the cytokine ω.

Basic model (Figure 3.7)

In the basic model (3.78) we can see that immune cells have been able to enter the left chamber Ω_l. In the right chamber Ω_r we can clearly see through the contour lines (black indicates decrease) that there is a negative slope of immune cell density M towards the channel entries, whereas in the one-dimensional channels an accumulation occurs. This is caused by the chemoattractant ϕ, which accumulates around the entries of the channels in the right chamber causing the immune cells to migrate according to the chemoattractant gradient and getting trapped at the highest chemoattractant concentration around the entry areas of the channels. We want to remark however that a direct comparison of the concentration between the one-dimensional channels and two-dimensional chambers can be misleading due to the fact that in the one-dimensional channels the color bar values indicate the concentration per length unit but for the two-dimensional chambers area per unit.0

Receptor-Saturation model (Figure 3.8)

For the receptor saturation model (3.79), we notice that more immune cells have migrated towards the left chamber, as can also be seen by the shift of density in the channels towards the left.
In the receptor saturation model, once the immune cells M are reaching those chemoattractant concentration peaks around the entry areas of the channels, they are able to move through because the chemotaxis is being reduced in presence of high chemoattractant concentrations through saturation. This makes the diffusive component of the movement of the immune cells influential enough to continue their migration through the channel and into the left chamber.
Additionally, there is a higher concentration of immune cells around the middle, top and bottom channels, explained by the fact that a larger quantity of chemoattractant is present in these areas. Also there is a larger accumulation around those channel entries, caused by the strong convective force.

Overcrowding model (Figure 3.9)

Comparing the basic and receptor-saturation model with the overcrowding model (3.80) we notice that much more cells have been able to migrate towards the left chamber as in the receptor saturation model. Comparing the contour lines shows similar dynamics between overcrowding and receptor-saturation model. The major difference is in the slightly shifted concentration peaks in the channels caused by the additional factor term $\left(1 - \frac{M}{M_{max}}\right)$ that reduces the chemotactic migration through channels compared to the receptor-saturation model.

Interaction (chemoattractant free) model (Figure 3.10)

We briefly investigate the interaction (chemoattractant free) model (3.82). In this model the chemotactic movement of the immune cells are solely influenced directly by the tumour cells T.
It is interesting to notice that the dynamics of the right chamber is inverted compared to the receptor-saturation and overcrowding models.
Whereas in the other models, the immune cells in the right chamber are accumulating near the entries of the channels, in the chemoattractant free model, the immune cells are gathering in the upper and lower right corner of the chamber, away from the channel entries.
Looking at the immune cell density distribution of the channels, we can deduce that the convective migration is more dominant than in the chemotaxis models.
An indication is the density distribution M in the channels which reflects the tumour cell distribution of the left chamber.
In the right chamber on the other hand, the immune cells are accurately migrating towards the highest concentrations of tumour cells T in the middle and corners of the chamber.
This behaviour can be explained by the fact that with the chemoattractant ϕ the accurate position of the tumour cells can only be determined indirectly because of the higher diffusion, the migration is being slowed down and diffused.

Receptor-Saturation model (1D-parabolic model in channels) (Figure 3.11)

Modelling the one-dimensional channels with the parabolic equation (3.3) and comparing the contour lines of the receptor-saturation model between the hyperbolic and parabolic one-dimensional modelled channels shows similar dynamics in the chambers. However, the propagation of immune cells

from the right to the left chamber seems to be faster compared to the hyperbolic Receptor-Saturation model, which is also noticeable in Figure 3.13 where the peak of immune cell concentration in the channels migrated further than in the hyperbolic model.

Receptor-Saturation model with low cell diffusion (Figure 3.12)

Lastly, we investigate the dynamics of the receptor-saturation model, where we have lowered the cell diffusion and increased the chemical diffusion in order to enforce a stronger convective situation.
Comparing this model with the other two receptor-saturation models shows clearly that according to the contour lines, the dynamics in the left chamber are similar between these three models but in the right chamber the situation is inverted with a strong decline of immune cell concentration around the channel entries caused by the strong convection.

Particle Representation (Figure 3.14)

Our mathematical model tries to describe the laboratory experiment of the microfluidic chip in a macroscopic way in form of density fields.
This is of importance considering that Organ on Chips (OOC) are developed to simulate tissues, organs and other biomaterial with millions of cells.
The available data and observations of the laboratory experiment on the other hand are on a microscopic level where only the trajectories of cells as data are available.
For this reason we also have a particle representation of the numerical simulations for each model at time $t = 500$ shown in Figure 3.14.
The particles for each model has been obtained with the acceptance-rejection method (7.1) that will be described in Section 7.1 where we will give a more detailed explanation. The particle representation allows us to see the different dynamics of migration of immune cells M (blue) and the decay of tumour cells T (red) for different chemotactic terms f.
We do not include a particle representation in the one-dimensional channels due to the fact that the combination between particles in a two-dimensional domain with one-dimensional domains would create a particles in channels which are very close to each other and does not represent the observation made in the laboratory experiments [15, 31].

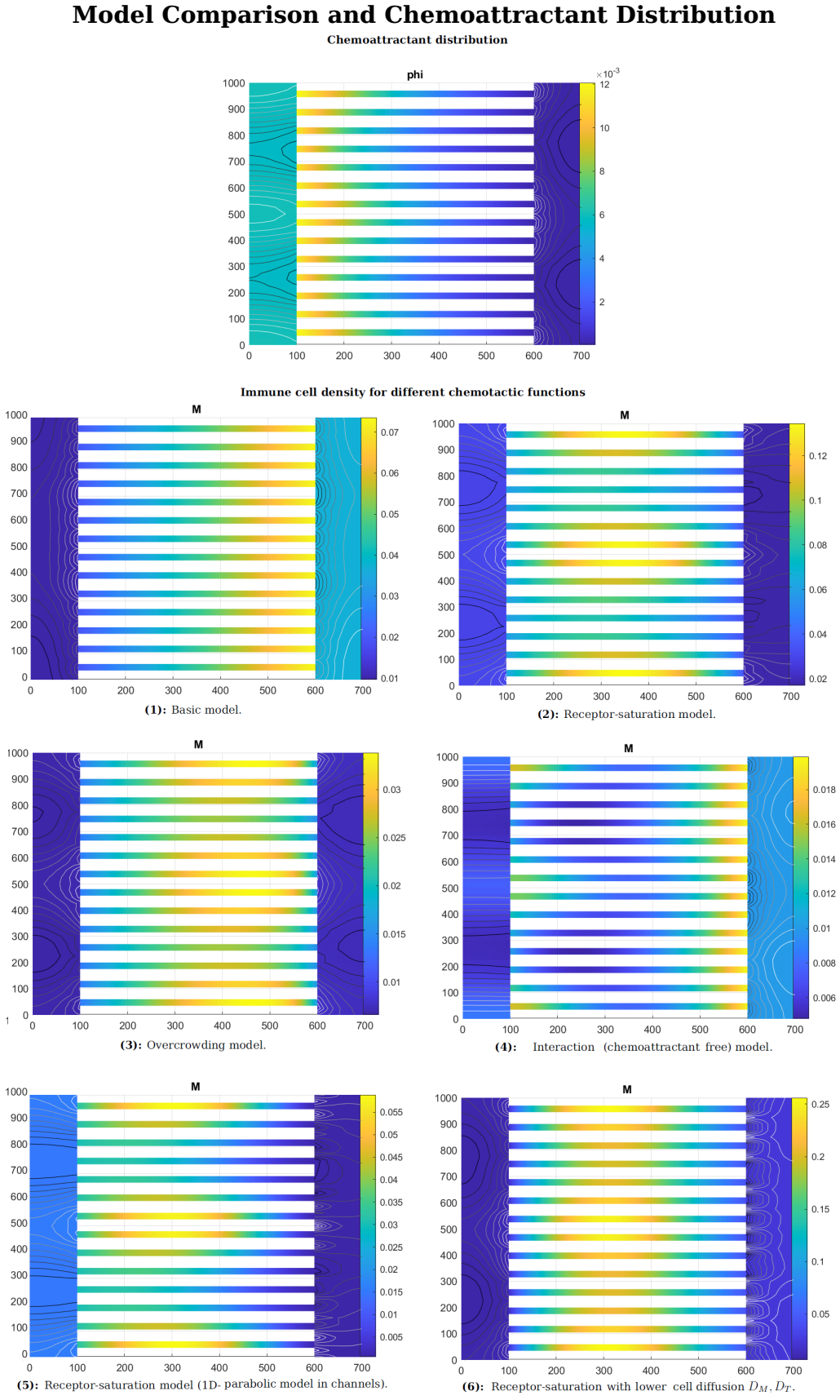
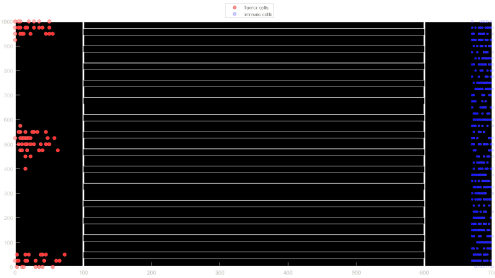
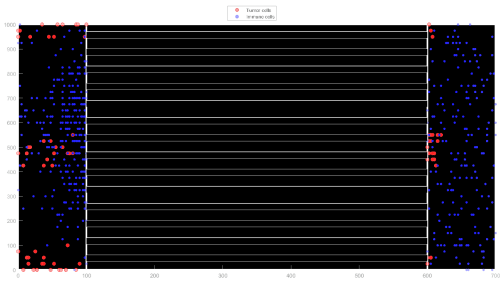


Figure 3.13: Chemoattractant distribution ϕ and immune cell distribution M at time $t = 250$ for different chemotactic functions f and different parameter values.

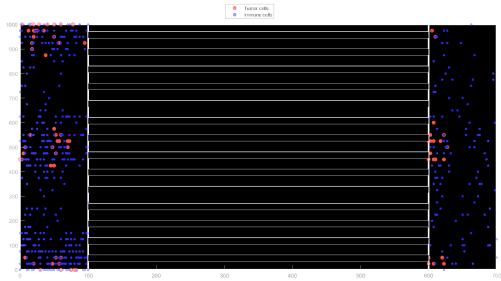
Particle Representation



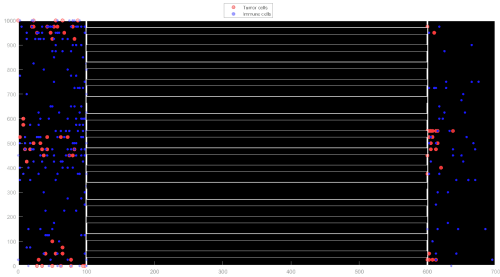
Initial distribution $t = 0$



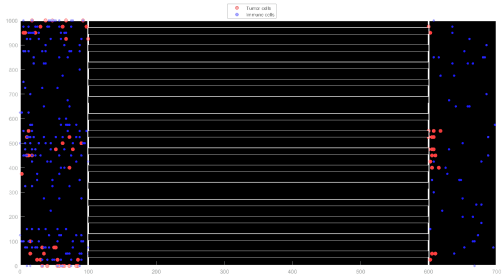
(c) I **Basic model**



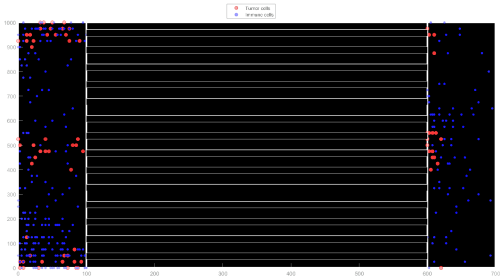
(d) II **Receptor saturation**



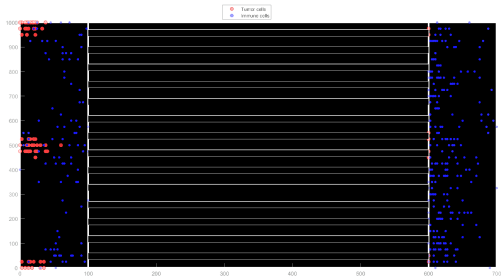
(e) III **Overcrowding**



(f) V **Interaction(chemotractant free)**



(g) **Fully parabolic in channels**



(h) **Receptor saturation with low diffusion for T and M and high diffusion for ϕ and ω**

Figure 3.14: Particle representation of solution in the two-dimensional chambers for different chemotactic terms f for time $t = 500$. Blue points: immune cells M. Red points : tumour cells T. Particles generated with the acceptance-rejection algorithm 7.1. No particles are visualized in the channels because they are mathematically modelled as one-dimensional domains, for which no satisfactory particle representation was possible.

Part II

Estimation of Parameters in Convection-Diffusion-Systems

In the previous two parts we have constructed a novel multi-domain transmission model for the OOC (1.45)-(1.47) based on the results made from the laboratory experiments [15, 31] and presented a numerical approximation of the model describing the qualitative behaviour of different cell species living in a confined environment.
The intrinsic parameters used for the model have been obtained from the literature, indicating typical values for each quantity (see table 3.1).
The following part deals with some background necessary to derive a method that is able to calibrate the OOC model with parameter values such that the model replicates the actual dynamics observed in the laboratory experiment based on real data. These methods to recover model parameters are called parameter estimation methods and are known as inverse problems [48] and have a wide application in a variety of scientific fields such as geophysics, medical imaging, machine learning and many more [10]. It is the first study about such methods applied to mathematical models on Organ-on-Chips.
The outline of Part II is the following.
In Chapter 4 we introduce the basic theory for solving equation systems. We begin with linear equation systems and explain the derivation of the most standard iterative methods, the relaxation methods, and define the general theory of convergence and consistency for iterative methods in general. We then further discuss the theory of linear multigrid methods as another iterative method, which takes advantages of error smoothing properties of relaxation methods [28, 29, 138].
Lastly, we present the Conjugated Gradient Method (CG-method), a prototype of Krylov-Subspace methods, a class of iterative methods, which are especially designed for solving large and sparse equation systems [100].
The following chapter focuses on root-finding numerical methods for non-linear equation systems such as Newton's methods and its variations [41, 113]. We explain the numerical challenges involved in solving non-linear equations and analyse certain requirements for convergence. We then proceed with the non-linear counterpart of multigrid methods by introducing multigrid methods for linear equation systems within non-linear solvers such as the Newton-Multigrid method (Newton-MG) [22] and the Full Approximation Scheme (FAS) which adopts the main idea of multigrid methods for the non-linear case [23].
Chapter 5 then serves as an introduction to non-linear optimization problems where numerical methods for finding local minima of non-linear objective functions are presented with their proper derivations and some convergence theory [4, 111, 132]. We will mainly focus on line search methods, which are numerical methods that search for a descendent direction that decreases the value of the objective function iteratively. Important theorems like the Wolfe and Armijo conditions [111] are discussed that are necessary to compute suf-

ficient step lengths for the descend direction in order to ensure convergent minimization methods.
We will then proceed with constrained optimization problems and analyse techniques to solve them under previously presented methods [111].
In Chapter 6 we finally define the parameter estimation problem as an inverse problem, explain the challenges of finding minima of non-linear ill-posed problems and introduce several ideas such as regularization techniques that helps in the identification of correct parameters [10, 48, 151].
In the same chapter we will also review certain methodology for parameter estimation problems found in the current scientific literature such as [8, 49, 80, 91, 92, 110] and analyse and interpret their results. In the last chapter of Part II we apply all the previously introduced theories and methods of parameter estimation problem to the general convection-diffusion equation. We modify existing methods and regularization techniques in order to create a robust parameter estimation method for OOC-models.

Chapter 4

Solving Linear and Non-Linear Systems

This chapter focuses on the solution of linear and non-linear systems which arise from the discretization of partial differential equation in form of implicit finite difference schemes.
We differentiate between direct and iterative methods for solving these systems. The advantage of direct methods is that they provide the exact solution (besides rounding errors) within a finite number of operations. The disadvantages are the computational costs and the storage required to perform the necessary operations, which are not feasible for big systems.
Iterative methods on the other hand, begin with a first initial approximation of the solution and iteratively improve it such that it converges towards the exact solution with much less computational cost and storage.
In the following section we investigate linear equation systems of the form

$$Ax = b \tag{4.1}$$

with a matrix $A \in \mathbb{R}^{n \times n}$ and vectors $b, x \in \mathbb{R}^n$, where the linear equation system has the unique solution $x = A^{-1}b$.
For an approximated solution of (4.1) $\tilde{x} \in \mathbb{R}^n$ let the error $e \in \mathbb{R}^n$ be defined as

$$e = x - \tilde{x} \tag{4.2}$$

and the residual $r \in \mathbb{R}^n$

$$r = b - A\tilde{x} \tag{4.3}$$

with the relation between error e and r

$$Ae = r. \tag{4.4}$$

The most noteworthy direct methods are

- **Gauß-Elimination**: $A = LU$

- **Cholesky factorization**: $A = LL^T$
- **QR factorization**: $A = QR$,

where the system matrix $A \in \mathbb{R}^{n\times n}$ is decomposed into a product of two matrices $L, U \in \mathbb{R}^{n\times n}$ that are lower and upper tridiagonal matrices and $Q, R \in \mathbb{R}^{n\times n}$ with Q orthogonal and R upper tridiagonal matrix.
Although direct methods are not used for large systems and will not be discussed any further in this thesis, they can be used as preconditioners to increase the rate of convergence and stability of iterative methods. We refer to the literature [29, 100] where the algorithms for direct methods are being described and preconditioners discussed.

If the mathematical problem involves the solution of a system of equations with $F : \mathbb{R}^n \to \mathbb{R}^n$, $F(x) = 0$ with at least one equation that does not depend linearly on some variables, the system is named non-linear. These equations commonly appear in modelling mechanical systems, chemical reactions, optimization problems and many other topics. Because the numerical solution of such non-linear systems often involves solving a related linear system, we discuss them first before proceeding with non-linear systems.

4.1 Iterative Methods

Solving linear equation systems with direct methods is efficient for small systems and gives exact solutions. But usually systems are large, storage is limited and one is only interested in approximated solutions because the linear equation systems might arise for instance from approximated methods, for example from finite difference methods, which already contain discretization errors.
Before we can define iterative methods, we need the following definitions.

Definition 4.1.1 (Fixed point)**.** *An element x of a set $D \subset X$ is called fixed point of a mapping $F : D \to X$, if*

$$F(x) = x. \tag{4.5}$$

Definition 4.1.2. *Let X be a normed vector space. A mapping $F : D \subset X \to X$ is called a contraction, if there exists a q with $0 \leq q < 1$ such that*

$$\| F(x) - F(y) \| \leq q \| x - y \|, \quad \forall x, y \in D. \tag{4.6}$$

Theorem 4.1.3. *A contraction mapping is continuous and has at most one fixed point [29].*

The following theorem is a useful tool that provides information on the existence and uniqueness of fixed points of maps. It is not only being used for deriving convergence conditions for many iterative methods but it also gives insight in their speed of convergence.

Theorem 4.1.4 (Banach Fixed Point Theorem [100])**.** *Let X be a normed vector space with $D \subset X$ and with contraction map $F : D \to D$, then F admits one and only one fixed point $x \in D$ of F and the sequence*

$$x_{n+1} = F(x_n), \qquad \text{for } n = 0, 1, 2, \ldots \tag{4.7}$$

converges for every initial value $x_0 \in D$ towards x.

We are now able to construct an iterative method to gradually approximate the solution of a linear equation system

$$Ax = b \tag{4.8}$$

with $A \in \mathbb{R}^{n \times n}$ and $x, b \in \mathbb{R}^n$.

Definition 4.1.5 (Iterative method)**.** *An iterative method given by the mapping*

$$\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \tag{4.9}$$

is linear, if matrices $M, N \in \mathbb{R}^{n \times n}$ exists such that

$$\phi(x, b) = Mx + Nb. \tag{4.10}$$

A vector $\tilde{x} \in \mathbb{R}^n$ is called fixed point of the iterative method ϕ to $b \in \mathbb{R}^n$, if

$$\tilde{x} = \phi(\tilde{x}, b). \tag{4.11}$$

As of now, we established the definitions for general iterative methods. In order for these methods to give approximated solutions for linear equation systems (4.8), we need to define consistency and convergence to create this link.

Definition 4.1.6. *An iterative method ϕ is consistent to matrix $A \in \mathbb{R}^{n \times n}$, if for all $b \in \mathbb{R}^n$ the solution $A^{-1}b$ is a fixed point of ϕ to b. It is convergent if for all $b \in \mathbb{R}^n$ and for all initial values $x_0 \in \mathbb{R}^n$ a limit $\tilde{x} \in \mathbb{R}^n$*

$$\tilde{x} = \lim_{m \to \infty} x_m = \lim_{m \to \infty} \phi(x_{m-1}, b) \tag{4.12}$$

independent of the initial value x_0 exists.

Consistency is a necessary condition for every iterative method in order to have a reasonable connection between the numerical method and the linear equation system. An equivalent definition to 4.1.6 is that a linear iterative method is consistent to matrix A, iff $M = I - NA$.

We remark that a consistent linear iterative method (4.10) can be also written in the form

$$x_{m+1} = x_m + Nr_m \tag{4.13}$$

with residual $r_m = b - Ax_m$. This equation shows the idea that the current approximation x_m is being updated by some sort of residual correction term with residual $r_m \in \mathbb{R}^n$.
The iterative method (4.13) shows that matrix N is be an approximation of A^{-1} ideally less costly to compute than the inverse matrix of A.
Consistency only gives a necessary condition for the linear iterative method to produce approximated solutions to a linear equation system. We need convergence in order to have a sufficient condition that the solutions given by the iterative method ϕ is, in fact, an approximated solution to the linear equation system (4.8).

Proposition 4.1.7. *A linear iterative method ϕ is convergent if and only if the spectral radius of the iteration matrix M is*

$$\rho(M) < 1. \tag{4.14}$$

The proof can be shown with the Banach Fixed Point theorem.

With this we conclude the main concepts of iterative methods and proceed with a specific class of iterative methods.

4.1.1 Relaxation Methods

Relaxation methods are a class of iterative methods to solve linear equation systems $Ax = b$ based on the partition of matrix $A \in \mathbb{R}^{n\times n}$ with

$$A = B + (A - B), \qquad B \in \mathbb{R}^{n\times n} \tag{4.15}$$

such that with the equivalent linear equation system

$$\begin{array}{rrcl} & Ax & = & b \\ \Leftrightarrow & Bx & = & (B-A)x + b \\ \Rightarrow & x & = & B^{-1}(B-A)x + B^{-1}b \end{array} \tag{4.16}$$

we obtain the linear iterative method

$$x_{m+1} = \phi(x_m, b) = Mx_m + Nb, \qquad \text{for } m = 0, 1, \ldots \tag{4.17}$$

with matrices

$$\begin{aligned} M : &= B^{-1}(B - A), \\ N : &= B^{-1} \end{aligned} \tag{4.18}$$

which is also called relaxation method.
For a splitting of matrix A, such that

$$A = D + L + U, \tag{4.19}$$

where D is the diagonal matrix of A, and L and U are the strictly lower and upper triangular part of A respectively we have the following well known relaxation methods:

- **Jacobi method**: $M := D^{-1}(D - A)$ and $N := D^{-1}$.
- **damped Jacobi method** $M := I - \omega D^{-1}A$ and $N := \omega D^{-1}$.
- **Gauß-Seidel method** $M := -(D + L)^{-1}U$ and $N := (D + L)^{-1}$.
- **SOR method** $M(\omega) := (D+\omega L)^{-1}\left[(1 - \omega)D - \omega U\right]$ and $N := \omega(D+\omega L)^{-1}$.
- **Richardson method** $M := (I - \omega A)$ and $N := \omega I$.

The relaxation methods can be derived by modifying the residual correction term of (4.13) with a weight $\omega \in \mathbb{R}$.

$$x_{m+1} = x_m + \omega N r_m. \tag{4.20}$$

Each of the indicated methods are consistent. However, convergence will be only guaranteed if the spectral radius of the iteration matrix satisfies $\rho(M) < 1$. We refer to [29, 100] for a more detailed explanations and analysis on relaxation methods, their speed of convergence and convergence assertions which infer certain conditions to be fulfilled by the matrix A in order to obtain convergence, such as diagonal dominance and positive-definiteness.
Although relaxation methods can provide approximated solutions to linear equation systems, they are typically slower in convergence than other iterative methods such as Krylov subspace methods. Yet, these methods have their usefulness as preconditioners or as smoothers for multigrid methods we will present in the next sections.

4.1.2 Krylov Subspace Methods

Another kind of iterative methods is called Krylov-subspace methods, belonging to the class of projection methods.
In relaxation methods the approximated solution x_m is determined by a fixed iteration function ϕ with $x_m = \phi\left(x_{m-1}, b\right)$ for a linear equation system $Ax = b$.

These solutions are always in the preimage of $\mathbb{R}^n$ with $x_m \in \mathbb{R}^n$. In projection methods however, the approximated solution x_m are only valid in the $m < n$-dimensional affine-linear subspace $x_0 + \mathcal{K}_m$ such that if $u_1, \ldots, u_m \in \mathbb{R}^n$ is a base of the Krylov subspace $\mathcal{K}_m \subset \mathbb{R}^n$, then

$$x_m := x_0 + \alpha_1 u_1 + \cdots + \alpha_m u_m \tag{4.21}$$

with $\alpha_i \in \mathbb{R}$. The m degrees of freedom are clearly defined by the orthogonality condition

$$(b - Ax_m) \perp \mathcal{L}_m, \tag{4.22}$$

with the m-dimensional subspace $L_m \subset \mathbb{R}^n$. This condition means that the residual r_m is orthogonal to the Krylov-subspace $\mathcal{L}_m$.
It follows that the iterations are defined implicitly by the orthogonality conditions (4.22) and each projection method differs solely on the choice of the Krylov subspaces $\mathcal{K}_m$ and $\mathcal{L}_m$.

Definition 4.1.8 (Krylov-Subspace Method)**.** *A Krylov-subspace method is a projection method for the solution of the linear equation system $Ax = b$ with $A \in \mathbb{R}^{n\times n}$ and $x, b \in \mathbb{R}^n$, with the Krylov subspace*

$$\mathcal{K}_m = \mathcal{K}_m(A, r_0) = span\{r_0, Ar_0, \ldots, A^{m-1}r_0\} \tag{4.23}$$

with residual $r_0 = b - Ax_0$.

For $\mathcal{K}_m = \mathcal{L}_m$ the residual $r_m = b - Ax_m$ is orthogonal to the Krylov subspace $\mathcal{K}_m$ and hence we call the projection method orthogonal with orthogonality condition $r_m \perp \mathcal{K}_m$ (also called Galerkin condition).
For $\mathcal{K}_m \neq \mathcal{L}_m$ the orthogonality condition (4.22) is also called Petrov-Galerkin condition.
Krylov subspace methods are often described through a reformulation of the linear equation system into a minimization problem. In the following we will give a brief derivation of the conjugated gradient method (CG) as such a method.
At this point we consider symmetric positive-definite matrices $A \in \mathbb{R}^{n\times n}$ for the linear equation system $Ax = b$.
We consider the functional

$$F : \mathbb{R}^n \to \mathbb{R} : \quad F(x) := \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \tag{4.24}$$

with scalar product $\langle \cdot, \cdot \rangle$. The functional (4.24) can also be defined as the energy norm $\| \cdot \|_A := \sqrt{\langle Ax, x\rangle}$ of a symmetric positive-definite matrix A.
Looking at the gradient of the functional

$$\nabla F(x) = Ax - b \tag{4.25}$$

we can see that the solution $x^* = A^{-1}b$ of the linear equation system is equivalent to the global minimum of the functional F if the matrix A is symmetric positive-definite.

Theorem 4.1.9. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite and $b \in \mathbb{R}^n$, then for functional F(4.24) the vector $x^* \in \mathbb{R}^n$*

$$x^* = \arg\min_{x\in\mathbb{R}^n} F(x) \tag{4.26}$$

is the global minimum if and only if

$$Ax^* = b. \tag{4.27}$$

Thus solving the linear equation system is equivalent to the minimization of the functional F

$$x^* = \arg\min_{x\in\mathbb{R}^n} F(x). \tag{4.28}$$

We anticipate methods for solving minimization problems which will present in Chapter 5 in more detail.

For now we define another function

$$f_{x,p} : \mathbb{R} \to \mathbb{R}, \quad f_{x,p}(\lambda) := F(x + \lambda p), \tag{4.29}$$

that transform the minimization problem (4.28) into a one-dimensional minimization problem for a given search direction $p_m \in \mathbb{R}^n$ and step length $\lambda_m \in \mathbb{R}$ along which we want to minimize the value of the function $f_{x,p}$.
We therefore create the iterative method

$$x_{m+1} = \arg\min_{x\in x_m+\mathrm{span}\{p_m\}} F(x), \tag{4.30}$$

which is equivalent to

$$\lambda_m = \arg\min_{\lambda\in\mathbb{R}} f_{x_m,p_m}(\lambda), \quad x_{m+1} = x_m + \lambda_m p_m. \tag{4.31}$$

This is a prototypical method for line search methods which will present in more detail in the next chapters.
For the optimal step length λ_{opt} we can make use of the symmetric positive definite property of matrix A.

Theorem 4.1.10. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite and $b, x, p \in \mathbb{R}^n$ with $p \neq 0$ then*

$$\lambda_{opt} = \lambda_{opt}(x, p) := \arg\min_{\lambda\in\mathbb{R}} f_{x,p}(\lambda) = \frac{\langle r, p\rangle}{\langle Ap, p\rangle} \tag{4.32}$$

with residual $r := b - Ax$ [100].

This keeps the question open which sequence of search directions p_m can be used.
An intuitive choice is using the steepest decent of the functional defined as

$$\nabla F(x) = Ax - b = -r \tag{4.33}$$

and as such obtain the search direction

$$p_m := \begin{cases} \frac{r_m}{\|r_m\|}, & \text{for } r_m \neq 0, \\ 0, & \text{for } r_m = 0. \end{cases} \tag{4.34}$$

Using a search direction (4.34) brings us to the **gradient descent method**, an iterative optimization method for finding local minima.
In the context of Krylov subspace method, the gradient descent method is an orthogonal projection method with $K_m = L_m = \operatorname{span}\{r_{m-1}\}$.
In order to find better search directions, we need to define what an optimal search direction is.

Definition 4.1.11. *Let $F : \mathbb{R} \to \mathbb{R}$, then $x \in \mathbb{R}^n$ is called*

- *i) optimal in regards to the search direction $p \in \mathbb{R}^n$ if*

$$F(x) \leq F(x + \lambda p), \quad \text{for all } \lambda \in \mathbb{R}. \tag{4.35}$$

- *ii) optimal in regards to a subspace $U \subset \mathbb{R}^n$, if*

$$F(x) \leq F(x + \zeta), \quad \text{for all } \zeta \in U. \tag{4.36}$$

For a functional F(4.24), $x \in \mathbb{R}^n$ is only optimal in regards to $U \subset \mathbb{R}^n$, iff

$$r = (b - Ax) \perp U. \tag{4.37}$$

The iteration x_m of the gradient descent method are only optimal in regards to the single search direction $r_{m-1} = b - Ax_{m-1}$.
A desirable extension would be that x_m is optimal in regards to the whole subspace $U = \{r_0, \ldots, r_{m-1}\}$ which would mean that for linear independent residuals, the exact solution can be found in the n-th iteration.
In other words we want that the next iteration x_{m+1} is not only optimal for the search direction p_m but also for all previous search directions $p_{m-1}, \ldots, p_0$.
How to choose the search direction such that the optimality is inherited is shown in the following theorem.

Theorem 4.1.12. *Let the vector $x \in \mathbb{R}^n$ be optimal in regards to the subspace $U = \operatorname{span}\{p_0, \ldots, p_{m-1}\} \subset \mathbb{R}^n$ for function F in (4.24), then $\tilde{x} = x + \zeta$ is optimal in regards to U if, and only if,*

$$A\zeta \perp U. \tag{4.38}$$

Using theorem 4.1.12, we need to choose p_m such that

$$\begin{aligned} Ap_m &\perp U_m = \text{span}\{p_0, \dots p_{m-1}\} \\ \Leftrightarrow \quad Ap_m &\perp p_j, \text{ for } j = 0, \dots, m-1, \end{aligned} \tag{4.39}$$

so that iteration x_{m+1} is optimal in regards to U_m.
This condition can be summarized under the term **A-conjugated**, which means that vectors $p_i, p_j, \dots \in \mathbb{R}^n$ are pairwise A-conjuagted with

$$\langle p_i, p_j \rangle_A := \langle Ap_i, p_r \rangle_2 = 0, \quad \text{for all } i, j \in \{0, \dots, m\} \text{ and } i \neq j. \tag{4.40}$$

If we have pairwise A-conjugated search directions $p_0, \dots, p_m$ and the iteration x_m is optimal in regards to $U_m = span\{p_0, \dots, p_{m-1}\}$, then the next iteration with

$$x_{m+1} = x_m + \lambda p_m \tag{4.41}$$

is optimal in regards to U_{m+1} if

$$\lambda = \frac{\langle r_m, p_m \rangle_2}{\langle Ap_m, p_m \rangle_2}, \tag{4.42}$$

which can be shown by using (4.37).
With pairwise A-conjugated search directions $p_0, \dots, p_m \in \mathbb{R}^n$ and λ_m (4.42) we can guarantee iterations x_{m+1} which are optimal towards the subspace U_{m+1}.
As for the appropriate choice of search directions, we extend the gradient descent method approach by choosing

$$p_0 = r_0 \tag{4.43}$$

and iteratively calculating

$$p_m = r_m + \sum_{j=0}^{m-1} \alpha_j p_j \tag{4.44}$$

with coefficients $\alpha_j \in \mathbb{R}$ which can be determined by requiring the pairwise A-conjugation of the search directions $p_0, \dots, p_m$:

$$0 = \langle Ap_m, p_i \rangle_2 = \langle Ar_m, p_i \rangle_2 + \sum_{j=0}^{m-1} \alpha_j \langle Ap_j, p_i \rangle_2, \tag{4.45}$$

which leads to the coefficients

$$\alpha_i = -\frac{\langle Ar_m, p_i \rangle_2}{\langle Ap_i, p_i \rangle}. \tag{4.46}$$

The iterative method

$$x_{m+1} = x_m + \lambda_m p_m \tag{4.47}$$

with search direction (4.43), (4.44), (4.46) and search step length λ_m (4.42) is the conjugated gradient method.
Because the CG-method is defined as

$$x_m \in x_0 + \text{span}\{p_0, \dots p_{m-1}\} = \text{span}\{r_0, \dots, A^{m-1} r_0\} = K_m \tag{4.48}$$

and is by definition optimal in regards to the Krylov subspace K_m, which means

$$r_m \perp K_m = L_m \tag{4.49}$$

the method is an orthogonal projection method.
Although the requirement to use the CG-method on linear equation system $Ax = b$ is that the matrix $A \in \mathbb{R}^{n \times n}$ is symmetric positive-definite, CG-methods can also be used on the normal equation $A^T Ax = A^T b$ since the matrix $A^t A$ is symmetric positive-definite for regular A.
However this squares the condition number which leads to slower convergence rates.

4.1.3 Multi Grid Methods

This section provides some numerical background about multi grid methods. The main references for this chapter [28, 63, 138] from which most of the content are taken and we refer to them for a more detailed presentation about convergence.
Just as relaxation methods, multi grid methods are iterative methods, which are also often used to solve linear equation systems arising from discretizations of partial differential equations but they can be applied to non-linear equation systems [22, 23, 69], too.
Multi grid methods have the following properties:

- The convergence rate is independent of the mesh grid size and the system can be solved with only a small number of iterations.
- The distribution of the error is divided into two parts which are smooth and coarse error parts. The coarse error parts are being dealt with by a smoother, which is a method that eliminates the coarse error parts, whereas the smooth error parts are being dealt with by a coarse-grid correction.

Using these two parts recursively is the core of a multigrid method.

For the following, we will consider the one-dimensional Poisson equation

$$\begin{aligned} -\partial_{xx} u(x) &= f(x), \quad \text{for } x \in \Omega, \\ u(x) &= 0, \qquad \text{for } x \in \delta\Omega \end{aligned} \tag{4.50}$$

with $u \in \mathcal{C}^2(\Omega, \mathbb{R}) \cap \mathcal{C}(\overline{\Omega}, \mathbb{R})$ and given $\Omega = (0,1)$, $\delta\Omega = \{0,1\}$ and $f \in \mathcal{C}(\Omega; \mathbb{R})$. The eigenfunctions of the homogenous boundary value problem (4.50) are

$$u(x) = c\sin(j\pi x) \quad \text{with } j \in \mathbb{N} \text{ and } c \in \mathbb{R}. \tag{4.51}$$

We now define the mesh grid refinement $\triangle x_l$ with $\triangle x_0 = \frac{1}{2}$ and $\triangle x_l = \frac{\triangle x_0}{2^l} = 2^{-(l+1)}$ and the grid refinement $\Omega_l := \left\{ j \triangle x_l | j = 1, \ldots, 2^{l+1} - 1 \right\}$ for $l = 0, \ldots$. When we discretize the Poisson equation (4.50) by using a central difference method in space, we obtain the following linear equation system

$$A_l u^l = f^l \tag{4.52}$$

with

$$A_l = \frac{1}{\triangle x_l} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{(N_l)\times(N_l)}, \tag{4.53}$$

$$u^l = \begin{pmatrix} u_1^l \\ \vdots \\ u_{N_l}^l \end{pmatrix} \text{ and } f^l = \begin{pmatrix} f_1^l \\ \vdots \\ f_{N_l}^l \end{pmatrix}.$$

For the eigenvectors of matrix A_l we have

$$v^{l,j} = \sqrt{2\triangle x} \begin{pmatrix} \sin(j\pi \triangle x_l) \\ \vdots \\ \sin(j\pi N_l \triangle x_l) \end{pmatrix}, \text{ for } j = 1, \ldots, N_l, \tag{4.54}$$

which are clearly the discretized version of the eigenfunction (4.51).
Fourier modes are more oscillatory for larger j and thus less smooth. We can henceforth distinguish between low-and high frequency as follows.

Definition 4.1.13. *The eigenvectors $v^{l,j}$ can be classified into*

- *low frequency component $v^{l,j}$ with $1 \leq j < \frac{N_l}{2}$.*
- *high frequency component $v^{l,j}$ with $\frac{N_l}{2} \leq j \leq N_l$.*

Theorem 4.1.14. *[29] The vectors $v^{l,j}$ defined in (4.54) are eigenvectors of matrix A_l with the associated eigenvalues $\lambda^{l,j} = \frac{4}{\triangle x_l^2}\sin^2\left(\frac{j\pi\triangle x_l}{2}\right)$. These eigenvectors form an orthonormal basis of $\mathbb{R}^{N_l}$.*

If we apply the linear equation system (4.52) to the damped Jacobi relaxation method with $\omega = \frac{1}{2}\tilde{\omega}$ we obtain

$$\begin{aligned} u_{m+1}^l &= u_m^l + \tilde{\omega} D_l^{-1}\left(f^l - A_l u_m^l\right) \\ &= u_m^l + \omega \triangle x_l^2\left(f^l - A_l u_m^l\right) \\ &= \underbrace{\left(I - \omega\triangle x_l^2 A_l\right)}_{=:M_l(\omega)} u_m^l + \underbrace{\omega\triangle x_l^2 I}_{=:N_l(\omega)} f^l, \quad \text{for } m = 0, 1 \ldots . \end{aligned} \tag{4.55}$$

It can be easily seen that the iteration matrix $M_l(\omega)$ has the same eigenvectors as the matrix A_l.

Theorem 4.1.15. *The matrix $M_l(\omega) := I - \omega\triangle x_l^2 A_l$ has the same eigenvectors as matrix A with eigenvalues*

$$\lambda^{l,j}(\omega) = 1 - 4\omega\sin^2\left(\frac{j\pi\triangle x_l}{2}\right), \quad \textit{for } j = 1, \ldots, N_l. \tag{4.56}$$

By using this theorem we can show how the error components develop during iterations.
Let u_*^l be the exact solution of (4.52) and u_{m+1}^l the $m+1$ iteration of the damped Jacobi method (4.55).
Then the error e_{m+1} is

$$e_{m+1} = u_*^l - u_{m+1}^l = M_l(\omega)e_m = M_l\left(\sum_{j=1}^{N_l}\alpha_m^{l,j}v^{l,j}\right) = \sum_{j=1}^{N_l}\alpha_m^{l,j}\lambda^{j,l}v^{l,j} = \sum_{j=1}^{N_l}\alpha_0^{l,j}\left(\lambda^{l,j}\right)^m v^{l,j}, \tag{4.57}$$

which is the form of iterative methods (4.10).
As we can read from the iterative error (4.57), each mode $v^{l,j}$ converges with different speeds towards zero, dictated by the corresponding eigenvalue $\lambda^{l,j}$.
The choice of ω affects the eigenvalue distribution and with $\omega = \frac{1}{4}$ we have an optimal dampening effect such that high frequency error components converge towards zero the fastest, leaving a smooth error since the low frequency components have the slowest rate of convergence.
Because of this property of "smoothing" the error, we call the damped Jacobi method also smoother.
Also worth mentioning is that for a finer mesh grid (smaller $\triangle x_l$) the eigenvalues are approaching 1 which means even less dampening on the smooth and coarse modes. A finer discretization means higher computational cost

but also slower convergence.

In order to decrease also the low frequency components of the error, we use the coarse-grid correction. The idea is that a smooth error with only low frequency components can be represented well on a coarser mesh grid and is more oscillatory on a coarser grid. Oscillatory errors on the other hand get aliased into smooth error on a coarser mesh. Hence we need to use a smoother first to eliminate high order frequency components before and then eliminate the low order frequency components which are high order frequencies on the coarse grid. By repeating this on coarser grids, we can eliminate both low and high frequency error components.

Definition 4.1.16. *A map*

$$F : \mathbb{R}^{N_l} \to \mathbb{R}^{N_{l-1}} \tag{4.58}$$

is called a restriction operator from finer mesh grid Ω^l to the coarser mesh grid Ω^{l-1}, if it is linear and surjective.

There are many ways for the restriction such that

$$u^{l-1} = R_l^{l-1} u^l \tag{4.59}$$

with restriction matrix $R_l^{l-1} \in \mathbb{R}^{N_{l-1} \times N_l}$ but in this context just show the trivial restriction matrix

$$R_l^{l-1} = \begin{pmatrix} 0 & 1 & 0 & & & & & & & \\ & & 0 & 1 & 0 & & & & & \\ & & & & 0 & 1 & 0 & & & \\ & & & & & & & \ddots & & \\ & & & & & & & & 0 & 1 & 0 \end{pmatrix} \tag{4.60}$$

and the linear restriction matrix

$$R_l^{l-1} = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 & & & & & & & \\ & & 1 & 2 & 1 & & & & & \\ & & & & 1 & 2 & 1 & & & \\ & & & & & & & \ddots & & \\ & & & & & & & & 1 & 2 & 1 \end{pmatrix}. \tag{4.61}$$

The benefit of using the linear restriction is that the trivial restriction neglect the values at the inbetween grid points such that information gets lost.
The opposite of the restriction is the prolongation where we map the solution or error from a coarse grid Ω^{l-1} to a finer grid Ω^l.

Definition 4.1.17. *A map*

$$G : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l} \tag{4.62}$$

is called a prolongation from Ω^{l-1} to Ω^l, if it is linear and injective.

An example for a prolongation matrix for

$$u^l = P_{l-1}^l u_{l-1} \tag{4.63}$$

is a linear interpolation which give us the linear prolongation matrix

$$P_{l-1}^l = \frac{1}{2} \begin{pmatrix} 1 & & & & \\ 2 & & & & \\ 1 & 1 & & & \\ & 2 & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & 2 & \\ & & & 1 & \end{pmatrix} \in \mathbb{R}^{N_l \times N_{l-1}}. \tag{4.64}$$

For the sake of simplicity, we focus on the description of a two-grid method which can be easily expanded by recursion to a multigrid method.
In the following, let

$$A_l e^l = A_l \left(u_m^l - u_*^l \right) = r_l \tag{4.65}$$

be the defect equation. If we assume the error e^l to be smooth, we can compute an approximation to the error by solving the coarser defect equation

$$A_{l-1} e^{l-1} = r_{l-1} \tag{4.66}$$

and the transform it back to the finer grid.

Coarse-grid correction

After the high frequency components have been eliminated through smoothing, the coarse-grid correction is applied in the following way:

1. Compute the residual: $r_l = f_l - A_l u_l$.
2. Restrict the residual to the coarse grid: $r_{l-1} = R_l^{l-1} r_l$.
3. Compute the error on the coarse grid by solving the system: $A_{l-1} e_{l-1} = r_{l-1}$.
4. Prolongate the error to the fine grid: $e_l = P_{l-1}^l e_{l-1}$.
5. Update the approximation $u_l = u_l + e_l$.

Definition 4.1.18. *Let u_m^l be an approximated solution of $A_l u^l = f^l$, then the method with*

$$u_m^{l,\mathsf{new}} = \phi_l^{CGC} \left(u_m^l, f^l \right) \tag{4.67}$$

is called coarse grid correction.

Lemma 4.1.19. *[29] The coarse grid correction ϕ_l^{CGC} is a linear consistent iterative method with*

$$M_l^{CGC} = I - P_{l-1}^l A_{l-1}^{-1} R_l^{l-1} A_l \tag{4.68}$$

and

$$N_l^{CGC} = P_{l-1}^l A_{l-1}^{-1} R_l^{l-1}. \tag{4.69}$$

Lemma 4.1.20. *[29] The coarse grid correction ϕ_l^{CGC} is not convergent.*

Two-grid method

With all the components defined, we can set up a two-grid method.
The two-grid method consists of a fixed number of pre-smoothing steps ν_1 followed by a coarse grid correction, which then ends with additionally post-smoothing steps ν_2.

Definition 4.1.21. *Let $\nu_1, \nu_2 \in \mathbb{N}_0$ with $\nu_1 + \nu_2 \geq 1$. Let S_l be a linear iterative scheme for the solution of $A_l u_l = f_l$ and ϕ_l^{CGC} be the coarse grid correction. Then the method defined by*

$$\phi_k^{TGM(\nu_1,\nu_2)} = S_l^{\nu_2} \circ \phi_k^{CGC} \circ S_l^{\nu_1} \tag{4.70}$$

is called a two grid method with ν_1 presmoothing and ν_2 postsmoothing steps.

Although the coarse grid correction itself is not a convergent method, the combination with the smoothing makes it convergent.

Theorem 4.1.22. *[29] Considering the two grid method (4.70), if the smoothing operator S_l is consistent, i.e $S_l(u_l^*) = u_l^*$ and convergent, then the two grid method $\phi_l^{TGM(\nu_1,\nu_2)}$ is consistent and convergent.*

The two-grid method algorithm is depicted in Algorithm 4.1.

Multigrid methods

Although two-grid methods are quite efficient, they are impracticable for larger linear equation systems because the defect equation

$$A_{l-1} e^{l-1} = r^{l-1} \tag{4.71}$$

needs be calculated. However the recursive application of the two-grid method to the defect equation does not differ from applying the two grid method to the original problem. By introducing more grids and recursively applying the two-grid method until a direct solution is possible, leads to the multigrid methods. Depending on the number of recursive two-grid steps $\gamma > 0$ we can differentiate between different types of multigrid methods [28]. In practice, only $\gamma = 1$ (V-cycle) and $\gamma = 2$ (W-Cycle) are used. The names arise from how the two-grid steps move through the hierarchy of grids.
Algorithm 4.2 depicts the multigrid method.

Algorithm 4.1
two-grid method $\phi_l^{TGM(\nu_1,\nu_2)}$

procedure Two-grid method
  for $i = 1, \ldots, \nu_1$ **do**
    $u^l := S_l\left(u^l, f^l\right)$.
  end for
  $r^{l-1} := R_l^{l-1}\left(A_l u^l - f^l\right)$
  $e^{l-1} := A_{l-1}^{-1} r^{l-1}$
  $u^l := u^l - P_{l-1}^l e^{l-1}$
  for $i = 1, \ldots, \nu_2$ **do**
    $u^l := S_l\left(u^l, f^l\right)$.
  end for
end procedure

Algorithm 4.2
multigrid method $\phi_l^{MGM(\nu_1,\nu_2)}$

procedure Multigrid method $\phi_l^{MGM}(\nu_1, \nu_2)$
  if $l = 0$ **then**
    $u^0 := A_0^{-1} f^0$
  else
    for $i = 1, \ldots, \nu_1$ **do**
      $u^l := S_l\left(u^l, f^l\right)$.
    end for
    $r^{l-1} := R_l^{l-1}\left(A_l u^l\right)$
    $e_0^{l-1} := 0$
    for $i = 1, \ldots, \gamma$ **do**
      $e_i^{l-1} := \phi_{l-1}^{MGM(\nu_1,\nu_2)}\left(e_{i-1}^{l-1}, r^{l-1}\right)$
    end for
    $u^l := u^l - P_{l-1}^l e_\gamma^{l-1}$
    for $i = 1, \ldots, \nu_2$ **do**
      $u^l := S_l\left(u^l, f^l\right)$.
    end for
  end if
end procedure

Full multigrid method

All iterative methods such as the relaxation methods and the multigrid methods as previously presented require an initial guess u^l on the finest grid. The idea behind the full multigrid method (FMGM) is the application of the coarse grids to improve the initial guess u^l.
The main steps of the method are:

a) Solve $A_0 u^0 = f^0$ on Ω_0 exactly.

b) Prolongation of the solution to the next finer grid and smooth the solution.

c) Repeat (b) until an initial guess u^l on the finest grid Ω_l is calculated.

d) Apply the multigrid method with the new initial guess.

Full multigrid methods are of special interest to us because when extending the idea of multigrid methods to non-linear problems, using a good initial guess, which is already close to the solution, helps to reduce the number of iteration steps, thus reducing computational costs. In regards to multigrid methods as optimization methods to find minima, a good initial guess also reduces the risk of getting trapped in a local minimum, which can be far away from the desired minimum.

4.2 Solving Non-Linear Systems

Finding a solution to a non-linear system through a direct method as for linear systems is often impossible and thus iterative methods must be used as well [29].
The general non-linear equation problem is

$$F(x) = \begin{pmatrix} F_1(x_1, \ldots, x_n) \\ \vdots \\ F_n(x_1, \ldots, x_n) \end{pmatrix} = 0, \tag{4.72}$$

for a sufficiently differentiable function $F : \mathbb{R}^n \to \mathbb{R}^n$.
Any fixed point equation

$$F(x) = x$$

can be transformed into a root finding problem with

$$G(x) := F(x) - x.$$

If for (4.72) we have $F(x) = Ax - b$ with square matrix $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, then we obtain a linear equation system which can be solved with the methods presented previously in this chapter.
Solving non-linear equations analytically, i.e with direct methods is often impossible, so iterative methods must be used.
For this reason we recall some background on iterative methods of the form

$$x^{k+1} = \phi\left(x^k\right) \tag{4.73}$$

with iteration function $\phi : \mathbb{R}^n \to \mathbb{R}^n$ with iteration sequence $\left(x^k\right)_{k \in \mathbb{N}_0}$.
One can already use the fixed point equation $F(x) = x$ and apply the intuitive

iterative method $x^{k+1} = F(x^k)$.
Generally there exist many different iterative methods, each converging at varying speeds towards an appropriate approximated solution.
To define the speed of convergence we use again Banach Fixed Point theorem 4.1.4:

Definition 4.2.1. *[113] The iteration sequence $\left(x^k\right)$ converges at least with order $p \geq 1$ towards x^* if there exist a constant $K \geq 0$ (for $p = 1$ $K < 1$) and an index k_0 such that for all $k \geq k_0$ the inequation*

$$\| x^{k+1} - x^* \| \leq K \| x^k - x^* \|^p \tag{4.74}$$

holds. For $p = 1$ we have linear convergence and for $p = 2$ quadratic convergence.

Based on the definition of iterative methods we can reformulate theorem 4.2.1 and additionally define local convergence and convergence domain.

Definition 4.2.2. *Let $\phi : \mathbb{R}^n \to \mathbb{R}^n$ be an iteration function for the fixed point x^*, then there is a neighbourhood $U(x^*) \subset R^n$ and a $p \geq 1$ with a constant $K \geq 0$ (with $K < 1$ for $p = 1$), such that for all $x \in U(x^*)$*

$$\| \phi(x) - x^* \| \leq K \| x - x^* \|^p \tag{4.75}$$

hold. Furthermore there exists a neighbourhood $V(x^*) \subset U(x^*)$ such that the iterate $x^{k+1} = \phi\left(x^k\right)$ converges for every initial guess $x^0 \in V(x^*)$ with convergence speed p towards x^*. We call such iteration local convergent with convergence domain $V(x^*)$.

With these tools at hand we can now define the Newton methods which are widely used for solving non-linear equation systems [29, 41, 113].

4.2.1 Newton's Method

The main idea of the Newton's method to find the root of the non-linear equation system $F(x) = 0$ is to linearize F at the current iterate x^k. Applying the first order Taylor expansion on F about the point x^k we obtain

$$F(x) \approx T_F^1(x^k, x) = F(x^k) + J_F(x^k)\left(x - x^k\right) = 0 \tag{4.76}$$

with Jacobian matrix $J_F\left(x^k\right) \in \mathbb{R}^{n \times n}$.
Now instead of solving $F(x) = 0$ we want to solve the linearized version $T_F^1(x^k, x) = 0$. Defining the next iterate as $x^{k+1} = x$ gives us Newton's method

$$x^{k+1} = x^k - J_F\left(x^k\right)^{-1} F\left(x^k\right). \tag{4.77}$$

For the termination of the iteration, there are two reasonable choices: $\frac{\|\triangle x^{k+1}\|}{\|x^0\|}$ or $\frac{\|F(x^k)\|}{\|F(x^0)\|}$.
We can define the algorithm of Newton's method for non-linear equation with a termination condition $\tau_k > 0$ in the following way:

- Choose an initial guess $x^0 \in \mathbb{R}^n$.
- Do
  - Solve the linear equation system

$$J_F\left(x^k\right) \triangle x^{k+1} = -F\left(x^k\right) \tag{4.78}$$

    for $\triangle x^{k+1} = x^{k+1} - x^k$.
  - Update $x^{k+1} = x^k + \triangle x^{k+1}$
- While $\frac{\|\triangle x^{k+1}\|}{\|x^0\|} < \tau_k$ or $\frac{\|F(x^k)\|}{\|F(x^0)\|} < \tau_k$.

With Newton's method we have transformed the non-linear equation system into an iterative method where an iteration sequence of linear equation systems must be solved. This can be done with direct methods but also with iterative methods as we have already presented in Chapter 4. Newton's methods in which the linear equation systems (4.78) are solved only approximately and in some unspecified manners are called inexact Newton's method [40].
With the purpose of analyzing the convergence properties of the Newton's method (4.77) we can make use of theorem 4.2.1.
From this it is possible to show that under certain assumptions on F, an initial guess x^0 that is sufficiently close to a root x^* Newton's method will converge at a linear speed of convergence.
Under some additional conditions also superlinear convergence speed up to quadratic convergence speed ($1 < p \leq 2$) can be guaranteed. We refer to [40, 41] where are more detailed discussion about convergence theorems can be found.

The main difficulties in iterative methods for solving non-linear equation systems are that , firstly, it can be quite difficult to foresee to which solution x^* the iterative method converges to for a given initial guess x^0. Secondly, the results about the speed of convergence do not indicate how close the initial guess x^0 has to be to the solution x^* to guarantee convergence.
Some answers to this equation can be given by the Kantorovich's theorems [29].
Lastly, another issue concerns the computation of each iteration x^{k+1} which involves solving the linear equation system $J_F(x^k)\triangle x^{k+1} = -F(x^k)$.
For complicated functions, the derivation of F, i.e the derivation of the Jacobi

matrix can be challenging, computational demanding or even analytically impossible.
For these reasons the idea is to not use the exact Jacobian matrix $J_F(x^k)$ evaluated at x^k but approximations which are good enough to guarantee convergence of Newton's method.
These methods are called **Quasi-Newton Methods**.
There is a variety of Quasi-Newton methods, each differing for the chosen approximation of the Jacobi matrix $J_F(x^k)$.
Some algorithms are using a fixed Jacobian $J_F^(x^0)$ for the whole iteration process. Another idea is Broyden's method [29], which extends the idea to calculate only $J_F(x^0)$ but then to update it by a sequence of matrices $B_1, B_2, \ldots$ with the formula

$$B_{k+1} = B_k + \frac{\left(F(x^{k+1}) - F(x^k) - B_k\left(x^{k+1} - x^k\right)\right)}{\left(x^{k+1} - x^k\right)^T \left(x^{k+1} - x^k\right)} \left(x^{k+1} - x^k\right)^T . \tag{4.79}$$

It can be shown that under certain assumptions, that Broyden's method also converges superlinearly [41].

4.2.2 Non-Linear Multigrid Method

In Section 4.1.3 we have presented with multigrid methods, an iterative method to solve linear equation system by recursively solving the residual equation on different grids, using relaxation method to dampen the high frequency error components and apply a coarse grid correction in order to decrease the low frequency error components as well.
This method works because the residual equation (4.65) is linear.
An application of multigrid methods for non-linear equation systems would greatly benefit from the advantages given by the nature of multi grid methods such as robustness and accelerated convergence.
There are two main approaches to apply the multigrid methods to non-linear equation systems.
The first is to use the linear multigrid method to solve the linear equation system that arises from non-linear optimization methods, such as Newton's method, which is called Newton-Multigrid method (Newton-MG) [79, 138].
The other approach is to apply the idea of multigrid directly to the non-linear equation system. This approach is also called Full Approximation Scheme (FAS) [23, 69]. Newton-MG methods have already been used in a wide area of elliptic and parabolic non-linear problems [79] with only little theory existing about convergence of these kind of methods (see [138]).
We refer to the indicated literature for a more detailed investigation of Newton-MG method and the recent paper [22] where a comparison between FAS and Newton-MG is made.

We consider the discrete non-linear equation system

$$F_l(x_l) = f_l \tag{4.80}$$

with non-linear operator F_l and exact solution vector x_l defined on different discretization grids Ω_l. Let y_l be the approximation of the fine-grid equation

$$F_l(x_l) = f_l. \tag{4.81}$$

We also define the error as $e_l := x_l - y_l$ with residual $r_l = f_l - F_l(y_l)$.
Since F is non-linear, we cannot use the residual equation since $F(e) \neq r$ and use instead

$$F_l(x_l) - F_l(y_l) = r_l \Leftrightarrow F_l(y_l + e_l) - F_l(y_l) = r_l. \tag{4.82}$$

The coarse-grid residual equation is then defined as

$$F_{2l}(y_{2l} + e_{2l}) - F_{2l}(y_{2l}) \quad = \quad r_{2l}. \tag{4.83}$$

By applying the restriction operator $I_l^{2l} : \Omega_l \to \Omega_{2l}$, as in the linear multigrid method (4.59), on the fine-grid residual r_l and approximated solution y_l to the coarse-grid Ω_{2l} we obtain

$$\begin{aligned} r_{2l} &= I_l^{2l} r_l = I_l^{2l}(f_l - F_l(y_l)), \\ y_{2l} &= I_l^{2l} y_l. \end{aligned} \tag{4.84}$$

Substituting (4.84) into the coarse-grid residual equation (4.83) leads to

$$F_{2l}(\underbrace{I_l^{2l} y_l + e_{2l}}_{=x_{2l}}) = \underbrace{F_{2l}(I_l^{2l} y_l) + I_l^{2l}(f_l - F_l(y_l))}_{=f_{2l}}. \tag{4.85}$$

After solving (4.85) for x_{2l} we get the coarse-grid error

$$e_{2l} = x_{2l} - I_l^{2l} y_l, \tag{4.86}$$

which then can be prolongated with the prolongation operator $I_{2l}^l : \Omega_{2l} \to \Omega_l$ to correct the fine-grid approximation y_l

$$y_{l,new} = y_l + I_{2l}^l e_{2l}. \tag{4.87}$$

This describes the two grid Full Approximation Scheme (FAS).
If we apply (4.87) to the non-linear function F_l

$$F_l(y_{l,new}) = F_l(y_l + I_{2l}^l e_{2l}), \tag{4.88}$$

we can already assume the idea of using FAS as a non-linear optimization method such that

$$F_l(y_{l,new}) = F_l(y_l + I_{2l}^l e_{2l}) \leq F_l(y_l), \tag{4.89}$$

which assumes a monotone decrease of the function F_l (objective function in the optimization context).
Sufficient conditions for convergence and usage of FAS as a non-linear optimization method can be found in [20, 112]. We want to remark that as for the linear multigrid methods, a relaxation operator (i.e. smoother) S_l for solving the non-linear system in (4.85) is required to ensure that the error is smooth before prolonging and restricting it to the finer grids (resp. coarser grids).
There is a wide variety of possible non-linear smoothers such as regularized Newton-method, Levenberg-Marquardt method, non-linear Gauß-Seidel method [111, 132]. In this work for all the numerical simulations the regularized Newton-methods as smoother S_l have been used.
As with the linear equivalent, it is only necessary to apply the smoothing a couple of times to smooth the error sufficiently enough to apply FAS. We depicted the algorithm of FAS in Algorithm 4.0, where the two-grid FAS is recursively applied in order to solve the non-linear equation (4.85).
Also here V- and W-cycles can be used as in linear multigrid methods.
As for the choice of smoother, it heavily depends on the non-linear equation that is being solved. We refer to [20] where different non-linear smoothers are being used for different non-linear problems.

Algorithm 4.1 Full Approximation Scheme in pseudocode

1: **procedure** FAS(l, y_l^j, f_l, ν)
2: Set $y_l^{'} = S_l^{\nu} y_l^j$
3: Set $y_{2l}^{'} = I_l^{2l} y_l^{'}$
4: Set $f_{2l} = F_{2l}(I_l^{2l} y_l) + I_l^{2l}(f_l - F_l(y_l^{'}))$
5: **if** $l = 2$ **then** Solve $F_{2l}(x_{2l}) = f_{2l}$
6: **else**Set $x_{2l} = FAS(2l, y_{2l}^{'}, f_{2l}, \nu)$
7: **end if**
8: **return** $y_l^{j+1} = S_l^{\nu}(y_l + I_{2l}^l e^{2l})$
9: **end procedure**

We already want to pre-empt that although FAS can be used to solve non-linear equation system quite efficiently, it can be also used as a minimization method for non-linear optimization problems. Indeed, under appropriate assumptions such that the coarse grid correction provides a descent direction and combining this with a minimization smoothing method such as regularized Newton-method with step size as a line search methods to ensure Wolfe conditions (see next Chapter 5, global convergence can be obtained. We refer again to Borzi [20] for a deeper analysis.

Chapter 5

Optimization Techniques

Optimization problems arise in many different fields from physics, biology to computer science, engineering and financials.
In its core, an optimization problem is the problem of finding the best solution among a set of possible solutions and mostly is formulated as minimizing/maximizing a function.
These functions are objectives, specific to the context of the problem and depend on certain characteristics such as energy in physical systems, profit and time in financial and economic systems, and weather forecast.
Each objective function has characteristics which are parameters and the goal of any optimization problem is to find values of the parameters that optimize the objective functions, i.e minimize/maximize them.
When these parameters are restricted in some way, the problem is called constrained optimization problem, otherwise unconstrained.
The goal of finding the ideal parameters consists constructing first of all an appropriate objective function for the problem. After such an objective function is formulated, an optimization method must be applied to find the solution. There exist a variety of optimization methods and the best choice depends on nature of the problem.
Once a solution is found, it is necessary to be able to determine whether a minimum/maximum is found. So called optimality conditions can be used for this.

The main references are [4, 111, 132] from which most of the contents are taken.
Mathematically we can formulate the minimization problem (the transformation to a maximization problem is straightforward) in the following manner:

Definition 5.0.1 (Minimization problem)**.** *Let $f : D \to \mathbb{R}$ be the objective function with subset $D \subset \mathbb{R}^n$ with vector $x \in \mathbb{R}^n$ as the parameters. Let*

the vectorial function $c : D \to \mathbb{R}^c$ be the constraints that are placed on the parameters x. Then the general minimization problem is defined as

$$\min_{x \in \mathbb{R}^n} f(x) \quad \textit{subject to} \begin{cases} c_i(x) = 0, & i \in \mathcal{E}, \\ c_j(x) \geq 0, & j \in \mathcal{I} \end{cases} \tag{5.1}$$

with $\mathcal{I}, \mathcal{E}$ being the sets of indices for inequality (resp. equation) constraints.

For a more compact notation we define the **feasible set**

$$\Omega := \{x \in \mathbb{R}^n | c_i(x) = 0, i \in \mathcal{E}, c_j(x) \geq 0, j \in \mathcal{I}\},$$

such that (5.1) can be rewritten as

$$\min_{x \in \Omega} f(x). \tag{5.2}$$

The minimization problem (5.1) can be classified into many categories according to the type and structure of objective function f and constrained functions c_i (linear, convex, non-linear, discrete) and so on.

In our work we devote ourselves with the parameter estimation of partial differential equations, we focus on continuous non-linear objective functions and categorize optimization problems into the two categories: unconstrained and constrained optimization problems.
Constrained optimization problems arise from problems where restrictions in form of equations and inequalities $c_i(x)$ have been put on parameters. This depends on the formulation of the model. Often constrained optimization problems can be transformed into unconstrained optimization problems and be solved as such, as it is commonly used in the Lagrangian Multiplier method. However, search steps taken by unconstrained optimization methods may be not acceptable for the constrained optimization problem, which often leads to a lack of convergence. This is referred to as the Maratos effect [136].
Another distinction about the optimization method must be made between global and local optimization methods for which we first need to define two types of minima.

Definition 5.0.2 (Local minima)**.** *Let $f : \mathbb{R}^n \supset D \to \mathbb{R}$ be a real valued function. Then $x^* \in \mathbb{R}^n$ is called local minimum of f if there is an $\epsilon > 0$ such that*

$$f(x^*) \leq f(x), \quad \textit{for all } x \in B(x^*, \epsilon) \tag{5.3}$$

with open ball $B(z, \epsilon) := \{x \in \mathbb{R}^n : \| x - z \| < \epsilon\}$.

Definition 5.0.3 (Global minima)**.** *Let $f : \mathbb{R}^n \supset D \to \mathbb{R}$ be a real valued function. Then $x^* \in \mathbb{R}^n$ is called global minimum of f if*

$$f(x^*) \leq f(x), \quad \text{for all } x \in D. \tag{5.4}$$

Every global minima is therefore also a local minima.

As expected, global optimization methods aim to find global minima. Well known global optimizers are simulated annealing, a generic probabilistic metaheuristic method, evolutionary algorithms and swarm-based optimization algorithms such as particle swarm and ant colony. All these methods are using very different approaches to find the global minimum of an objective function but they all have in common that the application procedure is far more difficult and computational demanding and as such usually much slower than local optimization methods [74, 150].
Local optimization method on the other hand are far easier to apply, computationally less demanding but do not guarantee that the minima found is the global minimum. Only for an important class of minimization problems, such as convex problems, where all local minima are global minima, can this feature be guaranteed.
Despite this, we will investigate local optimization methods since for good initial guesses of the optimal solutions/parameters, they often deliver sufficiently good and optimal results.
The optimization methods we will investigate are all iterative and begin with an initial guess x^0 of the parameters. The iteration tries to improve the value of the objective function f until a local minimum is reached.
Important for the optimization method are the following three properties:

- **Robustness**: The method should work well on a variety of different objective functions.
- **Computational cost**: An appropriate solution must be computed in reasonable time and require reasonable storage.
- **Accuracy**: The method should not be too sensitive to errors in the observation data.

Identifying a local minimum solely with definition 5.0.2 is rather difficult but important for iterative optimization methods to verify when a sufficient solution is reached.
For smooth objective functions f we can derive conditions that can be used to find possible local minima or to decide whether a solution is in fact a local minimum.

Definition 5.0.4 (Necessary and sufficient optimality condition)**.** *We call $\mathcal{C}$ a **necessary optimality condition** for a feasible point x of the optimization problem (5.2) if*

$$x \text{ is optimal} \Longrightarrow \mathcal{C} \text{ is satisfied} \tag{5.5}$$

*and $\mathcal{C}$ is called **sufficient optimality condition** if*

$$\mathcal{C} \text{ is satisfied} \Longrightarrow x \text{ is optimal.} \tag{5.6}$$

*We make a further classification of the optimality condition between **first order** conditions if first derivatives of the objective function f are involved and **second order** conditions if second or higher derivatives are involved.*

We now state optimality conditions for unconstrained problems which are elementary for any minimization/maximization problems.

Theorem 5.0.5 (First-Order necessary conditions [111])**.** *Let $f : D \to \mathbb{R}$ be continuously differentiable and $x^* \in \mathbb{R}^n$ be a local minimum, then*

$$\nabla f(x^*) = 0. \tag{5.7}$$

*Points $x \in \mathbb{R}^n$ with $\nabla f(x) = 0$ are also called **stationary points**.*

This theorem also shows that finding local minimum can be achieved by solving a non-linear equation system (5.7). Nevertheless it is only a necessary condition. An example are saddle points which are stationary points and solve (5.7) but are not local extrema (minima or maxima).

Theorem 5.0.6 (Second-order necessary conditions [111])**.** *Let $f : D \to \mathbb{R}$ be a smooth function and the Hessian matrix $\nabla^2 f$ continuous in an open neighbourhood of the local minimum $x^* \in \mathbb{R}^n$. Then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.*

The next theorem gives sufficient conditions on the derivatives of the objective function f that guarantee that a point x^* is a local minimum. We note that the following condition is not necessary as a local minimum that fulfils the second-order necessary condition could fail to satisfy the second order sufficient condition.

Theorem 5.0.7 (Second-order sufficient conditions [111])**.** *Let $f : D \to \mathbb{R}$ be a smooth function and the Hessian matrix $\nabla^2 f$ continuous in an open neighbourhood of the $x^* \in \mathbb{R}^n$ with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ positive definite. Then $x^* \in \mathbb{R}^n$ is a local minima.*

The second-order conditions foreshadows that the information of the Hessian matrix of the objective functions can be used in optimization methods and probably lead to faster and better convergence which we will see later.
We want to mention at this point that for nonsmooth objective functions the identification of local minima must be done with other methods. For functions with nondifferentiable points we can examine the generalized gradient [14], a generalization of the concept of gradient to the nonsmooth case.

All iterative optimization methods begin with an initial guess x_0. The closer the initial guess is to the actual optimal solution, the better and faster the convergence usually is.
For a starting point x_0 the optimization method constructs a sequence of iterations $x_0, x_1, \ldots, x_k, x_{k+1}, \ldots$ that also uses the information on the objective function $f(x_k)$ and possibly also from earlier iterates until an appropriate stop criterion has met. Examples are that the changes between iterations are sufficiently small $\| x_{k+1} - x_k \| < \delta$.
Most optimization methods are monotone, which means they require that each iteration decreases the value of the objective function further $f(x_{k+1}) \leq f(x_k)$. Although non-monotone methods exist, we do not discuss them in the following.
There are two fundamental strategies on how the new iterate x_{k+1} is being derived: Line search and trust region methods [111].

In line search methods first determines a descent direction $p_k \in \mathbb{R}^n$ along which the value of the objective function f is being reduced. Afterwards an appropriate step size α_k must be computed to determine how far to move along the direction of p_k to arrive at the next iterate x_{k+1}

$$\min_{\alpha_k > 0} f\left(x_k + \alpha_k p_k\right). \tag{5.8}$$

The other strategy is trust region. Here a model function m_k is being constructed having a behaviour near the current iteration x_k similar to the objective function f. Then the sub-problem

$$\min_{p \in \mathbb{R}^n} m_k\left(x_k + p\right) \tag{5.9}$$

is being solved for a $p \in \mathbb{R}^n$ instead where $x_k + p$ lies inside some region around x_k.
In other words, line search methods fix a search direction p_k and then search for an appropriate step size α_k whereas trust region method fix the maximum search distance first and then searches for an appropriate descent direction p.
An example of a trust region method are the Levenberg-Marquardt method

[132] where the objective function is iteratively approximated.
In the following section we will investigate line search methods to which steepest descent, conjugated gradient and Newton methods belong to.

5.1 Non-Linear Optimization (Line-Search)

In line search methods [111], each iteration is computed by a search direction p_k and an appropriate step length $\alpha_k > 0$

$$x_{k+1} = x_k + \alpha_k p_k, \tag{5.10}$$

such that

$$f(x_{k+1}) \leq f(x_k). \tag{5.11}$$

An intuitive way to analyze the rate of change in f is the application of the Taylor approximation

$$f(x_k + \alpha p) \approx f(x_k) + \alpha p^T \nabla f(x_k) + \mathcal{O}(\alpha). \tag{5.12}$$

By setting the convention of normalized search directions p_k with $\parallel p_k \parallel = 1$ we can observe in (5.12) that the highest decrease of f can be obtained at

$$\min_p p^T \nabla f(x_k).$$

Using the definition of scalar product we have

$$p_k^T \nabla f(x_k) = \parallel \nabla f(x_k) \parallel \cos(\theta). \tag{5.13}$$

Intuitively we would obtain the minimum by choosing $\theta = \pi$ such that $\cos(\pi) = -1$, which corresponds to

$$p_k = -\frac{\nabla f(x_k)}{\parallel \nabla f(x_k) \parallel}, \tag{5.14}$$

which is the **steepest descent method**.

More generally any descent direction p_k can be used as search direction as long as the next iteration x_{k+1} decreases the value of the objective function properly.
If we want to fulfil the condition $f(x_k + \alpha_k p_k) \leq f(x_k)$, the necessary condition for the choice of the search direction is

$$p_k^T \nabla f(x_k) = \parallel p_k \parallel \parallel \nabla f(x_k) \parallel \cos(\theta_k) < 0, \tag{5.15}$$

for sufficiently small value $\alpha_k \in \mathbb{R}_{>0}$ to ensure that the higher order terms of the Taylor expansion of f do not influence the rate of decrease too much.
In the literature [4, 111, 132] most search directions can be summarized into the category

$$p_k = -B_k^{-1} \nabla f(x_k) \tag{5.16}$$

with matrix $B_k \in \mathbb{R}^{n \times n}$ being symmetric and nonsingular.
For the steepest descent method this means for matrix $B_k = I$ being the identity matrix. In the following sections we will see that also Newton's method and non-linear conjugated gradient methods can be categorized as search directions methods.
Having now defined appropriate search directions, we will focus on the question of finding an appropriate step length α_k that is not too large so that we do not overshoot and not continuously decreasing the objective function f in each iteration $k+1$, but also not too small to avoid the convergence rate to be too slow to be feasible.
The ideal choice would be to solve the one-dimensional minimization problem

$$\min_{\alpha_k > 0} f(x_k + \alpha_k p_k) \tag{5.17}$$

but this involves high computational costs to solve a minimization at each iteration.
A simpler heuristic search would be to try out a sequence of possible step lengths (α_k) and to stop at an acceptable one that fulfils the condition $f(x_k + \alpha_k p_k) < f(x_k)$.
However this will not guarantee convergence since we are not always able to have a sufficient decrease in the objective function f.
Another inexact line search condition to find an appropriate step length α_k that sufficiently decreases the objective function f along the search direction p_k is

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^T p_k, \tag{5.18}$$

for a $c \in (0, 1)$, which is also called **Armijo condition** [111].
The condition (5.18) means that a decrease of the objective function needs to be proportional to the step length α_k and the directional derivative $\nabla f(x_k)^T p_k$.
Although condition (5.18) gives us a sufficient decrease, the inequality is satisfied for many small values of α_k.
In order to exclude short step lengths, a second condition is introduced

$$\nabla p_k f(x_k + \alpha_k p_k)^T \geq c_2 \nabla f(x_k)^T p_k, \tag{5.19}$$

for a $c_2 \in (c_1, 1)$, which is called **curvature condition**.
Figuratively this condition means that for strongly negative slopes at the new iteration point x_{k+1} we have the indication that we can reduce the objective

function f even further by moving further along the search direction p_k and for slightly negative or even positive slopes, that we cannot expect a further decrease of the value of the objective function by moving further along the search direction.
Both conditions (5.18) and (5.19) are known as the **Wolfe conditions** and are being used to find and verify an appropriate step length during the iteration.

5.1.1 Newton and Quasi-Newton Method

In the previous chapter we have introduced Newton's method as an iterative method to solve non-linear equation system of the form

$$f(x) = 0. \tag{5.20}$$

If applied to the gradient equation $\nabla f(x) = 0$, Newton's method would give us as solution a local minima.
Linearizing the gradient equation, we obtain

$$\nabla f(x_k + p) \approx \nabla f(x_k) + p^T \nabla^2 f(x_k). \tag{5.21}$$

Assuming that the Hessian matrix $\nabla^2 f(x_k)$ of the objective function is positive definite, we can obtain the Newton-direction

$$p_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k). \tag{5.22}$$

Regarding the general search directions (5.16) the Newton-direction is defined with $B_k = \nabla^2 f(x_k)$.
The fact that $\nabla^2 f$ is positive definite ensures that the Newton-direction satisfy the descent property condition $\nabla f(x_k)^T p_k < 0$.
For the case where the Hessian matrix is not positive definite, certain modifications must be made to enforce a descent direction [111].
An advantage of the Newton-direction is the superlinearly convergence rate.
The disadvantage is that the calculation of the Hessian matrix can be computationally expensive or not even analytically possible.
Similar to the non-linear equation setting, quasi Newton search direction can be applied to use approximations of the Hessian matrix in the same way as in the Boyden methods [41].

5.1.2 Non-Linear Conjugated Gradient Method

Lastly we present the non-linear conjugated gradient method as a line search method for the non-linear optimization problem.
The idea behind the non-linear conjugated gradient method is to extend the

linear CG-method, which minimizes the energy functional, to general non-linear functions f.
As the linear CG-method was an improvement of the gradient descent method in the linear case, also the non-linear CG-method can be regarded as an improvement to the steepest descent method (5.14).
Similar to the iteratively chosen search directions of the linear-CG-method (4.44), we have

$$p_0 = -\nabla f(x_0) \tag{5.23}$$

and

$$p_m = -\nabla f(x_m) + \beta_{m-1} p_{m-1} \tag{5.24}$$

with coefficients $\beta_m \in \mathbb{R}$ and we perform a line search for

$$\alpha_m = \arg\min_{\alpha} f(x_m + \alpha_m p_m), \tag{5.25}$$

such that we obtain the new iterate

$$x_{m+1} = x_m + \alpha_m p_m. \tag{5.26}$$

Firstly, unlike for the linear case where the coefficients β_m were clearly determined by the A-conjugation requirement (4.40) to ensure that each further iteration x_{m+1} is optimal in regards to all previous search directions, this is not the case for general non-linear functions f. In the literature there are many choices for the coefficients β_m and we refer to [65] for more details about this topic.
All the mentioned formulas for β_m are equivalent for the energy function F (4.24).
Fletcher and Reeves [51] were the first to propose coefficients defined as

$$\beta_m = \frac{\| \nabla f(x_m) \|^2}{\| \nabla f(x_{m-1}) \|^2}, \tag{5.27}$$

that are proven to make the non-linear CG-method globally convergent for non-linear functions f.
Secondly, α_m must be determined by a line search method in (5.25) as for the other presented methods.
This is not necessary for the energy function F because the optimal step length can be explicitly determined by the formula (4.42).

5.2 Non-Linear Optimization with Constraints

In this section we will extend the theory of unconstrained optimization problems (5.2) where $x \in \mathbb{R}^n$ to the constrained case subjected to the feasible

set $x \in \Omega$ for the purpose to derive conditions of solutions x^* similar to the conditions defined for the unconstrained problems.

Definition 5.2.1 (Local solution [111]). *A vector $x^* \in \Omega$ is called local solution of (5.2) if there is a neighbourhood $\mathcal{N}$ of x^* such that $f(x) \geq f(x^*)$ $\forall x \in \mathcal{N} \cap \Omega$.*

Definition 5.2.2 (Lagrangian [111]). *The function defined as $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for the constrained optimization problem (5.2) with*

$$\mathcal{L}(x, \lambda) := f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x) \tag{5.28}$$

is called Lagrangian function $\mathcal{L}$ with Lagrangian multipliers $\lambda_i \in \mathbb{R}$.

Definition 5.2.3 (LICQ [111]). *Let $x^* \in \mathbb{R}$ and the active set $A(x^*)$ defined as*

$$A(x) = \mathcal{E} \cup \{i \in \mathcal{I} | c_i(x) = 0\}, \tag{5.29}$$

then we say the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients $\{\nabla c_i(x^*, i \in A(x^*)\}$ is linearly independent.

Theorem 5.2.4 (First-Order Necessary Condition [111]). *Let $x^* \in \Omega$ be a local solution of (5.2) and LICQ holds at x^*. Then there is a Lagrange multiplier vector $\lambda^* := (\lambda_i^*)_{i \in \mathcal{E} \cup \mathcal{I}}$ such that the following conditions, also called Karush-Kuhn-Tucker conditions, hold at (x^*, λ^*)*

$$\begin{array}{lcll} \nabla_x \mathcal{L}(x^*, \lambda^*) & = & 0, & \\ c_i(x^*) & = & 0, & \forall i \in \mathcal{E}, \\ c_i(x^*) & \geq & 0, & \forall i \in \mathcal{I}, \\ \lambda_i^* & \geq & 0, & \forall i \in \mathcal{I}, \\ \lambda_i^* c_i(x^*) & = & 0, & \forall i \in \mathcal{E} \cup \mathcal{I}. \end{array} \tag{5.30}$$

Definition 5.2.5 ([111]). *For a point $x^* \in \mathbb{R}^n$ and the active set $\mathcal{A}(x^*)$ (5.29), we define the set $\mathcal{F}_1$ and its subset $\mathcal{F}_2(\lambda^*) \subset \mathcal{F}_1$ with Lagrange multiplier λ_i^* as*

$$\begin{array}{lcl} \mathcal{F}_1 & = & \left\{ \alpha d | d \in \mathbb{R}^n, \alpha > 0, \quad \begin{array}{ll} d^T \nabla c_i(x^*) = 0, & \forall i \in \mathcal{E} \\ d^T \nabla c_i(x^*) \geq 0, & \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \end{array} \right\} \\ \mathcal{F}_2(\lambda^*) & = & \left\{ w \in \mathcal{F}_1 | \nabla c_i(x^*)^T w = 0, \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* > 0 \right\}. \end{array} \tag{5.31}$$

Theorem 5.2.6 (Second-Order Necessary Condition [111]). *Let $x^* \in \Omega$ be a local solution of (5.2) and LICQ condition holds. Further let the Lagrange multipliers $\lambda_i^* \in \mathbb{R}$ satisfy the Karush-Kuhn-Tucker conditions (KKT) (5.30), then*

$$w^T \nabla_{xx} \mathcal{L}(x^*, \lambda^*) w \geq 0, \quad \forall w \in \mathcal{F}_2(\lambda^*). \tag{5.32}$$

Theorem 5.2.7 (Second-Order Sufficient Condition [111]). *Let $x^* \in \mathbb{R}^n$ be a feasible point for the optimization problem (5.2) and the Lagrange multipliers $\lambda_i^* \in \mathbb{R}$ satisfy the KKT conditions (5.30). If*

$$w^T \nabla_{xx} \mathcal{L}(x^*, \lambda^*) w > 0, \quad \forall w \in \mathcal{F}_{\in}(\lambda^*), w \neq 0, \tag{5.33}$$

then x^* is a strict local solution for the constrained optimization problem (5.2).

Having derived optimality conditions to describe solutions of constrained optimization problems we cast an eye at iterative algorithms that can solve such problems.
Although algorithms for constrained optimization exist, we will only present algorithms which take use of the already introduced optimization methods for the unconstrained case and adapt them for the constrained case.
The essential idea behind the following methods is to transform the constrained optimization problem into a sequence of unconstrained sub problems.
This motivates penalty methods, where an additional term for each constraint $c_i(x)$ is added to the objective function f of the optimization problem.
These additional terms are positive whenever the current x violates the constraints, thus increasing the value of the objective function, and in general multiplied by a sequence of so called penalty parameters $\mu_k > 0$ which increases for each iteration and penalizes violations of the constrained more severely.

Definition 5.2.8 (Penalty Method [111]). *Let $Q(x, \mu)$ be the quadratic penalty function defined as*

$$Q(x, \mu) := f(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i^2(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{I}} \left([c_i(x)]^- \right)^2 \tag{5.34}$$

with $[c_i(x)]^- = \max(-c_i(x), 0)$, and penalty parameter $\mu > 0$ then the penalty method can be defined as indicated in Algorithm 5.0 with a penalty parameter sequence $\{\mu_k\}$ with $\mu_k \searrow 0$ for $k \to \infty$.

The simplicity of the penalty methods comes with some disadvantages.
The minimization of $Q(x, \mu_k)$ becomes more difficult when μ_k decreases and the Hessian $\nabla_{xx}^2 Q(x, \mu_k)$ ill-conditioned when we are getting closer to a local minima.

Theorem 5.2.9 ([111]). *Let x_k^* be the exact global minimum of $Q(x, \mu_k)$ of the penalty method. Then for every sequence with $x_k^* \to x^*$ for $k \to \infty$, x^* is the solution of the constrained optimization problem (5.2).*

Of course the result of theorem 5.2.9 only holds when in each iteration k the global minimum is calculated which is not practical.
Most of the time iterative methods for the minimization as the one presented in the previous sections are being used to calculate inexact minimum x_{k^*} after a certain amount of iterations such that

$$\| \nabla Q\left(x_{k^*}\right) \| \leq \tau_k. \tag{5.35}$$

Theorem 5.2.10 ([111]). *If the tolerance $\tau_k > 0$ satisfies $\lim \tau_{k\to\infty} = 0$ with penalty parameter sequence $\mu_k \searrow 0$,then for all sequences $\{x_k\}$ with $\lim_{k\to\infty} x_k = x^*$ at which the constraint gradients $\nabla c_i\left(x^*\right)$ are linearly independent, x^* satisfies the KKT conditions (5.30). Then for an infinite subsequence $\mathcal{K}$ with $\lim_{k\in\mathcal{K}} x_k = x^*$*

$$\lim_{k\in\mathcal{K}} -\frac{c_i\left(x_k\right)}{\mu_k} = \lambda_i^*, \quad \text{for all } i \in \mathcal{E} \tag{5.36}$$

with λ^* satisfying the KKT conditions as well.

Algorithm 5.1 Penalty Method

1: **procedure** Penalty Method ($\mu_0 > 0$, tolerance $\tau_0 > 0$)
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Calculate $x_k = \arg\min Q(x, \mu_k)$
4: Update $\mu_{k+1} \in (0, \mu_k)$
5: **end for**
6: **end procedure**

In an effort to reduce the ill-conditioning of the minimization problems within the penalty method for each iteration k, another term is being added to the penalty function to transform the objective function into a Lagrangian function with the penalty term added to it.

Definition 5.2.11 (Augmented Lagrangian Method [111]). *Let $\mathcal{L}_A\left(x, \lambda, \mu\right)$ be the augmented Lagrangian function defined as*

$$\mathcal{L}_A\left(x, \lambda, \mu\right) := f(x) - \sum_{i\in\mathcal{E}} \lambda_i c_i\left(x\right) + \frac{1}{2\mu} \sum_{i\mathcal{E}} c_i\left(x\right)^2 \tag{5.37}$$

with penalty parameter $\mu > 0$ and Lagrangian multiplier λ. Then the augmented Lagrangian method can be defined as described in Algorithm 5.1.

The major advantage of the augmented Lagrangian methods over penalty methods is that the convergence does not require the penalty parameter μ be too small, thus reducing ill conditioning induced by too small μ's.
We refer to [111] where the convergence theorems for the augmented Lagrangian method can be found and how to extend the method for inequality constraints with the introduction of slack variables.

Algorithm 5.2 Augmented Lagrangian Method

procedure ALM ($\mu_0 > 0$, tolerance $\tau_0 > 0$)
    for $k = 0, 1, 2, \ldots$ **do**
        Calculate $x_k = \arg\min \mathcal{L}_A(x, \lambda^k, \mu_k)$
        Update $\lambda_i^{k+1} = \lambda_i^k - \frac{c_i(x_k)}{\mu_k}$.
        Choose $\mu_{k+1} \in (0, \mu_k)$
    end for
end procedure

Chapter 6

Inverse Problem: Parameter Estimation

The main purpose of a mathematical model is to accurately describe the real world problem it was derived for, such as our model for the OOC laboratory experiment (see Section 1.5) that describes the interactions between immune cells and tumour cells on the microfluidic chip environment. It contains assumptions about the intrinsic biological mechanisms such as chemotaxis. All these models contain model parameters such as diffusivity coefficients D, cellular drift velocity k_1 and many other parameters which are all specific to the type of cells and the environmental setting.
An operator F which maps the model parameters to the data which the model predicts is called **forward operator** and the problem of calculating a solution based on the model is called **forward problem**.
We already presented finite difference schemes to solve the forward problem for our mathematical model in Section 3.
A model is considered "good" in the sense that the data predicted by the model should be coherent with the observation and measurements made of the real problem.
The inversion of a forward problem is its **inverse problem**. Instead of using the model with its problem defining model parameters to predict data, we want to determine the specific model parameters that were involved to produce the data that have been made available in experiments.
In other words in the forward problem we want to know the effects, given the causes and in the inverse problem we want to know the causes given the effects.
The aim of this chapter in regards to our mathematical model of the OOC (1.45) is that with the observation data of the laboratory experiment on the OOC, we want to find appropriate model parameters such that it successfully describes the dynamics observed in the experiment.

This kind of inverse problem is also called parameter estimation problem to which the following sections are devoted to.

In the first part of this chapter we will give a brief introduction about inverse problems and the challenges of solving them and continue to appropriately formulate the parameter estimation problem as in inverse problem. What follows after are techniques such as regularizations that are designed to improve the parameter estimate. In order to control their effectiveness appropriate regularization parameters are required which need to be determined.
Afterwards we will present several parameter estimation methods designed to recover the model parameters of convection-diffusion equations and modify them in order to be adaptable for the model calibration of the OOC-model.

6.1 Introduction to Parameter Estimation

As we have already thoroughly investigated, the approximated solution of partial differential equations by means of finite different schemes is straightforward with the established definitions of consistency, stability and convergence from Chapter 2.
Inverse problems on the other hand frequently suffer from mathematical problems which do not fulfil Hadamard's definition of well-posedness [64].

Definition 6.1.1 (Hadamard)**.** *A problem is well-posed if the following three properties hold*

- ***Existence****: For all suitable data, a solution exist.*
- ***Uniqueness****: For all suitable data, the solution is unique.*
- ***Stability****: The solution depends continuously on the data.*

A problem that violates any of these properties of well-posedness is called ill-posed problem.

Inverse problems are usually ill-posed and violate one or more properties of well-posedness [48].
Nevertheless not all violations possess the same drastic significance. The violation of the existence of solutions can be avoided by relaxing the definition of solution for exact data, on the other hand for data that already contain noise, produced by measurement and observation errors, discretization errors and others, the inverse problem needs to be regularized and hence the solution will be altered.

The violation of uniqueness is a more severe problem. If more than one solution exists, some additional requirements must be made to choose an appropriate solution. In physical models, this involves for example positivity, maximal entropy or a minimal norm.
The violation of stability is the severest issue among the three properties in the numerical context and is being found when the solution of the inverse problem does not depend continuously on the data which creates instabilities in numerical methods designed to solve them.
In the following sections we restrict our work to the parameter estimation problem of the convection-diffusion equation

$$\partial_t u(x,y,t) = D\triangle u(x,y,t) - \operatorname{div}\left(\boldsymbol{v}(x,y,t)u(x,y,t)\right) \tag{6.1}$$

with diffusion coefficient $D \in \mathbb{R}_{>0}$ and velocity field $\boldsymbol{v} \in \mathcal{C}^2$.
The initial conditions are defined as

$$u\left(x,y,0\right) = u_0\left(x,y\right), \quad \text{for } (x,y) \in \Omega \tag{6.2}$$

and homogeneous Neumann boundary conditions

$$\left.\frac{\partial u}{\partial \boldsymbol{n}}\left(x,y,t\right)\right|_{\partial\Omega} = 0, \quad \text{for } (x,y,t) \in \partial\Omega \times [0,T). \tag{6.3}$$

The numerical solution $u(x_i,y_j,t_n) \approx u^n_{i,j}$ will be calculated with the finite difference scheme (3.24)

$$\begin{aligned}
u^{n+1}_{i,j} \;=\; u^n_{i,j} \;& + D\mu_x \left[\frac{\left(u^n_{i+1,j}-2u^n_{i,j}+u^n_{i-1,j}\right)}{2} + \frac{\left(u^{n+1}_{i+1,j}-2u^{n+1}_{i,j}+u^{n+1}_{i-1,j}\right)}{2}\right] \\
& + D\mu_y \left[\frac{\left(u^n_{i,j+1}-2u^n_{i,j}+u^n_{i,j-1}\right)}{2} + \frac{\left(u^{n+1}_{i,j+1}-2u^{n+1}_{i,j}+u^{n+1}_{i,j-1}\right)}{2}\right] \\
& - \tfrac{\lambda_x}{2}\left(\widehat{f}^{x,n}_{i+1,j}u^n_{i+1,j} - \widehat{f}^{x,n}_{i-1,j}u^n_{i-1,j}\right) - \tfrac{\lambda_y}{2}\left(\widehat{f}^{y,n}_{i,j+1}u^n_{i,j+1} - \widehat{f}^{y,n}_{i,j-1}u_{i,j-1}\right) \\
& + \tfrac{\lambda_x}{2}\left(|\widehat{f}^{x,n}_{i+1,j}|u^n_{i+1,j} - 2|\widehat{f}^{x,n}_{i,j}|u^n_{i,j} + |\widehat{f}^{x,n}_{i-1,j}|u^n_{i-1,j}\right) \\
& + \tfrac{\lambda_y}{2}\left(|\widehat{f}^{y,n}_{i,j+1}|u^n_{i,j+1} - 2|\widehat{f}^{y,n}_{i,j}|u^n_{i,j} + |\widehat{f}^{y,n}_{i,j-1}|u^n_{i,j-1}\right) \\
& + \tfrac{\triangle t}{2}\left(g(t_{n+1},x_i,y_j) + g(t_n,x_i,y_j)\right)
\end{aligned} \tag{6.4}$$

with appropriate mesh grid size $\triangle x, \triangle y > 0$ and time step size $\triangle t > 0$ that satisfy the stability conditions (3.25).
We can now define the forward problem as follows:

Definition 6.1.2 (Forward problem)**.** *Let the solution of the convection-diffusion equation (6.1) with initial and boundary conditions (6.2), (6.3) be the forward*

problem. Then with the finite difference scheme (6.4) we define the non-linear operator F as

$$F(\Theta) = u \tag{6.5}$$

with model parameter set $\Theta := (D, \boldsymbol{v}) \in \mathbb{R}^p$, which contains the diffusion coefficient $D \in \mathbb{R}_{>0}$ and the discretized velocity field $\boldsymbol{v} := (\boldsymbol{v}^x, \boldsymbol{v}^y)$, with $\boldsymbol{v}^x := (\boldsymbol{v}^{x,0}, \boldsymbol{v}^{x,1}, \dots, \boldsymbol{v}^{x,n})$, $\boldsymbol{v}^y := (\boldsymbol{v}^{y,0}, \boldsymbol{v}^{y,1}, \dots, \boldsymbol{v}^{y,n})$ and $\boldsymbol{v}^{x,n} := \left(v^{x,n}_{0,0}, v^{x,n}_{1,0}, \dots, v^{x,n}_{N_x+1,N_y+1}\right)$, $\boldsymbol{v}^{y,n} := \left(v^{y,n}_{0,0}, v^{y,n}_{1,0}, \dots, v^{y,n}_{N_x+1,N_y+1}\right)$ and approximated solution $u := (u^0, \dots, u^n)$ with $u^n = \left(u^n_{0,0}, \dots, u^n_{N_x+1,N_y+1}\right)$.

Furthermore when we speak of inverse problem, we will now exclusively refer to the inversion of this forward problem as the parameter estimation problem for parameters $\Theta \in \mathbb{R}^p$ and refer to observation/measurement data used to obtain those parameters as u_{data}.
We let $\Theta^* \in \mathbb{R}^p$ be defined as the exact model parameters and $\Theta_{opt} \in \mathbb{R}^p$ as the optimal parameter values obtained numerically.

There are several ways to mathematically define the inverse problem for the forward problem (6.5). The simplest is to use numerical methods for solving non-linear equation systems such as Newton's method or non-linear conjugated gradient method as described in Chapter 4 to solve $F(\Theta) = u_{data}$ for Θ. However, this approach would not lead to any satisfying results because neither the existence nor the uniqueness of the solution Θ is guaranteed, nor are the observation data u_{data} the exact solution of $F(\Theta)$ for a specific Θ which also does not necessarily depend continuously on such observation data.
For this reason instead of treating the inverse problem as a non-linear equation system, it is much better to reformulate (6.5) as a minimization problem that creates a connection between the forward problem $F(\Theta)$ and observation data u_{data}.

Definition 6.1.3 (Inverse Problem)**.** *Let the forward problem be defined as in Definition 6.1.2. Then we define the inverse problem as the minimization problem*

$$\Theta_{opt} = \arg\min_{\Theta \in \mathbb{R}^p} K(\Theta) \tag{6.6}$$

with non-linear least square functional

$$K(\Theta) := \| F(\Theta) - u_{data} \|^2 \tag{6.7}$$

with a norm $\| \cdot \|$ and observation data u_{data}.

Remark 6.1.4 (Observation data)**.** *In this work we consider the observation data u_{data} to be of the form $u_{data} := (u^0_{data}, u^1_{data}, \dots, u^m_{data})$ with*

$$u^m_{data} := \left(u^m_{data_{k,l}}\right)_{k=0,\dots,N_x+1, l=0,\dots,N_y+1}$$

with $x_k = k\triangle x$, $y_l = l\triangle y$ for $m+1$ different times, where the observation data are available on the discretized meshgrid of the whole domain Ω.
The forward operator F is defined such that $(F(\theta))^{n^*}_{i,j} = u^{n^*}_{i,j}$ coincide with the value of $(u_{data})^{n^*}_{i,j}$ at the same meshgrid point at time t_{n^*}. This is usually achievable by interpolating the numerical solution to match the observation data.

The smaller the value of the objective functional $K(\Theta)$ gets for a $\Theta \in \mathbb{R}^p$, the smaller the error between model prediction and observation and the more realistic the model can be considered.
This minimization problem (6.6) can be solved with the methods for non-linear optimization introduced in the previous Chapter 5 such as line search methods or non-linear conjugated method.
However solving the inverse problem (6.6) has three difficulties:

- **Ill-posedness**: The violation of Hadamard's properties, as mentioned above.
- **Nonlinearity**: The nonlinearity of the objective function might result in the presence of several local minima [8]. Depending on the initial guess and the non-linear optimization method, convergence can only be guaranteed towards one of those local minima which is not necessarily the global minimum.
- **Computational costs**: Solving a non-linear minimization problem does involve many nested iterative computations of the forward problem (6.5) itself for a variety of parameter sets Θ. If the forward problem is complicated, computationally costly and additionally has stability conditions depending on the parameters, as it is the case for finite difference methods, then the computational cost can be immense.

The latter two problems can be dealt with the application of multigrid methods.
Due to the coarser grid, the number of parameters are reduced and also computations on the coarser grid are more efficient [20].
The first problem can be dealt with by imposing additional information on the unknown parameter values $\Theta \in \mathbb{R}^p$ and modify the objective function (6.7) such that the inverse problem is less sensitive to any perturbations.

6.2 Parameter Estimation Methods For Convection-Diffusion Problems

With the techniques described in the previous sections, we are now able to develop the methods used for the parameter estimation problems.
Two different approaches have been investigated by Tao Liu [91, 130] and Nielsen et al [110] and deal with the problem of parameter estimation for the convection dominated non-linear convection-diffusion equation in the multi-phase porous media.
The equation the mentioned papers address is

$$\partial_t u(x,y,t) + \partial_x \phi(u) + \partial_y \psi(u) - \nabla \cdot ((q(x,y)\, N(u) \nabla u(x,y,t)) \quad = \quad s(x,y,t), \tag{6.8}$$

for $\Omega \times (0,T)$ and initial-boundary value conditions

$$\begin{aligned} u\,(x,y,0) &= \phi(x,y), && \text{for } \Omega, \\ u\,(x,y,t) &= 0, && \text{for } \delta\Omega \times (0,T), \end{aligned} \tag{6.9}$$

where u is the concentration field, q the constant in time permeability coefficient, N a positive non-linear function, s the piecewise smooth source function and ϕ, ψ non-linear S-shaped flux function of Buckley-Leverett type. The goal of such works is to recover the unknown permeability coefficient q which is a function that varies in space (x,y) and is constant in time t, but the two methods differ in how the inverse problem is defined.

In Sections 6.2.3 and 6.2.4 we will present the work of Liu [91, 130] and Nielsen [110] and their main ideas. Afterwards we extend and modify their approaches on the general convection-diffusion equation

$$\partial_t u(x,y,t) = D \triangle u(x,y,t) - \operatorname{div}\left(\boldsymbol{v}(x,y,t) u(x,y,t)\right) \tag{6.10}$$

with constant diffusion coefficient $D \in \mathbb{R}_{>0}$ and velocity field function $\boldsymbol{v} \in \mathcal{C}^2(\mathbb{R}^2 \times (0,T))$, which can vary not only in space but also in time.
The novelty in this parameter estimation problem consists of two extensions to the methods presented in [91, 110].

First Extension: Diffusion and Convection parameters

Firstly, we want to identify not only the diffusion coefficient D but also the velocity field function $\boldsymbol{v}$ in the convection term of the equation (6.10), which was fixed and known in their previous work. Despite the fact that our diffusion coefficient parameter D is a constant scalar, the velocity field function

$\boldsymbol{v}(x, y, t)$ is of similar complexity as the permeability coefficient $q(x, y)$. We are trying to identify two parameters sets which control two different dynamics; the diffusion and the convection. Depending on the observation data u_{data}, these two parameter sets can be highly correlated, which means that small changes in one parameter set causes inevitable changes to the other parameter set. This causes multiple problems during the process of finding the correct minima.

Second Extension: time dependence of unknown parameter

Secondly, and the most challenging extension, is that the velocity field function $\boldsymbol{v}(x, y, t)$ not only varies in space (x, y), but also in time t. Problems as such have not been investigated before for these kind of parameter estimation problems and this is an original contribution of this thesis, which we investigate through numerical tests in Subsection 6.2.5.

Additionally we will also introduce another parameter estimation method, presented in the paper of Ferguson et al. [49]. There, a parameter estimation problem for an one-dimensional non-linear convection diffusion-reaction model for the collective movement of two cell types is being solved which is very similar to our laboratory model (3.1) .
In their work, they intend to estimate up to 20 unknown model parameters, each involved in different terms such as diffusion, chemotaxis, crowding effects and other cell migration characteristics.
These parameters are constant in space, but vary in time, which is in contrast with [91, 110] where the parameters are constant in time but vary in space.
Another difference is the availability of observation data for the parameter estimation problem. Although the mathematical model used in [49] is a macroscopic one, which would need macroscopic density as observation data, the observation data are available in form of cell trajectories. In order to use the microscopic data for the macroscopic model in [49] without introducing further errors due to transformations, the so called Maximum-Likelihood method (see [106]) is used.
The aim in [49] is not in particular the recovery of the unknown parameters, but the model inference and drawing conclusions about the drivers of the cell movements by comparing each of their possible models and using model selection criteria such as the Akaike information criterion (AIC) [145] and the WAIC (widely applicable information criterion) [2, 146].

6.2.1 Regularization

The most severe issue with inverse problem is that the data do not depend continuously on the solution and arbitrary small perturbations can lead to arbitrary large perturbations in the solution.
The condition number as defined in [17] is a measurement of how sensitive a function is to changes in the input and how much the resulting output changes from the input.

Definition 6.2.1 (Condition number)**.** *The relative condition number of a function* $f : \mathbb{R}^n \to \mathbb{R}^m$ *at point* $x \in \mathbb{R}^n$ *is the smallest number* $\mathrm{cond}_{rel}(f) > 0$ *such that:*

$$\frac{\| f(x) - f(\tilde{x}) \|}{\| f(x) \|} \leq \mathrm{cond}_{rel}(f)(x) \cdot \frac{\| x - \tilde{x} \|}{\| x \|}, x \to \tilde{x}. \tag{6.11}$$

If the function f *is continuously differentiable, then*

$$\mathrm{cond}_{rel}(f)(x) = \frac{\| f'(x) \| \| x \|}{\| f(x) \|}, \tag{6.12}$$

where $f'(x)$ *is the Jacobi matrix of* f*.*

Inverse problem are usually suffering from very large condition numbers [48] and hence they are very ill-conditioned problems. For our parameter estimation problem (6.6) small errors between the solution of the forward problem and the observation data

$$\| F(\Theta) - u_{data} \| < \delta \tag{6.13}$$

lead to arbitrary large error between the estimated parameter solution and the exact solution $\| \Theta - \Theta^{exact} \|$. Errors in form of noise are already present because the forward problem as defined in definition 6.1.2 is not solved exactly but through a discretization method which introduces an error as discretization noise. In order to stabilize the approximation of the solution, so called regularization methods have to be used

Definition 6.2.2 (Tikhonov regularization)**.** *[151] For the inverse problem (6.6) with non-linear least square functional*

$$K_0(\Theta) = \| F(\Theta) - u_{data} \|_2^2 \tag{6.14}$$

we define the Tikhonov regularization term as

$$R(\Theta) = \lambda \| T(\Theta - \Theta_0) \|_2^2 \tag{6.15}$$

and extend the objective function K_0 with (6.15)

$$K(\Theta) = \| F(\Theta) - u_{data} \|_2^2 + \underbrace{\lambda \| T(\Theta - \Theta_0) \|_2^2}_{\text{general Tikhonov regularization}} \tag{6.16}$$

with Tikhonov matrix $T \in \mathbb{R}^{p \times p}$, regularization parameter $\lambda > 0$ and a-priori estimate $\Theta_0 \in \mathbb{R}^p$, which represents a-priori knowledge about the solution.

There is a variety of Tikhonov matrices T to choose from, each with their own specific advantages and disadvantages which are discussed further in [48, 95].
Choosing the identity matrix for the Tikhonov matrix $T = I \in \mathbb{R}^{p \times p}$ gives a preference to parameter solutions Θ with smaller norms, i.e smaller values.
However also first and second order difference operators, T_1 and respectively T_2, in form of derivative matrices can be used to favour smooth parameter solutions:

$$T_1 = \begin{pmatrix} -1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & & \vdots & \vdots & \\ 0 & \cdots & 0 & -1 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{(p-2) \times p}$$

$$and \tag{6.17}$$

$$T_2 = \begin{pmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & & \vdots & \vdots & \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(p-2) \times p}.$$

The choice of the Tikhonov matrix T depends on the noise and the characteristic of the parameters Θ and in most cases must be chosen experimentally. In [122] different Tikhonov matrices have been applied to an inverse problem with the results being not significantly different.
The regularization term not only imposes a priori information about the parameters, such as the value Θ_0 or characteristics such as smoothness or continuity, but also imposes numerical stability.
The strength of regularization is controlled by the regularization parameter $\lambda > 0$. If the regularization is too strong, the dampening effect on the parameter solution Θ will be large, hence the data-fitting term will be large as well. If the regularization is too small, then the solution will be a good fit but the solution may be dominated by features introduced through noise.
If the observation data u_{data} contain noise, then the observation data fit the model such that for the error $\| F(\Theta) - u_{data} \|^2 \leq \delta$. But there may be many local minima as parameter solutions $\Theta^{opt} \in \mathbb{R}^p$. For this reason with the help of

the regularization term, only solutions will be considered that minimize the regularization term as well and ensure that certain properties such as small values or smoothness will be included. We illustrate the different Tikhonov regularizations in the following example.

Example 2. *We consider the one-dimensional convection-diffusion equation (6.1) with fixed diffusion coefficient $D = 0.1$ and two different velocity field $\boldsymbol{v}$ which we want to estimate:*

(a)

$$\boldsymbol{v} := \cos(2\pi x), \quad \text{for } 0 \leq x \leq 1, \tag{6.18}$$

(b)

$$\boldsymbol{v} := \begin{cases} -1, & \text{for } 0 \leq x < 0.5, \\ 1.5, & \text{for } 0.5 \leq x \leq 1. \end{cases} \tag{6.19}$$

The observation data u_{data} for different times t have been obtained by solving the equation by finite difference scheme on a fine grid and added with 2% Gaussian noise.
The parameter estimation for parameter $\boldsymbol{v}$ can be seen in Figure 6.1 for three different Tikhonov matrices T:

- *Identity matrix $T = I$.*
- *First order difference operator (6.17) $T = T_1$.*
- *Second order difference operator (6.17) $T = T_2$.*

The regularization parameter λ has been obtained experimentally by choosing a λ such that for the parameter estimate $\tilde{\boldsymbol{v}}$ we have

$$\lambda_{opt} = \arg\min_{\lambda} \left(\frac{\| \boldsymbol{v} - \tilde{\boldsymbol{v}} \|}{\| \boldsymbol{v} \|} \right). \tag{6.20}$$

Results: *We can conclude the following: Due to the 2% Gaussian noise, which have been added to the observation data, the solution of the non-regularized parameter estimation $\tilde{\boldsymbol{v}}$ contains spurious oscillations, whereas all solutions of the regularized parameter estimation with Tikhonov matrices $T = I, T_1, T_2$ have dampen most of these oscillations well, which result in a smoother parameter estimate $\tilde{\boldsymbol{v}}$. However, there are only small differences between the three different Tikhonov matrices that have been used on the regularized objective functional $K(\Theta)$. For $T = I$ identity matrix, oscillations are more strongly present compared to $T = T_1$ and $T = T_2$, where the regularized solutions $\tilde{\boldsymbol{v}}$ are much smoother and barely contain oscillations.*

Although the regularized solutions $\tilde{v}$ with $T = T_1$ and $T = T_2$ for velocity field v in b) (6.19) are almost identical and contain small oscillations, especially around the discontinuity $x = 0.5$, for the smoother velocity field v in a) (6.18) the regularized solution $\tilde{v}$ with T_2 shows much lower oscillations compared to $T = T_1$.

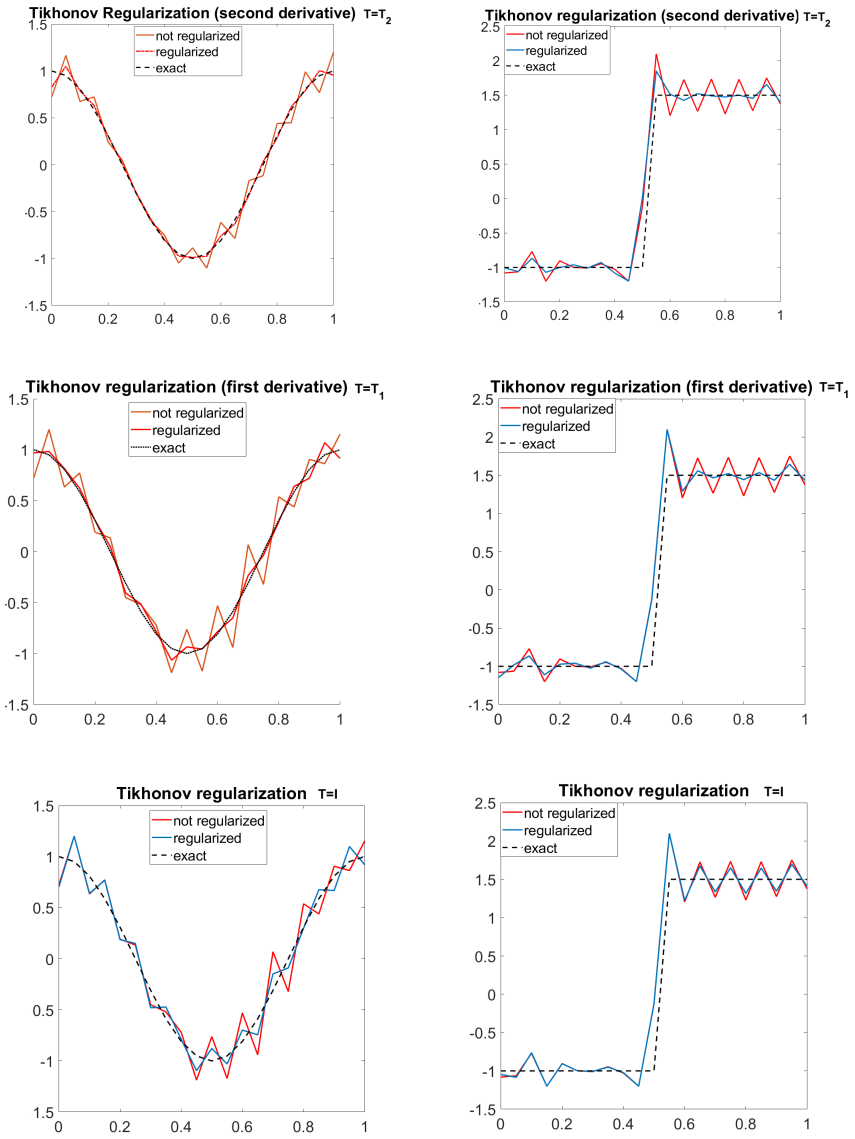


Figure 6.1: Parameter estimate of velocity function v for a) (6.18) left, and b) (6.19) right. The black curve shows the exact solution, the red curve the non-regularized parameter estimate and the blue curve the regularized one. In all examples does the regularized solution dampens oscillations and are in general a better fit compared to the non-regularized solution.

Usually the exact solution is not available which makes it impossible to find the optimal regularization parameter. Also the value of the optimal regularization parameter $\lambda_{opt} > 0$ depends on the forward problem and the noise and

is usually unknown. Often in practical problems the regularization parameter is determined experimentally through trial and error but there exist many sophisticated approaches to find the optimal regularization parameter such as discrepancy principle, cross-validation and the L-curve method, which we present in the next section [10, 12, 68].

L-curve Criterion

We will briefly introduce the idea behind choosing the regularization parameter based on the L-curve criterion described in [68].
The L-curve is a log-log plot for all valid regularization parameters $\lambda > 0$ of the norm of a regularized parameter solution Θ versus the norm of the corresponding residual norm, which is the value of the non-regularized objective functional $K_0(\Theta)$ (6.7).
It is a convenient graphical tool for displaying the trade-off between the size of a regularized solution and its fit to the given data, as the regularization parameter varies.
Heuristically, choosing a $\lambda_{Lcurve} > 0$ such that it seeks to balance the two quantities $\| \Theta \|$ and $K(\Theta)$ often leads to a sufficient estimate of the optimal regularization parameter λ_{opt} for the minimization problem (6.6) which gives the best approximation of Θ_{opt}.
As can be seen in Figure 6.2, by increasing the value of λ, only the norm of $\| \Theta \|_2$ decreases until a certain point. Then the change of the value of $\| \Theta \|_2^2$ flattens whereas the value of the residual $\| F(\theta) - u_{data} \|_2^2$ increases.
The estimated optimal λ_{opt} can be usually found at the corner of the curve, or mathematically, the estimated optimal λ_{opt} is usually located at the highest curvature of the L-curve, before for larger regularization parameter λ, the trade-off between the size of the regularized solution and its goodness of fit for the given observation data begins to shift towards a worsening of fit without a significant decrease of the size of the regularized solution.
Although the L-curve criterion can give sufficient estimates of the optimal regularization parameter λ_{opt}, there are two limitations:

1. If the exact parameter solution is very smooth, then the L-curve criterion will fail because the optimal regularization parameter is located not at the point of the highest curvature of the L-curve. This was pointed out in Hanke [67].

2. For an increasing problem size p, which in the context of our parameter estimation problem means on a finer mesh grid, the λ_{Lcurve} obtained from the L-curve criterion does not behave consistently with the optimal regularization parameter λ_{opt}, and tends to oversmooth the solution.

This was worked out in Vogel [142], where a more detailed explanation can be found.

Although both limitations do apply to our inverse problem of the parameter estimation (6.6) where part of the unknown model parameters is the velocity field $\boldsymbol{v}$ which in most application is a smooth function, the L-curve criterion benefits from the discretization on a coarser grid level making it possible to find a sufficient regularization parameter through the L-curve criterion. We will demonstrate with the following example.

Example 3. *We consider the inverse problem (6.6) with the regularized objective functional $K(\Theta)$ (6.16) for the one-dimensional convection-diffusion equation (6.1) and choose as diffusion coefficient $D = 1$ and for the velocity field $\boldsymbol{v}(x) = \cos(2\pi x)$ with initial condition $u_0(x) = 10e^{-(x-0.5)^2}$. The numerical solution is calculated with the finite difference scheme (3.22) with homogeneous Neumann boundary conditions on the interval $\Omega := [0, 1]$, respecting the stability condition (3.23).*
We fix $\triangle x = 0.05$ and choose the discretized velocity field as unknown parameter for the parameter estimation with $\Theta = \boldsymbol{v} = (v_0, v_1, \ldots, v_{N_x+1})^T$.
In Figure 6.2 the L-Curve is plotted with its respective optimal regularization parameter λ_{opt} and the regularization parameter λ_{Lcurve} through the L-Curve criteria at the L-curves largest curvature. The value of $\lambda_{Lcurve} = 0.003$ is very close to the optimal value $\lambda_{opt} = 0.0048$. The optimal λ_{opt} has been obtained experimentally by calculating the smallest relative error $err_{rel} = \frac{\|\Theta - \tilde{\Theta}\|}{\|\Theta\|}$.

Results: *As we solve the regularized minimization problem on the fine and coarse grid, it can be clearly seen from the L-Curves in Figure 6.2 that the regularization parameter λ_{Lcurve} estimated through the L-curve criterion is close to the optimal regularization parameter λ_{opt} despite the fact that v is in fact a smooth function.*
This leads to the conclusion that the first limitation of the L-Curve criterion can be overcome by using a coarser discretization of the velocity field $\boldsymbol{v}$ such that the estimated regularization parameter $\lambda_{Lcurve} \approx \lambda_{opt}$ gives a sufficiently good estimate. This is important because generally the exact solution Θ^* is not known, hence minimizing the relative error to find the optimal λ_{opt} is not possible.

6.2.2 L^1-L^2-Norm

In the previous sections we have defined the inverse problem for the parameter estimation of the convection-diffusion equation as the unconstrained

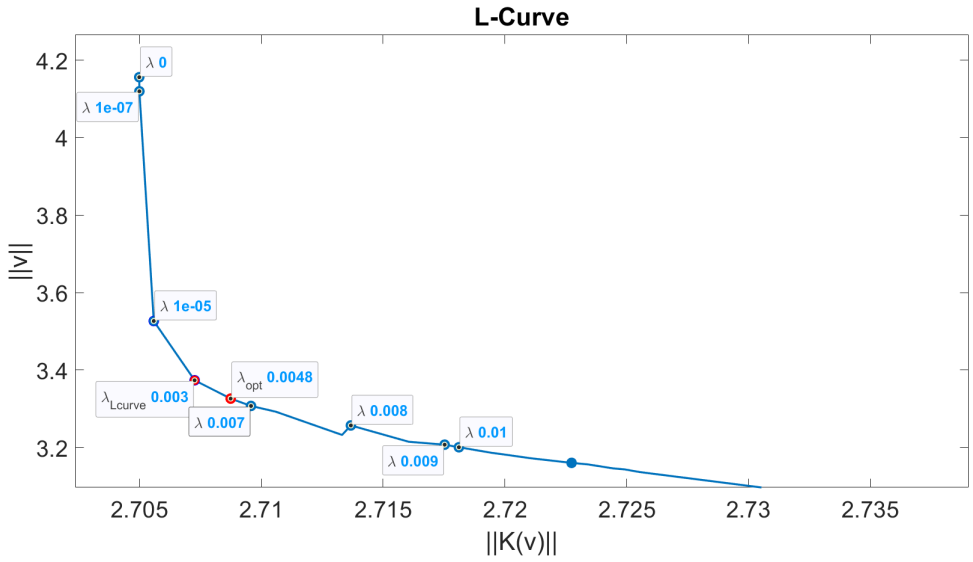


Figure 6.2: L-curve for example 3 with parameter $\Theta = \boldsymbol{v}$. X-axis shows $\parallel K(\boldsymbol{v}) \parallel$ and Y-axis $\parallel \boldsymbol{v} \parallel$ for different regularization parameters λ. Optimal regularization parameter λ_{opt} and L-curve regularization parameter $\lambda_{Lcurve}$are indicated in red.

minimization problem and introduced with the Tikhonov regularization the regularized minimization problem

$$\min_{\Theta} K(\Theta) = \parallel F(\Theta) - u_{data} \parallel_2^2 + \lambda \parallel T(\Theta - \Theta_0) \parallel_2^2 \tag{6.21}$$

with forward operator F, observation data u_{data} and regularization term with Tikhonov matrix T, a-priori estimate Θ_0 and regularization parameter λ.
This leads to the intuitive idea of choosing the L^1-norm for the regularization term instead of the L^2-norm.
In context of least square problems, this problem is also known as LASSO in statistics [152].
To investigate the differences of both approaches we define the two objective functions

$$\begin{aligned} K_1(\Theta) &:= \parallel F(\Theta) - u_{data} \parallel_2^2 + \lambda_1 \parallel \Theta \parallel_1, \\ K_2(\Theta) &:= \parallel F(\Theta) - u_{data} \parallel_2^2 + \lambda_2 \parallel \Theta \parallel_2^2 . \end{aligned} \tag{6.22}$$

The same theory applies as described for the Tikhonov regularization for the regularization parameters $\lambda_1, \lambda_2 > 0$, where a large λ increases the damping effect and can cause over-smoothing of the solution, and a small λ could fit the solution better but might be dominated by noise.
The essential difference between these two regularization terms is that the L^1-regularization penalizes the sum of the absolute values of Θ. This encourages the elements $(\Theta)_i$ of Θ to shrink to zero faster and can set them to zero, whereas the L^2-regularization penalizes the sum of the squared values of Θ

and does not have this ability but shrinks all parameters more evenly as seen in [85].
For the parameter estimation this can be beneficial if redundant parameters are present in the model, which then can be completely excluded and thus making the model more interpretable.
It was also shown in [85] that L^1-regularization performs significantly better than the L^2-norm when the observation data contain noise.
At the same time this can also be unfavourable for not redundant highly correlated parameters since the L^1-regularization simply picks one parameter from a group of correlated parameters and ignores the remaining ones which increase the overall instability in the parameter estimation and also leading to multiple solutions.
Another disadvantage is that the L^1-regularization results in a non-smooth objective functional which needs special treatment for the minimization process.
Zou and Hastie [152] introduced the elastic net regularization which combines the L^1- and L^2-norm in the objective function.

$$K_3(\Theta) := \| F(\Theta) - u_{data} \|_2^2 + \lambda_1 \| \Theta \|_1 + \lambda_2 \| \Theta \|_2^2 . \tag{6.23}$$

Their results presented in their paper [152] is that the elastic net regularization provides the benefits of the L^1-norm where noise is handled better and parameters tend to zero but also benefits from the L^2-norm by allowing grouping effect, i.e it shrinks highly correlated parameters simultaneously instead of choosing one and setting the others to zero.
We will try to verify the different results in the following example.

Example 4. *We consider the inverse problem (6.6) with the following three objective functional $K(\Theta)$:*

1. $K_1(\Theta) := \| F(\Theta) - u_{data} \|_2^2 + \lambda_1 \| \Theta \|_1$,
2. $K_2(\Theta) := \| F(\Theta) - u_{data} \|_2^2 + \lambda_1 \| \Theta \|_2^2$,
3. $K_3(\Theta) := \| F(\Theta) - u_{data} \|_2^2 + \lambda_1 \| \Theta \|_1 + \lambda_2 \| \Theta \|_2^2$,

for parameter $\Theta = \boldsymbol{v}$ with initial condition $u_0(x) = 10e^{-(x-0.5)^2}$ and no-flux Neumann boundary conditions $\frac{\partial u}{\partial n}|_{\delta\Omega} = 0$ for the domain $\Omega := [0, 10]$.
For the diffusion coefficient we choose $D = 1$ and the constant in time velocity field $v(x) = 10\sin(x)$.
For the choice of optimal regularization parameters for $\lambda_1 > 0$ and $\lambda_2 > 0$ we used the L-curve criterion whereas for the regularization parameters λ_1, λ_2 of the elastic net (6.23), the regularization parameters have been determined experimentally. We remark that a generalized L-curve criterion can be applied for more than one regularization parameter in which the L-curve is a

multidimensional L-surface [68].
On purpose no observation data have been used for the interval $[0, 2]$ *in order to test the performance of each regularization technique in absence of data. In Figure 6.3 the discretized velocity field for the* L^1*,* L^2 *and elastic net regularization is plotted.*

Results: *As expected the* L^1*-regularization sets the velocity field parameters in the interval with no observation data close to zero whereas the parameters in the* L^2*-norm regularization are distributed randomly around the x-axis. However, the* L^2*-norm has more accurate velocity field parameters in areas where observation data were available compared to the* L^1*-norm. The elastic net regularization on the other hand stands between both* L^1 *and* L^2 *norms such that where no observation data were available, the velocity field parameters are closer to zero than for the* L^2*-norm and for the remaining area where observation data were available, the parameters lie between the results of the* L^1*- and* L^2*-norm. As for the relative error* $err = \frac{\|v^{exact} - v\|_2}{\|v^{exact}\|}$ *the smallest error with* $6,8\%$ *has the* L^1*-norm regularization, with* 8% *for the elastic net regularization and* $10,5\%$ *for the* L^2*-norm regularization. Although this can be explained with the fact that the velocity field parameters of the* L^2*-norm regularization in the interval with no observational data are much larger compared to the other two regularization techniques.*

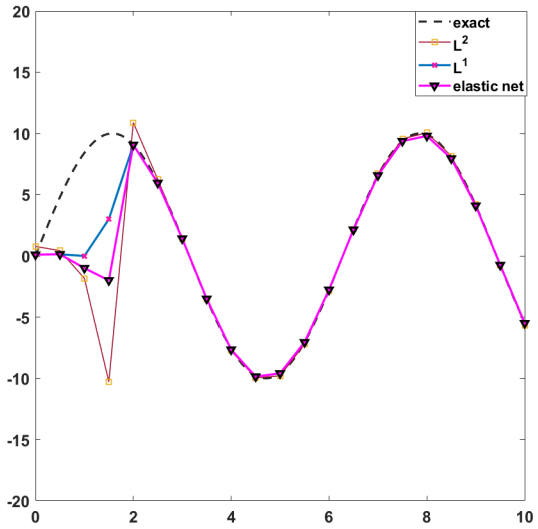


Figure 6.3: Different parameter estimation of example 4.

6.2.3 Multigrid Parameter Estimation

In Liu [91, 130] the forward problem (6.8) is solved with a finite difference scheme and the inverse problem defined as a non-linear minimization problem with a Tikhonov-regularized non-linear least square functional.
The complete numerical method to solve the inverse problem is a combination of solving the forward problem for different parameter sets in each nested iteration of a regularized Gauß-Newton method as the optimization method. This methodology is applied on non-linear multigrid method.
There are multiple benefits of applying a non-linear multigrid approach that was pointed out in [91], which highly motivate the multigrid application on inverse problems:

- **Reduction of computational cost**: Solving the inverse problem involves solving the forward problem numerically with a finite difference scheme. On a very fine mesh grid this means high computational cost. Furthermore, finite difference schemes' stability conditions depend in general on the model parameters as well which can be a challenge during the optimization iteration when different model parameter sets are being used and might require very small time step sizes $\triangle t$. Computations on coarse grids however not only reduce the computational cost of the finite difference scheme but also allows in general much larger time step sizes $\triangle t$.
- **Reduction of local minima**: Computations on coarse grids lead automatically to a reduced number of parameters which not only decreases the number of local minima tremendously but also put those remaining local minima further apart from each other. This reduces the risk of getting trapped in an undesired minimum when the initial guess for the optimization method is not close enough to the optimal parameter set [91].

The first point, the reduction of computational cost due to a coarser mesh grid $\triangle x, \triangle y > 0$ is trivial. To demonstrate the effect of the reduction of local minima, we investigate the following example.

Example 5. *We consider the one-dimensional convection-diffusion equation (6.10) as the forward problem, which is solved with the finite difference scheme (3.22) with stability conditions (3.23) on the interval* $\Omega = [0, 1]$*, with initial condition* $u_0(x) = 10e^{-(x-0.5)^2}$*, diffusion coefficient* $D = 0.5$ *and constant velocity field* $v = c = 5$*.*
We solve the inverse problem (6.6) with non-regularized objective functional

$$K_0(\Theta) = \| F(\Theta) - u_{data} \|_2^2, \tag{6.24}$$

for $\Theta = (D, c)^T \in \mathbb{R}^2$ for two mesh grid sizes

a) Fine mesh grid with $\triangle x = 0.01$.

b) Coarse mesh grid with $\triangle x = 0.1$.

The results are represented in Figure 6.4 where the values of $K(\theta) = K(D, c)$ are shown in a two-dimensional color surface for $(D, c) \in [0, 2] \times [-2, 16]$ and local minima labeled for a) and b).

Results: *Figure 6.4 shows clearly that firstly the number of local minima on the coarse grid is extremely reduced compared to the finer grid. Secondly, the numerical test indicates that the descent direction of the gradient of the objective functional ∇K leads towards the global minima for the coarse grid, which allows the initial guess Θ_0 for the optimization to be much further away from the exact solution $\Theta^* = (0.5, 5)^T$ whereas for the fine grid, this is not the case. These results support the motivation to apply non-linear multi grid methods for parameter estimation problems.*

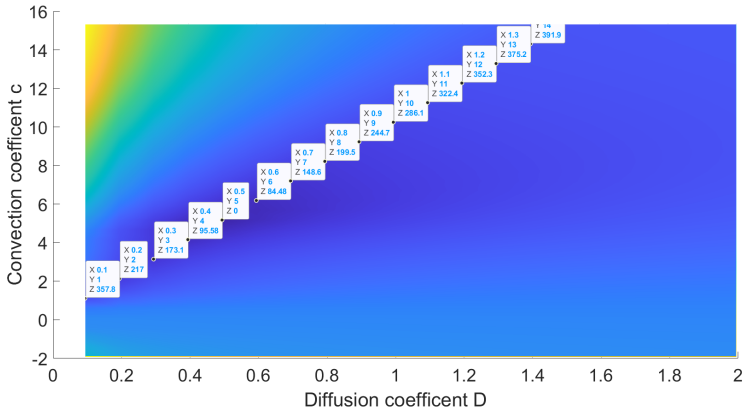
We present the methodology for the parameter estimation which is introduced in [91] for the permeability (diffusion) coefficient $q(x, y)$ in (6.8) in a more generalized form for our convection-diffusion equation (6.10) where the velocity field $\boldsymbol{v}(x, y)$ is the parameter that needs to be determined for fixed and known diffusion coefficient $D \in \mathbb{R}$ and refer to [91, 130] for (6.8).

Forward problem

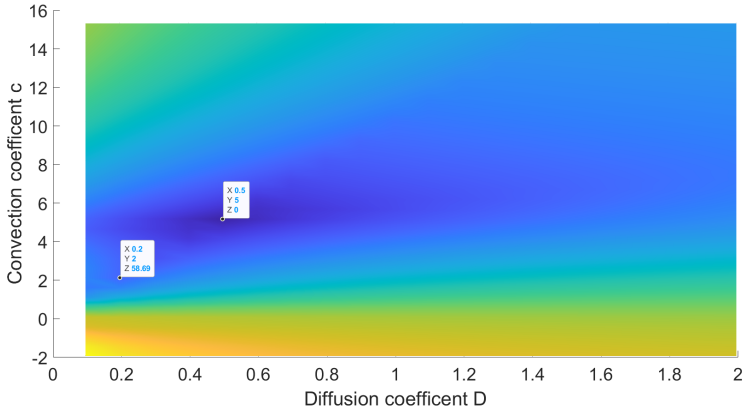
As the forward problem we consider the two-dimensional convection-diffusion problem

$$\begin{cases} \partial_t u\,(x, y, t) &= D \triangle u\,(x, y, t) - \operatorname{div}\left(\boldsymbol{v}\,(x, y)\, u\,(x, y, t)\right), \\ u\,(x, y, 0) &= u_0\,(x, y), \\ \frac{\partial u}{\partial \boldsymbol{n}}\big|_{\partial\Omega} &= 0 \end{cases} \tag{6.25}$$

with diffusion coefficient $D \in \mathbb{R}_{>0}$ and velocity field $\boldsymbol{v}\,(x, y) \in \mathcal{C}^2\,(\mathbb{R}^2)$.



(a) Objective functional K on finer mesh grid $\triangle x = 0.01$



(b) Objective functional K on coarser mesh grid $\triangle x = 0.1$

Figure 6.4: Both plots show the value of the objective functional K for the one-dimensional convection diffusion equation with D and constant c. K has on the finer grids many local minima. On the coarser grid most of the local minima disappear leaving only the global minima $D = 0.5$ and $c = 5$ and another local minima. Additionally the region around the global minima on the coarser grid assumes a better convergence and bigger basin than on the finer grid.

Discretization

First, the partial differential equation (6.25) for the two-dimensional domain Ω is discretized with the explicit finite difference scheme (3.24)

$$
\begin{aligned}
u_{i,j}^{n+1} \quad = \quad u_{i,j}^{n} \quad & + D\mu_x \left[\frac{\left(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}\right)}{2} + \frac{\left(u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}\right)}{2} \right] \\
& + D\mu_y \left[\frac{\left(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}\right)}{2} + \frac{\left(u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}\right)}{2} \right] \\
& - \frac{\lambda_x}{2} \left(v_{i+1,j}^{x,n} u_{i+1,j}^{n} - v_{i-1,j}^{x,n} u_{i-1,j}^{n} \right) - \frac{\lambda_y}{2} \left(v_{i,j+1}^{y,n} u_{i,j+1}^{n} - v_{i,j-1}^{y,n} u_{i,j-1} \right) \\
& + \frac{\lambda_x}{2} \left(|v_{i+1,j}^{x,n}| u_{i+1,j}^{n} - 2|v_{i,j}^{x,n}| u_{i,j}^{n} + |v_{i-1,j}^{x,n}| u_{i-1,j}^{n} \right) \\
& + \frac{\lambda_y}{2} \left(|v_{i,j+1}^{y,n}| u_{i,j+1}^{n} - 2|v_{i,j}^{y,n}| u_{i,j}^{n} + |v_{i,j-1}^{y,n}| u_{i,j-1}^{n} \right)
\end{aligned}
\tag{6.26}
$$

with boundary conditions (3.29) and stability conditions (3.25). Because we solve the forward problem on different grid levels, we need a proper definition

Definition 6.2.3 (Grid Levels)**.** *Let (6.26) be the numerical solution for the forward problem (6.25). Then we define the grid level* (g) *as the discretization of the domain* Ω *as* $\Omega^{(g)}$ *with mesh grid size* $\triangle x^{(g)} = 2^g \triangle x$ *and* $\triangle y^{(g)} = 2^g \triangle y$ *with grid level* $g = 0, \dots, G$*, where grid level* (0) *corresponds to the finest grid with* $\triangle x^{(0)} = \triangle x$ *and* $\triangle y^{(0)} = \triangle y$ *and grid level* (G) *corresponds to the coarsest possible grid with* $\triangle x^{(G)} = 2^G \triangle x$*,* $\triangle y^{(G)} = 2^G \triangle y$*.*
The corresponding discretized velocity field $\boldsymbol{v}$ *is thus defined as*

$$\boldsymbol{v}_{i,j}^{(g)} = \begin{pmatrix} v_{i,j}^{x,(g)} \\ v_{i,j}^{y,(g)} \end{pmatrix} \tag{6.27}$$

with indices $i = 0, \dots, N_x^{(g)} + 1$ *and* $j = 0, \dots, N_y^{(g)} + 1$ *according to the discretized domain* $\Omega^{(g)}$ *for grid level* g*.*
The same applies to the discretized solution $u\left(x_i^{(g)}, y_j^{(g)}, t_n\right) \approx u_{i,j}^{n,(g)}$ *with*

$$u^{n,(g)} := \left(u_{0,0}^{n,(g)}, \dots, u_{N_x+1,N_y+1}^{n,(g)}\right). \tag{6.28}$$

The superscript (g) *at indices such as* i, j *is omitted whenever a variable is denoted with the superscript* (g) *and only reintroduced when required.*
The solution of the forward problem with non-linear forward operator F *is consequently defined as*

$$F^{(g)}\left(\boldsymbol{v}^{(g)}\right) = u^{(g)}. \tag{6.29}$$

Inverse Problem

Next, we derive the inverse problem as the minimization problem for the parameter estimation.
Here we will consider the velocity field $\boldsymbol{v}$ as the unknown parameter with $\Theta = \boldsymbol{v}$ whereas the diffusion coefficient $D > 0$ will be assumed known.
We define the objective functional K as a Tikhonov-regularized non-linear least square functional

$$K^{(g)}\left(\boldsymbol{v}^{(g)}\right) := \| F^{(g)}\left(\boldsymbol{v}^{(g)}\right) - u_{data}^{(g)} \|_2^2 + \lambda^{(g)} \| \boldsymbol{v}^{(g)} \|_2^2 \tag{6.30}$$

with regularization parameter $\lambda^{(g)} \in \mathbb{R}_{>0}$ for Tikhonov matrix $T = I$ and observation data $u_{data}^{(g)}$ at grid level (g).

Remark 6.2.4. *We assume that the observation data* u_{data} *are defined over the whole domain* Ω *on the finest grid level* $g = 0$*. Values for coarser grid levels* g *are obtained by neglecting the in-between values.*

The inverse problem on the fixed grid level (g) is then defined as the non-linear minimization problem

$$\boldsymbol{v}^{(g)} = \arg\min_{\boldsymbol{v}^{(g)}} K^{(g)}\left(\boldsymbol{v}^{(g)}\right), \tag{6.31}$$

which then can be solved with the non-linear optimization methods presented in Chapter 5.
As it is, the inverse problem (6.31) can be used for the parameter estimation of $\boldsymbol{v}$ but does not benefit from the application of the non-linear multigrid approach which we discussed in Section 4.2.2. In the following we introduce the necessary multigrid components to apply the non-linear multigrid approach to the inverse problem.

Multigrid components

For the application we have already defined the grid refinement with $\Omega^{(g)}$ for grid levels $g = 0, \ldots, G$ in Definition 6.2.3.
For the non-linear multigrid method also appropriate restriction and prolongation operators are necessary to switch between grid levels appropriately:

- Forward problem: For the forward operator $F^{(g)}$ the switch to different grid levels g can be realized trivially by changing the mesh grid size $\triangle x^{(g)}$ and $\triangle y^{(g)}$ for the finite difference scheme (6.26) and does not require any special treatment.
- Discretized velocity field: For the discretized velocity field $\boldsymbol{v}^{(g)}$ which is defined at the nodes of the mesh grid $(x_i^{(g)}, y_j^{(g)})$ it is necessary to define an appropriate restriction operator $I_g^{g+1} : \Omega^{(g)} \to \Omega^{(g+1)}$ and prolongation operator $I_{g+1}^{g} : \Omega^{(g+1)} \to \Omega^{(g)}$.

In the following definitions we will use the stencil notation which illustrates matrix structures as discretization stencils. This notation allows us to represent the discretized operators in a more natural way where each element in the stencil form is placed according to its geometrical position. We refer to [138].

In the following we refer with $i^{(g)}$ and $j^{(g)}$ to the indices according mesh grid $\Omega^{(g)}$ and omit superscript (g) when it becomes clear from the context.

Restriction operator

Typical choices for a restriction operator $I_g^{g+1} : \Omega^{(g)} \to \Omega^{(g+1)}$ is the simple injection operator which only considers values that coincide at both grid levels g and $g + 1$.

Definition 6.2.5 (Injection operator)**.** *The injection operator* $I_g^{g+1} : \Omega^{(g)} \to \Omega^{(g+1)}$ *for* $\boldsymbol{v}^{(g)}$ *is defined in stencil form as*

$$I_g^{g+1} := \rceil \ 1 \ \lfloor \tag{6.32}$$

such that

$$\left(I_g^{g+1}\boldsymbol{v}^{(g)}\right)_{i^{(g+1)},j^{(g+1)}} = v_{i^{(g)},j^{(g)}} \tag{6.33}$$

However this loss of information results in worse approximation, thus using a weighted restriction operator that takes into account the adjacent points to form a mean value is more beneficial.

Definition 6.2.6 (Weighted restriction operator)**.** *The weighted restriction operator* $I_g^{g+1} : \Omega^{(g)} \to \Omega^{(g+1)}$ *for* $\boldsymbol{v}^{(g)}$ *is defined in stencil form as*

$$I_g^{g+1} := \rceil \begin{matrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{matrix} \lfloor \tag{6.34}$$

such that

$$\begin{aligned} \left(I_g^{g+1}\boldsymbol{v}^{(g)}\right)_{i^{(g+1)},j^{(g+1)}} = \tfrac{1}{16}\big(& v_{i^{(g)}-1,j^{(g)}-1} + 2v_{i^{(g)},j^{(g)}-1} + v_{i^{(g)}+1,j^{(g)}-1} \\ & +2v_{i^{(g)}-1,j^{(g)}} + 4v_{i^{(g)},j^{(g)}} + 2v_{i^{(g)}+1,j^{(g)}} \\ & +v_{i^{(g)}-1,j^{(g)}+1} + 2v_{i^{(g)},j^{(g)}+1} + v_{i^{(g)}+1,j^{(g)}+1}\big). \end{aligned} \tag{6.35}$$

The weighted restriction therefore carries more "information" from the finer grid to the coarser grid.

Prolongation operator

Similar to the weighted restriction operator (6.34) we define a prolongation operator $I_{g+1}^{g} : \Omega^{(g+1)} \to \Omega^{(g)}$ that is able to carry more information to the finer grid

Definition 6.2.7 (Linear interpolation prolongation operator)**.** *The two-dimensional linear interpolated prolongation operator* $I_{g+1}^{g} : \Omega^{(g+1)} \to \Omega^{(g)}$ *in stencil form is defined as*

$$I_{g+1}^{g} := \rceil \begin{matrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{matrix} \lfloor . \tag{6.36}$$

Smoothing

The last component for the non-linear multigrid method is an appropriate relaxation i.e smoothing operator

$$S^{(g)}\left(\boldsymbol{v}^{(g)}, K^{(g)}\right) \tag{6.37}$$

with an appropriate initial guess $\boldsymbol{v}^{(g)}$ which can be any non-linear optimization method such as regularized Gauß-Newton, non-linear conjugated gradient method or other line search methods.

Two-grid Approach

A non-linear multigrid method for two levels as in Section 4.2.2 Algorithm 4.0, consists of the following steps to perform a correction of $\boldsymbol{v}^{(g)}$ to the corrected $\widehat{\boldsymbol{v}}^{(g)}$:

- Initialization: Let $\boldsymbol{v}^{(g)}$ be the initial discretized velocity field for grid level (g).
- Restriction: $\boldsymbol{v}^{(g+1)} = I_g^{g+1}\boldsymbol{v}^{(g)}$.
- Smoothing: $\tilde{\boldsymbol{v}}^{(g+1)} = S^{(g+1)}\left(\boldsymbol{v}^{(g+1)}, K^{(g+1)}\left(\boldsymbol{v}^{(g+1)}\right)\right)$.
- Prolongation and Correction: $\widehat{\boldsymbol{v}}^{(g)} = \boldsymbol{v}^{(g)} + I_{g+1}^{g}\left(\tilde{\boldsymbol{v}}^{(g+1)} - \boldsymbol{v}^{(g+1)}\right)$.

Even though the application of the non-linear multigrid method with restriction (6.34), prolongation (6.36) and smoothing operator on the minimization problem (6.31) have been performed, two aspects must be ensured first: Consistency and convergence.
Consistency cannot be guaranteed without adjusting the objective functional $K^{(g)}$ (6.30) such that it consistent on every grid level (g).
For this reason we modify the original objective function $K^{(g)}$(6.31) with an additional term to ensure consistency:

$$\widehat{K}^{(g)} := \underbrace{\parallel F^{(g)}\left(\boldsymbol{v}^{(g)}\right) - u_{data}^{(g)} \parallel_2^2 + \lambda^{(g)} \parallel \boldsymbol{v}^{(g)} \parallel_2^2}_{=K^{(g)}\left(\boldsymbol{v}^{(g)}\right)} - r^{(g)}\boldsymbol{v}^{(g)} \tag{6.38}$$

with row vector $r^{(g)}$ to adjust the gradient of the new objective function $\widehat{K}^{(g)}$ and ensure consistency.

Condition 1: Consistency

In order to ensure consistency, we need certain requirements such that

$$\widehat{K}^{(g+1)}(\widehat{\boldsymbol{v}}^{(g+1)}) = \widehat{K}^{(g)}(\widehat{\boldsymbol{v}}^{(g)}) + constant \tag{6.39}$$

and need to adjust $u_{data}^{(g+1)}, \lambda^{(g+1)}$ and $r^{(g+1)}$ accordingly.
The assumption we made are the following:

- For the observation data $u_{data}^{(g+1)}$ we require that the initial deviation between the solution of the forward problem and the observation data is the same at the coarse and fine grids:

$$\begin{aligned} F^{(g+1)}(I_g^{g+1}\boldsymbol{v}^{(g)}) - u_{data}^{(g+1)} &= I_g^{g+1}\left(F^{(g)}(\boldsymbol{v}^{(g)}) - u_{data}^{(g)}\right) \\ \Leftrightarrow \quad u_{data}^{(g+1)} &= I_g^{g+1}u_{data}^{(g)} - \left[I_g^{g+1}F^{(g)}(\boldsymbol{v}^{(g)}) - F^{(g+1)}(I_g^{g+1}\boldsymbol{v}^{(g)})\right], \end{aligned} \tag{6.40}$$

  where the square bracket term compensates for the forward problem mismatch between grids.

- For the regularization parameter $\lambda^{(g+1)}$ we require that

$$\lambda^{(g+1)} \parallel I_g^{g+1}\boldsymbol{v}^{(g)} \parallel_2^2 = \lambda^{(g)} \parallel \boldsymbol{v}^{(g)} \parallel_2^2 \quad \Leftrightarrow \lambda^{(g+1)} = \frac{\parallel \boldsymbol{v}^{(g)} \parallel_2^2}{\parallel I_g^{g+1}\boldsymbol{v}^{(g)} \parallel_2^2}\lambda^{(g)}. \tag{6.41}$$

  Conditions (6.40) and (6.41) are especially important in regards to comparing the same inverse problem on fixed grids with different sizes.

- In order to find a condition for the vector $r^{(g+1)}$ we require that the gradient of the objective functional must be equal on different grid levels.

$$\nabla\widehat{K}^{(g+1)}(I_g^{g+1}\boldsymbol{v}^{(g)}) = \nabla\widehat{K}^{(g)}(\boldsymbol{v}^{(g)})I_{g+1}^{g} \tag{6.42}$$

  where the prolongation operator I_{g+1}^{g} actually plays the role of the restriction operator because it multiplies the gradient on the right [112]. The equality (6.42) can be enforced by choosing

$$r^{(g+1)} = \nabla K^{(g+1)}(I_g^{g+1}\boldsymbol{v}^{(g)}) - \nabla\widehat{K}^{(g)}(\boldsymbol{v}^{(g)})I_{g+1}^{g}. \tag{6.43}$$

  On the finest grid $g = 0$ this means $\widehat{K}^{(0)} = K^{(0)}$, thus $r^{(0)} = 0$.

Fulfilling conditions (6.40), (6.41) and (6.43) guarantees that the objective functional $\widehat{K}$ on different grid levels is equal within an additive constant (6.39). This ensures consistency.

Condition 2: Convergence

For convergence we require that the value of the objective functional $\widehat{K}^{(g)}$ for $\widehat{\boldsymbol{v}}^{(g)}$ is decreasing after the coarse grid correction on $\boldsymbol{v}^{(g)}$.

$$\widehat{K}^{(g)}(\widehat{\boldsymbol{v}}^{(g)}) \leq \widehat{K}^{(g)}(\boldsymbol{v}^{(g)}). \tag{6.44}$$

This is a typical condition for line search methods as they appear in non-linear optimization methods (5.11).
We refer to [20, 112] where sufficient conditions are being derived to guarantee convergence for the non-linear multi grid method as an optimization method.
The two-grid-level method for this parameter estimation problem can be extended to full V and W-cycles as described in Section 4.1.3 in the exact same way as for FAS methods (4.2.2), with the only modification of adjusting the objective functional dynamically accordingly to ensure consistency.

We conclude this section by stating that the results achieved in [91] shows that the application of multi grid methods does reduce computational time by more than 50% and at the same time improves the relative error compared to a fixed grid method.
We also want to remark that the approach of [91] can be modified and improved by choosing different regularization terms and using a Full multigrid method (FMG) (see Section 4.0), which can improve the initial guess.

6.2.4 Residual Constrained Parameter Estimation

Another approach for the parameter estimation of (6.8) is described in the paper of Nielsen et al. [110].
As in [91], the inverse problem is defined as a non-linear minimization problem with Tikhonov-regularized non-linear least square functional without the application of multi grid methods.
The significant difference between these two approaches consists in how the forward problem is being solved in [110]. Instead of calculating the solution of the forward problem (6.8) $F(\boldsymbol{q}) = u$ numerically with a finite difference scheme as in definition 6.1.2, it is computed by a constrained minimization problem where the distance between solution u and observation data u_{data} is being minimized under a constraint. This constraint is the discretized residual of (6.10). This approach eliminates the need of solving the forward problem with finite difference schemes entirely, and reduces the computational cost, but at the same time increases the number of unknown parameters that need to be estimated in the minimization procedure.

Definition 6.2.8 (Inverse Problem)**.** *Let the solution of the convection-diffusion equation (6.1) with initial and boundary condition (6.2), (6.3) be the forward problem. Furthermore, let the discretized residual $r_{i,j}^n$ be defined with the finite difference scheme from (6.26) as*

$$\begin{aligned} r_{i,j}^n(u) \;:=\; & \tfrac{u_{i,j}^{n+1}-u_{i,j}^n}{\triangle t} - D\tfrac{\mu_x}{\triangle t}\left[\tfrac{\left(u_{i+1,j}^n-2u_{i,j}^n+u_{i-1,j}^n\right)}{2} + \tfrac{\left(u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}\right)}{2}\right] \\ & -D\tfrac{\mu_y}{\triangle t}\left[\tfrac{\left(u_{i,j+1}^n-2u_{i,j}^n+u_{i,j-1}^n\right)}{2} + \tfrac{\left(u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}\right)}{2}\right] \\ & +\tfrac{\lambda_x}{2\triangle t}\left(v_{i+1,j}^{x,n}u_{i+1,j}^n - v_{i-1,j}^{x,n}u_{i-1,j}^n\right) + \tfrac{\lambda_y}{2\triangle t}\left(v_{i,j+1}^{y,n}u_{i,j+1}^n - v_{i,j-1}^{y,n}u_{i,j-1}\right) \\ & -\tfrac{\lambda_x}{2\triangle t}\left(|v_{i+1,j}^{x,n}|u_{i+1,j}^n - 2|v_{i,j}^{x,n}|u_{i,j}^n + |v_{i-1,j}^{x,n}|u_{i-1,j}^n\right) \\ & -\tfrac{\lambda_y}{2\triangle t}\left(|v_{i,j+1}^{y,n}|u_{i,j+1}^n - 2|v_{i,j}^{y,n}|u_{i,j}^n + |v_{i,j-1}^{y,n}|u_{i,j-1}^n\right). \end{aligned} \tag{6.45}$$

Then we define the inverse problem of the residual constrained parameter estimation as

$$\min \parallel u - u_{data} \parallel_2^2 \quad \textit{subject to } r_{i,j}^n(u) = 0. \tag{6.46}$$

We will now transform the constrained minimization problem (6.46) into an unconstrained problem by using the augmented Lagrangian method (5.37) with Lagrangian functional $\mathcal{L}_c$

$$\mathcal{L}_c(\boldsymbol{v}, u, \lambda) \;:=\; \parallel u - u_{data} \parallel_2^2 + \lambda \parallel T\boldsymbol{v} \parallel_2^2 + \sum_{i,j}\left(\lambda_{i,j}^n r_{i,j}^n + \frac{c}{2}\left(r_{i,j}^n\right)^2\right) \tag{6.47}$$

with Tikhonov matrix T, regularization parameter $\lambda > 0$, Lagrangian multiplier $\lambda_{i,j}^n \in \mathbb{R}$ and penalty constant $c > 0$.
The minimization problem not only needs to estimate the discretized velocity field function $\boldsymbol{v} = (\boldsymbol{v}^x, \boldsymbol{v}^y)$, but also the Lagrangian multiplier $\lambda_{i,j}^n$ and the discretized solution of the forward problem u as well.
Comparing the approach of Liu in Section 6.2.3 for a fixed grid size such that the discretized velocity field is of $v_{i,j}^x, v_{i,j}^y$ and $i = 0, \ldots, N_x+1$, $j = 0, \ldots, N_y+1$ means that instead of $2(N_x+2)(N_y+2)$ unknown parameters, the augmented Lagrangian method has additionally $M(N_x+2)(N_y+2)$ unknown parameters because of the Lagrangian multiplier $\lambda_{i,j}^n$ for each time steps t_n with $n = 0, \ldots, M-1$.
Although a multi grid approach is not used in [110], it reduces the number of unknown parameters by choosing a coarser representation by prolongation and interpolating the intermediate values of $\boldsymbol{v}$. The algorithm of solving the minimization problem (6.47) is:

1. Set initial values for all $(u)^{(0)}, \lambda_{i,j}^{(0),n} \in \mathbb{R}$ and $\boldsymbol{v}^{(0)}$
2. Solve the first minimization problem for $\boldsymbol{v}$

$$\boldsymbol{v}^{(k+1)} = \arg\min_{\boldsymbol{v}} \mathcal{L}(\boldsymbol{v}, u^{(k)}, \lambda^{(k)}). \tag{6.48}$$

3. Solve the second minimization problem for u

$$u^{(k+1)} = \arg\min_{u} \mathcal{L}(\boldsymbol{v}^{(k+1)}, u, \lambda^{(k)}). \tag{6.49}$$

4. Update Lagrangian multiplier

$$\lambda^{(k+1)} = \lambda^{(k)} + cr(u^{(k+1)}, \boldsymbol{v}^{(k+1)}). \tag{6.50}$$

5. Repeat minimizations with $k+1$.

Each minimization problem is solved with a non-linear optimization method. Nevertheless we want to note that both works [91, 110] are solving the same problem with similar data and obtain a similar quality of approximation. However, it is mentioned in Nielsen [110] that for convection dominant problems the algorithm encounters difficulties.

6.2.5 Time-varying Parameters

In both the multigrid- and residual constraint parameter estimation methods we discuss the parameter estimation of the velocity field function $\boldsymbol{v}(x, y)$ which is constant in time, and only vary in space. A similar setting is investigated in the original work [91, 110] for the permeability coefficient $q(x, y)$ in (6.8).
The discretization of $\boldsymbol{v}(x, y)$ with $\boldsymbol{v}^x_{i,j}, \boldsymbol{v}^y_{i,j}$ has been obtained according to the chosen mesh grid.
However the more general case for time-varying parameters has not been considered and will be presented in this section as an original contribution of this thesis.

Let the time-varying velocity field $\boldsymbol{v}$ be defined as

$$\begin{aligned} &v : \Omega^2 \times (0, T) \to \mathbb{R}, \\ &v(x, y, t) := \begin{pmatrix} v^x(x, y, t) \\ v^y(x, y, t) \end{pmatrix}. \end{aligned} \tag{6.51}$$

For a fixed spatial mesh grid (see definition 6.2.3) we obtain the discretized velocity field $\boldsymbol{v}^n_{i,j} \in \mathbb{R}^{2(N_x+2)\times(N_y+2)}$, for fixed time t_n.
This is the setting we have used for the multigrid parameter estimation method in Section 6.2.3 and the residual constraint parameter estimation method in Section 6.2.4, where the velocity field $\boldsymbol{v}$ only depends on space.
However, for time dependent velocity field $\boldsymbol{v}(x, y, t)$, solving the forward problem requires not only the discretized velocity field $\boldsymbol{v}_{i,j}$ in space but also in time $\boldsymbol{v}^n_{i,j}$.

Under these circumstances the number of unknown parameters increases by a factor of $M \in \mathbb{N}$ where M depends on the number of time step iterations the forward problem needs to perform.

In the following we will consider the inverse problem as defined in Definition 6.1.3 with

$$\arg\min \parallel F(\boldsymbol{v}) - u_{data} \parallel_2^2, \tag{6.52}$$

where the non-linear forward operator F is the numerical solution of the two-dimensional convection-diffusion equation (6.1) with the finite difference scheme (6.4) for a fixed mesh grid with $\triangle x, \triangle y > 0$ and fixed time step size $\triangle t > 0$ that respects the stability condition (3.25). Then the discretized velocity field is defined as $\boldsymbol{v}_{i,j}^n = \left(v_{i,j}^{n,x}, v_{i,j}^{n,y}\right)$ with $i = 0, \ldots, N_x+1$, $j = 0, \ldots, N_y+1$ and $n = 0, \ldots, M$ with $M = \frac{T}{\triangle t}$ and $t_M = T$.
We will neglect the superscript x (resp. y) that indicates the x- and y-component to make the reading easier and we will make distinction only when necessary.

In order to reduce the number of parameters $v_{i,j}^n$ we only choose a reduced number of time iterations $M^* < M$ which motivates the following methods as an original contribution of this thesis.

Temporal Multigrid Approach

The idea of the temporal multigrid approach (TMA) is based on the splitting of the time interval $\mathcal{T} := [0, T]$ into disjoint subintervals $\mathcal{T}_k := [t_k^*, t_{k+1}^*]$ for a time discretization with variable time step $\triangle t_k^*$ with $0 = t_0, t_{1^*}, \ldots, t_{M^*} = T$ with $M^* < M$.
If we choose a regular splitting with $\triangle t^* = 2^h \triangle t$ for $t_n^* = n^* \triangle t^*$ and $M^* = \frac{T}{\triangle t^*} = \frac{M}{2^h}$ for an integer $h > 0$, we obtain

$$\boldsymbol{v}_{i,j}^{n^*} = \boldsymbol{v}_{i,j}^{2^h n} \tag{6.53}$$

and consequently reducing the number of parameters $v_{i,j}^n$ by the factor of $\frac{1}{2^h}$. However, also non-equidistant subintervals $\mathcal{T}_k := [t_k^*, t_{k+1}^*]$ can be used with variable time step size $\triangle t_k^*$.

Remark 6.2.9. *Intuitively, in time intervals $[a, b]$ with larger time derivative $\frac{\partial \boldsymbol{v}}{\partial t}$, finer subintervals can be used, whereas in intervals with smaller time derivatives $\frac{\partial \boldsymbol{v}}{\partial t}$ coarser subintervals can be used because not much change is to be expected and thus not many parameters are needed for an accurate approximation.*
Nevertheless, this is only feasible if a-priori information about the time derivation $\frac{\partial \boldsymbol{v}(x,y,t)}{\partial t}$ of the velocity field is available.

Interpolation of intermediate values

Because the finite difference methods of the forward problem require also intermediate values of $\boldsymbol{v}_{i,j}^n$, where $t_n \notin \{t_0, \ldots, t_n^*, \ldots, t_M^*\}$, we need to interpolate them accordingly [29]. We obtain

$$v_{i,j}^n = s_n(t_n) \quad \text{for } t_n \in [t_k^*, t_{k+1}^*] \text{ with } t_k^* \leq t_n < t_{k+1}^*, \tag{6.54}$$

where s_n is a polynomial function. The simplest choice is a linear interpolation such that

$$\boldsymbol{v}_{i,j}^n = \boldsymbol{v}_{i,j}^{k^*} + \frac{\boldsymbol{v}_{i,j}^{k^*+1} - \boldsymbol{v}_{i,j}^{k^*}}{\triangle t_k^*}(t_n - t_k^*), \qquad \text{for } t_k^* < t_n < t_{k+1}^*. \tag{6.55}$$

We summarize the results in the following proposition.

Proposition 6.2.10. *Let the inverse problem of the convection-diffusion-equation be defined as in definition 6.1.3, for fixed meshgrid $\triangle x, \triangle y > 0$ in the time interval $\mathcal{T} := [0, T]$.*
Then we define with a variable time step size $\triangle t_k^x > 0$ the disjoint subintervals

$$\mathcal{T} = [0, T] = [t_0^*, t_1^*] \cup \cdots \cup [t_k^*, t_{k+1}^*] \cup \cdots \cup [t_{M^*-1}^*, t_{M^*}^*]. \tag{6.56}$$

Then the discretized velocity field $\boldsymbol{v}$ is defined as

$$v_{i,j}^n = \begin{cases} v_{i,j}^{n^*}, & \text{if } t_n \in \{t_0^*, \ldots, t_{M^*}^*\}, \\ s_{k^*}(t_n), & \text{if } t_n \notin \{t_0^*, \ldots, t_{M^*}^*\} \text{ for } t_k^* \leq t_n < t_k^*. \end{cases} \tag{6.57}$$

The approach of a coarser temporal representation of the discrete velocity field $\boldsymbol{v}$ is similar to the multigrid approach and allows us to reduce the effective parameters in the minimization methods and hence deliver more stable results as explained in [92] for the general multigrid approach.

We will demonstrate the temporal multigrid approach in the proceeding examples in Section 6.3.

6.2.6 Function Reconstruction

In the previous sections we applied the multigrid parameter estimation method to develop a robust algorithm for the parameter estimation of the diffusion coefficient D and velocity field $\boldsymbol{v}$, which can vary not only in space but also in time.
By using non-linear solvers as smoother in each grid level, we are able to converge towards local minima and with good initial guess obtain sufficient results for the parameter estimation.

Hitherto we assumed an arbitrary velocity field function $\boldsymbol{v}(x, y, t)$ for the convection-diffusion equation.
However sometimes additional knowledge of the velocity field can be used to further improve the parameter estimation process which depends on the form of the additional available information.
If the velocity field $\boldsymbol{v}$ is restricted through some constraint function

$$G(\boldsymbol{v}) = 0, \tag{6.58}$$

we can use a constrained minimization problem similar to the residual constrained parameter estimation method.
Another example is a parabolic reaction-diffusion equation for chemotaxis

$$\partial_t \phi \quad = \quad D_\phi \triangle \phi + \alpha u - \beta \phi \tag{6.59}$$

for the model equation

$$\partial_t u = D_u \triangle u - \operatorname{div}(\boldsymbol{v} u), \tag{6.60}$$

where the velocity field $\boldsymbol{v}$ is defined as the gradient function of the chemoattractant

$$\boldsymbol{v} = \nabla \phi \tag{6.61}$$

and the dynamics of ϕ determined by the parabolic equation (6.59).
There are three main approaches how the additional information can be included into the parameter estimation method:

1. Include ϕ in the forward and inverse problem, such that

   $$F(\Theta) = \begin{pmatrix} u \\ \phi \end{pmatrix}, \tag{6.62}$$

   where the model parameters $D_\phi, \alpha, \beta \in \mathbb{R}_{>0}$ are also being estimated with Θ.

2. Post-smoothing step: Include a second inverse problem to improve the estimate of the velocity field $\tilde{\boldsymbol{v}}$. The parameter estimation of $\boldsymbol{v}$ obtained through (6.7) will be used in a second inverse problem

   $$\arg\min \parallel H(\Theta_{\boldsymbol{v}}) - \boldsymbol{v} \parallel_2^2, \tag{6.63}$$

   where the non-linear forward operator H corresponds to the second parabolic equation of ϕ with model parameters $\Theta_{\boldsymbol{v}}$. In a second step the forward problem is then being solved with the estimated Θ_v to obtain a new corrected parameter estimate of $\boldsymbol{v}$.

3. Construct a constraint function

   $$G(\boldsymbol{v}) = 0 \tag{6.64}$$

   and solve the inverse problem as a constrained minimization problem.

The first approach is very simple to apply to the parameter estimation method, however it increases not only computational cost but also involves a greater number of parameters that need to be estimated.
This leads to more local minima and thus worsens the parameter estimation process.
The second approach however is independent of the parameter estimation and can be considered as a post-smoothing step, where the already acquired parameter estimate gets corrected or smoothed.
The last approach defines a constraint function G which then can be used in constrained minimization methods such as the penalty method (5.34).

Definition 6.2.11. *Let the inverse problem of the convection-diffusion equation be defined as in Definition 6.1.3. Furthermore let G be a constraint function of the velocity field $\boldsymbol{v}$ with*

$$G(\boldsymbol{v}) = 0, \tag{6.65}$$

then the modified non-regularized objective functional K_0 of the inverse problem is defined as

$$K_0(\Theta) := \| F(\Theta) - u_{data} \|_2^2 + c\left(G(\boldsymbol{v})\right)^2 \tag{6.66}$$

with penalty coefficient $c > 0$.

Definition 6.2.12. *Let the inverse problem of the convection-diffusion equation be defined as in Definition 6.1.3, ϕ be the solution of the parabolic reaction-diffusion equation (6.59) and the velocity field $\boldsymbol{v}$ defined as the gradient function of ϕ. Furthermore let $\boldsymbol{v}$ be the parameter estimate solution of the inverse problem of the convection-diffusion equation. Then we define the postsmoothing step with corrected velocity field $\tilde{\boldsymbol{v}}$*

$$\tilde{\boldsymbol{v}} = \arg\min \| H(\Theta_{\boldsymbol{v}}) - \boldsymbol{v} \|_2^2 \tag{6.67}$$

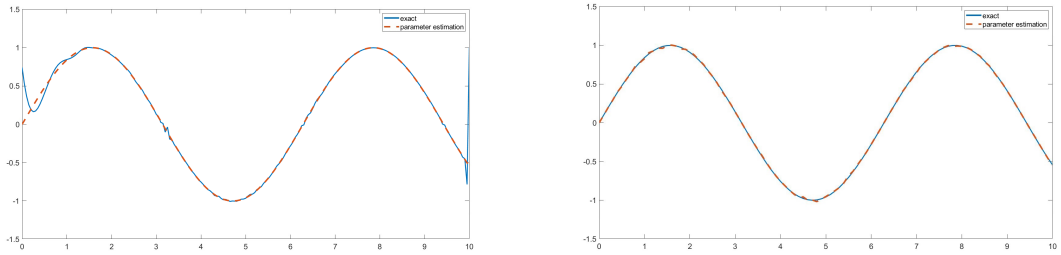
as the second inverse problem for the reaction-diffusion equation (6.59) with model parameters $\Theta_{\boldsymbol{v}}$, where H the function composition of a non-linear forward operator of the reaction-diffusion equation and a difference operator.

Example 6. *We demonstrate the post-smoothing effect on the one-dimensional convection diffusion equation, where the unknown velocity field $\boldsymbol{v}$ is estimated by solving the parameter estimation problem with objective functional*

$$K_0(\Theta) := \| F(\Theta) - u_{data} \|_2^2 \tag{6.68}$$

with $\Theta = \boldsymbol{v}$ and

$$K_0(\Theta) := \| F(\Theta) - u_{data} \|_2^2 + c\left(G(\boldsymbol{v})\right)^2 \tag{6.69}$$



(a) Velocity field v without post-smoothing step (b) Velocity field v with post-smoothing step

Figure 6.5: A comparison between the velocity field before and after the post-smoothing step. Left (orange): The oscillations which are very dominant at the borders and in areas with high second derivative. Right (blue): The post-smoothing has reduced some oscillations present at the borders.

with penalty parameter $c = 10^{-4}$ *and constraint function*

$$G(\boldsymbol{v}) := \sin(x_i) - v_i. \tag{6.70}$$

The effect the penalty constraint has on the parameter estimate solution $\tilde{v}$ *can be seen in Figure 6.5.*

Results: *The overall solution is improved by reduced oscillations, however the relative error* $err_{rel} = \frac{\|v-\tilde{v}\|_2}{\|v\|_2}$ *in the middle segment* $[3, 7]$ *has not improved with* $err_{rel} = 0.129$ *without the postsmoothing and with* $err_{rel} = 0.135$ *with postmoothing.*

6.3 Results on Parameter Estimation Methods

In the following section we will apply the previously presented and modified multigrid parameter estimation methods and the residual constraint parameter estimation method on several test equations to investigate their performance and robustness and compare the methods with each other.

6.3.1 Preparation

Before we present the results, we will give an overview of the several test equations and techniques used for the numerical experiments.

Test equations

In the following we will consider the following numerical tests:

1) Parameter estimation of the **one-dimensional convection-diffusion equation**

$$\begin{cases} \partial_t u &= D\partial_{xx} u - \partial_x (vu), \\ u_0(x) &= 10e^{-\frac{1}{2}(x-0.5)^2}, \\ \frac{\partial u}{\partial x}|_{\partial\Omega} &= 0 \end{cases} \tag{6.71}$$

with diffusion coefficient $D = 4$ and velocity field:

i) **Constant**: $\boldsymbol{v}$:=c=25.

ii) **Constant in time** t**, varying in space** x**:** $\boldsymbol{v} := c(x) = 15\sin(2\pi x)$

iii) **Varying in time** t **and space** x**:** $\boldsymbol{v} := c(x,t) = 15\sin(2\pi x - 4t)$.

iv) **Varying in time** t **and space** x **with constraint**: $\boldsymbol{v} := \nabla\phi(x,t)$ with

$$\begin{cases} \partial_t \phi &= 0.5\partial_{xx}\phi, \\ \phi_0(x) &= 15e^{-10(x-0.25)^2}, \\ \frac{\partial \phi}{\partial x}\big|_{\partial\Omega} &= 0. \end{cases} \tag{6.72}$$

2) Parameter estimation of the **two-dimensional convection-diffusion equation**

$$\begin{cases} \partial_t u &= D\Delta u - \operatorname{div}(\boldsymbol{v}u), \\ u_0(x,y) &= 10e^{-\frac{1}{2}\left((x-0.5)^2+(y-0.5)^2\right)}, \\ \frac{\partial u}{\partial \boldsymbol{n}}\big|_{\partial\Omega} &= 0 \end{cases} \tag{6.73}$$

with diffusion coefficient $D = 1$ and velocity field:

a) **Varying in space** (x,y):

$$v(x,y) := \begin{pmatrix} -10\,(x-0.5)\,e^{-5(x-0.5)^2-5(y-0.5)^2} \\ -10\,(y-0.5)\,e^{-5(x-0.5)^2-5(y-0.5)^2} \end{pmatrix}. \tag{6.74}$$

Parameter Estimation Methods

We apply the following parameter estimation methods on the test equations for the model parameters Θ with regularization functional $R(\Theta)$:

I) **Multigrid Parameter Estimation Method (MGPE)** (according to Section 6.2.3):

$$\widehat{K}^{(g)}(\Theta^{(g)}) := \| F^{(g)}\left(\Theta^{(g)}\right) - u_{data}^{(g)} \|_2^2 + R(\Theta^{(g)}) - r^{(g)}\Theta^{(g)}. \tag{6.75}$$

II) **Residual Constraint Parameter Estimation Method (RCPE)** (according to Section 6.2.4):

$$\mathcal{L}_c(\Theta, u, \lambda) \;:=\; \| u - u_{data} \|_2^2 + R(\Theta) + \sum_{i,j} \left(\lambda_{i,j}^n r_{i,j}^n + \frac{c}{2} \left(r_{i,j}^n \right)^2 \right). \tag{6.76}$$

III) **Maximum-Log-Likelihood Method (MLL)** (see [49])

$$\log \mathcal{L} := \sum_{i=1}^{n_p} \log \left(P(x_{i_d}, y_{j_d} | \Theta) \right) + R(\Theta) \tag{6.77}$$

with

$$P\left(x_{i_d}, y_{i_d} | \Theta\right) = \tfrac{F_{i_d,j_d}^{n_d}(\Theta)}{I^{n_d}}, \tag{6.78}$$

where $F_{i_d,j_d}^{n_d}(\Theta)$ is defined as the solution of the forward problem $F_{i_d,j_d}^{n_d}(\Theta) = u_{i_d,j_d}^{n_d}$ at data point $(x_{i_d}, y_{j_d}, t_{n_d})$ with model parameter Θ and $I^{n_d} := \triangle x \triangle y \sum_{i,j} u_{i,j}^{n_d}$.

For all three objective functions (6.75)-(6.77), we investigate the effects of the non-linear multigrid approach, which we derived and analysed in Section 6.2.3 for general objective non-linear functions, in regards to region of convergence, initial guess sensitivity, computational cost and robustness towards noise.

Regularization Term

For the regularization functional $R(\Theta)$ in (6.75) and (6.76) we have

$$R(\Theta) = b_D R_D(D) + b_v R_v(v) + b_{v_t} \sum_{i=0}^{N+1} R_{v_t}(v_i) + b_{v_x} \sum_{n=0}^{M} R_{v_x}(v^n) \tag{6.79}$$

with regularization parameters $b_D, b_{v_x}, b_{v_t} \in \mathbb{R}_{>0}$ and

$$\begin{aligned} R_D(D) &:= |D|, \\ R_{v_x}(v^n) &:= \mu \| T_x v^n \|_1 + (\mu - 1) \| T_x v^n \|_2^2, \\ R_{v_t}(v_i) &:= \mu \| T_t v_i \|_1 + (\mu - 1) \| T_t v_i \|_2^2, \\ R_v(v) &:= \mu \| v \|_1 + (\mu - 1) \| v \|_2^2 \end{aligned} \tag{6.80}$$

with appropriate Tikhonov matrices $T_x \in \mathbb{R}^{N+1 \times N+1}$ and $T_t \in \mathbb{R}^{M \times M}$ and elastic net parameter $\mu \in [0, 1]$.
In our setting we will use second order difference operator for the spatial discretization of the velocity field $\boldsymbol{v}^n \in \mathbb{R}^N$ and the first order difference operator for the time discretization of the velocity field $\boldsymbol{v}_i \in \mathbb{R}^M$. The regularization parameters are being obtained experimentally through MATLAB© and the L-Curve criterion.

Convergence Criterion

The convergence criterion is reached when

$$\| \Theta_{k+1} - \Theta_k \| \leq 10^{-3} \tag{6.81}$$

and we define the relative error as

$$\begin{aligned} err_D &:= \frac{\| D^{exact} - D^{opt} \|}{\| D^{exact} \|}, \\ err_v &:= \frac{\| v^{exact} - v^{opt} \|}{\| v^{exact} \|}. \end{aligned} \tag{6.82}$$

Artificial Data

For the artificially derived observation data u_{data} we apply a finite difference scheme with very fine mesh grid size $\triangle x, \triangle y > 0$ and time step size $\triangle t$ to have reasonable accurate observation data.
In order to replicate the situation of inherent noise, we add a Gaussian noise to the artificial data with

$$u_{data}^{\text{noise}} = u_{data} + \sigma u_{data} \Upsilon \tag{6.83}$$

with $\Upsilon \sim N(0,1)$ a normally distributed number with zero mean and standard deviation 1, whereas $\sigma \in [0,1]$ controls the noise percentage.
In order to have a reasonable comparison between the multigrid and residual constraint parameter estimation method with the maximum-likelihood method, we create microscopic data of sample size 1000 from the artificial data u_{data} with the acceptance-rejection method in Algorithm 7.3 which we present later in Chapter 7.1.

6.3.2 One-dimensional case

We begin by discretizing the one-dimensional convection diffusion equation (6.73) by the previously introduced finite difference scheme (3.22)

$$\begin{aligned} u_i^{n+1} &= u_i^n + D\mu_x \left[\tfrac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{2} + \tfrac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{2} \right] - \tfrac{\lambda_x}{2} \left(u_{i+1}^n v_{i+1}^n - u_{i-1}^n v_{i-1}^n \right) \\ &\quad + \tfrac{\lambda_x}{2} \left(|v_{i+1}^n| u_{i+1}^n - 2|v_i^n| u_i^n + |v_{i-1}^n| u_{i-1}^n \right) \end{aligned} \tag{6.84}$$

with $\lambda_x := \frac{\triangle t}{\triangle x}$ and $\mu_x := \frac{\triangle t}{\triangle x^2}$ with $\triangle x, \triangle t > 0$.
We also define several discretization grid refinements $\Omega^{(g)}$ for the later use of the multigrid approach and set $\Omega^{(0)}$ as the finest grid with $\triangle x_0 := \triangle x = 10^{-3}$ and the subsequently coarser grids $\Omega^{(g)}$ with $\triangle x_g := 2^g \triangle x$.

Furthermore we define the residual as

$$r_i^n := \frac{u_i^{n+1}-u_i^n}{\triangle t} - D\frac{\left[u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}+u_{i+1}^n-2u_i^n+u_{i-1}^n\right]}{\triangle x^2} + \frac{u_{i+1}^n v_{i+1}^n - u_{i-1}^n v_{i-1}^n}{2\triangle x} - \frac{|v_{i+1}^n|u_{i+1}^n - 2|v_i^n|u_i^n + |v_{i-1}^n|u_{i-1}^n}{2\triangle x}. \tag{6.85}$$

Regarding the observation data, we choose datasets $u_{data_i}^n$ for times $t_0 = 0$ to $t_M := 10$, and using equidistant time samples between $t = 0$ to $t = 10$.

6.3.3 i) Constant

For the constant case, the discretized velocity field is $v_i^n = c \in \mathbb{R}$ for all i and time steps n, which results in just two model parameters.
We choose $D = 4$ and $v = c = 25$ in order to have a high Peclet number, simulating a convection-dominated model as this is the case for chemotaxis models.
As initial guess for the optimization method we choose randomized values such that for each initial parameter guess

$$\Theta_0 \in [\Theta^{exact} - 2\Theta^{exact}, \Theta^{exact} + 2\Theta^{exact}]. \tag{6.86}$$

Results i)

The results are being collected in Figure 6.6, 6.7 and 6.8. We can deduce from results of 6.6 and 6.7 that for a fixed grid level MGPE recovers the parameters almost exactly with relative errors below 1%, even with 5% Gaussian noise, whereas RCPE performs in general slightly poorer with a higher relative error of the diffusion coefficient of 4.3% and even a higher relative error for the velocity field $\boldsymbol{v}$ with 3.5%, especially under Gaussian noise where the relative errors of both parameters are higher than 10%. This result reflects what was mentioned in [110] that the RCPE method performs less accurately and robustly for convection-dominant problems. The MLL however performs worse on the recovery of the velocity field $\boldsymbol{v}$ with a relative error of 12.7%.
The application of the non-linear multigrid to MGPE and RCPE improves the performance and as such they are able to recover both parameters with a sufficiently high accuracy, reducing the relative error of RCPE by a half whereas MLL fails to converge. This is due to the nature of the likelihood method by maximizing the density field in presence of a particle/cell [106]. Due to the coarser grid in the MLL approach, these positions are being distorted, hence the divergence. Also with Gaussian noise the MLL is not able to converge, meaning that imprecise cell position measurements in observation data lead to insufficient results.
In Figure 6.8 the influence of the initial guess is being analyzed.

It can be seen that MGPE for fixed grid level is the most stable method, allowing large initial guess intervals without significantly changing the relative error of the parameter estimates.
RCPE however is more sensitive to the initial guess, which results in bigger relative errors when the initial guess is further away from the exact parameters.
The MLL performs well only when the initial guess is close to the optimal value, giving similar relative errors as MGPE and RCPE. However, for initial guesses further away from the optimal value (in our example for $\Theta^{exact} \pm 10\Theta^{exact}$), MLL fails to converge to any solution.
Also here does the addition of the non-linear multigrid increases the interval of the initial guess for the parameter Θ_0 which is beneficial in situations where no prior knowledge is available about the parameters.
MGPE and RCPE do give the best results in performance and parameter recovery. Due to the insufficient performance of the maximum likelihood method, we will not consider it in the next numerical tests.

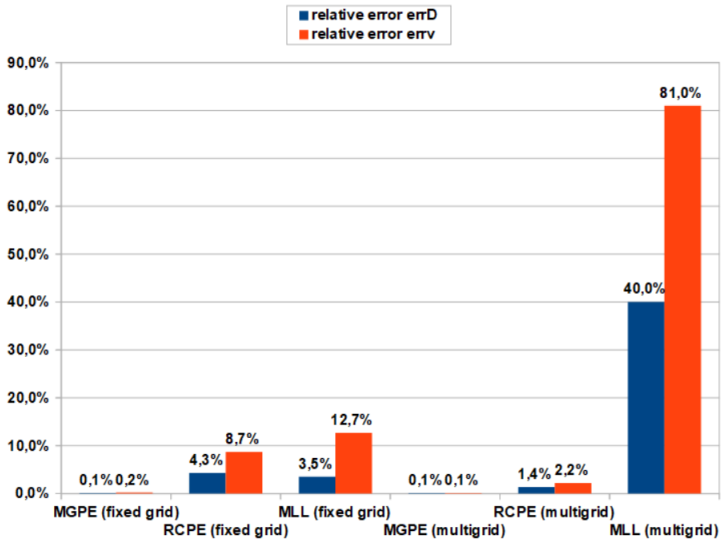


Figure 6.6: Parameter estimation with no noise. MGPE finds the optimal parameters whereas RCPE has slightly higher errors in both parameters, similar to MLL. The addition of multi grid improves RCPE but leads to false values for MLL.

6.3.4 ii) Constant in time t, varying in space x

The velocity field is now $v_i^n = c(x_i)$ and varies in space. We choose for our investigation $c(x) = 15\sin(2\pi x)$.

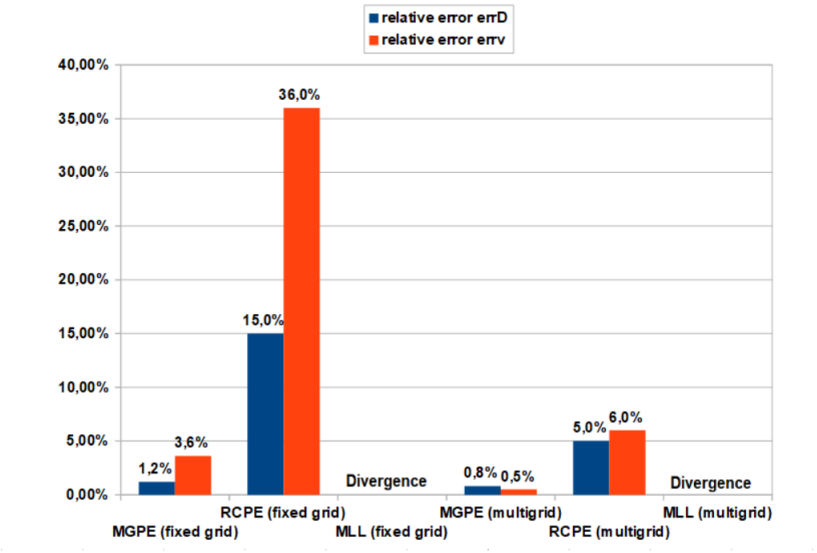


Figure 6.7: Parameter estimation with 5% Gaussian noise.

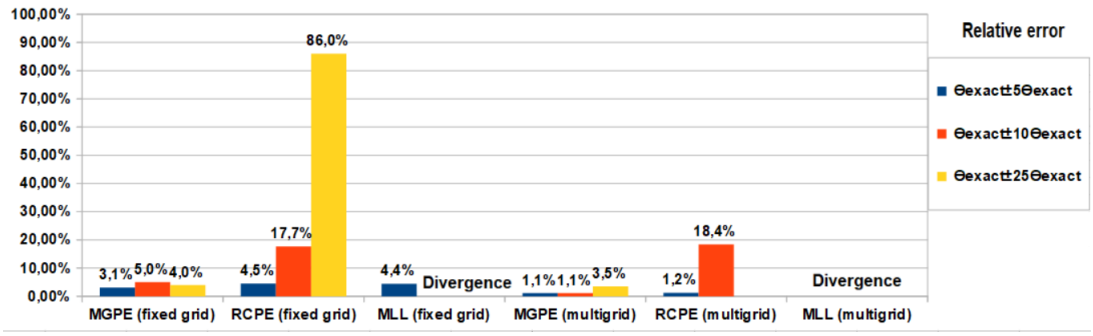


Figure 6.8: Relative error of Θ with observation data u_{data} with 2% Gaussian noise. As initial guesses we have chosen Θ_0 randomly within three intervals around the exact solution $[\Theta - 5\%\Theta, \Theta + 5\%\Theta]$, $[\Theta - 10\%\Theta, \Theta + 10\%\Theta]$, $[\Theta - 25\%\Theta, \Theta + 25\%\Theta]$.

Results ii)

Considering the results from Figure 6.9, 6.10 and 6.11, we see similar good results for MGPE. The relative error has increased to 5.2% and 8.1% compared to the constant velocity field with relative errors of 0.1% and 0.2% and the results are improved by the addition of the non-linear multigrid. This is due to the fact that on a coarser grid, less parameters of the velocity field are needed, reducing the number of local minima as well. RCPE fails to converge for fixed grid level for velocity fields v that are dependant on space, leading to no reasonable parameter estimations.
The application of multigrid to RCPE helps to ensure convergence although the errors are quite large with 8.4% and 11.4% compared to the MGPE which delivers parameter estimates with lower relative error already on a fixed grid

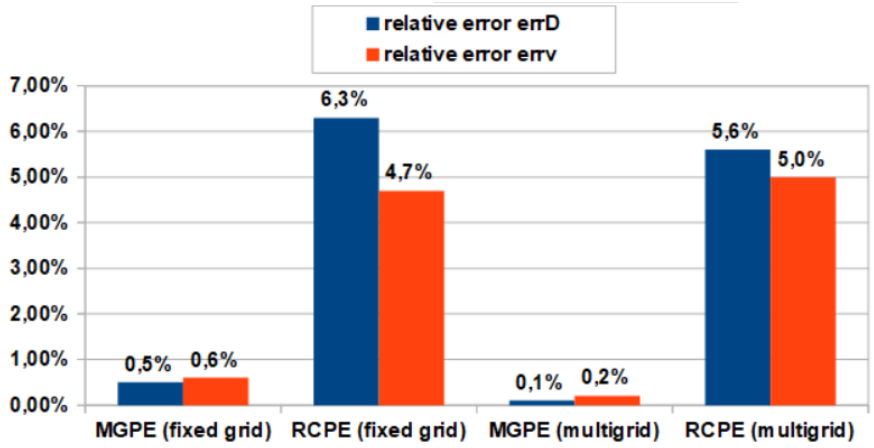


Figure 6.9: Parameter estimation with no noise for space-varying velocity field.

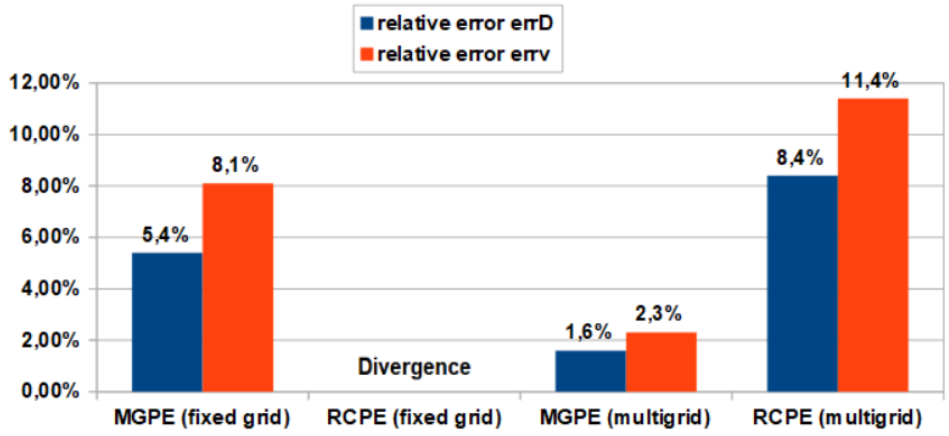


Figure 6.10: Parameter estimation with 5% Gaussian noise for space-varying velocity field.

level.
As for the initial guesses, still convergence is guaranteed with MGPE for both fixed grid level and multi grid, although the relative error increases the further the initial guess is from the actual value from 4.5% to 8% for the fixed grid level and from 2% to 5.5% for multi grid.
Since RCPE already shows to fail for non-constant velocity fields, we will proceed with the remaining numerical tests only for the MGPE.

6.3.5 iii) Varying in time t, varying in space x

In this case, we deal with velocity fields of the type $v_i^n = c(x_i, t_n)$ and choose $c(x,t) = 15\sin(2\pi x - 4t)$.
The difficulty here is the fact that the velocity field changes for each time iteration, increasing the number of parameters. For this reason we apply the temporal multi grid approach presented in Proposition 6.2.10. We will

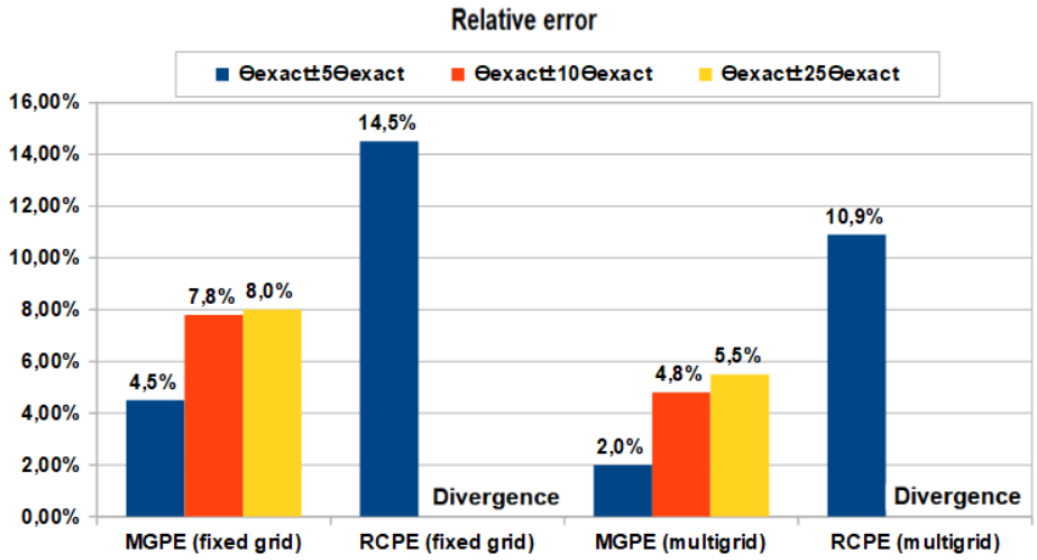


Figure 6.11: Relative error of Θ with observation data with 2% Gaussian noise and different initial guesses Θ_0 for space-varying velocity field.

investigate the following two approaches consisting in

1. Choosing for time data set $u_{i_d}^{n_d}$ with $n_d = 0, \dots, 100$ the equivalent number of velocity field parameters $v_i^{n_d}$.

2. Using a reduced number for the time discretization of $v_i^{\widehat{n}}$ with $\widehat{n} < n_d$ and interpolating the other values linearly as proposed in proposition 6.2.10.

The results are shown in Figure 6.12.

Results iii)

Comparing the two approaches through Figure 6.12 shows that in overall the estimated parameters have a higher relative error compared to the constant-in-time velocity field. However, by reducing the number of parameters by the factor 10, and interpolating the in-between values of the velocity field in time, we improve the overall performance reducing the relative error in both parameter estimates by half. This is due to the fact that a reduced number of parameters leads to a reduction of local minima, allowing better results during the optimization.

6.3.6 iv) Varying in time t and space x with constraint

In this test case, the velocity field $\boldsymbol{v}$ is defined through the partial differential equation (6.72) where the initial condition is known but not the diffusion

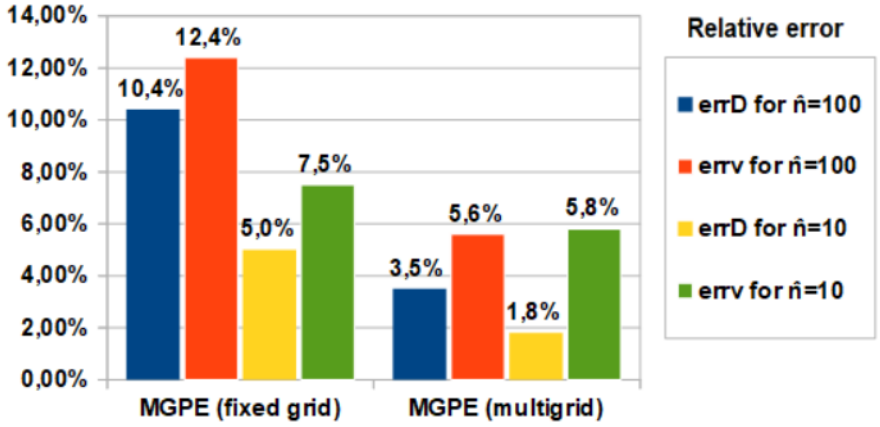


Figure 6.12: Parameter estimation with no noise for space-time-varying velocity field for $\widehat{n} = n_d = 100$ and $\widehat{n} = n_d/10 = 10$.

coefficient D_ϕ.
Here we apply first the MGPE on a fixed grid but afterwards apply the post-smoothing step as proposed in proposition 6.2.12 after the initially applied MGPE.

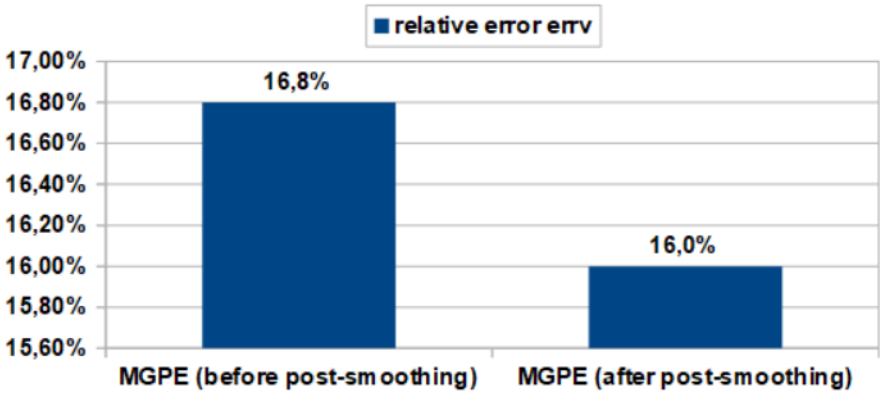


Figure 6.13: Relative error of the velocity field before and after postsmoothing.

Results iv)

In Figure 6.13 the difference between MGPE without post-smoothing and with post-smoothing step is shown. Due to the post-smoothing step the relative error of the velocity field v slightly reduced from 16.8% to 16%.
Although this improvement is not significant large, it has managed to further improve the parameter estimation although it has involved solving a

second inverse problem. However the trade-off between computational cost and relative error reduction strongly depends on the problem for which the parameter estimation method is applied on.

Remark 6.3.1. *We want to mention at this point that it is possible to further improve the results by consecutively applying the MGPE with post-smoothing step in order to obtain a first estimate for the diffusion coefficient $\tilde{D}$ and an improved estimate velocity field $\tilde{v}$ due to the postsmoothing step, and continue the MGPE with post-smoothing step to improve the overall results of post of both estimates, $\tilde{D}$ and $\tilde{v}$. However, because of the additional computational time, only one application of the MGPE with post-smoothing step has been used.*

6.3.7 Conclusion

We conclude the one-dimensional case with the following observations. Overall RCPE and MLL underperform and often diverge in presence of noise and convection-dominant convection diffusion equations. MGPE is able to recover the parameters with a sufficiently high accuracy and with the additional application of non-linear multigrid, MPGE improves in robustness, accuracy and has an increased initial guess interval to ensure convergence towards the local minima. The addition of a postsmoothing step within the non-linear multigrid method improves the overall accuracy of the velocity field only slightly but more importantly dampens otherwise present oscillations.

6.3.8 Two dimensional case

For the two dimensional case we continue in a similar way as for the one-dimensional case in Section 6.3.2.
In the following computations we will consider MGPE, since both RCPE and MLL are working poorly for convection dominated models for the one-dimensional case and consequently are not suited for the two-dimensional case which is more complex.
For the two-dimensional convection-diffusion model (6.73) we use the finite

difference scheme (3.24)

$$\begin{aligned} u_{i,j}^{n+1} &= u_{i,j}^{n} \\ &+D\mu_x \left[\frac{\left(u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}\right)}{2} + \frac{\left(u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}\right)}{2} \right] \\ &+D\mu_y \left[\frac{\left(u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}\right)}{2} + \frac{\left(u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}\right)}{2} \right] \\ &-\tfrac{\lambda_x}{2}\left(v_{i+1,j}^{x,n}u_{i+1,j}^{n} - v_{i-1,j}^{x,n}u_{i-1,j}^{n}\right) - \tfrac{\lambda_y}{2}\left(v_{i,j+1}^{y,n}u_{i,j+1}^{n} - v_{i,j-1}^{y,n}u_{i,j-1}\right) \\ &+\tfrac{\lambda_x}{2}\left(|v_{i+1,j}^{x,n}|u_{i+1,j}^{n} - 2|v_{i,j}^{x,n}|u_{i,j}^{n} + |v_{i-1,j}^{x,n}|u_{i-1,j}^{n}\right) \\ &+\tfrac{\lambda_y}{2}\left(|v_{i,j+1}^{y,n}|u_{i,j+1}^{n} - 2|v_{i,j}^{y,n}|u_{i,j}^{n} + |v_{i,j-1}^{y,n}|u_{i,j-1}^{n}\right) \end{aligned} \tag{6.87}$$

for the discretized domain $\Omega := [0,1] \times [0,1]$ with $\triangle x, \triangle y, \triangle t > 0$.

As for the velocity field $\boldsymbol{v} = (v^x, v^y)$, we now have the vectors $v^n := (v_{i,j}^{x,n}, v_{i,j}^{y,n})_{i,j}$ for $i = 0, \ldots, N_x + 1$, $j = 0, \ldots, N_y + 1$ for fixed time iteration n and vectors $v_{i,\cdot}^{x,n} = (v_{i,j}^{x,n})_i$, $v_{i,\cdot}^{y,n} = (v_{i,j}^{y,n})_i$ for fixed j and n with $i = 0, \ldots, N_x + 1$ and respectively vectors $v_{\cdot,j}^{x,n} = (v_{i,j}^{x,n})_j$, $v_{\cdot,j}^{y,n} = (v_{i,j}^{y,n})_j$ with fixed n and i for $j = 0, \ldots, N_y+1$.

The objective functional $K(\Theta)$ of MGPE is extended for the two-dimensional case with regularization term

$$\begin{aligned} R(\Theta) &= b_D R_D(D) + b_v R_v(v) + b_{v_t} \sum_{n=0}^{M} R_{v_t}(v^n) \\ &\quad + b_{v_x} \sum_{i=0}^{N_x+1} \sum_{j=0}^{N_y+1} R_{v_x}(v_{i,j}^{x,n}) + b_{v_y} \sum_{i=0}^{N_x+1} \sum_{j=0}^{N_y+1} R_{v_y}(v_{i,j}^{y,n}) \end{aligned} \tag{6.88}$$

and regularization parameters $b_D, b_{v_x}, b_{v_y}, b_{v_t} \in \mathbb{R}_{>0}$ and

$$\begin{aligned} R_D(D) &:= |D|, \\ R_{v_x}(v^n) &:= \mu \parallel T_x v^n \parallel_1 + (\mu - 1) \parallel T_x v^n \parallel_2^2, \\ R_{v_y}(v^n) &:= \mu \parallel T_y v^n \parallel_1 + (\mu - 1) \parallel T_y v^n \parallel_2^2, \\ R_{v_t}(v_{i,j}) &:= \mu \parallel T_t v_{i,j}^n \parallel_1 + (\mu - 1) \parallel T_t v_{i,j}^n \parallel_2^2, \\ R_v(v) &:= \mu \parallel v \parallel_1 + (\mu - 1) \parallel v \parallel_2^2 . \end{aligned} \tag{6.89}$$

We remark that in the two-dimensional case, the parameter estimation is much more sensitive with respect to the initial guess. For values further away from the optimal values, the parameter estimation method leads to divergence in most cases, which also highly depends on the observation data used for the parameter fitting.

Compared to the one-dimensional parameter estimation problems, the number of parameters are squared which in consequence increases the number of local minima and hence the sensitivity to initial guesses.
A stronger regularization does increase the chances of convergence but distorts all the other estimated parameters, leading to much higher relative errors.

Results a)

The results for the case a) are reported in Figure 6.14.

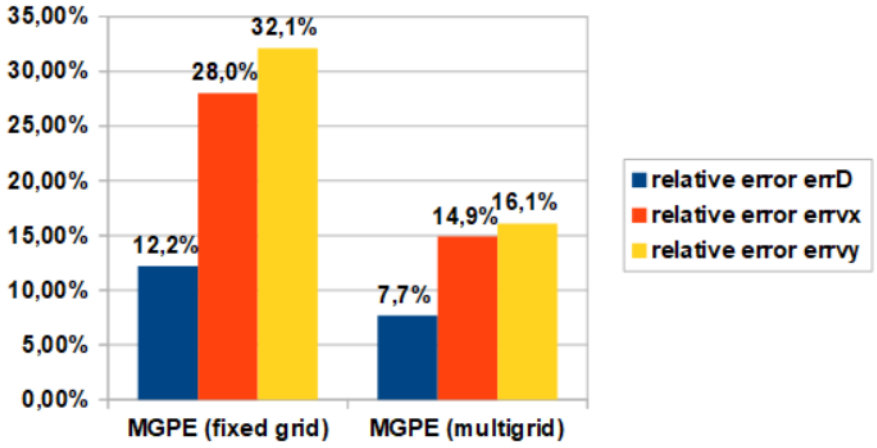


Figure 6.14: a) Parameter estimation with 1% Gaussian noise for space-varying velocity field.

We can deduce that the parameter estimation is able to recover reasonable parameter values with a low relative error of around under 12.2% for the diffusion coefficient but higher relative errors of 28% and 32.1% for the velocity field, which is much higher than for the one-dimensional case due to the squared number of unknown parameters of the velocity field.
However, also here the application of multigrid reduces the relative error of all parameter estimates by half with 7.7% relative error for the diffusion coefficient and 14.9% and 16.1% for the velocity field.

Remark 6.3.2. *In the previous numerical tests for the parameter estimation, the relative errors have been reduced, in average, by a factor of two, due to the application of multi grid methods.*
We noticed during the simulations that by using three grid levels $g = 0, 1, 2$ *optimal parameter values were obtained with minimal relative errors. Using coarser grids, i.e. applying the multi grid method beyond grid level* $g = 2$

does not improve the parameter estimates. However the computational time can be reduced by around 25% *for the two-dimensional parameter estimation and by around* 10% *for the one-dimensional parameter estimation.*

Remark 6.3.3. *For the parameter estimation of the two-dimensional convection diffusion equation (6.73) we were only able to obtain satisfying results with small relative errors for the case of space-varying velocity fields* $\boldsymbol{v}(x, y)$*. Introducing space- and time-varying velocity fields* $\boldsymbol{v}(x, y, t)$ *in most cases has led to divergence or solutions with high relative errors of at least* 50%*. This is caused mostly by the highly increased number of parameters which are needed when the velocity field depends on time.*

In the next part of the thesis, we will finally prepare the parameter estimation of the full laboratory model (1.45)-(1.47) with the data available, i.e. the positions of tumour cells and immune cells.
For that we need to find means to transform these data in order to be compatible with our mathematical model.

Part III

Parameter Fitting of Organ on Chip(OOC) Model with Real Data

This part of the thesis is dedicated to the parameter fitting of the multi-domain transmission model for Organs-on-Chip against real data.
Having derived a mathematical model of the OOC (1.45)-(1.47) based on the laboratory experiment [31] conducted on a microfluidic chips, we are now interested in calibrating our model based on the data acquired during the experiments.
We have already presented with the multigrid parameter estimation algorithm a robust and reliable method to recover model parameters from observation data. However, the observation data used during the parameter estimation were macroscopic data in form of density in which also our model is defined, whereas the data of the laboratory experiment are only available as microscopic observation data in form of cell trajectories and positions.
In order to use such microscopic data in our macroscopic model, we will introduce the kernel density estimate [147], which allows a non-parametric transformation of the cell trajectories into density without a-priori knowledge about the underlying density function.
The transformation between microscopic to macroscopic data through the kernel density estimation will be used to extend the framework of the multigrid parameter estimation method to work on microscopic observation data reliably. After numerical tests to validate the robustness and sufficient performance of the method, we proceed with the calibration of the multi-domain transmission model for the Organ-on-Chip based on the data obtained from the laboratory experiment in [15, 31].
Due to the limited access of data available to fit the model, we need to make additional simplifications based on assumptions about the experiment to deal with the lack of any data regarding the chemical concentrations and gradients of chemoattractant and cytokine.
This part is structured in the following way.
In the first two sections we present methods to create microscopic data from macroscopic data artificially and with the introduction of kernel density estimation method, we present a non-parametric transformation that can transform microscopic data back into macroscopic data. We then proceed with including the kernel density estimation method into the multigrid parameter estimation framework in order to obtain a method that is able to recover model parameters from microscopic data.
After numerical tests to verify the robustness of the method, we proceed with the model calibration of the OOC model against the data obtained from the laboratory experiment. We propose several model preparations and simplifications in order to calibrate our model well enough to reproduce the experiment conducted on the microfluidic chip and conclude this part with a comparison between real data from the experiment and from our model.

Chapter 7

Density-Particle Estimation

This last chapter before the main numerical simulation of the OOC-full model is dedicated to the proper transformation of the available observation data with the purpose of using the parameter estimation methods presented in the previous chapter and recover the intrinsic model parameters.
The mathematical model of the OOC (1.45)-(1.47), constructed in chapter 1.5, the numerical method to derive an approximated solution in chapter 3 and the multigrid parameter estimation method to recover the model parameters in Chapter 6.2.3 are all defined for macroscopic data in form of density quantities for the cells and chemical substances.
For this reason the observation data u_{data} used in the parameter estimation process need to be in the same form, i.e. must describe the density for a given time t. Otherwise it would not be possible to recover feasible model parameters that are specific to the given dynamics within our framework.
Unfortunately, in the laboratory experiments [31] and the observation data accession through TrackMate© [137], only the trajectories for a limited number of cells are available to us as explained in Section 1.1.
Such microscopic observation data are not conform with the macroscopic model that we derived and used to mathematically describe the dynamical behaviour of the OOC.
In consideration of this issue, either the mathematical model must be described in a microscopic way which involves stochastic partial differential equation theory [93] or the microscopic observation data must be transformed into macroscopic observation data. The latter choice is the one we discuss further in detail in the upcoming sections.

In the following sections we will first describe the numerical practice to create microscopic data according to a given density field. This is especially important to generate microscopic observation data artificially that hereinafter can be used to test the numerical quality and stability of the transformation

of the given microscopic observation data into their macroscopic equivalent, which we present in the second section of this chapter.
Lastly, an extension of the numerical simulations for the parameter estimation of the convection-diffusion equation given in chapter 6 is going to be created that involves the additional step of microscopic-macroscopic data transformation and compare its result.

7.1 Density to Particle Transformation

Creating macroscopic observation data of general functions $F(x)$ or probability density functions $f(x)$ artificially is a simple task [134]. One just need to choose points $(x_i)_{i=1,...,}$ to evaluate $F(x_i)$ (resp. $f(x_i)$) and obtain the desired data.
If a closed analytical form of the solution is not available, which is often the case for partial differential equations [30], we can use finite difference methods and calculate approximation with high accuracy by using high order methods or very small time steps $\triangle t$ and mesh grid size $\triangle x, \triangle y$ to obtain artificial observation data. However, we are only able to obtain artificial data of the same type as the functions we used to derive them. For probability density functions it is the probability and for partial differential equations it is the macroscopic quantity used in the partial differential equation.
If we want to derive microscopic data in such a way, we would need the function to be defined for such quantities.
For the convection-diffusion equation this can be done if we view the partial differential equation as a stochastic differential equation, where each random movement of a microscopic particle can be described as a stochastic process, such as the random walk, where at each step the particle moves to a certain direction according to a given probability [93]. In the limit of the number of particles N, one can obtain the partial differential equation, where the quantities are not the trajectories, i.e. location, of each particle any more but a probability density which indicates the probability that a particle is located at a certain point in space. We call the samples that are distributed according to a given probability distribution also random numbers.
The most common examples are uniform distributed random numbers according to the uniform distribution.

Definition 7.1.1. *A random variable on a probability space $(\Omega, \mathcal{F}, P)$ is a function*

$$X : \Omega \to \mathbb{R}, \tag{7.1}$$

that assigns a real number to each element in the sample space of a random experiment.

A random variable X is called continuous is possible values comprise either a single interval (either finite or infinite) of real numbers or a union of disjoint intervals. Usually random variables are represented with capital letters and specific values of a random variable x are represented with small letters and are also called random variates/random numbers.

Definition 7.1.2. *A function f with*

$$f : \mathbb{R} \to \mathbb{R} \tag{7.2}$$

is called probability density function of a continuous random variable X such that

$$\begin{cases} f(x) & \geq & 0, \\ \int_{-\infty}^{\infty} f(x)\,dx & = & 1, \\ P(a \leq X \leq b) & = & \int_a^b f(x)dx. \end{cases} \tag{7.3}$$

Definition 7.1.3. *We say that a random variable X has uniform distribution on the interval $[a, b]$, if it has the probability density function f given by*

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{for } a \leq x \leq b \\ 0, & \text{else.} \end{cases} \tag{7.4}$$

We write $X \sim U[a, b]$. The probability of a random number to be at a specific point in the interval $[a, b]$ is equally distributed.

Definition 7.1.4. *A random variable X is Gaussian, also known as normal, with parameters $\mu \in \mathbb{R}$ (mean value) and $\sigma^2 > 0$ (variance) if its probability density function is as follows*

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{(x-\mu)}{\sigma}\right)^2}. \tag{7.5}$$

We write $X \sim N(\mu, \sigma^2)$.

The computation of such uniform distributed random numbers can be done numerically and is called uniform pseudo-random number sampling [133]. Pseudo-random in the sense that the computed numbers are not truly random since they are determined completely by an initial value, the seed.
This gives rise to non-uniform pseudo-random number sampling to generate pseudo-random numbers that are distributed not according to a uniform probability distribution (7.4) but to a different probability distribution.

The procedure to generate such numbers consists in transforming a sequence of in dependent uniformly distributed random numbers on $[0, 1]$ into a sequence of independent random numbers of a different probability distribution.
The algorithms can be categorized in the following [73]:

- Inversion methods.
- Rejection methods.
- Composition methods.
- Methods specifically designed for a certain probability distribution.

The inversion method requires the additional computation of the cumulative distribution function $F(x) := \int_{-\infty}^{x} f(z)dz$ and composition methods are based on the assumption that the probability distribution f can be represented as a decomposition of several other probability distributions.
Thus, in this thesis we only present the acceptance-rejection method that has been used to generate all the artificial microscopic observation data for the numerical simulations.
It is a method that can be applied to all kinds of distributions in $\mathbb{R}^m$ with a probability density and as such is a standard algorithm to generation samples from non-uniform uni- and multivariate distributions.
The idea of the acceptance-rejection method is that if a point $(X, Y) \in \mathbb{R}^2$ is uniformly distributed in the domain between the image of the density function $\mathrm{im}(f)$ and the x-axis, then X is distributed according to f.
The theoretical foundation for the acceptance-rejection method is given by the following theorem.

Theorem 7.1.5 ([73])**.** *Let $f(x)$ be a density function with a constant $\alpha > 0$. If X and Y are uniformly distributed on*

$$G_{\alpha,f} := \{(x, y)|0 < y \leq \alpha f(x)\}, \tag{7.6}$$

then X is a random variable with density $f(x)$.

The acceptance-rejection method described in Algorithm 7.1 can be easily generalized to higher dimensions (see Algorithm 7.1 for 2D) . However there is an exponential increase in volume associated with increasing the spatial dimension, which results in much more sample rejections. Since we only work in one and two-dimensions, this does not affect us much.

As stated this is the acceptance-rejection method in its simplest form by using uniformly distributed random variables to generate random variates according to a given density $f(x)$. Since the enveloping function is the uniform hat

Algorithm 7.1
1D-Acceptance-Rejection-Method

procedure acceptanacerejection1D ($f : [a, b] \to \mathbb{R}_{>0}$, sample size N)
  Set $\alpha > 0$ as upper bound: $\alpha \geq f(x)$, $\forall x \in [a, b]$
  Set $\mathcal{X} := \{\}$
  $counter = 0$
  while $counter < N$ **do**
    Generate random variate $x \sim U(a, b)$
    Generate random variate $y \sim U(0, 1)$
    if $y \leq \alpha f(x)$ **then**
      Add x to the sample set $\mathcal{X}$
      $counter = counter + 1$
    end if
  end while
end procedure

Algorithm 7.2
2D-Acceptance-Rejection-Method

procedure acceptanacerejection2D($f : [a, b] \times [c, d] \to \mathbb{R}_{>0}$, sample size N)
  Set $\alpha > 0$ as upper bound $\alpha \geq f(x, y)$, $\forall x \in [a, b], y \in [c, d]$
  Set $\mathcal{X} := \{\}$
  $counter = 0$
  while $counter < N$ **do**
    Generate random variate $x \sim U(a, b)$
    Generate random variate $y \sim U(c, d)$
    Generate random number $z \sim U(0, 1)$
    if $z \leq \alpha f(x, y)$ **then**
      Add (x, y) to the sample set $\mathcal{X}$
      $counter = counter + 1$
    end if
  end while
end procedure

function, many of the computed sample points will be rejected. Choosing different enveloping functions can improve the generation speed but we will not further discuss these details and refer to [73, 133] where not only improvements of the rejection methods are stated but also other methods presented.

Acceptance-Rejection method for partial differential equations

With regards to partial differential equations such as the convection-diffusion equation where the analytical solution is not available and must be calculated numerically with a finite difference scheme to obtain the approximation at discrete points (x_i, y_j) we can either use interpolation method [29] to obtain a continuous density function $f(x, y)$ or a discrete version of the rejection method.
For the latter a small change must be made with the choice of the random variables X, Y since we only allow values at discrete points where the finite difference scheme was defined on.

Theorem 7.1.6 ([73])**.** *Let u^n be the discrete values of a density field, i.e $u^n_{i,j} \geq 0$ of a partial differential equation on the domain $\Omega = [a, b] \times [c, d]$, obtained through a finite difference schemes and discretized domain with $x_i := a + i\triangle x$, $y_j := c + j\triangle y$ with $i = 0, \ldots, N_x + 1$, $j = 0, \ldots, N_y + 1$, such that $x_{N_x+1} = b$ and $y_{N_y+1} = d$.*
If X and Y are distributed according to a discrete uniform distribution with $X \sim U\{0, N_x + 1\}$, $Y \sim U\{0, N_y + 1\}$ and Z uniform distributed $Z \sim U(0, 1)$ on

$$G_{\alpha,u^n} := \left\{(i, j, z)\,|0 < z \leq \alpha u^n_{i,j}\right\}, \tag{7.7}$$

then (X, Y) is a random variable with density u^n.

Algorithm 7.3 is based on theorem 7.1.6.

Algorithm 7.3
discrete Acceptance-Rejection-Method

procedure discreteacceptanacerejection2D (discrete density field u^n, sample size N)
  Set $\alpha := \max_{i,j} u^n_{i,j}$
  Set $\mathcal{X} := \{\}$
  $counter = 0$
  while $counter < N$ **do**
    Generate random variate $i \sim U\{0, \ldots, N_x + 1\}$
    Generate random variate $j \sim U\{0, \ldots, N_y + 1\}$
    Generate random variate $z \sim U(0, 1)$
    if $z \leq \alpha u^n_{i,j}$ **then**
      Add (x_i, y_j) to sample set $\mathcal{X}$
      $counter = counter + 1$
    end if
  end while
end procedure

With Algorithm 7.3 we are now able to generate microscopic observation data artificially, which we show in the following example

Example 7. *We will apply the acceptance-rejection method on the following examples:*

i) ***One-dimensional****:*

a)

$$f_1(x) = \begin{cases} \frac{e^{-18(x-0.5)^2}+2e^{-18(x-0.3)^2}}{\int_0^1 e^{-18(x-0.5)^2}+2e^{-18(x-0.3)^2}dx}, & \text{for } 0 \leq x \leq 1, \\ 0, & \text{else,} \end{cases} \tag{7.8}$$

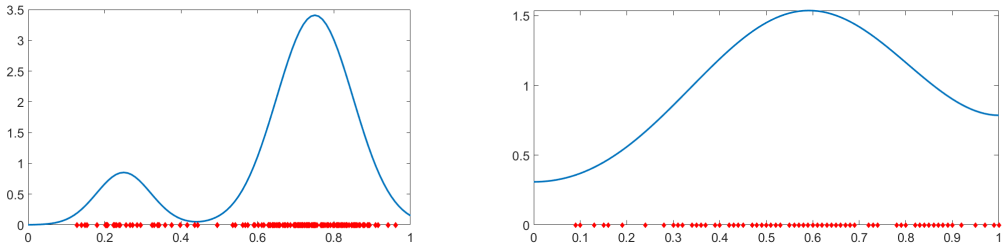
b)

$$\begin{cases} \partial_t u(x,t) &= D\partial_{xx}u(x,t) - c\partial_x u \\ u(x,0) &= u_0(x) = f_1(x) \\ \frac{\partial u}{\partial x}(x,t)\big|_{x=0,x=1} &= 0 \end{cases} \tag{7.9}$$

with $D = 1$ *and* $c = 3$ *for* $x \in [0, 1]$ *and* $t = 0.03$.

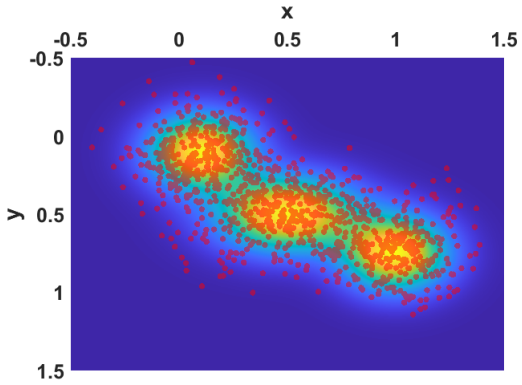
ii) two-dimensional

$$f_2(x,y) = \frac{e^{-18(x-1)^2-10(y+1)^2} + e^{-18(x+0.4)^2-18(y-0.3)^2} + e^{-18x^2-18(y+0.4)^2}}{\int_\Omega e^{-18(x-1)^2-10(y+1)^2} + e^{-18(x+0.4)^2-18(y-0.3)^2} + e^{-18x^2-18(y+0.4)^2} d\Omega}, \tag{7.10}$$



(a) (b)

Figure 7.1: Blue curve shows the density function of a) (7.8) and b) (7.9) with their corresponding sample set (red) derived through acceptance-rejection method for $N = 100$.



(a) Overlay of density field (7.10) with $N = 1000$ sample points(red) obtained through acceptance-rejection method.

Figure 7.2

for the domain $\Omega = [-0.5, 1.5] \times [-0.5, 1.5]$.
The resulting sample points can be seen in Figure 7.1 and Figure 7.2.

In the same way we have generated particle representation of the numerical experiment in Chapter 3 in the Figure 3.14.
Of course the generation is random for a finite number of sample points.
We remind that the purpose of generating such microscopic data artificially is on the one hand to create a test environment for parameter estimation methods in presence of microscopic observation data and on the other hand to have a tool to visualize the dynamics of mathematical models in particle form.

7.2 Particle to Density Transformation

Having established the acceptance-rejection method to create artificial microscopic data for the purpose of testing the efficiency of parameter estimation methods, we will now deal with the question of how to transform microscopic data into macroscopic data, i.e positions into density.

Definition 7.2.1. *Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be an independent and identically distributed sample set of n observations according to an unknown probability density function $f(x)$. Then the density estimate $\widehat{f}_h(x)$ of $f(x)$ assigns each sample point x_i a function $K_h(x_i, x)$ called a kernel function with*

$$\widehat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x_i, x) \tag{7.11}$$

with bandwidth parameter $h \in \mathbb{R}_{>0}$ that determines the amount of smoothing on the density estimate.
The kernel function $K_h(x_i, x)$ is non-negative and bounded for all x

$$0 \leq K_h(x_i, x) < \infty, \quad \forall x \in \mathbb{R} \tag{7.12}$$

and

$$\int_{-\infty}^{\infty} \widehat{f}_h(x) = 1. \tag{7.13}$$

For symmetric kernel functions we can rewrite (7.11) in the most frequently used form

$$K_h\left(x_i, x\right) = \frac{1}{h} K\left(\frac{x - x_i}{h}\right). \tag{7.14}$$

That the kernel density estimate (7.11) is a good approximation of the underlying density function f is shown in the following theorem.

Theorem 7.2.2 ([107])**.** *Let $K_h(x_i, x)$ be a kernel function of bounded variation. Furthermore let the probability density function $f(x)$ be uniformly continuous.*
The bandwidth is defined as $h(n) = \frac{c}{n^\alpha}$ with $0 < \alpha < \frac{1}{2}$, $c > 0$ and sample size $n \in \mathbb{N}$. Then the sequence of kernel density estimates $\widehat{f}_h(x)$ uniformly converges in probability to $f(x)$

$$P\left(\limsup_{n \to \infty} |\widehat{f}_{h(n)}(x) - f(x)| = 0\right) = 1. \tag{7.15}$$

Theorem 7.2.2 states that for a sample size $n \to \infty$ and bandwidth $h \to 0$ the kernel density estimate $\widehat{f}_h$ is converging in probability towards the actual

probability density function f, independently of the choice of kernel function K_h and in a weaker sense that with an appropriate bandwidth h an arbitrarily accurate estimate of the density function can be made with an appropriate large sample size n [147].
Before we continue, we define some important properties regarding kernel density estimation.

Definition 7.2.3 (Bias)**.** *Let $\widehat{f}_h(x)$ be a kernel density estimate (7.11) for the unknown probability density $f(x)$. Then the bias is defined as*

$$\begin{aligned} Bias\left[\widehat{f}_h(x)\right] &= E\left[\widehat{f}_h(x)\right] - f(x) \\ &= \tfrac{1}{n}\sum_{i=1}^{n} E\left[K_h\left(x - x_i\right)\right] - f(x) \\ &= E\left[K_h\left(x - x\right)\right] - f(x), \end{aligned} \tag{7.16}$$

which leads with a second-order Taylor expansion to

$$Bias\left[\widehat{f}_h(x)\right] = \frac{h^2}{2} f''(x) \int s^2 K(s) ds + o\left(h^2\right), \quad \text{as } h \to 0. \tag{7.17}$$

Definition 7.2.4 (Variance)**.** *Let $\widehat{f}_h(x)$ be a kernel density estimate (7.11) for the unknown probability density $f(x)$. Then the variance is defined as*

$$\begin{aligned} Var\left[\widehat{f}_h(x)\right] &= Var\left[\tfrac{1}{n}\sum_{i=1}^{n} K_h\left(x - x_i\right)\right] \\ &= \tfrac{1}{n} Var\left[K_h\left(x - x\right)\right], \end{aligned} \tag{7.18}$$

which with a Taylor expansion leads to

$$Var\left[\widehat{f}_h(x)\right] = \frac{1}{nh} f(x) \int K^2(s) ds + o\left(\frac{1}{nh}\right), \quad \text{as } nh \to 0. \tag{7.19}$$

From the asymptotic behaviour of the bias (7.17) and the variance (7.19) it is clear that the bias is proportional to h^2. Hence a decrease in the bandwidth h results in a reduction of the bias whereas for the variance it is proportional to $\frac{1}{nh}$ and moreover a larger bandwidth h is necessary to reduce the variance of the kernel density estimate.
This shows the trade-off between bias and variance since for fixed sample size n, and reduction of the bias by decreasing bandwidth h leads to a increase in variance and vice versa.
Both quantities (7.17) and (7.19) depend on the unknown probability function

$f(x)$ and its second derivative f''.

Looking at the mean squared error MSE [57]

$$MSE\left[\widehat{f}_h(x)\right] = E\left[\left(\widehat{f}_h(x) - f(x)\right)^2\right] = Var\left[\widehat{f}_h(x)\right] + Bias\left[\widehat{f}_h(x)\right]^2 \tag{7.20}$$

we can deduce that asymptotically with (7.17) and (7.19) the mean squared error converges towards zero if $h \to 0$ and $nh \to \infty$ thus the kernel estimate is consistent to the probability density function f.

The univariate kernel density estimation in definition 7.2.1 can be easily extended to the multivariate case [125].

Definition 7.2.5. *Let $\mathcal{X} = \{\boldsymbol{X_1}, \ldots, \boldsymbol{X_n}\}$ be sample set of m-dimensional random vectors $\boldsymbol{X_i}$ which are distributed according to a density function $f(x_1, \ldots x_n)$. Then the multivariate kernel density estimate $\widehat{f}_{H,n}$ is defined as*

$$\widehat{f}_{H,n}(x) = \frac{1}{n}\sum_{i=1}^{n} K_H\left(x - X_i\right) = \frac{1}{n \det H}\sum_{i=1}^{n} K(H^{-1}\left(x - X_i\right)) \tag{7.21}$$

with $x, X_i \in \mathbb{R}^m$, bandwidth matrix $H \in \mathbb{R}^{m\times m}$ which is symmetric and positive definite and multivariate kernel function K

As we will only apply the kernel density estimation (7.21) in two spatial dimensions at most, we can state that the bandwidth matrix $H \in \mathbb{R}^{2\times 2}$ is in the form

$$H := \begin{pmatrix} h_1 & h_2 \\ h_2 & h_3 \end{pmatrix} \tag{7.22}$$

under the condition

$$\begin{aligned} h_1 &> 0, \\ h_1 h_3 - h_2^2 &> 0. \end{aligned} \tag{7.23}$$

The form of the bandwidth matrix H (7.22) ensures symmetry and conditions (7.23) positive-definite, which are derived under the theorem that a matrix is positive-definite iff all its minors are positive.
For continuous independent random variables $(X_1, \ldots, X_m) \in \mathbb{R}^m$ the multivariate density $f\left(x_1, \ldots, x_m\right)$ can be defined as the product

$$f\left(x_1, \ldots, x_m\right) = f_1(x_1) \cdot \cdots \cdot f_m(x_m). \tag{7.24}$$

Many types of kernel functions can be found in the literature [57, 128, 131, 147] and basically any function $K : \mathbb{R} \to \mathbb{R}$ that satisfies $\int_{-\infty}^{\infty} K(x)dx = 1$ qualifies as a kernel function. However since we are interested in estimating probability densities, we require $K(x) \geq 0$.

However there exist a variety of other not non-negative kernel functions that possess certain properties to deal with bounded domains but we refer to [57]. In the following we mention the three most used kernel function

- **Gaussian**
  - univariate
$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{7.25}$$
  - multivariate
$$K(x_1, \ldots, x_n) = \frac{1}{\sqrt{(2\pi)^2}} e^{-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}} \tag{7.26}$$
- **Uniform**
$$K(x) = \begin{cases} \frac{1}{2} & \text{for } |x| < 1 \\ 0 & \text{for } |x| \geq 1 \end{cases} \tag{7.27}$$
- **Epanechnikov**
  - univariate
$$K(x) = \begin{cases} \frac{3}{4}(1 - x^2), & \text{for } |x| < 1, \\ 0, & \text{for } |x| \geq 1 \end{cases} \tag{7.28}$$
  - multivariate
$$K(x_1, \ldots, x_n) = \begin{cases} \frac{1}{2c_n}(n+2)\left(1 - \sum_{i=1}^n x_i^2\right), & \text{if } \sum_i^n x_i^2 \leq 1 \\ 0, & \text{else,} \end{cases} \tag{7.29}$$

    where c_n denoted the volume of the n-dimensional unit circle.

7.2.1 Boundary Correction

In the definition of the kernel density estimate we have assumed that the density is supported on the entire real domain $(-\infty, \infty)$ (resp. $(-\infty, \infty) \times (-\infty, \infty)$.) However if the density is only defined on a bounded domain, it is well known that the kernel density estimation performs poorly [38]. With no loss of generality, we assume the bounded interval $[0, 1]$ (resp. $[0, 1] \times [0, 1]$) for a probability density field $f(x)$ (resp. $f(x, y)$).
For a given kernel estimate $\widehat{f}(x)$ with bandwidth h we call $[0, h)$ and $(h, 1]$ the boundary region.
The main reason for the so called boundary effects is that the kernel density estimate $\widehat{f}(x)$ exhibits large boundary bias at the boundaries [35, 38]. With

a Gaussian kernel (7.25) for example, we will always have non-zero mass outside the domain

$$\widehat{f}(x) > 0, \text{ for } x < 0 \text{ and } x > 1. \tag{7.30}$$

For cases of densities where the majority of data are not around the boundary regions, the boundary effects are negligible. However for the opposite case, special care needs to be taken.
This is especially true for large bandwidth h which is affected by the noise level of data as well which requires a stronger smoothing effect [57].
Secondly, the kernel estimate does not respect the possible boundary conditions of the domains.
In the literature there are several techniques applied to this problem on removing the boundary effects on the kernel estimate [35, 38]:

- **Reflection method**
  This method is especially designed for homogeneous Neumann conditions $\frac{\partial f}{\partial \boldsymbol{n}}\big|_{\delta\Omega} = 0$ for the outer normal vector $\boldsymbol{n}$ [35].
- **Boundary kernel method**
  At each point in the boundary region a different kernel function K is used [125].
- **Transformation method**
  Mapping the data to the real line $(-\infty, \infty)$ and apply the kernel density estimation, which afterwards is mapped back to the bounded interval.
- **Pseudo-data method**
  Creation of data beyond the boundary.

In the following we will only consider the reflection method due to its easy implementation.

Definition 7.2.6. *Let the definition of 7.2.1 hold and $f(x)$ be supported on $[a, b]$. Then the reflection method for the boundary correction of a kernel density estimate (7.11) is defined as*

$$\widehat{f}_h^{reflect}(x) = \frac{1}{nh}\sum_{i=1}^{n}\left(K\left(\frac{x - x_i^-}{h}\right) + K\left(\frac{x - x_i}{h}\right) + K\left(\frac{x - x_i^+}{h}\right)\right) \tag{7.31}$$

for $a \leq x \leq b$ where $x_i^+ = 2b - x_i$ and $x_i^- = 2a - x_i$.

Since each estimator $\widehat{f}_h(x)$ differs from the original unknown $f(x)$, some measures of discrepancy between these two quantities must be established to evaluate the accuracy of the estimate.

Definition 7.2.7. *Let $f(x)$ be a probability density function and $\mathcal{X} = \{x_1, \ldots, x_n\}$ an independent sample distributed according to $f(x)$.*
Also let $\widehat{f}_h(x)$ be the kernel density estimate (7.11).
Then the mean integrated square error (MISE), asymptotic mean integrated squared error (AMISE) and integrated square error (ISE) are defined as

$$\begin{aligned} MISE\left(\widehat{f}_h(x)\right) &= \int_{-\infty}^{\infty} MSE\left(\widehat{f}_h(x)\right) dx \\ &= \int_{-\infty}^{\infty} E\left[\left(\widehat{f}_h(x) - f(x)\right)^2\right] dx \\ &= \int_{-\infty}^{\infty} Bias^2\left(\widehat{f}_h(x)\right) dx + \int_{-\infty}^{\infty} Var\left(\widehat{f}_h(x)\right) dx, \end{aligned} \tag{7.32}$$

$$\begin{aligned} AMISE\left(\widehat{f}_h(x)\right) &= \tfrac{h^4}{4}\left(\int_{-\infty}^{\infty} s^2 K(s) ds\right)^2 \int f''(x)^2 dx \\ &\quad + \tfrac{1}{nh}\int_{-\infty}^{\infty} K^2(s) ds, \end{aligned}$$

$$\begin{aligned} ISE\left(\widehat{f}_h(x)\right) &= \int_{-\infty}^{\infty}\left(\widehat{f}_h(x) - f(x)\right)^2 dx \\ &= \int_{-\infty}^{\infty} \widehat{f}_h^2(x) dx - 2\int_{-\infty}^{\infty} \widehat{f}_h(x) f(x) dx + \int_{-\infty}^{\infty} f^2(x) dx. \end{aligned} \tag{7.33}$$

7.2.2 Bandwidth Parameters

We have already seen in the definition of bias, variance and mean squared error that the bandwidth h in the one-dimensional case and bandwidth matrix H in the multi-dimensional case control the smoothness of the estimate and is essential for the quality of the estimate [57, 125].
A large h yields to an oversmoothed estimate which increases the bias and important features of the true density function are obscured, whereas a small h yields to an undersmoothed estimate, which increased the variance of the estimate such that the smallest feature such as features introduced by noise, will be present in the estimate. This is depicted in Figure 7.3.
In the univariate kernel density estimation (7.11), the uniform kernel function is equivalent to a histogram where the width of the bin is in fact the bandwidth h.
For the multivariate kernel density estimation the bandwidth H not only controls the width of the bin but also the orientation, which considerably influence the shape of the density estimate [125]. The effect can be seen in Figure 7.4, where we show the kernel density estimate for three different bandwidth matrices H:

- positive scalar times the identity matrix: $H = \begin{pmatrix} \frac{1}{20} & 0 \\ 0 & \frac{1}{20} \end{pmatrix}$.

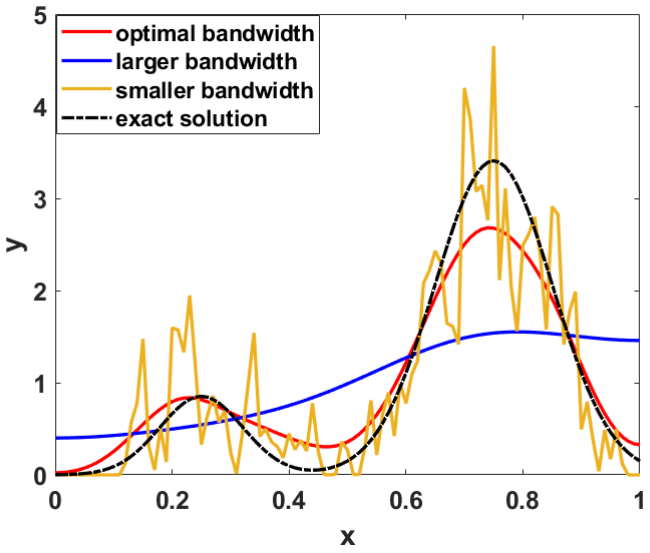


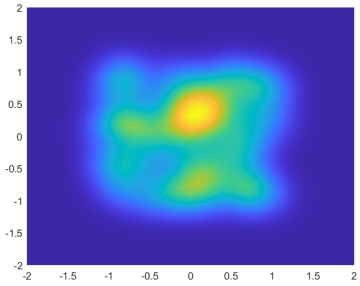
Figure 7.3: Effect of the bandwidth size on kernel density estimate. Blue: $h > h^{opt}$ creates an oversmoothed estimate. Red: optimal bandwidth h^*. Yellow: $h < h^{opt}$ creates an undersmoothed estimates with many spurious oscillations.

- diagonal matrix $H = \begin{pmatrix} \frac{1}{100} & 0 \\ 0 & \frac{\frac{1}{20}^2}{\frac{1}{100}} \end{pmatrix}$.
- symmetric positive definite matrix $H = \begin{pmatrix} \frac{3}{40} & \sqrt{\frac{3}{40}^2 - \frac{1}{20}^2} \\ \sqrt{\frac{3}{40}^2 - \frac{1}{20}^2} & \frac{3}{40} \end{pmatrix}$.

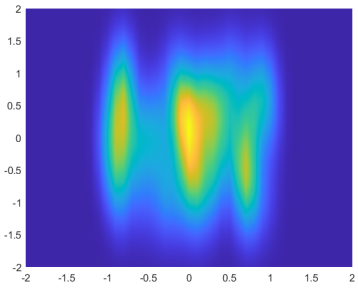
7.2.3 Kernel and Bandwidth Selection

Choosing the optimal bandwidth is essential for the quality of the kernel density estimate. The error estimation in Definition 7.2.7 can be used to determine the optimal bandwidth h^{opt} by minimizing either the MISE, AMISE (7.32) or ISE (7.33).
This also means that each optimal bandwidth differs depending on which error estimate has been used.
Because each error estimate involves the unknown density function $f(x)$, using the error estimates to determine h exactly is impossible. However, there exists a variety of methods to calculate the bandwidth parameter and we will describe two of the most commonly used ones.

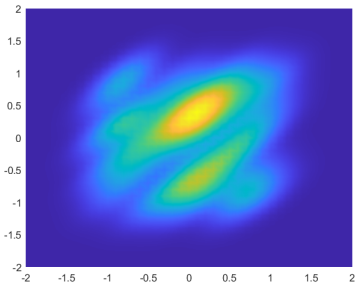
The first one we will present is Silverman's Rule-of-Thumb [131] as a plug-in method.



(a) Kernel density estimate for positive scalar times the identity matrix.



(b) Kernel density estimate for diagonal matrix.



(c) Kernel density estimate for symmetric positive definite matrix.

Figure 7.4: Kernel density estimation for different bandwidth matrices H. The same microscopic data is used for the kernel density estimation. However, the choice of bandwidth matrix H considerably changes the shape of the resulting density estimate. In particular the density estimate (b) and (c) are seemingly stretched out compared to (a) which uses the identity matrix as bandwidth matrix.

Rule of Thumbs Bandwidth

The main idea behind this method is based on AMISE and to replace the unknown density function $f(x)$ with an estimate.
Silverman [131] assumes the density function $f(x)$ to be a normal probability distribution $N(\mu, \sigma^2)$ with mean μ and variance σ^2 and a Gaussian kernel function $\phi(x)$.
By definition of AMISE (7.32) we have

$$\| f'' \|_2^2 = \int_{-\infty}^{\infty} f''(x)^2 dx = \frac{1}{\sigma^5} \int \phi''(x)^2 dx = \frac{1}{\sigma^5} \frac{3}{8\sqrt{\pi}} \approx 0.212 \frac{1}{\sigma^5}. \tag{7.34}$$

Replacing the unknown standard deviation σ in (7.34) by an estimate $\widehat{\sigma}$ according to

$$\widehat{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{7.35}$$

with the arithmetic mean $\bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i$ we can obtain the Rule-of-Thumb bandwidth by using AMISE (7.32), differentiating it with respect to the bandwidth h and solve the first-order condition for h which leads to

$$h = \left(\frac{\int_{-\infty}^{\infty} K^2(s)ds}{n \int_{-\infty}^{\infty} f''(x)^2 dx \left(\int_{-\infty}^{\infty} s^2 K(s)ds \right)^2} \right)^{\frac{1}{5}}. \tag{7.36}$$

We take as the kernel function the Gaussian kernel function $K = \phi$ and use (7.34) in (7.36) to obtain the Rule-of-Thumb bandwidth

$$\widehat{h}_{ROT} = \left(\frac{4\widehat{\sigma}^5}{3n} \right)^{\frac{1}{5}} \approx 1.06 \frac{\widehat{\sigma}}{n^{\frac{1}{5}}}. \tag{7.37}$$

Although the Rule-of-Thumb bandwidth is a data-based bandwidth which is frequently used in kernel density estimations in statistic program [131] and also in MATLAB©, the more the estimate can decrease in quality the more the unknown density function f differs from a normal distribution.

Least Square Cross Validation Bandwidth

The second method to obtain an optimal bandwidth is the Least squares cross validation method (LSCV) which uses the ISE [147].
In the ISE (7.33) expression, $\int_{-\infty}^{\infty} \widehat{f}_h^2(x)dx$ can be calculated from data, and $\int_{-\infty}^{\infty} f^2(x)dx$ does not depend on the bandwidth h and thus does not need to be considered in the minimization of ISE over h.
The term $\int_{-\infty}^{\infty} \widehat{f}_h(x)f(x)dx$ is equivalent to the expected value of $\widehat{f}_h(X)$. As such, we can estimate the expected value with the leave-one-out estimator

$$\widehat{E\left[\widehat{f}_h(X)\right]} = \frac{1}{n}\sum_{i=1}^{n} \widehat{f}_{h,-i}(X_i) \tag{7.38}$$

with

$$\widehat{f}_{h,-i}(x) = \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} K_h\left(x - X_j\right). \tag{7.39}$$

Using these we obtain the Least Squares Cross-Validation criteria

$$LSCV(h) = \int \widehat{f}_h^2(x)dx - \frac{2}{n(n-1)} \sum_{i=1}^{n}\sum_{j\neq i}^{n} K_h(X_i - X_j). \tag{7.40}$$

Since $LSCV(h)$ does not depend on any unknown quantities anymore, we can apply a minimization method (see chapter 5) to obtain the optimal bandwidth parameter $\widehat{h}_{LSCV}$.

Bandwidth Selection through Minimization

The third method is the most intuitive one to use under the requirement that the density function f is actually known.
Then any of the three error estimates MISE, AMISE (7.32) and ISE (7.33) can be minimized without any further assumptions.
This approach is of special interest for the future numerical simulations of this thesis because we want to fit a density function f to the observation data which we already assume to be distributed according to said density function. Of course, this leads to a considerable bias since in real observation data we have constructed a mathematical model based on assumptions, which only approximately represents the actual dynamics of the density function from which the observation data are obtained from.
However, the bias induced towards the observation data is not strong in view of the fact that the model parameters obtained in such a way can be used to infer the quality of the approximation by logical reasoning and a-priori knowledge of the magnitudes of the model parameters.

Multivariate Bandwidth Selection

The derivation of optimal bandwidth matrices $H \in \mathbb{R}^{m\times m}$ for the multivariate kernel density estimation can be obtained in a very similar way to the univariate case [125].
For Silverman's Rule-of-Thumb bandwidth matrix [131] we assume that the multivariate probability density function $f(x_1, \ldots, x_m)$ is a multivariate normal distributed $N(\boldsymbol{\mu}, \Sigma)$, and the kernel function used is the multivariate Gaussian.
Making the additional assumption that the bandwidth matrix H and covariance matrix Σ are diagonal matrices leads to

$$\widehat{h}_j = \frac{4}{m+2}^{\frac{1}{m+4}} n^{-\frac{1}{m+4}} \sigma_j, \tag{7.41}$$

which gives us the previously derived Silverman's Rule-of-Thumb for $m = 1$.
Making a more general assumption that H is proportional to the square root of the covariance matrix $\Sigma^{\frac{1}{2}}$ we obtain the multivariate Rule-of Thumb bandwidth

$$\widehat{H} = \frac{1}{n^{\frac{1}{m+4}}} \widehat{\Sigma}^{\frac{1}{2}}. \tag{7.42}$$

Also the least square cross-validation method can be generalized for the multivariate case. We refer to [46] for the derivation which is very similar for the one-dimensional case.

$$LSCV(H) = \frac{1}{n^2 \det H} \sum_{i=1}^{n} \sum_{j=1}^{n} \int_{-\infty}^{\infty} K(H^{-1}(X - X_i)) K(H^{-1}(X - X_j)) dX. \quad (7.43)$$

Minimizing $LSCV(H)$ (7.43) over H means solving a m^2-dimensional minimization problem.

7.2.4 Induced Transformation Error

In this section we will investigate of the numerical error induced by the density to particle to density transformation where we first create artificial macroscopic observation data, derive microscopic observation data with sample size n with the acceptance-rejection method (see Algorithm 7.1-7.3) and transform the sample back to macroscopic observation data in form of density with the kernel density estimation for a variety of bandwidth selection methods.
We will compare each kernel density estimation for the Rule-of-Thumb bandwidth and through minimization and investigate the influence of the sample size n with regards to the error between the artificial macroscopic observation data with the transformed artificial macroscopic observation data.

One-dimensional

Example 8. *For the following numerical simulations we will focus on two kinds of probability density function:*

(a) An explicitly defined probability density function

$$f_1(x) = \begin{cases} \frac{e^{-18(x-0.5)^2} + 2e^{-18(x-0.3)^2}}{\int_{-1}^{1} e^{-18(x-0.5)^2} + 2e^{-18(x-0.3)^2} dx} & \text{for } -1 \leq x \leq 1 \\ 0 & \text{else,} \end{cases} \quad (7.44)$$

which is a superposition of two truncated normal distributions, bounded at the interval $[-1, 1]$.

(b) The second density is the solution of the one-dimensional convection-diffusion equation

$$\begin{cases} \partial_t u(x,t) &= D\partial_{xx} u(x,t) - c\partial_x u, \\ u(x,0) &= u_0(x) = f_1(x), \\ \frac{\partial u}{\partial x}(x,t)\big|_{x=-1,x=1} &= 0 \end{cases} \quad (7.45)$$

with diffusion coefficient $D = 1$, convection coefficient $c = 3$ on the domain $\Omega := [-1, 1]$.
We also assume $\int_{-1}^{1} u(x,t)dx = \int_{-1}^{1} u(x,0)dx = 1$ for all $t > 0$.
We solve (7.45) with a finite difference scheme for time $T = 1$ and investigate the probability density at $u(x, 1)$.

The microscopic observation data in form of the random sample sets $\mathcal{X}^1$ for a) $f_1(x)$ are being obtained by the acceptance-rejection method and $\mathcal{X}^2$ for b) $u(x, T)$ are being obtained by the acceptance-rejection method for sample sizes $N = 100, 1000$.
For the transformation back to density we use the kernel density estimation with the reflection method (7.31) at the boundaries $x = -1$ and $x = 1$.
As for the bandwidth selection we compute the bandwidth with both the Rule-of-Thumb method (7.37) $\widehat{h}_{ROT}$, and the minimization method for the $ISE(h)$ (7.33) $\widehat{h}_{ISE}$.
In Figure 7.5 and in Figure 7.6 we can see the comparison between the different experiments.

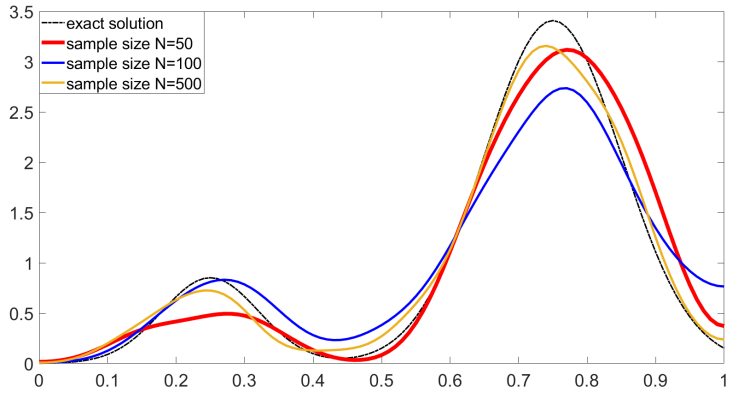
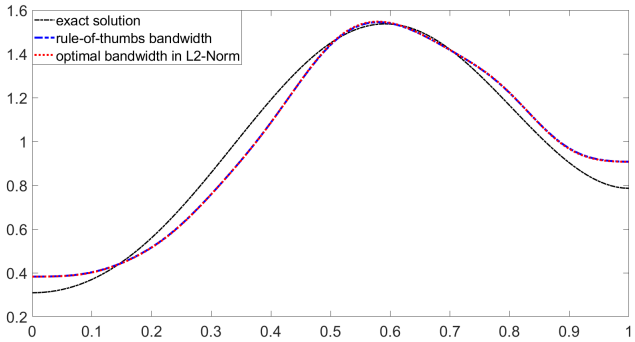


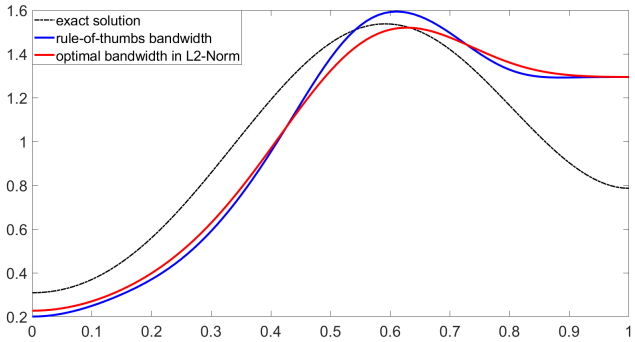
Figure 7.5: KDE with different sample sizes $N = 50, 100, 1000$. The more samples are available, the more accurately the kernel density estimate approximates the exact solution.

Two-dimensional

Next, we will conduct the same experiment for the two-dimensional probability density.



(a) KDE of 1D-convection-diffusion equation for sample size $N = 1000$. Bandwidths are $h_{ROT} = 0.0671$ and $h^* = 0.0658$.



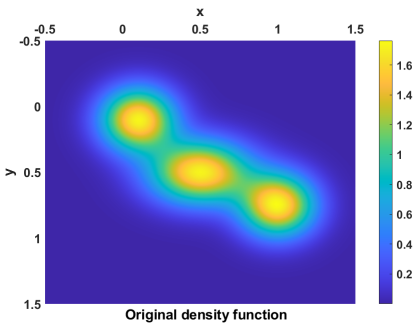
(b) KDE of 1D-convection-diffusion equation for sample size $N = 100$. Bandwidths are $h_{ROT} = 0.094$ and $h^* = 0.12$.

Figure 7.6

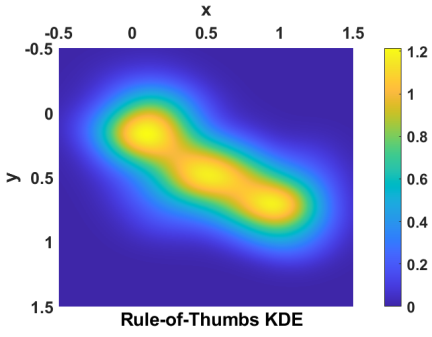
Example 9. *As in example 8 we investigate the multivariate truncated normal distribution*

$$f_2(x,y) = \begin{cases} \frac{e^{-18(x-1)^2-10(y+1)^2}+e^{-18(x+0.4)^2-18(y-0.3)^2}+e^{-18x^2-18(y+0.4)^2}}{\int_\Omega e^{-18(x-1)^2-10(y+1)^2}+e^{-18(x+0.4)^2-18(y-0.3)^2}+e^{-18x^2-18(y+0.4)^2}d\Omega}, & \text{for } -1 \le x \le 1 \\ & \text{and } -1 \le y \le 1, \\ 0, & \text{else} \end{cases} \tag{7.46}$$

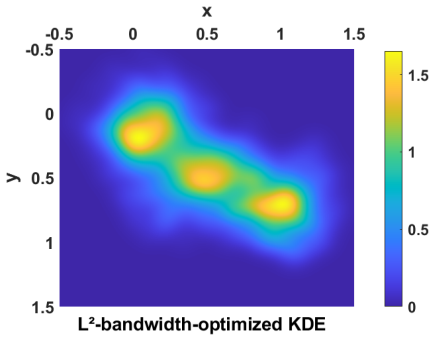
on the bounded domain $\Omega := [-1,1] \times [-1,1]$.
Similarly to the one-dimensional test we create a two-dimensional sample set $\mathcal{X}^1$ *with the acceptance-rejection method with a sample size of* $N = 1000$ *and apply the multivariate kernel density estimation (7.21) with both the multivariate Rule-of-Thumb method (7.42) and the minimization of* $ISE(H)$ *method as bandwidth selection methods to select the bandwidth matrix* H*. The results are shown in Figure 7.7.*



(a) Original density function.



(b) KDE: Rules-of-Thumb $H_{ROT} = \mathrm{diag}\,(0.1087, 0.1605)$.



(c) KDE:L^2-norm minimized $H = \mathrm{diag}\,(0.0739, 0.0757)$.

Figure 7.7: Comparison between the original density function (a) with kernel density estimates with the Rule-of-Thumb bandwidth and L^2-norm minimized bandwidth. (b) is smoother but dissolves the transition areas between the three areas of highest concentration. (c) is less smooth but magnitudes of highest concentration are recovered.

Conclusion of Example 8 and 9

We can conclude from the results from examples 8 and 9, that in general an increase of the sample size N does not only decrease the error between the exact artificial data and the kernel density estimation through particle transformation but also allows a smaller bandwidth size h and H.
Furthermore, it is becoming clear that the Rule-of-Thumb bandwidth allows the kernel density estimation to perform quite well, however does the bandwidth obtained from the minimization of ISE delivers smaller relative errors in the L^2-norm.
In the two-dimensional case the difference becomes more apparent due to the fact that the Rules-of-Thumb only allows diagonal bandwidth matrices H whereas with the ISE minimization we are able to include full bandwidth matrices H.
The relative error err_{rel} of the kernel density estimation between the Rules-of-Thumb bandwidth matrix with $err_{rel} = 0.2593$ and the L^2-norm minimized

bandwidth matrix with $err_{rel} = 0.1644$ shows the difference. This is also due to the fact that the magnitude of the spikes are much more accurately represented for the L^2-norm minimized kernel density estimate with 1.6 compared to the original magnitude of 1.78, whereas the Rule-of-Thumb kernel density estimate only reaches the magnitude of around 1.2. However we need to remark that the Rule-of-Thumb kernel density estimate, by definition, minimizes over the MISE.

7.3 Parameter Estimation with Kernel Density Estimation

With the kernel density estimation at hand we have a tool that can transform microscopic observation data into macroscopic observation data u_{data} which then can be used in our parameter estimation method presented in chapter 6.2.

Definition 7.3.1. *Let the microscopic observation data $\mathcal{X}$ be*

$$\mathcal{X} := \left(\mathcal{X}^l\right)_{l=0,\dots,M} := (x_m^l, y_m^l)_{m=1,\dots,M(l)}, \tag{7.47}$$

where $\mathcal{X}^l$ with $l \in \mathbb{N}$ is the set of the $M(l)$ particle positions (x_m^l, y_m^l) at time t_l.
Then we define the corresponding macroscopic observation data u_{data} through the kernel density estimation as

$$u_{data}\,(H) := \begin{pmatrix} u_{data}^0 \\ \vdots \\ u_{data}^M \end{pmatrix} \tag{7.48}$$

with

$$u_{data}^l(H) := \widehat{f}_{H,M(l)}(x) \tag{7.49}$$

where $\widehat{f}_{H,M(l)}(x)$ is the kernel density estimation (7.21) for bandwidth matrix $H \in \mathbb{R}^{2\times 2}$ with $M(l)$ particles in the set $\mathcal{X}^l$.

There are two approaches to apply a parameter estimation method with microscopic data $\mathcal{X}$:

1. Transforming the microscopic observation data $\mathcal{X}$ into macroscopic observation data u_{data} through kernel density estimation and applying a parameter estimation method afterwards.

2. Applying the kernel density estimation within the parameter estimation method.

In the first approach the bandwidth h (resp. H) is selected with a bandwidth selection method and then we proceed with the parameter estimation.
Nevertheless, there is a trade-off between simplicity and accuracy. Depending on the sample size N and choice of bandwidth, we inevitably introduce a certain level of noise to the obtained macroscopic observation data, as we have seen in Examples 8 and 9.
The second approach consists in taking advantage of the fact that we expect that the observation data u_{data} are well modelled by the real density function f which models parameters we want to fit. Hence the choice of bandwidth can be plugged into the parameter estimation method itself as an additional unknown parameter. This is especially reasonable since most of the bandwidth selectors are data-based methods since the density function f is either unknown or only weak assumptions are available. In our problem however, we assume that the data is coming from a known density function where only the model parameters are unknown.
Essentially this approach is similar to the approach for the estimation of the optimal bandwidth through the minimization of the ISE (7.33), where the exact density function f is being used. Although in case of real observation data, where the true density function f is often unknown and just an approximation of the true density function is available, using such an approximated density function for the minimization of the ISE to obtain the bandwidth can improve the overall quality of the kernel density estimation and as such the parameter estimation.
This is the motivation of the Kernel Density Estimation Parameter Estimation Method **(KDE-PE**).

KDE-Parameter Estimation Method (KDE-PE)

In this context of parameter estimation where the density is known, but the model parameters need to be estimated, we define the non-regularized objective functional for the KDE-PE.

Definition 7.3.2. *Let the forward problem be defined as in Definition 6.1.2. Furthermore, let $\mathcal{X}$ be the microscopic observation data. Then we define the inverse problem with kernel density estimate as the minimization problem to estimate model parameter set Θ and bandwidth matrix H*

$$\{\Theta_{opt}, H_{opt}\} = \arg \min_{\Theta \in \mathbb{R}^p, H \in \mathbb{R}^{2\times 2}} K(\Theta, H) \tag{7.50}$$

with the non-regularized nonlinear least square functional

$$K(\Theta, H) := \| F(\Theta) - u_{data}(H) \|_2^2, \tag{7.51}$$

where the observation data $u_{data}(H)$ are defined in (7.48), (7.49) and depend explicitly on the bandwidth matrix H.

The objective functional K (7.51) can be interpreted as the discrete equivalent of the integrated squared error (ISE) (7.33) such that the bandwidth selection within the parameter estimation methodology results in better estimations, i.e. smaller relative error err_{rel} in the L^2-norm, than obtaining the bandwidth from the previous mentioned methods such as the Rule-of-Thumb (7.37) as previously shown in the examples of Section 7.2.4.

Regularization of Kernel Density Estimation

However this could lead to a heavily over-smoothing, i.e large bandwidths H, which can deteriorate the quality of the estimates of model parameters Θ.
For this reason we need to introduce an additional regularization term to the objective functional $K(\Theta, H)$ in (7.51)

$$K\left(\Theta, H\right) := \parallel F\left(\Theta\right) - u_{data}(H) \parallel_2^2 + p_H \parallel H - H_0 \parallel_2^2 \tag{7.52}$$

with constant $p_H > 0$ and the a-priori bandwidth H_0 which can be any of the previously mentioned bandwidth selections such as the Rule-of-Thumb (7.37).
With the additional regularization of the bandwidth we not only avoid over-smoothing with large bandwidths H but also avoid a possible under-smoothing due to the introduced H_0.
This enforces the bandwidth parameter H to be around the Rules-of-Thumb bandwidth H_0 which is already a decent bandwidth choice as we have shown in example 9.
For large values $p_H > 0$, the term (7.52) enforces the bandwidth parameter H to be close to the actual a-priori estimate H_0.
A decrease of p_H eases the restriction such that greater changes of H away from H_0 are allowed to "fine-tune" the kernel density estimate $u_{data}(H)$.

7.3.1 Numerical Simulation: KDE-PE Method

We will extend the two numerical test simulations from chapter 6.3 for the one-dimensional and two-dimensional convection-diffusion equation by transforming the used artificial observation data u_{data} obtained through finite difference schemes to a microscopic sample with the acceptance-rejection method and apply the two approaches of kernel density estimation to this sample as described in the preceding section.

One-dimensional

As forward problem we consider the following one-dimensional convection-diffusion equation

$$\begin{cases} \partial_t u & = & D\partial_{xx}u - \partial_x\left(vu\right), \\ u(x,0) & = & u_0(x), \\ \frac{\partial u}{\partial x}\big|_{\partial\Omega} & = & 0 \end{cases} \tag{7.53}$$

with diffusion coefficient $D > 0$ and velocity field function $\boldsymbol{v}(x) \in \mathcal{C}^2$. We use the multigrid method for the parameter estimation with the only difference of using the modified objective functional (7.52) which contains the regularization term for the bandwidth $h > 0$ and a-priori bandwidth h_0 computed according to the Rule-of-Thumb-method (7.37).
For the artificial observation data, we created a microscopic sample of size $N = 10^5$ with the acceptance-rejection method and apply the kernel density estimation with the reflection method as boundary effect correction (7.31).

Example 10. *We consider the convection-diffusion equation (7.53) as forward problem with the forward operator F defined according to the finite difference scheme (3.22).*
The artificial observation data have been computed for a diffusion coefficient $D = 4$ and a velocity field function $\boldsymbol{v}(x) = c(x) = 15\sin(2\pi x)$ on the domain $\Omega = [0,1]$ and transformed into the microscopic observation data $\mathcal{X}$ through acceptance-rejection method for $N = 10^5$ sample points per time t.
We then conduct the following two numerical experiments:

*a) The parameter estimation **(PE)** according to Definition 6.1.3 with regularized objective functional*

$$K\left(\Theta\right) = \parallel F(\Theta) - u_{data} \parallel_2^2 + \lambda \parallel \Theta \parallel_2^2 \tag{7.54}$$

with the artificial macroscopic observation data u_{data}.

*b) KDE-parameter estimation **(KDE-PE)** according to Definition 7.3.2 with regularized objective functional*

$$K\left(\Theta, h\right) = \parallel F(\Theta) - u_{data}(h) \parallel_2^2 + \lambda \parallel \Theta \parallel_2^2 + p_h \parallel h - h_{ROT} \parallel_2^2 \tag{7.55}$$

with regularization parameters $\lambda > 0$ and $p_h > 0$, where the artificial macroscopic data $u_{data}(h)$ are defined with (7.48), (7.49) where the regularization parameters have been estimated experimentally.

The results can be seen in Figure 7.8 where the relative error of each parameter is indicated.

Conclusion

The results obtained from example 10 for a) and b) are represented in Figure 7.8. We can infer from the results that the parameters from the parameter estimation a) are recovered with a very low relative errors for the diffusion coefficient $err_D = 0.1\%$ and velocity field $err_v = 0.1\%$. The relative errors for the KDE-PE method b) on the other hand are 4.2% for the diffusion coefficient and 8.7% for the velocity field, which are higher than in a). This can be explained by the fact that we did not use the artificial data u_{data} as in a), but created a microscopic sample set $\mathcal{X}$ which we then re-transformed back with the kernel density estimation. These transformations naturally induce some noise and error during the parameter estimation process which explains the higher relative errors in the parameters.
Furthermore we remark that the estimated bandwidth h is slightly different from the Rule-of-Thumb bandwidth h_0. This can be explained by the fact that the Rule-of-Thumb is only optimal for normal distributed densities but due to the present convection term, the optimal bandwidth has a different value.

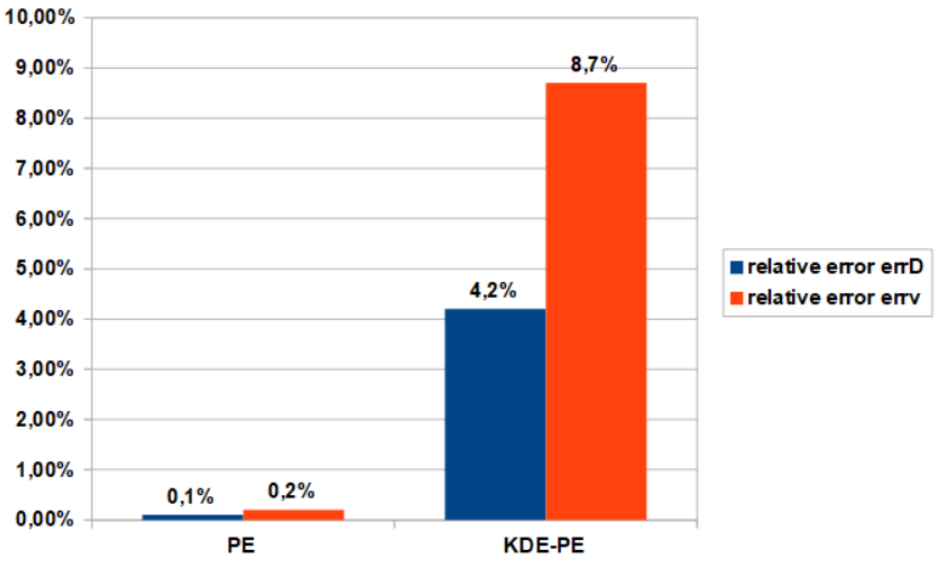


Figure 7.8: The relative errors of the estimated parameters: Diffusion coefficient D and velocity field $\boldsymbol{v}$. The parameter estimation method (PE) with artificial macroscopic observation data u_{data} performs significantly better than the KDE-PE method (with $|h - h_0| = 0.022$) due to the induced noise and error because of the transformation between microscopic and macroscopic data.

Two-dimensional

Similar to the one-dimensional test equation, we now investigate the two-dimensional convection-diffusion equation

$$\begin{cases} \partial_t u & = & D\triangle u - \operatorname{div}(\boldsymbol{v}u)\,, \\ u(x,y,0) & = & u_0(x,y), \\ \frac{\partial u}{\partial \boldsymbol{n}}\big|_{\partial\Omega} & = & 0, \end{cases} \tag{7.56}$$

as the forward problem for $D > 0$ and velocity field function $\boldsymbol{v} \in \mathcal{C}^2$. We use the multigrid method for the parameter estimation with the modified objective functional (7.52) with diagonal bandwidth matrix $H = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix}$ and Rule-of-Thumb bandwidth H_{ROT} (7.42).
For the artificial observation data, we create a microscopic sample set of size $N = 10^7$ with the acceptance-rejection method for different times t and apply the kernel density estimation with the reflection method as boundary effect correction (7.31).

Example 11. *We consider the forward problem as the two-dimensional convection-diffusion equation (7.56) with forward operator F defined according to the finite difference scheme (3.24).*
The artificial observation data have been computed for a diffusion coefficient $D = 1$ and velocity field function

$$v(x,y) := \begin{pmatrix} -10\,(x-0.5)\,e^{-5(x-0.5)^2-5(y-0.5)^2} \\ -10\,(y-0.5)\,e^{-5(x-0.5)^2-5(y-0.5)^2} \end{pmatrix} \tag{7.57}$$

on the domain $\Omega := [0,1]\times[0,1]$ and transformed into microscopic observation data $\mathcal{X}$ through the acceptance-rejection method for $N = 10^7$ sample points per time t.
We then conduct the following two numerical experiments:

a) *The parameter estimation **(PE)** according to Definition 6.1.3 with regularized objective functional*

$$K(\Theta) = \| F(\Theta) - u_{data} \|_2^2 + \lambda \| \Theta \|_2^2 \tag{7.58}$$

with artificial macroscopic observation data u_{data}.

b) ***KDE-PE** according to Definition 7.3.2 with regularized objective functional*

$$K(\Theta, H) = \| F(\Theta) - u_{data}(H) \|_2^2 + \lambda \| \Theta \|_2^2 + p_H \| H - H_{ROT} \|_2^2 \tag{7.59}$$

with regularization parameter $\lambda > 0$ and $p_H > 0$, where the artificial macroscopic data $u_{data}(H)$ are defined (7.48), (7.49). The regularization parameters have been estimated experimentally.

The results can be seen in Figure 7.9, where the relative error of each parameter is indicated

Conclusion

The relative errors of the 2D problem is in general much higher than for the 1D problem for both, the parameter estimation method with artificial macroscopic observation data u_{data} and the KDE-PE method with microscopic observation data $\mathcal{X}$.
However the discrepancy between these two methods is much bigger, with the largest difference of relative errors for $\boldsymbol{v}_y$.
It is also worth mentioning that the difference between estimated bandwidth matrix H and Rule-of-Thumb bandwidth matrix H_{ROT} is much larger with $\| H - H_{ROT} \|_2 = 0.107$. This indicates that in spatial dimensions greater than 1 the estimated bandwidth matrix H improves the parameter estimation more significantly than the Rules-of-Thumb method.

The numerical simulations of the KDE-PE method has shown that by allowing the bandwidth to be treated as an additional unknown parameter in the parameter estimation method with its own regularization term $p_h \| h - h_{ROT} \|_2^2$ regularization parameter $p_h \in \mathbb{R}_{\geq 0}$ and a-priori computed Rule-of-Thumb bandwidth h_{ROT}, we are able to use microscopic observation data for the parameter calibration of macroscopic models and obtain satisfactory results.
In the next chapter we will make use of the KDE-PE method where we calibrate model parameters of the OOC-model with real observation data from the laboratory experiments described in [31, 141] which are available in microscopic form.

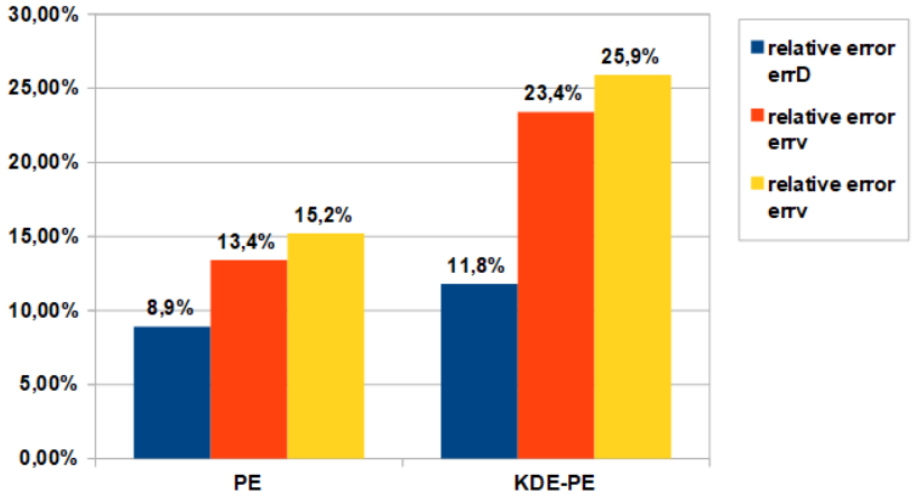


Figure 7.9: The relative errors of the estimated parameters: Diffusion coefficient D and velocity field $\boldsymbol{v_x}$, $\boldsymbol{v_y}$. The parameter estimation method (PE) with artificial macroscopic observation data u_{data} performs significantly better than KDE-PE (with $\| H - \widehat{H}_0 \|_2 = 0.107$) due to the induced noise and error because of the transformation between microscopic and macroscopic data. Both methods however produce higher relative errors compared to the one-dimensional parameter estimation problem.

Chapter 8

Numerical Implementation and Results of Parameter Estimation of OOC Model

The final chapter of this work makes use of all the techniques and methods previously presented and tested to construct a meaningful parameter estimation process for the full mathematical model (1.45)-(1.47) of the laboratory experiment on the OOC.
As presented in Section 1.1, the observation available to us in form of the video recording does not contain the complete domain of the OOC, but only a small section, which contains part of the left and right chambers Ω_l and Ω_r connected through one-dimensional micro channels.

OOC-model

We remind that the dynamics of cells and chemicals in the two-dimensional chambers are described with the parabolic Keller-Segel like model (1.45)

$$\begin{cases} \partial_t T &= D_T \Delta T - \lambda_T(\omega) T, \\ \partial_t M &= D_M \Delta M - \operatorname{div}\left(f\left(M, \phi\right)\right) \\ \partial_t \phi &= D_\phi \Delta \phi + \alpha_\phi T - \beta_\phi \phi, \\ \partial_t \omega &= D_\omega \Delta \omega + \alpha_\omega M - \beta_\omega \omega, \end{cases} \tag{8.1}$$

and for the one-dimensional channels we have either the parabolic Keller-Segel like model (1.46)

$$\begin{cases} \partial_t T_c &= D_T \partial_{xx} T_c - \lambda_T(\omega) T_c, \\ \partial_t M_c &= D_M \partial_{xx} M_c - \partial_x f_c \\ \partial_t \phi_c &= D_\phi \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c, \\ \partial_t \omega_c &= D_\omega \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_\omega \omega_c, \end{cases} \tag{8.2}$$

or the hyperbolic Cattaneo-like model (1.47)

$$\begin{cases} \partial_t T_c + \partial_x v_c^T = -\lambda_T(\omega) T_c, \\ \partial_t v_c^T + \frac{D_T}{\tau_T} \partial_x T_c = \frac{-v_c^T}{\tau_T}, \\ \partial_t \omega_c = D_{\omega_c} \partial_{xx} \omega_c + \alpha_\omega M_c - \beta_c \omega_c, \\ \partial_t M_c + \partial_x v_c^M = 0, \\ \partial_t v_c^M + \frac{D_M}{\tau_M} \partial_x M_c = \left(f_c - v_c^M \right) \frac{1}{\tau_M}, \\ \partial_t \phi_c = D_{\phi_c} \partial_{xx} \phi_c + \alpha_\phi T_c - \beta_\phi \phi_c. \end{cases} \tag{8.3}$$

Simplifications

To simplify the parameter estimation and reduce computational time we propose the following simplifications to the model:

1. Parameter estimation method applied in one chamber only.
2. Excluding cytokine ω.
3. Tumour cells T are stationary.
4. Excluding chemoattractant ϕ.

The reasoning behind these simplifications are:

1. The first simplification is justified by the fact that no cell tracking data are available within the one-dimensional micro channels [15] and introducing additional parameters such as the Kedem-Katchalsky parameter K and the other micro channel specified parameters increases the complexity of the parameter estimation unnecessary.

2. For second simplification we consider that no data are available about cytokine ω.
   The cytokine ω is directly involved in the decay rate of tumour cells T which is not relevant for the time window in which the observation data have been obtained and hence can be neglected [31].

3. The third simplification is based on the fact that in the laboratory experiment, the tumour cells T appear almost stationary within the time frame of the available video recording.
   Inclusion is only necessary for experiment settings, where the tumour cells are more mobile (see [15]).
   For the experiment, where the tumour cells T are virtually stationary, we can assume $T(x, y, t) = T(x, y, 0) = T_0$ and exclude its partial differential equation.

4. The fourth and last simplification is the initial neglect of the diffusion equation describing the dynamics of the chemoattractant ϕ.
   Similar to the cytokine ω, no data are available for ϕ as well, making the parameter estimation very challenging. But in contrast to the cytokine ω which has almost no influence in the dynamics of the experiment for the time interval we are working with [31, 141], the chemoattractant field ϕ fundamentally dictates the dynamics of the immune cells M due to the chemotactic function $f := \widehat{f}\,(M, \phi)\, M$.
   As a result, we introduce the velocity field function $\boldsymbol{v} : \Omega \times [0, \infty) \to \mathbb{R}^2$ with
   $$\boldsymbol{v}(x, y, t) := \widehat{f}\,(M, \phi) \tag{8.4}$$
   to replace the chemotactic dynamics induced by the chemoattractant ϕ by the unknown parameter $\boldsymbol{v}$, which can be used after the parameter estimation in combination with diffusion equation of ϕ to obtain some additional information about the chemoattractant ϕ and its model parameters.

Simplified Model for Parameter Estimation

Applying these ideas, we get the simplified model

$$\partial_t M \;=\; D_M \Delta M - \operatorname{div}(\boldsymbol{v} M) \tag{8.5}$$

with immune cell density M, diffusion coefficient $D_M > 0$ and velocity field function $\boldsymbol{v}$.
We want to remark, that although we have introduced $\boldsymbol{v}$ in order to avoid

the necessity of having information about the chemoattractant ϕ, the relation between them through the chemoattractant equation

$$\partial_t \phi = D_\phi \Delta \phi + \alpha_\phi T_0 - \beta_\phi \phi \tag{8.6}$$

with

$$\boldsymbol{v} = \widehat{f}\,(M, \phi)\,. \tag{8.7}$$

can be used to infer additional information about the chemoattractant distribution.

We have reduced the full model (1.45) to a single two-dimensional convection-diffusion equation (8.5) with unknown parameters D_M and $\boldsymbol{v}$.
We can apply the multi-grid parameter estimation method on this problem which includes the method of time-varying parameters of Section 6.2.5 and the function reconstruction for the post-smoothing step of Section 6.2.6.
Our aim is to use the available observation data of the laboratory experiment in order to parameter fit the mathematical model that describes the dynamics of the real experiment. This way the mathematical model describes the real behaviour on the OOC experiment and obtain feasible parameters in form of the diffusion parameter D_M and the velocity field parameter $\boldsymbol{v}$ which afterwards can be used to deduce the actual chemoattractant distribution on the domain and recover some of the intrinsic parameters of the chemotactic function χ.

8.1 Model And Data Preparation

Before we can proceed with the parameter estimation of the full model, we need to make adjustments to the model and to the observation data.

Data Preparation

The data acquired through TrackMate© as described in Section 1.1 are microscopic data in form of trajectories of immune cells and tumour cells. More precisely we have two sets $\mathcal{M}$ and $\mathcal{T}$ containing the tuples $\left(k, x, y, T^{frame}\right)$ labelling a cell with a number $k \in \mathbb{N}$, its location (x, y) relatively to the corresponding video crop of the laboratory experiment (see Figure 1.8), and the frame time T^{frame} when it was registered, from $\text{frame} = 1, \ldots, 720$, where frames are 2 minutes apart from each other, thus $T^{\text{frame}} = 2 * \text{frame}$ min.
Being able to follow the trajectories of each cell provides us with additional information which can be used in certain methods such as time weighted kernel density estimate [143] and directed kernel density estimation for time series [86] but in our framework it is sufficient to have a density field for

each time frame T^{frame} for which we can use the kernel density estimation as described in Section 7.1.
However there are two complications which need to be addressed in order to have a consistent transformation between the trajectories and the density estimates for each time frame T^{frame}: boundary conditions and non-constant mass.
As our data from the OOC experiment were acquired not from the whole chip domain but only from a small region (the region of the video recording), we do not have homogeneous Neumann boundary conditions as we assumed in the numerical experiment in Chapter 3. At each boundary of the domain cells can pass through freely.
This has two consequences which require a modification of our KDE-PE method procedure.

1. Mass transfer through boundaries

The first consequence of not having homogeneous Neumann boundary conditions at the outer boundary $\partial_{out}\Omega$ is that the number of cells is not conserved due to the income and outcome of the cells at the boundaries.
Up to this point we have applied the kernel density estimation (7.21) as a method to estimate probability density functions f, i.e

$$\int_\Omega f(x,y,t)d\Omega = \int_\Omega f(x,y,0)d\Omega = 1 \quad \text{for all } t \in [0,T]. \tag{8.8}$$

This condition is not true for the solution $u(x,y,t)$ of a convection-diffusion equation for other than homogeneous Neumann boundary conditions.
For this reason we need to modify the kernel density estimate to take into account the change of mass due to the flux of cells at the boundaries.

Definition 8.1.1 (Kernel Density Estimate for non-constant total mass)**.** *Let $\mathcal{X}^l = (k,x,y,t_l)$ be a set of microscopic observation data of cells at time t_l, where each cell is labelled with $k \in \mathbb{N}$, positioned at (x_k^l, y_k^l) at time t_l. Then we define the kernel density estimate $\widehat{u}(x,y,t_l)$*

$$\widehat{u}_H(x,y,t_l) = \frac{1}{\det H}\sum_{k=1}^{M_l} K_H\left(\left(x-x_k^l, y-y_k^l\right)\right) \tag{8.9}$$

with bandwidth matrix $H \in \mathbb{R}^{2\times 2}$, where $M_l \in \mathbb{N}$ is the number of total cells at time t_l in the microscopic observation data $\mathcal{X}$.

Remark 8.1.2. *The difference between the definition of the multivariate kernel density estimate (7.21) and (8.9) is that the latter is not normalized. In (8.9) we take into account that the number of cells in the domain Ω can change and thus the total mass of cells.*

Also, although we have different density estimates $\widehat{u}_H(x, y, t_l)$ (8.9) for each time t_l, where the number of cells varies with M_l, we choose a constant bandwidth matrix H for all times t. This way we avoid creating a too huge discrepancy of density estimates between times t. However this requires that the number of cells M_l for each time t_l is relatively large.

Remark 8.1.3. *The density estimate $\widehat{u}_H(x, y, t_l)$ (8.9) is defined on a domain $\Omega \subset \mathbb{R}^2$, and as such needs to be modified for the domain $\Omega = [0, L_x] \times [0, L_y]$ to reduce the boundary effects.*
The boundary at $\{L_x\} \times [0, L_y]$ is closed except at the entry areas of the one-dimensional channels. However due to much faster migration of cells through the channels, we assume also here closed boundaries.
Hence the reflective method (7.31) can be used for $\{L_x\} \times [0, L_y]$.
The outer boundaries $\delta_{out}\Omega$ are open and allow the income and outcome of cells, thus a special treatment is not necessary. A possible improvement can be to take advantage of the additional information from the trajectories of each cell to improve the boundary values but we will not apply this idea in the further calculations.

2. Boundary Condition

Firstly, we need to equip the full model (1.45) defined at the domain $\Omega := [0, L_x] \times [0, L_y]$ with other than homogeneous Neumann boundary conditions: Dirichlet boundary conditions or non-homogeneous Neumann boundary conditions.
In the following we will refer with $\partial_{out}\Omega := \{0\} \times [0, L_y] \cup [0, L_x] \times \{0, L_y\}$ to the outer boundaries of the domain Ω which are open and allow cells to move through.

Dirichlet Boundary Condition

For the sake of an easy adjustment to our numerical methods we focus on Dirichlet boundary conditions

$$u(x, y, t)|_{\partial_{out}\Omega} = g(x, y, t), \tag{8.10}$$

for an unknown function $g : \partial_{out}\Omega \times [0, \infty) \to \mathbb{R}$ defined on the outer boundary $\partial_{out}\Omega$. From a finite difference point of view, we need to modify the finite difference scheme for the full model (3.24) with Dirichlet boundary conditions (8.10). This can be easily done by setting the values $u^n_{i,j}$ on the outer

boundary

$$u_{i,j}^{n+1} = g\left(x_i, y_j, t_{n+1}\right) = g_{i,j}^{n+1}, \quad \text{for} \begin{cases} i = N_x + 1, \quad j = 0, \ldots, N_y + 1, \\ j = \{0, N_y + 1\}, \quad i = 0, \ldots, N_x + 1. \end{cases} \tag{8.11}$$

For the unknown function g we can use the kernel density estimates $\widehat{u}_H(x, y, t_l)$ (8.9) at the outer boundary $\partial_{out}\Omega$ such that

$$u_{i,j}^n = g_{i,j}^n = \widehat{u}_H(x_i, y_j, t_l), \quad \text{if } t_n = t_l. \tag{8.12}$$

For the in-between time values, a linear interpolation can be used between the density estimates for each time l.

Non-Homogeneous Neumann Boundary Condition

We can also choose non-homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial \boldsymbol{n}}\left(x, y, t\right)\Big|_{\delta_{out}\Omega} = p\left(x, y, t\right), \tag{8.13}$$

for an unknown flux function $p : \delta_{out}\Omega \times [0, \infty) \to \mathbb{R}$ defined at the outer boundary $\partial_{out}\Omega$ to take the flux of cells into account.
For the non-homogeneous Neumann boundary conditions we discretize (8.13) and obtain ghost values

$$\begin{aligned} u_{i,-1}^{n+1} &= u_{i,1}^{n+1} - 2\triangle y p_{i,0}^{n+1}, \\ u_{i,N_y+2}^{n+1} &= u_{i,N_y}^{n+1} + 2\triangle y p_{i,N_y+1}^{n+1}, \\ u_{-1,j}^{n+1} &= u_{1,j}^{n+1} - 2\triangle x p_{0,j}^{n+1}, \end{aligned} \tag{8.14}$$

which include into the finite difference scheme (3.24). We show the additional term exemplary for u^{n+1}_{i,N_y+1} and get with (8.14)

$$\begin{aligned}
u^{n+1}_{i,N_y+1} \quad = \quad u^{n}_{i,N_y+1} \quad & +D\mu_x \left[\frac{\left(u^n_{i+1,N_y+1} - 2u^n_{i,N_y+1} + u^n_{i-1,N_y+1}\right)}{2} + \frac{\left(u^{n+1}_{i+1,N_y+1} - 2u^{n+1}_{i,N_y+1} + u^{n+1}_{i-1,N_y+1}\right)}{2} \right] \\
& +D\mu_y \left[\left(u^n_{i,N_y} - u^n_{i,N_y+1}\right) + \left(u^{n+1}_{i,N_y} - u^{n+1}_{i,N_y+1}\right) \right] \\
& -\tfrac{\lambda_x}{2} \left(\widehat{f}^{x,n}_{i+1,N_y+1} u^n_{i+1,N_y+1} - \widehat{f}^{x,n}_{i-1,N_y+1} u^n_{i-1,N_y+1} \right) \\
& +\lambda_y \left(\widehat{f}^{y,n}_{i,N_y} u^n_{i,N_y} + \widehat{f}^{y,n}_{i,N_y+1} u_{i,N_y+1} \right) \\
& +\tfrac{\lambda_x}{2} \left(|\widehat{f}^{x,n}_{i+1,N_y+1}| u^n_{i+1,N_y+1} - 2|\widehat{f}^{x,n}_{i,N_y+1}| u^n_{i,N_y+1} + |\widehat{f}^{x,n}_{i-1,N_y+1}| u^n_{i-1,N_y+1} \right) \\
& +\lambda_y \left(|\widehat{f}^{y,n}_{i,N_y}| u^n_{i,N_y} - |\widehat{f}^{y,n}_{i,N_y+1}| u^n_{i,N_y+1} \right) \\
& \underbrace{+D\frac{\triangle t}{\triangle y} \left(p^n_{i,N_y+1} + p^{n+1}_{i,N_y+1} \right)}_{\text{additional boundary flux term}} .
\end{aligned} \tag{8.15}$$

We can hence modify the boundary values such that

$$\begin{aligned}
u^{n+1,\text{non-homogeneous}}_{i,0} &= u^{n+1,\text{homogeneous}}_{i,0} - D\tfrac{\triangle t}{\triangle y} \left(p^n_{i,0} + p^{n+1}_{i,0} \right), \\
u^{n+1,\text{non-homogeneous}}_{i,N_y+1} &= u^{n+1,\text{homogeneous}}_{i,N_y+1} + D\tfrac{\triangle t}{\triangle y} \left(p^n_{i,N_y+1} + p^{n+1}_{i,N_y+1} \right), \\
u^{n+1,\text{non-homogeneous}}_{0,j} &= u^{n+1,\text{homogeneous}}_{0,j} - D\tfrac{\triangle t}{\triangle x} \left(p^n_{0,j} + p^{n+1}_{0,j} \right), \\
u^{n+1,\text{non-homogeneous}}_{0,0} &= u^{n+1,\text{homogeneous}}_{0,0} - D\triangle t \left(\tfrac{1}{\triangle x} + \tfrac{1}{\triangle y} \right) \left(p^n_{0,0} + p^{n+1}_{0,0} \right), \\
u^{n+1,\text{non-homogeneous}}_{0,N_y+1} &= u^{n+1}_{0,N_y+1,\text{homogeneous}} - D\triangle t \left(\tfrac{1}{\triangle x} - \tfrac{1}{\triangle y} \right) \left(p^n_{0,N_y+1} + p^{n+1}_{0,N_y+1} \right),
\end{aligned} \tag{8.16}$$

for $(0, j)$ for $j = 1, \ldots, N_y$ and $(i, 0)$, $(i, N_y + 1)$ for $i = 1, \ldots, N_x + 1$.
To define the values of flux function p at the boundary we need information about the flux at the boundary. However no observation data are available outside the domain.
Nevertheless we propose an approach for the case where observation data are available outside the computational domain Ω.
Data availability outside the domain is equivalent to the availability of the ghost values $u^{n+1}_{i,-1}, u^{n+1}_{i,N_y+1}, u^{n+1}_{-1,j}$, ergo we can approximate the boundary flux $p^{n+1}_{i,j}$ with a finite difference using the available ghost values.

Bandwidth

Lastly, we need to define the bandwidth matrix H.
Since we plug in H as additional parameters as described in the KDE-PE

method, we use the a-prior bandwidth matrix $\widehat{H}_0$ obtained by the Rule-of-Thumb method (7.42).
In Figure 8.1 the density fields obtained according to the tracked trajectories of the cells are shown for different times t.
Although the a-priori bandwidth matrix $\widehat{H}_0$ has been used, one can clearly deduce the dynamical behaviour of the immune cells M, showing a diagonal migration from the top right side of the right chamber Ω^r towards the left chamber Ω^l through the one-dimensional channels which we have not considered in our density estimate since no cell tracking is available within the channels.
As the immune cells continue their migration diagonally on the left chamber leaving the domain, we can see a accumulation of immune cell density M in high concentration of tumour cells T.
These observations made solely based on the visual representation of the kernel density estimates in Figure 8.1 correspond with the observation of the trajectories of the cells.

8.2 Model Inference and Comparison

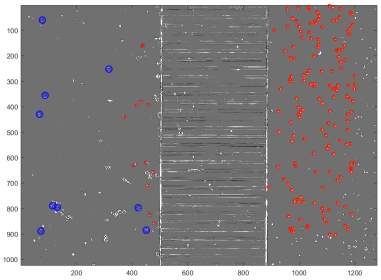
The introduction of different chemotactic terms $\widehat{f}$ in chapter 1.6 describes the different characteristics and intrinsic mechanics of the chemotactic movement of immune cells M in regards to the chemoattractant concentration ϕ.
We have presented five different chemotactic terms (1.52)-(1.56) for the OOC model (1.45)-(1.47) with different model parameters to describe the dynamical behaviour of the laboratory experiment on the OOC [31].
Fitting the available observation data u_{data} to the models allows us to estimate the intrinsic model parameters Θ but also gives us a measurement of the L^2-norm error between the data and the model solution through the objective functional of the parameter estimation problem with

$$K_0(\Theta) = \| F(\Theta) - u_{data} \|_2^2 . \tag{8.17}$$

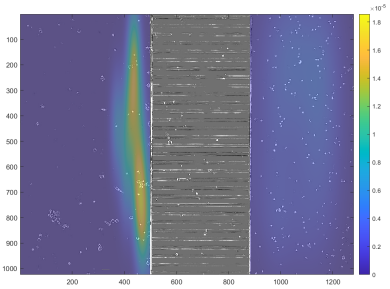
Choosing the model with the lowest error would be the "best fit" candidate among all the other models and consequently describes the model best.
However, it is also logical that models that involve more model parameters and hence more degrees of freedom, allow in many cases a better fit, i.e a smaller discrepancy between the observation data and the numerical solutions of the model [124].
The question that arises is which model does describe the actual reality in the best way?
We remind that we have two goals in mathematically modelling:

Tracked cells
Immune (red) and tumour (blue)

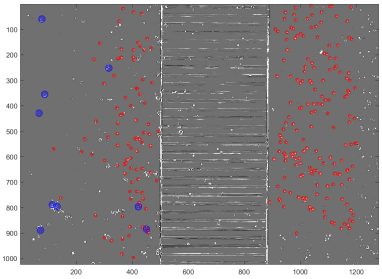
Density field of immune cells



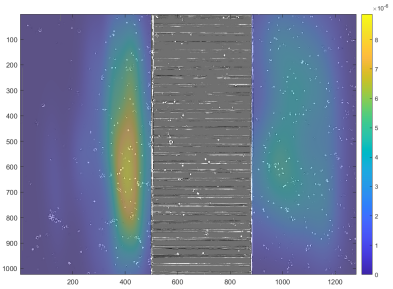
(1): At frame 1 ($T = 0min$)



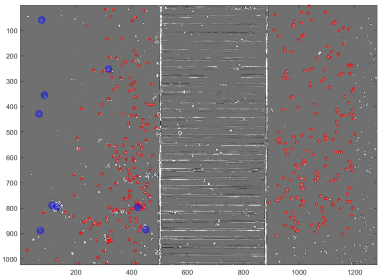
(2): At frame 1 ($T = 0min$)



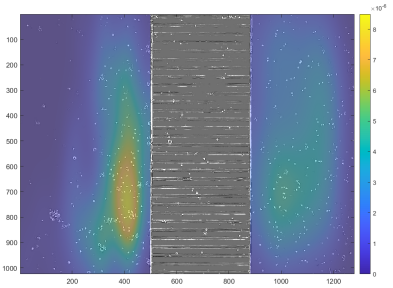
(3): At frame 200 ($T = 398min$)



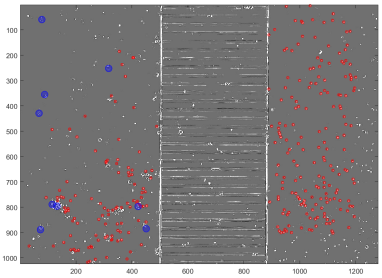
(4): At frame 200 ($T = 398min$)



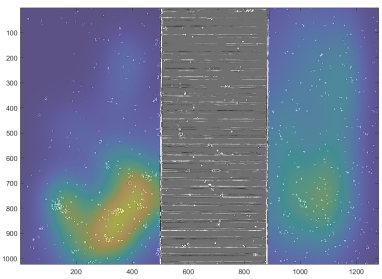
(5): At frame 400 ($T = 798min$)



(6): At frame 400 ($T = 798min$)



(7): At frame 720 ($T = 1438min$)



(8): At frame 720 ($T = 1438min$)

Figure 8.1: left: Tracked immune and tumour cells of the video recording of the experiment through TrackMate©. Right: Computed density fields by using the kernel density estimation on the available cell location data.

1. The development of a mathematical model able to describe the actual experiment in a reliable way and keeping the complexity as small as possible.

2. The model should be able to predict future events and unseen observations such that for different laboratory settings, thus different model parameters, the model is still able to describe the real behaviour in an approximated way.

These two objectives result in the model selection for inference and the model selection for prediction [44]. Thus a good model must be based on more than just the error. For this reason additional statistical measurements must be taken, which are known as information criteria [2], which are designed to balance between model complexity and good fit.
Information criteria are methods used to rank models and are derived from decision theory and the minimization of some measure of information loss [2].
The model that corresponds to the lowest information criteria value is taken to be the best model from that given stand point.
There are a variety of information criteria to be found in the literature which have benefits and disadvantages over one another, but we will restrict ourselves with the two most commonly used criteria: The AIC (Akaike information criterion) [3, 145] and BIC (Bayesian information criterion) [124].
In order to apply these criteria on our models, we first need to define the likelihood function $\mathcal{L}$ for the models.

Definition 8.2.1 (Likelihood Function [106])**.** *Let $u_{i,j}^n \approx u(x_i, y_j, t_n)$ be the approximated solution of the convection diffusion equation (6.1) with the finite difference scheme (3.24) with model parameter set Θ. Furthermore let $\mathcal{X}$ be the set of the microscopic observation data containing the tuples (x_k, y_k, t_k) representing the data location point (x_k, y_k) at time t_k for $k = 1, \ldots, N$. Then the likelihood of Θ can be calculated for each data point as*

$$P((x_k, y_k, t_k)|\Theta) := \frac{u(x_k, y_k, t_k)}{\int_\Omega u(x, y, t_k)\, d\Omega}. \tag{8.18}$$

The Likelihood function $\widehat{L}$, which is the likelihood over all microscopic observation data, can be calculated through the product as

$$\widehat{L}(\Theta) := \prod_{k=1}^{N} P((x_k, y_k, t_k)|\Theta) \tag{8.19}$$

and the log-likelihood function $\ln \widehat{L}$ *as the sum*

$$\ln \widehat{L}\left(\Theta\right) = \sum_{k=1}^{N} \ln P\left(\left(x_k, y_k, t_k\right)|\Theta\right) = \sum_{k=1}^{N} \ln \frac{u\left(x_k, y_k, t_k\right)}{\int_\Omega u\left(x, y, t_k\right) d\Omega} \tag{8.20}$$

The Likelihood function $\widehat{L}$ is equal to the probability that the microscopic data $\mathcal{X}$ are observed when the model parameter is Θ.
We proceed with the definitions of two information criteria.

Definition 8.2.2 (Akaike information criterion (AIC) [3])**.** *Let* $N \in \mathbb{N}$ *be the number of model parameters* $\Theta \in \mathbb{R}^N$ *that need to be estimated and* $\ln \widehat{L}$ *the log-likelihood function for the model (8.20). Then the AIC value of the model is defined as*

$$\mathrm{AIC} = 2N - 2\ln \hat{L}, \tag{8.21}$$

where the favourable model corresponds to the one with the minimum AIC value.

Definition 8.2.3 (Bayesian information criterion (BIC) [2])**.** *Let* $N \in \mathbb{N}$ *be the number of model parameters* $\Theta \in \mathbb{R}^N$ *that need to be estimated,* $n \in \mathbb{N}$ *the sample size of the microscopic observation data* $\mathcal{X}$ *and* $\ln \widehat{L}$ *the log-likelihood function for the model (8.20). Then the BIC value is defined as*

$$\mathrm{BIC} = N \ln n - 2\ln \widehat{L}, \tag{8.22}$$

where the favourable model corresponds to the one with the minimum BIC value.

Comparing both definitions (8.21) and (8.22) the difference is found at the factor of N where in AIC (8.21) it is 2, and in the definition of BIC (8.22) it is $\ln n$ and hence depends on the sample size n.
This means that BIC penalizes models with a higher number of model parameters much stronger with respect to AIC [124].
In the following example we will investigate, in regards to our OOC-model (1.45), whether it is possible to determine the correct type of chemotactic function $\widehat{f}$ present in the two-dimensional chemotaxis equation (3.5).

Example 12. *We consider the two-dimensional double-parabolic equation for chemotaxis (3.5)*

$$\begin{cases} \partial_t u &= D_u \Delta u - \mathbf{div}(f), \\ \partial_t \phi &= D_\phi \Delta \phi + \alpha T - \beta \phi, \end{cases} \tag{8.23}$$

on the two-dimensional domain $\Omega = [0, 50] \times [0, 50]$ *with cell density* $u(x, y, t)$ *and concentration of chemoattractant* $\phi(x, y, t)$*, growth rate* $\alpha > 0$*, consumption rate* $\beta > 0$*. In this model (8.23) we consider the tumour cell density*

$T(x,y)$ *to be stationary and to serve as a source for the chemoattractant* ϕ*. The initial-boundary conditions are*

$$\begin{aligned} u_0(x,y) &= \begin{cases} 10^{-2}, & \text{for } 37.5 \leq x \leq 50,\ y \in [0,50] \\ 0 & \text{else,} \end{cases} \\ \phi_0(x,y) &= 0, \\ \left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_{\partial\Omega} &= 0, \\ \left.\frac{\partial \phi}{\partial \boldsymbol{n}}\right|_{\partial\Omega} &= 0 \end{aligned} \tag{8.24}$$

and with chemoattractant producing tumour cells T

$$\begin{aligned} T(x,y) &= 10^{-3} e^{-\frac{1}{100}\left((x-5)^2+(y-15)^2\right)} + 10^{-3} e^{-\frac{1}{100}\left((x-15)^2+(y-10)^2\right)} \\ &\quad + 10^{-3} e^{-\frac{1}{100}\left((x-45)^2+(y-15)^2\right)}. \end{aligned} \tag{8.25}$$

For the chemotactic functions $f = \widehat{f}u = \chi u$ *we consider the following five functions as already presented in Section 1.6:*

I ***Basic model***

$$\chi := k_1 \tag{8.26}$$

with cellular drift velocity $k_1 \in \mathbb{R}$*.*

II ***Receptor saturation*** *[140]*

$$\chi(\phi) := \frac{k_1}{(k_2+\phi)^2} \tag{8.27}$$

with $k_2 \in \mathbb{R}_{>0}$ *representing the receptor dissociation constant.*

III ***Overcrowding*** *[71])*

$$\chi(u,\phi) := \frac{k_1}{(k_2+\phi)^2}\left(1 - \frac{u}{u_{max}}\right) \tag{8.28}$$

with maximum cell density $u_{max} \in \mathbb{R}_{>0}$*.*

IV ***Interaction*** *[49]*

$$\widehat{f} := \frac{k_1}{(k_2+\phi)^2}\nabla\phi + \frac{\eta_1}{1+\eta_2 u}\nabla u \tag{8.29}$$

with repulsion strength between cells $\eta_1 \in \mathbb{R}$ *and interaction strength* $\eta_2 \in \mathbb{R}_{>0}$

V ***Interaction*** *[49] (chemoattractant free)*

$$\widehat{f} := \frac{\eta_1}{(\eta_2 + T)^2} \nabla T. \tag{8.30}$$

For the following investigation we assume that the diffusion coefficient $D_\phi = 200$ *and* $\alpha = 1, \beta = 0.0001$ *are known and fixed, whereas only the diffusion coefficient* $D_u > 0$ *and the parameters in the chemotactic functions* $\widehat{f}$ *(8.26)-(8.30) are unknown.*
For the model calibration we create artificial observation data of (8.23) with the receptor saturation as the chemotactic function (8.27) with the parameters k_1 *and* k_2 *as indicated in table 3.1 for the times* t *until* $t = 0.1$ *and add* 2% *Gaussian noise.*
We want to remark that the artificial observation data not only contain data of $u(x, y, t)$ *but also of the chemoattractant* $\phi(x, y, t)$ *while in the real experimental data no information about the chemoattractant is available.*
We then proceed with the parameter estimation of (8.23) with all five chemotactic functions (8.26)-(8.30) against the data to recover the parameters and calculate the AIC (8.21) and BIC (8.22) which are presented in Figure 8.2.

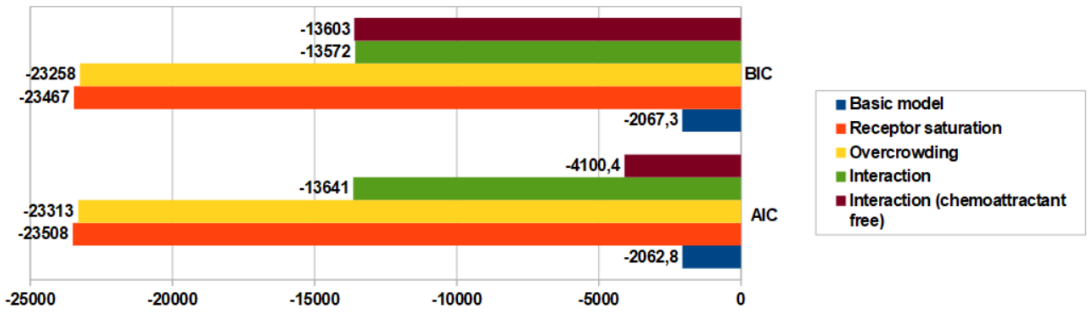


Figure 8.2: Different AIC and BIC values of the 2D-doubly parabolic model (8.23) with different chemotactic functions (8.26)-(8.30) against noisy artificial data of the receptor saturation model (8.27).

Conclusion

The comparison of the AIC and BIC values in Figure 8.2 shows that the Receptor-Saturation model (8.27) has the minimum (largest negative) AIC and BIC value compared to the other models, indicating that even with introduced noise, the values correctly indicate the model that represents the artificial data which have been used.
Furthermore we can deduce that among all the other models, the AIC and BIC value of the Overcrowding model (8.28) is very close to the AIC and BIC value

of the original model. This can be explained by the fact that the only difference between the Receptor-Saturation model and the Overcrowding model is the additional term of $\left(1 - \frac{u}{u_{max}}\right)$ with parameter u_{max} in the Overcrowding model which converges towards the Receptor-Saturation model for large u_{max}. However, the penalization of the additional parameter u_{max} in the definition of the AIC and BIC causes to favour the model with less parameters, hence the correct receptor-saturation model.
The Basic model (8.26) on the other hand has the least negative number compared to the other models which indicates that it is the least favourable model despite having the lowest number of model parameters.
For the other two models, interaction model (8.29) and chemoattractant free interaction model (8.30), the AIC and BIC values are less negative and hence less favourable models. The interaction model due to the larger number of model parameters and the chemoattractant free interaction model due to the larger error despite the lower number of parameters. We also want to remark that the ranking of models based on the AIC values and BIC values are identical except for the chemoattractant free Interaction model (8.30) which is the second least favourable model according to the AIC values but third least favourable based on its BIC value. This can be explained by the fact that BIC penalizes the number of parameter more severe than the AIC such that the chemoattractant free Interaction model with $N = 3$ model parameters are favoured over the interaction model with $N = 5$ model parameters.
We can conclude from this example that for the two-dimensional double-parabolic model for chemotaxis (8.23) with chemotactic terms (8.26)-(8.30) under the assumptions made in example 12 it is possible with the AIC and BIC information criterion to differentiate between the models and to determine the underlying chemotactic term of which the artificial data have been derived from.

8.3 Final Results

We summarize the parameter estimation settings for the OOC-model taking into account all previously made assumptions and simplifications:

- The simplified model

$$\partial_t M \quad = \quad D_M \Delta M - \operatorname{div}(\boldsymbol{v} M) \tag{8.31}$$

  with immune cell density M, diffusion coefficient $D_M > 0$ and unknown velocity field function $\boldsymbol{v}$

- We consider several regions Ω_{crop} of the left chamber Ω_l and right chamber Ω_r where we apply the parameter estimation method to reduce the

computational costs involved and average the estimated parameters to obtain averaged values for the whole domain.

- The objective functional of the parameter estimation method is (7.52)

$$K(\Theta, H) = \| F(\Theta) - u_{data}(H) \|_2^2 + \lambda \| \Theta \|_2^2 + p_H \| H - H_{ROT} \|_2^2, \quad (8.32)$$

  defined in the regions Ω_{crop}.

- The boundary conditions for the boundary $\partial\Omega_{crop}$ are the Dirichlet boundary conditions defined in (8.12).

- In order to further reduce the complexity of the system, we assume that the velocity field $\boldsymbol{v}$ is constant in time. This might not be true for long time scale problems but for the observation data available to us for a time range of just 24 hours, this assumption is valid, also due to the fact that tumour cells are producers of chemoattractant ϕ and mostly stationary and that the diffusion of chemical is slow enough to not affect the chemoattractant gradient too much during the time interval we are investigating.

Initialization of Model Calibration

As for the initial guess for the parameter estimation method, we will use the parameter values used in the numerical experiment in chapter 3 which can be found in table 3.1 and only allow the range of parameter values to be around the same magnitude. For the diffusion coefficient of the immune cells D_M we assume initial value $D_M = 9 \cdot 10^2 \frac{\mu m^2}{s}$ and allow the parameter only to range between 10^2 and 10^4.
Furthermore we want to remark that a direct model inference with AIC (8.21) and BIC (8.22) can not be investigated because of the simplification made. The model does not directly depend on the chemotactic terms χ, anymore, which defines the different models we are using. However a modification of the information criteria can be made that takes into account the relation between the velocity field $\boldsymbol{v}$ and the chemotactic term χ which we will not investigate here.

Domain Decomposition

In order to reduce computational cost, we will divide the computational domain Ω_{crop} into smaller regions Ω_i and conduct the parameter estimation on each region separately. The velocity field of the whole domain Ω_{crop} is henceforth defined as the union of the velocity fields defined on each subdomain Ω_i.

For the diffusion coefficient $D_M > 0$ we average its estimate over all subdomains.

Initialization of Velocity Field

Because the initial guess for the parameter estimation method is crucial to obtain feasible and satisfactory results, we propose the following initial guess for the velocity field based on the following assumptions:

i) According to the trajectories of the cells (see Figure 1.8 on the left) we can assume that in the right chamber the velocity field is constantly diagonally oriented. This is also partly true for the left chamber. This can be explained by the fact that the tumour cells are initially located in the reservoir, hence the highest concentration of chemoattractant must be present in that location. Considering the location of the domain Ω_{crop} (see Figure 1.8 on the right) this explains the diagonal movement of immune cells.

ii) Tumour cells are present in the left chamber, producing chemoattractant as well which causes some immune cells to migrate towards them. The evolution of chemoattractant is based on the parabolic reaction-diffusion equation (1.45).

Considering assumption i) we can propose that one component of the velocity field is defined as the constant velocity field

$$v^{(1)} := k_{v_1} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \tag{8.33}$$

with unknown constant $k_{v_1} > 0$.
The second component is based on assumption ii) as we assume

$$v^{i,(2)} := -k_{v_2} \begin{pmatrix} (x - x_i)\, e^{-k_{v_3}\left((x-x_i)^2+(y-y_i)^2\right)} \\ (y - y_i)\, e^{-k_{v_3}\left((x-x_i)^2+(y-y_i)^2\right)} \end{pmatrix} \tag{8.34}$$

with unknown constants $k_{v_2}, k_{v_3} > 0$ and tumour positions (x_i, y_i).

For the right chamber we assume that only $v^{(1)}$ (8.33) is present, where for the left chamber we have a superposition of $v^{(1)}$ and the summation over all tumour positions (x_i, y_i) of $v^{i,(2)}$ (8.34).

Model Calibration and Preliminary Results

The parameter estimation consists of two parts.

- We apply the parameter estimation method with objective functional (8.32), observation data (8.9) on several subdomains Ω_i with Dirichlet boundary conditions (8.10) for the parameter $\Theta_i = (D_M, k_{v_1}, k_{v_2}, k_{v_3})$ for each subdomain separately.

- We average D_M over all subdomains and apply a second parameter estimation method with objective functional (8.32), observation data (8.9) on the whole left and right chamber Ω_l, Ω_r for the parameter $\Theta = \boldsymbol{v}$. As initial guess the velocity field $\boldsymbol{v}$ obtained from the first parameter estimation methods is used.

The corrected velocity field $\boldsymbol{v}$ with the averaged diffusion coefficient will be used for the model (8.5).
The obtained density solutions M will then be transformed into microscopic data with the acceptance-rejection method and compared with the real cell positions.
For the averaged diffusion coefficient of the immune cells we obtain $D_M = 9,8892 \cdot 10^3 \frac{\mu m^2}{s}$.
In Figure 8.3 we directly compare visually the tracked immune and tumour cells on the left with immune cells artificially derived from the model (8.31) with the parameter estimation solutions for diffusion coefficient D_M and velocity field, through acceptance-rejection method. We chose $N = 230$ artificial immune cells for both the left and right chamber which explains the larger number of immune cells in the left domain compared to the tracked immune cells. As shown in Figure 8.3, the artificial immune cells coincide with the actual tracked immune cells. However we need to remark, that we cannot claim that this simplified model is reflecting the real model. The simplification made are based solely on assumptions made from the results of the laboratory experiment described in [15, 31, 141]. Additionally, no data are available about the chemoattractant distribution and the chemotactic movement of cells is only deduced by assumption i) and ii).

Reconstruction of Chemoattractant Distribution

The final part of the model calibration is dedicated to the reconstruction of the chemoattractant distribution ϕ. As mentioned before, the laboratory experiments described in [15, 31, 141] do not provide any data on the chemoattractant distribution. This has motivated the use of a simplified model (8.31) for the model calibration where the chemotactic function f has been replaced by a general velocity field $\boldsymbol{v}$ which has been further assumed to be constant in time t to reduce computational costs. However, we can use the relation

$$\boldsymbol{v} := \widehat{f} = \chi \nabla \phi \tag{8.35}$$

to infer additional information about the chemoattractant distribution ϕ.
For that we use the estimation of the model calibration for $\boldsymbol{v}$ from the previous section in a second parameter estimation problem

$$\min_{\Theta} K(\Theta) = \min \parallel \boldsymbol{v} - \chi \nabla_{\Delta x, \Delta y} \phi \parallel_2^2 + \lambda_1 \parallel \phi \parallel_2^2 + \lambda_2 \parallel T\phi \parallel_2^2 \tag{8.36}$$

with model parameter set $\Theta = (\phi, \chi)$, Tikhonov regularization matrix T as second-order-difference operator, and regularization parameters $\lambda_1, \lambda_2 > 0$, where $\nabla_{\Delta x, \Delta y}$ denotes the discretization of ϕ:

$$\nabla_{\Delta x, \Delta y} \phi = \begin{pmatrix} \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x} \\ \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y} \end{pmatrix}. \tag{8.37}$$

In the same way as for the model calibration for the diffusion D_M and velocity field $\boldsymbol{v}$, we apply the parameter estimation method on several subdomains Ω_i of Ω_l and Ω_r separately.
Regarding the chemotactic sensitivity function χ, we choose for each subdomain Ω_i a constant $\chi_i \in \mathbb{R}$ in order to reduce the complexity of the parameter estimation method.
Furthermore, it is also necessary to specify boundary conditions on each subdomain Ω_i for ϕ.
We make use of the assumption i) previously used for the initialization of the velocity field in (8.33) and set for the left and right chamber Ω_l, Ω_r

$$\begin{aligned} \phi_l|_{\delta\Omega_l} &= -k_{v_1}(x - y) + C_{\phi_l}, \\ \phi_r|_{\delta\Omega_r} &= -k_{v_1}(x - y) + C_{\phi_r} \end{aligned} \tag{8.38}$$

with constant $k_{v_1} \in \mathbb{R}_{\geq 0}$ which has already been estimated in the model calibration and $C_{\phi_l}, C_{\phi_r} \in \mathbb{R}_{\geq 0}$ which can be chosen arbitrarily but require

$$\begin{aligned} k_{v_1}(x - y) + C_{\phi_l} > 0, \quad &\text{for } (x, y) \in \delta\Omega_l, \\ k_{v_1}(x - y) + C_{\phi_r} > 0, \quad &\text{for } (x, y) \in \delta\Omega_r, \end{aligned} \tag{8.39}$$

in order to guarantee $\phi > 0$ for the whole domain Ω. We choose $C_{\phi_l} = C_{\phi_r} = 10^5$.

Results: The reconstructed chemoattractant distribution ϕ is shown in Figure 8.4. The optimization methods (MGPE with Levenberg-Marquardt and Gauss-Newton smoother) have not been able to converge due to the possible numerous local minima, hence a variety of randomized initial guesses Θ_0 have been used and the solution of the parameter estimation with smallest

value of $K(\theta)$ chosen.
From the reconstructed chemoattractant field we can deduce that in the right chamber, the concentration is spreading from the bottom-left corner and flows diagonally through the whole domain. The spreading of ϕ is not regularly though and shows regions of much lower chemoattractant such as in the top-right corner.
Furthermore, we can also see that the chemoattractant is entering the right domain through the micro-channels.
Regarding the left chamber, the distribution of chemoattractant ϕ is not as clear.
There are small areas of high concentrations in the bottom part, which corresponds with the location of tumour cells. However there seems to be an unusual low concentration at the bottom border of the left chamber and in the middle.
Additionally, a higher concentration is present around the top area of the left chamber, despite the fact that a very small number of tumour cells are located in this area which could have caused an increase of the concentration of chemoattractant.
Overall, the reconstructed chemoattractant distribution ϕ corresponds with the assumptions made on the velocity field $\boldsymbol{v}$ and the dynamics shown in the observation data of the immune cells but we need to remark that we were not able to obtain convergence and due to the chosen initial guesses used for the parameter estimation of ϕ we have induced a bias in the reconstruction of the chemoattractant distribution. This makes the preliminary chemoattractant distribution reconstruction only to a certain extent meaningful.
Some regions also show unusual low and high concentration of ϕ which might be caused by not estimated values of ϕ in regions where not much observation data have been available.
Additional observation data and more information about the chemoattractant distribution in the laboratory experiments on the OOC can improve the general reconstruction of chemoattractant.

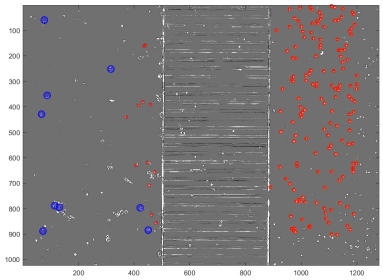
8.4 Computational Time

All the numerical simulations were performed in MATLAB© and performed on an Intel(R) Core(TM) i7-10750H CPU @ 2.6GHz and on a NVIDIA RTX 2070. The simulation was performed on the complete domain with $[1361 \times 1702]\mu m$ (two chambers) for mesh grid size $\triangle x = \triangle y = 2.5$.
The time step size is chosen to respect stability condition:
$\triangle t = 0.9 \left(D_M \frac{\triangle y^2}{\max\limits_{i,j} v_{i,j}^{x,n} \triangle y + \max\limits_{i,j} v_{i,j}^{y,n} \triangle x} + D_M \frac{\triangle x^2}{\max\limits_{i,j} v_{i,j}^{x,n} \triangle y + \max\limits_{i,j} v_{i,j}^{y,n} \triangle x} \right)$ and depends on

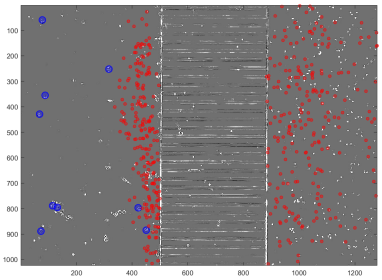
the magnitude of the velocity field. However, we can say that in average it takes 10 minutes to solve the forward problem once for time $t = 720$.
Despite the fact that the numerical computation of the solution of the forward problem (8.5) is relatively fast, the model calibration is significantly more time consuming due to the fact that the forward problem has to be computed several times during each iteration of the optimization process. More precisely the optimization method used in this work computes the gradient function for each unknown parameter. If we assume that the calculation of the solution of the forward problem takes the computational time of $t = t_{max}$ and that we choose a constant time step $\triangle t$ throughout the whole computations, then for N-unknown parameters, the forward problem has to be solved at least $2N$ times (to compute an approximation of the derivative of a parameter) for a single optimization iteration. The computational time squares for 2-dimensional problems where the velocity field $\boldsymbol{v}$ is two-dimensional. If we only consider the forward problem that contributes to the computational time for one optimization iteration, then the time is at least $(2N)^2 t_{max}$, which does not consider other operations involved in the optimization method such as matrix-vector products which are needed for solving linear equation systems inside the optimization methods. Application of multigrid reduces computational time on the coarsest grid up to 8.9 times (depending on the size of the finest grid). Additionally, the parallel computation of the sub-domains furthermore reduces computational times, making it possible to calibrate the model more efficiently.
For the multi-domain transmission model for OOC that includes one-dimensional channels, we are able to include them without any large increase in computational time, although in parameter estimation methods the introduction of additional parameters increases the overall computational time.

Tracked cells
Immune (red) and tumour (blue)

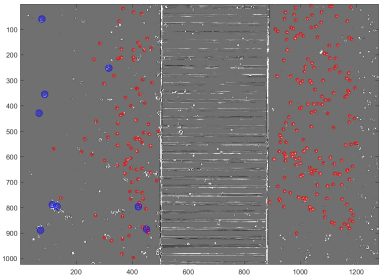
Particle representation of artificial immune cells



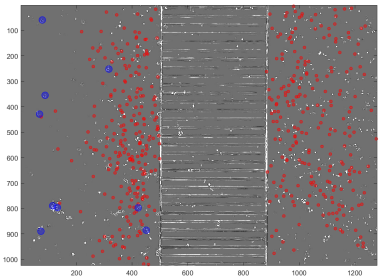
(1): At frame 1 ($T = 0$).



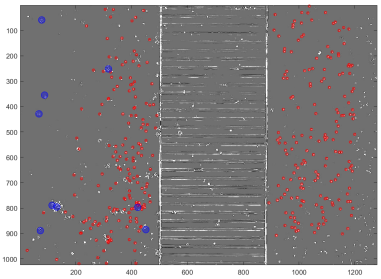
(2): At frame 1 ($T = 0$).



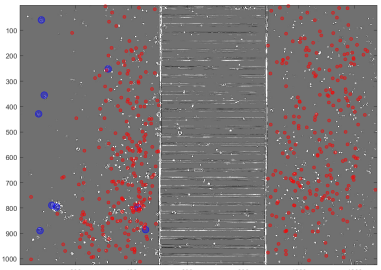
(3): At frame 200 ($T = 398min$).



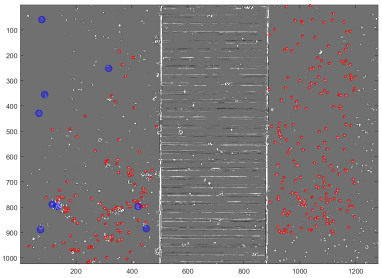
(4): At frame 200 ($T = 398min$).



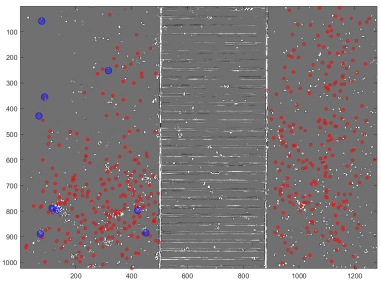
(5): At frame 400 ($T = 798min$).



(6): At frame 400 ($T = 798min$).



(7): At frame 720 ($T = 1438min$).



(8): At frame 720 ($T = 1438min$).

Figure 8.3: left: Tracked immune and tumour cell of the video recording of the experiment through TrackMate©. Right: Artificially created immune cells through model and derivation of particle representation with acceptance-rejection method for a fixed number of immune cells $N = 230$ for both the left and right domain.

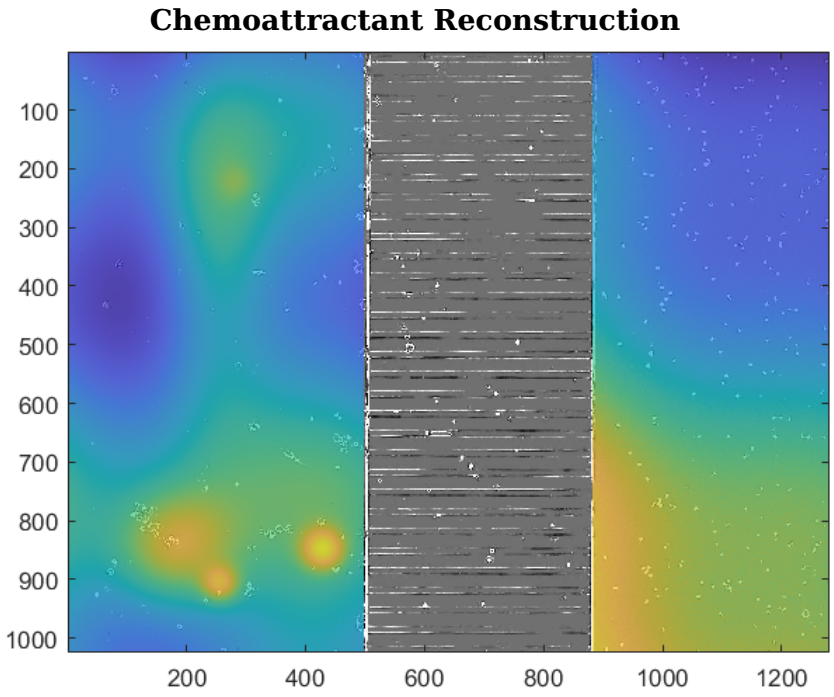


Figure 8.4: Reconstruction of chemoattractant distribution ϕ through parameter estimation with (8.36) with a-priori estimated velocity field $\boldsymbol{v}$ from (8.32). Due to not obtained convergence, the distribution is only to a limited extent meaningful. Right chamber: ϕ is concentrated in the left bottom of the domain. Left chamber: Highest concentration are concentrated around three areas in the bottom where also tumour cells are located. However, concentration peaks are much larger in the bottom area and the higher concentration in the top do not agree with the dynamics shown by immune cells M.

Part IV

Conclusion and Future Research

Conclusion and Future Research

The main feature of this thesis was the development of the following simulation tools:

- A mathematical model of chemotaxis to describe cell movements and interactions inside a microfluidic chip environment.
- A robust parameter estimation method which can be used to fit the model parameters against real microscopic data obtained in experiments.
- A reliable model inference to choose among different chemotactic functions and infer the intrinsic key features that are present.
- A particle representation in order to visually compare simulation with laboratory experiments.
- An attempt to reconstruct the chemoattractant distribution, for which no data were available.

We managed to simulate the microfluidic chip environment with its specific geometric features with two species of living cells moving in it similar to the experiment conducted on the OOC.
Moreover, we are also able to simulate more complicated situations where more than two cells and chemicals are present and modify the model to encompasses drug administrations and extend the chemotactic functions to include more intrinsic biological mechanisms.
With the inclusion of the multigrid method into the parameters estimation framework we are able to recover model parameters of highly convective dynamics which are often the case for chemotaxis and reduce computational time tremendously.
The numerical tests with increasing complexity show that our multigrid parameter estimation method with appropriate regularization is capable to recover the model parameters with a sufficiently high accuracy although during the tests it became clear that there are certain limits when noise is present and the initial guess is too far away from the actual optimal values. Especially in 2D these limits become apparent however the errors are acceptable.
By introducing the kernel density estimates into the parameter estimation framework, we successfully extended the fitting of data for microscopic data. However it becomes apparent that with a limited number of observation data, the quality of the parameter estimation suffers.
These two simulation tools, the mathematical model of the OOC and robust parameter estimation method, we tried to fit the available data in form of

cell positions to recover the model parameters. Although the data are limited to a number of cell positions with no information about the chemicals available, we managed with certain assumptions, to obtain some parameter values which can give some valuable insight about the present dynamics of the experiment.
Nevertheless, the choice of a continuous model to reproduce an experiment in a confined environment, with a relatively small number of cells is motivated by the fact that we want to describe the phenomena of immunosurveillance of cancer in tissues, where billions of cells are present.
With improved cell tracking and more data we could overcome the many challenges we faced during the data fitting process and make better inferences about the intrinsic cell mechanics within the experiment.
The simulation tool we developed are promising enough and can greatly contribute to the further advancement in the field of Organs-on-Chips.

Part V

Bibliography

Bibliography

[1] Elena Agliari et al. “Cancer-driven dynamics of immune cells in a microfluidic environment”. In: *Scientific reports* 4.1 (2014), pp. 1–15.

[2] Ken Aho, DeWayne Derryberry, and Teri Peterson. “Model selection for ecologists: the worldviews of AIC and BIC”. In: *Ecology* 95.3 (2014), pp. 631–636.

[3] Hirotugu Akaike. “A new look at the statistical model identification”. In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.

[4] Walter Alt. *Nichtlineare Optimierung: Eine Einführung in Theorie, Verfahren und Anwendungen*. Springer-Verlag, 2013.

[5] Philipp M Altrock, Lin L Liu, and Franziska Michor. “The mathematics of cancer: integrating quantitative models”. In: *Nature Reviews Cancer* 15.12 (2015), pp. 730–745.

[6] Denise Aregba-Driollet, Maya Briani, and Roberto Natalini. “Asymptotic high-order schemes for 2\times2 dissipative hyperbolic systems”. In: *SIAM Journal on Numerical Analysis* 46.2 (2008), pp. 869–894.

[7] Denise Aregba-Driollet, Maya Briani, and Roberto Natalini. “Asymptotic high-order schemes for 2\times2 dissipative hyperbolic systems”. In: *SIAM Journal on Numerical Analysis* 46.2 (2008), pp. 869–894.

[8] U Ascher and Eldad Haber. “A multigrid method for distributed parameter estimation problems”. In: *ETNA* 18 (2003), pp. 1–18.

[9] Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. “Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations”. In: *Applied Numerical Mathematics* 25.2-3 (1997), pp. 151–167.

[10] Richard C Aster, Brian Borchers, and Clifford H Thurber. *Parameter estimation and inverse problems*. Elsevier, 2018.

[11] Emmanuel Audusse et al. “A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows”. In: *SIAM Journal on Scientific Computing* 25.6 (2004), pp. 2050–2065.

[12] Murat Belge, Misha E Kilmer, and Eric L Miller. “Efficient determination of multiple regularization parameters in a generalized L-curve framework”. In: *Inverse problems* 18.4 (2002), p. 1161.

[13] Alfredo Bermudez and Ma Elena Vazquez. “Upwind methods for hyperbolic conservation laws with source terms”. In: *Computers & Fluids* 23.8 (1994), pp. 1049–1071.

[14] Dimitri P Bertsekas. “Nonlinear programming”. In: *Journal of the Operational Research Society* 48.3 (1997), pp. 334–334.

[15] Elena Biselli et al. “Organs on chip approach: a tool to evaluate cancer-immune cells interactions”. In: *Scientific reports* 7.1 (2017), pp. 1–12.

[16] Adrien Blanchet, Jean Dolbeault, and Benoît Perthame. “Two-dimensional Keller-Segel model: Optimal critical mass and qualitative properties of the solutions.” In: *Electronic Journal of Differential Equations (EJDE)* 2006 (2006), Paper–No.

[17] Lenore Blum et al. “The condition number for nonlinear problems”. In: *Complexity and Real Computation*. Springer, 1998, pp. 217–236.

[18] Jean-Pierre Boon and Brigitte Herpigny. “Model for chemotactic bacterial bands”. In: *Bulletin of mathematical biology* 48.1 (1986), pp. 1–19.

[19] Raul Borsche et al. “The scalar Keller–Segel model on networks”. In: *Mathematical Models and Methods in Applied Sciences* 24.02 (2014), pp. 221–247.

[20] Alfio Borzì and Volker Schulz. *Computational optimization of systems governed by partial differential equations*. SIAM, 2011.

[21] François Bouchut, Haythem Ounaissa, and Benoît Perthame. “Upwinding of the source term at interfaces for Euler equations with high friction”. In: *Computers & mathematics with applications* 53.3-4 (2007), pp. 361–375.

[22] Keeran J Brabazon, Matthew E Hubbard, and Peter K Jimack. “Nonlinear multigrid methods for second order differential operators with nonlinear diffusion coefficient”. In: *Computers & Mathematics with Applications* 68.12 (2014), pp. 1619–1634.

[23] Achi Brandt. “Multi-level adaptive solutions to boundary-value problems”. In: *Mathematics of computation* 31.138 (1977), pp. 333–390.

[24] Helmut Brass and Knut Petras. *Quadrature theory: the theory of numerical integration on a compact interval*. 178. American Mathematical Soc., 2011.

[25] Elishan Christian Braun, Gabriella Bretti, and Roberto Natalini. “Mass-preserving approximation of a chemotaxis multi-domain transmission model for microfluidic chips”. In: *Mathematics* 9.6 (2021), p. 688.

[26] Gabriella Bretti and Roberto Natalini. “Numerical approximation of nonhomogeneous boundary conditions on networks for a hyperbolic system of chemotaxis modeling the Physarum dynamics”. In: *Journal of Computational Methods in Sciences and Engineering* 18.1 (2018), pp. 85–115.

[27] Gabriella Bretti, Roberto Natalini, and Magali Ribot. “A hyperbolic model of chemotaxis on a network: a numerical study”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 48.1 (2014), pp. 231–258.

[28] William L Briggs, Van Emden Henson, and Steve F McCormick. *A multigrid tutorial*. SIAM, 2000.

[29] Martin Brokate et al. *Grundwissen Mathematikstudium*. Springer, 2016.

[30] Klemens Burg et al. *Partielle Differentialgleichungen und funktionalanalytische Grundlagen*. Springer, 2010.

[31] Luca Businaro et al. “Cross talk between cancer and immune cells: exploring complex dynamics in a microfluidic environment”. In: *Lab on a Chip* 13.2 (2013), pp. 229–239.

[32] John Charles Butcher and Nicolette Goodwin. *Numerical methods for ordinary differential equations*. Vol. 2. Wiley Online Library, 2008.

[33] Antonio Campo, Mohammad R Salimpour, and John Ho. “The method of lines with numerical differentiation of the sequential temperature–time histories for a facile solution of 1-D inverse heat conduction problems”. In: *Heat and Mass Transfer* 49.3 (2013), pp. 369–379.

[34] Carlo Cattaneo. “Sur une forme de l’equation de la chaleur eliminant la paradoxe d’une propagation instantantee”. In: *Compt. Rendu* 247 (1958), pp. 431–433.

[35] Matthew J Colbrook et al. “Kernel density estimation with linked boundary conditions”. In: *Studies in Applied Mathematics* 145.3 (2020), pp. 357–396.

[36] MC Comes et al. “The influence of spatial and temporal resolutions on the analysis of cell-cell interaction: a systematic study for time-lapse microscopy applications”. In: *Scientific reports* 9.1 (2019), pp. 1–11.

[37] Tine Curk, Davide Marenduzzo, and Jure Dobnikar. “Chemotactic sensing towards ambient and secreted attractant drives collective behaviour of E. coli”. In: *PLoS One* 8.10 (2013), e74878.

[38] Jing Dai and Stefan Sperlich. “Simple and effective boundary correction for kernel densities and regression with an application to the world income and Engel curve estimation”. In: *Computational Statistics & Data Analysis* 54.11 (2010), pp. 2487–2497.

[39] Kees Dekker. “Stability of Runge-Kutta methods for stiff nonlinear differential equations”. In: *CWI Monographs* 2 (1984).

[40] Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug. “Inexact newton methods”. In: *SIAM Journal on Numerical analysis* 19.2 (1982), pp. 400–408.

[41] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*. Vol. 35. Springer Science & Business Media, 2011.

[42] Ezio Di Costanzo, Roberto Natalini, and Luigi Preziosi. “A hybrid mathematical model for self-organizing cell migration in the zebrafish lateral line”. In: *Journal of mathematical biology* 71.1 (2015), pp. 171–214.

[43] Cristiana Di Russo. “Analysis and numerical approximations of hydrodynamical models of biological movements”. In: (2011).

[44] Jie Ding, Vahid Tarokh, and Yuhong Yang. “Model selection techniques: An overview”. In: *IEEE Signal Processing Magazine* 35.6 (2018), pp. 16–34.

[45] Y Dolak and T Hillen. “Cattaneo models for chemosensitive movement. Numerical solution and pattern formation.” In: *Journal of mathematical biology* 46.2 (2003), pp. 153–170.

[46] Tarn Duong and Martin Hazelton. “Plug-in bandwidth matrices for bivariate kernel density estimation”. In: *Journal of Nonparametric Statistics* 15.1 (2003), pp. 17–30.

[47] Casimir Emako et al. “Traveling pulses for a two-species chemotaxis model”. In: *PLoS computational biology* 12.4 (2016), e1004843.

[48] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*. Vol. 375. Springer Science & Business Media, 1996.

[49] Elaine A Ferguson et al. “Inference of the drivers of collective movement in two cell types: Dictyostelium and melanoma”. In: *Journal of The Royal Society Interface* 13.123 (2016), p. 20160695.

[50] Francis Filbet, Philippe Laurençot, and Benoît Perthame. “Derivation of hyperbolic models for chemosensitive movement”. In: *Journal of Mathematical Biology* 50.2 (2005), pp. 189–207.

[51] Reeves Fletcher and Colin M Reeves. “Function minimization by conjugate gradients”. In: *The computer journal* 7.2 (1964), pp. 149–154.

[52] Roseanne M Ford and Douglas A Lauffenburger. “Measurement of bacterial random motility and chemotaxis coefficients: II. Application of single-cell-based mathematical model”. In: *Biotechnology and Bioengineering* 37.7 (1991), pp. 661–672.

[53] Bengt Fornberg. “Generation of finite difference formulas on arbitrarily spaced grids”. In: *Mathematics of computation* 51.184 (1988), pp. 699–706.

[54] Joaquim Fort and Vicenç Méndez. “Wavefronts in time-delayed reaction-diffusion systems. Theory and comparison to experiment”. In: *Reports on Progress in Physics* 65.6 (2002), p. 895.

[55] Manuele Gori et al. “Investigating nonalcoholic fatty liver disease in a liver-on-a-chip microfluidic device”. In: *PLoS One* 11.7 (2016), e0159729.

[56] L Gosse. “A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms”. In: *Computers & Mathematics with Applications* 39.9-10 (2000), pp. 135–159.

[57] Artur Gramacki. *Nonparametric kernel density estimation and its computational aspects*. Springer, 2018.

[58] James M Greenberg and Wolfgang Alt. “Stability results for a diffusion equation with functional drift approximating a chemotaxis model”. In: *Transactions of the American Mathematical Society* 300.1 (1987), pp. 235–258.

[59] Joshua M Greenberg and Alain-Yves LeRoux. “A well-balanced scheme for the numerical processing of source terms in hyperbolic equations”. In: *SIAM Journal on Numerical Analysis* 33.1 (1996), pp. 1–16.

[60] Francesca Romana Guarguaglini et al. “Stability of constant states and qualitative behavior of solutions to a one dimensional hyperbolic model of chemotaxis”. In: *Discrete & Continuous Dynamical Systems-B* 12.1 (2009), p. 39.

[61] Francesca Romana Guarguaglini et al. “Stability of constant states and qualitative behavior of solutions to a one dimensional hyperbolic model of chemotaxis”. In: *Discrete & Continuous Dynamical Systems-B* 12.1 (2009), p. 39.

[62] Bertil Gustafsson, Heinz-Otto Kreiss, and Arne Sundström. “Stability theory of difference approximations for mixed initial boundary value problems. II”. In: *Mathematics of Computation* (1972), pp. 649–686.

[63] Wolfgang Hackbusch. *Multi-grid methods and applications*. Vol. 4. Springer Science & Business Media, 2013.

[64] Jacques Hadamard. “Sur les problèmes aux dérivées partielles et leur signification physique”. In: *Princeton university bulletin* (1902), pp. 49–52.

[65] William W Hager and Hongchao Zhang. “A survey of nonlinear conjugate gradient methods”. In: *Pacific journal of Optimization* 2.1 (2006), pp. 35–58.

[66] Ernst Hairer, Syvert P Nørsett, and Gerhard Wanner. *Solving ordinary differential equations. 1, Nonstiff problems*. Springer-Vlg, 1993.

[67] Martin Hanke. “Limitations of the L-curve method in ill-posed problems”. In: *BIT Numerical Mathematics* 36.2 (1996), pp. 287–301.

[68] Per Christian Hansen. “The L-curve and its use in the numerical treatment of inverse problems”. In: (1999).

[69] Jjje Henson. “Multigrid methods nonlinear problems: an overview”. In: *Proceedings of SPIE*. Vol. 5016. 2003, pp. 36–48.

[70] Thomas Hillen and Kevin Painter. “Global existence for a parabolic chemotaxis model with prevention of overcrowding”. In: *Advances in Applied Mathematics* 26.4 (2001), pp. 280–301.

[71] Thomas Hillen and Kevin J Painter. “A user’s guide to PDE models for chemotaxis”. In: *Journal of mathematical biology* 58.1 (2009), pp. 183–217.

[72] Thomas Höfer, Jonathan A Sherratt, and Philip Kumar Maini. “Dictyostelium discoideum: cellular self-organization in an excitable biological medium”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 259.1356 (1995), pp. 249–257.

[73] Wolfgang Hörmann, Josef Leydold, and Gerhard Derflinger. *Automatic nonuniform random variate generation*. Springer, 2004.

[74] Reiner Horst and Panos M Pardalos. *Handbook of global optimization*. Vol. 2. Springer Science & Business Media, 2013.

[75] Shi Jin. “A steady-state capturing method for hyperbolic systems with geometrical source terms”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 35.4 (2001), pp. 631–645.

[76] Shi Jin. “Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations”. In: *SIAM Journal on Scientific Computing* 21.2 (1999), pp. 441–454.

[77] Shi Jin and Yingzhe Shi. “A micro-macro decomposition-based asymptotic-preserving scheme for the multispecies Boltzmann equation”. In: *SIAM Journal on Scientific Computing* 31.6 (2010), pp. 4580–4606.

[78] Daniel D Joseph and Luigi Preziosi. “Heat waves”. In: *Reviews of Modern Physics* 61.1 (1989), p. 41.

[79] Guillaume Jouvet and Carsten Gräser. “An adaptive Newton multigrid method for a model of marine ice sheets”. In: *Journal of Computational Physics* 252 (2013), pp. 419–437.

[80] Maka Karalashvili et al. “Identification of transport coefficient models in convection-diffusion equations”. In: *SIAM Journal on Scientific Computing* 33.1 (2011), pp. 303–327.

[81] Saeed Kazem and Mehdi Dehghan. “Semi-analytical solution for time-fractional diffusion equation based on finite difference method of lines (MOL)”. In: *Engineering with Computers* 35.1 (2019), pp. 229–241.

[82] Ora Kedem and Aharon Katchalsky. “Thermodynamic analysis of the permeability of biological membranes to non-electrolytes”. In: *Biochimica et biophysica Acta* 27 (1958), pp. 229–246.

[83] Evelyn F Keller and Lee A Segel. “Initiation of slime mold aggregation viewed as an instability”. In: *Journal of theoretical biology* 26.3 (1970), pp. 399–415.

[84] Aleksandr Fedorovich Klement’ev. “Stability of the numerical integration of systems of ordinary differential equations arising in the application of the straight line method to some equations of oscillation theory”. In: *USSR Computational Mathematics and Mathematical Physics* 9.1 (1969), pp. 295–299.

[85] Joost N Kok et al. *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007, Proceedings*. Vol. 4701. Springer, 2007.

[86] Jukka Matthias Krisp and Stefan Peters. “Directed kernel density estimation (DKDE) for time series visualization”. In: *Annals of GIS* 17.3 (2011), pp. 155–162.

[87] I Richard Lapidus and Ralph Schiller. “Model for the chemotactic response of a bacterial population”. In: *Biophysical journal* 16.7 (1976), pp. 779–789.

[88] Randall J LeVeque et al. *Finite volume methods for hyperbolic problems*. Vol. 31. Cambridge university press, 2002.

[89] Chialiang Lin et al. “Evolutionary derivation of sixth-order P-stable SDIRKN methods for the solution of PDEs with the method of lines”. In: *Mediterranean Journal of Mathematics* 16.3 (2019), pp. 1–17.

[90] Roberto Lipsi et al. “The clinical use of regenerative therapy in COPD”. In: *International journal of chronic obstructive pulmonary disease* 9 (2014), p. 1389.

[91] Tao Liu. “A nonlinear multigrid method for inverse problem in the multiphase porous media flow”. In: *Applied Mathematics and Computation* 320 (2018), pp. 271–281.

[92] Tao Liu and Jie Song. “Estimation of a permeability field within the two-phase porous media flow using nonlinear multigrid method”. In: *Mathematical Problems in Engineering* 2017 (2017).

[93] Wei Liu and Michael Röckner. *Stochastic partial differential equations: an introduction*. Springer, 2015.

[94] Lucie A Low et al. “Organs-on-chips: Into the next decade”. In: *Nature Reviews Drug Discovery* (2020), pp. 1–17.

[95] Shuai Lu and Jens Flemming. “Convergence rate analysis of Tikhonov regularization for nonlinear ill-posed problems with noisy operators”. In: *Inverse problems* 28.10 (2012), p. 104003.

[96] Michael G Mauk et al. “Lab-on-a-chip technologies for oral-based cancer screening and diagnostics: Capabilities, issues, and prospects”. In: *Annals of the New York Academy of Sciences* 1098.1 (2007), pp. 467–475.

[97] Tengku Ibrahim Maulana et al. “Immunocompetent Cancer-on-Chip models to assess immuno-oncology therapy”. In: *Advanced Drug Delivery Reviews* (2021).

[98] Roberto Mayor and Carlos Carmona-Fontaine. “Keeping in touch with contact inhibition of locomotion”. In: *Trends in cell biology* 20.6 (2010), pp. 319–328.

[99] Előd Méhes and Tamas Vicsek. “Collective motion of cells: from experiments to models”. In: *Integrative biology* 6.9 (2014), pp. 831–854.

[100] Andreas Meister. *Numerik linearer gleichungssysteme*. Vol. 4. Springer, 2011.

[101] Andreas Meister and Jens Struckmeier. *Hyperbolic partial differential equations: theory, numerics and applications*. Springer Science & Business Media, 2012.

[102] Keith W Morton and David Francis Mayers. *Numerical solution of partial differential equations: an introduction*. Cambridge university press, 2005.

[103] Alexander Mosig, Janna Nawroth, and Peter Loskill. “Organs-on-a-Chip: neue Perspektiven in der Medikamenten-entwicklung und Personalisierten Medizin”. In: *Deutsche Zeitschrift fŘr Klinische Forschung* 1 (2017), pp. 7–14.

[104] James D Murray. *Mathematical biology: I. An introduction*. Vol. 17. Springer Science & Business Media, 2007.

[105] JD Murray. *II. Spatial models and biomedical applications*. Springer, 2003.

[106] In Jae Myung. “Tutorial on maximum likelihood estimation”. In: *Journal of mathematical Psychology* 47.1 (2003), pp. 90–100.

[107] EA Nadaraya. “On non-parametric estimates of density functions and regression curves”. In: *Theory of Probability & Its Applications* 10.1 (1965), pp. 186–190.

[108] Roberto Natalini. “Convergence to equilibrium for the relaxation approximations of conservation laws”. In: *Communications on pure and applied mathematics* 49.8 (1996), pp. 795–823.

[109] Roberto Natalini and Magali Ribot. “Asymptotic high order mass-preserving schemes for a hyperbolic model of chemotaxis”. In: *SIAM Journal on Numerical Analysis* 50.2 (2012), pp. 883–905.

[110] TK Nilssen et al. “Identification of diffusion parameters in a nonlinear convection–diffusion equation using the augmented Lagrangian method”. In: *Computational Geosciences* 13.3 (2009), pp. 317–329.

[111] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[112] Seungseok Oh et al. “A general framework for nonlinear multigrid inversion”. In: *IEEE Transactions on image Processing* 14.1 (2004), pp. 125–140.

[113] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[114] Stanley Osher. “Stability of difference approximations of dissipative type for mixed initial-boundary value problems. I”. In: *Mathematics of computation* 23.106 (1969), pp. 335–340.

[115] Stanley Osher. “Systems of difference equations with general homogeneous boundary conditions”. In: *Transactions of the American Mathematical Society* 137 (1969), pp. 177–201.

[116] Hans G Othmer and Thomas Hillen. “The diffusion limit of transport equations II: Chemotaxis equations”. In: *SIAM Journal on Applied Mathematics* 62.4 (2002), pp. 1222–1250.

[117] Benoît Perthame. *Transport equations in biology*. Springer Science & Business Media, 2006.

[118] Luigi Preziosi and Andrea Tosin. "Multiphase and multiscale trends in cancer modelling". In: *Mathematical Modelling of Natural Phenomena* 4.3 (2009), pp. 1–11.

[119] Alfio Quarteroni, Alessandro Veneziani, and Paolo Zunino. "Mathematical and numerical modeling of solute dynamics in blood flow and arterial walls". In: *SIAM Journal on Numerical Analysis* 39.5 (2002), pp. 1488–1511.

[120] PL Roe. "Upwind differencing schemes for hyperbolic conservation laws with source terms". In: *Nonlinear hyperbolic problems*. Springer, 1987, pp. 41–51.

[121] Carl Runge. "Über die numerische Auflösung von Differentialgleichungen". In: *Mathematische Annalen* 46.2 (1895), pp. 167–178.

[122] ETF Santos and Amin Bassrei. "L-and Θ-curve approaches for the selection of regularization parameter in geophysical diffraction tomography". In: *Computers & Geosciences* 33.5 (2007), pp. 618–629.

[123] William E Schiesser. *The numerical method of lines: integration of partial differential equations*. Elsevier, 2012.

[124] Gideon Schwarz et al. "Estimating the dimension of a model". In: *Annals of statistics* 6.2 (1978), pp. 461–464.

[125] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[126] Lee A Segel. "A theoretical study of receptor mechanisms in bacterial chemotaxis". In: *SIAM Journal on Applied Mathematics* 32.3 (1977), pp. 653–665.

[127] A Serafini. "Mathematical models for intracellular transport phenomena". In: *PhD in Applied Mathematics, Universita degli Studi di Roma" La Sapienza", Dottorato di Ricerca in Modelli e Metodi Matematici per la tecnologia e la societa* (2007).

[128] Simon J Sheather and Michael C Jones. "A reliable data-based bandwidth selection method for kernel density estimation". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 53.3 (1991), pp. 683–690.

[129] Jonathan A Sherratt. "Chemotaxis and chemokinesis in eukaryotic cells: the Keller-Segel equations as an approximation to a detailed model". In: *Bulletin of mathematical biology* 56.1 (1994), pp. 129–146.

[130] Wang Shuai, Xu Baiyan, and Liu Tao. “A Conjugate Gradient Method for Inverse Problems of Non-linear Coupled Diffusion Equations”. In: *Journal of Physics: Conference Series*. Vol. 1634. 1. IOP Publishing. 2020, p. 012165.

[131] BW Silverman. *Density estimation for statistics and data analysis, chapter 2-3*. 1986.

[132] Peter Spellucci. *Numerische verfahren der nichtlinearen optimierung*. Vol. 320. Springer-Verlag, 2013.

[133] J Michael Steele. *Non-Uniform Random Variate Generation (Luc Devroye)*. 1987.

[134] John C Strikwerda. *Finite difference schemes and partial differential equations*. SIAM, 2004.

[135] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.

[136] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods: nonlinear programming*. Vol. 1. Springer Science & Business Media, 2006.

[137] Jean-Yves Tinevez et al. “TrackMate: An open and extensible platform for single-particle tracking”. In: *Methods* 115 (2017), pp. 80–90.

[138] Ulrich Trottenberg, Cornelius W Oosterlee, and Anton Schuller. *Multigrid*. Elsevier, 2000.

[139] Laurie S Tsuruda, Michael W Lamé, and A Daniel Jones. “Formation of epoxide and quinone protein adducts in B6C3F 1 mice treated with naphthalene, sulfate conjugate of 1, 4-dihydroxynaphthalene and 1, 4-naphthoquinone”. In: *Archives of toxicology* 69.6 (1995), pp. 362–367.

[140] Luke Tweedy et al. “Distinct cell shapes determine accurate chemotaxis”. In: *Scientific reports* 3.1 (2013), pp. 1–7.

[141] Erika Vacchelli et al. “Chemotherapy-induced antitumor immunity requires formyl peptide receptor 1”. In: *Science* 350.6263 (2015), pp. 972–978.

[142] Curtis R Vogel. “Non-convergence of the L-curve regularization parameter selection method”. In: *Inverse problems* 12.4 (1996), p. 535.

[143] Xing Wang. “Time Dependent Kernel Density Estimation: A New Parameter Estimation Algorithm, Applications in Time Series Classification and Clustering”. In: (2016).

[144] Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II*. Vol. 375. Springer Berlin Heidelberg, 1996.

[145] David J Warne, Ruth E Baker, and Matthew J Simpson. “Using experimental data and information criteria to guide model selection for reaction–diffusion problems in mathematical biology”. In: *Bulletin of Mathematical Biology* 81.6 (2019), pp. 1760–1804.

[146] Sumio Watanabe. “A widely applicable Bayesian information criterion”. In: *Journal of Machine Learning Research* 14.Mar (2013), pp. 867–897.

[147] Stanisław Węglarczyk. “Kernel density estimation and its application”. In: *ITM Web of Conferences*. Vol. 23. EDP Sciences. 2018.

[148] Donald Wlodkowic and Jonathan M Cooper. “Tumors on chips: oncology meets microfluidics”. In: *Current opinion in chemical biology* 14.5 (2010), pp. 556–567.

[149] Jeffrey Wyckoff et al. “A paracrine loop between tumor cells and macrophages is required for tumor cell migration in mammary tumors”. In: *Cancer research* 64.19 (2004), pp. 7022–7029.

[150] Xin-She Yang. *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[151] Yu-Hua Zeng, Zheng Peng, and Yu-Fei Yang. “A hybrid splitting method for smoothing Tikhonov regularization problem”. In: *Journal of Inequalities and Applications* 2016.1 (2016), pp. 1–13.

[152] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2 (2005), pp. 301–320.