May 2017

Modeling of Anticancer Drug Delivery By Temperature-Sensitive Liposomes

Vera Franziska Loeser
University of Wisconsin-Milwaukee

Modeling of Anticancer Drug Delivery by Temperature-Sensitive Liposomes

by

Vera Franziska Loeser

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Mathematics

at

The University of Wisconsin–Milwaukee

May 2017

Abstract

Modeling of Anticancer Drug Delivery by Temperature-Sensitive Liposomes

by

Vera Franziska Loeser

The University of Wisconsin–Milwaukee, 2017
Under the Supervision of Professor Peter Hinow

Cytotoxic anticancer drugs are used to treat cancer, particularly tumors. These drugs themselves do not distinguish between healthy and tumor cells and attack all of them. Consequently physicians and chemists investigate safer ways of delivery that minimize damage to healthy cells. One of these ways are liposomal formulations of the anticancer drugs. Liposomes are vesicles that encapsulate the drug to shield the healthy parts of the body from the toxicity of the drugs. Due to the abnormal structure of tumors, especially their leaky vasculature, these macromolecules are able to diffuse into the tumor tissue whereas the normal vasculature prevents them to move into the tissue. After reaching the diseased tissue, the liposomes release their cargo triggered by a physical or chemical process.

In this thesis, we will investigate those liposomes that are activated upon entering regions of increased tissue temperature. Developing temperature-sensitive liposomes for targeted anticancer drug delivery is difficult to realize. The liposomes must be able to move far into the tumor against the interstitial pressure. The release of the drug must be triggered precisely in the tumor. We will show that the drug can move further into the tissue by diffusion. The aim of the mathematical modeling is to improve our understanding of the release and transport processes involved and (in the future) to optimize the scheduling between tissue heating and the liposome administration.

To my family.

Thank you for all your support.

I love you.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

BDF1	first backward differentiation formula
BDF2	second backward differentiation formula
CT	computer tomography
DNA	Deoxyribonucleic acid
DOX	Doxorubicin
EPR	enhanced permeability and retention
IFP	interstitial fluid pressure
LTLD	low temperature-sensitive liposomal formulation of Doxorubicin
ODE	ordinary differential equation
PDE	partial differential equation
RFA	radio frequency tumor ablation

Acknowledgements

First, I would like to thank **Professor Peter Hinow** for advising me on this project. Not knowing that medicine and more particularly oncology would have been my study field of choice, he offered me literally the perfect topic for this thesis. Without his support by providing the important literature and knowledge of the physics behind the processes I would not have been able to write this thesis. Thank you for your patience in our biweekly and weekly meetings.

I would also like to thank **Professor Bruce Wade** and **Professor Gabriella Pinter** for serving on my committee and taking time for me when I needed help.

Without the lectures of **Professor Martin Reißel** and **Professor Jonathan Kahl** this thesis would have been much harder. Thank you for building up a well-founded background knowledge about numerical analysis in general and advection equations in particular. I reviewed my lecture notes countless times.

Thanks to all my proofreaders who took time to help me finish this thesis.

Special thanks to **Professor Gerhard Dikta** and **Professor Jugal Ghorai** for bringing this Dual Master study into being. I also would like to thank **Ines**, **Nele**, **Rica** and **Thomas** for being the guinea pigs together with me for the exchange.

For sure I have to thank **Andi** and **Thomas** for their patience with me while we studied together for exams. You are great teachers and encouraged me to study when I tended to give up.

I would also like to thank my family and my friends for their love and support within the last two years. Especially **Thorsten**, **Jiayan**, **Qian** and **Jenny**, who are great roommates and indulged my every whim patiently.

Chapter 1

Introduction

The field of modeling and simulating tumor growth and tumor treatment is sometimes called "Mathematical Oncology" and is a growing field combining mathematical sciences and biomedicine. It provides tools for quantification of key parameters from patient-specific clinical data as well as customization of cancer treatment using mathematical models (Kuang et al., 2016). For a better understanding of mathematical oncology a brief definition of cancer follows: Tumors or neoplasia can be seen as a kind of loss of tissue stability due to mutation, where the proliferation of cells is "uncontrolled". Cancer is a malignant tumor, so that the tumor has the ability to invade the surrounded tissue and to build metastases. Metastasis describes the ability of a tumor to expand to other parts of the body (Kuang et al., 2016).

Mathematical modeling in general is a great tool for research and industry. The two major advantages of modeling are reduced costs and more precise results than those obtained by physical experiments. Therefore, mathematical modeling complements experimental work. Particular actions within a complex process can be simulated individually for a better understanding. In cancer research mathematical modeling of the medical treatment of tumor cells is a step to reduce the number of experiments in the petri dish or with animals. The drug and the tumor cells need to be developed first and then implanted into an animal. This procedure is time consuming and costly and does not guarantee that the setup of the experiments will succeed. Besides the economical point of view, also ethical issues arise.

On the other hand obtaining results from physical experiments is difficult and expensive as well, since methods like CT scans and coloring with radioactive elements are necessary. Some of these procedures are quite sensitive to errors.

Using mathematical modeling, the equations need to be found and parametrized and the

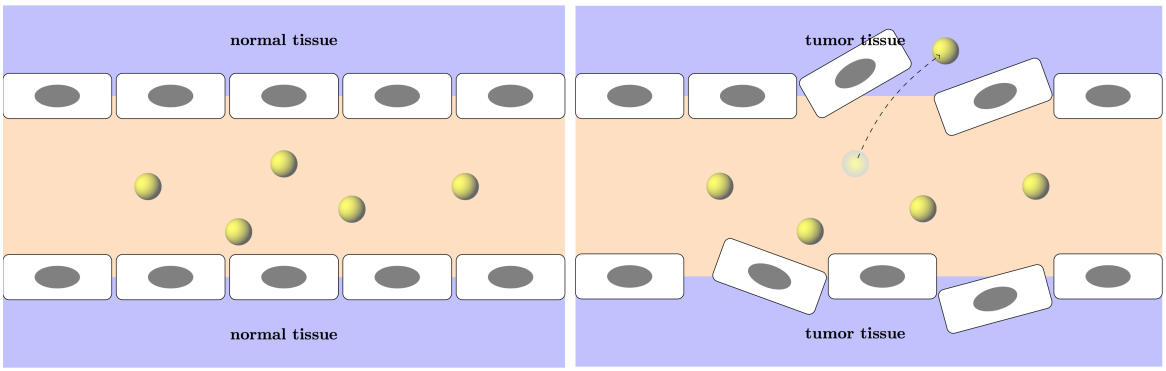
model needs to be implemented using programming languages. This can be time consuming and some sophisticated simulations need a lot of computational resources like high performance computing cluster. As soon as the simulations are running, multiple tests can be done without wasting material and the calculations are usually faster than physical experiments. The solutions of the simulations are obtained immediately without post-processing.

1.1 The Enhanced Permeability and Retention Effect

The enhanced permeability and retention (EPR) effect describes how molecules of certain sizes tend to accumulate preferentially in tumor tissue than in normal tissue. Hiroshi Maeda published as one of the first researchers new findings about this effect (Matsumura and Maeda, 1986; Maeda et al., 2003; Maeda, 2012). Various factors reinforce this effect and will be described in this chapter.

The EPR effect is a result of the rapid tumor growth and suppressed lymphatic drainage (Maeda et al., 2003). The tumor requires large amounts of oxygen and nutrients for growth. This results in the need to recruit the body's vasculature to the tumor. Tumor vasculature is different from healthy vasculature in that the endothelial cells have wide fenestrations (Taurin et al., 2012). The fenestrations result from accelerated growth since the cells cannot grow and align appropriately. Sustenance is faster through the bigger pores between the endothelial cells than in healthy tissue because more particles can diffuse into the tumor tissue at once. On the other hand, the lymphatic system within the tumor is insufficient since it is not built up appropriately. Thus, the uptake of the tumor tissue will not wash out as fast as in healthy tissue (Miao et al., 2015). The retention of micromolecular drugs inside normal tissue is usually a few minutes. In contrast, the retention time in malignant tissue can last days to weeks (Maeda, 2012).

Targeting drugs in general exploit the fact that a drug has a site of intended action and should be brought to ill tissue without having a side effect elsewhere. Therefore, a release of



(a) Normal vasculature (b) Leaky vasculature

Figure 1.1: Comparison of the vasculature in normal and in tumor tissue.

the drug in the bloodstream within the tumor would be targeted drug delivery as well. This also applies to other diseases. However, the EPR effect supports the usage of targeting drugs by facilitating the import of macromolecular drugs into the tumor tissue, which is unlikely in normal tissue (Maeda et al., 2003; Maeda, 2012; Wong et al., 2015). Since macromolecules are not able to diffuse passively into normal tissue, the EPR effect is exploited for deliberate treatment of tumor cells. Therefore, the likelihood for healthy cells to be killed by the cytotoxic drug decreases, while it is more likely that the drug treats the tumor instead of passing the tumor.

Disadvantages of the EPR effect are a higher interstitial fluid pressure (IFP) inside the tumor and thus enhanced diffusion of molecules out of the tumor (Wong et al., 2015; Minchinton and Tannock, 2006). Because a lot of nutrients and fluid can move into the tumor and remains inside, the interstitial pressure within the tumor is much higher than the vascular pressure outside of the tumor. The interstitial pressure gradient creates a outward directed velocity field within the tumor that drives the uptake from the core of the tumor back towards its edge (Baxter and Jain, 1989). As a result the EPR effect allows transport both into and out of the tumor.

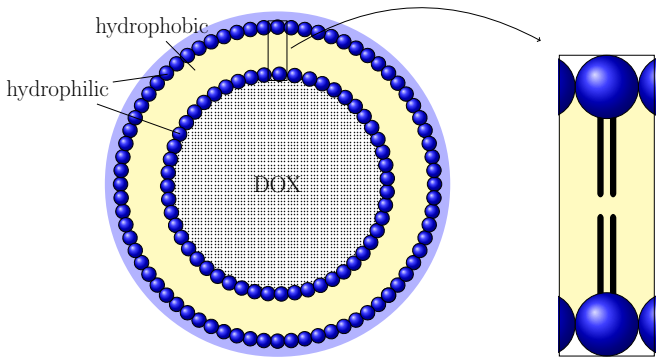


Figure 1.2: Structure of a liposome

1.2 Targeted Drug Delivery in solid tumors

Macromolecules with a molecular weight $> 45\ kDa$ will result in a selective uptake mainly within the tumor since the fenestrations of normal tissue are too small (Maeda et al., 2003). In normal tissue the endothelial cells are aligned close to each other, so that mainly small molecules diffuse through the fenestrations. There are also transporters in the endothelial cells for active transport that can even work against a gradient (Saxena et al., 1991), but in this thesis we focus on the passive transport by diffusion through the fenestrations. The macromolecules will mainly stay within the vasculature without passing the barrier into the tissue. The movement of the macromolecules through the fenestrations into the tumor tissue is depicted in Fig. 1.1.

The encapsulation of the cytotoxic anticancer drugs within liposomes primarily shields the drug but as a side effect it increases its size. One representative of these anticancer drugs is Doxorubicin (DOX), whose interference with DNA leads to cell death. Their adverse side effects include cardiomyopathy which can lead to heart failure and suppression of white blood cell production in the bone marrow (Hinow et al., 2016). These side effects can be reduced by targeted drug delivery.The liposomal formulation of DOX became increasingly common, for example as Doxil® or ThermoDox®. The structure of these liposomes is depicted in Fig. 1.2. They are spherical vesicles having the drug surrounded by at least one lipid bilayer. This layer is again surrounded by a coat which protects the liposome against destruction by

the immune system. The size as well as the shape of the liposomes affect the biodistribution and uptake, in addition to circulation time of molecules (Arnida et al., 2011). Therefore, targeted drugs are designed as bigger liposomes carrying the actual drug into the tumor cell, where they release their cargo.

The release is triggered by a chemical or physical reaction such as heat (Gasselhuber et al., 2010; Kneidl et al., 2014), ultrasound (Staruch et al., 2011; Rizzitelli et al., 2015) or the pH difference between healthy tissue and the more acidic malignant tissue (Liu et al., 2014). For example, the low temperature-sensitive liposomal formulation of Doxorubicin (LTLD) releases its cargo in a high concentration when it is heated to temperatures between 39.5 °C and 42 °C (Zagar et al., 2014). The tissue can be heated via radio frequency tumor ablation (RFA) (Gasselhuber et al., 2010) or ultrasound mediated hyperthermia (Gasselhuber et al., 2012a).

Our goal is to create an easily expandable model that can handle several targeted drug delivery methods. Therefore, we base on the work of Stapleton et al. (2013) and extend their model with the drug release using an adjustable "transfer function". They proposed a model for the liposome transport in solid tumors based on biophysical transport equations. These equations describe the pressure driven fluid flow across blood vessels and through the tumor interstitium. Their parametrization results from three preclinical animal tumor models. Within this thesis, the release of the drug in our mathematical model is simulated using temperature-sensitive liposomes based on the results of Gasselhuber et al. (2010). They proposed a spatio-temporal multicompartmental pharmacokinetic model to describe the transport and release of the drug from the low temperature-sensitive liposomal formulation of Doxorubicin into the tumor plasma space of a human. They parametrized their model based on data for human lung cancer cells.

1.3 Structure of this Thesis

In this thesis we study the anticancer drug delivery using temperature-sensitive liposomes. The relevant properties of tumors and anticancer drugs were explained in detail in Ch. 1 to create a solid background knowledge and to aid in the understanding of the mathematical model. In Ch. 2 we derive a mathematical model to improve the understanding of the ongoing processing during targeted drug delivery. In Ch. 3 we touch upon the numerical methods and the results of the numerical simulations will follow in Ch. 4. In Ch. 5 we will summarize the work in a discussion and prospects will be given.

Chapter 2

The Mathematical Model

We consider an isolated model tumor with spherical symmetry that is homogeneously perfused by blood vessels and lacks lymphatic drainage (Baxter and Jain, 1989; Stapleton et al., 2013). So the microscopic elements as blood vessels, interstitial tumor matrix and individual cells are not modeled explicitly. After a suitable normalization by the tumor radius R, the spatial variable is in the interval $r \in [0, 1]$.

A schematic depiction of the model of drug release of temperature-sensitive liposomes is given in Fig. 2.1. Liposomes are delivered from the blood plasma compartment. We assume that the liposome injection rapidly equilibrates in the plasma, so an ordinary differential equation is sufficient to describe their concentration within the blood plasma. Their concentration $c(t)$ follows simple first order kinetics

$$\frac{dc}{dt} = -kc, \quad c(0) = c_0. \tag{2.1}$$

Here k denotes the rate of clearance from the plasma (Ishida et al., 2002) and results from the assumption that the liposomes have a half-life of 24 hours. c_0 is the initial concentration. As long as the vascular pressure p_v is higher than the interstitial pressure p_i the liposomes are able to move from the blood plasma into the tumor tissue. Since the tumor volume is sufficiently small in comparison to the blood volume the passing of the liposomes into the tumor does not impact c significantly. As long as the temperature surrounding the liposomes is sufficiently high the liposomes can release their cytotoxic cargo. After the liposomes release their cargo they are no longer considered. The concentration of the drug inside the tumor is denoted as $w(r, t)$.

Baxter and Jain (1989) derived an analytic steady-state solution for the IFP for an

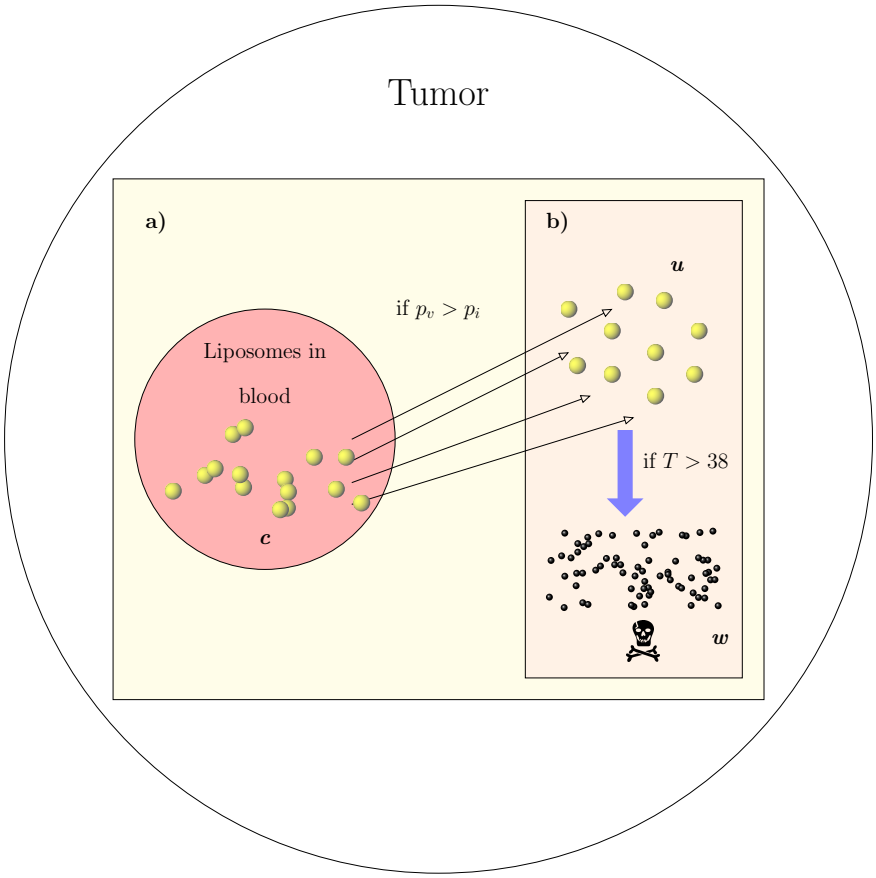


Figure 2.1: Scheme of temperature-sensitive liposomes. **a**: Transport of temperature-sensitive liposomes across the blood vessel barrier as long as the vascular pressure is higher than the interstitial pressure. **b**: Release of the liposome's cargo as long as the temperature within the tumor is sufficiently high.

isolated tumor without extravascular binding. Based on their aforementioned assumptions the interstitial fluid pressure is given by

$$p_i(r) = p_{i,max}\left(1 - \frac{\sinh(\alpha r)}{r\sinh(\alpha)}\right), \tag{2.2}$$

where $p_{i,max}$ is the maximal IFP. Stapleton et al. (2013) report values between 3.5 - 5.7 $mmHg$, 1.7 - 4.7 $mmHg$ and 4.56 - 42.3 $mmHg$ from experiments of ME180, H520 and VX2 tumors, respectively. These experiments were performed in mice (ME180, H520) and rabbits (VX2). α is a dimensionless parameter that is defined as the ratio of vascular to interstitial permeability. A smaller value of α means that the interstitial permeability is relatively high, whereas a larger value for α corresponds to a relatively low interstitial

permeability. Therefore, a smaller value of α means that macromolecules can move more easily into the tumor tissue.

The interstitial fluid velocity is obtained from Darcy's law,

$$v(r) = -K\frac{\partial p_i}{\partial r} = \frac{Kp_{i,max}}{R}\frac{\alpha r\cosh(\alpha r) - \sinh(\alpha r)}{r^2\sinh(\alpha)}, \tag{2.3}$$

where K denotes the hydraulic conductivity. This physical constant describes the permeability of the interstitium. This velocity field drives the transport of liposomes in the interstitial region of the tumor. Therefore $v(R) > 0$ on the edge of the tumor.

We assume a unidirectional flow of the liposomes such that the rate of transvascular exchange is proportional to the difference between vascular pressure p_v and p_i. Since the IFP is increasing towards the center of the tumor whereas p_v is assumed constant throughout the tissue, the exchange is only possible as long as the interstitial pressure is smaller than the vascular pressure. The concentration of the liposomes $u(r,t) \in (0,1)\times(0,t_e)$, with t_e as the end of the time interval, is given by

$$\frac{\partial u}{\partial t} = \frac{1}{V_r}cf_c(1-\sigma)(p_v - p_i)\cdot H(p_v - p_i) - \theta_l\frac{\partial}{\partial r}(uv) - h(T(r,t))u \quad \text{in} \quad (0,1)\times(0,t_e). \tag{2.4}$$

The initial condition and Neumann boundary condition, respectively, are given by

$$u(r,0) = u_0(r) \quad \text{for} \quad r \in \Omega := (0,1), \tag{2.5}$$

$$\frac{\partial u}{\partial r} = 0 \quad \text{on} \quad \Gamma := \partial\Omega\times(0,t_e). \tag{2.6}$$

If the IFP is higher than p_v, no transvascular exchange can occur. Therefore, the positivity is modeled by a Heaviside step function H. The only source of liposomes in the tumor is the plasma perfusing the tumor. Here $T(r,t)$ denotes the tissue temperature (which is also assumed to be spherically symmetric), and h is called a "transfer function", similar to that in Hinow et al. (2016). V_r is the ratio of the tumor volume to the blood volume. It ensures conservation of mass as the liposomes move between compartments of different volumes. The blood volume is 5 L whereas the tumor volume is 1 cm^3 and therefore significantly smaller.

V_r is given by

$$V_r = \frac{3 \cdot 5}{4\pi \cdot 10^3} \approx 10^{-3}. \tag{2.7}$$

The constant f_c denotes the capillary filtration coefficient, which is a constant of proportionality between capillary surface area and capillary hydraulic conductivity, and describes the permeability of the capillaries. The constant σ denotes the filtration reflection coefficient, which describes the fact that molecules bigger than a specific threshold are retained by the semipermeable membrane. Although the permeability is enhanced within tumors not every molecule will diffuse into the tumor tissue.

Stapleton et al. (2013) describe the fractional rate of liposome transport through interstitium to fluid flow as $\theta_l = 0.5$. This results from the inertia of the relatively large liposomes within the fluid. As the liposomes are rather large particles of molecular weights in the range of $100\,MDa$ (Stapleton et al., 2013), they are not subject to diffusion. This is different for the drug that is released from them. The concentration of the drug $w(r,t) \in (0,1) \times (0,t_e)$ within the tumor is governed

$$\frac{\partial w}{\partial t} = \frac{\partial^2}{\partial r^2}(Dw) - \theta_d \frac{\partial}{\partial r}(wv) + \beta h(T(r,t))u. \tag{2.8}$$

The initial condition and Neumann boundary condition, respectively, are given by

$$w(r,0) = w_0(r) \quad \text{for} \quad r \in \Omega, \tag{2.9}$$

$$\frac{\partial w}{\partial r} = 0 \quad \text{on} \quad \Gamma := \partial\Omega \times (0,t_e). \tag{2.10}$$

The drug is subject to transport and diffusion, and is supplied by the liposomes as they enter the heated region. The diffusion coefficient D, as found in Fick's second law of diffusion, describes the proportionality between the molar flux due to molecular diffusion and the gradient in the concentration. Since the tissue is heated up the diffusivity depends on the status of the tissue and therefore on the temperature: If the tissue is already ablated, the diffusivity decreases significantly within the necrotic tissue (Gasselhuber et al., 2010). For simplicity, we still assume the D to be constant. Suggestions for improvement can be found

in the discussion in Ch. 5.

The fractional rate of the transport of drug molecules through interstitium due to fluid flow is described by $\theta_d = 1$. This is an initial guess and needs to be parametrized more precisely. The constant $\beta > 0$ is the yield of drug molecules supplied by one liposome.

At the moment, Eq. 2.8 models only the release of the drug and its motion through the tumor. In the future, the metabolism and uptake of the drug need to be included. Ch. 5 will give more details on the improvement of the model.

The total amount of liposomes within the tumor is givenby the integral

$$\bar{u}(t) = \int_V u(r,t)dV = \int_0^R u(r,t) \cdot 4\pi r^2 dr. \tag{2.11}$$

The parameters and their values used within the simulations are summed up in Table 2.1.

Parameter	Unit	Interpretation	Value(s)	Reference
$p_{i,max}$	$mmHg$	maximal value of IFP	19.8	(Stapleton et al., 2013)
p_v	$mmHg$	vascular pressure	10	-
R	cm	radius of tumor	1	-
α	-	ratio of vascular to interstitial permeability	1; 3; 5; 10; 25	(Stapleton et al., 2013)
k	s^{-1}	rate of liposome clearance from the plasma	8.3×10^{-6}	(Ishida et al., 2002)
c_0	$pmol \cdot cm^{-3}$	Initial plasma concentration of liposomes	0.5	(Hinow et al., 2016)
K	$cm^2 \cdot mmHg^{-1} \cdot s^{-1}$	hydraulic conductivity or interstitial permeability	662×10^{-7}	(Stapleton et al., 2013)
θ_l	-	fractional rate of liposome transport through interstitium to fluid flow	0.5	(Stapleton et al., 2013)
σ	-	filtration reflection coefficient	0.19	(Stapleton et al., 2013)
f_c	$mmHg^{-1} \cdot s^{-1}$	capillary filtration coefficient	1088×10^{-7}	(Stapleton et al., 2013)
V_r	-	relation of tissue to blood volume	10^{-3}	-
D	$cm^2 \cdot s^{-1}$	Diffusion coefficient of DOX	6.7×10^{-7} (non ablated)	(Gasselhuber et al., 2010)
θ_d	-	fractional rate of DOX transport through interstitium to fluid flow	1	-
β	-	drug loading coefficient of DOX	1.3×10^{-7}	(Hinow et al., 2016)

Table 2.1: Summary of all parameters for numerical approximation

Chapter 3

Numerical Methods

Eq. (2.4) is an example of an Advection-Reaction Equation (see Chapter 3.1) and Eq. (2.8) is an example of an Advection-Reaction-Diffusion Equation (see Chapter 3.2). In this thesis, we use the Strang Splitting method for the numerical approximation of these equations, which is a representative of an Operator Splitting method (Strang, 1968; Hundsdorfer, 2000). In these methods the equations are broken up in their physical components and then these components are solved sequentially. This method can be applied when one component is stiffer than the other or to separate several dimensions. Stiffness describes equations that cannot be solved with an explicit method since at least one component of the equation decays much faster than the other components.

This chapter analyses the numerical methods used for each equation generated by the Operator Splitting Method (see Appendix A) .

3.1 The Advection-Reaction Equation

Eq. (2.4) is an example of an Advection-Reaction equation. Generally speaking the reaction term in this equation creates sources or sinks of the quantity, whereas the advection term describes the transport of this quantity by bulk motion.

Applying the Strang splitting described in Appendix A, we can write

$$\frac{\partial u}{\partial t} = \underbrace{\frac{1}{V_r} c f_c (1-\sigma)(p_v - p_i) \cdot H(p_v - p_i) - h(T)u}_{\text{reaction term}} \underbrace{- \theta_l \frac{\partial}{\partial r}(uv)}_{\text{advection term}}$$

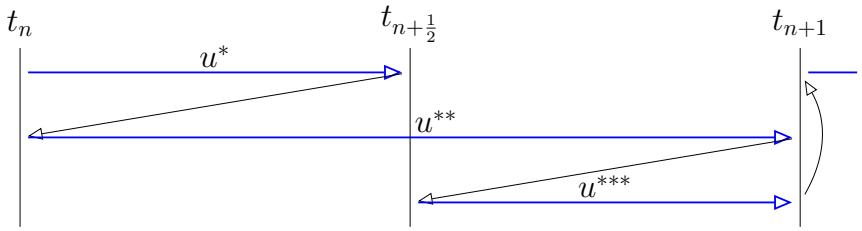


Figure 3.1: Scheme for the Strang Splitting of u

as

$$\frac{\partial u^*}{\partial \tau} = \frac{1}{V_r} c f_c (1-\sigma)(p_v - p_i) \cdot H(p_v - p_i) - h(T) u^* \qquad \text{for } t_n \leq \tau \leq t_{n+\frac{1}{2}} \tag{3.1}$$

$$\frac{\partial u^{**}}{\partial \tau} = -\theta_l \frac{\partial}{\partial r} (u^{**} v) \qquad \text{for } t_n \leq \tau \leq t_{n+1} \tag{3.2}$$

$$\frac{\partial u^{***}}{\partial \tau} = \frac{1}{V_r} c f_c (1-\sigma)(p_v - p_i) \cdot H(p_v - p_i) - h(T) u^{***} \qquad \text{for } t_{n+\frac{1}{2}} \leq \tau \leq t_{n+1} \tag{3.3}$$

with

$$u^*(t_n) = u_n, \qquad u^{**}(t_n) = u^*(t_{n+\frac{1}{2}}), \qquad u^{***}(t_{n+\frac{1}{2}}) = u^{**}(t_{n+1}).$$

This results in $u(t_{n+1}) = u^{***}(t_{n+1})$. Here the equation was split in a physical manner. First the reaction term will be evaluated and in the next step the transport will be calculated. The scheme is depicted in Fig. 3.1.

Eq. 3.1 and 3.3 are solved by the first representative of an L-stable method (definitions on stability can be found in Appendix B), the implicit Euler method

$$u_{\tau+1} = u_\tau + \Delta\tau g(r, \tau + \Delta\tau, u_{\tau+1}) \tag{3.4}$$

$$g(r, \tau + \Delta\tau, u_{\tau+1}) = \frac{1}{V_r} c(\tau + \Delta\tau) f_c (1-\sigma)(p_v - p_i) \cdot H(p_v - p_i) - h(T) u_{\tau+1}. \tag{3.5}$$

The implicit Euler method was implemented by using a fixpoint iteration with an initial step using the explicit Euler method. With this method no system of equation needs to be solved. The implicit Euler method has order $\mathcal{O}(\Delta t)$. The term $\mathcal{O}(\Delta t)$ denotes a quantity whose size is proportional to Δt or even smaller.

Eq. 3.2 was solved by using the first order upwind differencing scheme, which takes the flow direction into account. This discretization method creates the following equation which needs to be implemented:

$$u_r^{\tau+1} = \left(1 - \frac{\theta_l \Delta t}{\Delta r} v_r\right) u_r^{\tau} + \frac{\theta_l \Delta t}{\Delta r} v_{r-1} u_{r-1}^{\tau} \qquad \text{if } \theta_l \geq 0. \tag{3.6}$$

The advantages of this upwind differencing method are the properties of positivity and local preservation of mass and uniform stability. The method does not prevent a loss of mass due to outflow out of the domain. The disadvantage is artificial diffusion within the advection term, which could be eliminated by implementing a more complex method (Quarteroni et al., 2010, Remark 12.6). The artificial diffusion develops, because the mass within one grid cell moves only partly into the next grid cell. A method which attacks this issue will be discussed in Ch. 5.

The upwind differencing scheme is stable as long as the Courant-Friedrichs-Lewy condition is fulfilled. This condition describes the convergence behavior of x_i in its neighborhood. In this case the condition says that $x_i - \theta_l \Delta t \leq x_{i+1}$, otherwise the method is not convergent. Since $x_i = i\Delta r$, the condition for the upwind method is

$$0 \leq \frac{\theta_l \Delta t}{\Delta r} \leq 1 \qquad \text{for } \theta_l \geq 0 \tag{3.7}$$

3.2 The Advection-Reaction-Diffusion Equation

Eq. (2.8) is an example of an Advection-Reaction-Diffusion equation. In addition to the structure described in the previous Chapter 3.1 a diffusion term occurs in this type of equation. Applying the Strang splitting described in Appendix A, we can write

$$\frac{\partial w}{\partial t} = \underbrace{\beta h(T) u}_{\text{reaction term}} \underbrace{-\theta_d \frac{\partial}{\partial r}(wv)}_{\text{advection term}} + \underbrace{\frac{\partial^2}{\partial r^2}(Dw)}_{\text{diffusion term}}$$

as

$$\frac{\partial w^{(1)}}{\partial \tau} = \beta h(T) u_t \qquad \text{for } t_n \leq \tau \leq t_{n+\frac{1}{2}} \tag{3.8}$$

$$\frac{\partial w^{(2)}}{\partial \tau} = -\theta_d \frac{\partial}{\partial r}\left(w^{(2)} v\right) \qquad \text{for } t_n \leq \tau \leq t_{n+\frac{1}{2}} \tag{3.9}$$

$$\frac{\partial w^{(3)}}{\partial \tau} = \frac{\partial^2}{\partial r^2}(D w^{(3)}) \qquad \text{for } t_n \leq \tau \leq t_{n+1} \tag{3.10}$$

$$\frac{\partial w^{(4)}}{\partial \tau} = -\theta_d \frac{\partial}{\partial r}\left(w^{(4)} v\right) \qquad \text{for } t_{n+\frac{1}{2}} \leq \tau \leq t_{n+1} \tag{3.11}$$

$$\frac{\partial w^{(5)}}{\partial \tau} = \beta h(T) u \qquad \text{for } t_{n+\frac{1}{2}} \leq \tau \leq t_{n+1} \tag{3.12}$$

with

$$w^{(1)}(t_n) = w_n, \qquad w^{(2)}(t_n) = w^{(1)}(t_{n+\frac{1}{2}}), \qquad w^{(3)}(t_n) = w^{(2)}(t_{n+\frac{1}{2}}),$$
$$w^{(4)}(t_{n+\frac{1}{2}}) = w^{(3)}(t_{n+1}), \qquad w^{(5)}(t_{n+\frac{1}{2}}) = u^{(4)}(t_{n+1}).$$

The solution of the reaction equation for the drug can be found analytically. Since the right hand side does not depend on w, the equation is linear and can therefore just be multiplied by $\Delta\tau$ to evaluate the reaction. The reaction equation can also be calculated by the implicit Euler method as described in Ch. 3.1. This becomes important if the reaction equation would be extended by other terms like the uptake of the drug or metabolism. Then the reaction term would also depend on w itself and may not be solved by simple integration. For the advection term the upwind method is used as described in Ch. 3.1. The diffusion term is approximated by the Crank-Nicolson method (Grossmann et al., 2007, Section 2.6.1). This method is a combination of a half implicit and a half explicit Euler step and can be derived as follows:

$$\begin{aligned} \frac{\partial w}{\partial t} &= \frac{1}{2}\left(\left.\frac{\partial^2 D w}{\partial t^2}\right|_t + \left.\frac{\partial^2 D w}{\partial t^2}\right|_{t+1}\right) \\ &= \frac{1}{2}\left(\frac{D w^t_{r+1} - 2 D w^t_r + D w^t_{r-1}}{\Delta r^2} + \frac{D w^{t+1}_{r+1} - 2 D w^{t+1}_r + D w^{t+1}_{r-1}}{\Delta r^2}\right). \end{aligned} \tag{3.13}$$

Reordering with all w^{t+1} on the left hand side we get the final form of the Crank-Nicolson

method

$$-\kappa D w_{r+1}^{t+1} + (1+2\kappa D) w_r^{t+1} - \kappa D w_{r-1}^{t+1} = -\kappa D w_{r+1}^{t} + (1-2\kappa D) w_r^{t} + \kappa D w_{r-1}^{t}, \tag{3.14}$$

with

$$\kappa = \frac{\Delta t}{2\Delta r^2}. \tag{3.15}$$

This implicit method results in a tridiagonal matrix, which needs to be calculated in each time step. The advantage of this method is that the system of equations can be solved within $\mathcal{O}(r)$ work. This makes the Crank-Nicolson method basically as efficient as an explicit method.

With a series expansion in space and in time it can be shown that the Crank-Nicolson method is second order accurate in space and in time (Hundsdorfer, 2000). For stability the Courant-Friedrichs-Lewy condition is fulfilled if

$$\frac{\Delta t}{\Delta r^2} \leq \frac{1}{2}. \tag{3.16}$$

If this condition is violated, oscillations may occur.

Chapter 4

Results

The numerical methods were implemented using the Open Source computing language PYTHON (Python Software Foundation, 2017). The source code is attached in Appendix C.

4.1 Transfer function

The transfer function was determined from Gasselhuber et al. (2010, Figure 2). They experimentally determined the release of Doxorubicin by heating human blood plasma in single cuvettes to 25℃ and $37 - 47$℃ for at least 5 minutes. A liposome suspension (10 μL, 2 mg/mL DOX) of ThermoDox® (Celsion, Columbia, MD) was added to 750 μL of the heated plasma. The fluorescence intensity caused by the DOX release was monitored by a spectrofluorometer. A more precise description of the methods and materials can be found in Gasselhuber et al. (2010). We focused on the results depicted in Gasselhuber et al. (2010, Figure 2).

We are especially interested in the approximately linear slope within the first second (see Fig. 4.1). Therefore the release in Gasselhuber et al. (2010, Figure 2) is evaluated after one second with the slope taken from the gradient triangle. We consider only those temperatures which force the liposomes to release at least 50 % of their cargo, so the gradients were fitted to a piecewise linear function (see Fig. 4.1) dependent on the temperature.

$$h(T) = \begin{cases} 0 & T \leq 38 \\ 0.3789(T - 38) & 38 < T < 39 \\ 0.0176T - 0.3107 & 39 \leq T. \end{cases} \tag{4.1}$$

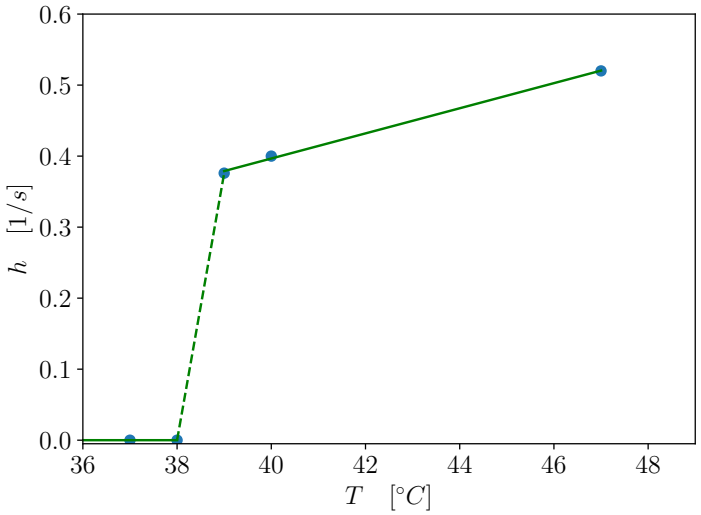


Figure 4.1: Piecewise linear fitting of the release of DOX. The data points base on Gasselhuber et al. (2010, Figure 2)

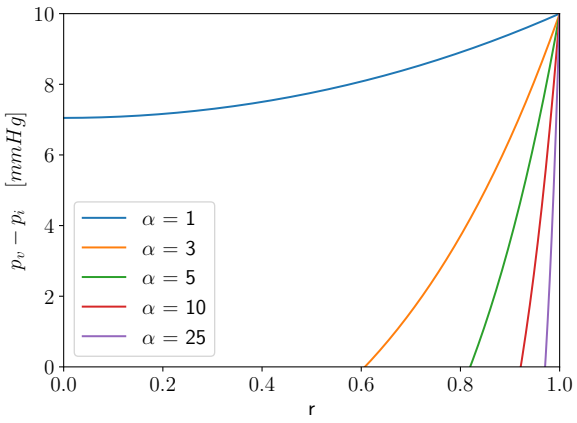


Figure 4.2: IFP and vascular pressure

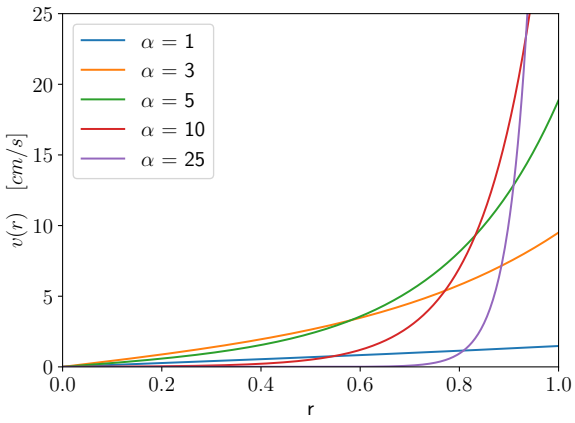


Figure 4.3: Interstitial Velocity

4.2 Pressure and Velocity

We assume a constant vascular pressure so that the transvascular exchange is driven by the IFP. This relation is depicted in Fig. 4.2. The liposomes cannot pass the point where the IFP outweighs the vascular pressure. However, the drug can diffuse further into the tumor.

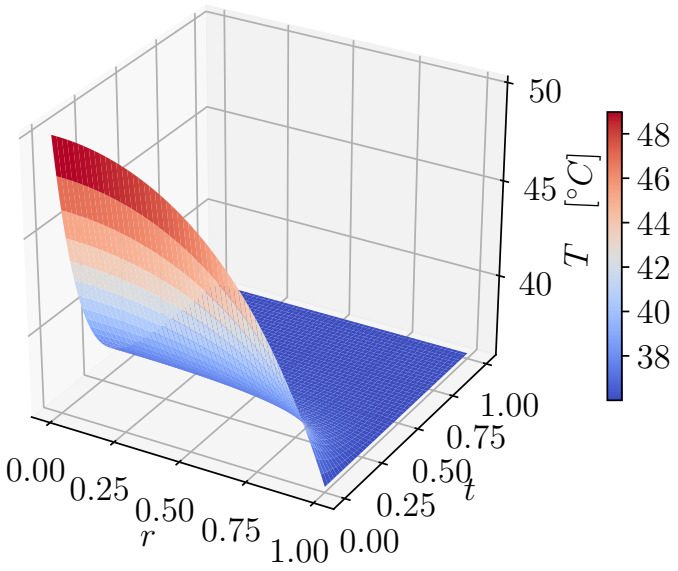


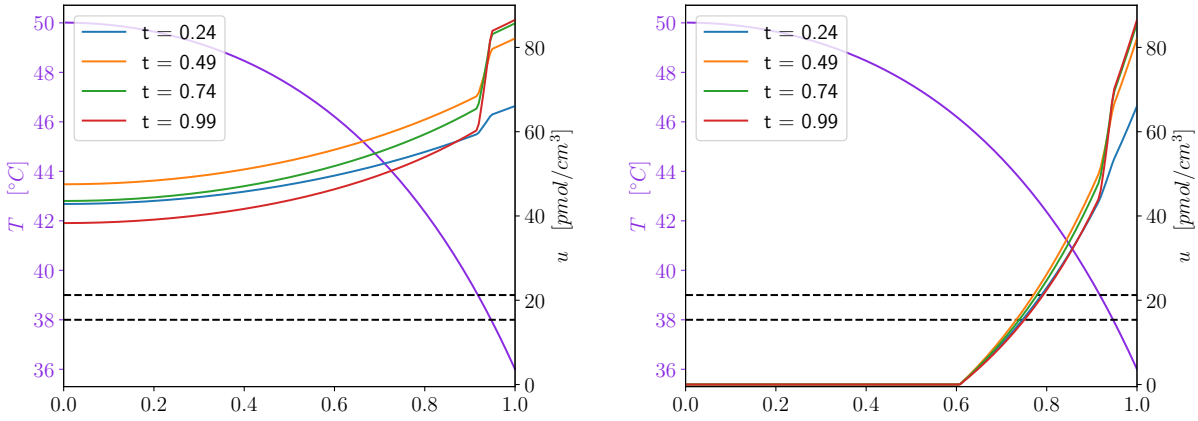
Figure 4.4: Temperature distribution within the tumor.

4.3 Temperature distribution

We assume a temperature field that has the value 50 $°C$ in the center of the tumor and decreases to 36 $°C$ body temperature at the edge of the tumor. We approximate the decay of the temperature similar to the interstitial pressure so that we use a hyperbolic sine, whereas the temporal aspect is taken from Gasselhuber et al. (2010, Figure 9). The exponential decay of the temperature was taken into account

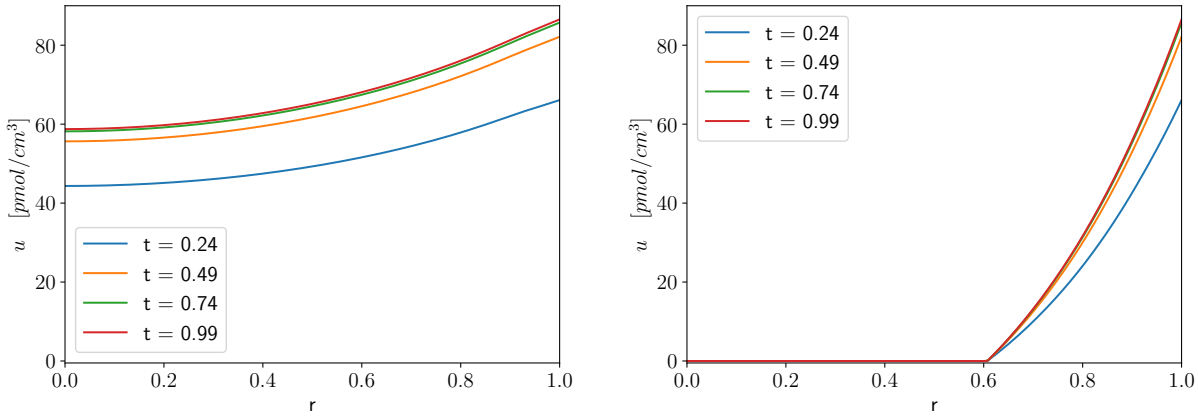
$$\begin{aligned} T(r,t) &= T_r \cdot T_t + T_0 \\ &= \left(20 - \frac{20\sinh(3r)}{r\sinh(3)}\right) e^{-\lambda t} + 36, \end{aligned} \tag{4.2}$$

where r is the normalized spacial component and t is the time. The exponential decay constant $\lambda = 8.32\ h^{-1}$ was estimated from the half-life of ca. 5 minutes (Gasselhuber et al., 2010, Figure 9). The function is depicted in Fig. 4.4.



(a) $\alpha = 1$ (b) $\alpha = 3$

Figure 4.5: Concentration of liposomes u in a steady-state temperature field in absence of transport. The transition zone of the drug release is marked by dashed lines.



(a) $\alpha = 1$ (b) $\alpha = 3$

Figure 4.6: Concentration of liposomes u in a time dependent temperature field in absence of transport.

4.4 Liposome concentration

4.4.1 Reaction term

At first, we a look at the reaction equation on its own (Fig. 4.5) in the steady state of the temperature field, such that $T(r,t) = T(r,0)$. Although the assumption of the steady state

is not realistic, the results from that calculation help to understand the impact of both terms within the reaction term.

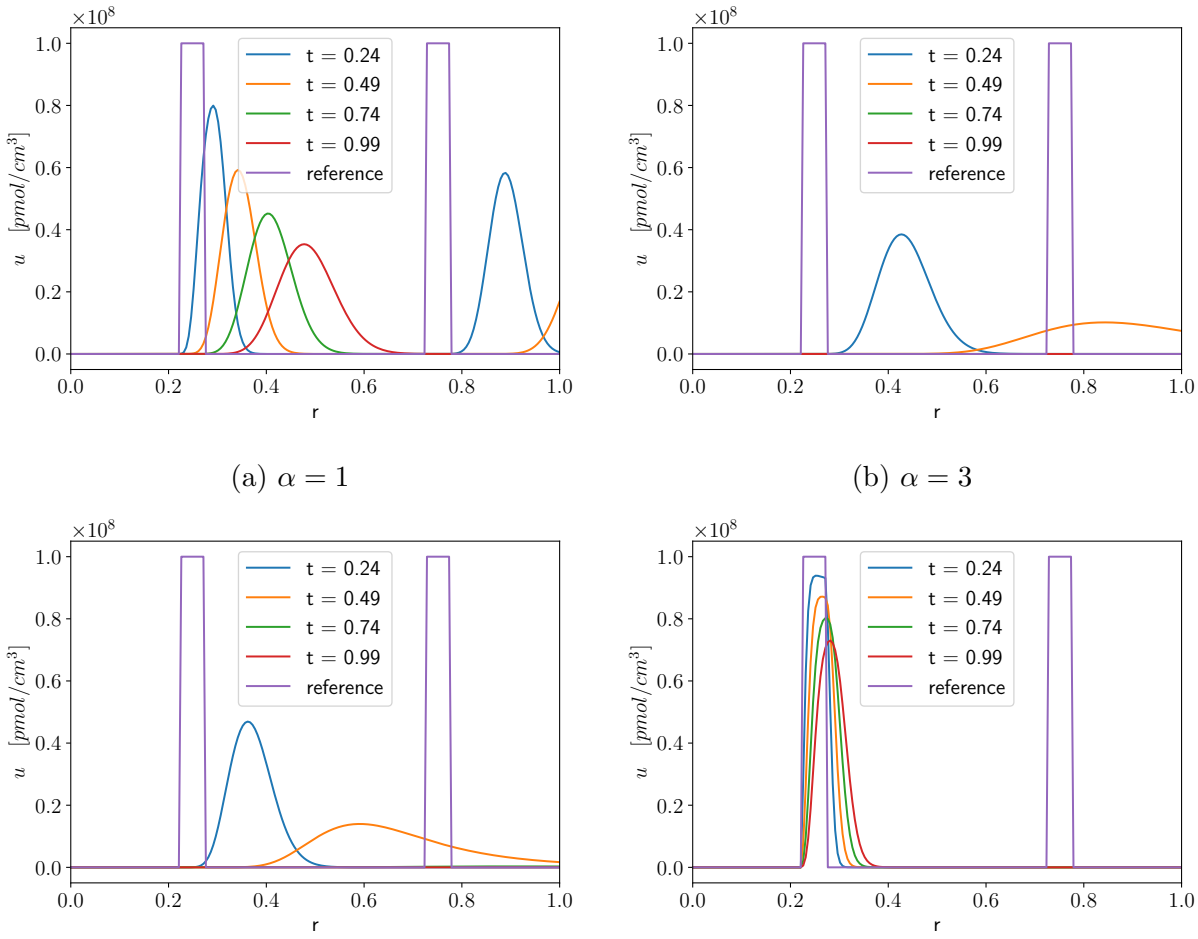
The temperature increases towards the center of the tumor and the transition zone for the drug release ($38 - 39\ ^{\circ}C$) is marked by a dashed line in Fig. 4.5. In the beginning, the liposomes enter the tumor at a rate proportional to the pressure difference. As soon as they reach the transition zone the liposomes start to release their cargo such that the concentration of liposomes decreases towards the center of the tumor. This explains the steep slope at approximately $r = 0.9$. With $\alpha = 3$ this structure of the individual terms is still visible, but the more α increases, the more the slope becomes uniformly and steep.

Since the temperature of the tissue cools down quickly with time, the liposomes decay in reality at a lower rate (Fig. 4.6). So in the realistic simulation the reaction term will be dominated by the exponential decay of the temperature and of the liposome concentration in the plasma.

4.4.2 Advection term

To examine the advection term, an initial block of higher concentration of liposomes was set on the domain in the interval [0.225, 0.275] and [0.725, 0.775]. These locations are chosen for analytic purposes and are not quite appropriate. The pure advection becomes visible without the differences in concentration due to the reaction terms (Fig. 4.7). This part of the analysis is important to show the impact of the interstitial velocity v (Fig. 4.3) on Eq. 2.4. Additionally the significant artificial diffusion resulting from the upwind differencing scheme becomes visible.

Since the velocity is increasing towards the edge of the tumor, the peaks of concentration are not symmetric but faster moving at the right end than at the left end. This increases the artificial diffusion of the upwind differencing method even more. The material tends to stream out of the tumor, although the likelihood depends on the distance, which it can pass during that amount of time. Therefore the likelihood of liposomes streaming out of the



(a) $\alpha = 1$ (b) $\alpha = 3$

(c) $\alpha = 5$ (d) $\alpha = 10$

Figure 4.7: Advection term of u

tumor depends on α.

Notable is the peak of advection in the plot $\alpha = 3$ (Fig. 4.7b). If α has a smaller value, then the velocity is too weak to move the material far away. With higher values of α the magnitude of velocity decreases to some extent in comparison to those values with lower α, but towards the edge of the tumor it increases more rapidly.

4.4.3 Full equation

The impact of α is even more visible when we have a look at how far the liposomes penetrate the tumor. Since the IFP depends on α, so does the reaction term. The liposomes cannot

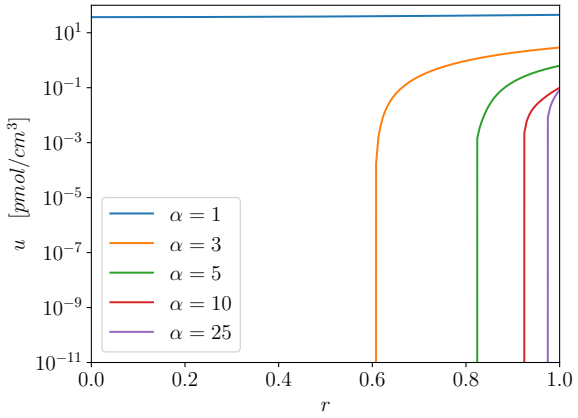


Figure 4.8: Liposome distribution over r during one hour

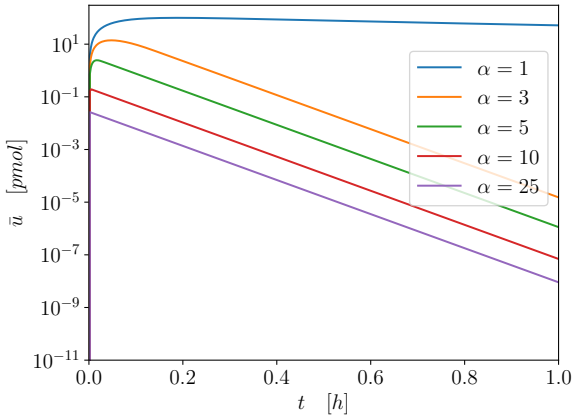


Figure 4.9: Total amount of liposomes over t

pass the point where the IFP outweighs the vascular pressure (Fig 4.2 and Fig. 4.8).

The total amount of liposomes within the tumor can be calculated by integrating u as described in Eq. 2.11 and is depicted in Fig. 4.9. This relatively high concentration in comparison to the initial concentration c_0 within the plasma results from the change of volume. The blood volume is ca. 5 L whereas the tumor volume is only 1 cm^3. The concentration is plotted on a logarithmic scale such that the decay is in fact exponential, resulting from Eq. 2.1.

4.5 Drug concentration

4.5.1 Diffusion term

Drug molecules are subject to diffusion. The diffusion acts autonomously since it does not depend on the liposome concentration u or the ratio of vascular to interstitial permeability α. Due to the concentration within w the drug molecules diffuse toward areas with less concentration as depicted in Fig. 4.10. This motion is independent of the flow directions such that the drug can move further into the tumor due to diffusion and especially pass the point where the IFP outweighs the vascular pressure.

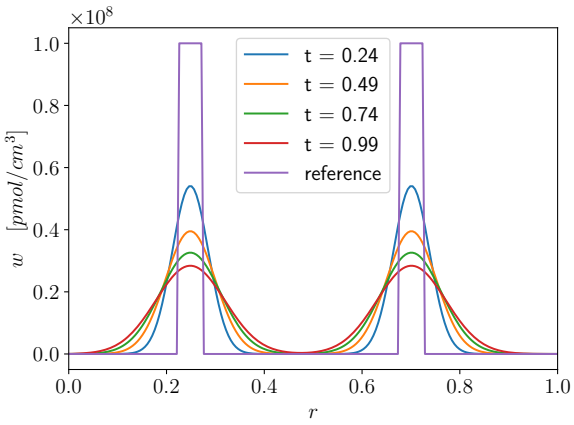


Figure 4.10: Diffusion term of w

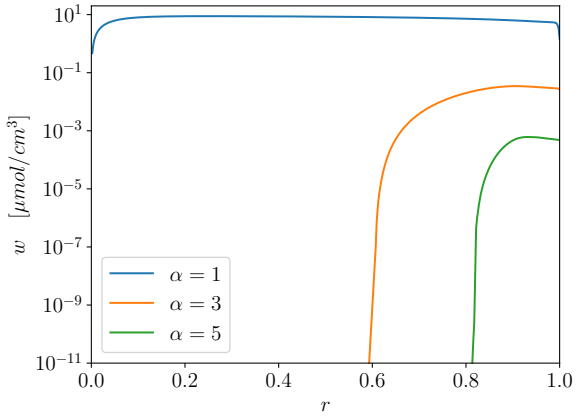


Figure 4.11: Drug distribution over r

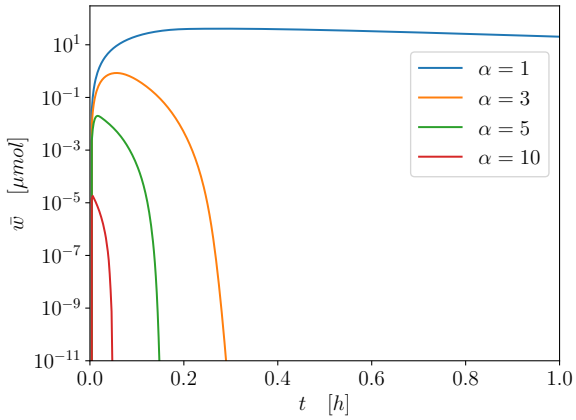


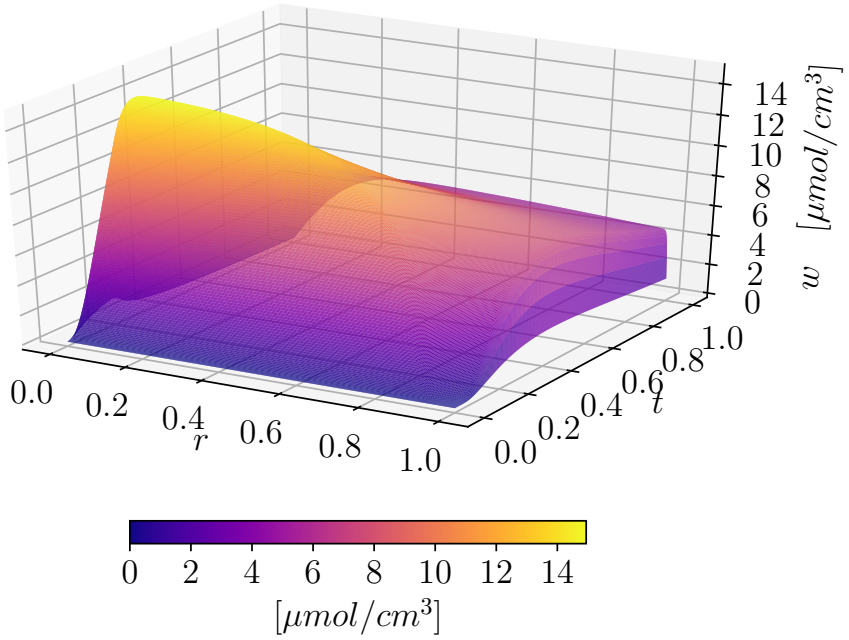
Figure 4.12: Total amount of drug over t

4.5.2 Full equation

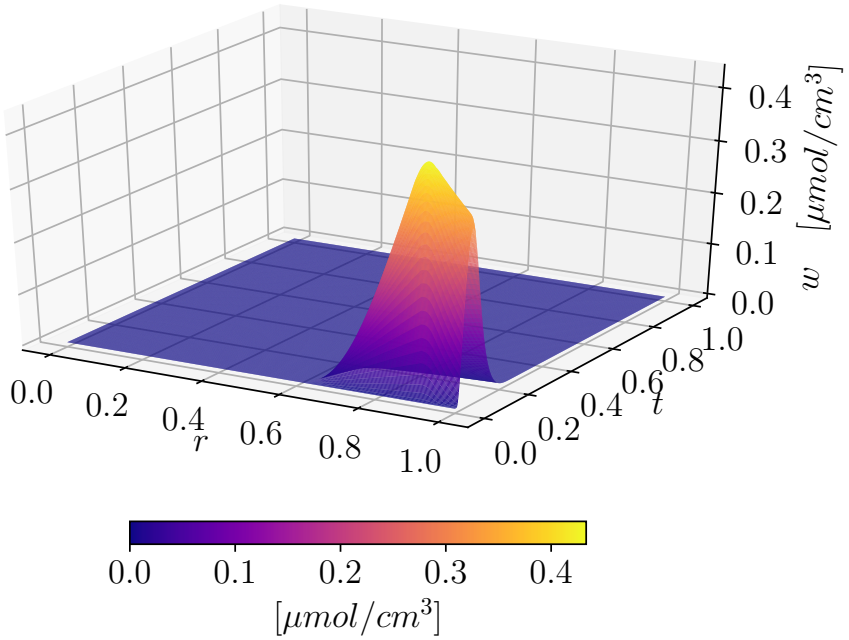
For the temperature distribution we assume that the tumor is heated by a probe located in the center of the tumor. If the liposomes cannot move far enough into the tumor such that their temperature reaches a specific threshold, no drug can be released. This is the reason why in the case of $\alpha \geq 25$ no drug concentration can be found in the tumor. Looking at the distance the drug moves into the tumor, the impact of the diffusion becomes clearly visible. The reaction term just generates the drug in place, whereas the advection term moves the drug outward. However, integrating over the time (Fig 4.11), the drug moves further inward than the liposomes (Fig. 4.8) due to the diffusion.

The distribution of the total amount of drug within the tumor can be calculated by integrating w as described in Eq. 2.11 and is depicted in Fig. 4.12. On the one hand, the amount of drug in the tumor is impacted by the amount of liposomes in the tumor. If less liposomes are present then less drug molecules can be released. On the other hand, the release of the drug is impacted by the temperature since the release rate h depends on the temperature. Thus, the release depends on the time. When the tumor cools down, less drug can be released from the liposomes. Since α controls the depth of the import of liposomes and the velocity and consequently the advection it has also an impact on the retention time of the drug within the tumor. An increasing α leads to a shorter retention time.

Notable is the decay of w with increasing α as described in the beginning of this section. The influence of α on the concentration of the drug becomes obvious, when w is plotted three-dimensional with respect to time and space (Fig. 4.13). Although the difference of α is not big, the difference of the drug concentration is clearly affected.



(a) $\alpha = 1$



(b) $\alpha = 3$

Figure 4.13: Concentration of the drug with respect to space and time

Chapter 5

Conclusion

5.1 Summary

The simulation of tumor growth and tumor treatment is one way for a better understanding of the particular components in these processes. Precise measurements of drug concentration in tumors requires a lot of pre- and post-processing steps, whereas mathematical simulation gives immediate results. On the other hand, mathematical simulations are only as reliable as the model. The more physical processes and side effects are neglected the less precise the simulation results. After setting up a mathematical model, the impact of single terms on the system can be analyzed.

We have set up a mathematical model that represents the delivery of temperature-sensitive liposomes and the release of their cargo within tumor tissue. These liposomes are spherical vesicles containing the drug and release their cargo in regions of increased temperature. We have modeled the elevated interstitial pressure and the interstitial velocity based on the model of Baxter and Jain (1989). For the concentration of the liposomes we extended the model given in Stapleton et al. (2013) and coupled that concentration with the concentration of the drug itself. The model for the drug release follows the idea of Hinow et al. (2016), where the drug is released by a transfer function. Therefore, we implemented some results for thermo-sensitive drug delivery referring to Gasselhuber et al. (2010). Our current model can be improved or extended as described in Ch. 5.2.

The parametrization of the mathematical model relies on parameters from available literature. A specific threshold exists for temperature-sensitive liposomes causing them to release their cargo as shown in Fig. 4.1 (Gasselhuber et al., 2010, Figure 2). This threshold is usu-

ally achieved by heating such as radio frequency tumor ablation. The temperature-sensitive liposomes are used to complement the tumor treatment by RFA. Around the area of ablated tissue the tumor will be treated additionally by the drug released from the liposomes (Gasselhuber et al., 2010).

The release of the drug depends on the temperature which itself has a temporal and a spatial component. We varied the ratio of vascular to interstitial permeability α in the simulation. As α increases the retention time of the drug within the tumor decreases due to a faster advection. An improvement of the efficiency of the treatment could be an incremental heating of the tumor, such that the liposomes have a better chance to release their cargo.

Developing temperature-sensitive liposomes for targeted anticancer drug delivery is difficult to realize. The liposomes must be able to move far into the tumor although the interstitial pressure counteracts the transport of the liposomes. The release of the drug must be triggered easily enough so that the tumor can be treated sufficiently. However, liposomes outside the tumor should tightly seal the drug to prevent toxic side effects. The diffusion of the drug molecules improves the penetration depth of the tumor.

5.2 Discussion

The mathematical model as well as the numerical approximation can be improved in the future. The most important step in the future is the improvement of the model of the drug. On the one hand, the drug uptake from the cells is not considered yet. This may impact the amount of concentration as well as the behavior of the drug molecules within the tumor in view of the advection and diffusion. The actual intention of the drug is the treatment of tumor cells. The decomposition and metabolism of the drug is not considered yet. An example for the metabolism can be found in Hinow et al. (2016). Those drug molecules which are not taken up from the tumor cells will start to dissociate after some time. After that they cannot treat the tumor cells any longer and should be neglected in the further

calculations.

Currently this model simulates the drug delivery within an isolated tumor. In the future, we can extend this model for tumors surrounded by healthy tissue. This will impact the interstitial pressure and the interstitial velocity at the edge of the tumor as well as the temperature field since the transition between the tissues has to be continuous.

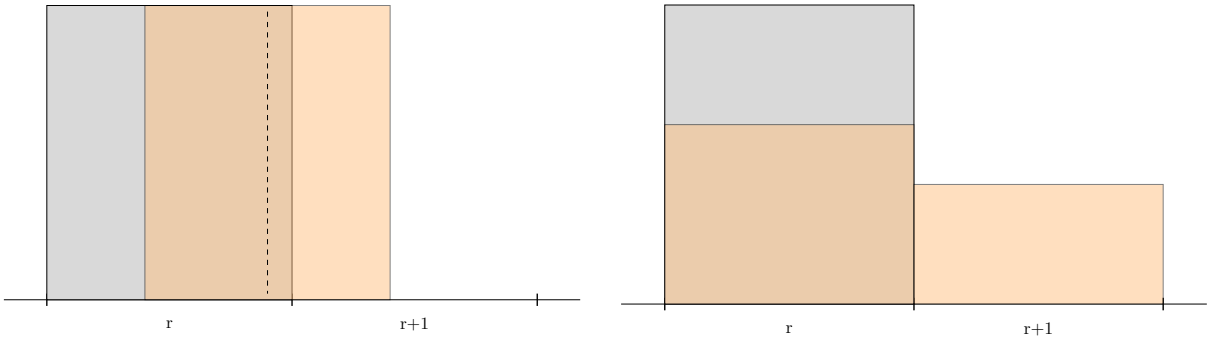
The Operator Splitting method has an order of $\mathcal{O}(\Delta t^2)$. However, this order can only be maintained if the numerical methods used in combination with the splitting are of at least the same order. Therefore the implicit Euler method and the upwind differencing method (both explained in Ch. 3.1) need to be replaced by more accurate methods. For sufficiently precise results the time step size and the grid resolution have to be very small which lead to long simulation times. By more precise numerical methods the number of calculations can be reduced since bigger resolutions will result in likewise precise simulation results.

The L-stable implicit Euler method is the first representative of the backward differentiation formulas and is sometimes abbreviated as BDF1. It is the first representative since it only uses one previous step. A suggestion for an improvement is using the second backward differentiation formula (BDF2), which has an order of $\mathcal{O}(\Delta t^2)$ and uses the two previous steps for the calculation of the next step. So instead of Eq. 3.4 we receive the following differentiation:

$$u_{\tau+1} = \frac{4u_\tau - u_{\tau-1} + 2\Delta\tau g(r, \tau + \Delta\tau, u_{\tau+1})}{3}, \tag{5.1}$$

where $g(r, \tau + \Delta\tau, u_{\tau+1})$ is still defined by Eq. 3.5.

Many numerical methods exist to approximate the advection equation. Therefore, the upwind differencing method could be replaced by a more accurate method. One of these methods was introduced by Egan and Mahoney (1972). This method reduces the effects of the artificial diffusion by advecting the subgrid center of mass and the subgrid spread of the molecules along with the molecules themselves. This counteracts the error of the advection that the molecules just move partly into the next grid cell. The previously mentioned



(a) Method of Egan and Mahoney (1972) (b) Upwind differencing

Figure 5.1: Comparison between the method of Egan and Mahoney (1972) and upwind differencing

upwind differencing method stores only the amount of concentration within one grid cell. The amount within one cell decreases while it advects into the next cell. The decrease of the concentration peak in this simulation due to numerical diffusion is showed in Fig. 4.7. Especially in the cases of a ratio of vascular to interstitial permeability of $\alpha = 3$ or $\alpha = 5$ the diffusion is significant. However, the peak concentration should not decrease and the width of the mass should not spread (Fig. 5.1). Because of the data storage of the method by Egan and Mahoney (1972) the previous shape of the concentration curve can be restored from the additional information. In a one-dimensional case with constant velocity this method is exact. The problem with this method is the complex data storage, since not only the concentration, but also the center of mass and the distribution of the molecules need to be stored. The reconstruction is complex and so the current source code of the simulation needs modification.

Instead of approximating the temperature by Eq. 4.2, thermodynamics, e. g. as found in Gasselhuber et al. (2010), could be implemented into the model. Consequently a new partial differential equation (PDE) has to be solved and the results are used for the calculation of Eq. 2.4 and Eq. 2.8. This will improve the accuracy of the reaction terms in both equations and therefore the whole mathematical model.

Also some constants may be improved by a sufficient modeling of their component. For

example, the diffusion coefficient of the drug is dependent on the status of the tissue. Gasselhuber et al. (2010) refer to two different diffusion constants. Within the ablated tissue, the drug diffusion constant is smaller than in the non-ablated tissue. Therefore, the diffusion coefficient D can be extended as $D(T(r,t))$. The simulation itself can already calculate the diffusion with a continuous function as diffusion coefficient. In the future the function needs to be set up and parametrized. The value of the diffusion constant has to be a continuous function to avoid discontinuity in the solutions. Additionally, the fractional rate of liposome transport through the interstitium to fluid flow θ_l, as well as θ_d for drug transport, require further investigations. θ_l was taken from Stapleton et al. (2013) although it is not obvious how they received this value. A value for θ_d (currently $\theta_d = 1$) needs to be estimated in the same way. Overall, our parametrization was taken from the literature. In a next step the parametrization could be verified or improved by physical experiments.

Although the assumption of radial symmetry is valid, the results would be more realistic if the mathematical model would be extended to all three dimensions. Therefore, the model itself needs some adjustments as well as the implementation for the simulation.

Bibliography

Arnida, M. M. Janát-Amsbury, A. Ray, C. M. Peterson, and H. Ghandehari, 2011: Geometry and Surface Characteristics of Gold Nanoparticles Influence their Biodistribution and Uptake by Macrophages. *Eur J Pharm Biopharm*, **77(3)**, 417–423.

Atkinson, K. E., 2014: *An Introduction to Numerical Analysis*. 2d ed., John Wiley & Sons, U.K.

Bangham, A. D., M. N. Standish, and J. C. Watkins, 1965: Diffusion of univalent ions across the lamellae of swollen phospholipids. *J. Mol. Biol.*, **13**, 238–252.

Baxter, L. T. and R. K. Jain, 1989: Transport of fluid and macromolecules in tumors. I. Role of interstitial pressure and convection. *Microvasc. Res.*, **37**, 77–104.

Casas, F., A. Murua, and M. Nadinic, 2012: Efficient computation of the Zassenhaus formula. *Computer Physics Communications*, **183**, 2386–2391.

Couture, O., J. Foley, N. F. Kassell, B. Larrat, and J.-F. Aubry, 2014: Review of ultrasound mediated drug delivery for cancer treatment: updates from pre-clinical studies. *Cancer Res*, **3(5)**, 494–511.

Egan, B. A. and J. R. Mahoney, 1972: Numerical modeling of advection and diffusion of urban area source pollutants. *Journal of Applied Meteorology*, **11 (2)**, 312–322.

Gasselhuber, A., M. R. Dreher, A. Negussie, B. J. Wood, F. Rattay, and D. Haemmerich, 2010: Mathematical spatio-temporal model of drug delivery from low temperature sensitive liposomes during radiofrequency tumour ablation. *Int. J. Hyperthermia*, **26**, 499–513.

Gasselhuber, A., M. R. Dreher, A. Partanen, P. S. Yarmolenko, D. Woods, B. J. Wood, and D. Haemmerich, 2012a: Targeted drug delivery by high intensity focused ultrasound mediated hyperthermia combined with temperature-sensitive liposomes: Computational modelling and preliminary *in vivo* validation. *Int. J. Hyperthermia*, **28**, 337–348.

Gasselhuber, A., M. R. Dreher, F. Rattay, B. J. Wood, and D. Haemmerich, 2012b: Comparison of conventional chemotherapy, stealth liposomes and temperature-sensitive liposomes in a mathematical model. *PLoS One*, **7**, e47 453.

Grossmann, C., H.-G. Roos, and M. Stynes, 2007: *Numerical Treatment of Partial Differential Equations*. Springer, translation and revision of the 3rd edition of "Numerische Behandlung Partieller Differentialgleichungen", Published by Teubner, 2005.

Grüll, H. and S. Langereis, 2012: Hyperthermia-triggered drug delivery from temperature-sensitive liposomes using MRI-guided high intensity focused ultrasound. *J. Control. Release*, **161**, 317–327.

Hinow, P., A. Radunskaya, S. M. Mackay, J. N. J. Reynolds, M. Schroeder, E. W. Tan, and I. Tucker, 2016: Signaled drug delivery and transport across the blood-brain barrier. *J. Liposome Res.*, **26**, 233–245.

Hundsdorfer, W., 2000: Numerical Solution of Advection-Diffusion-Reaction Equations. Thomas Stieltjes Institute, Lecture notes.

Ishida, T., H. Harashima, and H. Kiwada, 2002: Liposome clearance. *Bioscience Rep.*, **22**, 197–224.

Kneidl, B., M. Peller, G. Winter, L. H. Lindner, and M. Hossann, 2014: Thermosensitive liposomal drug delivery systems: state of the art review. *Int. J. Nanomed*, **9**, 4387–4398.

Kuang, Y., J. D. Nagy, and S. E. Eikenberry, 2016: *Introduction to Mathematical Oncology*. CHapman & Hall/CRC.

Liu, J., Y. Huang, A. Kumar, A. Tan, S. Jin, A. Mozhi, and X.-J. Liang, 2014: pH-Sensitive nano-systems for drug delivery in cancer therapy. *Biotechnology Advances*, **32**, 693–710.

Maeda, H., 2012: Macromolecular therapeutics in cancer treatment: The EPR effect and beyond. *J. Control. Release*, **164**, 138–144.

Maeda, H., J. Fang, T. Inutsukaa, and Y. Kitamoto, 2003: Vascular permeability enhancement in solid tumor: various factors, mechanisms involved and its implications. *International Immunopharmacology*, **3**, 319–328.

Matsumura, Y. and H. Maeda, 1986: A New Concept for Macromolecular Therapeutics in Cancer Chemotherapy: Mechanism of Tumoritropic Accumulation of Proteins and the Antitumor Agent Smancs. *Cancer Research*, **46**, 6387–6392.

Miao, L., C. M. Lin, and L. Huang, 2015: Stromal barriers and strategies for the delivery of nanomedicine to desmoplastic tumors. *Journal of Controlled Release*, **219**, 192–204.

Minchinton, A. I. and I. F. Tannock, 2006: Drug penetration in solid tumours. *Nat Rev Cancer*, **6(8)**, 583?92.

Pennes, H. H., 1948: Analysis of tissue and arterial blood temperatures in the resting human forearm. *J Appl Physiol*, **1**, 93–122.

Python Software Foundation, 2017: The Python Language. www.python.org, www.python.org.

Quarteroni, A., R. Sacco, and F. Saleri, 2010: *Numerical Mathematics*. 2d ed., Springer.

Reißel, M., 2015: Numerik für Differentialgleichungen I. Fachhochschule Aachen, Lecture Notes.

Rizzitelli, S., et al., 2015: Sonosensitive theranostic liposomes for preclinical in vivo mri-guided visualization of doxorubicin release stimulated by pulsed low intensity non-focused ultrasound. *Journal of Controlled Release*, **202**, 21–30.

Saxena, U., M. G. Klein, and I. J. Goldberg, 1991: Transport of lipoprotein lipase across endothelial cells. *Proceedings of the National Academy of Sciences of the United States of America*, **88 (6)**, 2254–2258.

Stapleton, S., M. Milosevic, C. Allen, J. Zheng, M. Dunne, I. Yeung, and D. A. Jaffray, 2013: A mathematical model of the enhanced permeability and retention effect for liposome transport in solid tumors. *PLoS One*, **8**, e81 157.

Staruch, R., R. Chopra, and K. Hynynen, 2011: Localised drug release using MRI-controlled focused ultrasound hyperthermia. *Int. J. Hyperthermia*, **27**, 391–398.

Strang, G., 1968: On the Construction and Comparison of Difference Schemes. *SIAM Journal on Numerical Analysis*, **5**, 506–517.

Taurin, S., H. Nehoff, and K. Greish, 2012: Anticancer nanomedicine and tumor vascular permeability; where is the missing link? *Journal of Controlled Release*, **164**, 265–275.

Taylor, A. E., 1981: Capillary Fluid Filtration. Starling Forces and Lymph Flow. *Circ Res.*, **49**, 557–575.

Wissler, E. H., 1998: Pennes' 1948 paper revisited. *J Appl Physiol*, **85**, 35–41.

Wong, A. D., M. Ye, M. B. Ulmschneider, and P. C. Searson, 2015: Quantitative Analysis of the enhanced permeability and retention (EPR) Effect. *PLoS One*, **10(5)**, e0123 461.

Zagar, T. M., et al., 2014: Two phase I dose-escalation/pharmacokinetics studies of low temperature liposomal doxorubicin (LTLD) and mild local hyperthermia in heavily pretreated patients with local regionally recurrent breast cancer. *Int. J. Hyperthermia*, **30**, 285–294.

Appendix A

The Operator Splitting Method

The idea of operator splitting can be explained most easily by considering a linear ordinary differential equation (ODE) system like

$$y'(t) = Ay(t), \tag{A.1}$$

with the analytical solution

$$y(t_{n+1}) = e^{A(t_{n+1}-t_0)}y(t_0) = e^{\Delta tA}y(t_n). \tag{A.2}$$

Here A denotes a matrix containing the coefficients for the system of equations and Δt denotes the time step size.

If A can be split into $A = \sum_{k=1}^{n} A_k$, then we can solve a system of sub-problems

$$\begin{aligned} y'(t) &= A_1 y(t) \\ &\vdots \\ y'(t) &= A_n y(t), \end{aligned} \tag{A.3}$$

such that the analytical solution (Eq. (A.2)) can be approximated by

$$y(t_{n+1}) = e^{\Delta tA_n} \dots e^{\Delta tA_1} y(t_n). \tag{A.4}$$

This method is the simplest splitting method with an convergence order of $\mathcal{O}(\Delta t)$. The term $\mathcal{O}(\Delta t)$ denotes a quantity whose size is proportional to Δt or even smaller. Definitions are summarized in Appendix A on page 38.

To achieve more accuracy and symmetry, the order of the A_k can be interchanged (Strang, 1968). Therefore the sequences of the sub-problems will be reversed, such that

$$\begin{aligned} y(t_{n+1}) &= \left(e^{\frac{\Delta t}{2}A_n} \dots e^{\frac{\Delta t}{2}A_1}\right)\left(e^{\frac{\Delta t}{2}A_1} \dots e^{\frac{\Delta t}{2}A_n}\right) y(t_n) \\ &= e^{\frac{\Delta t}{2}A_n} \dots e^{\frac{\Delta t}{2}A_2} e^{\Delta tA_1} e^{\frac{\Delta t}{2}A_2} \dots e^{\frac{\Delta t}{2}A_n} y(t_n). \end{aligned} \tag{A.5}$$

By applying a series expansion around $y(t + \Delta t)$ or using the Baker-Campbell-Hausdorff formula (Hundsdorfer, 2000) and the Zassenhaus formula (Casas et al., 2012), the local and global truncation error can be obtained. Overall, the operator splitting using the Strang method has an accuracy of $\mathcal{O}(\Delta t^2)$.

Appendix B

Definitions

To apply a numerical method within a simulation, the results gained by using that method should approximate the real solution closely. There exist some measurements for the quality of a numerical method, namely stability, consistence and convergence. Among others, the following definitions classify the methods used in Chapter 3.

Convergence of Numerical Methods

A fundamental theorem for the numerical solution of ODEs was developed by Peter Lax and Robert D. Richtmeyer and can be easily extended to PDEs. In the following the theorem itself and the necessary definitions will be given.

Theorem. Lax-Richtmeyer Theorem (Reißel, 2015, page 22)
A numerical method is convergent iff it is 0-stable and consistent. Furthermore, its order of convergence equals its oder of consistence.

A proof of this theorem can be found for example in Quarteroni et al. (2010, page 41). Therefore, a consistent method is convergent if it is 0-stable and a 0-stable method is convergent if it is consistent. The zero-stability describes numerical methods that are not sensitive to noisy input data.

Definition. Zero Stability (Reißel, 2015, page 21)
Given a general numerical method $\phi(\Delta t, t, y)$ approximating $y'(t) = f(t, y)$ we can calculate the solution and a noisy solution for the next time step by

$$y_{i+1} = \phi(\Delta t, t, y_i), \qquad y_0 = y_0, \tag{B.1}$$
$$\tilde{y}_{i+1} = \phi(\Delta t, t, \tilde{y}_i) + \Delta t \delta_{i+1}, \qquad y_0 = y_0 + \delta_0. \tag{B.2}$$

For

$$|\delta_i| \leq \varepsilon \quad \forall i \tag{B.3}$$

a method is called *zero-stable* if

$$\exists \Delta t_0 > 0 \;\; \exists c_\varepsilon > 0 : |y_i - \tilde{y}_i| \leq c_\varepsilon \varepsilon \quad \forall i, \; \forall \Delta t \leq \Delta t_0. \tag{B.4}$$

■

Consistence describes the precision of the solution of a numerical method. In other words, how accurate is the numerical approximation in comparison to the true solution. The order of consistence defines the order of convergence.

Definition. Consistence (Reißel, 2015, page 21)
Given the exact evolution $\Phi(\Delta t, t, y) = y(t) + \int_t^{t+\Delta t} f(\tau, y(\tau))d\tau$ for $y'(t) = f(t, y)$, a numerical approximation $\phi(\Delta t, t, y)$ is called *consistent* if

$$|\Phi(\Delta t, t, y) - \phi(\Delta t, t, y)| \xrightarrow{\Delta t \to 0} 0. \tag{B.5}$$

Furthermore, the *order of consistence* p is given by

$$|\Phi(\Delta t, t, y) - \phi(\Delta t, t, y)| = \mathcal{O}(\Delta t^{p+1}) \tag{B.6}$$

■

Stability of Numerical Methods

Beside the previous mentioned zero stability there exist more precise definitions of stability.

Definition. Absolute Stability (Quarteroni et al., 2010, page 489)
We consider the linear Cauchy problem (or so called *test problem*)

$$\begin{cases} y'(t) = \lambda y(t), & t > 0, \\ y(0) = 1 \end{cases} \tag{B.7}$$

with $\lambda \in \mathbb{C}$, whose solution is $y(t) = e^{\lambda t}$. As long as $\Re(\lambda) < 0$, $\lim_{t\to\infty} |y(t)| = 0$. A numerical method for approximating the test problem (B.7) is *absolutely stable* if

$$|y_n| \xrightarrow{t_n \to \infty} 0. \tag{B.8}$$

Let Δt be the discretization stepsize. Therefore, the numerical solution of y_n of (B.7) depends on Δt and λ and is not stable for all values of these parameters. The *region of absolute stability* of the numerical method is defines as a subset of the complex plane

$$\mathcal{S} = \{z = \Delta t\lambda \in \mathbb{C} : \text{(B.8) holds}\}. \tag{B.9}$$

So $\mathcal{S}$ contains all values of the product $\Delta t\lambda$ for which the solutions of the numerical method decay to zero as t_n tends to infinity. ■

A solution by a numerical method should behave similar to the real solution if it is applied to the test problem (B.7), for different λ and all $\Delta t > 0$. So the numerical method should be applicable independent of Δt. Since this strong requirement is not necessary for all applications the stability region $\mathcal{S}$ (B.9) gives information about those λ which can be represented by the method. The absolute stability can be classified even further.

Remark. A-Stability and L-Stability (Reißel, 2015, pages 64-68) A numerical method is called *A-stable* if

$$\mathbb{C}^- = \{z \mid z \in \mathbb{C}, \Re(z) \leq 0\} \subset \mathcal{S}, \tag{B.10}$$

so if the stability region $\mathcal{S}$ (B.9) coincides with the left complex half plane. Furthermore, a numerical method is called *L-stable* if it is A-stable and the solutions decay sufficiently fast such that

$$y_n \xrightarrow{\Re(\lambda) \to -\infty} 0. \tag{B.11}$$

Another definition of the L-Stability is the boundedness of the parameter λ

$$\Re(\lambda) < \infty, \qquad \Delta t \to \infty. \tag{B.12}$$

■

So the L-stable methods are suitable for stiff (or rapidly decreasing) problems whereas A-stable problems cannot represent the steep slope within the solution.

Appendix C

Source Code

```
#########################################################################
#                    G E N E R A L      I N F O                         #
#                                                                       #
#    This file gives helping functions, so implemented                  #
#    equations taken from  Stapleton2013 and Baxter1989. The            #
#    related equations are mentioned in the comments.                   #
#########################################################################
from numpy import sqrt, sinh, cosh, exp, pi
import logging, getopt, sys
from parameters import *

# Baxter1989, (8a)
# dimensionless interstitial pressure in isolated tumor
#
# Parameters:
# r : current radial position (cm)
# R : radius of tumor (cm)
# a : dimensionless ratio of vascular to interstitial
   ↪ permeability to fluid flow
def p_hat(r, a):
    if r==0:
        return 1 - a / sinh(a)
    r_hat = r/R
    return 1 - (sinh(a * r_hat) / (sinh(a) * r_hat))


# Baxter1989, (8a)
# interstitial pressure in isolated tumor (mmHg)
#
# Parameters:
# phat  : dimensionless interstitial pressure in isolated tumor
# p_e   : effective pressure (mmHg)
# p_inf : surrounding pressure (mmHg)
def p_i(phat):
    # Baxter1989, (8a)
    # return phat*(p_e - p_inf) + p_inf
    #### own choice
    return p_imax*phat
```

```
# Baxter1989, (8b)
# dimensionless interstitial velocity in isolated tumor
#
# Parameters:
# r : current radial position (cm)
# R : radius of tumor (cm)
# a : dimensionless ratio of vascular to interstitial
   ↪ permeability to fluid flow
def v_hat(r, a):
    if r==0:
        return 0.
    r_hat = r/R
    return (a * r_hat * cosh(a * r_hat) - sinh(a * r_hat)) / ((
       ↪ r_hat ** 2) * sinh(a))



# Baxter1989, (8b)
# interstitial velocity in isolated tumor (cm/sec)
#
# Parameters:
# vhat   : dimensionless interstitial velocity in isolated tumor
# R      : radius of tumor (cm)
# K      : hydraulic conductivity (cm**2/(mmHg * sec))
# p_e    : effective pressure (mmHg)
# p_inf  : surrounding pressure (mmHg)
def v_i(vhat):
    return p_imax* K_t  * vhat / R



# Stapleton2013, alpha (as in (2) )
# should be between 0.5 and 150
# dimensionless ratio of vascular to interstitial permeability to
   ↪  fluid flow
#
# Parameters:
# V     : Volume
# fc : capillary filtration coeff (1/(mmHg * sec))
# K     : hydraulic conductivity, interstitial permeability (cm
   ↪ **2/(mmHg * sec))
def alpha(V):
    return V**(1./3) * sqrt(fc_t/(K_t*V))
```

```
# first half of Stapleton2013, (1)
#
# Parameters:
# r : current radial position (cm)
# V    : Volume
# c    : plasma concentration of the nanoparticle
# fc : capillary filtration coeff
# p_v  : vasculature pressure (mmHg)
# phat : dimensionless interstitial pressure in isolated tumor
# pr_i   : interstitial pressure in isolated tumor (mmHg)
# sigma: filtration reflection coefficient
def lambd(r, a, t):
    phat = p_hat(r, a)
    pr_i = p_i(phat)
    if (p_v - pr_i) <= 0.:
        return 0.
    return fc_t * (p_v - pr_i) * (1 - sigma) * c(t)


# Concentration clearance
#
# Parameters:
# k : liposome clearance rate
# Cp: initial liposome concentration
# t : time
def c(t):
    return Cp*exp(-k*t)

# Ablation coefficient (Gasselhuber2010)
#
# Parameters:
# currently constant
def D(r, t):
    return 3600*6.7e-7 #cm^2/s


# Temperature distribution
#
# Parameters:
# r: current normalized radial position (cm)
def T(r, t):
    if r==0:
        r=0.00000001
    return (20 - (20. * sinh(3 * r))/(r * sinh(3)))*exp(-k2*t)+36
```

```
# heat transfer function
#
# Parameters:
# T : temperature
def h(T):
    if T <= 38:
        return 0
    elif T < 39:
        return (37.6 * T - 1428.8)/100
    else:
        return (1.76842105 * T  - 31.07368421)/100


##########################################################################
#                                                                        #
#          C A L C U L A T I O N   C O D E                               #
#                                                                        #
##########################################################################

# use Simpson rule
def integrate(u, t, r, Nr):
    h = 1.*R/Nr/3
    I = 0.0
    for i in range(1,int(Nr/2)):
        I += u[2*i-2,t]*r[2*i-2]**2 + 4*u[2*i-1,t]*r[2*i-1]**2 +
           ↪ u[2*i,t]*r[2*i]**2
    return I*h*4*pi

# use Operator splitting as explained in the thesis to calculate
   ↪ u and w
def getU(C0, Nr, tmin, tmax, al):
    dr = 1.*R/(Nr)
    dt = dr
    Ntau = 100
    dtau = dt/Ntau

    logging.info("dt = "+str(dt))
    logging.info("Nt = "+str(Nt))
    logging.info("dr = "+str(dr))
    logging.info("Nr = "+str(Nr))
    logging.info("alpha = "+str(al))


    # divide r into N equidistant points
    r = numpy.linspace(0, R, Nr)
```

```
    # create matrix
    u = numpy.zeros((Nr, Nt), dtype=numpy.float64)
    #initialize matrix
    u[:,0] = C0

    # Upwind differencing
    # for advection term: u_t + f*u_x = 0
    n = dtau/(dr)
    m = f*n
    logging.info("m = "+str(m))

    # speedup: calculate h and v
    local_h = numpy.zeros(Nr)
    local_v = numpy.zeros(Nr)

##########################################################################
# C O N C E N T R A T I O N    L I P O S O M E S
##########################################################################
    for t in range(Nt-1):
        # in each time step solve System with Strang Splitting
        # Strang splitting: 0.5A + B + 0.5A
        un = numpy.zeros((Nr, Ntau), dtype=numpy.float64)
        un[:,0] = u[:,t]
        logging.info("t="+str(t))
        logging.debug("Calculate u(r,t)")

        for x in range(Nr):
            local_h[x] = h(T(r[x], (t+1)*dt))
            local_v[x] = v_i(v_hat(r[x], al))

##########################################################################
# 0.5 EULER
##########################################################################
        for tau in range (int(Ntau/2)-1):
            for x in range(Nr):

                it=100
                error=1

                #fix point iteration for backward euler
                #initial euler step
                un[x,tau+1] = un[x,tau] + dtau*(lambd(r[x], al,
                    ↪ tau)/Vr - h(T(r[x], t*dt))*(un[x,tau]))
                while error > 1e-8:
```

```
                    u_n = un[x,tau] + dtau*(lambd(r[x], al, t+tau
                        ↪ )/Vr - local_h[x]*(un[x,tau+1]))
                    error_v = un[x, tau+1]-u_n
                    error = numpy.sqrt(numpy.dot(error_v,error_v)
                        ↪ )
                    un[x,tau+1] = u_n
                    it-=1
                    if it < 0:
                        print("Too many iterations")
                        break

        logging.debug("RK1_u")
        # set last value as start value
        un[:,0] = un[:,int(Ntau/2)-2]


#########################################################################
# 0.5 UPWIND
#########################################################################
        for tau in range (int(Ntau/2)-1):
            for x in range(1,Nr):

                # chain rule
                un[x,tau+1] = (1 - m*local_v[x])*un[x,tau] + m*
                    ↪ local_v[x-1]*un[x-1, tau]

        logging.debug("DC2_u")

#########################################################################
# 0.5 UPWIND
#########################################################################
        for tau in range(int(Ntau/2)-1, Ntau-1):
            for x in range(1,Nr):

                # chain rule
                un[x,tau+1] = (1 - m*local_v[x])*un[x,tau] + m*
                    ↪ local_v[x-1]*un[x-1, tau]

        logging.debug("DC2.2_u")
        # set last value as start value
        un[:,int(Ntau/2)-1] = un[:,Ntau-1]
```

```
#########################################################################
# 0.5 EULER
#########################################################################
        for tau in range(int(Ntau/2)-1, Ntau-1):
            for x in range(Nr):

                it=100
                error=1

                #fix point iteration for backward euler
                #initial euler step
                un[x,tau+1] = un[x,tau] + dtau*(lambd(r[x], al,
                   ↪ tau)/Vr - h(T(r[x], t*dt))*(un[x,tau]))
                while error > 1e-8:
                    u_n = un[x,tau] + dtau*(lambd(r[x], al, t+tau
                       ↪ )/Vr - local_h[x]*(un[x,tau+1]))
                    error_v = un[x, tau+1]-u_n
                    error = numpy.sqrt(numpy.dot(error_v,error_v)
                       ↪ )
                    un[x,tau+1] = u_n
                    it-=1
                    if it < 0:
                        print("Too many iterations")
                        break

        # set next value of u
        u[:,t+1] = un[:,-1]

        logging.debug("RK1.2_u")

    # fix boundary
    u[0,:] = u[1,:]

    return u


def getW(u, w0, Nr, tmin, tmax, al):
    dr = 1.*R/(Nr)
    dt = dr
    Ntau = 100
    dtau = dt/Ntau

    logging.info("dt = "+str(dt))
    logging.info("Nt = "+str(Nt))
    logging.info("dr = "+str(dr))
```

```
    logging.info("Nr = "+str(Nr))
    logging.info("alpha = "+str(al))

    # divide r into N equidistant points
    r = numpy.linspace(0, R, Nr)
    # create matrix
    w = numpy.zeros((Nr, Nt), dtype=numpy.float64)
    #initialize matrix
    w[:,0] = w0
    # Upwind differencing
    # for advection term: w_t + w_x = 0
    n = dtau/(dr)
    m = f*n
    logging.info("m = "+str(m))

    # speedup: calculate h, v and D
    local_h = numpy.zeros(Nr)
    local_v = numpy.zeros(Nr)
    local_D = numpy.zeros(Nr)

##########################################################################
# C O N C E N T R A T I O N      D R U G
##########################################################################
    for t in range(Nt-1):

        # in each time step solve System with Strang Splitting
        # Strang splitting: 0.5A + 0.5B + C + 0.5B + 0.5A
        wn = numpy.zeros((Nr, Ntau), dtype=numpy.float64)
        wn[:,0] = w[:,t]

        for x in range(Nr):
            local_h[x] = h(T(r[x], t*dt))
            local_v[x] = v_i(v_hat(r[x], al))
            local_D[x] = D(r[x], t*dt)

        logging.info("t="+str(t))
        logging.debug("Calculate w(r,t)")

##########################################################################
# 0.5 INTEGRATION
##########################################################################
        for tau in range (int(Ntau/2)-1):
            for x in range(Nr):
                wn[x,tau+1] = wn[x,tau] + beta*local_h[x]*u[x,t
                    ↪ ]*(dtau)
```

```
        logging.debug("RK1_w")
        # set last value as start value

        wn[:,0] = wn[:,int(Ntau/2)-2]

#########################################################################
# 0.5 UPWIND
#########################################################################
        for tau in range (int(Ntau/2)-1):
            for x in range(1,Nr):

                # chain rule
                wn[x,tau+1] = (1 - n*local_v[x])*wn[x,tau] + n*
                   ↪ local_v[x-1]*wn[x-1, tau]

        logging.debug("DC2_w")

        # set last value as start value
        wn[:,0] = wn[:,int(Ntau/2)-2]

        kappa = dtau/(2.0*dr**2)

#########################################################################
# CRANK NICOLSON
#########################################################################
        for tau in range(1,Ntau):

            temp = wn[:, tau-1] # get values of previous time
               ↪ step
            b_r = numpy.zeros((Nr), dtype='float64')
            A = numpy.zeros((Nr,Nr), dtype='float64') # only
               ↪ inner values
            for x in range(1,Nr-1):
                A[x,x] = ( 1 + 2*kappa*local_D[x] )
                A[x, x+1 ] = -kappa*local_D[x+1]
                A[x, x-1] = -kappa*local_D[x-1]
                b_r[x] = kappa*local_D[x+1]*temp[x+1] + (1 - 2*
                   ↪ kappa*local_D[x])*temp[x] + kappa*local_D[x
                   ↪ -1]*temp[x-1]

            # use boundary values
            A[0,0] = ( 1 + 2*kappa*local_D[0])
            A[0, 1 ] = -kappa*local_D[1]
            A[Nr-1,Nr-1] = ( 1 + 2*kappa*local_D[Nr-1] )
            A[Nr-1, Nr-2] = -kappa*local_D[Nr-2]
```

```
            # solve system of equations
            wn_new = numpy.linalg.solve(A,b_r)

            # save values from this time step
            wn[:, tau] = wn_new

        logging.debug("CN3_w")
        # set last value as start value
        wn[:,int(Ntau/2)-1] = wn[:,Ntau-1]

#########################################################################
# 0.5 UPWIND
#########################################################################
        for tau in range(int(Ntau/2)-1, Ntau-1):
            for x in range(1,Nr):

                # chain rule
                wn[x,tau+1] = (1 - n*local_v[x])*wn[x,tau] + n*
                   ↪ local_v[x-1]*wn[x-1, tau]

        logging.debug("DC2.2")
        # set last value as start value
        wn[:,int(Ntau/2)-1] = wn[:,Ntau-1]

#########################################################################
# 0.5 INTEGRATION
#########################################################################
        for tau in range(int(Ntau/2)-1, Ntau-1):
            for x in range(Nr):

                wn[x,tau+1] = wn[x,tau] + beta*local_h[x]*u[x,t
                   ↪ ]*(dtau)

        # set next value of w
        w[:,t+1] = wn[:,Ntau-1]

    # end of time loop
    w[0,:] = w[1,:]

    return w
```