Mathematical modelling of slow drug release from collagen matrices

Master thesis in Applied and Computational Mathematics

Birgitte Riisøen Erichsen

Department of Mathematics
University of Bergen
22 September 2014

Acknowledgements

First I would like to thank my supervisor Florin Radu for never-ending support and an open door throughout my work with this thesis.

I was fortunate enough to go on two trips during my work. From the Ludwig-Maximilians-University in Munich I would like to thank Prof. Dr. Wolfgang Frieß and his work group, especially Madeleine Witting for helping me in the lab and with making sense of the data. In Eindhoven University of Technology I would like to thank Prof. Sorin Pop for his hospitality and help with the new model.

Thank you also to all my friends from UiB for five great years. It has been a pleasure getting to know you all! Teach First Norway, Cohort 5, Arne, Andreas - I owe you one! Last but not least, thank you to my family, friends and Stian for much needed breaks, love and support, I couldn't have done it without all of you.

Birgitte Riisøen Erichsen
September 2014

Abstract

This master's thesis is about controlled drug release, which is a relatively new area of mathematical modelling. In this thesis there have been two major focuses. The first is to further understand the model for drug release from collagen matrices developed in [14], by solving it with a different numerical scheme, and the second to develop a new model based on a different geometry. Both models are based on mass conservation and Fick's law, and are therefore possible to compare. The two models have been discretized and implemented, and the results compared to experimental data.

Contents

Introduction

There are several different ways to insert a drug into the body. The best way depends not only on the drug itself, but also whether we want it to be released fast or slowly. For a fast drug release there are pills and tablets available if the drug can be absorbed in the digestive system and transported in the blood stream. If not, there are lotions that transport drugs through the skin. However, in order to have a controlled drug release, or sometimes to reach deep tissues, we need implants to be inserted into the body, remain stationary and release drugs gradually. These implants are known as drug deliver systems. They consist of a skeleton, known as the matrix, which traps the drug, and the drug itself. Figure 1 shows a matrix, its fiber structures and the trapped drug as red dots.

There are two kinds of drug delivery systems; degradable, made for instance of collagen, and non-degradable, made for instance of polymer nanocomposites [4]. Non-degradable drug delivery systems will need to be removed after the drug is released, because they do not naturally occur in the body, and they may therefore cause side-effects or be repelled. The degradable ones will not need removal as they will be broken down. For a degradable drug delivery system the risk associated with the matrix itself is small because the matter making the matrix is known to the body. However, it will
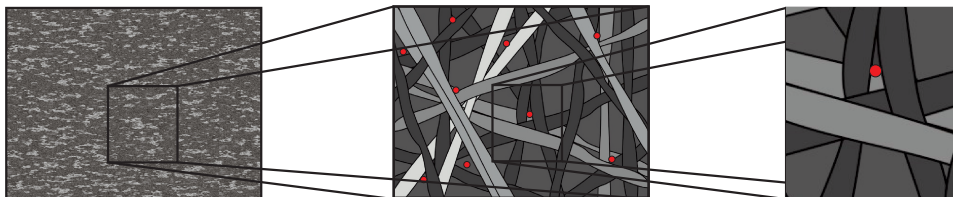

Figure 1: Collagen matrix with trapped drugs [14]

be more difficult to predict the release, which requires a more complicated mathematical model. In a non-degradable matrix the drug-release mostly happens due to diffusion, but in the degradable ones, the process is more complicated.

As the matrix is inserted into the body, it will absorb fluids from its surroundings. During this process the drug is released by diffusion. Simultaneously, the enzyme collagenase starts to break down the collagen. This enzyme is found everywhere collagen is found, which implies that the matrix should be inserted somewhere collagen naturally occurs. Collagenase continues to break down the collagen, which in turn releases more of the drug. This is the main cause of the drug release. When modelling this release, it can be argued that the two processes happen on a very different timescale. When the implant swells, only very small amounts of collagen is broken down, and after the matrix is fully swollen, there is hardly any diffusion.

Collagen matrices are a good degradable drug delivery system, because collagen is the main component in our connective tissue. This is the tissue that supports, connects or separates different kinds of tissues and organs in the body [9]. The body can also use the added collagen as it has positive effects on tissue regeneration. This is for instance good in the case of tumours, infections or formation of new tissue [12]. Collagen is a well-known and common substance, which has the added benefit that there has already been done a lot of research on it. Therefore a lot of its properties are known, and experiments will only be needed on the drug delivery itself. It is also cheap and easily accessible, since it is a leftover from the leather industry.

However, considering that experiments, both in the laboratory and on humans, are time-consuming and can be expensive, mathematical modelling of the release saves both time and money. Both the swelling and the degradation of the matrix can be modelled, but as they are on very different time scales, they may be modelled separately. There were many attempts to model drug release from collagen matrices, see e.g. [9, 12, 18, 17], but most of the papers so far do not properly include the effects of the evolving microscale. This was done for the first time in [14] for a grain geometry. This thesis continues the work in [14] by considering a channel geometry.

In this thesis we develop a new mathematical model for drug release from enzymatically degradable collagen matrices, which properly includes the effects of the evolving microstructure. The model is done for a simplified

geometry, e.g. a thin channel geometry. The model consists of a system of coupled partial and ordinary differential equations. Due to its complexity, the system has no known analytical solutions and therefore a numerical solution is needed. We will use the multipoint flux approximation method (MPFA) for the discretization in space and Euler's method for the temporal discretization. Numerical results based on the new mathematical model will be presented and compared with real data. A good correlation between the numerical and experimental results is observed, which sustain the developed model.

This thesis is structured as follows.

Chapter 1 concerns collagen and its properties giving some background information about collagen itself and why we want to use collagen implants for controlled drug release.

In the second chapter we focus on the mathematical modelling. The grain model, based on [14] will be described, and the new channel model will be derived from microscale and upscaled. There is also a comparison between the equations in the two models, where we look at how changing the geometry changed the sets of equations.

The third chapter is about the implementation of the equations. This also includes background theory on the methods used to discretize the equations. The implementations have been done in MATLAB, and there will be a description of how the sets of equations were adapted to be implemented.

Chapter 4 contains the results. These include convergence analysis from developing the programs, some stability tests and comparison between the numerical solutions and experimental data. Some of the experimental data is from [10, 9] and some from experiments we performed in October 2013.

The last chapter is devoted to conclusions and future work.

Chapter 1

Collagen based drug delivery systems

This chapter will be about drug delivery systems based on collagen for controlled release. We will look at the properties of collagen, and what makes is a good system for controlled drug release, as well as the devices themselves and the challenges connected to the research. Finally, we will give a quick overview of the experiments that we performed in October 2013.

1.1 Collagen

Collagen is the most common protein in the human body. At least 30 % of the protein is collagen and it is proposed to be a key structural component of load-bearing tissues, which includes bones, tendons and ligaments. Collagen is also found in the skin [5]. It is predominantly located in the extracellular matrix, which is the fluid between the cells. In the skin, more than 50 % of the extracellular protein consists of collagen, and for tendon and bone, it is more than 90 % [5, 7, 15]. The collagen spectrum ranges from tendons to placental membranes to the cornea. Hence, different collagen types with different properties, like mechanical strength and elasticity, are necessary in the various types of connective tissue [6, 7].

There is no proper definition of collagen, but it comprises a family of genetically distinct molecules. These have a unique triple-helix configuration in common. These helices consist consist of long chains of different amino

acids. Disparities of collagen types can be due to variation in the length of the helix, as well as the size of non-helical areas within it [6, 7, 15]. These areas vary from almost none (4 % for collagen I) to more than 90 % (for collagen XII). The helix is made of three polypeptide chains, referred to as α-chains, and it is the composition of these chains that determine the collagen type. Currently, at least 28 different types of collagen are known [15]. The predominant type of collagen is the type 1-collagen, which amongst others is found in skin, tendon, bone and large vessels. It consists of two identical $\alpha1$(I)-chains and one $\alpha2$(I)-chain with a different amino acid composition, or in some rare cases three $\alpha1$(I)-chains [7]. Because it is the most common type, it is the type mostly used for research, and the one used here.

Collagen is insoluble in organic solvents, and only a few percent of the total collagen is soluble in water [6]. This is why the implants need to be inserted into the body somewhere collagen naturally occurs. As we have seen, this includes a lot of different tissues, which makes collagen well suited for these kinds of implants.

1.1.1 Cross-linking

Cross-links in collagen are links between the α-chains, which makes the molecule more stable and difficult to degrade. These occur naturally both intra- and intermolecularly and are assembled within the non-helical areas of the molecules. They can dwindle away by acidic reactions, but new cross-links can be introduced in different ways [6]. Cross-linking a collagen matrix has the purpose of slowing the degradation, so the matrix will not crumble as fast. Because collagen-implants are used for controlled drug release, the ability to slow the process down can be useful. Changing the molecular structure of the collagen does, however, mean that there needs to be done new experiments, and the mathematical model needs to be altered, as this is very new with regards to drug release.

There are two ways to cross-link a collagen molecule; chemically or enzymatically. Chemical cross-linking will be badly solvant in water, and will therefore need to be solved in something else to react. This raises toxicological concerns, and both the chemicals and their solvents will need to be removed from the body [7]. Enzymatic cross-linking have very few toxicological concerns, because the cross-linking products are activated by enzymes that will be washed away before the implants are inserted. The cross-linked collagen used for the experiments in this thesis has been cross-linked by a

group of enzymes called tyrosinases. These catalyse the oxidation of tyrosine residues to dopaquinones, which makes the matrix stronger. The tyrosinases exist in higher plants, animals and microorganisms and were recognised for their ability to cross-link proteins, and specifically collagen, already in the 1960's. This work was recently confirmed and specified with data obtained from modern techniques, [6], which makes it a good choice for our enzymatical cross-linking.

1.2   The device

The collagen implant, often called a minirod, is made by homogenizing collagen and the drug we want to trap. Higher weight drugs, such as proteins or polysaccharides, are commonly used [10]. This mix is made into the minirod, formed as a cylinder, which is then inserted into the body. There are many uses for controlled drug delivery, ranging from treatment of cancer and diabetes to contraception and vaccination [10].

The challenges arise when we want to optimize the drug release. There are many factors that influence the process, both the shape of the minirod, whether the collagen is cross-linked or not and how much drug it contains. From a mathematical point of view, we want to model how quickly the collagen is degraded, and how fast the drug is released. However, due to concurring processes, this is difficult. There are also a lot of parameters, and although some can be determined experimentally, some will have to be fitted. The goal is to have as few parameters as possible fitted, and we hope that our new model will have fewer parameters that need fitting than the model from [14].

1.3   Experiments

During my stay with the Ludwig-Maximilians University of Munich, we did new experiments on the minirods. These were performed both with non-cross-linked collagen and cross-linked collagen. We did some short-term measuring of the collagen degradation that was used to help determine a new set of reaction rate constants. After I left, Madeleine Witting continued experimenting, and in combination with the experiments we performed together was able to give me a set of new parameters to use for my simula-

tions. There are still some experimenting to do, as we did no experiments on the release of drug, only the collagen degradation. However, earlier experiments suggest that the drug release as well as the collagen degradation is slowed down when the collagen is cross-linked.

These systems are very promising and their optimization is the subject of current research. As applied mathematicians we try to contribute through mathematical modelling and numerical simulation to the understanding and eventually the optimization of these systems.

Chapter 2

Mathematical model

This chapter will provide the mathematical background for the model, as well as explain the mathematical equations we will use to model the controlled drug release. The background consists of the equations for flow in a porous media on a small scale and the model for the chemical reactions. As this thesis uses two different mathematical models, they will be explained separately at the end of the chapter and then be compared to each other.

2.1 Flow in porous media

A porous medium is a material that contains pores, also known as voids. At least some of the pores are connected, which allows a fluid to flow through them. The skeletal part is often called a matrix, but we will avoid using that term in this thesis. This is due to the way the word matrix is used in controlled drug release research. The drug deliver matrix is the whole system, both the skeletal part and the fluid. However, since the collagen skeleton is a porous medium, we can use the equations associated with this. In our case, we will use mass conservation and Fick's law to find the diffusion and transport equations.

The equations for a single-phase flow in a porous media is based on the principle that the flow is conserved. In this section we will assume that the fluid, either liquid or gas, is in one homogeneous phase. This allows us to consider the fluid as only one component.

2.1.1 Mass conservation

The main principle for modeling flow and transport in porous media, and thus also in the minirod, is the principle of mass conservation. It comes from the conservation laws, which states that a physical quantity, in our case mass, is preserved in a closed system. That is, the change of mass inside a volume has to be equal to the amount of mass that enters or leaves the volume. We will derive the mass conservation equation based on [1].

For the conservation equation we will need four properties. The first is the porosity, ϕ, of the solid phase, which is given by

$$\phi = \frac{\text{pore volume}}{\text{total volume}}.$$ (2.1)

For the fluid there are three properties to be considered, density, ρ, volume flux, $\mathbf{u}$, and source density, Q. The density is given by

$$\rho = \frac{\text{mass}}{\text{volume}} = [\text{kg/m}^3].$$ (2.2)

A flux measures how much of something that flows through an area during a certain time interval. That is, a flux is any quantity that is defined on a "per area" and "per time" basis. The volume flux is therefore

$$\mathbf{u} = \frac{\text{volume}}{\text{time} \cdot \text{area}} = [\text{m}^3/(\text{s} \cdot \text{m}^2) = \text{m/s}],$$ (2.3)

which is the volume that flows through an area per time interval. The last quantity to be defined is the source density which is mass per volume per second

$$Q = \frac{\text{mass}}{\text{time} \cdot \text{volume}} = [\text{kg/(s} \cdot \text{m}^3)].$$ (2.4)

To derive the mass conservation equation, we start with an arbitrary domain as shown in fig. 2.1. Inside this volume we can have different substances, such as collagen, drug or enzyme. The concentration, or volume density, of a matter is given by its density times the porosity of the solid. Then

$$\int_\Omega \phi\rho \, dV,$$ (2.5)

gives the mass inside Ω. Furthermore,

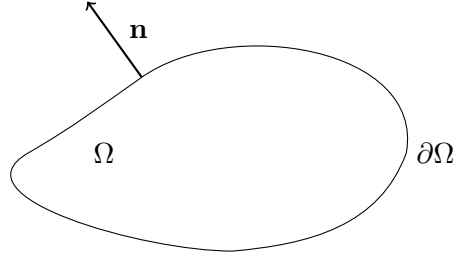$$\frac{\partial}{\partial t} \int_\Omega \phi\rho \, dV,$$ (2.6)

Figure 2.1: A domain Ω with boundary $\partial\Omega$ and outward unit normal $\mathbf{n}$.

gives the change in mass over time. For this derivative to be something else than zero there has to be a flux in to or out of the domain Ω, or a source or sink inside it.

The net flux of the species over the boundary $\partial\Omega$ is given by

$$\int_{\partial\Omega} (\rho\mathbf{u}) \cdot \mathbf{n}\, dS. \tag{2.7}$$

By letting Q denote the inner source density, the total production (or destruction) is given by

$$\int_{\Omega} Q\, dV. \tag{2.8}$$

From the mass conservation principle, the integral form of the mass conservation equation becomes

$$\frac{\partial}{\partial t} \int_{\Omega} \phi\rho\, dV + \int_{\partial\Omega} (\rho\mathbf{u}) \cdot \mathbf{n}\, dS = \int_{\Omega} Q\, dV. \tag{2.9}$$

By using the divergence theorem [2], the second term can be changed to

$$\int_{\partial\Omega} (\rho\mathbf{u}) \cdot \mathbf{n}\, dV = \int_{\Omega} \nabla \cdot (\rho\mathbf{u})\, dV, \tag{2.10}$$

where $\nabla \cdot (\rho\mathbf{v})$ is the divergence of the flux density $\rho\mathbf{v}$. In the first term, we can differentiate through the integral, that is taking the derivative inside the integral [2], to get

$$\frac{\partial}{\partial t} \int_{\Omega} \phi\rho\, dV = \int_{\Omega} \frac{\partial}{\partial t}(\phi\rho)\, dV. \tag{2.11}$$

Equation (2.9) can then be rewritten as

$$\int_{\Omega} \left(\frac{\partial}{\partial t}(\phi\rho) + \nabla \cdot (\rho\mathbf{u}) - Q \right) dV = 0. \tag{2.12}$$

The domain Ω is arbitrary, that is eq. (2.12) has to hold for all possible choices of domain. Thus, it can not be Ω that is making the integral 0, and the integrand can be removed such that the equation becomes

$$\frac{\partial}{\partial t}(\phi\rho) + \nabla \cdot (\rho\mathbf{u}) = Q. \tag{2.13}$$

This is the general form of the mass conservation equation.

2.1.2   Diffusion equations

We will also have mass conservation equations for species in the different phases in the pores. In this section we will find the diffusion equations based on [3, 16]. Letting $u = u(x, t)$ define their concentration at position x and time t, their mass conservation equation can be given as

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{J} = Q, \tag{2.14}$$

where $\mathbf{J} = \mathbf{J}(x, t)$ denotes the flux over the boundary of the domain. This flow can either be due to diffusion, transport, or a mixture of both.

Diffusion is the spreading due to molecular movements. On a molecular level, this is a continuous process. Physically, it can be considered a result of the random walk of the diffusing particles. It can be illustrated by a cup of tea. Initially all the colour and flavour of the tea exist inside the tea bag. However, as the particles diffuse in the water, they will be randomly and uniformly distributed in the cup.

On a higher level than the molecular, the diffusion happens from areas of high concentration to areas of lower concentration. Fick's first law states that the diffusion is proportional to the negative gradient of concentration. In equation form, this can be written as

$$\mathbf{J} = -D\nabla u, \tag{2.15}$$

with D as diffusion coefficient.

If the flux only happens due to diffusion, eq. (2.14) becomes

$$\frac{\partial u}{\partial t} - \nabla \cdot (D\nabla u) = Q. \tag{2.16}$$

This equation is called the diffusion equation for concentrations. For the homogeneous case, that is $Q = 0$, the equation is called Fick's second law.

2.2   Chemical reactions

Our model is based around the interaction between the collagen and colla-
genase, which is a chemical reaction. These reactions will provide a sink or
source term, depending on whether the species is produced or bound in the
reaction. Thus, they have to be put into the mass conservation equation in
our model.

A chemical reaction can be written as a chemical equation by

$$\text{reactants} \longrightarrow \text{products.} \tag{2.17}$$
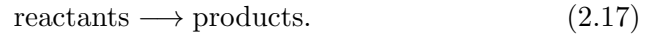
In our case, there are two processes that need to be modelled. The first
is the binding of the enzyme collagenase to the substrate collagen, which
creates the enzyme-substrate complex. The collagenase will then destroy
the collagen molecule, resulting in the separation of the enzyme and a rest
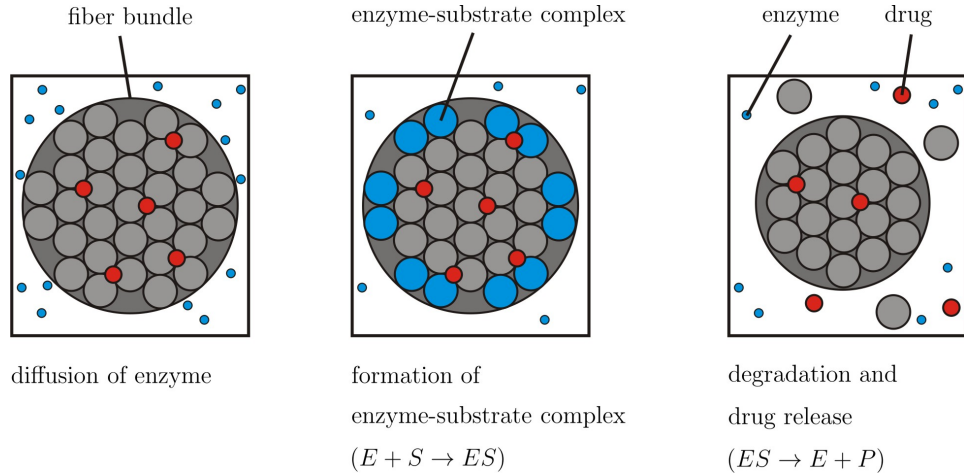product of hydrolized collagen. This is illustrated in fig. 2.2.

Figure 2.2: The degradation of the collagen matrix [13].

The reaction scheme used to describe the degradation of the matrix is the
common reaction scheme for an enzymatically catalyzed degradation of a
substrate. This was first proposed by Michaelis and Menten in 1913 [11],
and consists of the equations

$$E + S \xrightarrow{k_1} ES \tag{2.18}$$

and

$$ES \xrightarrow{k_2} E + P. \tag{2.19}$$

Here E is the enzyme, S the substrate, ES the enzyme-substrate complex and P the product. In our case of drug delivery systems, the enzyme is collagenase and the substrate is collagen. They form the enzyme-substrate complex, which in turn results in the product hydrolized collagen. The rate parameters k_1 and k_2 describe the absorption of enzyme to the active binding site of the collagen fibre and the cleavage process respectively [13]. We see that we have four unknowns, and the processes also result in four differential equations.

2.3 The mathematical models

There are two mathematical models used in this thesis. One, the model based on grain geometry, is the same as in [14], and will not be derived here, only explained. In [14], the model is two-dimensional, so what we will refer to as the grain model is a one-dimensional version of this. The second one has a channel geometry and will be derived as it is new in this thesis. Both models only describe the degradation of the matrix, that is, a fully swollen matrix and a homogeneous degradation is assumed.

2.3.1 A mathematical model for drug release and collagen degradation in a grain geometry

Figure 2.3 shows the domain on which the equations are valid. The circles represent bundles of collagen fibers with trapped drug as in fig. 2.2, and between them is the fluid that contains the enzymes. On the boundary between the collagen and the fluid, collagenase will attach itself to the collagen to break it down, and then form an enzyme-substrate complex. This species will only exist on this boundary. The model is based on four main equations. Two partial differential equations (PDE's) that are transport equations for the macroscopical concentrations, one ordinary differential equation (ODE) that describes the concentration of enzyme substrate and the surface concentration of the substrate, and one ODE that describes the geometry of the implant.

There are two transport equations for macroscopic concentrations - one for the enzyme concentration, C_E, and one for the concentration of the mo-
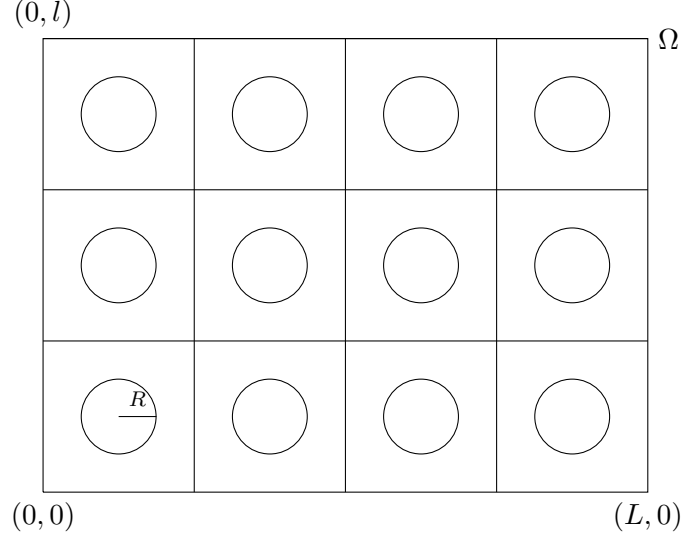
$(0, l)$

Ω

$(0, 0)$ $(L, 0)$

Figure 2.3: Schematic representation of the domain.

bile drug, C_A. Both are upscaled diffusion equations, with C_E described by

$$\partial_t(\theta(R)C_E) - \nabla_x \cdot (D_E^0 \bar{D}(R)\nabla_x C_E)$$
$$+ 2\pi R(k_1(N - C_{ES})C_E - k_2 C_{ES}) + \theta(R)k_{act}C_E = 0 \tag{2.20}$$

and C_A by

$$\partial_t(\theta(R)C_A) - \nabla_x \cdot (D_A^0 \bar{D}(R)\nabla_x C_A) - 2\pi R \frac{C_{A,im}^0}{C_C^0} k(C_{ES}R)^\gamma = 0. \tag{2.21}$$

The equations are similar in form, as the first two terms describe the diffusion process and the rest describes the reaction. They both have a diffusion coefficient, D_E^0 and D_A^0 respectively. The function $\bar{D}(R)$ is assumed to be the linear relation $\bar{D}(R) = \frac{1-2R}{1-2R^0}$, where R is the radius of the collagen fibre and R^0 is the initial radius. One more concentration is used, and that is the enzyme-substrate complex, C_{ES}. The constant N is a representation of C_{ES}^{max}, which means that $(N - C_{ES})$ represents how many places that are available for collagenase to attach. The porosity is $\theta(R) = 1 - \pi R^2$. All the k's are constant rate parameters, with k_1 and k_2 as in the Michaelis-Menten model. Finally, $C_{A,im}^0$ and C_C^0 are initial values of the concentration of immobile drug and collagen, respectively.

In eq. (2.20), the term $\theta(R)k_{act}C_E$ describes the death of the enzyme. The final term in this equation describes the process of enzymes attaching to the collagen, and the process where the collagen is broken down, such that they are released again. They are multiplied with the length of the boundary, $2\pi R$. The final term in eq. (2.21) has a fraction, $C_{A,im^0}/C_C^0$, which is the same as ρ_C/ρ_A, with ρ_C as the density of collagen and ρ_A as the drug density. Again $2\pi R$ describes the length of the border. The final part of the reaction, $k(C_{ES}R)^\gamma$ is a term also found in the description on how R changes.

The ODE for the enzyme-substrate complex and the surface concentration of the substrate is

$$\partial_t C_{ES} = k_1(N - C_{ES})C_E - k_2 C_{ES}. \tag{2.22}$$

Simply said, the equation tells us that the change in enzyme-substrate complex concentration is equal to how many enzymes that attaches to collagen, and subtracted by how many enzymes that are released again. We see that this equals the term from eq. (2.20) exactly, except that it is multiplied with the length of the boundary in eq. (2.20), and the sign is different. This is natural, because the subtraction in eq. (2.22) is the amount of enzymes becoming enzyme-substrate complexes minus the amount going back, and should be subtracted from the total amount of enzymes available.

The geometry of the substrate is described by

$$\partial_t R = -\frac{1 - \theta(R^0)}{C_C^0}k_2(C_{ES}R)^\gamma. \tag{2.23}$$

The fraction is the same as $\frac{1}{\rho_C}$, and tells us that the change in radius is inversely proportional to the density of the collagen.

2.3.2 A new mathematical model for drug release and collagen degradation in a channel geometry

To make a new model can be interesting, as it gives us an opportunity to compare two models. In turn, this could give more insight to the processes that are modelled. Our new model is based on channel geometry. Its upscaled version will be easier to compute, as it will be 1D. This geometry also makes it easier to look into the moving boundary, and it will be interesting to compare the equation sets.
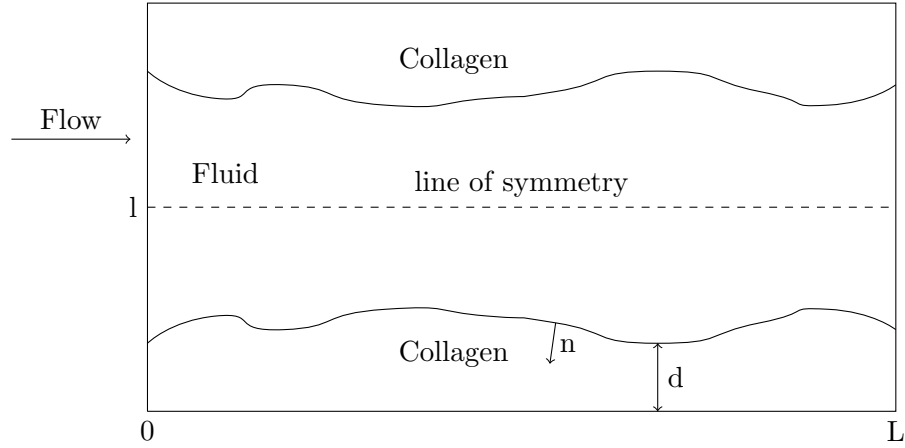
Figure 2.4: Schematic representation of the media.

Figure 2.4 is a schematic representation of the model at microscale, with collagen at the top and the bottom, and a fluid containing collagenase between them. Assuming symmetry along the dashed line allows us to only model half, and we will derive the model from the bottom half, which will then look as in fig. 2.5
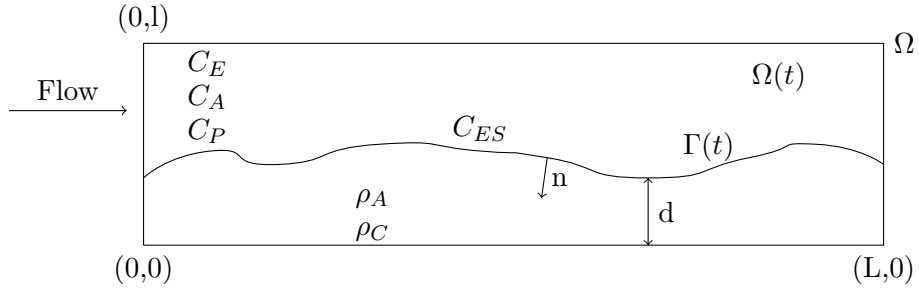
Figure 2.5: Schematic representation of the model.

This new domain is called Ω and defined as $\Omega = \{(x,y)|x \in (0,L), y \in (0,l)\}$. The part of the domain that contains the fluid with enzymes is called $\Omega(t)$ with $\Omega(t) = \{(x,y)|x \in (0,L), d(x(t)) < y < l)\}$, where d is the thickness of the collagen at point x at time t. All boundaries are denoted by Γ. The top boundary as $\Gamma_{sym} = \{(x,l)|x \in (0,L)\}$, the bottom boundary as $\Gamma_0 = \{(x,0)|x \in (0,L)\}$, the right one by $\Gamma_R = \{(0,y)|y \in (0,l)\}$ and left by $\Gamma_L = \{(L,y)|y \in (0,l)\}$. The boundary between the fluid and the col-

lagen is a moving boundary that depends on time, and it is defined by $\Gamma(t) = \{(x, d(x,t))|x \in (0, L)\}$.

In $\Omega(t)$ there are two species that need to be modelled, and we will model their concentrations. The first is the enzyme collagenase, denoted by C_E, and the second is the drug that is released, C_A. In the fluid there is also hydrolized collagen C_P, but we will assume that this is unnecessary to model. Sufficient assumptions for this is that the drug is evenly distributed both in the collagen and afterwards as released into the fluid. That is, they exist in fixed proportions. We also assume that the diffusion coefficients D_A and D_P are similar enough that it is enough to only model C_A. As C_P does not influence the processes, it is only modelled to see how much collagenase is released back into the fluid. However, the proportionality with C_A makes it possible to model the change in collagenase from C_A. The amount of drug still trapped within the collagen can be described by d and ρ_A, as ρ_A is the density of the drug. Likewise, by using d and ρ_C, with ρ_C as the density of collagen, it is possible to model the amount of collagen left. However, we will not use this in our final model. There is one more species to model; the enzyme-substrate complex C_{ES}. This complex will only exist where the collagenase can meet the collagen, that is on $\Gamma(t)$. It will result in the reactions described earlier

$$
\begin{aligned}
E + S &\xrightarrow{k_1} ES, \\
ES &\xrightarrow{k_2} E + P.
\end{aligned}
\tag{2.24}
$$

For deriving the mathematical model, we start by writing the equations on a dimensional microscale. We start by using mass balance in combination with Fick's law of diffusion for the species in $\Omega(t)$. This will in turn result in three diffusion equations, which are

$$
\partial_t C_E = D_E \Delta C_E - k_{act} C_E, \tag{2.25}
$$

$$
\partial_t C_A = D_A \Delta C_A, \tag{2.26}
$$

$$
\partial_t C_P = C_P \Delta C_P, \tag{2.27}
$$

where $D_i, i = A, E, P$ is the diffusion coefficient for each of the three species. The term $k_{act} C_E$, with k_{act} as a constant, describes the death of the enzymes.

On $\Gamma(t)$ we have a mass balance equation for C_{ES}. This describes the process from eq. (2.24), where the change in C_{ES} over time depends both

on how much collagenase that attaches to the collagen, as well as how much
of the collagen that is broken down, and thereby released into the fluid. The
equation becomes

$$\partial_t C_{ES} = H(\rho d)[k_1 C_E (N - C_{ES}) - k_2 C_{ES}] + (1 - H(\rho d))[-k_3 C_{ES}]. \quad (2.28)$$

The function $H(\rho d)$ is the Heaviside function, which is defined as

$$H(u) = \left\{ \begin{array}{ll} 1, & u > 0 \\ 0, & u \leq 0. \end{array} \right. \quad (2.29)$$

There are again constants, k_1, k_2 and k_3 that are rate parameters, and there
is a constant N which describes the number of places available for the col-
lagenase to attach to the collagen. The term $k_1 C_E (N - C_{ES})$ describes
the process where collagenase attaches to the collagen, whereas the terms
$-k_i C_{ES}$, $i = 2, 3$ describes the release of collagenase, hydrolized collagen,
and thereby drugs, back into the fluid. We see that as long as there is col-
lagen left, the Heaviside function is 1, that is, both processes are present.
When there is no more collagen left under the collagen-enzyme substrate,
the Heaviside function will be 0. This assures that the binding process
stops, so nothing will happen when the remaining substrate is released. We
do not know whether the remaining collagen is influencing the cleavage pro-
cess, which is the reason for the possibly different rate parameters k_2 and
k_3.

We can also describe the speed of the boundary in the normal direction.
This requires the normal vector into the collagen which will be

$$\mathbf{n} = (\partial_x d(x, t), -1) \cdot \frac{1}{\sqrt{1 + (\partial_x d)^2}}. \quad (2.30)$$

The speed with which the boundary moves is given by

$$\mathbf{v} = (x'(t), y'(t)) = (x'(t), \partial_t d + x' \partial_x d). \quad (2.31)$$

These two equations can be used to find that the speed of the boundary in
the normal direction is

$$v_n = \mathbf{v} \cdot \mathbf{n} = \frac{1}{\sqrt{1 + (\partial_x d)^2}} (x' \partial_x d - x' \partial_x d - \partial_t d) = -\partial_t d \cdot \frac{1}{\sqrt{1 + (\partial_x d)^2}}. \quad (2.32)$$

Using this we can find an equation for how the boundary moves,

$$v_n \rho = H(\rho d) k_4 C_E [N - C_{ES}]. \quad (2.33)$$

The Heaviside function is used again to make sure that the boundary stops moving as it reaches the bottom of the domain, and k_4 is just another rate parameter.

Mass balance gives us the final three equations

$$-D_E \nabla C_E \cdot \mathbf{n} = \partial_t C_{ES} + v_n C_E, \tag{2.34}$$

$$-D_A \nabla C_A \cdot \mathbf{n} = v_n(C_A - \rho_A), \tag{2.35}$$

$$-D_P \nabla C_P \cdot \mathbf{n} = v_n(C_P - \rho_P). \tag{2.36}$$

This introduces ρ_A and ρ_P as the mass density of the drug and the collagen respectively. Equation (2.34) shows that the speed of the moving boundary gives the rate of the degradation of collagen. This is proportional to the available enzyme, C_E and the number of available places for it to attach, $(N - C_{ES})$. Boundary and initial conditions completes the model.

Adimensionalization

So far, all quantities have had dimensions. We will now make the model dimensionless, and we will then denote the quantities with dimensions as $\hat{\cdot}$. For example $\hat{C}_E$ will be the concentration of collagenase with dimensions, whereas C_E is dimensionless. We will also introduce a new quantity, ε, which is defined as $\varepsilon = \hat{l}/\hat{L}$. This is a dimensionless constant, as $\hat{l}$ and $\hat{L}$ have the same dimension. By assuming that $\hat{L} >> \hat{l}$, it is clear that ε is very small. This will be used later, in the upscaling process.

We will now define the dimensionless quantities. We start by defining the variables x and y. By defining x as $x = \hat{x}/\hat{L}$ and y as $y = \hat{y}/\hat{l}$, Ω will be the area with $0 < x < 1$ and $0 < y < 1$. This also allows us to find their dimensionless derivatives

$$\frac{\partial}{\partial \hat{x}} = \frac{\partial}{\partial x} \cdot \frac{1}{\hat{L}} \quad \text{and} \quad \frac{\partial}{\partial \hat{y}} = \frac{\partial}{\partial y} \cdot \frac{1}{\hat{l}}. \tag{2.37}$$

Because $\hat{d}$ measures length in the y-direction, we define that the same way as y, which gives us $d = \hat{d}/\hat{l}$. The new variable to measure time will be t, defined as $t = \hat{t}/\hat{T}_R$, where $\hat{T}_R$ is a reference time. We can then find the derivative with respect to time

$$\frac{\partial}{\partial \hat{t}} = \frac{\partial}{\partial t} \cdot \frac{1}{\hat{T}_R}. \tag{2.38}$$

The diffusion coefficients will be divided by a reference diffusion coefficient defined as $\hat{D}_R = \hat{L}^2/\hat{T}_R$, giving $D_E = \hat{D}_E/\hat{D}_R$ and $D_A = \hat{D}_A/\hat{D}_R$. These are both assumed to be of order $O(1)$.

Finally, we need to make the concentrations adimensional. This will be done by dividing each of them with a reference concentration C_i^R, $i =$ E, A, ES. This will define them as $C_E = \hat{C}_E/\hat{C}_E^R$, $C_A = \hat{C}_A/\hat{C}_A^R$ and $C_{ES} = \hat{C}_{ES}/\hat{C}_{ES}^R$. Finally, ρ is defined such that $\rho = \hat{\rho}_A/\hat{C}_A^R$. The necessary constants will be defined later.

The first equation we will adimensionalize is eq. (2.25) which will become

$$\frac{1}{\hat{T}_R}\hat{C}_E^R\frac{\partial C_E}{\partial t} = D_E\hat{D}_R\hat{C}_E^R\left(\frac{1}{\hat{L}^2}\frac{\partial^2 C_E}{\partial x^2} + \frac{1}{\hat{l}^2}\frac{\partial C_E}{\partial x^2}\right) - \hat{k}_{act}\hat{C}_E^R C_E. \qquad (2.39)$$

Taking $1/\hat{L}^2$ out of the parenthesis and multiplying with $\hat{T}_R$ yields

$$\frac{\partial C_E}{\partial t} = \frac{\hat{D}_R\hat{T}_R}{\hat{L}^2}D_E\left(\frac{\partial^2 C_E}{\partial x^2} + \frac{1}{\hat{\varepsilon}^2}\frac{\partial C_E}{\partial x^2}\right), \qquad (2.40)$$

where

$$\frac{\hat{D}_R\hat{T}_R}{\hat{L}^2} = 1. \qquad (2.41)$$

The equations for C_A and C_{ES} will be found similarly.

To make eq. (2.33) dimensionless we first need to make eq. (2.32) dimensionless. To do this, we start with finding $\partial_{\hat{t}}\hat{d}$ which will be

$$\frac{\partial\hat{d}}{\partial\hat{t}} = \frac{1}{\hat{T}_R}\frac{\partial(\hat{l}d)}{\partial t}. \qquad (2.42)$$

This gives us

$$\hat{v}_n = -\frac{\hat{l}}{\hat{T}_R}\partial_t d\frac{1}{\sqrt{1 + \varepsilon^2(\partial_x d)^2}}, \qquad (2.43)$$

as an expression for $\hat{v}_n$.

This can be substituted into eq. (2.33) to find

$$\rho\hat{C}_A^R\frac{\hat{l}}{\hat{T}_R}\frac{1}{\sqrt{1 + \varepsilon^2(\partial_x d)^2}}\partial_t d = H(\hat{\rho}\hat{l}d)\hat{k}_4\hat{C}_E^R\hat{C}_{ES}^R C_E\left(\frac{\hat{N}}{\hat{C}_{ES}^R} - C_{ES}\right), \qquad (2.44)$$

which is the same as

$$\rho \partial_t d = \sqrt{1 + \varepsilon^2 (\partial_x d)^2} \frac{\hat{T}_R \hat{k}_4 \hat{C}_E^R \hat{C}_{ES}^R}{\hat{l} \hat{C}_A^R} H(\hat{\rho} \hat{l} d) C_E \left(\frac{\hat{N}}{\hat{C}_{ES}^R} - C_{ES} \right). \qquad (2.45)$$

Finally, we want to adimsionalize eqs. (2.34) and (2.35). We start with eq. (2.35) which becomes

$$\frac{1}{\sqrt{1 + \varepsilon^2 (\partial_x d)^2}} (\varepsilon \partial_x d, -1) \frac{\hat{D}_R}{\hat{L}} D_A \hat{C}_A^R \left(\frac{\partial C_A}{\partial x}, \frac{1}{\varepsilon} \frac{\partial C_A}{\partial y} \right)$$
$$= \frac{\hat{l}}{\hat{T}_R} \frac{1}{\sqrt{1 + \varepsilon^2 (\partial_x d)^2}} \partial_t d \hat{C}_A^R (C_A - \rho). \qquad (2.46)$$

This can be rewritten as

$$-D_A \left(\varepsilon \partial_x d \partial_x C_A - \frac{1}{\varepsilon} \partial_y C_A \right) = -\frac{\hat{L}^2}{\hat{D}_R \hat{T}_R} \frac{\hat{l}}{\hat{L}} \partial_t d (C_A - \rho). \qquad (2.47)$$

Finally, eq. (2.34) will become

$$-\frac{1}{\sqrt{1 + \varepsilon^2 (\partial_x d)^2}} (\varepsilon \partial_x d, -1) \frac{\hat{D}_R}{\hat{L}} \hat{C}_E^R D_E \left(\frac{\partial C_E}{\partial x}, \frac{1}{\varepsilon} \frac{\partial C_E}{\partial y} \right)$$
$$= \frac{1}{\hat{T}_R} \hat{C}_{ES}^R \partial_t C_{ES} - \frac{\hat{l}}{\hat{T}_R} \frac{1}{\sqrt{1 + \varepsilon^2 (\partial_x d)^2}} \partial_t d \hat{C}_E^R C_E, \qquad (2.48)$$

which is the same as

$$-D_E \left(\varepsilon \partial_x d \partial_x C_E - \frac{1}{\varepsilon} \partial_y C_E \right)$$
$$= \sqrt{1 + \varepsilon^2 (\partial_x d)^2} \frac{\hat{L}}{\hat{D}_R \hat{T}_R} \frac{\hat{C}_{ES}^R}{\hat{C}_E^R} \partial_t C_{ES} - \frac{\hat{l} \hat{L}}{\hat{T}_R \hat{D}_R} \partial_t d C_E. \qquad (2.49)$$

Because C_{ES} only exists on $\Gamma(t)$ and C_E exists in $\Omega(t)$ the fraction $\hat{C}_{ES}^R / \hat{C}_E^R$ is of order $\hat{l}$. This can be used to find that

$$\frac{\hat{L}}{\hat{D}_R \hat{T}_R} \frac{\hat{C}_{ES}^R}{\hat{C}_E^R} = \frac{\hat{L}^2}{\hat{D}_R \hat{T}_R} \frac{k_5 \hat{l}}{\hat{L}} = k_5 \varepsilon, \qquad (2.50)$$

with k_5 such that $\hat{C}_{ES}^R = k_5 \hat{l} \hat{C}_E^R$.

Equations (2.40), (2.45), (2.47) and (2.49) with the adimensionalized equations for C_A and C_{ES} result in an adimensionalized system of six equations

$$\partial_t C_E = D_E \left(\partial_{xx} C_E + \varepsilon^{-2} \partial_{yy} C_E \right) - k_0 C_E, \tag{2.51}$$

$$\partial_t C_A = D_A \left(\partial_{xx} C_A + \varepsilon^{-2} \partial_{yy} C_A \right), \tag{2.52}$$

$$\partial_t C_{ES} = H(\rho \hat{l} d)[k_1 C_E(\eta - C_{ES}) - k_2 C_{ES}] - (1 - H(\rho \hat{l} d))k_3 C_{ES}, \tag{2.53}$$

$$-\partial_t d = \frac{k_4}{\rho} H(\rho \hat{l} d) \sqrt{1 + \varepsilon^2 (\partial_t d)^2} \, C_E(\eta - C_{ES}) \tag{2.54}$$

$$-D_E \left(\varepsilon \partial_x d \partial_x C_E - \frac{1}{\varepsilon} \partial_y C_E \right) = -\varepsilon \partial_t d C_E + \varepsilon k_5 \sqrt{1 + \varepsilon^2 (\partial_t d)^2} \, \partial_t C_{ES}, \tag{2.55}$$

$$-D_A \left(\varepsilon \partial_x d \partial_x C_A - \frac{1}{\varepsilon} \partial_y C_A \right) = \varepsilon \partial_t d(\rho - C_P). \tag{2.56}$$

The first two equations are valid in $\Omega(t)$, whereas the last four are valid on $\Gamma(t)$. We also assume that $\partial_y C_E = \partial_y C_A = 0$ on $\Gamma(t)$. Again, the system is completed with boundary conditions at Γ_R and Γ_L, and initial conditions.

The constant k_0 is defined as $\hat{k}_{act} \hat{T}_R$ and is assumed to be of order $O(1)$. Notice that this means that $k_0/\hat{k}_{act} = [s]$, which can be seen as an activity timescale assumed here to be of order $O(\hat{T}_R)$. Both k_2 and k_3 are defined similarly, with $k_2 = \hat{k}_2 \hat{T}_R$ and $k_3 = \hat{k}_3 \hat{T}_R$. The fractions $1/\hat{k}_2$ and $1/\hat{k}_3$ can be seen as time scales for the consumption of collagen if there is more collagen underneath $\Gamma(t)$ and if collagen is absent respectively. The final k-constant is k_1, defined as $k_1 = \hat{k}_1 \hat{T}_R \hat{C}_E^R$. There is one remaining constant, η, which is defined as $\eta = \hat{N}/\hat{C}_{ES}^R$. As $\hat{C}_{ES}$ is a reference value for the concentration, it is natural to assume that $\hat{C}_{ES}^R \leq \hat{C}_{ES}^{max}$. However, C_{ES}^{max} is represented by N, which gives us that $\eta \geq 1$.

Upscaling

From this porescale dimensionless model, we want to find an upscaled dimensionless model. An upscaled model allows us to compare our results with experimental results, as experiments can not be performed at the pore scale. The upscaling is done by assuming that all the variables admit a representation as an expansion series in ε. In the upscaling process, we would also like

to eliminate the y-dependence, which will be done by an integration with respect to y. Doing this, the two-dimensional model becomes one-dimensional, which is a huge advantage from a computational point of view. By assuming that $l << L$, this is done without losing too much accuracy. Expansions are made for C_E, C_A, C_{ES} and d making them

$$C_E = C_E^\varepsilon(x, y, t) = C_E^0(x, y, t) + \varepsilon C_E^1(x, y, t) + \varepsilon^2 C_E^2(x, y, t) + \cdots ,$$
$$(2.57)$$

$$C_A = C_A^\varepsilon(x, y, t) = C_A^0(x, y, t) + \varepsilon C_A^1(x, y, t) + \varepsilon^2 C_A^2(x, y, t) + \cdots , \quad (2.58)$$

$$C_{ES} = C_{ES}^\varepsilon(x, y, t) = C_{ES}^0(x, y, t) + \varepsilon C_{ES}^1(x, y, t) + \varepsilon^2 C_{ES}^2(x, y, t) + \cdots ,$$
$$(2.59)$$

$$d = d^\varepsilon(x, y, t) = d^0(x, y, t) + \varepsilon d^1(x, y, t) + \varepsilon^2 d^2(x, y, t) + \cdots . \quad (2.60)$$

We start the upscaling with C_E by substituting eq. (2.57) in eq. (2.51) which yields

$$\partial_t(C_E^0 + \varepsilon C_E^1 + \cdots) = D_E[\partial_{xx}(C_E^0 + \varepsilon C_E^1 + \cdots)$$
$$+ \frac{1}{\varepsilon^2}\partial_{yy}(C_E^0 + \varepsilon C_E^1 + \varepsilon^2 C_E^2 + \cdots)] \quad (2.61)$$
$$- k_0(C_E^0 + \varepsilon C_E^1).$$

This can be rearranged to

$$\varepsilon^0(\partial_t C_E^0) + \varepsilon^1(\partial_t C_E^1) + \cdots$$
$$= \varepsilon^{-2}(D_E\partial_{yy}C_E^0) + \varepsilon^{-1}(D_E\partial_{yy}C_E^1) + \varepsilon^0(D_E\partial_{xx}C_E^0 + D_E\partial_{yy}C_E^2 - k_0C_E^0)$$
$$+ \varepsilon^1(D_E\partial_{xx}C_E^1 + D_E\partial_{yy}C_E^3 - k_0C_E^1) + \cdots .$$
$$(2.62)$$

Because ε is a very small number, we will look at the terms with different powers of ε separately. The idea is that ε is so small that two terms with different powers of ε never will be equal, and that terms of order $\varepsilon^i, i = 1, 2, \cdots$ will be so small that they do not matter.

We start by looking at the terms including ε^{-2}. This gives us

$$D_E\partial_{yy}C_E^0(x, y, t) = 0, \ \ y \in (d^0, 1). \quad (2.63)$$

as the only term in eq. (2.62) that includes ε^{-2} is $\varepsilon^{-2}(D_E\partial_{yy}C_E^0)$ and we know that $\varepsilon \neq 0$. The diffusion coefficient D_E can be neglected as it is a nonzero constant, and thereby not the reason the expression is zero. The

limitation in the y-direction is due to C_E only existing in the fluid. The terms multiplied with ε^{-1} gives us

$$D_E \partial_{yy} C_E^1 = 0, \ \ y \in (d^0, 1), \tag{2.64}$$

with D_E again nonzero. Finally, we look at the terms of order $O(1)$. They result in a longer equation, namely

$$\partial_t C_E^0 = D_E[\partial_{xx} C_E^0 + \partial_{yy} C_E^2] - k_0 C_E^0, \ \ y \in (d^0, 1). \tag{2.65}$$

As $\partial_y C_E^\varepsilon = 0$ on Γ_{sym}, it follows that $\partial_y C_E^0 = \partial_y C_E^1 = \partial_y C_E^2 = 0$ also on Γ_{sym}. On $\Gamma(t)$ we can expand eq. (2.55) to

$$
\begin{aligned}
& - D_E[\varepsilon \partial_x (d^0 + \varepsilon d^1 + \cdots) \partial_x (C_E^0 + \varepsilon C_E^1 + \cdots) \\
& \quad - \frac{1}{\varepsilon} \partial_y (C_E^0 + \varepsilon C_E^1 + \varepsilon^2 C_E^2 + \cdots)] \\
& = - \varepsilon \partial_t (d^0 + \varepsilon d^1 + \cdots)(C_E^0 + \varepsilon C_E^1 + \cdots) \\
& \quad + \varepsilon k_5 \sqrt{1 + \varepsilon^2 (\partial_t d^0 + \varepsilon \partial_t d^1 + \cdots)^2} \ \partial_t (C_{ES}^0 + \varepsilon C_{ES}^1 + \cdots).
\end{aligned}
\tag{2.66}
$$

The square root will be of order $O(1)$, which gives us that the only terms of order less than ε are the terms from $-\frac{1}{\varepsilon} \partial_y (C_E^0 + \varepsilon C_E^1)$. This yields that $\partial_y C_E^0 = 0$ and $\partial_y C_E^1 = 0$ at $y = d^0$. For the terms of order $O(\varepsilon)$ we find the equation

$$-D_E[\partial_x d^0 \partial_x C_E^0 - \partial_y C_E^2] = -\partial_t d^0 C_E^0 + k_5 \partial_t C_{ES}^0, \tag{2.67}$$

where k_5 now also includes the square root. This can be rearranged to

$$D_E \partial_y C_E^2 = D_E \partial_x d^0 \partial_x C_E^0 - \partial_t d^0 C_E^0 + k_5 \partial_t C_{ES}^0, \tag{2.68}$$

which holds on $\Gamma(t)$.

From eq. (2.63) we know that $\partial_y C_E^0$ has to be constant everywhere on $y \in (d^0, 1)$, and since $\partial_y C_E^0 = 0$ on $y = 1$ and $y = d^0$ it is 0 everywhere. This gives us that C_E^0 is independent of y, and we can write $C_E^0 = C_E^0(x, t)$, which is the upscaled C_E. Similarly, by starting with eq. (2.64) and using the same arguments, $C_E^1 = C_E^1(x, t)$. The final variable we need to address is C_E^2, which will be eliminated by integrating eq. (2.65) in the y-direction, from $d^0(x, t)$ to 1. This yields

$$
\begin{aligned}
\partial_t C_E^0 (1 - d^0) = \ & D_E \partial_{xx} C_E^0 (1 - d^0) - k_0 C_E^0 (1 - d^0) + D_E \partial_y C_E^2 \big|_{d^0}^1 \\
= \ & D_E \partial_{xx} C_E^0 (1 - d^0) - k_0 C_E^0 (1 - d^0) \\
& + (0 - D_E \partial_x d^0 \partial_x C_E^0 + \partial_t d^0 C_E^0 - k_5 \partial_t C_{ES}^0).
\end{aligned}
\tag{2.69}
$$

The expression for $D_E \partial_y C_E^2 \big|_{d^0}$ comes from eq. (2.68). Rearranging the terms result in the first of four upscaled equations

$$\partial_t[(1-d^0)C_E^0 + k_5 C_{ES}^0] = D_E \partial_x[(1-d^0)\partial_x C_E^0] - k_0(1-d^0)C_E^0, \quad (2.70)$$

which is valid on $x \in (0, L)$, $t > 0$.

Now we proceed by deriving the upscaled equation for the drug. This is done by using the expansion eq. (2.58) on eq. (2.52) which gives us

$$
\begin{aligned}
&\partial_t(C_A^0 + \varepsilon C_A^1 + \cdots) \\
&= D_A \left[\partial_{xx}(C_A^0 + \varepsilon C_A^1 + \cdots) + \frac{1}{\varepsilon^2} \partial_{yy}(C_A^0 + \varepsilon C_A^1 + \varepsilon^2 C_A^2 + \cdots) \right],
\end{aligned}
\quad (2.71)
$$

and expanding eq. (2.56) by eqs. (2.58) and (2.60) to

$$
\begin{aligned}
&- D_A \left[\varepsilon \partial_x(d^0 + \varepsilon d^1 + \cdots)\partial_x(C_A^0 \varepsilon C_A^1 + \cdots) - \frac{1}{\varepsilon}\partial_y(C_A^0 + \varepsilon C_A^1 + \cdots) \right] \\
&= \varepsilon \partial_t(d^0 + \varepsilon d^1 + \cdots)[\rho - (C_A^0 + \varepsilon C_A^1 + \cdots)].
\end{aligned}
\quad (2.72)
$$

It can then be shown in a similar way as for C_E that the upscaled equation for C_A becomes

$$\partial_t[d^0 \rho + (1-d^0)C_A^0] = D_A \partial_x[(1-d^0)\partial_x C_A^0], \quad (2.73)$$

in $\Omega(t)$.

As C_{ES} only exists on $y = d^0$, it is clear that it does not depend on the variable y. Its upscaled equation can then be found by using the terms of order less than $O(\varepsilon)$. The expansion is

$$
\begin{aligned}
\partial_t(C_{ES}^0 + \varepsilon C_{ES}^1 + \cdots) \\
= H(\rho \hat{l} d^0)[k_1(C_E^0 + \varepsilon C_E^1 + \cdots)(\eta - [C_{ES}^0 + \varepsilon C_{ES}^1 + \cdots]) \\
- k_2(C_{ES}^0 + \varepsilon C_{ES}^1 + \cdots)] - (1 - H(\rho \hat{l} d^0))k_3(C_{ES}^0 + \varepsilon C_{ES}^1 + \cdots).
\end{aligned}
\quad (2.74)
$$

This gives us that the upscaled equation will be

$$\partial_t C_{ES}^0 = H(\rho \hat{l} d^0)[k_1 C_E^0(\eta - C_{ES}^0) - k_2 C_{ES}^0] - (1 - H(\rho \hat{l} d^0))k_3 C_{ES}^0 \quad (2.75)$$

on $\Gamma(t)$.

The final equation that needs upscaling is eq. (2.54). Its expansion is

$$-\partial_t(d^0 + \varepsilon d^1 + \cdots) = \frac{k_4}{\rho} H(\rho \hat{l} d^0)\sqrt{1 + \varepsilon^2(\partial_t(d^0 + \varepsilon d^1 + \cdots))^2}$$
$$\cdot (C_E^0 + \varepsilon C_E^1 + \cdots)(\eta - C_{ES}^0 + \varepsilon C_{ES}^1 + \cdots)). \tag{2.76}$$

The square root is again of order $O(1)$ and will be combined with k_4 to another constant, still called k_4. Because d is independent of the variable y, we can find the upscaled equation by choosing the terms the same way we did for C_{ES}. That is, our fourth upscaled equation will be

$$-\partial_t d^0 = \frac{k_4}{\rho} H(\rho \hat{l} d^0) C_E^0 (\eta - C_{ES}^0) \tag{2.77}$$

on $\Gamma(t)$. For simplicity, the zeros will not be written from now on.

The final step in developing this model, is going back to the dimensions again. This will be done by substituting back for all the variables and constants we used to make the porescale model dimensionless. We will then find an upscaled dimensional system of four equations, based on eqs. (2.70), (2.73), (2.75) and (2.77). By abuse of notation we drop the hats, making them

$$\frac{\partial}{\partial t}[(l - d)C_E + C_{ES}] = D_E \frac{\partial}{\partial x}[(l - d)\frac{\partial}{\partial x}C_E] - k_{act}(l - d)C_E, \tag{2.78}$$

$$\frac{\partial}{\partial t}[(l - d)C_A + d\rho] = D_A \frac{\partial}{\partial x}[(l - d)\frac{\partial}{\partial x}C_A], \tag{2.79}$$

$$\frac{\partial}{\partial t}C_{ES} = H(\rho d)[k_1 C_E(N - C_{ES}) - k_2 C_{ES}]$$
$$- (1 - H(\rho d))k_3 C_{ES}, \tag{2.80}$$

$$\frac{\partial}{\partial t}(\rho d) = -k_4 H(\rho d)C_E(N - C_{ES}). \tag{2.81}$$

The model is again completed by boundary and initial conditions.

2.4   Comparison of the two sets of equations

The two models give us two sets of equations that seem very different at first glance. However, as the geometry is different, this is expected. We would

therefore like to take a closer look at the equations to find their similarities and differences. We start with the equations for C_E. The grain geometry has the equation

$$\partial_t(\theta(R)C_E) - \nabla_x \cdot (D_E^0 \bar{D}(R)\nabla_x C_E) + 2\pi R(\partial_t C_{ES}) + \theta(R)k_{act}C_E = 0, \quad (2.82)$$

and the channel geometry

$$\partial_t((l-d)C_E) - D_E\partial_x[(l-d)\partial_x C_E] + \partial_t C_{ES} + (l-d)k_{act}C_E = 0. \quad (2.83)$$

Remembering that $\theta(R) = 1 - \pi R^2$, it is clear that the first term in both equations is the time derivative of porosity times concentration. The second term contains a constant diffusion coefficient and a term that depends on the geometry. Both equations depend on the change in enzyme-substrate complex, but in the grain geometry, the term is multiplied with the length of the border. This difference is due to the differences in geometry. The grain geometry is based on two dimensions, with one calculation for each square, that is, enzymes may attach around the whole cylinder of collagen. Calculations on the channel model is done in the x-direction only, which means that the calculations on each node is independent of the geometry. The final term is again the same, with porosity times a death rate parameter times the concentration.

We saw that the equations for C_E have a very similar structure. However, they both depend on C_{ES}, so it is natural to look at the equations for that before concluding that the equations for C_E are built the same way. For the grain model, the equation is

$$\partial_t C_{ES} = k_1(N - C_{ES})C_E - k_2 C_{ES}, \quad (2.84)$$

and for the channel model

$$\partial_t C_{ES} = H(\rho d)[k_1(N - C_{ES})C_E - k_2 C_{ES}] - (1 - H(\rho d))k_3 C_{ES}. \quad (2.85)$$

These seem quite different on the surface, but this is again due to geometric differences between the models. We see that as long as the Heaviside function is 1, they are both containing the terms $k_1(N - C_{ES})C_E - k_2 C_{ES}$. However, the new model uses the Heaviside function to stop the process as explained earlier. The grain model does not stop naturally, which needs to be taken into consideration when implementing it. This will be explained further in section 3.7

Both models also include an equation for the concentration of released drug. The equation in the grain model is

$$\partial_t(\theta(R)C_A) - \nabla_x \cdot (D_A^0 \bar{D}(R)\nabla_x C_A) - 2\pi R \frac{C_{A,im}^0}{C_C^0}k(C_{ES}R)^\gamma = 0, \quad (2.86)$$

and in the new model

$$\partial_t[(l-d)C_A] - D_A\partial_x[(l-d)\partial_x C_A] + \partial_t(d\rho) = 0. \quad (2.87)$$

From the comparison of the C_E-equations, we know that the differences in the first two terms can be explained by the different geometries. This leaves one term in each of the equations. For the grain model we know that $CA, im^0 = \rho_A/(1-\theta)$ and $C_C^0 = \rho_S/(1-\theta)$. The term $(C_{ES}R)^\gamma$ is proportional to $-\partial_t R$, and by adjusting k the term can be rewritten as

$$-2\pi R\frac{C_{A,im}^0}{C_C^0}k(C_{ES}R)^\gamma = 2\pi R\rho_A k_{adj}\partial_t R, \quad (2.88)$$

with k_{adj} as the adjusted proportionality constant. This constant is also adjusted to contain ρ_S, which is possible because all densities are assumed to be constant. Apart from the geometric part $2\pi R$, this expression is exactly the same as the term $\partial_t(d\rho)$ from the channel model, apart from the proportionality constant. However, from eq. (2.81) we know that $\partial_t(d\rho)$ contains a proportionality constant, making the terms equal when ignoring the geometric difference.

The equations for R and d respectively, are depending a lot on the geometry, but they still have some similarities. They are both inversely proportional to ρ_S and depend on C_{ES}. This is seen clearly by using $C_C^0 = \rho_S/(1-\theta)$ to rewrite eq. (2.23) as

$$\partial_t R = -\frac{k_2}{\rho_S}(C_{ES}R)^\gamma, \quad (2.89)$$

and rewriting eq. (2.81) to

$$\partial_t d = -\frac{k_4}{\rho_S}H(d\rho)C_E(N - C_{ES}). \quad (2.90)$$

It is easy to see that $\partial_t R = 0$ when $R = 0$, which means that the process stops when there is no more collagen. For the channel geometry, this is achieved by using the Heaviside function. That is, the only differences

between the equations are the terms depending on C_{ES}, which is to be expected from the geometric differences.

In this chapter we have derived a new mathematical model for controlled drug release from collagen minirods, which is based on diffusion and mass conservation. Comparisons shows that the new model is consistent to the model in [14]. The new model assumes a thin channel geometry, making the upscaled model one-dimensional. The two models will form the basis of the numerical simulations presented in the next chapters.

Chapter 3

Numerical modelling

We have developed a one-dimensional model, but as this has no known analytical solutions we need numerical approximations. For this, we will use MPFA for the spatial discretization and forward Euler for the temporal. We will start this chapter by looking at different grids, and what will be used in our modelling. Afterwards we will derive the finite difference approximation, which forms the basis for the Euler method. For the space discretization, there will be a short introduction to the control volume methods, and a derivation of MPFA in one dimension. We will also look at two different kinds of boundary conditions before showing how the equations are discretized. The equations are both ODE's and PDE's. Some of them are coupled, and for those we will also show the coupling. The last section explains more about how the implementations were done.

3.1 Grid

To make a numerical representation of a function, we need to know which point to use for the calculation. The same procedure applies for solving a mathematical problem numerically. This is done by defining points X_i where the calculations are done. Defining these points is the same as defining a grid on the function domain.

Fundamentally, a grid is made by laying out some points and defining a neighbourhood around each point. The neighbourhood must be chosen so that lines between pairs of points are not crossing. Each element surrounded

by these lines, is called a cell, and the lines are called cell walls. Then we have to decide whether we want a grid that is point distributed or cell centred. A point distributed grid is based on the corner points, whereas a cell centred grid is based on the points in the centre of the cells, see figs. 3.1 and 3.2 [1].
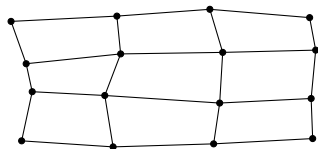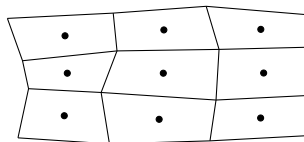

Figure 3.1: Point distributed grid.   Figure 3.2: Cell centered grid.

Note that for a cell centred grid, one must choose the grid first, and then use the midpoints generated - opposed to choosing the nodes first, and then constructing a grid around them.

In the one-dimensional case, all nodes will be on the same line. If the distance between the nodes is the same for all cells, we call it equidistant. An example is shown in fig. 3.3.
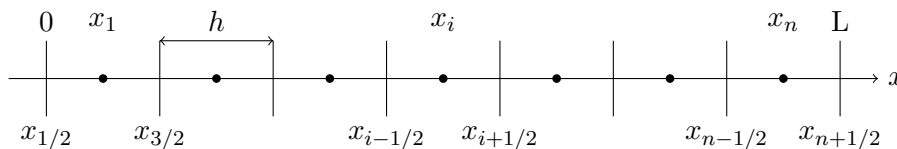
Figure 3.3: Cell centered equidistant grid in 1D.

Figure 3.3 shows a domain, the interval $[0, L]$, and its grid. As the grid is cell centred, we start by defining the cell walls, and then place the points at their midpoints.

3.2 Finite difference approximation

The finite difference approximation is based on Taylor series, and is the most elementary discretization method. It can be used in both space and time discretization, and to approximate derivatives of several orders. The finite difference methods are defined by the Taylor series for $u(x + h) = u(x_{i+1})$

and $u(x - h) = u(x_{i-1})$. The series for $u(x_{i+1})$ is

$$u(x_{i+1}) = u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i) + \frac{h^3}{3!}u'''(x_i) + \cdots . \qquad (3.1)$$

This gives us the forward difference approximation for the first derivative:

$$u'(x_i) = \frac{u(x_{i+1}) - u(x_i)}{h} + O(h). \qquad (3.2)$$

The term $O(h)$ gives the order of the method, which comes from the terms truncated from the series. In this case, they are of order h.

Similarly, from

$$u(x_{i-1}) = u(x_i) - hu'(x_i) + \frac{h^2}{2}u''(x_i) - \frac{h^3}{3!}u'''(x_i) + \cdots , \qquad (3.3)$$

we can find the backward difference approximation for the first derivative:

$$u'(x_i) = \frac{u(x_i) - u(x_{i-1})}{h} + O(h). \qquad (3.4)$$

To find the second order derivative we use

$$\frac{u(x_{i+1}) - u(x_i)}{h} = u'(x_i) + \frac{h}{2}u''(x_i) + \frac{h^2}{3!}u'''(x_i) + \frac{h^3}{4!}u'''(x_i) + \cdots \qquad (3.5)$$

and

$$\frac{u(x_i) - u(x_{i-1})}{h} = u'(x_i) - \frac{h}{2}u''(x_i) + \frac{h^2}{3!}u'''(x_i) - \frac{h^3}{4!}u'''(x_i) + \cdots . \qquad (3.6)$$

By subtracting eq. (3.6) from eq. (3.5) we get

$$u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} + O(h^2). \qquad (3.7)$$

This is the central difference formula for the second derivative $u''(x_i)$.

3.3 Temporal discretization

We start with an ordinary differential equation

$$u'(t) = f(t, u(t)). \qquad (3.8)$$

If we use eq. (3.2) to approximate $u'(t)$, we get the forward Euler scheme. The numerical method will then be

$$u^j = u^{j-1} + \tau f(t_{j-1}, u^{j-1}). \tag{3.9}$$

If we instead use eq. (3.4), we get the backward Euler scheme,

$$u^j = u^{j-1} + \tau f(t_j, u^j). \tag{3.10}$$

Here u^j denotes the approximated value of the exact solution at time t_j. The index j denotes the time steps for 1 to T, $j = 1, \ldots, T$ and $\tau = t_j - t_{j-1}$ is the size of the time step. For equidistant grid this will be Δt.

In this case, the lowest order term truncated is $\frac{h^2}{2} u''(x_i)$. The term is the local trunction error, which is the error made in each step, and of order h^2. The global trunction error is the cumulative effect of all the local truncation errors. Because $h = L/n$, it is easy to see that $n \sim 1/h$. This means that the global truncation error is proportional to h, and hence the method is of order h [8].

3.4   Discretization in space

The space discretization is based on the central difference formula for the second derivative, that is, eq. (3.7), because the only spatial derivatives in or models are second order. It includes the point where we want to make the calculation as well as the points on both sides. For the differential equation

$$u''(x) = f(x, u(x)), \tag{3.11}$$

the scheme will be

$$u(x_i) = \frac{1}{2} h^2 f(x_i, u(x_i)) + \frac{1}{2} u(x_{i+1}) + \frac{1}{2} u(x_{i-1}). \tag{3.12}$$

3.4.1   Control volume methods

The starting point for the control volume methods, is to define a grid over the domain of the differential equation. This divides the domain into cells, like we did in fig. 3.3 for the 1D case. These cells are called control volumes, hence the name of the method. On each of these cells, the mass conservation principle will be applied, which holds when the fluxes on each side of a cell

wall are equal. The integral form of the mass conservation equation is the applied to each cell. The flux is given across the cell walls, averaging the permeability. The theory in this chapter is based on [1].

Two-point flux approximation

Two-point flux approximation (TPFA) is the one-dimensional version of MPFA. Both models used in this thesis are one-dimensional, so we will only derive the TPFA. We start with the ordinary differential equation

$$-(Ku_x)_x = Q, \tag{3.13}$$

where $K = K(x) > 0$ can denote the permeability and Q the source term, which depends on x and may depend on t. The index denotes the partial derivative with respect to x, such that $u_x = \partial u/\partial x$. If we let the flux q be denoted by

$$q = -Ku_x, \tag{3.14}$$

the equation becomes

$$q_x = Q, \tag{3.15}$$

which we want to solve for q.

To do this, we have to discretize the domain. The grid used will be cell centred, as shown in fig. 3.4.
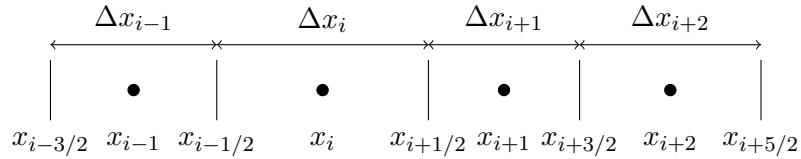
Figure 3.4: Cell centered grid in 1D.

This means that the grid points x_i for $i = 0, \ldots, n$ are the middle points of the cells. For the cell with midpoint x_i, the walls are given by $x_{i-1/2}$ and $x_{i+1/2}$. The distance between the walls of cell i is denoted by Δx_i that is $\Delta x_i = x_{i+1/2} - x_{i-1/2}$.

Integrating eq. (3.15) over the i-th cell yields

$$q_{i+1/2} - q_{i-1/2} = \int_{x_{i-1/2}}^{x_{1+1/2}} Q(x) \, dx, \tag{3.16}$$

with $q_{i+1/2}$ as the flux between cells i and $i+1$ and $q_{i-1/2}$ the flux between the cells $i-1$ and i. To find an expression for $q_{i+1/2}$ in terms of K and u, we rewrite eq. (3.14) as

$$u_x = -\frac{q}{K}. \tag{3.17}$$

We then integrate from middle point x_i to middle point x_{i+1} to get

$$u_{i+1} - u_i = -q_{i+1/2} \int_{x_i}^{x_{i+1}} \frac{1}{K(x)} \, dx, \tag{3.18}$$

for u. Rearranging this gives us the following expression for $q_{i+1/2}$ in terms of u_i and u_{i+1},

$$q_{i+1/2} = -\frac{u_{i+1} - u_i}{\int_{x_i}^{x_{i+1}} \frac{1}{K(x)} \, dx}. \tag{3.19}$$

We now need an approximation of the integral

$$\int_{x_i}^{x_{i+1}} \frac{1}{K(x)} \, dx. \tag{3.20}$$

The integral starts at x_i and ends at x_{i+1}. These are the midpoints in two neighbouring cells, which means that we integrate over two cells. We assume that $K(x)$ is almost constant at each interval, and thus can be denoted by its value in the cell centre, that is $K_i \approx K(x_i)$. We can then approximate the integral in eq. (3.20) by taking the average over the two cells involving x_i and x_{i+1}

$$\int_{x_i}^{x_{i+1}} \frac{1}{K(x)} \, dx = \frac{1}{2} \left(\frac{\Delta x_{i+1}}{K_{i+1}} + \frac{\Delta x_i}{K_i} \right). \tag{3.21}$$

This gives us the expression

$$q_{i+1/2} = -\frac{u_{i+1} - u_i}{\frac{1}{2} \left(\frac{\Delta x_{i+1}}{K_{i+1}} + \frac{\Delta x_i}{K_i} \right)}, \tag{3.22}$$

for $q_{i+1/2}$. By defining an a_i as

$$a_i = \frac{1}{\frac{1}{2} \left(\frac{\Delta x_i}{K_i} + \frac{\Delta x_{i-1}}{K_{i-1}} \right)}, \tag{3.23}$$

eq. (3.22) can be written as

$$q_{i+1/2} = -a_{i+1}(u_{i+1} - u_i). \tag{3.24}$$

If the grid is equidistant, and still cell centred, this expression can be simplified. As $\Delta x_i = h$ for all $i = 1, \ldots, n$, eq. (3.23) can be written as

$$a_i = \frac{1}{\frac{h}{2}\left(\frac{1}{K_i} + \frac{1}{K_{i-1}}\right)}, \tag{3.25}$$

In the rest of this thesis, an equidistant, cell centred grid will be used.

Now, using the expression for q for an equidistant, cell centred grid, eq. (3.16) becomes

$$\frac{u_i - u_{i-1}}{\frac{h}{2}\left(\frac{1}{k_i} + \frac{1}{K_{i-1}}\right)} - \frac{u_{i+1} - u_i}{\frac{h}{2}\left(\frac{1}{k_{i+1}} + \frac{1}{K_i}\right)} = \int_{x_{i-1/2}}^{x_{i+1/2}} Q(x)\, dx. \tag{3.26}$$

For simplicity we define b_i such that

$$b_i = \int_{x_{i-1/2}}^{x_{i+1/2}} Q(x)\, dx, \tag{3.27}$$

and eq. (3.26) can be written as

$$a_i(u_i - u_{i-1}) - a_{i+1}(u_{i+1} - u_i) = b_i. \tag{3.28}$$

Rearranging the terms give

$$-a_i u_{i-1} + (a_i + a_{i+1})u_i - a_{i+1}u_{i+1} = b_i. \tag{3.29}$$

Remembering that $i = 1, \ldots, n$, the result is a system with n equations and $[u_0, u_1, \ldots, u_n, u_{n+1}]$ as $n + 2$ unknowns. Thus, in order to get a unique solution we need to add two more equations. These will be determined through the boundary conditions. The equations can be written as a matrix system, $Au = b$. The matrix A is an $n \times n + 2$-matrix that will be

$$A = \frac{1}{h}\begin{bmatrix} -a_1 & a_1 + a_2 & -a_2 & & & & \\ & -a_2 & a_2 + a_3 & -a_3 & & \text{\Large 0} & \\ & & \ddots & \ddots & \ddots & & \\ & \text{\Large 0} & & -a_{n-1} & a_{n-1} + a_n & -a_n & \\ & & & & -a_n & a_n + a_{n+1} & -a_{n+1} \end{bmatrix} \tag{3.30}$$

This is multiplied with the $n + 2$ long u-vector, and the right hand side b-vector is n long and given as

$$
b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}
\tag{3.31}
$$

Before we look at how to cope with the boundary conditions, we want to compare this discretization scheme, eq. (3.29), with the one obtained by the finite difference approximation, eq. (3.7). If we let $K_i = K$, a constant for all i, then the finite difference discretization of eq. (3.13) by eq. (3.7) is

$$
-K\frac{u_{i+1} - 2u_i + u_{i-1}}{2h} = Q(x_i).
\tag{3.32}
$$

If $K \neq K(x)$, eq. (3.25) becomes $a_i = K/h$. Using this and approximating b_i by the midpoint rule, that is $b_i \approx hQ(x_i)$, eq. (3.29) will be the same as eq. (3.32). From eq. (3.7) we then know that this is a second order method, so the error is of order $O(h^2)$.

3.5   Boundary conditions

In order to find u_0 and u_{n+1}, we need to know what kind of boundary conditions we have. The most common are Dirichlet, where the function value is given at the boundary, and Neumann, where the value of the function derivative is given at the boundary. There may also be mixed boundary conditions, with Dirichlet at one boundary, and Neumann at the other. We will show how to incorporate both Dirichlet and Neumann boundary conditions both at the beginning and the end.

3.5.1   Dirichlet boundary conditions

For Dirichlet boundary conditions we know the function value at the end points. Because the grid is cell centred, these points will be located on $x_{1/2}$ and $x_{n+1/2}$, see fig. 3.5. This means that the boundary conditions

$$
u(0) = u_0,
\tag{3.33}
$$

and
$$u(L) = u_L, \tag{3.34}$$

will be discretized to
$$u_{1/2} = u_0, \tag{3.35}$$

and
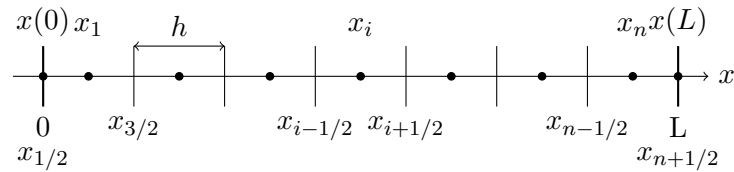$$u_{n+1/2} = u_L. \tag{3.36}$$

Figure 3.5: Cell centred equidistant grid with boundary conditions.

This poses a problem. To solve it, we add ghost cells on the ends of the interval, assume that u is constant on these cells and use the points x_0 and x_{n+1}. An equidistant 1D grid with boundary conditions is shown in fig. 3.5, and the same grid with added ghost cells can be seen in fig. 3.6.
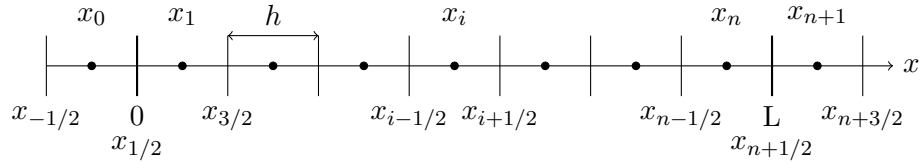
Figure 3.6: Cell centred equidistant grid with ghost cells.

By adding the ghost cells, the boundary conditions are basically handled the same way they would have been if the grid had been vortex centred. The new points in the middle of the new cells will be two additional points, called x_0 and x_{n+1}. The assumption that u is constant on the ghost cells gives us that
$$u_0 = u_0, \tag{3.37}$$

and
$$u_{n+1} = u_L, \tag{3.38}$$

when the problem is discretized.

In the system $Au = b$, we can implement this as the first and last row of the matrix, by setting $A(0,0) = A(n+2, n+2) = 1$, $A(0, j) = 0$ for $j = 1, \cdots, n+2$ and $A(n+2, k) = 0$ for $k = 0, \cdots, n+1$. The first and last entries in b will then be $b(0) = u_0$ and $b(n+2) = u_L$. The matrix A, now an $n+2 \times n+2$-matrix, and right hand side vector b, now $n+2$ long, will be

$$A = \frac{1}{h}
\begin{bmatrix}
1 & & & & & & & 0 \\
-a_1 & a_1 + a_2 & -a_2 & & & & & \\
& -a_2 & a_2 + a_3 & -a_3 & & & & \\
& & \ddots & \ddots & \ddots & & & \\
& & & -a_{n-1} & a_{n-1} + a_n & -a_n & & \\
& & & & -a_n & a_n + a_{n+1} & -a_{n+1} & \\
0 & & & & & & 1 &
\end{bmatrix}
\tag{3.39}$$

and

$$b =
\begin{bmatrix}
u_0 \\
b_1 \\
\vdots \\
b_n \\
u_L
\end{bmatrix}
\tag{3.40}$$

respectively. As all entries a_i, $i = 1, \ldots, n+1$ are positive, this is a diagonally dominant matrix, that is, the equation system $Au = b$ has only one solution by the invertible matrix theorem, and our system is complete.

3.5.2   Neumann boundary conditions

When we have Neumann boundary conditions, we know the value of the first derivative on the boundary. For the 1D case this is the value at the end points, which will be

$$u_x(0) = u_\alpha, \tag{3.41}$$

and

$$u_x(L) = u_\beta. \tag{3.42}$$

Again, we have the situation from fig. 3.5 with $x_{1/2} = 0$ and $x_{n+1/2} = L$. Adding ghost cells, as shown in fig. 3.6, allows us to treat it as we would have on a point distributed grid. The first boundary condition, $u_x(0)$ can

be approximated by $\frac{u_1 - u_0}{\Delta x} = \alpha$. To fit this into our matrix, we multiply by Δx on both sides, which results in $u_1 - u_0 = \alpha \Delta x$. This can be written as $u_0 - u_1 = -\alpha \Delta x$. Similarly, for the last term we write $\frac{u_{n+1} - u_n}{\Delta x} = \beta$ as $u_{n+1} - u_n = \beta \Delta x$. This makes A and b

$$
A = \frac{1}{h}
\begin{bmatrix}
1 & -1 & & & & & & \\
-a_1 & a_1 + a_2 & -a_2 & & & & \text{\Large 0} & \\
 & -a_2 & a_2 + a_3 & -a_3 & & & & \\
 & & \ddots & \ddots & \ddots & & & \\
 & & & -a_{n-1} & a_{n-1} + a_n & -a_n & & \\
 & \text{\Large 0} & & & -a_n & a_n + a_{n+1} & -a_{n+1} \\
 & & & & & -1 & 1
\end{bmatrix}
\tag{3.43}
$$

and

$$
b =
\begin{bmatrix}
-\alpha \Delta x \\
b_1 \\
\vdots \\
b_n \\
\beta \Delta x
\end{bmatrix}
\tag{3.44}
$$

respectively. We see that once again the matrix is diagonally dominant which ensures a unique solution for our equation.

3.6 Discretizing the equations

The equations used to simulate controlled drug release contain a combination of components than the ones described earlier in this chapter. In this section we will look at how they are discretized and implemented for each of the two models.

3.6.1 Implementation of the grain model

As the main equations are partly coupled, the implementation has to solve them together. However, they are not fully coupled, and by solving them in

the right order, they can be solved separately on each time step. The first equation to be solved is

$$\partial_t C_{ES} = k_1(N - C_{ES})C_E - k_2 C_{ES}. \tag{3.45}$$

Although $C_{ES} = C_{ES}(x,t)$, it is only differentiated with respect to time. The space discretization will happen naturally by solving it for each spatial grid point. To discretize it in time we will use forward Euler, that is, the scheme will be

$$C_{ES,i}^{j+1} = \Delta t[k_1(N - C_{ES,i}^j)C_{E,i}^j - k_2 C_{ES,i}^j] + C_{ES,i}^j. \tag{3.46}$$

The second equation is the equation for the radius of the implant. As the previous equation, the equation for R only contains time differentiation, and will also be solved by Euler's method. The scheme will then be

$$R_i^{j+1} = \Delta t \left[-\frac{1 - \theta(R_0^j)}{C_C^0} k_2(C_{ES,i}^j R_i^j)^\gamma \right] + R_i^j. \tag{3.47}$$

It uses $C_{ES,i}$ from the previous time step, because forward Euler is used. Later, it would however be nice to use backward Euler on one or both of these equations. In order to do that on the last equation, we will need $C_{ES,i}^{j+1}$. If we solve for C_{ES} first, they are still not fully coupled. If however, we want to solve for $C_{ES,i}$ implicitly, they become fully coupled as it will depend on $C_{E,i}^{j+1}$. Note that although both C_{ES} and R may vary in the spatial direction, their discretizations are not dependent on x. The reason is that each element x_i is independent of the other elements, x_k, $k \neq i$. Because the spatial discretization solves the equations on each element, the new value for both C_{ES} and R only depends on the previous value at the same interval. Writing them as vectors with one entry for each element then eliminates their dependence on x.

The PDE's will be solved implicitly, and they will therefore need both $C_{ES,i}^{j+1}$ and R_i^{j+1}. However, the two PDE's are independent of each other. For the time discretization backward Euler is used, and for the space discretization the two point flux approximation for equidistant grid eq. (3.29). This results in a matrix system $Au = b$ with a tridiagonal matrix. The discretization of the equation for the enzyme will be

$$\frac{\theta(R_i^{n+1})C_{E,i}^{n+1} - \theta(R_i^n)C_{E,i}^n}{\Delta t} - \frac{a_{i-1}C_{E,i-1}^{n+1} - (a_{i-1} + a_i)C_{E,i}^{n+1} - a_i C_{E,i+1}^{n+1}}{(\Delta x)^2}$$
$$= -2\pi R_i^{n+1}(k_1(N - C_{ES,i}^{n+1})C_{E,i}^{n+1} - k_2 C_{ES,i}^{n+1}) - \theta(R_i^{n+1})k_{act}C_{E,i}^{n+1}. \tag{3.48}$$

The coefficients a are computed as in section 3.4.1, that is, the equation for a_i will be

$$a_i = \frac{2}{\Delta x \left(\frac{1}{D_E^0 \bar{D}(R_i^{n+1})} + \frac{1}{D_E^0 \bar{D}(R_{i-1}^{n+1})} \right)}. \tag{3.49}$$

This can be written with the terms for the different C_E's separated. It will then look like

$$
\begin{aligned}
C_{E,i-1}^{n+1} &\left[-\frac{a_{i-1}}{(\Delta x)^2} \right] \\
+ C_{E,i}^{n+1} &\left[\frac{\theta(R_i^{n+1})}{\Delta t} + \frac{a_{i-1} + a_i}{(\Delta x)^2} + 2\pi R_i^{n+1} k_1 (N - C_{ES,i}^{n+1}) + \theta(R_i^{n+1}) k_{act} \right] \\
+ C_{E,i+1}^{n+1} &\left[-\frac{a_i}{(\Delta x)^2} \right] \\
&= 2\pi R_i^{n+1} k_2 C_{ES,i}^{n+1} + \frac{\theta(R_i^n) C_{E,i}^n}{\Delta t}.
\end{aligned}
\tag{3.50}
$$

Similarly the equation for the drug release is discretized to

$$
\begin{aligned}
&\frac{\theta(R_i^{n+1}) C_{A,i}^{n+1} - \theta(R_i^n) C_{A,i}^n}{\Delta t} \\
&- \frac{a_{i-1} C_{A,i-1}^{n+1} - (a_{i-1} + a_i) C_{A,i}^{n+1} + a_i C_{A,i+1}^{n+1}}{(\Delta x)^2} \\
&= 2\pi R_i^{n+1} \frac{C_{A,im}^0}{C_C^0} k (C_{ES,i}^{n+1} R_i^{n+1})^\gamma,
\end{aligned}
\tag{3.51}
$$

with

$$a_i = \frac{2}{\Delta x \left(\frac{1}{D_A^0 \bar{D}(R_i^{n+1})} + \frac{1}{D_A^0 \bar{D}(R_{i-1}^{n+1})} \right)}. \tag{3.52}$$

Grouping the terms together as we did for C_E yields

$$
\begin{aligned}
C_{A,i-1}^{n+1} &\left[-\frac{a_{i-1}}{(\Delta x)^2} \right] + C_{A,i}^{n+1} \left[\frac{\theta(R_i^{n+1})}{\Delta t} + \frac{a_{i-1} + a_i}{(\Delta x)^2} \right] + C_{A,i+1}^{n+1} \left[-\frac{a_i}{(\Delta x)^2} \right] \\
&= 2\pi R_i^{n+1} \frac{C_{A,im}^0}{C_C^0} k (C_{ES,i}^{n+1} R_i^{n+1})^\gamma + \theta(R_i^n) C_{A,i}^n.
\end{aligned}
\tag{3.53}
$$

It is easy to see that both matrices will still be diagonally dominant. In absolute value, the diagonal elements contain the sum of the off-diagonal elements, with the addition of other positive terms.

3.6.2 Implementation of the channel geometry model

As for the grain model, the equations for the new model need to be discretized. The new model also consists of four partially coupled equations. The equations for d and C_{ES} will be solved explicitly, and C_E and C_A will be solved implicitly. This allows us to solve the equations separately on each time step. However, this is complicated slightly by the introduction of the Heaviside function. We start by discretizing the equation for d to solve it explicitly

$$\rho \frac{d_i^{n+1} - d_i^n}{\Delta t} = -k_4 H(\rho d_i^n) C_{E,i}^n (N - C_{ES,i}^n). \qquad (3.54)$$

This results in the scheme

$$d_i^{n+1} = \Delta t \left[-\frac{k_4}{\rho} H(\rho d_i^n) C_{E,i}^n (N - C_{ES,i}^n) \right] + d_i^n. \qquad (3.55)$$

Although C_{ES} will be solved explicitly, it makes more sense to use the Heaviside function for time step $n + 1$, because this tells us whether or not there is collagen present at the time we want to calculate. This is possible, as d^{n+1} already is found on each element, and ρ is constant. The discretization then becomes

$$\begin{aligned} \frac{C_{ES,i}^{n+1} - C_{ES,i}^n}{\Delta t} =& H(\rho d_i^{n+1})[k_1 C_{E,i}^n (N - C_{ES,i}^n) - k_2 C_{ES,i}^n] \\ & - (1 - H(\rho d_i^{n+1})) k_3 C_{ES}, \end{aligned} \qquad (3.56)$$

with the scheme

$$\begin{aligned} C_{ES,i}^{n+1} =& \Delta t \big[H(\rho d_i^{n+1})[k_1 C_{E,i}^n (N - C_{ES,i}^n) - k_2 C_{ES,i}^n] \\ & - (1 - H(\rho d_i^{n+1})) k_3 C_{ES} \big] + C_{ES,i}^n, \end{aligned} \qquad (3.57)$$

The equations for C_E and C_A are again independent of each other, so it does not matter which is solved first. We will here go through the scheme for C_E first. It is important to make sure that mass is conserved, which is done by choosing the right points for the fluxes. If we look at eq. (2.78), the term

that determines the flux is $D_E \partial_x[(l-d)\partial_x C_E]$, which can be rewritten as $D_E \partial_x(F)$, with F as the flux. For point i, this can be discretized to

$$\frac{1}{\Delta x}[F_{i+1/2} - F_{i-1/2}], \tag{3.58}$$

where $F_{i+1/2}$ is the flux over cell wall $x_{i+1/2}$. There are no sources or sinks inside the domain, so all mass changes are due to the flux. We see that for each interval the mass change is equal to flux in minus flux out, which means that mass is conserved on each interval. We want to be sure that this is the case for the whole domain as well. Summing the fluxes over all the boundaries, that is $i = 1, \ldots, n$ gives

$$\begin{aligned}
&[\cancel{F_{3/2}} - F_{1/2}] \\
+&[\cancel{F_{5/2}} - \cancel{F_{3/2}}] \\
+&[\cancel{F_{7/2}} - \cancel{F_{5/2}}] \\
&\qquad \vdots \\
+&[\cancel{F_{n-1/2}} - \cancel{F_{n-3/2}}] \\
+&[F_{n+1/2} - \cancel{F_{n-1/2}}].
\end{aligned} \tag{3.59}$$

This is reduced to $F_{n+1/2} - F_{1/2}$, that is the change in mass for the whole domain is the change in flux over the edges. This means that mass is conserved over the whole domain.

When we want to implement this, we need the value of d on the cell walls. However, as the TPFA method uses mid points, we do not know this value of d. We will approximate it by the values of the two adjoining mid points, so that the implementation for $F_{i+1/2}$ is

$$F_{i+1/2} = [(l-d)\partial_x C_E]_{i+1/2} \approx \left(l - \frac{d_{i+1} + d_i}{2}\right) \frac{C_{E,i+1} - C_{E,i}}{\Delta x}. \tag{3.60}$$

Using this and taking into account the time discretization the implemented equation for C_E is

$$\begin{aligned}
&\frac{(l - d_i^{n+1})C_{E,i}^{n+1} - (l - d_i^n)C_{E,i}^n}{\Delta t} + \partial_t C_{ES}|_i^{n+1} \\
=&\frac{D_E}{(\Delta x)^2} \left[\left(l - \frac{d_{i-1}^{n+1} + d_i^{n+1}}{2}\right) C_{E,i-1}^{n+1} - \left(2l - \frac{d_{i-1}^{n+1} + 2d_i^{n+1} + d_{i+1}^{n+1}}{2}\right) C_{E,i}^{n+1} \right. \\
&\left. + \left(l - \frac{d_i^{n+1} + d_{i+1}^{n+1}}{2}\right) C_{E,i+1}^{n+1} \right] + k_{act}(l - d_i^{n+1})C_{E,i}^{n+1}.
\end{aligned}$$

$$\tag{3.61}$$

This has the scheme

$$C_{E,i-1}^{n+1}\left[-\frac{D_E}{(\Delta x)^2}\left(l-\frac{d_{i-1}^{n+1}+d_i^{n+1}}{2}\right)\right]$$

$$+C_{E,i}^{n+1}\left[\frac{l-d_i^{n+1}}{\Delta t}+\frac{D_E}{(\Delta x)^2}\left(2l-\frac{d_{i-1}^{n+1}+2d_i^{n+1}+d_{i+1}^{n+1}}{2}\right)-k_{act}(l-d_i^{n+1})\right]$$

$$+C_{E,i+1}^{n+1}\left[-\frac{D_E}{(\Delta x)^2}\left(l-\frac{d_{i-1}^{n+1}+d_i^{n+1}}{2}\right)\right]$$

$$=\frac{l-d_i^n}{\Delta t}C_{E,i}^n-\partial_t C_{ES}|_i^{n+1}.$$

$$(3.62)$$

Similarly the discretized equation for C_A will be

$$\frac{(l-d_i^{n+1})C_{A,i}^{n+1}-(l-d_i^n)C_{A,i}^n}{\Delta t}+\partial_t\rho d|_i^{n+1}$$

$$=\frac{D_A}{(\Delta x)^2}\left[\left(l-\frac{d_{i-1}^{n+1}+d_i^{n+1}}{2}\right)C_{A,i-1}^{n+1}-\left(2l-\frac{d_{i-1}^{n+1}+2d_i^{n+1}+d_{i+1}^{n+1}}{2}\right)C_{A,i}^{n+1}\right.$$

$$\left.+\left(l-\frac{d_i^{n+1}+d_{i+1}^{n+1}}{2}\right)C_{A,i+1}^{n+1}\right].$$

$$(3.63)$$

The implemented scheme is

$$C_{A,i-1}^{n+1}\left[-\frac{D_A}{(\Delta x)^2}\left(l-\frac{d_{i-1}^{n+1}+d_i^{n+1}}{2}\right)\right]$$

$$+C_{A,i}^{n+1}\left[\frac{l-d_i^{n+1}}{\Delta t}+\frac{D_A}{(\Delta x)^2}\left(2l-\frac{d_{i-1}^{n+1}+2d_i^{n+1}+d_{i+1}^{n+1}}{2}\right)\right]$$

$$+C_{A,i+1}^{n+1}\left[-\frac{D_A}{(\Delta x)^2}\left(l-\frac{d_{i-1}^{n+1}+d_i^{n+1}}{2}\right)\right]$$

$$=\frac{l-d_i^n}{\Delta t}C_{A,i}^n-\partial_t\rho d|_i^{n+1}.$$

$$(3.64)$$

It is easy to see that the matrix for C_A will still be diagonally dominant, as the diagonal element is the sum of the absolute value of the off-diagonal elements and another positive term. The matrix for C_E will be diagonally dominant as long as

$$\frac{l-d_i^{n+1}}{\Delta t}>k_{act}(l-d_i^{n+1}). \tag{3.65}$$

3.7 Implementation

For the implementation of the grain model, we treated D_E, D_A, θ and the right hand side functions for C_E and C_A as functions by using MATLAB's function handle. For each time step, forward Euler was used to find C_{ES} and R, and a combination of forward Euler and MPFA was used to solve the equations for C_E and C_A. The matrices used for this are adapted depending on the boundary conditions, and can be used for both Dirichlet and Neumann boundary conditions, as well as a combination of the two. To make sure that the radius would not become negative, we used the maximum value of the computed R and 0 for each interval.

In the channel model, both D's were treated as constants. However, the right hand sides were still treated as function handles. For this implementation, we first used forward Euler to solve for d, then found the new H and used this in the forward Euler calculation of C_{ES}. Note that if we choose a small enough time step the Heaviside function ensures that the minimum value of d is 0. A combination of MPFA and forward Euler was again used on C_E and C_A, with the matrices depending on type of boundary condition.

We chose the MPFA method do discretize our equations in space, because it conserves mass. This along with the forward Euler method was adapted to our discretized equations, and the implementations are possible to use for both Neumann and Dirichlet boundary conditions, even though the current models only have Dirichlet boundary conditions. To make sure the implementations were correct, they were tested for both accuracy and convergence. The results of this, as well as a sensitivity analysis will be found in the next chapter.

When the models were fully tested, we used them on two sets of data. The boundary conditions from the data in [14] are Dirichlet conditions, but they are not constant. In the experiments, the solution containing collagenase is changed at times. This makes the boundary conditions change between $1.4 \times 10^{-6} \mu mol/cm^3$ and 0. At the time the solution is changed, the boundary condition will be $1.4 \times 10^{-6} \mu mol/cm^3$ and then descend linearly towards 0 before the next change. This is due to the collagenase becoming inactive after a certain period of time. This period is not known, as has to be fitted. The new data from October 2013 has similar boundary conditions, but with the concentration of collagenase in the solution $1.25 \times 10^{-6} \mu mol/cm^3$.

There are other constants that we do not have experimental values for, and

they will be fitted as well. In the grain model we have γ, and in the channel model there is k_4. When implementing, we discovered that the fully linear channel model would not fit the data, and added a non-linearity. This was done in the equation for d. This seemed most reasonable considering the non-linear term in the grain model is in the equation for R. This made the equation for d

$$\partial_t d = -\frac{k_4}{\rho_S} H(d\rho)(C_E(N - C_{ES}))^{\gamma_2}, \tag{3.66}$$

and its discretization

$$d_i^{n+1} = \Delta t \left[-\frac{k_4}{\rho} H(\rho d_i^n)(C_{E,i}^n(N - C_{ES,i}^n))^{\gamma_2} \right] + d_i^n. \tag{3.67}$$

Chapter 4

Results

This chapter concerns the numerical results and their analysis. We will start with a comment about the experimental results, before moving on to the implementations. Here we will first show how the methods were tested, and look at convergence rates. A sensitivity analysis is performed to identify the relevant parameters in the models. A mathematical fitting of two parameters in the grain model and three in the channel model has also been performed. These parameters could not be fitted experimentally. Finally, the final fitting was done in comparison with the experimental results. The sensitivity analyses and final fittings are done for three different sets of data, the first one from [14] and the second and third from the new experiments.

4.1 Experimental results

There are two sets of data that are used in this thesis. One is from [14], and one, shich is referred to as the new one, from October 2013. The new experimental results seem to have some inaccuracies, as k_1 and k_2 are of a different order than the constants found in [14]. This may be due to different measuring units, as scaling them made the fitting of the data easy, but it is not clear if this really is the case. The collagen degradation measured in the new data set is much slower than the one from [14]. It may be due to differences in how often the solution that contains collagenase was changed, and how much collagenase it contained, but there may also be mistakes in one or both sets of data. This should be explored in future research. The contents of collagenase in the solution will be implemented as boundary

conditions, and the longer the solution is used unchanged, the more collagen becomes inactive, which in turn slows the process down.

4.2 Testing the implementations

The implementation of both models was tested with the help of known functions to minimize the chance of errors in the code. This was done by computing the right hand side so that the equation admitted the analytical solution. We then plotted the calculated solution over the analytical one, and checked the convergence rate. Throughout the testing, we used an equidistant grid with both t and x on the interval $[0, 1]$, and the same number of steps in both spatial and temporal direction. At $T = 1$, the plot was made and convergence tested for the computational domain $x \in [0, 1]$. The error that is used, is the L^2-error, which is given by

$$E = \sqrt{\int_0^1 (u_{analytical} - u_{numerical})^2 dx}. \tag{4.1}$$

On the testing of the models, the step sizes have been halved between each test. That is, the error should approximate 2 for small step sizes, as shown in section 3.4.1.

4.2.1 Grain geometry

The grain model used $f(x,t) = tx(1 - x) + C$ as the main analytical solution, with C as a constant. Changing the C was important to check that the boundary conditions were handled properly. We used $f(x,t)$ to check convergence of both C_E and C_A. However, as the equations are very similar, their implementations are almost identical. In the development of the program, we have tested convergence and analytical solution with C_E, and in the end checked for C_A. Before each convergence calculation, we also changed some of the constants, to make sure they had been implemented correctly.

The first test was done with constant R, C_{ES} and D. Keeping R constant, meant that θ could also be treated as a constant, which made it easier to test just the implementation of C_E. Figure 4.1 shows the computed
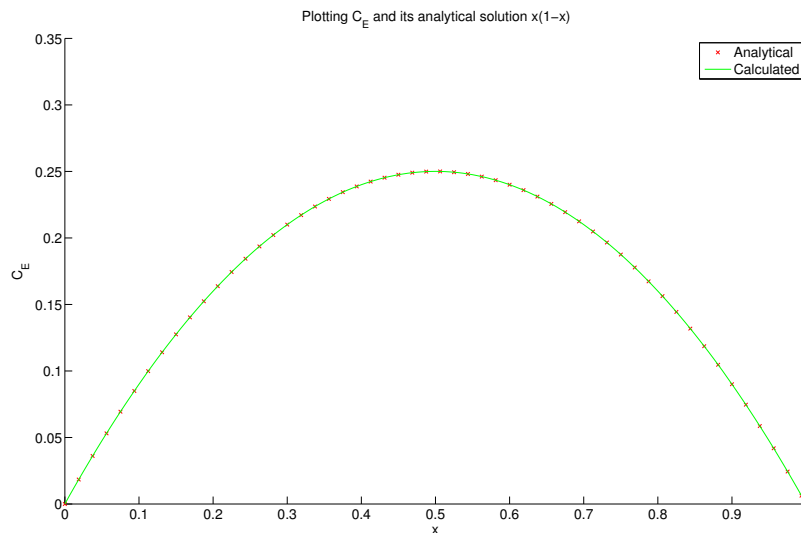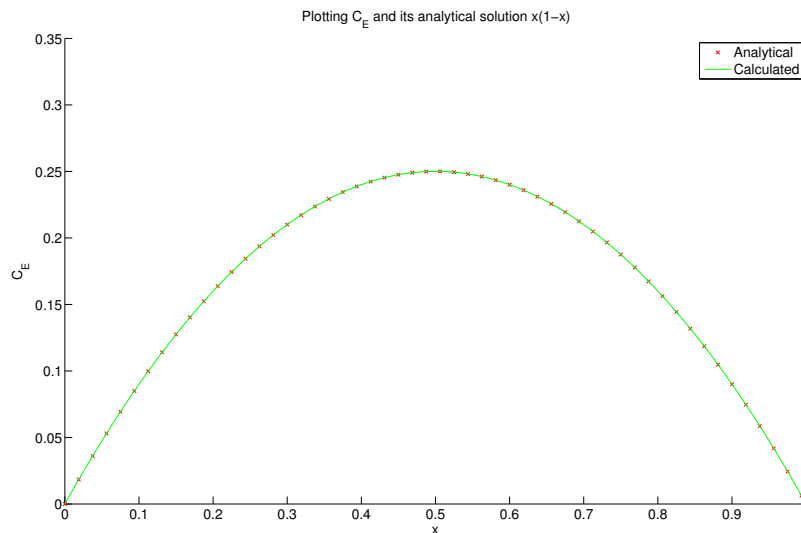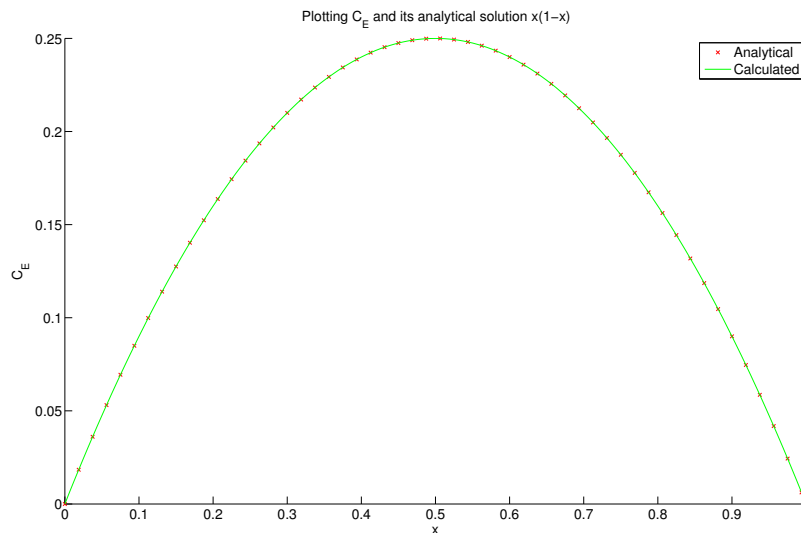
Figure 4.1: Solution of C_E with step length 0.00625 and $C = 0$.

Table 4.1: Error for C_E with constant R and C_{ES}

Step length (Δx)	Error	Convergence
0.1	0.035284557528755	1.861680616927416
0.05	0.018953067034124	1.927626317353635
0.025	0.009832334650911	1.963120830007375
0.0125	0.005008522399955	1.981400817217011
0.00625	0.002527768413354	

solution compared with the analytical, and the error and convergence rate
is in table 4.1.

We see that there is very small error between the computed and analytical
solutions, even on rather big steps. The convergence rate in table 4.1 is
calculated from the error on the same row, as well as the row below. It is
approaching 2 as the steps become smaller, and already when the step size
is reduced from 0.025 to 0.0125 it can be rounded off to 2.0. These results
are in agreement with the expected convergence rate.

In the next test R was found through forward Euler, which meant that
both D and θ changed for each time step. The results are given in fig. 4.2
and table 4.2.
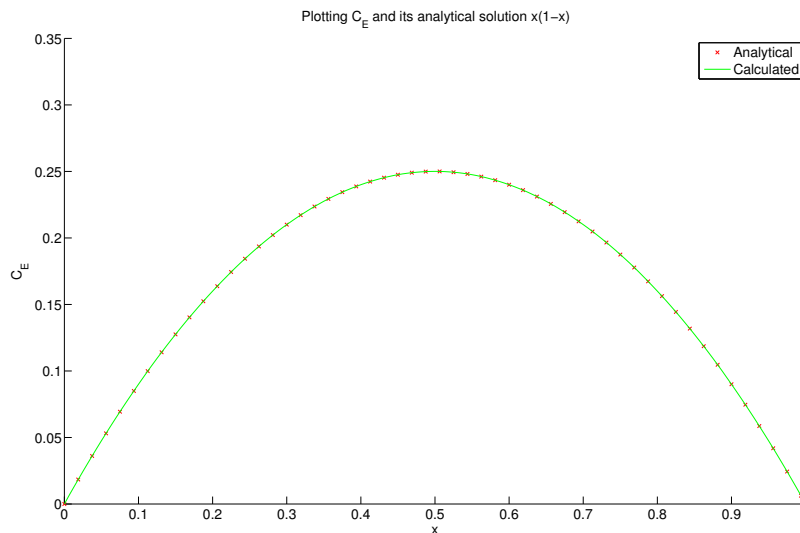
Figure 4.2: Solution of C_E with step length 0.00625 and $C = 0$.

Table 4.2: Error for C_E with constant C_{ES}

Step length (Δx)	Error	Convergence
0.1	0.035293454479411	1.861881753115380
0.05	0.018955798036237	1.927757263895274
0.025	0.009833083444300	1.963192411937406
0.0125	0.005008721195390	1.981437945934715
0.00625	0.002527821376221	

The results are again very good, with a small error and a convergence rate that approximates 2, which suggests a correct implementation. The analytical and computed solutions are almost identical, and the convergence rate is almost the same as in the previous test, as expected.

As it is difficult to test the results for R, θ, D and C_{ES} against analytical solutions, we chose to just test for C_E. To be sure that all calculations were correct, we therefore changed R back to being a constant when testing with C_{ES} as a function. This resulted in fig. 4.3 and table 4.3.

Once more, the results are good with a small error and the convergence rate that quickly approaches 2.

Figure 4.3: Solution of C_E with step length 0.00625 and $C = 0$.

Table 4.3: Error for C_E with constant R

Step length (Δx)	Error	Convergence
0.1	0.035296865086005	1.861928766771014
0.05	0.018957151162779	1.927812044021062
0.025	0.009833505927911	1.963228699155667
0.0125	0.005008843815364	1.981458428020343
0.00625	0.002527857130148	

The last test was done with both R and C_{ES} as functions, which meant that D and θ also were varying. This was tested for C_A as well to make sure that this method was working. Figure 4.4 shows the computed solution for C_E against the analytical solution $f(x, t)$, with Neumann boundary conditions $f(0, t) = f(1, t) = t$. At the end time $t = 1$, when the plot is made, the function is $f(x, t) = tx(1 - x) + 1$.

Although the computed solution of C_E differs a little from the analytical in the middle, as seen in fig. 4.4, table 4.4 shows that the convergence rate is right, that is the solution is quickly approaching the right one. We see that the convergence rate is even better than for the previous tests, with it being 2.0 from the beginning and 2.00 when the step size changes from 0.0125 to
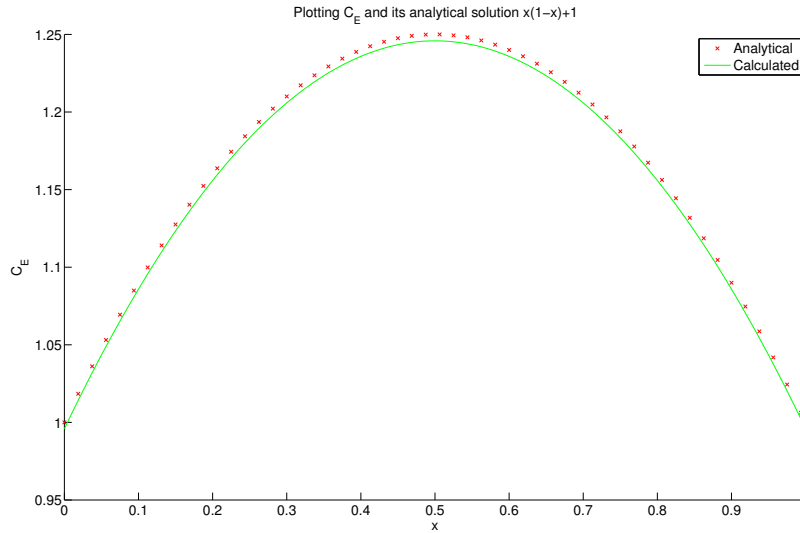
Figure 4.4: Solution of C_E with step length 0.00625 and Neumann boundary conditions.

Table 4.4: Error for C_E with calculated R and C_{ES}

Step length (Δx)	Error	Convergence
0.1	0.060801185635915	1.974156143422392
0.05	0.030798569727372	1.986976269955451
0.025	0.015500220205480	1.993430671771861
0.0125	0.007775650502910	1.996697114951157
0.00625	0.003894256392062	

0.00625.

The implementation of C_A was tested with Dirichlet boundary conditions, and $C = 0$. The results are shown in fig. 4.5 and table 4.5.

Figure 4.5 shows that the computed solution is very similar to the analytical one. This is confirmed by table 4.5, which additionally shows that the convergence rate is approaching 2 for C_A as well as the error being very small.

Figure 4.5: Solution of C_A with step length 0.00625 and $C = 0$.

Table 4.5: Error for C_A with calculated R and C_{ES}

Step length (Δx)	Error	Convergence
0.1	0.035278485658167	1.861562212163140
0.05	0.018951010837921	1.927530312242646
0.025	0.009831757621426	1.963064253573082
0.0125	0.005008372804675	1.981370443329528
0.00625	0.002527731662464	

4.2.2   Channel geometry

We defined $C_E(x, t) = tx(1 - x)$ and compared the analytical and computed solutions to test the method. Additionally, the error between the solutions, as well as the convergence rate was calculated, and the results are in table 4.6. The plot for C_E is given in fig. 4.6.

To test the implementation for solving d, we kept C_{ES} constant and solved d analytically by using the same function for C_E as before. When calculating d as in eq. (2.81) from this, we found its analytical solution to be $d = -x(1 - x)/2 + 0.3$. The plot for the computed d compared to its analytical solution is given if fig. 4.7.

Table 4.6: Error for C_E with computed C_{ES} and d

Step length	Error	Convergence
0.1	0.035383430465186	1.864094179248415
0.05	0.018981568023270	1.928980047580153
0.025	0.009840209621184	1.963791501754844
0.0125	0.005010821980027	1.981728685157167
0.00625	0.002528510596611	
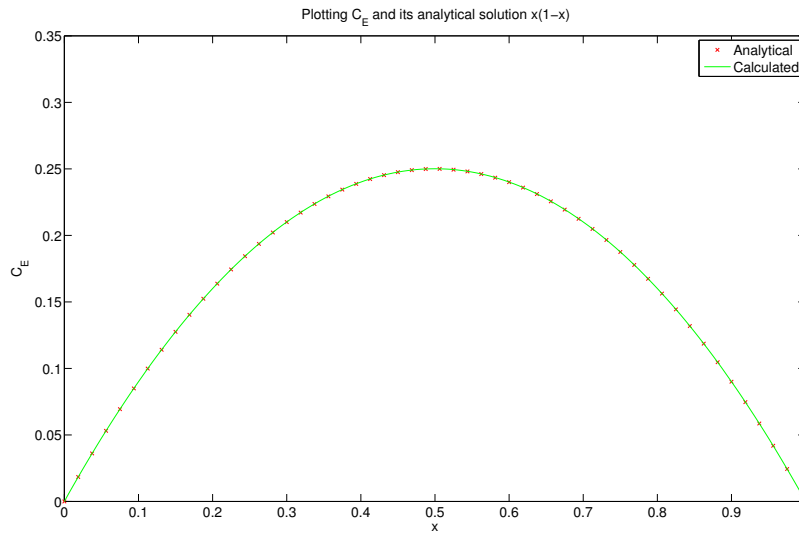
Figure 4.6: Solution of C_E with step length 0.00625.

We see that the plots have good accuracy, and that the error for C_E is small. From fig. 4.7 it is clear that although we have no convergence rate for d, the computation is becoming accurate very quickly. The error rate for C_E is approximately 2, and getting closer to 2 as the number of steps are increased. This is to be expected, and we see that already for the increase of step size from 0.025 to 0.0125, the error can be rounded to 2.0.
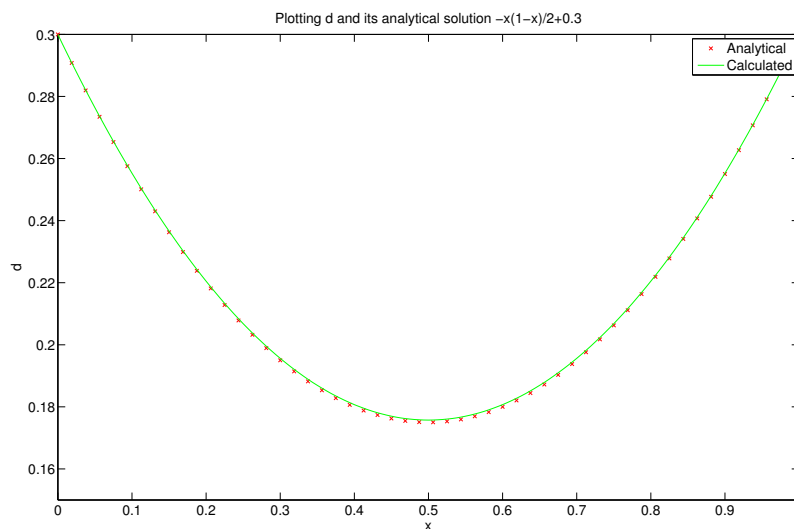
Figure 4.7: Solution of d with step length 0.00625.

4.3   Sensitivity analyses

We did a sensitivity analysis on all the parameters that need fitting. This is to see if small changes in the parameters that are uncertain make a big difference in the output and thereby identifying the relevant parameters.

We did sensitivity analyses with the data set from [14] as well as the new data sets for both models. This was done by systematically changing one of the parameters that need fitting, while not changing anything else. The sensitivity analyses was performed with the changing boundary conditions described in section 3.7.

4.3.1   Data from the paper of Ray et. al.

The first experimental data we used to test our models is the data from [14]. The sensitivity analysis was performed for the collagen degradation.

Grain geometry

For the model with grain geometry we did a sensitivity analysis for the two constants we needed to fit. That is, γ and how long the collagenase is active, which will be shortened to A_C and measured in parts of an hour. We started with γ, while keeping $A_C = 1/16$, and the results are in fig. 4.8.
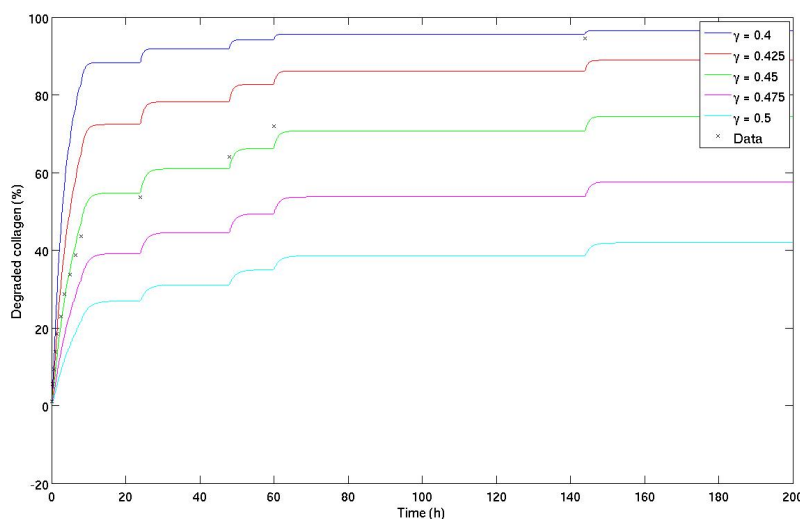
Figure 4.8: Sensitivity analysis for collagen degradation with different γ.

We see that small changes in γ have a big impact on how fast the collagen is degrading. In fig. 4.8, the smallest value of γ is still 80 % of the biggest, but the curves are still very different, especially in the beginning. The curves has some "jumps" where the degradation happens quickly before it slows down. This is due to the boundary conditions, where the jumps represent the changing of solution, thereby adding new, active collagenase to the system. In turn, this results in a quicker degradation of the collagen matrix. How quickly the curve flattens is due to how fast we set the enzyme to become inactive. This is A_C, the second parameter that needs to be fitted. We performed a sensitivity analysis for A_C as well, and this can be seen in fig. 4.9, with $\gamma = 0.6$.

We see that changing A_C has less influence than changing γ. Although the collagenase is active five times as long for $A_C = 1/4$ as for $A_C = 1/20$, the shape of the curve does not change much. The jumps are higher, but
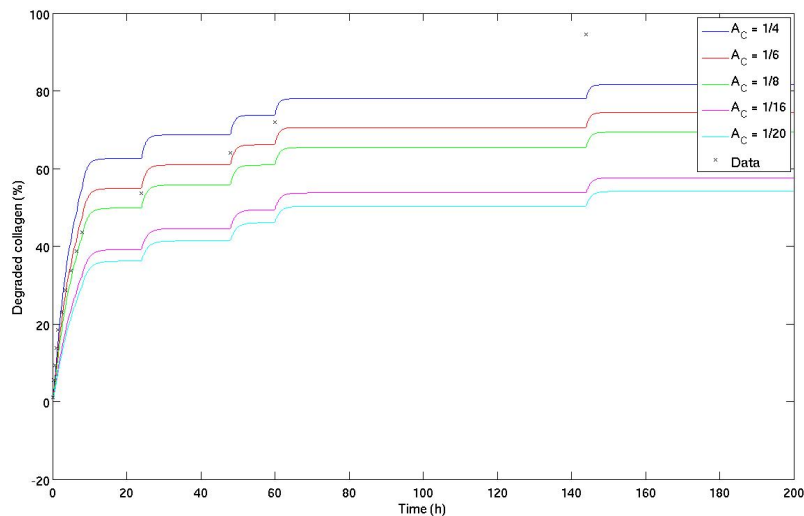
Figure 4.9: Sensitivity analysis with different A_C.

the processes stop quickly after the solution is changed for all A_C's. We have no good explanation for this, as we would have expected the time the collagenase is active to have a bigger impact, and this is something that will need more exploration in the future.

Channel geometry

In the model based on channel geometry, there are three constants that need fitting. These are γ_2 and A_C, which is similar to the grain geometric model, and the rate parameter k_4. We started with the sensitivity analysis for γ_2, shown in fig. 4.10. This was done with $k_4 = 200$ and $A_C = 1/100$.

We see that smaller γ_2 makes the curve steeper. Note also that as γ_2 gets bigger, the change in curves becomes more rapid. We think that $\gamma_2 = 1/3$ looks like a good fit, if it is slowed down towards the end.

The next parameter we fitted was A_C. We varied it between 1/10-th of an hour to 1/150-th of an hour, but it did not make a big difference as seen in fig. 4.11.

Here, $\gamma_2 = 1/3$ and $k_4 = 200$ were kept constant. We were expecting to
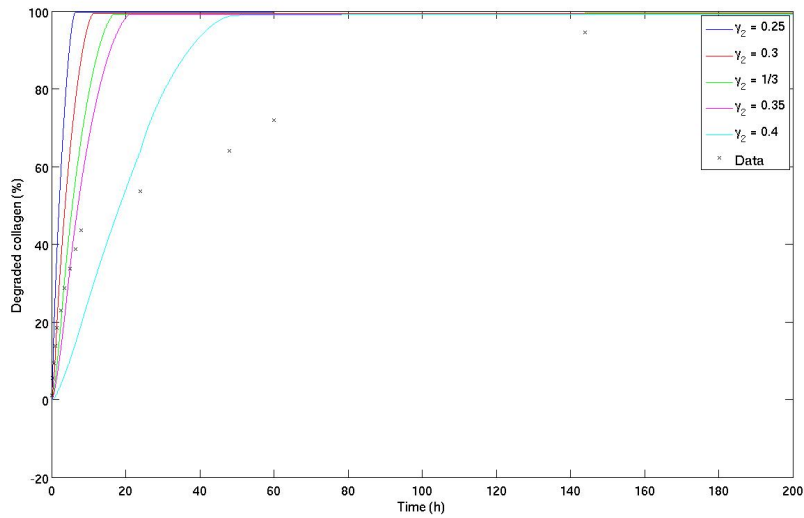
Figure 4.10: Sensitivity analysis with different γ_2.
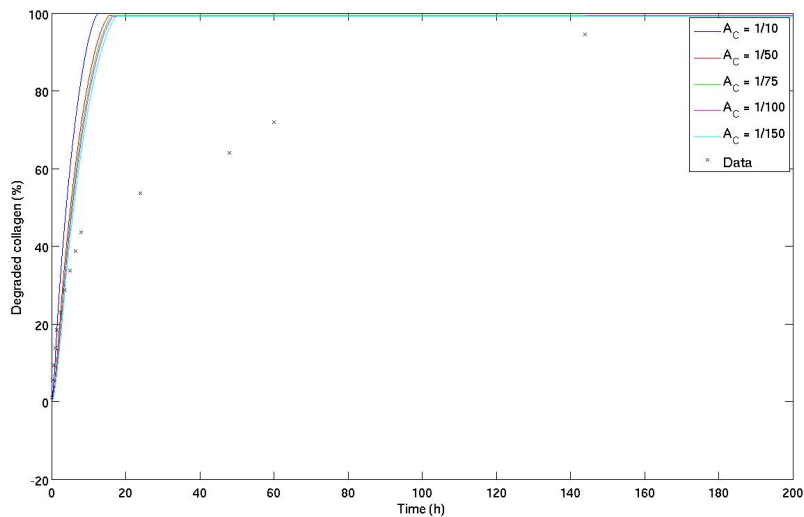
Figure 4.11: Sensitivity analysis with different time for active collagenase.

see jumps in the curve, as we did for the grain geometry, whenever new collagenase is added, but this effect is not visible, even when we set the

time the collagenase is active to be very small. With bigger γ_2 however, the jumps can be seen, if not as clear as we had thought. Figure 4.12 shows the collagen degradation with very big k_4, $\gamma_2 = 2/3$ and $A_C = 1/100$.
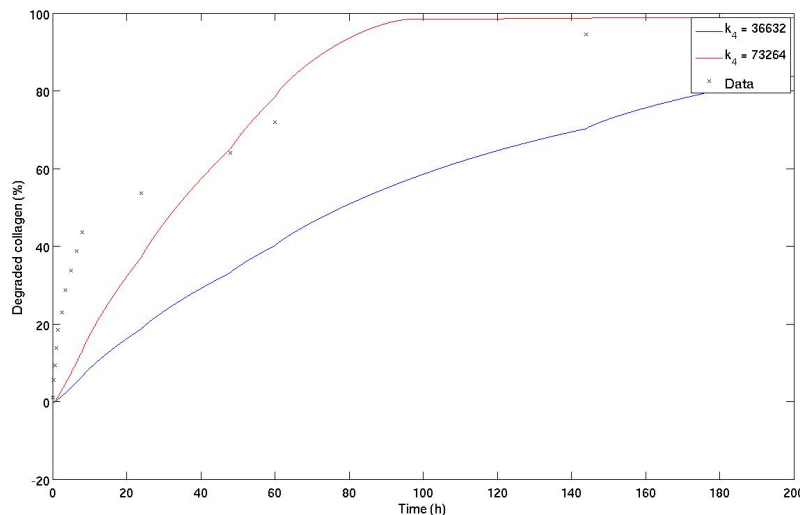
Figure 4.12: Sensitivity analysis with different k_4.

There are visible jumps, but the processes seem to not slow down as quickly as we anticipated. From the curve for $k_4 = 73\,264$, it is clear that even though the jumps become more distinct with higher γ_2, the computed curve still does not fit with the experimental data. Up to about 40 hours the computed concentration is too small, and then it gets too big, which means that the shape of the curve needs to be altered.

The final fitting was done for k_4. These results are in fig. 4.13.

We see that although the process slows down as k_4 decreases, the shape is not altered much. There seems to be something that our model is not considering properly that slows the process down with time. It is natural to think that the model allows the processes to happen too quickly, because we do not see the jumps. Especially towards the end, when collagenase is added infrequently, they should be visible. There are however some uncertainties regarding the boundary conditions. We do not know how long the enzyme is active, and it was not completely clear how often the solution was changed.
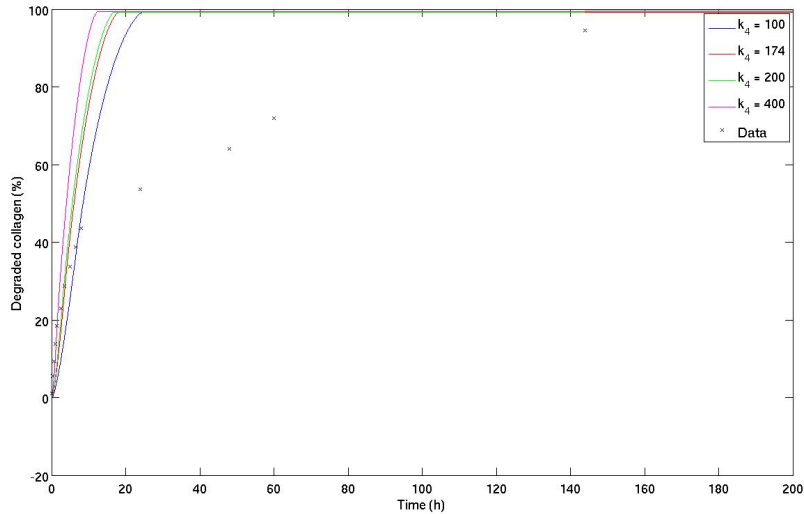
Figure 4.13: Sensitivity analysis with different k_4.

4.3.2 New data set (October 2013), non-cross-linked collagen

The first data set from October 2013 contains collagen that has not been cross-linked. These sets have much fewer data points than the set from [14].

Grain geometry

The sensitivity analysis for the data set with non-cross-linked collagen for the grain geometry can be seen in figs. 4.14 and 4.15. In fig. 4.14 A_C was kept constant at 2, while $\gamma = 0.375$ in fig. 4.15.

From the figures we see that γ has a bigger influence on the result than A_C. Another result worth noticing is that as γ increases or A_C decreases, the distance between the curves increases.
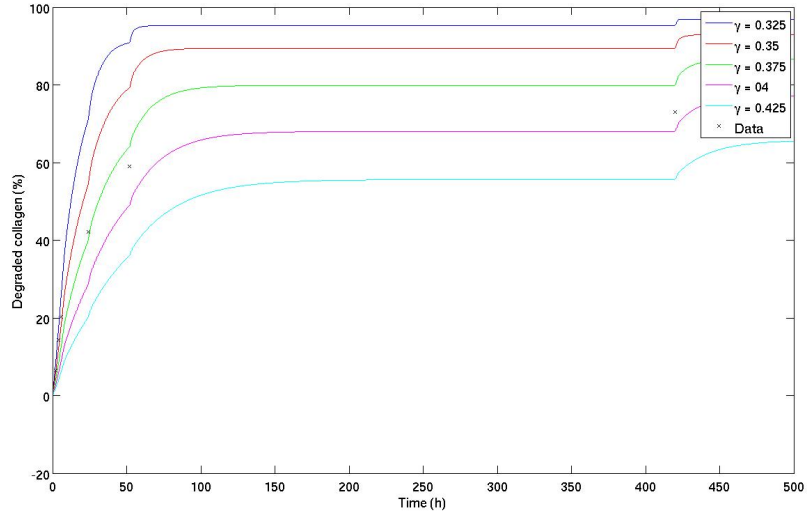
Figure 4.14: Sensitivity analysis with different γ.
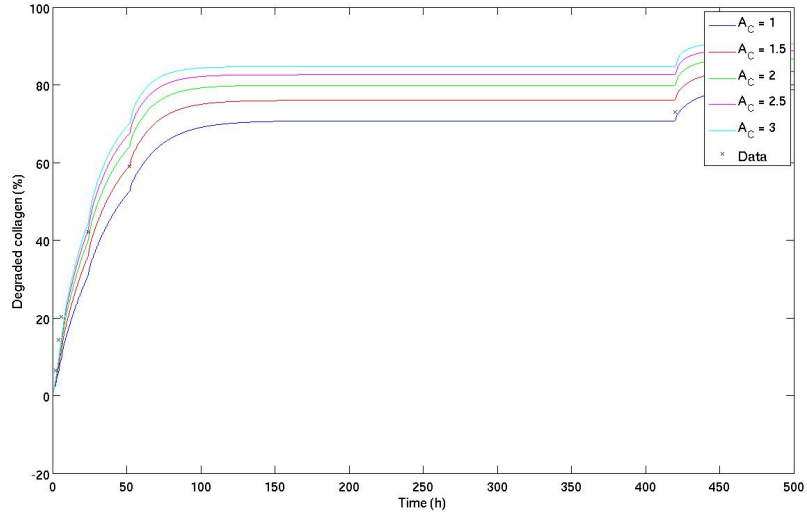
Figure 4.15: Sensitivity analysis with different A_C.

Channel geometry

The channel geometry again has three parameters to be fitted, as shown in figs. 4.16 to 4.18. The first analysis is for γ in fig. 4.16 with $A_C = 1/20$ and

$k_4 = 50$.
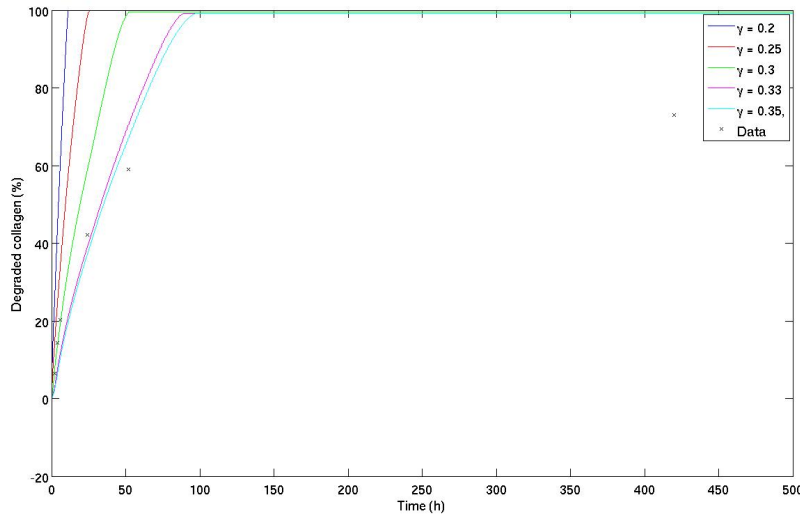
Figure 4.16: Sensitivity analysis with different γ.

It seems that $\gamma = 0.33$ would be a good fit, again if it would slow down towards the end.

To do the analysis on A_C, fig. 4.17, $\gamma = 1/3$ and $k_4 = 50$ were kept constant.

Although A_C varies a lot, with its biggest value being 60 times its smallest value, the curves are very similar. The clearest difference is towards the end of the increase. This is to be expected, because the new solution is added less frequent there, which means that the process depends a lot more on how long the collagenase is active.

Finally, the analysis of k_4 was done with constants $\gamma = 1/3$ and $A_C = 1/20$ as shown in fig. 4.18.

We see that changing the value of k_4 slightly changes the shape of the curve. Higher k_4 makes it slow down more towards the end of its ascendance, but we could not get it to slow down as much as the experimental results showed without loosing accuracy at the beginning.
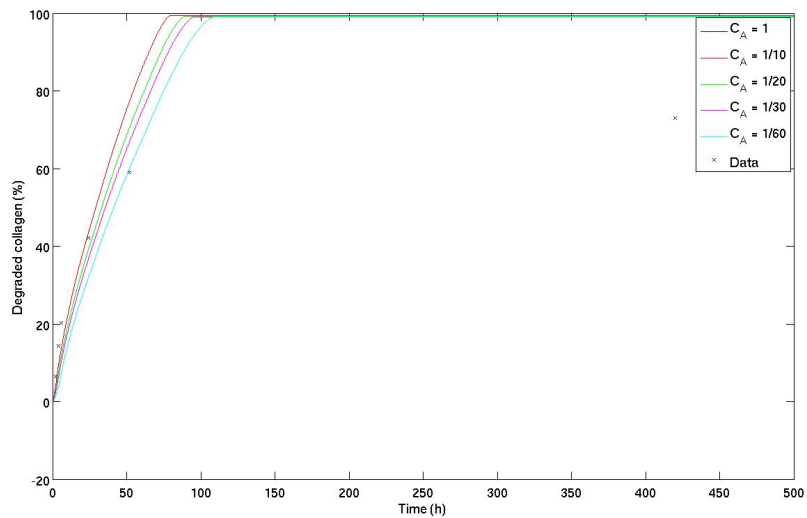
Figure 4.17: Sensitivity analysis with different A_C.
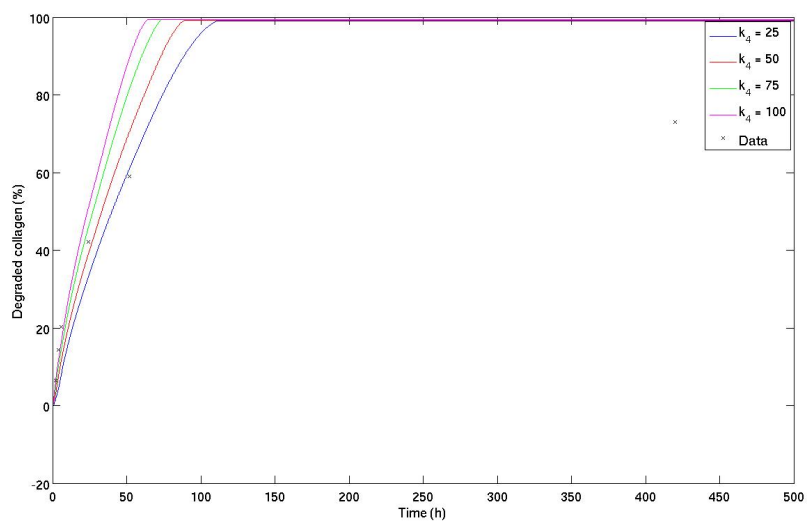
Figure 4.18: Sensitivity analysis with different k_4.

4.3.3   New data set (October 2013), cross-linked

The data set for the enzymatically cross-linked collagen matrices has a much slower degradation than the other data sets. This is to be expected because

the collagen matrices have more links that the collagenase have to break in order for the matrix to be broken down.

Grain geometry

The computations for the collagen degradation with cross-linked collagen can be found in figs. 4.19 and 4.20. In the analysis of γ, C_A was kept constant at 3, whereas γ was kept constant at 0.45 for the analysis of A_C.
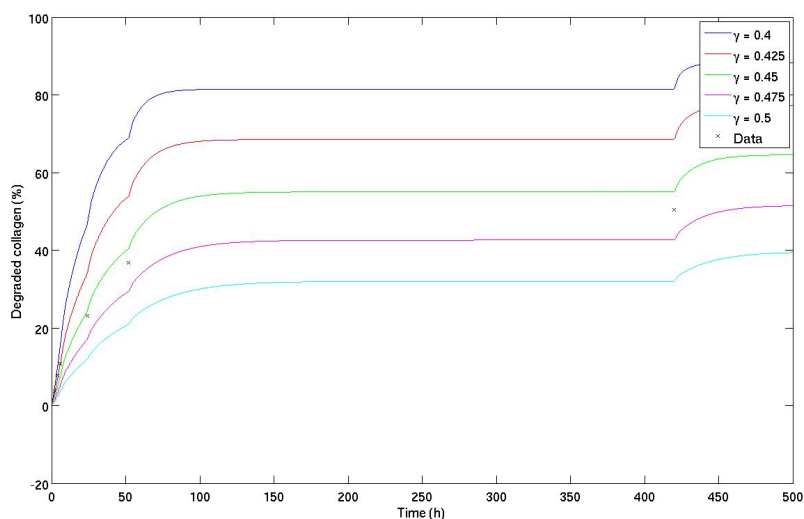
Figure 4.19: Sensitivity analysis with different γ.

There are still clearly visible jumps in the plots, and as earlier changes in γ has a big impact on the computations. The smallest γ is 80 % of the biggest, but calculates a degradation that is more than twice the slowest degradation after only a few hours.

The influence of A_C is still a lot smaller, even though it is bigger than for the non-cross-linked collagen from the previous section. From fig. 4.20 we see that although A_C is changed, the process slows down very quickly after each time the solution is changed. Note also that the scale of A_C differs between the data set from [14] and the data sets from October 2013. In the first, A_C is in parts of an hour, whereas the new data sets have A_C as bigger than an hour.
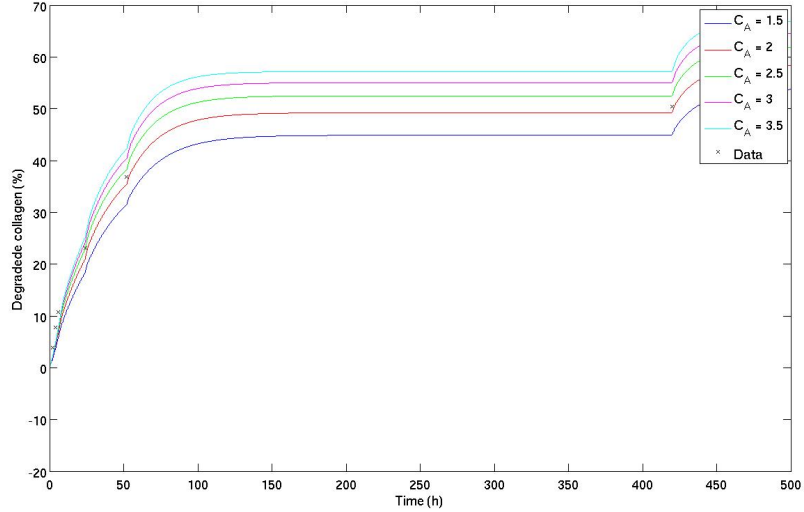
Figure 4.20: Sensitivity analysis with different A_C.

Channel geometry

We started with the sensitivity analysis of γ_2, with $A_C = 1/10$ and $k_4 = 50$. This is shown in fig. 4.21.

Figure 4.21 shows that bigger γ_2 slows the process down, but it is slowed down too much in the beginning, which makes the shape of the curve wrong in the beginning before it is correct towards the end. Again it seems that the new model misses an element that slows the process down as the solution with collagenase is changed with less frequency.

The next parameter to be analysed was A_C. Here, $\gamma_2 = 0.35$ and $k_4 = 25$, which resulted in fig. 4.22.

There is quite a big jump from $A_C = 1$ to $A_C = 1/10$, but apart from that, the choice of A_C does not influence the result greatly. This is consistent with the sensitivity analyses for the other data sets.

Finally, k_4 was analysed in fig. 4.23 with $\gamma_2 = 1/3$ and $A_C = 1/10$.

The fitting of k_4 has the same problem as the fitting of γ_2. That is, we are able to slow down the process, but then it slows down in the beginning as well. Even though it is slowed down more towards the end than in the
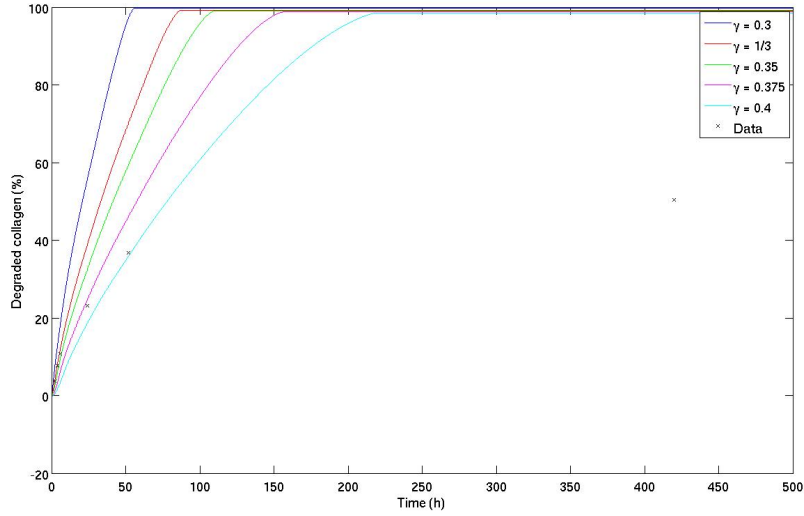
Figure 4.21: Sensitivity analysis with different γ_2.
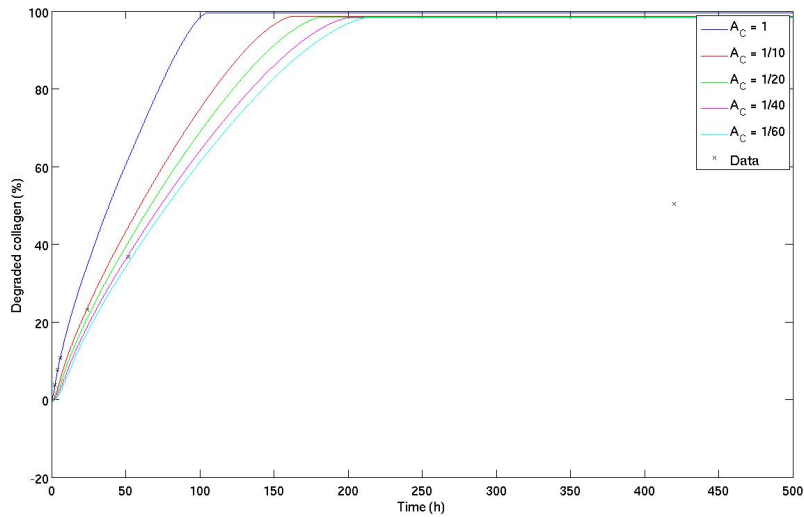
Figure 4.22: Sensitivity analysis with different A_C.

beginning, we see that already for $k_4 = 12.5$, the computation is too slow in the beginning. This means that changing k_4 is not the right approach to
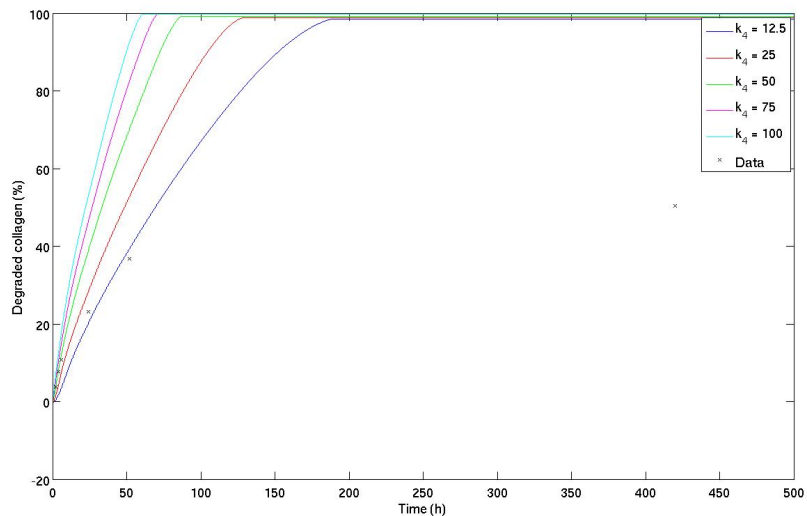
Figure 4.23: Sensitivity analysis with different k_4.

get the process to slow down towards the end, which supports our theory
that an element is missing from the model.

4.4   Comparison with experimental data

We compared our computations with different data sets. The first is the
one from [14], whereas the other two are the sets from October 2013. Based
on the sensitivity analyses for the parameters that can not be determined
experimentally, we found what we believe is the best fitting curve.

4.4.1   Data set from the paper by Ray et. al.

The first data set we adapted our implementation to is the data set from
[9, 10] used in [14].

Grain model

For the first set of data, we had a hard time fitting the grain model. Somehow the fit is not the same for the collagen degradation and the drug release. For the collagen degradation, $\gamma = 0.45$ and $A_C = 1/16$ seemed to be the best fit, but for the drug release $\gamma = 0.475$ and $A_C = 1/20$ is the best fit. These plots are shown in figs. 4.24 and 4.25. This may not seem like a big difference, but as we have seen in the sensitivity analysis for γ, small changes in γ have a big impact on the degradation of the collagen matrix. The same is true for the drug release, so we were unable to find a good fit where both the collagen degradation and drug release had the same parameters.
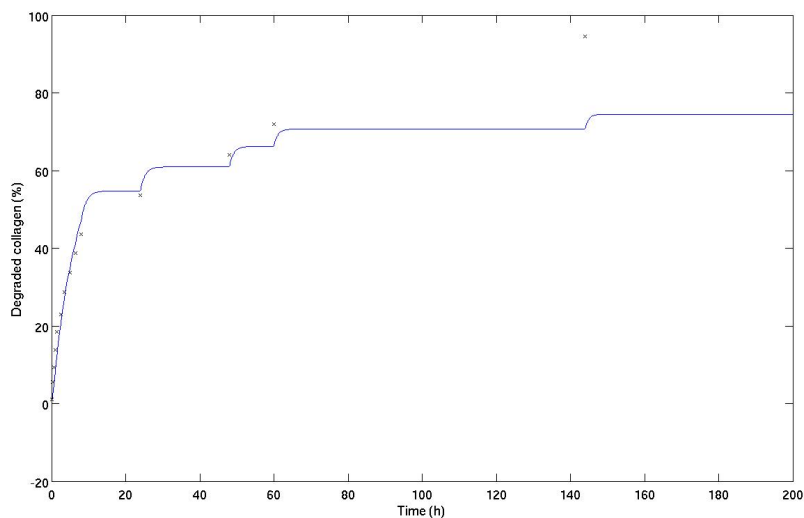
Figure 4.24: Computed degradation compared to experimentally measured degradation.

The plot for the collagen degradation fits well in the beginning, but is too small towards the end. The reason seems to be that the process stops too quickly after a new solution is added. However, changing A_C had very little impact as seen earlier, so it is not clear why this is happening.

Figure 4.25 shows that with the right parameters, the fit for the drug release is very good. Opposed to the computation for the degradation of the matrix, it does not slow down too much towards the end.
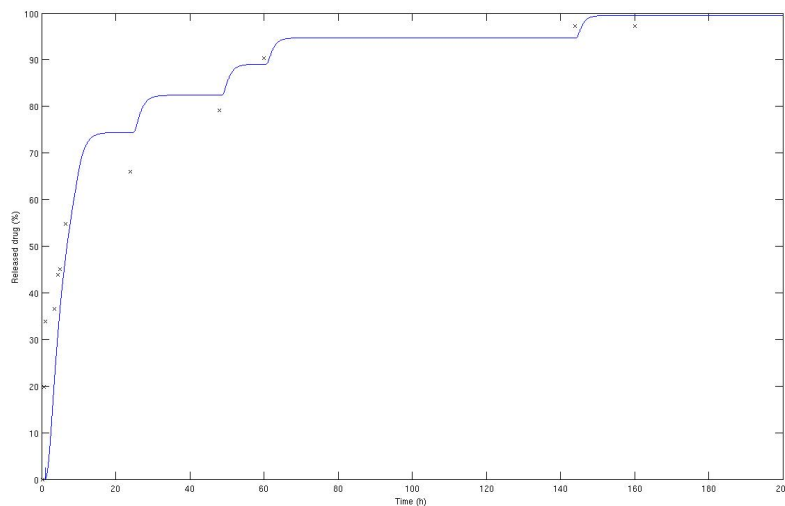
Figure 4.25: Computed drug release compared to experimentally measured drug release.

Channel model

For the channel model the best fit for the degradation of the collagen matrix seems to be with $\gamma_2 = 1/3$, $k_4 = 200$ and $A_C = 1/100$. The very small A_C is also indicating that there is something in the model is stopping it from slowing down as quickly as it should. The computed solution is plotted with the experimental data in fig. 4.26.

To find a good fit for the drug release was very difficult. The computed drug release was much to small compared to the experimental whenever the parameters were comparable with the parameters used to fit the curve for the degradation. However, with $\gamma_2 = 1/10$, $A_C = 1/100$ and $k_4 = 550^3$, we got the plot in fig. 4.27.

We could not find a good fit with the same parameters for the collagen degradation and the drug release, which may be due to the non-linearity in d. This theory is supported by the differences between the equations for C_A and C_E. While the equation for C_A contains $\partial_t d$ as well as d, the equation for C_E depends only on d, and not its derivative. This means that changing the equation for d will have a bigger impact on A_C than on C_E. Considering
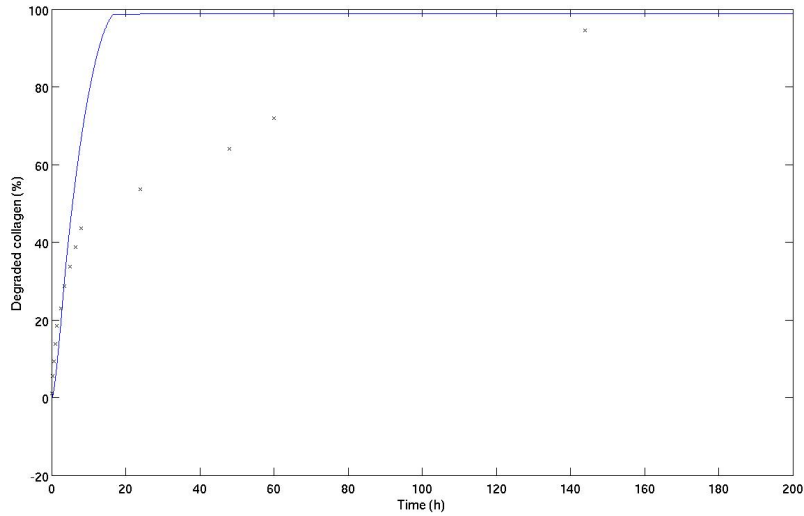
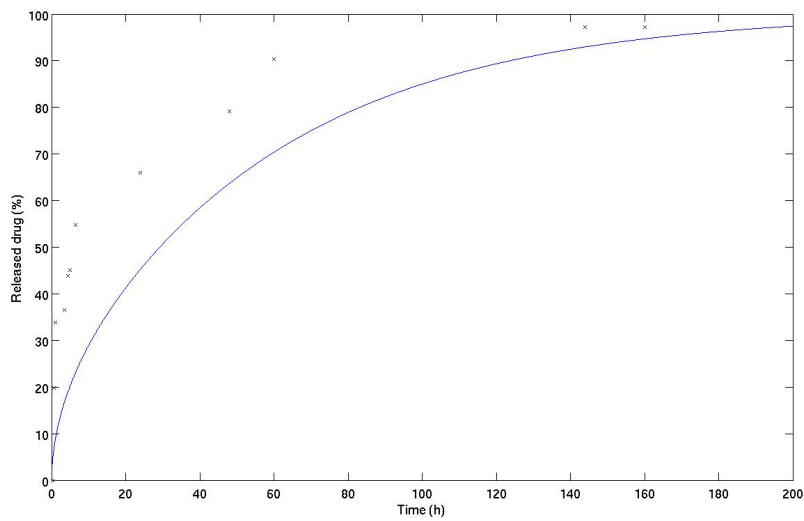Figure 4.26: Computed degradation compared to experimentally measured degradation.

Figure 4.27: Computed drug release compared to experimentally measured drug release.

the extremely big k_4 and small γ_2 that was needed to fit the drug release, this seems reasonable.

4.4.2   New data set (October 2013), non-cross linked collagen

Both data sets from October 2013 only contains data for the collagen degradation. The first mathematical modelling of a collagen implant with an evolving microstructure was done in [14]. Because this are of research is so new, this is the first opportunity to compare the fitted parameters for more than one set of data. They should not depend on the data, so it will be interesting to compare them.

Grain geometry

In the grain geometry we found the best fit to be for $\gamma = 0.375$ and $A_C = 1.5$. The plot for the computed solution is in fig. 4.28.
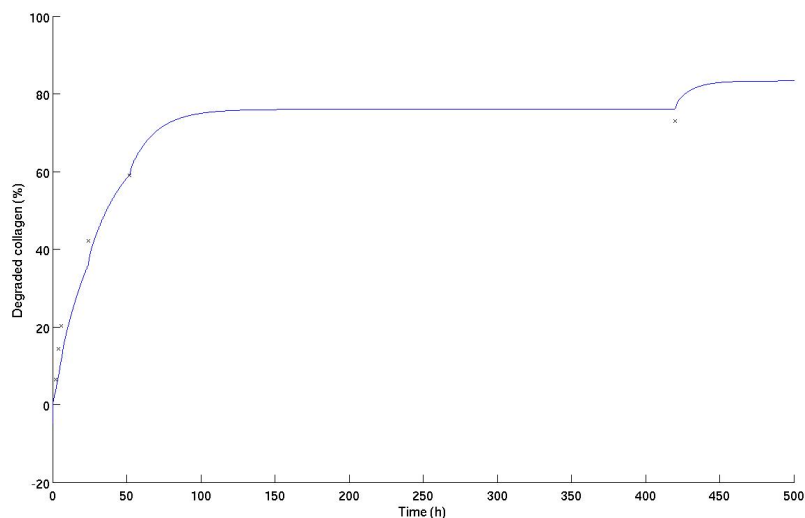

Figure 4.28: Computed degradation compared to experimentally measured degradation.

This fit is better than the fit we got with the data from [14]. The experimental data shows a much slower degradation towards the end than the previous data, which makes the model have a better fit.

Channel geometry

The best fit in the channel geometry has $\gamma = 1/3$, $A_C = 1/10$ and $k_4 = 50$, which can be seen in fig. 4.29.
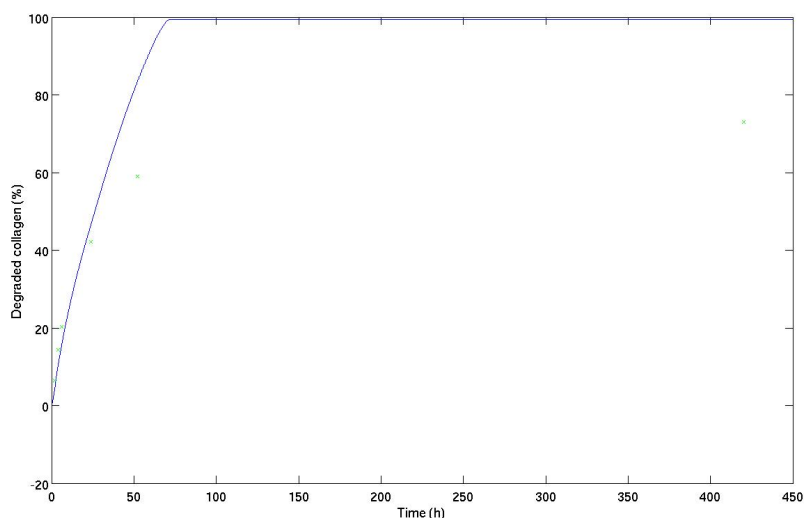

Figure 4.29: Computed degradation compared to experimentally measured degradation.

Again, we see that the model does not slow down quickly enough, but that the fit is pretty good in the beginning.

4.4.3   New data set (October 2013), cross-linked collagen

Collagen matrices are used for slow drug release, so anything that can slow the degradation down, is welcome. Enzymatic cross-linking is very promising, which makes it important that our models fit these experimental data as well.

Grain geometry

In the grain geometry, the best fit was found with $\gamma = 0.45$ and $A_C = 2$. This is shown in fig. 4.30.
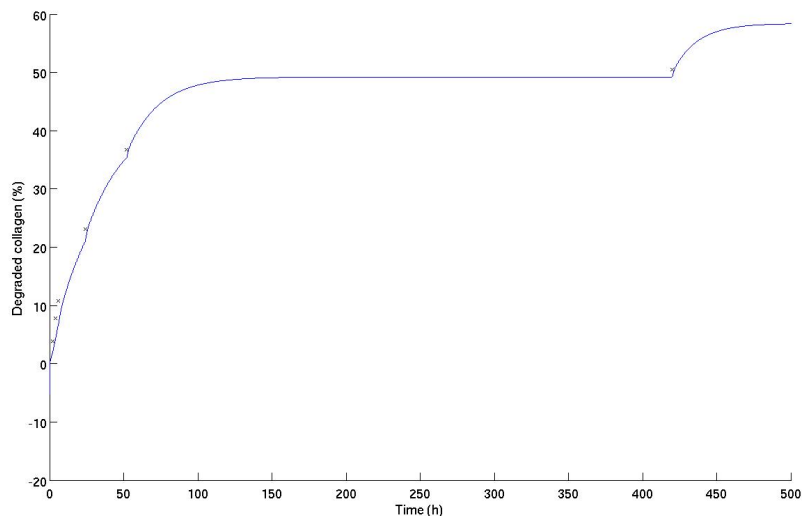

Figure 4.30: Computed degradation compared to experimentally measured degradation.

We see that this is a really good fit, with only small inaccuracies at the very beginning.

Channel geometry

The best fit for the experimental data with cross-linked collagen is shown in fig. 4.31.

Like all the plots from the model with channel geometry, it fits quite well in the beginning, but then continues to grow too rapidly.

For the new data set, there are no data for the drug release. This means that we could not compare the parameters for the degradation of the matrix with the drug release the way we could from the older data set. Considering the differences we found there, it would be very interesting to look at this in future research.
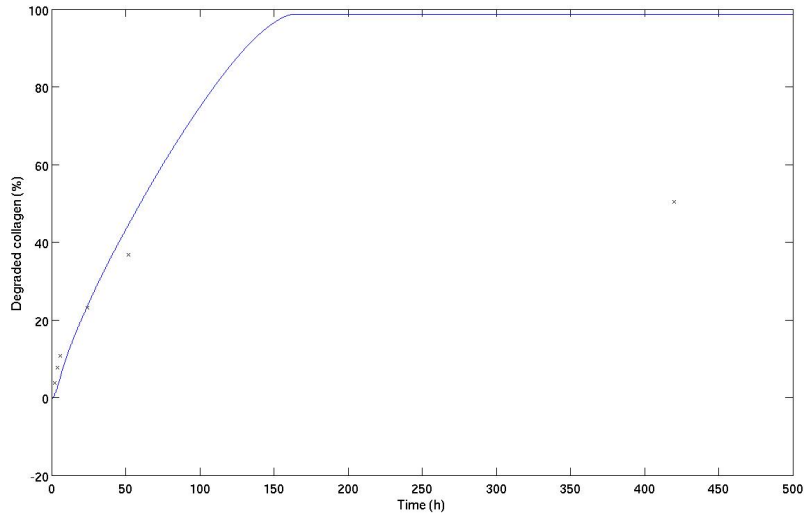
Figure 4.31: Computed degradation compared to experimentally measured degradation.

In this chapter we have seen how the implementations were tested to make sure they were correct, and how they were fitted to the experimental data. For the grain model we found that $\gamma = 0.45$ and $A_C = 1/16$ was the best fit for the data set from [14]. For the new data sets the fitted parameters were $\gamma = 0.375$ and $A_C = 1.5$, and $\gamma = 0.45$ and $A_C = 2$ respectively. We see that although they have some similarities, they are not identical enough to be certain of what the value should be. The channel model has three parameters that were $\gamma = 1/3$, $k_4 = 200$ and $A_C = 1/100$ for the first set of data. For the other two they were $\gamma = 1/3$, $k_4 = 50$ and $A_C = 1/10$, and $\gamma = 0.35$, $k_4 = 25$ and $A_C = 1/10$. These are rather similar, which is a good sign. For the first part of the collagen degradation at least, it looks like the model is a good fit, and the fact that the fitted parameters are almost constant for the two new data sets may indicate that they are correct.

Chapter 5

Conclusions and future work

This thesis looks at the mathematical modelling of slow drug release from collagen matrices, with a special focus on the degradation of the collagen matrix. This is a difficult, but mathematically very interesting problem. The collagen minirods show great potential in treating different diseases, and a good mathematical model can help to optimize the preparation of the minirods. Because it is a relatively new field of research, there is still much to explore. In this thesis we have tried to shed light both on an existing model, as well as creating a new model with different geometry.

We started with the grain model from [14], made it one-dimensional and chose a new method to implement it. This method was the TPFA, which is a control volume method, and thus has the advantage that it conserves mass. Afterwards we developed the channel model. We hoped that this would have fewer parameters to be fitted than the grain model, but this turned out not to be the case. If developed further, it would still be good to have this model, as it was developed in 1D, as opposed to the grain model. This would probably make it a better one dimensional model than the 1D version of the grain model, considering this was developed for two dimensions.

Both models were discretized and implemented in MATLAB. During the computations we discovered that the channel model is more stable than the grain model with the use of forward Euler for the equations with only temporal derivatives. The implementation of the grain model needed the

steps to be about three times shorter than the implementation of the channel model. Forward Euler has the advantage that the equations can be solved as only partially coupled, which allows for a much simpler implementation and fast calculations.

The implementations were tested theoretically as well as compared to experimental data. The first set of data we used was used previously in [14], but we also did some experiments to have a new set of data. The experiments were done for two types of collagen - both non-cross-linked and cross-linked. There is a clear difference in the time it takes to break down the two matrices, which is what we expected.

The solutions computed with the grain model in this thesis are not as good fits as the solutions from [14]. This is to be expected, as a two-dimensional model is better, and especially when the model was developed for a two-dimensional geometry. There are still some advantages with a 1D-model. It allows quicker computations and is easier to implement, which could be good for analyses. The change in method should not make a big difference, but it seems like we should have made some changes in the model when we changed dimensions. The natural place to start would be the porosity, as that depends heavily on the geometry, and therefore the number of dimensions. This may make the fit better. However, the model fits better for the new data, so it may be that there is an error in the data from [14]. Unfortunately, we have no new data for the drug release. In future research it would be good to look at how the parameters look like for this as well. If they are the same as for the collagen degradation, it could suggest an error in the data set. However, if they are very different again, it suggests am error in the model.

Generally, we see that the channel model has good fits at the beginning, but there is a problem with slowing it down towards the end. This may be due to too many simplifications, so a next step could be a 2D-model or more channels. After discovering that the model needed a non-linearity it seemed natural to have in the equation for d because the non-linearity in the grain model is in the equation for R. This should however be explored further, to find a mathematical explanation. It is also possible that there should be a non-linearity somewhere else, which may help the process slow down towards the end.

For future work it would be nice to have a new set of experimental data. Since the rate parameters k_1 and k_2 are so different in the sets we have, it would be useful to see which is right. That would also make it possible

to have more testing points, which would be good to know how well our model fits the experimental data. A new set of data should also consider the release of the drug. This is especially important for the cross-linked collagen matrices as there has been no modelling of this.

The new model with channel geometry needs more theoretical exploration. Both with the non-linear term, as mentioned before, and to find out why the process does not slow down. It would also be good to see it is possible to determine k_4 experimentally, as the fewer parameters that need fitting, the better.

References

[1] I Aavatsmark. Bevarelsesmetoder for elliptiske differensialligninger. *Studieretning beregningsvitenskap, UiB*, 2004.

[2] R A Adams. *Calculus*. Addison-Wesley, 1991.

[3] Peter Atkins and Julio De Paula. *Elements of physical chemistry*. Oxford University Press, 2013.

[4] J H-Y Chung, A Simmons, and L A Poole-Warren. Non-degradable polymer nanocomposites for drug delivery. *Expert opinion on drug delivery*, 8(6):765–778, 2011.

[5] G A Di Lullo, S M Sweeney, J Körkkö, L Ala-Kokko, and J D San Antonio. Mapping the ligand-binding sites and disease-associated mutations on the most abundant protein in the human, type i collagen. *Journal of Biological Chemistry*, 277(6):4223–4231, 2002.

[6] N Erlewein. Covalent linkage of proteins to collagen by tyrosinase. Master's thesis, Ludwig-Maximilians Universität München, 2012.

[7] W Friess. Collagen–biomaterial for drug delivery. *European Journal of Pharmaceutics and Biopharmaceutics*, 45(2):113–136, 1998.

[8] David Ronald Kincaid and Elliott Ward Cheney. *Numerical analysis: mathematics of scientific computing*, volume 2. American Mathematical Soc., 2002.

[9] I Metzmacher, F A Radu, M Bause, P Knabner, and W Friess. A model describing the effect of enzymatic degradation on drug release from collagen minirods. *European Journal of Pharmaceutics and Biopharmaceutics*, 67(2):349–360, 2007.

[10] Iris Metzmacher. *Enzymatic degradation and drug release behavior of dense collagen implants.* PhD thesis, lmu, 2005.

[11] L Michaelis and M L Menten. Die kinetik der invertinwirkung. *Biochem. z*, 49(333-369):352, 1913.

[12] F A Radu, M Bause, P Knabner, G W Lee, and W C Friess. Modeling of drug release from collagen matrices. *Journal of pharmaceutical sciences*, 91(4):964–972, 2002.

[13] FA Radu, M Bause, P Knabner, W Friess, and I Metzmacher. Numerical simulation of drug release from collagen matrices by enzymatic degradation. *Computing and visualization in science*, 12(8):409–420, 2009.

[14] N Ray, T van Noorden, FA Radu, W Friess, and P Knabner. Drug release from collagen matrices including an evolving microstructure. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 2013.

[15] S Ricard-Blum. The collagen family. *Cold Spring Harbor perspectives in biology*, 3(1), 2011.

[16] Tineke Røe. Modeling and simulation of concrete carbonation in 1-d using two-point flux approximation. 2013.

[17] A R Tzafriri, M Bercovier, and H Parnas. Reaction diffusion model of the enzymatic erosion of insoluble fibrillar matrices. *Biophysical journal*, 83(2):776–793, 2002.

[18] AR Tzafriri. Mathematical modeling of diffusion-mediated release from bulk degrading matrices. *Journal of controlled release*, 63(1):69–79, 2000.