universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

Modelling, Analysis, and Simulation of Actin-driven Wound Healing in Cells

angestrebter akademischer Grad

Magistra der Naturwissenschaften (Mag. rer. nat.)

Verfasser:	Stefanie Hirsch
Matrikelnummer:	0426472
Studienrichtung:	A 405 Mathematik
Betreuer:	Univ. Prof. Dr. Christian Schmeiser

Wien, am 11. März 2011

Abstract

This diploma thesis is devoted to the modelling, analysis, and simulation of wound healing within cells that use a meshwork of polymer filaments, called lamellipodium, to close a hole within the cell that either appears spontaneously or is created artificially by a microneedle.
It is part of a larger project concerned with the question of how these types of cells move along a surface, since the mechanisms are the same in both cases. The mechanisms which facilitate protrusion have been under investigation for some time in the scientific community, resulting in various models. The works of Christian Schmeiser, Dietmar Ölz and John Victor Small of the Institute of Molecular Biology at the Österreichische Akademie der Wissenschaften provide a mathematical framework for my diploma thesis. They consider the lamellipodium to consist of a meshwork of thin rods that are connected to each other and the substrate via proteins and enclosed by the cell membrane. In previous works ([3],[4], and [5]) they derived equations of motion resulting from various forces and considerations about the lifetime of the aforementioned proteins.

In the first chapter I will provide the biological background which will justify the modelling assumptions in chapter two. A strong assumption will be made about the geometry of the lamellipodium, namely that it is radially symmetric. This assumption implies that only the evolution of a single filament, the reference filament, is necessary to provide information about the evolution of the whole lamellipodium. In chapter three a numerical scheme will be derived which will be used to facilitate simulations. These simulations are done in Matlab, using a code developed by Dietmar Ölz that I have adapted for my purposes. The analysis of the outcome will show that certain modifications of the model will be necessary in order to produce feasible results. The last chapter is devoted to a first attempt in modelling the more complex situation of multiple directions of filaments in a lamellipodium.

Zusammenfassung

Diese Diplomarbeit behandelt die Modellierung, Analyse und Simulation von Wundheilung innerhalb von Zellen, die ein Netzwerk aus Polymerfilamenten (das Lamellipodium genannt wird) nutzen, um Löcher innerhalb der Zelle zu schließen. Diese Löcher können spontan auftreten oder mithilfe einer Mikronadel künstlich erzeugt werden.
Das Projekt, in dessen Rahmen die Arbeit verfasst wurde, befasst sich mit der Frage, wie sich diese Arten von Zellen entlang einer Oberfläche vorwärts bewegen. Die Mechanismen, die dieser Vorwärtsbewegung zugrunde liegen, sind schon lange Gegenstand wissenschaftlichen Interesses und haben zu verschiedenen Modellen geführt. Christian Schmeiser, Dietmar Ölz und John Victor Small vom Institut für Molekularbiologie an der Österreichischen Akademie der Wissenschaften geben mit ihrer Arbeit den Rahmen für meine Diplomarbeit vor. Sie gehen davon aus, dass das Lamellipodium aus dünnen Stäben besteht, die miteinander und mit dem Substrat über Proteine verbunden sind, und dass das Lamellipodium von der Zellmembran umhüllt wird. In früheren Arbeiten ([3],[4] und [5]) wurden Gleichungen hergeleitet, die aus dem Einfluss verschiedener Kräfte resultierten, sowie aus Überlegungen in Bezug auf die Lebenszeit der zuvor genannten Proteine.

Im ersten Kapitel werde ich einen Überblick über den biologischen Hintergrund geben, auch um die Modellierungsannahmen im zweiten Kapitel hinreichend zu motivieren. Dort wird die stark vereinfachende Annahme gemacht, dass das Lamellipodium radialsymmetrisch sei. Diese Annahme impliziert, dass die Evolution eines einzelnen Filaments, des Referenzfilaments, Information über das gesamte Lamellipodium enthält. Im dritten Kapitel wird ein numerisches Schema hergeleitet, auf dem Simulationen aufgebaut werden können. Diese Simulationen werden in Matlab durchgeführt, wobei ein Code verwendet wird, der von Dietmar Ölz entwickelt wurde, und den ich für meine Zwecke angepasst habe. Die Analyse der Simulationen zeigt, dass bestimmte Modifikationen im Modell notwendig sind, um realistische Ergebnisse zu erhalten. Das letzte Kapitel behandelt die ersten Versuche die komplexere Situation eines Lamellipodiums zu modellieren, das mehrere Richtungen für Filamente zulässt.

Danksagung

Allen voran danke ich meinen Betreuern Christian Schmeiser und Dietmar Ölz, sowie John Victor Small, die es mir ermöglicht haben, mich in dieses sehr interessante Gebiet der Zellbiologie einzuarbeiten. Großer Dank gebührt auch Nikolaos Sfakianakis, der mir immer wieder mit Ratschlägen, Hilfe und einer Tasse Kaffee zur Seite stand.

Besonderer Dank geht an meine Mutter und meinen Bruder, wie auch an meinen Freund David, für all die vielen kleinen Gespräche zwischendurch, die mich in meinem Ziel Mathematikerin zu werden, bestärkt haben.
Bei David möchte ich mich außerdem für das Korrekturlesen meiner Arbeit bedanken.

Diese Arbeit wurde durch ein Stipendium aus der Bestellung Christian Schmeisers zum Professor unterstützt, weshalb ich mich auch bei der Universität Wien recht herzlich bedanken möchte.

Contents

1 Introduction and Biological Background **7**

1.1 The Cytoskeleton . . . 7

1.1.1 Microtubuli . . . 8

1.1.2 Actin Filaments . . . 8

1.1.3 Intermediate Filaments . . . 9

1.2 The Process of Cell Movement . . . 9

1.3 Wound Closure in a Cell . . . 10

1.3.1 Experimental Setup . . . 10

1.3.2 Results . . . 10

1.4 The Process of Wound Healing in Different Cell Types . . . 13

2 Modelling **16**

2.1 Assumptions . . . 17

2.2 Deriving the Model . . . 18

2.3 Nondimensionalization . . . 21

2.4 Short Lifetime Limit, Variational Equation and the Euler-Lagrange Equations . . . 22

2.5 Finding an Initial Condition . . . 23

3 Discretization **26**

3.1 Numerical Scheme . . . 26

4 Implementing the Algorithm **30**

4.1 Simulations . . . 33

4.1.1 Appropriate Choice of Parameters . . . 33

5 Simulation Results **36**

5.1 The Scenarios . . . 37

5.1.1 Scenario 1, 2 and 3 - Full Problem at the Lower Fringe . . . 37

5.1.2 Scenario 4 - Full Problem at the Upper Fringe . . . 38

5.1.3 Scenarios 5, 6, and 7 - Trying to Counteract the Effects of Cross Links . . . 39

5.1.4 Results . . . 40

6 Towards a Model for Multiple Directions of Filaments **41**

6.1 Adaption of the Existing Model . . . 41

6.2 Outlook . . . 43

7 Bibliography **44**

8 Appendix A **45**
8.1 The Main Part of the Program 45
8.2 The Subprogrammes 48

9 Appendix B **55**

Chapter 1

Introduction and Biological Background

Cell migration is a vital part in the maintenance and development of multicellular organisms. Unfortunately, this process is not yet fully understood.
It has been shown, that for cells with a lamellipodium (see below), wound closure happens by the same mechanism as cell movement on surfaces. In order to gain more insight into cell motility, one can also look at the process of wound healing within a cell. Before going into more detail about wound closure, I want to explain some basic facts about the process of cell migration (though I will not go into detail about chemotaxis, the process by which a cell chooses a specific direction in reaction of a chemical signal outside the cell).
Cells migrate on surfaces by protruding at the 'front', and retracting at the 'rear' (as opposed for instance to swimming with 'tails', *flagellae*, for movement in a liquid). It has been shown that a main player in the protrusion and retraction process is actin polymerization. At the edges of the cell there is a thin sheet of a mesh work of actin filaments, termed *lamellipodium*.
Movement of the cells occurs when these filaments polymerize at the cell tip, and depolymerize at the 'inner' side of the lamellipodium, by dragging the cell body along.
This chapter shall provide an overview of the main structures and processes involved in cell motility and wound closure.

1.1 The Cytoskeleton

The following (chapter (1.1)) has been excerpted from [1]: The *cytoskeleton* is a dynamic structure that permeates the cell (except for the nucleus). It stabilizes its shape, holds the organelles in place (and moves them if necessary), and helps in separating the cell into daughter cells after cell division. It is made up of three main components: *actin filaments* (also termed *microfilaments*), *intermediate filaments*, and *microtubules* (*microtubuli*). Each of those is itself made up of long chains of (intertwined) proteins.

1.1.1 Microtubuli

Microtubuli are long, and relatively stiff cylinders of 23 nm diameter, consisting of chains of two subunits, α- and β-tubulin.
In animal cells they originate in MTOCs (microtubule organizing centers) attached to the sides of the nucleus (during interphase the spindle pole body also plays the role of an MTOC).
α- and β-tubulin form a dimer by attaching to each other. Many of those build a hollow cylinder, by subsequently connecting 13 pairs of dimers in a circular row, and packing these up one another. Within this cylinder, there are protofilaments, consisting of alternately arranged α- and β-tubulin monomers. Since the tubulin molecules have all the same orientation, the protofilament is polarized. On one side of the cylinder, tubulin polymerizes faster than on the other - this has lead to the notion of *plus ends* (faster polymerization) and *minus ends*.
Polarity is one of the main features of microtubuli, allowing them to facilitate vesicular transport, and (in some cells) to orientate the nucleus.
Another important feature is *dynamic instability*: every free tubulin dimer contains GTP that is hydrolyzed to GDP shortly after the tubulin dimer has attached to the cylinder. Molecules with GTP bind better to the structure than those with GDP. In a fast polymerization, molecules attach faster to the end than GTP hydrolyzes to GDP, so at the end of the microtubule one finds only GTP-subunits, forming a 'GTP-cap' (this cap prevents the cylinder from depolymerizing). At some point, hydrolysis catches up with polymerization, resulting in a catastrophic depolymerization process: Only GDP molecules are found at the end of the microtubule, but their attachment is much weaker than those of the GTP tubulin molecules, leading to rapid depolymerization and therefore shrinkage of the microtubule. After this catastrophic event a rescue event takes places, meaning that the polymerization starts anew.

1.1.2 Actin Filaments

Actin filaments are usually thinner, more flexible and shorter than microtubuli, but there are more of them present in the cell. They do not appear singularly, but in a mesh work or in bundles, which show a higher stability than single filaments.
An actin filament consists of two identical intertwined monomer chains, forming a double helix of 7 nm diameter. Similar to microtubule polymerization, actin monomers can be attached on both ends of the filament. At the plus end, the polymerization rate is higher than at the minus end. Also, depolymerization can happen at both ends.
Actin can be found everywhere in the cytoplasm. There are many actin binding proteins within the cell, most of which bind to filaments rather than to single actin monomers. Depending on the function of these proteins, actin filaments can be arranged as a mechanically stable mesh work, the *lamellipodium* (which plays an important role in cell motility and is located at the edge of the cell), in bundles of locally parallel filaments (resulting, for instance, in the formation of a *filopodium* - which is assumed to be a tactile sensor for the cell), or in contractile structures, enabling (like microtubules) vesicular transport.

An important phenomenon that can be observed with actin filaments is the process of *treadmilling*: Since at the plus end more polymerization happens than depolymerization and vice versa at the minus end, a single monomer seems to be moving along the filament from plus end to minus end.
This is an important factor in the mathematical modelling of cell motility, and will be referred to later.

1.1.3 Intermediate Filaments

Intermediate filaments provide mechanical support for the cell. The building process of an intermediate filament is as follows: two monomers are intertwined, forming a so called *coiled-coil* dimer. If arranged side by side and shifted slightly, two of those make up a tetramer. Out of these building blocks, chains are formed by assembling them into longer structures. These Structures are intertwined into a helical shape of approximately 10 nm diameter consisting of several of these chains.
As opposed to microtubuli and actin filaments, intermediate filaments do not have a plus end or a minus end, so they are not polarized. Also they do not show treadmilling.

1.2 The Process of Cell Movement

As mentioned in the introduction of this chapter, cells move by protrusion of the front and retraction at the rear (of course, 'front' and 'rear' only make sense relative to the direction in which the cell moves).
At the front of the cell, a thin sheet of an actin filament meshwork is found, 0.2-0.3 μm thick and several microns long, the lamellipodium (see above). The plus ends of the actin filaments point to the edge of the cell, and by polymerization, the cell membrane is pushed outwards. To stabilize the mesh work, there are *cross linker proteins* that bind to two different filaments, exhibiting mechanical resistance to stretching and twisting. *Integrins* are part of molecules which provide connection to the surface, thus enable adhesion and transfer of momentum (we will also call them *adhesions*). These are proteins which connect filaments with the surface through the membrane. They too are sensitive to stretching.

Movement is a result of the interplay of these factors:
By polymerizing at the front, new cross links and adhesions can be built, stabilizing the meshwork and connecting it to the surface. All filaments are subject to forces resulting from stretching and twisting the cross links and stretching the adhesions. In turn, each filament exerts forces on the meshwork, the cross links, and the adhesions by being bent or moved (as a result of the aforementioned forces), thus dragging the attached filaments with it, and in doing so, stretches and twists the cross links. Because of treadmilling, movement of integrins and cross linker proteins relative to the filament can be observed. This does not contribute to the dynamics, but plays an important part in modelling and simulating cell movement.

1.3 Wound Closure in a Cell

In some cells, for instance sea urchin coelomocytes, spontaneous opening and closing of wounds within the cell can be observed. It is assumed that this happens when the cell spreads out too thinly on the surface, and the cell membrane ruptures. In the *lamellum* (the region between the lamellipodium and the nucleus) a physical hole appears, that transverses through the cell membrane and connects to the surface of the substrate. In order to study the process of wound closure, one can produce similar wounds by mechanical manipulation with a microneedle. It has been shown that for wound closure in sea urchin coelomocytes, actin polymerization plays a crucial role ([2]).

1.3.1 Experimental Setup

In experiments conducted by Henson et al. ([2]) sea urchins, *Strongylocentrotus droebachiensis*, were collected from shore waters, coelomocytes were extracted, and put on cover slips. Cells were either treated with a microneedle, or left alone (in the latter case spontaneous opening of wounds has been observed). The process of wound opening/closure has been recorded using *time lapse methods*: pictures are taken every few seconds, and (in order to show the evolution of this process) are put together to a film.

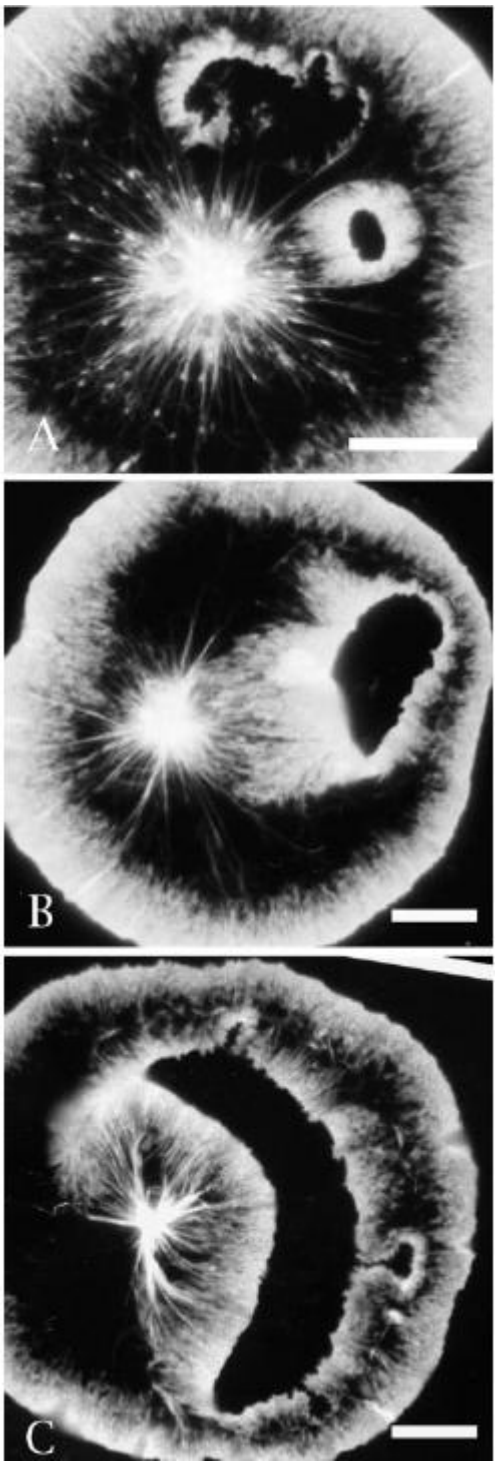
1.3.2 Results

When a wound appears (spontaneously or facilitated by a microneedle), the membrane ruptures and retracts. It is assumed that this retraction continues until there is physical resistance in the form of *actomyosin strings* (mechanically very stable polymer strings), that prevent the wound from opening further. A fringe of actin filaments appears at the wound margin, similar to the meshwork at the leading edge, and by polymerization at the edge of the wound this meshwork of actin filaments gets broader, at the same time smoothing out the wound edge.
The part of the wound margin that faces the cell center doesn't seem to move, whereas the part lying on the opposite side seems to move towards the center at a speed that is approximately the polymerization speed.

This is for two reasons: There is steady flow throughout the cell from the outer edge of the cell towards the center due to polymerization and according depolymerization. If there was no mechanism for closing the wound, the hole would seem to be drawn inwards almost radially. On the other hand, independent of this centripetal flow there is polymerization due to the actin meshwork around the wound. On the margin facing the center, the polymerization counteracts the effect of the flow, and therefore this side seems not to be moving (however, a ridge of accumulated actin is built up), whereas on the side facing the edge of the cell, this part of the wound margin seems to move inwards, with the flow, thus making the wound smaller. This also explains why the

speed of this inward-movement is roughly the polymerization speed.
After the wound has resealed, the freshly healed part of the membrane moves towards the center, again by centripetal flow ([2]).

2, Bar 10μm
These images were produced by *phalloidin staining*: phalloidin (a mycotoxin that binds to actin and inhibits depolymerization, thus 'freezing' the wound closure process) has been labelled with fluorescent markers, in order to make it visible.
In all three of the pictures labelled actin in white can be seen, appearing at the outer edge, forming the lamellipodium. In the center there is also actin associated with the nucleus, and actin fringes around small holes ((A) and (B)), and a large hole in (C) can be observed.



Picture (A) shows a wound at the top that has just opened and an actin fringe that surrounds it only partially. The wound at the right is more advanced in its healing process, the margin is already smooth.

In picture (B) a larger wound is depicted; note the bulk of actin on the side of the wound facing the cell center. Unfortunately, Henson et al. do not present an explanation as to why this occurs. Picture (A) seems to indicate that the actin is assembled from the side opposite the center, therefore this phenomenon cannot be explained saying that the large amount of filaments near the center is due to earlier assembly. It seems though, since there is almost no overlap of the two filamentous regions on the outer side of the wound, that filaments growing in opposite directions depolymerize before they abut.

Picture (C) shows a large convoluted wound and the actin fringe surrounding it. Again the part of this fringe facing the center is broader, and it seems as if the actin is drawn radially towards the center.

1.4 The Process of Wound Healing in Different Cell Types

In this section the focus is on how different cells use different mechanisms for closing a (spontaneous) wound.
B16 mouse melanoma cells, and *fish fibroblasts*, which have very different ways of using actin polymerization to close a wound or move in a specific direction, are often used for investigation.

John Victor Small and his group [1] at the IMBA [2] kindly provided videos of both cell types which show the opening and closing of (both spontaneous and induced) wounds.

In figure (1.1) a detail of a B16 melanoma cell in which a wound just opens can be seen. Picture 1 and 2 show this opening, pictures 3 and 4 depict the formation of an actin fringe around the wound margins, in pictures 5 to 8 we see how the wound gets smaller. We also see how the wound margin smoothens in this process. Notice how the actin fringe at the upper site gets thinner and thinner, until it is only a string (picture 9). Pictures 9, 10, and 11 show an interesting phenomenon: In picture 9 there is a protuberance at the lower end of the wound, in picture 10 this is smoothened out relative to the shape of the wound, and a darker region in the lamellipodium appears where the protuberance was shortly before. It can be assumed therefore that some sort of overlap of filamentous sheets happens in this region.
In the first four pictures there is a 'knot' in the middle which seems to be a physical barrier that inhibits further wound opening at this point (see also below). Picture 4 shows that around this knot the lamellipodium is very broad compared to the rest of the wound margin. Another important question (for the modelling and the simulations) is concerned with the dissolution of filaments into the surrounding region: Pictures 9 to 14 show that the lamellipodium still covers the initial wound area, whereas from picture 15 (here the wound area is only a fraction of the initial area) to the end, the lamellipodium seems to get dissolved.

[1] http://www.imba.oeaw.ac.at/research/vic-small/
[2] http://www.imba.oeaw.ac.at/

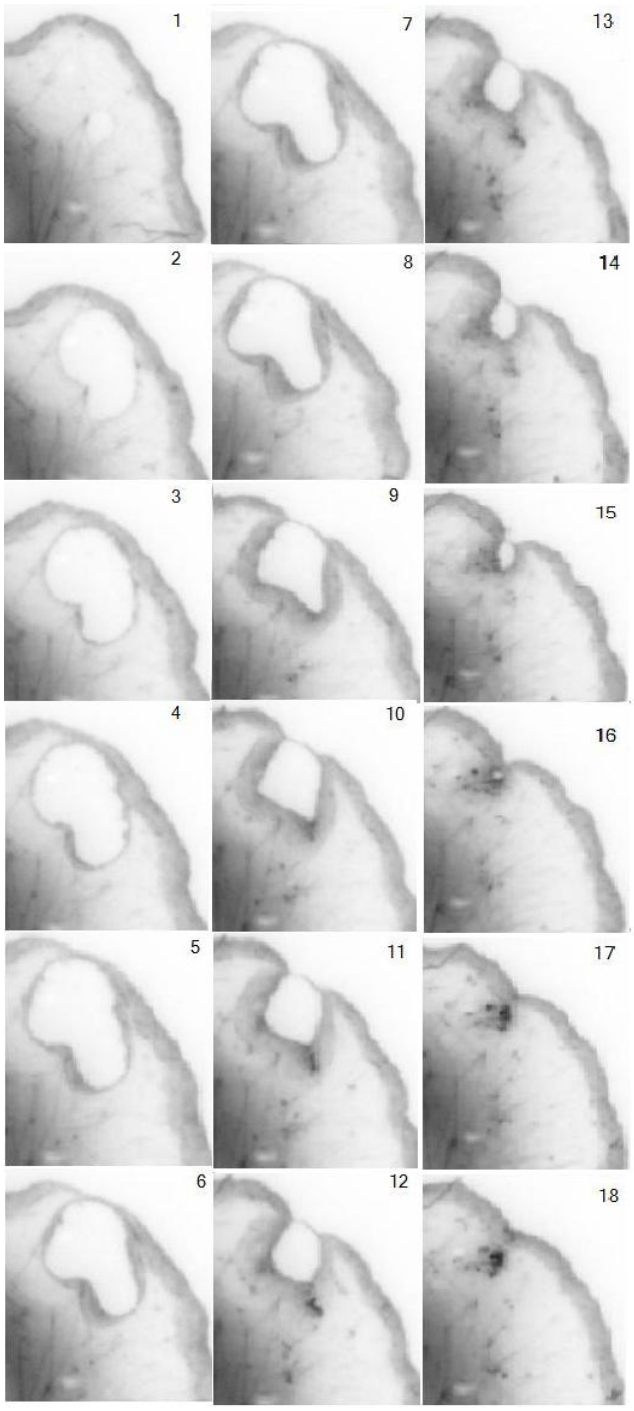


Figure 1.1: Time lapse series of a B16 melanoma cell showing the opening and closure of wound

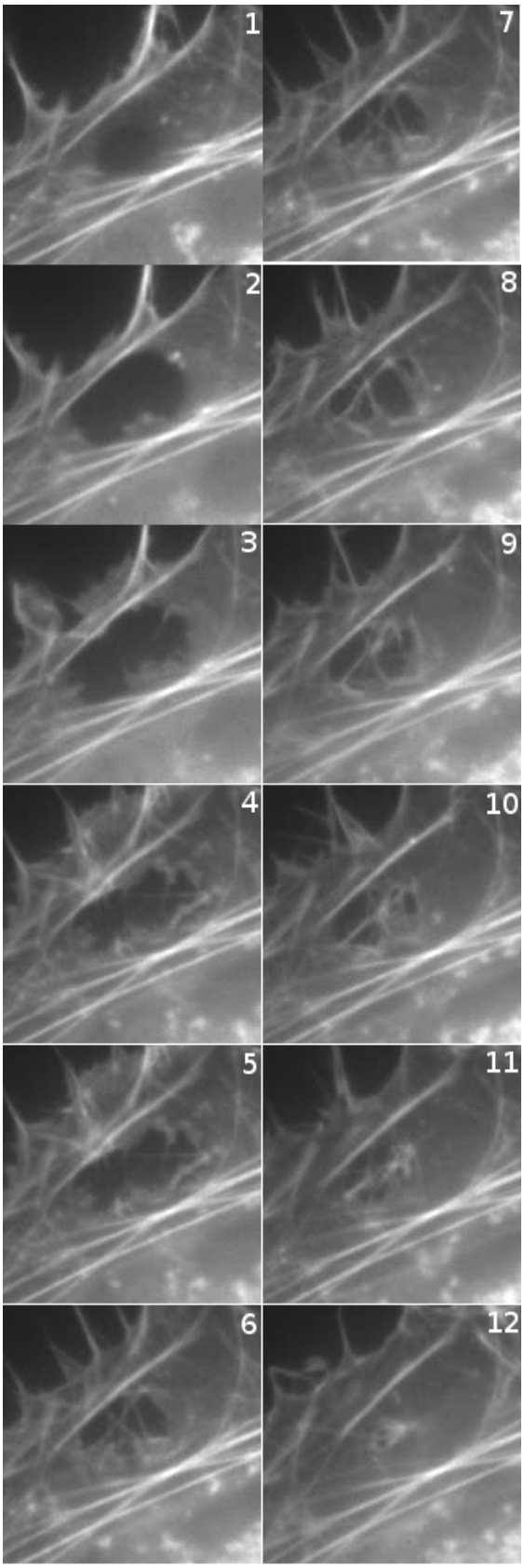


Figure 1.2: This series of pictures shows the opening (pictures 1 and 2) and closing of a wound in a fish fibroblast. Here the process of wound healing works differently than before. As can be seen in the first two pictures, the wound opens until a string of filaments provides a physical barrier. In the melanoma cells before there was an actin fringe which got tighter in a uniform way. Here the formation of filopodia can be observed that eventually transverse the wound(4, 5 and 6). Around the filopodia there is increased polymerization which shortens the time until the wound is closed.

Chapter 2

Modelling

In this chapter we will derive a model very similar to that presented by Dietmar Ölz and Christian Schmeiser ([3], [4], [5]). The process by which a wound heals within a cell is the same as that of cell movement (see chapter (1)), therefore it makes sense to choose a similar model.
They start on a microscopic level, describing cross link and adhesion molecule dynamics first in a discrete manner for finitely many filaments. Letting the number of filaments tend to infinity results in deterministic interpretations for previously stochastic equations for the evolution of cross links and adhesions. A minimization problem is introduced by assuming that the position functions for the filaments minimize a potential energy functional. Passing to a certain limit (assuming the lifetime of cross links and adhesion molecules is short compared to that of an actin monomer within a filament), the Euler-Lagrange equations (an equivalent for the equations of motion) are derived, where friction due to rapid turnover of cross links and adhesions appears.
It is important to note that their model describes movement for a fully developed lamellipodium, whereas in our case we are also interested in what happens at the beginning of the wound closure process. What do we expect to happen, if the lamellipodium is just beginning to grow?
Assuming all the necessary 'equipment' is present at the freshly sealed membrane around the wound margin (so called *nucleation centers*, providing the tools for growing filaments), we expect to see very short filaments after a certain time (the time corresponding to a few nucleation cycles). These filaments are too short to exert a notable pushing force on the membrane, therefore they will seem to grow away from the center of the wound as they are pushed away by the membrane. Nevertheless, we probably will already see adhesions and a few cross links. As more time passes, the filaments will grow longer, and we assume that at this stage the filaments still polymerize without subsequent depolymerization (therefore all filaments have the same length). If the filaments are long enough, there will be enough cross links to stabilize the meshwork, and also enough adhesions to root the lamellipodium, so we can expect the lamellipodium to push the membrane away, therefore decreasing the diameter of the wound.
So we first expect growth away from the wound center, into the cell, and after a time (depending on the diameter of the wound, of course), we expect the lamellipodium to grow inwards, towards the wound center.

2.1 Assumptions

The modelling assumptions are as follows (see also [3]):

A1: *After a certain time (see above) the circular wound will be surrounded by a lamellipodium which is two dimensional and has the topology of a ring, i.e. it lies between two closed curves.*

A2: *All actin filaments belong to one of two families, called clockwise and anti-clockwise. Filaments of the same family do not cross each other. Crossings of clockwise with anti-clockwise filaments are transversal. In the beginning, all barbed ends touch the leading edge of the lamellipodium, i.e. the inner curve of the previous assumption. Filaments are inextensible.*

A3: *Filaments polymerize at the barbed ends and depolymerize at the pointed ends with given polymerisation speeds.*

A4: *A cross link is a connection between a material point on a clockwise and a material point on an anti-clockwise filament. Cross links can be created spontaneously at the crossing between two filaments and they can also break. Creation and breaking are stochastic processes. There exists at most one cross link for any pair of filaments at any time.*

A5: *A μ-adhesion is a connection between a filament and the material within the cell surrounding the wound. μ-adhesions can be created spontaneously outside the wound margin and they can also break. Creation and breaking are stochastic processes. The number of μ-adhesions per filament length is restricted.*

A6: *An adhesion is a connection between a material point on a filament and a point on the substrate via a transmembrane linkage. Adhesions can be created spontaneously and they can also break. Creation and breaking are stochastic processes. The number of adhesions per filament length is restricted.*

A7: *The cell membrane exerts a constant force at the edge of the wound such that the wound tends to get bigger.*

A8: *The position of the filaments is determined by a quasistationary balance of elastic forces resulting from bending the filaments, stretching and twisting the cross links, stretching the μ-adhesions and adhesions, and from the force of the cell membrane around the leading edge.*

2.2 Deriving the Model

Since we are dealing with a symmetrical lamellipodium, we assume further that the position of any filament can be determined by rotation resp. reflection and rotation of a single reference filament. The position at time t of the reference filament is denoted by $z(t,s) = |z(t,s)|(\cos(\phi(t,s)), \sin(\phi(t,s)))$. s is the arc length parameter, $0 \leq s \leq L(t)$, so $|\partial_s z(t,s)| = 1$. $s = L(t)$ corresponds to the barbed end, $s = 0$ to the pointed end. Notice that the length of the reference filament is changing with time. Let $v(t)$ denote the polymerization speed, and $v_1(t)$ an average depolymerization speed. If we assume that at the pointed end only depolymerization happens, the s-value of a fixed actin monomer is decreasing over time (which means that this monomer is moving towards the pointed end), and we can provide a formula for $L(t)$:

$$L(t) = L_0 + \int_0^t v(t') - v_1(t') \; dt'.$$

In [3], [4], and [5] the parameter L was used to describe the maximum length of filaments, we use it here to denote the length of the reference filament (assuming that it has maximum length compared to the other filaments in the lamellipodium). Due to the inextensibility assumption, the path of a fixed actin monomer along the reference filament is given by $z(t, \sigma - \int_0^t v_1(t') \; dt')$. We use this to introduce Lagrange variables along a filament via

$$\sigma = s + \int_0^t v_1(t') \; dt'.$$

The matrices

$$R(\alpha) := \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}, \text{ and } D(\alpha) := R(\alpha) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

lead to the ansatz $F^+(t,\phi,s) = R(\alpha)z(t,s)$, $F^-(t,\alpha,s) = D(-\alpha)z(t,s)$ for the positions of clockwise resp. anticlockwise filaments, where α denotes the angle between a clockwise filament and the reference filament.
Let $\eta(t,s) : [0,\infty) \times [0, L(t)] \to [0,1]$ be the fraction of filaments, whose length at time t is at least $L(t) - s$ as measured from the barbed end, $\eta(t, L(t)) = 1$ at the beginning. At later times (when the radius of the wound gets smaller) we may assume that the number of filaments decreases at the leading edge, meaning that we have $\eta(t, L(t)) < 1$. $\partial_s \eta(t,s) \geq 0$ means that there are only pointed ends throughout the lamellipodium, i.e. all barbed ends touch the leading edge.
If we write η in terms of Lagrange variables, it is a decreasing function of time (we assume that only depolymerization happens at the pointed ends):

$$\partial_t \eta(t,s) - v_1(t) \partial_s \eta(t,s) \leq 0.$$

We introduce parameters describing stretching of cross links and adhesions and twisting of cross links: In this description, a denotes the age of a cross link or (μ-)adhesion molecule. For a cross

link of age a, connecting the reference filament to an anticlockwise filament with label α,

$$z(t-a, s+\int_{t-a}^{t} v_1(t')\ dt') = F^-(t-a, \alpha, s+\int_{t-a}^{t} v_1(t')\ dt')$$

must hold. This leads to $\alpha = -2\phi(t-a, s+\int_{t-a}^{t} v_1(t')\ dt')$.
$S_\mu = S_{adh} := |z(t,s) - z(t-a, s+\int_{t-a}^{t} v_1(t')\ dt')|$,
$S = z(t,s) - F^-(t, -2\phi^*, s)$, with $\phi^* := \phi(t-a, s+\int_{t-a}^{t} v_1(t')\ dt')$,
$T := \arccos(\partial_s z(t,s) \cdot \partial_s F^-(t, -2\phi^*, s)) - \phi_0$,
$T_0 = T(a=0)$, and ϕ_0 is the equilibrium angle for the cross links.
Let $\rho(t,s,a)$ be the density of cross links along a filament, depending on time, the position along the filament and the age of the respective cross link.
We assume that $\rho(t,s,a)$ satisfies

$$\begin{aligned} D_t\rho + \partial_a\rho &= -\zeta(S,T)\rho, \\ \rho(a=0) &= \beta(T_0)\left(1 - \int_0^\infty \rho\ da\right), \\ \rho(t,L,a) &= 0, \quad \rho(t=0) = \rho_I, \end{aligned}$$

where $D_t := \partial_t - v_1(t)\partial_s$. Similarly, let $\rho_{adh}(t,s,a)$ and $\rho_\mu(t,s,a)$ be the density of adhesions and μ-adhesions along a filament, then they satisfy

$$\begin{aligned} D_t\rho_{adh}(t,s,a) + \partial_a\rho_{adh}(t,s,a) &= -\zeta_{adh}(S_{adh})\rho_{adh}(t,s,a), \\ \rho_{adh}(a=0) &= \beta_{adh}\left(\bar\rho_{adh} - \int_0^\infty \rho_{adh}(t,s,a)\ da\right), \\ \rho_{adh}(t,L,a) &= 0, \quad \rho_{adh}(t=0) = \rho^I_{adh}, \end{aligned}$$

$$\begin{aligned} D_t\rho_\mu(t,s,a) + \partial_a\rho_\mu(t,s,a) &= -\zeta_\mu(S_\mu)\rho_\mu(t,s,a), \\ \rho_\mu(a=0) &= \beta_\mu\left(\bar\rho_\mu - \int_0^\infty \rho_\mu(t,s,a)\ da\right), \\ \rho_\mu(t,L,a) &= 0, \quad \rho_\mu(t=0) = \rho^I_\mu, \end{aligned}$$

where ζ, ζ_{adh}, ζ_μ are the destruction rates of cross links and (μ-)adhesions, β_{adh} is the constant creation rate of adhesions, $\bar\rho_{adh}$ and $\bar\rho_\mu$ are the maximum numbers of (μ-)adhesions along a filament, and the creation rate β_μ depends on the position of filaments:
$\beta_\mu = \beta_\mu(|z(t,s)|) = \bar\beta_\mu \in \mathbb{R}$ if $|z(t,s)| \geq R$ and 0 otherwise. As a next step we take into account assumption **A8** by introducing potential energy functionals. At any time $t \geq 0$ the filament position $z(t,s)$ is a minimizer of a potential energy functional:

$$U(t)[w] := U_{bending}(t)[w] + U_{scl}(t)[w] + U_{tcl}(t)[w] + U_{membrane}(t)[w] + U_\mu(t)[w] + U_{adh}(t)[w]$$

$$z(t,.) = \underset{|\partial_s w(s)|=1}{\operatorname{argmin}}\ U(t)[w],$$

where $w(s) = |w(s)|(\cos(\psi(s)), \sin(\psi(s)))$.

- *Stiffness of filaments*: $U_{bending}(t)[w] = \frac{\kappa^B}{2} \int_0^{L(t)} |\partial_s^2 w|^2 \eta \; ds$, where κ^B can be interpreted as the product of the bending stiffness of the reference filament and the total number $2n$ of filaments.
- *Membrane*: $U_{membrane}(t)[w] := M|w(t, L(t))|\eta(t, L(t))$ describes the energy contribution which stems from the constant force, with which the membrane pushes on a single filament times $2n$. By multiplying with $\eta(t, L(t))$ we take into account that after a certain time we may have fewer filaments touching the membrane. We assume that this contribution is only relevant within a circle of radius R around the center of the wound: $M = \bar{M}$, $\bar{M} \in \mathbb{R}$ for $|w(L(t))| \leq R$ and 0 otherwise.
- *Stretching the cross links*: $U_{scl}(t)[w] = \frac{\kappa^S}{2} \int_0^{L(t)} \int_0^{\infty} |S|^2 \rho \eta^2 |\partial_s \phi^*| \; dads$,
  $S = w(s) - D(2\phi^*)w(s)$, κ^S gives the Hooke constant describing the stretching stiffness of the cross links times the total number of filaments, and $|\partial_s \phi^*(t, s)|$ gives the number of crossings per unit length along the reference filament: $\alpha = -2\phi(t - a, s + \int_{t-a}^{t} v_1(t') \; dt')$ leads to $d\alpha = 2|\partial_s \phi^*| ds$ with $|\partial_s \phi^*| := -\partial_s \phi(t - a, s + \int_{t-a}^{t} v_1(t') \; dt') ds$.
- *Twisting the cross links*: $U_{tcl}(t)[w] = \frac{\kappa^T}{2} \int_0^{L(t)} \int_0^{\infty} |T|^2 \rho \eta^2 |\partial_s \phi^*| \; dads$,
  $T = \arccos(\partial_s w \cdot \partial_s D(2\phi^*)w) - \phi_0$, κ^T is the Hooke constant for the torsional stiffness of the cross links again multiplied by the total number of filaments.
- *Stretching the μ-adhesions*:
  $U_\mu(t)[w] = \frac{\kappa^\mu}{2} \int_0^{L(t)} \int_0^{\infty} |w - z(t - a, s + \int_{t-a}^{t} v_1(t') \; dt')|^2 \rho_\mu \eta \; dads$
  κ^μ is the Hooke constant describing the stretching stiffness of the μ-adhesions multiplied by $2n$.
- *Stretching the adhesions*:
  $U_{adh}(t)[w] = \frac{\kappa^A}{2} \int_0^{L(t)} \int_0^{\infty} |w - z(t - a, s + \int_{t-a}^{t} v_1(t') \; dt')|^2 \rho_{adh} \eta \; dads$
  κ^A is the Hooke constant describing the stretching stiffness of the adhesions multiplied by $2n$.

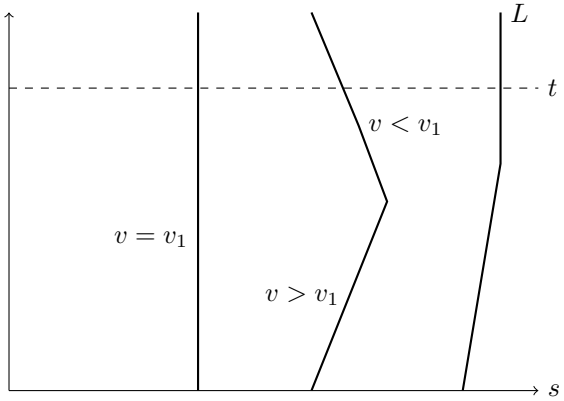


Figure 2.1: Possible maximum age depending on L(t)

In the terms for the cross links, adhesions, and μ-adhesions, the age integral could be restricted to the maximum age possible for a cross link or adhesion protein. It depends on the length of the filament at time t, which in turn depends on the polymerization and depolymerization velocities.

The figure above shows some of the possible scenarios for the evolution of the filament length over time.

2.3 Nondimensionalization

Now we nondimensionalize the equations and the energy functionals. We multiply each parameter and function with a reference value (the details can be found in [3]).
The main scaling assumption is that the parameter $\epsilon := \frac{\bar{a}v_0}{L}$ is small. It gives the ratio of $\bar{a}$, the reference parameter for the typical lifetime of cross links and adhesions, and L/v_0, the reference value for the typical time an actin molecule spends as part of the filament (v_0 is the reference value for the polymerization and depolymerization speed).
The scaled version of ρ satisfies

$$\begin{aligned} \epsilon D_t\rho(t,s,a) + \partial_a\rho(t,s,a) &= -\zeta(S,T)\rho(t,s,a), \\ \rho(a=0) &= \beta(T_0)\left(1 - \int_0^\infty \rho(t,s,a)\ da\right), \\ \rho(t,L(t),a) &= 0, \quad \rho(t=0) = \rho_I. \end{aligned}$$

For $\rho_{adh}(t,s,a)$ and $\rho_\mu(t,s,a)$ we get

$$\epsilon D_t\rho_{adh}(t,s,a) + \partial_a\rho_{adh}(t,s,a) = -\zeta_{adh}(S_{adh})\rho_{adh}(t,s,a),$$

with the boundary conditions

$$\rho_{adh}(a=0) = \beta_{adh}\left(1 - \int_0^\infty \rho_{adh}(t,s,a)\ da\right),\ \rho_{adh}(s=L(t)) = 0,$$

and

$$\epsilon D_t\rho_\mu(t,s,a) + \partial_a\rho_\mu(t,s,a) = -\zeta_\mu(S_\mu)\rho_\mu(t,s,a),$$

with similar boundary conditions

$$\rho_\mu(a=0) = \beta_\mu\left(1 - \int_0^\infty \rho_\mu\ da\right),\ \rho_\mu(s=L(t)) = 0.$$

The scaled versions of the energy functionals read:

- *Stiffness of filaments*: $U_{bending}[w] = \frac{1}{2}\int_0^{L(t)} |\partial_s^2 w|^2\eta\ ds$,
- *Membrane*: $U_{membrane}(t)[w] := M|w(L(t))|\eta(t,L(t))$,
- *Stretching the cross links*: $U_{scl}(t)[w] = \frac{\kappa^S}{2\epsilon}\int_0^{L(t)}\int_0^\infty |S|^2\rho\eta^2|\partial_s\phi^*|\ dads$, with $S = w(s) - D(2\phi^*)w(s)$, and $\phi^* = \phi(t-\epsilon a, s + \int_{t-\epsilon a}^t v_1(t')\ dt')$
- *Twisting the cross links*: $U_{tcl}(t)[w] = \frac{\kappa^T}{2}\int_0^{L(t)}\int_0^\infty |T|^2\rho\eta^2|\partial_s\phi^*|\ da\ ds$, with

$$T = \arccos(\partial_s w(s)\cdot\partial_s D(2\phi^*)w(s)) - \phi_0,$$

which can be written as

$$2\arccos(\partial_s|w|) + 2(\phi^* - \psi) - \phi_0.$$

- *Stretching the μ-adhesions*:
  $U_\mu(t)[w] = \frac{\kappa^\mu}{2\epsilon}\int_0^{L(t)}\int_0^\infty |w - z(t-\epsilon a, s + \int_{t-\epsilon a}^t v_1(t')\ dt')|^2 \rho_\mu \eta\ da\ ds$
- *Stretching the adhesions*:
  $U_{adh}(t)[w] = \frac{\kappa^A}{2\epsilon}\int_0^{L(t)}\int_0^\infty |w - z(t-\epsilon a, s + \int_{t-\epsilon a}^t v_1(t')\ dt')|^2 \rho_{adh} \eta\ da\ ds$

2.4 Short Lifetime Limit, Variational Equation and the Euler-Lagrange Equations

In the equations for ρ, ρ_{adh} and ρ_μ we let $\epsilon \to 0$. The equation for the cross link density now reads

$$\partial_a \rho = -\zeta(0, T_0)\rho, \quad \rho(a=0) = \beta(T_0)\left(1 - \int_0^\infty \rho\ da\right)$$

with the solution (we ignore the initial layer)

$$\rho = \frac{\beta(T_0)\zeta(0,T_0)}{\beta(T_0) + \zeta(0,T_0)} e^{-\zeta(0,T_0)a}.$$

For the μ-adhesions we obtain

$$\partial_a \rho_\mu = -\zeta_\mu(0)\rho_\mu, \quad \rho_\mu(a=0) = \beta_\mu\left(1 - \int_0^\infty \rho_\mu\ da\right)$$

which leads to

$$\rho_\mu = \frac{\beta_\mu \zeta_\mu(0)}{\beta_\mu + \zeta_\mu(0)} e^{-\zeta_\mu(0)a}.$$

By the same procedure we get for the adhesions

$$\rho_{adh} = \frac{\beta_{adh}\zeta_{adh}(0)}{\beta_{adh} + \zeta_{adh}(0)} e^{-\zeta_{adh}(0)a}.$$

For the energy functionals we look at the variational problem, and let $\epsilon \to 0$: At every time t the displacement $z(t,.)$ of the reference filament has to satisfy $\delta U(t)[z]\delta z = 0$ for all admissible variations δz, where $\delta U(t)$ is the variation of the total energy. An admissible variation has to satisfy $\partial_s z \cdot \partial_s \delta z = 0$.

- *Stiffness of filaments*: $\delta U_{bending}(t)[z]\delta z = \int_0^{L(t)} (\partial_s^2 z \cdot \partial_s^2 \delta z)\eta\ ds$, with the same limit as $\epsilon \to 0$.
- *Membrane*: $\delta U_{membrane}(t)[z]\delta z := M\eta(t, L(t))\frac{z(L(t))}{|z(L(t))|}\cdot\delta z(L(t))$, with the same limit as $\epsilon \to 0$.
- *Stretching the cross links*: In the limit $\epsilon \to 0$, a material derivative occurs:
  $\delta U_{scl}(t)[z]\delta z = \int_0^{L(t)} \mu^S |\partial_s \phi| D_t\phi (z^\perp \cdot \delta z)\eta^2\ ds$,
  $\mu^S = \frac{4\kappa^S \beta(T_0)}{\zeta(0,T_0)(\beta(T_0)+\zeta(0,T_0))}$.
- *Twisting the cross links*: In the limit $\epsilon \to 0$, we obtain

$\delta U_{tcl}(t)[z]\delta z = \int_0^{L(t)} \mu^T |\partial_s \phi|(2\arccos(\partial_s|z|) - \phi_0)(-\partial_s z^\perp \cdot \partial_s \delta z)\eta^2 \; ds$,
with $\mu^T = \frac{2\kappa^T \beta(T_0)}{\beta(T_0)+\zeta(0,T_0)}$.

- *Stretching the μ-adhesions*: $\delta U_\mu(t)[z]\delta z = \frac{\kappa^\mu}{2} \int_0^{L(t)} \int_0^\infty (z - z^*) \cdot \delta z \; \rho_\mu \eta \; dads$
  By a similar procedure as before we get for $\epsilon \to 0$
  $\delta U_\mu(t)[z]\delta z = \mu^M \int_0^{L(t)} D_t z \cdot \delta z \eta \; ds$, with $\mu^M = \frac{\kappa^\mu \beta_\mu}{\zeta_\mu(0)(\beta_\mu + \zeta\mu(0))}$.
- *Stretching the adhesions*: $\delta U_{adh}(t)[z]\delta z = \frac{\kappa^A}{2} \int_0^{L(t)} \int_0^\infty (z - z^*) \cdot \delta z \; \rho_{adh} \eta \; dads$
  Analogously we get for $\epsilon \to 0$
  $\delta U_{adh}(t)[z]\delta z = \mu^A \int_0^1 L D_t z \cdot \delta z \eta \; ds$, with $\mu^A = \frac{\kappa^A \beta_{adh}}{\zeta_{adh}(0)(\beta_{adh} + \zeta_{adh}(0))}$.

Collecting the terms leads to the variational equation

$$\eta M \frac{z}{|z|} \cdot \delta z|_{s=L(t)} + \int_0^{L(t)} [(\partial_s^2 z \cdot \partial_s^2 \delta z)\eta + \mu^M (D_t z \cdot \delta z)\eta + \mu^A (D_t z \cdot \delta z)\eta +$$
$$\mu^S |\partial_s \phi| D_t \phi (z^\perp \cdot \delta z)\eta^2 - \mu^T |\partial_s \phi|(2\arccos(\partial_s|z|) - \phi_0)(\partial_s z^\perp \cdot \partial_s \delta z)\eta^2] ds = 0$$

We introduce the Lagrange multiplier function $\lambda(t, s)$, and add the contribution

$$U_L[z, \lambda] := \int_0^{L(t)} \frac{\lambda}{2}(|\partial_s z|^2 - 1) ds$$

to the energy functional. Its z-variation reads

$$\delta U_L[z, \lambda]\delta z = \int_0^{L(t)} \lambda(\partial_s z \cdot \partial_s \delta z) \; ds$$

Combining this with the variational equation and a formal integration by parts leads to the Euler-Lagrange equation

$$\partial_s^2(\eta \partial_s^2 z) - \partial_s(\lambda \partial_s z) + \mu^M \eta D_t z + \mu^A \eta D_t z +$$
$$\mu^S |\partial_s \phi| \eta^2 D_t \phi z^\perp + \partial_s \left(\mu^T |\partial_s \phi| \eta^2 (2\arccos(\partial_s|z|) - \phi_0) \partial_s z^\perp \right) = 0$$

For the boundary conditions we get:

- s=0: $-\partial_s(\eta \partial_s^2 z) + \lambda \partial_s z - \mu^T \eta^2 (2\arccos(\partial_s|z|) - \phi_0)\partial_s z^\perp = 0$,
  $\partial_s^2 z = 0$.
- s=L(t): $-\partial_s(\eta \partial_s^2 z) + \lambda \partial_s z - \mu^T \eta^2 (2\arccos(\partial_s|z|) - \phi_0)\partial_s z^\perp + \eta M \frac{z}{|z|} = 0$,
  $\partial_s^2 z = 0$.

2.5 Finding an Initial Condition

We are interested in what happens right at the beginning of the formation of a lamellipodium, so we assume $\eta(t, s) = 1$ for $0 < s \leq L(t) = vt$, 0 otherwise. This condition means that at the beginning only polymerization at a constant speed v happens (and therefore $D_t = \partial_t$). Substituting that into

the equations we obtain

$$\partial_s^4 z - \partial_s(\lambda \partial_s z) + (\mu^M + \mu^A)\partial_t z + \tag{2.1}$$
$$\mu^S |\partial_s \phi| \partial_t \phi z^\perp + \partial_s(\mu^T |\partial_s \phi|(2\arccos(\partial_s |z|) - \phi_0)\partial_s z^\perp) = 0 \tag{2.2}$$

Boundary conditions:

- s=0: $-\partial_s^3 z + \lambda \partial_s z - \mu^T |\partial_s \phi|(2\arccos(\partial_s |z|) - \phi_0)\partial_s z^\perp = 0$,
  $\partial_s^2 z = 0$.

- s=L(t): $-\partial_s^3 z + \lambda \partial_s z - \mu^T |\partial_s \phi|(2\arccos(\partial_s |z|) - \phi_0)\partial_s z^\perp + M\frac{z}{|z|} = 0$,
  $\partial_s^2 z = 0$.

In order to do simulations based on this model it is necessary to have realistic initial conditions. One way to do that is not to start right from the beginning, but shortly afterwards. Furthermore, we will neglect the influence of cross links, since we assume the filaments to be too short to exhibit a noticeable meshwork. It also makes sense to assume that for short times the filament is rather straight. We make the ansatz

$$z(t,s) = R\begin{pmatrix}1\\0\end{pmatrix} + (s - L(t))\begin{pmatrix}w_1\\w_2\end{pmatrix}$$

with fixed R, $w = \binom{w_1}{w_2}$. Since $|\partial_s z| = 1$, we can conclude $w = \binom{w_1}{w_2} = \binom{\cos(\arg(\partial_s z))}{\sin(\arg(\partial_s z))}$. We substitute this into the simplified equation

$$\partial_s^4 z - \partial_s(\lambda \partial_s z) + (\mu^M + \mu^A)\partial_t z = 0.$$

From that we infer

$$\partial_s \lambda = -(\mu^M + \mu^A)v \quad \Longrightarrow \lambda = \lambda_0 - s(\mu^M + \mu^A)v$$

for some λ_0.
As mentioned in the introduction to this chapter, we expect the filaments to grow away from the wound center due to pushing by the membrane. Mathematically this means a balance of the forces from the membrane functional and the adhesion functionals:

$$\int_0^{L(t)} (\mu^M + \mu^A)(-vw)ds = -L(t)(\mu^M + \mu^A)vw = -t(\mu^M + \mu^A)v^2 w = -M\begin{pmatrix}1\\0\end{pmatrix}.$$

$z(t,s)$ has its barbed end on the x-axis for $t < T$ (T being the time the adhesion forces are greater than the membrane forces). We are only interested in the radial component of the force (where the pushing is strongest), so we get for $t = T$

$$T(\mu^M + \mu^A)v^2 w_1 = M \tag{2.3}$$
$$\Rightarrow \quad L_T = vT = \frac{M}{(\mu^M + \mu^A)vw_1}. \tag{2.4}$$

We will use the length and shape at time T for the starting values of our simulations.

Chapter 3

Discretization

In this chapter we will derive a numerical scheme which will again be very similar to that developed by Ölz and Schmeiser in [4], [5]. We will use the dimensional form of the model. However, since our model is different in some significant aspects, we have to adapt it properly. Also, we will not scale with respect to length and polymerization and depolymerization rates. This will later enable us to compare simulation results with videos taken of a specific wound in a B16 melanoma cell.

3.1 Numerical Scheme

In order to satisfy the inextensibility condition it is convenient to make an ansatz for $z(t,s)$:

$$z(t,s) = |z(t,s)|\begin{pmatrix}\cos(\phi(t,s))\\ \sin(\phi(t,s))\end{pmatrix} = R_0(t)\begin{pmatrix}\cos(\tilde{\gamma}(t))\\ \sin(\tilde{\gamma}(t))\end{pmatrix} + \int_0^s \begin{pmatrix}\cos(\gamma(t,s'))\\ \sin(\gamma(t,s'))\end{pmatrix} ds'$$

Then we have $|\partial_s z(t,s)| = 1$ for any time. $R_0(t)$ denotes the radius at the pointed end, and $\gamma_0(t)$ the angle with the x-axis there.
We introduce a time stepping procedure: Let

$$L^n = L^{n-1} + (v^{n-1} - v_1^{n-1})\Delta t, \quad n \geq 1,$$

where Δt is the time step, L^0 the length of the reference filament at $t = 0$, $v^n \Delta t \approx \int_{t-\Delta t}^t v(t')\, dt'$ and $v_1^n \Delta t \approx \int_{t-\Delta t}^t v_1(t')\, dt'$. Let N be the number of subintervals of L^n, $h^n := \frac{L^n}{N}$ be the spatial stepwidth, then we have $N+1$ evaluation points for z^n:

$$z_j^n \approx z(t^n, s_j^n) = z(n\Delta t, (j-1)h^n) \text{ for } n \geq 0,\ j = 1, \ldots, N+1.$$

Discretizing the parametrization from before we get

$$z_j^n = R_0^n \begin{pmatrix}\cos(\tilde{\gamma}^n)\\ \sin(\tilde{\gamma}^n)\end{pmatrix} + \sum_{i=1}^{j} \begin{pmatrix}\cos(\gamma_i^n)\\ \sin(\gamma_i^n)\end{pmatrix} h^n, \quad j = 1, \ldots, N+1$$

The energy contributions due to stretching and twisting the cross links and stretching the adhesions involve z-values of earlier positions. In order to introduce a time stepping procedure, we choose modified energy functionals (their variations at time $t - \Delta t$ have the same limit for $\Delta t \to 0$ as $\epsilon \to 0$ at time t):

$$\bar{U}^n(z)[w] := \bar{U}^n_{adh}(z)[w] + \bar{U}^n_{scl}(z)[w] + \bar{U}^n_{tcl}(z)[w] + U^n_{bending}(z)[w] + U^n_{membrane}(z)[w]$$

$$\begin{aligned}
\bar{U}^n_{adh}(z)[w] &:= (\mu^A + \mu^M) \int_0^{L^{n+1}} \frac{|w(s) - \hat{z}(s)|^2}{2\Delta t} \eta(t^{n+1}, s) ds \\
\bar{U}^n_{scl}(z)[w] &:= \mu^S \int_0^{L^{n+1}} |z(t,s)|^2 \frac{(\psi(s) - \hat{\phi}(s))^2}{2\Delta t} |\partial_s \hat{\phi}(s)| \eta(t^{n+1}, s)^2 ds \\
\bar{U}^n_{tcl}(z)[w] &:= \mu^T \int_0^{L^{n+1}} (\arccos(\partial_s w(s) \cdot D(2\hat{\phi}(t,s)) \partial_s w(s)) - \phi_0)^2 |\partial_s \hat{\phi}(s)| \eta(t^{n+1}, s)^2 ds
\end{aligned}$$

Here, $w(s) = |w(s)| \binom{\cos\psi(s)}{\sin\psi(s)} = \tilde{R}_0 \binom{\cos(\omega_0)}{\sin(\omega_0)} + \int_0^s \binom{\cos(\omega(s))}{\sin(\omega(s))} ds$, and μ^A, μ^M, μ^S, and μ^T are the dimensional counterparts for the factors in the nondimensionalized energyfunctionals from before. Using $v(t)$ and $v_1(t)$ as the polymerization and depolymerization rates, then $\hat{z}(t,s)$ we get by the following:

For $0 \le s \le L(t - \Delta t) - \int_{t-\Delta t}^t v_1(t') dt'$ we have

$$z\left(t - \Delta t, s + \int_{t-\Delta t}^t v_1(t') \, dt'\right),$$

and for $L(t - \Delta t) - \int_{t-\Delta t}^t v_1(t') dt' < s \le L(t)$:

$$z\left(t - \Delta t, L(t - \Delta t)\right) + \left(s - L(t - \Delta t) + \int_{t-\Delta t}^t v_1(t') dt'\right) \partial_s z\left(t - \Delta t, L(t - \Delta t)\right). \tag{3.1}$$

An analogous formula for $\hat{\phi}$ can be given.

In [4] Ölz and Schmeiser used $\hat{z}(t,s) = z(t - \Delta t, s + \Delta t)$, this was assuming equal (unit) polymerization and depolymerization speeds. In their case the filament maintained the same length for all times, and treadmilling had to be taken into account. Here we assume growing length of the filament in the beginning, and only after a certain time we introduce depolymerization.

Substituting the ansatz into the functionals and using $\partial_s w = \binom{\cos\omega(s)}{\sin\omega(s)}$ we obtain for twisting the cross links

$$T = \arccos(\partial_s w \cdot D(2\hat{\phi}) \partial_s w) - \phi_0 = \arccos\left(\binom{\cos(\omega)}{\sin(\omega)} \cdot \binom{\cos(2\hat{\phi} - \omega)}{\sin(2\hat{\phi} - \omega)}\right) - \phi_0 = 2\omega - 2\hat{\phi} - \phi_0,$$

and for the bending energy

$$|\partial_s^2 w(s)|^2 = |\partial_s \omega(s) \begin{pmatrix} \cos(\omega) \\ \sin(\omega) \end{pmatrix}|^2 = (\partial_s \omega(s))^2 .$$

We introduce a recursion

$$z^{n+1} = \underset{|\partial_s w|=1}{\operatorname{argmin}} \bar{U}(z^n)[w],$$

with $z^0 = z_I$.
In order to compute values for the energy functionals we have to interpolate previous positions of z^{n+1}. Furthermore, we have to take into account that the length L^n is different in each time step (due to polymerization and depolymerization), so the s-values at time $n\Delta t$ are not necessarily the same as at time $(n+1)\Delta t$.
One way to anticipate previous positions is to interpolate linearly between two evaluation points:

$$\hat{z}_j^n = z_j^n \left(1 - (N-(j-1))\,\frac{v_1^n \Delta t}{L^n} - (j-1)\frac{v^n \Delta t}{L^n}\right) + z_{j+1}^n \left((N-(j-1))\,\frac{v_1^n \Delta t}{L^n} + (j-1)\frac{v^n \Delta t}{L^n}\right), \tag{3.2}$$

for $j = 1, \ldots, N$. In the discretized version of the energy functionals (see below) we only need N values for $\hat{z}^n$. We assume that cross links and adhesions at newly polymerized sites do not exert a notable force on the meshwork, so we skip the value at $\hat{z}_{N+1}^n$. Analogously we define the first N positions of $\hat{\phi}_j^n$. For $\hat{\phi}_{N+1}^n$, which is needed for numerical differentiation in the discrete versions of the energy functionals (see below), we use formula (3.1):

$$\hat{\phi}_{N+1}^n = \phi_{N+1}^n + v^n \Delta t \left(\frac{\phi_{N+1}^n - \phi_N^n}{h^n}\right). \tag{3.3}$$

It is important to notice that this expression only makes sense, if $\hat{z}_j^n$ lies between z_j^n and z_{j+1}^n, so we need $0 \leq (N-(j-1))\frac{v_1^n \Delta t}{L^n} + (j-1)\frac{v^n \Delta t}{L^n} \leq 1$. From that we infer ($v^n$ and v_1^n are both nonnegative, so $\left(1 - (N-(j-1))\frac{v_1^n \Delta t}{L^n} + (j-1)\frac{v^n \Delta t}{L^n}\right)$ is also nonnegative)

$$\Delta t \leq \frac{L^n}{(N-(j-1))v_1^n + (j-1)v^n} \quad \forall\, j = 1, \ldots, N+1.$$

Since this has to hold for all j, we can guarantee this only if

$$\Delta t \leq \frac{L^n}{N max(v^n, v_1^n)},$$

or equivalently

$$\frac{h^n}{\Delta t} \geq max(v^n, v_1^n), \tag{3.4}$$

which can be considered a CFL condition. The grid speed needs to be at least the maximum of the polymerization and depolymerization speeds.

Discretizing the energy functionals we obtain

$$\bar{U}_{adh}(z^n)[w] = \frac{\mu^A + \mu^M}{2\Delta t} \sum_{j=1}^{N} h^{n+1}(w_j - \hat{z}_j^n)^2 \frac{\eta_j^{n+1} + \eta_{j+1}^{n+1}}{2}, \tag{3.5}$$

$$\bar{U}_{scl}(z^n)[w] = \frac{\mu^S}{2\Delta t} \sum_{j=1}^{N} |z_j^n|^2 h^{n+1}(\psi_j - \hat{\phi}_j^n)^2 \left| \frac{\hat{\phi}_{j+1}^n - \hat{\phi}_j^n}{h^{n+1}} \right| \left(\frac{\eta_j^{n+1} + \eta_{j+1}^{n+1}}{2} \right)^2, \tag{3.6}$$

$$\bar{U}_{tcl}(z^n)[w] = \frac{\mu^T}{2} \sum_{j=1}^{N} h^{n+1}(2\omega_j - 2\hat{\phi}_j^n - \phi_0)^2 \left| \frac{\hat{\phi}_{j+1}^n - \hat{\phi}_j^n}{h^{n+1}} \right| \left(\frac{\eta_j^{n+1} + \eta_{j+1}^{n+1}}{2} \right)^2, \tag{3.7}$$

$$U_{bending}(z^n)[w] = \frac{1}{2} \sum_{j=1}^{N} h^{n+1} \left(\frac{\omega_{j+1} - \omega_j}{h^{n+1}} \right)^2 \frac{\eta_j^n + \eta_{j+1}^n}{2}, \tag{3.8}$$

$$U_{membrane}(z^n)[w] = \eta_{N+1}^{n+1} M |w_{N+1}|, \tag{3.9}$$

where we interpolated η-values at the midpoints of s-intervals.

Chapter 4

Implementing the Algorithm

In this section we will implement the modified algorithm and do some further modifications to the original program provided by Dietmar Ölz.
We start by looking at the original program and give an overview:
It is written in Matlab, and its purpose is to simulate a fully developed lamellipodium with realistic parameters and draw conclusions about its long time behaviour.
Due to the choice of parametrisation, instead of cartesian coordinates for z^n it uses the radius at the barbed end, the angle with the x-axis at the barbed end, as well as the angle of the s-derivative, which is given as a $N+1$-vector (the first component corresponds to the pointed end and the last to the barbed end).
The minimization is with respect to these variables, although at several points throughout the programme cartesian coordinates are calculated and used to compute delays and for plotting. For the minimization itself a built-in function of Matlab is used.

The algorithm works as follows:

1. Choose specific parameters (radius of cell, equilibrium angle, bending stiffness, etc.).
2. Choose L, N, and Δt.
3. Compute η_j^n ('etafunc'), assume a linear decay from the pointed end, take averages.
4. Construct an initial condition ('computerad_phi0'): Provide values for radius at barbed end ('RL'), angle at barbed ('phiL') end and angles of s-derivatives. Subject them to forces due to twisting of cross links and bending the filament. Output is vector containing the angle at s-derivatives ('phi').
5. With these values compute cartesian coordinates of filament position ('computexieta'). Output is two $(N+2)$-vectors, 'xi' and 'eta'. The first component corresponds to the pointed end, where $s = 0$.
6. Compute angles of z^n ('argz'), and the respective s-derivatives ('argzp'), then drop the last component of the angle (which was necessary in order not to lose information in the process of numerical differentiation).
7. Start the time loop:
   For fixed number of iterations compute cartesian coordinates of the values from the previous

loop ('xi' and 'eta').

8. From these infer the angles and their s-derivatives for the factors in the energy functionals ('mu' and 'alpha').
9. Minimize with respect to radius at barbed end, angle at barbed end and angles of s-derivatives ('symfilament_by_minimisation'):
   Provide interpolated values for the old positions of z^{n+1}: For stretching and twisting the cross links compute cartesian coordinates, then the angles and the s-derivatives ('argzold' and 'argzoldp'). Interpolate linearly between the values at j and $j+1$.
   For the adhesions interpolate the cartesian coordinates ('computeoldpos').
   Put the angle at the barbed end, the radius at the barbed end, and the angles of s-derivatives into one vector, and use the built-in Matlab function 'fminunc' to minimize the energyfunctional with respect to this vector (the discretized versions of the energy functionals are contained in 'energyfunctional').
10. From the output of 'fminunc' retrieve the angle at the barbed end, the radius at the barbed end, and the angles of s-derivatives. These are the values at z^{n+1}.
11. Plot the lamellipodium ('plotsymfilament_evolution').
12. Start the next time loop.

The interpolation used for anticipating old positions at the next timestep is done directly within the energy functionals for stretching and twisting the cross links. For the adhesions a subprogramme has been written ('computeoldpos').

There are some changes that have to be made to adapt this programme for our purposes:

- *The minimization is with respect to the radius at the pointed end, the angle at the pointed end, and the angles of the s-derivatives*: This is to ensure that the lamellipodium grows in the direction of the barbed end.
- *The lamellipodium needs to point inwards, not outwards*: In order to let the lamellipodium point inwards instead of outwards, and using the parametrisation from before, we will change the sign in front of the s-integration in the subprogramme 'computexieta' (the angles of the s-derivatives of a clockwise filament with the barbed end at the x-axis are greater than π).
- *η needs to be changing with time*: For η we first assume equal length of filaments throughout the lamellipodium and only after a certain time we introduce graduation of lengths, starting from the pointed end. In the simulations, a minimum value of 0.1 is assumed at the pointed ends. For lower values the number of cross links and adhesions is so small that we may neglect their influence.
- *We choose specific initial conditions*: We start with a straight clockwise filament that has its barbed end at the x-axis. We provide values for the radius at the barbed end R_L^0, the filament length L^0, and the angles of the s-derivatives γ_i^0, and (using elementary trigonometry) compute the radius and the angle at the pointed end via:

$$R_0^0 = \sqrt{(R_L^0)^2 + (L^0)^2 - 2R_L^0 L^0 \cos(\pi - \gamma_{N+1}^0)}, \quad \tilde{\gamma}^0 = \arcsin\left(\frac{L^0 \sin(\pi - \gamma_{N+1}^0)}{R_0^0}\right).$$

- *We have an additional factor for the μ-adhesions, which is only relevant outside a certain radius*: the minimization is with respect to a vector, so we only add this factor in a component if the norm of the component is greater or equal to a certain radius.
- *We need to compute the delays differently, involving both polymerization and depolymerization at possibly different rates and possibly changing with time. This needs to be done such that at all times the aforementioned CFL condition is satisfied*: If we look at recursion (3.5), we see that delays appear in $\bar{U}^n_{adh}$ (namely $\hat{z}^n_j$), and in $\bar{U}^n_{scl}$, and $\bar{U}^n_{tcl}$ (via $\hat{\phi}^n_j$). Since there is also a numerical differentiation of these angles we need to provide $N+1$ components. So we need to have an additional program taking care of the delays in the angles. The first N we compute by the same procedure as for $\hat{z}^n_j$ (using formula (3.2)), and the component with number $N+1$ we get by formula (3.3).

4.1 Simulations

4.1.1 Appropriate Choice of Parameters

In order to be able to do realistic simulations we need to have realistic parameters and reference values.

As mentioned in chapter (1.4), I was provided with video material showing wound closure in different cells. I chose a video of a small, approximately symmetric wound in a B16 melanoma cell. A needle was poked into that cell resulting in the formation of a hole. In the first and second

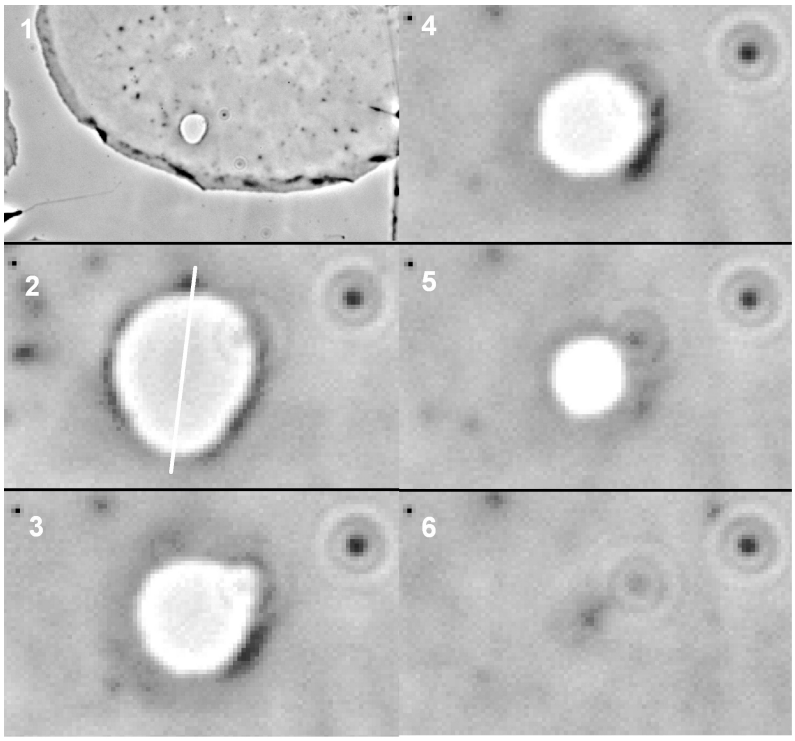


Figure 4.1: 1: B16 melanoma cell at t=0, 2-6: The wound heals

picture of figure (4.1) we see the whole cell and the wound at time $t = 0$. Picture 3 was taken at time $t = 0.4255$min, picture 4 at time $t = 0.5735$ min, picture 5 at time $t = 0.74$ min, and the last picture was taken at time $t = 0.894$ min, the time it took for the wound to heal.
The wound heals very symmetrically, so if we base our simulations on data from this specific wound, we can hope for a realistic outcome. The program used to retrieve data is called *Metamorph*. It has a feature that records pictures taken along a predefined line (see picture 2) from the beginning of the video to the end, and puts them together into a new picture below one another. I chose two lines along the axes where the wound expands the most.

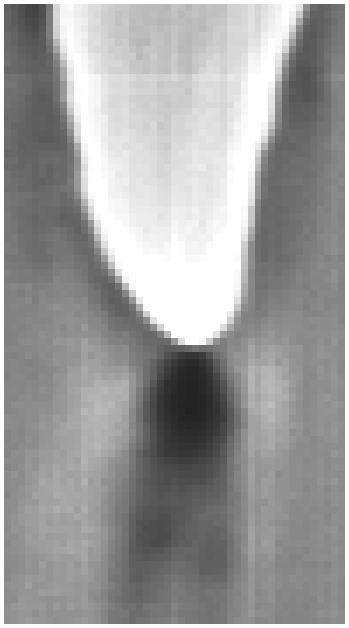


Figure 4.2: Evolution of a part of the lamellipodium along a thin band

In figure (4.2) we see such a picture: the top corresponds to $t = 0$, and the left to the upper fringe of the wound. The cone shows the movement of the wound margin over time. From this picture I was able to retrieve data such as the diameter of the wound, the length of the lamellipodium and protrusion and retraction rates at different times (if the wound would heal symmetrically we would see a straight symmetric cone). The goal is to reproduce the closure time, by providing the program with the diameter at the beginning, the length of a filament, and different protrusion and retraction rates, whenever they change significantly. We have to keep in mind though that these protrusion rates are not actual polymerization rates, since we have to assume that the filaments are not radial.

The other parameters of our model (except the parameters for the μ-adhesions and the membrane force), were taken from [5]. Also, in [5], $\bar{\rho}_{adh}$ and β_{adh} were determined to produce a stable lamellipodium. Integrins are rather stable, and therefore $\bar{\rho}_{adh}$, the maximum number of integrins along a filament, should be high. Since they were dealing with a lamellipodium in steady state, they took into account the strong displacement of monomers relative to the substrate in each timestep by taking a low value for $\bar{\rho}_{adh}$. In our case however, in some simulations we will take a larger value for we have a moving lamellipodium: $\bar{\rho}_{adh} = 0.67125$. In others we will take $\bar{\rho}_{adh} = 0.491$ to see if we can counteract some undesired effects.

Parameter	Value	Description
$\#F$	$90(R_L^0 + 0.2)\pi$ μm (varies)	Number of (anti-)clockwise filaments ([6], approximately 90 filaments cross a line of width $1\mu m$ drawn 0.2μm behind the wound edge with radius R_L^0)
$\eta(t, s)$	$(0.1 + 0.9\frac{s}{L})$ to 1	Fraction of filaments having at least length $L - s$
ϕ_0	$70°$	Equilibrium angle of cross links
β	1.3 sec^{-1}	Rate of cross link attachment
ζ	0.6 sec^{-1}	Rate of cross link detachment
β_{adh}	0.03 sec^{-1}	Rate of integrin attachment
ζ_{adh}	$0.012 \exp(\frac{z^{n+1}}{0.04\mu m})$ sec^{-1} (varies)	Rate of integrin detachment. z^{n+1} gives the position at time $t + \Delta t$. We will use $(v - v_1)\Delta t$ as an approximative value for z^{n+1}.
β_{adh}	0.03 sec^{-1}	Rate of integrin attachment

Energy functionals	Value	Description
$M/(4\pi^2)$	$900pN/\mu m$	Stretching elasticity of the membrane. Due to differences in the model for the membrane force compared to [3] we have an additional factor in M for the length of a filament.
$\kappa^B/(2\#F)$	$7 \times\ 10^{-2}$ $pN \times \mu m^2$	Flexural rigidity of one filament.
$\kappa^A/(2\#F)$	250 $pN/\mu m$	Stretching elasticity of one integrin-fibronectin complex.
$\kappa^S/\#F^2$	1000 $pN/\mu m$	Stretching elasticity of one cross link (filamin)
$\kappa^T/\#F^2$	0.1 $pN \times \mu m$	Torsional stiffness of one cross link

Chapter 5

Simulation Results

In this chapter the outcome of the simulations will be presented as well an analysis thereof.
The goal is to reproduce the time it takes for the wound to heal (0.894min), taking into account different protrusion and retraction rates, as well as the radius in the beginning, the closure time, and the thickness of the lamellipodium, which can be read from figure (4.2).
Seven scenarios were simulated with Matlab, two contain the parameters for the upper fringe, one for the lower fringe, and the other four were conducted at the lower fringe with slightly different parameters to determine their influence. In the following, these scenarios will be explained, the pictures can be found in Appendix B (9).
The pictures produced by the simulation show the state of the lamellipodium at different times. At $t = 0$ the simulation is started, with each filament having the same length. Two red circles at the wound margin and the outer fringe of the lamellipodium at $t = 0$ are drawn in each timestep to provide a reference for the evolution. The blue line is a visualisation of the reference filament in each timestep; it starts as a straight line with its barbed end at the x-axis. In formula (2.5) an initial condition for the length was derived, which corresponded to a balance of forces at the membrane. We incorporate this length by calculating the radius for the acto-myosin force from it. The number of evaluation points was chosen to be 40, together with a timestep of 0.002 min this ensured that violations of the grid-speed condition (3.4) were only for biological reasons.

5.1 The Scenarios

5.1.1 Scenario 1, 2 and 3 - Full Problem at the Lower Fringe

The values that can be read from figure (4.2) for the 'lower' fringe (farther from the cell center) are:

Start time	Stop time	Protrusion rate
0	0.426	2.358 μm/min
0.426	0.707	0.927 μm/min
0.707	∞	4.1772 μm/min
Start time	Stop time	Retraction rate
0	0.426	0
0.426	0.707	2.3178 μm/min
0.707	∞	7.658 μm/min
Other parameters	Thickness at $t = 0$	Radius at $t = 0$
	0.8862 μm	2.152 μm

One of the things that cannot be read directly from figure (4.2) is the length of a typical filament. We can only see the lamellipodium along a thin band in the direction of the radius, and we know that filaments tend to lie askew to the membrane, not normal. Therefore we have to assume an angle, and calculate the length of the reference filament from it.

For the first scenario an angle of 35° with the normal to the membrane was assumed, thus providing the equilibrium angle of filaments at the membrane right from the start. After a certain time (0.426 min) the η-function gradually is diminished starting from the pointed end. The time was chosen according to the pictures, which show a growing lamellipodium in the beginning, with (noticeable) depolymerization starting at time $t = 0.426$. Scenario 3 shows that the time for introducing gradual diminishing of η could have been chosen differently. For the simulations at the upper fringe, the starting time for the depolymerization was chosen to be $t = 0.1867$.
The pictures at time $t = 0.892$ and $t = 0.894$ show that the wound is closed at $t = 0.894$ (the pictures at time $t = 0$, $t = 0.3$, $t = 0.6$, and $t = 0.8$ have been provided to depict the evolution) (9.1). Unfortunately, though the exact time was reproduced, the simulation was not realistic: the filaments are bent too much. Filaments which have been observed in vivo or in vitro are only slightly bent.
The parameters which enable this effect are κ^S and κ^T. In order to get the numerical values for them, we had to multiply with the square of the filament number (see (4.1.1)). All other parameters only had a linear dependence in the filament number. This enforces high values for the respective energy functionals which in turn lead to filaments that are bent stronger. The effect is more pronounced with a smaller radius of the wound. A smaller radius in comparison to the thickness of the lamellipodium means a higher density of filaments at the wound margin. This means also that there are more cross links at the wound margin and therefore more force is applied

to ensure the equilibrium angle.

In scenario 2 we only assume 20° with the membrane normal, the rest of the parameter stays the same. Comparison of the pictures with scenario 1 shows that the influence of the initial angle is negligible. It only takes two timesteps longer to get to the same state. It also shows that different initial angles cannot counteract the effects of strong cross-linking, since in the end the reference filament is bent in the same (unnatural) way as in scenario 1. Picture (9.11) shows a detail of the wound at time $t = 0.7$. The reference filament is bent like an S-figure, and we can see that there is a slight drag away from the initial state. As an approach to counteract the effects of cross-linking, one could neglect connections to the substrate, thus letting the reference filament move more freely and adapt to forces from the cross links while being straight. The outcome of these simulations is shown in scenario 6.
Scenario 3 (9.12) was conducted to determine the influence of the η-function. The initial angle was again 35°, only this time all filaments have the same length throughout the simulation. Since it takes only one timestep more to reach the same state as in scenario 1, this influence can be neglected as well, and therefore η can be chosen arbitrarily.

5.1.2 Scenario 4 - Full Problem at the Upper Fringe

This scenario is divided into two subcases; in scenario 4a 9.16 we use 4.6671 μm/min as protrusion rate in the last phase of the simulation, in scenario 4b (9.20) it is 5.5492 μm/min. This is due to the fact that in figure (4.2) the part where the lamellipodium just closed is more than one pixel wide. For the lower fringe the right pixel was chosen, for the upper fringe both the right and left pixel were taken into account. Taking the right pixel produces a protrusion rate that is slightly too high (which leads to closure too early), and vice versa for the left pixel.
The values for the upper fringe are:

Start time	Stop time	Protrusion rate
0	0.597	1.4844 μm/min
0.597	∞	4.6671 μm/min (4a) resp. 5.5492 μm/min (4b)

Start time	Stop time	Retraction rate
0	0.1867	0
0.1867	0.3734	3.3905 μm/min
0.3734	0.597	3.129 μm/min
0.597	∞	4.6671 μm/min

Other parameters	Thickness at $t = 0$	Radius at $t = 0$
	1.1394 μm	2.532 μm

All other parameters are the same as in scenarios 1, 2, and 3.
As mentioned above, we have different closure times for the two scenarios, in scenario 4a it takes

0.928min to heal, in scenario 4b it takes 0.880min.

5.1.3 Scenarios 5, 6, and 7 - Trying to Counteract the Effects of Cross Links

For scenario 5 and 6 the term for the bending energy was made large (by a factor 1000). That way we ensure that filaments cannot be bent too much.
Scenario 5a (9.23) was a simulation for the lower fringe, 5b (9.26) at the upper fringe. The pictures for these scenarios show that the reference filament becomes more and more tangential. The polymerization rate and therefore the evaluation of old positions strongly depends on the angle with the membrane (we get higher polymerization rates for lower angles). At some point this leads to a violation of the grid speed condition (3.4), and therefore to the breakdown of the simulation. Scenario 5b breaks down at a steeper angle of the reference filament with the membrane than 5a, this is due to the shorter length of the filament at the upper fringe during most of the simulation. Scenarios 6a (9.29) and 6b (9.32) are an attempt to save scenario 5 by taking into account that we have a certain range for the values of $\bar{\rho}_{adh}$. The idea is to loosen the grip with the substrate in order to let the filament move more freely; the twisting forces would rotate the filament while the high stiffness kept them straight.

For scenario 6a, $\bar{\rho}_{adh} = 0.491$ was chosen, but the pictures show clearly that we only gain 0.002min. Because of this, for scenario 6b the adhesion forces were disregarded completely. There is only friction between the filaments and no connection to the substrate. This rather 'unphysical' simulation was done to show that even for very small adhesion forces the simulation breaks down.[1].

Scenarios 7a and 7b were simulated with the original energy terms, but η was diminished at the membrane in two different ways: for scenario 7a, it was assumed that if the density of filaments at the membrane gets too high, some filaments detach and are depolymerized instantaneously. This was implemented by making the slope of the length distribution η more shallow over time. Figure (9.38) shows the length distribution in the last timestep, the x-axis corresponds to possible s-values (assuming a maximum length of 8 μm for the filament), the y-axis to the fraction of filaments having at least length $L - s$. Interestingly, this produced the same results as the original simulation at the lower fringe. The filaments are again bent in an unrealistic fashion.
In scenario 7b detachment was simulated not by instantaneous depolymerization but by keeping the detached filaments in the meshwork. They are assumed not to polymerize and depolymerization happens at the same rate as for the attached filaments (this means that we also have barbed ends within the meshwork - which is considered unrealistic). Picture (9.41) shows that in this case the shape of the length distribution function is that of a hat function. Since there is less contribution by the cross links close to the membrane (the cross link energy terms depend on η

[1] The simulation was remarkable in another regard: in the first timestep, the membrane pushes the lamellipodium outwards, and since the mean angle between filaments is not the equilibrium angle there is further movement outwards to ascertain a low deviation from the mean angle between filaments. This phenomenon does not occur if notable adhesion forces are involved (above a very small threshold of 0.001 compared to the original value of the energy functional).

quadratically, whereas the other terms depend on η only linearly), the filament straightens out, and becomes tangential eventually, causing the breakdown of the simulation.
Scenarios 7a and 7b together show that the influence of cross links determine the shape of the reference filament. In scenario 7a, the overall number of filaments was diminished, but the relation of the cross link energy terms to the other energy terms along the filament roughly stayed the same compared to the relation in the scenario with the original η-function. That led to strong bending of the reference filament. In scenario 7b however, there is less cross link distribution closer to the membrane.

5.1.4 Results

The simulations show that the model depends on two main factors: the polymerization velocity (which was implicitly enforced on the system via the prescribed protrusion rate) and strong twisting forces of cross links. All the other factors have only little influence on the outcome (the membrane force could be used to push very hard, but this scenario is considered unrealistic by biologists [7]).
The problem with the model as it is can be summarized as follows: strong twisting leads to unrealistically bent filaments. However, we cannot neglect them, since only strong twisting ascertains a favorable angle at the membrane, in the sense that the polymerization velocities attain a realistic value of $1\mu m/\text{min}$ to $8\mu m/\text{min}$.
Nevertheless, the works of Ölz and Schmeiser ([4], [5]) showed that the symmetric model can reproduce realistic situations. Since the model presented here is a variation of this, we hope to be able to modify it in order to prevent unrealistically bent filaments. What we need is a way to ensure a certain angle at the membrane, while keeping the filaments straight. One way to do that can be to introduce new filaments via nucleation. The next chapter is devoted to the first attempts of deriving such a new model.

Chapter 6

Towards a Model for Multiple Directions of Filaments

So far, we have considered the lamellipodium to consist of two families of filaments; we named them clockwise and anticlockwise. If we only allow for two directions for the filaments, it may happen that in the process of wound closure the filaments become tangential to the circle marking the wound edge (see above). From that moment on, the filaments' polymerization does not close the wound any further, but will cause an overlap of filaments within the same family away from the wound edge (this phenomenon has not been shown in the simulations due to the violation of the grid speed condition (3.4)). It is therefore reasonable to consider more than two possible directions for the filaments. If we do this, however, we have to face a new problem: if a filament can polymerize in any direction, it could be that more than two filaments (even infinitely many) cross at any given point in space.

In order to prevent an unrealistically large pile of stacked filaments one has to restrict the number of filaments crossing at a given point in space. To do so, we need to know how many filaments are presently occupying this point, and we need approaching filaments to grow away from this point, should the number be high.
Furthermore, we get the same types of problems, if the filaments overlap in a small region (not necessarily a single point in space). So the questions arising are this: how many filaments may overlap at a given point in space? How far apart does the next knot of filaments need to be? Can this distance be reduced, if we don't have the maximum number of filaments overlapping at these two knots? The answers will certainly involve the bending stiffness of filaments.
Mathematically, the description of these situations will be done by deriving equations involving the curvature. This will also allow us to incorporate 3d-informations into our 2d-model.

6.1 Adaption of the Existing Model

We start with the discrete model by looking at the spatial situation at any given time. Although we allow the filaments to point in any direction (provided the barbed end is nearer to the membrane

than the pointed end), we assume the filament density to become smaller with distance to the membrane due to depolymerization.
Fix a point in space, $\mathbf{x} = (x, y) \in \mathbb{R}^2$. Let $K_h(\mathbf{x})$ be the ball around $\mathbf{x}$ with radius h, and $F_j'(\tilde{\mathbf{x}})$ shall denote the s-derivative of the j-th filament at $F_j(s)$, where s is chosen such that $F_j(s) = \tilde{\mathbf{x}}$, and j ranges form 1 to N.
By $w_j(\mathbf{x}) := \frac{1}{2h} l(K_h(\mathbf{x}) \cap F_j)$, $\mathbf{x} \in \mathbb{R}^2$, $j = 1, \ldots, N$, $F_j := F_j(s), s$, we measure for all filaments the ratio of the length of the filament within $K_h(\mathbf{x})$ to the diameter of this ball. Since the filaments can be bent, this ratio may be greater than one.
Let $\omega \in \mathbf{S}^1$, $s_j = min\{s' : F_j(s') \in K_h(\mathbf{x}) \cap F_j\}$ then

$$f_h(\mathbf{x}, \omega) := \frac{1}{h} \sum_{j=1}^{N} w_j(\mathbf{x}) \delta(F_j'(s_j) - \omega)$$

can be interpreted as the number of filaments that cross $K_h(\mathbf{x})$ and point in a certain direction ω, times a weight that measures how much space they occupy within $K_h(\mathbf{x})$.
If we integrate $f_h(\mathbf{x}, \omega)$ with respect to ω, we get a quantity that can be interpreted as a length density. This is the quantity that needs to be restricted:

$$\rho_h(\mathbf{x}) := \int_{S^1} f_h(\mathbf{x}, \omega) d\omega = \frac{1}{h} \sum_{j=1}^{N} w_j(\mathbf{x}) \leq \bar{\rho} \ \forall h \in \mathbb{R}.$$

Furthermore, since we assume that the density of filaments decreases with distance to the membrane, we have for ω pointing away from the membrane

$$f_h(\mathbf{x} + \lambda\omega, \omega) \geq f_h(\mathbf{x}, \omega) \ \forall \lambda > 0.$$

Dividing this by λ, and letting $\lambda \to 0$, we get

$$\omega \cdot \nabla_x f_h(\mathbf{x}, \omega) \geq 0.$$

As a next modelling step, we assume that $\mathbf{x}$ depends on s. It means that we identify a spatial position with a filament position. Since the s-derivative of a filament-position is a unit vector, we also may let ω be s-dependent: $\omega = \binom{\cos(\varphi(s))}{\sin(\varphi(s))}$. In fact, we have

$$\frac{d}{ds}\begin{pmatrix}\mathbf{x}\\ \omega\end{pmatrix} = \begin{pmatrix}\omega\\ \kappa(\mathbf{x}, \omega)\end{pmatrix} = \begin{pmatrix}\omega(\varphi)\\ \omega(\varphi)^{\perp}\partial_s\varphi\end{pmatrix} = \begin{pmatrix}\omega(\varphi)\\ \tilde{\kappa}\omega(\varphi)^{\perp}\end{pmatrix}, \tag{6.1}$$

where $\kappa(\mathbf{x}, \omega)$ is the curvature of a filament.
From the above considerations we may conclude that $f_h(\mathbf{x}(s), \omega(s))$ is increasing as a function of s:

$$L(f_h) := \frac{d}{ds} f_h(\mathbf{x}(s), \omega(s)) = \nabla_x f_h \cdot \omega + \tilde{\kappa} \nabla_\omega f_h \cdot \omega^{\perp} \geq 0.$$

If we further assume that we are looking at a small region such that there are no pointed ends, we may even say that we have equality in the statement above. Having no pointed ends, we have the

same density of filaments everywhere, and therefore the same value for f_h, if we let the ball with radius h 'travel' along a filament. From this we infer that f_h only depends on the local density of filaments, in other words

$$f_h(\mathbf{x}(s), \omega(s)) = c\eta(s),$$

where $\eta(s)$ denotes the collected length distribution of the filaments, and c is a constant (meaning that c only changes on a scale that is considerably larger than the one we consider here, and only transversal to the filaments).
Differentiating this equation we get

$$\nabla_x f_h \cdot \omega + \tilde{\kappa} \nabla_\omega f_h \cdot \omega^\perp = c\eta'(s) \tag{6.2}$$

From the structure of (6.1) and (6.2) we see that the solutions of (6.1) act as characteristics for (6.2).
In order to be able to solve this equation (for f_h) we need to know $\kappa(\mathbf{x}, \omega)$.

6.2 Outlook

The previous section shows a first attempt in involving multiple directions of filaments in the existing model. Further work on this needs to put an emphasis on finding reasonable functions for $\bar{\rho}$, and then on deriving equations of motion, while incorporating the equations for κ.

Chapter 7

Bibliography

1. B.Alberts, D.Bray, A.Johnson, J.Lewis, M.Raff, K.Roberts, P.Walter, **Lehrbuch der Molekularen Zellbiologie**, Wiley-VCH, 2.korr.Auflage 1998, ISBN 3-527-30493-2
2. Henson, Nazarian, Schulberg, Trabosh, Kolnik, Burns, McPartland: **Wound Closure in the Lamellipodia of Single Cells: Mediation by Actin Polymerization in the Absence of an Actomyosin Purse String**, Molecular Biology of the Cell Vol 13, 2002
3. Ölz, Schmeiser: **How do Cells Move? Mathematical Modelling of Cytoskeleton Dynamics and Cell Migration**, Cell mechanics: from single scale-based models to multiscale modelling,eds. A. Chauviere, L. Preziosi, and C. Verdier, Chapman and Hall / CRC Press, 2009
4. Ölz, Schmeiser: **Derivation of a model for symmetric lamellipodia with instantaneous cross-link turnover**, Arch. Rational Mech. Anal., published online first, 2010
5. Ölz, Schmeiser, Small: **Modelling of the Actin-cytoskeleton in symmetric lamellipodial fragments**, Cell Adhesion and Migration, 2/2(2008), 117 - 126
6. Köstler, Auinger, Vinzenz, Rottner, Small: **Differentially oriented populations of actin filaments generated in lamellipodia collaborate in pushing and pausing at the cell front**, Nature Cell Biology, Vol. 10, No. 3, March 2008
7. Vallotton, Danuser, Bohnet, Meister, Verkhovsky: **Tracking Retrograde Flow in Keratocytes: News from the Front**, Molecular Biology of the Cell, Vol. 16, 12231231, March 2005

Chapter 8

Appendix A

8.1 The Main Part of the Program

In this Appendix the code used in programming scenario 1 will be shown. The other scenarios were programmed with different parameters.

```
1  global  nfilaments nub num nus nut nuf
2  global N h deltat kappa4 omega L t phi0 RL0 svec Rold
3  global hfig e1 e2 e3 e4 e5 e6 omegavalue cfl radm
4  global  etafav etaf nplot groups  anglebegin
5  global   Mfactor v1 v  Rbegin muA muT muS angleend
6  maxframes=1530; %max number of loops
7  t=0; %time at the beginning, time is in minutes
8  k=0; %counter for the loops
9  count1=0; %counter for gradual diminishing of eta from the back
10 count2=0; %counter for gradual diminishing of eta at the front
11 %Space- and timestep
12 N=40; %number of evaluation points along the filament minus 1
13 deltat=0.002; %length of timestep
14
15 % turn on/off energy functionals
16 e1=1; %bending
17 e2=1; %membrane
18 e3=1; %adhesions
19 e4=1; %cross link stretching
20 e5=1; %twisting
21 e6=1; %mu-adhesion factor
22
23 %Initial condition
24 anglebegin=35;
25 phi= repmat(anglebegin*pi/180,N,1);%angles of s-derivatives with x-axis at the
26 %evaluation points; guarantees straight filament
27 RL0=17*10/79; %Radius of barbed end below
28
29 %Polymerization and depolymerization
30 %protrusion rate given->later: divide by cos(angle of s-derivative
31 %at the front)
32 protbegin=2.358;
33 protmiddle=0.01545*60;
```

```
protend=0.06962*60;
retrmiddle=0.03863*60;
retrend=0.12764*60;
v=protbegin/cos(anglebegin*pi/180);
v1=0;

nplot=322;  %nplot & groups relevant for plotting
groups=nplot/2;
nfilaments=floor(90*2*(RL0+0.2)*pi/2);  %number of filaments in a family, 90
    filaments cross a micron

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
omegavalue=70; %equilibrium angle for cross links
omega=omegavalue*pi/180;

Mfactor=0.01; %Membrane factor

%biological parameters
elasticity_integrins=250;
elasticity_crosslinks=1000;
torsional_stiffness_crosslinks=0.1;
rigidity_filaments=0.07;
elasticity_membrane=900;

kappa=rigidity_filaments*2*nfilaments;
kappa1=nfilaments^2*elasticity_crosslinks;
kappa1int=2*nfilaments*elasticity_integrins;
kappa2=nfilaments^2*torsional_stiffness_crosslinks;
kappa4=1;
kappa3=4*pi^2*elasticity_membrane;

zeta=0.6*60;
beta=1.3*60;
betaint=.03*60;
zeta0=0.6*60;
zetaint0=0.012*60*exp(deltat*(v-v1)/0.04);
maxrhoint=0.67125;

nub=1;
num=kappa3/kappa;
nut=kappa2/kappa;
nuf=kappa1int/kappa;
nus=kappa1/kappa;

muA=nuf*maxrhoint*betaint/(zetaint0*(betaint+zetaint0));
muM=e6*muA;
muS=4*nus*beta/(zeta*(beta+zeta));
muT=2*nut*beta/((beta+zeta));
Lm=Mfactor*num/((muM+muA)*v*cos(phi(end)));
L=7*.1266/cos(phi(end)); %length of filament in the beginning
h=L/N;
%radius and angle of the pointed end at t=0
R0=sqrt(L^2+RL0^2-2*L*RL0*cos(pi-phi(end)));
phi0=asin(L*sin(pi-phi(end))/R0);
```

```
%radius for the acto-myosin force
radm=R0-Lm*cos(phi(end));
preparefilamentplot; %provides frame for plotting

[xi, eta]=computexieta(phi, R0, phi0,h); %gives 2 (N+2)-vectors containing cartesian coordinates of the evaluation points of the filament; 1 component too many (last), due to numerical differentiation of argzold below

Rbegin=norm([xi(end),eta(end)]);%radius of barbed end at t=0;for plotting

svec=repmat(L,1,N+1);
etaf=etafunc2(svec,k,0); %etafunction (constant in the beginning)
%plot radii of barbed and pointed end in the beginning
xifront=xi(end);
etafront=eta(end);
xiback=xi(1);
etaback=eta(1);

[hfig]=plotsymfilament_evolution(phi, R0, phi0, k, sprintf('t=%.4f', t),xifront,
    etafront,xiback,etaback,L);
saveas(hfig,['test' num2str(k)],'jpeg');
while  k<maxframes
    %need to satisfy cfl-condition in each timestep
    cflt=h/max(v,v1); % also possible:L/(N*max(v1,v));
    cflN=L/(deltat*max(v1,v));
    cfl =(cflt>=deltat);
    if   cfl

        k=k+1;
        t=k*deltat;
        L =L+(v-v1)*deltat;
        h=L/N;

        %initialize non-constant eta depending on time
        svec=repmat(L,1,N+1);
        if (t>=25.553/60)
            count2=count2+1;
            svecb=0:h:L;
            svec=[svecb(end-min(count2,N):end),svec(min(count2,N)+1:end-1)];
        end
        etaf=etafunc2(svec,k,count1);
        etafav=(etaf(1:end-1)+etaf(2:end))/2;

        %measure protrusion
        [xi,eta]=computexieta(phi,R0,phi0,h);
        Rold=sqrt(xi(end)^2+eta(end)^2);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %THE MINIMIZATION
        [phi, R0, phi0]=symfilament_by_minimisation(phi, R0, phi0);
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %plot the lamellipodium
        [hfig]=plotsymfilament_evolution(phi, R0, phi0, k, sprintf('t=%.4f', t),
```

```
            xifront,etafront,xiback,etaback,L);
138         saveas(hfig,['test' num2str(k)],'jpeg');
139
140         %divide protrusion rate by cos(angle of s-derivative at front with the
                membrane), same for retraction rate at the back
141         [xi,eta]=computexieta(phi,R0,phi0,h);
142         phipointed=arcustangens(xi(1),eta(1));
143         phiL=arcustangens(xi(end),eta(end));
144         angleend=phi(end)+abs(phiL);
145         cosend=cos(angleend);
146         angleend=angleend/pi*180;
147         v=(protbegin*(t>=0&&t<25.553/60)+protmiddle*(t>=25.553/60&&t<42.391/60)+
                protend*(t>=42.391/60))/cosend;
148         v1=(retrmiddle*(t>=25.553/60&&t<42.391/60)+retrend*(t>=42.391/60))/cos((phi
                (1)+abs(phipointed)));
149         %modify velocity-dependent parameters in the energy-functionals
150         zetaint0=0.012*60*exp(deltat*(v-v1)/0.04);
151         muA=nuf*maxrhoint*betaint/(zetaint0*(betaint+zetaint0));
152
153     else
154         hfig=breakplot(cflt, cflN);
155         break;
156     end
157
158 end
```

8.2 The Subprogrammes

```
1  function [x]=arcustangens(xi,eta)
2  %deals with the problem of atan being not unique; calculates angle using
3  %quadrant, the jump in angle of height 2*pi is at the positive eta-axis (to
4  %keep differences in angles low)
5  atan1=atan(eta./xi);
6  a=(xi<0);
7  b=(eta>=0);
8  c=a.*b;
9  d=(eta<0);
10 e=a.*d;
11 x=atan1-pi*c-pi*e;
```

```
1  function hfig=breakplot(cflt,cflN);
2  global h_all fontsize
3  axes(h_all);
4  cla;
5  text(0.6, 0.6 , ['Timestep too large, computation not realistic, simulation stopped
       '], 'Fontsize', fontsize);
6  text(0.6, 0.55, ['Please choose either a timestep smaller than ' num2str(cflt,'%5.7
       f')], 'Fontsize', fontsize);
7  text(0.6, 0.5, ['or the number of evaluation points smaller than ' num2str(floor(
       cflN),'%5.0f')], 'Fontsize', fontsize);
8  hfig=gcf;
```

```
function [argzold]=computeoldargz(phi, R0, phi0)
%produces an N+1-vector
%Use current angles to interpolate old position,
%according to shift in s-variable due to polymerization and
%depolymerization
global N deltat L v v1
L_old=L-(v-v1)*deltat;
h_old=(L_old)/N;
[xi, eta]=computexieta(phi, R0, phi0, h_old);
argz=arcustangens(xi,eta);
w1=[N:-1:1]'*(v1*deltat/L_old);
w=[0:N-1]'*(v*deltat/L_old);
argzoldbasic=argz(1:end-1).*(1-w1-w)+argz(2:end).*(w1+w); %length N
argzold=[argzoldbasic;argz(end)+v/h_old*deltat*(argz(end)-argz(end-1))];
```

```
function [xiold, etaold]=computeoldpos(phi, RL, phiL)
%Use current position to interpolate old position,
%according to shift in s-variable due to polymerization and
%depolymerization
global L N deltat h v v1
L_old=L-(v-v1)*deltat;
h_old=L_old/N;
[xi, eta]=computexieta(phi, RL, phiL, h_old);
w1=[(N):-1:1]'*(v1*deltat/L_old);
w=[0:(N-1)]'*(v*deltat/L_old);
xiold=xi(1:end-1).*(1-w1-w)+xi(2:end).*(w1+w);
etaold=eta(1:end-1).*(1-w1-w)+eta(2:end).*(w1+w);
```

```
function [xi, eta] = computexieta(phi, R0, phi0, h)
%Computes cartesian coordinates, using radius at pointed end, angle at
%pointed end, and angles at s-derivatives
c=[0;(cos(phi))];
xi=R0*cos(phi0)-h*cumsum(c);
s=[0;(sin(phi))];
eta=R0*sin(phi0)-h*cumsum(s);
end
```

```
function [eng ] = energyfunctional(x)
%Contains the discretized versions of the energy functionals;
%The minimization is with respect to this function
%phi, R0, and phi0 (the angle at the s-derivatives,
%radius at pointed end and angle at pointed end)
%are to be minimized, whereas argzold, normzold, xiold, and etaold
%(the interpolated previous values/positions) remain fixed throughout the
    minmization
global N h omega deltat muA muT muS
global stretch twist engmembrane engbend
global etafav e1 e2 e3 e4 e5 e6 etaf num
global Mfactor argzold normzold argzoldp
global xiold etaold radm

phi=x(1:N);
R0=x(N+1);
```

```
phi0=x(N+2);

[xi, eta]=computexieta(phi, R0, phi0, h);

engbend=e1*1/2*diff([phi; phi(end)]).^2/(h)^2;
memnorm=norm([xi(end),eta(end)]);
%except for bending and membrane forces there is no contribution to the
%energy by the newly polymerized part of the filament
xi=xi(1:end-1);
eta=eta(1:end-1);
argz=arcustangens(xi,eta);
normz=sqrt(xi.^2+eta.^2);

%thinning out of eta at the membrane has to be taken into account, and
%membrane force only within a certain radius
rad=(norm([xi(end),eta(end)])<=radm);
engmembrane = -e2*Mfactor*num*memnorm*rad*etaf(end);

stretch=normzold.*(argz-argzold(1:end-1));

%fixed: argzold, argzoldp, omega, to be minimized: phi; stems from
%T=arccos(z_s.D(-2argz)*z_s)-omega
twist=normalizeangle(2*(phi-argzold(1:end-1))-omega);
stretch_substrate_squared=(xi-xiold).^2+(eta-etaold).^2;

%different friction coefficients for adhesions, depending on position
muaint=(1+e6*(normz>=radm));

engstretch_substrate=e3*muA/(2*deltat)*sum(stretch_substrate_squared.*etafav'.*
    muaint.*h);
engtwist=e5*muT/4*twist.^2.*abs(argzoldp).*etafav';
engstretch=e4*muS/(2*deltat)*stretch.^2.*abs(argzoldp).*etafav';

eng=engmembrane+sum((engbend+engstretch+engtwist).*etafav'.*h)+engstretch_substrate
    ;
```

```
function res = etafunc2(s,k,count1)
global L N  RL0 Rnew eta_rad
%Gradual diminishing from the front via kreal
d=0.1;
if count1==0
    kreal=0;
else
   if (k-count1<=N/2)
       kreal=k-count1;
   else
       kreal=floor(N/2);
   end
end

res=(1-d)*s/L*(1-kreal/N)+d;
```

```
function a = normalizeangle(x)
 a=mod(x+pi, 2*pi)-pi;
```

```
function [hfig]=plotsymfilament_evolution(phi, R0, phi0, k, tit, xifront,etafront,
    xiback,etaback,L);
%Plots the lamellipodium by using the reference filament;
%Copies of the  reference filament are turned and switched to span the
%whole lamellipodium
%Furthermore, certain parameters are put on the screen in order to easily
%compare simulations
global hfilaments  h_all RL0 heta nfilaments N angleend
global  etaf nplot groups v fontsize v1 anglebegin deltat f
global  Mfactor  e1 e2 e3 e4 e5 e6 omegavalue
h=L/N;
[xi, eta]=computexieta(phi, R0, phi0, h);

figure(f);
%plot relevant information
axes(h_all);
cla;

text(0.8,0.5,['The hole should be closed at time t=0.894'],'FontSize',fontsize);
text(0.6 ,0.93 ,['Loop # ' num2str(k,'%11.0f') '   Timestep ' num2str(deltat,'%11.5
    f') 'min, grid speed ok, # of evaluation points: ' num2str(N,'%4.0f')], '
    FontSize',fontsize);

text(0.6 ,0.87 ,['Bending energy: ' num2str(e1,'%11.3f')], 'FontSize',fontsize);
text(0.6 ,0.84 ,['Adhesion energy: ' num2str(e3,'%11.3f')], 'FontSize',fontsize);
text(0.6 ,0.81 ,['Acto-myosin energy: ' num2str(e6,'%11.3f')],'FontSize',fontsize);
text(0.6 ,0.78 ,['Cross link stretching: ' num2str(e4,'%11.3f') ], 'FontSize',
    fontsize);
text(0.6 ,0.75 ,['Cross link twisting: ' num2str(e5,'%11.3f') ], 'FontSize',
    fontsize);
text(0.62,0.72 ,['Equilibrium angle: ' num2str(omegavalue,'%11.3f')],'FontSize',
    fontsize);
text(0.6 ,0.66 ,['Membrane energy: ' num2str(e2,'%11.3f')], 'FontSize',fontsize);
text(0.62,0.63 ,['Membrane force: ' num2str(Mfactor,'%11.7f')], 'FontSize',fontsize
    );

text(0.6 ,0.57 ,['Initial condition:'],'FontSize',fontsize);
text(0.62,0.54 ,['Radius at the beginning: ' num2str(RL0,'%11.3f') ', angle at the
    beginning: ',num2str(anglebegin,'%11.3f')], 'FontSize',fontsize);
text(0.6 ,0.51 ,['Polymerization speed: ' num2str(v,'%11f') ], 'FontSize',fontsize)
    ;%'                Phase ' num2str(phase,'%1.0f')
text(0.6 ,0.48 ,['Depolymerization speed: ',num2str(v1,'%11f') ], 'FontSize',
    fontsize); %, initialized at time ',num2str(v1time,'%11.0f')
text(0.6 ,0.45 ,['Radius of barbed end: ' num2str(norm([xi(end), eta(end)]),'%11.7f
    ')], 'FontSize',fontsize);
text(0.6 ,0.42 ,['Radius of pointed end: ' num2str(norm([xi(1),eta(1)]),'%11.7f')
    ], 'FontSize',fontsize);
%text(0.6 ,0.39 ,['Protrusion. ' num2str(protrusion, '%3.6f') 'micron/min'],'
    FontSize',fontsize);
text(0.6 ,0.36 ,['Filament length: ' num2str(L,'%11f') ', # of filaments: ' num2str
    (nfilaments,'%11.0f')], 'FontSize',fontsize);

text(0.8,0.78,['Blue line: reference filament'],'FontSize',fontsize);
```

```
40 text(0.8, 0.66,['Angle at barbed end with membrane ' num2str(90-angleend,'%3.2f')],
       'FontSize',fontsize);
41
42 %plot etafunction
43 axes(heta);
44 cla;
45 title(sprintf('Length distribution'));
46 set(gca,'DataAspectRatio',[.9 .9 .9]);
47 xlim([0 8]);
48 ylim([0 1.2]);
49 set(gca,'XGrid','on');
50 set(gca,'YGrid','on');
51 hold on;
52 plot(0:h:L,etaf,'LineWidth',1.9,'Color','red');
53 hold off;
54
55 %plot the lamellipodium
56 axes(hfilaments);
57 cla;
58 title(tit);
59 set(gca,'DataAspectRatio',[.9 .9 .9]);
60 maxlim=3.8;
61 xlim([-maxlim maxlim ]);
62 ylim([-maxlim maxlim]);
63 set(gca,'XGrid','on');
64 set(gca,'YGrid','on');
65 hold on;
66 m2=100;
67 p=2*pi/m2*(0:m2);
68
69 plot(norm([xi(end),eta(end)])*cos(p),norm([xi(end),eta(end)])*sin(p), 'Color','blue
       ', 'LineWidth', 1.9);
70 plot(norm([xifront,etafront])*cos(p),norm([xifront,etafront])*sin(p), 'Color','red'
       , 'LineWidth', 2.0);
71 plot(norm([xiback,etaback])*cos(p),norm([xiback,etaback])*sin(p), 'Color','red', '
       LineWidth', 2.0);
72
73 % groups needs to be a prime and nplot a multiple of it
74 groupsize=nplot/groups;
75 dbeta=2*pi/nplot;
76 offset=0;
77 for j=1:groups
78     sind=(etaf/max(etaf) >= (j-1)/groups);
79     for i=0:(groupsize-1)
80         beta=offset+i*dbeta*groups;
81         RM=[[cos(beta), -sin(beta)]; [sin(beta), cos(beta)]];
82         turnedcoord=RM*[xi(sind)';eta(sind)'];
83         plot(turnedcoord(1,:), turnedcoord(2,:), 'Color', [0.5 0.5 0.5], 'LineWidth
               ', 0.5);
84         turnedcoord=RM*[xi(sind)';-eta(sind)'];
85         plot(turnedcoord(1,:), turnedcoord(2,:), 'Color', [0.5 0.5 0.5], 'LineWidth
               ', 0.5);
86     end
87     offset=offset+dbeta*11;
```

```
88 end
89 % plot the reference filament
90 plot(xi, eta, 'Color','blue', 'LineWidth', 2.0);
91 hold off;
92 drawnow;
93
94 hfig=gcf;
```

```
1  function preparefilamentplot;
2  %Provides frame for the plots, scales the size of the frame according to
3  %the screensize
4  global hfilaments  f figsize h_all fontsize heta hxivec g hfilament
5  fontsize=8;
6  scrsz = get(0,'ScreenSize');
7
8  f=figure('units','inches','Position',[0 0  scrsz(3)/10 scrsz(4)/10], 'Name', 'Actin
       -skeleton','Visible','Off');
9  get(0,'children')
10 figure(f);
11
12 h_all = axes('Position',[0 0 1 1],'Visible','off');
13 figsize=[0 0  scrsz(3)/10 (scrsz(4))/10];
14
15 hfilaments=axes('Position', [0.03 0.05 0.52 0.89]);
16 heta=axes('Position',[0.6 0.08 0.32 0.29]);
17 set(heta,'Fontsize',fontsize);
18 set(hfilaments, 'FontSize', fontsize);
19 set(h_all, 'FontSize', fontsize);
```

```
1  function [phi, R0, phi0] = symfilament_by_minimisation(phi, R0, phi0)
2  %Computes the argmin of the energyfunctionals with respect to radius at
3  %pointed end, angle at pointed end, and angles of s-derivatives
4  %computes interpolated previous positions in cartesian coordinates and
5  %angles, which are needed in the function 'energyfunctional';
6  %xiold, etaold, normzold, and argzold are to remain fixed;
7  %the built-in Matlab function 'fminunc' is used to carry out the
8  %minimization
9  global argzold xiold etaold N h normzold argzoldp L deltat v v1
10 h_old=(L-(v-v1)*deltat)/N;
11 %need to compare the angle that has to be minimized with the interpolated
12 %old angle
13 argzold=computeoldargz(phi,R0,phi0); %length N+1
14 argzoldp=diff(argzold)/h;
15
16 %need to compare xi/eta-values that have to be minimized with interpolated
17 %old xi- and eta-values
18 [xiold, etaold]=computeoldpos(phi, R0, phi0); %length 2N
19
20 [xi,eta]=computexieta(phi,R0,phi0,h_old);
21 normzold=sqrt(xi(1:end-1).^2+eta(1:end-1).^2);
22
23 options = optimset('Display','iter','LargeScale','on','MaxFunEvals', 15000, 'TolX',
       10^(-30) ,'TolFun', 10^(-30));
24 x0=[phi; R0; phi0];
```

```
25 [x] = fminunc(@energyfunctional,x0,options);
26 phi=x(1:N);
27 R0=x(N+1);
28 phi0=x(N+2);
```

Chapter 9

Appendix B

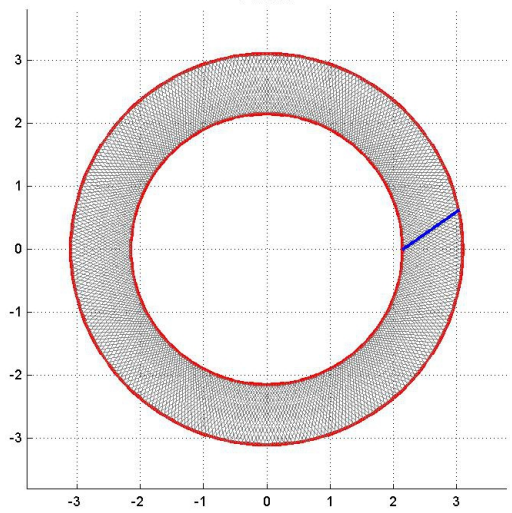


Figure 9.1: Scenario 1 at time t=0

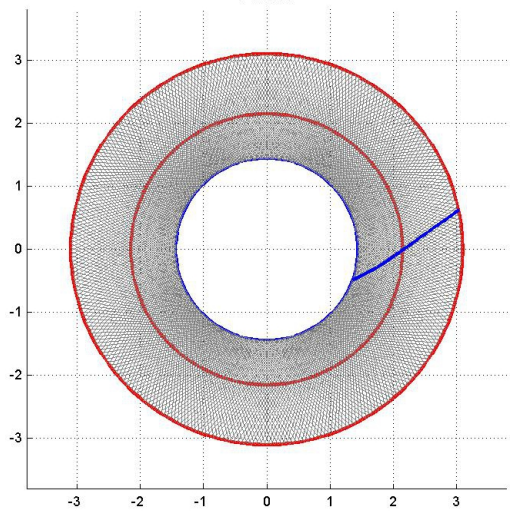


Figure 9.2: Scenario 1 at time t=0.3

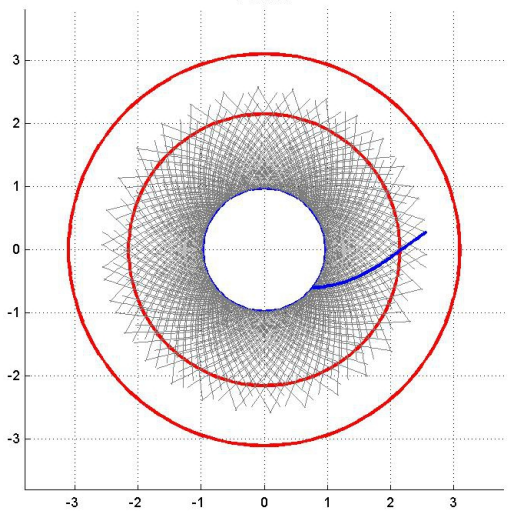


Figure 9.3: Scenario 1 at time t=0.6

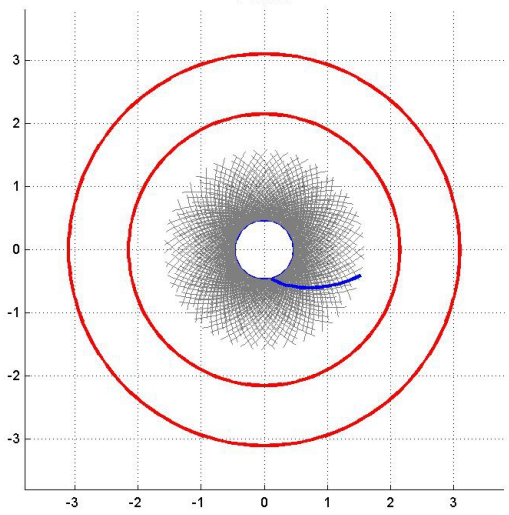


Figure 9.4: Scenario 1 at time t=0.8

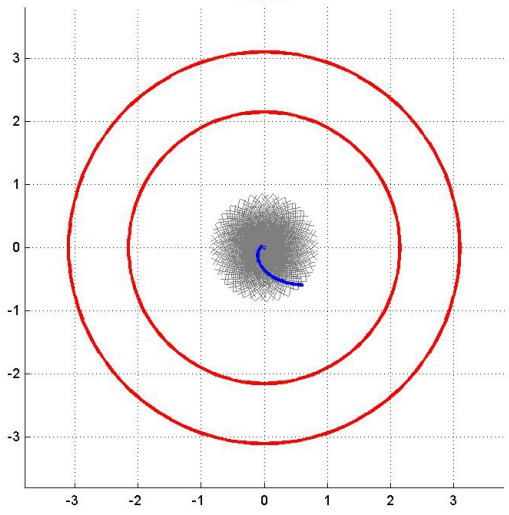


Figure 9.5: Scenario 1 at time t=0.892

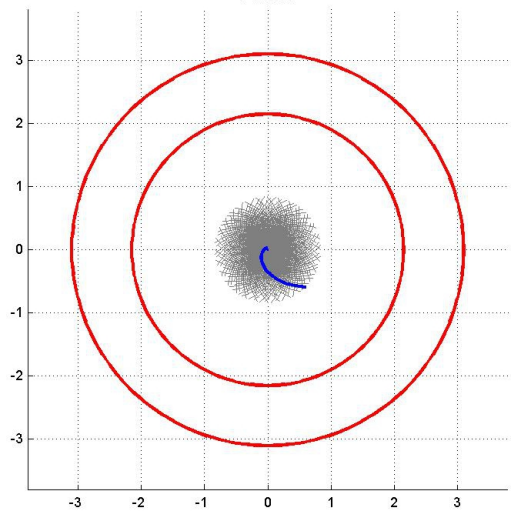


Figure 9.6: Scenario 1 at time t=0.894

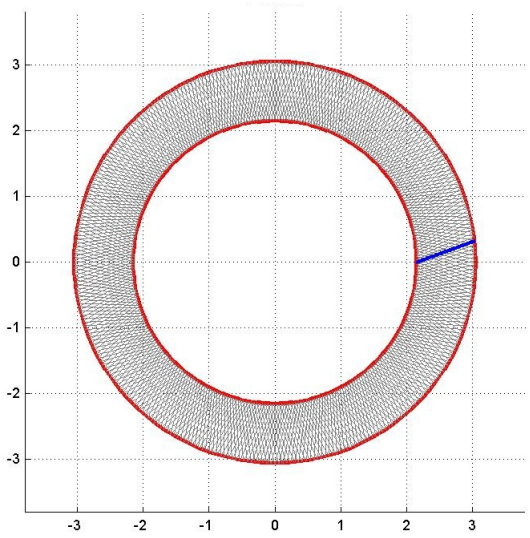


Figure 9.7: Scenario 2 at time t=0

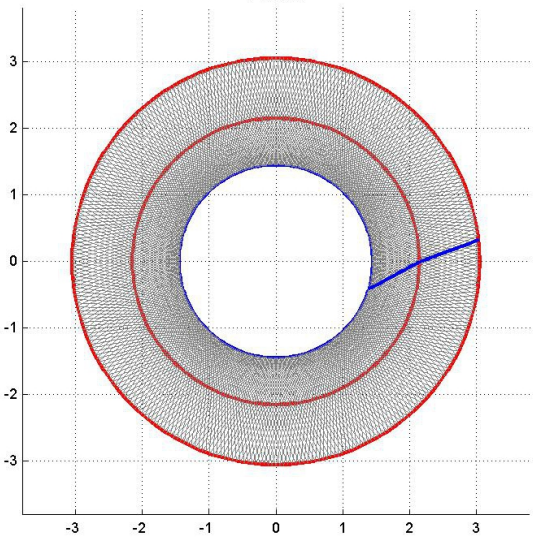


Figure 9.8: Scenario 2 at time t=0.3

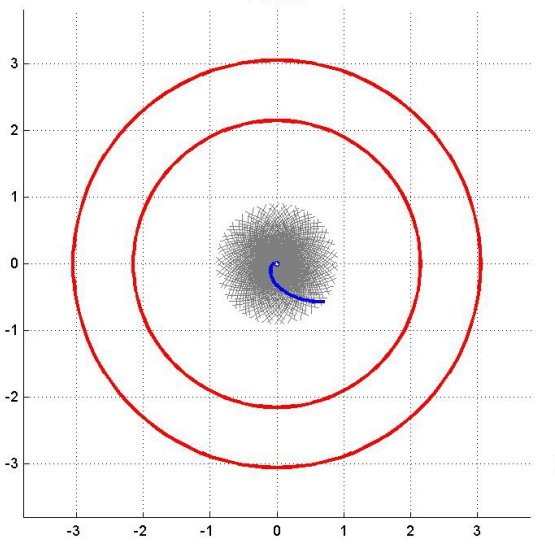


Figure 9.9: Scenario 2 at time t=0.894

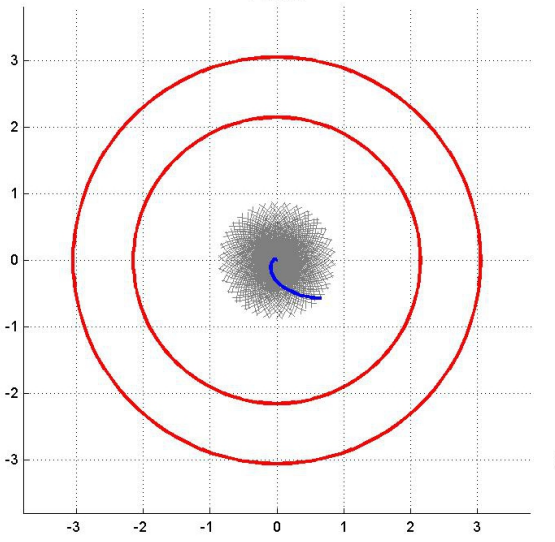


Figure 9.10: Scenario 2 at time t=0.898

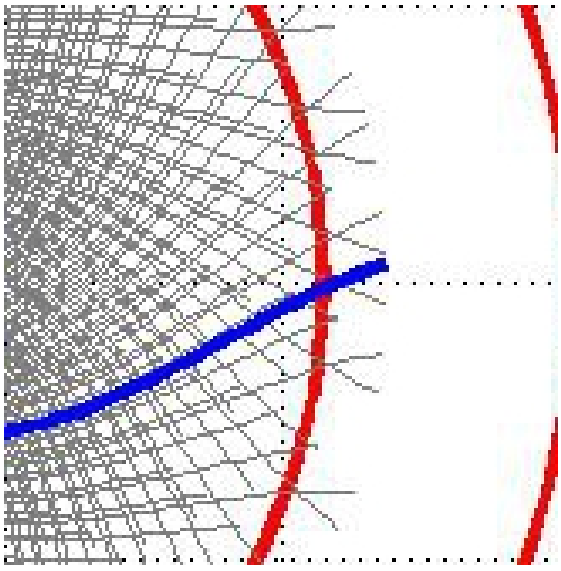


Figure 9.11: Scenario 2, detail at time t=0.7

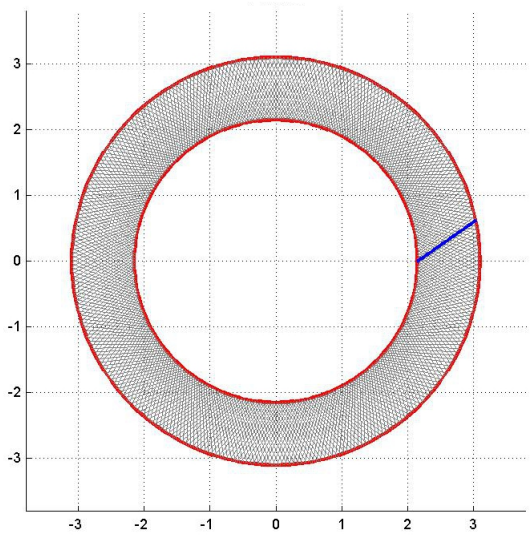


Figure 9.12: Scenario 3 at time t=0

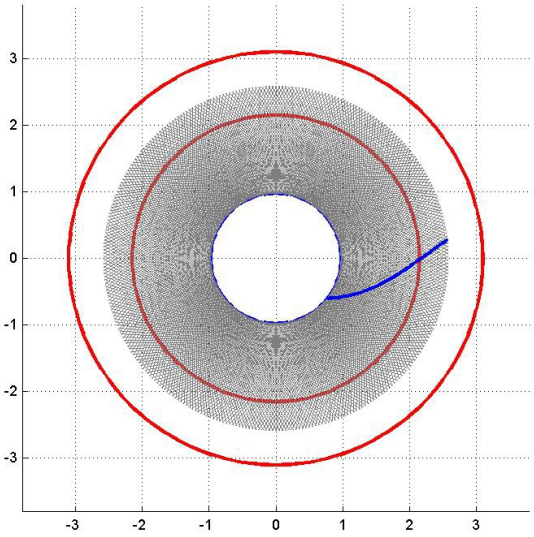


Figure 9.13: Scenario 3 at time t=0.6

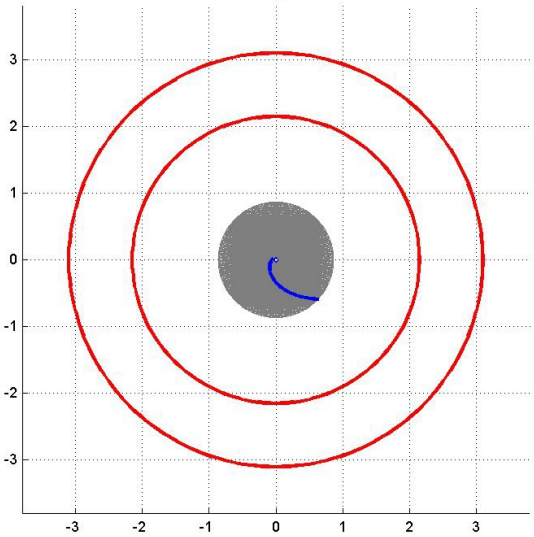


Figure 9.14: Scenario 3 at time t=0.894

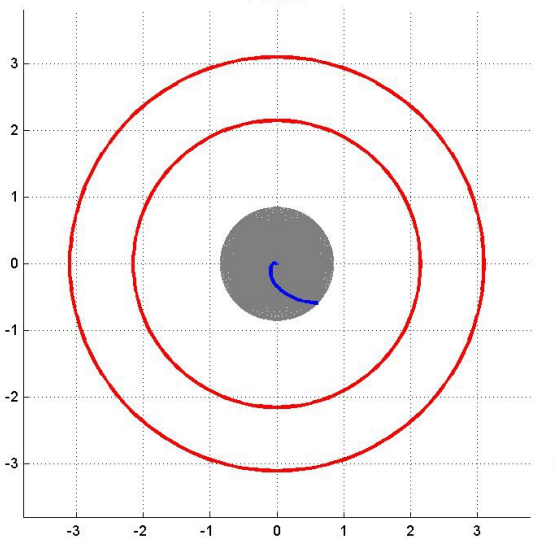


Figure 9.15: Scenario 3 at time t=0.896

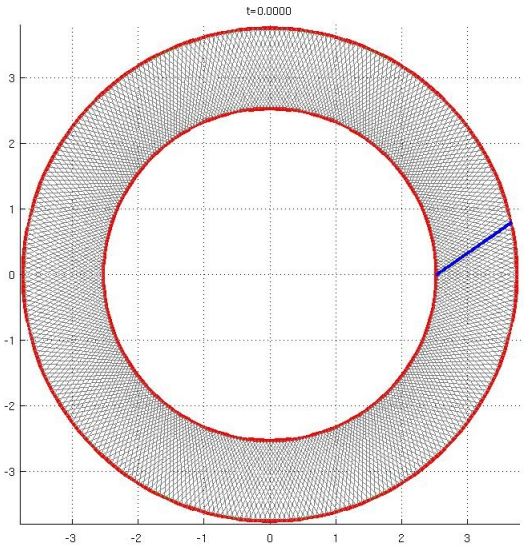


Figure 9.16: Scenario 4a at time t=0

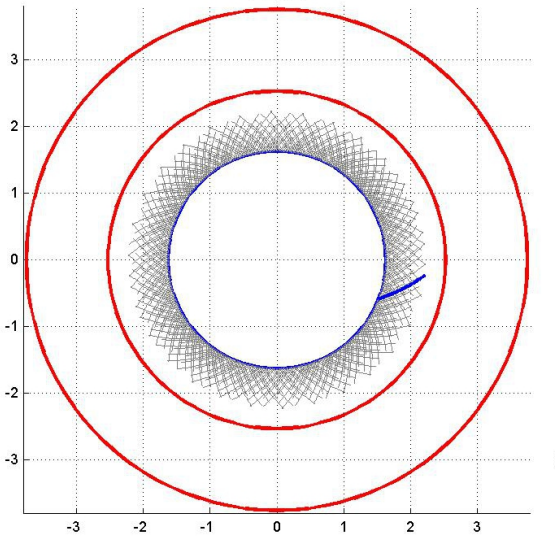


Figure 9.17: Scenario 4a at time t=0.6

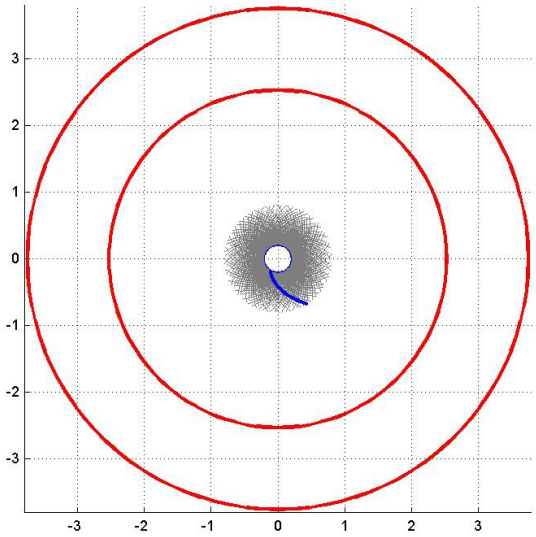


Figure 9.18: Scenario 4a at time t=0.894

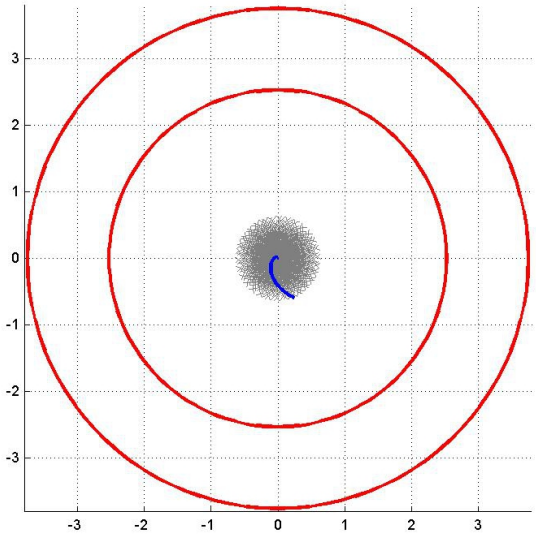


Figure 9.19: Scenario 4a at time t=0.928

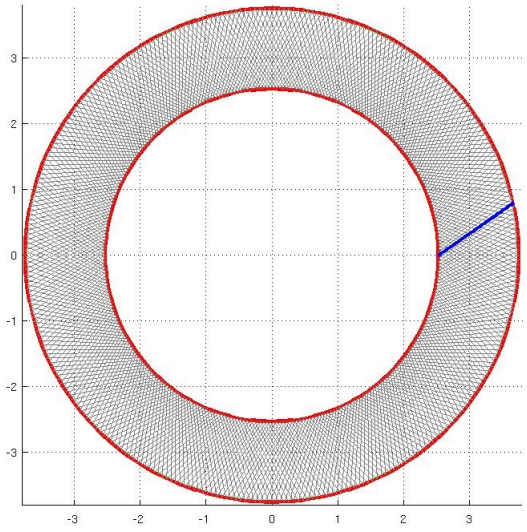


Figure 9.20: Scenario 4b at time t=0

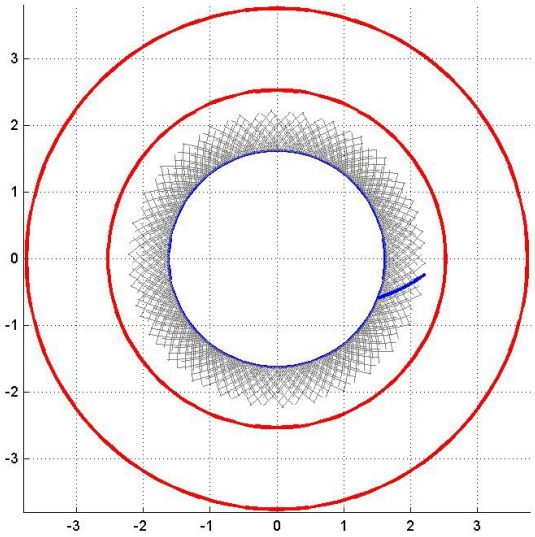


Figure 9.21: Scenario 4b at time t=0.6

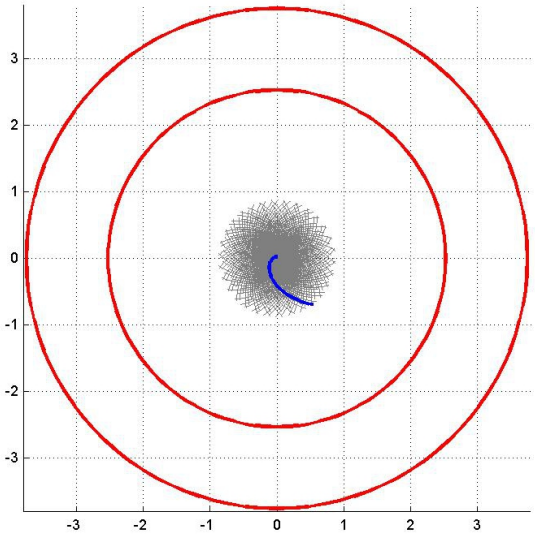


Figure 9.22: Scenario 4b at time t=0.880

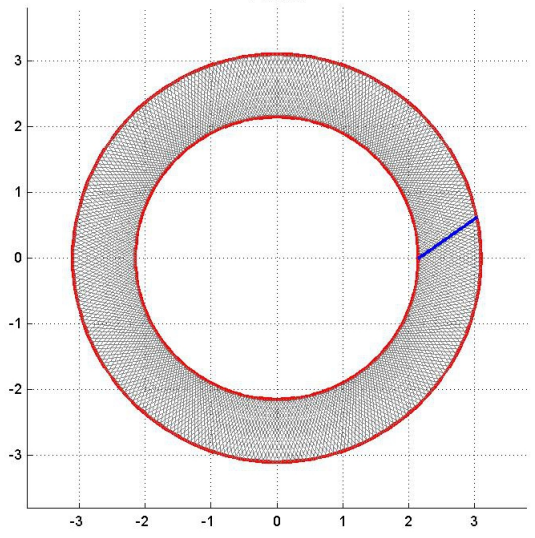


Figure 9.23: Scenario 5a at time t=0

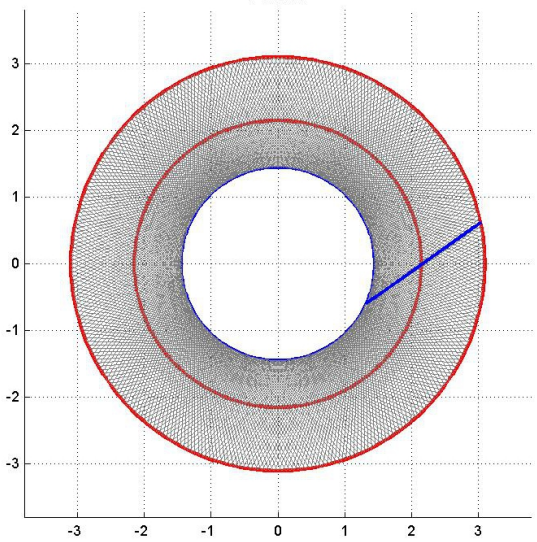


Figure 9.24: Scenario 5a at time t=0.3

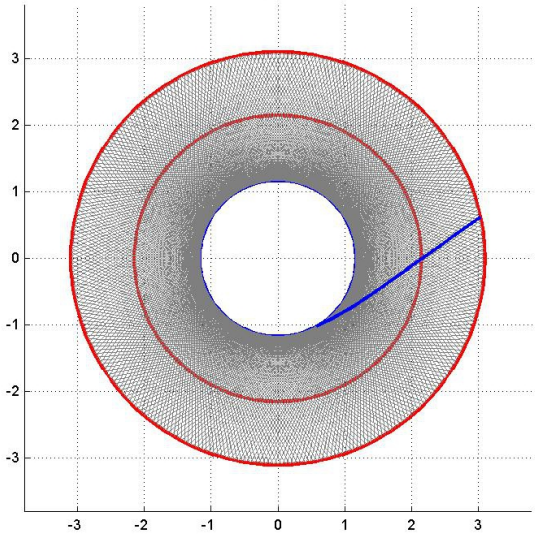


Figure 9.25: Scenario 5a at time t=0.406

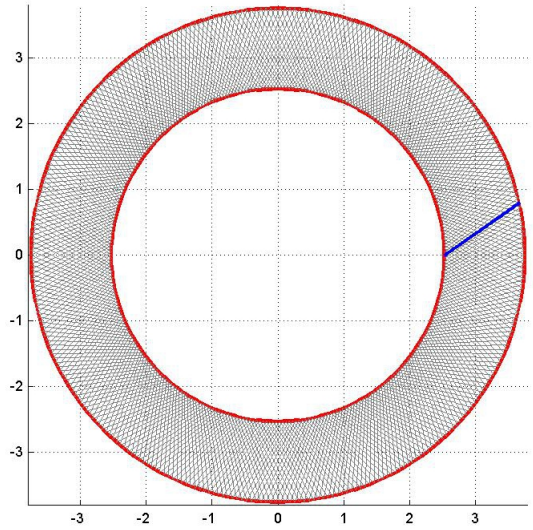


Figure 9.26: Scenario 5b at time t=0

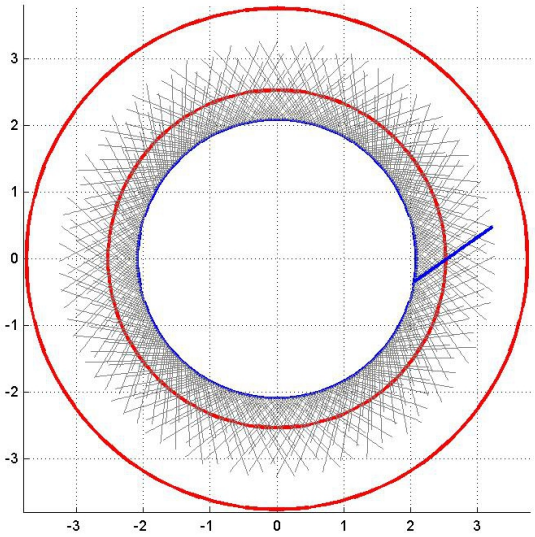


Figure 9.27: Scenario 5b at time t=0.3

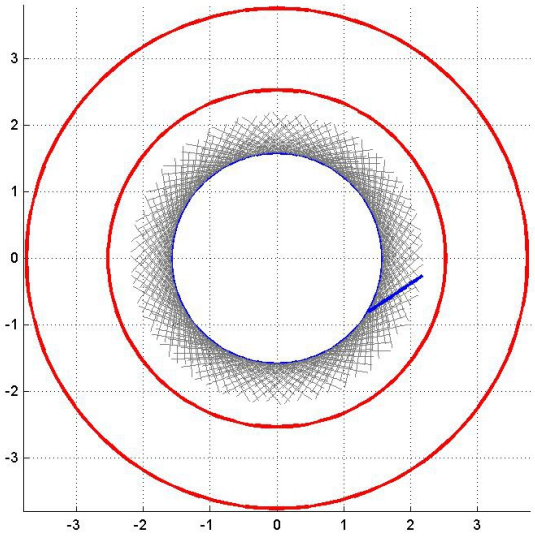


Figure 9.28: Scenario 5b at time t=0.608

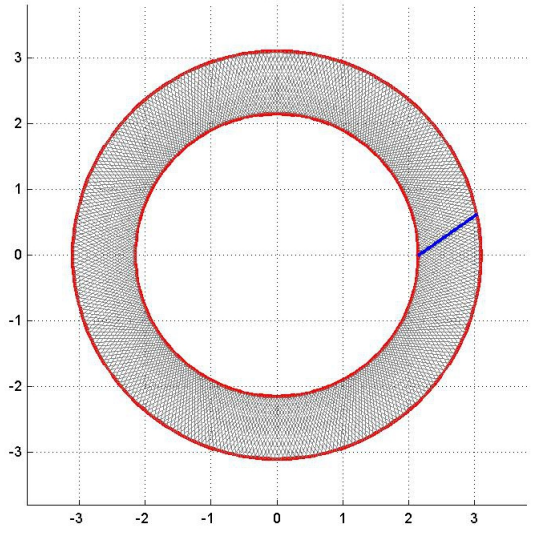


Figure 9.29: Scenario 6a at time t=0

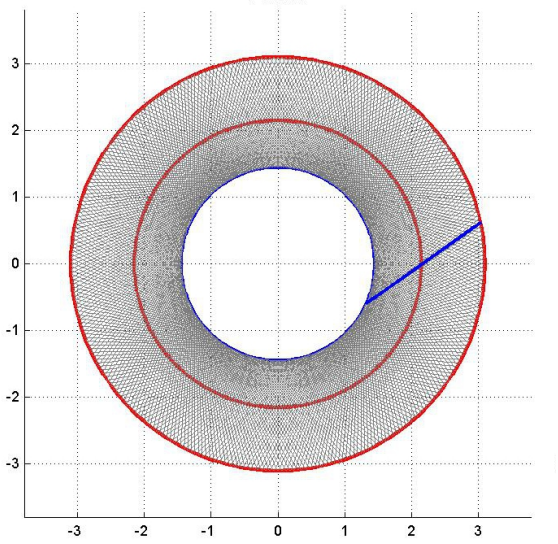


Figure 9.30: Scenario 6a at time t=0.3

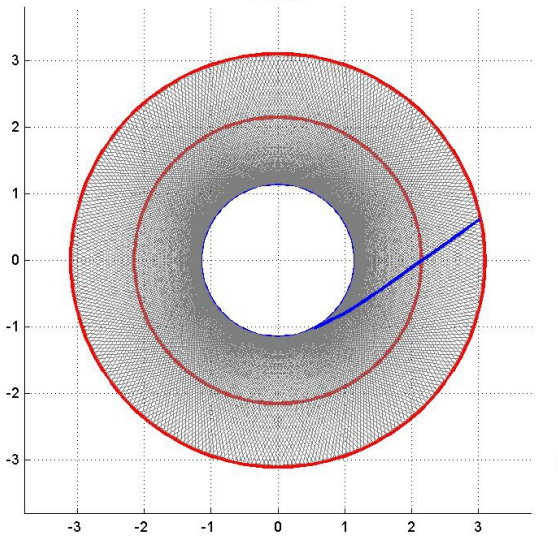


Figure 9.31: Scenario 6a at time t=0.408

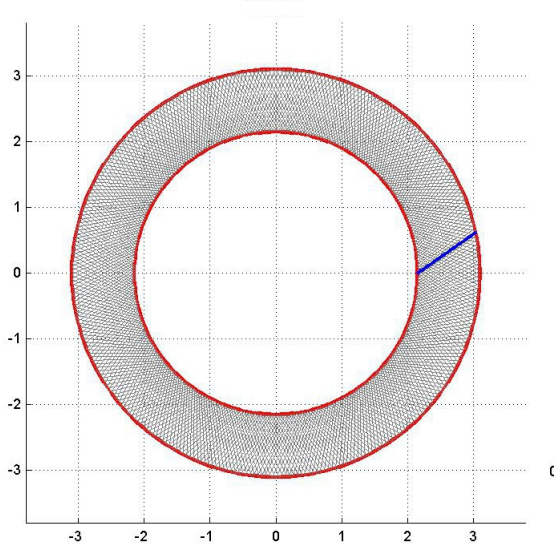


Figure 9.32: Scenario 6b at time t=0

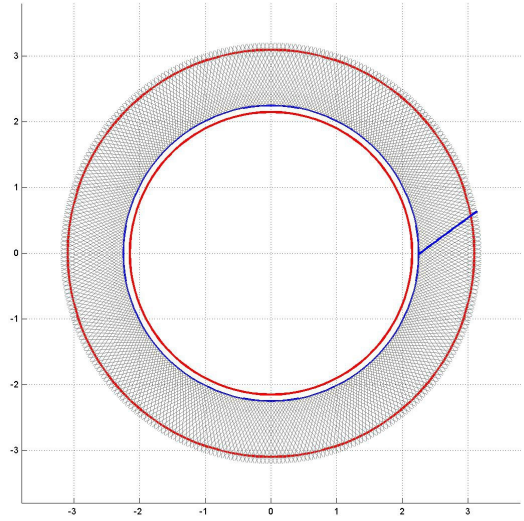


Figure 9.33: Scenario 6b at time t=0.006

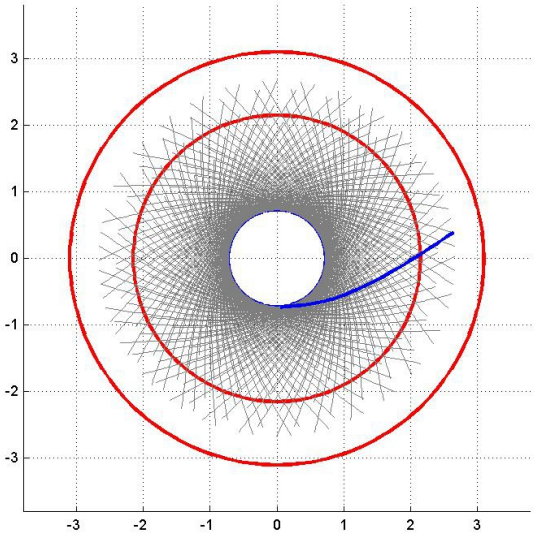


Figure 9.34: Scenario 6b at time t=0.534

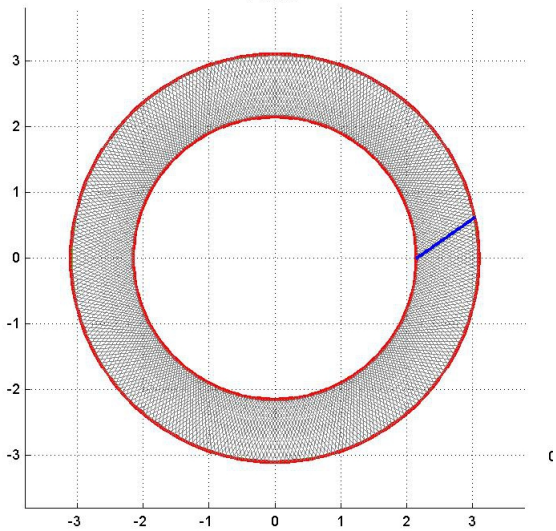


Figure 9.35: Scenario 7a at time t=0

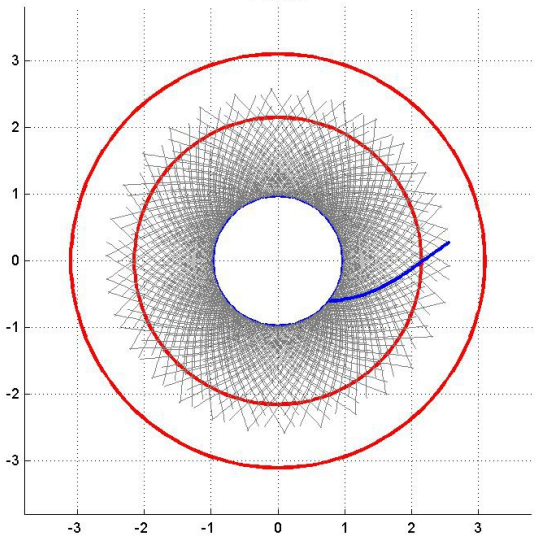


Figure 9.36: Scenario 7a at time t=0.6

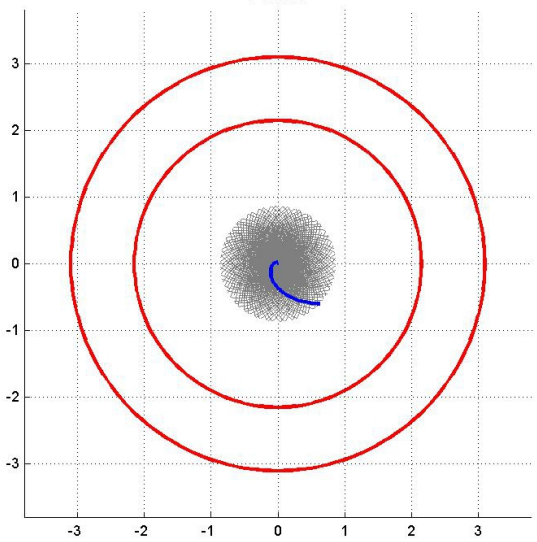


Figure 9.37: Scenario 7a at time t=0.894

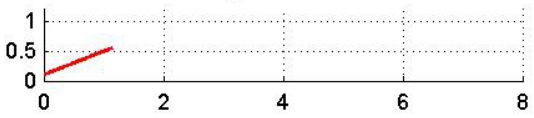


Figure 9.38: Length distribution for scenario 7a at time t=0.894

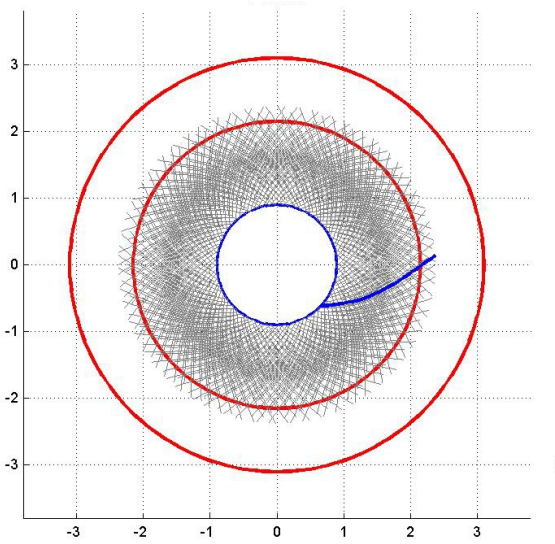


Figure 9.39: Scenario 7b at time t=0.68

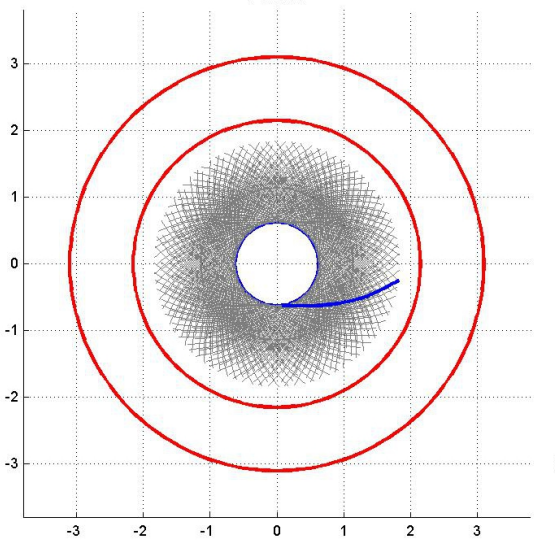


Figure 9.40: Scenario 7b at time t=0.774

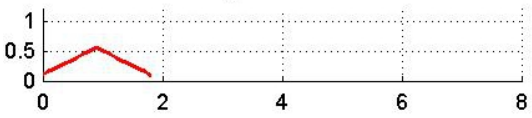


Figure 9.41: Length distribution for scenario 7b at time t=0.774

Curriculum Vitae

Personal Data

- Name: Stefanie Hirsch
- Born on June 3rd, 1980 in Vienna, Austria
- Nationality: Austria

Education

- 1986-1987: Volksschule Margaretenstraße in Vienna
- 1987-1990: Volksschule Lutherplatz in Vienna
- 1990-1998: Gymnasium Amerlingstraße in Vienna, Matura on 09th of June, 1998
- 1998-2000: Apprenticeship as a Bookseller at Buchhandlung Kuppitsch, Vienna
- 2004-2006: Diploma Studies of Technical Mathematics at the Technical University of Vienna, Austria
- 2006- : Diploma Studies of Mathematics at the University of Vienna, Austria
- Summer 2009: Summer School for Applied Mathematics and Modelling in Alba Adriatica, Italy

Work experience related to mathematics

- Tutor at the Faculty of Mathematics of the University of Vienna (2009 and 2010)