



Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

**Het modelleren van celdeformatie en -migratie:
Toepassing op het immuunsysteem
(Modeling Cell Deformation and Migration:
Application to the Immune System)**

Verslag ten behoeve van het
Delft Institute of Applied Mathematics
als onderdeel ter verkrijging

van de graad van

**BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE**

door

MATHIAS MUL

**Delft, Nederland
6 Juli 2012**

Contents

1	Introduction	2
2	The Model	3
2.1	One Cell One Source	3
2.2	Multiple Sources	7
2.3	Multiple Cells	9
2.4	Cell Shape Index	10
3	The Model for the Immune Response	11
3.1	White Blood Cell Concentration	12
3.2	Blood Flow	13
3.3	The Endothelium	15
3.4	Bacteria Behaviour	17
3.5	Degree of Infection	20
4	Numerical Solution Method	21
4.1	Time Discretization	21
4.2	Space Discretization	21
4.3	Time Integration	21
4.4	Convergence one cell one source	22
5	Numerical experiments	23
5.1	One Cell One Source	23
5.2	Multiple Sources	24
6	Computer Simulations of the Blood Vessel	28
6.1	Snapshots	29
6.2	Parameter Variation	34
7	Discussion and Conclusion	35
8	Appendix	36
9	References	47

Abstract

Shape transition of biological cells is simulated by a phenomenological model. The movement of cell walls is caused by its environment. Chemico-mechanically induced cell shape changes are treated with formal mathematical expressions in which lies the originality of this model. Visualizations of output will be presented in order to intuitively explain content. Furthermore, results on elemental research using this model applicated to the immune system and diabetes are available.

1 Introduction

The main reasons for cells to move are chemical factors in their surroundings and the presence of other cells. The form of a resting cell is usually a bi-polar or multi-polar shape, while during cellular motion the cell will change to an elongated oval geometry.

In order to understand biological processes like wound healing, tumor growth, organ development or infection by invading bacteria we have to gain a deeper understanding on a cellular level. Such understanding may lead to development of improved healthcare, which explains the need of understanding biology. In order to attain a higher level of understanding, it is important to observe the relationship between various biological parameters. Mathematical relations between these parameters are essential since it is the basis of mathematical modeling.

For the modeling of biological processes there exist several continuum-based methods incorporating partial differential equations^{[6][7]}. Mathematical literature contains simplified models for which analytic solutions are constructed, as well as more complicated modeling studies where finite-element simulators are developed^[8]. The main issue about these models is that they describe the biological processes representing cells by cell densities instead of treating individual cells.

The information about processes that take place in individual cells is lost when using cell density models. This strengthens the importance of individual cell based models. Of course some research effort devoted to the transition between the two models should be made, to show that they correspond with each other. In this report, a cell deformation model is applied to the immune system under diabetic conditions. Diabetis is a chronic disease in which there are high levels of sugar in the blood, which affects the immune system. Normally when a healthy person gets a bacteria infection, white blood cells in the blood circulation will push themselves through the blood vessel wall towards the infected tissue. In case of a person with diabetis, these blood vessel walls will become stiffer. Fewer white blood cells will arrive at the critical area and it will take longer to clean up the infected tissue. This phenomena will be investigated by using a mathematical model.

2 The Model

The general goal is to model the migration and deformation of individual biological cells. For simplification of these phenomena, the spatial domain of all simulations have two dimensions. The approach on this model will start as general and mathematical as possible. The basis of these ideas can be found in an article by *F.J. Vermolen* and *A.Gefen*.^[1]

2.1 One Cell One Source

We consider one arbitrarily shaped cell with a specified geometry and size which we call the *original shape* of the cell. The average position of the cell wall defines a point, the cell center, which coincides with the nucleus. The cell wall is discretized and represented by $N \geq 3$ points. At least three cell wall points are needed to define the representation of the cell area in a non-trivial way. We assume that the cell wall contracts to its original shape if it is not in that shape already.

In this context, a *source* is a point from which a certain chemical concentration is dispersed. We let this kind of sources represent bacteria, since we assume that bacteria are very small in comparison to cells. Cells are attracted by these chemical agents. This phenomenon is called *chemotaxis*.

The simple model assumes that the movement of a cell boundary only proceeds due to chemotaxis and contraction. Thus, for each node $\mathbf{x}_i$, we define the velocity vector $\mathbf{v}_i$ at point i on the cell wall at time t by

$$\mathbf{v}_i(t) = \mathbf{v}_i^{\text{contraction}}(t) + \mathbf{v}_i^{\text{chemotaxis}}(t), \quad \text{for } i \in \{1, \dots, N\}. \quad (1)$$

Cell contraction We assumed that the cell wall would contract to its original shape if it was not in that configuration already. This assumption is made because cells simply tend to do this when we observe them. In reality, there are thread-like connections between the cell nucleus and its cell wall, called *ligaments*.

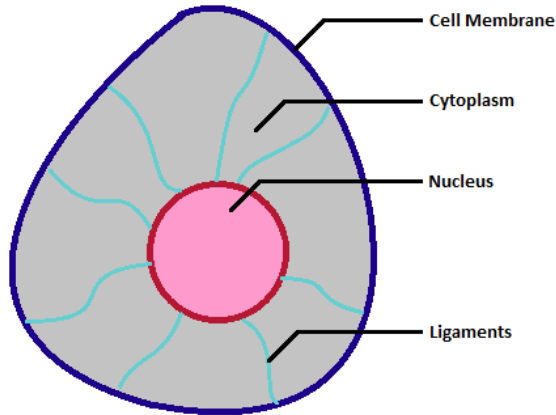


Figure 1: Scheme of the cell anatomy.

We bring this mechanism in our model by assuming that the cell center is connected to the cell wall by the cytoskeleton, which is represented by a series of springs. Since we made a space discretization, every node on the cell wall is connected to the center by a spring, see figure 2a. Using Hooke's law of elasticity we find an expression for the velocity as a result of contraction,

$$\mathbf{v}_i^{\text{contraction}}(t) = \alpha(\mathbf{x}_c(t) + \hat{\mathbf{x}}_i - \mathbf{x}_i(t)) . \quad (2)$$

Here α is the contraction parameter which defines the stiffness of the cell. The cell center at time t is denoted by $\mathbf{x}_c(t)$ and the original shape of the cell around the origin is defined by the set of points $\hat{\mathbf{x}}_i$, $i \in \{1, \dots, N\}$. The current location of the node i on the cell wall is denoted by $\mathbf{x}_i(t)$. Intuitively, the location of node i on the cell wall will have a velocity directed towards the point where the node would be located if the cell was not deformed, see 2b. Remark that $\mathbf{v}_i^{\text{contraction}}(t) = \mathbf{0}$ for every node i in the case of a resting cell.

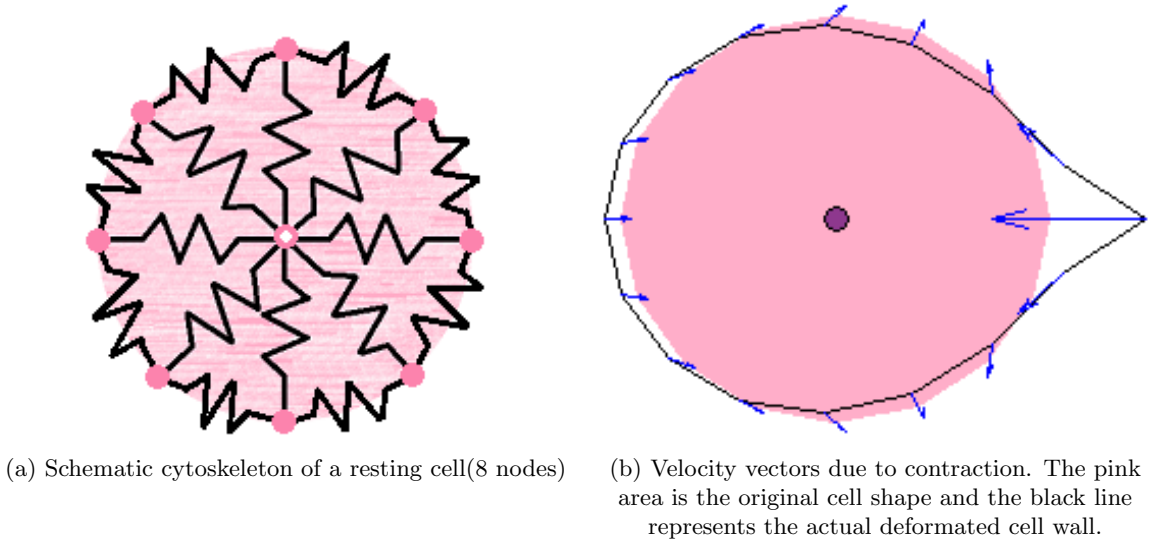


Figure 2: Contraction mechanism

The area of the cell is assumed to be bounded, therefore we define the contraction parameter in terms of the natural (A_0), minimum ($A_{\min}$), maximum ($A_{\max}$) and current $A(t)$ cell area as follows:

$$\alpha = \alpha_0 \frac{(A_0 - A_{\min})(A_{\max} - A_0)}{(A(t) - A_{\min})(A_{\max} - A(t))} . \quad (3)$$

This results in an increasing force towards the original shape as the area A of the projection of the cell differs more from the natural cell area. In [1], it is noted that the cell is a visco-elastic domain due to the interactions between the cytoskeleton and cytoplasm. Hence this would require a PDE to be solved using a finite-element method within each cell and therewith the computation time will increase enormously. Since our aim is to develop an efficient model that is based on closed form expressions for the diffusion of a chemical that attracts the cell, we simplify these notions and keep our current model of contraction the way we defined it.

Chemotaxis: Solutions to Diffusion Equations for Point-Sources We will make use of solutions to diffusion equations with point sources. We will treat both steady-state solutions as well as transient solutions. For simplicity, we assume that the domain of computation of the diffusion equations are unbounded. Boundedness can be dealt with in several ways and is not really an important issue at this point.

Quasi steady-state solutions The size of the bacteria that release the chemical agent, which attracts cells, is assumed to be very small compared to the cells and therefore the bacteria are approximated by *point sources*. We allow point sources to move, from which the assumption follows that the diffusion-field that surrounds it sets in instantaneously. Consider a point source $(X(t), Y(t))$, and $(x, y) \in \mathbb{R}^2$, $t > 0$ of which releases a diffusing chemical, then we have

$$-D\Delta c = \gamma(t)\delta(x - X(t), y - Y(t)). \quad (4)$$

We assume that $(X(t), Y(t))$ represents a given path in $\mathbb{R}^2$ which could be either the result of some (biased) random motion or from a complete deterministic motion. The concentration of the chemical is denoted by c and the diffusion coefficient is denoted by D . The strength of the point source at time t is defined as $\gamma(t)$. Lastly, $\delta(\cdot)$ denotes the Dirac Delta Function, which satisfies

$$\int_{\mathbb{R}^2} \delta(x, y) d\Omega = 1, \quad \text{with } \delta(x, y) = 0, \text{ if } (x, y) \neq (0, 0). \quad (5)$$

The solution to equation (4) is given by the Green's function

$$c(t, (x, y)) = -\frac{\gamma(t)}{2\pi D} \ln((x - X(t))^2 + (y - Y(t))^2), \quad (6)$$

which can be found in mathematical literature about this subject, see *Evans, 1998*^[2], among many others. Note that this is an analytic explicit solution of the concentration c in an unbounded domain.

Transient solutions In this case we arrive in a similar way to the equation

$$\frac{\partial c}{\partial t} - D\Delta c = \gamma(t)\delta(x - X(t), y - Y(t)), \quad \text{for } (x, y) \in \mathbb{R}^2, t > 0. \quad (7)$$

As in the quasi steady-state case, we assume that $(X(t), Y(t))$ represents a given path in $\mathbb{R}^2$ which could be either the result of some (biased) random motion or from a complete deterministic motion. The concentration of the chemical is denoted by c and diffuses with diffusion coefficient D . The strength of the point source at time t is defined as $\gamma(t)$. Lastly, $\delta(\cdot)$ denotes the Dirac Delta Function. We assume that the concentration is zero initially. Then, by *Evans 1998*^[2], we obtain the solution for the concentration c .

$$c(t, (x, y)) = \frac{1}{4\pi D} \int_0^t \frac{\gamma(s)}{t-s} \exp\left(-\frac{\|\mathbf{x} - \mathbf{X}(s)\|^2}{4D(t-s)}\right) ds \quad (8)$$

Chemotaxis: Migration of the cell

Active range The *active range* of a cell is the range $\rho > 0$ in which a cell deactivates all sources. If a source moves within the active range of a cell it will no longer emit any chemical agent and is said to be annihilated. Biologically, we assume that the cell mobility – its migration and deformation behaviour – influences the time it takes to neutralize bacteria. We define the active range AR_ρ to be the area defined by the union of all circles with radius ρ around the nodes of the cell wall:

$$AR_\rho = \{\mathbf{x} \in \mathbb{R}^2 : |\mathbf{x} - \mathbf{x}_i| < \rho, i \in \{1, \dots, N\}\}, \quad (9)$$

where N represents the number of points on the cell wall.

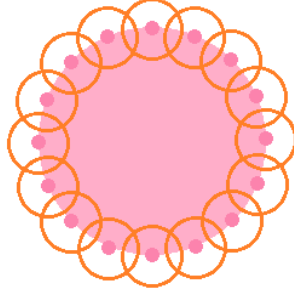


Figure 3: An example of the area AR_ρ : The union of orange circles with radius ρ .

Migration In the case that the cell and the source do not make any physical contact – the source is not within the *active range* of the cell – we describe the motion of the cell wall at time t due to the scalar field $c(t, (x, y))$. We assume that the direction of the velocity $\mathbf{v}_i^{\text{chemotaxis}}$ of all discretized cell wall points i is directed towards the largest increase in concentration c and the magnitude is proportional to this change of concentration. Hence, $\mathbf{v}_i^{\text{chemotaxis}}$ is proportional to the gradient of the concentration, $\nabla c(t, \mathbf{x}_i(t))$. Thus, we define the motion as

$$\mathbf{v}_i^{\text{chemotaxis}}(t) = \beta \nabla c(t, \mathbf{x}_i(t)), \text{ for } i \in \{1, \dots, N\}, \quad (10)$$

where β is the mobility parameter of the cell wall. Note that this can be made spatially dependent over the cell wall. Again, the position of the discretized cell wall points are $\mathbf{x}_i(t)$. This particular part of the model contributes to the efficiency of the model. We are able to calculate *explicit* expressions of $\nabla c(t, (x, y))$ in order to make fast calculations possible. We will give general explicit expressions at the end of the next section.

2.2 Multiple Sources

In this section we expand the model and we will present solutions for multiple and continuous sources. Let the total number of sources be $n \geq 2$ with intensities $\gamma_j \in \mathbb{R}_{>0}$ and positions $(X_j(t), Y_j(t)) \in \mathbb{R}^2$, $j \in \{1, \dots, n\}$. Linearity of the diffusion equation justifies the use of the superposition principle.

Solutions to Diffusion Equations for multiple Point-Sources First, we show that the superposition principle holds for the quasi steady-state. We have

$$-\Delta c_j = \gamma_j(t) \delta(x - X_j(t), y - Y_j(t)), \quad \forall j \in \{1, \dots, n\} \quad (11)$$

Thus, we have j equations. We add the left-hand side and the right-hand side of these equations and use linearity of the differential operator Δ ,

$$-\Delta \sum_{j=1}^n c_j = -\sum_{j=1}^n \Delta c_j = \sum_{j=1}^n \gamma_j(t) \delta(x - X_j(t), y - Y_j(t)). \quad (12)$$

Hence,

$$c(t, (x, y)) = \sum_{j=1}^n c_j(t, (x, y)) \quad (13)$$

satisfies the differential equation with multiple sources,

$$-\Delta c = \sum_{j=1}^n \gamma_j(t) \delta(x - X_j(t), y - Y_j(t)). \quad (14)$$

Now, we have shown that the superposition principle holds for the quasi steady-state. The proof of the transient case is analogous.

Quasi steady-state solutions By superposition:

$$c(t, (x, y)) = -\frac{1}{2\pi D} \sum_{j=1}^n \gamma_j(t) \ln(\|\mathbf{x} - \mathbf{X}_j(t)\|^2) . \quad (15)$$

For a continuously distributed source function $Q(t, \mathbf{x})$, $\mathbf{x} = (x, y)$ that is non-zero in $\Omega \subset \mathbb{R}^2$, we find a convolution-based solution:

$$c(t, (x, y)) = -\frac{1}{2\pi D} \int_{\Omega} Q(t, \bar{\mathbf{x}}) \ln(\|\mathbf{x} - \bar{\mathbf{x}}(t)\|^2) d\Omega . \quad (16)$$

Transient solutions By superposition:

$$c(t, (x, y)) = \frac{1}{4\pi D} \sum_{j=1}^n \int_0^t \frac{\gamma_j(s)}{t-s} \exp\left(-\frac{\|\mathbf{x} - \mathbf{X}_j(s)\|^2}{4D(t-s)}\right) ds. \quad (17)$$

Again, for a continuously distributed source function $Q(t, \mathbf{x})$, $\mathbf{x} = (x, y)$ that is non-zero in $\Omega \subset \mathbb{R}^2$, we find a convolution-based solution:

$$c(t, (x, y)) = \frac{1}{4\pi D} \int_0^t \int_{\Omega} \frac{Q(s, \bar{\mathbf{x}})}{t-s} \exp\left(-\frac{\|\mathbf{x} - \bar{\mathbf{x}}\|^2}{4D(t-s)}\right) d\Omega ds. \quad (18)$$

Note that we assumed that $\Omega \neq \Omega(t)$ and used Fubini's theorem for interchanging the integrals over space and time.

Explicit expressions of the concentration gradient As stated earlier, we will now present general expressions of the concentration gradient. These expressions are used in numerical experiments and simulations which can be found in later chapters.

Quasi steady-state

$$\nabla c(t, (x, y)) = \frac{1}{2\pi D} \sum_{j=1}^n \frac{\gamma_j(t)(\mathbf{X}_j(t) - \mathbf{x})}{\|\mathbf{X}_j(t) - \mathbf{x}\|^2} \quad (19)$$

Transient case

$$\nabla c(t, (x, y)) = \sum_{j=1}^n \int_0^t \frac{\gamma_j(s)(\mathbf{x} - \mathbf{X}_j(s))}{8\pi D^2(t-s)^2} \exp\left(-\frac{\|\mathbf{x} - \mathbf{X}_j(s)\|^2}{4D(t-s)}\right) ds \quad (20)$$

2.3 Multiple Cells

When we consider multiple cells there are two different approaches, because some cells interact mutually and some do not. At this point we introduce the terms hard impingement and soft impingement.

- *Hard impingement* is the case in which cells interact because they physically touch each other.
- *Soft impingement* is the case in which cells notice each other from a distance, but are not in physical contact.

Cells interact mutually This case will not be treated elaborately, since it is not used for the simulations and experiments in this thesis. We explain an idea to model soft impingement for M cells. For each cell $m \in \{1, \dots, M\}$, the cell wall is spatially discretized into N nodes. We assume that every defined node works as a source that releases a certain chemical agent. Then, cell m reacts to the concentration chemicals released by all other cells in the set $\{1, \dots, M\} \setminus m$ and moves towards the largest increase of chemical concentration. This is the chemotaxis principle. Once cells are getting close to each other, hard impingement has to be taken into account. We will explain a way to model this in the next paragraph. More details about mutually interacting cells can be found in other literature.^[1]

Cells ignore cells Before clarifying this case, we redefine the velocity vector $\mathbf{v}_i$ at point i on the cell wall at time t by

$$\mathbf{v}_i(t) = \mathbf{v}_i^{\text{contraction}}(t) + \mathbf{v}_i^{\text{chemotaxis}}(t) + \mathbf{v}_i^{\text{anti-overlap}}(t), \quad \text{for } i \in \{1, \dots, N\}. \quad (21)$$

At this point, we basically have to do nothing but avoid collisions. Physically it is not possible that two cells are in the same position at the same time, because they push each other away when they get close. For the sake of simplicity, we only consider the cell centers. Assume we have two cells, A and B , both with radius r at rest. They define two cell centers, positioned at $\mathbf{x}_c^A$ and $\mathbf{x}_c^B$. The distance between the two cell centers is denoted by $d(A, B)$. The length of overlap is defined as $\Delta = \max(2r - d(A, B), 0)$. Whenever they are too close to each other they will both be pushed aside.

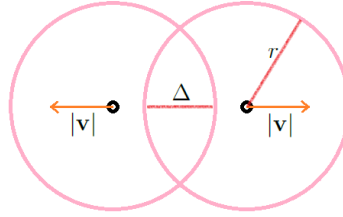


Figure 4: Two cells close to each other

We define the velocity $\mathbf{v}_i^{\text{anti-overlap}}(t)$ of cell A due to cell B as

$$\mathbf{v}_i^{\text{anti-overlap}(A)}(t) = \sqrt{\frac{\Delta}{r - \Delta}} \cdot \frac{\mathbf{x}_c^A - \mathbf{x}_c^B}{\|\mathbf{x}_c^A - \mathbf{x}_c^B\|}, \quad (22)$$

Notice that this dimensionless velocity is equal for every node on the cell wall of A . This is a consequence of the fact that we use only the cell center for the computation. The magnitude of this velocity is $\sqrt{\frac{\Delta}{r - \Delta}}$. Note that this velocity tends to infinity for $\Delta \rightarrow r$ and equals zero for $\Delta = 0$. Physically, this means that two cell centers can never be at the same location at the same time. Also, if the cells have no contact they will not push. Finally, remark that both cells are pushed aside with the same velocity but with opposite direction. Formula (22) is used in all simulations and experiments that involve multiple cells.

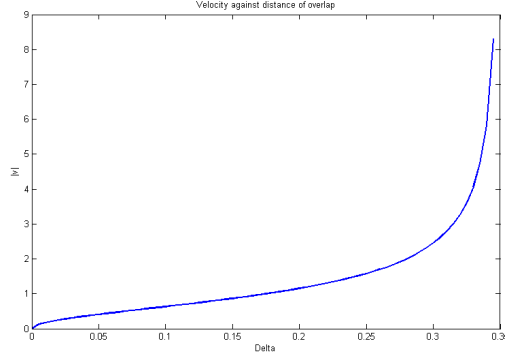


Figure 5: For $r = 0.35$, a plot of the function $\sqrt{\frac{\Delta}{r-\Delta}}$.

2.4 Cell Shape Index

Once a deforms, the cell wall will stretch. In two dimensions, this is described by a relationship between the cell area and the length of the cell wall. We define the Cell Shape Index (CSI) of a circle-shaped cell as

$$\text{CSI}(t) = \frac{4\pi A(t)}{l(t)^2} \quad (23)$$

In this equation, $A(t)$ denotes the cell area and $l(t)$ denotes the perimeter of the cell area. Note that this index is scaled so in the case of a resting cell the CSI equals one. For a lower value of CSI, the cell is stretched. An animation of a cell with a changing CSI over time can be found in the Numerical Experiments section.

3 The Model for the Immune Response

First, the global idea of the model will be given. Then, all aspects will be treated separately with an exact mathematical approach.

The immune response model we consider basically consists of two elements, a blood vessel and an area infected by bacteria. We assume that the blood vessel is straight and we define the blood flow from left to right. Depending on their concentration, white blood cells will enter the blood vessel at the left boundary – that is, at $x = 0$ – according to a *Poisson process*. Without any infection they will just follow the blood stream to the right and disappear when they leave the area of computation, see figure 6.

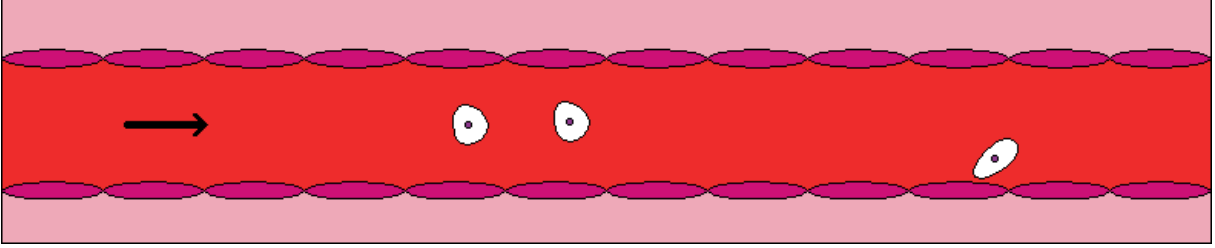


Figure 6: Simulation: Some white blood cells in a blood vessel. (Snapshot at a certain time t)

When we include an infected area – which could represent infected tissue or an infected organ – the white blood cells will push themselves through the endothelium towards the bacteria. The process of white blood cells pushing themselves outwards of the blood vessel is called *diapedesis*^[5], see figure 7. In order to clean the infected area, white blood cells annihilate the present bacteria. If the concentration of bacteria has decreased sufficiently the damaged area will start to restore itself.

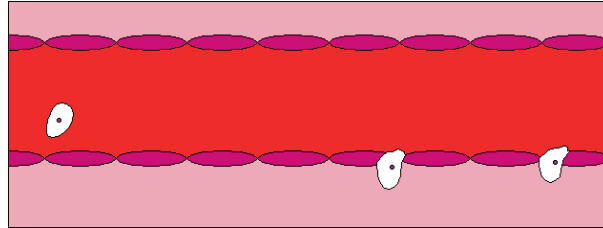


Figure 7: Simulation: *Diapedesis* – White blood cells leaving the blood vessel by pushing themselves through the endothelium. (Snapshot at a certain time t)

3.1 White Blood Cell Concentration

Define the concentration of white blood cells present inside one unit length of blood vessel as c_{wbc} . Also, define the maximum blood velocity as f_{max} . Then, white blood cells enter the blood vessel at the left side by a Poisson process with parameter $\lambda = c_{\text{wbc}} \cdot f_{\text{max}}$, which represents the ammount of white blood cells per time unit. We define W to be the stochastic variable that represents the total number of white blood cells entering the blood vessel during one time unit,

$$W \sim \text{Poisson}(\lambda). \quad (24)$$

Since λ does not depend on time, the expected number of white blood cells entering during the time interval $[t_1, t_2]$, for $t_1 < t_2$ is

$$\lambda_{t_1, t_2} = (t_2 - t_1) \cdot \lambda. \quad (25)$$

Positioning new white blood cells Once a white blood cell enters the blood vessel, we need to define its position. We define its cell center and the cell in its original shape. The x -position of the cell center is 0, because this is the left side of the vessel. The y -position is randomly determined using a stochastic variable B , $B \sim \text{Beta}(2, 2)$. The Beta(2, 2) distribution is chosen because we expect that a white blood cell is more likely to be positioned in the center of the blood vessel than on the vessel walls. The probability density function of B is plotted in figure 8.

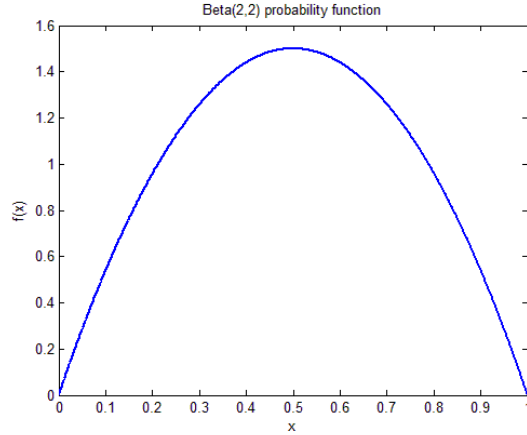


Figure 8: Probability density function of the Beta(2, 2) distribution.

We shift the mean of the sample value to 0 and stretch it according to the thickness h of the blood vessel and the radius r_{wbc} of the cell. To be exact, we define a new cell center $\mathbf{x}_c = (x_c, y_c)$ as

$$\begin{cases} x_c = 0 \\ y_c = (h - r_{\text{wbc}}) \cdot (B - \frac{1}{2}) \end{cases} \quad (26)$$

This definition guarantees that $-h + r_{\text{wbc}} < y_c < h - r_{\text{wbc}}$, which means that the new cell will be located inside the blood vessel.

3.2 Blood Flow

The blood flow inside the blood vessel is modeled by a Poiseuille flow and time-dependent blood pressure. We represent the blood vessel by a rectangle R with height $h > 0$ and length $L > 0$ as

$$R = \{(x, y) \in \mathbb{R}^2 : 0 < x < L, -h/2 < y < h/2\}. \quad (27)$$

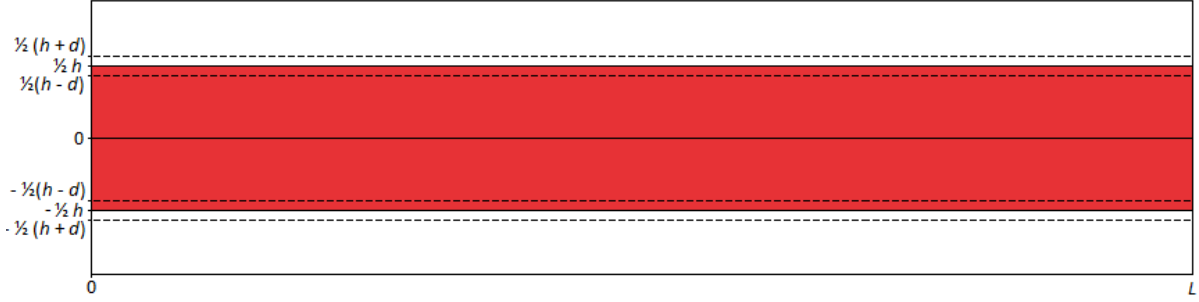


Figure 9: A scheme of the blood vessel. The red part represents the blood vessel R .

Poiseuille Flow We assume that the blood inside the blood vessel only has a velocity in the positive x -direction. Also, the velocity equals zero at the borders of the blood vessel. The approximate solution $u(y)$ to the Navier-Stokes equations is then given by ^[3]

$$u(y) = \frac{1}{2\mu} \frac{dp}{dx} y^2 + C_1 y + C_0. \quad (28)$$

Here, μ is the dynamic viscosity coefficient of blood and $\frac{dp}{dx}$ is the negative blood pressure gradient. Furthermore, C_1 and C_2 are constants that are determined by the boundary conditions. As for the model, we define a poiseuille flow pressure parameter $\zeta(t)$ that defines the maximum blood flow velocity inside the blood vessel as

$$\zeta(t) = -\frac{1}{2\mu} \frac{dp}{dx}(t). \quad (29)$$

After the introduction of this parameter the solution for the blood flow becomes

$$u(t, (x, y)) = \begin{cases} \zeta(t)(y - \frac{h}{2})(y + \frac{h}{2}) & \text{for } (x, y) \in R \\ 0 & \text{for } (x, y) \notin R \end{cases} \quad (30)$$

The equation for the velocity of a cell wall node i due to the blood flow becomes

$$\mathbf{v}_i^{\text{bloodflow}}(t) = u(t, (x, y)) \cdot \hat{\mathbf{i}}, \quad (31)$$

where $\hat{\mathbf{i}}$ is the unit vector pointing in the positive x -direction.

Blood pressure The blood pressure is periodic and depends on the heart beat. Therefore, $\zeta(t)$, which is the blood pressure multiplied by a constant, is also periodic. Denote the length of the cardiac cycle by L_{cc} . The blood pressure parameter during one cardiac cycle is plotted in figure 10, observe that $\zeta(t) \in [\frac{2}{3}\max_\zeta, \max_\zeta]$. The length of the systolic phase is defined as $\frac{1}{4}L_{cc}$, here the blood pressure is increasing. The length of the diastolic phase is defined as $\frac{3}{4}L_{cc}$, where the blood pressure decreases again. The disruption of pressure decrease during the diastolic phase can be explained by the fact that after every heart beat, the diameter of the aorta changes a little due to tightening and relaxing.

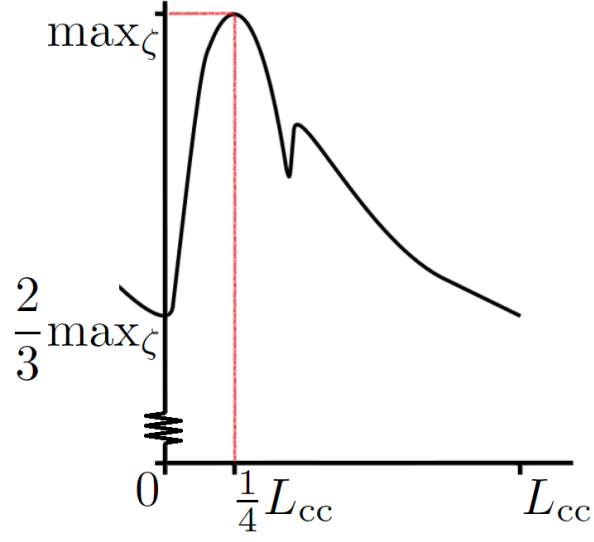


Figure 10: One period of the blood pressure.

For simplification, we approach this pressure by quadratic spline interpolation.^[4] In order to do this we interpolate twice, once between the two points $(0, \zeta(0))$ and $(\frac{1}{4}L_{cc}, \zeta(\frac{1}{4}L_{cc}))$, and once between $(\frac{1}{4}L_{cc}, \zeta(\frac{1}{4}L_{cc}))$ and $(L_{cc}, \zeta(L_{cc}))$. An example of this interpolation is given by the following plot, see figure 11. The timestep equals 0.1, the other parameters are $\max_{\zeta} = 4.5$, $L_{cc} = 2$

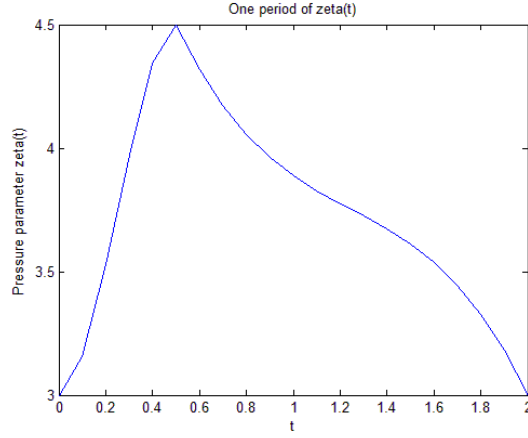


Figure 11: One period of $\zeta(t)$.

3.3 The Endothelium

The endothelium is a thin layer of cells at the side of the blood vessel. The cells that form the endothelium are called endothelial cells. This part is about the resistance that occurs when white blood cells push themselves through the endothelium. The stiffness of the endothelium is described by the parameter $\mu > 1$. The resistance is modeled by two components, decreased mobility of the white blood cell and a force that presses white blood cells back in the blood vessel. It is important to note that for these two components two different parameters (β_μ and γ_μ) are used that do not depend on each other. For a simplification of the model we choose to use one stiffness parameter μ that determines both parameters β_μ and γ_μ .

Decreased cell mobility The cell mobility parameter β , which can be found in equation (10) is assumed to decrease if a white blood cell tries to push itself through the endothelium. Figure 12 shows the situation we model. The diameter of the blood vessel is h , the thickness of the endothelium is d .



Figure 12: A scheme of the blood vessel. The green part represents the endothelium area.

Define the endothelium area $E = E_1 \cup E_2$ by the two rectangles E_1 and E_2

$$E_1 = \{(x, y) \in \mathbb{R} : 0 < x < L, \quad \frac{1}{2}(h-d) < y < \frac{1}{2}(h+d)\}, \quad (32)$$

$$E_2 = \{(x, y) \in \mathbb{R} : 0 < x < L, \quad -\frac{1}{2}(h+d) < y < -\frac{1}{2}(h-d)\}. \quad (33)$$

At this point we will consider the part of the cell wall that is inside the area of the endothelium. This part can be approximated by the fraction of nodes that is located in E at time t by $\theta(t) \in [0, 1]$,

$$\theta(t) = \frac{1}{N} \sum_{i=1}^N \chi_E(\mathbf{x}_i(t)), \quad (34)$$

with

$$\chi_E(\mathbf{x}_i(t)) = \begin{cases} 1 & \text{for } \mathbf{x}_i(t) \in E \\ 0 & \text{for } \mathbf{x}_i(t) \notin E \end{cases}. \quad (35)$$

The idea is to replace the mobility parameter β by $\beta^*(t)$, a convex combination of β and a smaller cell mobility β_μ defined as

$$\beta^*(t) = \beta(1 - \theta(t)) + \beta_\mu \theta(t), \quad (36)$$

with

$$\beta_\mu = \frac{\beta}{\mu}. \quad (37)$$

Remark that $\beta_\mu < \beta$, because $\mu > 1$. Since $\theta(t) \in [0, 1]$, observe that $\beta^*(t) \in [\beta_\mu, \beta]$. At this point we replace equation (10) of the previous chapter with

$$\mathbf{v}_i^{\text{chemotaxis}}(t) = \beta^*(t) \nabla c(t, \mathbf{x}_i(t)), \text{ for } i \in \{1, \dots, N\}. \quad (38)$$

Notice that the decreased cell mobility influences all nodes on the cell wall equally, depending on the fraction inside the endothelium area.

Force Completely depending on the parameter for the stiffness of the endothelium μ , we define γ_μ as

$$\gamma_\mu = \frac{\mu}{10}. \quad (39)$$

Then, the result of the force pushing the cell back in the blood vessel is the velocity

$$\mathbf{v}_i^{\text{endothelium}}(t) = \theta(t) \cdot \gamma_\mu \hat{\mathbf{n}}. \quad (40)$$

In this equation, $\hat{\mathbf{n}}$ denotes the unit vector pointing perpendicular to the x -axis from the cell center towards the line $x = 0$. The magnitude of this velocity is proportional to the part of the cell wall that is inside the area of the endothelium which is approximated by $\theta(t)$. This velocity is equal for all nodes i . An example is given in figure 13, notice that $\theta = \frac{3}{8}$ because three of eight cell wall nodes are located in the endothelium area E .

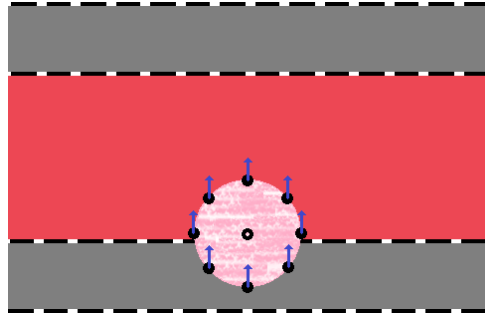


Figure 13: A white blood cell is pushed into the blood vessel.

3.4 Bacteria Behaviour

Area of infection The infected area $B \subset \mathbb{R}^2$ is assumed to be an ellipse with parameters a and b and center $B_c = (x_c^B, y_c^B)$, such that

$$B = \{(x, y) \in \mathbb{R}^2 : \frac{(x - x_c^B)^2}{a^2} + \frac{(y - y_c^B)^2}{b^2} < 1\}. \quad (41)$$

Initial distribution of bacteria The initial bacteria density δ_{initial} and the area of B are used to calculate the initial number of bacteria $N_{\text{bac}} \in \mathbb{N}_{\geq 0}$

$$N_{\text{bac}} = \lfloor \delta_{\text{initial}} \cdot \pi ab \rfloor. \quad (42)$$

We round the expression by taking the floor because N_{bac} is an integer. The initial position $\mathbf{X}_j(0) = (X_j(0), Y_j(0))$ of bacterium $j \in \{0 \dots N_{\text{bac}}\}$ is randomly distributed in B , by

$$\begin{cases} X_j(0) = a(2U_1 - 1) + x_c^B \\ Y_j(0) = \sqrt{b^2(1 - \frac{(X_j(0) - x_c^B)^2}{a^2})} \cdot (2U_2 - 1) + y_c^B \end{cases} \quad (43)$$

In this equation U_1 and U_2 are independent and identically distributed $U(0, 1)$ variables. The bacteria are not uniformly distributed over the ellipse, but this is not an elementary matter.

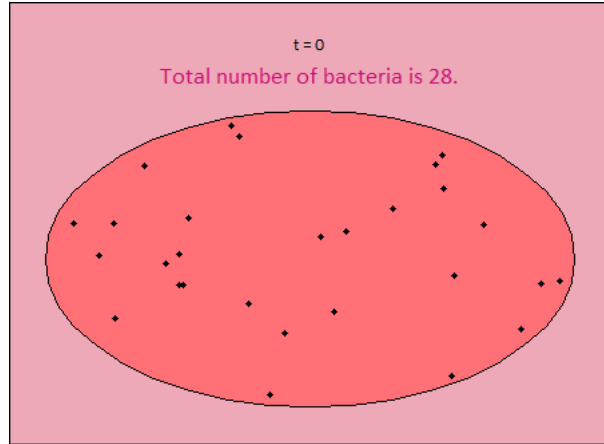


Figure 14: An example of a possible initial distribution of bacteria in an ellipse.

Bacteria movement The movement of the bacteria inside the ellipse B is a random walk in the sense that the velocity in the x - and y -direction is normally distributed with mean 0 and standard deviation σ . Outside the ellipse the movement is also normally distributed, but in this case the mean is 2σ directed towards the ellipse B . This means that whenever a bacterium leaves the ellipse, its movement is heavily biased in order to move back into B . We define the velocity $\mathbf{V}_j(t) = (V_j^x(t), V_j^y(t))$ of bacterium j for discretized time $t \in \{t_0, t_1, t_2, \dots, t_K\}$ by

$$\begin{cases} V_j^x(t_k) = [\sigma Z_1(t_k) + 2\sigma \cdot \chi_B(\mathbf{X}_j(t_k))] \frac{x_c^B - X_j(t_k)}{\|x_c^B - X_j(t_k)\|} \\ V_j^y(t_k) = [\sigma Z_2(t_k) + 2\sigma \cdot \chi_B(\mathbf{X}_j(t_k))] \frac{y_c^B - Y_j(t_k)}{\|y_c^B - Y_j(t_k)\|} \end{cases} \quad (44)$$

In equation (44), $Z_1(t_k)$ and $Z_2(t_k)$ are independent and identically normally distributed $N(0, 1)$ variables. Also,

$$\chi_B(\mathbf{X}_j(t)) = \begin{cases} 0 & \text{for } \mathbf{X}_j(t) \in B \\ 1 & \text{for } \mathbf{X}_j(t) \notin B \end{cases} \quad (45)$$

We could expand this definition to continuous time by interpolating between all defined points $\mathbf{V}_j(t_k)$. Because simulations are made in discretized time, this is not necessary.

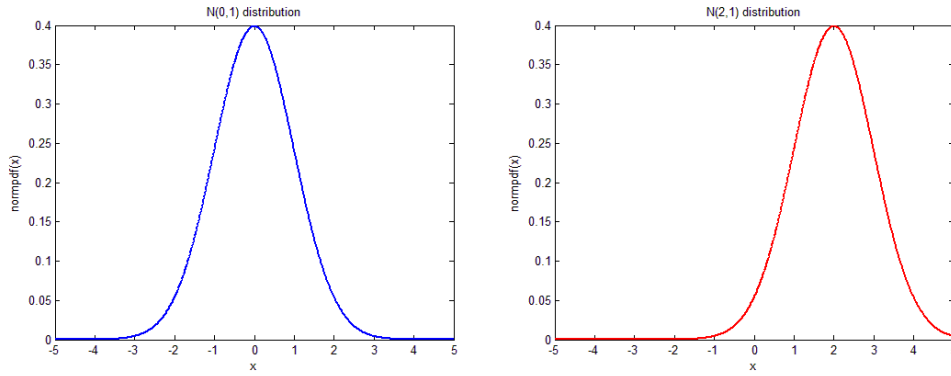


Figure 15: Left: $N(0, 1)$ distribution, Right: $N(2, 1)$ distribution. These distributions define the movement of the bacteria.

Population model For the bacteria population we use a population model with a death rate γ_d and a division rate γ_g . Death and division is simulated by a Poisson process. Also, death occurs if the position of a bacterium is within the active range of a white blood cell. Consider a living bacterium j at time $t = t_1$. Then, for $t_2 > t_1$ and $t_2 - t_1$ small, the following procedure is executed to determine the population status at t_2 .

1. If $X_j(t_1)$ is within the active range of a white blood cell, the bacterium dies. If dead, STOP.
2. With probability $(t_2 - t_1)\gamma_d$, the bacterium dies. This is executed by simulating a uniform $U(0, 1)$ variable. If dead, STOP.
3. With probability $(t_2 - t_1)\gamma_g$, the bacterium reproduces itself. This is also executed by simulating a uniform $U(0, 1)$ variable. The location of the new bacterium is the same as its parent.

At t the bacteria population is $N_{\text{bac}}(t)$. Without interference from white blood cells the expected number of bacteria at $t + \Delta t$, $\Delta t > 0$ small, is

$$N_{\text{bac}}(t + \Delta t) = (N_{\text{bac}}(t)(1 - \gamma_d))\gamma_b. \quad (46)$$

3.5 Degree of Infection

We would like to define how serious an infection really is, because we want to investigate this. Once *enough* bacteria are annihilated by white blood cells, the tissue is able to restore itself again. We choose to define the degree of infection $\text{DoI}(t)$ as

$$\text{DoI}(t) = \frac{\#\text{bacteria alive}}{\text{Area of infection}} = \frac{N_{\text{bac}}(t)}{\pi ab}. \quad (47)$$

Also, we say the infection is not serious anymore if the degree of infection drops below $\epsilon > 0$. The first time this happens is denoted by τ ,

$$\tau = \inf \{t > 0 : \text{DoI}(t) < \epsilon\}. \quad (48)$$

As an example, we show the result of one simulation. The degree of infection is not smooth, because the total number of bacteria is a natural number. We observe an increase and decrease in the degree of infection. Just before $t = 130$ there is no single bacteria left. The degree of infection will be used to investigate the effects of diabetes on the immune system.

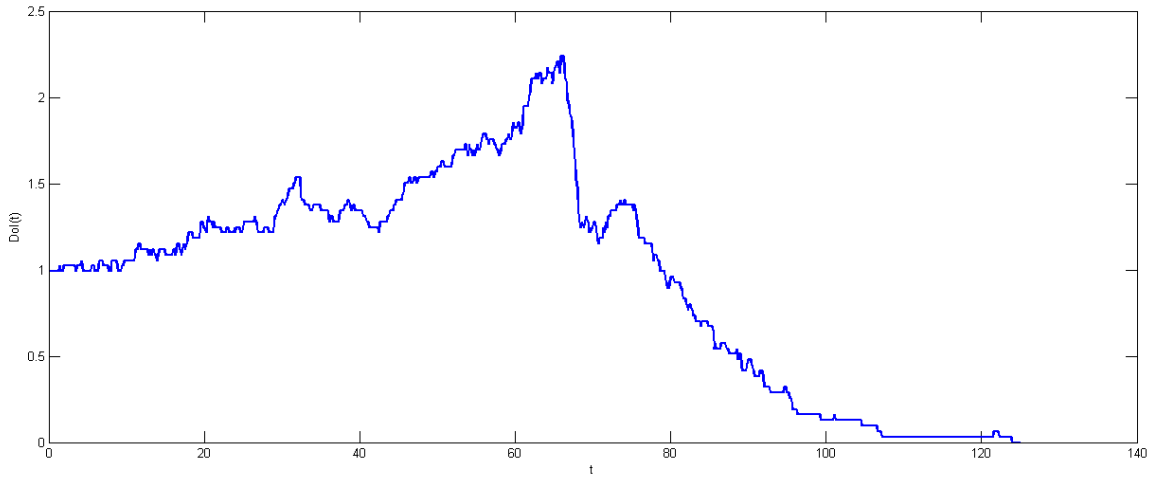


Figure 16: Simulation: The degree of infection plotted against time.

4 Numerical Solution Method

The numerical solution method is based on time and space discretization. After this has been done we can do the time integration and calculate a solution. We treat the solutions from the perspective of one cell influenced by its environment. This environment consists of other white blood cells, bacteria, the blood flow and the endothelium.

4.1 Time Discretization

We consider solutions for $t \in [0, T]$, $T \in \mathbb{R}_{>0}$. For timestep $\Delta t > 0$ we discretize time by defining equidistant points $t_k = k \cdot \Delta t$ in the interval $[0, T]$, $k \in \{0, 1, 2, \dots, K\}$ such that $t_K = T$. When T is divisible by Δt , we meet this condition and $K = \frac{T}{\Delta t} + 1$.

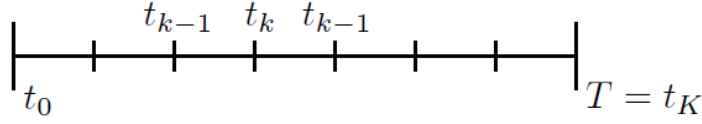


Figure 17: Scheme of the time grid.

4.2 Space Discretization

White blood cell Consider one circle shaped cell with radius r and initial cell center position $(0, 0)$. The cell wall discretization is defined by N equidistant points $\mathbf{x}_i = (x_i, y_i)$, $i \in \{0, 1, 2, \dots, N\}$ as

$$\begin{cases} x_i &= r \cos(\frac{2\pi i}{N}) \\ y_i &= r \sin(\frac{2\pi i}{N}) \end{cases} . \quad (49)$$

Bacterium A bacterium $X_j = (X_j, Y_j)$ already is a point, so its location is discrete by definition.

4.3 Time Integration

The velocity of well wall node i at time t is defined as the sum of all components.

$$\mathbf{v}_i(t) = \mathbf{v}_i^{\text{contraction}}(t) + \mathbf{v}_i^{\text{chemotaxis}}(t) + \mathbf{v}_i^{\text{anti-overlap}}(t) + \mathbf{v}_i^{\text{bloodflow}}(t) + \mathbf{v}_i^{\text{endothelium}}(t). \quad (50)$$

For completeness, we summarize the definitions of these components here.

$$\mathbf{v}_i^{\text{contraction}}(t) = \alpha(\mathbf{x}_c(t) + \hat{\mathbf{x}}_i - \mathbf{x}_i(t)) \quad (51)$$

$$\mathbf{v}_i^{\text{chemotaxis}}(t) = \beta^*(t) \nabla c(t, \mathbf{x}_i(t)) \quad (52)$$

$$\mathbf{v}_i^{\text{anti-overlap}}(t) = \sqrt{\frac{\Delta}{r - \Delta}} \cdot \frac{\mathbf{x}_c^A - \mathbf{x}_c^B}{\|\mathbf{x}_c^A - \mathbf{x}_c^B\|} \quad (53)$$

$$\mathbf{v}_i^{\text{bloodflow}}(t) = u(t, (x, y)) \cdot \hat{\mathbf{i}} \quad (54)$$

$$\mathbf{v}_i^{\text{endothelium}}(t) = \theta(t) \cdot \gamma_\mu \hat{\mathbf{n}} \quad (55)$$

It is important to remember that all these expressions are explicitly given and only depend on the location of node i and time t .

Euler Forward As time integration method we use the well-known Euler forward method. We choose this method instead of more stable methods like Modified Euler or Runge-Kutta, because it is simple to implement and does not distract us by complicating the model. When we applicate this method we have the following expression for the time integration:

$$\mathbf{x}_i(t_{k+1}) = \mathbf{x}_i(t_k) + \Delta t(\mathbf{v}_i(t_k)). \quad (56)$$

4.4 Convergence one cell one source

We test the numeric convergence by using a basic model, which is graphically shown in the section of numerical experiments. We have the initial situation of a cell with radius $r = 1$ with its center on the origin and a source on the positive x -axis, at $x = 2.85$, away from the cell. The cell will move only in the x -direction since this is the direction of largest increase of concentration. At $t = 10$ we write down the x -coordinate of the cell center. We do this for different timesteps. This results in the following table. The relative difference is the difference related to the radius of the cell.

timestep	0.01	0.025	0.05	0.1	0.5
x -coordinate cell center	1.1217	1.1238	1.1277	1.1538	1.4063
difference	0.0021	0.0039	0.0261	0.2525	-
relative difference	0.21%	0.39%	2.6%	25%	-

We observe that in this test case we reach numerical convergence for small enough timesteps. It is not necessary to use timesteps smaller than 0.025, since this hardly influences the result. For larger experiments it is acceptable to use a timestep of 0.1 at most. By halving the timestep we would improve only 0.39%.

5 Numerical experiments

In this section we will show the results of some basic numerical experiments. In every simulation we use the quasi steady-state solution to the diffusion equation, because it does not require temporal integration. This makes the calculations faster. In all simulations we use the input-data of the table below.

Parameter	Symbol	Value
Timestep	Δt	0.1
Cell mobility	β	1
Strength of point source	γ	1
Cell contraction	α_0	1
Cell area min	$A_{\min}$	$\frac{A_0}{\sqrt{2}}$
Cell area max	$A_{\max}$	$A_0\sqrt{2}$

5.1 One Cell One Source

The following experiment involves one cell and one bacteria. The white area is the cell with its nucleus in the center. One bacterium is located close to the cell and does not move. As expected, the cell migrates towards the bacterium and deformats because of its flexibility. When the bacterium is gone, the cell changes back to its original shape.

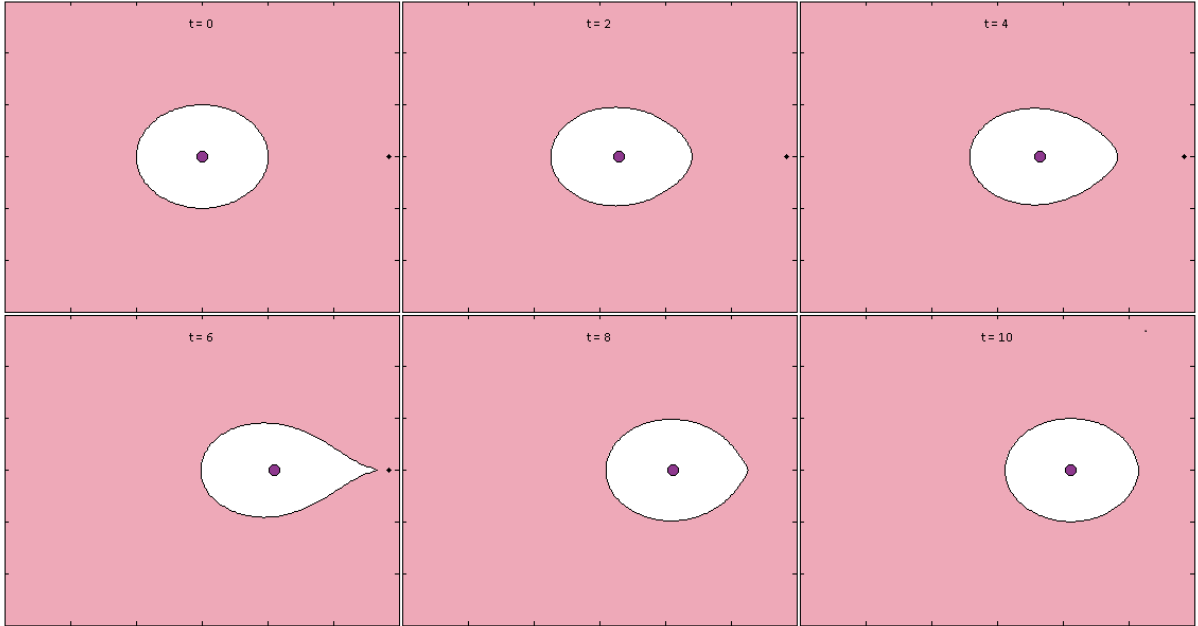


Figure 18: One cell annihilating a bacterium.

5.2 Multiple Sources

We will now show the results of two experiments involving multiple sources.

1 In the first multiple sources experiment, we locate four bacteria around the cell. The distances between the bacteria and the cell are all equal. We observe that the cell center does not move, but the cell deforms and creates some sort of tentacles. Again, when all bacteria are gone, the cell changes back to its original shape.

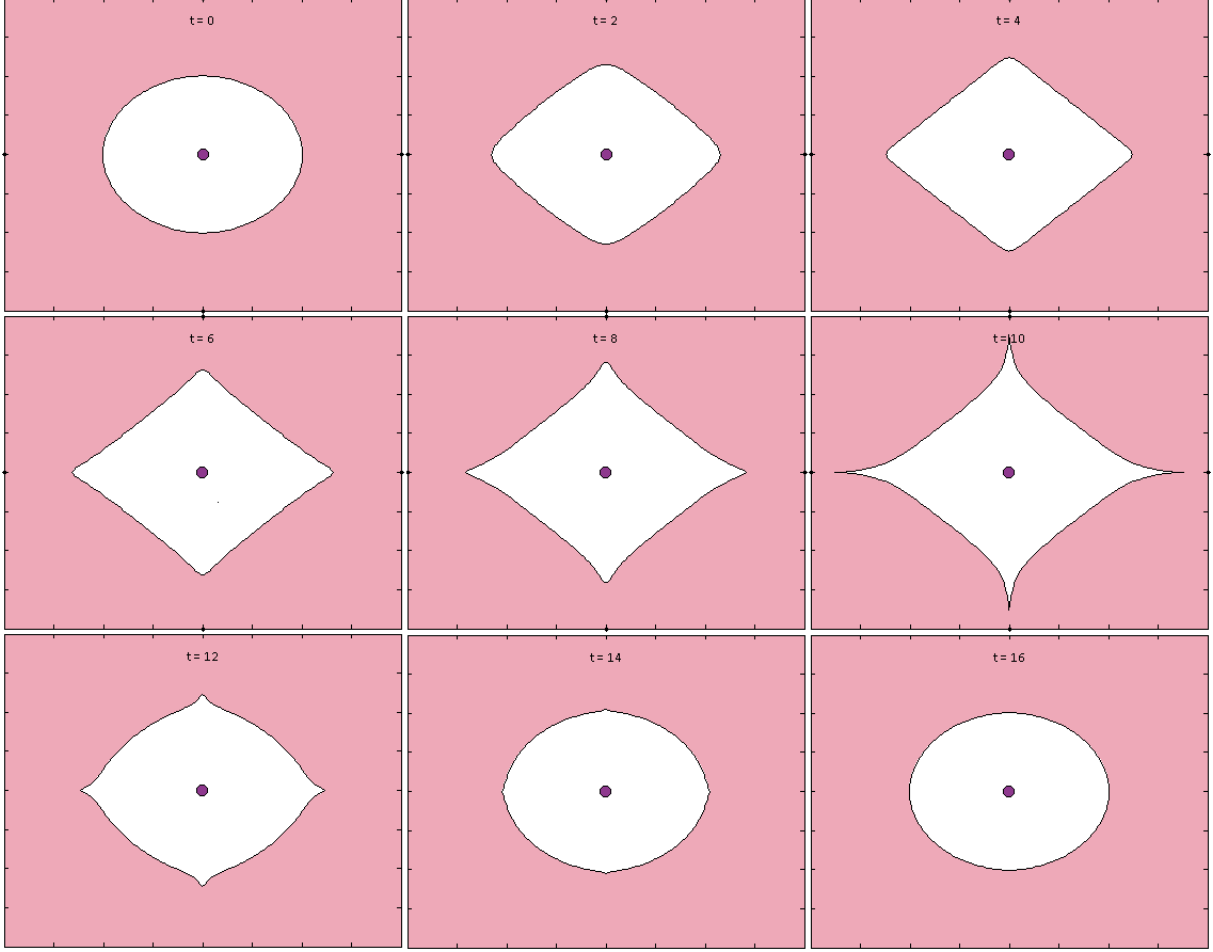


Figure 19: One cell annihilating four bacteria.

2 As for the second experiment, we define five moving bacteria around one cell. These bacteria travel according the following parametric equations:

$$\begin{aligned}
 X_1(t) &= 3 + 2.2 \sin(0.2t), & Y_1(t) &= -4 + 1.5 \cos(0.2t), \\
 X_2(t) &= -2.5 - 0.008t, & Y_1(t) &= 1.5, \\
 X_3(t) &= 4 \sin(0.08t), & Y_1(t) &= 4 \cos(0.08t), \\
 X_4(t) &= -3.5 - 0.008t, & Y_1(t) &= -3.5, \\
 X_5(t) &= 3.5 - 0.008t, & Y_1(t) &= 1.5.
 \end{aligned}$$

We observe that the cell eventually annihilates all bacteria.

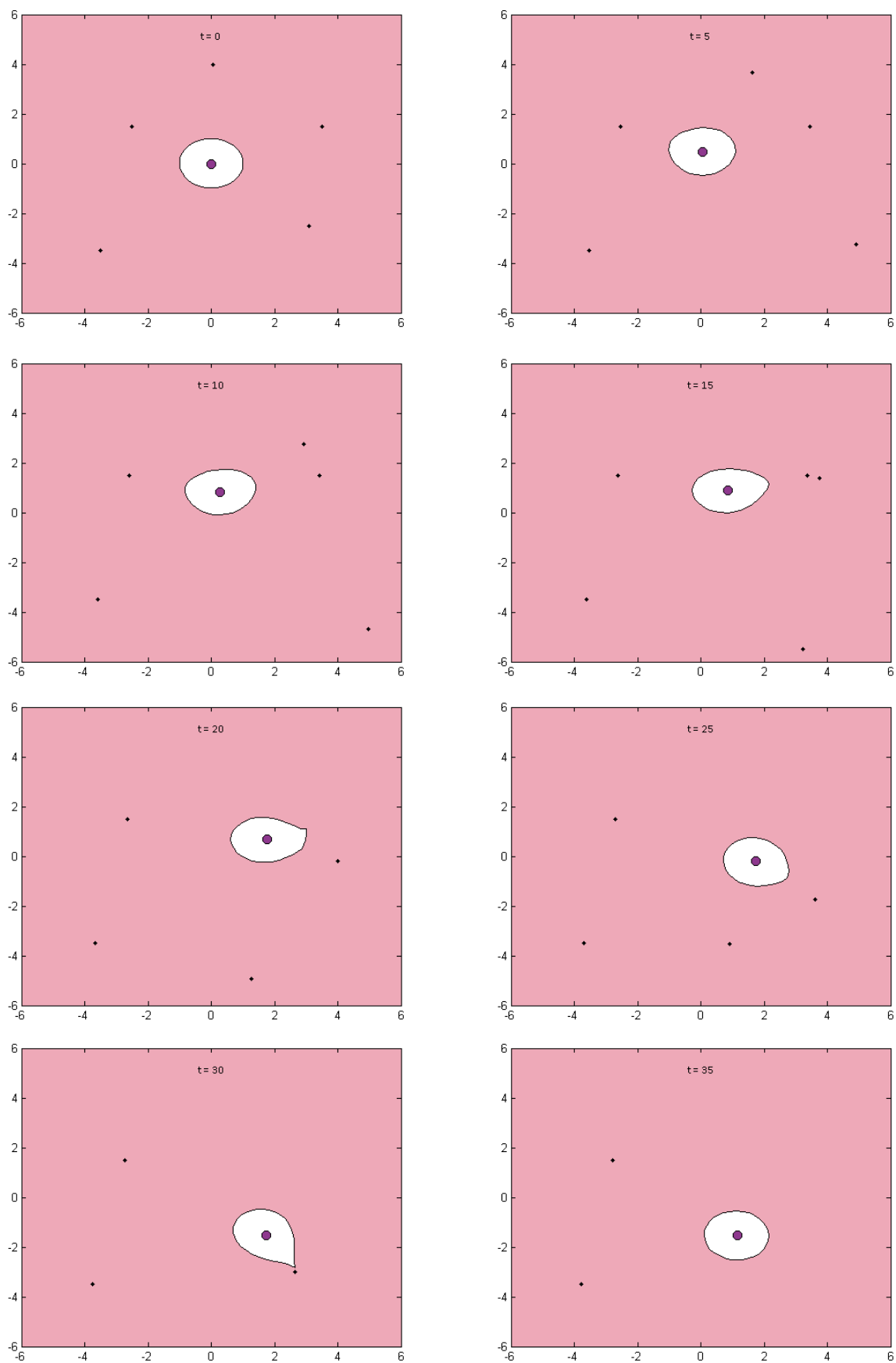
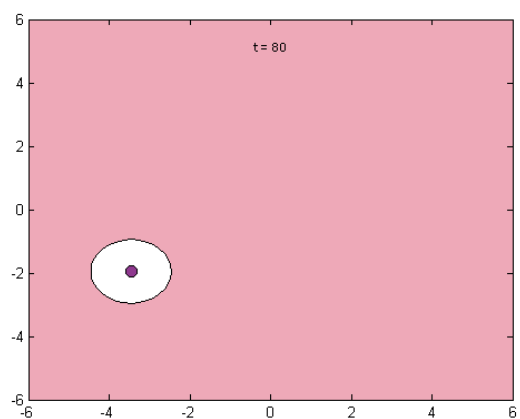
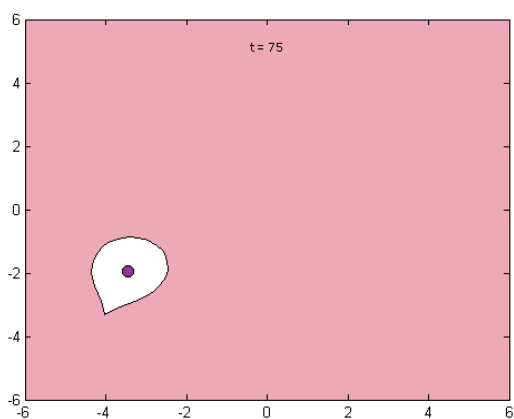
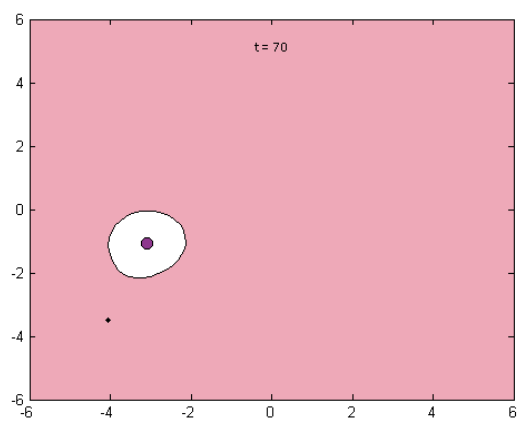
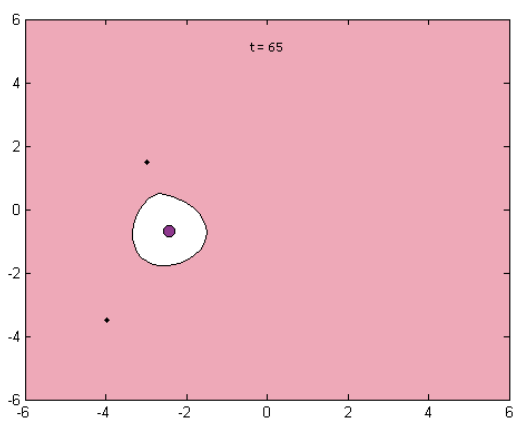
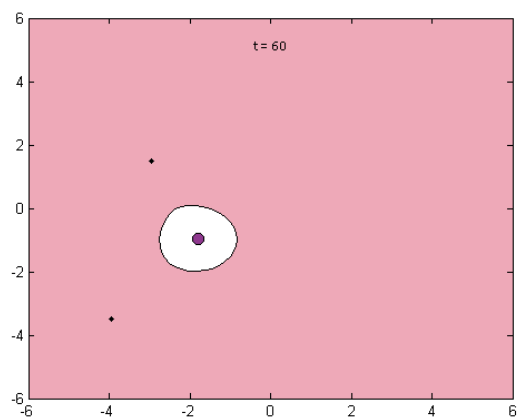
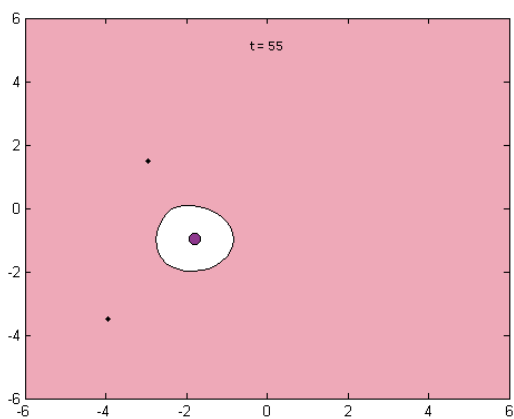
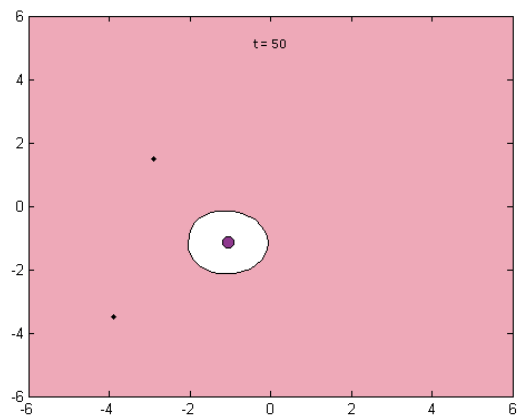
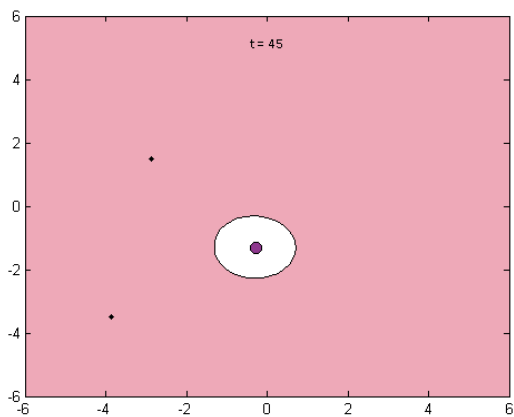


Figure 20: One cell annihilating five bacteria(1), we skip $t = 40$ because nothing happens there and this fits better on the page.



Cell Shape Index During the process of annihilating bacteria, the cell shape changes, which is plotted below in a graph of the cell shape index. The local minima of the CSI are exactly the moments when the cell neutralizes a bacterium.

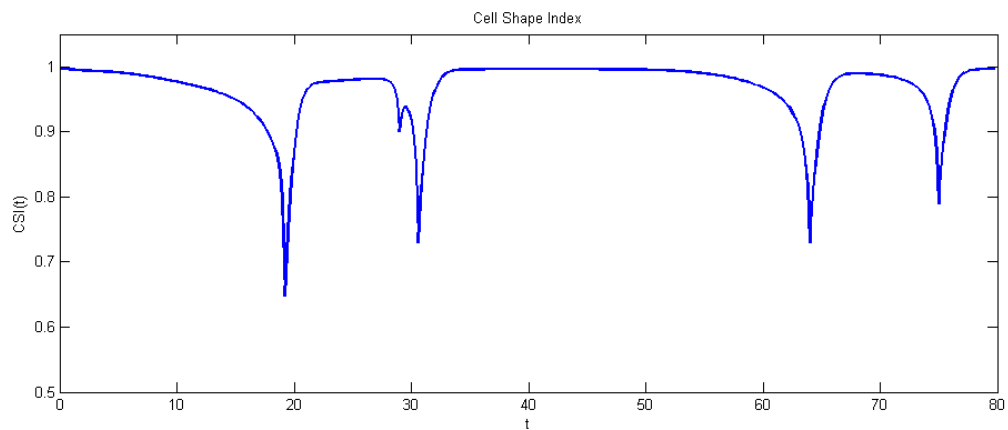


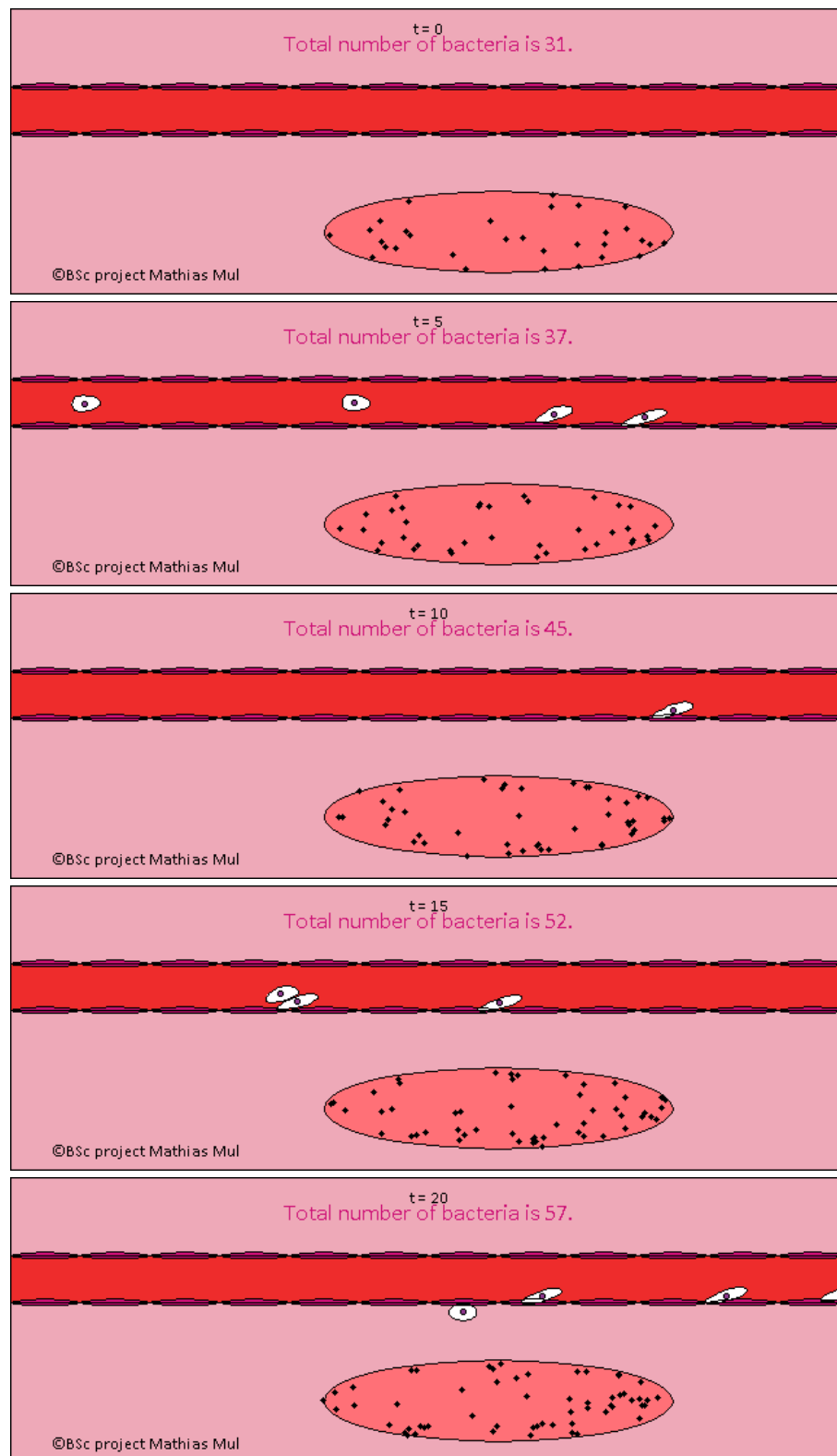
Figure 22: The cell shape index changes in time.

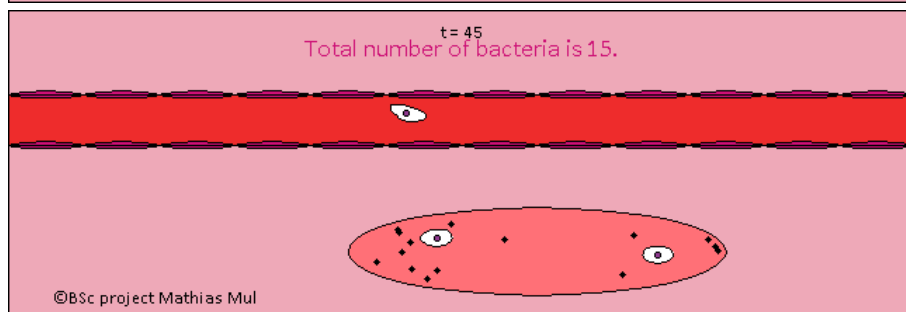
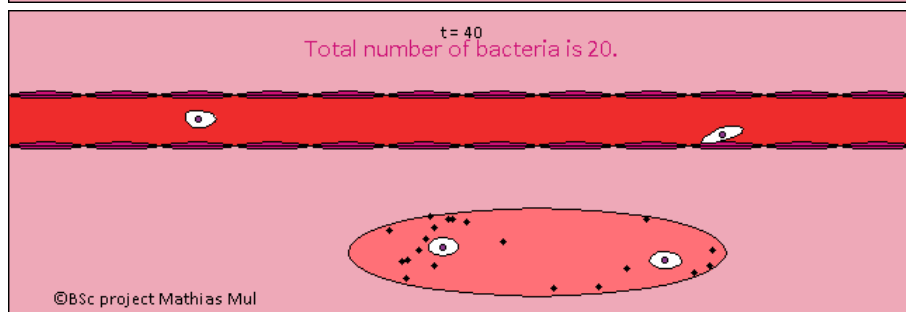
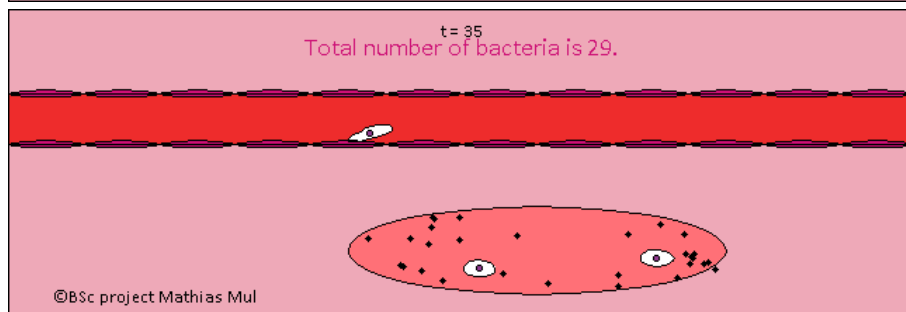
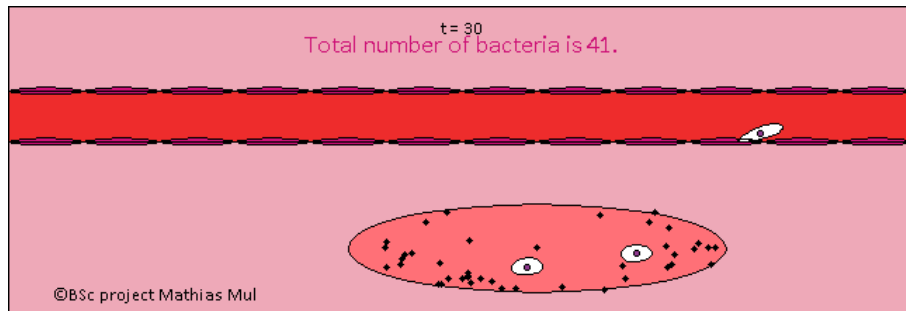
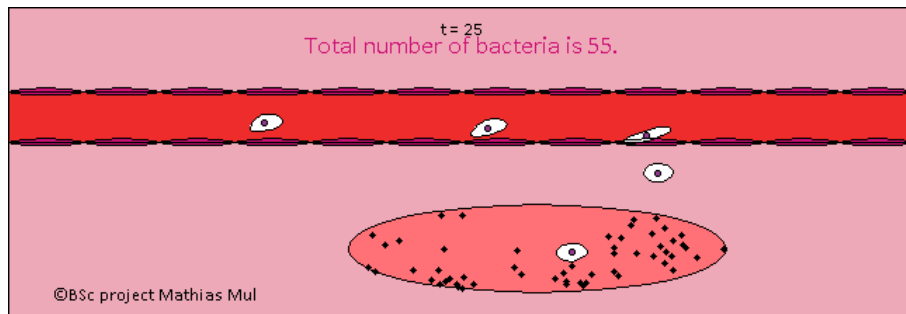
6 Computer Simulations of the Blood Vessel

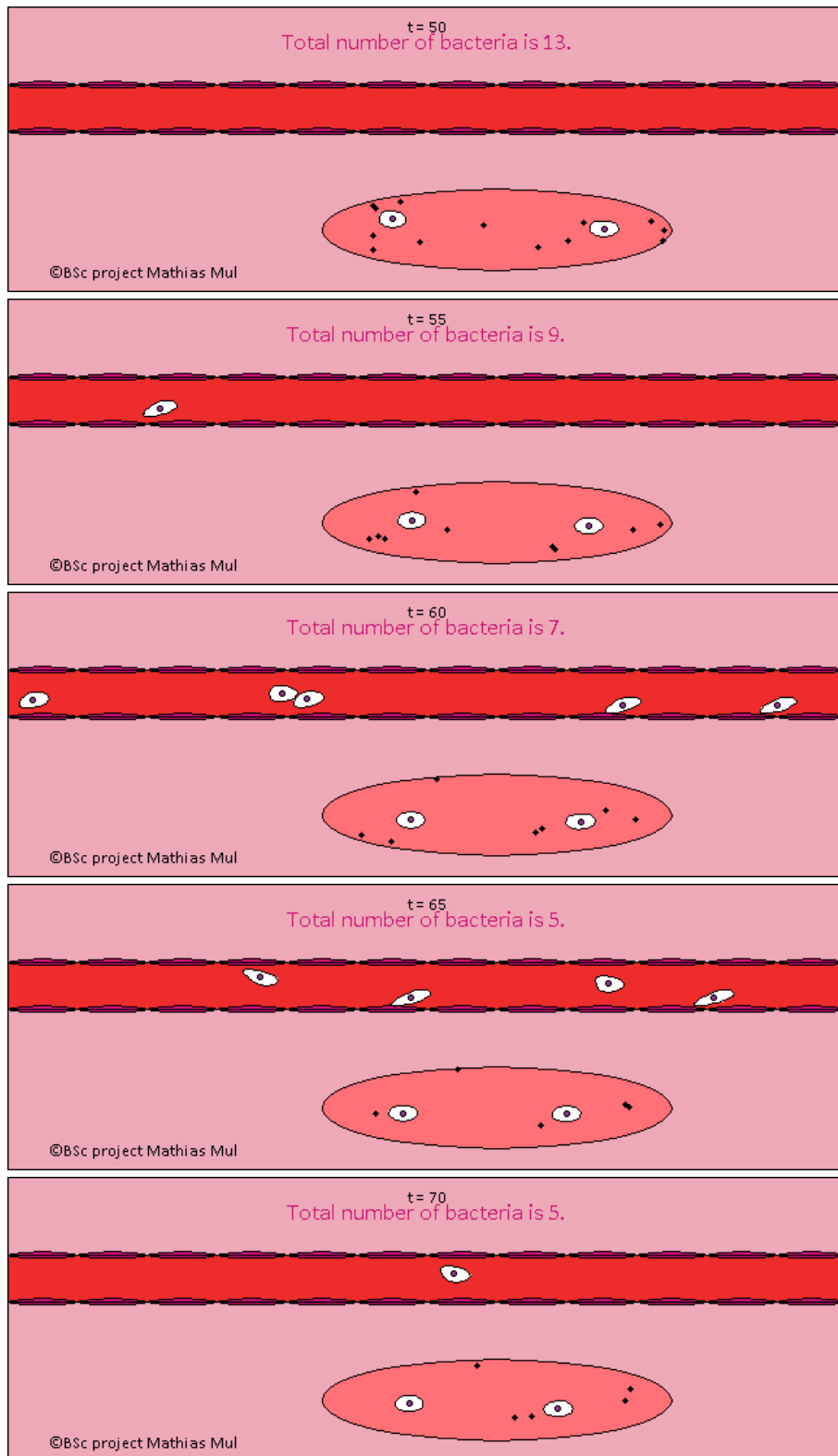
This section will treat a simulation of the immune system responding to an infection. At $t = 0$ there is an infected area close to a blood vessel. At first, white blood cells don't seem to notice the bacteria and pass by. After a while, the number of bacteria increased to a certain level and the white blood cells are starting to respond to them. They move through the endothelium out of the blood vessel towards the infected area. Then, they start to clean the infection and in the end all bacteria are killed – the tissue is restored. For this simulation we have used the following set of input values:

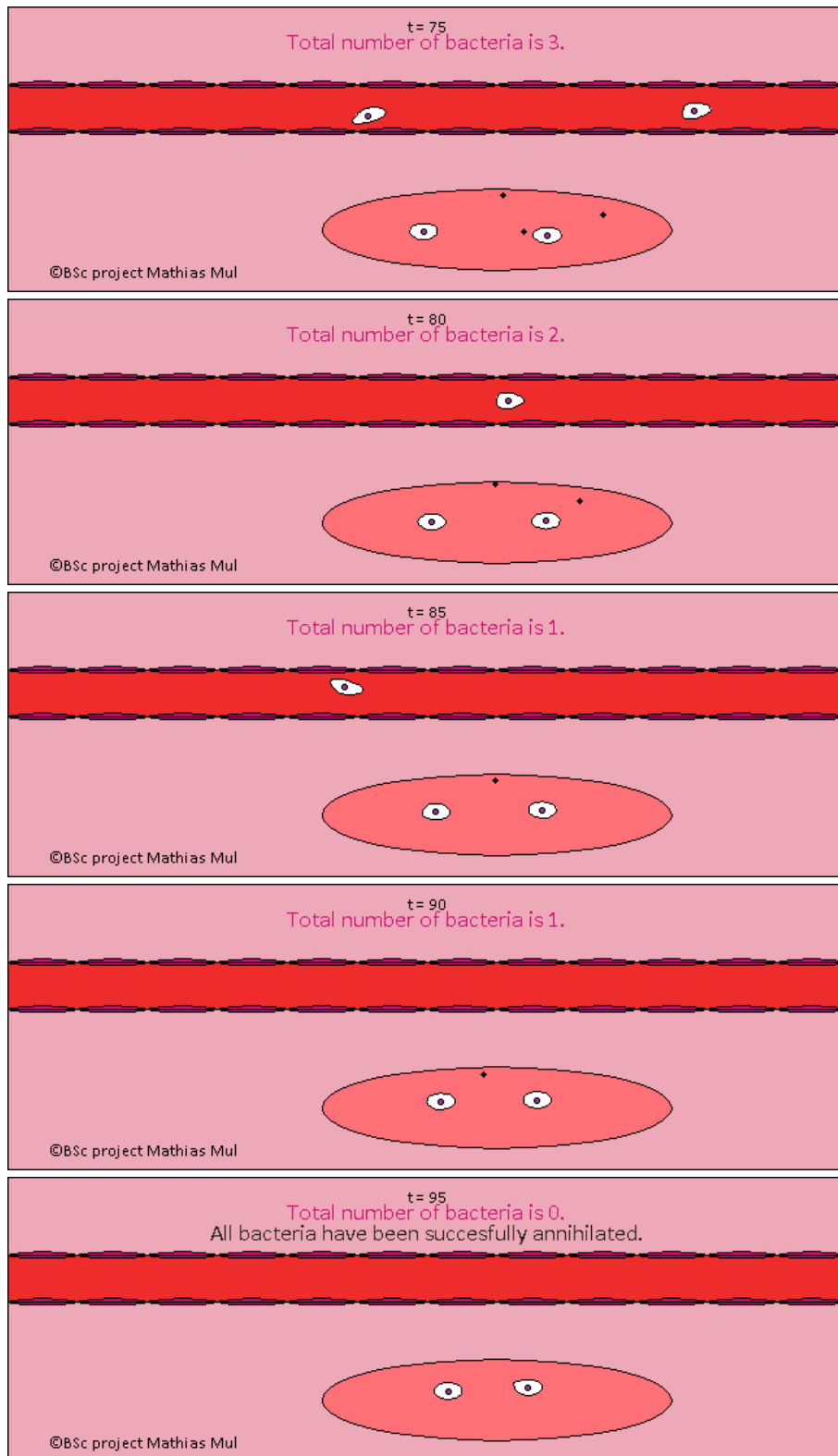
Parameter	Value
Timestep (Δt)	0.1
Radius white blood cells	0.4
Cell area min	$\frac{A_0}{\sqrt{2}}$
Cell area max	$A_0\sqrt{2}$
White blood cell concentration	0.05
Active range	0.15
Cell mobility(β)	1
Cell contraction	5
Wound center	(2, -6)
Wound size horizontal	10
Wound size vertical	4
Length of blood vessel(L)	24
Thickness blood vessel(h)	2
Thickness endothelium(d)	0.3
Stiffness endothelium(μ)	10
Initial bacteria density	1
Mobility of bacteria (σ)	0.8
Diffusion coefficient(D)	2
Strength of point source(γ)	1
Reproduction rate bacteria(γ_b)	0.05
Death rate bacteria (γ_d)	0.03
Blood velocity (maximum)	4.5
Length of cardiac cycle	2

6.1 Snapshots









The degree of infection, which directly depends on the total number of bacteria, is plotted against the time below. We notice at first an increase of the DoI, but once white blood cells entered the infected area it will start to decrease. After $t = 95$ the DoI is zero, since all bacteria are dead.

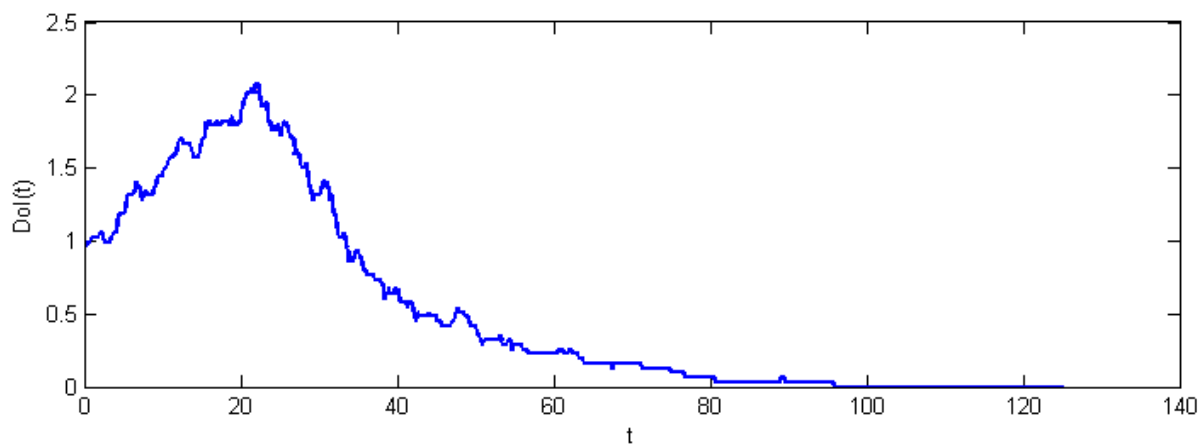


Figure 23: Degree of Infection plotted against time.

It is also interesting to see the average result of multiple simulations. We repeat the simulation with exactly the same input 10 times. As we can see in figure 24, there is a lot of uncertainty in the degree of infection.

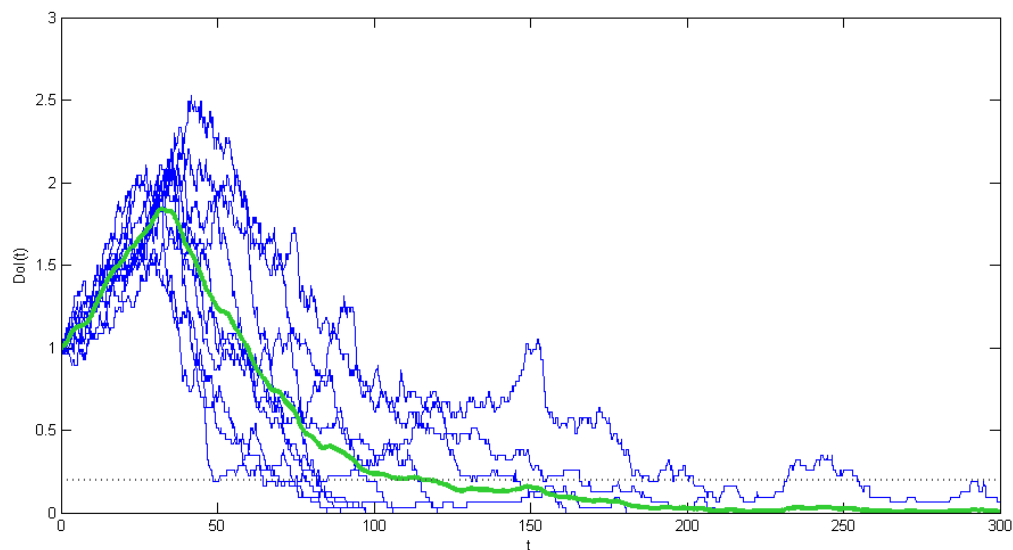


Figure 24: Ten different simulations of the immune system responding to an infection. The green line is the average.

The first time that the degree of infection is lower than $\epsilon = 0.2$ is denoted by τ . For every simulation we obtain a time τ . By observing the list of different values of τ we see that this stochastic variable τ has a high standard deviation.

τ
48.8
69.7
74.0
81.8
82.5
83.1
97.8
131.1
154.1
185.7

The standard deviation of τ is $\sigma_\tau = \sqrt{\text{var}(\tau)} = \sqrt{\text{E}[\tau^2] - (\text{E}[\tau])^2} \approx 40.47$. The mean of τ is 100.86.

6.2 Parameter Variation

One of the effects of diabetes is an increased stiffness of the endothelial cells. This has an effect on the immune system, because white blood cells have to put more effort in pushing themselves through the endothelium. We investigate this phenomenon by varying the endothelium stiffness parameter μ in our model. With ten simulations we determine for each μ the average of τ and its standard deviation. The resulting values are listed in the table below. All values of other parameters are unchanged.

μ	$\text{E}[\tau]$	σ_τ
10	101	40
12.5	122	19
15	146	24
17.5	175	48
20	181	54
22.5	212	54
25	220	43

When we put this data into a graph, we arrive at figure 25. Our expectation that τ would be larger for a higher endothelium stiffness is confirmed. As we can derive from the plot, it seems to be a linear relationship.

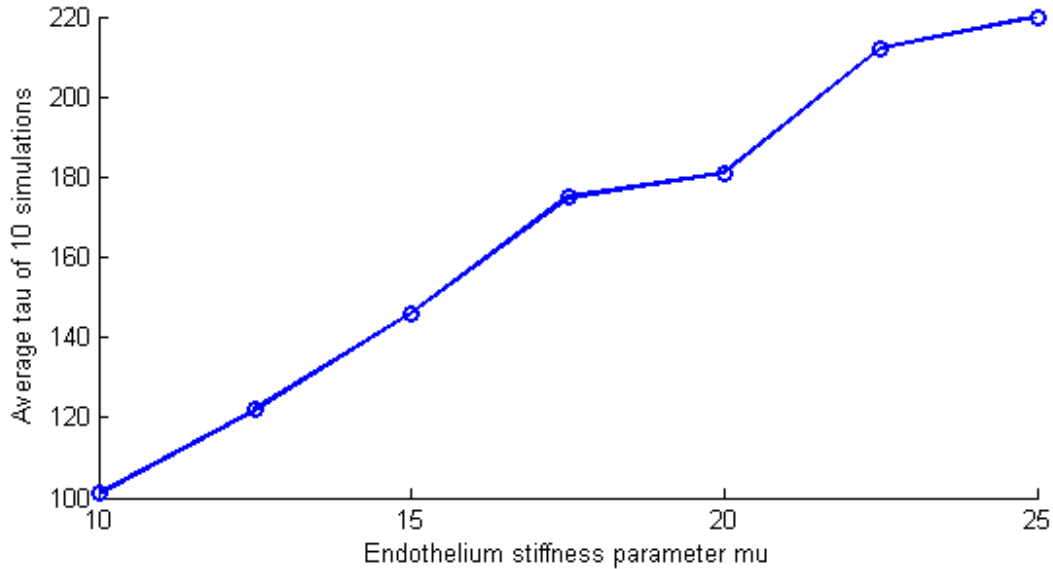


Figure 25: The results of varying the endothelium stiffness parameter.

7 Discussion and Conclusion

Discussion For all experiments and simulations we have used the quasi-steady state solution. For a large diffusion coefficient and little movement of sources, this is an accurate method of describing chemotaxis. But, in the case of a small diffusion coefficient and fast moving source this model does not do what we want. In this case a cell is likely to move not directly towards a source, thus we would need the transient case solution. However, this will increase the computation time drastically, since we need to additionally integrate over time. It would be a good idea to investigate the relationship between bacteria movement and diffusion of chemicals. If we know more about this relationship we can continue optimizing models for cell migration and deformation influenced by bacteria in their environment. Also, we assume that the domain of computation for the chemical concentration is infinite, which approaches the case of a large domain of computation. If this domain would be finite, we could still use specified Green's functions. This has the disadvantage that we end up with infinite series that have to be truncated. However, we assume that our computational domain is large compared to the cell sizes and intercellular distances, thus we are able to approach this by using an infinite domain. The fact that our method is based on a single-term solution makes it elegant.

The anti-overlap term used to describe hard impingement, cells touching each other, is not based on mechanics. The model could be improved by investigating literature about this subject as well as approaching hard impingement node-wise. Even though we could change this, we note that our current method is very practical and time efficient to use.

Concerning the model for the immune response, several improvements can be made. The blood vessel is described by a rectangle, but we would like to expand this to an arbitrary shape. Furthermore, the behaviour of bacteria could be changed, because bacteria are not always restricted to a certain area. If a bacterium enters the blood vessel by itself it should move along with the blood flow. The process of diapedesis is also an opportunity for improvement. If we would have a node-based model for hard impingement we could define special endothelial cells with nodes on their cell walls, so we are able to visualize the process of white blood cells pushing themselves through the endothelium.

Conclusion A model for the immune system has been constructed. White blood cells enter a blood vessel and push themselves through the vessel walls towards infected tissue. White blood cells move by making use of the chemotaxis principle. This movement is modeled and the simulations are time efficient because we use of explicit expressions. The time needed to simulate the immune response model without visualizing is less than half a minute. If the number of bacteria is not significant, white blood cells will not penetrate the vessel walls. The immune response model has been used to investigate the effect of diabetes on the immune response system. The time necessary to clean infected tissue is significantly longer for higher values of endothelium stiffness.

8 Appendix

This is the Matlab-code used in the computer simulations of the blood vessel. The code below might not be inviting to read, but it is written down. For this or any other file with code, feel free to send me an e-mail with your request to the adress just below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BSc project M.M. Mul (mathias_mul@hotmail.com)
Coordinator Dr.ir. F.J. Vermolen (f.j.vermolen@tudelft.nl)
Period february-june 2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Advanced wound
clear all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Model parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%White blood cell parameters
white_blood_cell_size = 0.4; % Radius of white blood cell (space
    units)
wbc_concentration = 0.05; % Concentration of white blood
    cells in blood
active_range_wbc = 0.15; % Range in which bacteria are
    annihilated (space units)
cap_velocity_bac_influence = 10; % The velocity of wbc induced by
    bacteria influence is capped
rvw_wbc = 0.1; % Random vertical movement of wbc
    inside blood vessel
beta = 1; % Mobility parameter of the white
    blood cell boundary
alpha = 5; % Contraction coefficient white
    blood cells

%Wound parameters
wound_center = [2, -6]; % Center of wound
wound_horizontal_size = 10; % Horizontal size of wound (space
    units)
wound_vertical_size = 4; % Vertical size of wound (space
    units)

%Blood vessel parameters
L_blood_vessel = 24; % The length of the blood vessel (
    integer space units)
thickness_blood_vessel = 2; % Thickness of blood vessel (space
    units)
thickness_blood_vessel_cell = 0.3; % Thickness of blood vessel cell (
    space units)
%Resistance parameters
mu = 50; % Stiffness coefficient of blood
    vessel cells
resistance_bv = mu/10; % Resistance of the endothelium (
    based on endothelium stiffnes)
beta_mu = beta/mu; % Decreased mobility parameter for
    white blood cells (based on endothelium stiffnes)
```

```

%Bacteria parameters
Victory_factor_bacteria = 20;           % When the bacteria population has
    been multiplied by this factor , the simulation stops
bac_density = 1;                        % bac density at t=0
sigma_bac = 0.8;                        % Mobility of bacteria
D = 2;                                  % Diffusion coefficient of bacteria
br_bac = 0.05;                          % Birth rate of bac population
dr_bac = 0.03;                          % Death rate of bac population

%Heartbeat parameters
Blood_flow = 4.5;                       % Blood flow velocity (velocity
    units)
L_cc = 2;                               % Length of cardiac cycle (time
    units)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Plot Window %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Complete situation
plot_window = [-L_blood_vessel/2,L_blood_vessel/2,min(-5,wound_center(2)
    -1.5*0.5*wound_vertical_size),max(5,wound_center(2)+1.5*0.5*
    wound_vertical_size)];
%Close-up on the wound
% plot_window = [wound_center(1)-wound_horizontal_size,wound_center(1)+
    wound_horizontal_size,wound_center(2)-wound_vertical_size,wound_center
    (2)+wound_vertical_size];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Time discretization %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T = 300;                                % calculations are made on time interval
    [0,T]
dt = 0.1;                               % time interval
K = T/dt+1;                             % total number of timesteps

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Blood vessel cells %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% & Blood vessel %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N_bvc = 12;                             % Number of
    nodes on blood vessel cell
L_blood_vessel_cell = 2;                 % Length of
    blood vessel cell
N_blood_vessels = ceil(L_blood_vessel/L_blood_vessel_cell); % Number of
    blood vessels on one side
%Defining starting position blood vessel cell centers
xc_bvc_up = (0:L_blood_vessel_cell:L_blood_vessel)-L_blood_vessel/2+
    L_blood_vessel_cell/2;
yc_bvc_up = ones(1,N_blood_vessels+(L_blood_vessel/L_blood_vessel_cell~=0))
    *(1/2)*(thickness_blood_vessel+thickness_blood_vessel_cell);
xc_bvc_down = xc_bvc_up;
yc_bvc_down = ones(1,N_blood_vessels+(L_blood_vessel/L_blood_vessel_cell
    ~=0))*(-1/2)*(thickness_blood_vessel+thickness_blood_vessel_cell);
% Defining the shape of the blood vessel cells

```

```

x_initial_bvc = (1/2)*L_blood_vessel_cell*cos(((0:N_bvc-1)')*2*pi/N_bvc)';
y_initial_bvc = (1/2)*thickness_blood_vessel_cell*sin(((0:N_bvc-1)')*2*pi/
    N_bvc)';
%Defining starting postition nodes on cell boundary of blood vessel cells
x_bvc_up = ones(N_blood_vessels+(L_blood_vessel/L_blood_vessel_cell~=0),1)*
    x_initial_bvc + (ones(N_bvc,1)*xc_bvc_up)';
y_bvc_up = ones(N_blood_vessels+(L_blood_vessel/L_blood_vessel_cell~=0),1)*
    y_initial_bvc + (ones(N_bvc,1)*yc_bvc_up)';
x_bvc_down = ones(N_blood_vessels+(L_blood_vessel/L_blood_vessel_cell~=0),
    1)*x_initial_bvc + (ones(N_bvc,1)*xc_bvc_down)';
y_bvc_down = ones(N_blood_vessels+(L_blood_vessel/L_blood_vessel_cell~=0),
    1)*y_initial_bvc + (ones(N_bvc,1)*yc_bvc_down)';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Heart beat %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
max_flow = Blood_flow; %Blood flow velocity
min_flow = max_flow*(2/3);
lsf = (1/4); %length of the systolic phase of one heart
    beat [length systolic phase + length diastolic phase=1]
%Calculation of periodic heart beat function using splines for
    interpolation.
hbt_alpha = 0;
hbt_beta = L_cc*lsf;
hbt_gamma = L_cc;
x_hb_1 = 0:dt*lsf:hbt_beta;
x_hb_2 = hbt_beta:dt*lsf:L_cc;
t_hb_1=(1/(hbt_beta-hbt_alpha))*(x_hb_1-hbt_alpha);
t_hb_2=(1/(L_cc-hbt_beta))*(x_hb_2-hbt_beta);
f_hb_1 =(1-t_hb_1).*min_flow+t_hb_1.*max_flow+t_hb_1.*(1-t_hb_1).*(-(
    max_flow-min_flow).*(1-t_hb_1)+(max_flow-min_flow).*t_hb_1);
f_hb_2 =(1-t_hb_2).*max_flow+t_hb_2.*min_flow+t_hb_2.*(1-t_hb_2).*(-(
    max_flow-min_flow).*(1-t_hb_2)+(max_flow-min_flow).*t_hb_2);
f_hb=[f_hb_1 ,f_hb_2(2:end-1)];
x_hb=[x_hb_1 ,x_hb_2(2:end-1)];
pv = [ repmat(f_hb,1,(ceil(T/L_cc))), f_hb_2(end) ];
x_pv = 0:dt*lsf:(ceil(T/L_cc)*L_cc;
poiseuille_velocity_coefficient = interp1(x_pv,pv,(0:dt:(ceil(T/L_cc))*L_cc
    ));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% White blood cells %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Space discretization %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining starting postition nodes on white blood cell boundary
N_wbc = 32; %Number of nodes on white blood cell
% Defining the shape of the white blood cells
x_initial_wbc = white_blood_cell_size*cos(((0:N_wbc-1)')*2*pi/N_wbc)';
y_initial_wbc = white_blood_cell_size*sin(((0:N_wbc-1)')*2*pi/N_wbc)';
% Poisson parameter for white blood cells in blood: Assigned value by other
    parameters.
lambda = wbc_concentration*max_flow;

% Contraction white blood cells

```

```

E_Twbc = round(2*lambda*T+1); % Memory allocation: Twice the
    expected total white blood cells
alpha_wbc_0 = alpha * ones(1,E_Twbc); % cell contraction forcing
    parameter
alpha_wbc = zeros(1,E_Twbc);
alpha_wbc(1,:) = alpha_wbc_0;
A_wbc_0 = polyarea(x_initial_wbc , y_initial_wbc);
A_wbc_min = 0.25*A_wbc_0;
A_wbc_max = 2*A_wbc_0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% bacteria %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Defining area of bac population (ellipsoid)
ec = wound_center; %ellipsoid_center;
ea = 0.5*wound_horizontal_size; %horizontal length;
eb = 0.5*wound_vertical_size; %vertical length;
x_bac_ellipsoid = ea*cos(((0:39)')*2*pi/40)'+ec(1);
y_bac_ellipsoid = eb*sin(((0:39)')*2*pi/40)'+ec(2);
N_bac_initial = floor(bac_density*polyarea(x_bac_ellipsoid , y_bac_ellipsoid)
    );
N_bac = N_bac_initial;
max_N_bac = Victory_factor_bacteria*N_bac;
N_bac_allocation = ceil(2*max_N_bac);
bac_alive = logical([ones(1,N_bac),zeros(1,N_bac_allocation-N_bac)]);
%Defining population parameters

%Each time interval there is a
    number of r_bac timesteps where
    reproduction is allowed

%Define random start positions of the bacteria
x_bac = zeros(1,N_bac_allocation);
y_bac = zeros(1,N_bac_allocation);
x_bac(1:N_bac) = ea*(2*rand(1,N_bac)-1)+ec(1);
y_bac(1:N_bac) = sqrt(eb^2*(1-((x_bac(bac_alive)-ec(1)).^2/ea^2))).*(2*rand
    (1,N_bac)-1)+ec(2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Allocating memory %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x_wbc = zeros(E_Twbc,N_wbc); %Allocation of memory for white blood
    cells
y_wbc = zeros(E_Twbc,N_wbc);
xc = zeros(E_Twbc,1);
yc = zeros(E_Twbc,1);
vx_wbc = zeros(E_Twbc,N_wbc);
vy_wbc = zeros(E_Twbc,N_wbc);
wbc = 0; %Number of defined white blood cells
wbc_visible=(1:wbc);
vx_bac = zeros(1,N_bac_allocation);
vy_bac = zeros(1,N_bac_allocation);
Animation(1:K) = struct('cdata', zeros(343,436,3),'colormap',colormap(hsv
    (256)));
DoI = zeros(K,1); % Degree of infection
DoI(1) = bac_density; % By definition

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% Initial image and pause %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cla;
set(gca,'ytick',[0]);
set(gca,'xtick',[0]);
box on;
%Plot the area of the wound
patch(x_bac_ellipsoid,y_bac_ellipsoid,(1/255)*[255, 112, 119]);
hold on
%coloring the blood vessel
patch([-L_blood_vessel/2,-L_blood_vessel/2,L_blood_vessel/2,L_blood_vessel
/2],[-(thickness_blood_vessel+thickness_blood_vessel_cell)/2,(
thickness_blood_vessel+thickness_blood_vessel_cell)/2,(
thickness_blood_vessel+thickness_blood_vessel_cell)/2,-(
thickness_blood_vessel+thickness_blood_vessel_cell)
/2],1/255*[238,44,44]);
hold on;
%plotting the bacteria
plot(x_bac(bac_alive),y_bac(bac_alive),'k. ');
%plotting the blood vessel cells
for i=1:N_blood_vessels
    patch(x_bvc_up(i,:),y_bvc_up(i,:),1/255*[205,16,118]);
    hold on
    patch(x_bvc_down(i,:),y_bvc_down(i,:),1/255*[205,16,118]);
    hold on
end;
%Define the final plot properties, and create an awesome animation!
axis(plot_window); %Define x- and y-axis
whitebg([238/255,169/255,184/255]); %Change background color of plot
%Define text in plot
text_time = text((plot_window(1)+(plot_window(2)-plot_window(1))/2),(
plot_window(4)-(plot_window(4)-plot_window(3))/14),'t = 0');
set(text_time,'FontSize',10,'Color',(1/255)*[0,0,0],'FontName','Calibri',
'HorizontalAlignment','center');
text_N_bac = text((plot_window(1)+(plot_window(2)-plot_window(1))/2),(
plot_window(4)-(plot_window(4)-plot_window(3))/8),['Total number of
bacteria is ',num2str(sum(bac_alive)),'.']);
set(text_N_bac,'FontSize',14,'Color',(1/255)*[205,16,118],'FontName','
Calibri','HorizontalAlignment','center');
text_start = text((plot_window(1)+plot_window(2))/2,2,'< Press any key to
start the simulation >');
set(text_start,'FontSize',10,'Color',(1/255)*[0,0,0],'FontName','Calibri',
'HorizontalAlignment','center');
text_copyright = text((plot_window(1)+(plot_window(2)-plot_window(1))/20),(
plot_window(3)+(plot_window(4)-plot_window(3))/15),[char(169),'BSc
project Mathias Mul']);
set(text_copyright,'FontSize',10,'Color',(1/255)*[0,0,0],'FontName','
Calibri');
Animation(1)=getframe;
pause;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Time integration %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k = 1:K-1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%% Image plot %%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    cla;
    %Plot the area of the wound
    patch(x_bac_ellipsoid , y_bac_ellipsoid ,(1/255)*[255, 112, 119]);
    hold on
    %coloring the blood vessel
    patch([-L_blood_vessel/2,-L_blood_vessel/2,L_blood_vessel/2,
        L_blood_vessel/2],[-(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2,(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2,(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2,-(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2],1/255*[238,44,44]);
    hold on;
    %plotting the bacteria
    bac_plot = plot(x_bac(bac_alive),y_bac(bac_alive),'k. ');
    %plotting the blood vessel cells
    for i=1:N_blood_vessels
        patch(x_bvc_up(i,:),y_bvc_up(i,:),1/255*[205,16,118]);
        hold on
        patch(x_bvc_down(i,:),y_bvc_down(i,:),1/255*[205,16,118]);
        hold on
    end;
    %Plot the white blood cells
    for j = wbc_visible
        patch(x_wbc(j,:),y_wbc(j,:), 'w');
        hold on;
        plot(xc(j,:),yc(j,:), 'ko', 'MarkerSize',3, 'MarkerFaceColor',(1/255)
            *[142 56 142]);
        hold on;
    end;
    %Define the final plot properties , and create an awesome animation!
    axis(plot_window); %Define x- and y-axis
    whitebg([238/255,169/255,184/255]); %Change background color of plot
    %Define text in plot
    text_N_bac = text((plot_window(1)+(plot_window(2)-plot_window(1))/2),(
        plot_window(4)-(plot_window(4)-plot_window(3))/8),[' Total number of
        bacteria is ',num2str(sum(bac_alive)),'. ']);
    set(text_N_bac, 'FontSize',14, 'Color',(1/255)*[205,16,118], 'FontName', '
        Calibri', 'HorizontalAlignment', 'center');
    text_time = text((plot_window(1)+(plot_window(2)-plot_window(1))/2),(
        plot_window(4)-(plot_window(4)-plot_window(3))/14),[' t = ',num2str(
        floor(k*dt))]);
    set(text_time, 'FontSize',10, 'Color',(1/255)*[0,0,0], 'FontName', ' Calibri
        ', 'HorizontalAlignment', 'center');
    text_copyright = text((plot_window(1)+(plot_window(2)-plot_window(1))
        /20),(plot_window(3)+(plot_window(4)-plot_window(3))/15),[char(169)
        , 'BSc project Mathias Mul']);
    set(text_copyright, 'FontSize',10, 'Color',(1/255)*[0,0,0], 'FontName', '
        Calibri');

```

```

if (sum(bac_alive)>max_N_bac)

    text_failure = text((plot_window(1)+(plot_window(2)-plot_window(1))
        /2),(plot_window(4)-(plot_window(4)-plot_window(3))/5),'The
        bacteria claimed victory. ');
    set(text_failure,'FontSize',14,'Color',(1/255)*[41,36,33],'FontName',
        'Calibri','HorizontalAlignment','center');
    Animation(k+1)=getframe;
    break;
end;

Animation(k+1)=getframe;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Define white blood cells entering the screen %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%New white blood cells enter following a poisson process
if((rand<(dt*lambda)) || (k==1)) %If there comes a new white blood cell
    its location and velocity will be defined below
    wbc = wbc+1;
    % Initial position
    xc(wbc) = (1/2)*(-L_blood_vessel);
    yc(wbc) = (thickness_blood_vessel-white_blood_cell_size)*(betainv(
        rand,2,2)-(1/2)); %Using a beta(2,2)-distribution we define the
        spawnlocation of the white bloodcell random in the y-axis
    x_wbc(wbc,:) = xc(wbc) + x_initial_wbc;
    y_wbc(wbc,:) = yc(wbc) + y_initial_wbc;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Define bacteria movement %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Annihilation by white blood cells: Annihilate bacteria that are close
to white blood cells
if ~(sum(bac_alive)==0)
    distance_bac = zeros(1,sum(bac_alive));
    annihilation_check = zeros(1,sum(bac_alive));
    for j = 1:wbc
        distance_bac(bac_alive) = min(((repmat(x_wbc(j,:),sum(bac_alive)
            ),1)-repmat(x_bac(bac_alive)',1,N_wbc)).^2+(repmat(y_wbc(j
            ,:),length(x_bac(bac_alive)),1)-repmat(y_bac(bac_alive)',1,
            N_wbc)).^2).^(1/2),[],2);
        annihilation_check = annihilation_check + (sum(bac_alive)~=0)
            .*(distance_bac(bac_alive) < active_range_wbc);
    end;
    bac_alive_indices = find(bac_alive);
    bac_alive(bac_alive_indices(annihilation_check>=1)) = 0;
end;
%Natural death: bacteria are only allowed to die if there are 2 or more
.
random_death=(sum(bac_alive)>1).*(rand(1,sum(bac_alive))<(dt*dr_bac));
indices_random_death=(find(bac_alive).*random_death);
bac_alive(indices_random_death(indices_random_death~=0))=0; %

%Final image plot in case all bacteria died.

```

```

if sum(bac_alive)==0
    cla;
    patch(x_bac_ellipsoid,y_bac_ellipsoid,(1/255)*[255, 112, 119]);
    hold on
    patch([-L_blood_vessel/2,-L_blood_vessel/2,L_blood_vessel/2,
        L_blood_vessel/2],[-(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2,(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2,(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2,-(thickness_blood_vessel+
        thickness_blood_vessel_cell)/2],1/255*[238,44,44]);
    hold on;
    bac_plot = plot(x_bac(bac_alive),y_bac(bac_alive),'k. ');
    for i=1:N_blood_vessels
        patch(x_bvc_up(i,:),y_bvc_up(i,:),1/255*[205,16,118]);
        hold on
        patch(x_bvc_down(i,:),y_bvc_down(i,:),1/255*[205,16,118]);
        hold on
    end;
    for j = wbc_visible
        patch(x_wbc(j,:),y_wbc(j,:), 'w');
        hold on;
        plot(xc(j,:),yc(j,:), 'ko', 'MarkerSize',3, 'MarkerFaceColor',
            ',(1/255)*[142      56      142]');
        hold on;
    end;
    axis(plot_window);
    whitebg([238/255,169/255,184/255]);
    text_N_bac = text((plot_window(1)+(plot_window(2)-plot_window(1))
        /2),(plot_window(4)-(plot_window(4)-plot_window(3))/8),['Total
        number of bacteria is ',num2str(sum(bac_alive)),'.']);
    set(text_N_bac,'FontSize',14,'Color',(1/255)*[205,16,118],'FontName',
        ', Calibri','HorizontalAlignment','center');
    text_time = text((plot_window(1)+(plot_window(2)-plot_window(1))/2),
        (plot_window(4)-(plot_window(4)-plot_window(3))/14),['t = ',
        num2str(floor(k*dt))]);
    set(text_time,'FontSize',10,'Color',(1/255)*[0,0,0],'FontName',',
        Calibri','HorizontalAlignment','center');
    text_copyright = text((plot_window(1)+(plot_window(2)-plot_window(1))/20),
        (plot_window(3)+(plot_window(4)-plot_window(3))/15),[
        char(169),'BSc project Mathias Mul']);
    set(text_copyright,'FontSize',10,'Color',(1/255)*[0,0,0],'FontName',
        ', Calibri');
    text_succes = text((plot_window(1)+(plot_window(2)-plot_window(1))/2),
        (plot_window(4)-(plot_window(4)-plot_window(3))/5),'All
        bacteria have been succesfully annihilated. ');
    set(text_succes,'FontSize',14,'Color',(1/255)*[41,36,33],'FontName',
        ', Calibri','HorizontalAlignment','center');
    Animation(k+1)=getframe;
break;
end;
%Reproduction: bacteria can only be reproduced if there is at least 1
    bac present.
    random_birth = ~(sum(bac_alive)==0).*(rand(1,sum(bac_alive))<(dt*br_bac
    ));
    bac_dead_indices = find(~bac_alive);

```

```

x_bac_alive = x_bac(bac_alive);
y_bac_alive = y_bac(bac_alive);

x_bac(bac_dead_indices(1:sum(random_birth))) = x_bac_alive(random_birth
==1);
y_bac(bac_dead_indices(1:sum(random_birth))) = y_bac_alive(random_birth
==1);
bac_alive(bac_dead_indices(1:sum(random_birth))) = ones(1,sum(
random_birth));
%Movement of bacteria is normally distributed with mean 0 and sigma^2=
sigma_bac
vx_bac(bac_alive) = -sign(x_bac(bac_alive)-ec(1)).*(((x_bac(bac_alive)
-ec(1))./ea).^2+((y_bac(bac_alive)-ec(2))./eb).^2)>1)*2*sigma_bac +
randn(1,sum(bac_alive)).*sigma_bac.^2;
vy_bac(bac_alive) = -sign(y_bac(bac_alive)-ec(2)).*(((x_bac(bac_alive)
-ec(1))./ea).^2+((y_bac(bac_alive)-ec(2))./eb).^2)>1)*2*sigma_bac +
randn(1,sum(bac_alive)).*sigma_bac.^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define white blood cell movement %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
wbc_visible=1:wbc;
wbc_visible=wbc_visible(xc(1:wbc)<(0.5*L_blood_vessel+
white_blood_cell_size));
for j = wbc_visible
%Define new contraction parameter alpha for all blood cells
alpha_wbc(j) = alpha_wbc_0(j)*((A_wbc_0-A_wbc_min)*(A_wbc_max-
A_wbc_0))/((polyarea(x_wbc(j,:),y_wbc(j,:))-A_wbc_min)*(
A_wbc_max-polyarea(x_wbc(j,:),y_wbc(j,:))));

%Calculate bac influence on the velocity of the
%white blood cells and add the influences using superposition.
vx_partial=zeros(sum(bac_alive),N_wbc);
vy_partial=zeros(sum(bac_alive),N_wbc);
bv_bordercheck_up = ((y_wbc(j,:) < -thickness_blood_vessel/2 & y_wbc
(j,:) > -(thickness_blood_vessel_cell+(thickness_blood_vessel)/2)
));
bv_bordercheck_down = ((y_wbc(j,:) > thickness_blood_vessel/2 &
y_wbc(j,:) < (thickness_blood_vessel_cell+(thickness_blood_vessel
)/2)));
bv_bordercheck = (bv_bordercheck_up|bv_bordercheck_down);
theta = sum(bv_bordercheck)/N_wbc;
beta_star = beta*(1-theta)+beta_mu*theta;
for s = find(bac_alive)

    vx_partial(s,:) = beta_star.*((1/(2*pi*D)).*((x_bac(s)-x_wbc(j
,:),:))./((x_bac(s)-x_wbc(j,:),:).^2+(y_bac(s)-y_wbc(j,:),:).^2)));
    vy_partial(s,:) = beta_star.*((1/(2*pi*D)).*((y_bac(s)-y_wbc(j
,:),:))./((x_bac(s)-x_wbc(j,:),:).^2+(y_bac(s)-y_wbc(j,:),:).^2)));
end;

distance_wbc = zeros(1,wbc);
vx_anti_overlap_partial = zeros(1,wbc);
vy_anti_overlap_partial = zeros(1,wbc);
W_loop = wbc_visible;

```

```

W_loop(W_loop==j) = [];
for w = W_loop
    distance_wbc(w) = sqrt((xc(j)-xc(w))^2+(yc(j)-yc(w))^2);
    vx_anti_overlap_partial(w) = (2*white_blood_cell_size -
        distance_wbc(w)>0)*real(sqrt(((2*white_blood_cell_size -
        distance_wbc(w))*white_blood_cell_size)/(
        white_blood_cell_size - (2*white_blood_cell_size - distance_wbc(
        w))))*(1/distance_wbc(w))*(xc(j)-xc(w));
    vy_anti_overlap_partial(w) = (2*white_blood_cell_size -
        distance_wbc(w)>0)*real(sqrt(((2*white_blood_cell_size -
        distance_wbc(w))*white_blood_cell_size)/(
        white_blood_cell_size - (2*white_blood_cell_size - distance_wbc(
        w))))*(1/distance_wbc(w))*(yc(j)-yc(w));
end;

vx_wbc_blood_flow = -poiseuille_velocity_coefficient(k)*(y_wbc(j,:)
    > -(thickness_blood_vessel+thickness_blood_vessel_cell)/2 &
    y_wbc(j,:) < (thickness_blood_vessel+thickness_blood_vessel_cell)
    /2).*(y_wbc(j,:).^2 - ((thickness_blood_vessel+
    thickness_blood_vessel_cell)^2)/4);
vx_wbc_contraction = alpha_wbc(j)*(xc(j)+x_initial_wbc-x_wbc(j,:));
vx_wbc_random_horizontal_movement = zeros(1,N_wbc);
vx_anti_overlap = ones(1,N_wbc)*sum(vx_anti_overlap_partial);
vx_wbc_endothelium = zeros(1,N_wbc);

vy_wbc_blood_flow = zeros(1,N_wbc);
vy_wbc_contraction = alpha_wbc(j)*(yc(j)+y_initial_wbc-y_wbc(j,:));
vy_wbc_random_vertical_movement = rvm_wbc*(rand-0.5)*(sum((y_wbc(j
    ,:)> -thickness_blood_vessel/2 & y_wbc(j,:) <
    thickness_blood_vessel/2))==N_wbc);
vy_anti_overlap = ones(1,N_wbc)*sum(vy_anti_overlap_partial);
vy_wbc_endothelium = theta * resistance_bv * (-max(
    bv_bordercheck_down>0)+max(bv_bordercheck_up>0));

vx_wbc_bac_influence = ~(sum(bac_alive)==0)*sum(vx_partial,1);
vy_wbc_bac_influence = ~(sum(bac_alive)==0)*sum(vy_partial,1);
v_wbc_bac_influence = (vx_wbc_bac_influence.^2+vy_wbc_bac_influence
    .^2).^(1/2);
vx_wbc_bac_influence = (v_wbc_bac_influence -
    cap_velocity_bac_influence <= 0).*vx_wbc_bac_influence + (
    v_wbc_bac_influence - cap_velocity_bac_influence > 0).*((
    cap_velocity_bac_influence)./v_wbc_bac_influence).*
    vx_wbc_bac_influence;
vy_wbc_bac_influence = (v_wbc_bac_influence -
    cap_velocity_bac_influence <= 0).*vy_wbc_bac_influence + (
    v_wbc_bac_influence - cap_velocity_bac_influence > 0).*((
    cap_velocity_bac_influence)./v_wbc_bac_influence).*
    vy_wbc_bac_influence;

vx_wbc(j,:) = vx_wbc_bac_influence + vx_wbc_blood_flow +
    vx_wbc_contraction + vx_wbc_random_horizontal_movement +
    vx_anti_overlap + vx_wbc_endothelium;
vy_wbc(j,:) = vy_wbc_bac_influence + vy_wbc_blood_flow +
    vy_wbc_contraction + vy_wbc_random_vertical_movement +
    vy_anti_overlap + vy_wbc_endothelium;

```

```

end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% Euler forward %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x_bac = x_bac + dt*vx_bac;
y_bac = y_bac + dt*vy_bac;
x_wbc = x_wbc + dt*vx_wbc;
y_wbc = y_wbc + dt*vy_wbc;
xc = (1/N_wbc)*sum(x_wbc,2);      %Define cell center
yc = (1/N_wbc)*sum(y_wbc,2);

DoI(k+1) = sum(bac_alive)/polyarea(x_bac_ellipsoid,y_bac_ellipsoid);
end;

Animation(k+2:K) = [];
%Save movie as .avi file.
%movie2avi(Animation, 'filename.avi','fps',<insert number of frames per
    second here, e.g. 30>)
%Compress file with free program virtualdub

```

9 References

- [1] F.J. Vermolen, A. Gefen, 2012
A Phenomenological Model for Chemico-Mechanically Induced Cell Shape Changes during Migration and Cell-Cell Contacts
- [2] Evans, 1998
Partial Differential Equations, American Mathematical Society
ISBN 0-8218-0772-2
- [3] A.E.P. Veldman, A. Velická, , 2008
Natuurwetenschappen en Stromingsleer, Rijksuniversiteit Groningen
- [4] C.Vuik, P. van Beek, F. Vermolen, J. van Kan, , 2006
Numerieke Methoden voor Differentiaalvergelijkingen, Published by VSSD
- [5] Christopher V. Carman
Mechanisms for transcellular diapedesis: probing and pathfinding by ‘invadosome-like protrusions’
- [6] Adam JA, 1999
A simplified model of wound healing (with particular reference to the critical size defect), Math Comput Model 30:23-32
- [7] Schugart RC, Friedman A, Zhao R, Sen CK, 2008
Wound angiogenesis as a function of tissue oxygen tension: a mathematical model, Proc Nation Ac Sc USA 105(7):2628-2633
- [8] Vermolen FJ, Javierre E, 2011
A finite-element model for healing of cutaneous wounds combining contraction angiogenesis and closure, J Math Biol: in press