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

A cell based model for Atherosclerosis

Report on behalf of the
Delft Institute of Applied Mathematics
as part of obtaining

the title of

BACHELOR OF SCIENCE
in
APPLIED MATHEMATICS

by

Kevin Visser

Delft, Netherlands
November 2015

Copyright © 2015 by Kevin Visser. All rights reserved.



BSc verslag TECHNISCHE WISKUNDE

Een cel-gebaseerd model voor Atherosclerose
A cell based model for Atherosclerosis

Kevin Visser

Technische Universiteit Delft

Begeleiders

Dr. J.L.A. Dubbeldam

Dr. ir. F.J. Vermolen

Overige commissieleden

Drs. E.M. van Elderen

November, 2015

Delft

Samenvatting

In dit rapport is een semi-stochastisch cel-gebaseerd model opgesteld voor de beweging van cellen die in de plak voorkomen bij Atherosclerose. Dit is gekoppeld aan een systeem dat cellen van type omzet en daarnaast bepaalde cellen de plak in laat gaan en juist er uit laat gaan. Uit het bloed komen monocyten en geoxideerd cholesterol binnen waardoor de plak in de loop der tijd zal groeien.

Er zijn vergelijkingen opgesteld aangaande de beweging van de cellen. Aantrekkingskracht, Afstotingskracht, Random Walk en Chemotaxis zijn besproken. In een grafische weergave is te zien hoe de cellen zich door de plak bewegen.

Een korte analyse met meerdere beginsituaties is gedaan. Er is gekeken naar wat het verschil in beginsituaties in de loop der tijd met zich meebrengt.

Voorwoord

Voor u ligt het eindverslag van een bacheloreindproject dat uitgevoerd is ter afsluiting van de bachelor Technische Wiskunde aan de TU Delft. Hierin is onderzoek gedaan naar een op cellen gebaseerd model van Atherosclerose.

Ik wil graag mijn begeleiders, Johan Dubbeldam en Fred Vermolen, bedanken voor hun advies en begeleiding tijdens het project. Na jaren van interesse in zowel wiskunde als biologie, heb ik deze met name door hen prachtig weten te combineren in dit project. Een project waar veel tijd in het doorgronden van de biologische achtergrond zat, als het probleem oplossend denken vanuit de studie. Beide heren wil ik bedanken voor het mij ~~me~~ gang laten gaan en mij zelfstandig dit project laten doen, maar waar nodig was, te sturen.

Contents

1 Inleiding **1**
- 1.1 Belang van het onderzoek . . . 1
- 1.2 Doel van het onderzoek . . . 1
- 1.3 Indeling van de scripte . . . 1

2 Biologische achtergrond **2**
- 2.1 Atherosclerose . . . 2
- 2.2 Opbouw aderwand . . . 2
- 2.3 Samenstelling van het bloed . . . 3
- 2.4 Vorming van de plak . . . 3

3 Wiskundige model **5**
- 3.1 Algemene eigenschappen . . . 5
- 3.2 Celmigratie . . . 5
  - 3.2.1 Actieve migratie . . . 5
  - 3.2.2 Passieve migratie . . . 7
  - 3.2.3 Celverplaatsing . . . 7
  - 3.2.4 Vector-Wienerproces . . . 8
  - 3.2.5 Chemotaxis . . . 9
  - 3.2.6 Totale Verplaatsing . . . 9
- 3.3 Celdeling en celsterfte . . . 9
  - 3.3.1 Celsterfte . . . 9
  - 3.3.2 Celdeling . . . 9
- 3.4 Differentie en toename van de cellen . . . 10
  - 3.4.1 Opbouw vergelijkingen . . . 10
  - 3.4.2 Numerieke implementatie . . . 10
  - 3.4.3 Volumegroei . . . 11

4 Resultaten van de simulaties **12**
- 4.1 Verscheidene situaties . . . 12
- 4.2 Simulatie 1 . . . 12
- 4.3 Simulatie 2 . . . 12
- 4.4 Overige bevindingen bij overige simulaties . . . 13

5 Conclusie en discussie **14**
- 5.1 Conclusie . . . 14
- 5.2 Discussie en aanbevelingen . . . 14

A Simulatie 1 **16**

B Simulatie 2 **19**

C Parameters **22**

D Main Code **24**

E Omzet cellen **32**

F Omzet Cellen 2 **33**

G Celtellen **36**

H Tellen van monocyten **37**

I Tellen van macrofagen **38**

J Tellen van LDL **39**

K Tellen van LDLox **40**

L Tellen van Foam **41**

M Monocyten naar Macrofagen **42**

N Macrofagen naar schuimcellen **43**

O LDL naar LDLox **44**

P Nieuwe cellen plaatsen **45**

Q Verwijderen van monocyten uit de plak **46**

R Verwijderen van LDLox **48**

S Volumegroei **50**

Chapter 1

Inleiding

In dit hoofdstuk zal er motivatie gegeven worden achter de beweegredenen voor dit onderzoek. Daarnaast zal het doel beschreven worden. Tot slot zal er kort de indeling van het verslag beschreven worden.

1.1 Belang van het onderzoek

Per jaar sterven in Nederland 40.000 mensen aan hart- dan wel vaatzieken. Een van de oorzaken hiervan is Atherosclerose, in de volksmond beter bekend als aderverkalking. Het is dan ook van belang dat hier onderzoek naar gedaan wordt. Een stap in de goede richting zou dan ook het hebben van wiskundige modellen zijn waarmee verscheidene situaties gesimuleerd kunnen worden. Met een goed model zou er bijvoorbeeld naar meerdere beginsituaties gekeken worden, maar ook de invloed van medicijnen.

1.2 Doel van het onderzoek

Het doel van dit onderzoek is om een semi-stochastisch cel-gebaseerd model te maken. In dit model wordt elke cel beschouwd als zijn eigen ~~identiteit,~~ met eigen celtype, radius en meer. De cellen waar naar gekeken gaan worden zijn diegene in de plak en cellen die deze plak in en uit gaan. Einddoel van het project is om te simuleren hoe de cellen bewegen, migreren, delen en sterven en te kijken of hier patronen in te vinden zijn.

1.3 Indeling van de scripte

Het eerste onderdeel zal bestaan uit de biologische achtergrond achter Atherosclerose. Dit zodat de lezer de latere processen van het model kan begrijpen. De processen die besproken worden zullen onder andere het ontstaan van plak zijn, de structuur van een aderwand en de migratie van cellen.

Het tweede onderdeel zal het model aangaande celmigratie zoals vermeld in het artikel van F.J. Vermolen[7] besproken worden. Vervolgens komt het verplaatsen van cellen in plak en de omzetting van cellen zoals vermeld in het artikel van J.L.A. Dubbeldam[2].

In het derde onderdeel zullen de resultaten van de simulaties van het uiteindelijke Matlab-model besproken worden.

Tenslotte zal er over deze resultaten geconcludeerd en gediscussieerd worden. Niet enkel over de resultaten, maar ook over wat er verbeterd kan worden aan het model.

Chapter 2

Biologische achtergrond

2.1 Atherosclerose

Atherosclerose, in de volksmond beter bekend als aderverkalking, is een ziekte waarbij de wanden van een slagader verharden door kalkaanslag. In de wanden ontstaat een laag plak die de bloedbaan verdunnen. Hoe dit zich in de loop der tijd ~~ontwikkeld~~ zal hierna behandeld worden.

2.2 Opbouw aderwand

De wand van een ader bestaat uit drie lagen, namelijk de *intima*, de *media* en de *adventitia*, zie ook het onderstaande figuur. Van deze is de intima het dunst en bestaat uit een dunne laag endotheelcellen die ondersteund worden door een elastische weefsellaag. Hierdoor wordt de rest van de aderwand ~~beschermt~~ tegen directe invloeden van het bloed. Daarnaast zorgt deze laag voor de transport van stoffen vanuit het bloed naar de weefsels. De media is de dikste laag van de aderwand en bestaat uit elastische vezels en glad spierweefsel. De adventitia is de derde en buitenste laag van de aderwand. Naast elastische vezels bestaat deze laag voornamelijk uit collageen (bindweefsel). Dit zorgt ervoor dat de ader aan omgevend weefsel vast blijft zitten door onder andere druk uitgeoefend door stromend bloed in de bloedbaan.

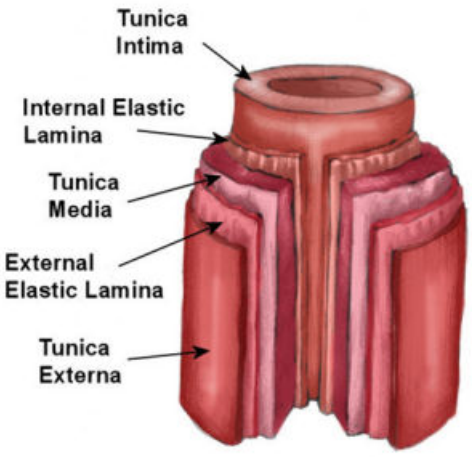

Figure 2.1: Schematische weergave van een aderwand

2.3 Samenstelling van het bloed

In het bloed zitten een aantal van de stoffen die atherosclerose kunnen veroorzaken. De voornaamste hiervan is cholesterol. Cholesterol is een vetachtige stof die behoort tot de lipiden. Lipiden zijn niet oplosbaar en zodoende cholesterol ook niet. Daar lipiden niet oplossen, moeten deze door het bloed getransporteerd worden door lipoproteïnen. Lipoproteïnen worden doorgaans in dichtheid ingedeeld. Van laag naar hoog zijn er: Very Low Density Lipoprotein (VLDL), Intermediate Density Lipoprotein (IDL), Low Density Lipoprotein (LDL), en High Density Lipoprotein (HDL). Twee hiervan zijn de voornaamste in het bloed:

- Low Density Lipoprotein (LDL), welke in de volksmond bekend staat als 'slecht' cholesterol daar deze vetmoleculen kan vervoeren de aderwanden in, welke in de loop der tijd kunnen ophopen. Dit kan zorgen voor dichtslibbende bloedvaten, gescheurde bloedvaten en soms infarcten.
- High Density Lipoprotein (HDL), welke in de volksmond bekend staat als 'goed' cholesterol daar deze vetmoleculen weg van de aderwanden kunnen vervoeren richting de lever.

De Low Density Protein ~~is diegene van belang voor dit project,~~ en zal hierna volgend enkel LDL genoemd worden.

Een andere factor bij de vorming van atherosclerose zijn de monocyten. Dit zijn een type witte bloedcellen. De monocyten in de bloedbaan worden met name opgenomen door de milt, een aantal gaan echter op den duur in weefsel zitten, waar ze gaan differentiëren.

2.4 Vorming van de plak

1. LDL circuleert door het bloed en passeert door het endotheel van de aderwand en hoopt zich op in de intima, vlak onder het endotheel.
2. Hier oxideert een deel van het LDL nadat het zich heeft gebonden aan proteoglycanen.
3. Geoxideerd LDL zorgt voor het loslaten van stoffen die op hun beurt monocyten en nog meer LDL aantrekken.
4. Daarnaast zorgt geoxideerd LDL ervoor dat monocyten in de intima omgezet worden naar macrofagen. Dit zijn grote cellen die resten van dode of beschadigde cellen veranderen naar lichaamsvreemde cellen. Dit doen ze door onder andere LDL op te nemen. Daarnaast worden er ook meer monocyten aangetrokken.
5. De macrofagen worden op den duur histocyten (schuimcellen dan wel 'foam'-cellen).
6. Deze schuimcellen zullen op den duur afsterven. Bij een te grote hoeveelheid afgestorven cellen komt er plakvorming. Deze cellen vormen de kern van de plak
7. Schuimcellen produceren cytokines, welke ervoor zorgen dat het gladde spierweefsel uit de media zich vermenigvuldigt en dat de aanmaak van de extracellulaire matrix bevorderd wordt.
8. Hierdoor wordt een kap over de plak gevormd, waardoor de vettige substanties niet uit de plak kan.
9. Calcium wordt geproduceerd en de plak verhardt. Hierdoor de naam 'aderverkalking'.

Na vele jaren zal de plak steeds groter ~~en groter~~ worden daar vettige substanties blijven ophopen en ~~verharderen~~ door het hierboven beschreven proces. Door continue vergroting van de plak zal de ader vernauwd raken en de aderwand verstijft. In figuur (2.2) is een schematische weergave van het verloop van het proces te zien. Hier is ook te zien wat het effect van de ziekte op de aderwand heeft.

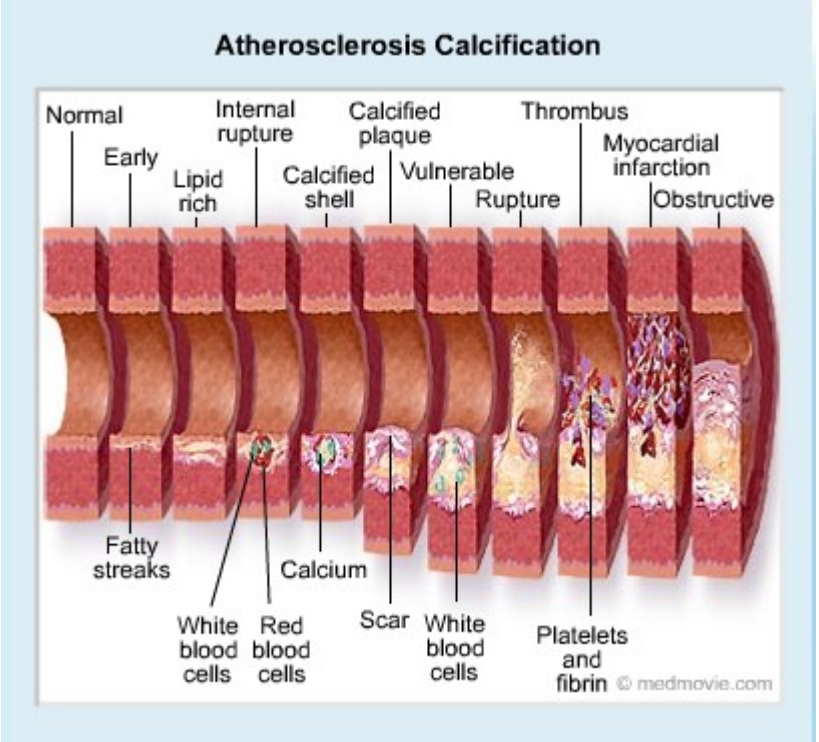

Figure 2.2: Schematische weergave van Atherosclerose.

Chapter 3

Wiskundige model

In dit hoofdstuk zal de achtergrond achter het wiskundig model besproken worden. Merk op dat dit wiskundig model een tweedimensionaal model is. De krachten die op de cellen van toepassing zijn, zullen besproken besproken. Daarnaast zullen er vergelijkingen aangaande omzetting van de cellen, zoals besproken in het vorige hoofdstuk, gegeven worden en zo ook voor de volumegroei van de plak.

3.1 Algemene eigenschappen

We bekijken een tweedimensionaal substraat waarin de cellen leven. Biologisch gezien is dit de plak waarin de cellen bewegen. We beschouwen het substraat als homogeen en isotoop. Dit houdt in dat elk punt van het substraat dezelfde eigenschappen heeft en deze in elke richting gelijk zijn. We nemen aan dat de substraat aan vier kanten geraakt wordt: aan de zuidkant door de ader en aan de andere kanten door de bloedbaan. Voor de cellen wordt aangenomen dat deze hele bollen zijn met een gegeven straal wat bij ieder anders is. In werkelijkheid zijn cellen niet noodzakelijk bolvormig, maar varieert hun vorm onder invloeden van beweging en botsing. We beschouwen driedimensionale cellen in een tweedimensionaal substraat. De celpopulatie zal in de loop van de tijd veranderen door de toevoer en verwijdering van cellen na processen beschreven in het vorige hoofdstuk, dan wel door celdeling en celsterfte.

3.2 Celmigratie

De beweging van een cel is in twee categorieën op te delen: actieve en passieve migratie. De interpretatie van deze processen worden besproken en bewegingsvergelijkingen opgesteld. Hierbij wordt gebruik gemaakt van het onderzoek van Gefen en Vermolen et al, uit respectievelijk [1], [7] en [8].

3.2.1 Actieve migratie

Elke levende cel beweegt door de plak heen en oefent een aantrekkingskracht uit welke een mechanisch signaal uitoefent over het substraat. Deze kracht levert een stress (kracht per oppervlakte-eenheid) op welke resulteert in een kleine deformatie van het substraat. Dit levert de mechanische energie op, welke de strain energy density (energie per volume-eenheid) is. Deze strain energy density reprensenteert een scalar, niet een vector en kan dus gemakkelijk geëvalueerd worden als een verstuurd en gedetecteerd signaal door het substraat heen. Laat M_i^0 de strain energy denisity zijn als gevolg van de trekkracht F_i op de positie van cel i. De vergelijking voor strain

energy density is als volgt gegeven:

$$M_i^0 = \frac{1}{2}\sigma\epsilon \tag{3.1}$$

Vanwege de eerder aangenomen homogeniteit kunnen we stellen dat de elasticiteit van het substraat $E_s(\mathbf{r}_i)$ in elk punt $\mathbf{r}_i$ gelijk is, en volgt er $E_s(\mathbf{r}_i) = E_s$. Wet van Hooke levert vervolgens:

$$\sigma = E_s\epsilon \tag{3.2}$$

Hierbij is σ de normaalspanning en ϵ de rek van het substraat in het midden van de cel. Laat L en d respectievelijk de dikte en lokale ~~certikale~~ verplaatsing van het substraat veroorzaakt door opwaartse trekkracht van de cel op het substraat. De rek ϵ is te berekenen als volgt

$$\epsilon = \frac{d}{L} \tag{3.3}$$

Als A_i de oppervlakte van de projectie van de cel op het substraat is en R_i de straal van de cel is, dan geldt er voor de spanning:

$$\sigma = \frac{F_i}{A_i} = \frac{F_i}{\pi R_i^2} \tag{3.4}$$

Combinatie van (3.1), (3.2) en (3.4) levert het volgende:

$$M_i^0 = \frac{1}{2\pi^2}\frac{F_i^2}{E_s R_i^4} \tag{3.5}$$

In dit project wordt aangenomen dat de trekkracht van de cel F_i gelijk is aan de maximale effectieve opwaartse trekkracht uitgeoefend door een levende cel, welke $\hat{F}$ is indien cel i levend is. Er wordt aangenomen dat alle cellen van hetzelfde fenotype zijn en dat dus $\hat{F}$ voor alle cellen gelijk zijn. Daar dode cellen geen trekkracht uitoefenen, wordt aangenomen dat indien cel i gestorven is, de trekkracht 0 is. In andere woorden, de volgende relatie geldt:

$$F_i = \begin{cases} \hat{F} & \text{als cel } i \text{ levend is} \\ 0 & \text{indien cel } i \text{ gestorven is} \end{cases}$$

Door het toedoen van cel i is er door vervorming energie opgeslagen in het substraat. Deze vervorming is het grootst in het midden van de cel, daar wordt dan ook het hoogste mechanische celsignaal gemeten. Hoe verder van de cel af, hoe lager het gemeten signaal is. De vervorming is hier dan ook minder. Uit eerder onderzoek is voor de sterkte van het signaal een exponentieel verband gevonden na signaalmetingen [7]. Een benadering voor de potential energy density is dan ook als volgt:

$$M_i(\mathbf{r}) = M_i^0 \exp\left\{-\lambda_i \frac{\|\mathbf{r} - \mathbf{r}_i\|}{R_i}\right\} \tag{3.6}$$

Hierin is $\mathbf{r}_i$ de locatie van cel i en $\lambda_i = \frac{E_s}{E_c}$ de ratio tussen de elasticitiet van het substraat E_s en de elasticiteit van de cel E_c, waarbij de elasticiteit van elke cel i als gelijk wordt beschouwen. Daar energie een scalaire kwantiteit is, is deze additief en kan om de totale strain energy density te vinden, de strain energy density van de verzameling cellen van grootte n als volgt geschreven worden voor positie $\mathbf{r}$:

$$M(\mathbf{r}) = \sum_{j=1}^{n} M_i(\mathbf{r}) = \sum_{j=1}^{n} M_i^0 \exp\left\{-\lambda_i \frac{\|\mathbf{r} - \mathbf{r}_i\|}{R_i}\right\} \tag{3.7}$$

3.2.2 Passieve migratie

Wanneer twee cellen dicht in de buurt van elkaar zijn, maar elkaar niet raken, bestaat er enkel een aantrekkingskracht en geen afstotende kracht. Echter wanneer twee cellen elkaar fysiek raken vindt er wel een afstotende kracht plaats. Deze afstotende kracht is groter dan de aantrekkende kracht en de cellen zullen elkaar afstoten. Zij $h = \frac{1}{2}\max(0, R_i + R_j - \|\mathbf{r}_i - \mathbf{r}_j\|)$. Hierin zijn R_i en R_j de radii van respectievelijk cellen i en j. Zo zijn $\|\mathbf{r}_i - \mathbf{r}_j\|$ de positie van de middelpunten van cellen i en j. Combinatie van Hertz' model voor contactkrachten, zie [1] en [7], en integratie over de strain, vinden we de strain energy density behorende bij de afstoting van cel i op cel j:

$$M^{ij} = \frac{16}{25}\frac{E_c}{\sqrt{2}\pi}\left(\frac{h}{R_i}\right)^{5/2} \tag{3.8}$$

Het uiteindelijke resultaat van de totale strain energy density function is als volgt:

$$\hat{M}_i(\mathbf{r}) = M_i(\mathbf{r}) - M^{ij} \tag{3.9}$$

3.2.3 Celverplaatsing

Daar de totale strain energy density function gevonden is, is het tijd om te kijken naar de verplaatsing van elke cel i zodra het mechanische signaal gedetecteerd is. De aanname is dat de verplaatsingsvector over een tijdsstap Δt geldt en een lineaire combinatie is van alle eenheidsvectoren die cel i verbindt met alle overige cellen j. De weegfactor voor ieder paar (i,j), $i \neq j$ wordt bepaald door de sterkte van het signaal dat ontvangen wordt. We vinden dus de volgende verplaatsingsvector:

$$\mathbf{z}_i = \sum_{j=1, j\neq i}^{n} \hat{M}_j(\mathbf{r}_i(t))\mathbf{v}_{ij}(t) \quad \forall i \ \in \{1, ...n\} \tag{3.10}$$

waarin $\mathbf{v}_{ij}$ de eenheidsvector weergeeft die cel i met cel j koppelt, welke gegeven is als:

$$\mathbf{v}_{ij} = \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|} \tag{3.11}$$

Merk op dat deze als 0 gegeven wordt als $\|\mathbf{r}_j - \mathbf{r}_i\| = 0$. Daar we zuiver over richting praten normaliseren we de vector $\mathbf{z}_i$:

$$\hat{\mathbf{z}}_i = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|} \tag{3.12}$$

Daar de verplaatsing gedurende een tijdsstap aangenomen is in de richting van $\mathbf{z}_i$ en de grootte van de verplaatsing is aangenomen om proportioneel te zijn aan de sterkte van het mechanische signaal, hebben we:

$$\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t) = \Delta t \alpha_i \hat{M}(\mathbf{r}_i)\hat{\mathbf{z}}_i \ \forall i \in \{1, ..., n\} \tag{3.13}$$

waarin α_i een parameter is met dimensie $\left[\frac{m^2 s}{kg}\right] = \left[\frac{m^3}{Ns}\right]$ en gegeven is als:

$$\alpha_i = \left(\frac{F_i}{\hat{F}}\right)\beta_i \frac{R_i^3}{f} \tag{3.14}$$

Hierin is $\hat{F}$ zoals eerder gezegd de effectieve opwaartse trekkracht uitgeoefend door een levende cel. Daar aangenomen is dat de cellen dezelfde fenotype hebben, is deze $\hat{F}$ voor elke cel hetzelfde. Ook geldt $f = \mu F_i$. Hierin is f de cel-substraat frictie en μ de cel-frictiecoëfficiënt, welke gesteld

is op $\mu = 0.2$ volgens [1]. Daarnaast is β_i met eenheid $\frac{m}{s}$ kwantificeert de beweegbaarheid van het celoppervlak van een levende cel. Er volgt dus voor de parameter α_i:

$$\alpha_i = \left(\frac{F_i}{\hat{F}}\right)\beta_i\frac{R_i^3}{f} = \frac{\beta_i R_i^3}{\mu\hat{F}^2}F_i \tag{3.15}$$

Vergelijkingen (3.13) en (3.15) leveren dus nu samen:

$$\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t) = \Delta t\frac{\beta_i R_i^3}{\mu\hat{F}^2}F_i\hat{M}(\mathbf{r}_i)\hat{\mathbf{z}}_i \;\forall i \in \{1,....,n\} \tag{3.16}$$

Detectiedrempel

Tot nu toe is er vanuit gegaan dat indien twee levende cellen aanwezig zijn, deze ook altijd elkaars signalen ontvangen wanneer deze verzonden zijn. Daar signaalsterkte nauwelijks de snelheid van de cel beïnvloedt, de snelheid wordt met name beïnvloedt door beweegbaarheid, diameter, frictiecoëfficiënt en levensvatbaarheid, ~~lijkt~~ er een detectiedrempel te ~~horen zijn~~ voor de strain energy density. Nemen we een detectiedrempel van een ϵ dan vinden we met vergelijking (3.6) dat:

$$M_i(\mathbf{r}) = M_i^0\exp\left\{-\lambda_i\frac{\|\mathbf{r}-\mathbf{r}_i\|}{R_i}\right\} \geq \epsilon \tag{3.17}$$

Uit onderzoek van [6] blijkt dat bij een elasticiteit van het substraat en cel van respectievelijk ongeveer 5 kPa en 0.5 kPa de maximale afstand waarover cellen elkaar detecteren ongeveer $\hat{d} = 30\mu m$. Gevonden is dus een detectiedrempel van:

$$\epsilon = M_i^0\exp\left\{\lambda_i\frac{\hat{d}}{R_i}\right\} \tag{3.18}$$

3.2.4 Vector-Wienerproces

Naast mechanotaxis en chemotaxis is er van cellen ook bekend dat deze ~~random walk tentoonstellen~~. Deze wordt gemodelleerd door middel van een vector-Wiener proces, $\sigma d\mathbf{W}(t)$. Hierin is σ de variantie van de celverplaatsing, welke gelijk is aan $\sigma = \sqrt{2D}$. Het vector-Wienerproces $\mathbf{W}(t)$ is een vector ~~die voldoet~~ aan de volgende vier eisen van een Wienerproces voor $W = W(t)$:

1. $W(0) = 0$
2. $W(t)$ is een stoschastisch continu proces. Met andere woorden:
$$\lim_{t\to\infty} P(|\,W(t) - W(s)| > \epsilon) = 0 \;\forall\epsilon > 0$$
3. Voor alle $0 \leq s \leq t$ heeft de toename $W_{t+s} - W_s$ een normale verdeling met gemiddelde 0 en variantie $t - s$, dus $W(t) - W(s) \sim N(0, t-s)$
4. De toenames $W(t) - W(s)$en $W(s) - W(k)$ zijn onafhankelijk voor alle $0 < k < s < t$.

Er is ook een factor aangaande de grootte van het substraat meegenomen in het model. Dit omdat de anders de waardes van het Vector-Wienerproces te groot worden en de cellen zomaar uit het substraat zouden verdwijnen.

3.2.5 Chemotaxis

Chemotaxis is de verplaatsing van een organisme, in ons geval een cel, als antwoord op een chemische stimulans. Deze stimulans is in het model een extra signaal. Dit omdat er niet gewerkt wordt met concentraties of iets dergelijks, maar met een cel-gebaseerd model. Indien cel i en cel j van hetzelfde type zijn zal er een sterker signaal tussen deze cellen plaatsvinden. De chemotaxisterm noemen we $c(\mathbf{x}($

3.2.6 Totale Verplaatsing

Nu zowel de chemotaxis, de mechanotaxis en Vector-Wienerproces aan bod zijn gekomen, is het tijd om de gehele verplaatsing weer te geven:

$$d\mathbf{x}_i = \Delta t \alpha_i M(\mathbf{x}_i(t))\hat{\mathbf{z}}_i dt + \sigma d\mathbf{W}(t) + c(\mathbf{x}(\tag{3.19}$$

3.3 Celdeling en celsterfte

3.3.1 Celsterfte

De levensvatbaarheid is gekwantificeerd met F_i welke om aan te geven dat cel gestorven is op $F_i = 0$ gezet wordt. In elke tijdsstap wordt de kans dat een cel sterft gezet op waarde p. In het model wordt deze gezet op de hypothetische waarde van $p = 0.01 * dt$. In werkelijke situaties kan deze kans afhangen van onder meer temperatuur van het substraat, pH-waardes en toxiciteit.

Laat x_i een random variabele die rekening houdt met eventuele sterfte van cel i, zet $x_i = rand(1)$ en indien $x_i > 0.99$ dan $F_i = 0$. Er zal een aantal aanames aangaande celsterfte waar rekening mee rekening gehouden moet worden:

- Celsterfte is een acute gebeurtenis waarna er geen beweging meer plaatsvindt.
- Gestorven cellen oefenen geen trekkracht uit op het substraat en bewegen niet.
- Gestorven cellen blijven op de positie waar ze gestorven zijn.
- In werkelijkheid laten gestorven cellen stoffen los; in dit project negeren we alle relaties die de stoffen hebben op het proces. Een gestorven cel zal dus geen invloed hebben op de verplaatsing van andere cellen.

3.3.2 Celdeling

Celdeling wordt door middel van een stochastisch porces het model in gebracht. De kans dat er een dochtercel ontstaat wordt gezet op waarde q. Deze waarde wordt op $q = 0.05$. Dit volgt uit [7]. Ook hier gelden een aantal aannames:

- De positie van dochtercel j is zo, dat het middelpunt overeenkomt met de originele positie van moedercel i.
- Indien de moedercel niet in beweging was wordt een willekeurige richting aangegeven waarheen cel i zich na de deling naar toe verplaatst.
- Ook celdeling wordt beschouwd als een acute gebeurtenis.

3.4 Differentie en toename van de cellen

Celdeling en celsterfte zijn niet de enige manier waarop nieuwe cellen het model in en uitgaan. Door de aanwezigheid van geoxideerd cholesterol zullen er in de loop van tijd monocyten en meer geoxideerd cholesterol de plak binnentreden. Daarnaast differiënteren de cellen in de plak. Er wordt gemodelleerd voor vijjf verschillende cellen: namelijk monocyten (m), macrofagen (M), LDL-cholesterol (L), gexodieerd cholesterol (L_{ox}) en schuimcellen (F).

3.4.1 Opbouw vergelijkingen

Het model dat M.A.K. Bulezai en J.L.A. Dubbeldam[2] ontwikkeld hebben is een nul-dimensionaal model dat bestaat uit vier gewone, met elkaar gekoppelde differentiaalvergelijkingen. Een vijfde relatie in lijn met het model van Bululzai-Dubbeldam aangaande het niet-geoxideerde cholesterol is hierbij toegevoegd. De differentiaalvergelijkingen volgens het model Bulelzai-Dubbeldam aangaande de differentie en toename zijn als volgt:

$$\frac{dm}{dt} = (\Gamma(\sigma_w, L_{ox}) - d_m)m - \rho_1 m \tag{3.20}$$

$$\frac{dM}{dt} = \rho_1 m - \frac{\rho_{in} L_{ox}}{1 + L_{ox}/L_{th}} M \tag{3.21}$$

$$\frac{dL}{dt} = -\frac{qm}{1 + m/m_{th}} \tag{3.22}$$

$$\frac{dL_{ox}}{dt} = \frac{qm}{1 + m/m_{th}} - \rho_L L_{ox} M - d_L L_{ox} \tag{3.23}$$

$$\frac{dF}{dt} = \frac{\rho_{in} L_{ox}}{1 + L_{ox}/L_{th}} M \tag{3.24}$$

$$\Gamma(\sigma_w, L_{ox}) = \frac{\gamma_0 L_{ox}}{(1 + \sigma_w/\sigma_0)(1 + L_{ox}/L_{th})} \tag{3.25}$$

De ~~fysieke~~ interpretatie van de vergelijkingen is als volgt: het oxideren van LDL-cholesterol is een ontstekingsreactie. Monocyten komen hierop af om dit tegen te werken. De term $\Gamma(\sigma_w, L_{ox})m$ geeft de reactie weer van het endotheel wanneer geoxideerd cholesterol de intima binnendringt. Deze term is lineair afhankelijk van m en heeft een sigmoïde afhankelijkheid van L_{ox}. De toename van monocyten is dus afhankelijk van de hoeveelheid geoxideerd cholesterolcellen in de plak. Ook is in deze term de wall shear stress meegenomen. De term $\rho_1 m$ geeft de mate aan waarin monocyten omgezet worden in macrofagen. Deze komt dan ook terug in vergelijking (3.21).

Vergelijking (3.21) heeft naast de term aangaande monocyten, ook de term die de mate weergeeft hoeveel de monocyten geoxideerd cholesterol opnemen, welke later schuimcellen worden. Deze term is dus ook in vergelijking (3.24) terug te vinden.

Vergelijking (3.23) geeft de verandering van geoxideerd cholesterol weer. De nog niet benoemde termen representeren respectievelijk de oxidatie van cholesterol, de opname van geoxideerd cholesterol door macrofagen en de diffusie van geoxideerd cholesterol weg uit de plak weer.

3.4.2 Numerieke implementatie

Voor numerieke implementatie kiezen we voor de volgende transformaties:

$$\tilde{L} = L_{ox}/L_{th}, \tilde{m} = m/L_{th}, \tilde{M} = M/L_{th}, \tilde{F} = F/L_{th},$$

$$\tilde{\sigma} = \sigma_w/\sigma_0, \tilde{t} = d_L t, \epsilon = d_m/d_L, a = \gamma_0 L_{th}/d_L$$
$$c = \rho_1/d_L, b = \rho_{in} L_{th}/d_L, d = \frac{q m_{th}}{d_L L_{th}}, e = \rho_L L_{th}/d_L, f = m_{th}/L_{th}$$

Deze transformaties leveren de volgende vergelijkingen op:

$$\dot{m} = \left(\frac{aL}{(1+\sigma)(1+L)} - \epsilon - c \right) \tag{3.26}$$
$$\dot{M} = cm - \frac{bML}{1+L} \tag{3.27}$$
$$\dot{L} = -\frac{dm}{f+m} \tag{3.28}$$
$$\dot{L_{ox}} = \frac{dm}{f+m} - eLM - L \tag{3.29}$$
$$\dot{F} = \frac{bLM}{1+L} \tag{3.30}$$

Hierin representeert de stip de afgeleide naar $\tilde{t}$. De tildes zijn voor gemak weggelaten. De daadwerkelijke simulaties worden in het volgende hoofdstuk besproken.

3.4.3 Volumegroei

Nu de betreffende differentiaalvergelijkingen geïntroduceerd zijn is het mogelijk de volumegroei te bepalen. Het volume van de plak op tijdstip $t+\Delta t : V(t+\Delta t)$ kan in mate van Δt weergegeven worden als:

$$V(t+\Delta t) = V(t) + \left(v_M \frac{dM}{dt} + v_{Lox} \frac{dL_{ox}}{dt} + v_m \frac{dm}{dt} + v_F \frac{dF}{dt} \right) V(t) \Delta t \tag{3.31}$$

waarin de verandering van cellen van macrofagen, geoxideerd cholesterol, monocyten en schuimcellen meegenomen wordt. Vanuit literatuur worden waardes voor molucelaire volumes gevonden: $v_F = 10^{-13} m^3$[4], $v_{Lox} = 10^{-21} m^3$[5], $v_m = 10^{-17} m^3$[4] en $v_M = 10^{-14} m^3$[3].

Chapter 4

Resultaten van de simulaties

In deze sectie worden de resultaten van het model besproken. Hier wordt een visuele representatie van de cellen gegeven. Er is te zien hoe deze in de plak bewegen. De gebruikte Matlab-code is te vinden in de bijlages. Ook zijn de gekozen parameters hier te vinden. Er is gekozen om voor vijf jaar het verloop te bekijken met tijdsstappen van een maand. Enkel in leven zijnde cellen zijn in beeld gebracht. Gestorven cellen zijn dus niet in het beeld aanwezig. De plotten zijn te vinden in de bijlages.

4.1 Verscheidene situaties

Er zal gesimuleerd worden met verscheidene situaties. Deze zijn als volgt:

- Beginnend met 100 cellen, random verdeeld tussen de types.

- Beginnend met 100 cellen, verdeeld tussen macrofagen en geoxideerd cholesterol.

4.2 Simulatie 1

De eerste simulatie bestaat uit 100 cellen al in de plak aanwezig, waarvan het type cel random verdeeld is. Te zien is dat er zoals verwacht in de loop der tijd schuimcellen ontstaan in de plak. Ook is te zien dat er veel meer cellen aanwezig zijn in de plak, met name de monocyten nemen een prominente rol aan. Te zien is ook dat het niet-geoxideerde cholesterol nauwelijks omgezet wordt en gedeeld heeft. Ook is op te merken dat er geen clusters van cellen ontstaan van hetzelfde type. Wel is te zien dat met uitzondering van de monocyten, cellen van hetzelfde type op ongeveer dezelfde x-positie zitten en voornamelijk verschillen in y-positie.

4.3 Simulatie 2

De tweede simulatie bestaat uit 100 cellen, verdeeld tussen macrofagen en geoxideerd cholesterol. Dit bleken 10 macrofagen en 90 geoxideerd cholesterolcellen t zijn. Te zien is dat zoals verwacht er na vijf jaar een overtal van schuimcellen aanwezig zijn. Daarnaast is ook hier te zien dat de cellen rond bepaalde x-posities stagneren. Wat niet geheel in de plot, maar wel in de matlab-output is dat er meerdere clusters ontstaan zijn van schuimcellen op ~~meerdere~~ plekken. Dit is gedeeltelijk in de plots te zien door de overlap van cellen.

4.4 Overige bevindingen bij overige simulaties

- Een hogere waarde random walk die niet afgesteld is op de grootte van het substraat levert stilstaande cellen op daar deze anders de plak uit gaan wat niet mag volgens het voorschrift.
- Indien de mate van omzetting van monocyten naar macrofagen wordt verdubbeld, zal de hoeveelheid monocyten sterk afnemen en sneller macrofagen worden dan bijvoorbeeld in de hierboven genoemde simulaties. Daar er minder monocyten zijn, loopt de oxidatie van cholesterol langzamer. Vervolgens zorgt een lagere hoeveelheid geoxideerd cholesterol voor langzamere omzetting van macrofagen naar schuimcellen.
- Het verdubbelen van de wall shear stress zorgt voor een langzamere binnenkomst van monocyten.
- Volumegroei is in vijf jaar, zelfs bij simulatie 2 waar veel schuimcellen in voorkwamen, amper van te spreken. Dit komt mede doordat er in het model minder cellen meegenomen worden dan dat er in werkelijkheid zijn.
- Indien er een klein aantal typen cellen aanwezig zijn, bijvoorbeeld enkel macrofagen en schuimcellen, ~~dan~~ vormen er sneller clusters van cellen. Indien alle vijf aanwezig zijn in grotere mate, dan vormen er minder clusters.

Chapter 5

Conclusie en discussie

5.1 Conclusie

In dit verslag is een semi-stochastisch cel-gebaseerd model gemaakt voor Atherosclerose. De celtypes die hier in voorkwamen zijn de monocyten, macrofagen, LDL, geoxideerd LDL en schuimcellen. Random Walk, Mechanotaxis en Chemotaxis zijn verwerkt in de bewegingen van de cellen. Van deze bewegingen zijn plots gemaakt en geplaatst in het verslag. Naast de bewegingen zijn ook celsterfte en celdeling in het model verwerkt. Ook zijn vergelijkingen toegepast aangaande het differiënteren van de cellen.

Te concluderen is dat het basismodel af is en werkt. Er zijn nog vele uitbreidingingen mogelijk. De Matlab-code achter is zo gebouwd dat onderzoek naar het gevolg van aanpassen van parameters gebruiksvriendelijk is. Daarnaast is de code toegankelijk voor vele uitbreidingen. Deze zullen hieronder besproken worden.

5.2 Discussie en aanbevelingen

- Er zijn in het model vele aannames gedaan. Zo is er bijvoorbeeld gekozen om chemotaxis te verwerken als het afgeven van sterkere signalen tussen dezelfde type cellen. Er is dus niet gekozen voor een invloed van concentraties, chemicaliën of iets dergelijks. In vervolgonderzoek zou hier naar gekeken kunnen worden hoe dit beter te implementeren valt.

- Te merken is dat er simulaties zijn waarbij Matlab vele malen sneller is dan bij andere. Dit zal te maken hebben met het uitvoeren van meer dan wel minder opdrachten dan wel aanroepen van subprogramma's. Hier ziet ook zeker verbeterruimte.

- In het model is uitgegaan van parameters gevonden in artikelen van andere onderzoekers. Er kan een uitgebreide studie gedaan worden naar hoe deze gevarieerd kunnen worden, maar ook of er meer parameters mogelijk zijn.

- Veel in de simulaties is gerandomized. De radii, startposities van de cellen en nog meer. Er is bijvoorbeeld bekend dat macrofagen in het algemeen een grotere radius hebben dan monocyten. Dit is in het model niet van toepassing. Vervolgonderzoek zou ook plaats kunnen vinden in het vast leggen van dit soort gegevens en kijken wat dit voor invloed heeft.

- Verwijdering van monocyten en geoxideerd LDL heeft random plaats gevonden na het doorlopen van alle cellen. Een mogelijkheid is of bijvoorbeeld cellen dichter aan de rand een grotere kans hebben om uit de plak te gaan. Dergelijke zaken zouden bekeken kunnen worden.
- Er is geen onderzoek gedaan naar of de hoeveelheid keren dat een cel gedeeld heeft invloed heeft op het nogmaals kunnen delen. Uitzoeken of dergelijke zaken biologisch correct zijn en te verwerken zijn is een optie.
- De plak is in werkelijkheid niet rechthoekig en zal ook niet enkel in de y-richting groeien. Vervolgonderzoek zou plaats kunnen vinden naar de invloed van het bloed buiten de plak en de cellen in de plak met betrekking tot de vervorming en uitzetten van de plak.
- In de inleiding is de invloed van medicijnen op dergelijke modellen naar voren gekomen. Ook de invloed van gezondheid, roken, sporten, leeftijd en meer zou verwerkt kunnen worden in het model. Zoals bijvoorbeeld in de eerder genoemde parameters.
- In betere rekenprogramma's zou een driedimensionaal variant van het model gemaakt kunnen worden.

Appendix A

Simulatie 1

Rood	Monocyten
Magenta	Macrofagen
Cyaan	Geoxideerd Cholesterol
Groen	LDL
Geel	Schuimcellen

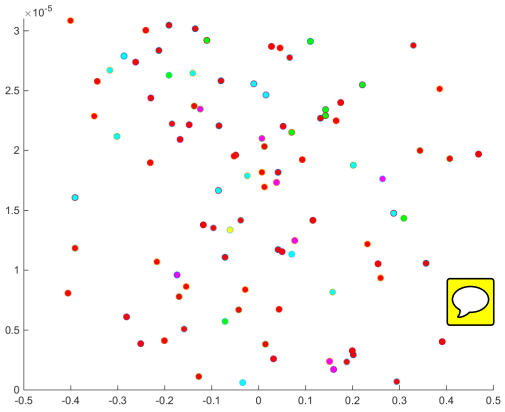

Figure A.1: Tijdstip 0 situatie 1

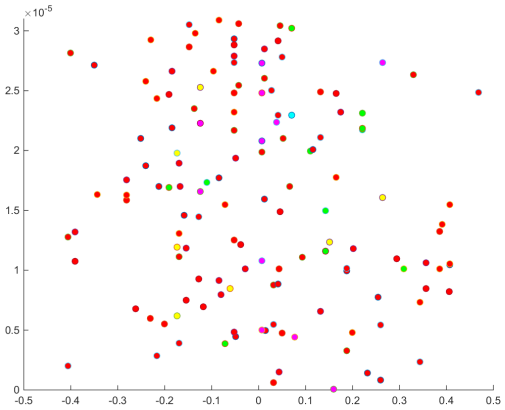

Figure A.2: Situatie 1 na 1 jaar

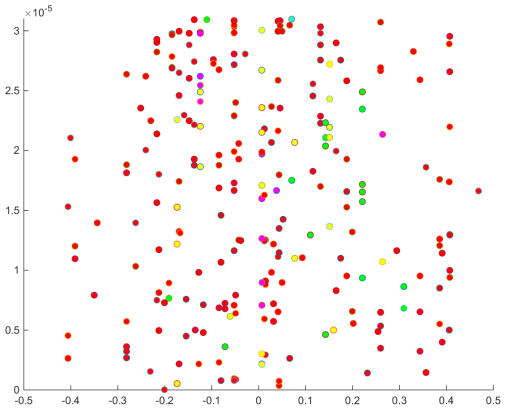

Figure A.3: Situatie 1 na 2 jaar

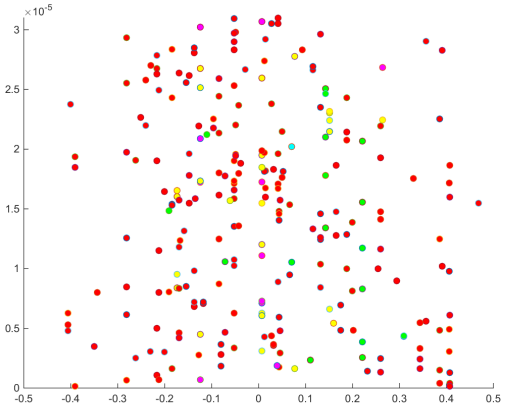

Figure A.4: Situatie 1 na 3 jaar

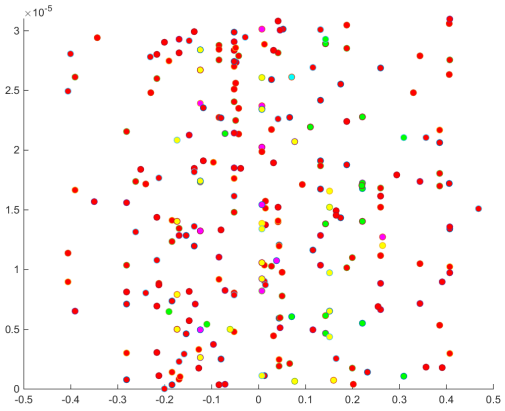

Figure A.5: Situatie 1 na 4 jaar

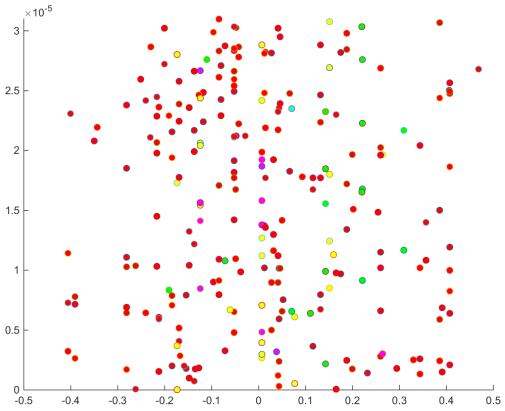

Figure A.6: Situatie 1 na 5 jaar

Appendix B

Simulatie 2

Rood	Monocyten
Magenta	Macrofagen
Cyaan	Geoxideerd Cholesterol
Groen	LDL
Geel	Schuimcellen

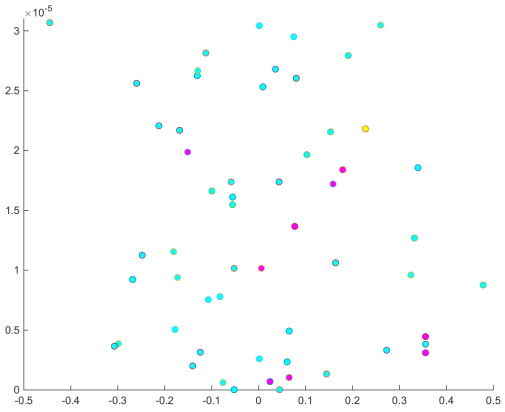

Figure B.1: Na een maand situatie 2

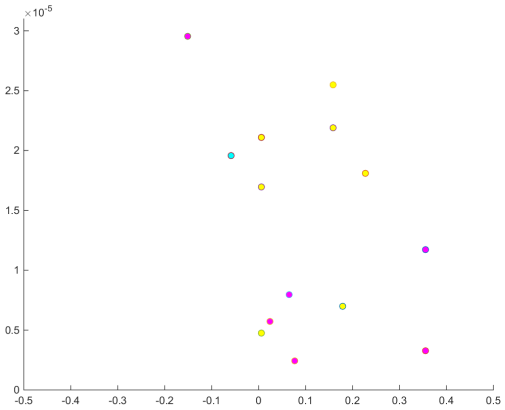

Figure B.2: Situatie 2 na 1 jaar

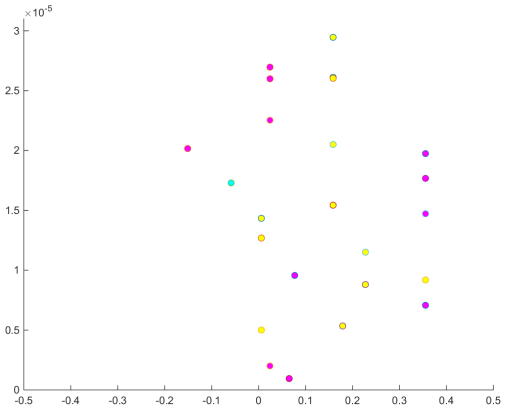

Figure B.3: Situatie 2 na 2 jaar

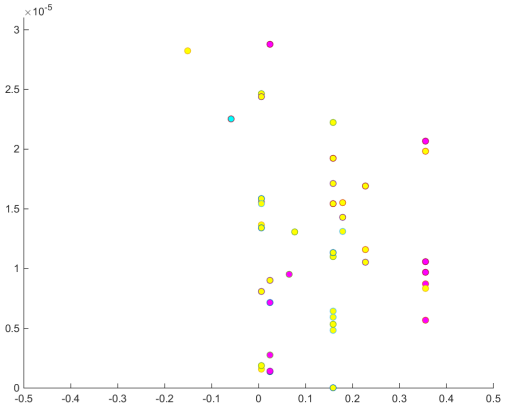

Figure B.4: Situatie 2 na 3 jaar

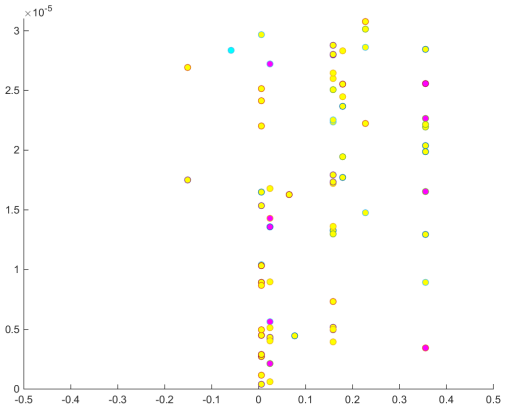

Figure B.5: Situatie 2 na 4 jaar

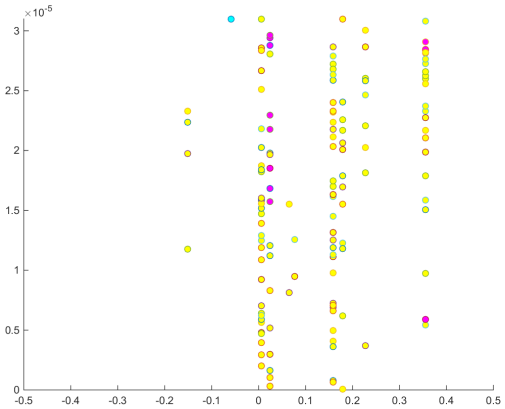

Figure B.6: Situatie 2 na 5 jaar

Appendix C

Parameters

Parameter	Waarde	Eenheid	Betekenis
ρ_1	0.125	1/j	Parameter voor omzetting van monocyten naar macrofagen
ρ_L	2.5	1/j	Parameter voor opname van geoxideerd LDL door macrofagen
ρ_{in}	10	$cm^2/(mol \cdot j)$	Parameter voor omzetting van macrofagen naar schuimcellen
q	7.5	$cm^2/(mol \cdot j)$	Parameter omzetting LDL naar geoxideerd LDL
L_{th}	1	mol/cm^2	Grenswaarde voor geoxideerd cholesterol
m_{th}	1	mol/cm^2	Grenswaarde voor monocyten
γ_0	20	$cm^2/(mol \cdot j)$	De snelheid waarmee monocyten de intima binnenkomen bij lage wall shear stress en lage L_{ox}
σ_w	1	Pa	Wall shear stress
σ_0	1	Pa	Wall shear stress
a/b/e/f	1		Waardes voor numerieke implementatie
ϵ	0.01		Waarde voor numerieke implementatie
c	0.05		Waarde voor numerieke implementatie
d	0.30		Waarde voor numerieke implementatie
r	0.31	cm	Radius van de ader
L_x	1	cm	Breedte van de plak op tijdstip 0
L_y	0.01·r	cm	Hoogte van de plak op tijdstip 0
E_s	5	kPa	Elasticiteit van het substraat
E_c	5	kPa	Elasticiteit van de cel
μ	0.2		Celfrictiecoëfficiënt
β	10	1/s	beweegbaarheidskwantificeerder van het celoppervlak van een levende cel
p	0.01		Kans van sterfte van een cel
q	0.05		Kans van deling van een cel
$\hat{F}$	1	nN	Effectieve opwaartse trekkracht uitgeoefend door een cel
R_c	0.04	μm	Gemiddelde radius van een cel
R_0	0.03	μm	Minimale radius van een cel
R_M	0.06	μm	Maximale radius van een cel
CelTotaal	100		Starthoeveelheid cellen
T	5	jaar	Te bestuderen tijd
dt	1/12	maand	Tijdsstap

Appendix D

Main Code

```
clc;
clear all;

%%% Tijdinterval
T = 5; %Berekeningen op tijdsinterval [0,T]
stap = 0.1; %Stapgroottes
TotStap = round(T/stap); %Totale hoeveelheid tijdstappen

%%%Parameters
Dimensies = 2;
CelTotaal = 100; %Hoeveelheid cellen

%%%Monocyten, LDL, LDL Oxided, Macrophagen, Foam-cellen
Dmp = 0.1; %Diffusie monocyten plak
rho1 = 0.125; %Monocyten naar Macrophagen
rho2 = 2.5; %Opname Lox door Macrophagen
q = 0.3; %Deel van LDL wat oxideert
rho3 = 10; %Macrophagen naar vetten

%%%Celaantrekking&botsing
E_s = 5; %Elasticiteit substraat in kPa
E_c = 0.5; %Elasticiteit cel in kPa
F_dak = 1e-9;%Aantrekkingskracht in N
F_levend = F_dak;
F_dood = 0;
mu = 0.2; %Celfrictiecofficient
cd = 30e-6; %Afstand waarover cellen elkaar maximaal kunnen aantrekken
beta = 10;
%%%Startcondities
%L_0 = 10;
%Lox_0 = 5;
radius = 0.31e-2;
R_0 = 3e-6;
R_c = 4e-6;
R_m = 5e-6; %Radii hemi-spherical cellen
R = [R_c + (R_m-R_0)*(rand(CelTotaal,1)-rand(CelTotaal,1))];
```

```
Volume = zeros(TotStap,2);

%%%Grootte plak bepalen aan de hand van de grootte van de bloedbaan.
%Randen plak bepalen ivm bepalen uiterste posities waar cellen in de plak
%zich later uiterlijk mogen bevinden.
Lpy = 0.01*radius; % Lengte plak y-richting
Lpx = 1; % Lengte plak x-richting

rand_plak_links_x = -0.5*Lpx;
rand_plak_rechts_x = 0.5*Lpx;
rand_plak_boven_y = Lpy;
rand_plak_beneden_y = 0;

%%%Beweegbaarheid in cel
lambda = E_s/E_c;
sigma = sqrt(2*Dmp)*(Lpy/Lpx); %Variantie Random Walk % Daar Random Walk
%te hoge waardes levert, is er geschaald naar substraatgrootte.

%%% Celmatrix vormen
% Kolom 1: Xpositie, Kolom 2: Ypositie, Kolom 3: Indien levend:1, indien
% dood:0
% Kolom 4: Celtype, 1 = monocyt, 2 = Macrofaag, 3 = LDL Geoxideerd , 4 =
% LDL, 5 = Foam
% Kolom 5: Celradius
% Kolom 6: Hoeveel er gedeeld is
Celmatrix = zeros(CelTotaal,6);
%Angeven dat ze levend zijn
for i = 1:CelTotaal
    Celmatrix(i,3) = 1;
end
%Aangeven welk type ze zijn
%Situatie 2
cel_m = 0;
cel_Lox = 40;
cel_M = 60;
cel_L = 0;
for i = 1:(CelTotaal-cel_Lox)
    Celmatrix(i,4) = 2;
end
for i = (CelTotaal-cel_Lox):CelTotaal
     Celmatrix(i,4) = 3;
end

%Situatie 1
rand_m = rand; %randomiseer monocyten
rand_L = rand; %randomiseer LDL
% cel_m = ceil(rand_m/(rand_m+rand_L)*CelTotaal); %Cellen
%die monocyten bij begin
% cel_L = floor(rand_L/(rand_m+rand_L)*CelTotaal); %Cellen
```

```
%die LDL zijn bij begin
% cel_Lox = round(q*cel_L); %Cellen die LDL geoxideerd zijn.
% cel_M = round(rho1*cel_m); %cellen die Macrophagen zijn.
% for i = 1:(cel_m-cel_M)
%     Celmatrix(i,4) = 1;
% end
% for i = (cel_m-cel_M+1):cel_m
%     Celmatrix(i,4) = 2;
% end
% for i = (cel_m+1):(cel_m+(cel_L-cel_Lox))
%     Celmatrix(i,4) = 3;
% end
% for i = (cel_m+(cel_L-cel_Lox)+1):CelTotaal
%     Celmatrix(i,4) = 4;
% end

%Situatie 3
% rand_m = rand;
% rand_Lox = rand;
% cel_m = ceil(rand_m*CelTotaal);
% cel_Lox = floor(rand_Lox*CelTotaal);
% for i = 1:(CelTotaal-cel_Lox)
%     Celmatrix(i,4) = 1;
% end
% for i = (CelTotaal-cel_Lox):CelTotaal
%      Celmatrix(i,4) = 3;
% end
%Random celradius geven
for i = 1:CelTotaal
   Celmatrix(i,5) = R(i,1);
end
%Random positie
rand_pos_x = -0.5*Lpx*rand(CelTotaal,1)+0.5*Lpx*rand(CelTotaal,1);
rand_pos_y = Lpy*rand(CelTotaal,1);
for i = 1:CelTotaal
    Celmatrix(i,1) = rand_pos_x(i,1);
    Celmatrix(i,2) = rand_pos_y(i,1);
end
Totaalaantalcellen = zeros(TotStap,7);
Aantalkeerverwijderen = 0;
for q = 1:35%TotStap
    close all
    %%%Brownian Motion
    dW_1 = zeros(CelTotaal,TotStap);
    dW_2 = zeros(CelTotaal,TotStap);
    for j = 1:CelTotaal
        dW_3 = zeros(1,TotStap);
        dW_4 = zeros(1,TotStap);
        dW_3(1) = sqrt(stap)*randn;
```

```
        dW_4(1) = sqrt(stap)*randn;
        for i = 2:TotStap
            dW_3(i) = sqrt(stap)*randn;
            dW_4(i) = sqrt(stap)*randn;
        end
        dW_1(j,:) = dW_3;
        dW_2(j,:) = dW_4;
    end
    %%%Mechanotaxis
    z = zeros(CelTotaal,2);
    random_leven = rand(CelTotaal,1);
    random_sterfte = rand(CelTotaal,1);
    for i = 1:CelTotaal
        M_0(i) = ((F_levend*Celmatrix(i,3))^2)/(2*(pi^2)...
            *E_s*(Celmatrix(i,5)^4)); %Strain energy density cel i
        epsilon(i) = M_0(i)*exp(-lambda*cd/Celmatrix(i,4));
        %Detectie Threshold
        alpha(i) = F_levend*Celmatrix(i,3)*(Celmatrix(i,5)^3)...
            *beta/(mu*(F_dak^2)); %
    end
    %%% Beweging
    M1 = zeros(CelTotaal, CelTotaal);
    M2 = zeros(CelTotaal,CelTotaal);
    A = zeros(CelTotaal,CelTotaal);
    V = zeros(CelTotaal,2);
    for i = 1:CelTotaal
        %Factor voor de chemotaxis bepalen. Hoe meer van een type cel er
        %is, hoe hoger de aantrekkingsfactor
        Totaal = celtellen(Celmatrix);
        if Celmatrix(i,4) == 1 && Totaal(1,1)>0
            factor = size(Celmatrix,1)/Totaal(1,1);
        end
        if Celmatrix(i,4) == 2 && Totaal(1,2)>0
            factor = size(Celmatrix,1)/Totaal(1,2);
        end
        if Celmatrix(i,4) == 3 && Totaal(1,4)>0
            factor = size(Celmatrix,1)/Totaal(1,4);
        end
        if Celmatrix(i,4) == 4 && Totaal(1,3)>0
            factor = size(Celmatrix,1)/Totaal(1,3);
        end
        if Celmatrix(i,4) == 5 && Totaal(1,5)>0
            factor = size(Celmatrix,1)/Totaal(1,5);
        end
        %Aantrekkingskracht en afstotingskracht
        for k = 1:CelTotaal
            if i~= k
                A(i,k) = norm([Celmatrix(i,1),Celmatrix(i,2)] -...
                    [Celmatrix(k,1),Celmatrix(k,2)]);
```

```
                h = 0.5*max(0,Celmatrix(i,5)+Celmatrix(k,5)-A(i,k));
                %overlapping
                M1(i,k) = (16/25)*(E_c/(sqrt(2)*pi))*(h/Celmatrix(i,5))...
                    ^(5/2); %Afstoting tussen cellen
                V(i,:) = ([Celmatrix(k,1),Celmatrix(k,2)]-[Celmatrix(i,1)...
                    ,Celmatrix(i,2)]);
                if V(i,1) == 0 &&V(i,2) ==0
                    V(i,:) =[0,0];
                else
                    V(i,:) = ([Celmatrix(k,1),Celmatrix(k,2)] -...
                        [Celmatrix(i,1),Celmatrix(i,2)])/...
                        norm([Celmatrix(k,1),Celmatrix(k,2)] -...
                        [Celmatrix(i,1),Celmatrix(i,2)]);
                end
                M2(i,k) = strain(M_0(i), [Celmatrix(i,1),Celmatrix(i,2)]...
                    ,[Celmatrix(k,1),Celmatrix(k,2)],Celmatrix(i,5),lambda);
                if Celmatrix(i,4) == Celmatrix(k,4)
                    M2(i,k) = factor*M2(i,k);
                end
                if epsilon(i)<M2(i,k)
                   M2(i,k)=0;
                end
            else
            M1(i,k) = 0;
            V(i,:) = [0,0];
            M2(i,k) = 0;
            end
            z(i,1) = z(i,1)+M2(i,k)*V(i,1)-M1(i,k)*V(i,1);
            z(i,2) = z(i,2)+M2(i,k)*V(1,2)-M1(i,k)*V(i,2);
        end
        %Om NaN te voorkomen als z 0 is
        if (z(i,1)&&z(i,2))==0
        else
        z(i,:) = z(i,:)/norm([z(i,1),z(i,2)]);
        end
    end

    %%%Celsterfte en Celdeling
    %Celdeling
    ind =0;
    for i = 1:CelTotaal
        if (random_leven(i)>0.95&&Celmatrix(1,3)~=0)
            Celmatrix = [Celmatrix;Celmatrix(i,:)];
            Celmatrix(i,6) = Celmatrix(i,6)+1;
            ind = ind+1;
            Celmatrix(CelTotaal+ind,6)= 0;
        end
    end
    %Celsterfte
```

```
for i = 1:CelTotaal
    if random_sterfte(i)>(1-0.01*stap)
        Celmatrix(i,3) = 0;
    end
end
for i = 1:CelTotaal
    Verandering(i,1) = alpha(i)*sum(M1(i,:)-M2(i,:))*stap*...
        z(i,1)+Celmatrix(i,3)*sigma*dW_1(i,q);
    Verandering(i,2) = alpha(i)*sum(M1(i,:)-M2(i,:))*stap*...
        z(i,2)+Celmatrix(i,3)*sigma*dW_2(i,q);

    if (random_leven(i)>0.95&& Verandering(i,1)==0&&...
            Verandering(i,2)==0 && Celmatrix(i,3) == 1 )
        Celmatrix(i,1) = Celmatrix(i,1)+rand*Celmatrix(i,5);
        Celmatrix(i,2) = Celmatrix(i,2)+rand*Celmatrix(i,5);
    else %Zorgen dat de cellen niet voorbij de rand van de plak gaat.
        if (Celmatrix(i,1)+Verandering(i,1))<=-0.5*Lpx || ...
                (Celmatrix(i,1)+Verandering(i,1))>= 0.5*Lpx || ...
                (Celmatrix(i,2)+Verandering(i,2))<=0 || ...
                (Celmatrix(i,2)+Verandering(i,2))>=Lpy
            Celmatrix(i,1) = Celmatrix(i,1);
            Celmatrix(i,2) = Celmatrix(i,2);
        else
            Celmatrix(i,1) = Celmatrix(i,1)+Verandering(i,1);
            Celmatrix(i,2) = Celmatrix(i,2)+Verandering(i,2);
        end
    end
end
CelTotaal = size(Celmatrix,1);

%%% Binnenkomst nieuwe cellen & posities van die cellen bepalen
Extra_cellen = nieuweposities(Celmatrix,stap,R_0,R_c,R_m,Lpx,Lpy);
Omgezet = omzetcellen(Celmatrix,stap);
%%% Omzetten van Monocyten naar Macrofagen
if Omgezet(1,2)<0
    Celmatrix = monnaarmac(Celmatrix,stap);
end
%%% Omzetten van LDL naar LDLox
if Omgezet(1,3)>0
    Celmatrix = ldlnaarldlox(Celmatrix,stap);
end
%%% Omzetten van Mac naar Foam
if Omgezet(1,5)>0
    Celmatrix = macnaarfoam(Celmatrix,stap);
end
%%% Het verwijderen van Lox uit plak
if Omgezet(1,4)<0
    Celmatrix = verwijderinglox(Celmatrix, Omgezet,stap);
    Aantalkeerverwijderen = Aantalkeerverwijderen+1;
```

```
end
%%% Het verwijderen van monocyten de plak uit
if Omgezet(1,1)<0
   Celmatrix = verwijderingmon(Celmatrix, Omgezet, stap);
end
Celmatrix = [Celmatrix;Extra_cellen];
CelTotaal = size(Celmatrix,1);
Tussenwaarde = celtellen(Celmatrix);
Totaalaantalcellen(q,1) = CelTotaal;
Totaalaantalcellen(q,2) = Tussenwaarde(1,1);
Totaalaantalcellen(q,3) = Tussenwaarde(1,2);
Totaalaantalcellen(q,4) = Tussenwaarde(1,3);
Totaalaantalcellen(q,5) = Tussenwaarde(1,4);
Totaalaantalcellen(q,6) = Tussenwaarde(1,5);
Totaalaantalcellen(q,7) = dodecellentellen(Celmatrix);
%%% Volumegroei bepalen
Tellen = celtellen(Celmatrix);
if q == 1
    Vol = volumegroei(Lpx*Lpy,Tellen(1,1),Tellen(1,2),...
        Tellen(1,3),Tellen(1,4),Tellen(1,5),stap);
    Volume(q) = Lpx*Lpy + Vol;
else
Vol = volumegroei(Volume(q-1),Tellen(1,1),Tellen(1,2),...
    Tellen(1,3),Tellen(1,4),Tellen(1,5),stap);
Volume(q) = Volume(q-1) + Vol;
end
Lengte_y_erbij = Vol/Lpx;
Lpy = Lpy + Lengte_y_erbij;

%%% Plotten van de cellen
clf;
axis off;
set(gcf, 'Renderer','OpenGL');
fig = figure;
hold on
pl = zeros(1,CelTotaal);

for i = 1:CelTotaal
    if Celmatrix(i,4) == 1 %cel is monocyt
       scatter(Celmatrix(i,1),Celmatrix(i,2),'MarkerFaceColor','r');
       %set(pl(i),'EraseMode','Normal');
    elseif Celmatrix(i,4) == 2 %Cel is Macrophaag
        scatter(Celmatrix(i,1),Celmatrix(i,2),'MarkerFaceColor','m');
        %set(pl(i),'EraseMode','Normal');
    elseif Celmatrix(i,4) == 3 %Cel is LDL Ox
        scatter(Celmatrix(i,1),Celmatrix(i,2),'MarkerFaceColor','c');
        %set(pl(i),'EraseMode','Normal');
    elseif Celmatrix(i,4) == 4 %Cel is LDL
        scatter(Celmatrix(i,1),Celmatrix(i,2),'MarkerFaceColor','g');
```

```
            %set(pl(i),'EraseMode','Normal');
        elseif Celmatrix(i,4) == 5 %Cel is Foam
            scatter(Celmatrix(i,1),Celmatrix(i,2),'MarkerFaceColor','y');
            %set(pl(i),'EraseMode','Normal');
        end
    end
    xlim([-0.5*Lpx,0.5*Lpx]);
    ylim([0,Lpy]);
    set(gcf, 'color', 'white');
    set(gca,'color','white');
    set(gcf, 'InvertHardCopy', 'off');
    hold off;
    %export_fig('Fig%d.png',q);
    saveas(fig,sprintf('FIG%d.png',q)); %figuren opslaan
end
disp('Klaar')
```

Appendix E

Omzet cellen

```
function y = omzetcellen(Celmatrix,h)

Totaal = celtellen(Celmatrix);
Informatie = omzetting(Totaal(1,1),Totaal(1,2),...
    Totaal(1,3),Totaal(1,4),Totaal(1,5),h);
%Indien Nieuwe_Monocyten < 0 dan geen monocyten uit het bloed er
%Indien ldlox > 0 dan extra ldlox uit endotheel naar binnen.
%Indien < 0 dan ldlox het bloed weer in.
MacnaarFoam = Informatie(1,5)-Totaal(1,5);
MonnaarMac = Informatie(1,2)-(Totaal(1,2)-MacnaarFoam);
LDLnaarLox = Informatie(1,8);
Lox = Informatie(1,4)-Totaal(1,4)- LDLnaarLox;
Monocytenver = Informatie(1,1)-(Totaal(1,1)-MonnaarMac);
y = [Monocytenver,MonnaarMac,LDLnaarLox,Lox,MacnaarFoam];
end
```

Appendix F

Omzet Cellen 2

```
function y = omzetting(TotMon,TotMac,TotLDL,TotLox,TotFoam,h)
%Indien niet numerieke implementatie gebruikt wordt
rho1 = 0.125;
rhoL = 2.5;
rhoin = 10;
q = 7.5;
Lth = 1;
mth = 1;
p1 = 5;
p2 = 0;
dl = 2.4e-5;
dm = 5.75e-6;
gamma0 = 20;
sigmaw = 1;
sigma0 = 1;

%indien wel numerieke implementatie gebruikt wordt
a = 1;
b = 1;
c = 0.05;
d = 0.3;
e = 1;
f = 1;
eps = 0.01;
sigma = 1;
%indien niet numerieke implementatie gebruikt wordt
% predictor1 = TotMon + round((((gamma0/(1+sigmaw/sigma0))*
%(TotLox/(1+TotLox/Lth))-dm)*TotMon-rho1*TotMon)*h);
% predictor2 = TotMac + round((rho1*TotMon - (rhoin*TotLox*TotMac)/
%(1+TotLox/Lth))*h);
% predictor3 = TotLDL - round(((q*TotMon)/(1+TotMon/mth))*h);
% predictor4 = TotLox + round(((q*TotMon)/(1+TotMon/mth) -
%rhoL*TotMac*TotLox-dl*TotLox)*h);
%
% monocyt = TotMon + round(((((gamma0/(1+sigmaw/sigma0))*(TotLox/
%(1+TotLox/Lth))-dm)*TotMon-rho1*TotMon)+(((gamma0/(1+sigmaw/sigma0))*
```

```
%(TotLox/(1+TotLox/Lth))-dm)*predictor1-rho1*predictor1))*0.5*h);
% macrofaag = TotMac + round(((rho1*TotMon - (rhoin*TotLox*TotMac)/
%(1+TotLox/Lth))+(rho1*TotMon - (rhoin*TotLox*predictor2)/
%(1+TotLox/Lth)))*0.5*h);
% ldl = TotLDL +round((- ((q*TotMon)/(1+TotMon/mth))-
%((q*TotMon*-predictor3)/(1+TotMon/mth)))*0.5*h);
% ldlox = TotLox + round((((q*TotMon)/(1+TotMon/mth) -
%rhoL*TotMac*TotLox-dl*TotLox)+((q*TotMon)/(1+TotMon/mth) -
%rhoL*TotMac*predictor4-dl*predictor4))*0.5*h);
% foam = TotFoam + round((rhoin*TotLox*TotMac)/(1+TotLox/Lth));

%indien wel numerieke implementatie gebruikt wordt
predictor1 = TotMon + round(((a*TotLox)/((1+sigma)*(1+TotLox))...
    -eps-c)*TotMon*h);
predictor2 = TotMac + round((c*TotMon-(c*TotMac*TotLox)/(1+TotLox))*h);
predictor3 = TotLox + round(((d*TotMon)/(f+TotMon)-...
    e*TotLox*TotMac-TotLox)*h);
predictor4 = TotFoam + round(((b*TotLox*TotMac)/(1+TotLox))*h);
predictor5 = TotLDL +round(((-d*TotMon)/(f+TotMon))*h);
%

VerMon = round((((a*TotLox)/((1+sigma)*(1+TotLox))-eps-c)*TotMon...
    + ((a*TotLox)/((1+sigma)*(1+TotLox))-eps-c)*predictor1)*0.5*h);
VerMac = round(((c*TotMon-(c*TotMac*TotLox)/(1+TotLox))+...
    (c*TotMon-(c*predictor2*TotLox)/(1+TotLox)))*0.5*h);
VerLDL = round((((-d*TotMon)/(f+TotMon)))*0.5*h);
VerLox = round((((d*TotMon)/(f+TotMon)-e*TotLox*TotMac-TotLox)+...
    ((d*TotMon)/(f+TotMon)-e*predictor3*TotMac-predictor3))*0.5*h);
VerFoam = round((((b*TotLox*TotMac)/(1+TotLox))+((b*TotLox*TotMac)/...
    (1+TotLox)))*0.5*h);

monocyt = TotMon + round((((a*TotLox)/((1+sigma)*(1+TotLox))-eps-c)*...
    TotMon + ((a*TotLox)/((1+sigma)*(1+TotLox))-eps-c)*predictor1)*0.5*h);
macrofaag = TotMac + round(((c*TotMon-(c*TotMac*TotLox)/(1+TotLox))+...
    (c*TotMon-(c*predictor2*TotLox)/(1+TotLox)))*0.5*h);
ldlox = TotLox + round((((d*TotMon)/(f+TotMon)-e*TotLox*TotMac-TotLox)+...
    ((d*TotMon)/(f+TotMon)-e*predictor3*TotMac-predictor3))*0.5*h);
foam = TotFoam + round((((b*TotLox*TotMac)/(1+TotLox))+((b*TotLox*TotMac)/...
    (1+TotLox)))*0.5*h);
ldl = TotLDL + round((((-d*TotMon)/(f+TotMon)))*0.5*h);

%indien niet numerieke implementatie gebruikt wordt
% monocyt = TotMon + (((a*TotLox)/((1+sigma)*(1+TotLox))-eps-c)*TotMon +
%((a*predictor3)/((1+sigma)*(1+predictor3))-eps-c)*predictor1)*0.5*h;
% macrofaag = TotMac + ((c*TotMon-(c*TotMac*TotLox)/(1+TotLox))+
%(c*predictor1-(c*predictor2*predictor3)/(1+predictor3)))*0.5*h;
% ldlox = TotLox + (((d*TotMon)/(f+TotMon)-e*TotLox*TotMac-TotLox)+
%((d*predictor1)/(f+predictor1)-e*predictor3*predictor2-predictor3))*0.5*h;
% foam = TotFoam + (((b*TotLox*TotMac)/(1+TotLox))+
```

```
%((b*predictor3*predictor2)/(1+predictor3)))*0.5*h;
% ldl = TotLDL + (((-q*TotMon*TotLDL)/(1+(TotMon/mth)))+
%((-q*predictor1*predictor5)/(1+(predictor1/mth))))*0.5*h;

perc = [monocyt,macrofaag,ldl,ldlox,foam,...
    VerMon,VerMac,VerLDL,VerLox,VerFoam];
y = perc;
end
```

Appendix G

Celtellen

```
function y = celtellen(Celmatrix)
Mon = 0;
LDL = 0;
Lox = 0;
Mac = 0;
Foam = 0;
%Celtellen spreekt voor zich
for i = 1: size(Celmatrix,1)
    if Celmatrix(i,3) == 1 && Celmatrix(i,4) == 1
       Mon = Mon+1;
    elseif Celmatrix(i,3) == 1 && Celmatrix(i,4) == 2
       Mac = Mac+1;
    elseif Celmatrix(i,3) == 1 && Celmatrix(i,4) == 3
       Lox = Lox+1;
    elseif Celmatrix(i,3) == 1 && Celmatrix(i,4) == 4
       LDL = LDL+1;
    elseif Celmatrix(i,3) == 1 && Celmatrix(i,4) == 5
       Foam = Foam+1;
    end
end
y = [Mon,Mac,LDL,Lox,Foam];
end
```

Appendix H

Tellen van monocyten

```
function y = monocytmatrix(Celmatrix)
Totaal = celtellen(Celmatrix);
Mon = zeros(Totaal(1,1),6);
rijmon = 1;
for i = 1:size(Celmatrix,1);
    if  Celmatrix(i,4) == 1 && Celmatrix(i,3) == 1
        Mon(rijmon,:) = Celmatrix(i,:);
        rijmon = rijmon + 1;
    end
end
y = Mon;
end
```

Appendix I

Tellen van macrofagen

```
function y = macrofaagmatrix(Celmatrix)
Totaal = celtellen(Celmatrix);
Mac = zeros(Totaal(1,2),6);
rijmac = 1;
for i = 1:size(Celmatrix,1);
    if  Celmatrix(i,4) == 2 && Celmatrix(i,3) == 1
        Mac(rijmac,:) = Celmatrix(i,:);
        rijmac = rijmac + 1;
    end
end
y = Mac;
end
```

Appendix J

Tellen van LDL

```
function y = ldlmatrix(Celmatrix)
Totaal = celtellen(Celmatrix);
LDL = zeros(Totaal(1,3),6);
rijldl = 1;
for i = 1:size(Celmatrix,1);
    if  Celmatrix(i,4) == 4 && Celmatrix(i,3) == 1
        LDL(rijldl,:) = Celmatrix(i,:);
        rijldl = rijldl + 1;
    end
end
y = LDL;
end
```

Appendix K

Tellen van LDLox

```
function y = ldloxmatrix(Celmatrix)
Totaal = celtellen(Celmatrix);
LDLox = zeros(Totaal(1,4),6);
rijldlox = 1;
for i = 1:size(Celmatrix,1);
    if  Celmatrix(i,4) == 3 && Celmatrix(i,3) == 1
        LDLox(rijldlox,:) = Celmatrix(i,:);
        rijldlox = rijldlox + 1;
    end
end
y = LDLox;
end
```

Appendix L

Tellen van Foam

```
function y = foammatrix(Celmatrix)
Totaal = celtellen(Celmatrix);
Foam = zeros(Totaal(1,5),6);
rijfoam = 1;
for i = 1:size(Celmatrix,1);
    if  Celmatrix(i,4) == 5 && Celmatrix(i,3) == 1
        Foam(rijfoam,:) = Celmatrix(i,:);
        rijfoam = rijfoam + 1;
    end
end
y = Foam;
end
```

Appendix M

Monocyten naar Macrofagen

```
function y = monnaarmac(Celmatrix,h)
Informatie = omzetcellen(Celmatrix,h);
Totaal = celtellen(Celmatrix);
startkans = abs(Informatie(1,2))/Totaal(1,1);
StartMon = Totaal(1,1);
RandomDoorMatrix = randperm(length(Celmatrix));
for i = 1:length(Celmatrix)
    if i == 1
        kans = startkans;
        huidigomtezetten = abs(Informatie(1,2));
        nognategaan = StartMon;
    else
        kans = huidigekans;
    end
    if Celmatrix(RandomDoorMatrix(i),4) == 1 && Celmatrix(RandomDoorMatrix(i),
       if rand(1,1)>=kans
           Celmatrix(RandomDoorMatrix(i),4) = 2;
           huidigomtezetten = huidigomtezetten - 1;
       end
       nognategaan = nognategaan - 1;
       huidigekans = huidigomtezetten/nognategaan;
    else
        huidigekans = kans;
    end
end
y = Celmatrix;
end
```

Appendix N

Macrofagen naar schuimcellen

```
function y = macnaarfoam(Celmatrix,h)
Informatie = omzetcellen(Celmatrix,h);
Totaal = celtellen(Celmatrix);
startkans = Informatie(1,5)/Totaal(1,2);
StartMac = Totaal(1,2);
RandomDoorMatrix = randperm(length(Celmatrix));
for i = 1:length(Celmatrix)
    if i == 1
        kans = startkans;
        huidigomtezetten = Informatie(1,5);
        nognategaan = StartMac;
    else
        kans = huidigekans;
    end
    if Celmatrix(RandomDoorMatrix(i),4) == 2 && Celmatrix(RandomDoorMatrix(i),
       if rand(1,1)>=kans
           Celmatrix(RandomDoorMatrix(i),4) = 5;
           huidigomtezetten = huidigomtezetten - 1;
       end
       nognategaan = nognategaan - 1;
       huidigekans = huidigomtezetten/nognategaan;
    else
       huidigekans = kans;
    end
end
y = Celmatrix;
end
```

Appendix O

LDL naar LDLox

```
function y = ldlnaarldlox(Celmatrix,h)
Informatie = omzetcellen(Celmatrix,h);
Totaal = Celtellen(Celmatrix);
startkans = Informatie(1,3)/Totaal(1,3);
StartLDL = Totaal(1,3);
RandomDoorMatrix = randperm(length(Celmatrix));
for i = 1:length(Celmatrix)
    if i == 1
        kans = startkans;
        huidigomtezetten = Informatie(1,3);
        nognategaan = StartLDL;
    else
        kans = huidigekans;
    end
    if Celmatrix(RandomDoorMatrix(i),4) == 4 && Celmatrix(RandomDoorMatrix(i),
       if rand(1,1)>=kans
           Celmatrix(RandomDoorMatrix(i),4) = 3;
           huidigomtezetten = huidigomtezetten - 1;
       end
       nognategaan = nognategaan - 1;
       huidigekans = huidigomtezetten/nognategaan;
    else
       huidigekans = kans;
    end
end
y = Celmatrix;
end
```

Appendix P

Nieuwe cellen plaatsen

```
function y = nieuweposities(Celmatrix,h, R_0, R_c, R_m, Lpx,Lpy)
Informatie = omzetcellen(Celmatrix,h);
%Bepalen hoeveel cellen erbij moeten
if Informatie(1,1)>0
   Nieuw_Mon = Informatie(1,1);
else
    Nieuw_Mon =0;
end
if Informatie(1,4)>0
   Nieuw_Lox = abs(Informatie(1,4));
else
   Nieuw_Lox = 0;
end
%Zoals in de maincode, bepalen van radius en positie cellen
Nieuw_cellen = Nieuw_Mon + Nieuw_Lox;
A = zeros(Nieuw_cellen,6);
R = [R_c + (R_m-R_0)*(rand(Nieuw_cellen,1)-rand(Nieuw_cellen,1))];
rand_pos = -0.5*Lpx*rand(Nieuw_cellen,1)+0.5*Lpx*rand(Nieuw_cellen,1);
%Alles toekennen
for i = 1:Nieuw_cellen
   A(i,1) = rand_pos(i,1);
   A(i,2) = 0.01+R(i,1);
   A(i,3) = 1;
    if i<=Nieuw_Mon
       A(i,4) = 1;
    else
       A(i,4) = 3;
    end
   A(i,5) = R(i,1);
end
y = A;
end
```

Appendix Q

Verwijderen van monocyten uit de plak

```
function y = verwijderingmon(Celmatrix,Omgezet,h)
Totaal = celtellen(Celmatrix);
Informatie = Omgezet;
TotVerMon = abs(Informatie(1,1));
StartTotMon = Totaal(1,1);
startkans = abs(Informatie(1,1)/Totaal(1,1));
rij = 1;
TussenMatrix = zeros(length(Celmatrix)-abs(Informatie(1,1)),6);
RandomDoorMatrix = randperm(length(Celmatrix));
for i = 1:length(Celmatrix)
      if i == 1
          kans = startkans;
          huidigomtezetten = TotVerMon;
          nognategaan = StartTotMon;
      else
          kans = huidigekans;
      end
      if Celmatrix(RandomDoorMatrix(i),4) == 1 && ...
            Celmatrix(RandomDoorMatrix(i),3) == 1 && kans~=0
          if rand(1,1)>=kans
             huidigomtezetten = huidigomtezetten - 1;
          else
             TussenMatrix(rij,:) = Celmatrix(RandomDoorMatrix(i),:);
             rij = rij+1;
          end
             nognategaan = nognategaan-1;
             huidigekans = huidigomtezetten/nognategaan;
      else
          huidigekans = kans;
          TussenMatrix(rij,:) = Celmatrix(RandomDoorMatrix(i),:);
          rij = rij+1;
      end
end
```

```
y = TussenMatrix;
end
```

Appendix R

Verwijderen van LDLox

```
function y = verwijderinglox(Celmatrix,Omgezet,h)
Totaal = celtellen(Celmatrix);
Informatie = Omgezet;
TotVerLox = abs(Informatie(1,4));
StartTotLox = Totaal(1,4);
startkans = abs(Informatie(1,4)/Totaal(1,4));
rij = 1;
%Aanmaken van de matrix en om elke LDL deeltje gelijke
%kans te geven om de
%plak uit te gaan wordt er random door de matrix gelopen
TussenMatrix = zeros(length(Celmatrix)-abs(Informatie(1,4)),6);
RandomDoorMatrix = randperm(length(Celmatrix));
for i = 1:size(Celmatrix,1)
      if i == 1
          kans = startkans;
          huidigomtezetten = TotVerLox;
          nognategaan = StartTotLox;
      else
          kans = huidigekans;
      end
      if Celmatrix(RandomDoorMatrix(i),4) == 3 && ...
            Celmatrix(RandomDoorMatrix(i),3) == 1 && kans~=0
          if rand(1,1)>=kans
             huidigomtezetten = huidigomtezetten - 1;
          else
             TussenMatrix(rij,:) = Celmatrix(RandomDoorMatrix(i),:);
             rij = rij+1;
          end
             nognategaan = nognategaan-1;
             huidigekans = huidigomtezetten/nognategaan;
      else
          huidigekans = kans;
          TussenMatrix(rij,:) = Celmatrix(RandomDoorMatrix(i),:);
          rij = rij+1;
      end
end
```

```
y = TussenMatrix;
end
```

Appendix S

Volumegroei

```
function y = volumegroei(Volume, TotMon, TotMac,TotLDL,TotLox,TotFoam,h)
%Definiren moleculaire volumes
vM = 10e-14;
vLox = 10e-21;
vm = 10e-17;
vF = 10e-13;

Veranderingen = omzetting(TotMon,TotMac,TotLDL,TotLox,TotFoam,h);

predictor = Volume + Volume*(vm*Veranderingen(1,6)+vM*Veranderingen(1,7)+vLox*
groei = (Volume*(vm*Veranderingen(1,6)+vM*Veranderingen(1,7)+vLox*Veranderinge
y = groei;
end
```

Bibliography

[1] Gefen A. Effects of virus size and cell stiffness on forces, work and pressures driving membrane invagination in a receptor-mediated endocytosis. *J Biomech Eng*, 132, 2010.

[2] J.L.A. Dubbeldam. Long time evolution of atherosclerotic plaques. *Journal of Theoretical Biology*, 297:1–10, 2012.

[3] Krombach et al. Cell size of alveolar macrophages: an interspecies comparison. *Environ. Health Perspect.*, 20:1261–1263, 1997.

[4] R.G. Gerrity. Monocytes and atherogenesis. *Am. J. Pathol.*, 103:191–200, 1981.

[5] J. Hulthe. The metabolic syndrome, ldl particle size and atherosclerosis. *Atheroscler. Thromb. Vasc. Biol.*, 20:2140–2147, 2000.

[6] Hammer DA Reinhart-King CA, Dembo M. Cell-cell mechanical communication through compliant substrates. *Biophys Journal*, 95:6044–6051, 2008.

[7] F.J. Vermolen and A Gefen. A semi-stochastic cell-based formalism to model the dynamics of migration of cells in colonies. *Biomech Model Mechanobiol*, 11:183–195, 2012.

[8] F.J. Vermolen, R. van der Meijden, M. van Es, A. Gefen, and D. Weihs. Towards a mathematical formalism for semi-stochastic cell-level computational modeling of tumor initiation. pages 6–9.