



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

**Een verbeterd semi-stochastisch model voor
wondheling onder infectie**
**An improved semi-stochastic model for infected
wound healing**

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

Daan Smits

Delft, Nederland
Juli 2015

Copyright © 2015 door Daan Smits. Alle rechten voorbehouden.



BSc verslag TECHNISCHE WISKUNDE

“Een verbeterd semi-stochastisch model voor wondheling onder infectie”

“An improved semi-stochastic model for infected wound healing”

Daan Smits

Technische Universiteit Delft

Begeleider

Dr.ir. F.J. Vermolen

Overige commissieleden

Dr. J.L.A. Dubbeldam

Drs. E.M. Van Elderen

Juli, 2015

Delft

Inhoudsopgave

Abstract	5
Motivatie	5
Probleemstelling	6
1 Biologische achtergrond	7
1.1 Fases van een wondhelingsproces	7
1.2 Migratie, deling en sterfte	8
1.3 Bacteriën en witte bloedcellen	9
2 Het wiskundige model	11
2.1 Basismodel	11
2.1.1 Actieve migratie	12
2.1.2 Passieve migratie	13
2.1.3 Bewegingsrichting	15
2.1.4 Verstoringen in mechanische signalen	15
2.1.5 Celdeling	16
2.1.6 Apoptose	18
2.2 Bacteriën	18
2.2.1 Beweging en deling van bacteriën	18
2.2.2 Melkzuurconcentratie	19
2.2.3 Invloed op celmobiliteit	19
2.2.4 Invloed op de duur van een delingscyclus	19
2.3 Witte bloedcellen	20
2.3.1 Constructie bloedvatenstelsel	20
2.3.2 Creatie en sterfte	21
2.3.3 Beweging van witte bloedcellen	22
2.3.4 Melkzuurafbraak	22
2.4 Parameters	23
3 Numerieke analyse	25
3.1 Tijdsintegratie	25
3.1.1 Keuze integratiemethode	25
3.1.2 Grootte tijdstap	26
3.2 Benadering melkzuurconcentratie	26
3.2.1 Quasi-stationaire oplossing	27
3.2.2 Finite Volume Method	28
4 Resultaten	35
4.1 Numeriek vs. analytisch voor twee cellen	35

4.2	Beweging van vier cellen	36
4.3	Koloniegroei	37
4.4	Heling van een schone wond	38
4.5	Heling van een licht geïnfecteerde wond	39
4.6	Zware infectie	41
	Conclusie/Discussie	43
	Bijlagen	45

Abstract

Voor dit project is een bestaand semi-stochastisch cell-based model voor wondheling onder infectie bekeken, dat migratie, deling, sterfte, melkzuurproducerende bacteriën en witte bloedcellen omvat. Dit model wordt uitgebreid, onder andere met een dynamische implementatie van celdeling. Daarnaast wordt met behulp van numerieke methoden de benodigde rekenkracht gereduceerd, om zo simulaties mogelijk te maken die met dit model voorheen niet konden worden uitgevoerd.

Motivatie

Een wiskundig model is een beschrijving van een systeem in wiskundige formules. Het model kan worden gebruikt om het gedrag van het systeem te voorspellen. Daarnaast geeft het inzicht in het betreffende systeem, en dat inzicht kan weer worden teruggekoppeld op het systeem, om zo te sturen en te optimaliseren. Wiskundig modelleren vindt zijn toepassing in vele vakgebieden.

Soms zijn experimenten ingewikkeld of tijdrovend. Soms zijn ze zelfs praktisch of ethisch niet uitvoerbaar, waarbij bijvoorbeeld kan worden gedacht aan de verspreiding van een epidemie. De steeds groter wordende rekenkracht van computers en de ontwikkeling van nieuwe en betere rekenmethoden maken wiskundig modelleren almaar populairder.

In dit verslag wordt een wondhelingsproces bekeken. Experimenten met wondheling nemen vaak dagen of weken in beslag. Een model kan hier als alternatief een aanzienlijke tijdswinst opleveren. In *Semi-stochastische aanpak van migratie, sterfte en deling in geïnfecteerde celkolonies* door Liselot Arkesteijn [1] wordt een model beschreven van een wondhelingsproces, dat zich redelijk realistisch gedraagt. In de opgenomen discussie staat te lezen dat het model op sommige punten een wellicht te sterk vereenvoudigde weergave van de werkelijkheid geeft, en dat sommige simulaties veel tijd in beslag nemen vanwege de benodigde rekenkracht.

Dit verslag zal de constructie van een op [1] gebaseerd model beschrijven, dat meer naar de werkelijkheid neigt. Daarnaast zal de benodigde rekenkracht worden teruggebracht.

Probleemstelling

Bij aanvang van dit project was het de bedoeling om het model in [1] te verbeteren. Wat de verbeteringen precies zouden moeten zijn is toen niet vastgelegd; gaandeweg is bekeken waar er relevante aanpassingen aan het model konden worden gedaan. Zoals in de motivatie te lezen staat heeft dit uiteindelijk geleid tot een realistischer en minder rekenkracht behoevend model. De in dit verslag te beantwoorden hoofdvraag is: “Hoe kan een realistischer en tijdsefficiënt model van een wondhelingsproces worden geconstrueerd?”.

Hoofdstuk 1

Biologische achtergrond

Om een goed model te kunnen construeren, is het zaak kennis te nemen van de biologische processen die bij een wondhelingsproces horen. In deze paragraaf worden de belangrijkste te modelleren factoren toegelicht, alsmede hun samenhang bij wondheling. Met name worden besproken: actieve celmigratie, passieve celmigratie, celdeling, celsterfte, bacteriën, angiogenese en witte bloedcellen.

1.1 Fases van een wondhelingsproces

Een wondhelingsproces beslaat drie fasen: de ontstekingsfase, de proliferatiefase, en de maturatiefase. De ontstekingsfase is de eerste fase die een wond doorloopt. Deze fase begint met hemostase, een mechanisme dat het lichaam heeft om bloedverlies te voorkomen: bloedplaatjes, de kleinste cellen in het bloed, zorgen voor stolling, en ze zenden chemische signalen uit die onder andere immuuncellen activeren; slagaders verwijden en bloedvaten vernauwen. Als er genoeg stolling heeft plaatsgevonden om het bloeden te doen stoppen, verwijden de bloedvaten weer, om zo witte bloedcellen de kans te geven de wond binnen te treden. Deze witte bloedcellen zullen beginnen met het opruimen van vuil, dode cellen en bacteriën. Aan het eind van de ontstekingsfase zullen nog stoffen worden geproduceerd die een positieve invloed hebben op celmigratie en -deling, om de wond zo voor te bereiden op de proliferatiefase.

Tijdens de proliferatiefase is er, onder andere, sprake van angiogenese, vorming van granulatieweefsel en epithelisatie. Angiogenese, de vorming van nieuwe bloedvaten, vindt plaats om de cellen optimaal te kunnen voorzien van voedingsstoffen, en zorgt ervoor dat witte bloedcellen de wond sneller en in grotere mate binnen kunnen treden. Granulatieweefsel kan worden gezien als een tijdelijk stuk huid. Het oogt lichtroze en is wat vochtig. Verder is doet het geen pijn en voelt het zacht aan. Granulatieweefsel is veel minder stevig dan 'normale' huid. Epiheelcellen zijn de cellen die het oppervlak van het lichaam bekleden. Ze vormen samen het epitheel, het oppervlakkige deel van de huid. Een wond kan grofweg worden opgevat als een gat in het epitheel, dus een deelgebied van

het lichaam waar zich geen epiheelcellen meer bevinden. De epithelisatie is het proces waarbij het lichaam probeert dit gat te dichten.

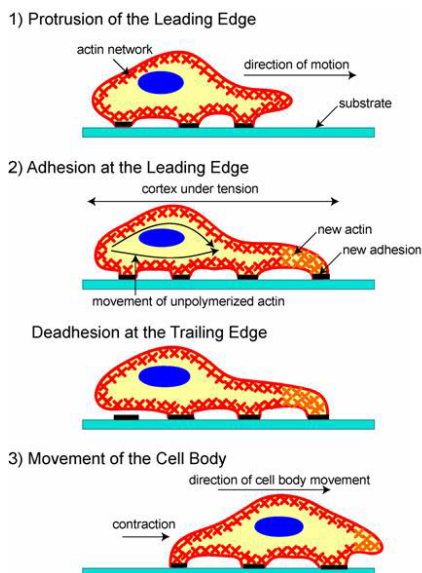
De laatste fase is de maturatiefase. Bij aanvang van de maturatiefase is de wond helemaal dicht. Het stuk huid waar de wond zich bevond is nog wat gevoelig en licht van kleur. Tijdens de maturatiefase zal het lichaam proberen dit stuk huid helemaal in de oorspronkelijke staat terug te brengen. Dit proces kan maanden of zelfs jaren duren.

Opgemerkt dient te worden dat de ontstekingsfase en de proliferatiefase elkaar kunnen overlappen. Zo kunnen er, bijvoorbeeld, tijdens de proliferatiefase nog steeds bacteriën in de wond aanwezig zijn, en daarom ook witte bloedcellen die proberen deze bacteriën op te ruimen.

1.2 Migratie, deling en sterfte

In dit verslag zal in eerste instantie de proliferatiefase worden gemodelleerd, waarbij voornamelijk wordt gekeken naar epithalisatie. Om epithalisatie verder uiteen te zetten zullen nu celmigratie, celdeling en celsterfte worden toegelicht.

Cellen liggen in het lichaam in de extra-cellulaire matrix, het deel van het weefsel dat niet uit cellen bestaat. De extra-cellulaire matrix wordt vaak aangeduid met de term 'substraat'. Figuur 2.1 laat zien hoe een cel over een substraat migreert:



Figuur 1.1: Een migrerende cel.

De afbeelding toont *actieve* migratie. Te zien is dat aan de voorkant van de cel steeds een nieuwe verbinding met het substraat wordt gecreëerd. Daarna wordt

de oude verbinding aan de achterkant van de cel opgeheven, zodat het cellichaam naar de nieuwe verbinding beweegt. Ten behoeve van dit proces trekt een cel aan het substraat, en zo wordt een vervorming van het substraat veroorzaakt. Een andere cel zal de vervorming van het substraat opmerken. Dit kan worden beschouwd als een vorm van communicatie tussen cellen. De grootte van een 'mechanisch signaal' dat een cel van een andere cel ontvangt hangt af van drie zaken. Ten eerste van de afstand tussen de cellen: hoe dichterbij een cel, hoe sterker de door hem veroorzaakte vervorming door een andere cel zal worden gevoeld. Ten tweede van het gemak waarmee het substraat zich door cellen laat vervormen: de substraatstijfheid. Als laatste speelt natuurlijk de kracht waarmee een cel aan het substraat trekt een rol. De combinatie van alle signalen die een cel ontvangt bepaalt in grote mate de bewegingsrichting en -snelheid van de cel; verondersteld wordt dat de beweging van de cellen voor een klein deel willekeurig is, door verstoringen in de signalen die zij ontvangen.

Naast actieve migratie is er nog een andere vorm van migratie. Als twee cellen zich bij elkaar in de buurt bevinden, kan het zo zijn dat ze elkaar indeuken. In dat geval zullen ze direct een kracht op elkaar uitoefenen, en elkaar zo wegduwen: dit wordt *passieve* migratie genoemd.

Celdeling is het proces waarbij één cel verandert in twee identieke cellen, en wordt algemeen aangeduid met de term 'mitose'. Een delingscyclus bestaat grofweg uit twee fasen. In de eerste fase, de interfase, kopieert een cel zijn genetische materiaal. Daarbij groeit de cel tot hij ongeveer het volume heeft van twee cellen. In de tweede fase, de mitotische fase, worden de gekopieerde DNA-strengen van elkaar gescheiden, waarna de cel splitst tot twee cellen.

Er zijn twee soorten celsterfte: necrose en apoptose. Necrose is celsterfte die wordt veroorzaakt door schade van buitenaf, bijvoorbeeld een infectie of letsel. In dit verslag wordt niet naar necrose gekeken. De andere vorm van celsterfte, apoptose, is gecontroleerd. De cel gaat dan over tot een geprogrammeerde zelfmoord, en zorgt er zo voor dat er zo min mogelijk afval in het weefsel achterblijft. Apoptose doet zich bijvoorbeeld voor als een cel te oud is, maar kan ook andere oorzaken hebben. Zo speelt dit proces een belangrijke rol bij het scheiden van de vingers van een embryo. Ook kan apoptose optreden als gevolg van een te hoge celdruk: als zich in een weefsel te veel cellen bevinden, zullen zij druk op elkaar uitoefenen, en dat kan apoptose inleiden. Op deze manier kan de balans in de hoeveelheid cellen in een weefsel worden gehandhaafd.

1.3 Bacteriën en witte bloedcellen

Eerst is het belangrijk op te merken dat er heel veel verschillende bacteriën zijn, met verschillende eigenschappen. Hetzelfde geldt voor witte bloedcellen. In dit verslag wordt geen onderscheid gemaakt tussen verschillende soorten bacteriën en witte bloedcellen. De in het model gebruikte parameters zijn generiek. Als op een zeker moment de behoefte bestaat het model te gebruiken voor specifieke bacteriën danwel witte bloedcellen, kunnen dan de parameters worden aangepast.

Als zich in een wond bacteriën bevinden, zullen zij de epithelisatie bemoeilijken.

Ze zullen met de epitheelcellen gaan concurreren om de aanwezige voedingsstoffen. Dit zorgt ervoor dat de delingscyclus van een epitheelcel meer tijd in beslag zal nemen. De bacteriën zullen ook melkzuur produceren, en daarmee de pH-waarde in het weefsel verlagen. De beweeglijkheid van epitheelcellen is afhankelijk van de pH-waarde van het weefsel waarin ze zich bevinden: er wordt aangenomen dat de mobiliteit exponentieel afneemt als de melkzuurconcentratie toeneemt.

Witte bloedcellen komen de wond binnen via bloedvaten, en zijn dus afhankelijk van angiogenese. Zodra er nieuwe bloedvaten naar de wond zijn aangelegd, wordt het voor de witte bloedcellen veel gemakkelijker deze binnen te treden. Ze zullen de bacteriën gaan doden en opruimen, en daarnaast de melkzuurconcentratie weer verlagen.

Hoofdstuk 2

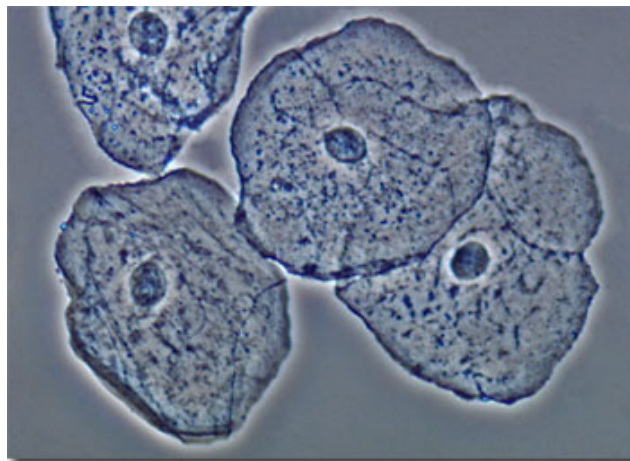
Het wiskundige model

In dit hoofdstuk zal het wiskundige model dat is geconstrueerd worden beschreven. Het basismodel is grotendeels gebaseerd op [1]. In [2] wordt een beschrijving gegeven van de delingscyclus die een cel doorloopt, alsmede een model daarvan. Deze celdelingscyclus wordt in het basismodel geïntegreerd. Bij het toevoegen van de bacteriën en witte bloedcellen wordt bekeken welke invloed deze hebben op de celdelingscyclus. Daarnaast worden in dit model witte bloedcellen in staat gesteld om de melkzuurconcentratie in de wond te verlagen.

2.1 Basismodel

In deze paragraaf wordt het basismodel toegelicht. In dit basismodel zijn interactie tussen epitheelcellen, celdeling en apoptose verwerkt.

In de onderstaande figuur zien we epitheelcellen op een substraat.



Figuur 2.1: Vier epitheelcellen.

We zien hier dat epitheelcellen complex van vorm zijn, en niet allemaal identiek. In dit model beschouwen we het substraat als een domein $\bar{\Omega} \subseteq \mathbb{R}^2$. Epitheelcellen veronderstellen we cirkelvormig, zodat het gedeelte van het substraat dat een cel in beslag neemt eenduidig kan worden gespecificeerd met een punt $\mathbf{r} = (x, y)^T \in \bar{\Omega}$, en een straal $R \in \mathbb{R}$.

2.1.1 Actieve migratie

Stel dat er zich op het substraat twee cellen bevinden. Beide cellen trekken aan het substraat en veroorzaken zo beide een vervorming. De cellen zullen elkaars vervorming waarnemen, en zodoende elkaars positie weten. Hierop zullen ze naar elkaar toe gaan bewegen. Complexer wordt dit, als er meer dan twee cellen op het substraat aanwezig zijn. In dit geval zal iedere cel meerdere vervormingen waarnemen, en zal hij zijn bewegingsrichting baseren op een combinatie van deze vervormingen. Dit zal nu formeel worden toegelicht.

Zeg dat er zich op het substraat $n \in \mathbb{N}$ cellen bevinden. Een cel wordt aangeduid met $i \in \mathbb{N}$, $1 \leq i \leq n$. Iedere vervorming zorgt ervoor dat er energie wordt opgeslagen in het substraat. Alle vervormingen bij elkaar opgeteld bepalen een verdeling van een zogenoemde rekenergiedichtheid. Het verband tussen de rekenergie E_{rek} , spanning σ en rek ϵ luidt:

$$E_{\text{rek}} = \frac{1}{2} \sigma \epsilon. \quad (2.1)$$

Als wordt aangenomen dat de rek van het substraat recht evenredig is met de spanning die erop wordt uitgeoefend, mag, omdat de vervormingen heel klein zijn, de Wet van Hooke worden toegepast. Als de elasticiteit van het substraat met E_s wordt genoteerd, luidt deze:

$$\sigma = E_s \epsilon. \quad (2.2)$$

De rekenergiedichtheid, ten gevolge van de kracht die cel i uitoefent, in het middelpunt van cel i , wordt genoteerd met $M_i(0)$. Deze notatie zal later aannemelijk worden. Samen induceren vergelijkingen (2.1) en (2.2):

$$M_i(0) = \frac{1}{2} E_s \epsilon^2. \quad (2.3)$$

Hierbij is ϵ de rek van het substraat ter hoogte van het middelpunt van cel i . De spanning is gedefinieerd als de kracht per oppervlakte-eenheid. Omdat cellen als cirkelvormig worden beschouwd, geldt dus:

$$\sigma = \frac{F_i}{\pi R_i^2}, \quad (2.4)$$

waarbij R_i de straal van cel i is, en F_i de kracht die cel i op het substraat uitoefent. Voor een levende cel wordt $F_i = \hat{F} \cdot (R_i/R_0)^2$ genomen, analoog aan [2]. R_0 is de straal van een cel direct na deling, en \hat{F} is de kracht die een cel op het substraat uitoefent direct na deling. Met behulp van (2.2) geldt nu:

$$\epsilon = \frac{F_i}{\pi E_s R_i^2}. \quad (2.5)$$

Als dit weer wordt ingevuld in (2.3), ontstaat de volgende uitdrukking voor $M_i(0)$:

$$M_i(0) = \frac{F_i^2}{2\pi^2 E_s R_i^4}. \quad (2.6)$$

Nu kan het mechanische signaal op een zekere afstand van cel i worden berekend. In [3] staat te lezen dat dit exact kan met behulp van Besselfuncties. Echter kan een goede benadering worden verkregen met e-machten. Omdat in dit verslag grote hoeveelheden cellen zullen worden bekeken, is het wat betreft de tijdsefficiëntie aantrekkelijker om niet voor de exacte oplossing met Besselfuncties te kiezen. De benadering voor het mechanische signaal op positie $\mathbf{r} \in \bar{\Omega}$ is:

$$M_i(\mathbf{r}) = \frac{F_i^2}{2\pi^2 E_s R_i^4} \cdot \exp\left(-\lambda \frac{\|\mathbf{r} - \mathbf{r}_i\|}{R_i}\right), \quad (2.7)$$

met $\mathbf{r}_i \in \bar{\Omega}$ het middelpunt van cel i , en λ een constante die de mate van uitdemping van het mechanische signaal representeert. Er wordt in dit verslag, net als in [1], $\lambda = E_s/E_c$ gebruikt. De stijfheid E_c van de epitheelcel wordt voor iedere cel gelijk verondersteld. Merk op dat invullen van $\mathbf{r} = 0$ in (2.7) de uitdrukking geeft als in (2.6).

De energiebijdragen van alle n cellen kunnen nu bij elkaar worden opgeteld. Als met $M(\mathbf{r})$ de totale energiedichtheid op positie \mathbf{r} wordt genoteerd, dan geldt:

$$M(\mathbf{r}) = \sum_{j=1}^n M_j(\mathbf{r}) = \sum_{j=1}^n \left(\frac{F_j^2}{2\pi^2 E_s R_j^4} \cdot \exp\left(-\lambda \frac{\|\mathbf{r} - \mathbf{r}_j\|}{R_j}\right) \right). \quad (2.8)$$

Hierin dient nog een aanpassing te worden gemaakt. In [5] wordt aangetoond dat er een grens is waaronder cellen een mechanisch signaal niet meer kunnen waarnemen. Deze grens wordt genoteerd met $\bar{\epsilon}$. In (2.8) dient iedere $M_j(\mathbf{r})$ dus te worden vervangen door:

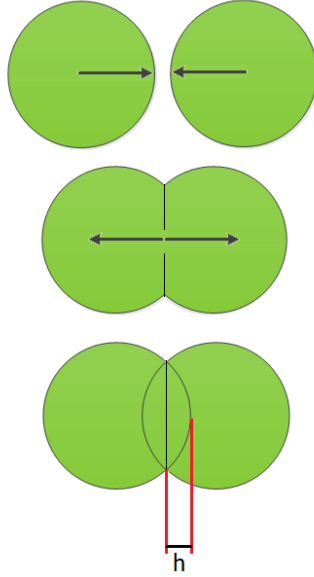
$$\begin{cases} M_j(\mathbf{r}), & \text{als } M_j(\mathbf{r}) \geq \bar{\epsilon} \\ 0, & \text{als } M_j(\mathbf{r}) < \bar{\epsilon} \end{cases}. \quad (2.9)$$

2.1.2 Passieve migratie

Als twee cellen elkaar indeuken zullen zij een kracht op elkaar gaan uitoefenen, zodat ze elkaar wegduwen. Voor het model is het makkelijker om te veronderstellen dat de cellen niet van vorm veranderen, zodat een cel in alle gevallen eenduidig kan worden beschreven met een middelpunt en een straal. Daarom wordt er in plaats van indeuking gewerkt met overlap. Hoe meer cellen elkaar overlappen, hoe groter de kracht die ze op elkaar uitoefenen. Figuur 2.2 maakt duidelijk wat er precies wordt bedoeld met ‘overlap’:

Er geldt dus:

$$h = \begin{cases} \frac{1}{2} \max(0, R_i + R_j - \|\mathbf{r}_i - \mathbf{r}_j\|), & \text{als } i \neq j \\ 0, & \text{als } i = j \end{cases}. \quad (2.10)$$



Figuur 2.2: Overlappende cellen.

In [10] wordt een uitdrukking afgeleid voor de bijdrage aan de potentiële energiedichtheid voor cel i , als functie van de overlap h met een andere cel. Voor de kracht Φ die op cel i wordt uitgeoefend, als functie van de overlap h met cel j , geldt:

$$\Phi(h) = \frac{4}{9} \sqrt{2h^3 E_c^2 R_i}. \quad (2.11)$$

Integreren van 0 tot h geeft de arbeid die is verricht om de overlap te bewerkstelligen:

$$\int_0^h \Phi(s) ds = \frac{8}{45} \sqrt{2h^5 E_c^2 R_i}. \quad (2.12)$$

De bijdrage aan de potentiële energiedichtheid voor cel i , als gevolg van de overlap met cel j , wordt genoteerd met M^{ij} . Deze is gelijk aan de verrichte arbeid gedeeld door het volume van cel i :

$$M^{ij} = \frac{2\sqrt{2}E_c}{15\pi} \left(\frac{h}{R_i} \right)^{5/2}. \quad (2.13)$$

Het rekenwerk tot nu toe bepaalt een vergelijking voor de totale energiedichtheid die geldt voor cel i . Deze totale energiedichtheid wordt genoteerd met $\hat{M}(\mathbf{r}_i)$:

$$\hat{M}(\mathbf{r}_i) = \sum_{j=1}^n |M_j(\mathbf{r}_i) - M^{ij}|. \quad (2.14)$$

Het minteken komt voort uit het feit dat iedere M^{ij} negatief werkt ten opzichte van iedere $M_j(\mathbf{r}_i)$. De absolutewaardestrepen waarborgen dat de energiedichtheid die hoort bij de interactie tussen cel i en cel j altijd positief is.

Vanaf nu zal de term aantrekkingskracht worden gebruikt voor de neiging die cellen hebben naar elkaar toe te bewegen als gevolg van de krachten die zij op het substraat uitoefenen. De term afstotingskracht zal worden gebruikt voor de kracht die op een cel wordt uitgeoefend als gevolg van overlapping van een andere cel.

2.1.3 Bewegingsrichting

De richting waarin een cel beweegt wordt nu gebaseerd op een gewogen gemiddelde van alle aantrekkings- en afstotingskrachten die de cel waarneemt. De gewichten voor dit gemiddelde worden bepaald door de grootte van iedere kracht. Beschouw een tijdstip t en noteer met \mathbf{v}_{ij} de eenheidsvector die cel i met cel j verbindt, dus:

$$\mathbf{v}_{ij} = \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|}. \quad (2.15)$$

Als $\|\mathbf{r}_j - \mathbf{r}_i\| = 0$ wordt $\mathbf{v}_{ij} = 0$ gesteld. De richting waarin cel i beweegt wordt nu dus gegeven door het hierboven beschreven gewogen gemiddelde:

$$\mathbf{z}_i = \sum_{j=1}^n ((M_j(\mathbf{r}_i) - M^{ij})\mathbf{v}_{ij}). \quad (2.16)$$

De vector \mathbf{z}_i wordt genormaliseerd, omdat bewegingssnelheid al in $\hat{M}(\mathbf{r}_i)$ is verwerkt:

$$\hat{\mathbf{z}}_i = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|}. \quad (2.17)$$

Als $\|\mathbf{z}_i\| = 0$, wordt $\hat{\mathbf{z}}_i = 0$ gesteld.

Het tot nu gedane rekenwerk induceert het volgende stelsel differentiaalvergelijkingen:

$$\frac{d\mathbf{r}_i}{dt} = \alpha_i \hat{M}(\mathbf{r}_i) \hat{\mathbf{z}}_i, \quad i = 1, \dots, n. \quad (2.18)$$

Hierbij is α_i een parameter die de mobiliteit van cel i indiceert. Deze wordt gelijk genomen aan die in [3]:

$$\alpha_i = \frac{\beta_i R_i^3}{\mu \hat{F}^2} F_i, \quad (2.19)$$

waarbij β_i staat voor de beweeglijkheid van het celoppervlak van cel i , en μ voor de frictiecoëfficiënt.

2.1.4 Verstoringen in mechanische signalen

Om rekening te houden met verstoringen in de mechanische signalen, wordt de beweging van de cellen als deels random gemodelleerd. In [3] leidt dit tot de vergelijkingen

$$\frac{d\mathbf{r}_i}{dt} = \alpha_i \hat{M}(\mathbf{r}_i) \hat{\mathbf{z}}_i + \sqrt{2D} \frac{d\mathbf{W}}{dt}, \quad i = 1, \dots, n. \quad (2.20)$$

Hierin is D een diffusieparameter, en $\mathbf{W} = (W_1(t), W_2(t))^T$ een wienerproces. In [4] wordt een formele definitie van een wienerproces gegeven. Voor dit verslag wordt alleen naar de belangrijkste eigenschappen van een Wienerproces gekeken: onafhankelijkheid over disjuncte intervallen, $W(0) = \mathbf{0}$, en $dW \sim \sqrt{dt}N(0, 1)$ op ieder interval dt .

Bij de implementatie van het model zullen de differentiaalvergelijkingen numeriek worden opgelost. Er wordt er dan voor gekozen dat geen enkele cel meer dan de helft van R_0 kan afleggen per tijdstap. In hoofdstuk 3 wordt gekozen voor Euler Voorwaarts als numerieke integratiemethode. Dit betekent dat, om te waarborgen dat $\|\Delta\mathbf{r}_i\| \leq R_0$, de volgende vergelijking voor Δt moet worden opgelost (met $N_1, N_2 \sim N(0, 1)$):

$$\|\Delta t \alpha_i \hat{M}(\mathbf{r}_i) \hat{\mathbf{z}}_i + \Delta t \sqrt{2D} \sqrt{\Delta t} (N_1, N_2)^T\| \leq R_0. \quad (2.21)$$

Dit zou veel extra rekenkracht vergen en is dus niet aantrekkelijk. De extra rekenkracht kan worden voorkomen door ervoor te kiezen de beweging op ieder moment óf deterministisch, óf stochastisch te laten zijn. Laat U_i voor $i \in \{1, \dots, n\}$ een uniform verdeelde stochast zijn op $[0, 1]$. Noteer met p_{rw} de kans dat een cel een willekeurige richting uitgaat. Definieer nu de functie

$$Y_i(U_i) = \begin{cases} 1, & \text{als } U_i \in [0, 1 - p_{rw}] \\ 0, & \text{als } U_i \in (1 - p_{rw}, 1] \end{cases}. \quad (2.22)$$

De vergelijkingen in (2.20) kunnen dan als volgt worden aangepast:

$$\frac{d\mathbf{r}_i}{dt} = Y_i \alpha_i \hat{M}(\mathbf{r}_i) \hat{\mathbf{z}}_i + (1 - Y_i) \sqrt{2D} \frac{d\mathbf{W}}{dt}, \quad i = 1, \dots, n. \quad (2.23)$$

2.1.5 Celdeling

In het basismodel in [1] is celdeling opgenomen, maar dit wordt behandeld als een instantane gebeurtenis. Realistischer is de aanpak die wordt gepresenteerd in [2], waarbij iedere cel een groei doormaakt alvorens te delen. Tijdens het groeien creëert de cel zo alvast een deel van de ruimte in het weefsel die nodig is voor twee cellen. Hier zal een vereenvoudigde en enigszins aangepaste versie van het delingsproces uit [2] worden behandeld.

De celdelingscyclus bestaat grofweg uit twee fasen, de interfase en de mitotische fase. Binnen de interfase kan nog tussen drie processen onderscheid worden gemaakt, en zo ontstaat de volgende cyclus:

1. **G1**. De eerste groeifase. Dit is de fase waarin een cel zich direct na deling bevindt. De straal van de cel groeit gedurende **G1** van R_0 tot R_S .
2. **S**. De rustfase. De cel wacht tot hij kan beginnen aan de tweede groeifase. Er vindt geen groei plaats.
3. **G2**. De tweede groeifase. De cel groeit van R_S tot R_M .
4. **M**. De mitotische fase. De cel verandert van één cel in twee identieke cellen.

In dit model wordt de tijdsduur van **G1**, **G2** en **M** deterministisch genomen. De duur van **S** wordt stochastisch gemodelleerd, om variatie te creëren tussen duren van delingscycli van cellen die uit dezelfde moedercel zijn voortgekomen. Met μ_D wordt de parameter genoteerd die de *verwachte* tijdsduur van een delingscyclus aangeeft. Met T_{G1} , T_{G2} en T_M wordt de tijdsduur van respectievelijk **G1**, **G2** en **M** genoteerd, en met T_S de (stochastische) tijdsduur van de **S**.

Er wordt aangenomen dat de cel tijdens **G1** en **G2** zal groeien volgens:

$$\frac{dR}{dt} = \frac{\xi}{R}. \quad (2.24)$$

Als met t_0 het moment wordt genoteerd dat een cel net heeft gedeeld, dan is de oplossing voor een cel in fase **G1**:

$$R((t - t_0)) = \sqrt{R(t_0)^2 + 2\xi(t - t_0)} \quad (2.25)$$

$$= \sqrt{R_0^2 + 2\xi(t - t_0)}. \quad (2.26)$$

Voor de oplossing voor een cel in fase **G2** moet worden opgemerkt dat tijdens **S** geen groei plaatsvindt. Voor een cel in fase **G2** geldt dus:

$$R((t - t_0)) = \sqrt{R_0^2 + 2\xi(t - t_0 - T_S)}. \quad (2.27)$$

In dit verslag wordt ervoor gekozen om een cel gedurende een hele delingscyclus te laten groeien tot een oppervlakte die twee keer zo groot is als die op t_0 . Zodoende moet $R(T_{G1} + T_S + T_{G2} + T_M) = \sqrt{2}R_0$. Omdat tijdens **S** en **M** geen groei plaatsvindt, volgt hieruit:

$$\xi = \frac{R_0^2}{2(T_{G1} + T_{G2})}. \quad (2.28)$$

Voor T_S wordt gekozen voor een exponentiële verdeling. Met μ_S wordt de verwachte tijdsduur van **S** genoteerd. Dan geldt dus:

$$P(T_S = t) = \frac{1}{\mu_S} \exp\left(-\frac{t}{\mu_S}\right). \quad (2.29)$$

Later zal deze verdeling worden benaderd met een geometrische, zodat de kans dat een cel gedurende dt van **S** naar **G2** gaat dan wordt gegeven door $p_{S \rightarrow G2} = dt/\mu_S$.

De laatste fase, **M**, zal hier worden behandeld als zijnde intstantaan, dus $T_M = 0$. Op basis van het werk in [2] wordt $T_{G1} = T_{G2} = 0.3\mu_D$ en $\mu_S = 0.4\mu_D$ genomen.

De richting van de splitsing van een cel wordt loodrecht ten opzichte van de bewegingsrichting verondersteld. De twee cellen die ontstaan worden geplaatst op afstand R_0 van elkaar, zodat ze elkaar nog een stukje moeten wegduwen om hun natuurlijke plaats in het weefsel te vinden.

2.1.6 Apoptose

Wat betreft apoptose wordt er niet naar specifieke oorzaken gekeken. Er wordt simpelweg verondersteld dat iedere cel na verloop van tijd tot apoptose overgaat. Op deze manier wordt ervoor gezorgd dat het aantal cellen in het weefsel niet explosief stijgt, en kan er door juiste parameterkeuze een balans worden bewerkstelligd. De levensduur van een cel wordt exponentieel verdeeld verondersteld, zodat de kans dat een cel gedurende een interval dt tot apoptose overgaat kan worden benaderd met dt/μ_a , met μ_a de verwachte tijd tussen het ontstaan van een cel en de intrede van apoptose. Als apoptose eenmaal zijn intrede heeft gedaan, zullen naburige cellen nog enige tijd de aanwezigheid van een gestorven cel blijven voelen, omdat er nog restmateriaal aanwezig is. Zeg dat dit restmateriaal na l seconden is opgeruimd, dan kan, op dezelfde manier als in [1], sterfte van cel i worden gemodelleerd door F_i als volgt aan te passen:

$$F_i = \begin{cases} \hat{F} \left(\frac{R_i}{R_0} \right)^2, & \text{als } t < t_a \\ \frac{\hat{F}}{l} \left(\frac{R_i}{R_0} \right)^2 (l - (t - t_a)), & \text{als } t_a \leq t < t_a + l \\ 0, & \text{als } t \geq t_a + l \end{cases} \quad (2.30)$$

Hierbij is t_a het moment van overlijden van de betreffende cel. Na het moment van overlijden neemt F_i dus lineair af totdat deze gelijk is aan 0.

2.2 Bacteriën

Bacteriën in een wond produceren melkzuur. Dit melkzuur verlaagt de pH-waarde in de wond, en dit zorgt ervoor dat epitheelcellen meer moeite hebben met bewegen. Ook zal de verwachtingswaarde van de duur van een delingscyclus hoger worden ten gevolge van een verhoogde pH-waarde. In deze paragraaf zal eerst worden gekeken naar de beweging van bacteriën, en daarna naar hoe de melkzuurconcentratie in een weefsel zich manifesteert. Vervolgens wordt de invloed ervan op celmobiliteit en duur van een delingscyclus beschreven.

2.2.1 Beweging en deling van bacteriën

Er wordt aangenomen dat bacteriën zo klein zijn, dat er geen sprake is van aantrekkings- en afstotingskrachten zoals bij de epitheelcellen. Ook wordt aangenomen dat de bacteriën, omdat ze zo klein zijn, vrij door het weefsel kunnen bewegen. Ze ondervinden dus geen hinder van epitheelcellen. De beweging van de bacteriën gedurende een interval dt wordt gemodelleerd als een verplaatsing ter grootte van UR_0dt , U een op $[0, 1]$ uniform verdeelde stochast, in willekeurige richting. In werkelijkheid zal de beweging complexer zijn. In [1] wordt toegelicht dat de invloed van de vereenvoudigde beweging op de melkzuurconcentratie zeer gering is.

Deling wordt voor bacteriën als instantane gebeurtenis behandeld. Gedurende dt is de kans dat een bacterie deelt gelijk aan p_bdt .

2.2.2 Melkzuurconcentratie

Noteer met n_{bac} het aantal bacteriën in het weefsel. De melkzuurconcentratie $c_k(\mathbf{r}, t)$ als gevolg van bacterie $k \in \{1, \dots, n_{\text{bac}}\}$, kan worden beschreven met de volgende partiële differentiaalvergelijking en beginvoorwaarde ([1]):

$$\begin{cases} \frac{\partial c_k}{\partial t} = D_b \cdot \nabla^2 c_k + \gamma_k^b(t) \delta(\mathbf{r} - \mathbf{r}_k(t)), & \mathbf{r} \in \mathbb{R}^2 \\ c_k(\mathbf{r}, 0) = 0, & \mathbf{r} \in \mathbb{R}^2 \end{cases} \quad (2.31)$$

Hierbij is $\mathbf{r}_k(t) \in \bar{\Omega}$ de positie van bacterie k op tijdstip t , en $\gamma_k^b(t)$ is een productieterm:

$$\gamma_k^b(t) = \begin{cases} \bar{\gamma}^b, & t \in [t_0^b, t_s^b) \\ 0 & t \in [0, t_0^b) \cup [t_s^b, \infty) \end{cases}, \quad (2.32)$$

met t_0^b het moment waarop de betreffende bacterie ontstaat, t_s^b het moment waarop een bacterie onschadelijk wordt gemaakt (door een witte bloedcel, zie paragraaf 2.3), en $\bar{\gamma}^b$ een parameter die de productie van melkzuur per kubieke micrometer per seconde door een levende bacterie aangeeft. D_b is de diffusiecoëfficiënt, en δ is de Diracdelta, zie [6]. Op basis van superpositie impliceert dit de volgende vergelijkingen voor de totale melkzuurconcentratie $c := c(\mathbf{r}, t) = \sum_{k=1}^{n_{\text{bac}}} c_k(\mathbf{r}, t)$:

$$\begin{cases} \frac{\partial c}{\partial t} = D_b \cdot \nabla^2 c + \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b(t) \delta(\mathbf{r} - \mathbf{r}_k(t)), & \mathbf{r} \in \mathbb{R}^2 \\ c(\mathbf{r}, 0) = 0, & \mathbf{r} \in \mathbb{R}^2 \end{cases} \quad (2.33)$$

In [1] wordt ook de oplossing gegeven:

$$c(\mathbf{r}, t) = \sum_{k=1}^{n_{\text{bac}}} c_k(\mathbf{r}, t) = \sum_{k=1}^{n_{\text{bac}}} \int_0^t \frac{\gamma_k^b(s)}{4\pi D_b \cdot (t-s)} \exp\left(-\frac{\|\mathbf{r} - \mathbf{r}_k(s)\|^2}{4D_b \cdot (t-s)}\right) ds. \quad (2.34)$$

2.2.3 Invloed op celmobiliteit

De melkzuurconcentratie c in een bepaald punt heeft invloed op de pH-waarde die daar heerst, en zodoende op de mobiliteit van een cel die zich daar bevindt. In [1] wordt op basis van [7] gekozen voor een exponentiële functie om de invloed van de melkzuurconcentratie op celmobiliteit te modelleren. De vergelijkingen in (2.23) moeten dan worden aangepast tot:

$$\frac{d\mathbf{r}_i}{dt} = \exp(-\nu_1 c) \left(Y_i \alpha_i \hat{M}(\mathbf{r}_i) \hat{\mathbf{z}}_i + (1 - Y_i) \sqrt{2D} \frac{d\mathbf{W}}{dt} \right), \quad i = 1, \dots, n. \quad (2.35)$$

Hierbij is $0 \leq \nu_1 \in \mathbb{R}$ een parameter die de gevoeligheid van de celmobiliteit voor de melkzuurconcentratie aangeeft.

2.2.4 Invloed op de duur van een delingscyclus

Om de invloed van de melkzuurconcentratie op de duur van een delingscyclus te modelleren moeten de drie fases **G1**, **G2** en **S** worden bekeken (aangezien **M** als

instantane gebeurtenis wordt behandeld). Voor **G1** en **G2** wordt aangenomen dat een cel gedurende dt ten gevolge van de melkzuurconcentratie nog maar het werk voor de celgroei kan verrichten dat hij normaal in $\exp(-\nu_2 c)dt$ zou hebben verricht. Dit geeft aanleiding tot het aanpassen van vergelijkingen (2.25)/(2.26) en (2.27) tot respectievelijk:

$$R((t - t_0)) = \sqrt{R_0^2 + 2\xi \int_{t_0}^t \exp(-\nu_2 \cdot c(\mathbf{r}(s), s)) ds} \quad (\mathbf{G1}\text{-fase}) \quad (2.36)$$

$$R((t - t_0)) = \sqrt{R_0^2 + 2\xi \left(\int_{\mathcal{I}} \exp(-\nu_2 \cdot c(\mathbf{r}(s), s)) ds - T_s \right)} \quad (\mathbf{G2}\text{-fase}), \quad (2.37)$$

waarbij $\mathcal{I} = [t_0, t_{G1 \rightarrow S}) \cup [t_{S \rightarrow G2}, t)$, voor $t_{G1 \rightarrow S}$ het tijdstip waarop de betreffende cel van fase **G1** naar **S** gaat, en $t_{S \rightarrow G2}$ het tijdstip waarop de cel van fase **S** naar **G2** gaat. $\mathbf{r}(s) \in \bar{\Omega}$ is de positie van de betreffende cel op tijdstip s .

Voor de kans dat een cel gedurende dt van fase **S** naar **G2** gaat kan een aanpassing worden gedaan die ongeveer analoog is aan die voor de celmobiliteit: $p_{S \rightarrow G2} = \exp(-\nu_3 c)dt/\mu_S$.

Er dient te worden opgemerkt dat de eerder gespecificeerde T_{G1} , T_{G2} en μ_S nu niet meer de (verwachte) tijdsduren zijn van, respectievelijk, **G1**, **G2** en **S**; het zijn de tijdsduren die gelden als een cel gedurende zijn gehele delingscyclus een melkzuurconcentratie gelijk aan 0 ondervindt.

2.3 Witte bloedcellen

Als zich in een wond bacteriën bevinden, zullen witte bloedcellen proberen deze te gaan doden. Daarnaast ruimen ze melkzuur op. Zodoende spelen ze een belangrijke rol in een wondhelingsproces. In deze paragraaf wordt eerst in grote lijnen het werk aangaande witte bloedcellen uit [1] besproken: er wordt beschreven hoe een bloedvatstelsel dat door de wond loopt wordt gemodelleerd (angiogenese); vervolgens wordt toegelicht hoe de hoeveelheid witte bloedcellen in een weefsel afhangt van de melkzuurconcentratie en het aantal bacteriën; daarna wordt de beweging van witte bloedcellen door een weefsel uitgelegd. Tot slot wordt een nieuw modelement behandeld: witte bloedcellen worden in staat gesteld melkzuur op te ruimen.

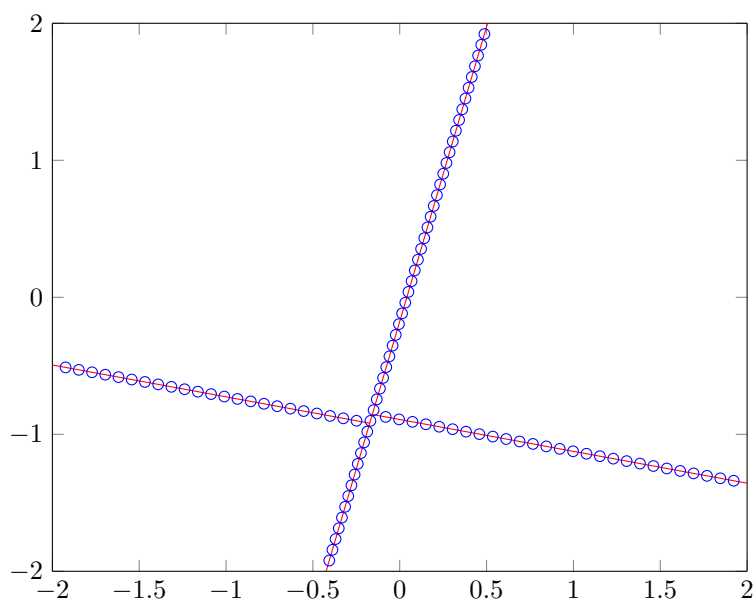
2.3.1 Constructie bloedvatstelsel

De constructie van een eenvoudig bloedvatstelsel geschiedt hier, zoals in [1], stochastisch, op basis van de volgende routine:

- Er wordt een ‘hoofdvat’ geplaatst, een rechte lijn die de y -as doorsnijdt tussen $(0, -0.5)$ en $(0, 0.5)$ om te waarborgen dat dit vat altijd door de wond heen loopt (bij de simulaties zullen steeds wonden worden gebruikt die tenminste $[-0.5, 0.5] \times [-0.5, 0.5] \subseteq \bar{\Omega}$ overdekken). De richting van het hoofdvat wordt stochastisch genomen door een hoek θ uit een uniforme verdeling op $[0, \pi)$ te trekken.

- Er worden twee vertakkingen aan het hoofdvat geplaatst. Deze vertakkingen staan loodrecht op het hoofdvat, en wijzen ten opzichte van elkaar in tegengestelde richting. De punten waar ieder van de vertakkingen het hoofdvat raakt worden wederom op basis van een uniforme verdeling gekozen.
- Over het nu ontstane bloedvatenstelsel worden evenredig zogenoemde ‘creatiepunten’ verdeeld: punten van waaruit witte bloedcellen in staat worden gesteld uit de bloedvaten te treden.

Figuur 2.3 toont een met bovenstaande routine geconstrueerd bloedvatenstelsel, met honderd creatiepunten:



Figuur 2.3: Een eenvoudig bloedvatenstelsel met honderd creatiepunten.

2.3.2 Creatie en sterfte

De kans p_{wb} dat gedurende een interval dt uit een bepaald creatiepunt een witte bloedcel het weefsel binnentreedt wordt als volgt gekozen:

$$p_{wb} = \frac{(1 - \exp(-\zeta c))dt}{\#(\text{creatiepunten})}. \quad (2.38)$$

Hierbij is $0 \leq \zeta \in \mathbb{R}$ een parameter die aangeeft hoe graag witte bloedcellen willen uittreden. De term $(1 - \exp(-\zeta c))$ laat zien dat p_{wb} afhankelijk is van de melkzuurconcentratie die in het creatiepunt geldt: als $c = 0$, is $p_{wb} = 0$, zodat er zoals gewenst geen witte bloedcellen uittreden; als c heel groot is, geldt $(1 - \exp(-\zeta c)) \approx 1$, zodat er zoals gewenst heel veel witte bloedcellen uittreden. Het is niet de bedoeling dat het aantal witte bloedcellen dat gedurende dt het weefsel binnentreedt afhangt van de hoeveelheid creatiepunten, $\#(\text{creatiepunten})$. Daarom wordt er door deze term gedeeld.

De witte bloedcellen zullen, nadat ze zijn uitgetreden, proberen de bacteriën te neutraliseren. In het model wordt verondersteld dat de witte bloedcellen, zodra ze vijf witte bloedcellen onschadelijk hebben gemaakt, doodgaan. De sterfte van witte bloedcellen wordt als instantaan gemodelleerd, dus er is geen sprake van restmateriaal.

2.3.3 Beweging van witte bloedcellen

Witte bloedcellen migreren deels op dezelfde manier als epitheelcellen. Praktisch betekent dat dat zij deelnemen aan de interactie die de basis vormt voor de beweging van epitheelcellen. Daarnaast zullen de witte bloedcellen actief op zoek gaan naar bacteriën, waarbij ze hun beweging baseren op de gradiënt van de melkzuurconcentratie. Voor de witte bloedcellen worden random bewegingen buiten beschouwing gelaten. Noteer met n_{wb} het aantal witte bloedcellen in het weefsel op een zeker moment. Dan gelden de volgende vergelijkingen:

$$\frac{d\mathbf{r}_j}{dt} = \alpha_j(\hat{M}(\mathbf{r}_j)\hat{\mathbf{z}}_j + v_{wb}\nabla c), \quad j = 1, \dots, n_{wb}. \quad (2.39)$$

Hierbij is dus de celmobiliteitsparameter α_j gelijk gedefinieerd aan die voor epitheelcellen. De parameters E_c en \hat{F} worden ook gebruikt voor de witte bloedcellen. Voor de celstraal wordt R_0 gebruikt. v_{wb} is een parameter die aangeeft hoe de snelheid van een witte bloedcel zich verhoudt tot de melkzuurconcentratie. Er moet worden opgemerkt dat vanaf nu voor de termen \hat{M} en $\hat{\mathbf{z}}$ steeds moet worden gekeken naar zowel epitheelcellen als witte bloedcellen (ook voor de beweging van epitheelcellen).

Als een witte bloedcel zich dicht in de buurt van een bacterie bevindt, kan ∇c zeer grote waarden aanemen (als gevolg van de Diracdelta in vergelijking (2.33)). Dit zou onevenredig grote bewegingen van witte bloedcellen in kunnen leiden. Daarom wordt een maximale waarde $(\nabla c)_{\max}$ voor ∇c ingevoerd.

2.3.4 Melkzuurafbraak

Witte bloedcellen worden in staat gesteld melkzuur te kunnen opruimen, zodat de melkzuurconcentratie niet meer alleen kan dalen als gevolg van diffusie. Om dit te modelleren moet de volgende aanpassing worden gedaan aan de vergelijkingen in (2.33):

$$\begin{cases} \frac{\partial c}{\partial t} = D_b \cdot \nabla^2 c + \sum_{k=1}^{n_{bac}} \gamma_k^b(t)\delta(\mathbf{r} - \mathbf{r}_k(t)) - \sum_{j=1}^{n_{wb}} \gamma_j^{wb}(t)\delta(\mathbf{r} - \mathbf{r}_j(t)), & \mathbf{r} \in \mathbb{R}^2 \\ c(\mathbf{r}, 0) = 0, & \mathbf{r} \in \mathbb{R}^2 \end{cases} \quad (2.40)$$

Hierbij is $\gamma_j^{wb}(t)$ een parameter die aangeeft hoeveel melkzuur witte bloedcel j opruimt per seconde. Voor een levende witte bloedcel is deze parameter gelijk aan $\bar{\gamma}^{wb}$, voor een dode gelijk aan 0.

2.4 Parameters

Voor het grootste deel van de parameters kan een beroep op [1] worden gedaan. Voor het overige zal een korte toelichting worden gegeven.

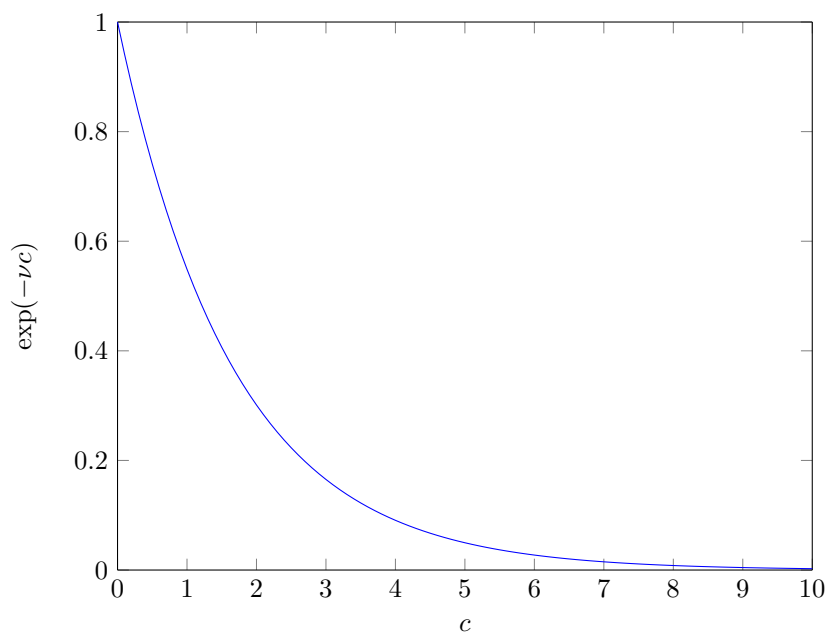
De drempel $\bar{\epsilon}$ waaronder cellen mechanische signalen niet meer kunnen voelen is in [2] berekend, voor een substraat vergelijkbaar met dat in dit project. Deze is zo klein, namelijk $\bar{\epsilon} \approx 2.7 \cdot 10^{-31}$, dat deze voor dit project nauwelijks invloed zal hebben. Daarom wordt $\bar{\epsilon} = 0$ genomen.

De in (2.19) gespecificeerde α_i hangt af van R_i . Omdat een cel kan groeien tot een straal van $\sqrt{2}R_0$ volgt uit de definitie van α_i dat een cel vlak voor deling ongeveer $4\sqrt{2}$ keer zo snel beweegt als een cel vlak na deling. Dit lijkt niet erg realistisch en daarom zal hier een constante $\alpha = 0.006$ worden genomen. Deze α is een stuk kleiner dan α_i geëvalueerd in $R_i = R_0$, maar hiervoor kan worden gecompenseerd door de tijdstap groter te nemen. Bijkomend voordeel is dat simulaties hierdoor sneller kunnen worden gedaan.

Een epitheelcel doet in werkelijkheid ongeveer een dag over één delingscyclus. Dat is te lang voor dit model, en daarom zal de duur van een delingscyclus worden geschaald. In plaats van een dag wordt vijf minuten gebruikt. Dit kan worden gedaan, op voorwaarde dat er goed in de gaten gehouden wordt dat cellen de tijd krijgen om uit elkaar te bewegen, en niet opeenhopen.

In [1] wordt een aantal simulaties gedaan met v_{wb} -waarden tussen 0.002 en 0.5, waarna wordt gekozen voor $v_{wb} = 0.002$. Dit omdat witte bloedcellen anders in sommige gevallen een te grote afstand per tijdstap afleggen, zodat ze als het ware over andere cellen heenspringen. Voor dit project zal wel met $v_{wb} = 0.5$ worden gewerkt, omdat er wordt verondersteld dat witte bloedcellen zich vrij gemakkelijk tussen de epitheelcellen door kunnen manoeuvreren.

Voor de parameters D , p_{rw} , $\bar{\gamma}^{wb}$, ν_2 en ν_3 is het lastig in de literatuur waarden te vinden. Voor ν_2 en ν_3 wordt verondersteld dat de invloed van melkzuur op de duur van een delingscyclus ongeveer hetzelfde is als die op de celmobiliteit. Daarom wordt $\nu_3 = \nu_2 = \nu_1 = \nu = 0.6$ gekozen. Figuur 2.4 toont de grafiek van $\exp(-\nu c)$ uitgezet tegen c . Verder wordt $D = 10^{-6}$, $p_{rw} = 0.1$, en $\bar{\gamma}^{wb}$ gekozen, om de simpele reden dat dit aannemelijke resultaten oplevert.



Figuur 2.4: Invloed ν op celmobilititeit en duur delingscyclus

In de onderstaande tabel staan de voor de simulaties gebruikte parameters.

Parameter	Waarde	Eenheid
E_s	5.0	kPa
E_c	2.0	kPa
R_0	0.04	μm
α	0.006	$\text{m}^3\text{N}^{-1}\text{s}^{-1}$
\hat{F}	0.01	nN
D	10^{-6}	mm^2s^{-1}
p_{rw}	0.1	-
μ_{D}	300	s
μ_{S}	120	s
l	10	s
D_b	0.01	$\mu\text{m}^2\text{s}^{-1}$
$\bar{\gamma}^b$	0.1	$\text{mol} \cdot \text{m}^{-3}\text{s}^{-1}$
$\bar{\gamma}^{\text{WB}}$	0.09	$\text{mol} \cdot \text{m}^{-3}\text{s}^{-1}$
ζ	0.1	-
$(\nabla c)_{\text{max}}$	30	-
p_s	0.001	-

Hoofdstuk 3

Numerieke analyse

Om het in hoofdstuk 2 beschreven model te kunnen gebruiken voor simulaties, zullen de beschreven bewegingsvergelijkingen numeriek moeten worden geïntegreerd. In dit hoofdstuk zal worden beschreven welke keuzes er wat dat betreft zijn gemaakt. In [1] bleken berekeningen omtrent de melkzuurconcentratie in veel gevallen zeer veel rekenkracht te behoeven, in sommige gevallen zelfs zo veel dat bepaalde simulaties niet konden worden uitgevoerd. In dit hoofdstuk zal daarom worden gepoogd de melkzuurconcentratie te benaderen, om simulaties zo te versnellen.

De verschillende methodes voor numerieke integratie die in dit hoofdstuk worden besproken zijn te vinden in [8]. Ook voor toelichting aangaande het symbool \mathcal{O} , dat wordt gebruikt voor uitspraken over benaderingsfouten, kan op [8] een beroep worden gedaan.

3.1 Tijdsintegratie

3.1.1 Keuze integratiemethode

In [1] is voor de tijdsintegratie van de bewegingsvergelijkingen Modified Euler gebruikt. Deze methode biedt een $\mathcal{O}(\Delta t^2)$ -fout (met Δt de tijdstap), maar leverde een probleem op. Als twee cellen elkaar overlappen, vallen op een bepaald punt de predictor en corrector die bij Modified Euler horen, tegen elkaar weg. Dit leidt een evenwicht in waarvan in werkelijkheid geen sprake is. In dit verslag wordt daarom Euler Voorwaarts gebruikt. Deze methode wordt vaak niet gebruikt vanwege de slechte stabiliteitseigenschappen, en de $\mathcal{O}(\Delta t)$ -fout, maar simulaties laten zien dat deze hier zonder problemen kan worden toegepast. Euler Voorwaarts levert voor de beweging van epitheelcellen de volgende vergelijkingen op:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t Y_i \alpha_i \hat{M}(\mathbf{r}_i(t)) \hat{\mathbf{z}}_i(t) + (1 - Y_i) \sqrt{2D\Delta t} (N_1, N_2)^T. \quad (3.1)$$

Hierbij geldt $i \in \{1, \dots, n\}$ en $N_1, N_2 \sim N(0, 1)$. Voor de beweging van bacteriën en witte bloedcellen kan Euler Voorwaarts op analoge wijze worden toegepast.

In paragraaf 4.1 wordt de juistheid van het gebruik van Euler Voorwaarts gemotiveerd.

3.1.2 Grootte tijdstap

Zodra een cel een andere overlapt, is er sprake van een afstotingskracht. Als de overlap groot genoeg is, zal de afstotingskracht zodanig zijn dat de cel wordt teruggeduwd. Als Δt te groot is, kan een cel in een tijdstap zo veel afstand afleggen, dat hij als het ware over een andere cel heenspringt, zonder een afstotingskracht te ondervinden. Dit is uiteraard niet de bedoeling, en daarom moet op ieder tijdstip worden bekeken welke tijdstap geschikt is. In [1] wordt een keuze voor Δt gepresenteerd die bovengenoemd probleem voorkomt. Als Δt wordt gekozen dat iedere cel hoogstens de helft van zijn straal af kan leggen, kan een cel nooit over een andere heenspringen. Nummer eerst de epitheelcellen van 1 tot en met n , en vervolgens de witte bloedcellen van $n + 1$ tot en met $n + n_{wb}$. Dan:

$$(\Delta t)_t = \min_{i \in \{1, \dots, n + n_{wb}\}} \left(\frac{R_i}{2 \cdot |v_i|} \right), \quad (3.2)$$

waarbij $(\Delta t)_t$ de te gebruiken tijdstap op tijdstip t is, en $|v_i|$ is de snelheid van cel i op tijdstip t .

Voor de simulaties zal iedere tijdstap in eerste instantie $\Delta t = 1$ worden gekozen. Vervolgens wordt bekeken of de tijdstap moet worden aangepast naar $(\Delta t)_t$.

3.2 Benadering melkzuurconcentratie

Voor de melkzuurconcentratie is in vergelijking (2.34) een analytische oplossing gegeven. Deze oplossing is een som van n_{bac} integralen, die numeriek moeten worden benaderd. De variabele t is steeds een bovengrens voor het interval waarover moet worden geïntegreerd, maar komt ook voor in het integrand. Dit heeft tot gevolg dat voor vaste \mathbf{r} en t de waarde $c(\mathbf{r}, t)$ niet kan worden gebruikt om $c(\mathbf{r}, t + \Delta t)$ te berekenen, zoals dat wel het geval zou zijn geweest als t niet in het integrand voorkwam. Als voor numerieke integratie bijvoorbeeld de trapeziumregel (zie [8]) wordt gebruikt, neemt de benodigde rekenkracht voor het bepalen van $c(\mathbf{r}, t)$ zodoende lineair toe met t . Voor sommige simulaties wordt zeker tot en met $t = 500$ gerekend en zijn $n = 1000$, $n_{bac} = 20$ redelijke aantallen. Als de trapeziumregel dan wordt toegepast met $\Delta t = 1$, moeten 10^7 waarden van het integrand worden bepaald.

Om de melkzuurconcentratie anders te benaderen, en zo de benodigde rekenkracht te reduceren, wordt eerst gekeken naar een quasi-stationaire oplossing. Deze blijkt niet geschikt, en daarom wordt uiteindelijk de Finite Volume Method ([9]) toegepast.

3.2.1 Quasi-stationaire oplossing

Een quasi-stationaire oplossing is er één die zich instantaan aanpast aan de verandering van externe factoren ([6]). In de context van de vergelijkingen in (2.33) betekent dit dat de oplossing slechts verandert als een bacterie verplaatst, of als n_{bac} kleiner of groter wordt. Hier zal de quasi-stationaire oplossing voor het geval van één bacterie worden bepaald. Dit gaat ongeveer analoog aan het bepalen van de Greense functie in poolcoördinaten in \mathbb{R}^2 zoals in [6]. Op basis van superpositie kan door te sommeren de oplossing voor een groter aantal bacteriën worden gevonden. Door $\partial c_k / \partial t = 0$ te stellen, verkrijgt men de vergelijking die de quasi-stationaire oplossing c_k^q bepaalt:

$$D_b \cdot \nabla^2 c_k^q + \gamma_k^b(t) \delta(\mathbf{r} - \mathbf{r}_k(t)) = 0, \quad \mathbf{r} \in \mathbb{R}^2. \quad (3.3)$$

In poolcoördinaten luidt deze vergelijking, als $\rho = \|\mathbf{r} - \mathbf{r}_k(t)\|$ wordt genoemd:

$$\frac{1}{\rho} \frac{d}{d\rho} \left(\rho \frac{dc_k^q}{d\rho} \right) = -\gamma_k^b(t) \delta(\rho), \quad \mathbf{r} \in \mathbb{R}^2. \quad (3.4)$$

Als nu $\rho > 0$, dan geldt:

$$\frac{1}{\rho} \frac{d}{d\rho} \left(\rho \frac{dc_k^q}{d\rho} \right) = 0, \quad \mathbf{r} \in \mathbb{R}^2. \quad (3.5)$$

Tweemaal integreren levert:

$$c_k^q(\rho, t) = a_1 \ln \rho + a_2, \quad (3.6)$$

voor $a_1, a_2 \in \mathbb{R}$ constant. Als (3.4) over een cirkel \mathcal{C} met middelpunt $\mathbf{r}_k(t)$ en kleine straal wordt geïntegreerd geldt:

$$\iint_{\mathcal{C}} \nabla^2 c_k^q d\Omega = -\gamma_k^b(t). \quad (3.7)$$

Als nu de divergentiestelling wordt toegepast volgt (omdat, voor \mathbf{n} de naar buiten gerichte eenheidsnormaal, $\mathbf{n} \cdot \nabla c = \partial c / \partial \rho$):

$$\oint_{\partial \mathcal{C}} \frac{\partial c_k^q}{\partial \rho} d\Gamma = -\gamma_k^b(t), \quad (3.8)$$

zodat $2\pi\rho \partial c_k^q / \partial \rho = -\gamma_k^b(t)$, omdat $\partial c_k^q / \partial \rho$ constant is op \mathcal{C} . Omdat dit nog steeds geldt in de limiet $\rho \rightarrow 0$, volgt nu dat $a_1 = -\gamma_k^b(t) / (2\pi)$. De gevonden quasi-stationaire oplossing luidt nu:

$$c_k^q(\mathbf{r}, t) = -\frac{\gamma_k^b(t)}{2\pi} \ln \|\mathbf{r} - \mathbf{r}_k(t)\| + a_2, \quad a_2 \in \mathbb{R}. \quad (3.9)$$

Meteen wordt duidelijk dat deze oplossing niet geschikt kan zijn. Ongeacht de keuze voor a_2 , altijd zullen $\mathbf{r} \in \mathbb{R}^2$ bestaan zodat $c_k^q(\mathbf{r}, t) < 0$. Sterker nog: $|c_k^q(\mathbf{r}, t) - c_k(\mathbf{r}, t)| \rightarrow \infty$ als $\|\mathbf{r}\| \rightarrow \infty$.

3.2.2 Finite Volume Method

In [9] wordt de Finite Volume Method uitgelegd. In deze subparagraaf zal deze worden toegepast op de vergelijkingen in (2.33). Uiteindelijk zal op ieder tijdstip een waarde voor de melkzuurconcentratie beschikbaar zijn in m^2 roosterpunten. Bij deze roosterpunten horen controlevolumes (die in de volgende paragraaf nader zullen worden toegelicht). Voor iedere epitheelcel zal worden bekeken in welk controlevolume hij ligt, en de waarde van het bijbehorende roosterpunt zal vervolgens aan de cel worden toegekend.

Semi-discretisatie

Voor alle simulaties in dit verslag is het voldoende om benaderingen van de melkzuurconcentratie te kunnen bepalen op $[-2, 2] \times [-2, 2] \subseteq \mathbb{R}^2$. Daarom wordt een rooster gebruikt dat is gebaseerd op dit domein. Zowel in de x - als de y -richting wordt gebruikgemaakt van m roosterpunten. De horizontale en verticale afstand tussen twee naast elkaar liggende roosterpunten is dan gelijk aan $\hat{h} = 1/(m-1)$.

Elk roosterpunt ligt in een controlevolume V . Bekijk een inwendig punt (dus een punt dat niet op $\partial([-2, 2] \times [-2, 2])$ ligt) en integreer de partiële differentiaalvergelijking van het beginwaardeprobleem in (2.33) over V :

$$\iint_V \frac{\partial c}{\partial t} d\Omega = \iint_V D_b \cdot \nabla^2 c d\Omega + \iint_V \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b(t) \delta(\mathbf{r} - \mathbf{r}_k(t)) d\Omega. \quad (3.10)$$

Nu wordt de divergentiestelling toegepast, $\partial/\partial t$ wordt buiten de eerste integraal gehaald, en de lineariteit van de integraal kan worden gebruikt zodat:

$$\frac{d}{dt} \iint_V c d\Omega = D_b \oint_{\nabla V} \mathbf{n} \cdot \nabla c d\Gamma + \sum_{k=1}^{n_{\text{bac}}} \iint_V \gamma_k^b(t) \delta(\mathbf{r} - \mathbf{r}_k(t)) d\Omega. \quad (3.11)$$

Hierbij is \mathbf{n} de naar buiten gerichte eenheidsnormaal. Merk nu op dat:

$$\iint_V \gamma_k^b(t) \delta(\mathbf{r} - \mathbf{r}_k(t)) d\Omega = \begin{cases} \gamma_k^b(t) & \text{als } \mathbf{r}_k(t) \in V \\ 0 & \text{als } \mathbf{r}_k(t) \notin V \end{cases}. \quad (3.12)$$

Noteer nu

$$I_{k,V}(t) = \begin{cases} 1 & \text{als } \mathbf{r}_k(t) \in V \\ 0 & \text{als } \mathbf{r}_k(t) \notin V \end{cases}, \quad (3.13)$$

dan geldt dus:

$$\frac{d}{dt} \iint_V c d\Omega = D_b \oint_{\partial V} \mathbf{n} \cdot \nabla c d\Gamma + \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b(t) I_{k,V}(t). \quad (3.14)$$

Voor $\mathbf{n} \cdot \nabla c$ worden nu voorwaartse differenties gebruikt. Dit wordt gedaan volgens het N,E,S,W,P -schema: het punt waarvoor ∇c wordt berekend wordt dan aangeduid met een P , het punt erboven met een N , het punt rechts ernaast

met een E , het punt eronder met een S , en het punt links ernaast met een W .

Verder worden de twee integralen met eenpuntsintegraties benaderd. Dit levert de volgende vergelijking voor de Finite Volume Method-benadering in roosterpunt P , die wordt genoteerd met $C_P = C_P(t)$:

$$\hat{h}^2 \frac{dC_P}{dt} = D_b(C_N + C_E + C_S + C_W - 4C_P) + \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,V}. \quad (3.15)$$

Dit is nog niet voldoende: er moet nog een uitdrukking voor de punten op $\partial([-2, 2] \times [-2, 2])$ worden bepaald. Omdat het beginwaardeprobleem in (2.33) voor een oneindig domein geldt, is er niet direct een randvoorwaarde beschikbaar. De randvoorwaarde $c(\mathbf{r}, t) = 0$ is niet voor de hand liggend, omdat dit zou corresponderen met een melkzuurconcentratie gelijk aan 0 buiten $[-2, 2] \times [-2, 2]$. Ook $\mathbf{n} \cdot \nabla c(\mathbf{r}, t) = 0$ lijkt niet logisch, want dit zou betekenen dat er geen uitstroom van melkzuurconcentratie plaatsvindt. Er wordt daarom gekozen voor $\mathbf{n} \cdot \nabla c(\mathbf{r}, t) = -\phi c(\mathbf{r}, t)$ op $\partial([-2, 2] \times [-2, 2])$, voor $0 \leq \phi \in \mathbb{R}$ een nader te bepalen constante. Informeel zegt deze randvoorwaarde: ‘‘Hoe groter c op de rand, hoe groter de uitstroom’’.

De controlevolumes in de hoeken van het domein hebben oppervlakte $(\hat{h}/2) \cdot (\hat{h}/2) = \hat{h}^2/4$. De controlevolumes voor de overige randpunten hebben oppervlakte $\hat{h} \cdot (\hat{h}/2) = \hat{h}^2/2$. Daarom is voor de vergelijkingen voor die punten, naast het implementeren van de randvoorwaarde, nog een extra aanpassing nodig. Er zal hier een vergelijking worden gegeven voor het hoekpunt $(-2, 2)$, en een voor een punt op de lijn $\{(x, y) \in \mathbb{R}^2 : -2 < x < 2, y = -2\}$. Voor de rest van de randpunten kan analoog een vergelijking worden bepaald. Voor $(-2, 2)$:

$$\frac{\hat{h}^2}{4} \frac{dC_P}{dt} = D_b \left(\frac{1}{2} C_N + \frac{1}{2} C_E - C_P - \hat{h} \phi C_P \right) + \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,V}. \quad (3.16)$$

Voor een punt op de lijn $\{(x, y) \in \mathbb{R}^2 : -2 < x < 2, y = -2\}$:

$$\frac{\hat{h}^2}{2} \frac{dC_P}{dt} = D_b \left(C_N + \frac{1}{2} C_E + \frac{1}{2} C_W - 2C_P - \hat{h} \phi C_P \right) + \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,V}. \quad (3.17)$$

De vergelijkingen voor alle punten tezamen induceren het volgende stelsel differentiaalvergelijkingen:

$$\frac{d\mathbf{C}}{dt} = \mathbf{A}\mathbf{C} + \mathbf{f}. \quad (3.18)$$

Herbij is $\mathbf{C} = \mathbf{C}(t) = (C_1(t), \dots, C_{m^2}(t))^T$, gebaseerd op een horizontale nummering. Voor de matrix \mathbf{A} geldt:

$$\mathbf{A} = \frac{D_b}{\hat{h}^2} \begin{pmatrix} B_1 & 2I & & & & \\ I & B_2 & I & & & \\ & \ddots & \ddots & \ddots & & \\ & & I & B_{m-1} & I & \\ & & & 2I & B_m & \end{pmatrix}, \quad (3.19)$$

waarbij I de $m \times m$ -identiteitsmatrix is,

$$B_1 = B_m = \begin{pmatrix} b_1 & 2 & & & \\ & 1 & b_2 & & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & b_2 & 1 \\ & & & & 2 & b_1 \end{pmatrix}, \quad (3.20)$$

met $b_1 = -4 - 4\hat{h}\phi$ en $b_2 = -4 - 2\hat{h}\phi$, en

$$B_2 = \dots = B_{m-1} = \begin{pmatrix} b_2 & 2 & & & \\ & 1 & -4 & & 1 \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -4 & 1 \\ & & & & 2 & b_2 \end{pmatrix}. \quad (3.21)$$

Voor de vector \mathbf{f} geldt:

$$\mathbf{f} = \frac{1}{\hat{h}^2} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{m-1} \\ f_m \end{pmatrix}, \quad (3.22)$$

waarbij

$$f_1 = \begin{pmatrix} 4 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,1} \\ 2 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,2} \\ \vdots \\ 2 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,(m-1)} \\ 4 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,m} \end{pmatrix}, \quad f_m = \begin{pmatrix} 4 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,((m-1)m+1)} \\ 2 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,((m-1)m+2)} \\ \vdots \\ 2 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,(m^2-1)} \\ 4 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,m^2} \end{pmatrix} \quad (3.23)$$

en

$$f_j = \begin{pmatrix} 2 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,((j-1)m+1)} \\ \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,((j-1)m+2)} \\ \vdots \\ \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,(jm-1)} \\ 2 \sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,jm} \end{pmatrix}, \quad (j = 2, \dots, (m-1)). \quad (3.24)$$

Met $I_{k,\bar{\alpha}}$ wordt in het bovenstaande steeds de indicatorfunctie bedoeld die aangeeft of bacterie k zich bevindt in het controlevolume dat hoort bij het roosterpunt met nummer $\bar{\alpha}$.

Witte bloedcellen

Het opruimen van melkzuur door witte bloedcellen kan nu heel gemakkelijk worden gemodelleerd door \mathbf{f} in (3.22) aan te passen: verander steeds $\sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,\bar{\alpha}}$ in

$$\sum_{k=1}^{n_{\text{bac}}} \gamma_k^b I_{k,\bar{\alpha}} - \sum_{j=1}^{n_{\text{wb}}} \gamma_j^{\text{wb}} I_{j,\bar{\alpha}}, \quad (3.25)$$

met $I_{j,\bar{\alpha}}$ de indicatorfunctie die aangeeft of witte bloedcel j zich in het controlevolume met nummer $\bar{\alpha}$ bevindt.

Tijdsintegratie

Omdat bij de Matlab-implementatie niet met een vaste tijdstap zal worden gewerkt, is het aantrekkelijk om een tijdsintegratiemethode te kiezen die onafhankelijk van Δt stabiel is. Euler Achterwaarts voldoet voor vergelijking (3.18) aan dit criterium als kan worden gewaarborgd dat de eigenwaarden van A negatief zijn. Om dit na te gaan, kan de stelling van Gershgorin worden toegepast ([8]). Deze zegt dat de eigenwaarden van A in de vereniging van de volgende cirkels liggen:

$$\left\{ z \in \mathbb{C} : \frac{D_b}{\hat{h}^2} |z - (-4 - \hat{h}\phi)| \leq 4 \frac{D_b}{\hat{h}^2} \right\}; \quad (3.26)$$

$$\left\{ z \in \mathbb{C} : \frac{D_b}{\hat{h}^2} |z - (-4 - 2\hat{h}\phi)| \leq 4 \frac{D_b}{\hat{h}^2} \right\}; \quad (3.27)$$

$$\left\{ z \in \mathbb{C} : \frac{D_b}{\hat{h}^2} |z - (-4)| \leq 4 \frac{D_b}{\hat{h}^2} \right\}. \quad (3.28)$$

Hieruit volgt dat het reële deel van de eigenwaarden in ieder geval niet positief is. Echter blijft wel een eigenwaarde gelijk aan 0 mogelijk. Daarom zal uiteindelijk een vergelijking tussen de Finite Volume Method-benadering en de analytische oplossing uitsluitel moeten bieden. Het voordeel van Euler Achterwaarts blijft wel staan: bij andere integratiemethoden zijn er altijd Δt die met zekerheid instabiliteit induceren. Het schema wordt nu:

$$\mathbf{C}(t + \Delta t) = (I - \Delta t A)^{-1} \mathbf{C}(t) + \Delta t \mathbf{f}(t). \quad (3.29)$$

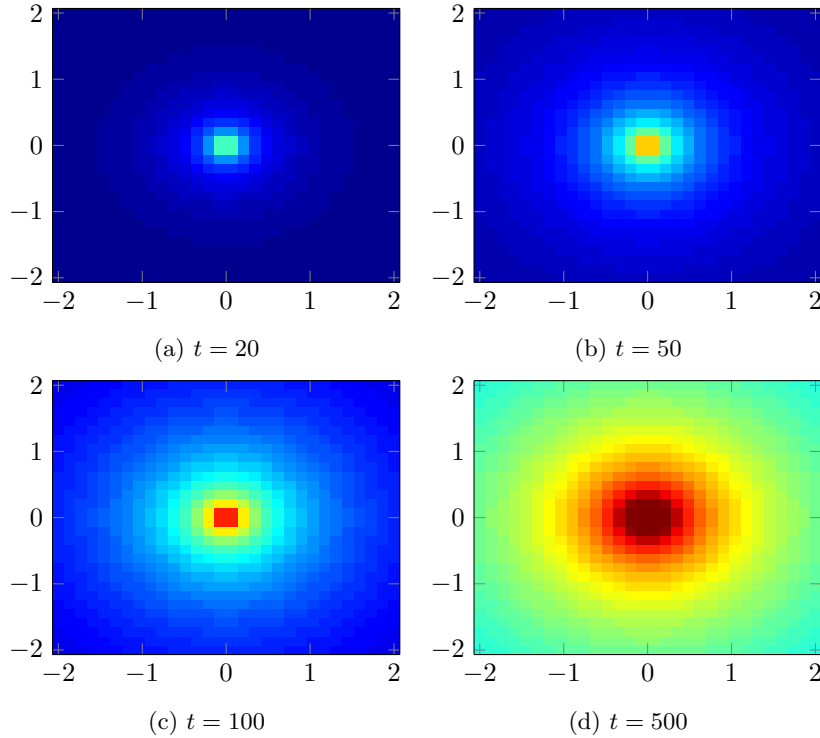
Schatting randvoorwaardeparameter ϕ

De waarde van de FVM-benadering in de randpunten geeft een goede indicatie van de juistheid van de keuze voor ϕ als deze wordt vergeleken met de analytische oplossing. Om goed te kunnen vergelijken is het zaak een grote t te kiezen, zodat de melkzuurconcentratie ongeveer stationair is. Er wordt één bacterie met $\mathbf{r}_k = (0, 0)^T$ bekeken. Voor $t = 500$, $m = 80$ en $\phi = 0.2$ is de maximale relatieve fout op de rand ongeveer gelijk aan 0.17. Deze relatieve fout is nog redelijk groot, maar wordt vooral veroorzaakt door grote fouten in de hoeken van het domein. Omdat zich daar tijdens simulaties bijna nooit cellen zullen bevinden is de fout in de melkzuurconcentratie die daadwerkelijk gevolgen heeft kleiner dan 0.17. Voor andere waarden van ϕ wordt geen kleinere maximale relatieve fout gevonden. Daarom zal vanaf nu $\phi = 0.2$ worden gebruikt.

Foutanalyse

Figuur 3.1 toont kleurenplots van de absolute fout tussen de FVM-benadering en de analytische oplossing voor het geval van één bacterie, met $\mathbf{r}_k = (0, 0)^T$, voor $t = 20$, $t = 50$, $t = 100$ en $t = 500$. De kleuren zijn geschaald voor

zuurconcentraties tussen 1 en 3, dat wil zeggen: als het absolute verschil groter is dan 3, is een rode kleur te zien, als het absolute verschil 0 is, een donkerblauwe. Te zien is dat op $t = 20$ de absolute fout zeer klein is. Voor $t = 50$ ontstaan er iets

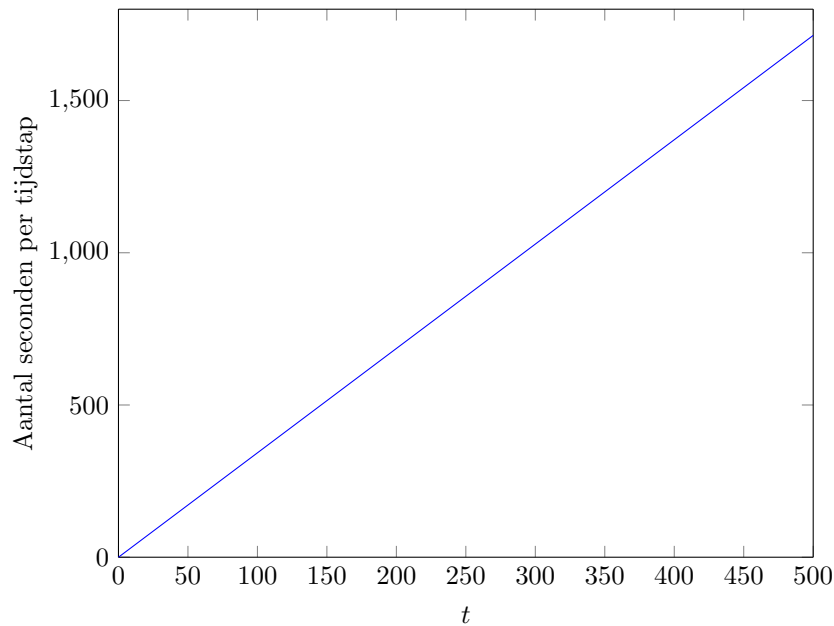


Figuur 3.1: Kleurenplots van de absolute fout tussen de FVM-benadering en de analytische oplossing, met $m = 30$.

grotere fouten. Voor de FVM-benadering maakt het, gegeven dat een bacterie zich in controlevolume V bevindt, niet uit wáár in het controlevolume deze bacterie zich bevindt. Immers: in één controlevolume ligt maar één roosterpunt. Voor de analytische oplossing maakt het juist een groot verschil waar in het controlevolume een bacterie zich bevindt. Dit komt door de Diracdelta, die een singulariteit in het integrand van de analytische oplossing oplevert. Door deze singulariteit is de grafiek van de melkzuurconcentratie heel steil in de buurt van een bacterie. Zodoende kan $c(\mathbf{r})$ sterk verschillen van bijvoorbeeld $c(\mathbf{r} + \hat{\mathbf{h}}\mathbf{n}/2)$ (\mathbf{n} een willekeurige eenheidsvector). Dit verklaart de rode kleur die rond de positie van de bacterie te zien is vanaf $t = 50$. Wat verder weg van de bacterie, in het lichtblauwe gebied, is de fout ongeveer 0.5. De invloed van deze fout op de celmobilititeit is nog redelijk beperkt. Voor $t = 100$ is al een groter gebied te zien waar ongeveer een fout van 0.5 plaatsheeft. Voor $t = 500$ zijn pas echt grote fouten te zien, zeker in de buurt van \mathbf{r}_k . Echter zijn dan wel de waarden van de FVM-benaderingen groter dan die van de analytische oplossing. Omdat de analytische oplossing rond de bacterie op $t = 500$ ongeveer gelijk aan 4 is, is de uiteindelijke invloed van de fout op de celmobilititeit nog steeds klein, immers: $\exp(-\nu_1 \cdot 4) \approx 0.09$, zodat de cellen sowieso nog maar nauwelijks zouden kunnen bewegen.

Tijdsbesparing

Het benodigde aantal seconden om de melkzuurconcentratie te berekenen met de analytische oplossing neemt lineair toe met t , als een constant aantal cellen en bacteriën wordt verondersteld. In figuur 3.2 is dit gevisualiseerd.



Figuur 3.2: Aantal seconden dat per tijdstap nodig is om de melkzuurconcentratie te berekenen, als functie van t .

Om de FVM-benadering zo snel mogelijk te maken, worden bij aanvang drie inversen berekend voor $(I - \Delta t A)$. Dit duurt enkele tientallen seconden. Daarna zal Δt steeds naar beneden worden afgerond om zo één van de drie inversen te kunnen gebruiken voor het Euler Achterwaarts-schema. Na het berekenen van de inversen is per tijdstap een verwaarloosbare hoeveelheid tijd nodig om de melkzuurconcentratie te berekenen. De grafiek in figuur 3.2 toont dus een bijna volledige tijdsbesparing.

Gradiënt

Om de beweging van witte bloedcellen (2.39) te kunnen bepalen, zijn de waarden van ∇c nodig. Deze kunnen worden berekend met de FVM-benadering van c . In eerste instantie lijken centrale differenties aantrekkelijk, omdat deze vaak kleinere fouten opleveren dan voorwaartse. Bijvoorbeeld:

$$\left(\frac{\partial C}{\partial x}\right)_P \approx \frac{C_E - C_W}{2\hat{h}}. \quad (3.30)$$

Echter, als het aantal bacteriën klein is, is er veel radiale symmetrie, waardoor soms $C_E \approx C_W$ en er onterecht een kleine waarde voor $\partial c/\partial x$ wordt bepaald.

Om dit te voorkomen zullen voorwaartste differenties worden gebruikt:

$$\left(\frac{\partial C}{\partial x}\right)_P \approx \frac{C_E - C_P}{2\hat{h}} \quad (3.31)$$

$$\left(\frac{\partial C}{\partial y}\right)_P \approx \frac{C_N - C_W}{2\hat{h}}. \quad (3.32)$$

Voor roosterpunten op de lijnen $\{(x, y) \in \mathbb{R}^2 : x = 2\}$ en $\{(x, y) \in \mathbb{R}^2 : y = 2\}$ moet de randvoorwaarde $\nabla \cdot c = -\phi c$ worden gebruikt.

Zelfs met gebruik van de voorwaartse differenties kunnen zich nog problemen voordoen. Als bijvoorbeeld een witte bloedcel zich links van een bacterie bevindt, maar wel in hetzelfde controlevolume, zal hij waarschijnlijk een negatieve $\partial C/\partial x$ toegekend krijgen. Om dit probleem te voorkomen moet m zodanig worden gekozen dat $\sqrt{\hat{h}^2 + \hat{h}^2} = \sqrt{2}\hat{h} < R_0$. Voor de simulaties zal $m = 80$ worden gebruikt.

Conclusie FVM-benadering

De FVM-benadering kan als redelijk worden beschouwd. Voor grote t zijn er grote verschillen met de analytische oplossing te zien, maar de invloed op de celmobilititeit blijft redelijk beperkt. Bovendien is het niet aannemelijk dat bacteriën 500 seconden vrij spel hebben: ze zullen waarschijnlijk al veel eerder onschadelijk zijn gemaakt door de witte bloedcellen. Ook kan worden beargumenteerd dat het misschien wel realistischer is om de melkzuurproductie te modelleren met een bronterm op een klein oppervlak \hat{h}^2 , dan met een Diracdelta. De Diracdelta is immers een wiskundig concept, in werkelijkheid zal een bacterie geen puntbron zijn. De FVM-benadering biedt een aantal voordelen:

- de benodigde tijd voor het berekenen van $c(\mathbf{r}, t)$ hangt niet meer af van t ;
- het aantal bacteriën heeft nauwelijks invloed op de benodigde rekenkracht voor het bepalen van $c(\mathbf{r}, t)$;
- het aantal epitheelcellen heeft geen invloed meer op het aantal berekeningen dat omtrent de melkzuurconcentratie moet worden gedaan per tijdstap;
- het opruimen van melkzuur door witte bloedcellen kan nu op een heel gemakkelijke manier worden gemodelleerd, en het aantal witte bloedcellen heeft nauwelijks invloed op de benodigde rekenkracht voor het bepalen van $c(\mathbf{r}, t)$.

Hoofdstuk 4

Resultaten

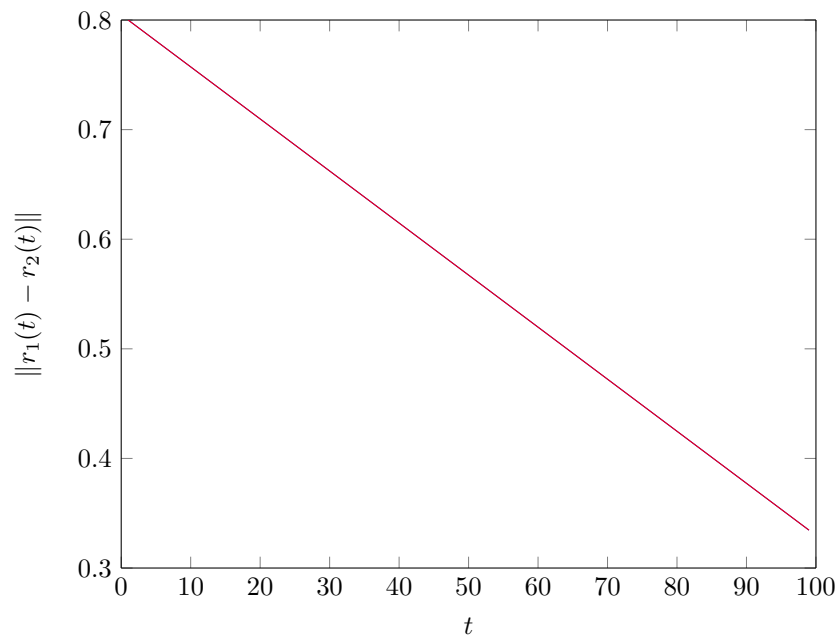
In dit hoofdstuk zullen enkele simulaties worden besproken die met het model zijn verkregen.

4.1 Numeriek vs. analytisch voor twee cellen

In [1] wordt de analytische oplossing gegeven voor de deterministische beweging van twee cellen, bij constante straal R_0 en afwezigheid van melkzuur. Voor de eerste cel geldt:

$$r_1(t) = \frac{R_0}{2\lambda} \ln \left(\exp \left(-\frac{2\lambda\alpha M_1(0)t}{R_0} \right) \cdot \left(\exp \left(\frac{2\lambda r_0}{R_0} \right) + 1 \right) - 1 \right). \quad (4.1)$$

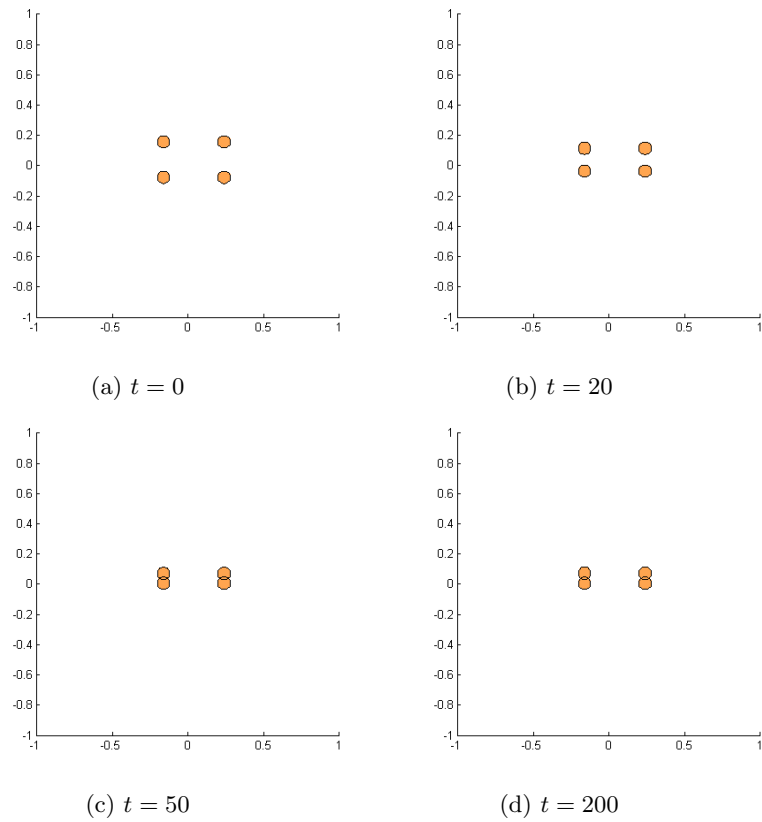
Hierbij is r_0 de beginpositie van de cel. Voor de tweede cel geldt $r_2(t) = -r_1(t)$. De beweging gaat dus in feite over de x -as en is symmetrisch in de oorsprong. Zonder verlies van algemeenheid kan dit worden verondersteld. Deze analytische oplossing geldt tot het moment dat de cellen elkaar raken. Zodra er van afstotingskrachten sprake is, is de oplossing niet meer geldig. De analytische oplossing wordt vergeleken met de numerieke, voor $r_0 = 10R_0$. In figuur 4.1 is $\|r_1 - r_2\|$ uitgezet tegen t , met in het rood de analytische oplossing, en in het blauw de numerieke. Zoals te zien is het verschil maar zeer klein. Het maximale absolute verschil tussen de twee is $1,51 \cdot 10^{-12}$. Dit motiveert de juistheid van het gebruik van Euler Voorwaarts voor de bewegingsvergelijkingen van de epitheelcellen en witte bloedcellen.



Figuur 4.1: Afstand tussen twee cellen uitgezet tegen de tijd. In het rood de analytische oplossing en in het blauw de numerieke.

4.2 Beweging van vier cellen

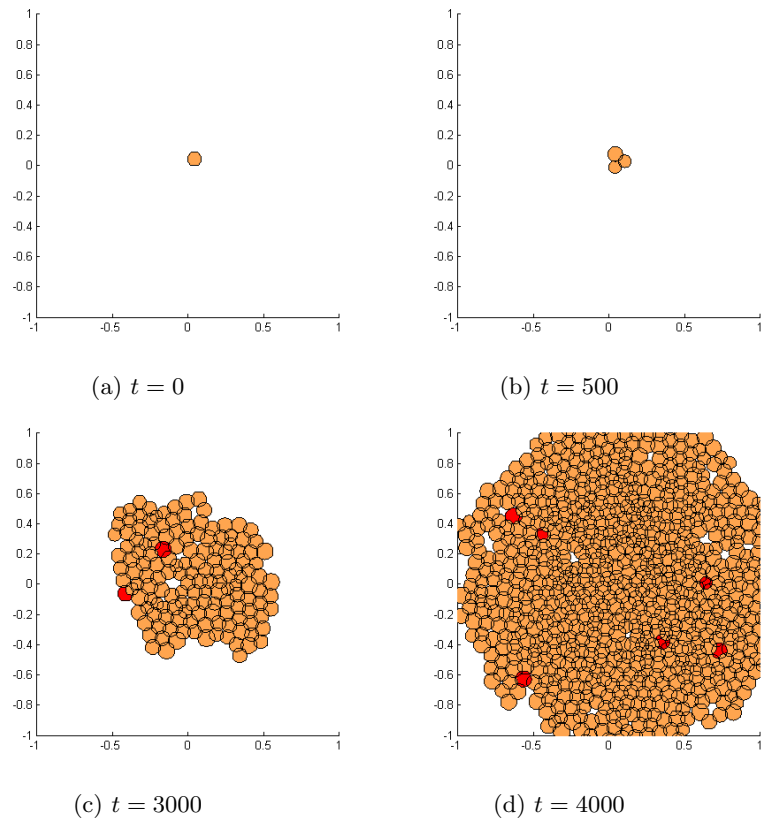
Figuur 4.2 toont de beweging van vier cellen. De cellen zullen in eerste instantie naar de dichtstbijzijnde cel migreren. Zodra ze deze hebben bereikt (4.3c) zullen de twee duo's van cellen niet meer naar elkaar toe bewegen. Hiermee toont deze situatie interessant gedrag: cellen willen niet alleen zijn, en zullen daarom altijd op zoek gaan naar een andere cel. Zodra deze is gevonden echter, zijn signalen van andere cellen vaak niet sterk genoeg meer om invloed op de beweging uit te oefenen. Slechts in het geval van heel veel cellen (duizenden of meer) gebeurt dit wel.



Figuur 4.2: Beweging van vier cellen.

4.3 Koloniegroei

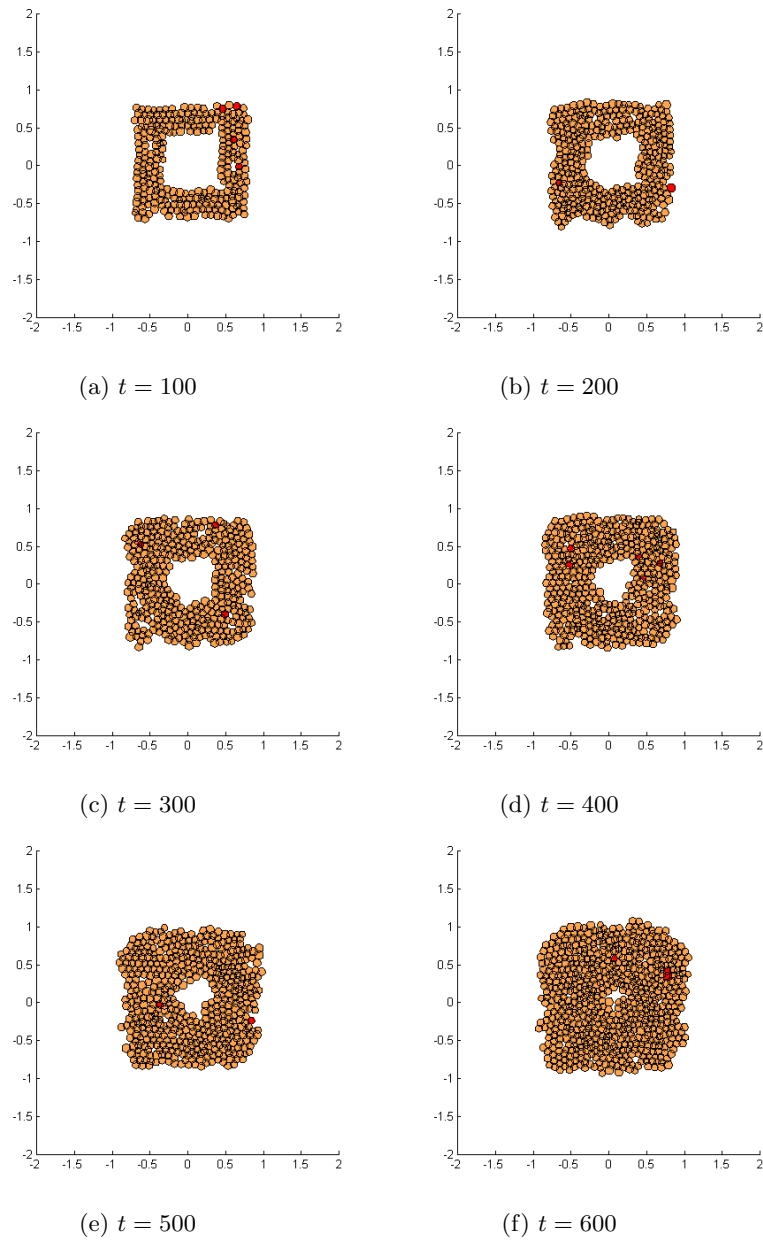
Figuur 4.3 toont de uitgroei van één cel tot een kolonie. Als parameters zijn $\mu_D = 300$ en $p_s = 0.001$ gebruikt. In [1] wordt ook koloniegroei behandeld. Daar moet de delingskans zorgvuldig worden gekozen om te voorkomen dat cellen zich ophopen. Als de delingskans te hoog is hebben de cellen geen tijd om elkaar weg te duwen. In het huidige model is van dit probleem in veel minder sterke mate sprake. Omdat een cel tijdens zijn delingscyclus door te groeien al ruimte vrijmaakt voor twee cellen, vindt er bijna geen ophoping plaats.



Figuur 4.3: Uitgroei tot een kolonie. Vitale cellen zijn beige gekleurd, dode rood.

4.4 Heling van een schone wond

In figuur 4.4 is een schone, helende wond te zien op verschillende tijdstippen. Bij afwezigheid van bacteriën en dus melkzuur, kunnen de cellen ongeremd bewegen en delen, zodat de wond vrij snel dichtgroeit. Op $t = 600$, wat overeen komt met twee dagen, is de wond zo goed als dicht.

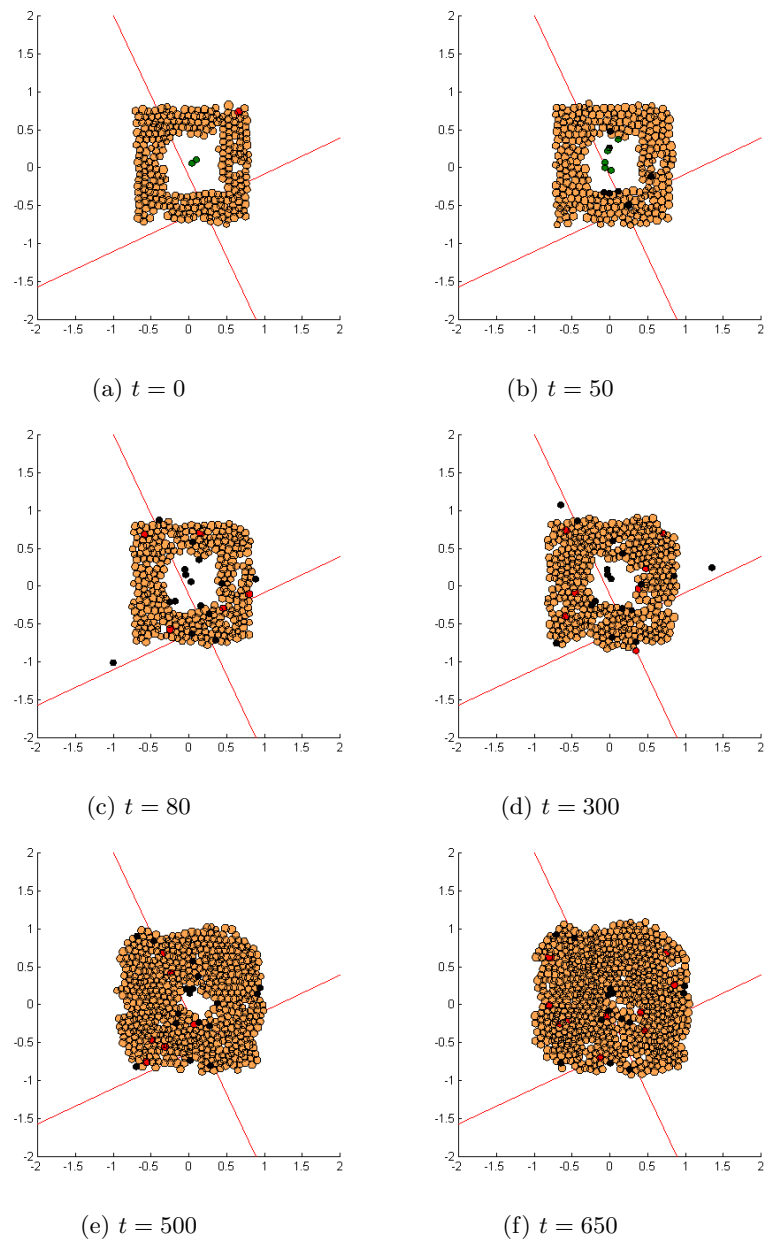


Figuur 4.4: Heling van een schone wond. Vitale cellen zijn beige gekleurd, dode rood.

4.5 Heling van een licht geïnfecteerde wond

Figuur 4.5 toont heling van een licht geïnfecteerde wond. Op $t = 0$ bevinden zich twee bacteriën in de wond, met delingskans 0.03. De bacteriën zijn groen gekleurd, en hebben een onrealistisch grote straal, maar dat is om ze goed zicht-

baar te maken. Voor de berekeningen worden ze als puntbronnen behandeld. Het duurt even voor er witte bloedcellen de wond binnenkomen, omdat het melkzuur zich eerst moet verspreiden tot aan de creatieputen. Op $t = 50$ is te zien dat de eerste witte bloedcellen de wond binnen zijn gekomen. Op basis van de zuurgradiënt bewegen ze richting de bacteriën, om ze onschadelijk te maken. Op $t = 80$ zijn alle bacteriën gedood. Er heerst dan wel nog steeds een melkzuurconcentratie, en om die reden zullen er nog een tijdje witte bloedcellen de wond binnenkomen. Op $t = 500$ is er bijna niets meer te merken van de aanwezigheid van bacteriën die er bij aanvang was, behalve dat er nog steeds witte bloedcellen in het weefsel zitten. Deze zullen pas verdwijnen na sterfte. Sterfte is voor witte bloedcellen (los van de sterfte die plaatsheeft als een witte bloedcel vijf bacteriën heeft gedood) op dezelfde manier gemodelleerd als voor epitheelcellen. Op $t = 650$ is de wond dicht. Uiteindelijk heeft de lichte infectie maar een kleine invloed gehad op de tijdsduur van het wondhelingsproces, vooral omdat de bacteriën snel onschadelijk konden worden gemaakt.

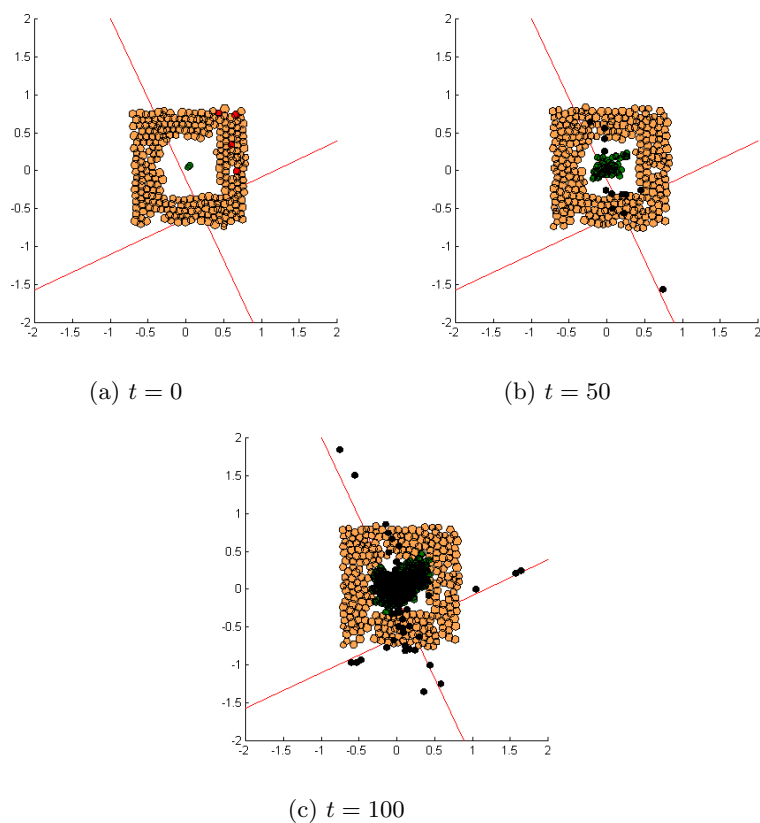


Figuur 4.5: Heling van een licht geïnfekteerde wond. Vitale cellen zijn beige gekleurd, dode rood. Bacteriën zijn groen en witte bloedcellen zwart.

4.6 Zware infectie

In figuur 4.6 is een zwaar geïnfekteerde wond te zien. Op $t = 0$ bevinden zich twee bacteriën in de wond, met delingskans 0.08. Op $t = 50$ komen de eerste witte bloedcellen de wond binnen, maar de hoeveelheid bacteriën is dan al zo

groot, dat ze er niet meer tegenop kunnen. De zuurconcentraties in de wond zijn zo hoog dat er wat betreft de epitheelcellen praktisch geen beweging en deling meer plaatsvindt. Op $t = 100$ is een waar offensief van de witte bloedcellen te zien, maar de bacteriën delen zo snel dat het nauwelijks effect heeft. Uiteindelijk zullen alle epitheelcellen sterven, alhoewel dit wel redelijk lang zal duren.



Figuur 4.6: Zware infectie. Vitale cellen zijn beige gekleurd, dode rood. Bacteriën zijn groen en witte bloedcellen zwart.

Conclusie/Discussie

De doelstelling bij aanvang van dit project was om het model in [1] te verbeteren. De conclusie is dat dit is gelukt. Het model beschikt nu over een realistischere implementatie van celdeling, de invloed van melkzuur op celdeling is toegevoegd en witte bloedcellen zijn in staat gesteld melkzuur op te ruimen. Zodoende is het model nu in theorie meer waarheidsgetrouw. Bovendien kunnen simulaties nu veel sneller worden uitgevoerd door de Finite Volume Method-benadering voor de melkzuurconcentratie.

Of de simulaties daadwerkelijk juist zijn, is op dit moment lastig te zeggen. Om hierover sluitende uitspraken te doen, zal het model moeten worden vergeleken met echte wonden en infecties. De simulaties leveren in de meeste gevallen aannemelijke resultaten op, maar sommige parameters zijn ook juist met die bedoeling gekozen, omdat er in de literatuur nauwelijks waarden te vinden waren.

De duur van een delingscyclus is van ongeveer een dag geschaald naar vijf minuten. Er werd bij simulaties steeds gekeken of er geen opeenhopingen ontstonden, maar nog steeds kan worden afgevraagd of een dergelijke schaling altijd juist is. In werkelijkheid zal de druk die cellen ondervinden remmend werken op hun groei.

De gebruikte Finite Volume Method-benadering is te verdedigen, omdat voor één cel de verschillen met de analytische oplossing voor het grootste deel redelijk klein zijn. Echter voor een groot aantal bacteriën zullen de verschillen zodanig groot zijn, dat moeilijk nog van een goede benadering kan worden gesproken. Eventueel zou verder onderzoek kunnen worden gedaan, bijvoorbeeld naar hoe de verschillen in de buurt van een bacterie kunnen worden ingedamd. Anderzijds moet worden opgemerkt dat de analytische oplossing in zichzelf ook al een benadering was. Melkzuur komt bij diffusie cellen tegen waar het omheen moet, en bevindt zich bovendien in de wond met nog allerlei andere stoffen. De essentie van het modelleren van melkzuur is uiteindelijk dat een wond langzamer dichtgroeit bij aanwezigheid van bacteriën. Deze essentie blijft overeind staan.

De grootte van de epitheelcellen zoals in dit project gebruikt is niet correct. In [1] was hiervan ook al sprake. Kwalitatief hoeft dit voor de simulaties niet per se een groot verschil te maken, echter kan wel de detectiedrempel $\bar{\epsilon}$ belangrijker worden als grotere cellen worden bekeken.

Bibliografie

- [1] Arkesteijn, L. (2012): Semi-stochastische aanpak van migratie, sterfte en deling in geïnfecteerde celkolonies.
- [2] Van der Meijden, R. (2013): Een semi-stochastisch cell-based model voor de ontwikkeling van een longcarcinoom.
- [3] Vermolen, F.J.; Arkesteijn, E.C.M.M.; Gefen, A. (2015): Modeling the Immune System Response to Epithelial Wound Infections
- [4] Steele, J.M. (2001): Stochastic Calculus and Financial Applications, Springer, New York
- [5] Reinhart-King, C.A., Dembo, M, Hammer, D.A. (2008): Cell-cell mechanical communication through compliant substrates, *Biophysics Journal*, 95, 6044-6051
- [6] Haberman, R. (2012): Applied Partial Differential Equations with Fourier Series and Boundary Value Problems
- [7] Vermolen, F.J., Gefen, A. (2013): A semi-stochastic cell based model for in-vitro infected ‘wound’ healing
- [8] Vuik, C., Van Beek, P., Vermolen, F., Van Kan, J. (2006): Numerieke Methoden voor Differentiaalvergelijkingen
- [9] Van Kan, J, Segal, A., Vermolen, F. (2008): Numerical Methods in Scientific Computing
- [10] Gefen, A. (2010): Effects of virus size and cell stiffness on forces, work, and pressures driving membrane invagination in a receptor-mediated endocytosis, *J. Biomech Eng*, 132: 084501-1-08405-5

Bijlagen

Onderstaand de gebruikte Matlabcode:

```
1 clear all; close all; clc;
2 randn('state',100); rand('state',100);
3
4
5 % parameters
6 F_hat = 0.01; l = 10; Es = 5; Ec = 2; R_0 = 0.04; alpha =
    0.006;
7 lambda = Es/Ec; nu = -0.6; Db = 0.01; D = 1e-6; p_rw =
    0.1;
8 p_s = 0.001; p_d_bac = 0.08; gamma_bac = 0.1; gamma_WB =
    0.001;
9 phi = 0; zeta = -0.1; mu_D = 300; numcr=100; tau_0 = 1;
10 maxkills_WB = 5; alpha_WB = alpha; v_WB = 0.5; gradmax =
    10; m=80;
11
12 % initiele celverdeling bepalen. de x-coördinaat van cel
    i staat op
13 % positie 2*i-1 in r, de y-coördinaat op positie 2*i.
14 % in r worden ook de coördinaten van WB-cellen opgeslagen
    . InitSubstrate
15 % retourneert als functie van een index een initiele
    celverdeling
16 [N,r] = InitSubstrate(6);
17 % bloedvatenstelsel bepalen, inclusief creatiepunten voor
    WB-cellen.
18 % bvatstelsel retourneert als functie van numcr een
    bloedvatenstelsel
19 [vatx ,vaty ,tak1x ,tak1y ,tak2x ,tak2y ,crx ,cry] = bvatstelsel
    (numcr);
20
21 % F als functie van tijdstip en celstraal
22 F = @(s,R)(F_hat/l*(R/R_0)^2*(1-s));
23
24 % matrices, vectoren, stapgrootte voor FVM bepalen
25 hFVM = 4/(m-1); % stapgrootte voor FVM
26 [A,fcoeff] = FVMmatrices(m); C = zeros(m^2,1);
```

```

27 % drie inverses, voor verschillende tau, om benodigde
    rekenkracht laag te
28 % houden
29 invA1 = inv(eye(m^2)-Db/h_FVM^2*A);
30 invA05 = inv(eye(m^2)-0.5*Db/h_FVM^2*A);
31 invA01 = inv(eye(m^2)-0.1*Db/h_FVM^2*A);
32 x = -2:h_FVM:2; y=x; % rooster op basis van h_FVM
33
34 % initiele celstralen bepalen
35 t_int = 0.6*mu_D*rand(N,1);
36 fase = zeros(N,1);
37 R = zeros(N,1);
38 xi = R_0^2/(2*0.6*mu_D);
39 for i=1:N
40     if t_int(i)<0.3*mu_D
41         fase(i) = 1; % G1-fase
42         R(i) = sqrt(R_0^2+xi*t_int(i));
43     elseif t_int(i)<0.7*mu_D
44         fase(i) = 2; % S-fase
45         R(i) = sqrt(R_0^2+xi*0.3*mu_D);
46     else
47         fase(i) = 3; % G2-fase
48         R(i) = sqrt(R_0^2+xi*(t_int(i)-0.4*mu_D));
49     end
50 end
51
52 % initiele bacterieverdeling
53 N_bac = 0; x_bac = []; y_bac = [];
54
55 kmax = 10000; N_WB = 0; kills_WB = zeros(N_WB,1); t_s =
    kmax*ones(N,1);
56 U_rw = rand(N,kmax); U_d = rand(N,kmax); U_s = rand(N+
    N_WB,kmax);
57 W = randn(2*N,kmax); U_d_bac = rand(1,kmax); U_StoG2 =
    rand(N,kmax);
58 U_WB = rand(numcr,kmax); dr = zeros(2*N,kmax); t = 0;
59
60 % fig = figure('units','normalized','outerposition',[0 0
    1 1]);
61 framestep = 1;
62 fig = figure;
63 c = [];
64 axis([-2 2 -2 2]); axis square;
65 hold on
66 plot(vatx,vaty,'r'); plot(tak1x,tak1y,'r'); plot(tak2x,
    tak2y,'r');
67
68
69
70

```

```

71
72 gradCx = zeros(m^2,kmax); gradCy = zeros(m^2,kmax); C =
      zeros(m^2,kmax);
73 for k = 2:kmax
74     if k == 100
75         N_bac = 1; x_bac = [0]; y_bac = [0];
76     end
77     k
78     M_att = zeros(N+N.WB); M_rep = zeros(N,N);
79     vx = zeros(N,N); vy = zeros(N,N); zx = zeros(N,1); zy
      = zeros(N,1);
80     drx = zeros(N,1); dry = zeros(N,1); tau = tau_0; mzc
      = zeros(N,1);
81     % bepalen aantrekkings- en afstotingskrachten tussen
      epitheel- en
82     % WB-cellen
83     for i = 1:N+N.WB
84         r_i = [r(2*i-1);r(2*i)];
85         for j = 1:N+N.WB
86             r_j = [r(2*j-1);r(2*j)];
87             M_0_j = F(max(t-t_s(j),0),R(j))^2/(2*Es*pi^2*
      R(j)^4);
88             M_att_ij = M_0_j*exp(-lambda*norm(r_i-r_j)/R(
      j));
89             M_att(i,j) = M_att_ij;
90             if i~=j
91                 h = max(R(i)+R(j)-norm(r_i-r_j),0);
92                 M_rep_ij = 6/15*sqrt(h^5*R(i)/2*(2*Ec/3)
      ^2)/pi/R(i)^3;
93                 M_rep(i,j) = M_rep_ij;
94             end
95             v_ij_x = r_j(1)-r_i(1);
96             v_ij_y = r_j(2)-r_i(2);
97             norm_v = norm([v_ij_x , v_ij_y]);
98             if norm_v ~= 0
99                 vx(j,i) = v_ij_x/norm_v;
100                vy(j,i) = v_ij_y/norm_v;
101             else
102                 vx(j,i) = 0;
103                 vy(j,i) = 0;
104             end
105         end
106     end
107     % verplaatsing epitheelcellen
108     for i=1:N
109         % bepalen melkzuurconcentratie waaraan cel i
      onderhevig is
110         I = find(r(2*i-1)*ones(1,m)>x-hFVM/2*ones(1,m));
      I = I(end);

```



```

111     J = find(r(2*i)*ones(1,m)>y-h_FVM/2*ones(1,m)); J
        = J(end);
112     mzc(i) = C((J-1)*m+I,k-1);
113     % bepalen richtingsvector cel i
114     zx(i) = (exp(nu*mzc(i))*M_att(i,:)-M_rep(i,:))*vx
        (:,i);
115     zy(i) = (exp(nu*mzc(i))*M_att(i,:)-M_rep(i,:))*vy
        (:,i);
116     normz_i = norm([zx(i),zy(i)]);
117     if normz_i ~ 0
118         zx(i) = zx(i)/normz_i; zy(i) = zy(i)/normz_i;
119     else
120         U_rw(i,k) = 1;
121     end
122     if t_s(i) == kmax
123         % deterministische beweging cellen
124         if U_rw(i,k) < 1-p_rw
125             M_i = sum(abs(exp(nu*mzc(i))*M_att(i,:)-
                M_rep(i,:)));
126             drx(i) = alpha*M_i*zx(i);
127             dry(i) = alpha*M_i*zy(i);
128             tau_temp = R_0/2/sqrt(drx(i)^2+dry(i)^2);
129             % random walk cellen
130         else
131             X = W(2*i-1,k); Y = W(2*i,k);
132             drx(i) = exp(nu*mzc(i))*sqrt(2*D)*X;
133             dry(i) = exp(nu*mzc(i))*sqrt(2*D)*Y;
134             tau_temp = R_0^2/8/D/exp(nu*mzc(i))/(X^2+
                Y^2);
135         end
136         % verplaatsing als de cel dood is
137     else
138         drx(i) = 0; dry(i) = 0;
139     end
140     % bepalen tijdstap tau zodanig dat geen enkele
        cel meer dan R_0/2
141     % aflegt
142     if tau_temp < tau
143         tau = tau_temp;
144     end
145 end
146 % verplaatsing wb-cellen
147 for i = N+1:N+N.WB
148     % bepalen zuurgradint die door WB-cel i wordt
        opgemerkt
149     I = find(r(2*i-1)*ones(1,m)>x-h_FVM/2*ones(1,m));
150     I = I(end);
151     J = find(r(2*i)*ones(1,m)>y-h_FVM/2*ones(1,m));
152     J = J(end);

```

```

153     gradCxi = gradCx((J-1)*m+I,k-1); gradCyi = gradCy
        ((J-1)*m+I,k-1);
154     % bepalen beweging als gevolg van interactie met
        andere cellen
155     M_i = sum(abs(M_att(i,:) - M_rep(i,:)));
156     zx(i) = (M_att(i,:) - M_rep(i,:)) * vx(:,i);
157     zy(i) = (M_att(i,:) - M_rep(i,:)) * vy(:,i);
158     normz_i = norm([zx(i), zy(i)]);
159     zx(i) = zx(i)/normz_i;
160     zy(i) = zy(i)/normz_i;
161     % begrenzen zuurgradient
162     normgrad = norm([gradCxi, gradCyi]);
163     if normgrad > gradmax
164         gradCxi = gradmax*gradCxi/normgrad;
165         gradCyi = gradmax*gradCxi/normgrad;
166     end
167     drx(i) = alpha_WB*(M_i*zx(i)+v_WB*gradCxi);
168     dry(i) = alpha_WB*(M_i*zy(i)+v_WB*gradCyi);
169     tau_temp = R_0/2/sqrt(drx(i)^2+dry(i)^2);
170     if tau_temp < tau
171         tau = tau_temp;
172     end
173 end
174 % tau naar beneden afronden naar 0.1, of 0.5 als dat
        kan, om zo van de
175 % vooraf bepaalde inversematrices voor de FVM gebruik
        te kunnen maken
176 if tau < 1
177     if tau > 0.5
178         tau = 0.5;
179     elseif tau > 0.1
180         tau = 0.1;
181     end
182 end
183 % nieuw tijdstip
184 t = t+tau;
185
186 % nieuwe plaats epitheelcellen
187 for i=1:N
188     if U_rw(i,k) < 1-p_rw
189         r(2*i-1) = r(2*i-1)+tau*drx(i);
190         r(2*i) = r(2*i)+tau*dry(i);
191     else
192         r(2*i-1) = r(2*i-1)+sqrt(tau)*drx(i);
193         r(2*i) = r(2*i)+sqrt(tau)*dry(i);
194     end
195     % afronden op 14 decimalen
196     r(2*i-1) = str2num(sprintf('%0.14f', r(2*i-1)));
197     r(2*i) = str2num(sprintf('%0.14f', r(2*i)));
198 end

```

```

199 % nieuwe plaats wb-cellen
200 for i=N+1:N+N.WB
201     r(2*i-1) = r(2*i-1)+tau*drx(i);
202     r(2*i) = r(2*i)+tau*dry(i);
203     r(2*i-1) = str2num(sprintf('%0.14f',r(2*i-1)));
204     r(2*i) = str2num(sprintf('%0.14f',r(2*i)));
205 end
206 % nieuwe plaats bacterien
207 for i = 1:N_bac
208     dir_bac_x = rand-0.5; dir_bac_y = rand-0.5;
209     v_bac = rand; norm_bac = norm([dir_bac_x,
210     dir_bac_y]);
211     dir_bac_x = dir_bac_x/norm_bac;
212     dir_bac_y = dir_bac_y/norm_bac;
213     x_bac(i) = x_bac(i)+tau*R_0*v_bac*dir_bac_x;
214     y_bac(i) = y_bac(i)+tau*R_0*v_bac*dir_bac_y;
215 end
216 % bacteriedeling
217 for i = 1:N_bac
218     if N_bac < 1000
219         if U_d_bac(i,k) > 1-p_d_bac*tau
220             N_bac = N_bac+1; U_d_bac = [U_d_bac;rand
221             (1,kmax)];
222             x_bac = [x_bac,x_bac(i)]; y_bac = [y_bac,
223             y_bac(i)];
224         end
225     end
226 end
227 % creatie wb-cellen
228 for i = 1:numcr
229     % bepalen zuurconcentratie die geldt in
230     creatiepunt i
231     I = find(crx(i)*ones(1,m)>x-h.FVM/2*ones(1,m)); I
232     = I(end);
233     J = find(cry(i)*ones(1,m)>y-h.FVM/2*ones(1,m)); J
234     = J(end);
235     mzc_WB.i = C((J-1)*m+I,k-1);
236     if UWB(i,k) > 1-tau*(1-exp(zeta*mzc_WB.i))/numcr
237         r = [r;crx(i);cry(i)]; N.WB = N.WB+1;
238         kills_WB = [kills_WB;0]; R = [R;R_0];
239         t_s = [t_s;kmax]; U_s = [U_s;rand(1,kmax)];
240     end
241 end
242 % bepalen welke bacterien worden gedood door WB-
243 cellen
244 for i = N+1:N+N.WB
245     for j = N_bac:-1:1
246         if kills_WB(i-N) < maxkills_WB
247             if norm([x_bac(j);y_bac(j)]-r(2*i-1:2*i))
248                 <R(i)

```

```

241         x_bac = [x_bac(1:j-1), x_bac(j+1:end)
242                ];
243         y_bac = [y_bac(1:j-1), y_bac(j+1:end)
244                ];
245         N_bac = N_bac-1;
246         kills_WB(i-N) = kills_WB(i-N)+1;
247     end
248 end
249 % witte bloedcel verwijderen als hij maxkills
250 % bacterien heeft gedood
251 for i=N+N.WB:-1:N+1
252     if kills_WB(i-N) == maxkills_WB
253         r = [r(1:2*(i-1)); r(2*(i+1)-1:end)]; R = [R
254                (1:i-1); R(i+1:end)];
255         N.WB = N.WB-1;
256         kills_WB = [kills_WB(1:i-N-1); kills_WB(i-N+1:
257                end)];
258     end
259 end
260 % bepalen celstraal
261 for i=1:N
262     if t_s(i) == kmax
263         if fase(i) == 1
264             t_int(i) = t_int(i)+exp(nu*mzc(i))*tau;
265             if t_int(i)>0.3*mu_D
266                 R(i) = sqrt(R_0^2+2*xi*0.3*mu_D);
267                 fase(i) = 2;
268             else
269                 R(i) = sqrt(R_0^2+2*xi*t_int(i));
270             end
271         elseif fase(i) == 2
272             if U_StoG2(i,k)>1-tau/0.4/mu_D*exp(nu*mzc
273                (i))
274                 fase(i) = 3;
275                 t_int(i) = 0.3*mu_D;
276             end
277         else
278             if t_int(i)+exp(nu*mzc(i))*tau<0.6*mu_D
279                 t_int(i) = t_int(i)+exp(nu*mzc(i))*
280                    tau;
281                 R(i) = sqrt(R_0^2+2*xi*t_int(i));
282             else
283                 U_rw = [U_rw; rand(1,kmax)]; U_d = [
284                    U_d; rand(1,kmax)];
285                 U_s = [U_s; rand(1,kmax)];
286                 U_StoG2 = [U_StoG2; rand(1,kmax)]; W =
287                    [W; randn(2,kmax)];

```

```

281         t_s = [t_s;kmax]; t_int(i) = 0; t_int
           = [t_int;0];
282         fase(i) = 1; fase = [fase;1]; dr_d =
           [drx(i),dry(i)];
283         drx_d1 = -dr_d(2); dry_d1 = dr_d(1);
284         norm_d1 = norm([drx_d1,dry_d1]);
285         drx_d1 = drx_d1/norm_d1*R_0/2;
286         dry_d1 = dry_d1/norm_d1*R_0/2;
287         drx_d2 = -drx_d1; dry_d2 = -dry_d1;
288         r1 = [r(2*i-1)+drx_d1;r(2*i)+dry_d1];
289         r2 = [r(2*i-1)+drx_d2;r(2*i)+dry_d2];
290         r = [r(1:2*(i-1));r1;r(2*i+1:2*N);r2;
           r(2*N+1:end)];
291         N = N+1; R(i) = R_0; R = [R(1:N-1);
           R_0;R(N:end)];
292     end
293 end
294 end
295 end
296 % celsterfte , tevens voor WB-cellen
297 for i=1:N+N.WB
298     if t_s(i)==kmax
299         if U_s(i,k) > 1-tau*p_s
300             t_s(i) = t;
301         end
302     end
303 end
304 % verwijderen dode cellen
305 verwijder = t*ones(N+N.WB,1)-t_s(1:N+N.WB) >= (1-
           tau_0)*ones(N+N.WB,1);
306 for i=N+N.WB:-1:N+1
307     if verwijder(i) == 1
308         r = r([1:2*(i-1),2*(i+1)-1:end]'); t_s = t_s
           ([1:i-1,i+1:end]');
309         R = R([1:i-1,i+1:end]'); N.WB = N.WB-1;
310     end
311 end
312 for i=N:-1:1
313     if verwijder(i) == 1
314         r = r([1:2*(i-1),2*(i+1)-1:end]'); t_s = t_s
           ([1:i-1,i+1:end]');
315         t_int = t_int([1:i-1,i+1:end]'); fase = fase
           ([1:i-1,i+1:end]');
316         R = R([1:i-1,i+1:end]'); N = N-1;
317     end
318 end
319 if N == 0
320     disp('alle cellen dood')
321     break
322 end

```

```

323 % bepalen melkzuurconcentratie
324 f = zeros(m^2,1);
325 % productie melkzuur door bacterin
326 for i=1:N_bac
327     I = find(x_bac(i)*ones(1,m)>x-h_FVM/2*ones(1,m));
328         I = I(end);
329     J = find(y_bac(i)*ones(1,m)>x-h_FVM/2*ones(1,m));
330         J = J(end);
331     f((J-1)*m+I) = f((J-1)*m+I)+coeff((J-1)*m+I)*
332         gamma_bac;
333 end
334 % opruimen melkzuur door WB-cellen
335 for i=N+1:N+N_WB
336     I = find(r(2*i-1)*ones(1,m)>x-h_FVM/2*ones(1,m));
337         I = I(end);
338     J = find(r(2*i)*ones(1,m)>x-h_FVM/2*ones(1,m)); J
339         = J(end);
340     f((J-1)*m+I) = f((J-1)*m+I)-coeff((J-1)*m+I)*
341         gamma_WB;
342     f((J-1)*m+I-1) = f((J-1)*m+I)-coeff((J-1)*m+I)*
343         gamma_WB;
344     f((J-1)*m+I+1) = f((J-1)*m+I)-coeff((J-1)*m+I)*
345         gamma_WB;
346     f((J-2)*m+I) = f((J-1)*m+I)-coeff((J-1)*m+I)*
347         gamma_WB;
348     f((J)*m+I) = f((J-1)*m+I)-coeff((J-1)*m+I)*
349         gamma_WB;
350     f((J-2)*m+I-1) = f((J-1)*m+I)-coeff((J-1)*m+I)*
351         gamma_WB;
352     f((J-2)*m+I+1) = f((J-1)*m+I)-coeff((J-1)*m+I)*
353         gamma_WB;
354     f((J)*m+I-1) = f((J-1)*m+I)-coeff((J-1)*m+I)*
355         gamma_WB;
356     f((J)*m+I+1) = f((J-1)*m+I)-coeff((J-1)*m+I)*
357         gamma_WB;
358 end
359 % gebruiken juiste inversematrix bij de huidige
360     tijdstap
361 if tau==1
362     C(:,k) = invA1*(C(:,k-1)+tau/h_FVM^2*f);
363 elseif tau==0.5
364     C(:,k) = invA05*(C(:,k-1)+tau/h_FVM^2*f);
365 elseif tau==0.1
366     C(:,k) = invA01*(C(:,k-1)+tau/h_FVM^2*f);
367 else
368     C(:,k) = inv(eye(m^2)-tau*Db/h_FVM^2*A)*(C(:,k-1)
369         +tau/h_FVM^2*f);
370 end
371 C(:,k) = max(C(:,k), zeros(m^2,1));
372 % bepalen gradient melkzuurconcentratie

```

```

357     for i=1:m^2
358         if mod(i,m) == 0
359             % randvoorwaarde
360             gradCx(i,k) = -phi*C(i,k);
361         else
362             % voorwaartse differenties
363             gradCx(i,k) = (C(i+1,k)-C(i,k))/hFVM;
364         end
365         if i<=(m-1)*m
366             % randvoorwaarde
367             gradCy(i,k) = (C(i+m,k)-C(i,k))/hFVM;
368         else
369             % voorwaartse differenties
370             gradCy(i,k) = -phi*C(i,k);
371         end
372     end
373     t
374     % visualisatie
375     if mod(k-2,framestep) == 0
376         delete(c); c = [];
377         % epitheelcellen
378         for i=1:N
379             if t_s(i)~=kmax
380                 c = [c,rectangle('Position',[r(2*i-1),r
381                     (2*i),2*R(i),2*R(i)],'Curvature'
382                     ,[1,1], 'FaceColor',[1,0,0])];
383             else
384                 c = [c,rectangle('Position',[r(2*i-1),r
385                     (2*i),2*R(i),2*R(i)],'Curvature'
386                     ,[1,1], 'FaceColor',[1,0.65,0.31])];
387             end
388         end
389         % WB-cellen
390         for i=N+1:N+N.WB
391             c = [c,rectangle('Position',[r(2*i-1),r(2*i)
392                 ,2*R(i),2*R(i)], 'Curvature',[1,1], '
393                 FaceColor','k')];
394         end
395         % bacterin
396         for i=1:N_bac
397             c = [c,rectangle('Position',[x_bac(i),y_bac(i)
398                 ),2*R_0,2*R_0], 'Curvature',[1,1], '
399                 FaceColor',[0,0.5,0])];
400         end
401         %
402         title(['t = ',num2str(t), ' N = ',num2str(N)])
403         mov((k-2)/framestep+1) = getframe(fig);
404     end
405 end

```

399
400 end