

Aus dem Institut für Theoretische Informatik
der Universität zu Lübeck

Direktor: Prof. Dr. math. K. Rüdiger Reischuk

Ein hybrides Automatenmodell
zur Simulation des Immunsystems

Diplomarbeit

vorgelegt von
Johannes Textor

27. März 2006



Betreuer: Prof. Dr. math. K. Rüdiger Reischuk

Zusammenfassung

Das Immunsystem ist eines der komplexesten Systeme des menschlichen Körpers. Bis heute weist unser Verständnis seines Aufbaus und seiner Funktion noch viele Lücken auf. In der immunologischen Forschung kommen seit einiger Zeit neben *in vivo* und *in vitro* Experimenten auch mathematische Modelle und Computersimulationen zum Einsatz. Diese können noch nicht das Immunsystem in seiner vollen Größe und Komplexität abbilden, wohl aber bei der Vorbereitung, Planung und Interpretation realer Experimente helfen.

Die mathematische Modellierung des Immunsystems geschieht meist durch Differentialgleichungssysteme, während zur Simulation im Computer agentenbasierte Systeme verwendet werden. In dieser Arbeit werden diese gegensätzlich erscheinenden Modellierungsverfahren miteinander in Verbindung gebracht: Das erwartete Verhalten eines probabilistischen Automatenmodells des Immunsystems wird durch Differentialgleichungen charakterisiert. Anhand dieser Charakterisierung wird aufgezeigt, wie die biophysikalische Plausibilität des Modells verbessert und die Performance durch hybride Simulation gesteigert werden kann. Abschließend wird das hybride Modell implementiert und getestet.

Vorwort und Dank

Mit der vorliegenden Diplomarbeit kommt mein Studium der Informatik nun zu seinem Abschluss. Ich widme diese Arbeit meinen Eltern, denen ich zutiefst dafür dankbar bin, dass sie mir dieses Studium ermöglicht haben. Obwohl meine Entscheidungen für sie nicht immer nachvollziehbar gewesen sein können, haben sie mich auf diesem langen Weg immer bedingungslos unterstützt. Es ist vor allem ihr Verdienst, dass ich nun diese Zeilen schreiben darf.

Auch mein Betreuer, Herr Prof. Reischuk, muss unerschütterliches Vertrauen in mich gehabt haben, um trotz der vielen Male, die ich im Laufe der Zeit Ideen entwickelt, untersucht und dann doch wieder verworfen habe, an einen Erfolg dieser Arbeit zu glauben. Zuletzt wurde aus einer vagen Idee dann doch noch eine, wie ich hoffe, runde Sache. Ihm gebührt mein Dank dafür, dass er mich trotz meiner zweifellos unberechenbaren Arbeitsweise immer wieder ermuntert und in meinem Glauben an ein glückliches Ende bestärkt hat.

Ohne Anspruch auf Vollständigkeit danke ich außerdem den hilfreichen Geistern, die mir zur Seite standen. Herr Prof. Prestin zeigte mir in wenigen Minuten den Ausweg aus einer ausweglos erscheinenden Differentialgleichungs-Sackgasse. Markus Hinkelmann gab wertvolle Starthilfe bei der Benutzung der *Mayflower*. Nicole Grabinski überprüfte und bereicherte mein laienhaftes Verständnis der Immunologie. In allen Problemen der Praxis konnte ich stets die Hilfe der guten Seelen des Instituts für theoretische Informatik, unserer Sekretärin Claudia Mamat und unseres Systemadministrators Jens Heinrichs, in Anspruch nehmen.

Schließlich danke ich allen, die dazu beigetragen haben, dass unsere schöne Universität wohl auch weiterhin die Universität zu Lübeck bleiben darf!

Inhaltsverzeichnis

1. Einleitung	8
1.1. Mathematische Modelle und Computersimulationen in der Immunologie . . .	8
1.1.1. Einsatzgebiete	8
1.1.2. Schwerpunkte und Vereinfachungen	10
1.2. Ziel der Diplomarbeit	11
1.3. Aufbau der Diplomarbeit	11
2. Grundlagen	12
2.1. Agentenbasierte Modellierung des Immunsystems	12
2.1.1. Das Celada-Seiden-Modell	12
2.1.2. CImmSim	21
2.1.3. Bedeutung des CS-Modells	25
2.2. Differentialgleichungsmodelle des Immunsystems	25
2.2.1. Die Massenwirkungsgesetze	26
2.2.2. Das Modell von Carneiro et al.	32
2.3. Reaktions-Diffusions-Systeme	37
2.3.1. Modellierung von Diffusionsprozessen	37
2.3.2. Reaktions-Diffusions-Gleichungen	41
2.3.3. Strukturelle Ähnlichkeiten zum CS-Modell	43
3. Analyse des erwarteten Verhaltens des Celada-Seiden-Modells	44
3.1. Simulationsprozesse im CS-Modell	44
3.2. Erwartetes Verhalten der Simulationsprozesse	45
3.2.1. Diffusion	45
3.2.2. Proliferation	54
3.2.3. Interaktion	55
3.3. Zusammenfassung	60
4. Erweiterung des Celada-Seiden-Modells	61
4.1. Beschränkung der Kapazität der Gitterpunkte	61
4.2. Modifikation des Interaktionsprozesses	62
4.2.1. Behandlung affinitätsabhängiger Interaktionen	64
4.3. Diffusion von Zellen und Molekülen	64

4.4. Beschleunigung durch hybride Simulation	64
4.4.1. Approximation des Diffusionsprozesses	65
4.4.2. Approximation des Proliferationsprozesses	66
4.4.3. Approximation des Interaktionsprozesses	67
5. Ergebnisse und Diskussion	69
5.1. LImmSim – Implementierung des erweiterten CS-Modells	69
5.1.1. Grundzüge der Implementierung	70
5.1.2. Experimente	75
5.2. Zusammenfassung	82
6. Zusammenfassung und Ausblick	84
6.1. Ergebnisse der Arbeit	84
6.2. Ausblick	85
6.3. Fazit	86
A. Immunologische Begriffe und Abkürzungen	87
B. Inhalt der CD	90
B.1. Kompilieren von LImmSim	90
B.2. Aufruf von LImmSim	90
B.2.1. Einstellung der Parameter	92
B.3. Die LImmSim-GUI	93
B.4. Credits	93
Literaturverzeichnis	96

1. Einleitung

Die Immunologie ist eine faszinierende Wissenschaft, gilt aber als besonders schwer zugänglich. Um sich mit Computermodellen des Immunsystems zu befassen, muss man natürlich die Grundzüge des Aufbaus und der Funktion des Immunsystems und seiner wichtigsten zellulären und humoralen Bestandteile kennen. Hierbei kommt es weniger auf die genauen mikrobiologischen Vorgänge an. Vielmehr sollte man ein gutes Verständnis für Strukturen, funktionale Zusammenhänge und Interaktionen im Immunsystem mitbringen.

Es würde den Rahmen dieser Arbeit sprengen, eine vorbereitende Einführung in die Immunologie zu geben. Der Leser sei hier auf die einschlägigen Lehrbücher verwiesen [JTW02, RBM95, SB80]. Für Informatiker dürfte *An Interpretative Introduction to the Immune System* von Stephen Hofmeyr [Hof01] von besonderem Interesse sein, die das Immunsystem aus Sicht der Informatik und mit Hinblick auf die Modellierung im Computer beschreibt. Einige wichtige Begriffe und gebräuchliche Abkürzungen aus der Immunologie sind in Anhang A erklärt.

1.1. Mathematische Modelle und Computersimulationen in der Immunologie

Im Gegensatz zur Mathematik und Informatik ist die Immunologie eine experimentelle Wissenschaft, die sich methodisch vor allem auf *in vivo* und *in vitro* Experimente stützt. Mathematische Modelle und Computersimulationen kommen daher meist erst dort zum Einsatz, wo die Durchführung von Experimenten zum Erkenntnisgewinn aus verschiedenen Gründen nicht möglich oder nicht praktikabel ist, oder wo durch die Eigenschaften solcher Modelle ein besonderer Mehrwert entsteht.

1.1.1. Einsatzgebiete

Stützung neuer Hypothesen zur Funktion des Immunsystems: Da Experimente im Labor teuer und zeitaufwändig sind, ist es verständlich, dass sich gerade bei neuen oder unkonventionellen Theorien über die Funktion des Immunsystems nicht immer ein Labor oder ein Geldgeber für eine experimentelle Überprüfung findet. Die Immunologie ist hier besonders geprägt durch die Erfahrung mit der Netzwerk-Theorie von Niels Jerne [Jer74], einem der bedeutendsten Immunologen des 20. Jahrhunderts. Für die Netzwerk-Theorie erhielt Jerne 1984 den Nobelpreis für Medizin. Allerdings konnte diese Theorie experimentell nie nachge-

wiesen werden, obwohl dies nicht zuletzt wegen Jernes großem Ansehen jahrelang von vielen Forschern versucht wurde. Nach etwa zehn Jahren intensiver und fruchtloser Forschungsaktivitäten stellte sich schließlich heraus, dass der in der Netzwerk-Theorie postulierte Effekt zwar in gerigem Maße nachweisbar ist, für die Entwicklung der Immunantwort aber praktisch keine Bedeutung hat. Diese Jahre nennt man heute *the dark age of immunology*. Dabei war Jerne zuvor von Max Delbrück, ebenfalls Nobelpreisträger für Medizin, mit folgendem Satz eindringlich gewarnt worden [Sö03]:

We have both seen the grand idolatry of intuition run his course. If anything, this is of the Devil.

Die ohnehin schon große Skepsis der experimentellen Immunologen gegenüber neuen, unkonventionellen Theorien nahm durch diese Erfahrungen noch zu. Dies führte zu einem höheren Druck auf theoretische Immunologen, ihre Hypothesen schlüssig zu belegen.

Die mathematische oder informatische Modellierung solcher Hypothesen liefert Möglichkeiten, sie durch formale Analysen und Beweise oder durch Computersimulation zu stützen. Zwar ist ein Computermodell einer neuen Theorie noch lange kein Beweis für ihre Richtigkeit (auch für die Netzwerktheorie existieren zahlreiche Computermodelle). Es ermöglicht aber eine exaktere Beschreibung und Untersuchung der postulierten Mechanismen. Zudem kann unter Umständen auf frühere Erfahrungen mit ähnlichen, experimentell nachweisbaren Modellen zurückgegriffen werden.

Im Literaturverzeichnis finden sich einige Referenzen auf derart motivierte Modelle [CCFS96, PW92, BSP92, HMR96]. Das Modell von Carneiro et al. [CCFS96] wird in Abschnitt 2.2.2 genauer vorgestellt.

Experimente *in machina*: Eine ehrgeizigere Idee ist die Abbildung des Immunsystems im Computer in solcher Genauigkeit, dass die Durchführung von Experimenten direkt anhand der Simulation möglich wird. Solche *in machina* Experimente könnten herkömmliche *in vitro* und *in vivo* Experimente vermutlich nie ersetzen, wohl aber bei deren Vorbereitung, Planung und Interpretation helfen. Die Vorteile lägen auf der Hand: Bei kostengünstigerer Durchführung wären alle erdenklichen Messwerte sofort zugreifbar und könnten auf nahezu beliebige Art visualisiert und ausgewertet werden.

Momentan liegt ein derart leistungsfähiges Modell allerdings noch in weiter Ferne. Das bekannteste agentenbasierte Modell des Immunsystems, das in Abschnitt 2.1.1 vorgestellte Celada-Seiden-Modell [SC92, PKSC02], wurde mit dem Ziel entwickelt, *in machina* Experimente zu ermöglichen. Es stellt noch einen zu kleinen Teil des Immunsystems dar, um universell nutzbar zu sein, wurde aber zum Beispiel schon zur Untersuchung der Effektivität von Grippeimpfungsstrategien verwendet [SFAP99].

Untersuchung der Evolution des Immunsystems: Viele Teilsysteme des Immunsystems, beispielsweise das Komplementsystem, bestehen aus einer Vielzahl miteinander interagierender und voneinander abhängiger Komponenten. Bis heute ist unklar, wie diese Systeme durch Evolution entstanden sein könnten. So griff Michael Behe, ein führender Vertreter der Intelligent-Design-Hypothese, das Komplementsystem als ein Argument gegen die Darwinsche Evolutionstheorie auf [Beh98].

Computermodelle können helfen, neue Einsichten über die Evolution des Immunsystems zu gewinnen. Anhand von Simulationen kann untersucht werden, inwieweit das Immunsystem bei Entfernung bestimmter Teilsysteme prinzipiell funktionsfähig bleiben würde. Da Hypothesen über die Entstehung der Immunabwehr in prähistorischen Organismen nicht direkt belegt werden können, ist die Modellierung im Computer hier besonders verbreitet.

Einsatz in der Lehre: Durch Erfahrungen mit dem Celada-Seiden-Modell wurde deutlich, dass sich Computermodelle des Immunsystems gut in der Lehre verwenden lassen. Auch hier kommen sie nur ergänzend zu den klassischen Lehrmethoden zum Einsatz, können aber durch ihre relative Einfachheit und gute Visualisierungsmöglichkeiten für ein besseres Verständnis der Vorgänge im Immunsystem hilfreich sein.

1.1.2. Schwerpunkte und Vereinfachungen

Keins der im Rahmen dieser Arbeit untersuchten Modelle erhebt den Anspruch, alle Organe des Immunsystems wie Thymus, Knochenmark, Milz und Lymphknoten inklusive deren räumlicher Anordnung im Körper oder gar der Interaktion mit anderen Systemen wie dem Blutkreislauf abbilden zu können. Im Gegenteil enthalten viele Modelle gar keinen Begriff des Raums, sondern geben nur quantitative Größen auf globaler Ebene an (Anzahl an T-Zellen, Antigen-Konzentration und ähnliches). Bei agentenbasierten Modellen wird als einfacher Raumbegriff ein Gitter eingesetzt, um für die Interaktionen der Zellen und Moleküle einen Begriff von Lokalität und Nachbarschaft zur Verfügung zu stellen.

Das Fehlen räumlicher Differenzierung liegt unter anderem darin begründet, dass sich die untersuchten Modelle auf Teilaspekte des Immunsystems konzentrieren. Sie bilden vor allem Vorgänge ab, die im Lymphknoten, insbesondere im Keimzentrum, stattfinden. Die Unterteilung des Keimzentrums in funktional verschiedene Zonen (dark zone, light zone, Mantelzone etc.) wird hier weggelassen.

Andere, hier nicht betrachtete Modelle, vor allem aus dem Gebiet der Computerimmunologie [FHS97, FPAC94, Hof99], konzentrieren sich auf die Entwicklung der T-Zell-Rezeptordiversität durch negative Selektion im Thymus. Auch hier werden räumliche Aspekte vernachlässigt.

Eine Konsequenz dieser Vereinfachungen ist, dass die mit den Modellen erhaltenen Werte meist keine direkte Entsprechung zur Realität haben, sondern nur qualitativ interpretiert

werden können.

1.2. Ziel der Diplomarbeit

Das bekannteste Modell zur Simulation des Immunsystems im Computer ist das probabilistische Automatenmodell von Celada und Seiden [SC92], das von den Autoren als “verallgemeinerter zellulärer Automat” bezeichnet wird. Ziel dieser Diplomarbeit war es, dieses Modell aus mathematischer und automatentheoretischer Sicht zu analysieren und zu charakterisieren. Eine Untersuchung dieses Modells mit formalen Mitteln wurde bislang nicht durchgeführt, wohl auch aus dem Grund, dass zelluläre Automaten als besonders schwer analysierbar gelten.

Durch diese Untersuchung sollte geklärt werden, wie das Modell aus informatischer Sicht einzuordnen ist und ob es Verwandtschaften zu anderen, besser analysierten und verstandenen Modelltypen gibt. Weiterhin sollten Ansätze zur Verbesserung des Modells herausgearbeitet werden.

Abschließend sollte eine Implementierung des veränderten oder erweiterten Modells erstellt werden, mit der die vorgenommenen Veränderungen getestet und evaluiert werden können.

1.3. Aufbau der Diplomarbeit

Im nächsten Kapitel (2) werden die Grundlagen dieser Diplomarbeit zusammengefasst. Dabei werden die wichtigsten Modelle des Immunsystems erläutert und ein verwandter Modelltyp aus der Chemie, das Reaktions-Diffusions-System, wird vorgestellt. In Kapitel 3 wird das Celada-Seiden-Modell mit mathematischen Mitteln analysiert und charakterisiert. Anhand der Ergebnisse dieser Analyse werden in Kapitel 4 Ansätze zur Verbesserung des Celada-Seiden-Modells entwickelt. Die Resultate der am verbesserten Modell durchgeführten Experimente werden in Kapitel 5 vorgestellt. In Kapitel 6 werden die Ergebnisse der Diplomarbeit zusammengefasst und Perspektiven für weitere Forschung aufgezeigt.

2. Grundlagen

2.1. Agentenbasierte Modellierung des Immunsystems

Die agentenbasierte Modellierung des Immunsystems ist gegenüber der in Abschnitt 2.2 behandelten Modellierung durch Differentialgleichungen weniger verbreitet und erst seit einiger Zeit überhaupt durchführbar. Im Vergleich zu den recht abstrakten Differentialgleichungssystemen sind agentenbasierte Modelle jedoch leichter verständlich, da sie eine möglichst direkte Abbildung und Simulation der immunologischen Vorgänge anstreben.

Ein agentenbasiertes Modell ist ein System ohne zentrale Kontrolle, dessen Verhalten nur von *lokalen* Interaktionen seiner Komponenten, der *Agenten*, abhängt. Beobachtet man das Verhalten eines Agentensystems auf globaler Ebene, spricht man von *emergentem Verhalten*. Der Zusammenhang zwischen emergentem Verhalten und lokalen Interaktionen kann hochkomplex sein [Wol02]. Bei der Modellierung mikroskopischer Vorgänge werden die Agenten auch als *Partikel* bezeichnet.

Da das Immunsystem ebenfalls ein verteiltes System ohne zentrale Kontrolle ist, ist die Modellierung durch ein agentenbasiertes System naheliegend. Die Agenten sind hier die einzelnen Zellen und Moleküle, aus denen sich das Immunsystem zusammensetzt. Für diese Partikel werden Interaktionsregeln formuliert, die die tatsächlichen immunologischen Vorgänge möglichst genau wiedergeben.

Das bis heute vollständigste Agentenmodell des Immunsystems wurde bereits im Jahr 1992 von dem Physiker Philip Seiden in Kooperation mit dem praktizierenden Immunologen Franco Celada entworfen [SC92]. Dieses Modell wurde seitdem unter dem Namen *ImmSim* kontinuierlich weiterentwickelt [PKSC02]. Das Celada-Seiden-Modell wurde als Schwerpunkt dieser Arbeit gewählt, weil es das vollständigste, bei weitem bekannteste und meist zitierte agentenbasierte Modell des Immunsystems ist. Die weiteren dem Autor bekannten agentenbasierten Modelle [MS01, SFAP99, Das98] sind Variationen, Verallgemeinerungen oder Verfeinerungen dieses Modells.

2.1.1. Das Celada-Seiden-Modell

Im Celada-Seiden-Modell (CS-Modell) werden die Interaktionen zwischen Antigen, Antigenpräsentierenden Zellen, T- und B-Zellen und Antikörper in einem Lymphknoten dargestellt.

Das Modell umfasst die folgenden Aspekte der Immunantwort:

- Unspezifische Antigenerkennung und Antigenpräsentation durch antigenpräsentierende Zellen (APCs)
- Aktivierung der spezifischen Abwehr durch Interaktion und Kostimulation der APCs, T- und B-Zellen
- Proliferation und Differentiation der B- und T-Zellklone, somatische Hypermutation
- Sekretion von Antikörper, Bildung von Antigen-Antikörper-Komplexen (Immunkomplexe)
- Zerstörung der Immunkomplexe durch Phagozytose
- Abklingen der Immunantwort
- Bildung des Immungedächtnisses durch B- und T-Gedächtniszellen

In der Realität finden diese Vorgänge an verschiedenen, teils weit auseinanderliegenden Stellen im Organismus statt: Die direkte Interaktion mit dem Antigen (Erkennung, Antikörperbindung, Neutralisierung) erfolgt am Infektionsherd, während alle anderen Vorgänge im Lymphknoten, vor allem in den Keimzentren, stattfinden. Im CS-Modell dagegen teilen sich alle Partikel einen gemeinsamen, nicht differenzierten Raum.

Aufbau des Modells

Der grundsätzliche Aufbau des CS-Modells ähnelt dem eines zellulären Automaten:

1. Der Simulationsraum ist ein diskretes Gitter
2. Die Simulation ist eine Serie diskreter Zeitschritte
3. Auf den Feldern des Gitters befinden sich Partikel aus einer endlichen Menge verschiedener Partikeltypen (B-Zellen, T-Zellen, Antigen, etc.)
4. Der Inhalt jedes Feldes entwickelt sich nach denselben probabilistischen Regeln
5. Die Regeln für die Entwicklung eines Feldes hängen nur von Inhalt dieses Feldes ab
6. Zwischen den Entwicklungsschritten bewegen sich die Partikel von Feld zu Feld (Diffusion)

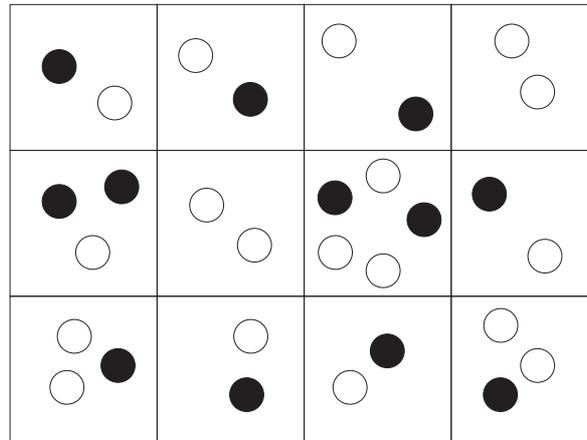


Abbildung 2.1.: Schematischer Aufbau des CS-Modells: Auf den Feldern des Gitters befinden sich Partikel verschiedener Arten, die miteinander interagieren.

Abbildung 2.1 zeigt schematisch den Aufbau des CS-Modells. Der hauptsächliche Unterschied zu herkömmlichen zellulären Automaten besteht darin, dass die Felder des Gitters beliebig viele Partikel aufnehmen können. Nur die Anzahl verschiedener Partikelarten ist endlich. Im Gegensatz dazu ist bei einem herkömmlichen zellulären Automaten die Menge der Zustände, in denen sich ein Feld befinden kann, endlich.

Die Bewegung der Partikel auf dem Gitter zwischen den Entwicklungsschritten wird realisiert, indem in jedem Schritt für jedes Partikel zufällig entschieden wird, ob es auf seinem Ursprungsfeld verbleibt oder auf eines der Nachbarfelder diffundiert. Diese Art der Diffusion wird in Kapitel 3 genauer untersucht.

Die zu simulierenden immunologischen Vorgänge werden als Regeln formuliert, nach denen sich die Partikel auf den Feldern gegenseitig beeinflussen und entwickeln. Durch diese Regeln kann der Inhalt eines Feldes auf verschiedene Weise verändert werden. Es können neue Partikel entstehen, etwa durch Zellteilung, oder vorhandene Partikel getötet werden, zum Beispiel durch Phagozytose. Viele Interaktionen führen auch dazu, dass Partikel ihren internen Zustand ändern, zum Beispiel von einer naiven B-Zelle zu einer B-Zelle, die Antigen erkannt hat.

Affinität

Wie im wirklichen Immunsystem hängt der Verlauf vieler Interaktionen im CS-Modell von der *Affinität* zwischen den beteiligten Partikeln ab. Ein Beispiel ist die Antigenerkennung durch B-Zellen. Der B-Zell-Rezeptor wird als hochaffin zu einem Antigen bezeichnet, wenn seine Form so gut zur Form des Antigens passt, dass eine Bindung des Antigens an den B-Zell-Rezeptor erfolgen kann (Schlüssel-Schloss-Prinzip, vergleiche Abbildung 2.2).

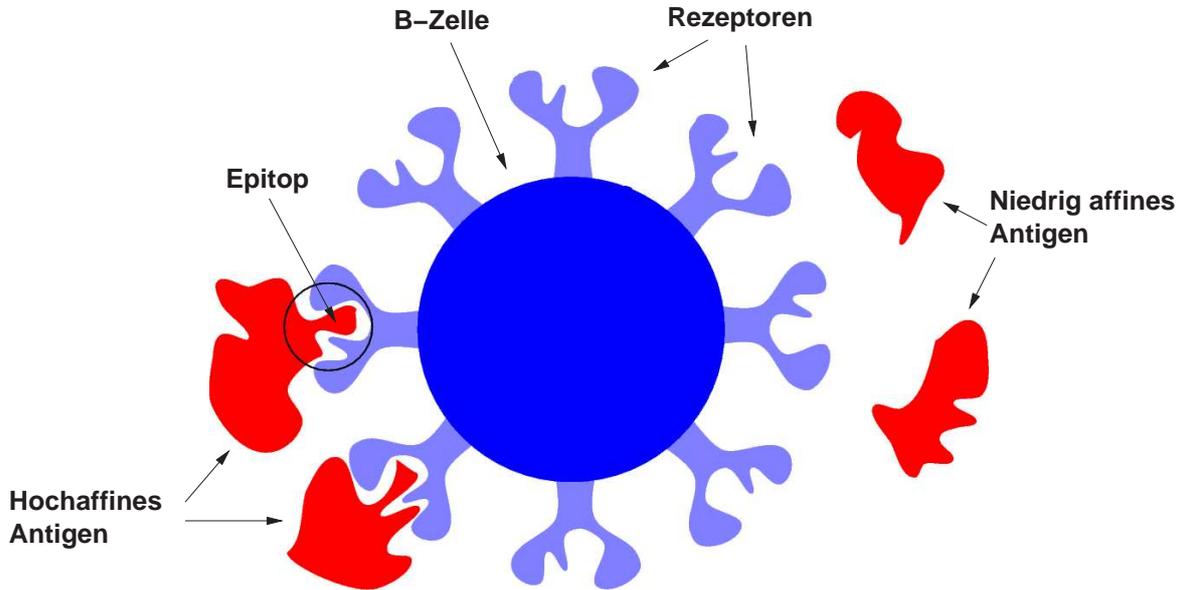


Abbildung 2.2.: Schlüssel-Schloss-Prinzip bei der Antigenerkennung durch B-Zellen: Links dargestellt hochaffines Antigen, das eine Form aufweist, die gut zum B-Zell-Rezeptor passt (Epitop). Rechts niedrig affines Antigen, das kein Epitop aufweist, das zum B-Zell-Rezeptor passt. Grafik entnommen aus [Hof01]

Im CS-Modell werden der Rezeptor und das Epitop als Bitstrings fester Länge modelliert (vergleiche Abbildungen 2.4 und 2.5). Eine B-Zelle kann an ein Antigen binden, wenn beide sich auf dem gleichen Gitterfeld befinden. Die Bindung findet nicht mit einer a priori festgelegten Wahrscheinlichkeit statt, sondern wird zur Laufzeit als Funktion der Affinität zwischen dem Rezeptor der B-Zelle und dem Epitop des Antigens ermittelt. Die Affinität ist eine Abstandsfunktion, die für identische Bitstrings minimal und für komplementäre Bitstrings maximal ist. Für $r, e \in \{0, 1\}^N$ ist die Affinitätsfunktion wie folgt definiert (d_h bezeichnet hier den Hamming-Abstand):

$$a(r, e) = f(d_h(r, e))$$

$$\begin{aligned} f(n_{\min}) &= \alpha_{\min} \\ f(n) &= 0 && \text{für } n < n_{\min} \\ f(n) &= \min\left(1, \alpha_{\text{inc}} f(n-1) \frac{n}{N-n+1}\right) && \text{für } n > n_{\min} \end{aligned}$$

Die Wahrscheinlichkeit der Bindung des Antigens mit Epitop e an die B-Zelle mit Rezeptor r ist im CS-Modell proportional zur Affinität zwischen e und r . Abbildung 2.3 zeigt die in der Implementierung von Castiglione [BC01] verwendete Affinitätsfunktion mit $N = 12$, $n_{\min} = 9$, $\alpha_{\min} = 0.05$, $\alpha_{\text{inc}} = 2.5$.

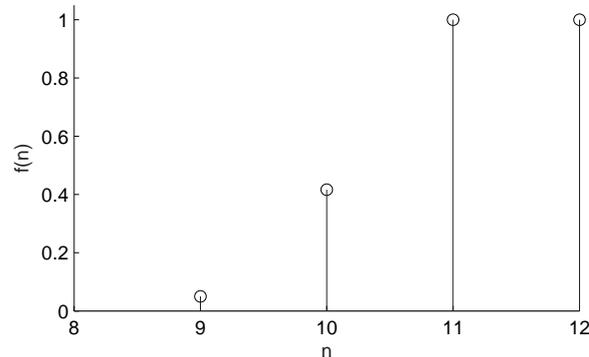


Abbildung 2.3.: Affinitätsfunktion aus der Implementierung des CS-Modells von Castiglione [BC01] mit $N = 12$, $n_{\min} = 9$, $\alpha_{\min} = 0.05$, $\alpha_{\text{inc}} = 2.5$

Für den Vorgang der Bindung eines Antigenpeptids an den MHC zur Antigenpräsentation auf der Oberfläche einer APC oder B-Zelle (siehe Abbildung 2.6, mittlere Grafik) wird eine andere Affinitätsfunktion verwendet. Diese gibt die Tatsache wieder, dass der MHC an sehr viele Antigenpeptide binden kann, während die B-Zell-Rezeptoren hochspezifisch sind. Die Affinitätsfunktion lautet hier:

$$a(p, mhc) = \left(\frac{1}{2}\right)^{d_h(p, mhc)}$$

Modellierte Moleküle und Zellen

Im CS-Modell wird zwischen molekularen und zellulären Partikeln unterschieden. Der Unterschied liegt dabei darin, dass Zellen einen internen Zustand haben, während Moleküle zustandslos sind. Es gibt drei verschiedene Arten molekularer Partikel:

- *Antigen (Ag)* besteht aus Epitopen und Peptiden, die als Bitstrings modelliert werden (siehe Abbildung 2.4).¹
- *Antikörper (Ab)* ist durch seine Form (Paratop) charakterisiert, die der Form des Rezeptors der B-Zelle entspricht, die den Antikörper hergestellt hat.
- *Immunkomplexe (IC)* entstehen durch die Bindung von Antigen an Antikörper.

Die zellulären Partikel des CS-Modells gliedern sich in vier Arten (siehe Abbildung 2.5):²

¹Neuere Versionen des CS-Modells kennen auch komplexere, zustandsbehaftete Antigene wie Krebszellen und das HI-Virus

²Freunde immunologischer Lehrbücher mögen dem Autor die etwas klobige Darstellung der Zellen in dieser Abbildung verzeihen. Sie ist schon ein leichter Fortschritt gegenüber der Quelle, aber in der Informatik werden eckige Formen bevorzugt.

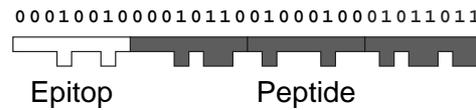


Abbildung 2.4.: Schematische Darstellung eines Antigenes im CS-Modell. Nach [SC92]

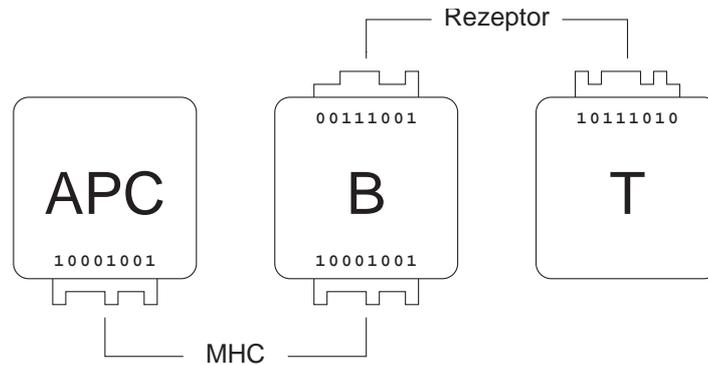


Abbildung 2.5.: Schematische Darstellung der zellulären Partikel im CS-Modell. Nach [SC92]

- *B-Zellen* (*B*) werden durch einen individuellen Rezeptor und durch das eventuell an den MHC gebundene Antigenpeptid charakterisiert. Ein Zustandsbit markiert, ob die B-Zelle eine Gedächtniszelle ist.
- *Plasmazellen* (in Abbildung 2.5 nicht dargestellt) sezernieren Antikörper. Sie entstehen aus proliferierenden B-Zellen und verlieren dabei ihre Fähigkeit, an Antigen zu binden.
- *T-Zellen* (*T*) besitzen einen Rezeptor zur Erkennung von MHC-Peptid-Komplexen. Sie dienen ausschließlich der Regulation der Immunantwort und interagieren nicht direkt mit Antigen. Ein Zustandsbit markiert, ob die T-Zelle eine Gedächtniszelle ist.³
- *Antigenpräsentierende Zellen* (*APC*) binden hochaffin an Immunkomplexe und niedrig affin an Antigen. Sie übernehmen die Aufgaben der unspezifischen Antigenerkennung und der Phagozytose der Immunkomplexe. APCs, die Antigen phagozytiert haben, präsentieren einen Teil des Peptids zusammen mit dem MHC.

Lebenszyklus der Partikel

Im Gegensatz zu einem herkömmlichen zellulären Automaten ist das CS-Modell kein geschlossenes System. In jedem Simulationsschritt wird ein Teil der zellulären und molekularen Partikel entfernt und dadurch der Zelltod bzw. die Molekülersetzung nachgebildet. Die

³Mit "T-Zellen" sind also eigentlich die T-Helferzellen (Th2) gemeint. Andere T-Zellen wie T-Killerzellen oder T-Supressorzellen kommen im ursprünglichen CS-Modell noch nicht vor.

Wahrscheinlichkeit, dass ein Partikel eines gegebenen Typs während eines Simulationsschrittes stirbt, ergibt sich wie folgt aus der mittleren Lebenszeit τ dieses Typs:

$$\Pr[\text{Tod}] = \frac{1}{\tau}$$

Es wird nur der Lebenszyklus von B- und T-Zellen modelliert. Da APCs keine individuell spezifischen Attribute besitzen, wird ihre Lebensdauer vereinfachend als unendlich groß angenommen. Dadurch bleibt die Population an APCs über die gesamte Simulation konstant.

Neue Partikel werden aus *peripheren Komponenten* in das CS-Modell eingefügt oder innerhalb des Modells durch Zellteilung oder Sezernierung (Antikörper) gebildet. Das CS-Modell kennt drei periphere Komponenten: Aus dem *Thymus* werden neue T-Zellen in das Modell eingefügt, aus dem *Knochenmark* stammen neue B-Zellen. Die Vorgänge innerhalb Thymus und Knochenmark werden dabei nicht mitmodelliert. Die Bildung neuer Zellen pro Zeitschritt soll dabei im Mittel gerade den Verlusten durch normalen Zelltod entgegenwirken. Die Anzahl n der neuen Zellen eines Typs pro Zeitschritt ist eine Funktion der mittleren Lebenszeit τ und der initialen Populationsgröße I :

$$n = I \frac{1}{\tau}$$

Die dritte periphere Komponente des CS-Modells ist die Umwelt. Aus ihr stammt das Antigen, das in den virtuellen Lymphknoten "injiziert" wird. In einem Simulationslauf wird typischerweise mindestens zwei Mal Antigen injiziert, um den Effekt des Immungedächtnisses nachzuweisen.

Spezifische Antigenerkennung

Der Vorgang der spezifischen Antigenerkennung ist der Modellierungsschwerpunkt des CS-Modells. Die Grundidee ist die Modellierung von Proteinformen als Bitstrings in Kombination mit dem oben erläuterten Begriff der Affinität. Die spezifische Antigenerkennung besteht aus drei Teilschritten, die in Abbildung 2.6 dargestellt sind.

Trifft eine B-Zelle auf ein zu ihrem Rezeptor affines Antigen, kann die B-Zelle das Antigen mit hoher Wahrscheinlichkeit phagozytieren (linke Grafik). Dabei wird das Antigen internalisiert und das Peptid in zwei Fragmente zerteilt. Es wird versucht, eines der Fragmente an die linke oder rechte Hälfte des MHC-Strings zu binden, wobei die günstigere Affinitätsfunktion für den MHC verwendet wird. Gelingt die Bindung, wird das andere Fragment auf dem MHC nach außen sichtbar (Antigenpräsentation, mittlere Grafik).

Die B-Zelle muss sich nun von einer T-Zelle bestätigen lassen, dass es sich bei dem Peptidfrag-

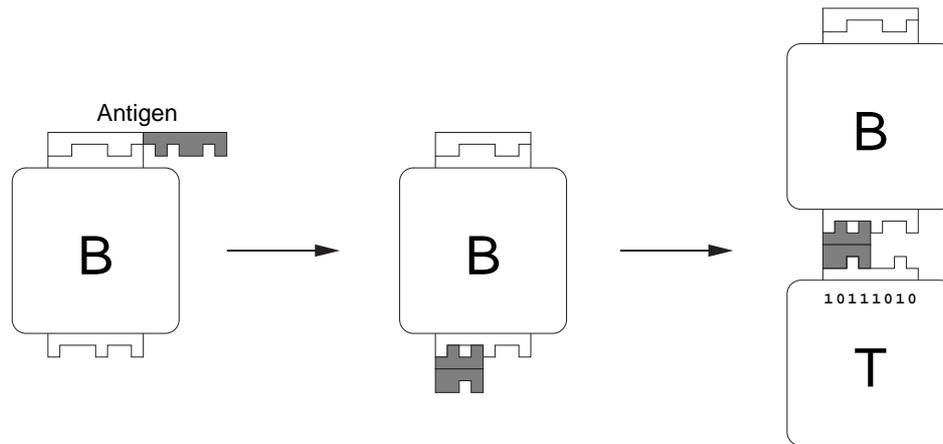


Abbildung 2.6.: Antigenerkennung im CS-Modell. Nach [Han01]

ment tatsächlich um ein fremdes und kein körpereigenes Protein handelt. Die T-Zellen wurden im Thymus trainiert, auf körpereigene Proteine *nicht* zu reagieren (negative Selektion). Dies schützt den Organismus vor seinem eigenen Immunsystem (*self tolerance*). Im CS-Modell erfolgt die Antigenerkennung durch die T-Zelle wieder über die Affinität des präsentierten Peptidfragments zum Rezeptor der T-Zelle (rechte Grafik). Es kommt also erst dann zu einer Immunantwort, wenn sowohl mindestens eine B-Zelle *und* eine T-Zelle das Antigen erkannt haben.

Jeder Teilschritt der Antigenpräsentation ist ein elementarer Simulationsschritt. Der komplette Vorgang sollte daher mindestens 3 Zeitschritte dauern, jedoch ist aufgrund der hohen Spezifität der Rezeptoren die Zahl der B-Zellen, die ein Antigen potenziell überhaupt erkennen können, sehr gering (abhängig von der Länge der Bitstrings). Es kann unter Umständen recht lange dauern, bis eine affine B-Zelle das Antigen erkannt und verarbeitet hat und die Hilfe einer affinen T-Zelle erhält. Dies ist auch im wirklichen Immunsystem der Fall und ist ein Grund für die relativ lange Zeit, bis eine Immunantwort erfolgt (*lag time*).

Ablauf der Simulation

Die übrigen Interaktionsmöglichkeiten des CS-Modells sollen hier nicht detailliert behandelt werden. In Abbildung 2.7 sind schematisch alle Interaktionswege dargestellt: Antigen und Antikörper können Immunkomplexe bilden, die hochaffin von APCs phagozytiert werden. B-Zellen und APCs phagozytieren Antigen und präsentieren es den T-Zellen. Haben eine T-Zelle und eine B-Zelle oder eine APC dasselbe Antigen erkannt, werden die T-Zelle und die B-Zelle zur Proliferation angeregt.

Nachdem die B-Zellen eine Reihe von Proliferationszyklen durchlaufen haben, entwickelt sich

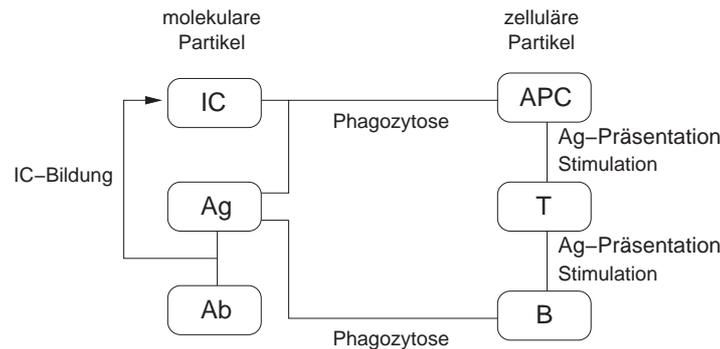


Abbildung 2.7.: Interaktionsmöglichkeiten im CS-Modell

ein Teil zu Plasmazellen, die dann Antikörper sezernieren. Die Form des Antikörpers entspricht genau der des B-Zell-Rezeptors, wodurch der Antikörper hochaffin an das Antigen binden kann. Durch diese Bindung wird das Antigen als gefährlich markiert (*Opsonierung*). Antigen, welches auf diese Weise mit Antikörper markiert wurde, wird von den unspezifischen Fresszellen phagozytiert.

Abbildung 2.8 zeigt, aus welchen Phasen ein Simulationsschritt besteht. Alle Phasen finden lokal auf den Feldern des Gitters und ohne Einfluss aus der Nachbarschaft statt. Nur in der Diffusionsphase wandern Zellen und Moleküle durch die Zufallsbewegung auf eines der Nachbarfelder und sorgen so für indirekten Informationsaustausch mit der Nachbarschaft. Im CS-Modell wird eine 6-Nachbarschaft verwendet.

Konfliktbehandlung

Aus Abbildung 2.7 wird deutlich, dass es für einige Partikelarten innerhalb eines Zeitschritts mehrere Interaktionsmöglichkeiten gibt. APCs könnten zum Beispiel sowohl einen Immunkomplex als auch ein Antigen phagozytieren.

Im CS-Modell werden diese Konflikte wie folgt behandelt: In jedem Zeitschritt wird die Liste der Interaktionsmöglichkeiten zufällig permutiert und dann in dieser Reihenfolge abgearbeitet. Dabei darf jede Interaktionsmöglichkeit von jedem Partikel höchstens ein Mal genutzt werden. Eine APC darf also pro Zeitschritt höchstens einen Immunkomplex oder ein Antigen phagozytieren. Ob ein Antigen mit einem Antikörper einen Immunkomplex bildet oder von einer B-Zelle phagozytiert wird, hängt davon ab, welche dieser Interaktionsmöglichkeiten zuerst betrachtet wird.

Die Reihenfolge, in der die Partikel selbst betrachtet werden, wird ebenfalls in jedem Schritt zufällig permutiert. Dadurch ist bei Konkurrenzsituationen - zum Beispiel wenn viele Antikörper nur wenigen affinen B-Zellen gegenüberstehen - sichergestellt, dass kein Partikel auf

Dauer eine höhere Interaktionswahrscheinlichkeit hat als ein anderes.

Um die Interaktionswege weiter zu entzerren, werden zuerst alle Zell-Zell-Interaktionen und dann die Zell-Molekül-Interaktionen behandelt.

2.1.2. CImmSim

Eine aktuelle, frei verfügbare Implementierung des CS-Modells stammt von Filippo Castiglione und trägt den Namen *CImmSim* [BC01]. Gegenüber dem ursprünglichen CS-Modell wurde diese Implementierung um dendritische Zellen und T-Killerzellen erweitert. Weiterhin ermöglicht das Programm die Simulation eines mutierenden HI-Virus und eines Epithelkarzinoms. Die in den vorherigen Abschnitten erläuterten grundlegenden Eigenschaften des CS-Modells wurden aber beibehalten.

Die folgenden Abbildungen zeigen die Ergebnisse eines Testlaufs mit CImmSim. In Abbildung 2.9 sind die Populationsgrößen von B-Zellen, T-Zellen, Antigen und Antikörper dargestellt. Plots 1 und 2 zeigen die Ausbildung von B- und T-Gedächtniszellen. In Abbildung 2.10 werden zu verschiedenen Zeitpunkten der Simulation zusätzlich die genauen Verteilungen von Antigen und Antikörper auf dem 101 mal 101 Felder großen Gitter dargestellt.

Immunologische Interpretation

Bei der Interpretation der Plots und Grafiken ist zu beachten, dass den meisten Größen wegen der vereinfachten Struktur des CS-Modells keine direkte biologische Bedeutung zugeordnet werden kann. Die y -Werte geben die absoluten Populationsgrößen an und stellen laut Castiglione eine Implementierung der "Leukozytenformel" dar, die Referenzwerte für die Anzahl von Granulozyten und Lymphozyten pro Mikroliter Blut in einem gesunden Erwachsenen liefert.

Die Verwendung der Leukozytenformel ist allerdings fragwürdig, da das CS-Modell im Wesentlichen die Vorgänge in einem Keimzentrum abbildet, während die Leukozytenformel für die Zellen des Blutkreislaufs gilt. Im Blut ist die Menge an T-Zellen etwa 2-3mal so hoch wie die der B-Zellen, was qualitativ durch die Initialwerte von CImmSim (siehe Abbildung 2.9) wiedergegeben wird. Im Keimzentrum allerdings überwiegen bei weitem die B-Zellen mit einem Anteil von etwa 90 Prozent (vgl. [JTW02], S. 377 ff).

Der verhältnismäßig niedrige Anstieg der B-Zell-Population erklärt sich durch den geringen Anteil an für das Antigen spezifischen B-Zellen. Weiterhin darf im Modell eine B-Zelle nur vier Teilungszyklen durchlaufen. Somit entstehen pro aktivierter B-Zelle nur 16 Effektorzellen. In der Realität liegt die Zahl der Teilungszyklen bei etwa 1000.

Auf der Zeitskala weist das Modell dagegen sehr realistische Werte auf: Die Initialphase

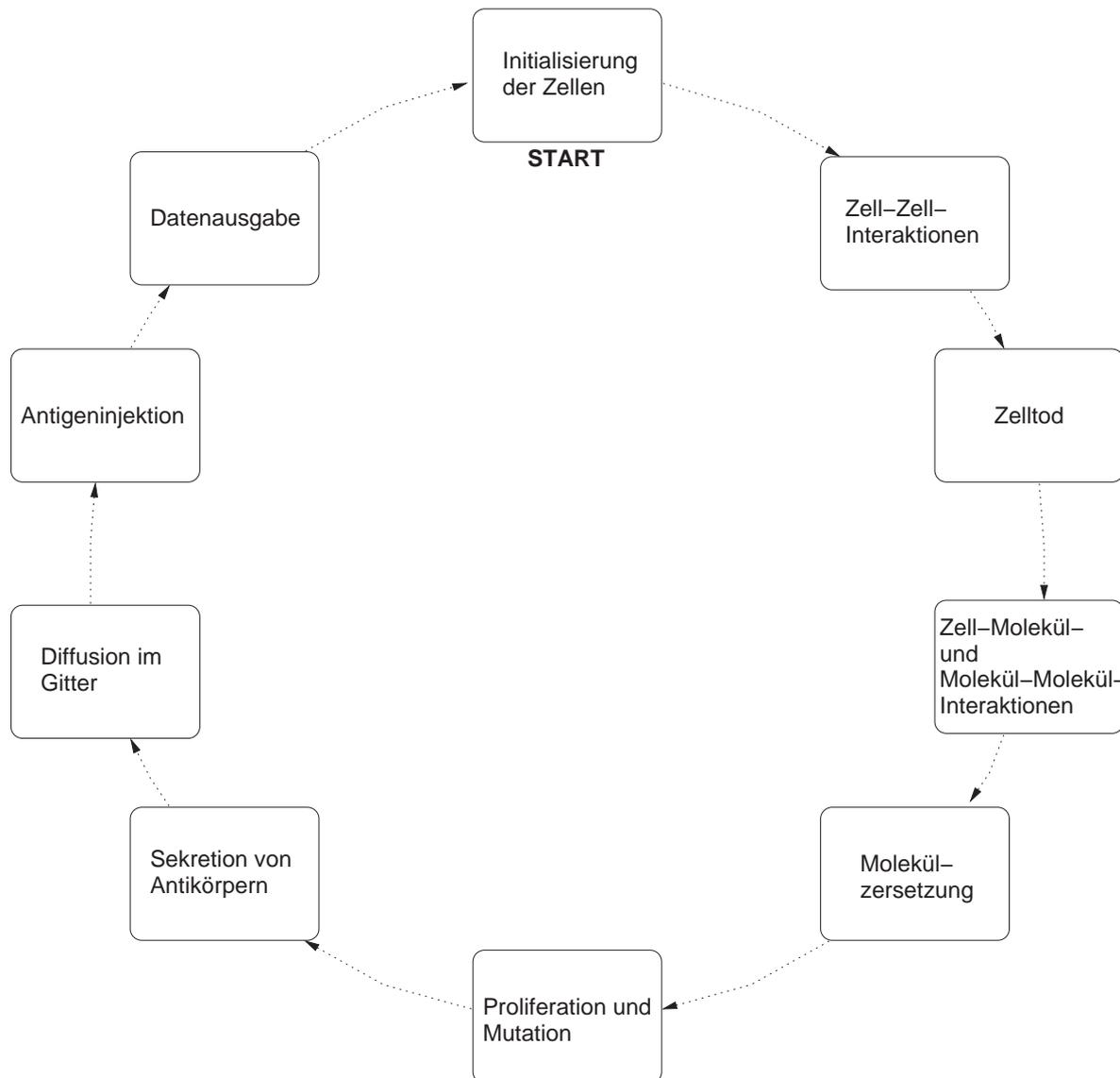


Abbildung 2.8.: Ein Simulationsschritt im CS-Modell. Aus [Han01]

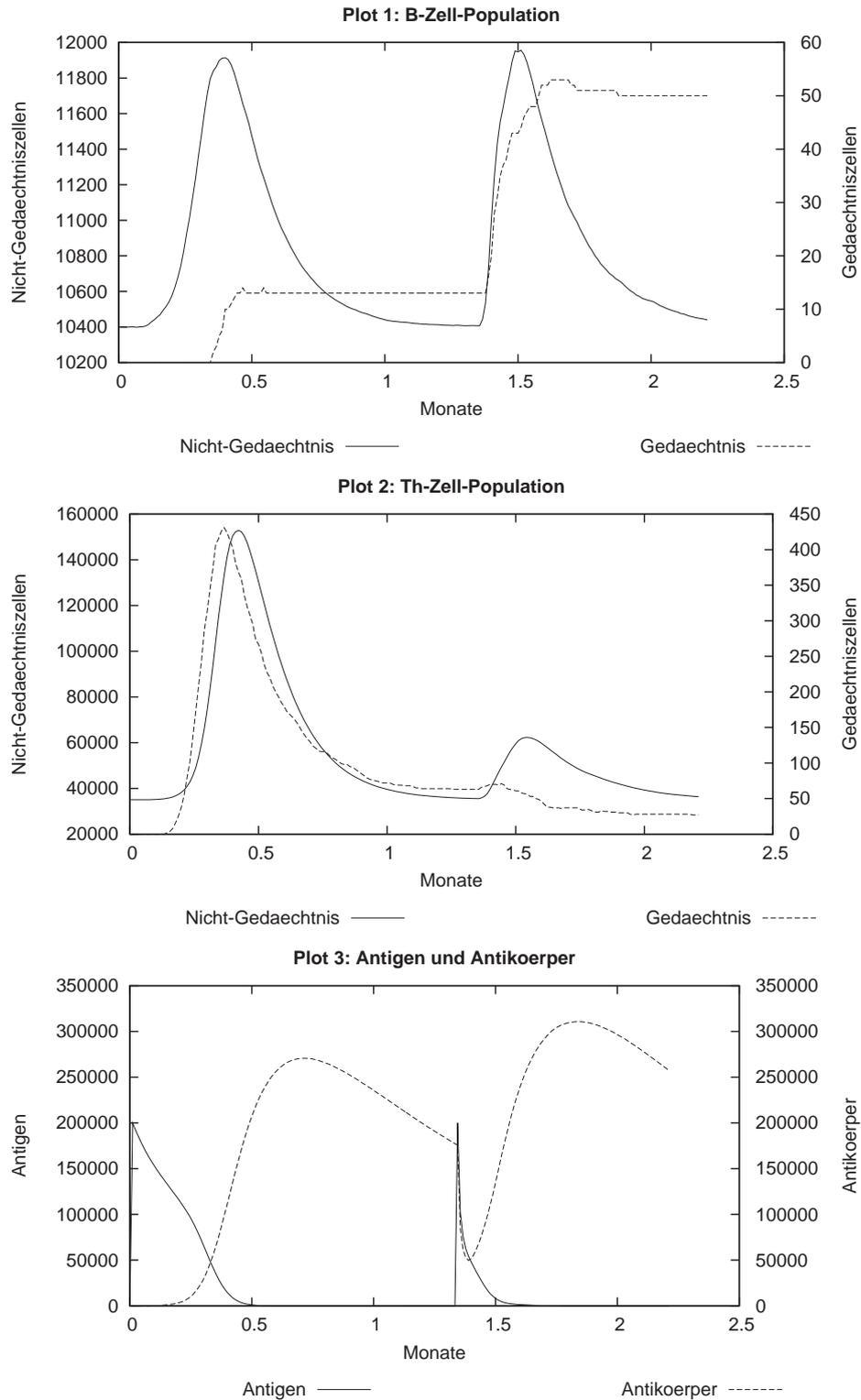


Abbildung 2.9.: Verlauf einer simulierten Immunantwort in CImmSim. Dargestellt sind die Populationsgrößen von B-Zellen (Plot 1), T-Zellen (Plot 2), Antigen und Antikörper (Plot 3).

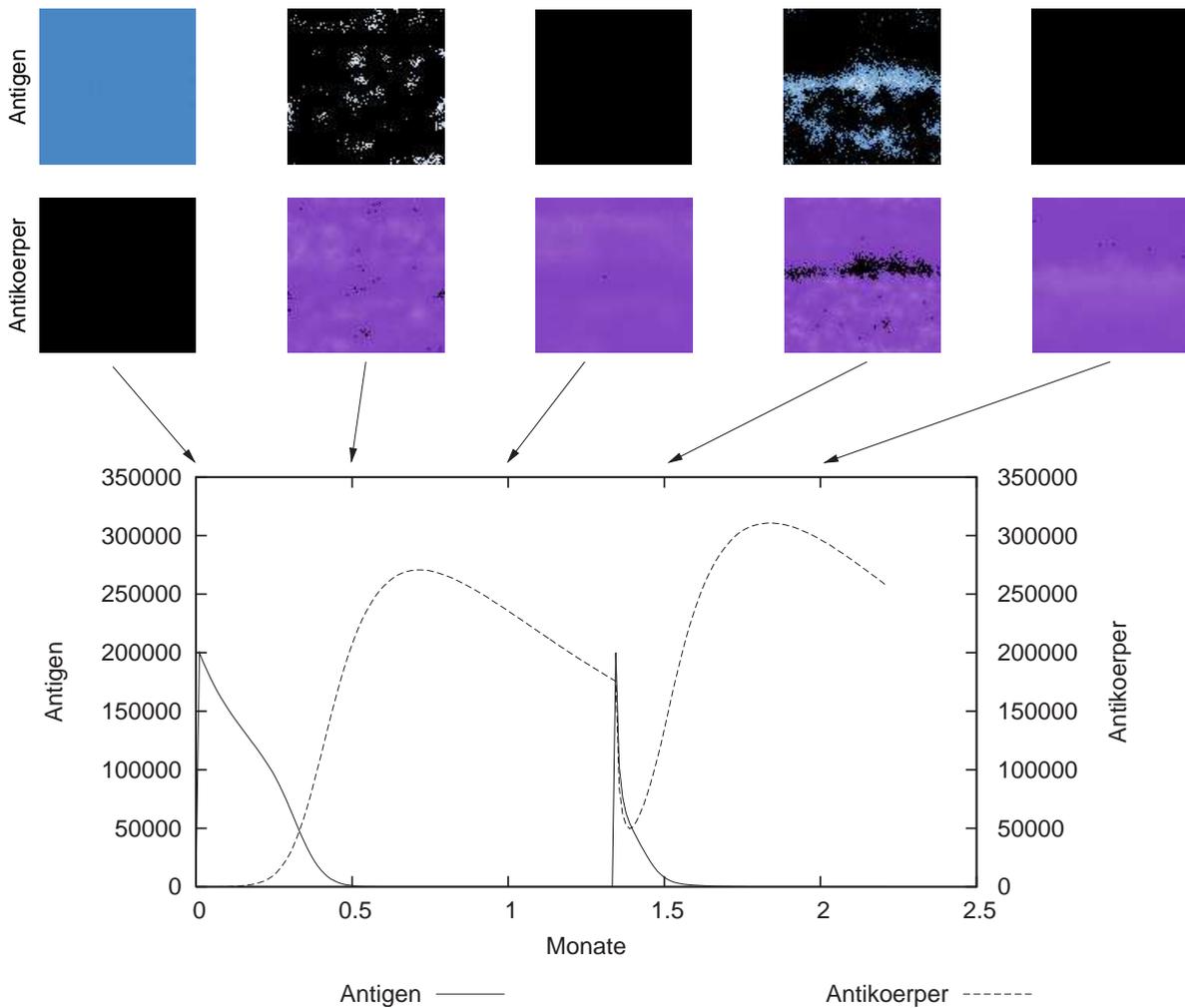


Abbildung 2.10.: Räumliche Verteilung von Antigenen und Antikörpern zu verschiedenen Zeitpunkten der in Abbildung 2.9 dargestellten Simulation. Eine hellere Färbung des Bildes bedeutet eine höhere Konzentration an Antigenen (obere Bildreihe) bzw. Antikörpern (untere Bildreihe).

bis zum Eintreten der Immunantwort dauert etwa eine halbe Woche. Der Peak des Antikörperspiegels ist nach etwa 3 Wochen erreicht. Diese Werte entsprechen einer typischen Immunantwort aus der Literatur ([JTW02], S. 22).

Gut beobachtbar ist auch der Effekt des Immungedächtnisses. Durch den immer noch vorhandenen Antikörper wird ein Großteil des Antigens direkt nach der zweiten Injektion gebunden. Trotzdem erfolgt eine stärkere und schnellere Immunantwort (höherer und steilerer Anstieg des Antikörper-Spiegels) als beim ersten Mal. Dies ist auf die beim ersten Antigenkontakt gebildeten Gedächtniszellen zurückzuführen.

2.1.3. Bedeutung des CS-Modells

Obwohl das CS-Modell (in der erläuterten Form) bereits 1992 veröffentlicht wurde und damit für immunologische Verhältnisse recht alt wirkt, ist es doch bis heute in seiner Bedeutung von keinem anderen agentenbasierten Modell des Immunsystems erreicht worden. So bildete es eine Grundlage für die Forschungen zu künstlichen Immunsystemen und zur Computerimmunologie, der Übertragung immunologischer Prinzipien auf das Gebiet der Computersicherheit [Das98, FHS97]. In der Literatur finden sich neben dem CS-Modell manchmal Hinweise auf SIMMUNE von Martin Meier-Schellersheim [MS01], das wie das CS-Modell mit dem Anspruch entwickelt wurde, das Immunsystem möglichst vollständig abzubilden, um dadurch eines Tages *in machina* Experimente zu ermöglichen. Es ist allerdings keine Implementierung von SIMMUNE bekannt.

Das CS-Modell wurde kontinuierlich weiterentwickelt und um weitere Zelltypen, wie dendritische Zellen und T-Killerzellen in CImmSim, ergänzt. Andere Verbesserungen konzentrierten sich auf technische Aspekte wie die Portierung nach C in CImmSim und die Parallelisierung [CBS97]. Dennoch ist es als Vertreter der agentenbasierten Modellierung gegenüber dem verbreiteteren Modelltyp, den Differentialgleichungen, insgesamt eher weniger bekannt.

2.2. Differentialgleichungsmodelle des Immunsystems

Differentialgleichungen sind bis heute der in der Immunologie bekannteste und verbreitetste Modelltyp. Dies liegt nicht nur daran, dass agentenbasierte Modelle in realistischer Größenordnung erst seit einiger Zeit überhaupt umsetzbar sind. Besonders auf dem Gebiet der formalen Analyse und Methodenkenntnis kann man mit Differentialgleichungen auf jahrhundertlange Erfahrung zurückblicken, während die formale Analyse agentenbasierter Systeme oder gar zellulärer Automaten noch vergleichsweise wenig entwickelt ist.

Differentialgleichungsmodelle setzen auf der Makroperspektive, dem globalen Verhalten, an. Auch hier bildet aber das lokale Verhalten immer die Grundlage für das Aufstellen der Glei-

chungen. Um aus dem zu modellierenden lokalen Verhalten Differentialgleichungen abzuleiten, bedient man sich der aus der Chemie bekannten *Massenwirkungsgesetze*. Diese ermöglichen es, Geschwindigkeitsgleichungen aufzustellen, die die Entwicklung der Zellen und Moleküle über die Zeit beschreiben.

Für diese Vorgehensweise lassen sich in der Literatur viele Beispiele finden [BSP92, PW92, CCFS96, HMR96], die sich durch die Zielsetzung der Modellierung und die Methodik der Untersuchung der Differentialgleichungen – analytisch oder numerisch – unterscheiden. Meist werden die Modelle benutzt, um den Einfluss bestimmter Parameter auf die Stabilität der Immunantwort oder des Immunsystems selbst zu untersuchen. Hier ist besonders die Fixpunktanalyse hilfreich, um Stabilitätsregionen zu ermitteln.

Im folgenden Abschnitt werden zunächst die Massenwirkungsgesetze eingeführt. In Abschnitt 2.2.2 wird dann ein vergleichsweise einfaches Differentialgleichungsmodell von Carneiro et al. detailliert vorgestellt und analysiert.

2.2.1. Die Massenwirkungsgesetze

Die Massenwirkungsgesetze dienen der Beschreibung chemischer Reaktionen, die wir ebenfalls als lokal gesteuerte Prozesse ohne zentrale Kontrolle auffassen können. Sie sind begründet durch die Tatsache, dass bei chemischen Reaktionen meist sehr große Mengen von Atomen und Molekülen beteiligt sind. Ausserdem wird angenommen, dass die reagierenden Stoffe sich frei im Raum bewegen, wie es etwa in einer Flüssigkeit oder in einem Gas der Fall ist, so dass sich eventuelle Konzentrationsunterschiede durch Diffusion ausgleichen. Da all diese Annahmen auch in den meisten Modellen des Immunsystems getroffen werden, werden die Überlegungen, die zu den Massenwirkungsgesetzen führen, auch dort angewendet. Dies ist so gängig geworden, dass es in der immunologischen Literatur meist nur noch kurz erwähnt wird. Es ist daher für das Verständnis der Differentialgleichungsmodelle des Immunsystems notwendig, sich eingehender mit den Massenwirkungsgesetzen zu befassen. Im Folgenden wird eine Einleitung zu diesem Thema gegeben, die auf dem Lehrbuch *Allgemeine und anorganische Chemie* von Erwin Riedel [Rie99] basiert.

Die Geschwindigkeit chemischer Reaktionen

Eine chemische Reaktion findet statt, wenn zwei oder mehr Reaktionspartner (Atome oder Moleküle) im Raum zusammentreffen und die eventuell benötigte Aktivierungsenergie vorhanden ist. Abbildung 2.11 zeigt eine solche Reaktion: Zwei Hydrogeniodid-Moleküle stoßen zusammen und reagieren zu einem Wasserstoff- und einem Iodmolekül. Man spricht hier von einer *molekularen Zerfallsreaktion* des Hydrogeniodids. Da zwei Moleküle an der Reaktion beteiligt sind, nennt man sie *bimolekular*.

Chemische Reaktionen werden durch *Reaktionsgleichungen* beschrieben, die Anfangs- und Endprodukte der Reaktion angeben. Die Reaktionsgleichung für den Zerfall von Hydrogeniodid lautet beispielsweise



Diese Reaktionsgleichung beschreibt die lokalen Vorgänge, die beim Zerfall von Hydrogeniodid auftreten. Betrachten wir nun eine große Anzahl von Hydrogeniodidmolekülen in einem geschlossenen System, werden wir global eine Abnahme der *Konzentration* des Hydrogeniodids über die Zeit beobachten. Verlauf und Geschwindigkeit dieses globalen Verhaltens sind der Gegenstand der *chemischen Kinetik*. Zur Beschreibung der Kinetik verwendet man Differentialgleichungen, mit denen die Veränderung der Stoffkonzentrationen beschrieben wird. Für die Konzentration des Stoffes HI schreibt man beispielsweise $[HI]$. Aus der Reaktionsgleichung lässt sich bereits ein grundlegender Zusammenhang zwischen den Molekülkonzentrationen ableiten:

$$-2 \frac{d[HI]}{dt} = \frac{d[I_2]}{dt} = \frac{d[H_2]}{dt}$$

Jeder dieser Terme kann als Geschwindigkeit der Zerfallsreaktion angesehen werden. Grundlage für die Bestimmung der Geschwindigkeit ist eine einfache Überlegung, die *Stoßtheorie*. Angenommen, Hydrogeniodid liegt in einer Konzentration von c Teilchen pro Raumeinheit vor: $[HI] = c$. Diese Teilchen bewegen sich frei im Raum. Wenn ein Teilchen eine Strecke der Länge l zurücklegt, wird es im Mittel den Weg von \bar{n} anderen Teilchen kreuzen.

Die Reaktionsgeschwindigkeit entspricht der Zahl der Elementarreaktionen im Gesamtsystem S pro Zeiteinheit. Diese ist proportional zur Summe der Zusammenstöße (nur ein Teil der Zusammenstöße führt tatsächlich zu einer Reaktion). Die Summe der Zusammenstöße ist proportional zum Produkt der Teilchenkonzentration mit dem Mittelwert der potenziellen Reaktionspartner \bar{n} . Insgesamt ergibt sich:

$$-2 \frac{d[HI]}{dt} \sim [HI] \sum_S \bar{n}$$

Abbildung 2.12 illustriert die Kernaussage der Stoßtheorie: Erhöht man die Konzentration des Hydrogeniodids um den Faktor k , so wird sich bei einer großen Zahl von Partikeln auch der Mittelwert \bar{n} der potenziellen Reaktionspartner um den Faktor k erhöhen. Auch \bar{n} ist also proportional zur Konzentration des Hydrogeniodids. Dadurch ergibt sich:

$$-2 \frac{d[HI]}{dt} \sim [HI][HI]$$

Allgemein lässt sich das Massenwirkungsgesetz so formulieren:

Die Geschwindigkeit einer chemischen Reaktion ist proportional zu den Konzentrationen der Ausgangsprodukte.

Tatsächlich kann für den Zerfall von Hydrogeniodid experimentell folgende Geschwindigkeitsgleichung gefunden werden:

$$-2\frac{d[HI]}{dt} = k[HI]^2 \quad (2.2)$$

k ist hierbei eine von verschiedenen Faktoren wie der Raumtemperatur und der für die Reaktion benötigten Aktivierungsenergie abhängige Konstante. Sie muss experimentell bestimmt werden.

Die Differentialform der Geschwindigkeitsgleichung kann durch Integration in Zeitgesetze für die Konzentrationen der Ausgangs- und Endprodukte umgeformt werden. Um eine solche Gleichung für $[HI]$ zu erhalten, stellen wir die Differentialgleichung um zu

$$-\frac{d[HI]}{[HI]^2} = \frac{k}{2} dt$$

und integrieren nach der Zeit:

$$\frac{1}{[HI(t)]} - \frac{1}{[HI(0)]} = \frac{k}{2}t$$

So erhalten wir die *integrierte Geschwindigkeitsgleichung*

$$[HI(t)] = \frac{1}{\frac{k}{2}t + \frac{1}{[HI(0)]}} \quad (2.3)$$

durch die der gesamte Reaktionsverlauf in Abhängigkeit von der Initialkonzentration $[HI(0)]$ errechnet werden kann. Abbildung 2.13 zeigt Plots der integrierten Geschwindigkeitsgleichung für verschiedene Anfangskonzentrationen.

Ordnung chemischer Reaktionen

Die *Ordnung* einer elementaren chemischen Reaktion entspricht der Anzahl der daran beteiligten Moleküle. In der Natur kommen fast ausschließlich Reaktionen einer Ordnung ≤ 3 vor. Die Ordnung einer chemischen Reaktion ist maßgeblich für das Aufstellen der Geschwindigkeitsgleichungen.

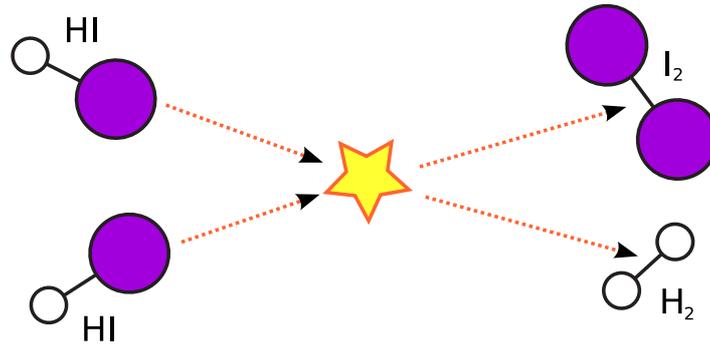


Abbildung 2.11.: Eine bimolekulare Reaktion: Zwei aufeinandertreffende Hydrogeniodid-Moleküle reagieren zu einem H_2 - und einem I_2 -Molekül. Aus [Rie99]

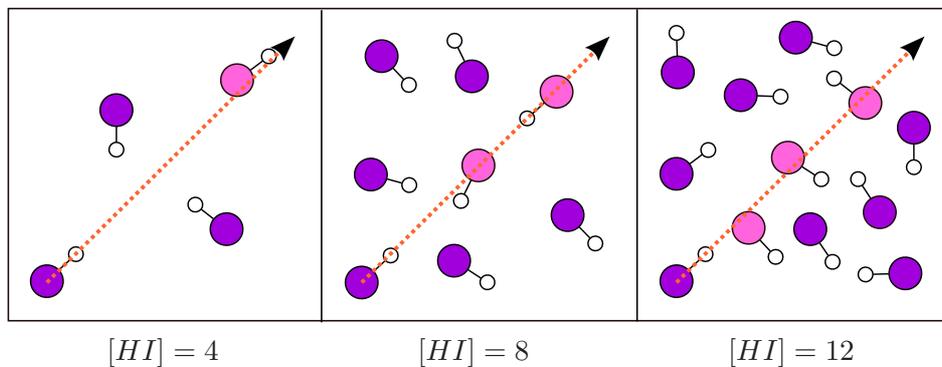


Abbildung 2.12.: Veranschaulichung der Stoßtheorie: Erhöht man die Konzentration des Hydrogeniodids um den Faktor k , so wird ein sich frei bewegendes Teilchen auf einer bestimmten Strecke im Mittel k mal so viele Teilchen treffen wie zuvor.

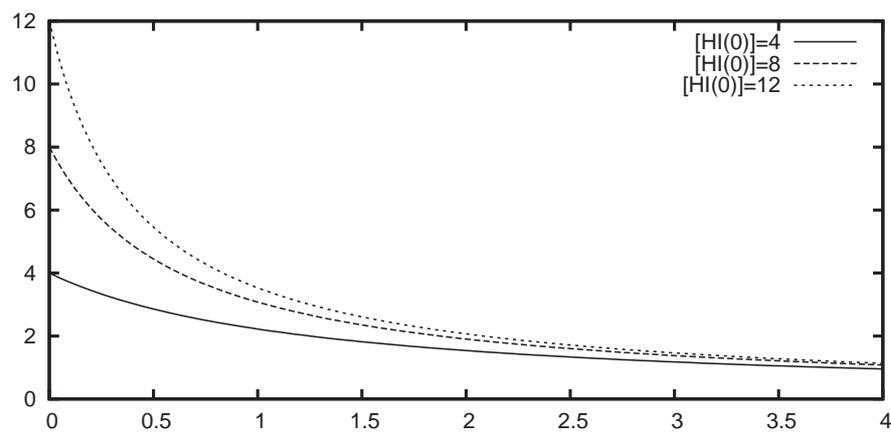
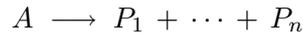


Abbildung 2.13.: Plots der integrierten Geschwindigkeitsgleichungen des Zerfalls von Hydrogeniodid in Iod und Wasserstoff. $[HI(0)]$ gibt die Initialkonzentration in Teilchen pro Raumeinheit an.

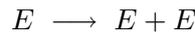
Reaktionen erster Ordnung: Eine Reaktion erster Ordnung (unimolekulare Reaktion) liegt vor, wenn ein Stoff ohne Einwirkung eines Reaktionspartners seine Beschaffenheit verändert. Ein Beispiel ist der radioaktive Zerfall. Bei einer Reaktion erster Ordnung hat die Reaktionsgleichung die Form



und führt unter der Annahme $A \notin \{P_1, \dots, P_n\}$ zu der Geschwindigkeitsgleichung

$$-\frac{d[A]}{dt} = \frac{d[P_i]}{dt} = k[A] \quad (2.4)$$

Auch die biologischen Vorgänge der Zellteilung und des Zelltods können als Reaktionen erster Ordnung modelliert werden. So könnte man die Teilung eines *E. coli* Bakteriums durch die Reaktionsgleichung



ausdrücken. Angenommen, dieser Vorgang findet pro Bakterium im Mittel einmal pro Zeiteinheit statt ($k = 1$), ergibt dies die Geschwindigkeitsgleichung

$$\frac{d[E]}{dt} = [E]$$

Reaktionen zweiter Ordnung: Der oben erläuterte Zerfall von Hydrogeniodid ist ein Beispiel für eine Reaktion zweiter Ordnung (bimolekulare Reaktion). In diesem Fall reagiert das HI mit sich selbst. Allgemein nimmt die Reaktionsgleichung die Form



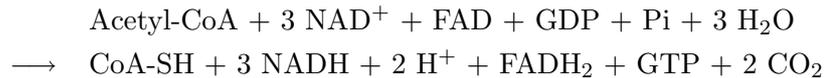
an, was unter der Annahme $A, B \notin \{P_1, \dots, P_n\}$ zu folgender Geschwindigkeitsgleichung führt:

$$\frac{-d[A]}{dt} = \frac{-d[B]}{dt} = \frac{d[P_i]}{dt} = k[A][B] \quad (2.5)$$

Reaktionen dritter und höherer Ordnung: Die Grundüberlegungen der Stoßtheorie lassen sich prinzipiell analog auf Reaktionen dreier oder mehrerer Moleküle anwenden. Jedoch ist das gleichzeitige Zusammentreffen von mehr als zwei Molekülen im Raum vergleichsweise sehr unwahrscheinlich, so dass solche Reaktionen in einem chemischen System selten geschwindigkeitsbestimmend sind. Bei trimolekularen Reaktionen ist meist eins der Ausgangsprodukte in

sehr hoher Konzentration vorhanden, so dass die Kinetik nicht messbar von der einer bimolekularen Reaktion abweicht. In einem solchen Fall spricht man von einer *pseudobimolekularen* Reaktion.

Komplexe biochemische Reaktionen sind oft aus zahlreichen Elementarschritten zusammengesetzt. Ein Beispiel ist der Citratzyklus. Für diesen gilt die Reaktionsgleichung

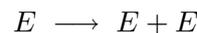


Es handelt sich hier aber nicht etwa um eine Reaktion zehnter Ordnung, obwohl auf der linken Seite der Gleichung zehn Ausgangsprodukte notiert sind. Vielmehr ist der Citratzyklus eine komplexe Reaktionskette aus zehn voneinander abhängigen Einzelreaktionen. Die Massenwirkungsgesetze gelten daher nicht für die Gesamtreaktion.

Parallel ablaufende Reaktionen

In realen chemischen und biologischen Systemen laufen meist mehrere Reaktionen gleichzeitig ab. Die Formulierung von Geschwindigkeitsgleichungen für solche Systeme erfolgt durch einfache Summation der elementaren Gleichungen nach den Massenwirkungsgesetzen.

Als Beispiel soll hier wiederum die Entwicklung einer Bakterienkultur, beispielweise von *E. coli*, dienen. Die Teilung der Bakterien genügt wie oben erläutert der Reaktionsgleichung



Den Zelltod könnte man folgendermaßen modellieren:



Dies entspricht den elementaren Geschwindigkeitsgleichungen

$$\frac{d[E]}{dt} = k_{PE}[E]$$

für die Teilung und

$$\frac{d[E]}{dt} = -k_{DE}[E]$$

für den Zelltod. k_{PE} gibt hier die Teilungsrate (*proliferation rate*) und k_{DE} die Todesrate (*death rate*) von *E. coli* unter den betrachteten Lebensbedingungen an, die experimentell

bestimmt werden müssen. Die Kinetik des Gesamtprozesses ergibt sich nun durch Summation der Geschwindigkeitsterme:

$$\frac{d[E]}{dt} = (k_{PE} - k_{DE})[E]$$

Es wird deutlich, dass die Massenwirkungsgesetze auch zur Modellierung biologischer Vorgänge gut geeignet sind, wenn man von sehr großen Mengen interagierender Partikel ausgehen kann. Die hier dargestellte Geschwindigkeitsgleichung für den Zell-Lebenszyklus findet dementsprechend auch in den meisten Differentialgleichungsmodellen des Immunsystems, wie dem im Folgenden erläuterten Modell von Carneiro et al., Verwendung.

2.2.2. Das Modell von Carneiro et al.

Das Modell von Carneiro et al. [CCFS96] dient der Untersuchung eines möglichen regulatorischen Mechanismus, mit dem das Immunsystem in Anwesenheit endogenen Antigens einen stabilen Zustand aufrechterhält. Modelliert werden dabei T- und B-Zellen und Antikörper, die von körpereigenem (endogenem) Antigen angeregt werden, dessen Konzentrationsrate als konstant angenommen wird. Das Immunsystem sollte solches Antigen zwar erkennen und neutralisieren, dabei aber eine “vernünftige” Intensität der Immunantwort garantieren, so dass es auch bei lang andauernder konstanter Stimulation durch das endogene Antigen nicht zu unkontrolliertem B- oder T-Zell-Wachstum kommt. ⁴

Die Wechselwirkungen zwischen T-Zellen, B-Zellen, Antigen und Antikörper sind in Abbildung 2.14 dargestellt. Antigen stimuliert die Proliferation von B- und T-Zellen. Monoklonale B- und T-Zellen stimulieren sich gegenseitig (Kostimulation). Stimulierte B-Zellen sezernieren Antikörper, dessen Präsenz ab einer gewissen Schwelle die Proliferation der T-Zellen inhibiert. Dadurch wird eine negative Feedback-Schleife erzeugt, die das unbegrenzte Wachstum der Lymphozyten verhindert und so das System stabilisiert.

Das allgemeine Modell bildet auch die B- und T-Zell-Rezeptordiversität ab, die analytischen und numerischen Untersuchungen wurden aber anhand vereinfachter Modelle durchgeführt, die nur ein oder zwei Antigenarten kennen. Wir behandeln hier das Modell mit nur einem Antigen.

⁴Nach der Netzwerktheorie [Jer74] wirkt Antikörper selbst als endogenes Antigen. Weiterhin entstehen moderneren Theorien zufolge konstant Tumorzellen im Körper, die normalerweise durch Absonderung bestimmter Stoffe als endogenes Antigen erkannt werden können und dann vom Immunsystem neutralisiert werden, bevor ein Tumor entstehen kann.

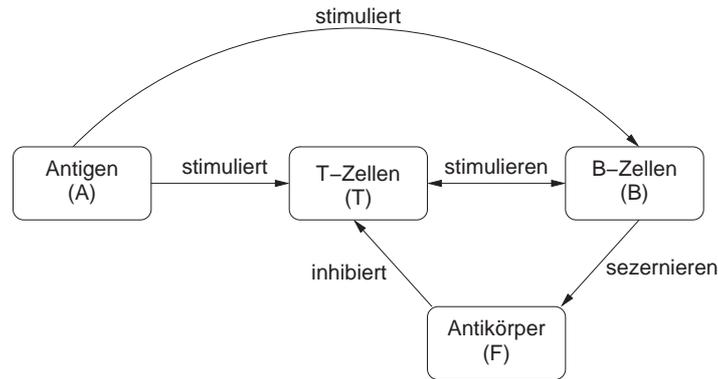


Abbildung 2.14.: Immunologischer Regelkreis im Modell von Carneiro et al. [CCFS96]

T-Zellen

T-Zellen werden durch Kontakt mit Antigenfragmenten, die sie z.B. von dendritischen Zellen präsentiert bekommen, zur Proliferation angeregt. Da die Präsenz des Antigens als konstant angenommen wird, entspricht dies einem normalen Wachstumsprozess. Zusammen mit dem normalen Zelltod führt dies wie in Abschnitt 2.2.1 zu folgender Differentialgleichung (Konzentrationen sind vereinfachend ohne die eckigen Klammern notiert):

$$\frac{dT}{dt} = (k_{PT} - k_{DT}) \cdot T \quad (2.6)$$

Hier gibt k_{PT} die Proliferationsrate und k_{DT} die Sterberate pro Zeitschritt an.

B-Zellen

B-Zellen werden zur Proliferation angeregt, wenn sie an Antigen gebunden haben und eine T-Zelle treffen, die das gleiche Antigen erkannt hat. Entsprechend dem Massenwirkungsgesetz einer bimolekularen Reaktion würde dies zu folgender Gleichung führen:

$$\frac{dB}{dt} = k_{PB} \cdot B \cdot T$$

Hier folgen Carneiro et al. aber einer Argumentation von DeBoer und Perelson [BP95], die in diesem Fall die Modellierung gemäß dem Massenwirkungsgesetz für unangemessen halten. Sie kritisieren, dass der Term $B \cdot T$ unbeschränkt wächst, wenn eine der beiden Konzentrationen gegen unendlich geht. In Wirklichkeit kann aber eine T-Zelle pro Zeiteinheit nur begrenzt viele B-Zellen stimulieren, so dass das B-Zell-Wachstum nach oben beschränkt sein sollte, wenn nur B oder T gegen unendlich geht.

In [BP95] wird ein modifizierter Wachstumsterm hergeleitet, der dieses Kriterium erfüllt:

$$\frac{dB}{dt} = k_{PB} \cdot \frac{B \cdot T}{B + T}$$

Im Gegensatz zum Massenwirkungsgesetz erhöht sich hier die Zahl neu gebildeter B-Zellen nur um den Faktor k (statt k^2), wenn die Zahl der B-T-Zellpaare um den Faktor k erhöht wird. Fügt man den Zelltod mit Sterberate k_{DB} hinzu, ergibt sich die Gleichung

$$\frac{dB}{dt} = -k_{DB} \cdot B + k_{PB} \cdot \frac{T \cdot B}{T + B} \quad (2.7)$$

Antikörper

Der Antikörperspiegel im Serum (im Modell durch F abgekürzt) sinkt durch Zerfallen der Moleküle mit Rate k_{DF} und durch Binden des Antikörpers an Antigen mit der Rate k_{DC} . Neuer Antikörper wird von Plasmazellen sezerniert, die sich mit T-Zell-Hilfe aus aktivierten B-Zellen entwickeln. Nimmt man die Menge des pro Plasmazelle produzierten Antikörpers als konstant an, ist die Zunahme an Antikörper durch Sezernierung proportional mit Faktor k_{SF} zur Zahl der durch Kostimulation angeregten B-Zellen (siehe oben). Zusammenfassend erhält man die Gleichung

$$\frac{dF}{dt} = -(k_{DF} + k_{DC}) \cdot F + k_{SF} \cdot \frac{T \cdot B}{T + B} \quad (2.8)$$

Inhibition der T-Zell-Aktivierung durch Antikörper

Das bisher entwickelte mathematische Modell führt zu unkontrolliertem Wachstum. Die in Abbildung 2.14 dargestellte negative Feedbackschleife wird erreicht, indem die Proliferation der T-Zellen gehemmt wird, falls die Antikörperkonzentration eine bestimmte Schwelle übersteigt.

Dieser Vorgang wird durch die *T-Zell-Inhibitionsfunktion* $\alpha_T(F)$ mit den zwei Parametern a_{T1} und a_{T2} modelliert:

$$\alpha_T(F) = \begin{cases} 1 & \log(F) < a_{T1} \\ \exp\left(-\left(\frac{\log(F)-a_{T1}}{a_{T2}}\right)^2\right) & \log(F) \geq a_{T1} \end{cases} \quad (2.9)$$

Die Funktion $\alpha_T(F)$ ist in Abbildung 2.15 dargestellt. Ist der Antikörperspiegel kleiner als der Schwellwert a_{T1} , findet keine Inhibition statt. Übersteigt der Antikörperspiegel den Schwellwert, nimmt die T-Zell-Aktivität exponentiell ab. Die Steilheit des exponentiellen Abstiegs

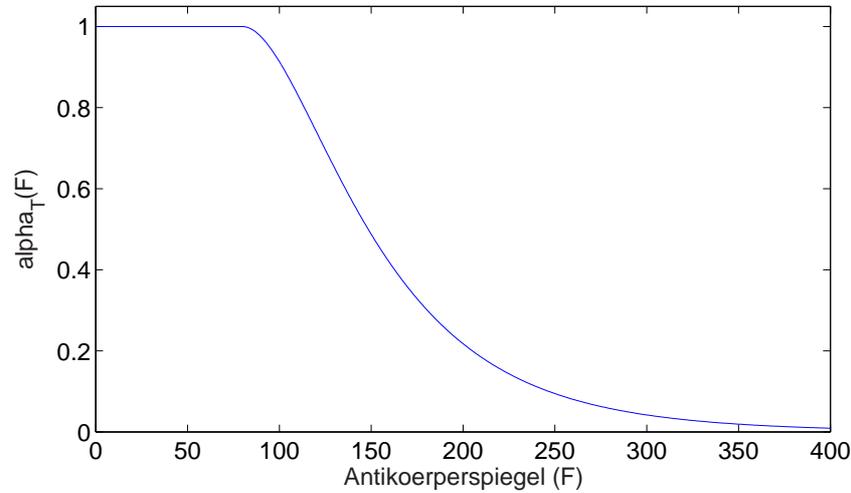


Abbildung 2.15.: Die T-Zell-Inhibitionsfunktion $\alpha_T(F)$ für $a_{T1} = \log(80)$, $a_{T2} = \log(2.1)$

wird durch den Parameter a_{T2} festgelegt. Die Werte aus Abbildung 2.15 wurden aus [CCFS96] übernommen.

Das Differentialgleichungsmodell wird nun komplettiert, indem jedes Vorkommen aktivierter T-Zellen mit der Inhibitionsfunktion multipliziert wird. Das Gesamtmodell lautet damit:

$$\begin{aligned}
 \frac{dT}{dt} &= (\alpha_T(F) \cdot k_{PT} - k_{DT}) \cdot T & (2.10) \\
 \frac{dB}{dt} &= -k_{DB} \cdot B + k_{PB} \cdot \frac{\alpha_T(F) \cdot T \cdot B}{\alpha_T(F) \cdot T + B} \\
 \frac{dF}{dt} &= -(k_{DF} + k_{DC}) \cdot F + k_{SF} \cdot \frac{\alpha_T(F) \cdot T \cdot B}{\alpha_T(F) \cdot T + B}
 \end{aligned}$$

Formale Analyse

Das Gleichungssystem besitzt einen einzigen nichttrivialen Fixpunkt, wenn folgende Bedingungen erfüllt sind [CCFS96, Mur02]:

$$k_{DB} < k_{PB}, \quad k_{DT} < k_{PT}, \quad \alpha_T(F) = k_{DT}/k_{PT}$$

Der Fixpunkt ist lokal stabil, wenn die Ungleichung $k_{PB} > k_{DB} + k_{DT}$ erfüllt ist. Die Werte für T , B und F am Fixpunkt ergeben sich durch:

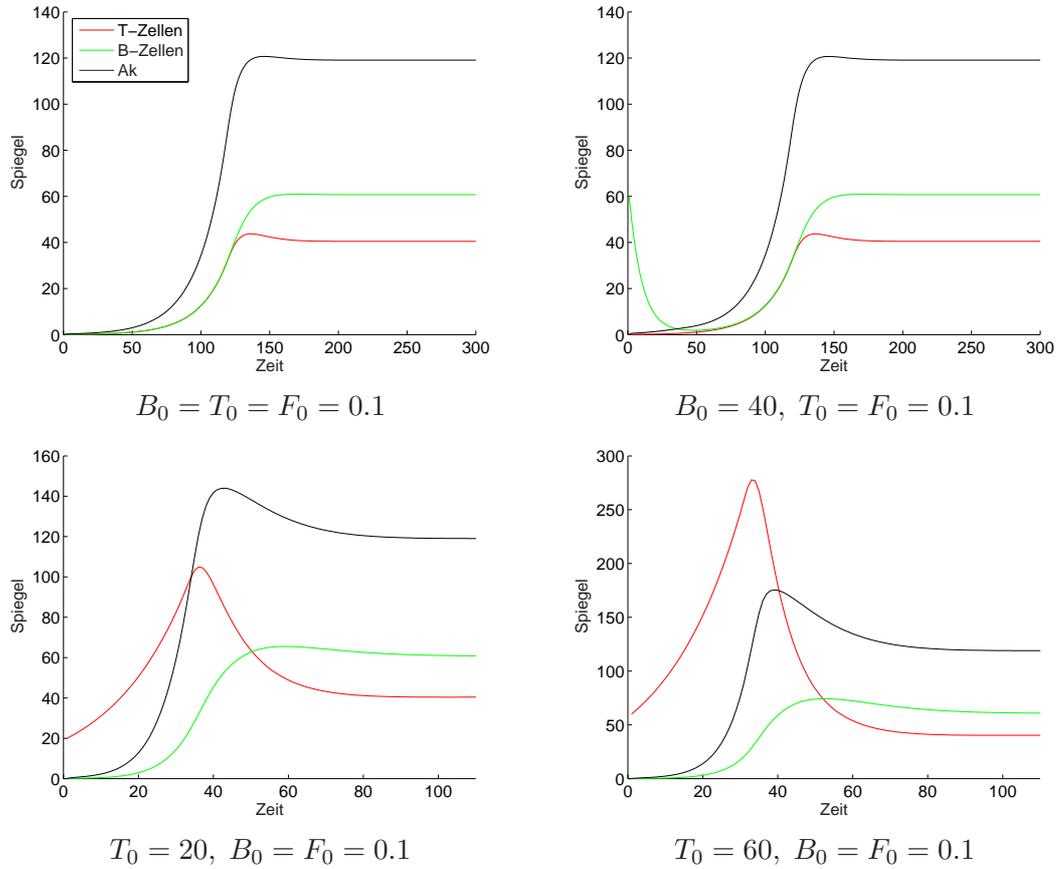


Abbildung 2.16.: Numerische Integration des Modells von Carneiro et al. [CCFS96] für verschiedene Anfangsbedingungen

$$\begin{aligned}
 F &= \alpha_T^{-1} \left(\frac{k_{DT}}{k_{PT}} \right) \\
 B &= \frac{k_{PB} \cdot (k_{DF} + k_{DC})}{k_{SF} \cdot k_{DB}} \cdot F \\
 T &= \frac{k_{PT} \cdot k_{PB} \cdot (k_{DF} + k_{DC})}{k_{DT} \cdot k_{SF} \cdot (k_{PB} - k_{DB})} \cdot F
 \end{aligned}$$

Numerische Analyse

Um die Plausibilität des Systems zu untersuchen, wurden die Gleichungen mit verschiedenen Initialwerten für T , B und F numerisch integriert. Die übrigen Modellparameter wurden aus [CCFS96] entnommen. Die Ergebnisse der Integration sind in Abbildung 2.16 dargestellt. Das Modell wies eine hohe Stabilität auf: Trotz stark variierender Anfangsbedingungen konvergierte es jedes Mal gegen den oben angegebenen Fixpunkt.

Immunologische Interpretation

Der Fixpunkt des Modells erfüllt die Eigenschaften, die ein Kontrollmechanismus zur Stabilisierung des Immunsystems gegenüber endogenem Antigen aufweisen muss: Er ist stabil auch gegenüber starker Variation der Anfangsbedingungen und allein durch die intrinsischen Systemparameter (Proliferations- und Sterberaten der Zellen und Moleküle, Parameter der T-Zell-Inhibitionsfunktion) bestimmt. Die Aussagen sind aber rein qualitativ zu sehen, Werte und Geschwindigkeiten können nicht mit realen Werten in Verbindung gebracht werden. Carneiro et al. bezeichnen ihr Modell selbst als sehr starke Vereinfachung der tatsächlichen Vorgänge im Immunsystem.

Die Einfachheit des Modells macht eine formale Analyse des entstehenden Gleichungssystems erst möglich, und dies auch nur im nochmals vereinfachten Fall eines einzigen Antigentyps. Komplexere Differentialgleichungsmodelle wie in [HMR96] können nur noch numerisch integriert werden.

2.3. Reaktions-Diffusions-Systeme

Kein im Rahmen dieser Arbeit betrachtetes Differentialgleichungsmodell enthält einen Begriff des Raums wie das CS-Modell, so dass man denken könnte, dies sei ein prinzipieller Nachteil dieses Modelltyps. Ein Blick in die Chemie zeigt aber, dass für chemische Reaktionen Differentialgleichungsmodelle existieren, die nicht nur den Verlauf der Reaktionen über die Zeit, sondern auch den Verlauf im Raum modellieren. Dies geschieht durch die Kombination von Termen für die Reaktionen mit Termen, die die Diffusion der Chemikalien im Raum modellieren. Die formale und numerische Analyse solcher *Reaktions-Diffusions-Systeme* geht auf Alan Turing zurück [Tur52].

Theoretisch könnten auch die Differentialgleichungsmodelle des Immunsystems um Diffusionsterme erweitert werden, um den Verlauf der Immunantwort im Raum zu modellieren. Dieser Ansatz konnte im Rahmen dieser Arbeit nicht untersucht werden. Die strukturellen Ähnlichkeiten der Reaktions-Diffusions-Systeme zum CS-Modell lieferten aber Hinweise, wie das CS-Modell mathematisch analysiert werden kann. Daher sollen Reaktions-Diffusions-Systeme im Folgenden kurz vorgestellt werden.

2.3.1. Modellierung von Diffusionsprozessen

Die mathematische Charakterisierung von Diffusionsprozessen ist identisch mit der von physikalischen Vorgängen wie der Ausbreitung elektrischer Energie in einem Leiter oder der Wärmeausbreitung in einem Festkörper. Hier kann also auf Resultate aus der Thermodynamik zurückgegriffen werden. Das wesentliche physikalische Resultat zur Beschreibung der Diffusion

ist das Ficksche Gesetz.

Das Ficksche Gesetz

Immer wenn Teilchen, Partikel oder Energien frei im Raum beweglich sind, kommt es zu einem Diffusionsprozess. Die simultane freie Bewegung der Partikel (*Brownsche Bewegung*) führt dazu, dass entlang eines Konzentrationsgefälles mehr Partikel in Richtung der abfallenden Konzentration wandern als umgekehrt. Mit der Zeit werden daher die Konzentrationsunterschiede im Raum ausgeglichen und es entsteht eine homogene Verteilung der Partikel.

Das Ficksche Gesetz erlaubt eine mathematische Charakterisierung von Diffusionsprozessen. Sei $[A]$ die Konzentration eines Stoffes oder einer Energie pro Raumeinheit, dann gilt für die Änderung der Konzentration nach der Zeit durch den Diffusionsprozess die Gleichung [Mur02]:

$$\frac{\partial[A]}{\partial t} = D\nabla^2[A] \quad (2.11)$$

In einem eindimensionalen Ring entspräche ∇^2 der zweiten Ableitung der Konzentration nach der Ringposition. Die Konstante D ist vom Stoff abhängig und gibt dessen Diffusionsgeschwindigkeit an. Die Diffusionsgeschwindigkeit von Gasen ist beispielsweise meist höher als die von in einer Flüssigkeit gelösten Salzen.

Numerische Integration der Diffusionsgleichung

Das Ficksche Gesetz ist die Grundlage für die mathematische Modellierung von Diffusionsprozessen und deren Simulation im Computer. Die folgende formale Modellangabe basiert auf einer Arbeit von Leppänen [Lep04].

Wir betrachten einen zusammenhängenden Bereich $\Omega \subset \mathbb{R}^n$ mit einem geschlossenen Randbereich $\partial\Omega$. Sei $[A](x, t)$, $x \in \Omega$ die Konzentration eines Stoffes im Punkt x zum Zeitpunkt t . Dann gilt nach dem Fickschen Gesetz die Gleichung

$$\frac{\partial[A](x, \cdot)}{\partial t} = D\nabla^2[A](x, \cdot) \quad \forall x \in \Omega$$

Für die Kompletierung des Modells benötigen wir die Anfangsbedingungen $[A](\cdot, 0)$ und die Randbedingungen. Häufig wählt man die *von Neumannsche Randbedingung*, wonach der Rand undurchlässig ist. Dies können wir wie folgt notieren:

$$\vec{n} \cdot \nabla[A] = 0 \quad \text{auf } \partial\Omega$$

\vec{n} ist dabei der Normalenvektor des Randes. Diese Bedingung kann erfüllt werden, indem die Funktionswerte in Ω über den Rand nach außen gespiegelt werden. Der Gradient $\nabla[A]$ verläuft dann tangential zum Rand, also orthogonal zu \vec{n} , das Skalarprodukt ist 0.

Bei der *periodischen Randbedingung* fordert man, dass sich die Werte von $[A]$ zyklisch wiederholen. Im Fall $\Omega = \mathbb{R}^2$ entspräche dies der Bedingung

$$[A]((x, y), t) = [A]((x + w, y + h), t) \quad \forall x \forall t$$

Bei periodischen Randbedingungen entspricht Ω einem Torus mit Breite w und Höhe h .

Um das Modell im Computer abzubilden, müssen Zeit und Raum diskretisiert werden. Die Approximation von ∇^2 kann durch finite Differenzen erfolgen. Dies führt zum bekannten diskreten Laplace-Operator. Im 2D-Fall ist dieser gegeben durch die Matrix

$$L = \left(\frac{1}{2dx}\right)^2 \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

wobei dx die Breite des diskreten Gitters ist. Die diskrete Approximation von $\nabla^2[A]$ erhält man dann durch die Faltung $[A] \otimes L$. Am Rand muss hier bei von Neumannschen Randbedingungen die Spiegelung der Funktionswerte durchgeführt werden, was am linken Rand beispielsweise durch die Verwendung der Faltungsmatrix

$$\left(\frac{1}{2dx}\right)^2 \begin{pmatrix} 0 & 1 & 0 \\ 0 & -4 & 2 \\ 0 & 1 & 0 \end{pmatrix}$$

erreicht wird. Bei periodischen Randbedingungen fasst man Ω direkt als Torus auf und führt die Faltung am Rand modulo der Torusgrößen durch.

Die einfachste Methode zur numerischen Integration des Gleichungssystems ist die sogenannte FTCS-Methode (Forward in Time, Centered in Space). Hier wird ausgehend von $[A]^{(0)}$ nach der Vorschrift

$$[A]^{(t+dt)} = [A]^{(t)} + dt D ([A] \otimes L) \tag{2.12}$$

jeweils eine Approximation für $[A]^{(t+dt)}$ berechnet. Bei der Durchführung ist zu beachten, dass numerische Instabilitäten auftreten können, wenn dx zu klein oder dt zu groß gewählt wird. Die Methode ist in jedem Fall stabil für:

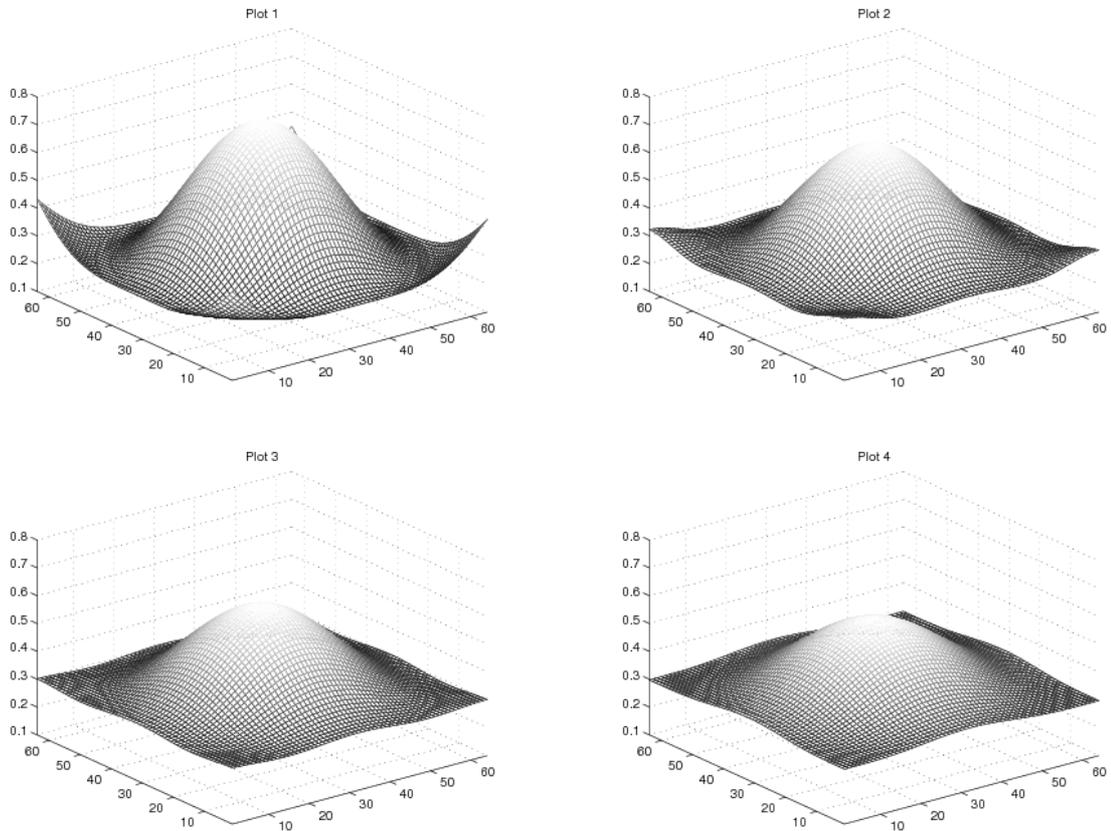


Abbildung 2.17.: Modellierung eines Diffusionsprozesses durch iterierte Anwendung des diskreten Laplace-Filters auf einen 64 mal 64 Felder großen 2D-Torus. Die Plots zeigen die Konzentration $[A]$ nach 1, 251, 501 und 701 Iterationen nach Vorschrift 2.12 mit den Parametern $dt = 0.3$ und $dx = 0.5$.

$$\frac{dt}{(dx)^2} \leq \frac{1}{2}$$

Die genauen Stabilitätsgrenzen hängen vom Spektrum der Anfangskonzentration ab.

Abbildung 2.17 illustriert einen so modellierten Diffusionsprozess. Hier wurde auf einem 2D-Torus ein Ausschnitt einer Sinuswelle als Initialkonzentration verwendet und daraufhin Gleichung 2.12 iteriert angewendet. Trotz der Parameter $dt = 0.3$ und $dx = 0.5$, die das Stabilitätskriterium für die FTCS-Methode nicht erfüllen, blieb das System numerisch stabil.

Turing-Muster

Reaktions-Diffusions-Systeme wurden 1952 von Alan Turing in seiner Arbeit *The chemical basis of morphogenesis* [Tur52] behandelt. Die Motivation war, eine mathematische Erklärung



Abbildung 2.18.: Turing-Muster aus dem Brüsselator. Aus [Lep04]

für die Bildung natürlicher Formen (*Morphogenese*) zu finden. Turing vermutete, dass diese durch das Zusammenspiel mehrerer miteinander reagierender Chemikalien zu Stande kommen könnten, die mit unterschiedlichen Geschwindigkeiten diffundieren.

Mittels der erläuterten Resultate aus Physik und Chemie stellte er ein Differentialgleichungssystem zu einer einfachen chemischen Reaktion auf und bewies, dass durch die unterschiedlichen Diffusionsgeschwindigkeiten der Reaktanten verschiedene Arten von Mustern im Raum entstehen können.

Abbildung 2.18 zeigt solche Turing-Muster aus dem Brüsselator, einem real existierenden chemischen System, mit dem zum ersten Mal die Bildung von Turing-Mustern experimentell nachgewiesen werden konnte. Die gezeigten Muster stammen allerdings aus einer Computersimulation des Brüsselators. Hier können die Diffusionsgeschwindigkeiten der Reaktanten beliebig variiert werden. Abhängig von den gewählten Werten kann das System dann verschiedene Formen, zum Beispiel die gezeigten Spiral- und Punktmuster, erzeugen.

2.3.2. Reaktions-Diffusions-Gleichungen

In Reaktions-Diffusions-Gleichungen werden die Massenwirkungsgesetze mit dem Fickschen Gesetz kombiniert, um chemische Reaktionen und ähnliche Prozesse sowohl im Raum als auch über die Zeit mathematisch zu beschreiben. Als Beispiel betrachten wir ein einfaches Konkurrenzsystem aus zwei Reaktanten A und B . Die Reaktionsgleichungen des Systems lauten:



Zwei Elemente des einen Reaktanten können also ein Element des anderen Reaktanten “konvertieren”. Die kinetischen Geschwindigkeitsgleichungen des Systems lauten damit wie folgt (siehe Abschnitt 2.2.1):

$$\frac{d[A]}{dt} = k_1[A][A][B] - k_2[B][B][A] \quad (2.15)$$

$$\frac{d[B]}{dt} = -k_1[A][A][B] + k_2[B][B][A] \quad (2.16)$$

Die Konstanten k_1 und k_2 geben wie üblich die Reaktionsgeschwindigkeiten an. Wir fügen nun Diffusionsterme nach Abschnitt 2.3.1 hinzu und erhalten dadurch ein Reaktions-Diffusions-System:

$$\begin{aligned} \frac{\partial[A]}{\partial t} &= D_A \nabla^2 A && + k_1[A][A][B] - k_2[B][B][A] \\ \frac{\partial[B]}{\partial t} &= D_B \nabla^2 B && - k_1[A][A][B] + k_2[B][B][A] \end{aligned} \quad (2.17)$$

Dieses Gleichungssystem beschreibt den Verlauf der Reaktion in Raum und Zeit in Abhängigkeit von den Diffusionsgeschwindigkeiten D_A und D_B und den Reaktionsgeschwindigkeiten k_1 und k_2 . Zur Komplettierung des Systems werden wieder die Anfangs- und Randbedingungen benötigt. Wegen der Nichtlinearitäten können Reaktions-Diffusions-Systeme meist nicht analytisch gelöst werden.

Wie in Abschnitt 2.3.1 beschrieben entspricht die numerische Integration des System einer Simulation der Reaktions- und Diffusionsvorgänge. Uns interessiert dabei der Verlauf der Simulation mehr als die Lösung der Gleichung im eigentlichen Sinne. Bei homogenen Anfangsbedingungen sind Verlauf und Endzustand trivial (vollständiges Verschwinden einer Spezies oder ein homogenes Gleichgewicht $[A] \equiv [B]$). Verrauschen wir aber die Anfangsbedingungen mit einem raumabhängigen Zufallswert, entwickelt das System fraktale Muster. Es sind auch Situationen möglich, bei denen sich die Konzentrationen in wellenförmigen Zyklen im Raum bewegen (traveling wave solution).

Abbildung 2.19 zeigt Höhenplots einer iterativen numerischen Integration dieses Gleichungssystems mit den Parametern $D_A = D_B = k_1 = k_2 = 0.5$, $dx = 0.5$, $dt = 0.3$, $[A]^{(0)} = 0.5 + \text{rand}(-0.05, 0.05)$, $[B]^{(0)} = 1 - [A]^{(0)}$. Die diskrete Integration wurde nach der FTCS-

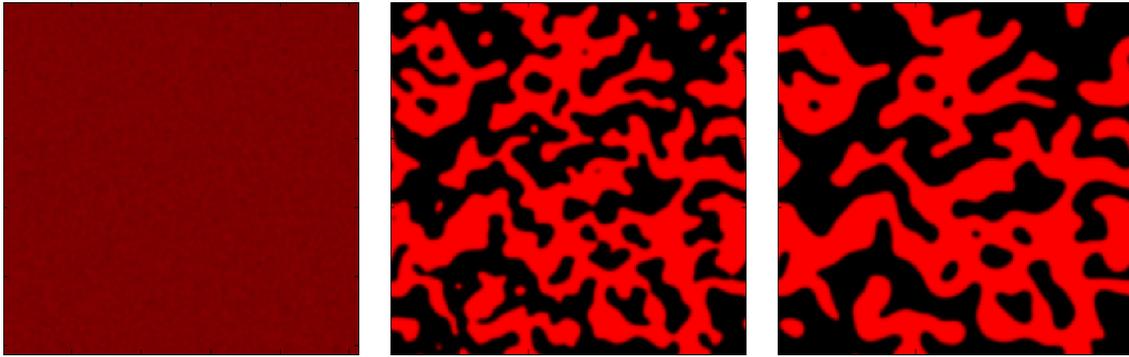


Abbildung 2.19.: Höhenplots der iterativen numerischen Integration des Reaktions-Diffusions-Systems 2.17 nach der FTCS-Methode mit $D_A = D_B = k_1 = k_2 = 0.5$; $dx = 0.5$; $dt = 0.3$; $[A]^{(0)} = 0.5 + \text{rand}(-0.05, 0.05)$; $[B]^{(0)} = 1 - [A]^{(0)}$

Methode gemäß Vorschrift 2.12, erweitert um die Reaktionsterme, durchgeführt.

2.3.3. Strukturelle Ähnlichkeiten zum CS-Modell

Vergleicht man den Ablauf der numerischen Integration eines Reaktions-Diffusions-Systems mit dem CS-Modell, werden einige Parallelen deutlich. Durch die Diskretisierung des Raums bei der numerischen Integration wird dieser zu einem diskreten Gitter. Für die Berechnung der Reaktionsterme an einem Gitterpunkt sind nur die Funktionswerte dieses Gitterpunkts von Bedeutung. In die Berechnung der Diffusionsterme fließen nur die Funktionswerte aus der unmittelbaren Nachbarschaft eines Gitterpunkts ein. Dies entspricht genau dem Lokalisierungsprinzip des CS-Modells.

Durch den bei der Diffusion verwendeten diskreten Laplace-Operator (siehe Abschnitt 2.3.1) wird der Funktionswert eines Gitterpunkts gleichmäßig auf die Nachbarfelder “verteilt”. Dies entspricht dem Erwartungswert der probabilistischen Diffusionsbewegung des CS-Modells, bei der die Partikel eines Gitterpunkts auf zufällig ausgewählte Nachbarfelder diffundieren. Wir können also den zu erwartenden Verlauf der Diffusion im CS-Modell durch eine Differentialgleichung, nämlich das Ficksche Gesetz, charakterisieren. Dies wird im nächsten Kapitel formalisiert.

Das erwartete Verhalten der Reaktionsvorgänge (Interaktionen) im CS-Modell erschließt sich dagegen nicht sofort. Hier ist eine ausführlichere mathematische Betrachtung nötig. Diese ist ebenfalls Gegenstand des nächsten Kapitels.

3. Analyse des erwarteten Verhaltens des Celada-Seiden-Modells

In diesem Kapitel wird das erwartete Verhalten des probabilistischen CS-Modells untersucht und durch Differentialgleichungen charakterisiert. Gegenstand der Analyse werden dabei nicht die speziellen Regeln sein, nach denen Zellen und Moleküle im CS-Modell interagieren. Vielmehr sollen die allgemeinen Algorithmen untersucht werden, die den verschiedenen Aktionen und Interaktionen zugrundeliegen. Diese Algorithmen werden im Folgenden auch *Simulationsprozesse* genannt. Beispielsweise werden Proliferation und Zelltod für alle zellulären Partikel nach dem gleichen Algorithmus simuliert.

Eine ähnliche Analyse findet sich in der Arbeit der Chemiker Malevanets und Kapral [MK96, MK97]. Dort wurde ein bekanntes Reaktions-Diffusions-System, das FitzHugh-Nagumo Modell, durch einen dem CS-Modell ähnlichen probabilistischen Automaten simuliert. In [MK97] wurde bewiesen, dass das erwartete Verhalten dieses Automaten den FitzHugh-Nagumo-Gleichungen entspricht. Dieser Beweis dient den Autoren als Hauptargument für die Plausibilität ihrer Simulation.

Beim Entwurf des CS-Modells wurde keine solche Analyse der biophysikalischen Plausibilität durchgeführt, da das Modell von vornherein nur zur *qualitativen* Simulation einer Immunantwort gedacht war. Die Implementierung der Simulationsprozesse erfolgte vor allem mit Hinblick auf algorithmische Einfachheit. Durch die im letzten Kapitel aufgezeigten strukturellen Ähnlichkeiten zu Reaktions-Diffusions-Systemen bietet sich an, die von Malevanets und Kapral durchgeführte Analyse auf das CS-Modell anzuwenden. Dadurch kann im Nachhinein geklärt werden, inwieweit das Modell biophysikalisch plausibel ist, und wo eventuell nachgebessert werden müsste.

3.1. Simulationsprozesse im CS-Modell

Folgende zentralen Simulationsprozesse des CS-Modells werden in diesem Kapitel untersucht (vergleiche zur Einordnung der Prozesse auch die schematische Darstellung eines Simulationsschritts in Abbildung 2.8):

- Diffusion

Diffusion wird in jedem Simulationsschritt einmal durchgeführt. Dabei verbleibt jede Zelle und jedes Molekül entweder auf ihrem Feld oder wandert auf eines der benachbarten Felder.

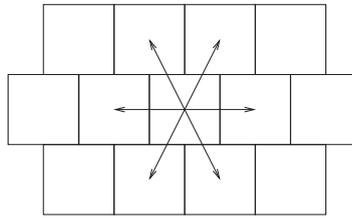


Abbildung 3.1.: Felder und 6-Nachbarschaft im CS-Modell

- Proliferation

Proliferation bezeichnet die Teilung von Zellen. Dies betrifft einerseits das Antigen (z.B. wenn es sich um bakterielles Antigen handelt) und andererseits die B- und T-Zellen, die das Antigen erfolgreich erkannt haben und sich dann teilen, um eine große Zahl von spezifischen Abwehrzellen zu erzeugen.

- Interaktion

Mit dem Begriff Interaktion bezeichnen wir hier alle Aktionen, die bei Zusammentreffen zweier Partikel im Raum stattfinden können. Vor allem betrifft dies die Phagozytose und die gegenseitige Stimulation von Zellen.

Zuerst werden wir für die Diffusion testen, ob sie das in Abschnitt 2.3.1 vorgestellte Ficksche Gesetz erfüllt. Bei den Interaktionen interessiert uns, ob sie den Massenwirkungsgesetzen (Abschnitt 2.2.1) genügen, die die Grundlage für die meisten Differentialgleichungsmodelle des Immunsystems sind. Bei der Proliferation interessiert uns vor allem, ob gegenüber der einfachen Gleichung aus Abschnitt 2.2.1, die unbeschränktes exponentielles Wachstum modelliert, ein Mechanismus implementiert ist, der das Zellwachstum nach oben beschränkt. Das erwartete Verhalten sollte möglichst mit einem bekannten Modell für bakterielles Wachstum (für eine Übersicht siehe [Mur02], Kapitel 1) übereinstimmen.

3.2. Erwartetes Verhalten der Simulationsprozesse

3.2.1. Diffusion

Wir beginnen mit dem erwarteten Verhalten der Diffusion auf dem Gitter. Dazu benötigen wir zuerst einen formalen Begriff dieses Gitters. Im CS-Modell wird ein modifiziertes quadratisches Gitter mit einer 6-Nachbarschaft verwendet (siehe Abbildung 3.1), in dem die Abstandsvektoren eines Feldes zu den Nachbarfeldern eine nichtisotropische Form bilden, die eine formale Analyse verkomplizieren würde. Wir führen die folgenden Betrachtungen daher zunächst auf einer leichter zu behandelnden klassischen 4-Nachbarschaft durch.

Diffusion und Fluss in einem rechteckigen 2D-Gitter

Wir definieren das Gitter des CS-Modells wie üblich als ungerichteten Graphen. Um den Vergleich des erwarteten Verhaltens zu einem physikalischen Diffusionsprozess im Raum durchzuführen zu können, fügen wir eine räumliche Interpretation der Knoten hinzu. Sei w die Breite des Gitters in Feldern in x -Richtung, h die Höhe. Das Gitter $G = (V, E)$ ist ein ungerichteter Graph mit den wie folgt gegebenen Knoten und Kanten:

$$V = \{ (x\Delta s, y\Delta s) \mid 0 \leq x < w, 0 \leq y < h \}$$

$$E = \{ ((x\Delta s, y\Delta s), ((x \pm 1)\Delta s \bmod w\Delta s, y\Delta s)) \mid 0 \leq x < w, 0 \leq y < h \}$$

$$\cup \{ ((x\Delta s, y\Delta s), (x\Delta s, ((y \pm 1)\Delta s \bmod h\Delta s))) \mid 0 \leq x < w, 0 \leq y < h \}$$

Durch die Definition der Kantenmenge entspricht das Gitter geometrisch einem 2D-Torus. Die Bezeichnung der Knoten interpretieren wir dabei direkt als Koordinaten in \mathbb{R}^2 . Die Kantenmenge entspricht dem Nachbarschaftsbegriff

$$\mathcal{N} : V \mapsto V^4$$

$$\mathcal{N}(x, y) = \{((x \pm \Delta s) \bmod w\Delta s, (y \pm \Delta s) \bmod h\Delta s)\}$$

Da im CS-Modell keine gegenseitige Beeinflussung der Partikelarten in Bezug auf die Diffusion stattfindet, betrachten wir hier vereinfachend nur einen einzigen Partikeltyp und formalisieren dies durch eine Knotengewichtung $c(x, y, t)$, die die Zahl der Partikel auf dem Knoten (x, y) zum Zeitpunkt / Simulationsschritt t angibt. Wir verstehen c dabei als diskrete Approximation einer reellwertigen Funktion \mathcal{C} :

$$c : V \times \mathbb{N} \mapsto \mathbb{N}$$

$$C : \mathbb{R}^2 \times \mathbb{R} \mapsto \mathbb{R}$$

$$c(x, y, t) \approx \int_{x' = x - \Delta s/2}^{x + \Delta s/2} \int_{y' = y - \Delta s/2}^{y + \Delta s/2} \mathcal{C}(x', y', t) dy' dx'$$

$$C(x, y, t) \approx \frac{1}{(\Delta s)^2} c(x', y', t) \quad \text{mit } (x', y') \in V : \|(x, y) - (x', y')\| = \min.$$

Während eines Diffusionsschritts im diskreten Modell werden entlang der Kanten Partikel ausgetauscht, wie in Abbildung 3.2 illustriert ist. Hier sehen wir den Partikelaustausch als Approximation eines reellwertigen Flussprozesses $J(x, y, t)$. Der Fluss $J(x, y, t)$ an einem Punkt

(x, y) zum Zeitpunkt t ist ein Vektor, der angibt, in welche Richtung und mit welcher Stärke die Partikel an diesem Punkt diffundieren:

$$J = \begin{pmatrix} J_x \\ J_y \end{pmatrix} : \mathbb{R}^2 \times \mathbb{R} \mapsto \mathbb{R}^2$$

$$\begin{aligned} o_1(t) - i_1(t) &\approx \int_{y'=y}^{y+\Delta s} J_y(x, y', t) dy' \\ i_3(t) - o_3(t) &\approx \int_{y'=y-\Delta s}^y J_y(x, y', t) dy' \\ o_2(t) - i_2(t) &\approx \int_{x'=x}^{x+\Delta s} J_x(x', y, t) dx' \\ i_4(t) - o_4(t) &\approx \int_{x'=x-\Delta s}^x J_x(x', y, t) dx' \end{aligned} \quad (3.1)$$

Der Fluss steht wie folgt mit der Änderung der Konzentration am betrachteten Punkt über die Zeit in Verbindung [Mur02]:

$$\frac{\partial \mathcal{C}}{\partial t} = -\nabla J$$

Unter Verwendung von Gleichung 3.1 erhalten wir folgenden Zusammenhang, der den Austausch von Partikeln im diskreten Modell als Approximation eines zeitkontinuierlichen Flussprozesses beschreibt:

$$\frac{\partial \mathcal{C}}{\partial t} = -\nabla J \approx \frac{1}{2(\Delta s)^2} \begin{pmatrix} i_2(t) + i_4(t) - o_2(t) - o_4(t) \\ i_1(t) + i_3(t) - o_1(t) - o_3(t) \end{pmatrix}$$

Durch Diskretisierung von t erhalten wir die Finite-Differenzen-Approximation

$$\frac{\mathcal{C}(x, y, t + \Delta t) - \mathcal{C}(x, y, t)}{\Delta t} \approx \frac{1}{2(\Delta s)^2} \left[\sum_j i_j(t) - \sum_j o_j(t) \right] \quad (3.2)$$

Bei einer randomisierten Implementierung der Diffusion sind die $i_j(t)$ und $o_j(t)$ Zufallsvariablen mit Ereignisraum \mathbb{N} . Um die konkrete Implementierung der Diffusion durch den approximierten kontinuierlichen Diffusionsprozess zu charakterisieren, werden wir die Erwartungswerte der $i_j(t)$ und $o_j(t)$ sowie des Ausdrucks 3.2 untersuchen.

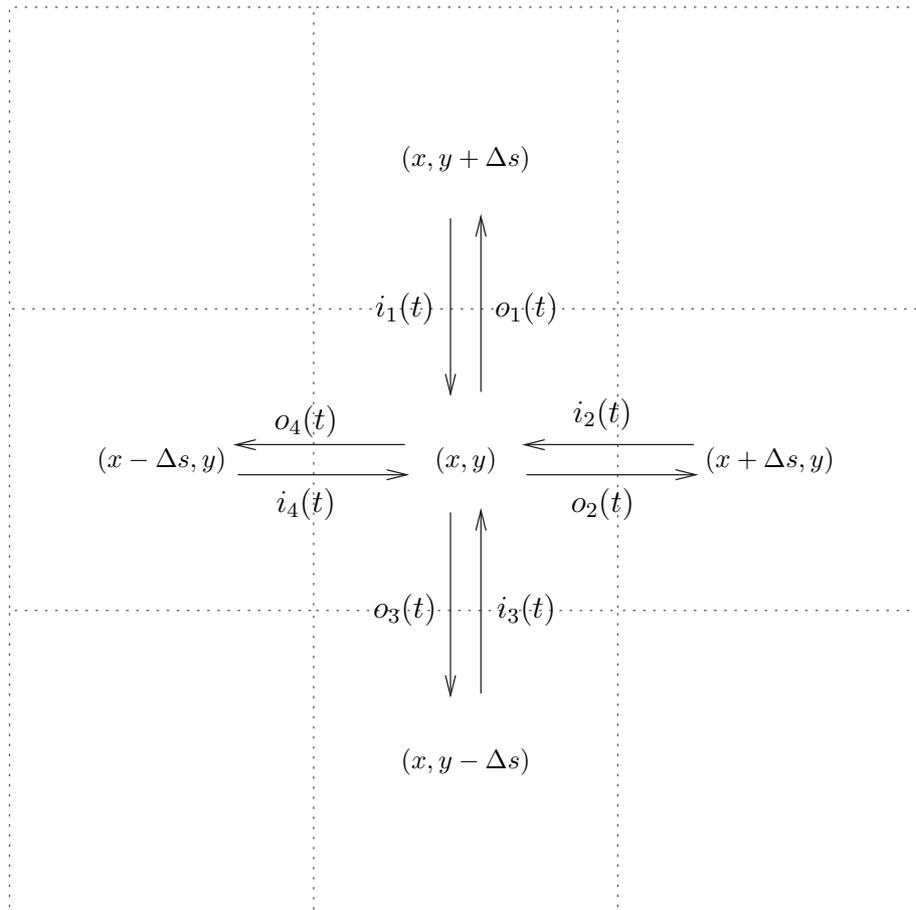


Abbildung 3.2.: Ein Diffusionsschritt in der Nachbarschaft eines Punktes (x, y) im rechteckigen Gitter

Diffusion durch einfache Brownsche Bewegung

Wir beginnen mit dem einfachsten Algorithmus, der in ImmSim für Zellen und Moleküle, in CImmSim nur für die Moleküle verwendet wird. Dieser simuliert eine gleichzeitige Zufallsbewegung (Brownsche Bewegung) aller Partikel:

```
01 procedure diffusion ( s: site )
02   for each m in { molecules at s }
03     with probability D do
04       target[m] = random neighbour of s
05     else
06       target[m] = s
07     end
08   end
09 end
```

Der Algorithmus behandelt alle Felder des Gitters parallel, so dass das Ergebnis der Diffusion auf einem Feld für die Nachbarschaft keine unmittelbaren Auswirkungen hat. Jedes Molekül wird also nur genau einmal betrachtet.

In CImmSim ist der Spezialfall $D = \#\mathcal{N}(\mathbf{s})/(1 + \#\mathcal{N}(\mathbf{s}))$ implementiert – die Wahrscheinlichkeit für den Verbleib eines Partikels auf seinem Ursprungsfeld ist genau so groß wie die Wahrscheinlichkeit der Diffusion auf ein bestimmtes Zielfeld.

Wir betrachten nun das erwartete Verhalten des Algorithmus. Sei $W_{uv}(t)$, $(u, v) \in E$, die Anzahl der Partikel, die durch einen Diffusionsschritt von u nach v verschoben werden. $W_{uv}(t)$ ist eine binomialverteilte Zufallsvariable:

$$\Pr[W_{uv}(t) = n] = \binom{c(u,t)}{n} \left(\frac{D}{4}\right)^n \left(1 - \frac{D}{4}\right)^{c(u,t)-n}$$

$$\mathbb{E}[W_{uv}(t)] = \frac{D}{4} c(u,t)$$

Dadurch ergeben sich die Erwartungswerte der Differenzen aus Ausdruck 3.2 wie folgt:

$$\begin{aligned} \mathbb{E} \left[\sum_j o_j(t) \right] &= D c(x, y, t) \\ \mathbb{E} \left[\sum_j i_j(t) \right] &= \frac{D}{4} (c(x - \Delta s, y, t) + c(x, y - \Delta s, t) + c(x + \Delta s, y, t) + c(x, y + \Delta s, t)) \\ &= \frac{D}{4} \sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t) \end{aligned}$$

Durch Einsetzen der Erwartungswerte in Gleichung 3.2 erhalten wir:

$$\frac{\mathcal{C}(x, y, t + \Delta t) - \mathcal{C}(x, y, t)}{\Delta t} \approx \frac{D}{2(\Delta s)^2} \left[-c(x, y, t) + \frac{1}{4} \sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t) \right]$$

Dies ist eine Finite-Differenzen-Approximation der Gleichung

$$\frac{\partial \mathcal{C}}{\partial t} = \frac{D}{2} \left[\frac{\partial^2 \mathcal{C}}{\partial x^2} + \frac{\partial^2 \mathcal{C}}{\partial y^2} \right] = \frac{D}{2} \nabla^2 \mathcal{C}$$

die wir bereits als Ficksches Gesetz kennengelernt haben (Abschnitt 2.3.1). Die betrachtete Implementierung der Diffusion kann also als diskrete Approximation eines physikalischen Diffusionsprozesses angesehen werden.

Diffusion von Zellen in CImmSim

Der eben analysierte Diffusionsalgorithmus simuliert eine gleichzeitige Brownsche Bewegung aller Partikel. Dieses physikalische Modell gilt aber nur für Teilchen in hinreichend geringer Konzentration, so dass die Brownsche Bewegung nicht wesentlich durch Kollisionen gestört wird. Bei höher konzentrierten oder größeren Partikeln (wie Zellen) werden häufig erweiterte Modelle verwendet, um die Dynamik des Diffusionsprozesses zu beschreiben. Hier erhöht sich der Diffusionsdruck bei hoher Konzentration der Partikel.

In [Mur02] wird die Diffusionsgleichung wie folgt verallgemeinert, um die Erhöhung des Diffusionsdrucks wiedergeben zu können:

$$\frac{\partial \mathcal{C}}{\partial t} = D \nabla \cdot \left[\left(\frac{\mathcal{C}}{c_0} \right)^m \nabla n \right] \quad \text{mit } m, c_0 > 0 \quad (3.3)$$

Auch in CImmSim wurde die Implementierung der Diffusion für Zellen geändert, um einen höheren Diffusionsdruck bei hohen Konzentrationen zu erreichen. Es wird dabei für jede Zelle

zunächst ein Zielfeld ausgewählt und die Verschiebung wird dann mit einer Wahrscheinlichkeit umgekehrt proportional zur Sättigung dieses Feldes durchgeführt:

```

01 procedure diffusion ( s: site )
02   for each c in {cells at s}
03     t = random neighbour of s
04     p = D * (1 - #{cells at t} / CMAX)
05     with probability p do
06       target[c] = t
07     else
08       target[c] = s
09     end
10   end
11 end

```

Dadurch soll neben der Anpassung des Diffusionsdrucks sichergestellt werden, dass nicht mehr als CMAX Zellen auf einem Gitterfeld zu liegen kommen. Dies ist allerdings durch den Algorithmus nicht ganz ausgeschlossen. Befinden sich auf einem Feld CMAX-1 Zellen, so ist die Wahrscheinlichkeit nicht 0, dass von jedem Nachbarfeld eine Zelle zu diesem Feld diffundiert.

Um die Anpassung des Diffusionsdrucks zu verifizieren, betrachten wir wieder den Erwartungswert der Zufallsvariable $W_{uv}(t)$:

$$E[W_{uv}(t)] = \frac{D}{4} \left(1 - \frac{c(v,t)}{\text{CMAX}} \right) c(u,t)$$

Die Erwartungswerte der eingehenden und ausgehenden Partikel sind also:

$$\begin{aligned}
 E \left[\sum_j o_j(t) \right] &= \frac{D c(x,y,t)}{4} \left(4 - \frac{\sum_{(x',y') \in \mathcal{N}(x,y)} c(x',y',t)}{\text{CMAX}} \right) \\
 E \left[\sum_j i_j(t) \right] &= \frac{D \sum_{(x',y') \in \mathcal{N}(x,y)} c(x',y',t)}{4} \left(1 - \frac{c(x,y,t)}{\text{CMAX}} \right)
 \end{aligned}$$

Wir sehen an der unteren Gleichung, dass in Regionen mit hoher durchschnittlicher Partikelkonzentration die Diffusion zum Stillstand kommt, während ein leeres Feld die Partikel aus der Umgebung “anzieht”. Dieser Effekt erscheint gerade konträr zu der erwünschten Erhöhung des Diffusionsdrucks bei hoher Konzentration. Wir untersuchen nun den Netto-Fluss nach Gleichung 3.2:

$$\begin{aligned}
 & \frac{\mathcal{C}(x, y, t + \Delta t) - \mathcal{C}(x, y, t)}{\Delta t} \\
 \approx & \frac{D \sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t)}{8 (\Delta s)^2} \left(1 - \frac{c(x, y, t)}{\text{CMAX}} \right) - \frac{D c(x, y, t)}{8 (\Delta s)^2} \left(4 - \frac{\sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t)}{\text{CMAX}} \right) \\
 = & \frac{D}{2 (\Delta s)^2} \left[-c(x, y, t) + \frac{1}{4} \sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t) \right]
 \end{aligned}$$

und stellen fest, dass wir wiederum die Differentialgleichung

$$\frac{\partial c}{\partial t} = \frac{D}{2} \left[\frac{\partial^2 \mathcal{C}}{\partial x^2} + \frac{\partial^2 \mathcal{C}}{\partial y^2} \right] = \frac{D}{2} \nabla^2 c$$

erhalten, wie auch beim Algorithmus ohne Modifikation des Diffusionsdrucks.

Damit ist gezeigt, dass die Änderung des Diffusions-Algorithmus in CImmSim für Zellen in Bezug auf das erwartete Verhalten wirkungslos ist. Bei hohen Konzentrationen tritt ein weiteres Problem auf: Wenn alle Felder einer Gitterregion mit der maximalen Anzahl an Zellen belegt sind, findet gar keine Diffusion mehr statt. Dies wäre nur vertretbar, wenn nur eine einzige Zellart simuliert werden würde, was im CS-Modell aber nicht der Fall ist. Es stellt sich daher die Frage, wie die Implementierung der Diffusionsbewegung korrigiert werden könnte.

Alternative Implementierung für die Zelldiffusion

Wir untersuchen folgende alternative Implementierung der Zelldiffusion:

```

01 procedure cell_diffusion ( s: site )
02   for each c in {cells at s}
03     p = D * #{cells at s} / CMAX
04     with probability p do
05       target[c] = random neighbour of s
06     else
07       target[c] = s
08     end
09   end
10 end

```

Dieser Algorithmus betrachtet zur Bestimmung des Diffusionsdrucks nicht das Ziel-, sondern das Quellfeld, funktioniert ansonsten aber analog. Dadurch wird nicht sichergestellt, dass nicht mehr als CMAX Zellen auf einem Feld zu liegen kommen. Es kommt aber bei starker

Sättigung nicht zum Diffusionsstillstand wie bei dem zuvor untersuchten Algorithmus.

Wir untersuchen nun wieder den Erwartungswert von $W_{uv}(t)$:

$$E[W_{uv}(t)] = \left(\frac{D}{4 \text{ CMAX}} \right) c(u, t)^2$$

$$\begin{aligned} E \left[\sum_j o_j(t) \right] &= \frac{D}{\text{CMAX}} c(x, y, t)^2 \\ E \left[\sum_j i_j(t) \right] &= \frac{D \sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t)^2}{4 \text{ CMAX}} \end{aligned}$$

und durch Einsetzen in Gleichung 3.2 erhalten wir:

$$\begin{aligned} & \frac{\mathcal{C}(x, y, t + \Delta t) - \mathcal{C}(x, y, t)}{\Delta t} \\ \approx & \frac{D}{8 (\Delta s)^2 \text{ CMAX}} \sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t)^2 - c(x, y, t)^2 \\ = & \frac{D}{8 (\Delta s)^2 \text{ CMAX}} \sum_{(x', y') \in \mathcal{N}(x, y)} (c(x', y', t) + c(x, y, t))(c(x', y', t) - c(x, y, t)) \\ = & \frac{D}{8 (\Delta s)^2 \text{ CMAX}} \sum_{(x', y') \in \mathcal{N}(x, y)} 2c(x, y, t)(c(x', y', t) - c(x, y, t)) + (c(x', y', t) - c(x, y, t))^2 \\ = & \frac{D c(x, y, t)}{(\Delta s)^2 \text{ CMAX}} \left[-c(x, y, t) + \frac{1}{4} \sum_{(x', y') \in \mathcal{N}(x, y)} c(x', y', t) \right] \\ & + \frac{D c(x, y, t)}{2 (\Delta s)^2 \text{ CMAX}} \left[\frac{1}{4} \sum_{(x', y') \in \mathcal{N}(x, y)} (c(x', y', t) - c(x, y, t))^2 \right] \end{aligned}$$

Die hierdurch approximierte Differentialgleichung lautet:

$$\begin{aligned}
 \frac{\partial \mathcal{C}}{\partial t} &= \frac{D}{\text{CMAX}} \left[c \left[\frac{\partial^2 \mathcal{C}}{\partial^2 x} + \frac{\partial^2 \mathcal{C}}{\partial^2 y} \right] + \left(\frac{\partial \mathcal{C}}{\partial x} \right)^2 + \left(\frac{\partial \mathcal{C}}{\partial y} \right)^2 \right] \\
 &= \frac{D}{\text{CMAX}} \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \cdot \begin{pmatrix} c \frac{\partial \mathcal{C}}{\partial x} \\ c \frac{\partial \mathcal{C}}{\partial y} \end{pmatrix} \\
 &= D \nabla \cdot \left[\frac{c}{\text{CMAX}} \nabla \mathcal{C} \right]
 \end{aligned}$$

und entspricht damit Gleichung 3.3 mit $c_0 = \text{CMAX}$, $m = 1$. Mit diesem Algorithmus kann also das erwartete Verhalten erreicht werden, dass in CImmSim angestrebt wurde: Bei größerer Zelldichte erhöht sich der Diffusionsdruck.

3.2.2. Proliferation

Abgesehen von der in jedem Zeitschritt stattfindenden Einfügung neuer B- und T-Zellen aus dem virtuellen Knochenmark und Thymus und der programmierten Injektion von Antigen können neue Zellen im Modell nur durch Proliferation entstehen.

Die Proliferation der Zellen wird dabei nach einem Algorithmus durchgeführt, der die auf dem betrachteten Feld bereits vorhandenen Zellen berücksichtigt. Bei höherer Belegungsichte wird der Proliferationsprozess verlangsamt. Im Pseudocode sieht der Algorithmus wie folgt aus: ¹

```

01 procedure proliferation( a: type, s: site )
02   p = exp( - (#{particles of type a at s}/K)2 )
03   for u in { particles of type a at s }
04     with probability p do
05       clone(u)
06     end
07   end
08 end

```

Wenn wir die Zahl der durch diesen Algorithmus im Gitterpunkt s zum Zeitpunkt t neu generierten Zellen mit $N_s(t)$ bezeichnen, ist $N_s(t)$ eine binomialverteilte Zufallsvariable mit dem von der Anzahl $c(s, t)$ der Partikel des Typs a im Gitterpunkt s zum Zeitpunkt t abhängigen Erwartungswert

¹In der ursprünglichen Definition des CS-Modells [SC92] wird dieser Prozess nicht explizit so festgelegt. Der angegebene Pseudocode wurde anhand von CImmSim erstellt.

$$E[N_s(t)] = c(s, t)e^{(-c(s,t)/K)^2}$$

Dies entspricht der Differentialgleichungsform

$$\frac{\partial C}{\partial t} = C e^{(-C/K)^2}$$

Es ist unklar, wie dieser Algorithmus biologisch motiviert sein könnte. Offensichtlich wurde als Modell für die Wachstumsgeschwindigkeit eine Gaußverteilung benutzt. Dies entspricht allerdings nicht den üblichen Modellen für beschränkte Wachstumsprozesse wie der logistischen Gleichung (vgl. [Mur02], Kapitel 1). Weiterhin führt Überbelegung eines Gitterfeldes zu keiner Konsequenz.

3.2.3. Interaktion

Wie bereits erwähnt, bezeichnen wir mit dem Begriff “Interaktion” alle Prozesse im CS-Modell, die beim Zusammentreffen von Partikeln im Raum stattfinden können (siehe auch Abbildung 2.7):

- Phagozytose
Makrophagen phagozytieren Antigen und Immunkomplexe, B-Zellen phagozytieren Antigen
- Kostimulation
Makrophagen stimulieren T-Zellen, B- und T-Zellen stimulieren sich gegenseitig
- Bindung von Antikörper an Antigen

Wie man sieht, sind im CS-Modell sind bei allen elementaren Interaktionen höchstens zwei verschiedene Partikeltypen beteiligt. Wie auch bei der Diffusion ist die Implementierung der Interaktion probabilistisch. Meist steht aber die Interaktionswahrscheinlichkeit nicht a priori fest, sondern wird beim Betrachten eines Partikelpaars aus der Affinität ermittelt (siehe Abschnitt 2.1.1). Nur die Phagozytose von Antigen und Immunkomplexen durch Makrophagen ist affinitätsunabhängig.

Wir wollen nun untersuchen, inwieweit die Implementierung dieser Interaktionen im CS-Modell die Massenwirkungsgesetze erfüllt. Der Interaktionsprozess ist durch einen einfachen Algorithmus realisiert:

```

01 procedure interaction ( a: type, b: particle type, s: site )
02   for u in { particles of type a at s }
03     for v in { particles of type b at s }
04       with probability  $p_{ab}$  do
05         interact(u,v); break
06       end
07     end
08   end
09 end

```

Es werden also alle Paare von Partikeln auf dem Feld s betrachtet, zwischen denen potenziell eine Interaktion stattfinden könnte. Für jedes Paar wird dann ein Zufallsexperiment mit der Wahrscheinlichkeit p_{ab} durchgeführt (für affinitätsabhängige Interaktionen ist $p_{ab} = a(u, v)$). Gelingt das Experiment, findet eine Interaktion zwischen den beiden Partikeln statt. Implizit wird hier davon ausgegangen, dass sich durch die Interaktion die Typen der beiden Partikel ändern, so dass beide für den Rest des Prozesses nicht mehr berücksichtigt werden. Dies erklärt die `break`-Anweisung in der inneren `for`-Schleife: Durch die Interaktion zwischen u und v haben beide Partikel entweder ihren Typ geändert oder sie sind verschwunden, so dass die innere Schleife abgebrochen und das nächste u betrachtet werden muss. Abbildung 3.3 zeigt einen möglichen Verlauf des Interaktionsprozesses für die Phagozytose von Antigen durch APCs.

Wie man sieht, ist jeder Durchlauf der äußeren Schleife von den Ergebnissen aller vorherigen Durchläufe abhängig. Dies macht es schwierig, für die Wahrscheinlichkeitsverteilung der Anzahl der stattfindenden Interaktionen eine geschlossene Formel anzugeben. Wir analysieren die Wahrscheinlichkeitsverteilung daher im Folgenden nur rechnergestützt. Weiterhin gehen wir davon aus, dass p_{ab} a priori feststeht, da auch nur für diesen Fall die Massenwirkungsgesetze gelten würden.

Sei $I_u(t)$ die Anzahl der Interaktionen zwischen den Partikeln des Typs a und b auf dem Gitterpunkt u zum Zeitschritt t , $a(u, t)$ und $b(u, t)$ seien die Konzentrationen der Partikel der Typen a und b . $P(k, n, m)$ bezeichne die Wahrscheinlichkeit, dass zwischen n Partikeln des Typs a und m Partikeln des Typs b k Interaktionen stattfinden:

$$P(k, n, m) := \Pr[I_u(t) = k \mid a(u, t) = n, b(u, t) = m]$$

Wir stellen zunächst fest, dass nicht mehr als $\min(n, m)$ Interaktionen auftreten können:

$$P(k, n, m) = 0 \quad \text{für } k > \min(n, m)$$

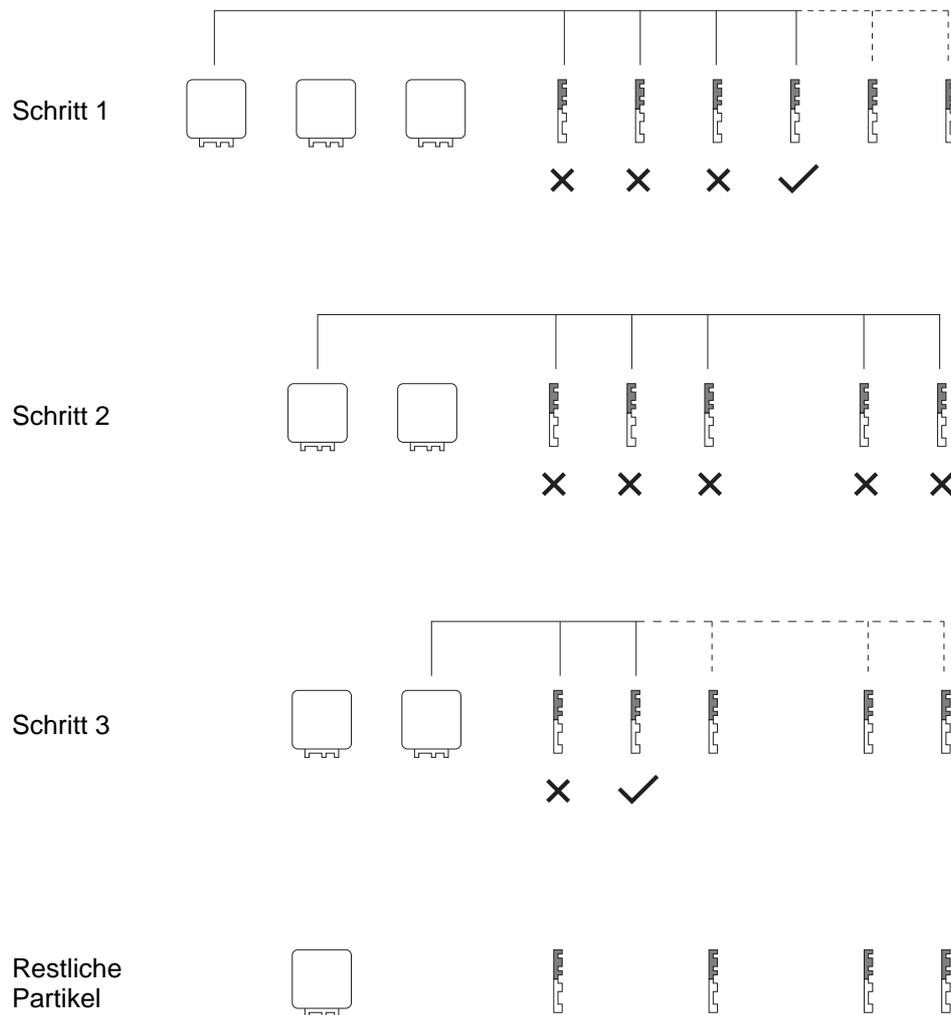


Abbildung 3.3.: Möglicher Verlauf eines Interaktionsprozesses im CS-Modell. Hier wird die unspezifische Phagozytose von Antigenen durch antigenpräsentierende Zellen (APCs) dargestellt. Zunächst stehen 6 Antigene 3 APCs gegenüber. Nach dem ersten Schritt (im Pseudocode ein Durchlauf der inneren Schleife) hat die erste APC das vierte Antigen phagozytiert. Das fünfte und sechste Antigen werden daher von dieser APC nicht mehr betrachtet (gestrichelte Linien). APC und Antigen ändern dadurch ihren Zustand und werden nicht weiter betrachtet. Der zweiten APC gelingt es nicht, ein Antigen zu phagozytieren. Im dritten Durchlauf phagozytiert die dritte APC ein weiteres Antigen. Damit ist der Interaktionsschritt abgeschlossen.

Für $k = 0$ gilt:

$$P(0, n, m) = (1 - p_{ab})^{nm}$$

Für $0 < k \leq \min(n, m)$ gilt der rekursive Zusammenhang

$$\begin{aligned} P(k, n, m) &= (1 - p_{ab})^{m-k} P(k, n - 1, m) \\ &+ (1 - (1 - p_{ab})^{m-k+1}) P(k - 1, n - 1, m) \end{aligned}$$

Dieser ergibt sich aus einer einfachen Überlegung: Haben beim Durchlauf des Programms k Interaktionen stattgefunden, so fanden entweder alle während der ersten $n - 1$ Durchläufe der äußeren Schleife statt und im letzten Durchlauf keine (oberer Term), oder es fanden $k - 1$ Interaktionen während der ersten $n - 1$ Durchläufe und im letzten Durchlauf dann eine Interaktion statt (unterer Term).

Mit dieser rekursiven Vorschrift kann die Wahrscheinlichkeitsverteilung für gegebene n, m und p_{ab} berechnet werden. Abbildung 3.4 zeigt Plots der Verteilung für $p_{ab} = 0.01$ und $m = 100$. In Abbildung 3.5 sind die Erwartungswerte für die Anzahl der Interaktionen in Abhängigkeit von n für verschiedene m mit $p_{ab} = 0.01$ aufgetragen. Erwartungsgemäß zeigt sich hier, dass der Interaktionsalgorithmus nicht nach dem Massenwirkungsgesetz arbeitet, denn dann müsste die Zahl der Interaktionen proportional zu den einzelnen Konzentrationen sein.

Weiterhin ist der Zusammenhang zwischen dem Parameter p_{ab} und der erwarteten Zahl der Interaktionen von Interesse. Auch dieser kann hier mangels einer geschlossenen Formel nur durch Plots verdeutlicht werden (Abbildung 3.6). Wiederum ergibt sich eine nichtlineare Abhängigkeit der mittleren Interaktionsgeschwindigkeit von p_{ab} .

Durch eine zusätzliche Überlegung wird die mangelnde physikalische Plausibilität des Interaktionsalgorithmus im CS-Modell verdeutlicht: Ist die Interaktionswahrscheinlichkeit p_{ab} sehr hoch, so kommt es fast sicher zur Maximalzahl an Interaktionen. Befindet sich zum Beispiel auf einem Gitterpunkt eine B-Zelle, die genau zu einem ebenfalls auf dem Gitterpunkt befindlichen Antigen passt, so ergibt sich p_{ab} aus der Formel für die Affinität (Abschnitt 2.1.1) zu 1. Die B-Zelle wird also in jedem Fall an ein Antigenpartikel binden, unabhängig davon, ob das Antigen in sehr geringer oder sehr hoher Konzentration vorliegt. Man würde erwarten, dass die Interaktionswahrscheinlichkeit bei geringerer Antigenkonzentration, beispielsweise wenn nur ein Antigenpartikel vorhanden ist, entsprechend abnimmt.

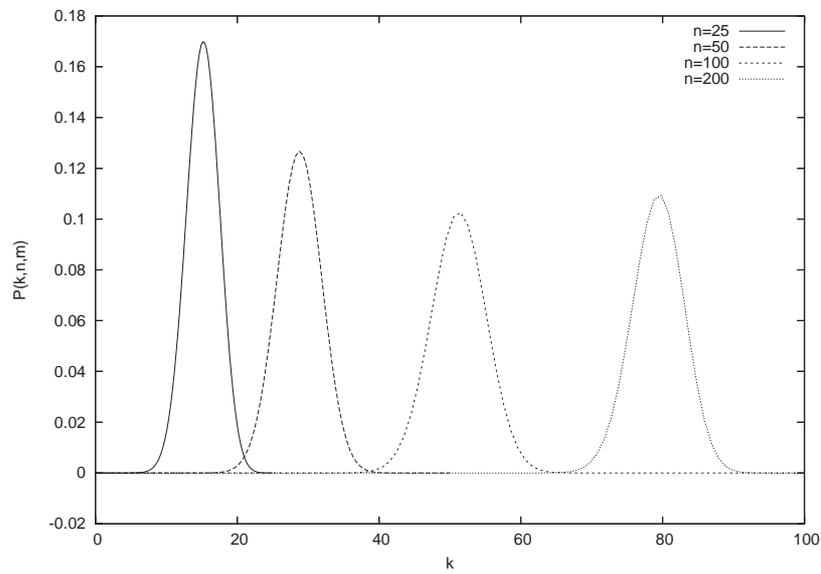


Abbildung 3.4.: Wahrscheinlichkeitsverteilung der Interaktionen bei einem Durchlauf des Interaktionsalgorithmus im CS-Modell für $m = 100$, $p_{ab} = 0.01$ und $n = 25$; $n = 50$; $n = 100$; $n = 200$.

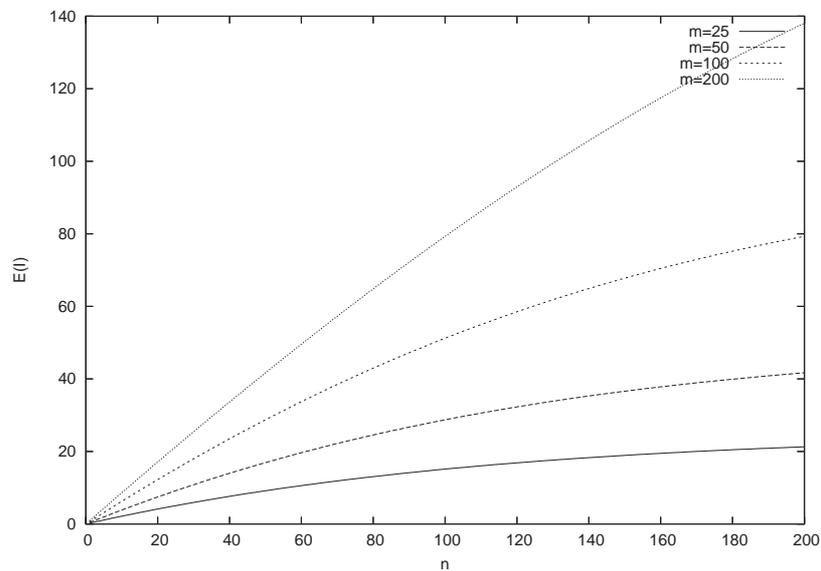


Abbildung 3.5.: Erwartungswerte der Interaktionen bei einem Durchlauf des Interaktionsalgorithmus im CS-Modell in Abhängigkeit von n mit $p_{ab} = 0.01$ und $m = 25$; $m = 50$; $m = 100$; $m = 200$.

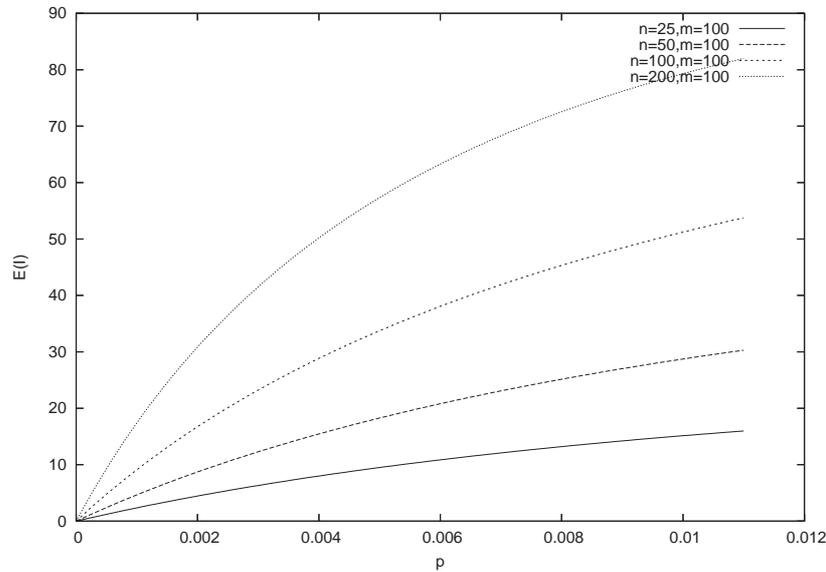


Abbildung 3.6.: Erwartungswerte der Interaktionen bei einem Durchlauf des Interaktionsalgorithmus im CS-Modell in Abhängigkeit von p_{ab} für $m = 100$ und $n = 25; n = 50; n = 100; n = 200$.

3.3. Zusammenfassung

Bei der Untersuchung des erwarteten Verhaltens des CS-Modells ergab sich ein gemischtes Bild. Der Diffusionsalgorithmus erfüllt das Ficksche Gesetz und kann daher als Approximation eines tatsächlichen Diffusionsprozesses angesehen werden. CImmSim weist einen abgewandelten Diffusionsalgorithmus für Zellen auf, der allerdings nach genauerer Analyse fragwürdig erscheint. Dazu wurde in diesem Kapitel ein alternativer Algorithmus erarbeitet, der das in CImmSim eigentlich angestrebte erwartete Verhalten aufweist.

Der Proliferationsalgorithmus kann mit keinem allgemein üblichen Modell für beschränkte Wachstumsprozesse in Verbindung gebracht werden. Seine biophysikalische Motivation ist unklar, das erwartete Verhalten jedoch noch relativ einfach zu bestimmen.

Für das erwartete Verhalten des Interaktionsalgorithmus konnte keine Differentialgleichung angegeben werden, da für die Wahrscheinlichkeitsverteilung keine geschlossene Formel gefunden werden konnte. Durch rekursive Berechnung der Wahrscheinlichkeitsverteilung für repräsentative Werte zeigte sich, dass das erwartete Verhalten nicht den Massenwirkungsgesetzen entspricht, die normalerweise für die Modellierung solcher Vorgänge durch Differentialgleichungen verwendet werden (siehe Kapitel 2). Besonders bei niedrigen Partikelkonzentrationen verhält sich dieser Algorithmus biophysikalisch nicht plausibel. Motiviert durch diese Ergebnisse werden im nächsten Kapitel alternative Algorithmen für Proliferation und Interaktion vorgeschlagen.

4. Erweiterung des Celada-Seiden-Modells

Wie aus der Untersuchung des erwarteten Verhaltens des CS-Modells deutlich wurde, spiegelt die Implementierung der grundlegenden Simulationsprozesse nur zum Teil die entsprechenden biophysikalischen Gesetzmäßigkeiten und Modelle wieder. Um die biophysikalische Plausibilität des Modells zu verbessern, sollte die Implementierung so geändert werden, dass diese Gesetzmäßigkeiten eingehalten werden.

Im Folgenden wird daher das CS-Modell so abgewandelt und erweitert, dass die implementierten Prozesse im Erwartungswert die entsprechenden Differentialgleichungen erfüllen. Das neue Modell trägt nach seinem Entstehungsort Lübeck den Namen *LImmSim*.

4.1. Beschränkung der Kapazität der Gitterpunkte

Zwar enthält das CS-Modell einen Mechanismus zur Verlangsamung der Proliferation bei zunehmender Sättigung eines Gitterpunkts. Die physikalische Motivation dieses Ansatzes ist jedoch unklar. In *LImmSim* wurde als Modell für das beschränkte Wachstum die bekannte logistische Gleichung ([Mur02], S. 3) gewählt. Diese führt einen Modellparameter K ein, der die maximale Aufnahmefähigkeit eines Gitterpunkts pro Partikeltyp angibt. Die Proliferation von Zellen eines Typs a wird dann nach folgendem Algorithmus durchgeführt:

```
01 procedure proliferation( a: type, s: site )
02    $p = v_a * (1.0 - \#\{\text{particles of type } a \text{ at } s\} / K)$ 
03   if  $p > 0$ 
04     for u in { particles of type a at s }
05       with probability  $p$  do
06         clone(u)
07       end
08     end
09   else
10     for u in { particles of type a at s }
11       with probability  $-p$  do
12         kill(u)
13       end
14     end
```

```

15   endif
16   end

```

Durch die Art der Implementierung ist K kein harter Grenzwert, der niemals überschritten werden kann. Eine Überschreitung führt aber zum Tod einiger Zellen und wird damit im Laufe der Zeit wieder ausgeglichen. Die Kapazitätsgrenze gilt hier für jeden Partikeltyp separat. Es können also gleichzeitig bis zu K B-Zellen, K T-Zellen und K Antigene auf einem Gitterpunkt vorhanden sein, aber niemals beispielsweise $2K$ B-Zellen und K T-Zellen.

Das erwartete Verhalten dieses Algorithmus wird charakterisiert durch die Zufallsvariable $N_s(t)$ der neu entstehenden Zellen auf dem Gitterpunkt s zum Zeitpunkt t :

$$E[N_s(t)] = v_a c(s, t) (1 - c(s, t) / K)$$

Hier gibt $c(s, t)$ wieder die Anzahl der Partikel des Typs a auf dem Feld s zum Zeitpunkt t an. Die dadurch approximierte Differentialgleichung lautet:

$$\frac{dC}{dt} = v_a C (1 - C / K)$$

Dies ist die logistische Gleichung für ressourcenbeschränktes Wachstum. Die Gesamtgeschwindigkeit des Wachstumsprozesses kann durch v_a festgelegt werden. Bei geringen Partikelkonzentrationen entspricht das Wachstum etwa einem einfachen exponentiellen Wachstumsprozess. Mit steigender Sättigung verlangsamt sich der Prozess, bis der Gitterpunkt gesättigt ist und gar kein Wachstum mehr stattfindet. Kommt es zu einer Übersättigung, z.B. durch Hinzuwanderung aus der Nachbarschaft durch Diffusion, wird der Wachstumsterm negativ und einige Partikel sterben. Abbildung 4.1 zeigt den Verlauf eines Wachstumsprozesses nach der logistischen Gleichung.

4.2. Modifikation des Interaktionsprozesses

Durch die Beschränkung der Partikelkapazität pro Gitterpunkt ist auch die Anzahl an Interaktionen zwischen Partikeln pro Gitterpunkt und Zeiteinheit beschränkt. Dies kann man ausnutzen, um den Interaktionsprozess des CS-Modells so zu modifizieren, dass die Massenwirkungsgesetze eingehalten werden. Im Pseudocode sieht der modifizierte Interaktionsprozess wie folgt aus:

```

01 procedure interaction ( a: type, b: type, s: site )
02   n = #{ particles of type a at s }
03   m = #{ particles of type b at s }
04   density = if n > m then n/K else m/K

```

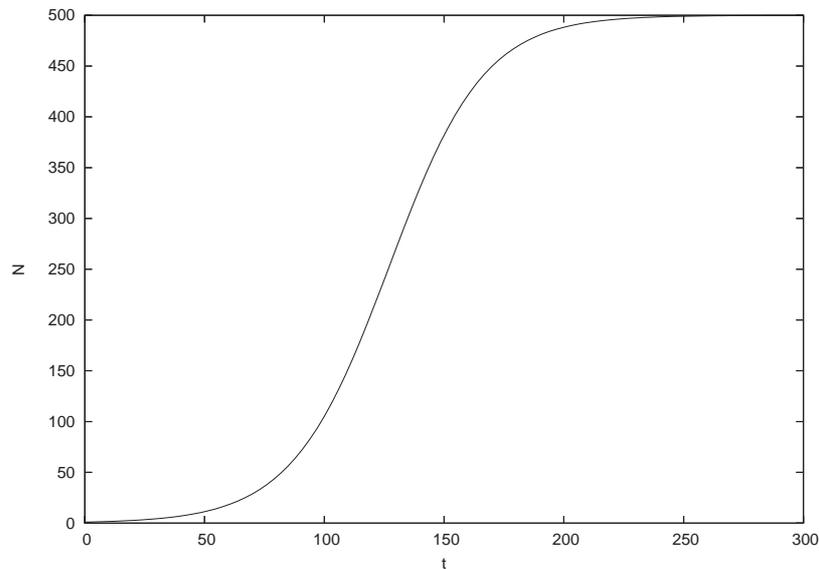


Abbildung 4.1.: Ein Wachstumsprozess nach der logistischen Gleichung mit $K = 500, r = 0.05, N(0) = 1$

```

05  max_interactions = if n < m then n else m
06  for i = 1 : max_interactions
07      with probability p_ab * density do
08          interact(a,b)
09      end
10  end
11  end
    
```

K gibt wie schon erwähnt die Partikelkapazität eines Gitterpunkts an. $p_{ab} \in [0, 1]$ ist ein Parameter, der die Basisgeschwindigkeit der Interaktion zwischen a und b beschreibt.

Sei $I_s(t)$ die Anzahl der Interaktionen zwischen den Partikeln des Typs a und b auf dem Feld s zum Zeitschritt t , $a(u, t)$ und $b(u, t)$ seien die Konzentrationen der Partikel der Typen a und b . Dann ist $I_s(t)$ eine binomialverteilte Zufallsvariable mit dem Erwartungswert:

$$E[I_s(t)] = p_{ab} \cdot \text{density} \cdot \text{max_interactions} = \frac{p_{ab}}{K} a(u, t) b(u, t)$$

Der Erwartungswert ist also proportional zu den Partikelkonzentrationen $a(u, t)$ und $b(u, t)$. Damit erfüllt der modifizierte Interaktionsalgorithmus die Massenwirkungsgesetze.

4.2.1. Behandlung affinitätsabhängiger Interaktionen

Durch die Modifikation des Algorithmus ist es nun notwendig, dass die Wahrscheinlichkeit p_{ab} a priori feststeht, da die vorhandenen Partikel lediglich gezählt, nicht aber individuell betrachtet werden. Affinitätsabhängige Interaktionen müssen daher nach den Spezifitäten gruppiert abgearbeitet werden. Für jedes Paar s_1, s_2 von Spezifitäten gilt dann $p_{ab} = a(s_1, s_2)$.

Dies soll am Beispiel der Phagozytose von Antigen durch B-Zellen erläutert werden, die von der Affinität des B-Zell-Rezeptors zum Epitop des Antigens abhängt. Dabei wird nach folgendem Prinzip verfahren:

```
01 procedure b_cell_phagocytosis ( s: site )
02   for each r in { receptors of B cells at s }
03     for each e in { epitopes of antigens at s }
04       interaction ( b cells with receptor r, antigens with epitope e, s )
05     end
06   end
07 end
```

4.3. Diffusion von Zellen und Molekülen

Der Diffusionsprozess für Zellen und Moleküle wird in LImmSim wie im CS-Modell durch eine simultane Brownsche Bewegung aller Partikel modelliert. Die modifizierte Diffusion für Zellen aus Abschnitt 3.2.1 kam zunächst nicht zum Einsatz, da sie für T- und B-Zellen nicht angebracht erschien. Ein erhöhter Diffusionsdruck auf diese Zellen bei größerer Zelldichte würde die Kostimulation zwischen den beiden Zelltypen erschweren.

Da im CS-Modell dem Diffusionsprozess generell keine hohe Bedeutung beigemessen wird, steuert dort ein einziger globaler Parameter die Diffusionsgeschwindigkeit für alle Partikeltypen. In LImmSim gibt es für jeden Partikeltyp einen separaten Parameter, so dass z.B. Antikörper als relativ kleines Molekül wesentlich schneller diffundiert als T- oder B-Zellen.

4.4. Beschleunigung durch hybride Simulation

Sowohl Diffusion als auch Proliferation und Interaktionen sind in LImmSim nun binomialverteilte Zufallsprozesse. Wir wissen aus dem schwachen Gesetz der großen Zahlen (vgl. etwa [Kal02], S.95), dass die relative Abweichung des Prozessausgangs vom Erwartungswert für große Partikelzahlen gegen 0 konvergiert. Um die zeitaufwändige Generierung von Zufallszahlen für jedes einzelne Partikel zu vermeiden, können wir ab einer genügend hohen Schranke den Mittelwert durch den Erwartungswert ersetzen, das Prozessergebnis also a priori abschätzen.

Durch diese schwellwertgesteuerte Abschätzung wird das Modell zu einem Hybrid: Im Bereich kleiner Partikelzahlen ist das Verhalten stark randomisiert, bei großen Partikelzahlen entspricht das Verhalten einer numerischen Integration der Differentialgleichungen für den Erwartungswert nach der FTCS-Methode (siehe Abschnitt 2.3.1). Hierbei muss allerdings beachtet werden, dass alle Konzentrationen ganzzahlig bleiben müssen, die einzusetzenden Erwartungswerte aber allgemein keine ganzen Zahlen sind.

4.4.1. Approximation des Diffusionsprozesses

Bei der Diffusion wird dieses Problem so gelöst, dass zunächst nur der durch 4 teilbare Anteil der zu transferierenden Partikel gleichmäßig auf die Nachbarfelder verteilt wird. Für die restlichen Partikel wird dann der exakte Diffusionsalgorithmus angewendet:¹

```

01 procedure approx_diffusion ( a: type, s: site )
02   c = D * #{ particles of type a at s }
03   with probability c - [c] do
04     move 1 particle of type a from s to random neighbour of s
05   end
06   c = [c]
07   for each t in N(s)
08     move [c/4] particles of type a from s to t
09   end
10   c = c mod 4
11   for i = 1 : c
12     move 1 particle of type a from s to random neighbour of s
13   end
14 end

```

Durch diesen Algorithmus werden also maximal 4 Partikel wirklich zufällig bewegt, der Rest wird gleichmäßig auf die Nachbarschaft verteilt. Dadurch wird bei großen Partikelzahlen gegenüber der simultanen Brownschen Bewegung Rechenzeit eingespart.

Sei $W_{uv}(t)$, $(u, v) \in E$, die Anzahl der Partikel, die durch den Diffusionsschritt zum Zeitpunkt t nach diesem Verfahren von u nach v verschoben werden. Der Erwartungswert ist

$$E[W_{uv}(t)] = D c(u, t)/4 \quad \forall v \in \mathcal{N}(u)$$

und entspricht damit dem der simultanen Brownschen Bewegung aus Abschnitt 3.2.1. Dieses Verfahren wird auch in CImmSim für die Diffusion von mehr als 10000 Molekülen verwendet.

¹Zum besseren Vergleich mit dem Algorithmus aus Kapitel 3 ist der Algorithmus hier für die 4-Nachbarschaft angegeben. In LImmSim wird die 6-Nachbarschaft des CS-Modells verwendet.

In LImmSim wird zwischen der exakten Brownschen Bewegung und der Approximation umgeschaltet, wenn die Anzahl der Partikel eines Typs auf dem Feld s einen Schwellwert θ_{diff} überschreitet. Mit dem Schwellwert kann so gegen einen Verlust von Randomisierung die Geschwindigkeit der Simulation erheblich gesteigert werden.

4.4.2. Approximation des Proliferationsprozesses

Auch der Proliferationsprozess kann beschleunigt werden, indem auf die Durchführung eines Zufallsexperiments für jedes Partikel verzichtet wird und als Erwartungswert eingesetzt wird. Nur mit einem Partikel muss ein Zufallsexperiment durchgeführt werden, um das Ergebnis nicht zu verfälschen, falls der Erwartungswert keine natürliche Zahl ist. Dies führt zu folgendem Algorithmus:

```

01 procedure approx_proliferation( a: type, s: site )
02    $p = v_a * (1.0 - \#\{\text{particles of type a at s}\} / K)$ 
03    $c = p * \#\{\text{particles of type a at s}\}$ 
04   if  $p > 0$ 
05     for  $i = 1 : \lfloor c \rfloor$  do
06       clone 1 particle of type a
07     end
08     with probability  $c - \lfloor c \rfloor$  do
09       clone 1 particle of type a
10     end
11   else
12     for  $i = 1 : \lfloor -c \rfloor$  do
13       kill 1 particle of type a
14     end
15     with probability  $-c - \lfloor -c \rfloor$  do
16       kill 1 particle of type a
17     end
18   endif
19 end

```

Für den Erwartungswert der Anzahl $N_u(t)$ neu erzeugter Zellen auf dem Feld u zum Zeitpunkt t gilt auch hier

$$E[N_u(t)] = v_a c(u, t) (1 - c(u, t) / K)$$

und damit bleibt als Differentialgleichungsform die logistische Gleichung

$$\frac{d\mathcal{C}}{dt} = v_a \mathcal{C} (1 - \mathcal{C}/K)$$

erhalten. In LImmSim wird dieser approximierte Proliferationsalgorithmus angewendet, wenn die Zahl der Partikel eines Typs auf einem Feld den Schwellwert θ_{inter} überschreitet.

4.4.3. Approximation des Interaktionsprozesses

Da im Gegensatz zum CS-Modell das Mittelwertverhalten des in Abschnitt 4.2 angegebenen Interaktionsalgorithmus von LImmSim durch eine geschlossene Formel ausdrückbar ist, kann auch dieser wie der Diffusionsprozess und der Proliferationsprozess für große Zahlen von Partikeln beschleunigt werden. Dies geschieht durch den folgenden Algorithmus:

```

01 procedure approx_interaction ( a: type, b: type, s: site )
02   n = #{ particles of type a at s }
03   m = #{ particles of type b at s }
04   density = if n > m then n/K else m/K
05   max_interactions = if n < m then n else m
06   c = pab * density * max_interactions
07   for i = 1 : [c]
08     interact(a,b)
09   end
10   with probability c - [c] do
11     interact(a,b)
12   end
13 end

```

Die Anzahl $I_u(t)$ der bei einem Durchlauf dieses Algorithmus auf dem Feld u zum Zeitpunkt t stattfindenden Interaktionen ist also eine Zufallsvariable mit einem höchstens zweielementigen Ereignisraum. Der Erwartungswert ergibt sich zu

$$E[I_u(t)] = p_{ab} \cdot \text{density} \cdot \text{max_interactions} = \frac{p_{ab}}{K} a(u, t) b(u, t)$$

und entspricht daher genau dem des Algorithmus aus Abschnitt 4.2.

In LImmSim wird die Umschaltung zwischen dem exakten und dem approximierten Interaktionsprozess ebenfalls schwellwertgesteuert durchgeführt. Der Schwellwert θ_{inter} wird hier mit der maximalen Zahl der Interaktionen, `max_interactions`, verglichen.

Damit wurden alle Änderungen, die in LImmSim gegenüber dem originalen CS-Modell vorgenommen wurden, vorgestellt. Durch die Einhaltung biophysikalischer Gesetzmäßigkeiten

im Erwartungswert wurde die Plausibilität des Modells verbessert. Die Kenntnis der Differentialgleichungen für den Erwartungswert ermöglicht eine hybride Simulation, mit der die Performance vermutlich gesteigert werden kann. Es stellt sich nun die Frage, wie das qualitative Verhalten der Simulation durch die Abwandlung des Modells beeinflusst wird. Um dies herauszufinden, wurde das abgewandelte Modell implementiert und getestet. Die Ergebnisse werden im nächsten Kapitel vorgestellt.

5. Ergebnisse und Diskussion

5.1. LImmSim – Implementierung des erweiterten CS-Modells

Die im letzten Kapitel vorgestellten Erweiterungen des CS-Modells wurden unter dem Namen *LImmSim* implementiert. Die wichtigsten Designziele bei der Implementierung von LImmSim waren:

1. Objektorientierter Aufbau

Die objektorientierte Implementierung des Modells soll die schnelle und einfache Erweiterung um neue Arten von Zellen und Molekülen ermöglichen. Im Rahmen dieser Diplomarbeit wurden zunächst nur die Zellen und Moleküle des ursprünglichen CS-Modells [SC92] berücksichtigt.

2. Multithreading-Fähigkeit

Zur Durchführung aufwändiger Berechnungen steht an der Universität zu Lübeck die *Mayflower*, ein Sun Fire 15K Multiprozessor-Rechner mit derzeit 72 Prozessoren je 800 MHz und 72 GB Hauptspeicher in einer Shared-Memory-Architektur zur Verfügung. Damit ein Programm auf einer solchen Architektur eine entsprechende Beschleunigung erfährt, muss die Arbeit auf mehrere Threads verteilt werden, die dann auf verschiedenen Prozessoren laufen können.

3. Effiziente Speichernutzung

Bei Simulationsläufen mit CImmSim fiel der relativ hohe Speicherverbrauch auf, durch den der Simulationsraum stark eingeschränkt ist. In der Version 6.3 von CImmSim kann ohne Änderungen am Quelltext höchstens ein Gitter von 96 mal 96 Feldern erzeugt werden. Dies entspricht einer initialen Populationsgröße von 94442 Zellen. Mit diesen Werten verbraucht CImmSim schon etwa 300 MB Speicher. Der Speicherbedarf wächst exponentiell in der Länge der zur Modellierung der Zellrezeptoren verwendeten Bitstrings, die daher momentan auf 12 beschränkt ist. Dies entspricht einer unrealistisch niedrigen Zahl von nur 4096 verschiedenen B-Zell-Spezifitäten. Für das menschliche Immunsystem wurde diese Zahl auf 10^8 geschätzt [Ton83], was einer Bitstring-Länge von 26 bis 28 Bit entsprechen würde ($2^{27} = 134217728$).

Bei der Implementierung von LImmSim sollte daher auf eine effiziente Nutzung des Hauptspeichers geachtet werden, damit eine Simulation in realistischer Größenordnung

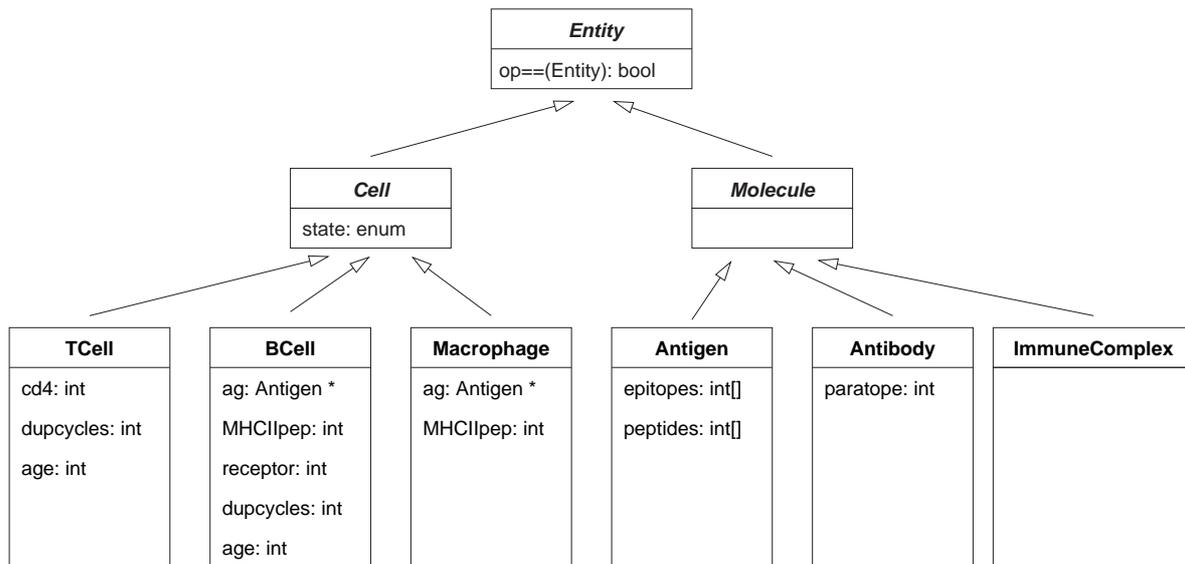


Abbildung 5.1.: Objektmodell von LImmSim. Die in kursiver Schrift dargestellten Klassen *Entity*, *Cell* und *Molecule* sind abstrakte Basisklassen. Als Basisklasse für *Antigen* wurde zunächst *Molecule* gewählt, da *Antigen* in LImmSim keinen internen Zustand hat. Es vermehrt sich aber trotzdem im simulierten Organismus, vergleichbar mit einem einfachen Bakterium.

möglich wird. Diese Zielsetzung steht allerdings teilweise in Konflikt zu einer effektiven Aufteilung des Programms in mehrere Threads.

5.1.1. Grundzüge der Implementierung

Als Programmiersprache wurde C++ gewählt, um bei größtmöglicher Kontrolle über die Speichernutzung gleichzeitig einen objektorientierten Entwurf durchführen zu können. Bei ersten Implementierungsversuchen mit Java konnte nicht die gewünschte Performance erzielt werden.

Das Objektmodell

Die Zellen und Moleküle wurden in einem einfachen hierarchischen Objektmodell angeordnet. Ausgehend von einer abstrakten Basisklasse *Entity* wurden zwei Hauptklassen *Cell* und *Molecule* definiert. Jede Unterklasse von *Entity* muss einen Vergleichsoperator implementieren, der auf semantische Gleichheit mit einem anderen Objekt der gleichen Klasse testet. Jede Unterklasse von *Cell* besitzt außerdem einen internen Zustand *state*. Ansonsten geben die Attribute der Klassen das CS-Modell wieder. Das vollständige Objektmodell ist in Abbildung 5.1 dargestellt.

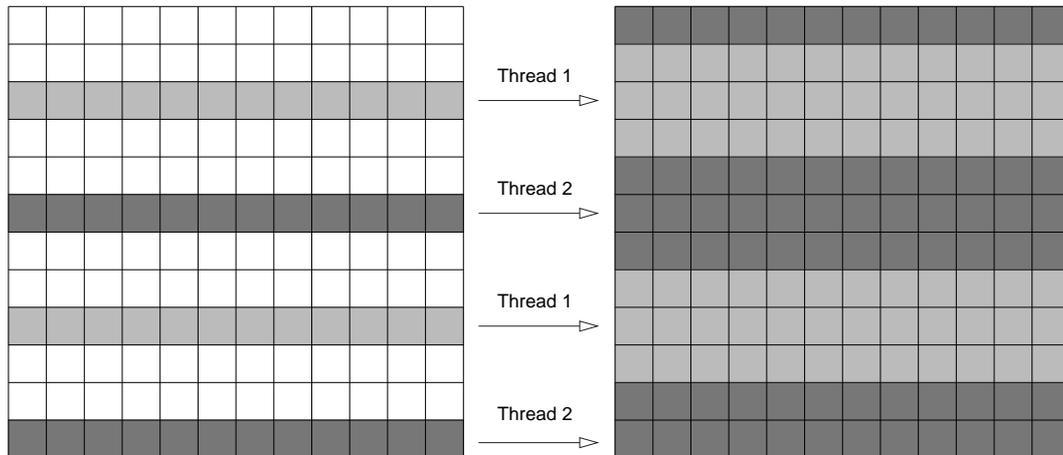


Abbildung 5.2.: Prinzip der Parallelisierung in LImmSim. Hier ist Phase 3 dargestellt, in der alle Zeilen i mit $i \equiv 2 \pmod{3}$ von zwei Threads abgearbeitet werden. Hell unterlegt sind die Zuständigkeitsbereiche des ersten Threads. Im linken Gitter sind die Zeilen markiert, aus denen gelesen wird, im rechten Gitter die Zeilen, in die nach Durchführung der Interaktionen und der Diffusion geschrieben wird.

Parallelisierung

Das CS-Modell lässt sich wegen der Lokalität der Interaktionen relativ gut parallelisieren bzw. auf mehrere Threads verteilen. Die Interaktionsprozesse können parallelisiert werden, indem die Zeilen des Gitters gleichmäßig auf die vorhandenen Threads verteilt werden. Bei der Diffusion ergibt sich das Problem, dass auch die Nachbarschaft eines Gitterpunkts durch den Ausgang der Diffusion beeinflusst wird und es bei Verwendung mehrerer Threads eventuell zu Schreibkonflikten kommen könnte.

Dieses Problem kann umgangen werden, indem die Durchführung eines Simulationsschritts in drei Phasen unterteilt wird. In jeder Phase werden dabei nur Zeilen behandelt, zwischen denen jeweils zwei unbenutzte Zeilen liegen, so dass sich die verschiedenen Threads nicht gegenseitig behindern können. Jeder Thread beschreibt dabei höchstens seine aktuelle Zeile sowie die Zeilen darüber und darunter. Zusätzlich wird das gesamte Gitter dupliziert, so dass die Ergebnisse der Reaktionen und Diffusionen in einem zweiten Gitter eingetragen werden können. Nach Durchführung des Simulationsschritts wird dieses Gitter dann zum Ausgangsgitter für den nächsten Schritt. Abbildung 5.2 zeigt schematisch dieses Parallelisierungsprinzip.

Speicherverwaltung

Durch die Verwendung globaler Lookup-Hashtabellen für alle Objekte wird erreicht, dass kein Objekt mehr als ein Mal im Speicher erzeugt werden muss. An den Gitterpunkten selbst wird

nur mit Zeigern gearbeitet, die auf die eigentlichen Objekte in den globalen Hashtabellen verweisen. Dadurch ergibt sich bei starker Vermehrung eines einzelnen Objekts, wie des Antigens oder einer B-Zelle mit einem bestimmten Rezeptortyp, kein erhöhter Speicherbedarf im Laufe der Simulation.

Ein Problem tritt auf, wenn durch Interaktionen Veränderungen an den Objekten vorgenommen werden. Diese müssen auch in den globalen Hashtabellen reflektiert werden. Bei der Verwendung mehrerer Threads kann es hierbei zu Konflikten kommen. Im Moment wird dieses Problem durch Semaphoren gelöst, die vor einer Änderung der globalen Hashtabellen reserviert werden müssen. Für jeden Hashwert wird dabei eine separate Semaphore verwendet, so dass meistens trotzdem von mehreren Threads gleichzeitig auf die Hashtabellen zugegriffen werden kann. Trotzdem kommt es im Verlauf der Simulation gelegentlich zu Engpässen.

Modellparameter

Gegenüber CImmSim wurde LImmSim durch eine Reihe von Parametern erweitert. So können zum Beispiel die Diffusionsgeschwindigkeiten der Partikeltypen individuell eingestellt werden. Zwei Schwellwerte steuern die Approximation der probabilistischen Interaktions-, Proliferations- und Diffusionsprozesse durch den Erwartungswert. Für jede Interaktion in LImmSim gibt es einen Parameter, mit dem die Basisgeschwindigkeit des Vorgangs gegenüber den anderen Vorgängen im Modell kalibriert werden kann. Die Rezeptor- und Antigendiversität kann mit den in Abschnitt 2.1.1 vorgestellten Parametern eingestellt werden.

In der folgenden Tabelle sind alle Modellparameter von LImmSim aufgelistet.

<i>Parameter</i>	<i>Standard</i>	<i>Bedeutung</i>
<code>threads</code>	2	Anzahl von Threads für die Parallelisierung
<code>gridsize</code>	10	Ausdehnung des Gitters in x und y -Richtung
<code>timesteps</code>	2000	Anzahl der Simulationsschritte
<code>nbitstr</code>	16	Länge der Bitstrings, die Antigen- und Rezeptordiversität abbilden
<code>minmatch</code>	14	Minimaler Hamming-Abstand zwischen zwei Bitstrings für eine Affinität > 0
<code>afflevel</code>	0.05	Affinität von zwei Bitstrings mit Hamming-Abstand <code>minmatch</code>
<code>affenhance</code>	0.77	Parameter zur Erhöhung der Affinität bei steigendem Hamming-Abstand

<i>Parameter</i>	<i>Standard</i>	<i>Bedeutung</i>
mhc	123	MHC Klasse II-Bitstring des Organismus, relevant für die Präsentation von Antigen durch B-Zellen und Makrophagen
capacity	512	Maximale Aufnahmekapazität eines Gitterpunkts pro Zelltyp, gilt auch für Antigen
init_macrophages	130	Initiale Anzahl von Makrophagen pro Gitterpunkt
init_bcells	50	Initiale Anzahl von B-Zellen pro Gitterpunkt
init_tcells	50	Initiale Anzahl von T-Zellen pro Gitterpunkt
ag_epitopes	1	Anzahl der Epitope pro Antigen (mehr als 1 derzeit nicht möglich)
ag_peptides	3	Anzahl der Peptide pro Antigen
v_prolif_b	0.05	Proliferationsgeschwindigkeit (neue Zellen pro Zeitschritt) der stimulierten B-Zellen
v_prolif_t	0.025	Proliferationsgeschwindigkeit der stimulierten T-Zellen
v_prolif_ag	0.025	Proliferationsgeschwindigkeit des Antigens
vdiff_ag	0.1	Diffusionsgeschwindigkeit des Antigens
vdiff_ic	0.1	Diffusionsgeschwindigkeit der Immunkomplexe
vdiff_ab	0.5	Diffusionsgeschwindigkeit des Antikörpers
vdiff_t	0.02	Diffusionsgeschwindigkeit der T-Zellen
vdiff_b	0.02	Diffusionsgeschwindigkeit der B-Zellen
vdiff_ma	0.01	Diffusionsgeschwindigkeit der Makrophagen
tau_ag	10000	Mittlere Lebenszeit des Antigens
tau_ic	1000	Mittlere Lebenszeit der Immunkomplexe
tau_ab	690	Mittlere Lebenszeit des Antikörpers
tau_t	200	Mittlere Lebenszeit aktiver T-Zellen
tau_t_armed	50	Mittlere Lebenszeit bewaffneter T-Zellen
tau_b	200	Mittlere Lebenszeit aktiver B-Zellen
tau_plasma	100	Mittlere Lebenszeit von Plasmazellen

<i>Parameter</i>	<i>Standard</i>	<i>Bedeutung</i>
p_ma_ag	0.0005	Geschwindigkeit der Phagozytose von Antigen durch Makrophagen
p_ma_ic	1	Geschwindigkeit der Phagozytose von Immunkomplexen durch Makrophagen
p_b_ag	1	Basisgeschwindigkeit der Phagozytose von Antigen durch B-Zellen
p_b_t	3	Basisgeschwindigkeit der Kostimulation von T- und B-Zellen
p_ma_t	0.8	Basisgeschwindigkeit der Stimulation von T-Zellen durch Makrophagen
p_remove_ag	0.01	Wahrscheinlichkeit, dass ein von einer B-Zelle oder Makrophage präsentierte Antigen während eines Zeitschritts entfernt wird
p_ag_ab	0.5	Basisgeschwindigkeit der Bindung von Antikörper an Antigen
th_diffusion	100	Schwellwert für die Anwendung des approximierten Diffusionsalgorithmus
th_reaction	100	Schwellwert für die Anwendung der approximierten Interaktions- und Proliferationsalgorithmen
prolif_cycles_t	6	Zahl der Proliferationszyklen, die T-Zellen durchlaufen, bevor sie sich zu bewaffneten T-Zellen entwickeln
prolif_cycles_b	6	Zahl der Proliferationszyklen, die B-Zellen durchlaufen, bevor sie sich zu Plasmazellen entwickeln
ab_secrete	16	Zahl der Antikörpermoleküle, die eine Plasma-Zelle pro Zeitschritt sezerniert

Tabelle 5.1.: Modellparameter in LImmSim

Die Interaktionen wurden gemäß der Beschreibungen in [SC92] mit den in Kapitel 4 erläuterten Änderungen implementiert. Dabei wurden die somatische Hypermutation und die Ausbildung von Gedächtniszellen zunächst weggelassen. Weiterhin wurde auf die Anergie (Deaktivierung) von B- und T-Zellen bei starker Immunaktivität in der Umgebung verzichtet,

da für diesen Vorgang in den aktuellen immunologischen Standardwerken [JTW02, RBM95] kein Beleg gefunden werden konnte.

5.1.2. Experimente

Anhand der durchgeführten Experimente mit LImmSim sollten die folgenden Fragen geklärt werden:

- Kann mit LImmSim ebenso wie mit dem CS-Modell eine Immunantwort simuliert werden?
- Welche Konsequenzen haben die vorgenommenen Modifikationen an den Interaktions- und Proliferationsalgorithmen auf die simulierte Immunantwort?
- Kann durch die hybride Simulation eine Verbesserung der Performance erreicht werden? Wie sind die Auswirkungen auf die simulierte Immunantwort?
- Wie gut skaliert LImmSim bei der Verwendung mehrerer Prozessoren?

Zur Klärung dieser Fragen wurden mehrere Testläufe mit LImmSim durchgeführt, deren Ergebnisse im Folgenden dargestellt werden. Soweit nicht anders angegeben, wurden dabei als Parameter die Standardwerte aus Tabelle 5.1 eingesetzt.

Simulation einer Immunantwort mit LImmSim

Beim ersten Testlauf wurden 3000 Simulationsschritte auf einem 180 mal 180 Felder großen Gitter durchgeführt. Die Ergebnisse dieses Testlaufs sind in den Abbildungen 5.3 bis 5.7 dargestellt. Es zeigte sich, dass auch mit den in Kapitel 4 vorgestellten Änderungen der typische Verlauf einer Immunantwort simuliert wurde. In der Anfangsphase konnte sich das Antigen auf dem Gitter ungehindert verbreiten und wurde zunächst nur von Makrophagen und T-Zellen erkannt. Durch die verwendeten Parameter für Rezeptordiversität und Affinität betrug die Wahrscheinlichkeit, auf einem gegebenen Gitterpunkt eine zum Antigen affine B-Zelle anzutreffen, etwa 36 Prozent. Dementsprechend dauerte es einige Zeit, nämlich 511 Simulationsschritte, bis die erste B-Zelle das Aktivierungssignal von einer T-Zelle erhielt. Nach 573 Zeitschritten kam es zur Bildung der ersten Antikörper. In Abbildung 5.7 ist gut sichtbar, wie zu Beginn der Immunantwort nur an wenigen Stellen des Gitters Antikörper gebildet wird. Im weiteren Verlauf stieg der Antikörperspiegel in etwa auf das Zehnfache des Antigenspiegels an. Der höchste Wert des Antikörperspiegels wurde nach 2154 Schritten erreicht. Zu diesem Zeitpunkt war bereits der größte Teil des Antigens mit Antikörper markiert und von den Makrophagen vernichtet worden. Nach 2381 Zeitschritten war das Antigen vollständig entfernt worden.

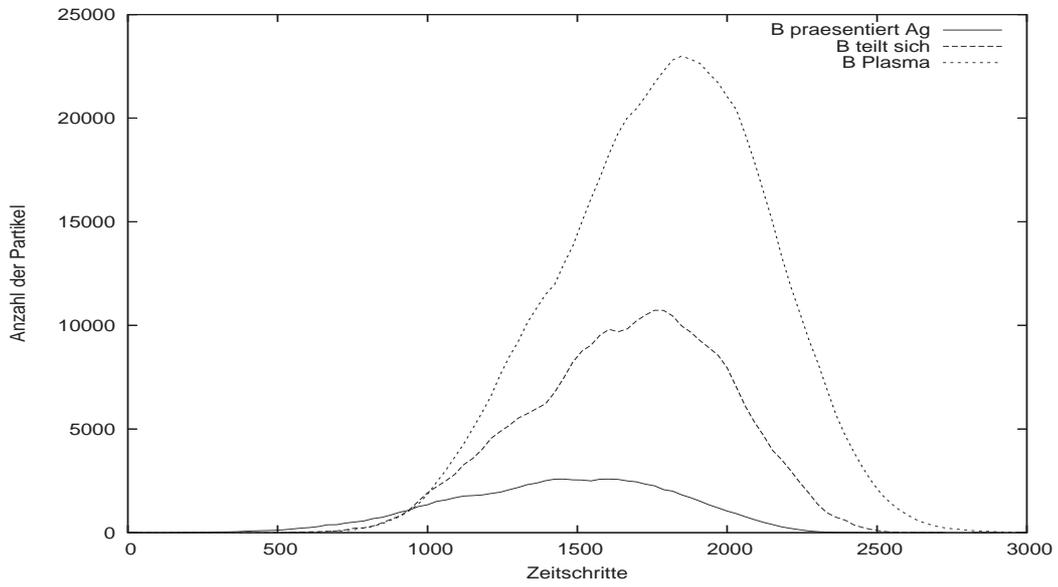


Abbildung 5.3.: Immunantwort in LImmSim: Aktivierung von B-Zellen

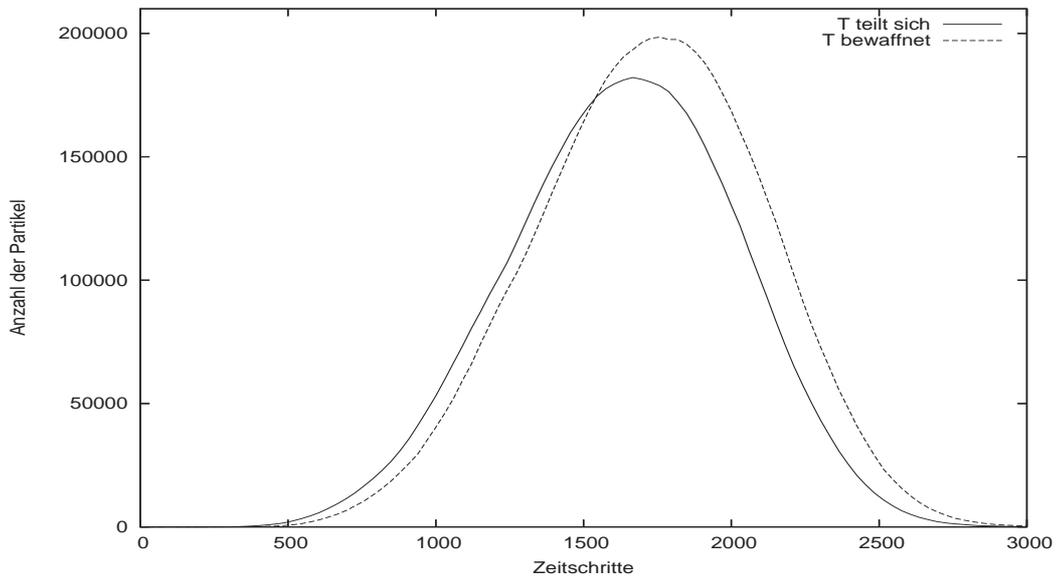


Abbildung 5.4.: Immunantwort in LImmSim: Aktivierung von T-Zellen

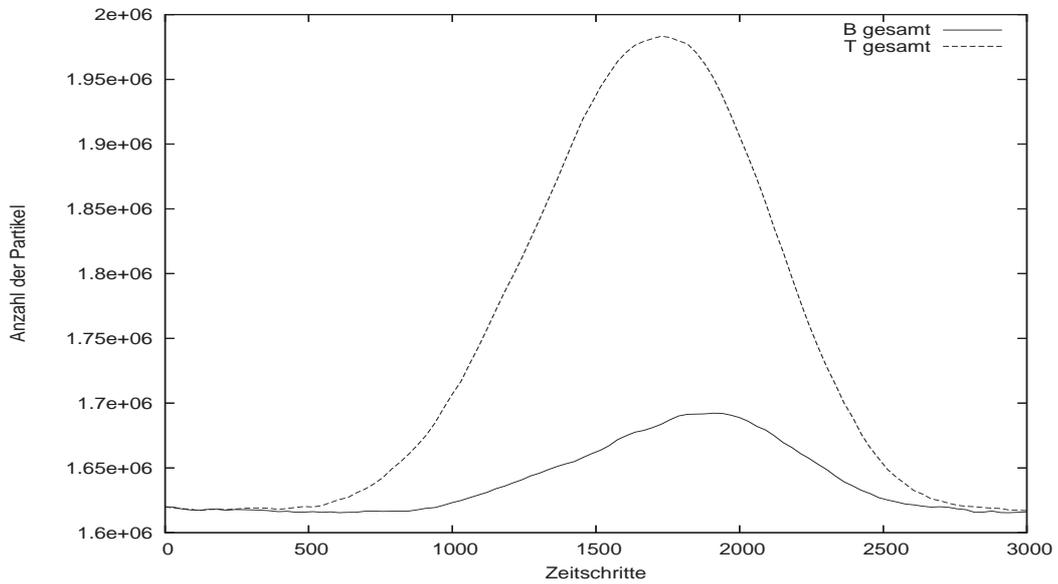


Abbildung 5.5.: Immunantwort in LImmSim: Gesamtzahl der B- und T-Zellen

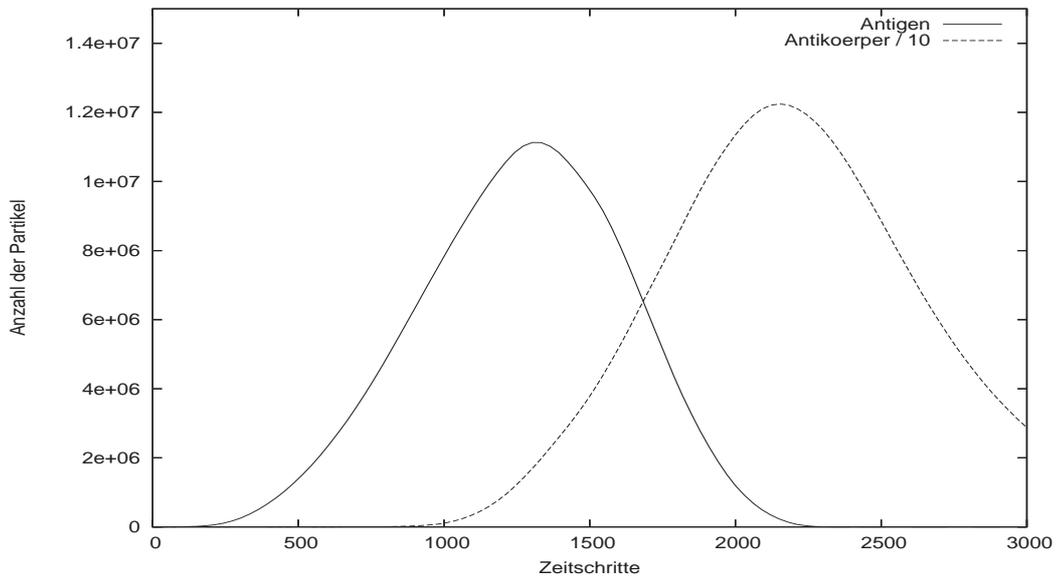


Abbildung 5.6.: Immunantwort in LImmSim: Spiegel von Antigen und Antikörper

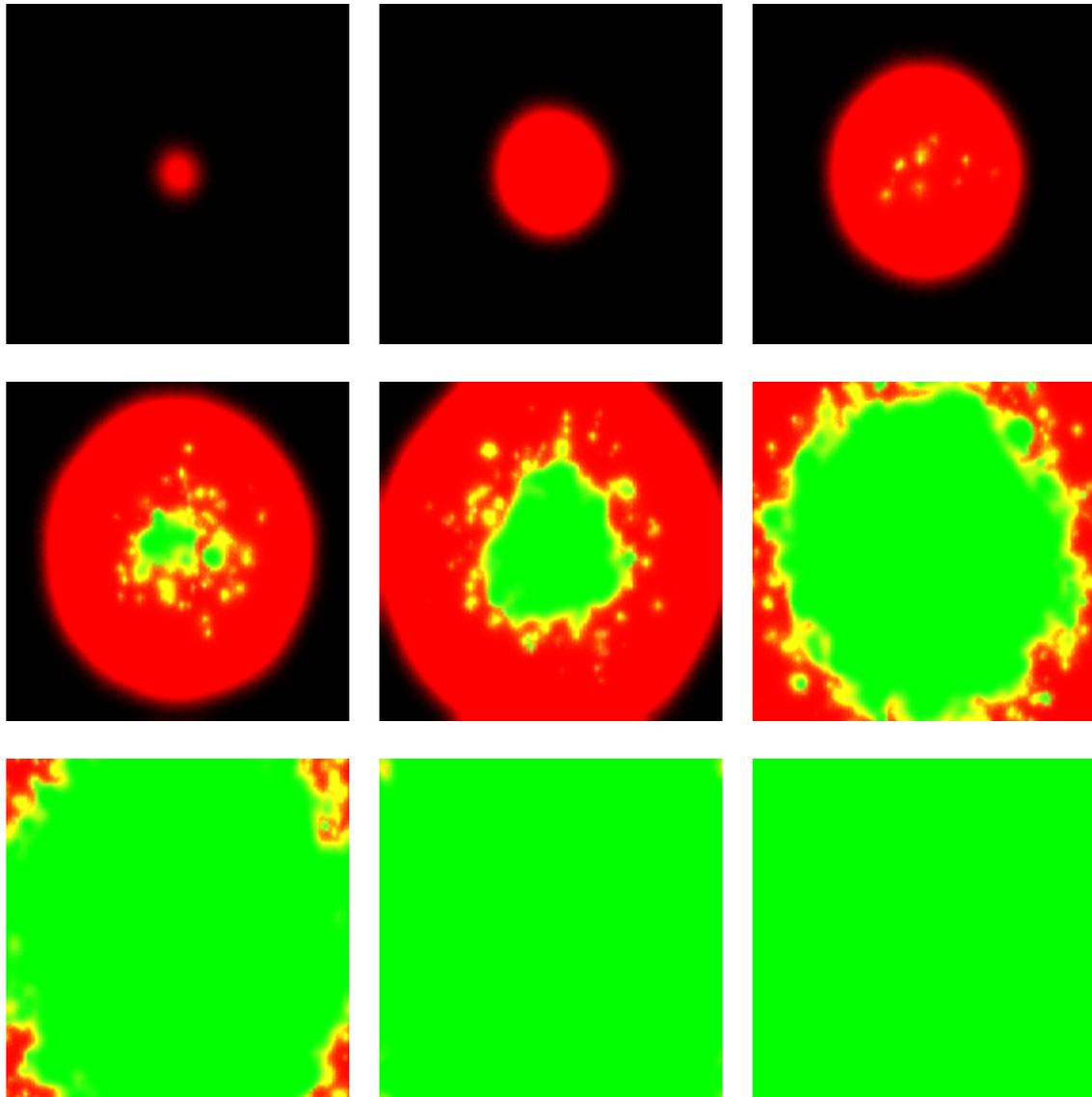


Abbildung 5.7.: Verteilung von Antigen und Antikörper im Raum während des Simulationsverlaufs nach 250, 500, 750, 1000, 1250, 1500, 1750, 2000 und 2250 Schritten. Nach 750 Schritten sind die ersten “Keimzentren” sichtbar, wo B- und T-Zellen das Antigen erkannt haben und es zur Kostimulation und Ausbildung von Antikörper sezernierenden Plasmazellen gekommen ist. Nach 1000 Schritten ist das Antigen in der Mitte des Simulationsraums bereits weitgehend vernichtet. Nach 2250 Schritten ist die Infektion vollständig überwunden. Der Antikörper bleibt im Simulationsraum prävalent. Die Population der B- und T-Zellen hat sich zu diesem Zeitpunkt bereits weitestgehend wieder zurückgebildet (hier nicht sichtbar, siehe dazu Abbildung 5.5).

Auswirkungen der hybriden Simulation

Als nächstes wurden auf Gittern der Größe 10 mal 10 und 40 mal 40 je 10 Testläufe durchgeführt. Dabei wurden die Schwellwerte zur Approximation der Interaktions-, Proliferations- und Diffusionsalgorithmen durch den Erwartungswert je einmal auf dem Standardwert 100 belassen und einmal auf den Wert 512 gesetzt, der der maximalen Aufnahmekapazität eines Gitterpunkts entspricht. Durch die Verwendung des Schwellwerts von 512 wird die hybride Simulation größtenteils deaktiviert.

Die Ergebnisse der Testlaufreihen sind in den Tabellen 5.2 bis 5.5 dargestellt. Für jeden Simulationslauf sind die Zeitpunkte der erstmaligen Bildung von Antikörper und der vollständigen Vernichtung des Antigens sowie die Maximalspiegel der B-Zellen, T-Zellen, Antigen und Antikörper angegeben. Zudem wurde die Laufzeit der Simulation mit dem Unix-Befehl `time` gemessen. Die Durchführung der Simulationen erfolgte auf einem handelsüblichen PC mit einer Taktfrequenz von 2GHz. Für jede gemessene Größe wurden der Mittelwert μ und die Standardabweichung σ ermittelt.

Die Resultate zeigen, dass bei der Verwendung der Approximation durch den Mittelwert für große Partikelzahlen das qualitative Simulationsverhalten erhalten bleibt. Überraschend ist, dass die Verwendung der hybriden Simulation eine *höhere* Varianz der Messwerte zur Folge zu haben scheint. Hier würde man eher erwarten, dass die geringere Randomisierung zu einer geringeren Schwankung der Messwerte führt.

Durch die Verwendung der approximierten Simulationsprozesse konnte die mittlere Geschwindigkeit bei der Gittergröße von 10 mal 10 um 58 Prozent, bei der Gittergröße von 40 mal 40 um 97 Prozent gesteigert werden. Die Zeiteinsparung durch die Approximation ist in der späteren Phase der Simulation am größten, wenn der Antikörperspiegel seinen Maximalwert erreicht.

Effektivität der Parallelisierung

Um die Effektivität des Multithreadings in LImmSim zu testen, wurden jeweils 10 Testläufe auf einem 192 mal 192 Felder großen Gitter mit verschiedenen Anzahlen von Threads durchgeführt. Die Gittergröße wurde so gewählt, da auf dem Parallelrechner *Mayflower* der Universität zu Lübeck 64 von 72 Prozessoren zur freien Verfügung stehen. Durch die Unterteilung jedes Simulationsschrittes in 3 Phasen werden also mindestens 192 Zeilen benötigt, damit jeder Thread in jeder Phase mindestens eine Zeile erhält. In jedem Testlauf wurden 100 Simulationsschritte durchgeführt. Die Mittelwerte der Laufzeiten ergaben sich wie folgt (siehe auch Abbildung 5.8):

<i>Lauf</i>	<i>Zeit/s</i>	<i>Erster Ab</i>	<i>Ag vernichtet</i>	<i>Max. Ag</i>	<i>Max. B</i>	<i>Max. T</i>	<i>Max. Ab</i>
1	54	469	796	50902	7411	10783	2494584
2	55	926	1202	50971	6415	7742	1658143
3	50	597	1126	50946	5770	6670	934892
4	52	340	741	50086	7243	9750	2453614
5	62	471	694	50889	9192	13758	4550016
6	52	705	1277	50971	5557	6191	594443
7	55	546	810	50932	6441	9248	1753319
8	53	600	932	50955	6438	9058	1896623
9	51	601	988	50945	5919	7892	1049920
10	49	634	927	50964	5883	7588	888509
μ	53	589	949	50856	6627	8868	1827406
σ	3	149	189	258	1031	2107	1099618

Tabelle 5.2.: Ergebnisse von 10 Testläufen mit LImmSim je 2000 Schritte bei einer Gittergröße von 10x10 Feldern und Verwendung der hybriden Simulation ab 100 Partikeln/Interaktionen

<i>Lauf</i>	<i>Zeit/s</i>	<i>Erster Ab</i>	<i>Ag vernichtet</i>	<i>Max. Ag</i>	<i>Max. B</i>	<i>Max. T</i>	<i>Max. Ab</i>
1	65	507	871	50861	6224	8383	1331766
2	93	596	990	50977	5860	8044	1069596
3	101	457	762	50844	7622	10870	2975102
4	83	744	1226	50945	5727	7173	715227
5	96	478	838	50854	6495	8568	1527684
6	69	593	1228	50943	5266	5870	282351
7	89	278	811	48562	6058	7995	1126456
8	85	569	838	50935	6407	9346	1567164
9	83	517	801	50913	6202	8278	1435343
10	72	525	828	50897	6510	9065	1724956
μ	84	526	919	50673	6237	8359	1375565
σ	11	113	164	705	589	1251	673259

Tabelle 5.3.: Ergebnisse von 10 Testläufen mit LImmSim je 2000 Schritte bei einer Gittergröße von 10x10 Feldern und Verwendung der hybriden Simulation ab 512 Partikeln/Interaktionen

<i>Lauf</i>	<i>Zeit/s</i>	<i>Erster Ab</i>	<i>Ag vernichtet</i>	<i>Max. Ag</i>	<i>Max. B</i>	<i>Max. T</i>	<i>Max. Ab</i>
1	1676	374	1390	779406	86554	107020	7958422
2	1733	648	1406	803919	86588	105882	8572627
3	1566	516	1306	788802	86690	104616	8181279
4	2627	485	1169	710807	94546	151964	19510467
5	2888	289	1068	669800	99915	174820	27850507
6	1808	455	1349	794878	86172	103653	8003319
7	2148	642	1253	803788	94658	142557	18853292
8	1954	585	1186	792823	92521	122286	15395447
9	1529	593	1393	804611	86261	103986	7469979
10	1990	562	1234	787421	91312	121813	14224532
μ	1992	515	1275	773626	90522	123860	13601987
σ	428	111	107	43345	4573	23502	6516024

Tabelle 5.4.: Ergebnisse von 10 Testläufen mit LImmSim je 2000 Schritte bei einer Gittergröße von 40x40 Feldern und Verwendung der hybriden Simulation ab 100 Partikeln/Interaktionen

<i>Lauf</i>	<i>Zeit/s</i>	<i>Erster Ab</i>	<i>Ag vernichtet</i>	<i>Max. Ag</i>	<i>Max. B</i>	<i>Max. T</i>	<i>Max. Ab</i>
1	2809	516	1257	766145	86220	105595	7666052
2	4903	581	1251	798765	94997	134400	17063286
3	3811	566	1251	784458	89353	120844	11355614
4	4229	570	1148	769348	93405	120125	15578134
5	4132	599	1194	785721	92747	132117	14152273
6	3481	593	1234	782522	89835	119365	12088058
7	2773	606	1361	807438	88540	106308	9033954
8	3138	538	1329	783382	88416	120005	9796272
9	4144	440	1208	791792	92038	125163	14059227
10	5904	469	1166	745457	96875	154803	20859629
μ	3932	548	1240	781503	91243	123873	13165250
σ	920	54	63	16705	3154	13558	3809035

Tabelle 5.5.: Ergebnisse von 10 Testläufen mit LImmSim je 2000 Schritte bei einer Gittergröße von 40x40 Feldern und Verwendung der hybriden Simulation ab 512 Partikeln/Interaktionen

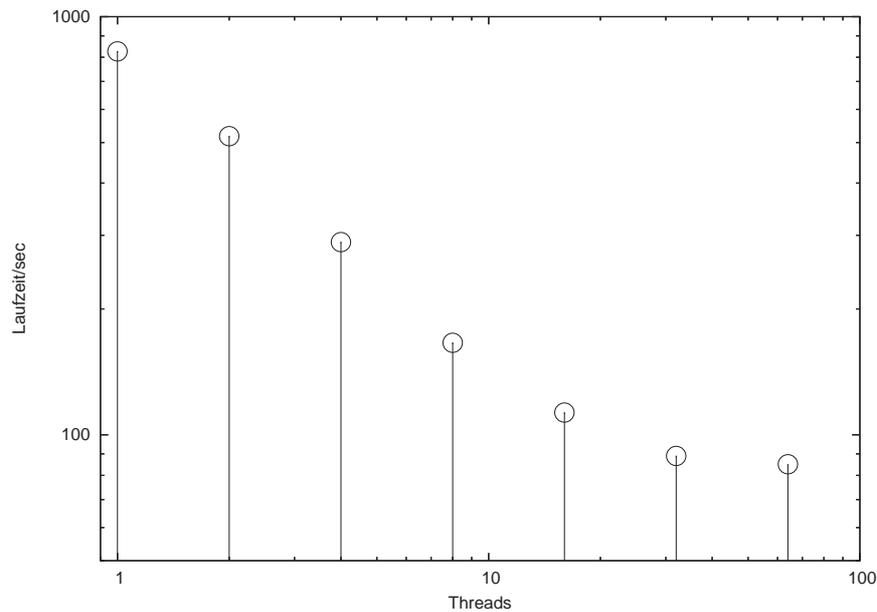


Abbildung 5.8.: Mittelwerte der Laufzeiten von 100 Simulationsschritten mit LImmSim über 10 Testläufe auf einem 192 mal 192 Felder großen Gitter bei der Verwendung von 1,2,4,8,16,32 und 64 Threads.

Threads	1	2	4	8	16	32	64
Laufzeit/sec	826	518	289	166	113	89	85

Bis zu einer Zahl von 16 Threads kam es also zu einer zufriedenstellenden Steigerung der Laufzeit. Ab 16 Threads aufwärts war die Steigerung nicht mehr ganz so effizient. Die Verwendung von 64 Threads bringt keinen nennenswerten Vorteil gegenüber 32 Threads mehr. Die Effizienzverluste dürften durch die oben erläuterte Verwendung globaler Caches zur Einsparung von Speicher zu erklären sein. Werden mehr Threads verwendet, so steigt die Wahrscheinlichkeit gegenseitiger Blockade.

5.2. Zusammenfassung

Die Ergebnisse der Testläufe mit LImmSim waren insgesamt zufriedenstellend. Auch mit den veränderten Proliferations- und Interaktionsalgorithmen wies das Modell das qualitative Verhalten einer Immunantwort auf. Durch die Änderungen wurde das Modell um zahlreiche Parameter erweitert, die zum Teil direkte Entsprechungen in der Realität haben (Diffusionsgeschwindigkeiten und Geschwindigkeitsparameter der unspezifischen Interaktionen). Auch bei der schwellwertgesteuerten Approximation der probabilistischen Prozesse durch den Erwartungswert blieb das qualitative Verhalten bestehen, es kam allerdings zu einer noch zu

klärenden Steigerung der Messwertvarianzen. Durch die hybride Simulation konnte die Geschwindigkeit der Simulation je nach Größe des Simulationsraums in etwa verdoppelt werden. Die Parallelisierung der Simulation auf einer Shared-Memory-Architektur brachte ebenfalls erhebliche Geschwindigkeitssteigerungen. Hier lag der Effizienzgrad allerdings wegen der Verwendung von Semaphoren für den Zugriff auf die globalen Caches unterhalb des theoretischen Maximums.

6. Zusammenfassung und Ausblick

6.1. Ergebnisse der Arbeit

Zu Beginn der Arbeit wurden die wichtigsten mathematischen und informatischen Modelle des Immunsystems aus der Literatur zusammengetragen. Es zeigte sich, dass zur Modellierung des Immunsystems hauptsächlich agentenbasierte Systeme und Differentialgleichungen eingesetzt werden. Der Schwerpunkt der weiteren Betrachtungen war das bekannteste und wichtigste agentenbasierte Modell des Immunsystems, das Celada-Seiden-Modell. Dabei handelt es sich um einen probabilistischen Automaten, dessen Aufbau einem zellulären Automaten ähnelt.

Daraufhin wurde eine Analyse der im Celada-Seiden-Modell implementierten Simulationsprozesse mit mathematischen Mitteln durchgeführt. Ziel dieser Analyse war es, festzustellen, ob die implementierten Prozesse biophysikalisch plausibel sind. Die dabei verwandte Methodik war inspiriert durch die dem Celada-Seiden-Modell ähnlichen Reaktions-Diffusions-Systeme aus der Chemie: Für das erwartete Verhalten der probabilistischen Simulationsprozesse wurden Differentialgleichungen hergeleitet, die dann mit den Naturgesetzen verglichen wurden, die für die simulierten Prozesse gelten.

Durch diese Analyse ergab sich, dass die Implementierung der Diffusion im Erwartungswert den physikalischen Gesetzen für Diffusionsprozesse genügt. In "CImmSim", der Open-Source-Implementierung des Celada-Seiden-Modells, wird für die Diffusion von Zellen allerdings ein abgewandelter Algorithmus verwendet, der ein fehlerhaftes erwartetes Verhalten aufweist. Den Algorithmen für die Simulation der Interaktions- und Proliferationsprozesse im Celada-Seiden-Modell konnte dagegen keine unmittelbare biophysikalische Bedeutung zugeordnet werden.

Im nächsten Schritt wurden für die Interaktions- und Proliferationsprozesse alternative Algorithmen angegeben. Der alternative Algorithmus zur Proliferation stellt sicher, dass pro Gitterpunkt eine gegebene Kapazität an Zellen nicht überschritten wird. Dies wird durch die Verwendung der logistischen Gleichung als Wachstumsmodell erreicht. Dadurch wurde es möglich, für die Interaktionen einen Algorithmus anzugeben, der die Massenwirkungsgesetze erfüllt. Durch die Einhaltung dieser Gesetzmäßigkeiten wird das abgewandelte Modell biophysikalisch plausibler. Weiterhin wird eine Steigerung der Performance durch hybride Simulation möglich. Dabei wird für große Partikelzahlen das Ergebnis der probabilistischen Simulationsprozesse durch den Erwartungswert abgeschätzt.

Abschließend wurde das erweiterte Celada-Seiden-Modell implementiert und getestet. Auch

unter Verwendung der modifizierten Algorithmen und der hybriden Simulation konnte eine Immunantwort simuliert werden. Da bei der Implementierung besonderer Wert auf eine effiziente Speichernutzung gelegt wurde, konnte eine wesentlich größere Menge an Zellen und eine höhere Antigendiversität simuliert werden als mit CImmSim. Auch die Performance konnte durch hybride Simulation und Parallelisierung gesteigert werden.

6.2. Ausblick

Mit der in dieser Arbeit vorgenommenen Erweiterung des Celada-Seiden-Modells sollte es möglich sein, die Modellparameter stärker als bisher mit Werten aus der Realität in Verbindung zu bringen. Dazu müssten die Modellparameter aus Sicht der Immunologie betrachtet und bewertet werden. Weiterhin sollte geklärt werden, ob die Verwendung der logistischen Gleichung als Modell für die Proliferation und der Massenwirkungsgesetze als Modell für die Interaktion plausibel ist, oder ob diese durch andere, komplexere Modelle ersetzt werden sollten.

Da das implementierte Modell auf der Arbeit von Celada und Seiden von 1992 [SC92] basiert, entsprechen die modellierten Regeln nicht mehr dem aktuellen Forschungsstand der Immunologie. Das Modell sollte daher um weitere Arten von Zellen und deren Funktionen ergänzt werden. Beispielsweise wäre eine Unterscheidung zwischen Th1- und Th2-Zellen und die Einbindung von T-Killerzellen und T-Suppressorzellen wünschenswert. Das zweidimensionale Gitter sollte durch ein dreidimensionales ersetzt werden, um den Realitätsbezug der Simulation zu steigern.

Momentan bewegen sich die Zellen und Moleküle im Simulationsraum nur zufällig. Im Immunsystem werden häufig Botenstoffe eingesetzt, um Lymphozyten an bestimmte Orte, etwa den Infektionsherd, zu locken. Diesen Vorgang nennt man *Chemotaxis*. Für die Chemotaxis existieren Differentialgleichungsmodelle, die im Wesentlichen Verallgemeinerungen der Gleichungen für Diffusion sind [Mur02]. Vermutlich könnte das Celada-Seiden-Modell durch entsprechende Änderungen an den Diffusionsalgorithmen recht einfach um Chemotaxis erweitert werden.

Ein weiterer wichtiger Schritt wäre die Unterteilung des simulierten Immunsystems in funktional verschiedene Zonen und Organe, wie Lymphknoten, Keimzentren, Blutkreislauf, Knochenmark, Thymus und Peripherie (Infektionsherd). Mittlerweile ist bekannt, dass viele Vorgänge der Immunabwehr auf bestimmte Organe beschränkt sind. So erfolgt zum Beispiel die somatische Hypermutation der B-Zellen nur in den Keimzentren der Lymphknoten [JTW02]. Direkte Interaktion mit dem Antigen findet dagegen vor allem am Infektionsherd statt. Eine solche Unterteilung könnte im simulierten System durch die Verwendung mehrerer Simulationsräume realisiert werden, die über Kanäle miteinander verbunden sind.

6.3. Fazit

Im Rahmen dieser Arbeit konnten auf dem langen Weg zu einer vollständigen Simulation des Immunsystems im Computer, anhand derer immunologische Forschung stattfinden könnte, nur vergleichsweise kleine Fortschritte erzielt werden. Es wurde jedoch erstmals eine Analyse des wichtigen Celada-Seiden-Modells mit mathematischen Mitteln durchgeführt und eine Verbindung zu den gegensätzlich erscheinenden Differentialgleichungsmodellen geschaffen. Diese Analyse ist aber keinesfalls als abgeschlossen zu betrachten. Die aufgezeigten Parallelen zu den Reaktions-Diffusions-Systemen aus der Chemie könnten auch für zukünftige, tiefergehende Analysen nützlich sein.

Es bleibt zu hoffen, dass sich der Dialog zwischen Immunologie und Informatik in Zukunft intensiviert, damit die Vision einer realistischen Simulation des Immunsystems im Computer eines Tages Wirklichkeit werden kann. Dazu gehören auf der einen Seite Informatiker, die bereit sind, sich in dieses komplexe Thema mit seinen vielen Details einzuarbeiten. Es gehören dazu aber auch Immunologen, die sich trotz des noch sehr frühen Stadiums der aktuellen Modelle dazu bereit erklären, an solchen Modellen mitzuarbeiten und sie mit ihren Erkenntnissen und Ideen zu bereichern. Das Beispiel der Zusammenarbeit von Franco Celada und Philip Seiden mag hier motivierend wirken: Ihr Modell ist auch 14 Jahre nach seiner Entstehung in seiner Bedeutung unübertroffen – eine Halbwertszeit der Erkenntnis, die für beide Fachgebiete eher ungewöhnlich ist.

A. Immunologische Begriffe und Abkürzungen

Antigen (Ag)	Bezeichnung für einen Stoff oder ein Lebewesen, dessen Vorhandensein im Körper eine Immunantwort hervorruft.
Antigenpräsentation	Vorgang, bei dem Zellen, die selbst nicht zwischen körpereigenen und körperfremden Substanzen unterscheiden können, Fragmente einer von ihnen verdauten Substanz auf der Zelloberfläche darbieten. Daraufhin kann von T-Helferzellen untersucht werden, ob die Fragmente von einem körperfremden, potenziell gefährlichen Stoff oder Lebewesen stammen.
Antikörper (Ab)	Vom Immunsystem hergestelltes Molekül, das sich spezifisch gegen eine Art von Antigen richtet. Antikörper kann an dieses Antigen binden und so dafür sorgen, dass die unspezifischen Abwehrzellen das so markierte Antigen angreifen und neutralisieren.
APC	Antigenpräsentierende Zelle, z.B. Makrophage oder dendritische Zelle. Notwendig zur Aktivierung der T-Zellen, die Antigen nicht direkt erkennen können, sondern es von APCs in verdauter Form präsentiert bekommen.
B-Zelle	B-Zellen stammen aus dem Knochenmark (b one marrow). Ihre Aufgabe ist die spezifische Erkennung von Antigenen. B-Zellen, die Antigen erkannt haben, benötigen die Hilfe von T-Zellen, um sich durch Zellteilung und Reifung zu Plasmazellen zu entwickeln. B-Zell-Rezeptoren sind hochspezifisch und werden durch zufällige Anordnung von Proteinen erzeugt. Durch die große Zahl verschiedener Rezeptoren ist das Immunsystem in der Lage, auch auf unbekanntes Antigen zu reagieren.
CS-Modell	Celada-Seiden-Modell. Das meistbekannte agentenbasierte Modell des Immunsystems [SC92].

CImmSim	C Immune System Simulator. Implementierung des CS-Modells von Filippo Castiglione. Derzeit die einzige Implementierung des CS-Modells mit frei verfügbarem Quelltext. Da Castiglione Mitglied der Forschungsgruppe um Philip Seiden war, kann davon ausgegangen werden, dass die Closed-Source-Versionen des Modells wie IMMSIM3 oder IMMSIM++ auf demselben oder einem ähnlichen Quelltext aufbauen.
Epitop	Charakteristische Form auf der Oberfläche eines Antigens, die dem Immunsystem als Erkennungsmerkmal für dieses Antigen dient.
Immunkomplex (IC)	Bezeichnung für ein Antigen, das an Antikörper gebunden hat (markiertes Antigen).
ImmSim	Erste Implementierung des CS-Modells von Philip Seiden in der Programmiersprache APL. Der Quelltext von ImmSim ist nicht verfügbar.
Komplementsystem	System mehrerer miteinander interagierender Proteine, die Abwehrmechanismen gegen viele Bakterien und Viren bereitstellen. Das Komplementsystem kann unter anderem durch Schädigung des Gewebes, aber auch durch Signale des spezifischen Immunsystems aktiviert werden.
Kostimulation	Signal zwischen B- und T-Zellen, die das gleiche Antigen erkannt haben. Durch Kostimulation werden die Zellen zur Vermehrung angeregt.
LImmSim	Lübecker Immunsystem Simulator. Modifizierte Version des CS-Modells, die im Rahmen dieser Diplomarbeit entwickelt wurde.
Lymphozyten	Zellen des Immunsystems.
MHC	Major Histocompatibility Complex. Ein Proteinkomplex, den alle körpereigenen Zellen auf ihrer Oberfläche tragen. Der MHC ist bei Wirbeltieren in jedem Individuum unterschiedlich und kann als mikrobiologische Definition des "Selbst" (Personalausweis des Körpers) angesehen werden. Die Präsentation von Antigenen auf der Zelloberfläche von antigenpräsentierenden Zellen erfolgt stets zusammen mit einem MHC.
Paratop	Form eines B-Zell-Rezeptors oder Antikörpers, die genau zum Epitop eines Antigens passt (Schlüssel-Schloss-Prinzip).

Phagozytose	Aufnahme und Verdauung einer Substanz oder einer fremden Zelle durch eine Fresszelle.
Plasmazellen	Zellen, die Antikörper sezernieren. Plasmazellen entwickeln sich aus B-Zellen. Die Form des sezernierten Antikörpers entspricht der des B-Zell-Rezeptors.
Proliferation	Zellteilung.
Spezifität	Besonderes Merkmal bzw. Form eines Antigens, Antikörpers, B-Zell- oder T-Zell-Rezeptors.
T-Zelle	Gemeinsam mit den B-Zellen die wichtigste Zelllinie des spezifischen Immunsystems. T-Zellen werden vor allem im Thymus (Bries) gebildet. Es gibt verschiedene Arten von T-Zellen mit unterschiedlichen Aufgaben bei der Immunantwort. Von besonderer Bedeutung sind dabei die T-Helferzellen.
Th-Zelle	T-Helferzelle. Dient der Steuerung der Immunantwort. T-Helferzellen werden im Thymus trainiert, <i>nicht</i> gegen körpereigene Proteine zu reagieren (negative Selektion). Bevor B-Zellen zu Antikörper produzierenden Plasmazellen reifen dürfen, benötigen sie ein Stimulationssignal von einer T-Helferzelle, der sie das von ihnen erkannte Antigen präsentieren. Dadurch wird sichergestellt, dass keine Antikörper gegen körpereigene Proteine gebildet werden.

B. Inhalt der CD

Die erstellte Implementierung von LImmSim ist auf der beigefügten CD im C++ Quellcode und als ausführbare Datei für x86-Linux verfügbar. Weiterhin wurde eine rudimentäre GUI zur grafischen Steuerung von LImmSim erstellt (siehe Abbildung B.1). LImmSim wurde unter Linux und Solaris getestet, sollte aber auch auf anderen Unix-Systemen lauffähig sein. Um LImmSim zu nutzen, kopieren Sie bitte das Verzeichnis `limmsim-0.1` von der CD in ein beliebiges Verzeichnis auf Ihr System. Stellen Sie mittels

```
chmod -R u+w limmsim-0.1
```

sicher, dass trotz des Kopierens von CD in das Verzeichnis geschrieben werden kann.

B.1. Kompilieren von LImmSim

Zum Kompilieren von LImmSim auf Linux-Systemen steht ein Makefile zur Verfügung. Der Übersetzungsvorgang wird gestartet durch den Aufruf

```
make distclean all
```

im LImmSim-Wurzelverzeichnis. Dadurch wird die ausführbare Datei `limmsim` generiert. In der Datei `Makefile.solaris` sind die Optionen zur Kompilation von LImmSim auf der Mayflower unter Solaris enthalten. Dies geschieht durch

```
make -f Makefile.solaris distclean all
```

B.2. Aufruf von LImmSim

LImmSim kann entweder über die Kommandozeile oder durch die LImmSim-GUI bedient werden. Der Aufruf

```
./limmsim
```

startet eine Simulation mit den Standardparametern (siehe Tabelle 5.1). Die Daten des Simulationsverlaufs werden im Verzeichnis `output` abgelegt:

- `output/data`: Die Dateien `t.dat`, `b.dat`, `ma.dat`, `ic.dat`, `ag.dat` und `ab.dat` enthalten Statistiken über die Gesamtzahl der verschiedenen Partikeltypen im Simulationsverlauf.

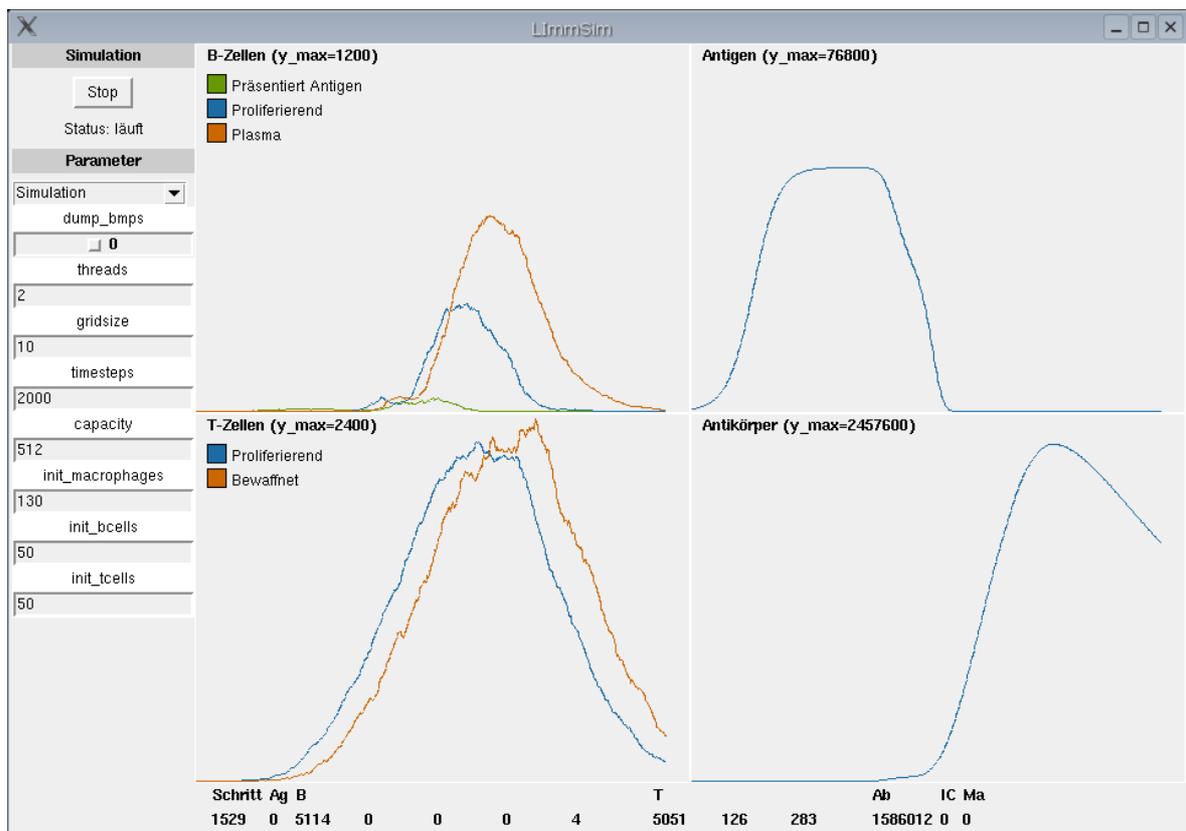


Abbildung B.1.: GUI zur Bedienung von LImmSim

- **output/dumps:** Wenn der Parameter `dump_bmps` den Wert 1 hat (was standardmäßig der Fall ist), werden hier nach jedem Simulationsschritt BMP-Grafiken abgelegt, die die Konzentrationen von Antigen und Antikörper (in `output/dumps/ag`) sowie B- und T-Zellen (in `output/dumps/b`) darstellen.

Zur Visualisierung der globalen Statistiken wurde ein Gnuplot-Skript erstellt. Dies kann aufgerufen werden durch:

```
cd output
./plot.sh
```

Hierdurch wird eine Reihe von Plots erzeugt, die mit Betätigung der Eingabetaste im Befehlsfenster durchgeblättert werden können.

B.2.1. Einstellung der Parameter

Um die Simulationsparameter zu verändern, rufen Sie `LImmSim` wie folgt auf:

```
./limmsim -i
```

Es können nun die zu verändernden Parameter eingegeben werden. Die Syntax dafür lautet:

```
[parametername] [wert]
```

Als Parameternamen können dabei alle in Tabelle 5.1 angegebenen Werte verwendet werden. Die Eingabe kann durch ein End-Of-File-Symbol (CTRL-D) oder eine nur aus einem Punkt bestehende Zeile beendet werden. Die Eingabe

```
threads 16
gridsize 120
.
```

führt beispielsweise dazu, dass `LImmSim` mit den Standardwerten, aber einer Gittergröße von 120 mal 120 gestartet wird und zur Berechnung 16 Threads verwendet werden.

Natürlich müssen die Parameter nicht per Hand eingegeben werden, sondern können auch per Umleitung der Standardeingabe aus einer Datei gelesen werden:

```
./limmsim < settings/default-settings.txt
```

startet `LImmSim` mit den Parametern aus der Datei `settings/default-settings.txt`. Diese Datei kann als Grundlage zur Definition eigener Parameterdateien benutzt werden.

B.3. Die LImmSim-GUI

Zur Vereinfachung der Bedienung wurde für LImmSim eine rudimentäre GUI erstellt. Zur Benutzung der GUI muss tcl/tk verfügbar sein, spezielle Bibliotheken für tcl oder tk sind nicht erforderlich. Die GUI wird mit dem Befehl

```
./limmsim-gui.tcl
```

oder, falls dies fehlschlagen sollte, mit

```
wish -f ./limmsim-gui.tcl.
```

aufgerufen. Dabei muss sich `limmsim` im gleichen Verzeichnis befinden. Der Aufbau der GUI wird in Abbildung B.1 dargestellt. Auf der linken Seite unter “Parameter” befinden sich Eingabefelder, mit denen die Simulationsparameter eingestellt werden können. Zur besseren Übersicht sind die Parameter in verschiedene Gruppen eingeteilt, die mit einem Ausklappfeld umgeschaltet werden können. Die Simulation kann durch den Knopf “Start” gestartet werden. Im Verlauf der Simulation werden auf den Ausgabefeldern Plots der Partikelkonzentrationen angezeigt. Im unteren Fensterbereich werden die Partikelkonzentrationen unformatiert ausgegeben. Diese Anzeige entspricht dem Dateiinhalt der entsprechenden Statistik in `output/data`.

B.4. Credits

LImmSim benutzt als schnellen, multithreading-fähigen Zufallsgenerator den *Mersenne Twister* von Matsumoto und Nishimura [MN98] in einer Implementierung von Richard J. Wagner, der seine Arbeit unter der BSD-Lizenz freigegeben hat. Die Implementierung wurde bezogen von:

```
http://www-personal.engin.umich.edu/~wagnerr/MersenneTwister.html
```

Zur Ausgabe der BMP-Grafiken wurde die Bibliothek `BMP_IO` von John Burkardt verwendet, die ohne Lizenz zur Verfügung steht. Die Bibliothek wurde bezogen von:

```
http://www.scs.fsu.edu/~burkardt/cpp\_src/bmp\_io/bmp\_io.html
```

Das Ausklappfeld für die Optionsgruppen in der GUI wurde durch das tk-Widget *combobox* von Bryan Douglas Oakley realisiert, das zur kostenlosen Verwendung freigegeben ist.

```
http://www1.clearlight.com/~oakley/tcl/combobox/index.html
```

Ich bin den Autoren dieser Bibliotheken zutiefst dafür dankbar, dass sie ihre Werke zur freien Verwendung zur Verfügung gestellt und mir dadurch einige Tage Arbeit erspart haben.

Literaturverzeichnis

- [BC01] BERNASCHI, M. ; CASTIGLIONE, F.: Design and Implementation of an Immune System Simulator. In: *Computers in Biology and Medicine* 31 (2001), Nr. 5, S. 303–331
- [Beh98] BEHE, Michal J.: *Darwin's Black Box: The Biochemical Challenge to Evolution*. Free Press, 1998
- [BP95] BOER, R.J. de ; PERELSON, A.S.: Towards a General Function Describing T Cell Proliferation. In: *J. Theor. Biol.* 175 (1995), S. 567–576
- [BSP92] BOER, R.J. de ; SEGEL, L.A. ; PERELSON, A.S.: Pattern-Formation in One-Dimensional and Two-Dimensional Shape-Space Models of the Immune System. In: *J. Theor. Biol.* 155 (1992), S. 295–333
- [CBS97] CASTIGLIONE, F. ; BERNASCHI, M. ; SUCCI, S.: Simulating the Immune Response on a Distributed Parallel Computer. In: *Int. J. Mod. Phys. C* 8 (1997), Nr. 3, S. 527–545
- [CCFS96] CARNEIRO, Jorge ; COUTINHO, Antonio ; FARO, Jose ; STEWART, John: A Model of the Immune Network with B-T Cell Co-operation. I-Prototypical Structures and Dynamics. In: *J. Theor. Biol.* 182 (1996), S. 513–529
- [Das98] DASGUPTA, Dipankar (Hrsg.): *Artificial Immune Systems and Their Applications*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 1998. – ISBN 3540643907
- [FHS97] FORREST, Stephanie ; HOFMEYR, Steven A. ; SOMAYAJI, Anil: Computer Immunology. In: *Communications of the ACM* 40 (1997), Nr. 10, S. 88–96
- [FPAC94] FORREST, Stephanie ; PERELSON, Alan S. ; ALLEN, Lawrence ; CHERUKURI, Ramesh: Self-Nonself Discrimination in a Computer. In: *SP '94: Proceedings of the 1994 IEEE Symposium on Security and Privacy*. Washington, DC, USA : IEEE Computer Society, 1994, S. 202
- [Han01] HANEGRAAF, Wouter: *Simulating the Immune System*, University of Amsterdam, Diplomarbeit, September 2001

-
- [HMR96] HEAD, Martha ; MERYHEW, Nancy ; RUNQUIST, Olaf: Mechanism and Computer Simulation of Immune Complex Formation, Opsonization, and Clearance. In: *J. Lab. Clin. Med.* 128 (1996), Nr. 1, S. 61–74
- [Hof99] HOFMEYER, Steven A.: *An Immunological Model of Distributed Detection and its Application to Computer Security*, University of New Mexico, Diss., 1999
- [Hof01] *Kapitel An Interpretative Introduction to the Immune System*. In: HOFMEYER, Steven: *Design Principles for the Immune System and Other Distributed Autonomous Systems*. Oxford University Press, 2001
- [Jer74] JERNE, Niels K.: Towards a Network Theory of the Immune System. In: *Ann. Immunol. (Inst. Pasteur)* 125C (1974), S. 373–389
- [JTW02] JANEWAY, Charles A. ; TRAVERS, Paul ; WALPORT, Mark: *Immunologie*. Spektrum Akademischer Verlag, 2002
- [Kal02] KALLENBERG, Olav: *Foundations of Modern Probability*. Springer, 2002. – ISBN 3540643907
- [Lep04] LEPPÄNEN, Teemu: *Computational Studies of Pattern Formation in Turing Systems*, TU Helsinki, Diss., 2004
- [MK96] MALEVANETS, Anatoly ; KAPRAL, Raymond: Links, Knots, and Knotted Labyrinths in Bistable Systems. In: *Physical Review Letters* 77 (1996), Nr. 4, S. 767–770
- [MK97] MALEVANETS, Anatoly ; KAPRAL, Raymond: Microscopic model for FitzHugh-Nagumo dynamics. In: *Physical Review E* 55 (1997), Nr. 5, S. 5657–5670
- [MN98] MATSUMOTO, M. ; NISHIMURA, T.: Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. In: *ACM Transactions on Modeling and Computer Simulation* 8 (1998), Nr. 1, S. 3–30
- [MS01] MEIER-SCHELLERSHEIM, Martin: *The Immune System as a Complex System: Description and Simulation of the Interactions of its Constituents*, Fachbereich Physik der Universität Hamburg, Diss., 2001
- [Mur02] MURRAY, J.: *Mathematical Biology*. Springer Verlag, 2002
- [PKSC02] PUZONE, R. ; KÖHLER, B. ; SEIDEN, P. ; CELADA, F.: IMMSIM, a Flexible Model for *in machina* Experiments on Immune System Responses. In: *Future Gener. Comput. Syst.* 18 (2002), Nr. 7, S. 961–972. – ISSN 0167–739X
- [PW92] PERELSON, A.S. ; WEISBUCH, G.: Modeling Immune Reactivity in Secondary Lymphoid Organs. In: *Bull. Math. Biol.* 54 (1992), S. 649–572

- [RBM95] ROITT, Ivan M. ; BROSTOFF, Jonathan ; MAILE, David K.: *Kurzes Lehrbuch der Immunologie*. Thieme, Stuttgart, 1995. – ISBN 3137021030
- [Rie99] RIEDEL, Erwin: *Allgemeine und Anorganische Chemie*. de Gruyter, 1999
- [SB80] SCHWICK, H. G. ; BRÄUER, H.: *Exempla Immunologica*. Frankfurt/M. : Behringwerke, 1980. – ISBN 3540643907
- [SC92] SEIDEN, P.E. ; CELADA, F.: A Model for Simulating Cognate Recognition and Response in the Immune System. In: *J. Theor. Biol.* 158 (1992), Nr. 3, S. 329–357
- [SFAP99] SMITH, D.J. ; FORREST, S. ; ACKLEY, D.H. ; PERELSON, A.S.: Variable Efficacy of Repeated Annual Influenza Vaccination. In: *Proc. Nat. Acad. Sci. USA* 96 (1999), Nr. 24, S. 14001:6
- [Sö03] SÖDERQVIST, Thomas: *Science as Autobiography: The Troubled Life of Niels Jerne*. Yale University Press, 2003
- [Ton83] TONEGAWA, S.: Somatic Generation of Antibody Diversity. In: *Nature* 302 (1983), S. 575–581
- [Tur52] TURING, A.: The Chemical Basis of Morphogenesis. In: *Philosophical Transactions of the Royal Society of London* 237 (1952), Nr. 641
- [Wol02] WOLFRAM, Stephen: *A New Kind Of Science*. Wolfram Media, 2002. – ISBN 1579550088