Improved Simulation Algorithms for Agent Based Models of the Immune Response

Johannes Textor, Björn Hansen
Institute for Theoretical Computer Science
University of Lübeck
Ratzeburger Allee 160, 23538 Lübeck, Germany
textor@tcs.uni-luebeck.de

Abstract

The immune response is a complex process which involves large numbers of highly diverse cells, making it a well suited application for agent based modelling. A major problem with current models is the lack of biophysically sound simulation methodology. In this paper, we analyze the key algorithms of an important agent based model of the immune system, point out methodological deficiencies and propose alternatives which correspond to discrete versions of established continuous models. To analyze the effects of these modifications, a detailed simulation of the immune response based on the proposed algorithms has been implemented and tested.

1 Introduction

The vertebrate immune system is one of the most complex systems to have emerged in nature. Our knowledge of how the cells and molecules which make up the immune system work and collaborate is still far from complete. Both continuous mathematical models and agent-based simulations have been applied to study the immune system [Forrest and Beauchemin, 2007]. Two key features of the immune system make it an ideal application for agent-based modelling:

- *Diversity:* To keep up with a huge, ever-changing environment of antigenic challenges, immune cells have evolved sophisticated mechanisms to diversify. Immune cells carry receptors which are generated randomly by recombination of DNA fragments. These receptors may be diversified even further by mutations of the coding sequence. The total number of possible receptors has been estimated to 10^8 [Janeway and Travers and Walport, 2005]. While such highly diverse and dynamically changing cell properties are difficult to model using differential equations, it is comparatively straightforward to implement them in an agent based model.
- *Stochasticity:* As a consequence of the high diversity of antigen receptors, a new antigen is normally only detected by a very low number of immune cells. Thus, stochastic events during the initial phase of the immune response may have dramatic consequences (see [Mosmann et al., 1986] for an example). This fact may be represented accurately using agent based models while continuous models are usually based on the assumption that stochastic perturbations may be neglected due to the large number of particles.

Many agent based models used today to study the immune system are based on the Celada-Seiden model [Seiden and Celada, 1992; Puzone et al., 2002]. This model uses bit strings to represent antigen, receptors and peptides. It is capable of reproducing high-level emergent behaviour like immune memory, affinity maturation and repertoire shift. However, the considerable advances in immunology in the last 15 years make the immunological basis of the model seem a little outdated. Much more is known today about cell migration and interactions in lymphoid tissue and so one should seek to incorporate this knowledge into agent-based modelling.

The work presented here follows two main objectives. On one hand, we seek a better understanding of the simulation algorithms used in the original version of the Celada-Seiden model (CS model). On the other hand, we wish to modify and extend the model to facilitate a more explicit representation of current immunological knowledge. To this end, we proceed as follows:

- We present the basic simulation algorithms of the CS model in terms of operations on abstract particle types, in contrast to the existing literature which is aimed at readers who are familiar with immunology. We investigate to what extent these algorithms correspond to established biophysical laws.
- We evaluate the appropriateness of these algorithms for immunological simulations in the light of new results about cell migration and interactions in the lymph node. We propose new algorithms for diffusion, proliferation and cell-cell interaction which all correspond to discrete versions of established continuous models. This link allows for hybrid simulations where the contin-

uous model is used directly for sufficiently large particle numbers, resulting in an increase of performance.

To evaluate our findings, we have implemented an agent-based simulation of the immune response based on the proposed algorithms. The implementation is discussed briefly and some results are presented in section 5. However, the main focus of this paper is on the basic properties of the simulation algorithms rather than the implementation.

2 Background

Since the CS model cannot be explained in full detail in this paper, the reader is asked to consult the literature for additional reference. This section will only give a short informal introduction.

The authors described their model as a modified cellular automata, but it is more similar to the lattice gas simulations used in computational physics and fluid dynamics. Implementations of the model normally use a two-dimensional hexagonal lattice as the simulation space. The simulation alternates between phases of interaction between agents and a phase of diffusion, where all agents may move to a random neighbour on the lattice with a fixed probability. During the interaction phase, agents may only interact with other agents on the same lattice site. The agents are also called particles in this paper and may be cells, molecules, or antigen.

It shows at some points that the simulation space of the CS model was not designed to explicitly represent a real space. For instance, there is no limit on the number of particles at a given lattice point. The diffusivity, i.e., the probability to move to a neighbouring site during the diffusion phase, is equal for all particles. However, the existence of the lattice and the random walk of the agents on it increase the stochasticity of the model. It allows for a delay between antigen invasion in the system and detection by a sensitive immune cell, which may be at a distant lattice site. Analogous effects exist in the course of infections in real organisms.

Recently, observations of immune cells in lymph nodes using two-photon microscopy revealed that the CS model is actually closer to the truth than one would have expected. Both B- and T-cells appear to search for antigen in secondary lymphoid tissue using a random walk [Miller et al., 2002; Halin et al., 2005]. It seems natural to try and incorporate these findings into the CS model. In this paper, we take a first step in this direction by looking at the basic simulation algorithms of the CS model and analyzing if they could be used to construct an explicit representation of the migration and interactions of lymphocytes in secondary lymphoid tissue.

3 Simulation algorithms

The algorithm description given here are based on [Seiden and Celada, 1992]. However, since the model was not formally defined in the cited source and some details remain unclear, the most recent implementation of the model, CImmSim version 6.3 [Baldazzi et al., 2006], was used for additional reference.

```
argument $\tau$: particle type
do for each site $S$
    do for each particle $\pi$ of type $\tau$ at $S$
        if COIN$\left(\frac{\Delta t}{(\Delta s)^2}\mathcal{D}_\tau\right)$
            target[$\pi$] $\leftarrow$ random neighbour of $S$
        else
            target[$\pi$] $\leftarrow S$
```

Figure 1: Diffusion algorithm from the CS model which has been generalized to support particle type dependent diffusivities and explicit model resolution parameters. After the algorithm has determined the targets for all particles on all sites, all particles are moved to their targets simultaneously.

3.1 Notation

All algorithms are given as pseudo code in terms of operations on abstract particle types τ. Two particles are assigned different types if there is any feature which makes them distinguishable with respect to the simulation, even if they represent the same kind of biological object. For example, two B cells which carry different antigen receptors are modelled as different types. The following model parameters are used in the algorithms:

$\Delta t > 0$	Temporal model resolution
$\Delta s > 0$	Spatial model resolution
$\mathcal{C}_{\max} > 0$	Capacity of a lattice point
$\mathcal{S}_\tau > 0$	Size of τ
$\mathcal{D}_\tau \in [0,1]$	Diffusivity of τ
$\mathcal{I}_{\tau,\tau'} \in [0,1]$	Affinity or interaction strength between τ and τ'
$\mathcal{P}_\tau \in [0,1]$	Proliferation rate of τ

The parameters $\mathcal{S}_\tau$ and $\mathcal{D}_\tau$ do not exist in the CS model, but will be used later in our own extensions. $\mathcal{S}_\tau$ representing the size of an object in spatial units and can be mapped to the real physical size of the object via $\mathcal{C}_{\max}$ and Δs (in the CS model, $\mathcal{S}_\tau = 1$ for all τ). $\mathcal{D}_\tau$ introduces particle type dependent diffusivity (in the CS model, $\mathcal{D}_\tau = \mathcal{D}_{\tau'}$ for all τ, τ'). Likewise, the CS model has no explicit notion of Δt and Δs (i.e., $\Delta t = \Delta s = 1$ in the CS model).

In our pseudo code notation, the function COIN(p) is used to denote a random experiment which succeeds with probability p. If $p \geq 1$, the experiment always succeeds. As we will see later in this section, this definition is necessary since $p > 1$ cannot always be avoided.

3.2 Diffusion

As described in Sec. 2, the simulated particles perform a simple undirected random walk during the diffusion phase. The corresponding pseudo code is given in Fig. 1. The mean-field limit (i.e., the behaviour for sufficiently large particle numbers) of this algorithm is described by the differential equation

```
argument $\tau$: particle type
do for each site $S$
    $\rho \leftarrow 0$
    do for each particle $\pi$ at $S$
        $\rho \leftarrow \rho + \left(\mathcal{S}_{\mathrm{type}(\pi)}/\mathcal{C}_{\max}\right)^m$
    do for each particle $\pi$ of type $\tau$ at $S$
        if COIN$\left(\rho \frac{\Delta t}{(\Delta s)^2} \mathcal{D}_\tau\right)$
            target[$\pi$] $\leftarrow$ random neighbour of $S$
        else
            target[$\pi$] $\leftarrow S$
```

Figure 2: Variation of the diffusion algorithm from Fig. 1 where diffusive pressure is increased proportionally with the local concentration of cells.

$$\frac{\partial C_\tau}{\partial t} = \mathcal{D}_\tau \nabla^2 C_\tau$$

where $C_\tau : \mathbb{R}^m \times \mathbb{R} \mapsto \mathbb{R}$ is a continuous function which represents the *concentration* of particles of type τ on the lattice. This can be shown by considering the expected change of the count of particles of type τ at a given site after one time step and proving that the resulting equation is a finite difference approximation of the above differential equation. Details have to be omitted, but the technique is fairly standard and has been applied elsewhere [Malevanets and Kapral, 1997].

The above equation is well known in physics as the Fickian law of diffusion. Thus, the diffusion algorithm can be seen as a correct agent-based simulation of a real diffusion process.

However, the Fickian law of diffusion only holds for sufficiently low concentrations. When modelling populations at higher concentrations where collisions between the objects increase diffusion pressure, one may use the following generalized version [Murray, 2002]:

$$\frac{\partial C_\tau}{\partial t} = \mathcal{D}_\tau \nabla \cdot \left[\left(\frac{C_\tau}{\mathcal{C}_{\max}} \right)^m \nabla C_\tau \right] \quad , m > 0$$

It can be shown that this equation is the mean-field limit of the algorithm given in Fig. 2 in which the probability to move to a neighbouring site increases with particle concentration on the source site. Interestingly, a similar algorithm has been implemented in CImmSim which modifies the probability of motion depending on the concentration of the *target* site. While this might be thought to have a similar effect as the algorithm in Fig. 2, it can be shown that the CImmSim variant is equivalent to Fickian diffusion in the mean field limit. This shows that the behaviour of an algorithm in the mean field limit is not always immediately obvious.

So far, it is unclear which of the algorithms is more appropriate for immune system simulations. How freely immune cells move in lymphoid tissue is still a matter of debate. No difference to a normal random walk was found in the two-photon experiments, but more recent findings suggest that the seemingly random motion may actually be a consequence of the densely-packed environment [Beltman et al., 2006].

```
arguments $\tau, \tau'$: particle types
do for each site $S$
    $L \leftarrow$ particles of type $\tau$ at $S$
    $L' \leftarrow$ particles of type $\tau'$ at $S$
    do for each $\pi \in L$
        do for each $\pi' \in L'$
            if COIN($\mathcal{I}_{\tau,\tau'}$)
                interact($\pi, \pi'$)
                remove $\pi'$ from $L'$
                continue using next $\pi$
```

Figure 3: Algorithm for particle-particle interactions in the CS model. Note that two particles of the same type are completely equal, and thus the order of particles in L and L' does not influence the outcome of the algorithm.

3.3 Interactions between particles

The actual immunological knowledge is implemented in the interactions which can take place between the particles and the consequences of these interactions. In the pseudo codes in Fig. 3 and 4, this knowledge is abstracted in the procedure call interact(π, π'). For example, an interaction procedure could move one of the particles to a different state (by changing its type), or remove one of the particles (think phagocytosis). In one interaction phase, all possible interactions between the particles are considered. Since the outcome of each interaction between a pair of particle types may affect the subsequently treated interactions, the order in which the interactions are considered is randomly determined at each site.

While this algorithm is easy to implement, it is surprisingly difficult to determine the mean field limit, because each outcome of the inner loop influences all following iterations of the outer loop. So far, we were unable to derive a closed formula for the mean field limit of this algorithm. In section 4, we will improve our understanding of this algorithm by studying it using a model of a simple chemical system.

In order to model the cell-cell and cell-molecule interactions in the lymph node more explicitly, the algorithm should be aware of the particle sizes and the spatial resolution. Again, several continuous models are available for interactions between particles, the most common being the kinetic rate laws from chemistry which are also often used in continuous models of the immune system (e.g. [Carneiro et al., 2005]). An algorithm which follows the law of mass action is shown in Fig. 4. It is easy to see that the rate of the changes induced by interact(π, π') in the mean field limit is equal to $\mathcal{I}_{\tau,\tau'} \cdot C_\tau \cdot C'_\tau$ and thus proportional to the rates of concentration. Thus, the algorithm obeys the law of mass action.

3.4 Cell proliferation

Only antigen and activated immune cells proliferate in the CS model. A continuous turnover of cells is im-

```
1  arguments $\tau, \tau'$: particle types
2  do for each site $S$
3      $L \leftarrow$ particles of type $\tau$ at $S$
4      $L' \leftarrow$ particles of type $\tau'$ at $S$
5      $I_{\max} \leftarrow \mathrm{size}(L)$
6      $\rho \leftarrow \mathrm{size}(L') \cdot (\mathcal{S}_{\tau'}/\mathcal{C}_{\max})$
7      do $I_{\max}$ times
8          if COIN($\rho\,\Delta t\,\mathcal{I}_{\tau,\tau'}$)
9              interact(head($L$),head($L'$))
10             $L$=tail($L$)
11             $L'$=tail($L'$)
```

Figure 4: Proposed interaction algorithm which obeys the law of mass action. For simplicity, it is assumed that size(L) $\leq$ size(L') holds at line 5 such that the maximum number of interactions does not exceed the possible number of interactions.

```
argument $\tau$: particle type
do for each site $S$
    $\rho \leftarrow 0$
    do for each particle $\pi$ at $S$
        $\rho \leftarrow \rho + \mathcal{S}_{\mathrm{type}(\pi)}/\mathcal{C}_{\max}$
    $L \leftarrow$ cells of type $\tau$ at $S$
    if $\Delta t \cdot \mathcal{P}_\tau \cdot (1-\rho) > 0$
        do for each $\pi \in L$
            if COIN($\mathcal{P}_\tau \cdot (1-\rho)$)
                clone($\pi$)
    else
        do for each $\pi \in L$
            if COIN($-\mathcal{P}_\tau \cdot (1-\rho)$)
                kill($\pi$)
```

Figure 5: Proposed proliferation algorithm. In contrast to previous implementations of the CS model, the capacity of each site is limited.

plemented by inserting new cells into the model and deleting existing cells at a fixed rate, determined by a half-life parameter for each cell type. In the original version of the CS model, proliferation of activated cells was completely unlimited. In CImmSim, the growth rate of cells is damped with a Gaussian kernel (algorithm not shown). Still, it is not possible for cells to die due to overpopulation.

Several models for resource-limited growth of cells exist [Murray, 2002]. The algorithm shown in Fig. 5 implements a discrete version of the *logistic equation*

$$\frac{\partial C_\tau}{\partial t} = \mathcal{P}_\tau C_\tau \frac{\mathcal{C}_{\max} - C_\tau}{\mathcal{C}_{\max}}$$

which imposes a stricter limit on the number of cells. While temporal overpopulation of a site is still possible, the algorithm will remove the exceeding cells after some time such that the maximum site capacity $\mathcal{C}_{\max}$ is no longer exceeded. This is important for the algorithms shown in Fig. 2 and 4 to provide meaningful results.

4 Simulation of a Bistable Chemical System

Given the high complexity of the CS model, special care must be taken to avoid simulation artifacts. These artifacts might arise, for example, as a consequence of setting Δs too small or Δt too high. As an analogy, think of stability criteria for finite difference methods to solve partial differential equations. However, a rigorous mathematical stability analysis of the entire model does not seem realistic. Instead, we applied our simulation algorithms to a comparatively simple bistable chemical system to estimate valid parameter ranges. The system consists of the four hypothetical species A, A^*, B, B^* and is defined by the following chemical equations:

$$2A + A^* \xrightarrow{\mathcal{I}_{A,A^*}} 3A$$

$$2A^* + A \xrightarrow{\mathcal{I}_{A^*,A}} 3A^*$$

$$\begin{tikzcd}
A^* + B \arrow[r, "\mathcal{I}_{A^*,B}"] & A + B \arrow[d, "\mathcal{I}_{A,B}"] \\
A^* + B^* \arrow[u, "\mathcal{I}_{A^*,B}"] & A + B^* \arrow[l, "\mathcal{I}_{A,B^*}"']
\end{tikzcd}$$

This system was studied extensively using a particle simulation [Malevanets and Kapral, 1997] which was designed to comply with the Fickian law of diffusion and the law of mass action. The particle simulation is capable of producing stable Turing patterns if the ratio of diffusivity vs. reaction rate is sufficiently high, i.e., if diffusion is “fast enough” to maintain sufficient independence between neighbouring lattice sites. Since the particle simulation used by Malevanets and Kapral is structurally similar to the CS model, it is used here as a testbed for the algorithms presented in the last section.

The simulations have been carried out on a 300x300 square lattice. As described above, a simulation alternates between interaction and diffusion phases. During the interaction phase, the order at which the 6 possible reactions are treated is chosen randomly at each lattice point. When using the interaction algorithm from the CS model, each particle is only allowed to interact once per interaction phase. This limitation is not applied to our modified interaction algorithm.

It is not immediately obvious how the algorithm from Fig. 4 can be extended to interactions of three particles, especially if two of the interacting species are identical as in the equation $2A + A^* \rightarrow 3A$. Fig. 7 shows an implementation which still obeys the law of mass action and ensures that no negative particle numbers arise. However, note that it is possible that A^* is converted to A even if there is only one particle of type A at the site. The extension of the interaction algorithm of the CS model is not shown. It is simply done by considering all possible triples of particles instead of all possible pairs.

Some simulation results are shown in Fig. 6. Both

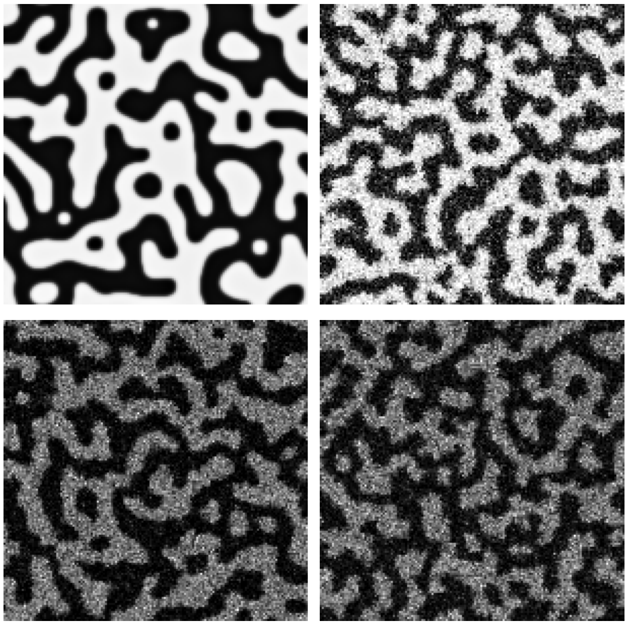

Figure 6: Simulations of a bistable chemical system. Top left: Solution of the system equations with perturbed initial conditions. Top right: Particle simulation as described by Malevanets and Kapral. Bottom left: Particle simulation using algorithms from the CS model (Fig. 1 and 3). Bottom right: Particle simulation using the diffusion algorithm from the CS model (Fig. 1) and our proposed interaction algorithm (Fig. 4). The colors are linearly mapped to the interval of concentrations of A on the lattice, white representing the maximum concentration. The bottom images appear darker due to higher variance.

```
do for each site $S$
    $L \leftarrow$ particles of type $A$ at $S$
    $L' \leftarrow$ particles of type $A^*$ at $S$
    $I_{\max} \leftarrow \mathrm{size}(L_2)$
    $\rho \leftarrow (\mathrm{size}(L) \cdot (\mathcal{S}_A/\mathcal{C}_{\max}))^2$
    do $I_{\max}$ times
        if COIN($\rho\,\Delta t\,\mathcal{I}_{A,A^*}$)
            move particle from $L'$ to $L$
```

Figure 7: Extension of the algorithm from Fig. 4 to interactions between three particles. Here, the equation $2A + A^* \rightarrow 3A$ is shown.

	CS model	Modified version
Δt	1	1
Δs	1	1
$\mathcal{C}_{\max}$	32	32
$\mathcal{S}_\tau (\forall \tau)$	1	1
$\mathcal{D}_A = \mathcal{D}_{A^*}$	0.032	0.032
$\mathcal{D}_B = \mathcal{D}_{B^*}$	0.32	0.32
$\mathcal{I}_{A,A^*} = \mathcal{I}_{A^*,A}$	0.0007	0.182
$\mathcal{I}_{A^*,B} = \mathcal{I}_{A,B^*}$	0.0006125	0.0104
$\mathcal{I}_{A,B} = \mathcal{I}_{A^*,B^*}$	0.001225	0.0208

Table 1: Parameters used in the simulations of the bistable chemical system shown in Fig. 6.

the original algorithms from the CS models and our modified version are able to form stable Turing patterns in certain parameter regions. The parameters from [Malevanets and Kapral, 1997] can be used directly with our proposed algorithms (bottom right image in Fig. 6). The corresponding values for the CS model had to be determined by isolating the diffusion and interaction parts and tuning them to the particle simulation described by Malevanets and Kapral. Note that these parameter values are close to maximal since $\mathcal{D}_B > 0.5$ would violate the numerical stability condition $\mathcal{D}_B \Delta t/(\Delta s)^2 \leq 0.5$.

Of course, the fact that the algorithms are able to generate stable Turing patterns in the studied parameter regions does not guarantee that no artifacts arise when the algorithms are applied to simulate the immune response. However, we are now able to characterize the difference between the two interaction algorithms a bit more precisely: for diffusion to play any role at all, the interaction algorithm of the CS model must be used with very low parameter values. Typical parameter values from CImmSim, for example, range from 0.01 (unspecific phagocytosis of antigen by macrophages) to 1 (specific phagocytosis of antigen by maximum-affinity B cells). In this parameter range, the number of interactions carried out is close to maximal with high probability. The diffusion algorithm no longer maintains a locally smooth particle distribution under these conditions.

5 Simulation of the Immune Response

We have created an implementation of the CS model using the algorithms proposed in this paper which includes B cells, T cells, macrophages, antigen and antibody and the interactions between them as described in [Seiden and Celada, 1992], including somatic hypermutation and a simple mechanism for immune memory which is also found in CImmSim.

The algorithms used were those given in Fig. 1 for diffusion, Fig. 4 for interactions and Fig. 5 for cell proliferation. The parameters $\mathcal{S}_\tau$ are not used yet. To increase the performance of the simulation, the user may define thresholds for each simulated process. If the number of particles participating in the process exceeds the threshold, the simulation switches to the continuous model. The user may also decide to not use this approximation by setting the threshold higher than $\mathcal{C}_{\max}$. Extensive simulations have been carried out to explore the effect of the hybrid simulation on the simulation outcome [Textor, 2006], but can not be discussed in detail in this paper. Briefly, the mean values of the measured quantities (duration of infection clearance, success rate, maximum cell titers among others) were not affected by the hybrid simulation mode. Surprisingly, the variance of these quantities increased when enabling the hybrid mode.

Results of a typical simulation run are shown in Fig. 8. Antigen is injected at a random site of the lattice at time steps $t = 100$ and $t = 2500$. It starts proliferating and spreads across the lattice by diffusion. Some B cells and macrophages ingest the foreign anti-

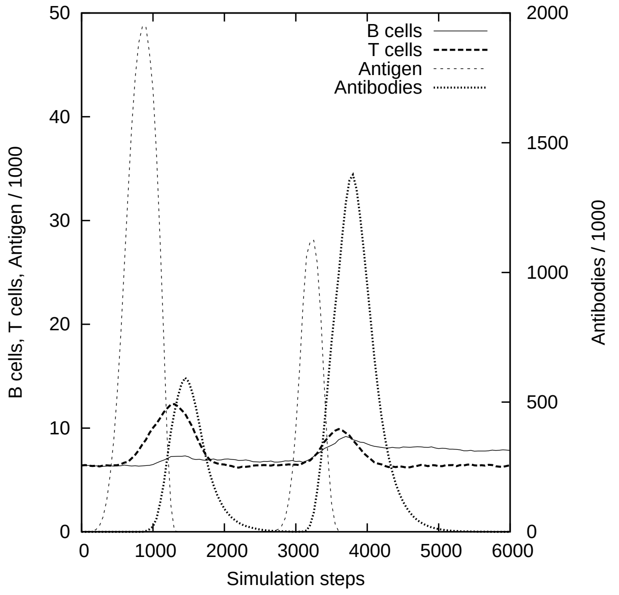

Figure 8: Results of a simulation of the immune response on a 40 × 40 hexagonal lattice using $\mathcal{C}_{\max} = 32$. Proliferating antigen is injected at $t = 100$ and $t = 2500$. Due to immune memory, the second infection is cleared faster than the first one and more antibody is produced.

gen and present peptide fragments (represented as bit strings) to the T cells. Once a T cell has recognized a presented antigen on a B cell surface, both cells start proliferating and produce clones with the same receptor. Eventually, B cells start to secrete antibody and the antigen is cleared. Some of the B cells remain in the system as B memory cells. Thus, upon reinfection, the antigen is cleared faster and more antibody is produced.

The implementation is available under the GNU General Public License and can be downloaded from `http://www.tcs.uni-luebeck.de/forschung/software/limmsim/` or obtained by e-mail from the authors for reference.

6 Conclusions and Perspectives

In this paper, the basic simulation algorithms used in the Celada-Seiden model, an agent-based simulation of the immune response, have been analyzed. It has been shown that using relatively simple modifications to the key simulation algorithms of the CS model, a link to well-established continuous models is established. It has been shown how reasonable parameter ranges for these algorithms can be obtained using a simulation of a bistable chemical system. A prototype of an immune system simulator based on the new algorithms has been implemented and tested.

Further research should extend the model to three dimensions and incorporate current immunological knowledge, heading towards an explicit simulation of a real secondary lymphoid organ, for example a lymph node. This challenging task raises a number of new questions. For example, it is not straightforward to model different tissue regions and the interfaces between them in a uniform simulation space.

The link to continuous models also enables hybrid simulations which may substantially increase the performance of large-scale simulations. While preliminary results indicate that these simulations still produce valid results, the effects of this hybridization need to be clarified further.

References

[Baldazzi et al., 2006] V. Baldazzi et al. An Enhanced Agent Based Model of the Immune System Response. *Cellular Immunology*, 244:77–79, 2006.

[Beltman et al., 2006] J. B. Beltman et al. Spatial modelling of brief and long interactions between T cells and dendritic cells. *Immunology and Cell Biology*, 85:306–314, 2007.

[Carneiro et al., 2005] J. Carneiro et al. Immunological Self-Tolerance: Lessons from Mathematical Modeling. *Journal of Computational and Applied Mathematics*, 184:77–100, 2005.

[Forrest and Beauchemin, 2007] S. Forrest and C. Beauchemin. Computer Immunology. *Immunology Reviews*, 16(1):176–197, 2007.

[Halin et al., 2005] C. Halin et al. In Vivo Imaging of Lymphocyte Trafficking. *Annual Review of Cell and Developmental Biology*, 21:581–603, 2005.

[Janeway and Travers and Walport, 2005] C. A. Janeway and P. Travers and M. Walport. *Immunobiology (6th edition).* Garland Science, 2005.

[Malevanets and Kapral, 1997] A. Malevanets and R. Kapral. Links, Knots, and Knotted Labyrinths in Bistable Systems. *Physical Review Letters*, 77(4):767–770, 1996.

[Miller et al., 2002] M. J. Miller et al. Two-Photon Imaging of Lymphocyte Motility and Antigen Response in Intact Lymph Node. *Science*, 296:1869–1873, 2002.

[Mosmann et al., 1986] T. Mosmann et. al. Two types of murine helper T cell clone. I. Definition according to profiles of lymphokine activities and secreted proteins. *Journal of Immunology*, 136:2348–57, 1986.

[Murray, 2002] J. D. Murray *Mathematical Biology (3rd edition).* Springer, 2002.

[Puzone et al., 2002] R. Puzone, B. Kohler, P. E. Seiden, F. Celada. IMMSIM, a flexible model for in machina experiments on immune system responses. *Future Generation Comp. Syst.*, 18(7):961–972, 2002.

[Seiden and Celada, 1992] P. E. Seiden and F. Celada. A Model for Simulating Cognate Recognition and Response in the Immune System. *Journal of Theoretical Biology*, 158:329–357, 1992.

[Textor, 2006] J. Textor. Ein hybrides Automatenmodell zur Simulation des Immunsystems. Masters Thesis, University of Lübeck, Germany, 2006.