

Vrije Universiteit Amsterdam
Faculty of Sciences

Master's Thesis
Modeling Tuberculosis
in Lung and Central Nervous System

by
Mohammed El-Kebir

Supervisors
prof. dr. A.M. van Furth
prof. dr. J. Heringa
prof. dr. D. Kirschner
dr. M. van der Kuip

Amsterdam, August 8, 2010

This work was financially supported by KNCV Tuberculosis Foundation, by Stichting Researchfonds Kindergeneeskunde VU Medisch Centrum and by Faculteit Exacte Wetenschappen VU University Amsterdam.

Abstract

Tuberculosis (TB) is caused by the bacterium *Mycobacterium tuberculosis* (Mtb). Most cases of TB are pulmonary, i.e. the main infection site is in the lung. In this work, we consider pulmonary TB as well as tuberculous meningitis (TBM). The latter is caused by infection of the meninges in the central nervous system (CNS) with Mtb. TBM is the most severe extra-pulmonary manifestation of TB; when left untreated it results in death of the patient. Even if the patient is treated, severe sequelae may result such as spasticity, other handicaps and serious mental problems.

We start this thesis by presenting the relevant biological background information, after which we proceed to describe and extend an existing agent-based model of the immune response in pulmonary TB. In addition, we introduce a new agent-based model describing the immune response to Mtb in the CNS. We have implemented the models by developing novel software that employs various visualization techniques. After subjecting both models to an experimental evaluation, we conclude by discussing possible future work.

Acknowledgments

It has been a long ride of about ten months, eight of which I spent in Ann Arbor, Michigan and the remaining two months in the Netherlands writing my thesis. Without the support of many, I would not have been able to finish this work. I sincerely hope to be able to express my gratitude with the following words.

First of all, I would like to thank my supervisors Marceline van Furth and Martijn van der Kuip for giving me the opportunity to do my thesis work on this interesting and very relevant topic, and for guiding and supporting me.

I am also indebted to Denise Kirschner who welcomed me to her lab at the University of Michigan, and made me feel at home immediately. It is thanks to her guidance that a great part of this work came about.

I had a wonderful time in Ann Arbor thanks to my colleagues and friends. I would like to thank Simeone Marino and his wife Kathya Arce for inviting me to their place at various occasions. I am also grateful to Dave Mazur for showing me around in Ann Arbor and patiently explaining me the rules of American Football.

I wish to thank Jaap Heringa and Gunnar Klau for allowing me to finish my Master's thesis during the first months of my PhD position.

Last but certainly not least, I would like to thank my colleagues Mohammad Fallahi-Sichani, Nick Cilfone, Jennifer Linderman, Paul Wolberg and Joe Waliga for the pleasant and productive time we had together.

- Mohammed El-Kebir

Contents

I	Introduction	1
1	Introduction	3
1.1	Immunobiology of TB	3
1.1.1	A primer on immunology	4
1.1.2	Pathogenesis of pulmonary TB	7
1.1.3	Pathogenesis of TBM	8
1.2	Previous and related work – theoretical and experimental	10
1.2.1	Modeling of TB	10
1.2.2	TBM	10
1.3	Our contribution	12
II	Methods	13
2	Modeling	15
2.1	Lung model	15
2.1.1	Model overview	16
2.1.2	Immune cell specific rules	18
2.1.3	Cytokine/chemokine specific rules	26
2.1.4	Mtb specific rules	33
2.1.5	Extensions and model parameters	35
2.2	CNS model	37
2.2.1	Model overview	37
2.2.2	Immune cell specific rules	40
2.2.3	Cytokine/chemokine specific rules	46
2.2.4	Mtb specific rules	50
2.2.5	Model parameters	50
2.3	Uncertainty analysis methodology	52

3	Visualization	57
3.1	Datasets	57
3.2	Agent visualization	58
3.3	Scalar visualization	59
3.3.1	Color maps	59
3.3.2	Isolines	62
3.3.3	Height plot	64
3.4	Vector visualization	66
3.5	Camera	69
3.6	Combining visualization techniques	70
4	Implementation	73
4.1	Top-level overview	73
4.2	Implementation details	74
4.2.1	Model	74
4.2.2	Visualization	76
4.2.3	User interface	79
4.2.4	LHS	81
III	Results	83
5	Experimental Evaluation	85
5.1	Baseline identification	85
5.1.1	Lung model	85
5.1.2	CNS model	91
5.2	In silico experiments	97
5.2.1	Lung model	97
5.2.2	CNS model	99
6	Conclusion and Discussion	105
6.1	Discussion and future work	106
	Bibliography	109
A	Diffusion Equation	113
A.1	FTCS	113
A.2	BTCS	113

B	Build Instructions	117
B.1	Windows	117
B.1.1	MinGW	117
B.1.2	Visual Studio	118
B.2	Linux	118
B.3	Mac OS X	118
C	Parameter Sets	119
C.1	Mapping tables	119
C.2	Lung – containment	121
C.3	Lung – clearance	122
C.4	Lung – dissemination	123
C.5	CNS – containment	124
C.6	Lung – TNF- α knock-out	126
C.7	Lung – IFN- γ knock-out	127
C.8	CNS – IFN- γ knock-out	128

List of Algorithms

1	LUNGABM(\mathcal{G}, I)	17
2	INITIALIZE(\mathcal{G}, I)	18
3	UPDATESTATES(\mathcal{G}, t)	19
4	CONTRIBUTETOCaseation(c)	19
5	UPDATETREGSTATE($T_{\text{reg}}, \mathcal{G}, t$)	20
6	UPDATETCYTSTATE(T_c, \mathcal{G}, t)	20
7	UPDATETGAMSTATE(T_γ, \mathcal{G}, t)	21
8	UPDATEMACSTATE(M, \mathcal{G}, t)	22
9	HANDLEMACRESTING(M, \mathcal{G}, t)	23
10	HANDLEMACINFECTED(M, \mathcal{G}, t)	24
11	HANDLEMACCHRONICALLYINFECTED(M, \mathcal{G}, t)	25
12	HANDLEMACACTIVE(M, \mathcal{G}, t)	26
13	MACSECRETE(\mathcal{G})	27
14	DIFFUSE(\mathcal{G})	29
15	DEGRADE(\mathcal{G})	30
16	RECRUIT(\mathcal{G}, t)	30
17	RECRUITMAC(c, \mathcal{G}, t)	31
18	RECRUITTCELL(c, \mathcal{G}, t)	32
19	MOVEMACROPHAGES(\mathcal{G}, t)	32
20	MOVETCELLS(\mathcal{G}, t)	33
21	MOVEIMMUNECELL($ic, \mathcal{G}, b_{\text{CCL2}}, b_{\text{CCL5}}, b_{\text{CXCL9}}, \alpha$)	34
22	GROWINTMTB(M)	34
23	GROWEXTMTB(\mathcal{G})	35
24	BRAINABM(\mathcal{G}, I)	40
25	INITIALIZE(\mathcal{G}, I)	41
26	UPDATESTATES(\mathcal{G}, t)	41
27	UPDATETCYTSTATE(T_c, \mathcal{G}, t)	42

28	UPDATEMGSTATE(m, \mathcal{G}, t)	43
29	HANDLEMGRAMIFIED(m, \mathcal{G}, t)	44
30	HANDLEMGRESTING(m, \mathcal{G}, t)	44
31	HANDLEMGINFECTED(m, \mathcal{G}, t)	45
32	HANDLEMGCHRONICALLYINFECTED(m, \mathcal{G}, t)	46
33	HANDLEMGACTIVE(m, \mathcal{G}, t)	46
34	MGSECRETE(\mathcal{G})	47
35	PROLIFERATEMICROGLIA(\mathcal{G}, t)	48
36	PROLIFERATEMICROGLIALCELL(m, \mathcal{G}, t)	48
37	RECRUIT(\mathcal{G}, t)	49
38	RECRUITMAC(c, \mathcal{G}, t)	49
39	MOVEMICROGLIA(\mathcal{G}, t)	50
40	MOVEIMMUNECELL($ic, \mathcal{G}, b_{CCL2}, b_{CCL5}, b_{CXCL9}, \alpha$)	51
41	LHS(P, n)	53

Part I

Introduction

Chapter 1

Introduction

With nearly one third of the global population infected, 10 million new cases every year and 2 million deaths annually [48, 49], *Mycobacterium tuberculosis* (Mtb), the causative agent of tuberculosis (TB), is one of the most successful pathogens. Even more impressive is that the pathogen has been around for quite a while: evidence of the DNA of the bacterium has been found in Egyptian mummies dated back to 3000-2400 BC [51].

Most cases of TB are pulmonary, i.e. the main infection site is in the lungs. In this thesis we consider both pulmonary TB and tuberculous meningitis (TBM). The latter is caused by infection of the meninges in the central nervous system with Mtb. TBM is the most severe extra-pulmonary manifestation of TB: it affects mainly young children and when left untreated, it results in the death of the patient. Even when treated severe sequelae result in half of the patients, such as spasticity, other handicaps and serious mental problems [40]. TBM accounts for 0.5-1% of TB cases [5].

It was not until 50 years ago that antibiotic treatment for TB was discovered. An effective vaccine, however, is yet to be found, although there is a vaccine currently used in many parts of the world, the efficacy of which is variable. With the appearance of multidrug-resistant TB (MDR-TB) and increasing incidence rates due to HIV, the need for an effective vaccine is even more apparent. A prerequisite for developing such a vaccine is a thorough understanding of the immunological response. In this thesis we aim to improve our understanding by extending mathematical models for studying the immune response to Mtb in the lung and developing a first time ever model of the brain interaction with Mtb.

In the remainder of this chapter we give a brief outline on the immunobiology of Mtb, review related and previous work and present an overview of the contribution of this work. As for the rest of this thesis, we present the developed models in Chapter 2, the visualization techniques used in Chapter 3, and the overall model implementation in Chapter 4. In Chapter 5, the models are subjected to an experimental evaluation. Conclusions and directions for future research are given in Chapter 6.

1.1 Immunobiology of TB

We start this section by giving a quick overview on the fundamental concepts of immunology. Next we discuss the chain of events (called *pathogenesis*) leading to pulmonary TB and TBM

infection and disease.

1.1.1 A primer on immunology

Immunology studies the *immune system*, which is the body's defense system against disease. Agents that cause disease — so called *pathogens* — can be viruses, bacteria, fungi or parasites. In order to fight off infection by a pathogen, an *immune response* is mounted. An immune response can be either *innate* or *adaptive*. An adaptive immune response is pathogen specific; it is developed as an adaptation to infection with that pathogen. On the other hand, an innate immune response is immediately available and not pathogen specific.

For an effective immune response to take place, the following four important tasks have to be performed by the immune system.

1. Immunological recognition.

Pathogens must be recognized as such by the immune system. Both innate and adaptive immunity are able to recognize pathogens.

2. Immune effector functions.

Once an infection by a pathogen has been recognized, the immune system must act upon it by containing the infection and if possible clearing it.

3. Immune regulation.

The immune system can be very harmful to the host's body if it is not kept in check. Therefore there must also be mechanisms for self-regulation.

4. Immunological memory.

One of the features of adaptive immunity is its ability of generating immunological memory: subsequent exposure to an already encountered pathogen will result in a stronger and faster immune response. This ability of the immune system is specifically exploited by vaccination.

In order to fulfill the previously mentioned tasks, the immune system depends on the activities of white blood cells or *leukocytes*. In Figure 1.1 the lineage of various types of leukocytes is shown. All leukocytes are derived from stem cells in the bone marrow. Leukocytes that comprise the innate immune system are NK cells (NK stands for natural killer), dendritic cells, macrophages and the granulocytes. NK cells are *cytotoxic cells*, i.e. they have the ability to kill other cells. They play a major role in killing tumors and cells infected by viruses. The granulocytes consist of neutrophils, eosinophils, mast cells and basophils. Eosinophils excel at killing parasites, whereas basophils and mast cells contribute to the recruitment of other leukocytes to the site of infection. Neutrophils are the most numerous leukocyte present in the blood. Together with macrophages and dendritic cells they are *phagocytic cells*. Phagocytic cells can ingest (extracellular) pathogens by a process called *phagocytosis*; once inside, the pathogen is killed.

Dendritic cells (and to a lesser extent macrophages) form the link between innate and adaptive immunity. They have a very distinct shape with dendrites that act like billboards on which *antigen* — that is, part of the ingested pathogen — is put on display (see Figure 1.2). The

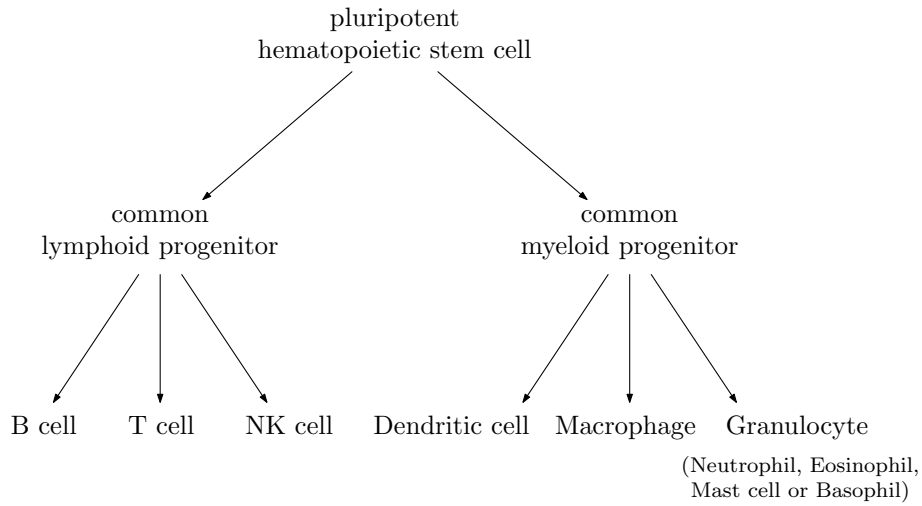


Figure 1.1: Leukocyte lineage. T cells, B cells and NK-cells comprise the lymphocytes. Only T cells and B cells are part of adaptive immunity. Cells of the other cell types participate in the innate immune response.

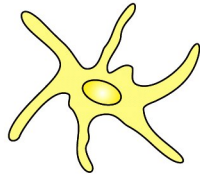

Cell	Activated function
<p>Dendritic cell</p> 	 <p>Antigen uptake in peripheral sites</p> <p>Antigen presentation</p>

Figure 1-4 part 2 of 6 Immunobiology, 7ed. (© Garland Science 2008)

Figure 1.2: Antigen presentation takes place on the dendrites of dendritic cells. (Taken from Janeway’s Immunobiology [27].)

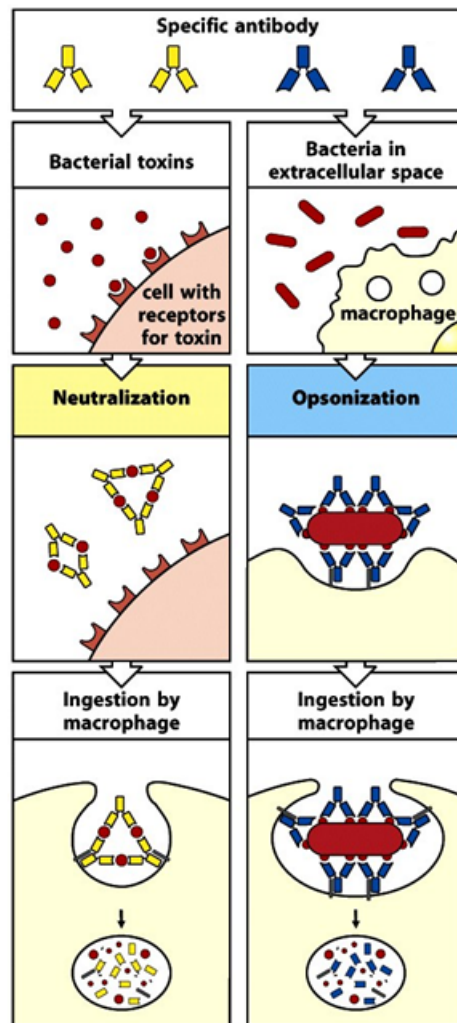


Figure 1-26 Immunobiology, 7ed. (© Garland Science 2008)

Figure 1.3: Antibodies can neutralize a toxin or can opsonize a pathogen after which it will be taken up by a phagocyte. (Slightly adapted from Janeway's Immunobiology [27].)

human body houses many lymph nodes placed at strategic sites and connected to one another through the lymphatics. It is through the lymphatics that a mature dendritic cell, i.e. a dendritic cell that is presenting antigen, migrates to the nearest lymph node. In this lymph node T cells and B cells specific to the antigen are primed.

T cells and B cells are part of the adaptive immune system. The main distinction between the two is that B cells do not directly interact with other cells or the pathogen itself. Rather they participate in *humoral immunity* by secreting antibodies that either neutralize toxins originating from the pathogen, or facilitate phagocytic clearance of the pathogen by attaching to it (called *opsonization*, see Figure 1.3). Primed B cells have the ability to turn into memory cells. Unlike B cells, T cells do participate in *cell-mediated immunity* by interacting directly with other (white blood) cells. Roughly four different types of T cells can be distinguished. T helper cells (T_H cells) assist other leukocytes in performing their immunological function. For instance, they can activate macrophages and cytotoxic T cells. Also they can induce antibody production in B cells. Cytotoxic T cells (T_C or CTLs) have the ability to kill infected cells. Regulatory

T cells (T_{reg} cells) down-regulate the adaptive immune response by shutting down other T cells via molecules they produce. Finally, memory T cells provide immunological memory.

As previously mentioned the innate immune response happens immediately. Whenever a pathogen invades the body, it is met by macrophages and other phagocytic cells. These phagocytic cells take up the pathogen and release special proteins called chemokines and cytokines. *Chemokines* attract other immune cells to the infection scene (in a process known as *chemotaxis*), whereas *cytokines* affect the behavior of nearby cells. Together the secreted chemokines and cytokines initiate a process known as inflammation. *Inflammation* is very beneficial to fighting infection: released cytokines increase the permeability of blood vessels allowing for a larger influx of immune cells which are then guided to the site of infection by the released chemokines. Also secreted cytokines increase the flow of lymph thereby allowing antigen-bearing cells to migrate faster to the nearest lymph node — also called tissue draining lymph node (LN) — where the adaptive immune response can be initiated. There are approximately 700 such LNs distributed throughout the body.

In summary, the immune system is the host's defense against infection with pathogens. This defense system is a layered system. The first line of defense consists of the physical barriers that prevent organisms from entering the host. In cases when this line of defense is breached, other innate immunity mechanisms will step up in a non-pathogen specific way. Only if innate immunity fails to clear the pathogen, adaptive immunity kicks in with the generation of pathogen specific T cells and B cells. In the next subsection we will describe how the immune system responds to an infection with Mtb.

1.1.2 Pathogenesis of pulmonary TB

Mycobacterium tuberculosis is an aerosol transmitted bacterium. It is a very infectious pathogen: inhalation of a single droplet containing just a few bacteria (as few as 2-5) can lead to infection [6,28]. This is remarkable, especially when considering that most infectious diseases require an initial dose of more than 1000 bacteria. Upon inhalation of an infected droplet, alveolar macrophages (macrophages resident in the lung tissue) will phagocytose the extracellular bacteria. For most pathogens this would be the end of the story. But not for Mtb: the very cells that are intended to destroy it prove to be an excellent breeding ground for Mtb. Normally a macrophage having taken up pathogen will try to dissolve it by fusing an intracellular structure called a lysosome to the encapsulated pathogen. Lysosomes contain enzymes that destroy bacteria. Mtb, however, has developed a mechanism to prevent this fusion from happening. Basically this means that the bacteria continue to reproduce inside a macrophage phagosome until that macrophage bursts after which the released bacteria will infect other macrophages and thus spreading the infection.

Despite the fact that Mtb is highly contagious, there is a large discrepancy between the number of infected people (in the order of billions) and the number of annual deaths (in the order of millions). This discrepancy points to another important fact about TB: not all infected people progress to active disease, rather most individuals achieve a latent state and while they are in this state they do not show clinical symptoms of disease nor are they able to infect others. So altogether we can conclude that the pathogen is able to achieve a very delicate balance with the immune system: the pathogen is not cleared while at the same time no symptoms of active disease develop. In order to achieve this balance, there must be mechanisms that counteract the seemingly uncontrolled intracellular growth of Mtb. One such mechanism is

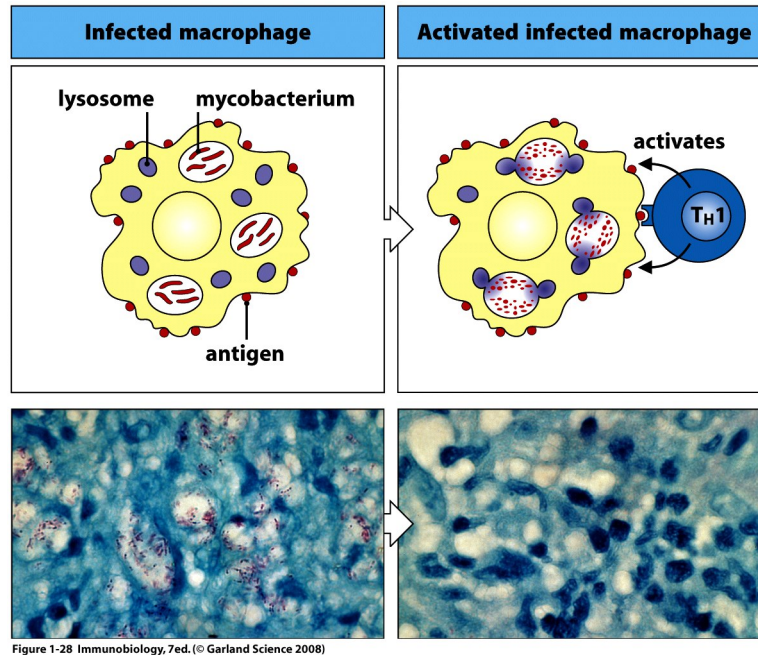


Figure 1-28 Immunobiology, 7ed. (© Garland Science 2008)

Figure 1.4: An infected macrophage that is unable to clear the ingested pathogen can be activated by T helper cells allowing lysosomes to fuse with the phagosomes and subsequently dissolving them. In the bottom pictures mycobacteria have red staining; observe that after macrophage activation no more bacteria are present. (Taken from Janeway’s Immunobiology [27].)

provided by T helper cells. T helper cells primed specifically for Mtb have the ability to activate infected macrophages allowing their lysosomes to fuse with and subsequently kill the encapsulated bacteria (see Figure 1.4). As a small side note, HIV, the causative agent of AIDS, attacks CD4+ T cells — exactly the T helper cells that activate infected macrophages. It is for this reason that HIV co-infection increases the risk of TB reactivation/development significantly.

A hallmark of TB and a direct consequence of the thus described balance is a so called granuloma. A granuloma is a spherical mass of immune cells aimed to wall off pathogens that cannot be eliminated by the immune system. A typical TB granuloma has a core of cheese-like caseous necrotic tissue (i.e. dead tissue) and extracellular bacteria surrounded by macrophages which on their turn are surrounded by a cuff of T cells (cf. Figure 1.5). People with latent TB have a 10% chance of reactivation during their lifetime [46]. Reactivation happens when a formed granuloma is unable to contain the infection and the bacteria disseminates. A person with active TB coughs up necrotic lung tissue containing extracellular bacteria that can infect others. When bacteria enter the blood stream they can disseminate even further to other parts of the body including the brain. If this happens then tuberculous meningitis can result.

1.1.3 Pathogenesis of TBM

In order to fully grasp the pathogenesis of TBM, a basic understanding of the anatomy of the central nervous system (CNS) is required. The central nervous system is comprised of the brain and the spinal cord, which are enveloped by *meninges*. The primary function of

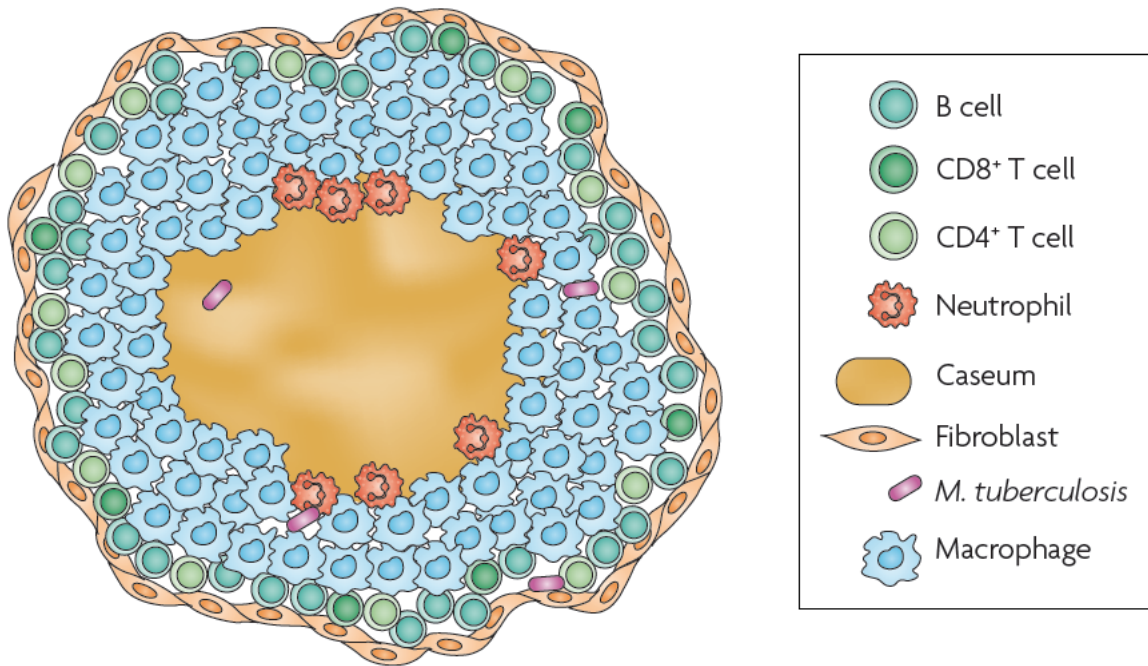


Figure 1.5: A typical TB granuloma consists of macrophages and lymphocytes surrounding a core of caseous tissue. Note that CD8+ T cells are typically cytotoxic T cells, whereas CD4+ T cells are T helper cells. (Taken from [3].)

the meninges is to offer protection to the CNS. Meninges are comprised by three layers: the dura mater, the arachnoid mater and the pia mater (see Figure 1.6a). In the subarachnoid space, the space between the arachnoid mater and the pia mater, cerebrospinal fluid (CSF) is present. Essentially this fluid acts as a protective cushion for the CNS. Ventricles, cavities in the CNS, also contain the same cerebrospinal fluid that flows through the subarachnoid space of the meninges (cf. Figure 1.6b).

The *blood-brain barrier* (BBB) separates the CNS from the regular systemic blood circulation. Because of the BBB, harmful substances normally do not reach the brain, protecting against damage that would be severe if not fatal. Unlike other organs, the CNS lacks a draining lymph node. For these reasons, the CNS used to be thought of as an *immunologically privileged* area, i.e. an area where no inflammatory immune response takes place upon the introduction of antigen. But recent studies indicate that this view is not entirely correct: both immune cells from outside the CNS, including lymphocytes, can cross the intact BBB, and microglia, the resident brain macrophages, do participate actively in immune responses in the CNS [9].

The most predominant manifestation of CNS TB is tuberculous meningitis. It is, as the name suggests, an infection of the meninges with *Mtb*. The pathogenesis of this disease was established in a meticulous work performed by Rich and McCordock in the 1930s [12, 34]. In their work they performed autopsies on deceased TBM patients and found that the majority had caseating granulomas in the brain parenchyma (tissue) or meninges. In honor of this work, such a caseating granuloma is called a Rich focus. Meningitis results when a disintegrating Rich focus ruptures either directly into the subarachnoid space of the meninges or through the ventricles. From there on, bacteria disseminate further causing wide-spread inflammation of the meninges (see Figures 1.7 and 1.8). As a result the flow of CSF is obstructed causing it to

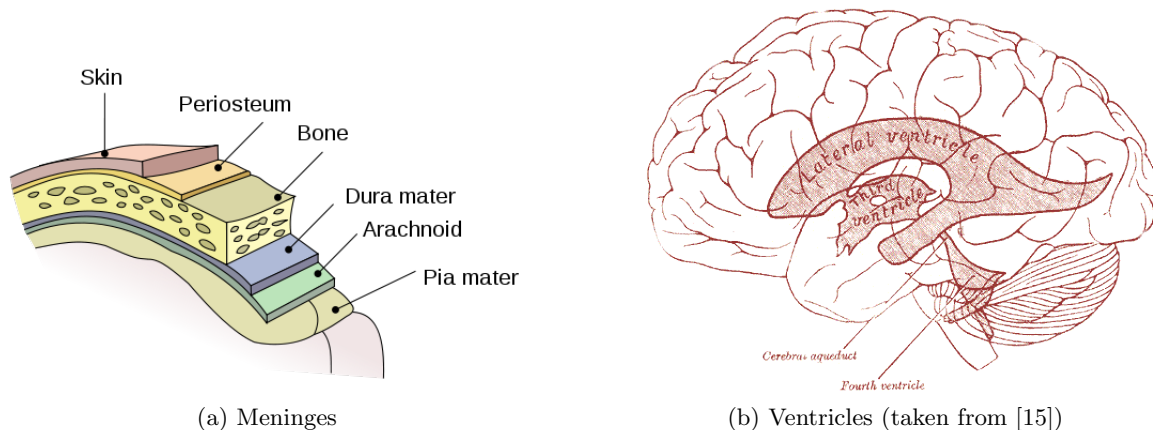


Figure 1.6: Cerebrospinal fluid is located in the subarachnoid space of the meninges and in the ventricles.

accumulate and resulting in *hydrocephalus*: increased pressure on the brain in the skull leading to irreversible brain damage. It is only during this late infection stage that TBM is typically diagnosed.

TBM is not the only form of CNS TB. When a Rich focus does not rupture into the meninges, it can continue to grow and become either a brain abscess — an enclosed collection of pus — or a tuberculoma — a solidified mass. In either case, depending on the location, severe neurological damage can result [39].

1.2 Previous and related work – theoretical and experimental

1.2.1 Modeling of TB

The first agent-based model describing the immune response in pulmonary TB was introduced by Segovia-Juarez et al. [36] Using this model the authors identified several factors that played a key role in granuloma formation, among which were chemokine diffusion and macrophage activation. Ray et al. extended the first model to include specific T cell functionality [33]. Using the extended model, the effect of individual functions of the cytokine $\text{TNF-}\alpha$ was studied. Note that such a study can only be performed in silico, as blocking or knocking-out $\text{TNF-}\alpha$ in vivo results in the loss of all $\text{TNF-}\alpha$ dependent functionality.

1.2.2 TBM

There are a lot of questions surrounding the pathogenesis of TBM. Most notably is the question on how the bacterium gains access to the CNS. Jain et al. tried to answer this question with an in vitro model of the human blood-brain barrier [17]. Their model suggested that *Mtb* have the ability to actively invade and traverse the endothelial cells lining up the BBB.

In addition to in vitro models, also animal models for TBM are described in the literature. The first murine model to study the pathogenesis of TBM was developed by van Well et al. After injecting *Mtb* directly into the brain of mice, the authors found that all the mice developed

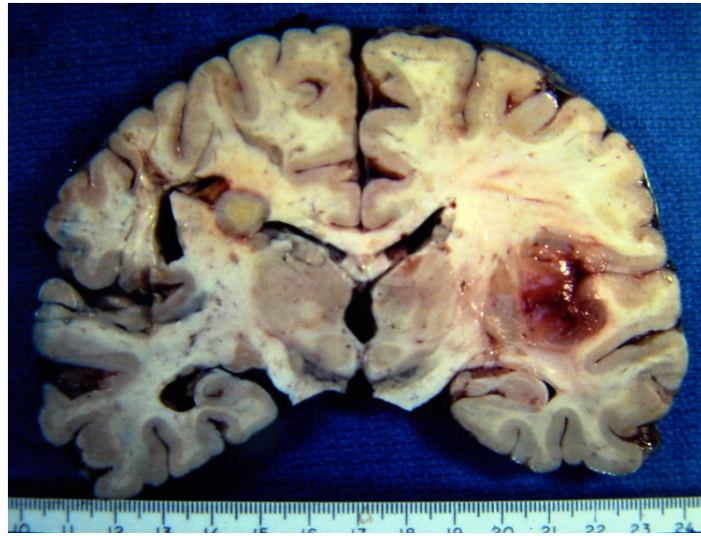
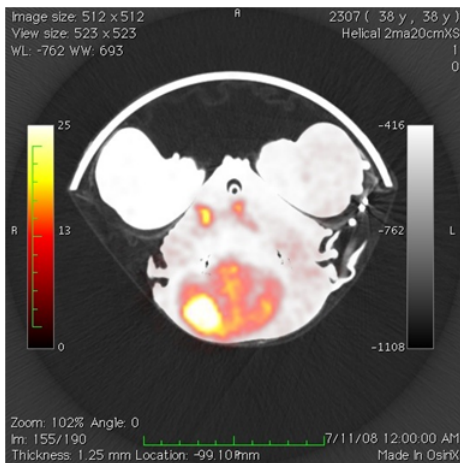
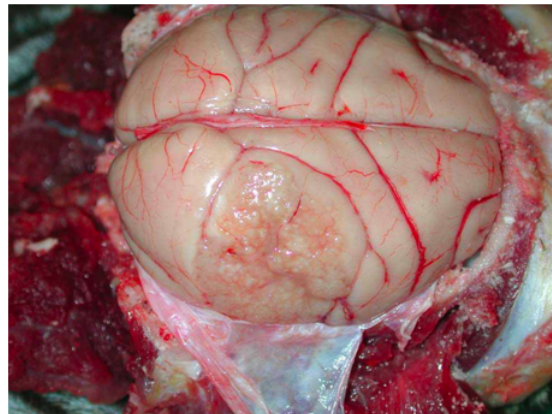


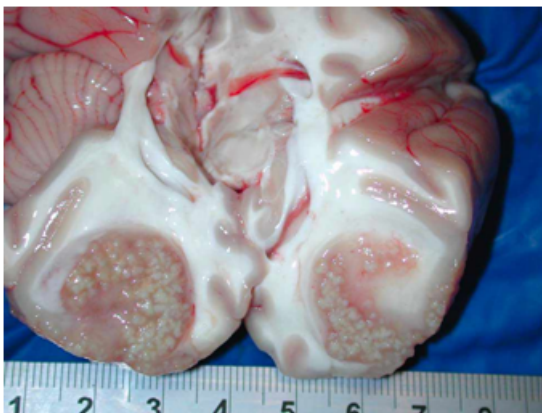
Figure 1.7: Cross-section of a brain affected by meningitis (irritation on the right side). Observe the large tuberculoma on the left side.



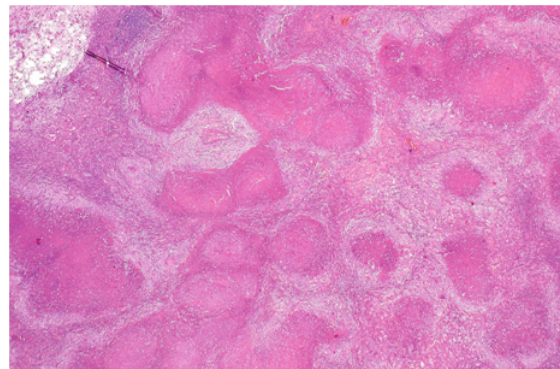
(a) MRI scan reveals brain anomalies.



(b) Observe the large lesion.



(c) Cross-section



(d) Staining

Figure 1.8: Granuloma in a monkey brain, courtesy by Dr. JoAnne Flynn.

meningitis and had elevated levels of chemokines and cytokines [45].

In another murine model Be et al. injected Mtb intravenously [4], i.e. directly into the bloodstream. This approach mimics the pathogenesis of TBM in humans where the disease results after hematogenous dissemination of Mtb. In their experiments the authors measured elevated bacteria numbers in the brain. However, no significant immune response could be detected there, as opposed to the lung tissue where extensive inflammation was detected.

Lee et al. chose to inject the BCG strain of *Mycobacterium bovis* into the brain of mice in their study [20], after which they were able to show that granulomas formed in the CNS. The authors also measured elevated numbers of microglia, macrophages, dendritic cells and T helper cells, as well as elevated levels of cytokines.

The group of Tsenova et al. used rabbit models to study TBM. In particular, they chose to study the effect of anti-inflammatory drugs on the pathogenesis of TBM [41–43]. They found that a combination of antibiotics and thalidomide, an anti-TNF- α drug, results in the highest survival rates.

1.3 Our contribution

In this work, we extend the agent-based model (ABM) by Ray et al. to include more down-regulatory mechanisms, and refined mechanisms for bacteria uptake by macrophages. In addition, we update a discrepancy in the computation of the diffusion of chemokines and cytokines. A new visualization framework is introduced allowing for a real-time overview of the grid, chemokine and cytokine concentrations and the immune cells.

Using the improved ABM, we developed to our knowledge the first in silico model describing the immune response and the pathogenesis of TBM. After fine-tuning this model, we perform several in silico experiments identifying parameters that dramatically affect the outcome of the disease.

Part II

Methods

Chapter 2

Modeling

In the previous chapter we described how the pathogenesis of TB and TBM proceeds. Also, we introduced some basic concepts of immunology. What one can notice is that there is a complex interplay between the individual parts that comprise the immune system. It is this interplay that allows the immune system to function properly and act appropriately on invading pathogens. In order to model the immune response to Mtb, we need to have a modeling method that captures the interplay in a bottom-up fashion starting from the individual parts — immune cells, Mtb, cytokines and chemokines — and resulting in the expected emergent behavior — the formation of a granuloma.

A modeling method specifically tailored toward modeling complex systems such as the immune system is agent-based modeling [7]. In an agent-based model (ABM) autonomous agents residing in a predefined environment make decisions based on a set of rules taking into account the agent's state and its neighborhood. This is in contrast to traditional differential equations models in which agents are assumed to be part of homogenous populations represented by continuous variables and where interactions take place on the population level and are inherently deterministic.

In this chapter we present two ABMs: in Section 2.1 we describe and extend an existing ABM of pulmonary TB, after which, in Section 2.2, we introduce a new ABM capturing the immune response to Mtb in the CNS. The methodology used for uncertainty analysis is presented in Section 2.3.

2.1 Lung model

In this section we describe and extend the Agent-Based Model (ABM) developed by Ray et al. [33] We start by giving an overview in Subsection 2.1.1. The rules specific to immune cells are presented in Subsection 2.1.2, followed by the cytokine and chemokine specific rules in Subsection 2.1.3. In Subsection 2.1.4 we describe the rules concerning Mtb. We conclude, in Subsection 2.1.5, by highlighting the extensions and giving an overview of the model parameters.

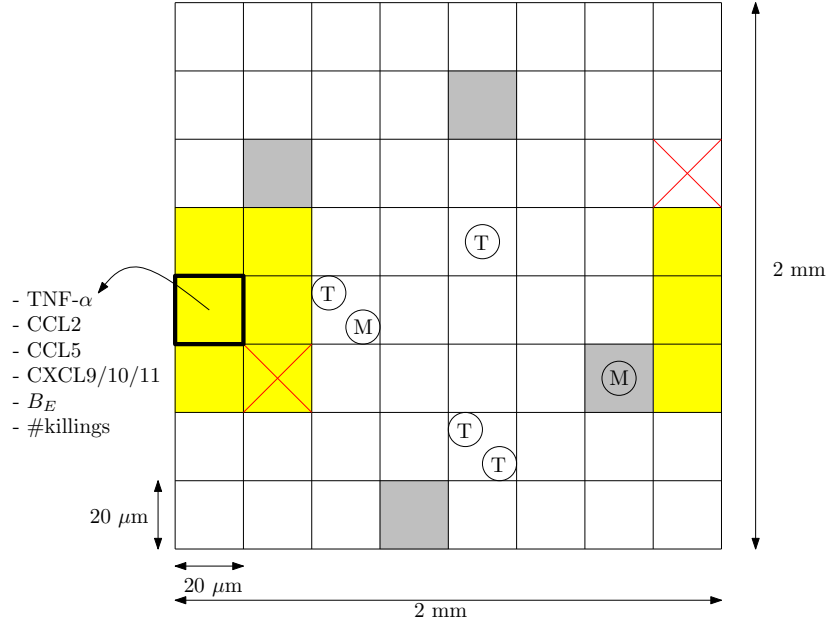


Figure 2.1: There are 10,000 micro-compartments on the grid, some of which are vascular sources (denoted by a gray background). Micro-compartments can also be caseated (denoted by a red cross). There is room for two immune cells in a micro-compartment; only one of which can be a macrophage (denoted by ‘M’), possibly accompanied by a T cell (denoted by ‘T’). Several continuous attributes are associated with every micro-compartment. The Moore neighborhood of the micro-compartment with fat borders is colored in yellow (it wraps over to the right edge, due to the grid being toroidal).

2.1.1 Model overview

The environment studied is a cross-section of $2\text{ mm} \times 2\text{ mm}$ of the alveolar lung tissue. The environment is modeled as a *uniform grid* comprised by 100×100 *micro-compartments*. Hence, the size of each micro-compartment is $20\text{ }\mu\text{m} \times 20\text{ }\mu\text{m}$. The grid has a toroidal boundary, i.e. the left edge and the right edge are connected, as well as the top and bottom edges. As such the Moore neighborhood of every micro-compartment consists of nine micro-compartments.

Out of the 10,000 micro-compartments, 50 are designated to be *vascular sources*. The number of sources was estimated from a pathologist working with Dr. Flynn’s group. In a single micro-compartment immune cells, extracellular bacteria and cytokines/chemokines can be present. Newly recruited immune cells enter the grid through vascular sources.

In this model, we consider two types of agents, namely discrete agents and continuous agents. Immune cells are modeled as *discrete agents*. The immune cells that are considered are macrophages and T cells. The latter are further classified into T_γ , T_c and T_{reg} cells. T_γ cells represent CD4+ and CD8+ T cells that produce the cytokine IFN- γ . Cytotoxic T cells are represented by T_c cells and T_{reg} cells correspond to regulatory T cells. The number of extracellular bacteria (denoted by B_E), and the levels of TNF- α , CCL2, CCL5, CXCL9/10/11 are modeled as *continuous attributes* of a micro-compartment. TNF- α is a cytokine, whereas CCL2, CCL5 and CXCL9/10/11 are chemokines. As mentioned in Subsection 1.1.1, movement of immune cells is governed by chemotaxis which in turn is dependent on the chemokine concentrations. TNF- α , working in concert with the chemokines, plays a role in the recruitment of new immune

cells (see Subsection 2.1.3).

In a micro-compartment at most two immune cells can be present. In case one of them is a macrophage, the other one can only be a T cell. This has to do with the diameter of a macrophage being $20\ \mu\text{m}$ [19] — the size of a micro-compartment. The size of a T cell is between $5\text{--}7\ \mu\text{m}$ [1]. While in theory more than two T cells fit in a micro-compartment, in the model only at most two T cells can be present in a single micro-compartment. Throughout the course of a simulation run of the model, micro-compartments can become caseous (dead tissue), in which case no immune cells can be present on those micro-compartments. In Figure 2.1 an example grid illustrating the described concepts is depicted.

In Algorithm 1 the pseudo code of the main loop of the ABM is given. There we can see that the model is initialized by calling the function INITIALIZE (see Algorithm 2). In this function a user-specified inoculum consisting of infected macrophages and extracellular bacteria is placed on the grid. In addition, N_M resting macrophages are placed on the grid uniformly at random. Furthermore, sources are chosen by first subdividing the grid into 49 equally sized clusters, per cluster one micro-compartment is flagged as a source, and finally one more additional source is chosen at random.

The model is run for 200 days, which corresponds to 28,800 10-minute time steps. The reason that we work with 10-minute time steps is due to the velocity of the fastest moving immune cell on the grid: namely the T cell which has an average velocity of $11\ \mu\text{m}/\text{min}$ within the lymph nodes of mice. Assuming that in alveolar lung tissue T cells move half an order of magnitude slower, we arrive at an approximate velocity of $20\ \mu\text{m}/10\ \text{min}$; in other words one micro-compartment per time step.

Algorithm 1: LUNGABM(\mathcal{G}, I)

Input: \mathcal{G} is a uniform grid and I is the initial inoculum

```

1 INITIALIZE( $\mathcal{G}, I$ )
2 for  $t \leftarrow 1$  to 28800 do
3   for  $t' \leftarrow 1$  to 100 do
4     MACSECRETE( $\mathcal{G}$ )
5     DIFFUSE( $\mathcal{G}$ )
6     DEGRADE( $\mathcal{G}$ )
7     MOVEMACROPHAGES( $\mathcal{G}, t$ )
8     MOVETCELLS( $\mathcal{G}$ )
9     RECRUIT( $\mathcal{G}, t$ )
10    UPDATESTATES( $\mathcal{G}, t$ )
11    GROWEXTMTB( $\mathcal{G}$ )

```

Chemokine and cytokine related rules proceed at a finer timescale of 6-second time steps. This has to do with the method used for solving the diffusion equation. After performing secretion, diffusion and degradation of TNF- α and the chemokines for 10 minutes, the immune cells are moved. Macrophage movement is dependent on the state of that macrophage, whereas T cells are moved every 10 minutes, in concordance with their reported velocity. Once all immune cells have moved, new immune cells are recruited through the sources taking into account the levels of TNF and chemokines at every source. This is followed by applying the immune cell specific rules and reproducing the extracellular bacteria. After that the next time step is computed in a similar fashion.

Algorithm 2: INITIALIZE(\mathcal{G}, I)

Input: \mathcal{G} is a uniform grid and I is the initial inoculum

- 1 Place initial inoculum I on \mathcal{G}
 - 2 Place N_M resting macrophages on random micro-compartments of the grid
 - 3 Subdivide \mathcal{G} in $7 \times 7 = 49$ clusters of neighboring micro-compartments
 - 4 In every such cluster, pick a micro-compartment c and set $c[\text{source}] \leftarrow \mathbf{true}$
 - 5 Pick, uniformly at random, one micro-compartment c that is not a source and set $c[\text{source}] \leftarrow \mathbf{true}$
-

Before we can start discussing the rules associated with the agents, we need to formalize the attributes associated to every micro-compartment. Let c be a micro-compartment; the following short hands for denoting the attributes of c will be used throughout the remainder of this chapter.

- $c[\text{TNF}]$ is the number of TNF- α molecules in c ;
- $c[\text{CCL2}]$ is the number of CCL2 molecules in c ;
- $c[\text{CCL5}]$ is the number of CCL5 molecules in c ;
- $c[\text{CXCL9}]$ is the number of CXCL9/10/11 molecules in c ;
- $c[B_E]$ is the number of extracellular bacteria in c ;
- $c[\text{killings}]$ is the number of killings contributing to caseation that have occurred in c ;
- $c[\text{caseous}]$ is a flag that is set when c is caseated;
- $c[\text{source}]$ is a flag that is set when c is a source.

2.1.2 Immune cell specific rules

In this subsection we look at the UPDATESTATES function invoked in Algorithm 1. This function concerns updating the states of the various immune cells on the grid. First all the macrophage state transitions are performed, followed by updating the states of the T_γ cells, T_c cells and finally the T_{reg} cells. If during a state transition an immune cell dies (e.g. due to age, or apoptosis), it is removed immediately from the grid. In order to prevent the order in which the various immune cell types are considered from playing a role, the state changes happen in a two-phase fashion: only after all immune cells are considered, the proposed new states are finalized. In Algorithm 3 the pseudo code of the UPDATESTATES function is given.

What all immune cells have in common are the attributes t_{birth} and t_{death} denoting, respectively, the birth and death time of an immune cell. The death time is determined during recruitment of the immune cell by taking the reported maximal lifespan into account (see Subsection 2.1.3). A macrophage has a maximal lifespan of 100 days [44], whereas effector T cells have a lifespan of at most 3 days [38].

In the remainder of this subsection we will describe the states and transitions of every immune cell. Sometimes the death of a macrophage contributes to caseation in the micro-compartment in which it resides; once the number of killings contributing to caseation reaches N_{caseous} , the

Algorithm 3: UPDATESTATES(\mathcal{G}, t)

Input: \mathcal{G} is a uniform grid and t is the current time step

- 1 Let \mathcal{L}_M , \mathcal{L}_γ , \mathcal{L}_c and \mathcal{L}_{reg} be lists containing all macrophages, T_γ cells, T_c cells and T_{reg} cells, respectively
 - 2 **foreach** $M \in \mathcal{L}_M$ **do**
 - 3 UPDATEMACSTATE(M, \mathcal{G}, t)
 - 4 **if** M [dead] **then** Remove M from \mathcal{G}
 - 5 **foreach** $T_\gamma \in \mathcal{L}_\gamma$ **do**
 - 6 UPDATETGAMSTATE(T_γ, \mathcal{G}, t)
 - 7 **if** T_γ [dead] **then** Remove T_γ from \mathcal{G}
 - 8 **foreach** $T_c \in \mathcal{L}_c$ **do**
 - 9 UPDATETCYTSTATE(T_c, \mathcal{G}, t)
 - 10 **if** T_c [dead] **then** Remove T_c from \mathcal{G}
 - 11 **foreach** $T_{reg} \in \mathcal{L}_{reg}$ **do**
 - 12 UPDATETREGSTATE(T_{reg}, \mathcal{G}, t)
 - 13 **if** T_{reg} [dead] **then** Remove T_{reg} from \mathcal{G}
 - 14 Finalize all state transitions, by updating the current states of all cells
-

micro-compartment in question becomes caseous and any accompanying T cell in that micro-compartment is killed. Since this is a commonly used routine, we have isolated the pseudo code in Algorithm 4.

Algorithm 4: CONTRIBUTETOCASEATION(c)

Input: c is a micro-compartment

- 1 c [killings] $\leftarrow c$ [killings] + 1
 - 2 **if** c [killings] = $N_{caseous}$ **then**
 - 3 Kill any T cell in c
 - 4 c [caseous] \leftarrow **true**
-

 T_{reg} cell

In the model, a T_{reg} cell has only one state: it is always active. If a T_{reg} cell dies due to age or due to TNF- α -induced apoptosis, its death does not contribute to caseation. As seen in the pseudo code of Algorithm 5, a T_{reg} cell down-regulates every immune cell in its Moore neighborhood. If the immune cell to be down-regulated is a macrophage then its transcription factor STAT1 is switched off. Note that T_{reg} cells themselves cannot be down-regulated.

 T_c cell

Just like other T cell types, the death of a T_c cell does not contribute to caseation. When a T_c is down-regulated by a T_{reg} cell, it does not perform its cytotoxic activities for a period of $\sigma_{c,deact}$ time steps. The cytotoxic activities of a T_c cell consist of killing infected or chronically infected macrophages in its local micro-compartment. Whenever such a macrophage is encountered, it is killed with a probability of $p_{c,kill}$. Killing a chronically infected macrophage has two

Algorithm 5: UPDATE_TREG_STATE($T_{\text{reg}}, \mathcal{G}, t$)

Input: T_{reg} is T_{reg} cell, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $T_{\text{reg}}$  resides
2 if  $t = T_{\text{reg}}[t_{\text{death}}]$  then
3    $T_{\text{reg}}[\text{dead}] \leftarrow \text{true}$ 
4 else if  $c[\text{TNF}] > \tau_{\text{TNF,apopt}}$  and with probability  $p_{\text{TNF,apopt}}$  then
5    $T_{\text{reg}}[\text{dead}] \leftarrow \text{true}$ 
6 else
7   foreach immune cell  $ic$  in the Moore neighborhood of  $c$  that is not a  $T_{\text{reg}}$  cell do
8      $ic[t_{\text{deact}}] \leftarrow t$ 
9      $ic[\text{deact}] \leftarrow \text{true}$ 
10    if  $ic$  is a macrophage then
11       $ic[\text{STAT1}] \leftarrow \text{false}$ 

```

possible outcomes: either the macrophage is killed cleanly and none of its intracellular bacteria are dispersed, or the killing proceeds in a less clean manner resulting in the dispersion of all intracellular bacteria over the Moore neighborhood. The probability of killing a chronically infected macrophage in a clean manner is given by $p_{c,\text{kill,cleanly}}$. Cytotoxic killing of a macrophage contributes to caseation. In Algorithm 6 the pseudo code of this procedure is given.

Algorithm 6: UPDATE_TCYT_STATE(T_c, \mathcal{G}, t)

Input: T_c is T_c cell, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $T_c$  resides
2 if  $t = T_c[t_{\text{death}}]$  then
3    $T_c[\text{dead}] \leftarrow \text{true}$ 
4 else if  $c[\text{TNF}] > \tau_{\text{TNF,apopt}}$  and with probability  $p_{\text{TNF,apopt}}$  then
5    $T_c[\text{dead}] \leftarrow \text{true}$ 
6 else
7   if  $T_c[\text{deact}]$  and  $T_c[t_{\text{deact}}] + \sigma_{c,\text{deact}} = t$  then
8      $T_c[\text{deact}] \leftarrow \text{false}$ 
9   if  $T_c[\text{deact}] = \text{false}$ 
10    and  $c$  has an infected or chronically infected macrophage
11    and with probability  $p_{c,\text{kill}}$  then
12      Let  $M$  be the macrophage in  $c$ 
13      if  $M[\text{state}] = \text{Chronically infected}$  and with probability  $(1 - p_{c,\text{kill,cleanly}})$  then
14        Disperse all intracellular bacteria over the Moore neighborhood of  $c$ 
15       $M[\text{dead}] \leftarrow \text{true}$ 
16      CONTRIBUTE_TO_CASEATION( $c$ )

```

 T_γ cell

In addition to playing a crucial in the activation of macrophages (by enabling the transcription factor STAT1 as we will see later on), T_γ cells have the ability to induce, with a probability of $p_{\text{Fas,apopt}}$, so called Fas/FasL apoptosis in infected and chronically infected macrophages resulting in the dispersion of half of the intracellular bacteria over the Moore neighborhood.

Unlike TNF- α -induced apoptosis, apoptosis due to the Fas/FasL pathway does contribute to caseation (see Algorithm 7). If a T_γ cell is down-regulated, it does not induce apoptosis for a period specified by $\sigma_{\gamma,\text{deact}}$.

Algorithm 7: UPDATETGAMSTATE(T_γ, \mathcal{G}, t)

Input: T_γ is T_γ cell, \mathcal{G} is a uniform grid and t is the current time step

- 1 Let c denote the micro-compartment in which T_γ resides
- 2 **if** $t = T_\gamma[t_{\text{death}}]$ **then**
- 3 $T_\gamma[\text{dead}] \leftarrow \text{true}$
- 4 **else if** $c[\text{TNF}] > \tau_{\text{TNF,apopt}}$ **and** with probability $p_{\text{TNF,apopt}}$ **then**
- 5 $T_\gamma[\text{dead}] \leftarrow \text{true}$
- 6 **else**
- 7 **if** $T_\gamma[\text{deact}]$ **and** $T_\gamma[t_{\text{deact}}] + \sigma_{\gamma,\text{deact}} = t$ **then**
- 8 $T_\gamma[\text{deact}] \leftarrow \text{false}$
- 9 **if** $T_\gamma[\text{deact}] = \text{false}$
- 10 **and** c has an infected or chronically infected macrophage
- 11 **and** with probability $p_{\text{Fas,apopt}}$ **then**
- 12 Let M be the macrophage in c
- 13 Kill half of the intracellular bacteria of M
- 14 Disperse the other half over the Moore neighborhood of c
- 15 $M[\text{dead}] \leftarrow \text{true}$
- 16 CONTRIBUTETOCASEATION(c)

Macrophage

Macrophages have the ability to phagocytose extracellular bacteria. In case of Mtb, the bacteria is not necessarily cleared (cf. Subsection 1.1.2). Therefore we keep track of the number of intracellular bacteria in a macrophage, which we denote by B_I . There are four states attainable by a macrophage: resting, infected, chronically infected and active. In addition to these four states, there are two transcription factors: NF κ B and STAT1. In case a macrophage is down-regulated by a T_{reg} cell, the property ‘deact’ is set and the time at which the macrophage was down-regulated is denoted by t_{deact} . The time span during which a macrophage remains down-regulated is a parameter denoted by $\sigma_{\text{M,deact}}$.

In case a macrophage dies due to aging, its intracellular bacteria are released to the micro-compartment the macrophage resides in. In addition, when the macrophage is in the active state, its death will contribute to caseation. Should a micro-compartment become caseous, i.e. the number of killings equals the parameter N_{caseous} , any T cell residing in that micro-compartment is also killed. The second cause of death is due to TNF- α -induced apoptosis (programmed cell death). This happens only when the number of TNF- α molecules in the micro-compartment is more than $\tau_{\text{TNF,apopt}}$ and on top of that a probability of $p_{\text{TNF,apopt}}$ is satisfied. In contrast to death by aging, TNF- α -induced apoptosis kills half of the intracellular bacteria and the other half is dispersed over the Moore neighborhood. Macrophages that are not down-regulated can execute their state-specific rules. Before this is done, the NF κ B property is reevaluated; it is enabled if

- the macrophage is infected or active, or

Algorithm 8: UPDATEMACSTATE(M, \mathcal{G}, t)

Input: M is a macrophage, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $M$  resides
2 if  $t = M[t_{\text{death}}]$  then
3    $c[B_E] \leftarrow c[B_E] + M[B_I]$ 
4   if  $M[\text{state}] = \text{Active}$  then
5     CONTRIBUTETOCASEATION( $c$ )
6    $M[\text{dead}] \leftarrow \text{true}$ 
7 else if  $c[\text{TNF}] > \tau_{\text{TNF,apopt}}$  and with probability  $p_{\text{TNF,apopt}}$  then
8   Kill half of the intracellular bacteria of  $M$ 
9   Disperse the other half over the Moore neighborhood of  $c$ 
10   $M[\text{dead}] \leftarrow \text{true}$ 
11 else
12  if  $M[\text{deact}]$  and  $M[t_{\text{deact}}] + \sigma_{M,\text{deact}} = t$  then
13     $M[\text{deact}] \leftarrow \text{false}$ 
14  if  $M[\text{deact}] = \text{false}$  then
15    Set  $M[\text{NF}\kappa\text{B}]$  to true if the state is infected or active, or the number of extracellular
    bacteria in the Moore neighborhood is larger than  $N_{M,\text{Be},\text{NF}\kappa\text{B}}$ . Otherwise set
     $M[\text{NF}\kappa\text{B}]$  to false.
16  switch  $M[\text{state}]$  do
17    case Resting HANDLEMACRESTING( $M, \mathcal{G}, t$ )
18    case Infected HANDLEMACINFECTED( $M, \mathcal{G}, t$ )
19    case Chronically infected HANDLEMACCHRONICALLYINFECTED( $M, \mathcal{G}, t$ )
20    case Active HANDLEMACACTIVE( $M, \mathcal{G}, t$ )

```

- the number of TNF- α molecules in the macrophage's micro-compartment is more than $\tau_{M,TNF,NF\kappa B}$, OR
- the number of extracellular bacteria in the macrophage's Moore neighborhood is more than $N_{M,Be,NF\kappa B}$.

If none of the previous conditions are met, the NF κ B property is switched off. The pseudo code corresponding to this description is given in Algorithm 8. In the next paragraphs we will look at the state-specific rules.

Algorithm 9: HANDLEMACRESTING(M, \mathcal{G}, t)

Input: M is a resting macrophage, \mathcal{G} is a uniform grid and t is the current time step

- 1 Let c denote the micro-compartment in which M resides
- 2 **if** $c[B_E] \leq 1$ **then**
 - /* Kill an extracellular bacterium if there is only 1 (or less) */
- 3 $c[B_E] \leftarrow 0$
- 4 $M[\text{state}'] \leftarrow \text{Resting}$
- 5 **else if** with probability $p_{M,\text{kill}}$ **then**
 - /* Kill an extracellular bacterium and remain resting */
- 6 $c[B_E] \leftarrow c[B_E] - 1$
- 7 $M[\text{state}'] \leftarrow \text{Resting}$
- 8 **else**
 - /* Become infected with an extracellular bacterium */
- 9 $c[B_E] \leftarrow c[B_E] - 1$
- 10 $M[B_I] \leftarrow 1$
- 11 $M[\text{state}'] \leftarrow \text{Infected}$
- 12 Enable $M[\text{STAT1}]$ with probability $\text{NUMBEROFACTIVE}T\text{GAMINMOORE}(c) * p_{M,\text{STAT1}}$
- 13 **if** $M[\text{STAT1}]$ **and** $M[\text{NF}\kappa\text{B}]$ **then**
- 14 $M[B_I] \leftarrow 0$
 - /* Activated macrophages have a shortened lifespan of 10 days */
- 15 $M[t_{\text{death}}] \leftarrow \min(M[t_{\text{death}}], t + 1440)$
- 16 $M[\text{state}'] \leftarrow \text{Active}$

Recall that the number of extracellular bacteria is modeled as a continuous attribute of every micro-compartment. It can therefore happen that there is just a fraction of less than one bacterium present in a micro-compartment. If a resting macrophage encounters only one extracellular bacterium, or a fraction of it, the macrophage will kill the bacterium and remain resting. Otherwise, if more than one extracellular bacterium is present, two things can happen: either with probability $p_{M,\text{kill}}$ the macrophage remains resting by killing a single extracellular bacterium, or the macrophage becomes infected with that same extracellular bacterium. For a resting macrophage to become activated, both NF κ B and STAT1 need to be enabled. The latter is enabled with a probability that is proportional to the number of active T_γ cells in the Moore neighborhood. In Algorithm 9 the pseudo code of this procedure is given.

We can see in Algorithm 10 that an infected macrophage is still able to phagocytose extracellular bacteria. The probability of this occurring, however, decreases with the number of intracellular bacteria present in the infected macrophage (see Figure 2.2). Activation of an infected macrophage occurs when both NF κ B and STAT1 are enabled; the latter is enabled with

Algorithm 10: HANDLEMACINFECTED(M, \mathcal{G}, t)

Input: M is an infected macrophage, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $M$  resides
2 GROWINTMTB( $M$ )
3  $p \leftarrow (N_{M,Bi,Ci} - M[B_I])/100$ 
4 if  $c[B_E] > 0$  and with probability  $p$  then
    /* The less intracellular bacteria are present, the higher the probability
       for taking up more will be */
5  $\Delta B \leftarrow \min(c[B_E], 1)$ 
6  $c[B_E] \leftarrow c[B_E] - \Delta B$ 
7  $M[B_I] \leftarrow M[B_I] + \Delta B$ 
8 if  $M[B_I] \geq N_{M,Bi,Ci}$  then
    /* Become chronically infected, if the number of intracellular bacteria
       exceeds  $N_{M,Bi,Ci}$  */
9  $M[\text{state}'] \leftarrow \text{Chronically infected}$ 
10 else
11 Enable  $M[\text{STAT1}]$  with probability  $\text{NUMBEROFACTIVE TGAMINMOORE}(c) * p_{M,\text{STAT1}}$ 
12 if  $M[\text{STAT1}]$  and  $M[\text{NF}\kappa\text{B}]$  then
13  $M[B_I] \leftarrow 0$ 
    /* Activated macrophages have a shortened lifespan of 10 days */
14  $M[t_{\text{death}}] \leftarrow \min(M[t_{\text{death}}], t + 1440)$ 
15  $M[\text{state}'] \leftarrow \text{Active}$ 
16 else
17  $M[\text{state}'] \leftarrow \text{Infected}$ 

```

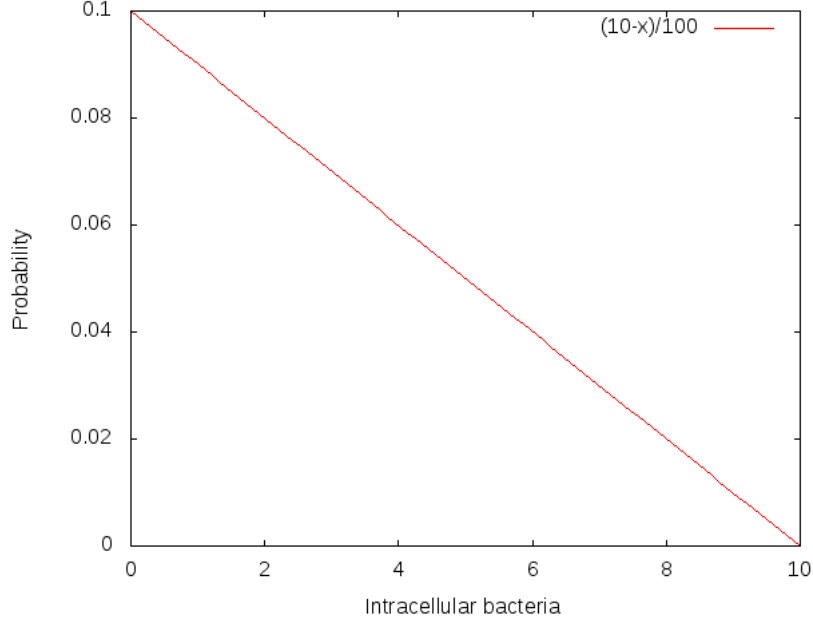


Figure 2.2: Probability of taking up bacteria decreases, as the number of intracellular bacteria increases; in this plot we have that $N_{M, Bi, CI} = 10$.

a probability proportional to the number of active T_γ cells in the Moore neighborhood, whereas the former depends on $TNF-\alpha$ in the local micro-compartment and extracellular bacteria in the neighborhood. When the number of intracellular bacteria exceeds the parameter $N_{M, Bi, CI}$, the macrophage becomes chronically infected. At that point the fate of the macrophage is sealed: the intracellular bacteria will continue to reproduce until they grow beyond $N_{M, Bi, burst}$ causing the macrophage to burst (see Algorithm 11). The bursting contributes to caseation of the local micro-compartment and the dispersion of all intracellular bacteria over the Moore neighborhood.

Algorithm 11: HANDLEMACCHRONICALLYINFECTED(M, \mathcal{G}, t)

Input: M is a chronically infected macrophage, \mathcal{G} is a uniform grid and t is the current time step

- 1 Let c denote the micro-compartment in which M resides
 - 2 GROWINTMTB(M)
 - 3 **if** $M[B_I] \geq N_{M, Bi, burst}$ **then**
 - 4 Disperse all intracellular bacteria over the Moore neighborhood of c
 - 5 $M[dead] \leftarrow \mathbf{true}$
 - 6 CONTRIBUTETOCASEATION(c)
 - 7 **else**
 - 8 $M[state'] \leftarrow \mathbf{Chronically\ infected}$
-

An activated macrophage has a shortened lifespan of at most 10 days. During its remaining lifespan it has the ability to kill extracellular bacteria in its local micro-compartment. Unlike a resting macrophage, an activated macrophage can kill more than one extracellular bacterium (as many as $N_{M, Be, kill}$) without being restricted by the total number of extracellular bacteria

in its local micro-compartment (see Algorithm 12).

Algorithm 12: HANDLEMACACTIVE(M, \mathcal{G}, t)

Input: M is an active macrophage, \mathcal{G} is a uniform grid and t is the current time step

- 1 Let c denote the micro-compartment in which M resides
 - 2 $c[B_E] \leftarrow c[B_E] - \mathbf{min}(c[B_E], N_{M,Be,kill})$
 - 3 $M[\text{state}] \leftarrow \text{Active}$
-

2.1.3 Cytokine/chemokine specific rules

In this subsection we describe the rules concerning the cytokine TNF- α and the chemokines CCL2, CCL5 and CXCL9/10/11. We start by describing how they are secreted, diffused and degraded, followed by a description of their effects on recruitment and movement of immune cells.

Secretion

In the model macrophages are the only type of immune cell able to secrete the cytokine TNF- α and the chemokines CCL2, CCL5 and CXCL9/10/11. Secretion by macrophages is dependent on the transcription factor NF κ B and the state of the macrophage. Macrophages that are not in the resting state and have the transcription factor NF κ B enabled secrete at maximal rate. Secretion at half rate occurs when NF κ B is enabled and the macrophage is resting. In addition, infected macrophages that do not have NF κ B enabled also secrete at half rate. Note that chronically infected macrophages always secrete at maximal rate, as NF κ B is enabled by definition for such macrophages (which makes sense biologically as Mtb has the ability to induce the NF κ B-signalling pathway via the interaction with specific types of Toll-like receptors on the cell surface of a macrophage [27]). The rates by which secretion occurs for TNF- α , CCL2, CCL5 and CXCL9 are denoted by s_{TNF} , s_{CCL2} , s_{CCL5} and s_{CXCL9} , respectively. In Algorithm 13 the pseudo code concerning secretion is given.

Diffusion

In every 6-second time step secretion is followed by diffusion. In the following paragraphs we derive the diffusion method used using a forward Euler scheme.

The assumption is that we are dealing with a constant diffusion coefficient D . Therefore the heat equation can be used to describe diffusion:

$$\frac{\partial u}{\partial t} = D\nabla^2 u = D \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right). \quad (2.1)$$

Here $u(x, y, t)$ is a function of two spatial variables x and y and the time variable t . The idea now is to discretize the equation in space and then to solve the resulting system of ordinary differential equations (ODEs) in order to calculate the solution. Note that this is a very natural thing to do, as the various concentrations are already stored in a discrete fashion on \mathcal{G} . The space discretization is done on a 102×102 grid. The grid is bigger than 100×100 , as we reserve

Algorithm 13: MACSECRETE(\mathcal{G})

Input: \mathcal{G} is a uniform grid

```

1 for  $i \leftarrow 1$  to 100 do
2   for  $j \leftarrow 1$  to 100 do
3     if  $\mathcal{G}_{ij}$  has a macrophage then
4       Let  $c$  be the micro-compartment at  $\mathcal{G}_{ij}$ 
5       Let  $M$  be the macrophage at  $c$ 
6       if  $M[\text{NF}\kappa\text{B}]$  and  $M[\text{state}] \neq \text{Resting}$  then
7          $c[\text{TNF}] \leftarrow c[\text{TNF}] + s_{\text{TNF}}$ 
8          $c[\text{CCL2}] \leftarrow c[\text{CCL2}] + s_{\text{CCL2}}$ 
9          $c[\text{CCL5}] \leftarrow c[\text{CCL5}] + s_{\text{CCL5}}$ 
10         $c[\text{CXCL9}] \leftarrow c[\text{CXCL9}] + s_{\text{CXCL9}}$ 
11      else if ( $M[\text{NF}\kappa\text{B}]$  and  $M[\text{state}] = \text{Resting}$ ) or  $M[\text{state}] = \text{Infected}$  then
12         $c[\text{TNF}] \leftarrow c[\text{TNF}] + 0.5 \cdot s_{\text{TNF}}$ 
13         $c[\text{CCL2}] \leftarrow c[\text{CCL2}] + 0.5 \cdot s_{\text{CCL2}}$ 
14         $c[\text{CCL5}] \leftarrow c[\text{CCL5}] + 0.5 \cdot s_{\text{CCL5}}$ 
15         $c[\text{CXCL9}] \leftarrow c[\text{CXCL9}] + 0.5 \cdot s_{\text{CXCL9}}$ 

```

the outer edges for the boundary conditions. The coordinates of the nodes in the grid are given by

$$\begin{aligned} x_i &= (i - 1)\Delta x, \\ y_j &= (j - 1)\Delta y \end{aligned} \tag{2.2}$$

where $0 \leq i, j \leq 101$.

From now on we define $1 \leq i, j \leq 100$ and we introduce the shorthand $u_{i,j}(t) = u(x_i, y_j, t)$. The boundary conditions are

$$\begin{aligned} u_{0,0}(t) &= 0, \\ u_{0,j}(t) &= 0, \\ u_{i,0}(t) &= 0, \\ u_{101,101}(t) &= 0, \\ u_{101,j}(t) &= 0, \\ u_{i,101}(t) &= 0, \\ u_{i,j}(t) &= g(i, j) \end{aligned} \tag{2.3}$$

where $g(i, j)$ is the initial concentration at \mathcal{G}_{ij} .

We can approximate the second-order partial derivatives in Equation 2.1 using central differ-

ences. We do this by considering the Taylor series expansion about the point (x_i, y_j) :

$$u_{i+1,j} = u_{i,j} + \Delta x \frac{\partial u_{i,j}}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{(\Delta x)^3}{6} \frac{\partial^3 u_{i,j}}{\partial x^3} + O((\Delta x)^4), \quad (2.4)$$

$$u_{i-1,j} = u_{i,j} - \Delta x \frac{\partial u_{i,j}}{\partial x} + \frac{(\Delta x)^2}{2} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{(\Delta x)^3}{6} \frac{\partial^3 u_{i,j}}{\partial x^3} + O((\Delta x)^4), \quad (2.5)$$

$$u_{i,j+1} = u_{i,j} + \Delta y \frac{\partial u_{i,j}}{\partial y} + \frac{(\Delta y)^2}{2} \frac{\partial^2 u_{i,j}}{\partial y^2} + \frac{(\Delta y)^3}{6} \frac{\partial^3 u_{i,j}}{\partial y^3} + O((\Delta y)^4), \quad (2.6)$$

$$u_{i,j-1} = u_{i,j} - \Delta y \frac{\partial u_{i,j}}{\partial y} + \frac{(\Delta y)^2}{2} \frac{\partial^2 u_{i,j}}{\partial y^2} - \frac{(\Delta y)^3}{6} \frac{\partial^3 u_{i,j}}{\partial y^3} + O((\Delta y)^4). \quad (2.7)$$

From Equations 2.4 and 2.5 we now have

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}. \quad (2.8)$$

Similarly, combining Equations 2.6 and 2.7 and discarding the error term gives

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2}. \quad (2.9)$$

Since \mathcal{G} is a uniform grid, we have that $\Delta x = \Delta y$. Therefore

$$\frac{\partial u}{\partial t} \approx D \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{(\Delta x)^2}. \quad (2.10)$$

We now have a system of $100 \cdot 100 = 10,000$ coupled ordinary differential equations. We can solve this system by using for instance the forward Euler method, which gives us

$$u_{i,j}^{n+1} = u_{i,j}^n + \mu(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n), \quad (2.11)$$

where $\mu = \Delta t D / (\Delta x)^2$.

Let $C_{i,j}(t)$ be the concentration value in micro-compartment $\mathcal{G}_{i,j}$ at time $t \in \mathbb{R}^+$. Using Equation 2.11 and the previously defined boundary conditions, we obtain the following recursive equation for $C_{i,j}$:

$$C_{0,0}(t) = 0, \quad (2.12)$$

$$C_{0,j}(t) = 0, \quad (2.13)$$

$$C_{i,0}(t) = 0, \quad (2.14)$$

$$C_{101,101}(t) = 0, \quad (2.15)$$

$$C_{101,j}(t) = 0, \quad (2.16)$$

$$C_{i,101}(t) = 0, \quad (2.17)$$

$$C_{i,j}(0) = g(i, j), \quad (2.18)$$

$$C_{i,j}(t + \Delta t) = C_{i,j}(t) + \mu(C_{i+1,j}(t) + C_{i-1,j}(t) + C_{i,j+1}(t) + C_{i,j-1}(t) - 4C_{i,j}(t)) \quad (2.19)$$

where $t \in \mathbb{R}^+$ and $g(i, j)$ is the initial concentration at $\mathcal{G}_{i,j}$. Note that rearranging the terms in Equation 2.19 results in the diffusion equation given in Reference [36] (see Section A.1).

We assume that the modeled chemokines have the same diffusion coefficient, which we denote by D_{CC} . The diffusion coefficient of TNF- α is denoted by the parameter D_{TNF} . A necessary but not sufficient condition for the Euler method to converge in our case is given by [11]

$$\frac{\Delta t D}{(\Delta x)^2} < 0.25. \quad (2.20)$$

Typically the unit used for diffusivity is cm^2/s . We have that $\Delta x = 20 \mu\text{m} = 2 \cdot 10^{-3} \text{cm}$. Choosing $\Delta t = 6 \text{s}$ yields the following bound for D :

$$D < 1.67 \cdot 10^{-7} \text{cm}^2/\text{s}. \quad (2.21)$$

In Algorithm 14 the pseudo code of the DIFFUSE function is given.

Algorithm 14: DIFFUSE(\mathcal{G})

Input: \mathcal{G} is a uniform grid; the micro-compartment at the bottom-left corner is $\mathcal{G}_{1,1}$

```

1 Let  $\mathcal{G}'$  be a uniform grid of the same dimensions as  $\mathcal{G}$ 
2  $\mu_{TNF} \leftarrow \frac{6 \cdot D_{TNF}}{4 \cdot 10^{-6}}$ 
3  $\mu_{CC} \leftarrow \frac{6 \cdot D_{CC}}{4 \cdot 10^{-6}}$ 
4  $c_0[\text{TNF}] \leftarrow 0$ 
5  $c_0[\text{CCL2}] \leftarrow 0$ 
6  $c_0[\text{CCL5}] \leftarrow 0$ 
7  $c_0[\text{CXCL9}] \leftarrow 0$ 
8 for  $i \leftarrow 1$  to 100 do
9   for  $j \leftarrow 1$  to 100 do
10    Let  $c$  and  $c'$  be the micro-compartment at  $\mathcal{G}_{ij}$  and  $\mathcal{G}'_{ij}$ , respectively
11    if  $i = 1$  then  $c_{\text{down}} \leftarrow c_0$  else  $c_{\text{down}} \leftarrow \mathcal{G}_{i-1,j}$ 
12    if  $i = 100$  then  $c_{\text{up}} \leftarrow c_0$  else  $c_{\text{up}} \leftarrow \mathcal{G}_{i+1,j}$ 
13    if  $j = 1$  then  $c_{\text{left}} \leftarrow c_0$  else  $c_{\text{left}} \leftarrow \mathcal{G}_{i,j-1}$ 
14    if  $j = 100$  then  $c_{\text{right}} \leftarrow c_0$  else  $c_{\text{right}} \leftarrow \mathcal{G}_{i,j+1}$ 
15     $c'[\text{TNF}] \leftarrow c[\text{TNF}] + \mu_{TNF} \cdot (c_{\text{down}}[\text{TNF}] + c_{\text{up}}[\text{TNF}] + c_{\text{left}}[\text{TNF}] + c_{\text{right}}[\text{TNF}])$ 
16     $c'[\text{CCL2}] \leftarrow c[\text{CCL2}] + \mu_{CC} \cdot (c_{\text{down}}[\text{CCL2}] + c_{\text{up}}[\text{CCL2}] + c_{\text{left}}[\text{CCL2}] + c_{\text{right}}[\text{CCL2}])$ 
17     $c'[\text{CCL5}] \leftarrow c[\text{CCL5}] + \mu_{CC} \cdot (c_{\text{down}}[\text{CCL5}] + c_{\text{up}}[\text{CCL5}] + c_{\text{left}}[\text{CCL5}] + c_{\text{right}}[\text{CCL5}])$ 
18     $c'[\text{CXCL9}] \leftarrow$ 
         $c[\text{CXCL9}] + \mu_{CC} \cdot (c_{\text{down}}[\text{CXCL9}] + c_{\text{up}}[\text{CXCL9}] + c_{\text{left}}[\text{CXCL9}] + c_{\text{right}}[\text{CXCL9}])$ 
19  $\mathcal{G} \leftarrow \mathcal{G}'$ 

```

Degradation

Degradation of TNF- α and the chemokines also proceeds at the finer timescale of 6-second time steps. The two parameters δ_{TNF} and δ_{CC} denote the degradation coefficients for TNF- α and the chemokines, respectively. In Algorithm 15 the pseudo code concerning degradation is given.

Recruitment

As mentioned in Subsection 2.1.1, out of the 10,000 micro-compartment 50 are vascular sources. It is through these 50 sources that new immune cells are recruited. Only non-caseous sources can

Algorithm 15: DEGRADE(\mathcal{G})

Input: \mathcal{G} is a uniform grid of 100×100 micro-compartments

```

1 for  $i \leftarrow 1$  to 100 do
2   for  $j \leftarrow 1$  to 100 do
3     Let  $c$  be the micro-compartment at  $\mathcal{G}_{ij}$ 
4      $c[\text{TNF}] \leftarrow c[\text{TNF}] - \delta_{\text{TNF}} \cdot c[\text{TNF}]$ 
5      $c[\text{CCL2}] \leftarrow c[\text{CCL2}] - \delta_{\text{CC}} \cdot c[\text{CCL2}]$ 
6      $c[\text{CCL5}] \leftarrow c[\text{CCL5}] - \delta_{\text{CC}} \cdot c[\text{CCL5}]$ 
7      $c[\text{CXCL9}] \leftarrow c[\text{CXCL9}] - \delta_{\text{CC}} \cdot c[\text{CXCL9}]$ 

```

recruit immune cells. Recruitment happens by considering the sources iteratively. If a source is not caseated and can accommodate a macrophage, we try to recruit one at that source. If there is still room for a T cell and the current time step is past t_{recr} , possibly a T cell will be recruited at the non-caseated source in question.

Algorithm 16: RECRUIT(\mathcal{G}, t)

Input: \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $\mathcal{L}_{\text{sources}}$  be the list of vascular sources on  $\mathcal{G}$  (by definition  $|\mathcal{L}_{\text{sources}}| = 50$ )
2 foreach  $c \in \mathcal{L}_{\text{sources}}$  do
3   if  $c[\text{caseous}] = \text{false}$  then
4     if #cells at  $c$  is less than 2 and no macrophage is present on  $c$  then
5       RECRUITMAC( $c, \mathcal{G}, t$ )
6   if #cells at  $c$  is less than 2 and  $t \geq t_{\text{recr}}$  then
7     RECRUITTCELL( $c, \mathcal{G}, t$ )

```

In the model there is a lower bound, denoted by N_M , on the number of macrophages on the grid. If the number of macrophages drops below the lower bound, new macrophages are actively recruited regardless of the TNF- α and chemokines levels. In the normal situation, i.e. when the number of macrophages is above the lower bound, macrophage recruitment at a source c depends on TNF- α , CCL2 and CCL5 in the following way:

$$\alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL2}} \cdot c[\text{CCL2}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] \geq \tau_{M,\text{recr}} \quad (2.22)$$

where α_{TNF} , α_{CCL2} and α_{CCL5} are configurable parameters. So the source c can recruit a macrophage when the value of the linear combination is above the threshold parameter $\tau_{M,\text{recr}}$. In addition, there is also a probability of $p_{M,\text{recr}}$ that should be satisfied. If both conditions are met, a new macrophage is recruited at c . Recall that a macrophage lives for at most 100 days [44]. We model this by assuming that all macrophages live 100 days, and we introduce some variation by setting the birth time of a macrophage, uniformly at random, to a value in the interval $[t - 14400, t]$ where t is the current time step (note that 100 days corresponds to 14,400 time steps). The pseudo code corresponding to this procedure is given in Algorithm 17.

Recruitment of T cells may take place at a non-caseated source c with a probability of $p_{T,\text{recr}}$, if on that source c at most one immune cell is present and one of the following conditions is satisfied: for T_γ cells

$$\alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL2}} \cdot c[\text{CCL2}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] + \alpha_{\text{CXCL9}} \cdot c[\text{CXCL9}] \geq \tau_{\gamma,\text{recr}}; \quad (2.23)$$

Algorithm 17: RECRUITMAC(c, \mathcal{G}, t)

Input: c is a non-caseated source not containing a macrophage and t is the current time step

- 1 Let n_{macs} be the number of macrophages on \mathcal{G}
 - 2 $b \leftarrow \alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL2}} \cdot c[\text{CCL2}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] \geq \tau_{\text{M,recr}}$
 - 3 **if** (b **and** with probability $p_{\text{M,recr}}$) **or** $n_{\text{macs}} < N_{\text{M}}$ **then**
 - 4 Let M be a new macrophage on micro-compartment c
 - 5 Set $M[t_{\text{birth}}]$ uniformly at random to a value in $[t - 14400, t]$
 - 6 $M[t_{\text{death}}] \leftarrow M[t_{\text{birth}}] + 14400$
 - 7 $M[\text{state}] \leftarrow \text{Resting}$
 - 8 $M[\text{deact}] \leftarrow \text{false}$
 - 9 $M[\text{NF}\kappa\text{B}] \leftarrow \text{false}$
 - 10 $M[\text{STAT1}] \leftarrow \text{false}$
 - 11 $M[B_I] \leftarrow 0$
-

for T_c cells

$$\alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] + \alpha_{\text{CXCL9}} \cdot c[\text{CXCL9}] \geq \tau_{c,\text{recr}}; \quad (2.24)$$

and for T_{reg} cells

$$\alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] \geq \tau_{\text{reg,recr}}. \quad (2.25)$$

In addition there are also probabilities associated with the recruitment of T cells: we have $p_{c,\text{recr}}$, $p_{\gamma,\text{recr}}$ and $p_{\text{reg,recr}}$ for T_c , T_γ and T_{reg} cells, respectively. We require that $p_{c,\text{recr}} + p_{\gamma,\text{recr}} + p_{\text{reg,recr}} = 1$. The maximal lifespan of a T cell is 3 days [38], which corresponds to 432 time steps. Similarly to macrophages, this is enforced by setting the birth time to at most 3 days in the past. The pseudo code of RECRUITTCELL is given in Algorithm 18.

In summary, the cytokines and chemokines that play a role in the recruitment of immune cells are given in Table 2.1b.

Movement

The speed by which macrophages move depends on their state. For instance, chronically infected macrophages do not move at all. Whereas resting, infected and active macrophages move every $v_{\text{M,r}}$, $v_{\text{M,i}}$, $v_{\text{M,a}}$ time steps, respectively (see Algorithm 19). In Algorithm 20, we can see that T cells move every time step.

Recall that the Moore neighborhood of every micro-compartment consists of nine compartments. An immune cell has the ability to move to every micro-compartment of its Moore neighborhood, provided that micro-compartment is not caseated nor occupied. Due to chemotaxis, there is a bias toward moving to micro-compartment with higher levels of chemokines. In Table 2.1a, the chemokines that influence the movement of the various cell types are shown.

Whenever an immune cells wants to move, we sum for every micro-compartment in its Moore neighborhood the levels of the involved chemokines. In case the immune cell is a macrophage, we multiply the highest chemokine level by 1.5 — in other words, there is an even stronger bias for macrophages to move to micro-compartment with the highest chemokine levels. Subsequently, the sums are normalized so that each micro-compartment has a probability of moving onto it.

Algorithm 18: RECRUITCELL(c, \mathcal{G}, t)

Input: c is a non-casated source not containing a macrophage and t is the current time step

```

1  $b_\gamma \leftarrow \alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL2}} \cdot c[\text{CCL2}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] + \alpha_{\text{CXCL9}} \cdot c[\text{CXCL9}] \geq \tau_{\gamma, \text{recr}}$ 
2  $b_c \leftarrow \alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] + \alpha_{\text{CXCL9}} \cdot c[\text{CXCL9}] \geq \tau_{c, \text{recr}}$ 
3  $b_{\text{reg}} \leftarrow \alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] \geq \tau_{\text{reg}, \text{recr}}$ 
4 if with probability  $p_{\text{T,recr}}$  then
5   Let  $p$  be a random real number in the range  $[0, 1]$ 
6   if  $b_\gamma$  and  $p \in [0, p_{\gamma, \text{recr}}]$  then
7     Let  $T_\gamma$  be a  $T_\gamma$  cell on micro-compartment  $c$ 
8     Set  $T_\gamma[t_{\text{birth}}]$  uniformly at random to a value in  $[t - 432, t]$ 
9      $T_\gamma[t_{\text{death}}] \leftarrow T_\gamma[t_{\text{birth}}] + 432$ 
10     $T_\gamma[\text{deact}] \leftarrow \text{false}$ 
11  else if  $b_c$  and  $p \in [p_{\gamma, \text{recr}}, p_{\gamma, \text{recr}} + p_{c, \text{recr}}]$  then
12    Let  $T_c$  be a  $T_c$  cell on micro-compartment  $c$ 
13    Set  $T_c[t_{\text{birth}}]$  uniformly at random to a value in  $[t - 432, t]$ 
14     $T_c[t_{\text{death}}] \leftarrow T_c[t_{\text{birth}}] + 432$ 
15     $T_c[\text{deact}] \leftarrow \text{false}$ 
16  else if  $b_{\text{reg}}$  and  $p \in [p_{\gamma, \text{recr}} + p_{c, \text{recr}}, p_{\gamma, \text{recr}} + p_{c, \text{recr}} + p_{\text{reg}, \text{recr}}]$  then
17    Let  $T_{\text{reg}}$  be a  $T_{\text{reg}}$  cell on micro-compartment  $c$ 
18    Set  $T_{\text{reg}}[t_{\text{birth}}]$  uniformly at random to a value in  $[t - 432, t]$ 

```

	CCL2	CCL5	CXCL9/10/11
Mac	x	x	
T_γ	x	x	x
T_c		x	x
T_{reg}		x	

(a) Chemokines that play a role in the movement of the immune cells.

	TNF	CCL2	CCL5	CXCL9/10/11
Mac	x	x	x	
T_γ	x	x	x	x
T_c	x	x	x	x
T_{reg}	x		x	

(b) Cytokines/chemokines that play a role in the recruitment of the immune cells.

Table 2.1: Role of TNF- α and chemokines in chemotaxis and recruitment of immune cells.

Algorithm 19: MOVEMACROPHAGES(\mathcal{G}, t)

Input: \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $\mathcal{L}_M$  be a list containing all the macrophages
2 foreach  $M \in \mathcal{L}_M$  do
3   if  $M[\text{state}] = \text{Resting}$  and  $t \bmod v_{M,r} = 0$  then
4     MOVEIMMUNECELL( $M, \mathcal{G}, \text{true}, \text{false}, \text{true}, 1.5$ )
5   if  $M[\text{state}] = \text{Active}$  and  $t \bmod v_{M,a} = 0$  then
6     MOVEIMMUNECELL( $M, \mathcal{G}, \text{true}, \text{false}, \text{true}, 1.5$ )
7   if  $M[\text{state}] = \text{Infected}$  and  $t \bmod v_{M,i} = 0$  then
8     MOVEIMMUNECELL( $M, \mathcal{G}, \text{true}, \text{false}, \text{true}, 1.5$ )

```

Algorithm 20: MOVE_TCELLS(\mathcal{G}, t)**Input:** \mathcal{G} is a uniform grid and t is the current time step

- 1 Let \mathcal{L}_γ , \mathcal{L}_c and \mathcal{L}_{reg} be lists containing all T_γ cells, T_c cells and T_{reg} cells, respectively
- foreach** $T_\gamma \in \mathcal{L}_\gamma$ **do**
- 2 MOVE_IMMUNE_CELL($T_\gamma, \mathcal{G}, \text{true}, \text{true}, \text{true}, 1$)
- 3 **foreach** $T_c \in \mathcal{L}_c$ **do**
- 4 MOVE_IMMUNE_CELL($T_c, \mathcal{G}, \text{false}, \text{true}, \text{true}, 1$)
- 5 **foreach** $T_{\text{reg}} \in \mathcal{L}_{\text{reg}}$ **do**
- 6 MOVE_IMMUNE_CELL($T_{\text{reg}}, \mathcal{G}, \text{false}, \text{true}, \text{false}, 1$)

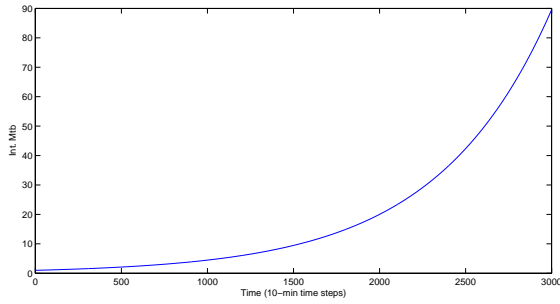
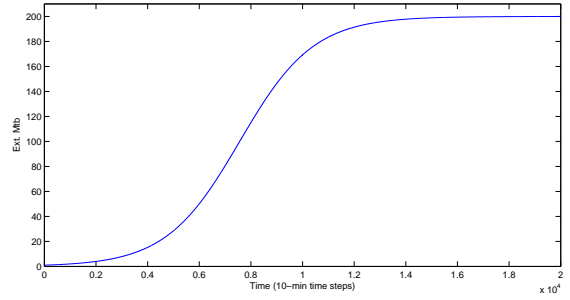
(a) Growth of intracellular bacteria with $B_I(0) = 1$ and $\alpha_{\text{Bi}} = 0.0015$.(b) Growth of extracellular bacteria with $B_E(0) = 1$, $\alpha_{\text{Be}} = 0.0007$ and $K_{\text{Be}} = 200$.

Figure 2.3: Growth of intracellular and extracellular bacteria

Using the probabilities a micro-compartment c is picked. If that micro-compartment is caseous or cannot accommodate the immune cell, the immune cell does not move. Otherwise, in case the immune cell is a macrophage it moves to c . In case the immune cell is a T cell and c is not empty, then there are additional probabilities to be met: if c contains a macrophage, the probability of moving to c is $p_{\text{T},\text{M}}$; should c contain a T cell then the probability of moving to it is $p_{\text{T},\text{T}}$.

2.1.4 Mtb specific rules

Mtb prefer the intracellular environment of macrophage over the extracellular environment. It is for this reason that we make use of two different growth functions. The growth function used for intracellular bacteria is:

$$B_I(t + \Delta t) = B_I(t) + \alpha_{\text{Bi}} * B_I(t) \quad (2.26)$$

where α_{Bi} is a parameter denoting the growth rate of intracellular Mtb (see Figure 2.3a and Algorithm 22).

Growth of extracellular Mtb is bounded by the dimensions of a micro-compartment. Assuming that a micro-compartment can sustain at most K_{Be} bacteria, the following logistic growth function can be used to model growth of extracellular Mtb:

$$B_E(t + \Delta t) = B_E(t) + \alpha_{\text{Be}} \cdot B_E(t) \cdot \left(1 - \frac{B_E(t)}{K_{\text{Be}}}\right) \quad (2.27)$$

Algorithm 21: MOVEIMMUNECELL($ic, \mathcal{G}, b_{\text{CCL2}}, b_{\text{CCL5}}, b_{\text{CXCL9}}, \alpha$)

Input: ic is an immune cell on the uniform grid \mathcal{G} ; $b_{\text{CCL2}}, b_{\text{CCL5}}, b_{\text{CXCL9}}$ are flags denoting the involved chemokines and α is a bonus coefficient

```

1 Let  $\mathcal{G}_{kl}$  be the micro-compartment on which  $ic$  resides
2 for  $i \leftarrow -1$  to 1 do
3   for  $j \leftarrow -1$  to 1 do
4     /* Note that we initialize  $p_{ij}$  to a very small value  $\epsilon$ , in order to
5      prevent a possible divide by zero later during normalization. In the
6      actual implementation the toroidicity of the grid needs to be taken
7      into account. */
8      $p_{ij} \leftarrow \epsilon$ 
9     if  $b_{\text{CCL2}}$  then  $p_{ij} \leftarrow p_{ij} + \mathcal{G}_{k+i, l+j}[\text{CCL2}]$ 
10    if  $b_{\text{CCL5}}$  then  $p_{ij} \leftarrow p_{ij} + \mathcal{G}_{k+i, l+j}[\text{CCL5}]$ 
11    if  $b_{\text{CXCL9}}$  then  $p_{ij} \leftarrow p_{ij} + \mathcal{G}_{k+i, l+j}[\text{CXCL9}]$ 
12 Multiply the highest value in  $p$  by  $\alpha$ 
13 Normalize  $p$ 
14 Let  $c$  be the micro-compartment picked according to the probabilities in  $p$ 
15 if  $c$  is not caseous and has room for  $ic$  then
16   /* Move with probability  $p_{\text{T,M}}$ , if  $c$  contains a macrophage */
17   /* Move with probability  $p_{\text{T,T}}$ , if  $c$  contains a T cell */
18   if  $ic$  is a T cell and  $c$  contains a macrophage then
19     With probability  $p_{\text{T,M}}$ , move  $ic$  onto  $c$ 
20   else if  $ic$  is a T cell and  $c$  contains a T cell then
21     With probability  $p_{\text{T,T}}$ , move  $ic$  onto  $c$ 
22   else
23     Move  $ic$  onto  $c$ 

```

Algorithm 22: GROWINTMTB(M)

Input: M is a macrophage (or a microglial cell)

```

1  $M[B_I] \leftarrow M[B_I] + \alpha_{\text{Bi}} \cdot M[B_I]$ 

```

where α_{Be} is a parameter denoting the growth rate of extracellular Mtb (see Figure 2.3b and Algorithm 23).

Algorithm 23: GROWEXTMTB(\mathcal{G})

Input: \mathcal{G} is a uniform grid of 100×100 micro-compartments

```

1 for  $i \leftarrow 1$  to 100 do
2   for  $j \leftarrow 1$  to 100 do
3     Let  $c$  be the micro-compartment at  $\mathcal{G}_{ij}$ 
4      $c[B_E] \leftarrow c[B_E] + \alpha_{Be} \cdot c[B_E] \cdot \left(1 - \frac{c[B_E]}{K_{Be}}\right)$ 

```

2.1.5 Extensions and model parameters

In the previous subsections we described the agent-based model of the immune response to Mtb in the lung. As mentioned earlier, this model was first described by Segovia-Juarez et al. [36] and later extend by Ray et al. [33] The differences and extensions to Ray's model are:

- T_γ cells used to have two different active states. These two states have been merged, as they were not significantly different and introduced unnecessary complexity to the model.
- In order to have more down-regulatory mechanisms in the model, not only T_γ cells can be down-regulated by T_{reg} cells but also T_c cells and macrophages.

When T_c cells are down-regulated by T_{reg} cells, they lose their cytotoxic capabilities for a fixed period of time.

If a macrophage is down-regulated by a T_{reg} cell then STAT1 is deactivated and the macrophage in question does nothing but moving around and secreting for a fixed period of time.

- In Ray's model, once NF κ B was turned on it always remained on. Now we check every time step whether the conditions pertaining to NF κ B activation are still satisfied; if that is not the case, we disable NF κ B.
- The extracellular bacteria condition of NF κ B activation has been extended to consider the Moore neighborhood, rather than the local micro-compartment.
- Resting macrophages that have NF κ B enabled, secrete at half rate instead of maximal rate.
- In order to reduce unnecessary complexity, the condition for enabling STAT1 is now the same for both resting and infected macrophages.
- Infected macrophages now have the ability to phagocytose additional extracellular bacteria; the probability of doing this is negatively proportional to the number of intracellular bacteria.
- Due to an error in Ray's model implementation, diffusion was computed for only a couple of 6-second time steps. Fixing this error resulted in a very slow model. For that reason we make use of a different method for solving the diffusion equation (see Section A.2 for more details).

To conclude, we list in Table C.2 all parameters together with their descriptions.

Table 2.2: Lung model parameters

Parameter	Description
N_{caseous}	Number of killings for a micro-compartment to become caseous
$\tau_{\text{TNF,apopt}}$	TNF- α level required for TNF- α -induced apoptosis
$p_{\text{TNF,apopt}}$	Probability by which TNF- α -induced apoptosis occurs
$\tau_{\text{M,TNF,NF}\kappa\text{B}}$	TNF- α level required for NF κ B activation in macrophages
$N_{\text{M,Be,NF}\kappa\text{B}}$	Number of extracellular bacteria in the Moore neighborhood required for NF κ B activation in macrophages
$\sigma_{\text{M,deact}}$	Time span during which a macrophage remains deactivated
$\sigma_{\text{c,deact}}$	Time span during which a T_c cell remains deactivated
$\sigma_{\gamma,\text{deact}}$	Time span during which a T_γ cell remains deactivated
$p_{\text{M,STAT1}}$	Probability of a T_γ cell to enable STAT1 in a macrophage
$p_{\text{M,kill}}$	Probability of a resting macrophage to kill an extracellular bacterium
$N_{\text{M,Bi,CI}}$	Number of intracellular bacteria for an infected macrophage to become chronically infected
$N_{\text{M,Bi,burst}}$	Number of intracellular bacteria for a chronically infected macrophage to burst
$N_{\text{M,Be,kill}}$	Number of extracellular bacteria that an active macrophage can kill every time step
$p_{\text{c,kill}}$	Probability of a T_c cell to kill a (chronically) infected macrophage
$p_{\text{c,kill,cleanly}}$	Probability of a T_c cell to kill a chronically infected macrophage cleanly
$p_{\text{Fas,apopt}}$	Probability of a T_γ cell to induce apoptosis via the Fas/FasL pathway
s_{TNF}	Maximal number of TNF- α molecules that a macrophage can secrete every 6-second time step
s_{CCL2}	Maximal number of CCL2 molecules that a macrophage can secrete every 6-second time step
s_{CCL5}	Maximal number of CCL5 molecules that a macrophage can secrete every 6-second time step
s_{CXCL9}	Maximal number of CXCL9/10/11 molecules that a macrophage can secrete every 6-second time step
D_{TNF}	Diffusion coefficient of TNF- α molecules
D_{CC}	Diffusion coefficient of chemokine molecules
δ_{TNF}	Degradation rate of TNF- α
δ_{CC}	Degradation rate of the chemokines
t_{recr}	Time after which T cell recruitment is enabled
N_{M}	Minimal number of macrophages on the grid
α_{TNF}	Effect of TNF- α on recruitment of immune cells
α_{CCL2}	Effect of CCL2 on recruitment of immune cells
α_{CCL5}	Effect of CCL5 on recruitment of immune cells
α_{CXCL9}	Effect of CXCL9/10/11 on recruitment of immune cells
$\tau_{\text{M,recr}}$	Recruitment threshold for macrophage recruitment
$\tau_{\text{c,recr}}$	Recruitment threshold for recruitment of T_c cells
$\tau_{\gamma,\text{recr}}$	Recruitment threshold for recruitment of T_γ cells
$\tau_{\text{reg,recr}}$	Recruitment threshold for recruitment of T_{reg} cells
$p_{\text{M,recr}}$	Probability of recruiting a macrophage

Continued on next page

Table 2.2 – continued from previous page

Parameter	Description
$p_{T,recr}$	Probability of recruiting a T cell
$p_{c,recr}$	Probability of recruiting a T_c cell
$p_{\gamma,recr}$	Probability of recruiting a T_γ cell
$p_{reg,recr}$	Probability of recruiting a T_{reg} cell
$v_{M,r}$	Time required for a resting macrophage to move one micro-compartment
$v_{M,a}$	Time required for an active macrophage to move one micro-compartment
$v_{M,i}$	Time required for an infected macrophage to move one micro-compartment
$p_{T,M}$	Probability of moving a T cell onto a micro-compartment containing a macrophage
$p_{T,T}$	Probability of moving a T cell onto a micro-compartment already containing a T cell
α_{Bi}	Growth rate of intracellular bacteria
α_{Be}	Growth rate of extracellular bacteria
K_{Be}	Number of extracellular bacteria a micro-compartment can accommodate

2.2 CNS model

In this section we introduce a new agent-based model describing the immune response and pathogenesis of TBM. In Subsection 2.2.1 we give a top-level overview of the model. Rules specific to immune cells are discussed in Subsection 2.2.2, followed by a discussion of the cytokine/chemokine related rules in Subsection 2.2.3 and Mtb specific rules in Subsection 2.2.4. This section is concluded by listing all parameters in Subsection 2.2.5.

2.2.1 Model overview

This time the environment that we study is a cross-section of $4\text{ mm} \times 4\text{ mm}$ of the cerebral cortex of the brain. Similarly to the lung model, the size of a micro-compartment is chosen to be the same as that of a macrophage, namely $20\ \mu\text{m} \times 20\ \mu\text{m}$. So the uniform grid that represents the environment is comprised by 200×200 micro-compartments. As we saw in Subsection 1.1.2, TBM results when the meninges are infected with Mtb. Since we want to capture the pathogenesis of TBM, we need to include meninges in the environment. We do this by defining one edge of the grid to correspond to meninges, whereas the other three edges correspond to brain tissue. Since the edges correspond to different types of tissue, we cannot make the grid toroidal. Instead, as soon as an immune cells moves out of the grid, we stop tracking it. So the Moore neighborhood of a micro-compartment does not always consist of nine micro-compartments.

The cerebral cortex is highly vascularized with a vascular volume of roughly 2% of the total volume [10]. Taking this number into account, we arrive at a number of 800 sources out 40,000 micro-compartments. In addition to considering the immune cells of the lung model (macrophages, T_γ , T_c and T_{reg} cells), we also consider a type of immune cell specific to the CNS: the microglial cell. Microglia are the resident macrophages of the brain and spinal cord. As such they have phagocytic abilities, and can thus ingest extracellular bacteria. Unlike the

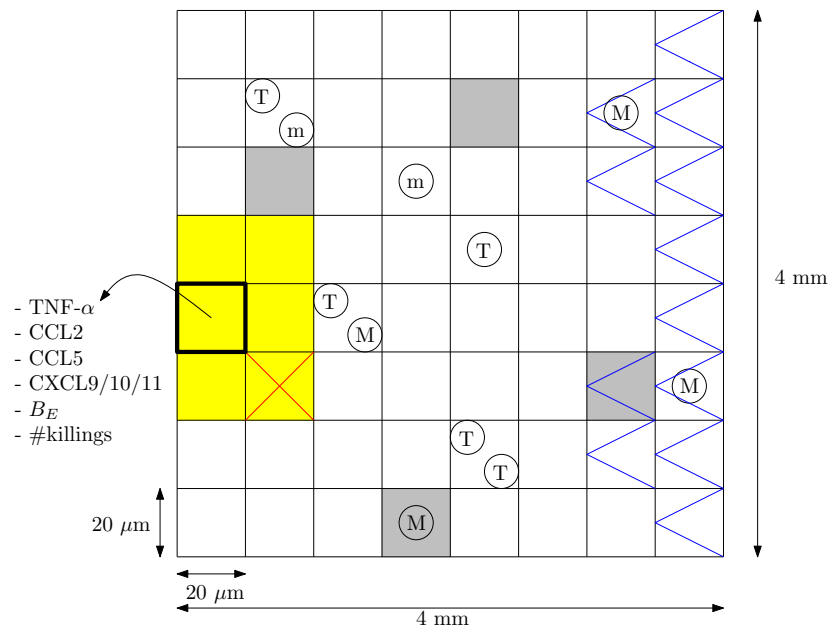


Figure 2.4: There are 40,000 micro-compartments on the grid, some of which are sources (denoted by a gray background), others are part of the meninges (on the right edge). Micro-compartments can also be caseated (denoted by a red cross). There is room for two immune cells in a micro-compartment; only one of which can be a macrophage (denoted by ‘M’) or a microglial cell (denoted by ‘m’), both can be accompanied by a T cell (denoted by ‘T’). Several continuous attributes are associated with every micro-compartment. The Moore neighborhood of the micro-compartment with fat borders is colored in yellow.

other modeled immune cells, microglia are not recruited through vascular sources. Rather they reside and proliferate on site.

There is a steady state population of meningeal macrophages in the brain that play a protective role during bacterial meningitis [32]. To include this in our model, we need to ensure that meningeal macrophages remain on the meninges, and that the population of such macrophages is in equilibrium. We address this during movement and recruitment of macrophages.

Just like the lung model, there can be at most two immune cells present in a micro-compartment. Should one of those be a macrophage or a microglial cell, then it can only be accompanied by a T cell. The same cytokines/chemokines of the lung model are also present in the CNS model. So for every micro-compartment we keep track of the levels of the cytokine TNF- α and the chemokines CCL2, CCL5 and CXCL9/10/11. Micro-compartments can become caseated over time, at which point they become inaccessible for immune cells. A micro-compartment can also be a source or part of the meninges. We denote the attributes of a micro-compartment c by:

- $c[\text{TNF}]$ is the number of TNF- α molecules in c ;
- $c[\text{CCL2}]$ is the number of CCL2 molecules in c ;
- $c[\text{CCL5}]$ is the number of CCL5 molecules in c ;
- $c[\text{CXCL9}]$ is the number of CXCL9/10/11 molecules in c ;
- $c[B_E]$ is the number of extracellular bacteria in c ;
- $c[\text{killings}]$ is the number of killings contributing to caseation that have occurred in c ;
- $c[\text{caseous}]$ is a flag that is set when c is caseated;
- $c[\text{source}]$ is a flag that is set when c is a source;
- $c[\text{meninges}]$ is a flag that is set when c is part of the meninges.

In Figure 2.4 an example grid illustrating the described concepts is given. The model is initialized by invoking the function INITIALIZE (see Algorithm 25). The first step of this function consists of initializing the micro-compartments that comprise the meninges. For every row of the grid, the width of the meninges at that row is chosen to be between 10 and 20 micro-compartments — corresponding to a width of 0.2–0.4 mm. The next step in the initialization is to put the initial inoculum — consisting of infected macrophages, infected microglia and extracellular bacteria — on the grid. This is followed by placing populations of $N_{\text{m,ram}}$ ramified inactive microglia and N_{M} resting meningeal macrophages on the grid. The latter are placed at random on micro-compartments that are part of the meninges. The final step of the initialization is to place 800 sources on the grid. Similarly to the lung model, this is done by subdividing the grid into $28 \times 28 = 784$ equally-sized clusters of micro-compartments; in each of these clusters a micro-compartment is picked at random and flagged to be a source. The remaining 16 sources are assigned in a random fashion by picking them from the micro-compartments that are not sources.

The ABM is run for 200 days by executing 28,800 10-minute time steps. The first step of the main loop concerns the secretion, diffusion and degradation of chemokines and cytokines. This happens at a finer timescale of 6-second time steps. Subsequently, all microglia are moved followed by moving macrophages and then T cells. After that microglial cells are proliferated

and new immune cells are recruited. Then the immune cell specific rules are performed, and the growth function of extracellular Mtb is applied. The pseudo code of the main loop of the model is given in Algorithm 24.

Algorithm 24: BRAINABM(\mathcal{G}, I)

Input: \mathcal{G} is a uniform grid and I is the initial inoculum

```

1 INITIALIZE( $\mathcal{G}, I$ )
2 for  $t \leftarrow 1$  to 28800 do
3   for  $t' \leftarrow 1$  to 100 do
4     MACSECRETE( $\mathcal{G}$ )
5     MGSECRETE( $\mathcal{G}$ )
6     DIFFUSE( $\mathcal{G}$ )
7     DEGRADE( $\mathcal{G}$ )
8     MOVEMICROGLIA( $\mathcal{G}, t$ )
9     MOVEMACROPHAGES( $\mathcal{G}, t$ )
10    MOVETCELLS( $\mathcal{G}$ )
11    PROLIFERATEMICROGLIA( $\mathcal{G}$ )
12    RECRUIT( $\mathcal{G}, t$ )
13    UPDATESTATES( $\mathcal{G}, t$ )
14    GROWEXTMTB( $\mathcal{G}$ )

```

2.2.2 Immune cell specific rules

Immune cell specific rules are applied in the function UPDATESTATES invoked in Algorithm 24. In this function we update the states of the immune cells on the grid, starting by macrophages followed by microglia, T_γ , T_c and finally T_{reg} cells. An immune cell is immediately removed from the grid, should it die during a state transition. Just like with the lung model, state changes occur in a two-phase fashion (see Algorithm 26).

The rules concerning state changes of macrophages, T_γ and T_{reg} cells are exactly the same as those described in Subsection 2.1.2. In the remainder of this subsection we will focus on the rules pertaining to T_c cells and microglia.

T_c cell

Besides being able to kill infected macrophages, a T_c cells has the ability to kill infected microglia as well. Specifically, infected and chronically infected microglia are targeted by T_c cells. The killing proceeds in the same way: with a probability of $p_{c,\text{kill}}$ an infected or chronically infected microglial cell is killed. In case the microglial cell is infected, it is killed in such a way that all intracellular Mtb are dispersed over the Moore neighborhood, unless a probability of $p_{c,\text{kill, cleanly}}$ is met in which case all intracellular Mtb are killed. Cytotoxic killing of microglia contributes to caseation. In Algorithm 27 the pseudo code of the updated UPDATETCYTSTATE function is given.

Algorithm 25: INITIALIZE(\mathcal{G}, I)

Input: \mathcal{G} is a uniform grid and I is the initial inoculum

```

/* Initialize meninges */
1 for  $i \leftarrow 1$  to 200 do
2   Let  $r$  be an integer, chosen uniformly at random, in the range  $[10, 20]$ 
3   for  $j \leftarrow 200 - r$  to 200 do
4     Let  $c$  be the micro-compartment at  $\mathcal{G}_{ij}$ 
5      $c[\text{meninges}] \leftarrow \text{true}$ 
/* Put inoculum and initial populations of macrophages and microglia on the
   grid */
6 Place initial inoculum  $I$  on  $\mathcal{G}$ 
7 Place  $N_M$  resting macrophages on random meningeal micro-compartment
8 Place  $N_{m,ram}$  ramified microglia on random micro-compartment of the grid
/* Initialize sources */
9 Subdivide  $\mathcal{G}$  in  $28 \times 28 = 784$  equally-sized clusters of neighboring micro-compartment
10 In every such cluster, pick a micro-compartment  $c$  and set  $c[\text{source}] \leftarrow \text{true}$ 
11 Pick, uniformly at random, 16 micro-compartment that are not sources and for each such
    micro-compartment  $c$  set  $c[\text{source}] \leftarrow \text{true}$ 

```

Algorithm 26: UPDATESTATES(\mathcal{G}, t)

Input: \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $\mathcal{L}_M, \mathcal{L}_m, \mathcal{L}_\gamma, \mathcal{L}_c$  and  $\mathcal{L}_{reg}$  be lists containing all macrophages, microglia,  $T_\gamma$  cells,  $T_c$ 
   cells and  $T_{reg}$  cells, respectively
2 foreach  $M \in \mathcal{L}_M$  do
3   UPDATEMACSTATE( $M, \mathcal{G}, t$ )
4   if  $M[\text{dead}]$  then Remove  $M$  from  $\mathcal{G}$ 
5 foreach  $m \in \mathcal{L}_m$  do
6   UPDATEMGSTATE( $m, \mathcal{G}, t$ )
7   if  $m[\text{dead}]$  then Remove  $m$  from  $\mathcal{G}$ 
8 foreach  $T_\gamma \in \mathcal{L}_\gamma$  do
9   UPDATETGAMSTATE( $T_\gamma, \mathcal{G}, t$ )
10  if  $T_\gamma[\text{dead}]$  then Remove  $T_\gamma$  from  $\mathcal{G}$ 
11 foreach  $T_c \in \mathcal{L}_c$  do
12  UPDATETCYTSTATE( $T_c, \mathcal{G}, t$ )
13  if  $T_c[\text{dead}]$  then Remove  $T_c$  from  $\mathcal{G}$ 
14 foreach  $T_{reg} \in \mathcal{L}_{reg}$  do
15  UPDATETREGSTATE( $T_{reg}, \mathcal{G}, t$ )
16  if  $T_{reg}[\text{dead}]$  then Remove  $T_{reg}$  from  $\mathcal{G}$ 
17 Finalize all state transitions, by updating the current states of all cells

```

Algorithm 27: UPDATE_TCYT_STATE(T_c, \mathcal{G}, t)

Input: T_c is T_c cell, \mathcal{G} is a uniform grid and t is the current time step

- 1 Let c denote the micro-compartment in which T_c resides
- 2 **if** $t = T_c[t_{\text{death}}]$ **then**
- 3 $T_c[\text{dead}] \leftarrow \text{true}$
- 4 **else if** $c[\text{TNF}] > \tau_{\text{TNF,apopt}}$ **and** with probability $p_{\text{TNF,apopt}}$ **then**
- 5 $T_c[\text{dead}] \leftarrow \text{true}$
- 6 **else**
- 7 **if** $T_c[\text{deact}]$ **and** $T_c[t_{\text{deact}}] + \sigma_{c,\text{deact}} = t$ **then**
- 8 $T_c[\text{deact}] \leftarrow \text{false}$
- 9 **if** $T_c[\text{deact}] = \text{false}$
- 10 **and** c has an infected or chronically infected macrophage or microglial cell
- 11 **and** with probability $p_{c,\text{kill}}$ **then**
- 12 Let m be the macrophage or microglial cell in c
- 13 **if** $m[\text{state}] = \text{Chronically infected}$ **and** with probability $(1 - p_{c,\text{kill,clearly}})$ **then**
- 14 Disperse all intracellular bacteria over the Moore neighborhood of c
- 15 $m[\text{dead}] \leftarrow \text{true}$
- 16 CONTRIBUTE_TO_CASEATION(c)

Microglia

Microglia are phagocytic cells, and thus have the ability to ingest extracellular Mtb. We denote the number of intracellular bacteria in a microglial cell by B_I . In the literature three states are distinguished in microglia: amoeboid, ramified and reactivated [31, 35]. Amoeboid microglia are only present during fetal brain development. Therefore we do not consider them in the model. In the mature brain, ramified microglial cells are found. Ramified microglia have a small cell body (5–10 μm) and thin branching dendrites; in this state they are stationary and do not participate in the immune response. Ramified microglia can regain their phagocytic abilities by becoming reactive following inflammation or infection [2]. In our model we only consider ramified and reactivated microglia. The latter are further classified in the following four states: resting, infected, chronically infected or activated.

In a reactivated microglial cell, the two transcription factors $\text{NF}\kappa\text{B}$ and STAT1 can be enabled. For $\text{TNF-}\alpha$ -induced apoptosis to occur the same pre-conditions as in the lung model have to be met: the level of $\text{TNF-}\alpha$ in the local micro-compartment has to be above $\tau_{\text{TNF,apopt}}$ and a probability of $p_{\text{TNF,apopt}}$ should be satisfied. An apoptosed microglial cell releases half of its intracellular bacteria to its Moore neighborhood; the other half is killed. The second cause of microglial death is due to aging. In the model microglia live — just like macrophages — at most 100 days. Death of an activated microglial cell contributes to caseation.

Microglia can be down-regulated by T_{reg} cells, in which case they stop performing their state-specific rules for a period of $\sigma_{m,\text{deact}}$ time steps. Microglia that are not down-regulated perform their state-specific rules after updating the $\text{NF}\kappa\text{B}$ property. $\text{NF}\kappa\text{B}$ is turned on if

- the microglial cell is infected or active, or
- the number of $\text{TNF-}\alpha$ molecules in the micro-compartment of the microglial cell is more than $\tau_{m,\text{TNF,NF}\kappa\text{B}}$, OR

- the number of extracellular bacteria in the Moore neighborhood of the microglial cell is more than $N_{m,Be,NF\kappa B}$.

If none of the above conditions are met, the NF κ B property is disabled. In Algorithm 28 the pseudo code corresponding to this description is given.

Algorithm 28: UPDATEMGSTATE(m, \mathcal{G}, t)

Input: m is a microglial cell, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $m$  resides
2 if  $t = m[t_{\text{death}}]$  then
3    $c[B_E] \leftarrow c[B_E] + m[B_I]$ 
4   if  $m[\text{state}] = \text{Active}$  then
5     CONTRIBUTETOCASEATION( $c$ )
6    $m[\text{dead}] \leftarrow \text{true}$ 
7 else if  $c[\text{TNF}] > \tau_{\text{TNF,apopt}}$  and with probability  $p_{\text{TNF,apopt}}$  then
8   Kill half of the intracellular bacteria of  $m$ 
9   Disperse the other half over the Moore neighborhood of  $c$ 
10   $m[\text{dead}] \leftarrow \text{true}$ 
11 else
12  if  $m[\text{deact}]$  and  $m[t_{\text{deact}}] + \sigma_{m,\text{deact}} = t$  then
13     $m[\text{deact}] \leftarrow \text{false}$ 
14  if  $m[\text{deact}] = \text{false}$  then
15    Set  $m[\text{NF}\kappa\text{B}]$  to true if the state is infected or active, or the number of extracellular
    bacteria in the Moore neighborhood is larger than  $N_{m,Be,NF\kappa B}$ . Otherwise set
     $m[\text{NF}\kappa\text{B}]$  to false.
16  switch  $m[\text{state}]$  do
17    case Ramified HANDLEMGRAMIFIED( $m, \mathcal{G}, t$ )
18    case Resting HANDLEMGRESTING( $m, \mathcal{G}, t$ )
19    case Infected HANDLEMGINFECTED( $m, \mathcal{G}, t$ )
20    case Chronically infected HANDLEMGCHRONICALLYINFECTED( $m, \mathcal{G}, t$ )
21    case Active HANDLEMGACTIVE( $m, \mathcal{G}, t$ )

```

A ramified microglial cell is stationary and does not actively participate in the immune response. It can become reactivated in the following two ways. If the TNF- α level is above $\tau_{m,\text{TNF,react}}$, the microglial cell in question becomes resting [18]. The other way involves extracellular bacteria in the Moore neighborhood [18]: if there is an extracellular bacterium in the Moore neighborhood, the ramified microglial cell becomes infected with that bacterium (see Algorithm 29).

A resting microglial cell has the ability to kill an extracellular bacterium. If there is only one bacterium (or a fraction of less than one) in the local micro-compartment of the microglial cell, the bacterium is killed. If there are several bacteria in the local micro-compartment then two things can happen: either with a probability of $p_{m,\text{kill}}$ one of the bacteria is killed by the microglial cell which afterwards will remain in the resting state, or the microglial cell becomes infected with one bacterium. The transcription factor STAT1 is enabled with a probability that is proportional to the number of non-down-regulated T_γ cells in the Moore neighborhood. In case both transcription factors — NF κ B and STAT1— are enabled, the microglial cells becomes activated (see Algorithm 30).

We can see in Algorithm 31 that the rules concerning infected microglia are the same as those

Algorithm 29: HANDLEMGRAMIFIED(m, \mathcal{G}, t)

Input: m is a ramified macrophage, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $m$  resides
2 if  $c[\text{TNF}] > \tau_{m, \text{TNF}, \text{react}}$  then
3    $m[\text{state}'] \leftarrow \text{Resting}$ 
4 else if there are extracellular Mtb in the Moore neighborhood of  $c$  then
5   Let  $c'$  be a micro-compartment in the Moore neighborhood of  $c$  containing at least one
   extracellular bacterium (i.e.  $c'[B_E] \geq 1$ )
6    $c'[B_E] \leftarrow c'[B_E] - 1$ 
7    $m[B_I] \leftarrow 1$ 
8    $m[\text{state}'] \leftarrow \text{Infected}$ 
9 else
10   $m[\text{state}'] \leftarrow \text{Ramified}$ 

```

Algorithm 30: HANDLEMGRESTING(m, \mathcal{G}, t)

Input: m is a resting microglial cell, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $m$  resides
2 if  $c[B_E] \leq 1$  then
3   /* Kill an extracellular bacterium if there is only 1 (or less) */
4    $c[B_E] \leftarrow 0$ 
5    $m[\text{state}'] \leftarrow \text{Resting}$ 
6 else if with probability  $p_{m, \text{kill}}$  then
7   /* Kill an extracellular bacterium and remain resting */
8    $c[B_E] \leftarrow c[B_E] - 1$ 
9    $m[\text{state}'] \leftarrow \text{Resting}$ 
10 else
11   /* Become infected with an extracellular bacterium */
12    $c[B_E] \leftarrow c[B_E] - 1$ 
13    $m[B_I] \leftarrow 1$ 
14    $m[\text{state}'] \leftarrow \text{Infected}$ 
15 Enable  $m[\text{STAT1}]$  with probability  $\text{NUMBEROFACTIVE TGAMINMOORE}(c) * p_{m, \text{STAT1}}$ 
16 if  $m[\text{STAT1}]$  and  $m[\text{NF}\kappa\text{B}]$  then
17    $m[B_I] \leftarrow 0$ 
18   /* Activated microglia have a shortened lifespan of 10 days */
19    $m[t_{\text{death}}] \leftarrow \min(m[t_{\text{death}}], t + 1440)$ 
20    $m[\text{state}'] \leftarrow \text{Active}$ 

```

Algorithm 31: HANDLEMGINFECTED(m, \mathcal{G}, t)

Input: m is an infected microglial cell, \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $c$  denote the micro-compartment in which  $m$  resides
2 GROWINTMTB( $m$ )
3  $p \leftarrow (N_{m, \text{Bi, CI}} - m[B_I])/100$ 
4 if  $c[B_E] > 0$  and with probability  $p$  then
    /* The less intracellular bacteria are present, the higher the probability
       for taking up more will be */
5  $\Delta B \leftarrow \min(c[B_E], 1)$ 
6  $c[B_E] \leftarrow c[B_E] - \Delta B$ 
7  $m[B_I] \leftarrow m[B_I] + \Delta B$ 
8 if  $m[B_I] \geq N_{m, \text{Bi, CI}}$  then
    /* Become chronically infected, if the number of intracellular bacteria
       exceeds  $N_{m, \text{Bi, CI}}$  */
9  $m[\text{state}'] \leftarrow \text{Chronically infected}$ 
10 else
11 Enable  $m[\text{STAT1}]$  with probability  $\text{NUMBEROFACTIVETGAMINMOORE}(c) * p_{m, \text{STAT1}}$ 
12 if  $m[\text{STAT1}]$  and  $m[\text{NF}\kappa\text{B}]$  then
13  $m[B_I] \leftarrow 0$ 
    /* Activated microglia have a shortened lifespan of 10 days */
14  $m[t_{\text{death}}] \leftarrow \min(m[t_{\text{death}}], t + 1440)$ 
15  $m[\text{state}'] \leftarrow \text{Active}$ 
16 else
17  $m[\text{state}'] \leftarrow \text{Infected}$ 

```

of infected macrophages in the lung model (cf. Subsection 2.1.2). Infected microglia have the ability to uptake more extracellular bacteria with a probability that is inversely proportional to the number of intracellular bacteria (see Figure 2.2). Should the number of intracellular bacteria exceed $N_{m, Bi, CI}$, the microglial cell in question becomes chronically infected. If this is not the case, there is still a chance for the microglial cell to become activated. This happens when both transcription factors NF κ B and STAT1 are enabled. The latter is enabled in the same way as in resting microglia and macrophages.

Algorithm 32: HANDLEMGCHRONICALLYINFECTED(m, \mathcal{G}, t)

Input: m is a chronically infected microglial cell, \mathcal{G} is a uniform grid and t is the current time step

- 1 Let c denote the micro-compartment in which m resides
 - 2 GROWINTMTB(m)
 - 3 **if** $m[B_I] \geq N_{m, Bi, burst}$ **then**
 - 4 Disperse all intracellular bacteria over the Moore neighborhood of c
 - 5 $m[dead] \leftarrow \mathbf{true}$
 - 6 CONTRIBUTETO CASEATION(c)
 - 7 **else**
 - 8 $m[state'] \leftarrow \mathbf{Chronically\ infected}$
-

In case a microglial cell is chronically infected, its intracellular bacteria will continue to reproduce until they grow beyond $N_{m, Bi, burst}$, at which point the microglial cell bursts and all intracellular bacteria are dispersed over the Moore neighborhood. The bursting of a chronically infected microglial cell contributes to caseation. In Algorithm 32 the pseudo code corresponding to this procedure is given.

Algorithm 33: HANDLEMGACTIVE(m, \mathcal{G}, t)

Input: m is an active microglial cell, \mathcal{G} is a uniform grid and t is the current time step

- 1 Let c denote the micro-compartment in which m resides
 - 2 $c[B_E] \leftarrow c[B_E] - \mathbf{min}(c[B_E], N_{m, Be, kill})$
 - 3 $m[state'] \leftarrow \mathbf{Active}$
-

Activated microglial cells excel at killing extracellular bacteria: every time step $N_{m, Be, kill}$ extracellular bacteria are killed in the local micro-compartment. The price that microglia pay for this increased phagocytic ability is that they have a shortened life span of at most 10 days (see line 14 of Algorithm 30 and line 13 of Algorithm 31). The pseudo code corresponding to the state transition of activated microglia is given in Algorithm 33.

2.2.3 Cytokine/chemokine specific rules

In this subsection we look at the rules concerning the chemokines CCL2, CCL5, CXCL9/10/11 and the cytokine TNF- α . Diffusion and degradation proceed in the same manner as in the lung model, and we refer to Subsection 2.1.3 for the details. In this subsection we will focus on secretion, proliferation of microglia, and recruitment and movement of immune cells.

Secretion

In addition to macrophages, also microglia have the ability to secrete the chemokines CCL2, CCL5, CXCL9/10/11 and the cytokine TNF- α [2, 31, 35]. For both types of phagocytes, secretion is dependent on the transcription factor NF κ B and the state of the phagocyte. For instance, ramified microglia do not secrete at all, whereas resting microglia that have NF κ B enabled secrete at half rate. In all other cases where NF κ B is enabled, secretion happens at maximal rate. Infected microglia secrete at half rate if NF κ B is disabled. The rates by which secretion occurs for TNF- α , CCL2, CCL5 and CXCL9 are denoted by s_{TNF} , s_{CCL2} , s_{CCL5} and s_{CXCL9} , respectively. The secretion rules concerning macrophages are the same as in the lung model (see Algorithm 13). In Algorithm 34 the pseudo code corresponding to secretion by microglia is given.

Algorithm 34: MGSECRETE(\mathcal{G})

Input: \mathcal{G} is a uniform grid

```

1 for  $i \leftarrow 1$  to 100 do
2   for  $j \leftarrow 1$  to 100 do
3     if  $\mathcal{G}_{ij}$  has a microglial cell then
4       Let  $c$  be the micro-compartment at  $\mathcal{G}_{ij}$ 
5       Let  $m$  be the microglial cell at  $c$ 
6       if  $m[\text{state}] \neq \text{Ramified}$  then
7         if  $m[\text{NF}\kappa\text{B}]$  and  $m[\text{state}] \neq \text{Resting}$  then
8            $c[\text{TNF}] \leftarrow c[\text{TNF}] + s_{\text{TNF}}$ 
9            $c[\text{CCL2}] \leftarrow c[\text{CCL2}] + s_{\text{CCL2}}$ 
10           $c[\text{CCL5}] \leftarrow c[\text{CCL5}] + s_{\text{CCL5}}$ 
11           $c[\text{CXCL9}] \leftarrow c[\text{CXCL9}] + s_{\text{CXCL9}}$ 
12        else if ( $m[\text{NF}\kappa\text{B}]$  and  $m[\text{state}] = \text{Resting}$ ) or  $m[\text{state}] = \text{Infected}$  then
13           $c[\text{TNF}] \leftarrow c[\text{TNF}] + 0.5 \cdot s_{\text{TNF}}$ 
14           $c[\text{CCL2}] \leftarrow c[\text{CCL2}] + 0.5 \cdot s_{\text{CCL2}}$ 
15           $c[\text{CCL5}] \leftarrow c[\text{CCL5}] + 0.5 \cdot s_{\text{CCL5}}$ 
16           $c[\text{CXCL9}] \leftarrow c[\text{CXCL9}] + 0.5 \cdot s_{\text{CXCL9}}$ 

```

Recruitment and proliferation

As opposed to macrophages and T cells, microglia are not recruited through vascular sources. Rather they proliferate on site. We make a distinction between ramified and reactivated microglia, as the first proliferate at a slower rate than the latter [37]. The parameter $N_{\text{m,ram}}$ denotes the number of ramified microglia on the grid. This number is maintained by proliferating ramified microglia with the following probability:

$$p_{\text{ram}} = \min\left(\frac{N_{\text{m,ram}} - n_{\text{ram}}}{N_{\text{m,ram}}}, 0\right). \quad (2.28)$$

In the previous subsection we mentioned that reactive microglia can either be resting, infected, chronically infected or active. As infected and chronically infected microglia contain intracellular bacteria, we chose to allow only activated and resting microglia to proliferate. Before a resting or active microglial cell can proliferate, the TNF- α level in the local micro-compartment

should be greater than $\tau_{m,\text{TNF},\text{prol}}$. To ensure that only a limited number of reactive microglia proliferate, the following probability is taken into account:

$$p_{\text{react}} = \frac{N_{m,\text{react},\text{prol}}}{n_{\text{res}} + n_{\text{act}}}. \quad (2.29)$$

So $N_{m,\text{react},\text{prol}}$ is roughly the number of resting and active microglia that can proliferate each time step. In Algorithm 35 the pseudo code for the proliferation of microglia is given.

We can see in Algorithm 36 that proliferation of a microglial cell results in the creation of a new microglial cell in the Moore neighborhood. In case the Moore neighborhood cannot accommodate a new microglial cell, the microglial cell in question does not proliferate. Since not much is known about the lifespan of microglia, we assign new microglial cells the same lifespan as macrophages (which is 100 days). In newly created microglia, the two transcription factors NF κ B and STAT1 are turned off initially.

Algorithm 35: PROLIFERATEMICROGLIA(\mathcal{G}, t)

- 1 Let \mathcal{L}_m be a list containing all microglia on \mathcal{G} Let n_{ram} , n_{res} and n_{act} be the number of ramified, resting and active microglia on \mathcal{G} , respectively
 - 2 $p_{\text{ram}} \leftarrow \min(\frac{N_{m,\text{ram}} - n_{\text{ram}}}{N_{m,\text{ram}}}, 0)$
 - 3 $p_{\text{react}} \leftarrow \frac{N_{m,\text{react},\text{prol}}}{n_{\text{res}} + n_{\text{act}}}$
 - 4 **foreach** $m \in \mathcal{L}_m$ **do**
 - 5 Let c be the micro-compartment on which m resides
 - 6 **if** $m[\text{state}] = \text{Ramified}$ **and** with probability p_{ram} **then**
 - 7 PROLIFERATEMICROGLIALCELL(m, \mathcal{G}, t)
 - 8 **if** ($m[\text{state}] = \text{Resting}$ **or** $m[\text{state}] = \text{Active}$)
 - 9 **and** $c[\text{TNF}] > \tau_{m,\text{TNF},\text{prol}}$ **and** with probability p_{react} **then**
 - 10 PROLIFERATEMICROGLIALCELL(m, \mathcal{G}, t)
-

Algorithm 36: PROLIFERATEMICROGLIALCELL(m, \mathcal{G}, t)

Input: \mathcal{G} is a uniform grid, m is a microglial cell and t is the current time step

- 1 Let c be a micro-compartment in the Moore neighborhood of m that can accommodate a microglial cell
 - 2 **if** c exists **then**
 - 3 Let m' be a new microglial cell
 - 4 $m'[t_{\text{birth}}] \leftarrow t$
 - 5 $m'[t_{\text{death}}] \leftarrow t + 14400$
 - 6 $m'[\text{NF}\kappa\text{B}] \leftarrow \text{false}$
 - 7 $m'[\text{STAT1}] \leftarrow \text{false}$
 - 8 $m'[\text{deact}] \leftarrow \text{false}$
 - 9 $m'[B_I] \leftarrow 0$
 - 10 $m'[\text{state}] \leftarrow m[\text{state}]$
-

There are 800 sources on the grid. It is through these sources that macrophages and T cells are recruited. Recruitment of T cells proceeds in the same way as in the lung model; only after t_{recl} is T cell recruitment enabled (see Algorithms 37 and 18). Almost the same rules apply for macrophage recruitment. While the same threshold condition of chemokines and TNF- α is used

(cf. Equation 2.22), this time the parameter N_M is a lower bound on the number of *meningeal* macrophages on the grid. This lower bound is enforced during recruitment: when the number of meningeal macrophages drops below the bound, macrophages are recruited through vascular sources on the meninges (see Algorithm 38).

Algorithm 37: RECRUIT(\mathcal{G}, t)

Input: \mathcal{G} is a uniform grid and t is the current time step

- 1 Let $\mathcal{L}_{\text{sources}}$ be the list of vascular sources on \mathcal{G} (by definition $|\mathcal{L}_{\text{sources}}| = 800$)
 - 2 **foreach** $c \in \mathcal{L}_{\text{sources}}$ **do**
 - 3 **if** $c[\text{caseous}] = \text{false}$ **then**
 - 4 **if** #cells at c is less than 2 **and** no macrophage or a microglial cell is present on c **then**
 - 5 RECRUITMAC(c, \mathcal{G}, t)
 - 6 **if** #cells at c is less than 2 **and** $t \geq t_{\text{recr}}$ **then**
 - 7 RECRUITTCELL(c, \mathcal{G}, t)
-

Algorithm 38: RECRUITMAC(c, \mathcal{G}, t)

Input: c is a non-caseated source not containing a macrophage or a microglial cell and t is the current time step

- 1 Let n_{macs} be the number of macrophages on the meninges of \mathcal{G}
 - 2 $b \leftarrow \alpha_{\text{TNF}} \cdot c[\text{TNF}] + \alpha_{\text{CCL2}} \cdot c[\text{CCL2}] + \alpha_{\text{CCL5}} \cdot c[\text{CCL5}] \geq \tau_{M,\text{recr}}$
 - 3 **if** (b **and** with probability $p_{M,\text{recr}}$) **or** ($n_{\text{macs}} < N_M$ **and** $c[\text{meninges}]$) **then**
 - 4 Let M be a new macrophage on micro-compartment c
 - 5 Set $M[t_{\text{birth}}]$ uniformly at random to a value in $[t - 14400, t]$
 - 6 $M[t_{\text{death}}] \leftarrow M[t_{\text{birth}}] + 14400$
 - 7 $M[\text{state}] \leftarrow \text{Resting}$
 - 8 $M[\text{deact}] \leftarrow \text{false}$
 - 9 $M[\text{NF}\kappa\text{B}] \leftarrow \text{false}$
 - 10 $M[\text{STAT1}] \leftarrow \text{false}$
 - 11 $M[B_I] \leftarrow 0$
-

Movement

Microglia have receptors for the chemokines CCL2 and CCL5 [2]. These two chemokines play a role in chemotaxis. Similarly to a macrophage, the speed by which a microglial cell moves depends on the state of that microglial cell. Ramified and chronically infected microglia do not move at all. Just like with macrophages, there is an additional bonus factor of 1.5 resulting in stronger bias of moving a microglial cell to the micro-compartment with the highest chemokine levels. In Algorithm 39 the pseudo code of this procedure is given.

In Subsection 2.2.1 we mentioned that the grid is not toroidal. There are several ways to deal with movement in a non-toroidal grid. One way would be to prevent cells from moving off the grid. This, however, would lead to crowding near the edges and corners of the grid. To deal with this issue, we chose to stop tracking immune cells as soon as they leave the grid. In order to model meningeal macrophages, we enforce that a macrophage residing on a micro-compartment that is part of the meninges can only move to other meningeal micro-compartments or off the grid. The probability of a T cell moving onto a micro-compartment already containing a

Algorithm 39: MOVEMICROGLIA(\mathcal{G}, t)

Input: \mathcal{G} is a uniform grid and t is the current time step

```

1 Let  $\mathcal{L}_m$  be a list containing all microglia on  $\mathcal{G}$ 
2 foreach  $m \in \mathcal{L}_m$  do
3   if  $m[\text{state}] = \text{Resting}$  and  $t \bmod v_{m,r} = 0$  then
4     MOVEIMMUNECELL( $m, \mathcal{G}, \text{true}, \text{false}, \text{true}, 1.5$ )
5   if  $M[\text{state}] = \text{Active}$  and  $t \bmod v_{m,a} = 0$  then
6     MOVEIMMUNECELL( $m, \mathcal{G}, \text{true}, \text{false}, \text{true}, 1.5$ )
7   if  $M[\text{state}] = \text{Infected}$  and  $t \bmod v_{m,i} = 0$  then
8     MOVEIMMUNECELL( $m, \mathcal{G}, \text{true}, \text{false}, \text{true}, 1.5$ )

```

microglial cell is given by $p_{T,m}$. The pseudo code corresponding to the movement of an immune cell is given in Algorithm 40.

2.2.4 Mtb specific rules

The growth function of intracellular bacteria in microglia is the same as that in macrophages (see Subsection 2.1.4). The same function for extracellular growth of Mtb is used as in the lung model.

2.2.5 Model parameters

The main differences between the lung and CNS model are:

- The environment is modeled as a grid of 200×200 micro-compartments instead of 100×100 micro-compartments.
- The grid is no longer toroidal: once an immune cell moves off the grid, it is no longer tracked.
- One side of the grid contains meninges. On these meninges meningeal macrophages are present; if a macrophage moves on the meninges it will be considered a meningeal macrophage, and as such will be unable to leave the meninges.
- There is a steady population of meningeal macrophages. If the number of such macrophages drops, new meningeal macrophages are actively recruited onto the meninges.
- The number of sources on the grid is 800 instead of 50.
- Microglia — a type of immune cells specific to the CNS — are included in the model.
- Microglia are not recruited through vascular sources. Rather they are proliferated on site.

Except for these differences, the CNS model is identical to the lung model. Therefore the parameters as presented in Table C.2 apply to the CNS model as well. In addition, in Table 2.3 the parameters that apply specifically to the CNS model are given.

Algorithm 40: MOVEIMMUNECCELL($ic, \mathcal{G}, b_{\text{CCL2}}, b_{\text{CCL5}}, b_{\text{CXCL9}}, \alpha$)

Input: ic is an immune cell on the uniform grid \mathcal{G} ; $b_{\text{CCL2}}, b_{\text{CCL5}}, b_{\text{CXCL9}}$ are flags denoting the involved chemokines and α is a bonus coefficient

```

1 Let  $\mathcal{G}_{kl}$  be the micro-compartment on which  $ic$  resides
2 for  $i \leftarrow -1$  to 1 do
3   for  $j \leftarrow -1$  to 1 do
4     /* Note that we initialize  $p_{ij}$  to a very small value  $\epsilon$ , in order to
5      prevent a possible divide by zero later during normalization. */
6      $p_{ij} \leftarrow \epsilon$ 
7     if  $b_{\text{CCL2}}$  then  $p_{ij} \leftarrow p_{ij} + \mathcal{G}_{k+i,l+j}[\text{CCL2}]$ 
8     if  $b_{\text{CCL5}}$  then  $p_{ij} \leftarrow p_{ij} + \mathcal{G}_{k+i,l+j}[\text{CCL5}]$ 
9     if  $b_{\text{CXCL9}}$  then  $p_{ij} \leftarrow p_{ij} + \mathcal{G}_{k+i,l+j}[\text{CXCL9}]$ 
10 Multiply the highest value in  $p$  by  $\alpha$ 
11 Normalize  $p$ 
12 Let  $c$  be the micro-compartment picked according to the probabilities in  $p$ 
13 if  $c$  is not on  $\mathcal{G}$  then
14   Kill  $ic$ 
15 else if  $c$  is not caseous and has room for  $ic$  then
16   /* Move with probability  $p_{\text{T,M}}$ , if  $c$  contains a macrophage */
17   /* Move with probability  $p_{\text{T,m}}$ , if  $c$  contains a microglial cell */
18   /* Move with probability  $p_{\text{T,T}}$ , if  $c$  contains a T cell */
19   if  $ic$  is a T cell and  $c$  contains a macrophage then
20     With probability  $p_{\text{T,M}}$ , move  $ic$  onto  $c$ 
21   else if  $ic$  is a T cell and  $c$  contains a microglial cell then
22     With probability  $p_{\text{T,m}}$ , move  $ic$  onto  $c$ 
23   else if  $ic$  is a T cell and  $c$  contains a T cell then
24     With probability  $p_{\text{T,T}}$ , move  $ic$  onto  $c$ 
25   else if  $ic$  is a macrophage and  $\mathcal{G}_{kl}[\text{meninges}]$  then
26     Only move  $ic$  onto  $c$  if  $c[\text{meninges}]$ 
27   else
28     Move  $ic$  onto  $c$ 

```

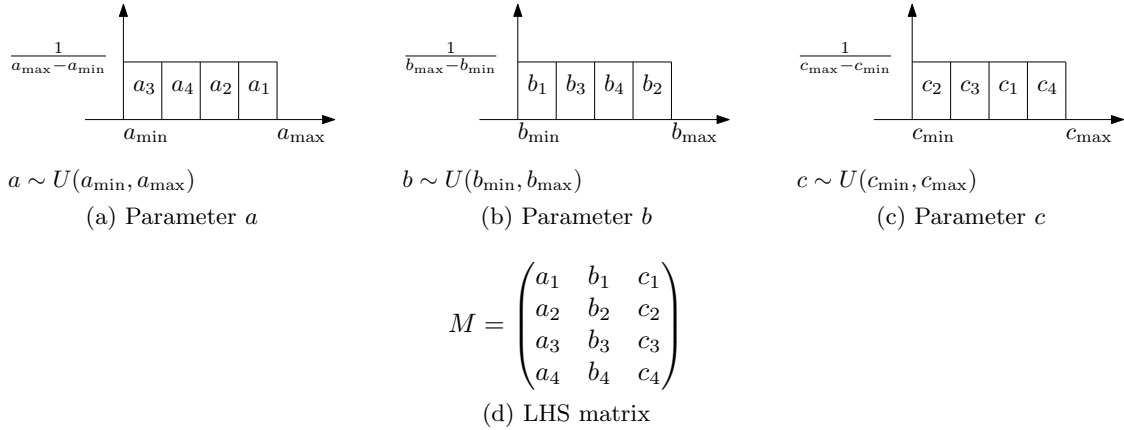
Table 2.3: CNS model parameters

Parameter	Description
$\tau_{m,TNF,NF\kappa B}$	TNF- α level required for NF κ B activation in microglia
$N_{m,Be,NF\kappa B}$	Number of extracellular bacteria in the Moore neighborhood required for NF κ B activation in microglia
$\sigma_{m,deact}$	Time span during which a microglial remains deactivated
N_M	Minimal number of meningeal macrophages on the grid
$N_{m,ram}$	Minimal number of ramified microglia on the grid
$\tau_{m,TNF,react}$	TNF- α level required for a ramified microglial cell to become resting
$p_{m,STAT1}$	Probability of a T_γ cell to enable STAT1 in a microglial cell
$p_{m,kill}$	Probability of a resting microglial cell to kill an extracellular bacterium
$N_{m,Bi,CI}$	Number of intracellular bacteria for an infected microglial cell to become chronically infected
$N_{m,Bi,burst}$	Number of intracellular bacteria for a chronically infected microglial cell to burst
$N_{m,Be,kill}$	Number of extracellular bacteria that an active microglial cell can kill every time step
$N_{m,react,prol}$	Number of reactive (either resting or active) microglia that can proliferate every time step
$\tau_{m,TNF,prol}$	TNF- α level required for proliferation of reactive (either resting or active) microglia
$v_{m,r}$	Time required for a resting microglial cell to move one micro-compartment
$v_{m,a}$	Time required for an active microglial cell to move one micro-compartment
$v_{m,i}$	Time required for an infected microglial cell to move one micro-compartment
$p_{T,m}$	Probability of moving a T cell to a micro-compartment containing a microglial cell

2.3 Uncertainty analysis methodology

There are 46 parameters in the lung model, and an additional 16 parameters in the CNS model. For most of these parameters no values are known. When assigning the unknown parameters different values, different model outputs may result. The uncertainty due to unknown parameter values is called *epistemic*. Since our models are stochastic by nature, they may result in different model outputs with the same parameter values. This introduces a second type of uncertainty, namely *aleatory* uncertainty. In order to deal with these two types of uncertainty, an uncertainty analysis is typically performed. In this section we will focus on the Latin Hypercube Sampling (LHS) method for uncertainty analysis [24].

In LHS a probability density function is assumed for every parameter. For simplicity we assume that we are dealing with uniform distributions only. Let k denote the number of parameters. The parameter values are sampled in a stratified manner and without replacement. Every parameter has the same number of strata, which we denote by n . Based on the probability density functions, the strata for every parameter correspond to equal probability intervals. The number of samples generated equals the number of strata. The n samples together with their

Figure 2.5: Example of LHS sampling on three parameters, with $n = 4$.

k values are stored in the so called LHS matrix which has dimensions $n \times k$. Each row of this matrix corresponds to a sample, which on its turn consists of k values. A parameter value in a row is generated by picking without replacement a stratum corresponding to that parameter. Subsequently a value within the stratum is picked uniformly at random. In Algorithm 41 the pseudo code of the LHS method is given; an example is given in Figure 2.5.

Algorithm 41: LHS(P, n)

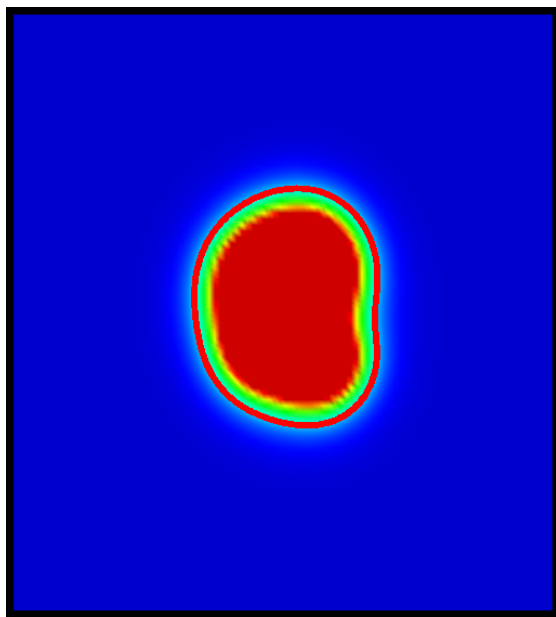
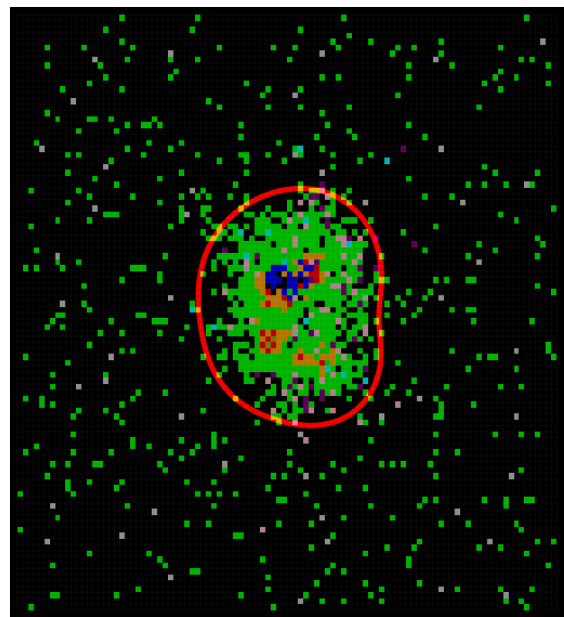
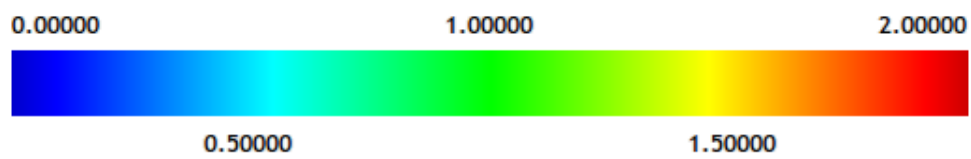
Input: $P = \{x_1, \dots, x_k\}$ is the set of parameters and n is the number of samples

```

/* Generate equally-probable strata for every parameter */
1 for  $i \leftarrow 1$  to  $k$  do
2   Let  $U(a, b)$  be the uniform distribution of  $x_i$ 
3    $w \leftarrow (b - a + 1)/n$ 
4   for  $j \leftarrow 1$  to  $n$  do
5      $S_i \leftarrow S_i \cup \{[a + (i - 1)w, a + iw]\}$ 
/* Generate the LHS matrix */
6 Let  $M$  be a  $n \times k$  matrix
7 for  $i \leftarrow 1$  to  $n$  do
8   for  $j \leftarrow 1$  to  $k$  do
9     Pick, uniformly at random, an interval  $I \in S_j$ 
10     $S_j \leftarrow S_j \setminus \{I\}$ 
11    Let  $x$  be a value in  $I$  uniformly at random
12     $M_{i,j} \leftarrow x$ 
13 return  $M$ 

```

Using the resulting LHS matrix, n model simulations are performed with parameters values as prescribed by the matrix. For every model simulation, we collect several output parameters. Interesting output parameters in our case are bacteria numbers (intracellularly as well as extracellularly) and the granuloma area. We consider micro-compartments whose TNF- α level is above a certain threshold to be part of a granuloma. This is a very natural thing to do, as in the model TNF- α is only secreted by macrophages participating actively in the immune response and these macrophages are most likely part of a granuloma (cf. Figure 2.6).

(a) TNF- α level(b) Immune cells: resting macrophages are green, infected macrophages orange, chronically infected macrophages red, T_γ cells are pink, T_{reg} cells cyan and T_c cells purple.(c) Color map for TNF- α levelsFigure 2.6: Granuloma area is determined using TNF- α levels.

In Section 5.1 we set out to find using LHS parameter values that result in the following scenarios.

- Clearance:

The bacteria is cleared, and after clearance the immune response terminates (i.e. no more cytokines/chemokines on the grid, T cells disappear and only resting macrophages remain present).

- Containment:

A stable granuloma is formed, the bacteria numbers remain constant as well as the granuloma area.

- Dissemination:

Bacteria numbers keep growing and no stable granuloma is formed.

We can see that there is a time element involved in containment and dissemination. Therefore we sample the outcomes over time and after every pre-configured interval perform a t-test between the samples collected in the previous time window and the current one. Let μ_1 be the mean and σ_1^2 the variance of the outcomes in the current time window. Conversely, μ_2 and σ_2^2 denote the mean and variance of the previous time window. For testing whether there is containment we formulate the following hypotheses:

$$\begin{aligned} H_0 : \mu_1 &= \mu_2, \\ H_1 : \mu_1 &< \mu_2. \end{aligned} \tag{2.30}$$

Similarly, we test for dissemination with the following two hypotheses:

$$\begin{aligned} H_0 : \mu_1 &= \mu_2, \\ H_1 : \mu_1 &> \mu_2. \end{aligned} \tag{2.31}$$

Since we cannot assume the sample variances σ_1^2 and σ_2^2 to be equal, we make use of Welch's t-test [26, 47]. The test statistic t_0 is defined as

$$t_0 = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{2.32}$$

the degrees of freedom is

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1+1} + \frac{(s_2^2/n_2)^2}{n_2+1}} - 2 \tag{2.33}$$

where n_1 and n_2 are the number of samples, \bar{x}_1 and \bar{x}_2 the sample means, and s_1^2 and s_2^2 the estimated sample variances in the current and previous time window, respectively.

The null-hypothesis in Equation 2.30 is rejected if

$$t_0 \geq t_{\nu;\alpha}. \tag{2.34}$$

Conversely, the alternative hypothesis in Equation 2.31 is accepted if

$$t_0 \leq -t_{\nu;\alpha}. \tag{2.35}$$

In order to determine in what way the input parameters affect the outcome, a sensitivity analysis can be performed [24]. After performing an extensive LHS, we can compute for every input parameter an accurate partial rank correlation coefficient. This coefficient describes the relation between the input parameter and the output variable. In Chapter 5 we do not perform a sensitivity analysis, instead we focus on finding stable baseline parameter sets using the just described classifications.

Chapter 3

Visualization

A picture is worth a thousand words. This is especially true when dealing with complex models such as the ones described in Chapter 2. In this chapter we present and describe the novel various visualization techniques that were employed to make the models more comprehensible. We start by giving an overview in Section 3.1 of the datasets that can be visualized. In Section 3.2 we discuss the method used for visualizing the grid and the immune cells. Methods for representing scalar datasets are described in Section 3.3 followed by methods for the visualization of vector datasets in Section 3.4. Rotating, panning and zooming is discussed in Section 3.5. We conclude by discussing in Section 3.6 how various visualization techniques can be combined.

Note that all visualization methods discussed in this chapter happen in real-time. So whenever a time step of the model is computed, the visualization is updated to match the latest model state. This allows the user to actually see in real-time how the immune cells, chemokine/cytokine concentrations and bacteria populations develop over time. As a final note we would like to mention that the focus will be on the CNS/TBM model throughout this chapter, as this model is an extension of the lung model.

3.1 Datasets

The environment of both our models are represented by uniform grids. The following scalar attributes present at every micro-compartment of the grid comprise the *scalar datasets*.

- TNF- α level;
- CCL2 level;
- CCL5 level;
- CXCL9/10/11 level;
- Number of extracellular bacteria B_E ;
- Number of intracellular bacteria B_I .

In Subsections 2.1.3 and 2.2.3 chemotaxis has been modeled by using probabilities proportional to the chemokine levels. A natural way of looking at chemotaxis mathematically is by

considering the gradient of a chemokine ρ :

$$\nabla\rho = \left(\frac{\partial\rho_x}{\partial x}, \frac{\partial\rho_y}{\partial y} \right) \quad (3.1)$$

So the gradient is by definition a vector. Since our scalar datasets are discrete, we have to approximate $\nabla\rho$. Suppose that we want to calculate $\nabla\rho_{i,j}$ where i is the column index and j the row index. Since the grid is uniform, we have that $\Delta x = \Delta y$, i.e. the horizontal and vertical mesh spacing are the same. Using central differences, we can approximate $\nabla\rho_{i,j}$ as

$$\nabla\rho_{i,j} = \left(\frac{\rho_{i+1,j} - \rho_{i-1,j}}{2 \cdot \Delta x}, \frac{\rho_{i,j+1} - \rho_{i,j-1}}{2 \cdot \Delta x} \right). \quad (3.2)$$

When using this approximation we have to deal with border cases. For instance if we want to compute $\nabla\rho_{0,0}$, we have to compute $\nabla\rho_{-1,0}$ and $\nabla\rho_{0,-1}$. We can deal with this by setting the boundary conditions to 0.

By considering the gradient of several scalar datasets, we arrive at the following *vector datasets*.

- $\nabla \text{TNF-}\alpha$;
- ∇CCL2 ;
- ∇CCL5 ;
- ∇CXCL9 .

The immune cells in a micro-compartment and the flags — signalling whether the micro-compartment is caseous, a source, or part of the meninges — are not part of a scalar or vector dataset. Instead, we visualize them separately as we will see in the next section.

3.2 Agent visualization

The grid is drawn using transparent edges. Every micro-compartment of the grid is colored according to the type of immune cell present in that micro-compartment. Recall that in a micro-compartment two immune cells can be present. If this happens, T cells take priority over macrophages and microglia. The flags denoting whether a micro-compartment is a source or caseated have the highest priority. Sources are depicted with a gray color. Caseated micro-compartments are visualized as such by a cross. In summary, the priorities by which various grid components are drawn are as follows.

1. Sources;
2. Caseated micro-compartments;
3. T_γ cells;
4. T_c cells;
5. T_{reg} cells;
6. Macrophages and microglia;

7. Extracellular Mtb.

In the CNS model, meninges are depicted with a transparent ‘v’ rotated sideways in the background of the grid. As such they are always visible without obstructing the visualization of other grid components. In Figures 3.1a and 3.1c example grids of the lung and CNS model are shown.

The visualization of the grid and the agents is parameterizable. We allow the user to specify which types of immune cells to show, and whether to mark caseated micro-compartments as such. In addition, the user can decide to show only vascular sources that are able to recruit certain types of immune cells (cf. Figures 3.1b and 3.1d). By clicking with the mouse on a micro-compartment in the visualized grid, all information about that micro-compartment and the immune cells residing on it is displayed to the user (see Figure 3.2).

3.3 Scalar visualization

Scalar datasets can be visualized in various ways. We look at color maps in Subsection 3.3.1, isolines in Subsection 3.3.2 and height plots in Subsection 3.3.3.

3.3.1 Color maps

The most natural way to depict a scalar dataset on a grid is using colors. A color map takes a scalar value as input and maps it to a color. The easiest way to implement a color map is to make use of two mapping functions:

$$\begin{aligned} g : \mathbb{R} &\rightarrow [0, 1], \\ f : [0, 1] &\rightarrow R \times G \times B. \end{aligned}$$

According to a provided minimal value a and a maximal value b , the map g first clamps the scalar value into the range $[a, b]$ and subsequently normalizes the clamped value such that result is in the range $[0, 1]$. The map $f \circ g$ takes the result of g and maps it to a color in the RGB color space.

We have implemented the following five different color maps (cf. Figure 3.3).

1. Gray-scale:

A simple black to white color map.

2. Rainbow:

The widely used rainbow color map (which by some is considered harmful [8]).

3. Fire:

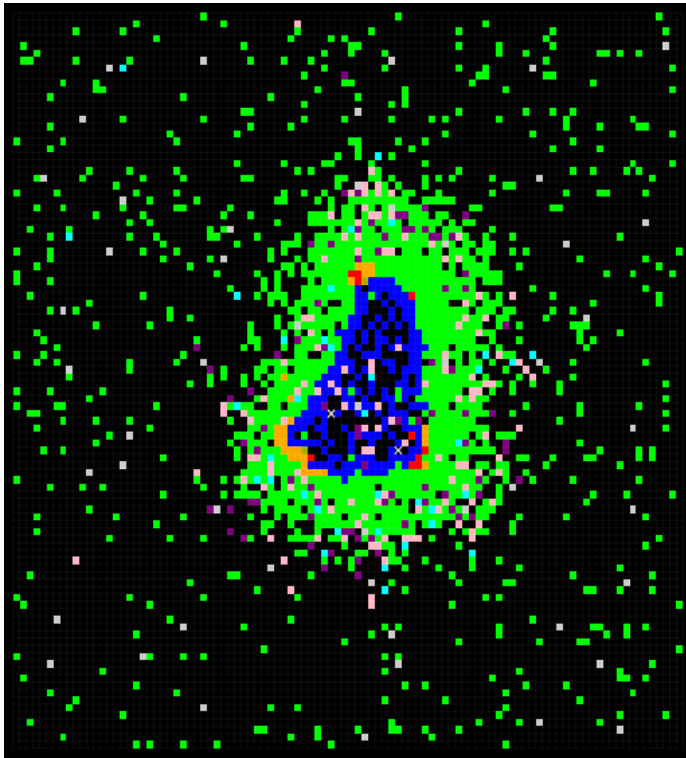
A color map that is inspired on black-body radiation.

4. Cool-warm:

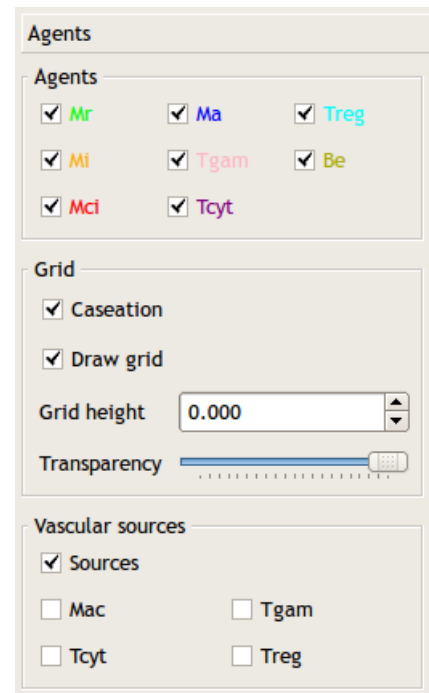
A simple color map that goes from blue to white to red.

5. Green-red:

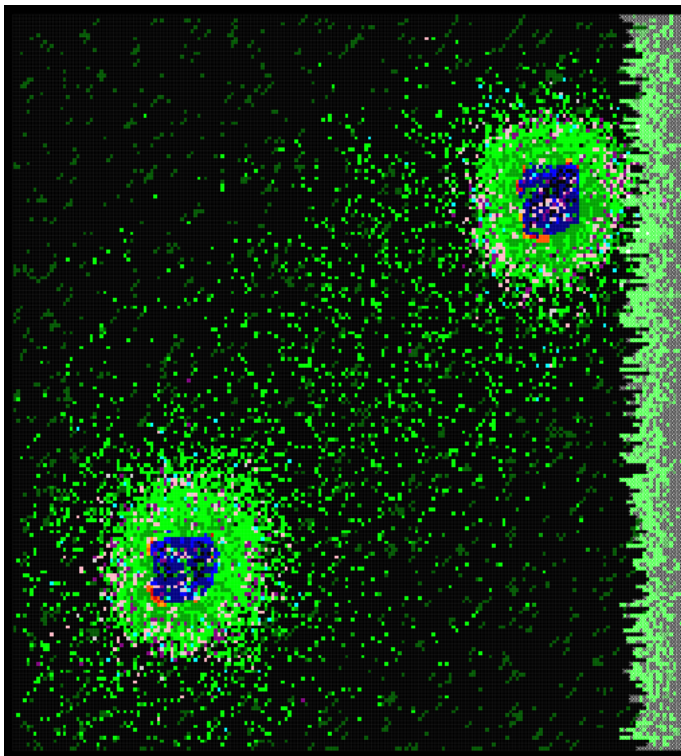
The color map that is used in micro-arrays.



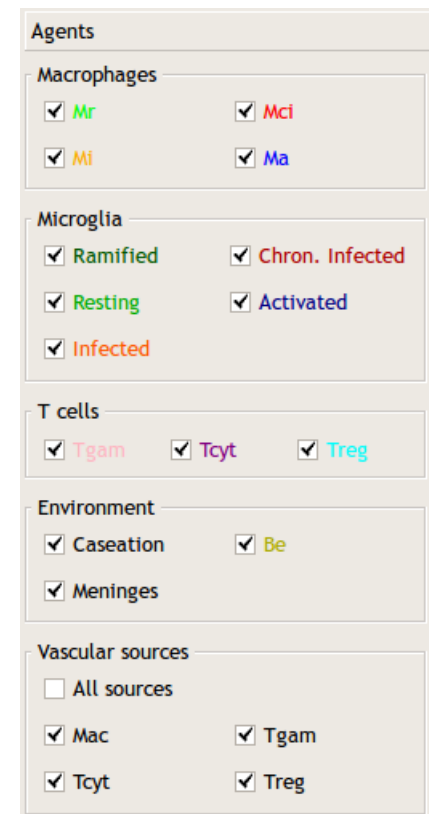
(a) Lung model: immune cells are visualized using different colors depending on their states. Vascular sources are colored in gray, and caseated compartments are depicted with a cross.



(b) In the user interface, the user can specify which types of immune cells to show.



(c) CNS model: meninges are on the right edge. It can be seen that the grid of this model is bigger.



(d) User interface: observe the additional cell type of microglia.

Figure 3.1: Visualization of immune cells and the grid in the lung and CNS model.

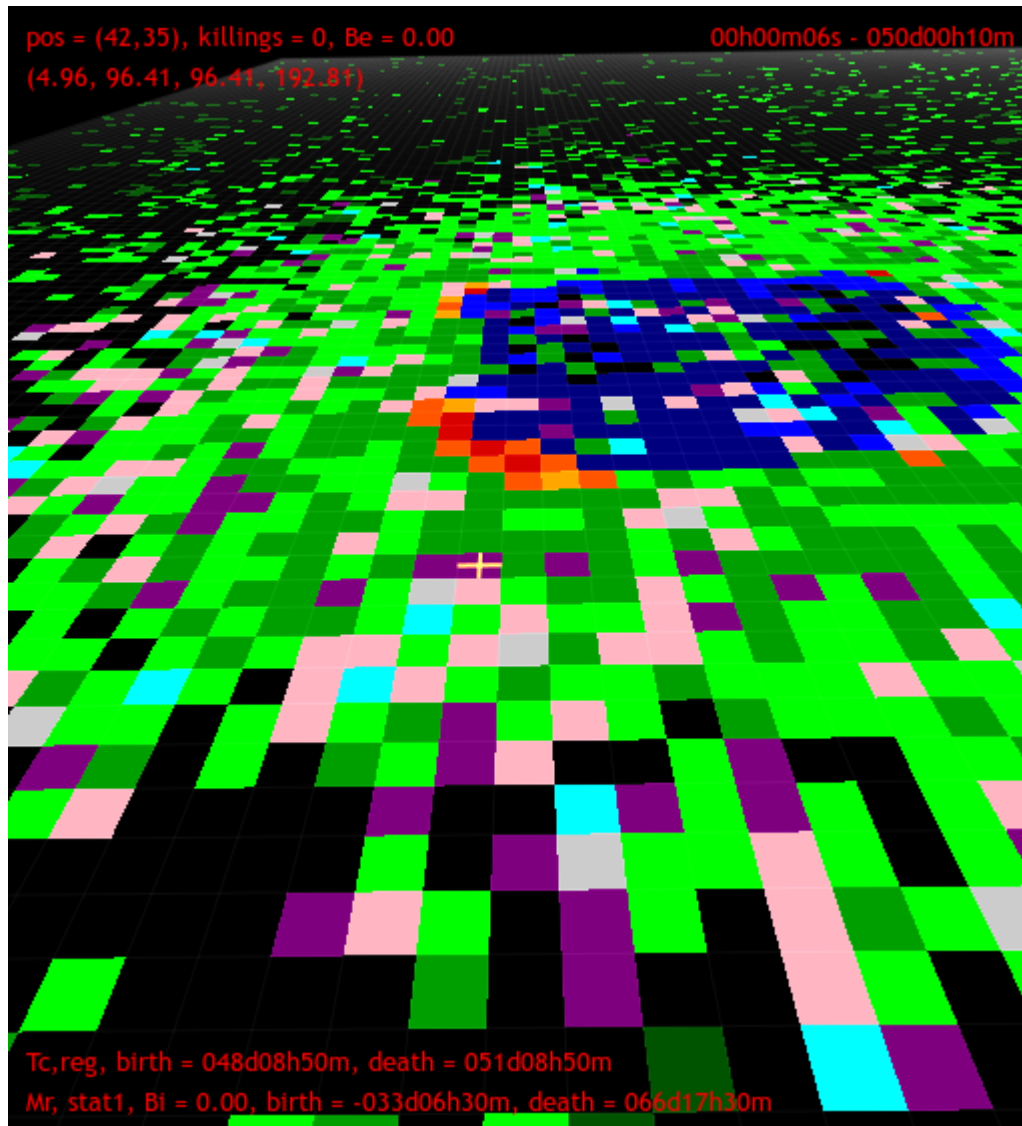


Figure 3.2: The yellow cross mark indicates the micro-compartment for which additional information is shown. At the top, the position is displayed along with the number of killings, number of extracellular bacteria and the levels of $\text{TNF-}\alpha$, CCL2, CCL5 and CXCL9/10/11 (second row, from left to right). The bottom two rows show that there is a down-regulated T_c cell present accompanied by a resting macrophage. The simulation is at day 50.

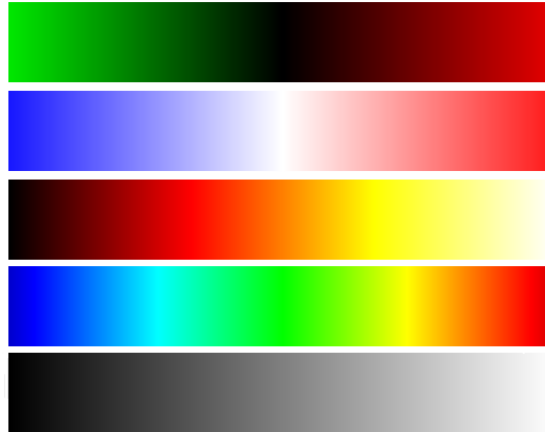


Figure 3.3: Implemented color maps, from top to bottom: green-red, cool-warm, fire, rainbow and gray-scale.

The grid is colored by drawing filled triangles whose vertices correspond to neighboring micro-compartments on the grid. The vertices of every triangle are assigned a color based on their scalar value and the used color map. The triangles are colored with a linear gradient that takes the colors of the comprising vertices into account (see Figure 3.5a for how the triangles are colored).

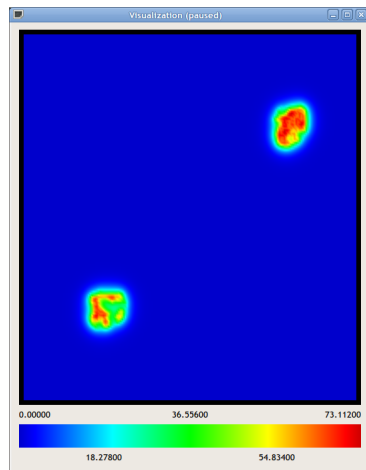
The number of colors to be used is a parameter that can be set. In addition to reversing the color map, it is also possible to specify a hue displacement, as well as adjusting the transparency, saturation and value of every color in the color map. Instead of specifying a fixed minimal and maximal value, the user has the possibility to use real-time scaling in which case the minimal and maximal values are updated every simulation time step to match the minimal and maximal value on the grid. In Figure 3.4 the TNF- α levels on the grid of Figure 3.1c are visualized using color maps.

3.3.2 Isolines

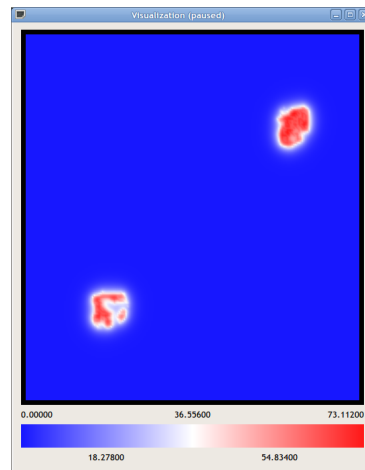
An often used method for displaying scalar datasets is contouring. With this visualization technique, a curve is drawn such that all points on that curve have the *same* scalar value — hence, the name isoline. Several isolines corresponding to different scalar values can be drawn at the same time.

For drawing isolines, we make use of the marching triangles algorithm, which is a simplified 2D version of the marching cubes algorithm [23]. As the name suggests, the marching triangles algorithm draws isolines by considering triangles. We consider the same triangles that were used in the color map visualization; that way the isoline will not conflict with the used coloring of the scalar dataset (see Figure 3.5a). For each vertex of a triangle we check whether the scalar at that vertex is larger than the target value. Doing this allows us to determine, using the case table in Figure 3.5b, on which edges the target value is located.

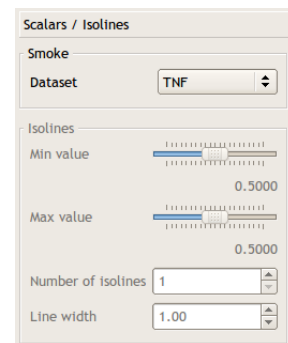
In the case table we can see that apart from the first and last case, there are always two vertices whose values equal the target value. In order to obtain the exact location of these vertices, we make use of linear interpolation. For instance, suppose that our target value v is located on the edge AB (see Figure 3.5c). Let v_A and v_B be the values of vertices p_A and p_B , respectively.



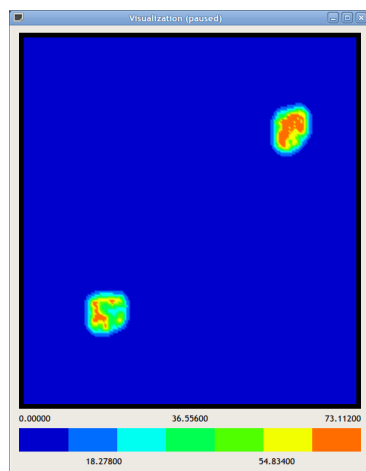
(a) The TNF- α levels of the grid in Figure 3.1c are visualized using the rainbow color map.



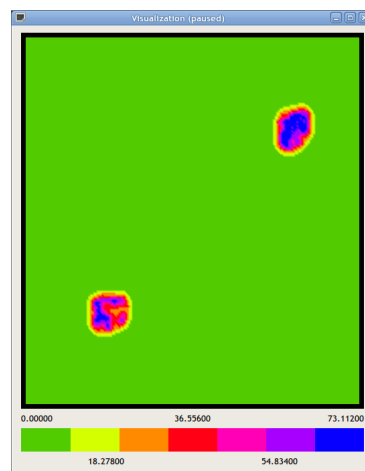
(b) Same dataset with the cool-warm color map.



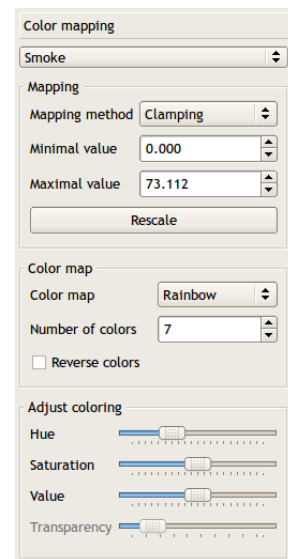
(c) In the user interface a scalar dataset can be selected.



(d) Rainbow colormap with 7 colors.

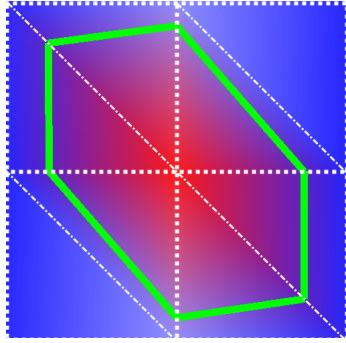


(e) Rainbow colormap with 7 colors and a hue displacement.



(f) User interface for color map parameters.

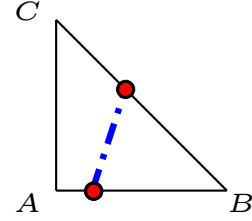
Figure 3.4: Visualization of a scalar dataset using a color map.



(a) Diagonals of the triangles used in marching triangles algorithm must be the same as those used in the color map visualization.

A	B	C	AB	BC	AC
0	0	0	-	-	-
0	0	1	-	*	*
0	1	0	*	*	-
0	1	1	*	-	*
1	0	0	*	-	*
1	0	1	*	*	-
1	1	0	-	*	*
1	1	1	-	-	-

(b) Marching triangles case table.



(c) Example triangle as used in the algorithm.

Figure 3.5: Marching triangles algorithm.

The coordinate of the vertex p whose value equals the target value v is then given by

$$p = p_A + \frac{v - v_A}{v_B - v_A} \cdot (p_B - p_A). \quad (3.3)$$

In the user interface a range $[v_1, v_2]$ can be set. The number n of isolines to be drawn is also configurable. The target value v_i for an isoline ℓ_i , where $0 \leq i < n$ is given by

$$v_i = v_1 + (v_2 - v_1) * i. \quad (3.4)$$

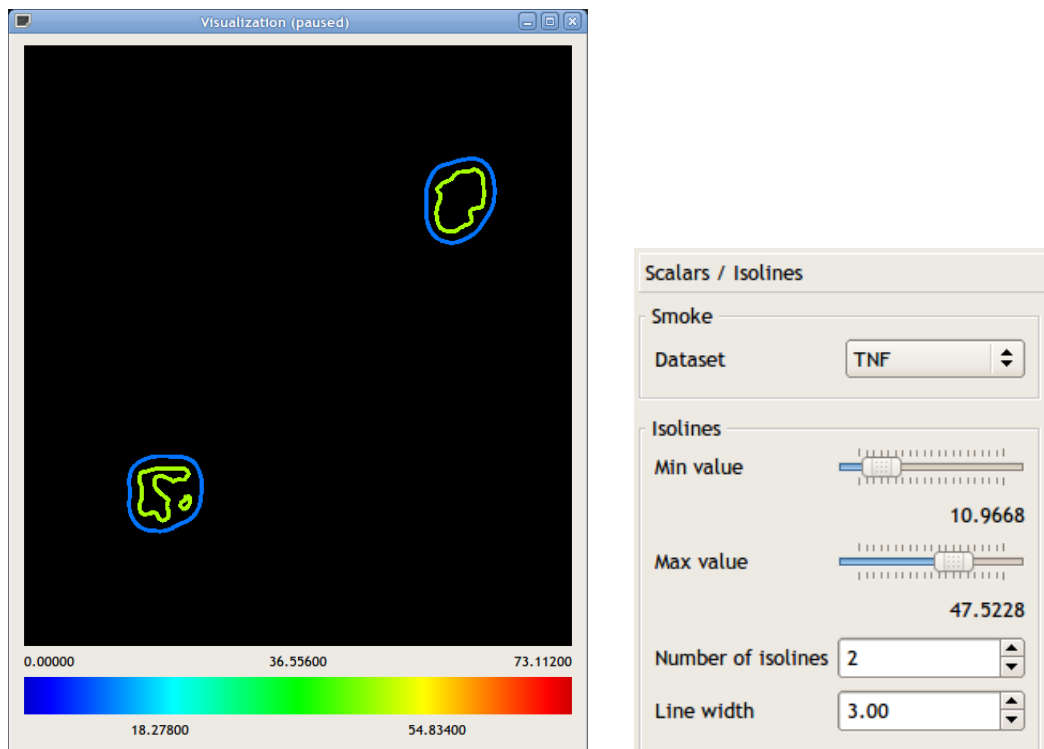
Since the dataset's range changes over time, we have decided to split the process of selecting the actual values of v_1 and v_2 in two parts. In the isoline settings a user selects a normalized range of interest. Subsequently, in the color map settings the actual minimal and maximal values can be set. It is also possible to select the scaling option, which updates the minimal and maximal value every time step. For increased visibility, the line width of isolines can be set. In Figure 3.6 the TNF- α levels on the grid of Figure 3.1c are visualized using two isolines.

3.3.3 Height plot

The third method of visualizing a scalar dataset is using an elevation plot, or a height plot. In such a plot scalar values are mapped to height coordinates. Similarly to the two previous methods, we draw the height plot using triangles. The z -coordinates of the vertices of these triangles represent the scalar values.

For drawing the height plot, we place a directional light source above the grid. We make use of Gouraud shading [14] with fixed material parameters. This method requires normal vectors at every vertex. We obtain these as follows. The height z of every grid point (x, y) is given by $z = f(x, y)$. We can now define the function F as $F(x, y, z) = f(x, y) - z = 0$. Since the gradient at any point is perpendicular to the level set, the normal at (x, y, z) is given by

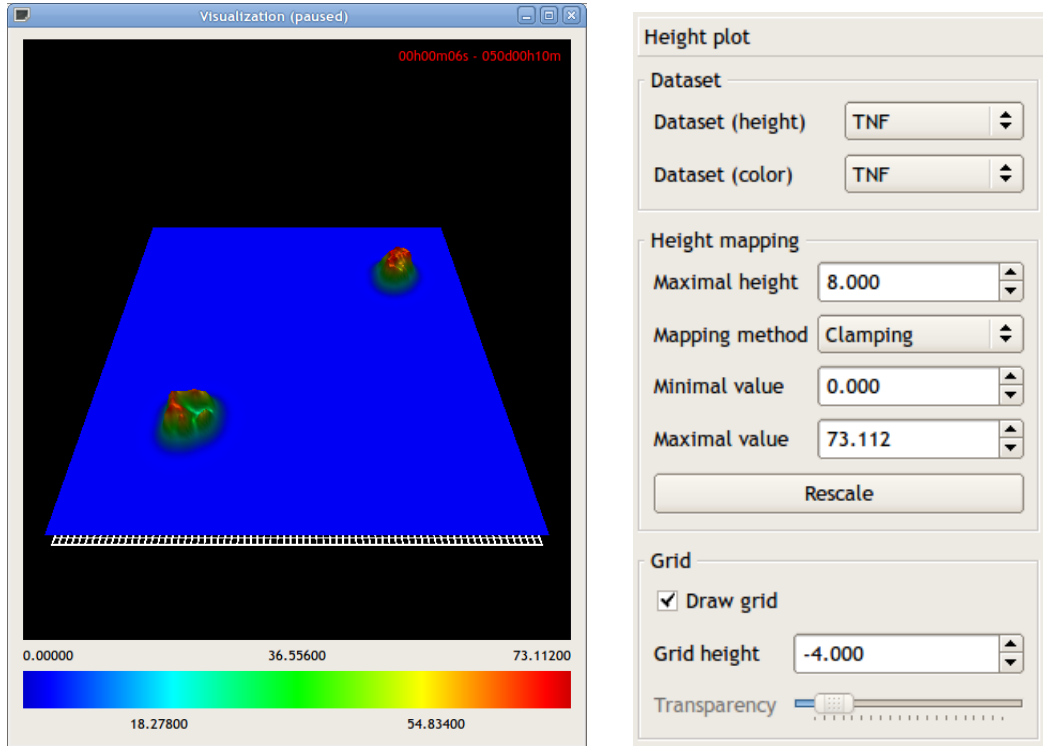
$$\nabla F = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, 1 \right). \quad (3.5)$$



(a) Visualization of two isolines of the TNF- α scalar dataset of the grid in Figure 3.1c.

(b) User interface of isoline specific settings.

Figure 3.6: Visualization of a scalar dataset using isolines.



(a) Height plot of the TNF- α scalar dataset of the grid in Figure 3.1c.

(b) User interface of settings specific to the height plot.

Figure 3.7: Visualization of a scalar dataset using a height plot.

Just like with the color maps, we allow the user to specify the clamping interval or to let the application dynamically update this interval. The maximal z -coordinate (i.e. the maximal height) can also be set in the user interface. In Figure 3.7 a height plot visualization of the TNF- α scalar dataset of the grid depicted in Figure 3.1c.

3.4 Vector visualization

In Section 3.1 we defined four vector datasets involving the gradient of TNF- α , CCL2, CCL5 and CXCL9/10/11. In this section we describe a visualization technique for visualizing these datasets.

The idea is to draw for every vector on the grid a *glyph* whose size is adjusted to reflect the value of the vector it represents. The glyph length increases linearly with the magnitude of the considered vector. The color, on the other hand, corresponds to a scalar dataset. Four different glyph shapes have been implemented: hedgehogs, cones, arrows and textures (see Figure 3.8).

In Figure 3.9 an example cone and arrow glyph are shown. The magnitude of the cone corresponds to the magnitude of the vector \vec{q} it represents, whereas the radius of the base of the cone is roughly one third of the vector magnitude. This results in a good ratio between the thickness and height of a cone glyph. Also a disc is added in order to make the cone closed at the bottom. An arrow consists of a cylinder and a cone. The radius of the cylinder is $0.2 \cdot |\vec{q}|$. The height of the cylinder is $0.6 \cdot |\vec{q}|$. On top of the cylinder a cone structure is placed. The

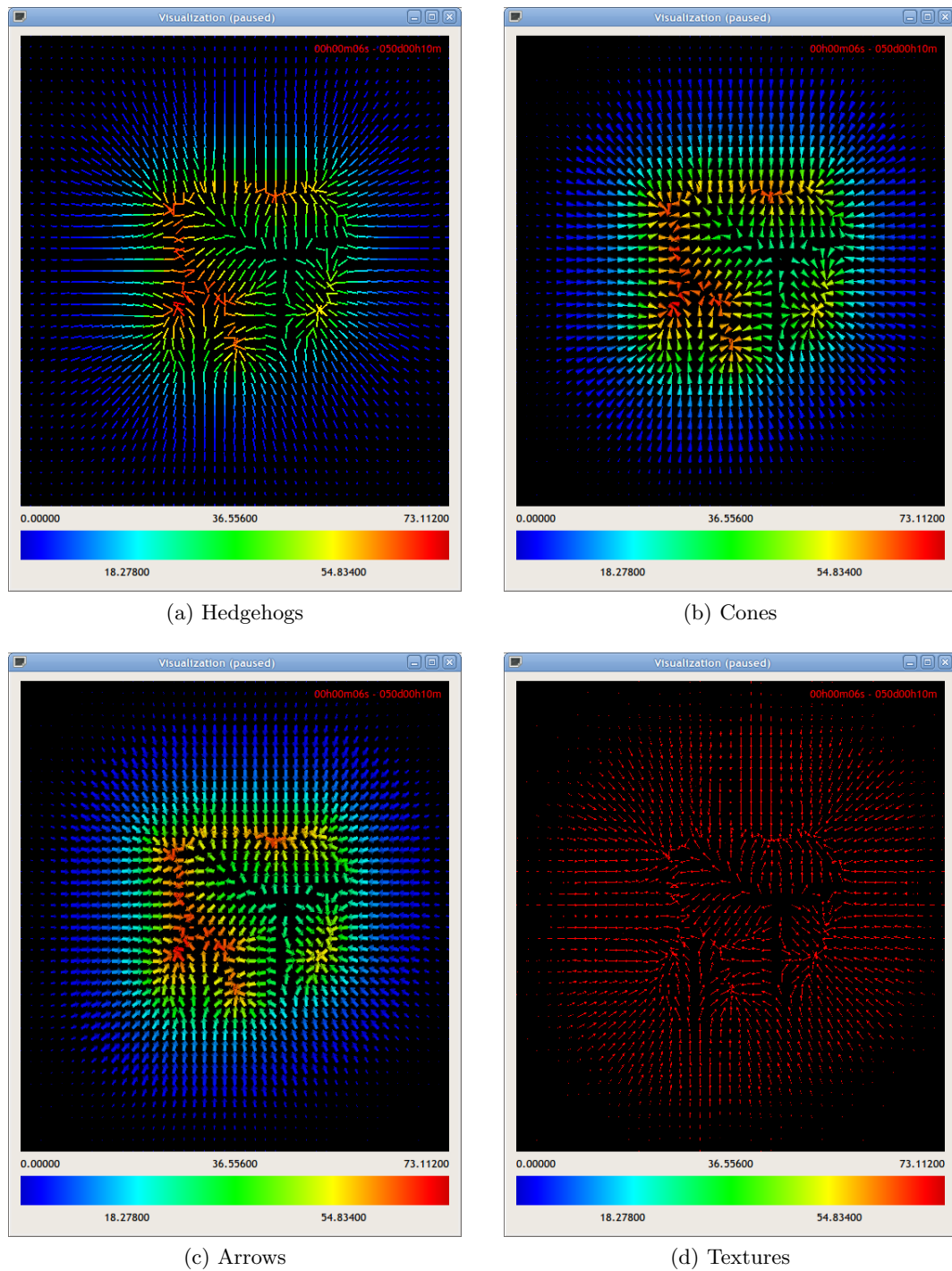


Figure 3.8: Visualization of $\nabla \text{TNF-}\alpha$, zoomed in on granuloma at bottom-left corner of Figure 3.1c.

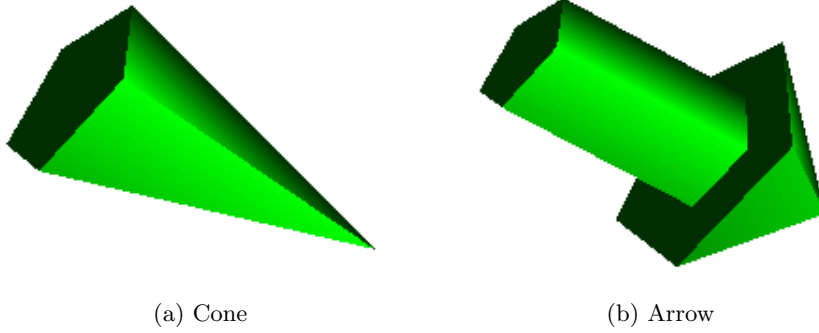


Figure 3.9: Cone and arrow design.

radius of the base of that cone is the same as its height, which is $0.4 \cdot |\vec{q}|$. Similarly to a cone glyph, two discs are drawn at the bases of the cylinder and the cone.

Unlike hedgehogs which are drawn directly on their coordinates, the primitives comprising the arrows and cones are drawn on the origin pointing toward the positive z -axis. So in order to draw a cone/arrow with its origin on an arbitrary grid point p and an angle of α , the following sequence of transformations is required.

1. Translate to (p_x, p_y) .
2. Rotate $\frac{\pi}{2}$ over the y -axis.
3. Rotate α over the x -axis.

This corresponds to [16]

$$\begin{pmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\frac{\pi}{2}) & 0 & \sin(\frac{\pi}{2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\frac{\pi}{2}) & 0 & \cos(\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3.6)$$

If we let \vec{q} denote the vector represented by the glyph, working out the matrix multiplications results in

$$\begin{pmatrix} 0 & \sin(\alpha) & \cos(\alpha) & p_x \\ 0 & \cos(\alpha) & -\sin(\alpha) & p_y \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & \frac{q_y}{|\vec{q}|} & \frac{q_x}{|\vec{q}|} & p_x \\ 0 & \frac{q_x}{|\vec{q}|} & -\frac{q_y}{|\vec{q}|} & p_y \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3.7)$$

The fourth glyph type is based on a user provided texture. We assume that a typical glyph texture is in the form of an arrow pointing in Northeast direction (see Figure 3.10a). For drawing the glyph texture we make use of the transformation described in Equation 3.7. For that reason we draw the squared texture placeholder in yz -plane with the diagonal lying on the positive z -axis. Since the grid is uniform, the angle of the diagonal — denoted by α in Figure 3.10b — is $\pi/4$.

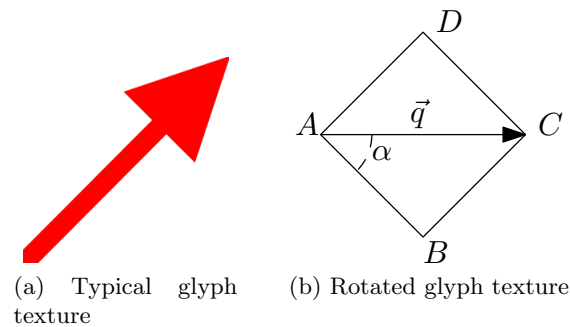
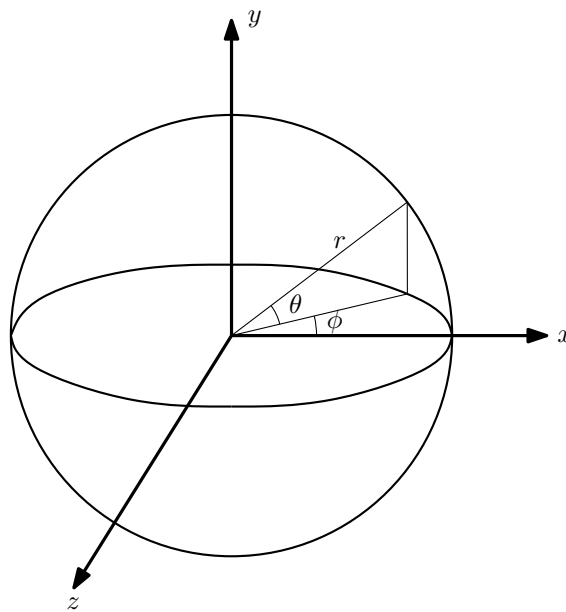


Figure 3.10: Design of a glyph texture.

Figure 3.11: Initially, the center of the grid coincides with the sphere's origin. A coordinate in the spherical coordinate system is defined by (r, θ, ϕ) .

3.5 Camera

The camera is placed on a sphere containing the grid. In order to keep the camera placed on the sphere, we make use of a spherical coordinate system. This coordinate system has three coordinates: r , θ and ϕ . The radial distance from a point to the origin is denoted by r . The angles θ and ϕ are defined as in Figure 3.11.

A spherical coordinate (r, θ, ϕ) can be mapped to the Cartesian coordinate (x, y, z) as follows.

$$x = r \cos \phi \cos \theta, \quad (3.8)$$

$$y = r \sin \theta, \quad (3.9)$$

$$z = -r \sin \phi \cos \theta. \quad (3.10)$$

The camera can be rotated by moving the mouse while pressing the middle mouse button.

Moving the mouse horizontally adjusts ϕ , whereas moving the mouse vertically modifies θ . The radius can be altered using the scroll wheel. Since we use perspective projection, increasing the radius corresponds to zooming-out and decreasing the radius to zooming-in.

Panning corresponds to changing the origin of the sphere; we do this by allowing the user to specify a displacement of Δx and Δy :

$$x' = \Delta x + x, \quad (3.11)$$

$$y' = \Delta y + y. \quad (3.12)$$

Moving horizontally while pressing the right mouse button changes the value of Δx . Conversely, moving vertically adjusts the value of Δy . The grid displayed in Figure 3.2 is obtained by rotating, zooming in and panning to the granuloma at the bottom-left corner of Figure 3.1c.

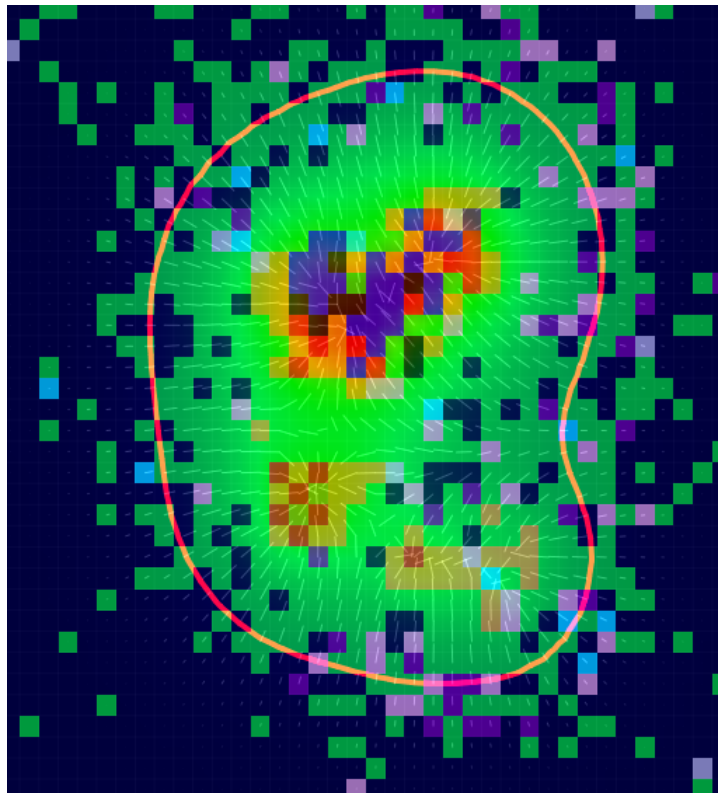
Regardless of how the camera is positioned, left clicking on a micro-compartment on the grid displays information about that micro-compartment. This is done by transforming view-port coordinates back to model coordinates. Since view-port coordinates lack a depth coordinate, we obtain the depth by reading out the depth buffer.

3.6 Combining visualization techniques

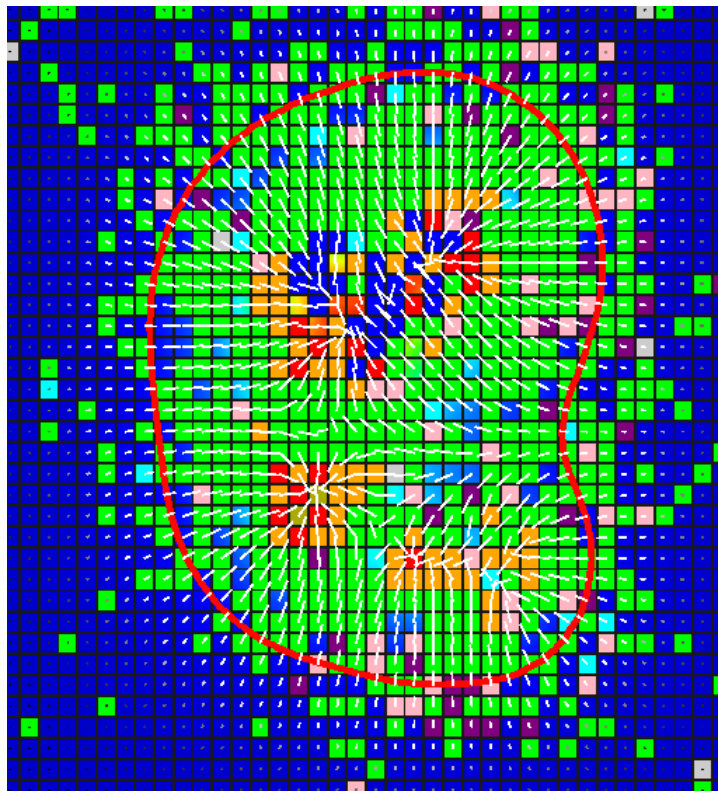
In the previous subsections we have introduced the following visualization techniques.

1. Color maps;
2. Isolines;
3. Height plot;
4. Glyphs.

With the exception of height plot visualization, all discussed visualization techniques can be combined allowing for the visualization of several datasets at the same time. When combining several visualization techniques, it is often necessary to use blending (see Figure 3.12).



(a) With blending



(b) Without blending

Figure 3.12: Immune cells are visualized, as well as the TNF- α concentration using the rainbow color map, ∇ TNF- α using hedgehogs and an isoline contouring everything with a TNF- α level of 3.5. Observe that the hedgehogs near the isoline are perpendicular to it.

Chapter 4

Implementation

The purpose of this chapter is to describe how the models and concepts introduced in the previous two chapters have been implemented. We give a top-level overview of the developed software in Section 4.1. This is followed by specific implementation details in Section 4.2.

4.1 Top-level overview

The software that we want to develop should satisfy the following requirements.

1. The software should be cross-platform: it should work on Windows, MacOS X and Linux.
2. The software should have a graphical user interface allowing the user to inspect the various model attributes in real-time.
3. The software should allow the user to inspect model statistics using the visualization techniques discussed in Chapter 3.
4. The software should be responsive: while the simulation is running, the user should be able to interact with the software without any observable delays.
5. The software should allow all parameters (as introduced in Sections 2.1 and 2.2) to be configured to user-specified values.
6. The software should be easily maintainable and extendible with new instrumentation or different models.

Since a simulation time step is computationally very expensive, we specifically chose not to use a managed programming language (such as Java or any of the .NET languages). Instead we decided to use C++. In order to satisfy the cross-platform requirement, we make use of the Qt framework. Qt is a C++ framework for developing cross-platform applications with a graphical user interface (GUI). The visualization techniques discussed in the previous chapter are implemented using OpenGL.

In Figure 4.1 a top-level overview of the developed software is given. There it can be seen that the software consists of four components: the model component, the visualization component,

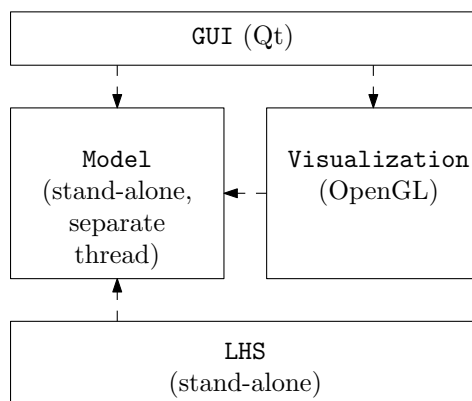


Figure 4.1: Top-level overview: there are four components: model, visualization, user interface and LHS. The model component is an independent component: it can be compiled as a separate executable. The same applies for the LHS component. The visualization component depends on the model component, and the user interface component depends on both the model and visualization components.

the user interface component and the LHS component. The model is a stand-alone component, i.e. it does not depend on the other components and it can be compiled into a separate executable. In order to satisfy the responsiveness requirement, the model component is run in a separate thread. The LHS component compiles into a separate executable. The visualization and LHS component only depend on the model component, whereas the GUI component needs both the model and visualization components. Modularizing the software in separate components serves both maintainability and extensibility.

4.2 Implementation details

In this section we present the implementation details of the components introduced in the previous section. In our description we will focus on the CNS model, as this model is an extension of the lung model.

4.2.1 Model

The class diagram describing the implemented classes is given in Figure 4.2. There we can see that the central class is `Simulation`. In this class a simulation time step can be performed; for that reason instances of the following classes are contained within the class.

- **Grid.**
This class contains all the micro-compartments, which on their turn are modeled by the class `GridCell`.
- **Stats.**
Contains all relevant statistics about the current model simulation.
- **Diffusion.**

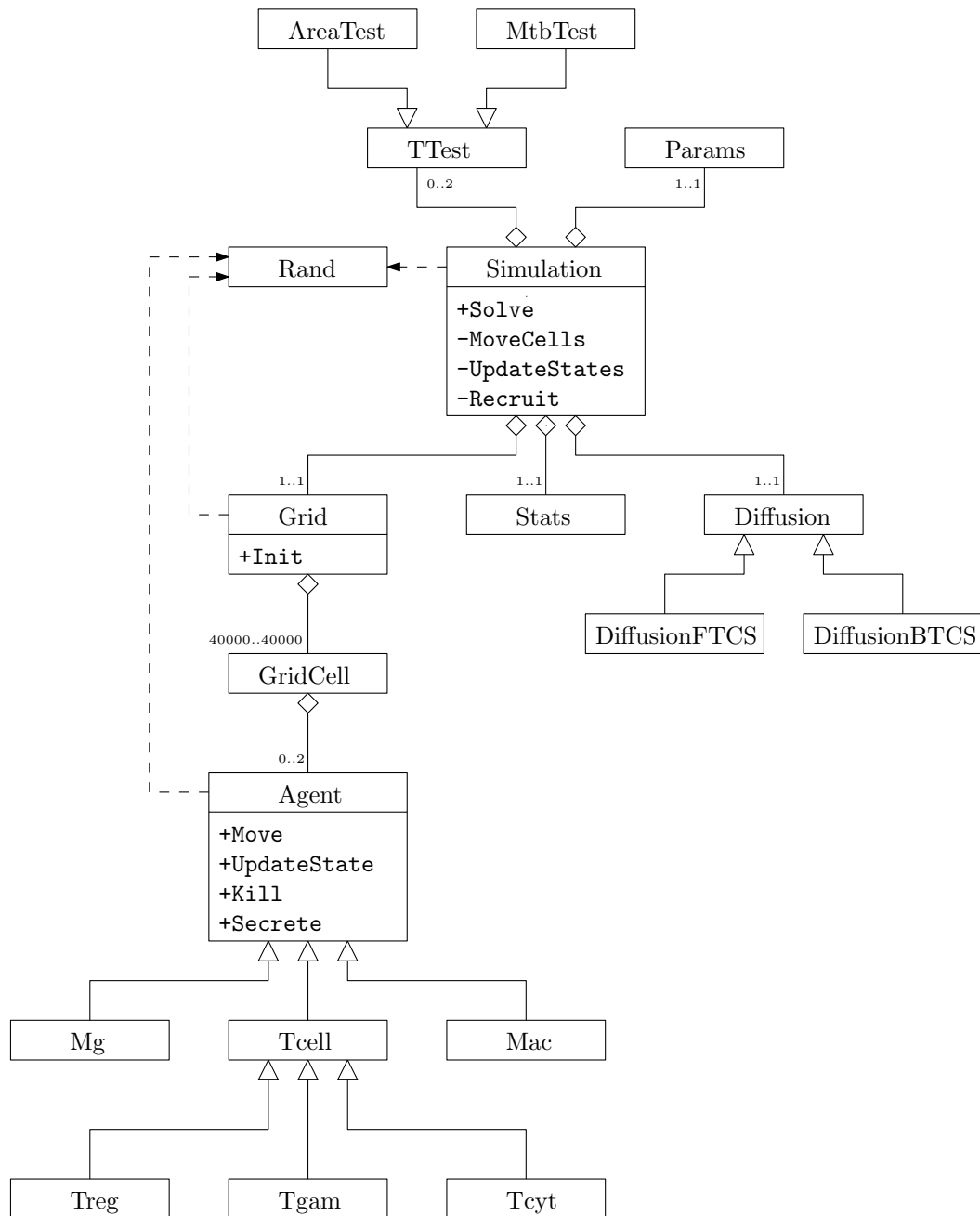


Figure 4.2: Class diagram of the brain model.

An abstract base class representing the used diffusion method. The method described in Subsection 2.1.3 is implemented by the class `DiffusionFTCS` (FTCS stands for forward time centered space). The class `DiffusionBTCS` — where BTCS stands for backward time centered space — implements an implicit method for solving the diffusion equation.

- **TTest.**

A base class of two t-test methods based on the number of bacteria — `MtbTest` — and the granuloma area — `AreaTest` — see Subsection 2.3. For evaluating Student's-t distribution, we make use of the Boost C++ libraries.

- **Params.**

This class contains the values of all parameters. It also provides functionality for reading and saving parameter values to an XML file. XML-related functionality is provided by the TinyXML library.

Micro-compartments are modeled by the class `GridCell`. In addition to chemokines, TNF- α , the number of caseous killings, and flags signalling whether a micro-compartment is a source or part of the meninges, a micro-compartment can also contain at most two immune cells. The abstract base class for immune cells is `Agent`. In this class the interface to which every immune cells must adhere to is defined: all immune cells are able to move, be killed, change state, and possibly secrete chemokines/cytokines. Since T cell movement proceeds for all T cell types similarly (except for the use of different chemokines), the common functionality has been isolated in the base class `Tcell`. The classes representing the immune cells are as follows.

- `Mg` represents a microglial cell;
- `Mac` represents a macrophage;
- `Treg` represents a T_{reg} cell;
- `Tgam` represents a T_{γ} cell;
- `Tcyt` represents a T_{c} cell.

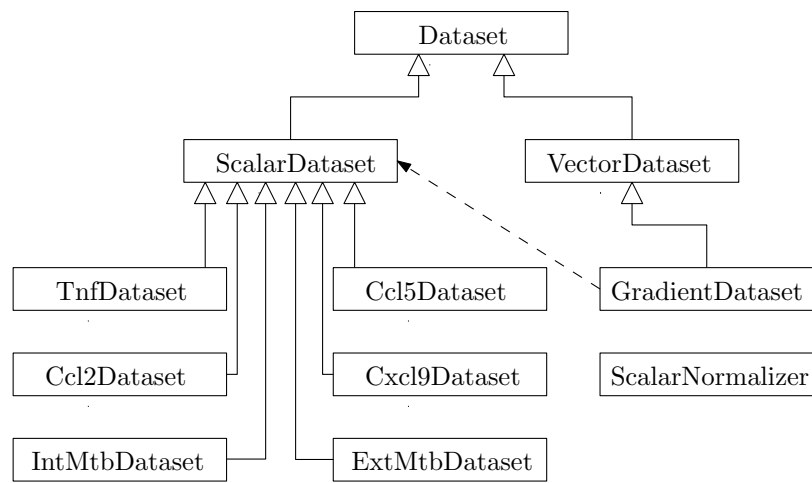
We make use of the Mersenne twister algorithm [25] in the Boost C++ library to generate pseudo random numbers. To allow for a different random number generator, we have introduced the wrapper class `Rand` for random number generation, which has been modeled as a singleton class.

Additional functionality for serializing and deserializing the model state has been implemented. As such, interesting model scenarios can be saved, and resimulated later on — possibly with different parameter values (such as depleting TNF- α) allowing the user to study the effect of certain parameters on the outcome in a specific run.

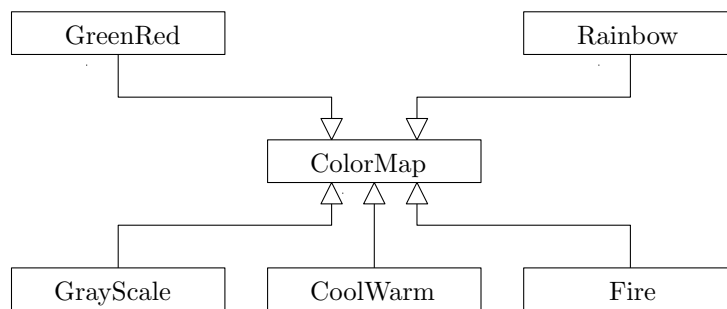
4.2.2 Visualization

The visualization techniques discussed in the previous chapter have been implemented using OpenGL. In Figure 4.3 all relevant classes of the visualization component are given.

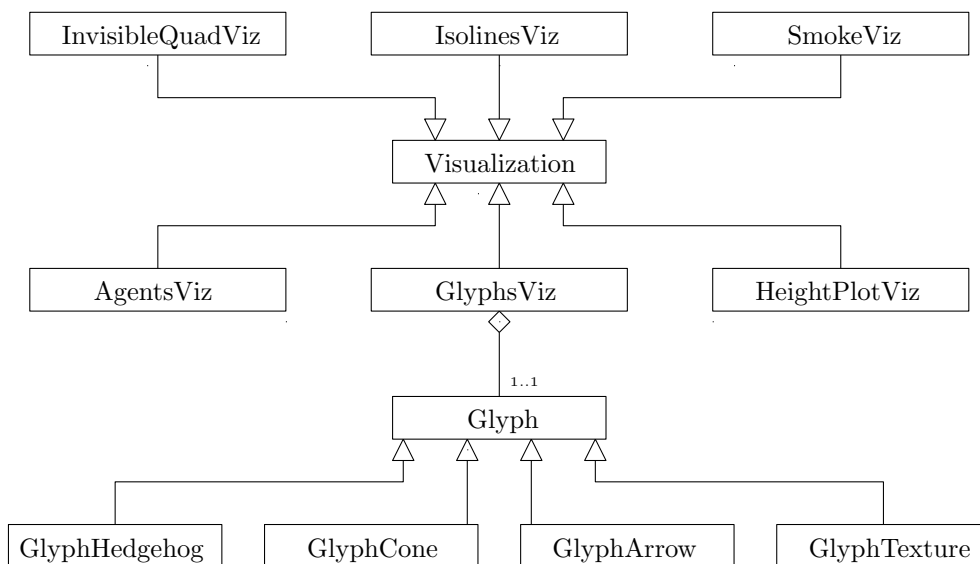
In Section 3.1 we introduced scalar and vector datasets. The base class of both types of datasets is `Dataset`; in this base class functionality for bilinear interpolation is provided.



(a) Datasets



(b) Color maps



(c) Visualization techniques

Figure 4.3: Class diagram of the visualization component.

The base class of scalar datasets is `ScalarDataset`. Conversely the base class of vector datasets is `VectorDataset`. There is only one type of vector dataset: namely the gradient of a scalar dataset, which is why the class `GradientDataset` is initialized with an instance of `ScalarDataset`. There are six scalar datasets in the model:

- `TnfDataset` is the dataset containing TNF- α levels;
- `Ccl2Dataset` is the dataset containing CCL2 levels;
- `Ccl5Dataset` is the dataset containing CCL5 levels;
- `Cxcl9Dataset` is the dataset containing CXCL9/10/11 levels;
- `ExtMtbDataset` is the dataset containing the number of extracellular bacteria of every micro-compartment;
- `IntMtbDataset` is the dataset containing the number of intracellular bacteria of every micro-compartment.

Using the class `ScalarNormalizer`, a scalar value within a specified range can be normalized into the range $[0, 1]$. Common color map functionality, including functionality for mapping coordinates in the HSV color space to the RGB color space, specifying the number of bands and applying a hue displacement is provided by the base class `ColorMap`. This base class also provides an abstract interface for picking a color based on a normalized scalar value. The five color maps described in Subection 3.3.1 are implemented by the following subclasses.

- `GreenRed` is the green-red color map;
- `Rainbow` is the rainbow color map;
- `GrayScale` is the gray-scale color map;
- `CoolWarm` is the cool-warm color map;
- `Fire` is the fire color map.

All implemented visualization techniques derive from the base class `Visualization`. Visualization of immune cells (as discussed in Section 3.2) is taken care of by the class `AgentsViz`. Functionality for scalar dataset visualization using color maps (cf. Subsection 3.3.1) is provided by the class `SmokeViz`. The class `GlyphsViz` implements visualization of vector datasets using glyphs. The four glyph types introduced in Section 3.4 are modeled by the classes `GlyphHedgehog`, `GlyphCone`, `GlyphArrow` and `GlyphTexture`; these classes derive from the base class `Glyph` where common functionality for translating and rotating the model is implemented. Visualization of isolines is done in the class `IsolinesViz` and a height plot is drawn in the class `HeightPlotViz`.

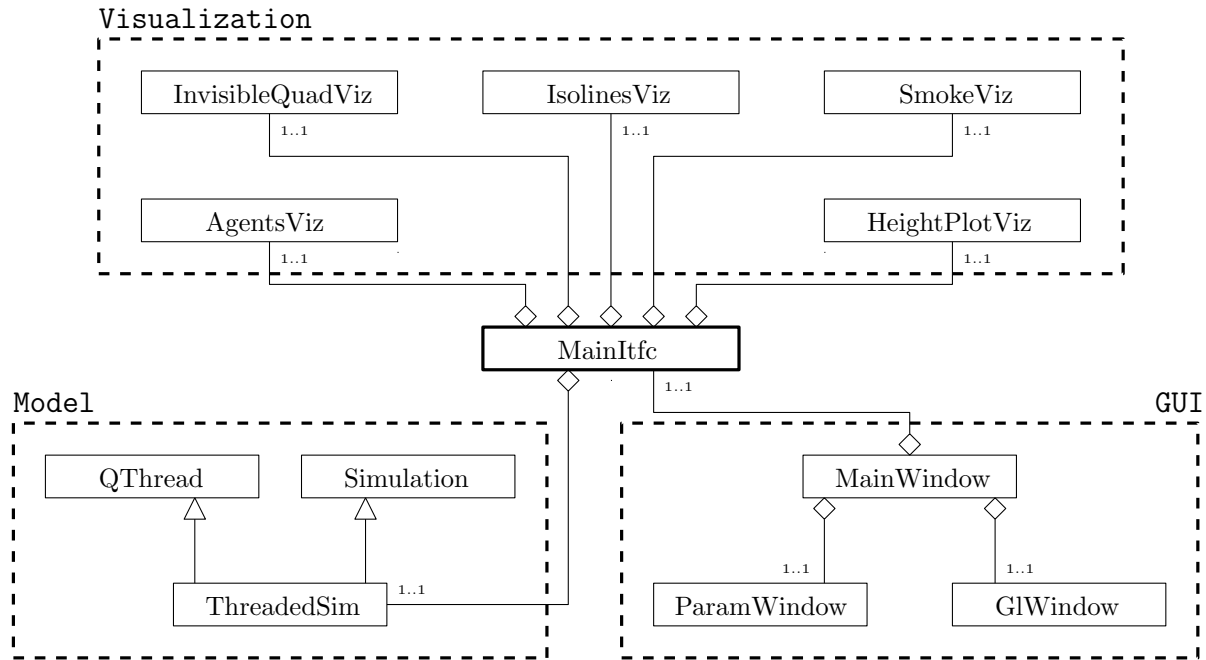


Figure 4.4: Class diagram of the GUI component.

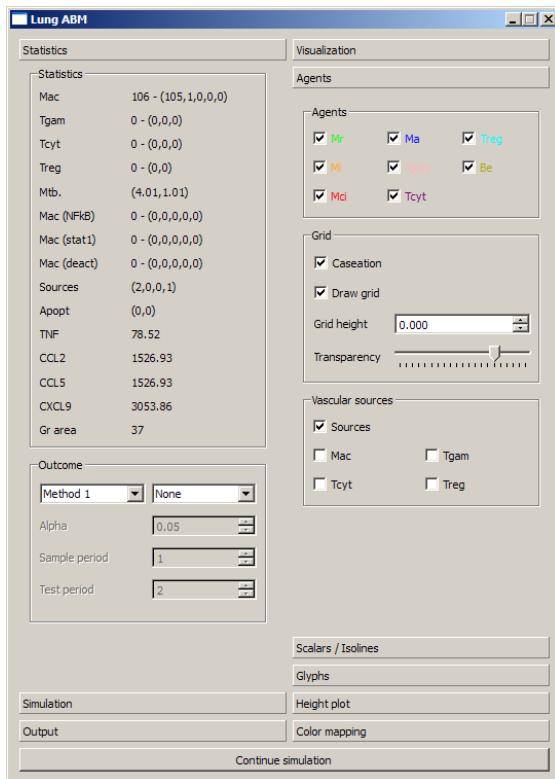
4.2.3 User interface

In Figure 4.4 the class diagram of the GUI component is given. There we can see that the class connecting the GUI component with the model and visualization components is `MainItfc`. This class contains instances of the visualization class, as well as an instance of the wrapper class `ThreadedSim`. The purpose of `ThreadedSim` is to extend the central class `Simulation` of the model component with threading functionality. This is done by inheriting from the Qt class `QThread`. In order to deal with concurrency issues, a mutex — provided by the class `QMutex` — is used.

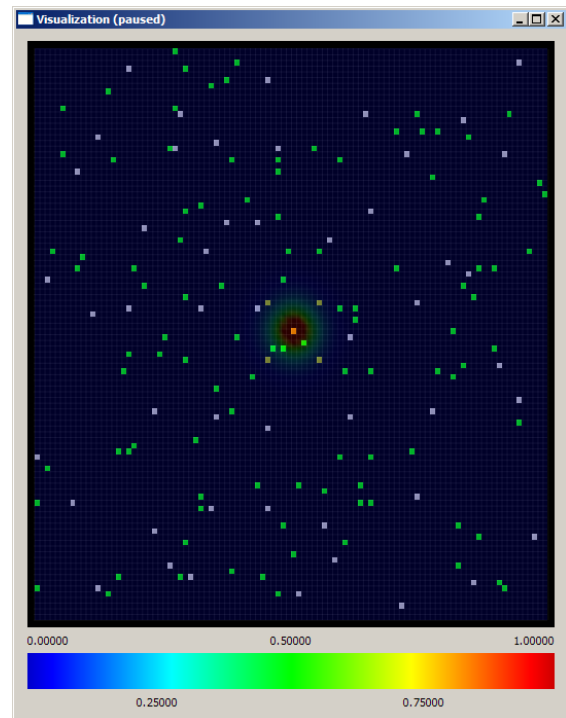
The main window is implemented by the class `MainWindow` (see Figure 4.5a). In the main window the user can configure how the visualization should proceed. Also, the user can specify the following four time intervals used for:

1. saving a snapshot of the grid (picture),
2. appending the current statistics to a CSV file,
3. saving the current model state, and
4. determining the current outcome by performing a t-test.

In the main window a timer is used to query `ThreadedSim` for new data. In case there is new data, the visualized grid is updated. The class `GIWindow` corresponds to the window where the grid is visualized using OpenGL (see Figure 4.5b). The user can pan, rotate and zoom by using the mouse on this window. Also when left-clicking on a micro-compartment, information about that micro-compartment is displayed. Since the grid can be moved to different positions, we need to map view-port coordinates back to model coordinates. As view-port coordinates lack a depth coordinate, we obtain the depth by reading out the depth buffer. Subsequently, we



(a) MainWindow



(b) GIWindow

Name	Value	Unit	Description
GR			
nrSources	50		Number of vascular sources on the grid
nrKillingsCaseation	10		Number of killings for a compartment to become caseated
diffusivityTNF	5.2e-08	cm ² /s	TNF diffusivity
diffusivityChemokines	5.2e-08	cm ² /s	Chemokine diffusivity
degRateTNF	0.9905		TNF degradation rate (per 6s)
degRateChemokines	0.9905		Chemokine degradation rate (per 6s)
thresholdApoptosisTNF	70	#molecules	TNF threshold for TNF-induced apoptosis
probApoptosisTNF	0.04		Probability of TNF-induced apoptosis
minChemotaxis	2	#molecules	Chemotaxis sensitivity range (lower bound)
maxChemotaxis	2000	#molecules	Chemotaxis sensitivity range (upper bound)
Mac			Macrophage specific parameters
maxAge	14400	#timesteps	Maximal macrophage age
maxAgeAct	1440	#timesteps	Maximal activated macrophage age
initNumber	105		Initial number of resting macrophages on the grid
maxTimeReg	71	#timesteps	Time span during which a macrophage remains down-regulated
movementRest	2	#timesteps	Time required for a resting macrophage to move one micro-compartment
movementAct	47	#timesteps	Time required for an active macrophage to move one micro-compartment
movementInf	144	#timesteps	Time required for an infected macrophage to move one micro-compartment
dTNF	1.8	#molecules/6s	Secretion rate of TNF
dCCL2	35	#molecules/6s	Secretion rate of CCL2
dCCL5	35	#molecules/6s	Secretion rate of CCL5
dCXCL9	70	#molecules/6s	Secretion rate of CXCL9

Buttons: Load Save

(c) ParamWindow

Figure 4.5: Windows of the graphical user interface.

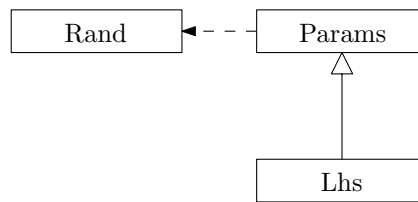


Figure 4.6: Class diagram of the LHS component.

supply these coordinates to the function `gluUnProject` and end up with model coordinates. The window where parameter values together with their descriptions can be inspected corresponds to the class `ParamWindow` (see Figure 4.5c). In this window new parameter values can be loaded — even when the model is already running.

4.2.4 LHS

The LHS component is a component that compiles to a separate executable. In Figure 4.6 we can see that only the `Params` and `Rand` classes from the model component are used. Similarly to Section 2.3, we have assumed in our implementation that all parameters are derived from a uniform distribution. The `lhs` executable takes as input an XML file containing ranges for every parameter, as well as the number of samples n and generates n XML parameter files that the model can run.

Perl scripts have been implemented to allow for automated simulation runs of parameter files generated by LHS. In each simulation run, we save at regular intervals snapshots of the grid as well as statistics. Using the resulting CSV files and snapshots, a `LATEX` report is compiled including plots — generated by `gnuplot` — of time series of various statistics.

Part III

Results

Chapter 5

Experimental Evaluation

The purpose of this chapter is twofold. Firstly, we determine in Section 5.1 baseline parameter sets resulting in containment, clearance and dissemination in the lung model; for the CNS model we determine a baseline that results in containment. Secondly, we showcase in Section 5.2 how *in silico* experiments can be performed using the baseline parameter sets.

5.1 Baseline identification

In Subsection 5.1.1 we identify baselines for the lung model, followed by a baseline parameter set resulting in containment for the CNS model in Subsection 5.1.2.

5.1.1 Lung model

In Subsection 2.1.5 we highlighted the differences between the model developed by Ray et al. and our lung model. A major difference is the use of a different method to compute diffusion. Because of this, the landscape of TNF- α and the chemokines on the grid is different from what it was previously. As a consequence, the parameter set identified by Ray et al. no longer results in containment. Therefore, in this subsection we set out to identify new parameter sets that result in containment, dissemination and clearance. Parameters that involve TNF- α or chemokine levels are varied using LHS in a systematic way as well as newly introduced parameters concerning down-regulatory mechanisms.

In Table 5.1 parameters whose values have been fixed are given. For some of these parameters, the values have been obtained from the literature:

- L1. Based on References [30, 50], we set the number of intracellular Mtb that a macrophage can accommodate $N_{M, Bi, burst}$ to 20.
- L2. Soluble TNF- α is secreted at a rate of

$$s_{TNF} = 1.8 \text{ molecules per 6 seconds [13].}$$

- L3. Soluble TNF- α has a molecular mass of 51 kDa. Using the following formula that relates the molecular mass of a molecule to its diffusion coefficient [29], we arrive at a diffusion

coefficient of:

$$D_{\text{TNF}} \approx a(M_r)^b = 1.778 \cdot 10^{-4} \cdot 51000^{-0.75} \approx 5.2 \cdot 10^{-8} \text{ cm}^2/\text{s}.$$

Note that the diffusion coefficient satisfies the stability condition formulated in Equation 2.21.

- L4. Resting macrophages move at a velocity of $1 \mu\text{m}/\text{min}$ [44]. This corresponds to moving one micro-compartment every two 10-minute time steps: $v_{\text{M,r}} = 2$ time steps.
- L5. Intracellular Mtb have a doubling time ranging from 28 to 96 hours [50]. For simplicity, we make use of a doubling time of 75 hours. This corresponds to a growth rate of $\alpha_{\text{Bi}} = 0.0015 (10 \text{ min})^{-1}$.

For other parameters the values have been kept the same as in Ray's model [33].

- R1. The probability of TNF- α -induced apoptosis is $p_{\text{TNF,apopt}} = 0.04$.
- R2. The probability of a resting macrophage to kill an extracellular bacterium is $p_{\text{M,kill}} = 0.02$.
- R3. A macrophage becomes chronically infected if the number of intracellular Mtb is half the maximal capacity, i.e. $N_{\text{M,Bi,CI}} = N_{\text{M,Bi,burst}}/2 = 10$.
- R4. The number of extracellular Mtb that an activated macrophage can kill every 10 minutes — denoted by $N_{\text{M,Be,kill}}$ — is assumed to be 10.
- R5. The probability of the cytotoxic killing of a (chronically) infected macrophage by a T_c cell is $p_{\text{c,kill}} = 0.12$.
- R6. The probability of the cytotoxic killing of a (chronically) infected macrophage in clean way by a T_c cell is $p_{\text{c,kill,cleanly}} = 0.75$.
- R7. The probability of Fas/FasL-induced apoptosis by a T_γ cell is $p_{\text{Fas,apopt}} = 0.06$.
- R8. The secretion rates of CCL2 and CCL5 are

$$s_{\text{CCL2}} = s_{\text{CCL5}} = 35 \text{ molecules per 6 seconds.}$$

- R9. The secretion rate of CXCL9/10/11 is

$$s_{\text{CXCL9}} = 71 \text{ molecules per 6 seconds.}$$

- R10. T cells show up after 20 days, i.e. $t_{\text{recr}} = 2880$ time steps.
- R11. The minimal number of resting macrophages on the grid is $N_{\text{M}} = 105$.
- R12. The probability of recruiting a macrophage is $p_{\text{M,recr}} = 0.04$.
- R13. The probability of recruiting a T cell is $p_{\text{T,recr}} = 0.1$.

R14. The probabilities of recruiting the various types of T cells are

$$\begin{aligned} p_{\gamma,\text{recr}} &= 0.54, \\ p_{c,\text{recr}} &= 0.36, \\ p_{\text{reg},\text{recr}} &= 0.1. \end{aligned}$$

Note that these probabilities sum up, as required, to 1.

R15. Infected macrophages move one micro-compartment once a day, i.e. $v_{M,i} = 144$ time steps.

R16. The probability of a T cell moving onto a micro-compartment already containing a macrophage is $p_{T,M} = 0.02$.

R17. The probability of a T cell moving onto a micro-compartment already containing another T cell is $p_{T,T} = 0.08$.

R18. Since Mtb prefer the intracellular environment of a macrophage over the extracellular environment, we assume that the growth rate of extracellular Mtb is half the intracellular growth rate: $\alpha_{Be} = \alpha_{Bi}/2 = 0.0007 (10 \text{ min})^{-1}$.

R19. A micro-compartment can accommodate 10 times the number of bacteria a macrophage can contain, i.e. $K_{Be} = 10 \cdot N_{M,Bi,\text{burst}} = 200$.

Values of the following parameters differ from Ray's parameter set:

A1. The probability of a T_γ cell to activate the transcription factor STAT1 in a macrophage is $p_{M,\text{STAT1}} = 0.08$.

A2. We assume that the modeled chemokines have the same diffusion coefficient as TNF- α , i.e. $D_{CC} = 5.2 \cdot 10^{-8} \text{ cm}^2/\text{s}$.

A3. The degradation coefficients of TNF- α and the chemokines are assumed to be

$$\delta_{\text{TNF}} = \delta_{CC} = 0.0095.$$

A4. The recruitment coefficients are chosen such that the effects of TNF- α , CCL2, CCL5 and CXCL9/10/11 are the same and independent of their levels on the grid — in other words we disregard the secretion rates:

$$\begin{aligned} \alpha_{\text{TNF}} &= 1, \\ \alpha_{\text{CCL2}} &= \frac{s_{\text{TNF}}}{s_{\text{CCL2}}} = \frac{1.8}{35} = 0.0514, \\ \alpha_{\text{CCL5}} &= \frac{s_{\text{TNF}}}{s_{\text{CCL5}}} = \frac{1.8}{35} = 0.0514, \\ \alpha_{\text{CXCL9}} &= \frac{s_{\text{TNF}}}{s_{\text{CXCL9}}} = \frac{1.8}{71} = 0.0254. \end{aligned}$$

A5. The recruitment threshold for macrophage recruitment is $\tau_{M,\text{recr}} = 0.025$.

A6. The recruitment threshold for recruitment of T_c cells is $\tau_{c,\text{recr}} = 0.4$.

A7. The recruitment threshold for recruitment of T_γ cells is $\tau_{\gamma,\text{recr}} = 0.4$.

A8. The recruitment threshold for recruitment of T_{reg} cells is $\tau_{\text{reg},\text{recr}} = 0.05$.

Table 5.1: Parameters with fixed values

Parameter	Description	Value	Note
$p_{\text{TNF,apopt}}$	Probability by which TNF- α -induced apoptosis occurs	0.04	R1
$p_{\text{M,STAT1}}$	Probability of a T_γ cell to enable STAT1 in a macrophage	0.08	A1
$p_{\text{M,kill}}$	Probability of a resting macrophage to kill an extracellular bacterium	0.02	R2
$N_{\text{M,Bi,CI}}$	Number of intracellular bacteria for an infected macrophage to become chronically infected	10	R3
$N_{\text{M,Bi,burst}}$	Number of intracellular bacteria for a chronically infected macrophage to burst	20	L1
$N_{\text{M,Be,kill}}$	Number of extracellular bacteria that an active macrophage can kill every time step	10	R4
$p_{\text{c,kill}}$	Probability of a T_c cell to kill a (chronically) infected macrophage	0.12	R5
$p_{\text{c,kill,cleanly}}$	Probability of a T_c cell to kill a chronically infected macrophage cleanly	0.75	R6
$p_{\text{Fas,apopt}}$	Probability of a T_γ cell to induce apoptosis via the Fas/-FasL pathway	0.06	R7
s_{TNF}	Maximal number of TNF- α molecules that a macrophage can secrete every 6-second time step	1.8	L2
s_{CCL2}	Maximal number of CCL2 molecules that a macrophage can secrete every 6-second time step	35	R8
s_{CCL5}	Maximal number of CCL5 molecules that a macrophage can secrete every 6-second time step	35	R8
s_{CXCL9}	Maximal number of CXCL9/10/11 molecules that a macrophage can secrete every 6-second time step	71	R9
D_{TNF}	Diffusion coefficient of TNF- α molecules	$5.2 \cdot 10^{-8}$	L3
D_{CC}	Diffusion coefficient of chemokine molecules	$5.2 \cdot 10^{-8}$	A2
δ_{TNF}	Degradation rate of TNF- α	0.0095	A3
δ_{CC}	Degradation rate of the chemokines	0.0095	A3
t_{recr}	Time after which T cell recruitment is enabled	2,880	R10
N_{M}	Minimal number of macrophages on the grid	105	R11
α_{TNF}	Effect of TNF- α on recruitment of immune cells	1	A4
α_{CCL2}	Effect of CCL2 on recruitment of immune cells	0.0514	A4
α_{CCL5}	Effect of CCL5 on recruitment of immune cells	0.0514	A4
α_{CXCL9}	Effect of CXCL9/10/11 on recruitment of immune cells	0.0254	A4
$\tau_{\text{M,recr}}$	Recruitment threshold for macrophage recruitment	0.025	A5
$\tau_{\text{c,recr}}$	Recruitment threshold for recruitment of T_c cells	0.4	A6
$\tau_{\gamma,\text{recr}}$	Recruitment threshold for recruitment of T_γ cells	0.4	A7
$\tau_{\text{reg,recr}}$	Recruitment threshold for recruitment of T_{reg} cells	0.05	A8
$p_{\text{M,recr}}$	Probability of recruiting a macrophage	0.04	R12
$p_{\text{T,recr}}$	Probability of recruiting a T cell	0.1	R13
$p_{\text{c,recr}}$	Probability of recruiting a T_c cell	0.36	R14
$p_{\gamma,\text{recr}}$	Probability of recruiting a T_γ cell	0.54	R14
$p_{\text{reg,recr}}$	Probability of recruiting a T_{reg} cell	0.1	R14

Continued on next page

Table 5.1 – continued from previous page

Parameter	Description	Value	Note
$v_{M,r}$	Time required for a resting macrophage to move one micro-compartment	2	L4
$v_{M,i}$	Time required for an infected macrophage to move one micro-compartment	144	R15
$p_{T,M}$	Probability of moving a T cell onto a micro-compartment containing a macrophage	0.02	R16
$p_{T,T}$	Probability of moving a T cell onto a micro-compartment already containing a T cell	0.08	R17
α_{Bi}	Growth rate of intracellular bacteria	0.0015	L5
α_{Be}	Growth rate of extracellular bacteria	0.0007	R18
K_{Be}	Number of extracellular bacteria a micro-compartment can accommodate	200	R19

In Table 5.2 the parameters that have been varied in the LHS are given. The ranges that we chose for these parameters are as follows:

- S1. The number of killings for a micro-compartment to become caseated, N_{caseous} , is varied between 5 and 10.
- S2. Based on observed TNF- α levels on the grid, the TNF- α level required for TNF- α -induced apoptosis, $\tau_{\text{TNF,apopt}}$, is varied between [65, 75].
- S3. Since NF κ B activation due to TNF- α occurs more frequently than TNF- α -induced apoptosis, the range of $\tau_{M,\text{TNF},\text{NF}\kappa\text{B}}$ is chosen to be [50, 60].
- S4. The second mechanism for NF κ B activation is extracellular Mtb. We vary the parameter $N_{M,\text{Be},\text{NF}\kappa\text{B}}$ in the range [10, 300].
- S5. A macrophage can remain down-regulated for 1 hour up to 1 day, i.e. $\sigma_{M,\text{deact}} \in [6, 144]$.
- S6. Since T cells do not live as long as macrophages (3 days opposed to 100 days), they remain down-regulated for a shorter time span of at most 10 hours: $\sigma_{c,\text{deact}}, \sigma_{\gamma,\text{deact}} \in [6, 60]$.
- S7. Activated macrophages move at a faster rate than infected macrophages. The range that we use for $v_{M,a}$ is [6, 100].

Table 5.2: Parameters varied in the LHS

Parameter	Description	Range	Note
N_{caseous}	Number of killings for a micro-compartment to become caseous	[5, 10]	S1
$\tau_{\text{TNF,apopt}}$	TNF- α level required for TNF- α -induced apoptosis	[65, 75]	S2
$\tau_{M,\text{TNF},\text{NF}\kappa\text{B}}$	TNF- α level required for NF κ B activation in macrophages	[50, 60]	S3

Continued on next page

Table 5.2 – continued from previous page

Parameter	Description	Value	Note
$N_{M,Be,NF\kappa B}$	Number of extracellular bacteria in the Moore neighborhood required for NF κ B activation in macrophages	[10, 300]	S4
$\sigma_{M,deact}$	Time span during which a macrophage remains deactivated	[6, 144]	S5
$\sigma_{c,deact}$	Time span during which a T_c cell remains deactivated	[6, 60]	S6
$\sigma_{\gamma,deact}$	Time span during which a T_γ cell remains deactivated	[6, 60]	S6
$v_{M,a}$	Time required for an active macrophage to move one micro-compartment	[6, 100]	S7

We performed several LHS experiments with parameter values according to Tables 5.1 and 5.2. The initial inoculum consisted of one extracellular bacterium and four macrophages infected with one intracellular bacterium each. Ultimately, we were able to obtain parameter sets resulting in clearance, containment and dissemination. The values of these parameter sets are given in Table 5.3.

Table 5.3: Parameter values resulting in containment, clearance and dissemination.

	Containment	Dissemination	Clearance
$N_{caseous}$	10	10	10
$\tau_{TNF,apopt}$	70	70	70
$\tau_{M,TNF,NF\kappa B}$	51	51	51
$N_{M,Be,NF\kappa B}$	11.6435	128.326	114.367
$\sigma_{M,deact}$	72	110	55
$\sigma_{c,deact}$	33	7	39
$\sigma_{\gamma,deact}$	36	12	32
$v_{M,a}$	31	79	11

In Figure 5.1 a run with the containment parameter set is given. In the plots we can see that a stable granuloma, i.e. a granuloma whose area remains constant over time, is formed and that bacterial numbers remain low. There is a spike in bacterial numbers at day 20; the reason for this is that then adaptive immunity is initiated by T cells appearing on the grid causing bacterial numbers to drop. We can also see that immune cells numbers remain constant after day 75.

A run with the parameter set resulting in dissemination is given in Figure 5.2. There we can see that the granuloma area as well as the number of immune cells and bacteria continue to increase over time. If we compare parameter sets for containment and dissemination (cf. Table 5.3), we observe that activated macrophages move at a slower pace in the dissemination scenario than in the containment scenario. Also, the number of extracellular bacteria needed for NF κ B activation is much higher in the dissemination parameter set than in the containment parameter set. So due to the parameter values, macrophages are able to do a better job at killing Mtb in the containment scenario than in the dissemination scenario.

Grid snapshots and plots of the parameter set that results in clearance are given in Figure 5.3. In the plots we can see that at day 90 there are no more bacteria, and from that point onwards the immune cell numbers decrease and the granuloma continues to shrink until it dissolves at day 165. From the three parameter sets, activated macrophages move at the fastest pace in the clearance parameter set. This may explain why that parameter set results in clearance.

5.1.2 CNS model

The parameter values that we use for the CNS model are largely based on the parameter values used in the previous subsection (Table 5.1). For parameters specific to the CNS model the following values were used:

- A1. The minimal number of meningeal macrophages is $N_M = 2000$.
- A2. The minimal number of ramified microglia on the grid is also $N_{m,ram} = N_M = 2000$.
- A3. The probability of a resting microglial cell to kill an extracellular bacterium is similar to that of a macrophage: $p_{m,kill} = p_{M,kill} = 0.02$.
- A4. Since microglia are typically smaller than macrophages, we set the number of intracellular bacteria needed to half that needed for a macrophage to become chronically infected: $N_{m,Bi,CI} = N_{M,Bi,CI}/2 = 5$.
- A5. Similarly, a microglial cell bursts when the number of intracellular bacteria is $N_{m,Bi,burst} = N_{M,Bi,burst}/2 = 10$.
- A6. The number of extracellular bacteria an activated microglial cell can kill is $N_{m,Be,kill} = N_{M,Be,kill}/2 = 5$.
- A7. Just like a resting macrophage, a resting microglial cells moves once every two 10-minute time steps: $v_{m,r} = v_{M,r} = 2$.
- A8. The same applies for the movement of infected microglia, they also move once a day: $v_{m,i} = v_{M,i} = 144$.
- A9. The probability of moving a microglial cell onto a micro-compartment already containing a T cell is $p_{T,m} = p_{T,M} = 0.02$.

Table 5.4: CNS parameters with fixed values

Parameter	Description	Value	Note
N_M	Minimal number of meningeal macrophages on the grid	2000	A1
$N_{m,ram}$	Minimal number of ramified microglia on the grid	2000	A2
$p_{m,kill}$	Probability of a resting microglial cell to kill an extracellular bacterium	0.02	A3
$N_{m,Bi,CI}$	Number of intracellular bacteria for an infected microglial cell to become chronically infected	5	A4
$N_{m,Bi,burst}$	Number of intracellular bacteria for a chronically infected microglial cell to burst	10	A5

Continued on next page

Table 5.4 – continued from previous page

Parameter	Description	Value	Note
$N_{m,Be,kill}$	Number of extracellular bacteria that an active microglial cell can kill every time step	5	A6
$v_{m,r}$	Time required for a resting microglial cell to move one micro-compartment	2	A7
$v_{m,i}$	Time required for an infected microglial cell to move one micro-compartment	144	A8
$p_{T,m}$	Probability of moving a T cell to a micro-compartment containing a microglial cell	0.02	A9

The parameters that have been varied in the LHS are given in Table 5.5. The ranges that we used for these parameters are as follows:

- S1. The number of killings for a micro-compartment to become caseated, $N_{caseous}$, is varied between 5 and 10.
- S2. Based on observed TNF- α levels on the grid, the TNF- α level required for TNF- α -induced apoptosis, $\tau_{TNF,apopt}$, is varied between [65, 75].
- S3. Since NF κ B activation due to TNF- α occurs more frequently than TNF- α -induced apoptosis, the ranges of $\tau_{M,TNF,NF\kappa B}$ and $\tau_{m,TNF,NF\kappa B}$ are chosen to be [50, 60].
- S4. The second mechanism for NF κ B activation is extracellular Mtb. We vary the parameters $N_{M,Be,NF\kappa B}$ and $N_{m,Be,NF\kappa B}$ in the range [10, 300].
- S5. A macrophage can remain down-regulated for 1 hour up to 1 day, i.e. $\sigma_{M,deact} \in [6, 144]$.
- S6. A microglial cell can remain down-regulated for 1 hour up to 1 day, i.e. $\sigma_{m,deact} \in [6, 144]$.
- S7. Since T cells do not live as long as macrophages (3 days opposed to 100 days), they remain down-regulated for a shorter time span of at most 10 hours: $\sigma_{c,deact}, \sigma_{\gamma,deact} \in [6, 60]$.
- S8. Activated macrophages and microglia move at a faster rate than their infected counterparts. The ranges that we use for $v_{M,a}$ and $v_{m,a}$ are [6, 100].
- S9. The TNF- α level required for reactivation of microglia, $\tau_{m,TNF,react}$, is varied in the range [0.01, 0.2].
- S10. The probabilities of a T_{γ} cell to enable STAT1 in a macrophage or microglial cell — $p_{M,STAT1}$ and $p_{m,STAT1}$, respectively — are varied in the range [0.01, 0.08].
- S11. The number of reactive (either resting or active) microglia that can proliferate every time step, $N_{m,react,prol}$, is in the range [10, 100].
- S12. The TNF- α level required for proliferation of reactive (either resting or active) microglia, $\tau_{m,TNF,prol}$, is in the range [0.1, 2].

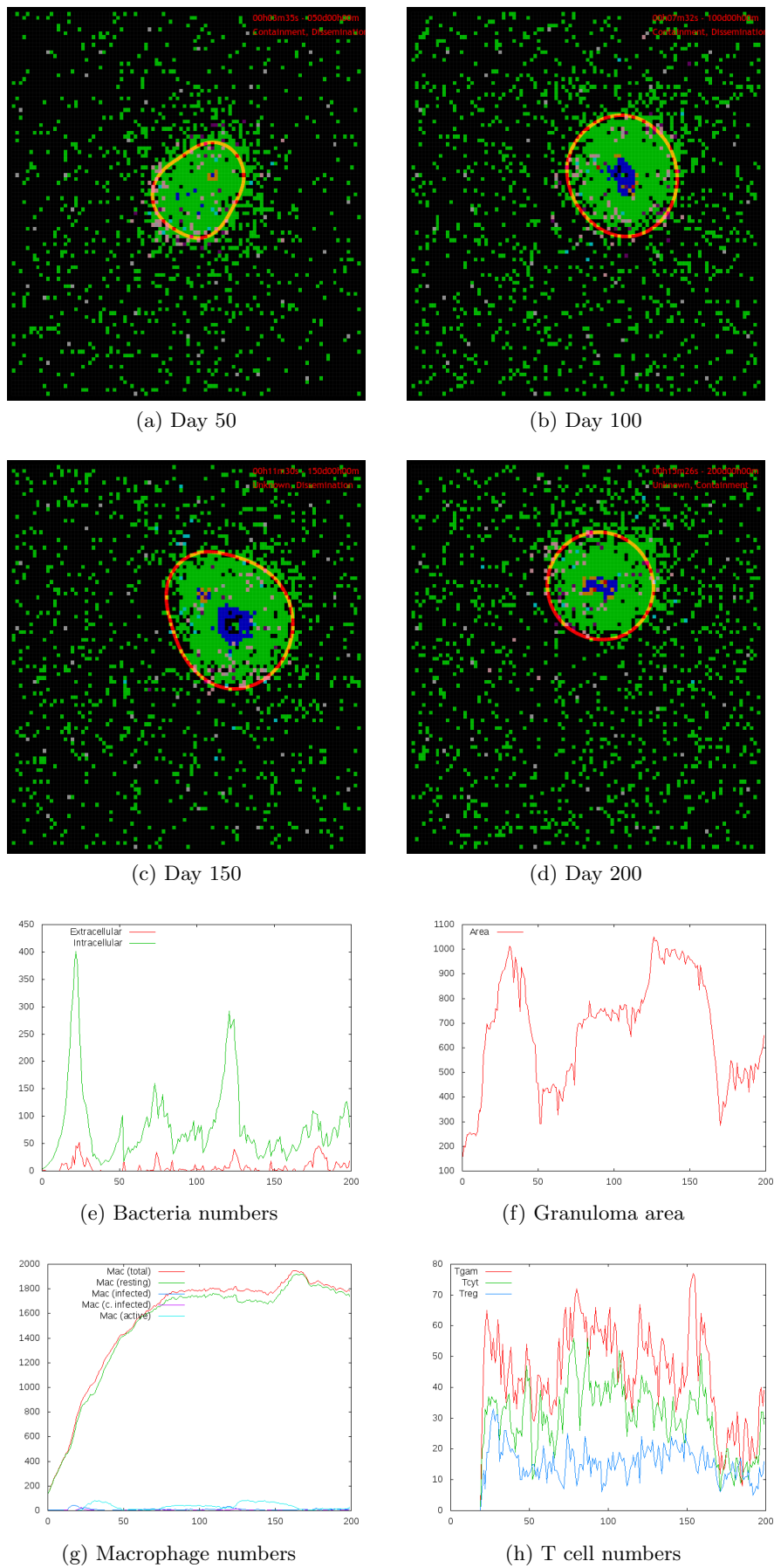


Figure 5.1: Containment in the lung model.

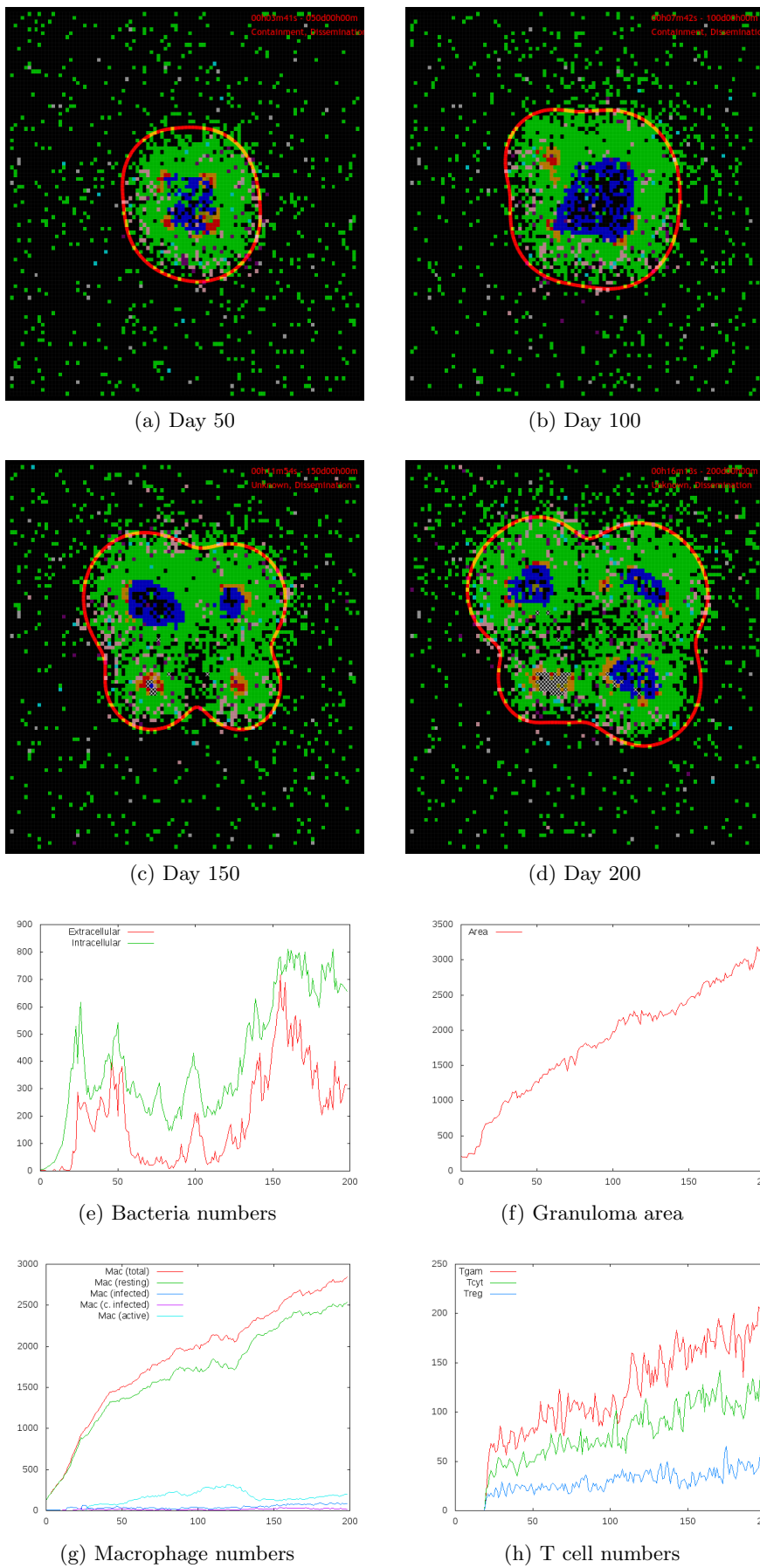


Figure 5.2: Dissemination in the lung model.

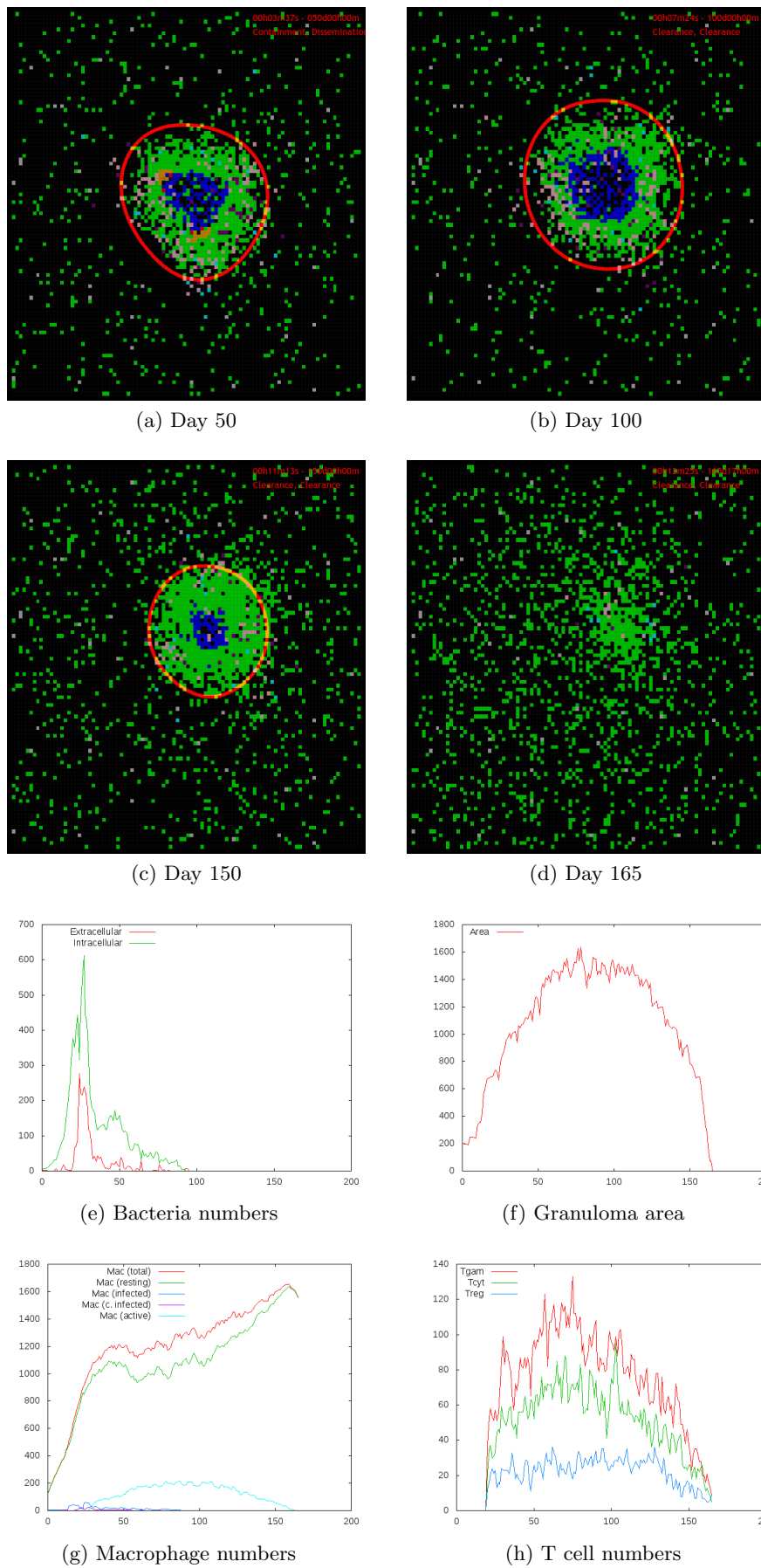


Figure 5.3: Clearance in the lung model.

Table 5.5: Parameters varied in the LHS

Parameter	Description	Range	Note
N_{caseous}	Number of killings for a micro-compartment to become caseous	[5, 10]	S1
$\tau_{\text{TNF,apopt}}$	TNF- α level required for TNF- α -induced apoptosis	[65, 75]	S2
$\tau_{\text{M,TNF,NF}\kappa\text{B}}$	TNF- α level required for NF κ B activation in macrophages	[50, 60]	S3
$N_{\text{M,Be,NF}\kappa\text{B}}$	Number of extracellular bacteria in the Moore neighborhood required for NF κ B activation in macrophages	[10, 300]	S4
$\tau_{\text{m,TNF,NF}\kappa\text{B}}$	TNF- α level required for NF κ B activation in microglia	[50, 60]	S3
$N_{\text{m,Be,NF}\kappa\text{B}}$	Number of extracellular bacteria in the Moore neighborhood required for NF κ B activation in microglia	[10, 300]	S4
$\sigma_{\text{M,deact}}$	Time span during which a macrophage remains deactivated	[6, 144]	S5
$\sigma_{\text{m,deact}}$	Time span during which a microglial cell remains deactivated	[6, 144]	S6
$\sigma_{\text{c,deact}}$	Time span during which a T_c cell remains deactivated	[6, 60]	S7
$\sigma_{\gamma,\text{deact}}$	Time span during which a T_γ cell remains deactivated	[6, 60]	S7
$v_{\text{M,a}}$	Time required for an active macrophage to move one micro-compartment	[6, 100]	S8
$v_{\text{m,a}}$	Time required for an active microglial cell to move one micro-compartment	[6, 100]	S8
$\tau_{\text{m,TNF,react}}$	TNF- α level required for a ramified microglial cell to become resting	[0.01, 2]	S9
$p_{\text{M,STAT1}}$	Probability of a T_γ cell to enable STAT1 in a macrophage	[0.01, 0.08]	S10
$p_{\text{m,STAT1}}$	Probability of a T_γ cell to enable STAT1 in a microglial cell	[0.01, 0.08]	S10
$N_{\text{m,react,prol}}$	Number of reactive (either resting or active) microglia that can proliferate every time step	[10, 100]	S11
$\tau_{\text{m,TNF,prol}}$	TNF- α level required for proliferation of reactive (either resting or active) microglia	[0.1, 2]	S12

We performed an LHS with parameter ranges and values as in Tables 5.4 and 5.5. The initial inoculum consisted of two infection sites containing two macrophages and one microglial cell each infected with one intracellular bacterium. The number of samples we simulated was 50. Out of the 50 runs we performed, several resulted in containment; we classified one of these as containment. In Table 5.6 the parameter values of this run are given.

Table 5.6: Parameters values resulting in containment.

Parameter	Value
N_{caseous}	9
Continued on next page	

Table 5.6 – continued from previous page

Parameter	Value
$\tau_{\text{TNF,apopt}}$	70
$\tau_{\text{M,TNF,NF}\kappa\text{B}}$	51
$N_{\text{M,Be,NF}\kappa\text{B}}$	211.513
$\tau_{\text{m,TNF,NF}\kappa\text{B}}$	51
$N_{\text{m,Be,NF}\kappa\text{B}}$	160.314
$\sigma_{\text{M,deact}}$	62
$\sigma_{\text{m,deact}}$	27
$\sigma_{\text{c,deact}}$	40
$\sigma_{\gamma,\text{deact}}$	54
$v_{\text{M,a}}$	7
$v_{\text{m,a}}$	9
$\tau_{\text{m,TNF,react}}$	1.67308
$p_{\text{M,STAT1}}$	0.0670123
$p_{\text{m,STAT1}}$	0.0520861
$N_{\text{m,react,prol}}$	6
$\tau_{\text{m,TNF,prol}}$	0.773001

The plots and snapshots of the parameter set corresponding to the containment run are given in Figure 5.4. In the plots we can see that the bacterial numbers remain low and are in a steady state. The granuloma area on the other hand keeps on increasing. Nonetheless, we classified this parameter set as containment because bacteria are controlled.

5.2 In silico experiments

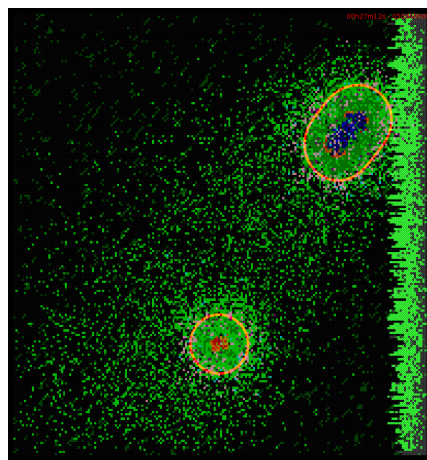
With the developed models *in silico* experiments can be performed. In this section we describe several of such experiments.

5.2.1 Lung model

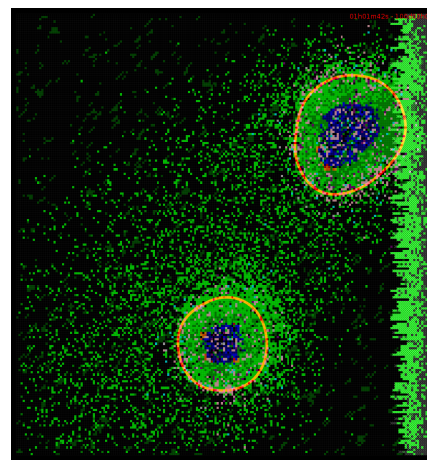
Using the baseline parameter set identified in Subsection 5.1.1 and depicted in Figure 5.1, we perform several in silico depletion and knock-out experiments. Note that throughout our simulation experiments we made sure that the same seed of the random number generator was used as in the containment scenario; as such vascular sources were distributed in the same way.

We start by simulating a knock-out of TNF- α by setting the secretion rate $s_{\text{TNF}} = 0$. By knocking-out TNF- α , we effectively disable all TNF- α related mechanisms including TNF- α -induced apoptosis and NF κ B activation due to TNF- α . In Figure 5.5 the results of the knock-out experiment are given. Recall that determining the granuloma area depends on TNF- α levels on the grid. Since due to the knock-out no TNF- α is present on the grid, the granuloma area could not be determined as we can see in Figure 5.5f.

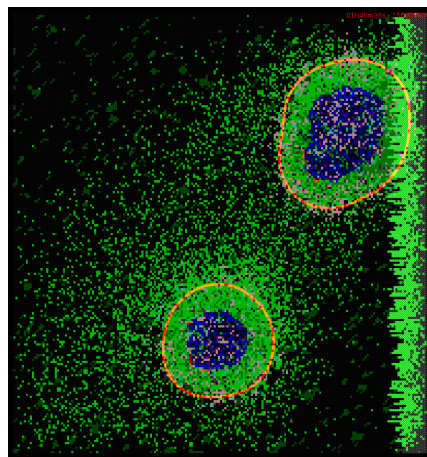
The bacterial numbers in the TNF- α knock-out scenario are much higher than in the baseline scenario. Also we can see in the plot of Figure 5.5e that the bacterial numbers do not stabilize



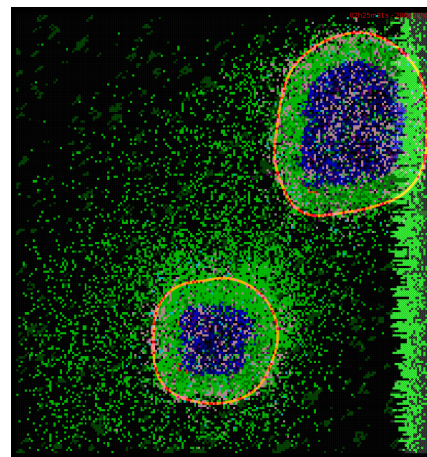
(a) Day 50



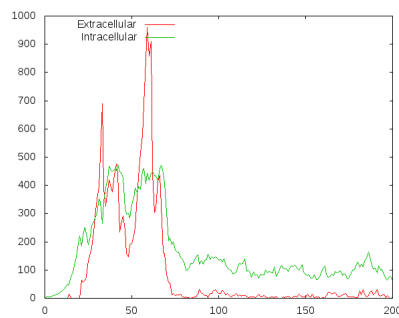
(b) Day 100



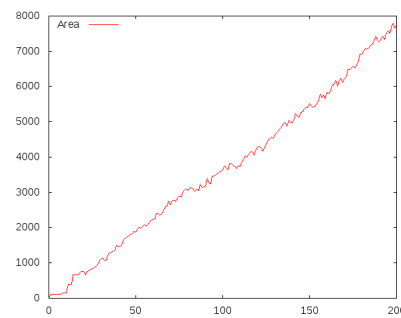
(c) Day 150



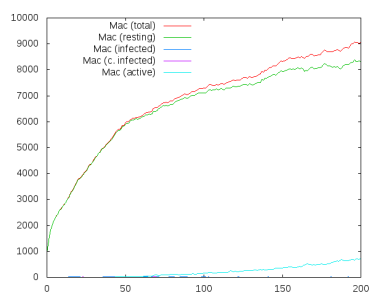
(d) Day 200



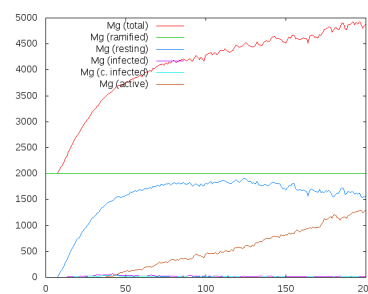
(e) Bacteria numbers



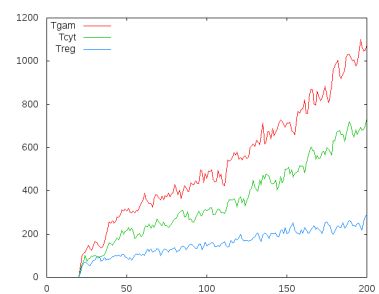
(f) Granuloma area



(g) Macrophage numbers



(h) Microglia numbers



(i) T cell numbers

Figure 5.4: Containment in the CNS model.

and keep increasing over time. If we compare the snapshots of the grids in Figures 5.1 and 5.5, we can see that in the knock-out scenario there is a lot more caseation. An explanation for this might be that infected and chronically infected macrophages killed as a result of TNF- α -induced apoptosis do *not* contribute to caseation. When disabling this mechanism, an infected or chronically infected macrophage can only die as a result of the following: (1) natural age, (2) become active and die due to a shortened lifespan, (3) bursting, (4) cytotoxic killing by a T_c cell and (5) Fas/FasL apoptosis induced by a T_γ cell. Except for death by natural age, all mentioned cases contribute to caseation. So by disabling TNF- α -induced apoptosis, it is much more likely to have caseation.

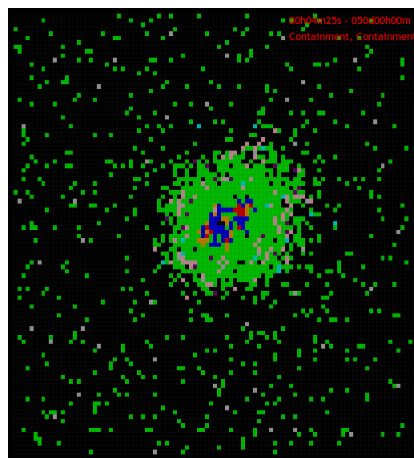
In our models we chose not to represent IFN- γ as a separate cytokine. Rather we decided to use T_γ cells as a proxy for this cytokine. We did this by allowing T_γ cells to activate the transcription factor STAT1 of macrophages in their proximity. We can simulate a knock-out of IFN- γ by setting the probability $p_{M,STAT1} = 0$. This effectively disables the mechanism of STAT1 activation by T_γ cells. Since STAT1 is a necessary condition for macrophage activation, we expect that no activated macrophages will be present on the grid. This is indeed the case if we look at the plot of Figure 5.6g. Furthermore, if we look at snapshots of the grid and the plots of bacterial numbers and granuloma area in Figure 5.6, we can see that there is an uncontrolled growth of Mtb: the granuloma area as well as bacterial numbers increase steadily over time. So based on this experiment, we can conclude that IFN- γ plays a major role in containing the bacteria.

In contrast with a knock-out, a depletion of TNF- α at day 100 is performed by setting the corresponding parameter to 0 at day 100 rather than from day 0. In Figure 5.7 the results are shown. From Figure 5.7f we can indeed see that TNF- α is knocked-out from day 100 onwards (recall that the granuloma area is determined by taking TNF- α levels into account). We can also see in Figure 5.7e that from day 100 onwards bacterial numbers increase steadily. This is to be expected when considering the results of the TNF- α knock-out experiment (cf. Figure 5.5) where we saw that bacterial numbers kept on increasing as opposed to the baseline scenario.

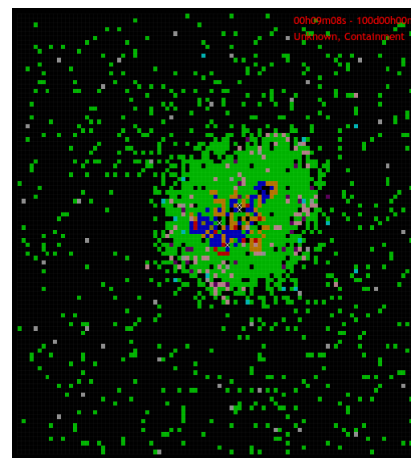
The second depletion experiment we performed concerns the cytokine IFN- γ . Similarly to the previous depletion experiment, we set $p_{M,STAT1} = 0$ from day 100 onwards. In Figure 5.8 the results of this experiment are given. We can see in Figure 5.8g that from day 100 onwards the number of activated macrophages drops to 0. There is also a dramatic increase of extracellular bacterial numbers from day 100 onwards. So clearly IFN- γ is a very important factor in containing Mtb.

5.2.2 CNS model

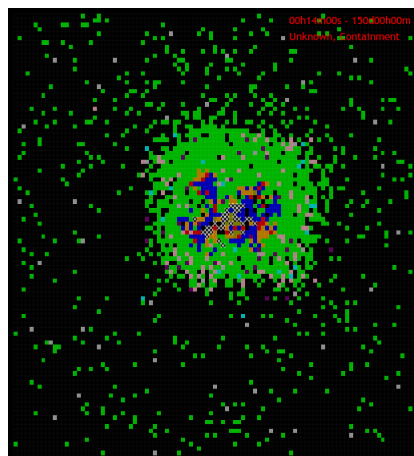
Just like with the lung model, *in silico* knock-out and depletion experiments can be performed in the CNS model as well. In this section we will illustrate one such experiment by knocking-out the cytokine IFN- γ . In the CNS model, there are two parameters concerning IFN- γ namely $p_{M,STAT1}$ and $p_{m,STAT1}$. We simulate a knock-out by setting both parameters to 0. The results of the knock-out are given in Figure 5.9. There we can see, as opposed to the baseline scenario (cf. Figure 5.4), that there are no activated macrophages and microglia and there is a steady increase in bacterial numbers and granuloma area over time.



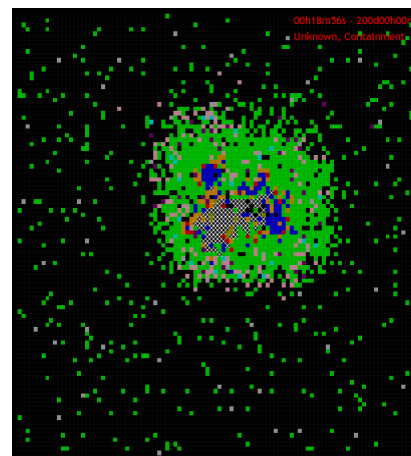
(a) Day 50



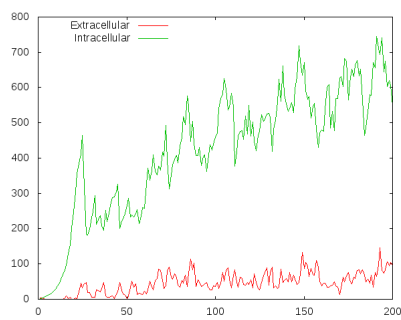
(b) Day 100



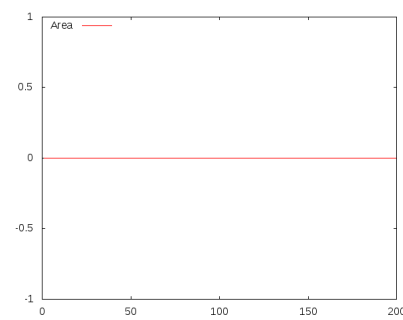
(c) Day 150



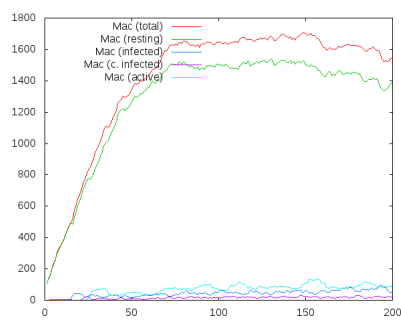
(d) Day 200



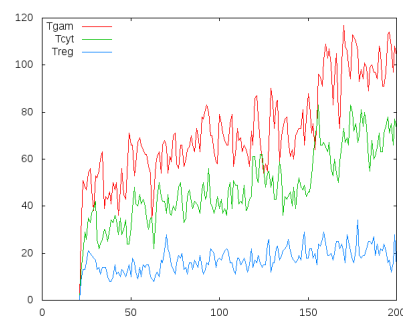
(e) Bacteria numbers



(f) Granuloma area

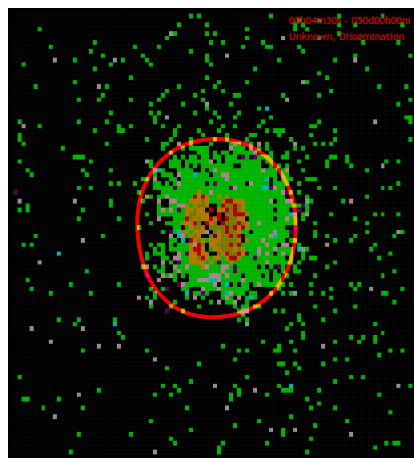


(g) Macrophage numbers

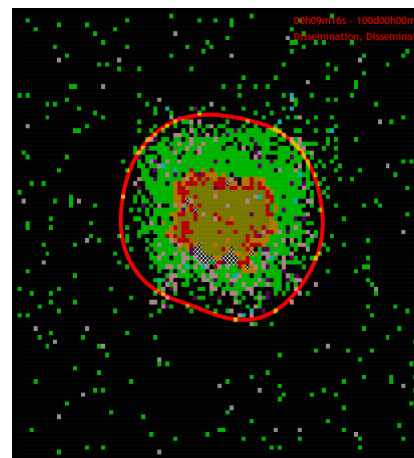


(h) T cell numbers

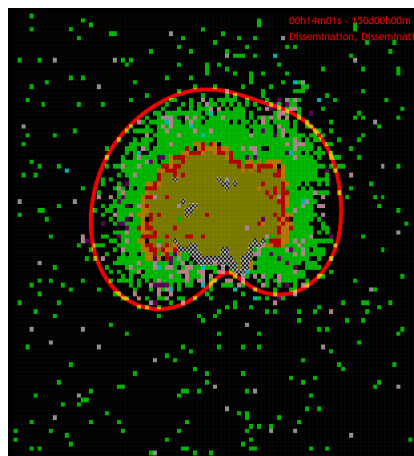
Figure 5.5: TNF- α knock-out in the lung model.



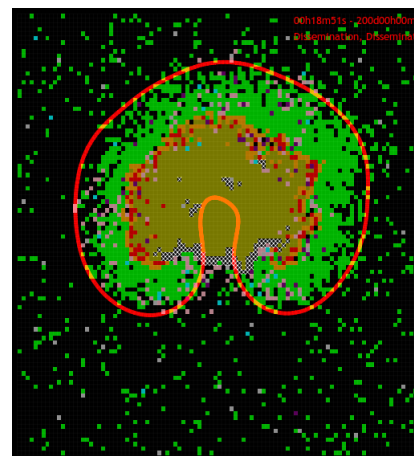
(a) Day 50



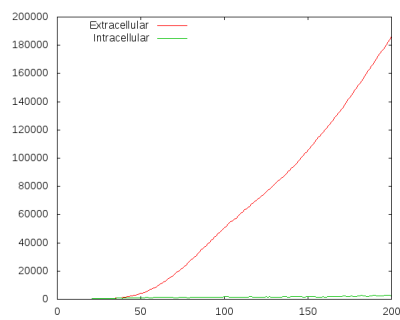
(b) Day 100



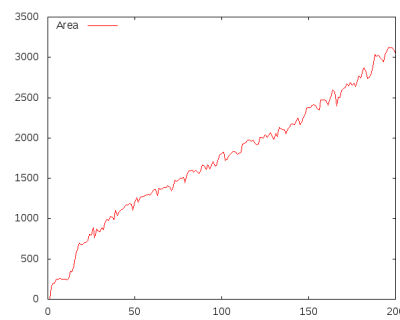
(c) Day 150



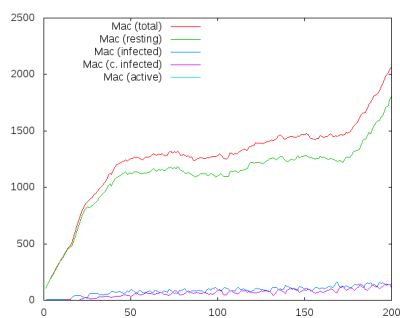
(d) Day 200



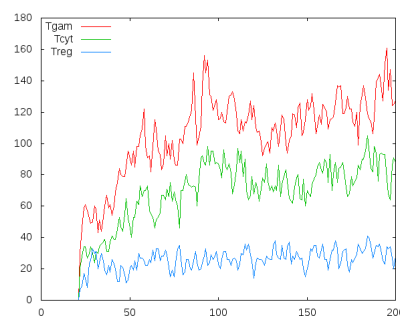
(e) Bacteria numbers



(f) Granuloma area

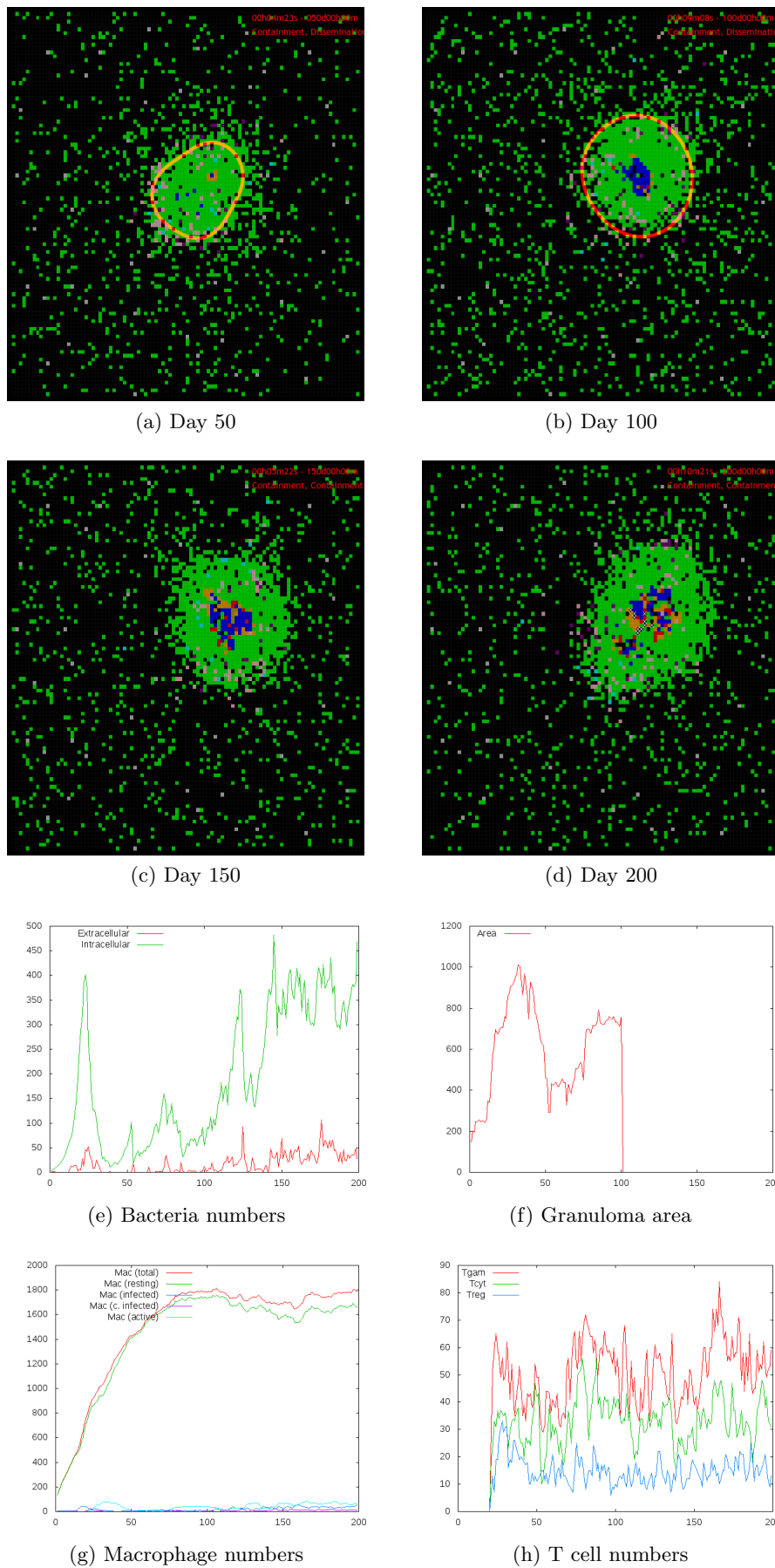


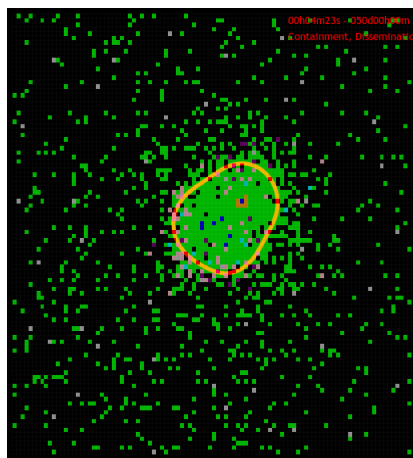
(g) Macrophage numbers



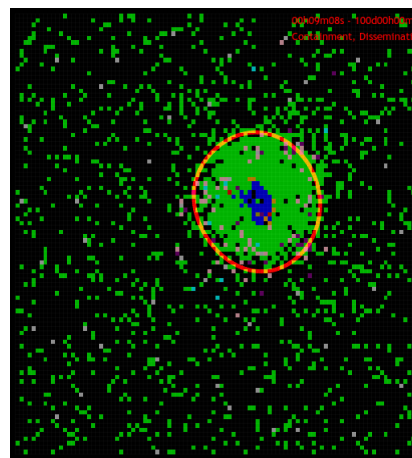
(h) T cell numbers

Figure 5.6: IFN- γ knock-out in the lung model.

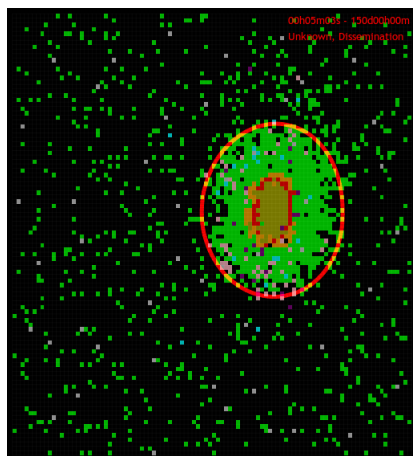
Figure 5.7: TNF- α depletion at day 100 in the lung model.



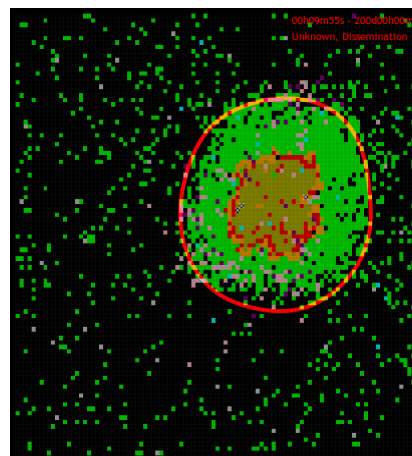
(a) Day 50



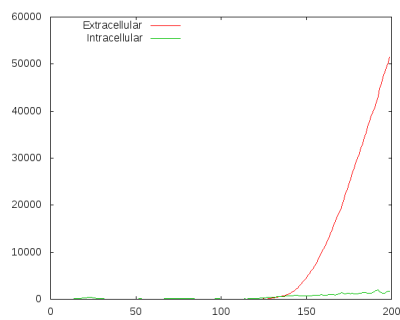
(b) Day 100



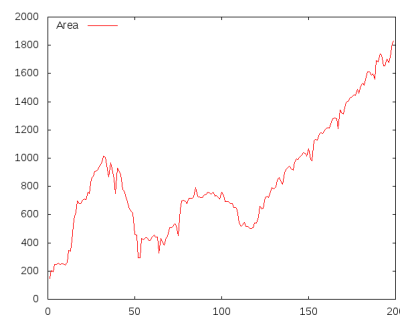
(c) Day 150



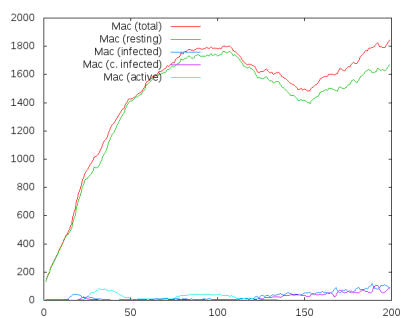
(d) Day 200



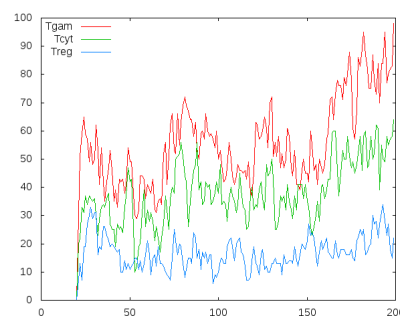
(e) Bacteria numbers



(f) Granuloma area

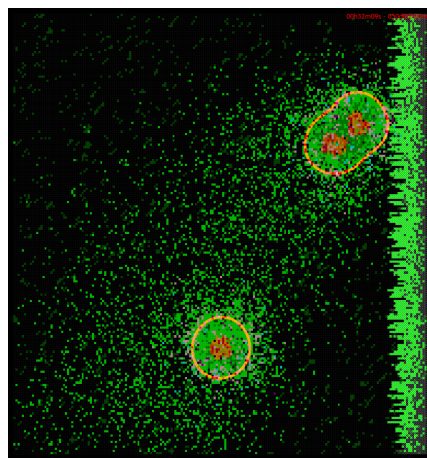


(g) Macrophage numbers

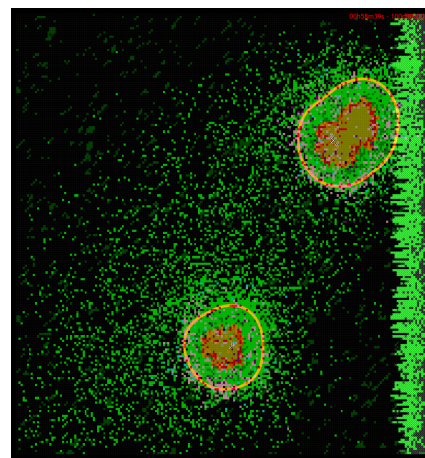


(h) T cell numbers

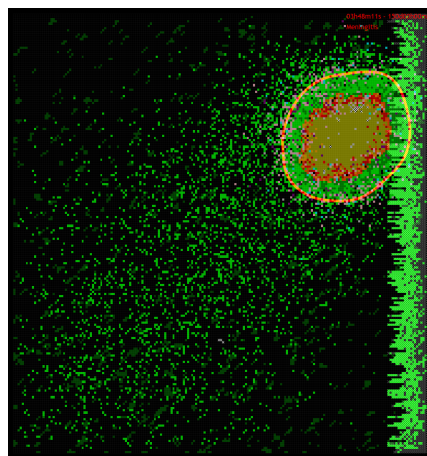
Figure 5.8: IFN- γ depletion at day 100 in the lung model.



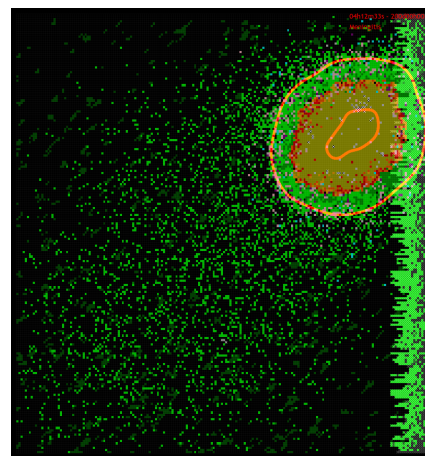
(a) Day 50



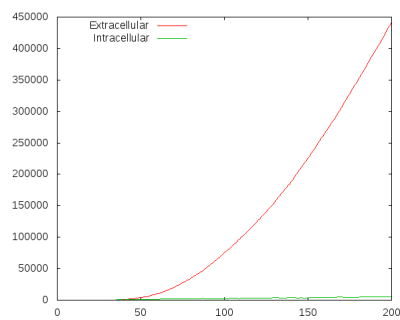
(b) Day 100



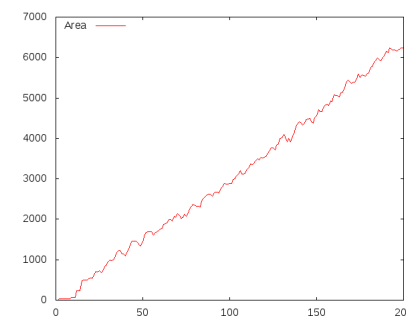
(c) Day 150



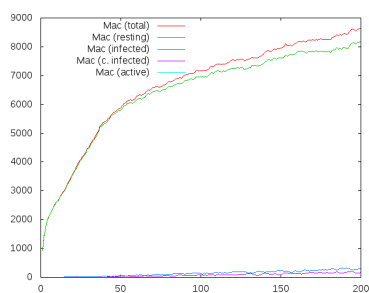
(d) Day 200



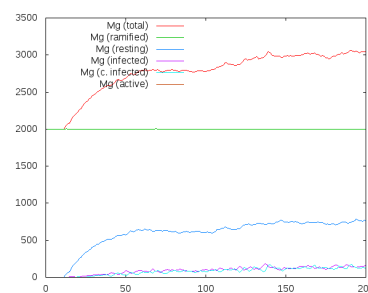
(e) Bacteria numbers



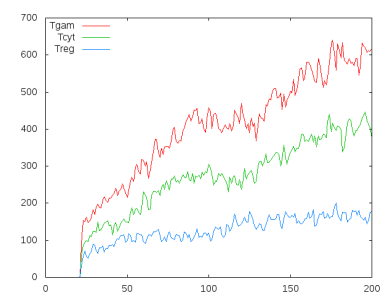
(f) Granuloma area



(g) Macrophage numbers



(h) Microglia numbers



(i) T cell numbers

Figure 5.9: IFN- γ knock-out in the CNS model.

Chapter 6

Conclusion and Discussion

In this thesis we explored pulmonary tuberculosis (TB) as well as tuberculous meningitis (TBM). The latter is a very serious disease that is hard to diagnose and treat. Even when treated, the disease often results in irreversible neurological damage. Both pulmonary TB and TBM are caused by the pathogen *Mycobacterium tuberculosis*. This bacterium has been around since ancient times. It was not until 50 years ago that effective antibiotic treatment was discovered. An effective vaccine, however, is yet to be found. Now with the appearance of multi-drug resistant TB and increasing incidence rates due to HIV coinfection, the need for effective treatment and prevention strategies is even more urgent. In order to derive such strategies, it is imperative to have a thorough understanding of the pathogenesis and immune response in TB and TBM. One way to improve our understanding is by means of mathematical modeling.

The modeling technique that we employed in this thesis is agent-based modeling. With agent-based modeling, autonomous agents residing in a predefined environment make decisions based on a set of rules taking into account the agent's state and its neighborhood. In our case the agents are immune cells such as macrophages, microglia and various types of T cells. It is through the interplay of the immune cells, that we can observe in our models the formation of a granuloma — the hallmark of TB infection.

In Chapter 2 we introduced two agent-based models. We started by describing and extending an existing agent-based model of pulmonary TB. A first time ever model describing the immune response and pathogenesis of TBM was presented in Section 2.2. A method specifically geared toward uncertainty analysis in mathematical models was described in Section 2.3. In that same section we also described how Welch's t-test can be used to determine the model outcome by taking into account the granuloma area and bacterial numbers. We introduced various visualization techniques in Chapter 3 including color mapping, vector visualization using glyphs, isolines and height plots. These visualization techniques were employed to obtain a better overview of a model simulation. Requirements and implementation details of the developed software were discussed in Chapter 4. In Chapter 5 both models were subjected to an experimental evaluation. We were able to obtain parameter sets resulting in containment, dissemination and clearance in the lung model as well as a parameter set resulting in containment in the CNS/TBM model. Using the identified baselines, several *in silico* experiments were performed in Section 5.2 including TNF- α and IFN- γ knock-out and depletion experiments. Using simulations, we confirmed the major role that IFN- γ plays in TB infection.

6.1 Discussion and future work

Open questions CNS model

In order to arrive at a model of the pathogenesis of TBM, we had to make assumptions about the following questions, as we could not find conclusive answers in the literature.

- Which cytokine(s) is mainly responsible for inducing proliferation of microglia? In the model we assumed this cytokine to be TNF- α .
- What is the typical thickness of the meninges? Meninges in our model are located on the right edge of the grid and are 0.3-0.4 mm thick.
- How many meningeal macrophages are present on the meninges? This corresponds to the value of the parameter N_M . In our baseline parameter set, we assumed that there are macrophages on roughly half of the meningeal micro-compartments.
- What is the lifespan of a microglial cell? We assumed that microglial cells live for at most 100 days.
- Can reactivated microglia become ramified again? In the model once a ramified microglial cell becomes reactivated, it will always remain in that state.
- How does the population of ramified microglia develop during infection? We assumed that this population is always in a steady state by maintaining the number of such microglia $N_{m,ram} = 2000$.
- What is the proliferation rate of reactivated microglial cells? We assumed that every 10-minute time step $N_{m,react,prol} = 6$ reactive microglial cells proliferate.

Finding conclusive answers to the above questions will result in a more accurate model. Also a more extensive uncertainty analysis needs to be performed in order to establish baseline scenarios for clearance and dissemination as well. Such an uncertainty analysis may be followed by a sensitivity analysis as to quantify the extent of the effect of various model parameters on the outcome.

Asynchronicity

Our models are based on discrete time steps. In addition, immune cells change state in a synchronous fashion: first all new states are computed and saved in a temporary store followed by copying the new states to the cells. Movement of immune cells happens semi-synchronously: first all microglia are moved followed by movement of macrophages, T_γ cells, T_c cells and finally T_{reg} cells. Within each type of immune cell, movement happens asynchronously. However, the order by which the immune cells move is fixed: the lists of immune cells of each type are ordered by time of appearance on the grid of the immune cells, and movement occurs according to this ordering. A similar thing applies for recruitment of immune cells: vascular sources are considered in a fixed order (from bottom left to top right corner) and at every source first a macrophage is (possibly) recruited followed by a T cell. Also proliferation of microglia happens by considering microglial cells in a fixed order.

In order to have more realistic models, we should aim for asynchronicity whenever possible. For state changes, this can be accomplished by considering immune cells in a random order and by immediately finalizing the computed new state. Asynchronous movement can be achieved by moving immune cells in a random order without taking their type into account. Recruitment can be made asynchronous by considering vascular sources in a random order, and by randomly flipping the order of whether to recruit a macrophage or a T cell first. Proliferation of microglia can be modeled more realistically by randomly shuffling the list of microglia on the grid.

Future model extensions

The focus of this work has been on developing methods, models and software for the modeling of TB in lung and CNS. We hope that the developed methodology proves to be a solid and sound foundation for future research. We believe that many intriguing aspects remain to be modeled.

An interesting avenue to explore is so called multi-scale modeling. The models that we built are at the cellular scale. It would be interesting to extend the models to include interactions at the molecular scale. One can think of including TNF- α molecular dynamics in immune cells individually.

Another interesting aspect would be to consider the cytokine IFN- γ as a separate entity in the model rather than having T_γ cells act as a proxy for it. Also down-regulatory mechanisms can be modeled more realistically by including the anti-inflammatory cytokine IL-10 in the model. Vitamine D plays a role in the antimicrobial activities of macrophages [21, 22]. It would be interesting to model and investigate the extent of the role of Vitamine D.

In addition to multi-scale modeling, we could consider multi-compartmental modeling. In Section 1.1.1 we described that the adaptive immune response is initiated in a draining lymph node. Having such a lymph node as a separate compartment in the lung model, captures the influx and appearance of T cells more realistically as opposed to having them phenomenologically appear from day 20 onwards (as specified by the parameter t_{regr}). Efforts are underway to incorporate an ordinary-differential equation based lymph node model as a separate compartment into our lung agent-based model.

Both our models are based on a two-dimensional grid. Since a granuloma is a three-dimensional structure, it would be better for the grid to be three-dimensional as well. When making the transition from 2D to 3D, we have to keep in mind that the number of compartments will increase considerably (i.e. a grid $2\text{ mm} \times 2\text{ mm}$ consists of 10,000 micro-compartments, whereas a grid of $2\text{ mm} \times 2\text{ mm} \times 2\text{ mm}$ consists of 1,000,000 micro-compartments). Such an increase would necessitate the need for a faster implementation.

Faster implementation

A simulation run of 200 days of the lung model takes about 25 minutes on a laptop machine. Since the grid is much bigger in the CNS model, simulating 200 days requires two hours on the same laptop. Using a profiler we have determined that 95% of the computational time is spent in the computation of diffusion. So in order to have a faster implementation, we need to focus on this bottleneck. Instead of solving diffusion on the complete uniform grid, we could use adaptive meshing to focus on relevant regions of the grid — for instance regions where the

cytokine/chemokine levels are nonzero. Another approach would be to consider parallelization; this can be done either on the GPU or on a cluster.

Experimental work

Ideally the CNS/TBM model should be based on and validated against experimental data. An example of an interesting in vivo experiment is to use protein-coated beads in the brains of mice to study the formation of a Rich focus. A similar experiment has been performed to study the availability of TNF- α within a pulmonary TB granuloma [13].

Bibliography

- [1] A. Abbas and A. Lichtman. *Cellular and Molecular Immunology*. Saunders, 5th edition, 2003.
- [2] F. Aloisi. Immune function of microglia. *Glia*, 36(2):165–179, 2001.
- [3] C. E. Barry 3rd, H. I. Boshoff, V. Dartois, T. Dick, S. Ehrt, J. Flynn, D. Schnappinger, R. J. Wilkinson, and D. Young. The spectrum of latent tuberculosis: rethinking the biology and intervention strategies. *Nat Rev Microbiol*, 7(12):845–855, 2009.
- [4] N. Be, G. Lamichhane, J. Grosset, S. Tyagi, Q.-J. Cheng, K. Kim, W. Bishai, and S. Jain. Murine model to study the invasion and survival of mycobacterium tuberculosis in the central nervous system. *J Infect Dis*, 198(10):1520–1528, 2008.
- [5] N. A. Be, K. S. Kim, W. R. Bishai, and S. K. Jain. Pathogenesis of central nervous system tuberculosis. *Curr Mol Med*, 9:94–99, 2009.
- [6] M. Behr, S. Warren, H. Salamon, P. Hopewell, A. P. de Leon, C. Daley, and P. Small. Transmission of Mycobacterium tuberculosis from patients smear-negative for acid-fast bacilli. *Lancet*, 353(9151):444–449, 1999.
- [7] E. Bonabeau. Agent-based modeling: methods and techniques for simulating human systems. *Proc Nat Acad Sci*, 99(Suppl. 3):7280–7287, 2002.
- [8] D. Borland and R. M. Taylor II. Rainbow color map (still) considered harmful. *IEEE Comput Graph Appl*, 27(2):14–17, 2007.
- [9] M. J. Carson, J. M. Doose, B. Melchior, C. D. Schmid, and C. C. Ploix. CNS immune privilege: hiding in plain sight. *Immunol Rev*, 213(1):48–65, 2006.
- [10] F. Cassot, F. Lauwers, C. Fouard, S. Prohaska, and V. Lauwers-Cances. A novel three-dimensional computer-assisted method for a quantitative study of microvascular networks of the human cerebral cortex. *Microcirculation*, 13:1–18, 2006.
- [11] S. Dalziel. Numerical methods lecture notes: partial differential equations. <http://www.amtp.cam.ac.uk/lab/people/sd/lectures/nummeth98/pdes.htm>, Last accessed: July 2010.
- [12] P. Donald, H. Schaaf, and J. Schoeman. Tuberculous meningitis and miliary tuberculosis: the Rich focus revisited. *J Infect*, 50(3):193–195, 2005.

- [13] M. Fallahi-Sichani, M. A. Schaller, D. E. Kirschner, S. L. Kunkel, and J. J. Linderman. Identification of key processes that control tumor necrosis factor availability in a tuberculosis granuloma. *PLoS Comput Biol*, 6(5):e1000778, 2010.
- [14] H. Gouraud. Continuous shading of curved surfaces. *IEEE Trans Comput*, 20(6):623–629, 1971.
- [15] H. Gray. *Anatomy of the human body*. Lea & Febiger, 20th edition, 1918.
- [16] D. D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*. Prentice Hall, 3rd edition, 2003.
- [17] S. Jain, M. Paul-Satyaseela, G. Lamichhane, K. Kim, and W. Bishai. Mycobacterium tuberculosis invasion and traversal across an in vitro human blood-brain barrier as a pathogenic mechanism for central nervous system tuberculosis. *J Infect Dis*, 193(9):1287–1295, 2006.
- [18] T. M. Kauppinen and R. A. Swanson. Poly polymerase-1 promotes microglial activation, proliferation, and matrix metalloproteinase-9-mediated neuron death. *J Immunol*, 174(4):2288–2296, 2005.
- [19] F. Krombach, S. Mnzing, A. M. Allmeling, J. T. Gerlach, J. Behr, and M. Dörger. Cell size of alveolar macrophages: an interspecies comparison. *Environ Health Perspec*, 105 Suppl 5:1261–1263, 1997.
- [20] J. Lee, C. Ling, M. M. Kosmalski, P. Hulseberg, H. A. Schreiber, M. Sandor, and Z. Fabry. Intracerebral mycobacterium bovis bacilli Calmette-Guerin infection-induced immune responses in the CNS. *J Neuroimmunol*, 213(1-2):112–122, 2009.
- [21] P. T. Liu and R. L. Modlin. Human macrophage host defense against mycobacterium tuberculosis. *Curr Opin Immunol*, 20(4):371–376, 2008.
- [22] P. T. Liu, S. Stenger, D. H. Tang, and R. L. Modlin. Cutting edge: Vitamin D-mediated human antimicrobial activity against mycobacterium tuberculosis is dependent on the induction of cathelicidin. *J Immunol*, 179(4):2060–2063, 2007.
- [23] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput Graph*, 21(4):163–169, 1987.
- [24] S. Marino, I. B. Hogue, C. J. Ray, and D. E. Kirschner. A methodology for performing global uncertainty and sensitivity analysis in systems biology. *J Theor Biol*, 254:178–196, 2008.
- [25] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans Model Comput Simul*, 8(1):3–30, 1998.
- [26] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2006.
- [27] K. Murphey, P. Travers, and M. Walport. *Janeway’s immunobiology*. Garland Science, 7th edition, 2008.

- [28] M. Nicas, W. W. Nazaroff, and A. Hubbard. Toward understanding the risk of secondary airborne infection: emission of respirable pathogens. *J Occup Environ*, 2(3):143–154, 2005.
- [29] L. J. Nugent and R. K. Jain. Extravascular diffusion in normal and neoplastic tissues. *Cancer Res*, 44:238–244, 1984.
- [30] S. Paul, P. Laochumroonvoranpong, and G. Kaplan. Comparable growth of virulent and avirulent *Mycobacterium tuberculosis* in human macrophages in vitro. *J Infect Dis*, 174(1):105–112, 1996.
- [31] P. K. Peterson, G. Gekker, S. Hu, and C. C. Chao. *Microglia: A double-edged sword*, chapter 3, pages 31–55. Blackwell Sci, 1997.
- [32] M. M. J. Polfiet, P. J. G. Zwijnenburg, A. M. van Furth, T. van der Poll, E. A. Döpp, C. R. de Lavalette, E. M. L. van Kesteren-Hendrikx, N. van Rooijen, C. D. Dijkstra, and T. K. van den Berg. Meningeal and perivascular macrophages of the central nervous system play a protective role during bacterial meningitis. *J Immunol*, 167:4644–4650, 2001.
- [33] J. C. J. Ray, J. L. Flynn, and D. E. Kirschner. Synergy between individual TNF-dependent functions determines granuloma performance for controlling mycobacterium tuberculosis infection. *J Immunol*, 182:3706–3717, 2009.
- [34] A. Rich and H. McCordock. The pathogenesis of tuberculous meningitis. *Bull Johns Hopkins Hosp*, 52:2–37, 1933.
- [35] R. B. Rock, G. Gekker, S. Hu, W. S. Sheng, M. Cheeran, J. R. Lokensgard, and P. K. Peterson. Role of microglia in central nervous system infections. *Clin Microbiol Rev*, 17(4):942–964, 2004.
- [36] J. L. Segovia-Juarez, S. Ganguli, and D. Kirschner. Identifying control mechanisms of granuloma formation during *M. tuberculosis* infection using an agent-based model. *J Theor Biol*, 231:357–376, 2004.
- [37] M. Shankaran, M. E. Marino, R. Busch, C. Keim, C. King, J. Lee, S. Killion, M. Awada, and M. K. Hellerstein. Measurement of brain microglial proliferation rates in vivo in response to neuroinflammatory stimuli: Application to drug discovery. *J Neurosci Res*, 85(11):2374–2384, 2007.
- [38] J. Sprent. Lifespans of naive, memory and effector lymphocytes. *Curr Opin Immunol*, 5(3):433–438, 1993.
- [39] G. E. Thwaites and T. T. Hien. Tuberculous meningitis: many questions, too few answers. *Lancet Neurol*, 4(3):160–170, 2005.
- [40] G. E. Thwaites and J. F. Schoeman. Update on tuberculosis of the central nervous system: pathogenesis, diagnosis, and treatment. *Clin Chest Med*, 30:745–754, 2009.
- [41] L. Tsenova, A. Bergtold, V. H. Freedman, R. A. Young, and G. Kaplan. Tumor necrosis factor α is a determinant of pathogenesis and disease progression in mycobacterial infection in the central nervous system. *Proc Natl Acad Sci*, 96:5657–5662, 1999.
- [42] L. Tsenova, B. Mangaliso, G. Muller, Y. Chen, and V. H. Freedman. Use of IMiD3, a thalidomide analog, as an adjunct to therapy for experimental tuberculous meningitis. *Antimicrob Agents Chemother*, 46(6):1887–1895, 2002.

- [43] L. Tsenova, K. Sokol, V. H. Freedman, and G. Kaplan. A combination of thalidomide plus antibiotics protects rabbits from mycobacterial meningitis-associated death. *J Infect Dis*, 177(6):1563–1572, 1998.
- [44] R. van Furth, M. M. D. Diesselhoff-den Dulk, and H. Mattie. Quantitative study on the production and kinetics of mononuclear phagocytes during an acute inflammatory reaction. *J Exp Med*, 186(6):1314–1330, 1973.
- [45] G. van Well, C. Wieland, S. Florquin, J. Roord, T. van der Poll, and A. van Furth. A new murine model to study the pathogenesis of tuberculous meningitis. *J Infect Dis*, 195(5):694–697, 2007.
- [46] E. Vynnycky and P. Fine. Life time risks, incubation, and serial interval of tuberculosis. *Am J Epidemiol*, 152:247–263, 2000.
- [47] B. L. Welch. The generalization of ‘Student’s’ problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35, 1947.
- [48] World Health Organization. Global tuberculosis control: A short update to the 2009 report, 2009.
- [49] World Health Organization. Tuberculosis facts, 2009.
- [50] M. Zhang, J. Gong, Y. Lin, and P. Barnes. Growth of virulent and avirulent *Mycobacterium tuberculosis* strains in human macrophages. *Infect Immun*, 66(2):794–799, 1998.
- [51] A. R. Zink, C. Sola, U. Reischl, W. Grabner, N. Rastogi, H. Wolf, and A. G. Nerlich. Characterization of *Mycobacterium tuberculosis* complex DNAs from Egyptian mummies by spoligotyping. *J Clin Microbiol*, 41(1):359–367, 2003.

Appendix A

Diffusion Equation

A.1 FTCS

In this section we show that the recursive equation that derived in Subsection 2.1.3 is the same as the one used by Segovia-Juarez et al. The equation that we derived is

$$C_{i,j}(t + \Delta t) = C_{i,j}(t) + \mu(C_{i+1,j}(t) + C_{i-1,j}(t) + C_{i,j+1}(t) + C_{i,j-1}(t) - 4C_{i,j}(t)) \quad (\text{A.1})$$

where $\mu = \Delta t D / (\Delta x)^2$. Segovia-Juarez et al. use the following recursive equation describing diffusion [36]

$$C_{i,j}(t + \Delta t) = (1 - \lambda) \cdot C_{i,j}(t) + \frac{\lambda}{4}(C_{i+1,j}(t) + C_{i-1,j}(t) + C_{i,j+1}(t) + C_{i,j-1}(t)) \quad (\text{A.2})$$

where $\lambda = 4\Delta t D / (\Delta x)^2$. We have that

$$\lambda = 4\mu. \quad (\text{A.3})$$

Substituting (A.3) in (A.2) results in

$$\begin{aligned} C_{i,j}(t + \Delta t) &= (1 - 4\mu) \cdot C_{i,j}(t) + \mu(C_{i+1,j}(t) + C_{i-1,j}(t) + C_{i,j+1}(t) + C_{i,j-1}(t)) \\ &= C_{i,j}(t) - 4\mu C_{i,j}(t) + \mu(C_{i+1,j}(t) + C_{i-1,j}(t) + C_{i,j+1}(t) + C_{i,j-1}(t)) \\ &= C_{i,j}(t) + \mu(C_{i+1,j}(t) + C_{i-1,j}(t) + C_{i,j+1}(t) + C_{i,j-1}(t) - 4C_{i,j}(t)). \end{aligned}$$

A.2 BTCS

Ray et al. used backward time centered space (BTCS) to solve the diffusion equation. As opposed to the forward time centered space (FTCS) scheme that we employed, BTCS is an implicit method and results in a linear system of equations. To solve this system, the authors made use of successive over-relaxation (SOR). In Listing A.1 the relevant source code is shown. There we can see that the constant `ok` is the convergence threshold. We can also see that every time step, another loop is executed for the SOR (line 15). Every iteration of the inner loop the residual error is computed and stored in `max` and `count` is incremented. The next iteration takes place when the following two conditions are satisfied:

1. `count` is less than 100, and
2. there is no convergence (i.e. `max > ok`).

Basically the first condition ensures that there is an upper bound on the number of iterations the SOR may take. The second condition tells the SOR to continue when there is no convergence. Suppose that the for-loop terminates because of convergence. In that case we have that `max` \leq `ok`. If we now look at the computation of the next time steps, we see that no iterations of the inner for-loop are performed as `max` has not been reset — we still have `max` \leq `ok`. Ultimately, this boils down to that every 10-minute time step diffusion is only computed for the first 6-second time steps. Fixing the error by resetting `max` every iteration (e.g. `max = 2 * ok`) results in an algorithm that is significantly slower than the FTCS scheme.

Listing A.1: BTCS

```

1 void diffusion() {
2     // allowable error for good enough convergence
3     const float ok = 0.001;
4     // the actual maximum error. needs to start out max > ok
5     float max = 0.002;
6
7     /* grid dimensions are MSIZE * MSIZE */
8     // v the iteration at the new time t+dt
9     double v[MSIZE + 2][MSIZE + 2];
10    // u iteration at time t
11    double u[MSIZE + 2][MSIZE + 2];
12
13    for (int t = 0; t < 100; t++) {
14        // then for each time step a loop for the SOR
15        for (int count = 0; count < 100 && max > ok; count++) {
16            max = ok;
17            double err = 0.0;
18
19            // Sort through the list.
20            for (i = MSIZE; i >= 1; i--) {
21                for (j = MSIZE; j >= 1; j--) {
22                    double res = u[i][j] - (a_plus_one) * v[i][j]
23                        + (a_over_four) * (v[i - 1][j] + v[i + 1][j]
24                        + v[i][j + 1] + v[i][j - 1]);
25                    v[i][j] = v[i][j] + w * res;
26
27                    err += fabs(res);
28                }
29            }
30
31            max = err;
32        }
33
34        // now we update the solution to the new time
35        for (i = MSIZE; i >= 1; i--) {
36            for (j = MSIZE; j >= 1; j--) {
37                // compute degradation
38                v[i][j] -= v[i][j] * DIFF_CHEMOKINES;
39                u[i][j] = v[i][j];
40            }
41        }
42    } // end of loop over time;
43 }

```


Appendix B

Build Instructions

In the build instructions we will focus on the lung model; the brain model can be built with the same instructions.

B.1 Windows

There are two possible ways for building the software in MS Windows: using Microsoft Visual Studio or using MinGW.

B.1.1 MinGW

In order to build using MinGW, the package ‘Qt SDK for Open Source C++ development on Windows’ needs to be downloaded. After installing this package, start a command prompt by clicking on ‘Start → Programs → Qt SDK by Nokia v2010.xy (open source) → Qt Command Prompt’.

The first thing to be done is to build Boost:

```
C:\boost>bjam --build-dir="C:\Qt\boost" --toolset=gcc stage
```

After building Boost, the software can be compiled as follows:

```
C:\src\GR-lung>qmake
```

```
C:\src\GR-lung>mingw32-make release
```

Since the default stack size is too small, we have to update this property in the PE-header of the executable as follows:

```
c:\src\GR-lung>editbin /STACK:10485760 grviz-lung.exe
```

B.1.2 Visual Studio

Building with Visual Studio requires the installation of the package ‘Qt for Open Source C++ development on Windows (VS2008)’. In addition, the ‘Visual Studio Add-in’ needs to be installed. Boost libraries pre-built for Visual Studio can be obtained from www.boostpro.com.

In order to build Visual Studio needs to be started by clicking on ‘Start → Programs → Qt by Nokia v4.6.x (VS2008 OpenSource) → Visual Studio with Qt 4.6.0’. After which a solution file can be made by clicking in the menu on Qt followed by Open Qt Project File (.pro).

It is important to properly set the stack size by right clicking on the project and clicking on properties, going to linker and setting ‘Stack reserve size’ to 10485760. A release build can be generated by setting the solution configuration to release and pressing ‘Ctrl+Shift+B’.

B.2 Linux

In order to build the software on a Debian based distribution (such as Ubuntu), the package build-essential needs to be installed:

```
mohammed@artemis:~$ sudo apt-get install build-essential
```

Subsequently development packages of Boost and Qt need to be installed:

```
mohammed@artemis:~$ sudo apt-get install libqt4-dev
```

```
mohammed@artemis:~$ sudo apt-get install libboost-all-dev
```

When all necessary packages are installed, the software can be built as follows:

```
mohammed@artemis:~$ qmake
```

```
mohammed@artemis:~$ make release
```

B.3 Mac OS X

The Qt SDK for Mac OS X can be downloaded from the Qt website. After building Boost, the software can be built in a terminal as follows:

```
$ qmake -spec macx-g++
```

```
$ make release
```

Appendix C

Parameter Sets

C.1 Mapping tables

The attributes of the XML parameter file corresponding to the parameters used in the description of the lung model are given in the following table.

Table C.1: Mapping of lung model parameters

Parameter	XML attribute
N_{caseous}	/GR/@nrKillingsCaseation
$\tau_{\text{TNF,apopt}}$	/GR/@thresholdApoptosisTNF
$p_{\text{TNF,apopt}}$	/GR/@probApoptosisTNF
$\tau_{\text{M,TNF,NF}\kappa\text{B}}$	/GR/Mac/@thresholdNFkBTFNF
$N_{\text{M,Be,NF}\kappa\text{B}}$	/GR/Mac/@nrExtMtbNFkB
$\sigma_{\text{M,deact}}$	/GR/Mac/@maxTimeReg
$\sigma_{\text{c,deact}}$	/GR/Tcell/Tcyt/@maxTimeReg
$\sigma_{\gamma,\text{deact}}$	/GR/Tcell/Tgam/@maxTimeReg
$p_{\text{M,STAT1}}$	/GR/Mac/@probStat1Tgam
$p_{\text{M,kill}}$	/GR/Mac/@probKillExtMtbRest
$N_{\text{M,Bi,Ci}}$	/GR/Mac/@nrIntMtbCInf
$N_{\text{M,Bi,burst}}$	/GR/Mac/@nrIntMtbBurstCInf
$N_{\text{M,Be,kill}}$	/GR/Mac/@nrExtMtbUptakeAct
$p_{\text{c,kill}}$	/GR/Tcell/Tcyt/@probKillMac
$p_{\text{c,kill,cleanly}}$	/GR/Tcell/Tcyt/@probKillMacCleanly
$p_{\text{Fas,apopt}}$	/GR/Tcell/Tgam/@probApoptosisFasFasL
s_{TNF}	/GR/Mac/@dTfNF
s_{CCL2}	/GR/Mac/@dCCL2
s_{CCL5}	/GR/Mac/@dCCL5
s_{CXCL9}	/GR/Mac/@dCXCL9
D_{TNF}	/GR/@diffusivityTNF
D_{CC}	/GR/@diffusivityChemokines
δ_{TNF}	1 - /GR/@degRateTNF
δ_{CC}	1 - /GR/@degRateChemokines

Continued on next page

Table C.1 – continued from previous page

Parameter	XML attribute
t_{recr}	/GR/Tcell/@timeRecEnabled
N_M	/GR/Mac/@initNumber
α_{TNF}	/GR/@effectRecTNF
α_{CCL2}	/GR/@effectRecCCL2
α_{CCL5}	/GR/@effectRecCCL5
α_{CXCL9}	/GR/@effectRecCXCL9
$\tau_{M,\text{recr}}$	/GR/Mac/@thresholdRec
$\tau_{c,\text{recr}}$	/GR/Tcell/Tcyt/@thresholdRec
$\tau_{\gamma,\text{recr}}$	/GR/Tcell/Tgam/@thresholdRec
$\tau_{\text{reg},\text{recr}}$	/GR/Tcell/Treg/@thresholdRec
$p_{M,\text{recr}}$	/GR/Mac/@probRec
$p_{T,\text{recr}}$	/GR/Tcell/@probRec
$p_{c,\text{recr}}$	/GR/Tcell/Tcyt/@probRec
$p_{\gamma,\text{recr}}$	/GR/Tcell/Tgam/@probRec
$p_{\text{reg},\text{recr}}$	/GR/Tcell/Treg/@probRec
$v_{M,r}$	/GR/Mac/@movementRest
$v_{M,a}$	/GR/Mac/@movementAct
$v_{M,i}$	/GR/Mac/@movementInf
$p_{T,M}$	/GR/Tcell/@probMoveToMac
$p_{T,T}$	/GR/Tcell/@probMoveToTcell
α_{Bi}	/GR/Mtb/@growthRateIntMtb - 1
α_{Be}	/GR/Mtb/@growthRateExtMtb - 1
K_{Be}	/GR/Mtb/@growthExtMtbBound

The mapping of parameters specific to the CNS model to XML attributes is given in the following table.

Table C.2: Mapping of CNS model parameters

Parameter	XML attribute
$\tau_{m,\text{TNF},\text{NF}\kappa\text{B}}$	/GR/Mg/@thresholdNFkB
$N_{m,\text{Be},\text{NF}\kappa\text{B}}$	/GR/Mg/@nrExtMtbNFkB
$\sigma_{m,\text{deact}}$	/GR/Mg/@maxTimeReg
$N_{m,\text{ram}}$	/GR/Mg/@initNumber
$\tau_{m,\text{TNF},\text{react}}$	/GR/Mg/@thresholdReactTNF
$p_{m,\text{STAT1}}$	/GR/Mg/@probStat1Tgam
$p_{m,\text{kill}}$	/GR/Mg/@nrExtMtbUptakeAct
$N_{m,\text{Bi},\text{CI}}$	/GR/Mg/@nrIntMtbCInf
$N_{m,\text{Bi},\text{burst}}$	/GR/Mg/@nrIntMtbBurstCInf
$N_{m,\text{Be},\text{kill}}$	/GR/Mg/@nrExtMtbUptakeAct
$N_{m,\text{react},\text{prol}}$	/GR/Mg/@nrProliferateReact * /GR/Mg/@probProliferateReact
$\tau_{m,\text{TNF},\text{prol}}$	/GR/Mg/@proliferateReactTNF

Continued on next page

Table C.2 – continued from previous page

Parameter	XML attribute
$v_{m,r}$	/GR/Mg/@movementRest
$v_{m,a}$	/GR/Mg/@movementAct
$v_{m,i}$	/GR/Mg/@movementInf
$p_{T,m}$	/GR/Tcell/@probMoveToMac

C.2 Lung – containment

```

1 <GR
2     nrKillingsCaseation = "10"
3     nrSources = "50"
4     diffusivityTNF = "5.2e-08"
5     diffusivityChemokines = "5.2e-08"
6     degRateTNF = "0.9905"
7     degRateChemokines = "0.9905"
8     thresholdApoptosisTNF = "70"
9     probApoptosisTNF = "0.04"
10    minChemotaxis = "2"
11    maxChemotaxis = "2000">
12    <Mac
13        initNumber = "105"
14        maxAge = "14400"
15        maxAgeAct = "1440"
16        maxTimeReg = "72"
17        movementRest = "2"
18        movementAct = "31"
19        movementInf = "144"
20        dTNF = "1.8"
21        dCCL2 = "35"
22        dCCL5 = "35"
23        dCXCL9 = "70"
24        thresholdNFkB TNF = "51"
25        nrExtMtbUptakeRest = "1"
26        probKillExtMtbRest = "0.02"
27        nrExtMtbNFkB = "11.6435"
28        nrIntMtbCInf = "10"
29        nrIntMtbBurstCInf = "20"
30        probStat1Tgam = "0.08"
31        nrExtMtbUptakeAct = "10"
32        thresholdRec = "0.025"
33        probRec = "0.04"/>
34    <Tcell
35        maxAge = "432"
36        timeRecEnabled = "2880"
37        probMoveToMac = "0.02"
38        probMoveToTcell = "0.08"
39        probRec = "0.1">
40        <Tgam
41            maxTimeReg = "36"
42            probApoptosisFasFasL = "0.06"
43            thresholdRec = "0.4"
44            probRec = "0.54"/>

```

```

45         <Tcyt
46             maxTimeReg = "33"
47             probKillMac = "0.12"
48             probKillMacCleanly = "0.75"
49             thresholdRec = "0.4"
50             probRec = "0.36"/>
51         <Treg
52             thresholdRec = "0.05"
53             probRec = "0.1"/>
54     </Tcell>
55     <Mtb
56         growthRateIntMtb = "1.0015"
57         growthRateExtMtb = "1.0007"
58         growthExtMtbBound = "200"/>
59     <Init>
60         <Mac row = "50" col = "50"/>
61         <ExtMtb row = "45" col = "45"/>
62         <ExtMtb row = "45" col = "55"/>
63         <ExtMtb row = "55" col = "45"/>
64         <ExtMtb row = "55" col = "55"/>
65     </Init>
66 </GR>

```

C.3 Lung – clearance

```

1 <GR
2     nrKillingsCaseation = "10"
3     nrSources = "50"
4     diffusivityTNF = "5.2e-08"
5     diffusivityChemokines = "5.2e-08"
6     degRateTNF = "0.9905"
7     degRateChemokines = "0.9905"
8     thresholdApoptosisTNF = "70"
9     probApoptosisTNF = "0.04"
10    minChemotaxis = "2"
11    maxChemotaxis = "2000">
12    <Mac
13        initNumber = "105"
14        maxAge = "14400"
15        maxAgeAct = "1440"
16        maxTimeReg = "55"
17        movementRest = "2"
18        movementAct = "11"
19        movementInf = "144"
20        dTNF = "1.8"
21        dCCL2 = "35"
22        dCCL5 = "35"
23        dCXCL9 = "70"
24        thresholdNFkB TNF = "51"
25        nrExtMtbUptakeRest = "1"
26        probKillExtMtbRest = "0.02"
27        nrExtMtbNFkB = "114.367"
28        nrIntMtbCInf = "10"
29        nrIntMtbBurstCInf = "20"
30        probStat1Tgam = "0.08"
31        nrExtMtbUptakeAct = "10"
32        thresholdRec = "0.025"
33        probRec = "0.04"/>

```

```

34     <Tcell
35         maxAge = "432"
36         timeRecEnabled = "2880"
37         probMoveToMac = "0.02"
38         probMoveToTcell = "0.08"
39         probRec = "0.1">
40     <Tgam
41         maxTimeReg = "32"
42         probApoptosisFasFasL = "0.06"
43         thresholdRec = "0.4"
44         probRec = "0.54"/>
45     <Tcyt
46         maxTimeReg = "39"
47         probKillMac = "0.12"
48         probKillMacCleanly = "0.75"
49         thresholdRec = "0.4"
50         probRec = "0.36"/>
51     <Treg
52         thresholdRec = "0.05"
53         probRec = "0.1"/>
54 </Tcell>
55 <Mtb
56     growthRateIntMtb = "1.0015"
57     growthRateExtMtb = "1.0007"
58     growthExtMtbBound = "200"/>
59 <Init>
60     <Mac row = "50" col = "50"/>
61     <ExtMtb row = "45" col = "45"/>
62     <ExtMtb row = "45" col = "55"/>
63     <ExtMtb row = "55" col = "45"/>
64     <ExtMtb row = "55" col = "55"/>
65 </Init>
66 </GR>

```

C.4 Lung – dissemination

```

1 <GR
2     nrKillingsCaseation = "10"
3     nrSources = "50"
4     diffusivityTNF = "5.2e-08"
5     diffusivityChemokines = "5.2e-08"
6     degRateTNF = "0.9905"
7     degRateChemokines = "0.9905"
8     thresholdApoptosisTNF = "70"
9     probApoptosisTNF = "0.04"
10    minChemotaxis = "2"
11    maxChemotaxis = "2000">
12    <Mac
13        initNumber = "105"
14        maxAge = "14400"
15        maxAgeAct = "1440"
16        maxTimeReg = "110"
17        movementRest = "2"
18        movementAct = "79"
19        movementInf = "144"
20        dTNF = "1.8"
21        dCCL2 = "35"
22        dCCL5 = "35"

```

```

23         dCXCL9 = "70"
24         thresholdNFkB TNF = "51"
25         nrExtMtbUptakeRest = "1"
26         probKillExtMtbRest = "0.02"
27         nrExtMtbNFkB = "128.326"
28         nrIntMtbCInf = "10"
29         nrIntMtbBurstCInf = "20"
30         probStat1Tgam = "0.08"
31         nrExtMtbUptakeAct = "10"
32         thresholdRec = "0.025"
33         probRec = "0.04"/>
34     <Tcell
35         maxAge = "432"
36         timeRecEnabled = "2880"
37         probMoveToMac = "0.02"
38         probMoveToTcell = "0.08"
39         probRec = "0.1">
40         <Tgam
41             maxTimeReg = "12"
42             probApoptosisFasFasL = "0.06"
43             thresholdRec = "0.4"
44             probRec = "0.54"/>
45         <Tcyt
46             maxTimeReg = "7"
47             probKillMac = "0.12"
48             probKillMacCleanly = "0.75"
49             thresholdRec = "0.4"
50             probRec = "0.36"/>
51         <Treg
52             thresholdRec = "0.05"
53             probRec = "0.1"/>
54     </Tcell>
55     <Mtb
56         growthRateIntMtb = "1.0015"
57         growthRateExtMtb = "1.0007"
58         growthExtMtbBound = "200"/>
59     <Init>
60         <Mac row = "50" col = "50"/>
61         <ExtMtb row = "45" col = "45"/>
62         <ExtMtb row = "45" col = "55"/>
63         <ExtMtb row = "55" col = "45"/>
64         <ExtMtb row = "55" col = "55"/>
65     </Init>
66 </GR>

```

C.5 CNS – containment

```

1 <GR
2     nrKillingsCaseation = "9"
3     nrSources = "800"
4     diffusivityTNF = "5.2e-08"
5     diffusivityChemokines = "5.2e-08"
6     degRateTNF = "0.9905"
7     degRateChemokines = "0.9905"
8     thresholdApoptosisTNF = "70"
9     probApoptosisTNF = "0.04"
10    minChemotaxis = "2"
11    maxChemotaxis = "2000">

```



```

12     <Mac
13         initNumber = "2000"
14         maxAge = "14400"
15         maxAgeAct = "1440"
16         maxTimeReg = "62"
17         movementRest = "2"
18         movementAct = "7"
19         movementInf = "144"
20         dTNF = "1.8"
21         dCCL2 = "35"
22         dCCL5 = "35"
23         dCXCL9 = "70"
24         thresholdNFkB TNF = "51"
25         nrExtMtbUptakeRest = "1"
26         probKillExtMtbRest = "0.02"
27         nrExtMtbNFkB = "211.513"
28         nrIntMtbCInf = "10"
29         nrIntMtbBurstCInf = "20"
30         probStat1Tgam = "0.0670123"
31         nrExtMtbUptakeAct = "10"
32         thresholdRec = "0.025"
33         probRec = "0.04"/>
34     <Mg
35         initNumber = "2000"
36         maxAge = "22100"
37         maxAgeAct = "1440"
38         maxTimeReg = "27"
39         movementRest = "2"
40         movementAct = "9"
41         movementInf = "144"
42         dTNF = "1.8"
43         dCCL2 = "35"
44         dCCL5 = "35"
45         dCXCL9 = "71"
46         thresholdNFkB TNF = "51"
47         nrExtMtbUptakeRest = "1"
48         probKillExtMtbRest = "0.02"
49         nrExtMtbNFkB = "160.314"
50         nrIntMtbCInf = "5"
51         nrIntMtbBurstCInf = "10"
52         probStat1Tgam = "0.0520861"
53         nrExtMtbUptakeAct = "5"
54         thresholdReactTNF = "1.67308"
55         nrProliferateReact = "34"
56         proliferateReactTNF = "0.773001"
57         probProliferateReact = "0.163126"/>
58     <Tcell
59         maxAge = "432"
60         timeRecEnabled = "2880"
61         probMoveToMac = "0.02"
62         probMoveToTcell = "0.08"
63         probRec = "0.1">
64         <Tgam
65             maxTimeReg = "54"
66             probApoptosisFasFasL = "0.06"
67             thresholdRec = "0.4"
68             probRec = "0.54"/>
69         <Tcyt
70             maxTimeReg = "40"

```

```

71         probKillMac = "0.12"
72         probKillMacCleanly = "0.75"
73         thresholdRec = "0.4"
74         probRec = "0.36"/>
75     <Treg
76         thresholdRec = "0.05"
77         probRec = "0.1"/>
78 </Tcell>
79 <Mtb
80     growthRateIntMtb = "1.0015"
81     growthRateExtMtb = "1.0007"
82     growthExtMtbBound = "200"/>
83 <Init>
84     <Mg row = "150" col = "165"/>
85     <Mac row = "140" col = "155"/>
86     <Mac row = "50" col = "100"/>
87 </Init>
88 </GR>

```

C.6 Lung – TNF- α knock-out

```

1 <GR
2     nrKillingsCaseation = "10"
3     nrSources = "50"
4     diffusivityTNF = "5.2e-08"
5     diffusivityChemokines = "5.2e-08"
6     degRateTNF = "0.9905"
7     degRateChemokines = "0.9905"
8     thresholdApoptosisTNF = "70"
9     probApoptosisTNF = "0.04"
10    minChemotaxis = "2"
11    maxChemotaxis = "2000"
12    effectRecTNF = "0"
13    effectRecCCL2 = ".0514"
14    effectRecCCL5 = ".0514"
15    effectRecCXCL9 = ".0257">
16    <Mac
17        initNumber = "105"
18        maxAge = "14400"
19        maxAgeAct = "1440"
20        maxTimeReg = "72"
21        movementRest = "2"
22        movementAct = "31"
23        movementInf = "144"
24        dTNF = "0"
25        dCCL2 = "35"
26        dCCL5 = "35"
27        dCXCL9 = "70"
28        thresholdNFkB TNF = "51"
29        nrExtMtbUptakeRest = "1"
30        probKillExtMtbRest = "0.02"
31        nrExtMtbNFkB = "11.6435"
32        nrIntMtbCInf = "10"
33        nrIntMtbBurstCInf = "20"
34        probStat1Tgam = "0.08"
35        nrExtMtbUptakeAct = "10"
36        thresholdRec = "0.025"
37        probRec = "0.04"/>

```

```

38     <Tcell
39         maxAge = "432"
40         timeRecEnabled = "2880"
41         probMoveToMac = "0.02"
42         probMoveToTcell = "0.08"
43         probRec = "0.1">
44         <Tgam
45             maxTimeReg = "36"
46             probApoptosisFasFasL = "0.06"
47             thresholdRec = "0.4"
48             probRec = "0.54"/>
49         <Tcyt
50             maxTimeReg = "33"
51             probKillMac = "0.12"
52             probKillMacCleanly = "0.75"
53             thresholdRec = "0.4"
54             probRec = "0.36"/>
55         <Treg
56             thresholdRec = "0.05"
57             probRec = "0.1"/>
58     </Tcell>
59     <Mtb
60         growthRateIntMtb = "1.0015"
61         growthRateExtMtb = "1.0007"
62         growthExtMtbBound = "200"/>
63     <Init>
64         <Mac row = "50" col = "50"/>
65         <ExtMtb row = "45" col = "45"/>
66         <ExtMtb row = "45" col = "55"/>
67         <ExtMtb row = "55" col = "45"/>
68         <ExtMtb row = "55" col = "55"/>
69     </Init>
70 </GR>

```

C.7 Lung – IFN- γ knock-out

```

1 <GR
2     nrKillingsCaseation = "10"
3     nrSources = "50"
4     diffusivityTNF = "5.2e-08"
5     diffusivityChemokines = "5.2e-08"
6     degRateTNF = "0.9905"
7     degRateChemokines = "0.9905"
8     thresholdApoptosisTNF = "70"
9     probApoptosisTNF = "0.04"
10    minChemotaxis = "2"
11    maxChemotaxis = "2000">
12    <Mac
13        initNumber = "105"
14        maxAge = "14400"
15        maxAgeAct = "1440"
16        maxTimeReg = "72"
17        movementRest = "2"
18        movementAct = "31"
19        movementInf = "144"
20        dTNF = "1.8"
21        dCCL2 = "35"
22        dCCL5 = "35"

```

```

23         dCXCL9 = "70"
24         thresholdNFkB_TNF = "51"
25         nrExtMtbUptakeRest = "1"
26         probKillExtMtbRest = "0.02"
27         nrExtMtbNFkB = "11.6435"
28         nrIntMtbCInf = "10"
29         nrIntMtbBurstCInf = "20"
30         probStat1Tgam = "0"
31         nrExtMtbUptakeAct = "10"
32         thresholdRec = "0.025"
33         probRec = "0.04"/>
34     <Tcell
35         maxAge = "432"
36         timeRecEnabled = "2880"
37         probMoveToMac = "0.02"
38         probMoveToTcell = "0.08"
39         probRec = "0.1">
40     <Tgam
41         maxTimeReg = "36"
42         probApoptosisFasFasL = "0.06"
43         thresholdRec = "0.4"
44         probRec = "0.54"/>
45     <Tcyt
46         maxTimeReg = "33"
47         probKillMac = "0.12"
48         probKillMacCleanly = "0.75"
49         thresholdRec = "0.4"
50         probRec = "0.36"/>
51     <Treg
52         thresholdRec = "0.05"
53         probRec = "0.1"/>
54 </Tcell>
55 <Mtb
56     growthRateIntMtb = "1.0015"
57     growthRateExtMtb = "1.0007"
58     growthExtMtbBound = "200"/>
59 <Init>
60     <Mac row = "50" col = "50"/>
61     <ExtMtb row = "45" col = "45"/>
62     <ExtMtb row = "45" col = "55"/>
63     <ExtMtb row = "55" col = "45"/>
64     <ExtMtb row = "55" col = "55"/>
65 </Init>
66 </GR>

```

C.8 CNS – IFN- γ knock-out

```

1 <GR
2     nrKillingsCaseation = "9"
3     nrSources = "800"
4     diffusivityTNF = "5.2e-08"
5     diffusivityChemokines = "5.2e-08"
6     degRateTNF = "0.9905"
7     degRateChemokines = "0.9905"
8     thresholdApoptosisTNF = "70"
9     probApoptosisTNF = "0.04"
10    minChemotaxis = "2"
11    maxChemotaxis = "2000">

```

```

12     <Mac
13         initNumber = "2000"
14         maxAge = "14400"
15         maxAgeAct = "1440"
16         maxTimeReg = "62"
17         movementRest = "2"
18         movementAct = "7"
19         movementInf = "144"
20         dTNF = "1.8"
21         dCCL2 = "35"
22         dCCL5 = "35"
23         dCXCL9 = "70"
24         thresholdNFkBTFN = "51"
25         nrExtMtbUptakeRest = "1"
26         probKillExtMtbRest = "0.02"
27         nrExtMtbNFkB = "211.513"
28         nrIntMtbCInf = "10"
29         nrIntMtbBurstCInf = "20"
30         probStat1Tgam = "0"
31         nrExtMtbUptakeAct = "10"
32         thresholdRec = "0.025"
33         probRec = "0.04"/>
34     <Mg
35         initNumber = "2000"
36         maxAge = "22100"
37         maxAgeAct = "1440"
38         maxTimeReg = "27"
39         movementRest = "2"
40         movementAct = "9"
41         movementInf = "144"
42         dTNF = "1.8"
43         dCCL2 = "35"
44         dCCL5 = "35"
45         dCXCL9 = "71"
46         thresholdNFkBTFN = "51"
47         nrExtMtbUptakeRest = "1"
48         probKillExtMtbRest = "0.02"
49         nrExtMtbNFkB = "160.314"
50         nrIntMtbCInf = "5"
51         nrIntMtbBurstCInf = "10"
52         probStat1Tgam = "0"
53         nrExtMtbUptakeAct = "5"
54         thresholdReactTNF = "1.67308"
55         nrProliferateReact = "34"
56         proliferateReactTNF = "0.773001"
57         probProliferateReact = "0.163126"/>
58     <Tcell
59         maxAge = "432"
60         timeRecEnabled = "2880"
61         probMoveToMac = "0.02"
62         probMoveToTcell = "0.08"
63         probRec = "0.1">
64         <Tgam
65             maxTimeReg = "54"
66             probApoptosisFasFasL = "0.06"
67             thresholdRec = "0.4"
68             probRec = "0.54"/>
69         <Tcyt
70             maxTimeReg = "40"

```

```
71         probKillMac = "0.12"
72         probKillMacCleanly = "0.75"
73         thresholdRec = "0.4"
74         probRec = "0.36"/>
75     <Treg
76         thresholdRec = "0.05"
77         probRec = "0.1"/>
78 </Tcell>
79 <Mtb
80     growthRateIntMtb = "1.0015"
81     growthRateExtMtb = "1.0007"
82     growthExtMtbBound = "200"/>
83 <Init>
84     <Mg row = "150" col = "165"/>
85     <Mac row = "140" col = "155"/>
86     <Mac row = "50" col = "100"/>
87 </Init>
88 </GR>
```