University of Alberta

Library Release Form

Name of Author:	Catherine Beauchemin
Title of Thesis:	Spatiotemporal Modelling of Viral Infection Dynamics
Degree:	Doctor of Philosophy
Year this Degree Granted:	2006

Catherine Beauchemin
Avadh Bhatia Physics Laboratory,
University of Alberta,
Edmonton, Alberta
Canada
T6G 2J1

November 16, 2005

Things should be made as simple as possible, but not any simpler.

Albert Einstein

University of Alberta

Spatiotemporal Modelling of Viral Infection Dynamics

by

Catherine Beauchemin

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Department of Physics

Edmonton, Alberta
Spring 2006

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Spatiotemporal Modelling of Viral Infection Dynamics** submitted by **Catherine Beauchemin** in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dr. Jack Tuszynski (Supervisor)

Dr. Richard Sydora (Chair)

Dr. Ivan L'Heureux (External)

Dr. John Samuel

Dr. Gerda de Vries

Dr. Zbyszek Gortel

DATE: _____

To

My mom, Kipp, Jack, Alan, and others.

Abstract

Viral kinetics have been studied extensively in the past through the use of ordinary differential equations describing the time evolution of the diseased state in a spatially well-mixed medium. However, emerging spatial structures such as localized populations of dead cells might affect the spread of infection, similar to the manner in which a counter-fire can stop a forest fire from spreading.

In the first phase of the project, a simple two-dimensional cellular automaton model of viral infections was developed. It was validated against clinical immunological data for uncomplicated influenza A infections and shown to be accurate enough to adequately model them.

In the second phase of the project, the simple two-dimensional cellular automaton model was used to investigate the effects of relaxing the well-mixed assumption on viral infection dynamics. It was shown that grouping the initially infected cells into patches rather than distributing them uniformly on the grid reduced the infection rate as only cells on the perimeter of the patch have healthy neighbours to infect. Use of a local epithelial cell regeneration rule where dead cells are replaced by healthy cells when an immediate neighbour divides was found to result in more extensive damage of the epithelium and yielded a better fit to experimental influenza A infection data than a global regeneration rule based on division rate of healthy cell. Finally, the addition of immune cell at the site of infection was found to be a better strategy at low infection levels, while addition at random locations on the grid was the better strategy at high infection level.

In the last project, the movement of T cells within lymph nodes in the absence of antigen, was investigated. Based on individual T cell track data captured by two-photon microscopy experiments in vivo, a simple model was proposed for the motion of T cells. This is the first step towards the implementation of a more realistic spatiotemporal model

of HIV than those proposed thus far [76, 90].

Acknowledgements

The research presented in this document was mainly supported by Dr. Jack Tuszynski's MITACS' Mathematical Modelling in Pharmaceutical Development (MMPD) project. Portions of this work were done under the auspices of the U.S. Department of Energy under contract W-7405-ENG-36 and supported by NIH grant AI 28433 awarded to Dr. Alan S. Perelson. This work was also partly funded through the UNM/LANL Joint Science and Technology Laboratory grant awarded to Dr. Alan S. Perelson and Dr. Stephanie Forrest.

I would also like to acknowledge the work of the developers and maintainers of: GTK+, GtkGLExt, XFig, Ipe, pdftoipe, Gimp, LaTeX, Transcode, UAthesis LaTeXdocument class, xine, and of course, Linux. Without these tools, a lot of things would have been a lot harder to do if not simply impossible.

Finally, I would like to thank Kipp Cannon, Jack Tuszynski, John Samuel, and Alan S. Perelson for helpful discussions and much needed moral support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

"Self-organization" is a process in which the entropy of a (normally open) system decreases automatically without being guided or managed by an outside source. Often the most interesting behaviours exhibited by such systems are neither exhibited by individual agents nor the result of a centralized information structure. Instead, these "emergent" properties arises from the multiple interactions between the simple agents. A good example of such systems is droplets accumulating slowly with simple rules giving rise to the intricate patterns observed in snowflakes. The immune system is also a perfect example of such systems. It is made of many types of agents, for instance B cells, T cells, etc., which interact directly, e.g. a B cell binding with an antigen, or through their environment, e.g. an activated T cell releasing cytokines that trigger the division of neighbouring B and T cells. From the interactions of those agents, a global complex system emerges which is capable of learning (e.g. adapt to new pathogens) and memory (e.g. retain affinity for previously encountered pathogens).

In the past two decades, many approaches have been chosen to model different aspects of the immune system. Differential equation models are perhaps the most common and are typically used to simulate the immunological and epidemiological dynamics of particular diseases to try and identify the critical parameters involved [11, 63]. Genetic algorithms have been applied to the modelling of the evolution of diversity and pattern recognition capability in the immune system [25, 36, 37]. On the theoretical front, some ingenious work [23, 62, 72, 74] has been done to understand aspects of the immune system from the perspective of optimization problems. Efforts to build an immune system tailored for computer networks [24, 73] have raised our understanding of the crucial nature of certain immune system mechanisms. Cellular automaton (CA) models have been developed to simulate the immune dynamics of particular diseases [76, 90], to model shape space interactions [89] based on the network theory introduced by Jerne [39], and have been chosen as the tool of choice to implement highly complex general immune system simulators [7, 14, 43, 71] that are to be used to run immune system experiments *in silico*. More recently, a promising stage-structured modelling approach was used to study cytotoxic T lymphocyte response to antigen [16].

Differential equation modelling is the most popular approach to understanding and

characterizing the dynamics of viral infections. The basic viral infection model, which was introduced by Perelson [63, 65], describes the temporal evolution of the population of susceptible or target cells, T, which become infected, I, as a result of their interactions with virus particles, V, namely

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \lambda - dT - kTV \ , \tag{1.1}$$

$$\frac{\mathrm{d}I}{\mathrm{d}t} = kTV - \delta I \ , \tag{1.2}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = pI - cV \ , \tag{1.3}$$

where λ represents the production rate of target cells, d and δ are the death rate of target and infected cells respectively, c and p are, respectively, the clearance rate and production rate of virions by infected cells, and k is the rate of infection of target cells by virions. This model is widely used with minor or major modifications to study the dynamics of various viral infections. Typically, these mathematical modelling efforts seek to determine crucial parameters of the dynamics of a specific viral infection which would be impractical or arduous to extract experimentally.

But those simple ODE models make the very important assumption that the various populations of cells and virus particles are uniformly distributed over the space where the infection takes place for all times; an assumption that is rarely realistic, and which may or may not affect in a significant way the resulting dynamics. For this reason, there is growing interest in probing the effect of spatial distribution on systems in ecology [21, 22, 88], epidemiology [34, 47] and immunology [29, 48, 76]. In the case of viral infections, for example, spatial structures emerging over the course of an infection, such as localized populations of dead cells, might affect the spread of infection, similar to the manner in which a counter-fire can stop a forest fire from spreading. The investigation of the effects of these emerging spatial structures on viral infection dynamics is the main motivation behind the research presented herein, which covers various aspects of spatiotemporal modelling of viral infections.

In my research, CA modelling was preferred over differential equation modelling. One of the advantages of using differential equations is that a lot is known about their behaviour. Given a set of differential equations, one can often tell, for instance, for which range of the parameters the system is in a stable or an oscillatory state and for which value of the parameters the system will bifurcate from one state to the other (see [77] for a good introduction to the theory of dynamical systems and differential equations). However, modelling self-organizing systems using differential equations is not always advantageous. The set of differential equations necessary to represent such systems can be too complex for an analytical solution to be obtained, perhaps because the set of equations is too large or because of nonlinearities or delays.

For cases where the differential equations can no longer be solved analytically, CA can be a useful modelling alternative. The penalty for using the latter method is that CA modelling is a fairly new method and although CA provide a natural way to model self-

organizing systems, they give little information on the landscape of the global dynamics in the parameter space. This information has to be extracted through averaging many simulations over the whole parameter space. On the other hand, CA can be simpler to implement, and adding nonlinearities, delays or additional compartments does not introduce any new difficulties in solving the model. Further, the description of the components and rules that make up the CA is often done in the same language as is used to describe the real system, such that the approximations made to simplify the simulations are usually more physical in character than mathematical. Finally, a CA, by its very definition, emphasizes the local dynamics of a system rather than representing averages of its dynamics, as is the case with the mean-field approach of differential equations. A CA model allows one to witness the subtle rules at play in the model and it is a good starting point to understanding the impact of local interactions on the resulting global dynamics of a complex system. This is why I chose CA to implement spatiotemporal models of viral infections.

In order to facilitate the implementation and visualization of CA models for the research project, I implemented MASyV, a Multi-Agent System Visualization platform. Its implementation and usage are described in Appendix A. In short, MASyV facilitates the visualization of the CA models and automates certain tasks such as batch runs, data logging to file, recording of the visual simulations to compressed movie formats, etc.

As was mentioned earlier, the present document covers various aspects of spatiotemporal modelling of viral infection for the purpose of investigating how emerging spatial structures affect the dynamics of viral infections. The first part of this document offers a survey of the literature of relevance to the research projects presented herein. More specifically, Chapter 2 introduces the main components of the immune system and offers a brief overview of the mechanisms of immune responses, with special attention to viral infections, particularly to influenza A. Then, Chapter 3 covers the basics of CA theory, including early work on elementary one-dimensional CA and the statistical properties and applications of CA. Finally, Chapter 4 reviews the literature on spatiotemporal models of immune processes, from the complex bit-string models to simpler CA toy models and models specific to a particular viral infection.

Then, as a first step towards investigating the effects of the spatial distribution of agents on localized viral infection, a simple two-dimensional CA model was developed. It was validated against clinical immunological data for uncomplicated influenza A infections and shown to be accurate enough to adequately model this disease. This model is the subject of Chapter 5.

In Chapter 6, the simple two-dimensional CA model is used to investigate the effects of relaxing the well-mixed assumption on viral infection dynamics. Particularly, the effects of the initial distribution of infected cells, the regeneration rule for dead epithelial cells, and the proliferation rule for immune cells are explored and shown to have an important impact on the development and outcome of the viral infection in the model.

Finally, in Chapter 7, the movement of T cells within lymph nodes in the absence of antigen is investigated. Based on individual T cell track data captured by two-photon microscopy experiments in vivo, a simple model is proposed for the motion of T cells. This is the first step towards the implementation of a more realistic spatiotemporal model

of HIV than those proposed thus far [76, 90].

Chapter 2

Immunology and Virology

In order to be able to model any self-organizing system adequately, one has to learn what constitutes the system of interest and what rules govern the movements and actions of these constituents. Here, a brief overview of the crucial components of the immune system will be introduced. This is a good chance for the uninitiated reader to collect some of the immunological knowledge necessary to understand the models that will be introduced later in this document. For a great overview of the immune system from the point of view of a physicist, see [66] and for an in-depth coverage of the subject, see [30].

2.1 The Specific and Non-specific Immune Systems

Antigens are molecules present on cells (both foreign and self) with which the immune system's cells and molecules can interact. "Foreign" antigens, components of the invading organisms, alert the immune system of invaders and initiate immune responses which can destroy the invading cells. To fight any antigen efficiently, the vast army of cells and structures that constitute the immune system is divided in two classes: the nonspecific (or innate) system and the specific (or acquired, adaptive) system.

The nonspecific system consists of anatomical barriers (skin, eyebrows, etc.), secretions (saliva, tears, etc.) and phagocytic cells such as macrophages, neutrophils and natural killer (NK) cells. As its name suggests, the defence mechanisms it carries are not specific to a particular antigen. In general, most of the microorganisms encountered by a healthy individual are readily cleared within a few days by the nonspecific defence mechanisms of the nonspecific system even before the specific immune system gets activated. It provides the first line of defence against any exposure to an antigen and will often serve as a backup force while the specific immune system builds up its specificity during a primary invasion, i.e. a first encounter of the system with a given pathogen. However, when an invading microorganism eludes the nonspecific system or is not cleared by it, the specific immune response of the specific system is triggered.

The specific immune system is composed mainly of lymphocytes (such as B and T cells), antibody molecules and other molecules produced by the lymphocytes. As its name suggests, this system has to be tailored to the specific intruder in order to be able

to fight it efficiently. The specific immune system's responses can be divided into two branches: the humoral and the cell-mediated response. Figure 2.1 illustrates both the humoral (left side) and cell-mediated (right side) response. The humoral response's main effectors are the antibodies. A humoral response entails the interaction of B cells with antigens and helper T cells, and the B cell's subsequent proliferation and differentiation into plasma and memory B cells, with the former producing great amounts of antibodies capable of binding an antigen to tag it for removal. The cell-mediated response's main effectors are T cells, both helper and cytotoxic T cells. Helper T cells are responsible for secreting cytokines enabling various phagocytic cells to phagocytose microorganisms more effectively, while cytotoxic T cells are responsible for killing altered (e.g. virus-infected) host cells (referred to as target cells).

The specificity of the specific immune system comes from the pattern recognition capabilities of the system. Both B and T cells possess receptor molecules on their surfaces that can recognize antigen or more specifically the antigen's recognizable part called the epitope. The T cell receptor is simply called a T cell receptor (TCR), while the B cell receptor is referred to as a surface or anchored B cell receptor when it is anchored in the surface of the B cell, and is referred to as an antibody (immunoglobin) when it is free (soluble) after having been secreted by an effector plasma B cell. Figure 2.1 presents a schematic view of the B and T cells' receptor.

Recognition of an epitope by a receptor occurs at a molecular level and is based on the length of the complementary regions between the strings typically made of amino acid[1] that constitute the receptor and the epitope. The particular amino acid string that composes a receptor determines the receptor's idiotype. Immunologists sometimes refer to the idiotype of a receptor as the "shape" of the receptor. It is important to point out that all the receptors on the surface of a given lymphocyte are made of the same amino acid chain and hence are said to all be of the same idiotype.

Receptors are constructed by a complex genetic process that insures that the receptors expressed on different lymphocytes have a different randomly chosen idiotype, hence maximizing the chance that any random antigen presented to the system can be recognized given that the repertoire of receptors is complete. For a clever calculation and discussion on the completeness of the immune repertoire, see [66]. The B cell receptor interacts with epitopes present on intact antigen molecules that may be soluble or bound to a surface. But since the function of a T cell is to kill or stimulate other cells, it needs to be able to recognize that it is interacting with a cell rather than a soluble molecule. To accomplish this, T cell receptors are designed to only recognize antigen when they are bound to a cell surface molecule called a major histocompatibility complex (MHC).

MHC molecules come in two classes. Class I molecules specialize in presenting proteins synthesized within the cells and are found on every cell (see right-hand side of Figure 2.1). Class II molecules specialize in presenting fragments of molecules picked up from the environment and are found only on professional antigen-presenting cells

[1]B cell receptors are mostly made of amino acids. However carbohydrates, lipids and even nucleic acids can serve as B cell epitopes. T cell epitopes are almost always made of peptides (i.e. a string of amino acids). For the purpose of our discussion, we will assume that the building blocks of all receptors and epitopes are amino acids.

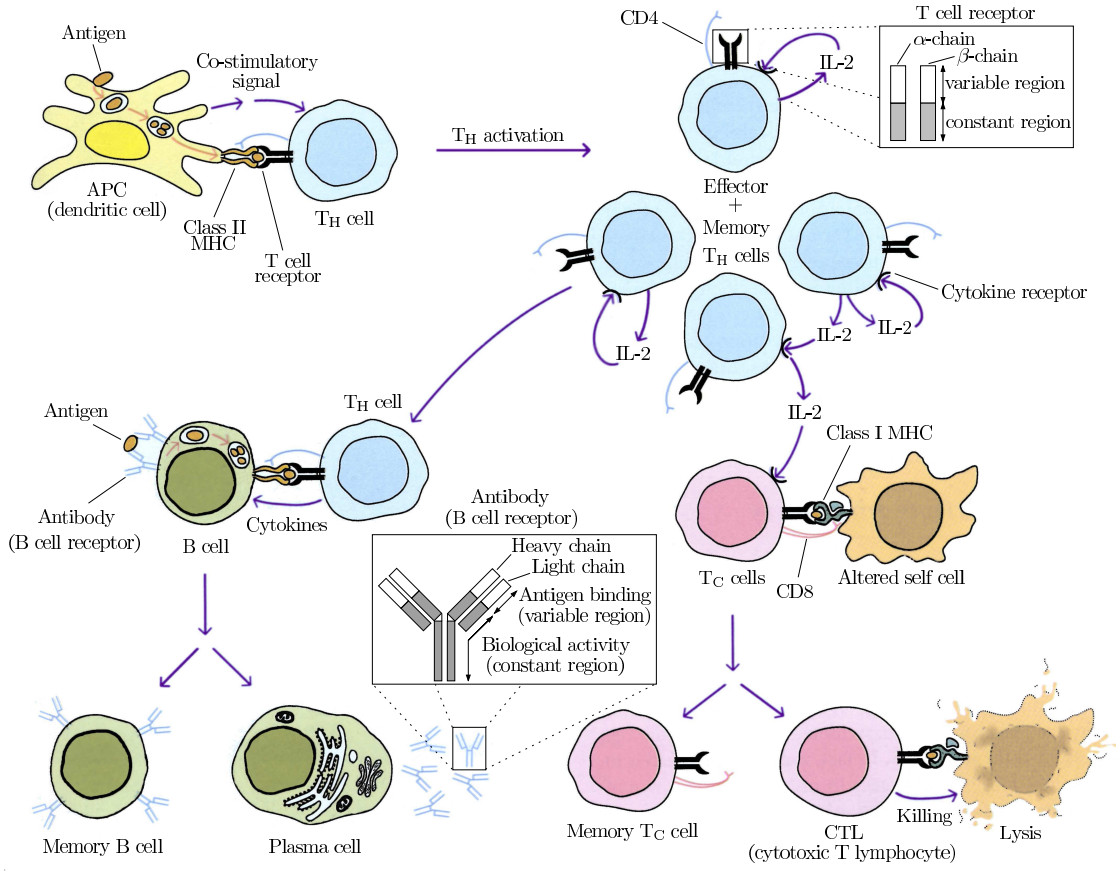


Figure 2.1: Representation of the specific immune response. The boxes show a schematic representation of the B and T cells' receptor. The left-hand side of the figure illustrates the humoral response while the right-hand side illustrates the cell-mediated response of the specific immune system. Figure adapted from [30].

(APCs) like B cells, macrophages and dendritic cells (see top-left corner of Figure 2.1). Both classes of MHC molecules bind peptides and present them to T cells. MHC class I typically presents peptides to cytotoxic T cells (Tc, $CD8^+$ T cell, or CTL for cytotoxic T lymphocyte) which then destroy the cell if it can bind the class I MHC-peptide complex. Class II MHC typically presents peptides to T helper cells (Th or $CD4^+$ T cell) which in turn secrete cytokines. Cytokines are proteins produced by various cells and serve to regulate the intensity and duration of the immune response by exerting a variety of effects on lymphocytes and other immune cells.

Once a B cell has successfully recognized an antigen and has been successfully bound by a helper T cell, co-stimulation will occur along with the release of cytokines which will initiate the division and differentiation of both the B cell and helper T cell into effector and memory cells. In the case of B cells, the effector cells are called plasma B cells and secrete large amounts of antibody of their idiotype. The antibodies' role is to tag cells and molecules as foreign. An antibody-antigen complex or a foreign cell with antibodies attached to its surface are quickly eaten by large phagocytic cells such as macrophages.

2.2 The Challenges Faced by the Immune System

The immune system constitutes only a few percent of the total cells in the body. During an immune response, the lymph nodes may swell to allow some increase in lymphocyte populations but that percentage cannot increase very much before affecting other bodily functions. Because of this size constraint on the immune system, specific strategies had to be adopted. Those include clonal selection, learning and memory, as well as self-nonself discrimination.

In the specific immune system, diversity of the receptors is necessary in order to maximize the chance of the immune system being able to recognize any random antigen presented to it. Affinity, i.e. the strength of the binding between two interacting agents of the specific immune system — a function of the compatibility between the cells' epitope and receptor — is also an important asset since the higher the affinity, the stronger the response. Diversity versus affinity is then the complex balance that the specific immune system has to always carefully maintain. One way the system accomplishes this is through clonal selection, the process by which only those cells that recognize the antigen are allowed to differentiate thus being selected against those which do not.

Clonal selection operates on both B and T cells, resulting in affinity maturation, i.e. the increase in average antibody affinity. Lymphocytes fight among themselves to bind an antigen resulting in the natural selection of clones (lymphocytes of a given idiotype) with a higher affinity for the antigen. For example, if two T cells can bind an antigen presented by an antigen presenting cell (APC), the one that can bind it better will generate a stronger response resulting in a larger clonal population carrying its idiotype. Note that when activated lymphocytes differentiate, point mutations (a mistake in the amino acid string) may be introduced in the receptor type of its clones such that their idiotype is slightly changed. Point mutation can result in the creation of higher affinity clones and can be seen as the perturbation method used by the immune system to explore

the affinity landscape to find higher affinity clones.

The specific immune system has also developed a capacity for learning and memory. Learning occurs through the process of clonal selection during which the lymphocytes that have proven themselves to be valuable by having recognized antigens see their population size increase. This introduces a bias of the repertoire from random towards a repertoire that more clearly reflects the actual antigenic environment. Upon primary invasion, often referred to as the original antigenic sin, the specific immune response may take days to build up. At that time, the immune response will be driven by the nonspecific system and by low affinity cells of the specific system. But upon secondary invasion, the immune system's response to the antigen has a larger amplitude and is faster than that of the primary invasion, suggesting that the system maintains some form of memory of the antigens it encounters.

Memory, like many other topics in immunology, is still not fully understood but it is likely the result of a combination of factors. It can be due to the fact that upon secondary infection with the same antigen, the repertoire that has been biased through affinity maturation during the first invasion now contains larger populations of high affinity clones. This would help avoid the delays seen in the primary invasion due to the fact that the cell population with high affinity for the pathogen had to be enlarged before substantial amounts of antibody could be secreted. Here, as in a typical predator-prey situation, the size of the lymphocyte subpopulation with high affinity for a specific antigen relative to the size of the antigen population is crucial in determining the outcome of infection. Memory can also be attributed to differences between naive cells and cells that have already encountered the antigen. For instance, memory cells may be easier to trigger than naive cells. Cells that have never been triggered are small and contain little cytoplasm. When triggered they make at least 50 new proteins. One would expect many of these molecules to remain in the cell, thus making subsequent triggering events easier and faster [66].

While being able to recognize any random antigen that is presented to the immune system is a highly desirable feature, it does not come without a price. Self-nonself discrimination is perhaps the most formidable challenge the immune system has to face. It has to be able to recognize as many antigen as possible while constantly working to avoid autoimmunity, i.e. an immune response against self. For this, the immune system has developed a "training period" for immature lymphocytes. New lymphocytes are created in the bone marrow [2]. B cells are immediately put in circulation while the T cells migrate to the thymus — hence the name, T cells — to begin what is referred to as their "thymic education". There, they will encounter self as well as nonself antigens and will be selected negatively for their ability to recognize self peptides presented by self MHC and positively for their ability to recognize nonself peptides presented by self MHC. It is also in the thymus that the T cells will evolve to typically become either helper T cells (Th) or cytotoxic T cells (Tc) by respectively expressing $CD4^+$ or $CD8^+$ membrane molecules on their surface. There is some level of deletion of self-reactive B

[2]Lymphocytes are also created during an immune response when cytokines initiate cell division and differentiation. However, since those are created from successful lymphocytes clones, they are very unlikely to recognize self.

cell clones taking place in the bone marrow, where B cells are produced — hence the name, B cells. But because B cells are required to bind with helper T cells in order to initiate differentiation into antibody producing plasma cells and memory cells, their selection is also made through helper T cell selection. Unsuccessful B cells, i.e. B cells that are not bound by any helper T cells, will eventually die without ever being activated. But nothing is perfect and the threat of an autoimmune disease is never fully eliminated.

2.3 Viral Infections

The modelling efforts presented in this document concentrate on the dynamics of viral infections. More specifically, experimental data of the dynamics of influenza A infections was used to calibrate the proposed viral infection model. For this reason, this section concentrates on the specific host-pathogen mechanisms involved in a viral infection, and more specifically, in an influenza A infection.

A viral infection is an infection caused by virus particles, referred to as virions. A virion does not have the ability to reproduce itself as it does not contain the necessary machinery. To proliferate, a virion needs to "borrow" the machinery of a host cell, also referred to as a target cell, and make use of the cell's mechanisms and building materials to create copies of itself. If the reproduction of the virions solicits too much of the host cell's resources, it can result in the death of the host cell.

A viral infection is initiated when virus particles, which have successfully escaped non-specific immune detection, enter a host cell. The adhesion and absorption process of the virions by the host cell is a complicated mechanism and varies across host cells and virus types. Once the virions are inside the host cell, they begin reproducing. The viral infection spreads as the newly produced virions are released from the host cell and move on to infect neighbouring cells. Release can occur as a sudden burst, as a continuous flow, or both. If the cell undergoes apoptosis (cell suicide) as a result of viral toxicity or immune attacks, virions will be released suddenly as a burst. But if the virions have the ability to extravasate out of the cell, they can be released in a continuous manner, as long as the host cell keeps producing new virions.

A viral infection is typically fought on two fronts by the two branches of the immune response: the humoral and the cell-mediated responses. The humoral response is responsible for the clearance of virions in order to prevent further infection. This is done through the action of the antibodies secreted by the plasma B cells. The cell-mediated response is responsible for the destruction of infected cells in order to eliminate the sources of virions. This is carried out by the cytotoxic T cells and the macrophages which respectively kill and phagocytose the infected cells.

2.3.1 Influenza A Viral Infections

Infection Characteristics

Influenza, in humans, is caused by a virus that attacks mainly the upper respiratory tract, the nose, throat and bronchi and rarely also the lungs. According to the World

Health Organization (WHO), the annual influenza epidemics affect from 5% to 15% of the population and are thought to result in between three and five million cases of severe illness and between 250,000 and 500,000 deaths every year around the world. Most deaths currently associated with influenza in industrialized countries occur among the elderly over 65 years of age [87].

Influenza viruses are divided into three groups: A, B, and C. Additionally, influenza viruses are defined by two different antigens present on their surface. They are proteins with spike-like features called haemagglutinin (H) and neuraminidase (N). Influenza A has two subtypes which are important for humans: A(H3N2) and A(H1N1), of which the former is currently associated with most deaths. Antibodies to H are strain-specific and neutralize the infectivity of the influenza A virus, while antibodies to N have a less protective effect. Thus, it is suggested in [11] that the antigenic properties of influenza A virus be associated with the H determinant.

It is the high mutation capability of the influenza virus that makes it a great public health concern. The genetic makeup of influenza viruses allows for frequent minor genetic mutations to take place. This makes it necessary to constantly monitor the global influenza situation in order to adjust the influenza vaccines' virus composition annually to include the most recent circulating influenza A(H3N2), A(H1N1) and influenza B viruses. Additionally, influenza A viruses, including subtypes from different species, can swap, reassort, and merge genetic material resulting in novel subtypes. Three times in the last century, influenza A viruses have undergone major genetic changes mainly in their H-component, resulting in global pandemics and large tolls in terms of both disease and deaths [86].

Infection Milieu

Influenza A is a viral infection in which virions infect the epithelial cells of the upper respiratory tract and major central airways. It is characterized by desquamation (shedding or peeling) of the epithelium of the nasal mucosa, the larynx, and the tracheobronchial tree. The airway epithelium consists of a single layer of cells (everywhere except in the trachea) [68] which is made up of at least four major cell types, including basal cells (progenitor cells), ciliated cells, goblet cells and Clara cells [61].

From below (basal side), the cells are attached to the basement membrane which is not affected by influenza infections [68]. From above (apical side), the cells are bathed in 2 distinct layers of fluid: a 7–10 μm periciliary fluid layer, topped by a < 2 μm layer of mucus[1]. The periciliary fluid (sol phase) must be watery enough to allow cilia to move freely and the mucus (gel phase) must be thick and elastic enough to get the cilia to engage the mucus gel layer pushing it out of the lungs. Through the action of the beating cilia, the mucus gel layer is swept from the airways towards the trachea at surface velocities of 1 – 10 mm/min [61]. The exact viscosity of the periciliary fluid is unknown, but it is assumed to be similar to that of serum or plasma, a dilute mixture of salts and proteins[2], and is typically about 1.43 cP [50]. The viscosity of mucus is orders of magnitudes greater than that of water and can easily reach 1–100 P[1]. The infection

milieu is illustrated in Figure 2.2.

Infection Dynamics

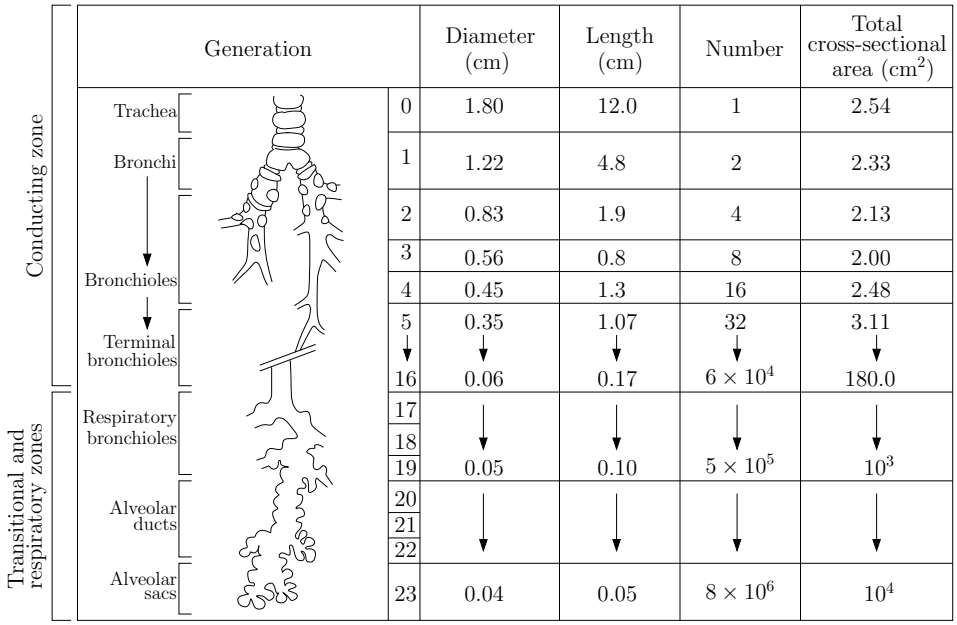
An influenza A infection is typically caused by inhalation of respiratory droplets from infected persons. These droplets which contain influenza virions (virus particles) then land on the mucus blanket lining the respiratory tract [68]. Many virions are destroyed by non-specific clearance such as mucus binding. The remaining virions escape the mucus and attach to receptors on the surface of target epithelial cells. Infection is initiated by adsorption of the virions to the cell surface, which results in receptor-mediated endocytosis of the virus particles approximately 20 min after infection [68]. Once inside the cell, the virions begin replicating, using the machinery and building materials that would normally be used by the host cell to maintain its functions. Virus budding, which takes place only at the apical surface membrane of infected cells [58], can be detected 5–6 h after infection, and is maximal 7–8 h after infection [68]. Virions are released at a rate of approximately 10^2–10^4 d^{-1} [11] and move on to infect neighbouring cells, repeating the infection process. Virus titer (virion concentration) peaks within 2–3 d and is cleared within 6–8 d post-infection [1]. Infected cells are destroyed as a result of the cytopathic effect of the virions and as a result of the immune response, with a lifespan of $\sim$ 24 h. At the peak of the disease, about 30%–50% of the epithelium of the upper airways is destroyed [11]. Cellular regeneration of the epithelium begins 5–7 d after infection but complete resolution can take up to one month [1].

Immune Response

The immune response to influenza virus infection involves both the humoral and cellular branch of the immune system. The humoral branch's major contribution is through the production of antibodies which clear free virions. Those antibodies are most often specific to the HA (haemagglutinin) and NA (neuraminidase) surface proteins of influenza virions. Unfortunately, these surface proteins vary widely between different virus strain, and thus, antibodies confer limited long term immunity or cross-reactivity to other subtypes of influenza A virus [56]. This is why influenza A vaccines have to be updated every year.

The cellular branch's contribution is dominated mainly by the action of cytotoxic T lymphocytes (CTL) which recognize and destroy infected cells. The CTL response offers hope for long term immunity because CTLs recognize the conserved epitopes of internal proteins of influenza virions, such as nucleoprotein (NP) and matrix protein (M) which are shared by influenza A subtypes [60]. It has been shown that CTL can limit influenza A virus replication and protect against lethal influenza A challenge in murine (mice) models [38]. However, CTLs also have the potential to cause immunopathology (immune system attacking the host) [15] and thus their response has to be carefully dosed. There is growing interest in the development of vaccine that would not only

[1]Michelle Dawson, Personal communication by email, 28 October 2004.
[2]Malcolm King, Personal communication by email, 28 October 2004.



Generation		Diameter (cm)	Length (cm)	Number	Total cross-sectional area (cm^2)
Trachea	0	1.80	12.0	1	2.54
Bronchi	1	1.22	4.8	2	2.33
	2	0.83	1.9	4	2.13
	3	0.56	0.8	8	2.00
Bronchioles	4	0.45	1.3	16	2.48
	5 ↓ 16	0.35 ↓ 0.06	1.07 ↓ 0.17	32 ↓ 6×10^4	3.11 ↓ 180.0
Terminal bronchioles					
Respiratory bronchioles	17	↓	↓	↓	↓
	18				
	19	0.05	0.10	5×10^5	10^3
Alveolar ducts	20	↓	↓	↓	↓
	21				
	22				
Alveolar sacs	23	0.04	0.05	8×10^6	10^4

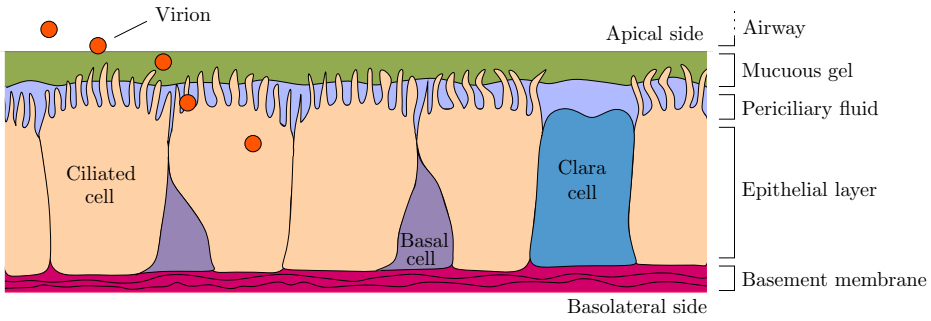


Figure 2.2: Physionomy of airways, modified from [61] (top panel) and cross-section of the lung epithelium (bottom panel).

elicit a strain-dependent neutralizing antibody response but would also efficiently prime strain-independent virus-specific memory cytotoxic T lymphocytes [31, 59].

CTLs expansion and contraction has been the subject of many studies [41, 78, 2, 49]. It is now believed that once activated, CTLs undergo a period of massive clonal expansion followed by a contraction phase in which 90% – 95% of activated CTLs die while the remainder differentiate into memory cytotoxic T cells [49]. This differentiation programme, often referred to as T cell programming, appears to depend on the condition of initial priming of the naive T cell but otherwise be independent of antigenic concentration. Once activated, the T cell will undergo intense proliferation even when the antigen is removed from the system [41, 78], and similarly, T cell contraction takes place whether the antigen has been cleared or still remains in the system [2]. After initial activation of a naive cytotoxic T cell, it is believed that it takes approximately 24 h before the first division occurs, and subsequent divisions are believed to occur every 5–6 h [78]. After a certain number of divisions, an activated cell will have acquired effector function and after approximately 15 divisions [9], it will either die or differentiate into a memory cell.

Chapter 3

Cellular Automata

Cellular automaton (CA) models are very powerful tools commonly used in statistical mechanics (e.g. the Ising spin model [46]). While statistical mechanics is good at describing systems where the number of agents, N, is very large (such that it is possible to take the continuum approximation), it gives much poorer results when N is not large enough to take the continuum approximation. Furthermore, statistical mechanics is very appropriate for defining a system at its equilibrium state but soon becomes too complex when the system is out of equilibrium. Finally, statistical mechanics is very limited at describing systems of strongly interacting agents. When interactions are not weak and therefore cannot be neglected, it might still be possible to explain a system in terms of statistical mechanics, but the formalism will be very particular and extremely limited in its applications. It goes without saying that this would involve numerous approximations which might lead to unsatisfying results. CA have proven extremely useful in modelling those kinds of systems and usually give very acceptable results. What makes the beauty of CA is that through the implementation of simple local rules, one can reproduce the behaviour exhibited by the highly complex system modelled. The level of interactions involved in self-organizing systems is usually quite well described by CA.

Cellular automata were originally introduced by John von Neumann and Stanislaw Ulam under the name of "cellular spaces" as possible idealization of biological systems. They sought to show that biological processes such as the reproduction and evolution of organized forms could be modelled by simple cells following local rules for changing a cell parameter with time [35]. Traditional CA usually consist of a regular uniform N-dimensional grid that can either be finite or infinite (periodic boundaries) in extent. The grid contains a discrete variable (or cell) at each site that can assume m possible discrete values. The state of a CA is completely specified by the values of all variables at each site. The CA evolves in discrete space with discrete time steps with the value of a variable at a given site being affected by the values of variables at sites in its neighbourhood at the previous time step. The neighbourhood of a site can be defined in numerous ways and can be as extended as one wishes. At a new time step, the variables at all sites are updated based on their own value and that of their defined neighbourhood at the preceding time step according to a definite set of local rules.

3.1 Elementary One-dimensional Cellular Automata

Stephen Wolfram has done most of the early work on CA. In the '80s, he published an extensive series of articles on CA [83, 84, 85]. He concentrated most of his work on elementary CA, namely one-dimensional CA with only two possible values at each site: 0 or 1. Typically, the neighbourhood he chose was defined to be the cell itself plus its two immediate neighbours: the cells to its left and right. This type of CA as been thoroughly studied by Wolfram and will be the main subject of this section.

From the definition of the neighbourhood, it follows that the local rules for the evolution of the CA will have to be defined for each combination of states of the 3 sites involved in defining the subsequent state of a single site. For the sake of clarity, let us label lc, cc and rc — for left, centre, and right cell, respectively — the 3 cells involved in the determination of the state of cc in the next time step. The set of possible configurations of lc, cc, rc is then defined as, for example,

$$\begin{array}{cccccccccc} lc, cc, rc & 111 & 110 & 101 & 100 & 011 & 010 & 001 & 000 \\ cc & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{array} \quad , \tag{3.1}$$

where the bottom line indicates the state of cc at the next time step given the top line's lc, cc, rc configuration at the previous step. This means that any rule can be defined by an eight digit binary number where each digit would represent the next state of the cc given its current state and that of its immediate neighbours. The rule detailed on the second line of (3.1) is called "Rule 90", from the binary number 01011010 defining it. Recall that a binary number is read in the following manner:

$$01011010 = 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 90 \ . \tag{3.2}$$

For a binary CA based on a 3-cell local rule, there are $2^3 = 8$ possible configurations for which a result must be specified which makes for a total of $2^8 = 256$ possible rules. However, those rules are usually reduced to a total number of 32 "legal" rules. The legal rules come from the fact that the CA were initially used for modelling biological processes such as the reproduction of organized forms. This meant, for example, that a starting configuration of all-zero sites, corresponding to the absence of any cells, should remain an all-zero configuration as it evolves in time — the triplet 000 should always give 0. Secondly, the rules should be symmetric so that the left and right cells are interchangeable. This forces the rules for configurations 100 and 001 and for configurations 110 and 011 to be equal [84]. Thus, the legal rules can be written as

$$\alpha_1\alpha_2\alpha_3\alpha_4\alpha_2\alpha_5\alpha_4 0 \ , \tag{3.3}$$

where each α_i can be 0 or 1, hence the $2^5 = 32$ legal rules.

If one decides to start the one-dimensional elementary CA with a single non-zero seed at its centre, interesting patterns develop. Based on these patterns, the 32 legal rules can be divided into 4 types as described in Table 3.1. Example patterns formed by rules from each of the 4 types are presented in Figure 3.1. Note that each of the 4

	patterns	rules
Type 1	disappear with time	0, 32, 72, 104, 128, 160, 200, 232
Type 2	single non-zero seed forever	4, 36, 76, 108, 132, 164, 204, 236
Type 3	uniform type of rows	50, 122, 178, 222, 250, 254
	uniform pairs of rows	54, 94
Type 4	nontrivial patterns	18, 22, 90, 126, 146, 150, 182, 218

Table 3.1: Classification of the 32 legal rules according to the type of patterns they developed when the one-dimensional CA is initialized with a single non-zero seed.

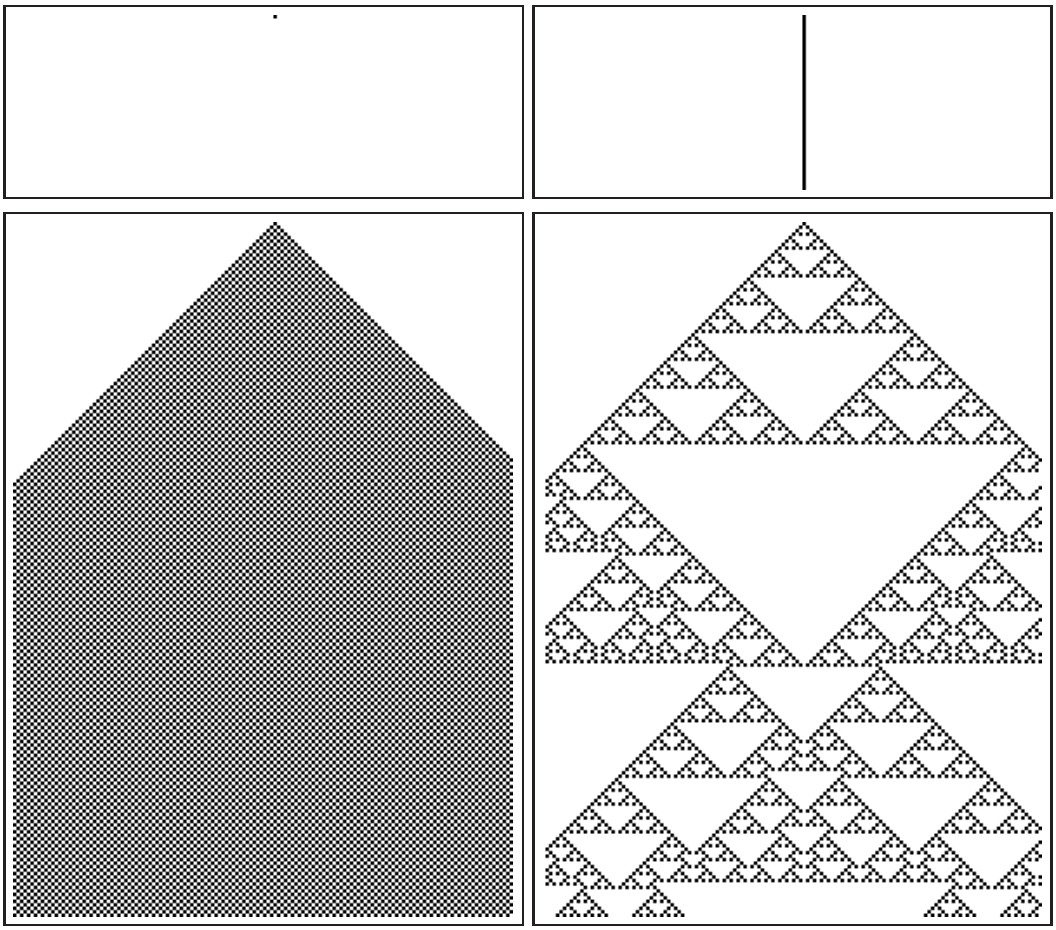


Figure 3.1: Evolution of a one-dimensional CA initialized with a single seed and submitted to Type 1 Rule 232 (top left), Type 2 Rule 36 (top right), Type 3 Rule 178 (bottom left), and Type 4 Rule 90 (bottom right). Each row represents the state of the 150-sites periodic CA at a given time step with black representing 1s and white representing 0s. The top row of each panel constitutes the initial single-seed configuration and subsequent rows represent successive time steps.

types contains 8 rules. This is no coincidence. What determines the "personality" of the outcome of a certain rule is the combination of its local rules. A closer look at the underlying local rules that are common to the rules of a same type reveals how the local rules affect the resulting structures.

Type 1: ($\alpha_4 = 0$ and $\alpha_5 = 0$) $\rightarrow$ all single-seed states (100, 010, 001) map to 0, leading to a permanent all-zero state.

Type 2: ($\alpha_4 = 0$ and $\alpha_5 = 1$) $\rightarrow$ the single-seed states 100 and 001 map to 0 while 010 maps to 1, leading to the permanent single-seed state.

Type 3 & 4: ($\alpha_4 = 1$) $\rightarrow$ both 100 and 001 map to 1 allowing the initial single-seed to disperse to both neighbour sites, leading to complex patterns.

where the α_i's are as in (3.3).

The three first types are said to be "simple" CA. The last type however, contains more elaborate CA that develop nontrivial patterns. At infinite time, those nontrivial patterns become self-similar fractals.

3.2 Statistical Properties of 1D Cellular Automata

Let us imagine a single row of N lattice points, where each lattice point represents a particle. Let us say that the values they can take is 0 or 1 indicating, for example, the spin orientation. A microstate of this system is a row of N particles with a given sequence of spins represented by the values 0 and 1. In total there are 2^N possible microstates all with an equal probability of occurring, provided that 0s and 1s have an equal probability of occurring. From this definition, it is possible to define a macrostate as the set of all N-site microstates containing x cells of value 1. If one of the microstates is submitted to a specific rule, for example Rule 90 as in (3.1), a new microstate will result. The resulting microstate (2nd line) when Rule 90 is applied to an original microstate (1st line) is

$$\begin{array}{rccccccccccc} \text{Original :} & (0) & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & (0) & \\ \text{Resulting after Rule 90 :} & & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & & \end{array} . \tag{3.4}$$

Note that periodic boundary conditions are used and denoted by adding in parenthesis the left-most state to the right of the right-most state and the right-most state to the left of the left-most state.

3.2.1 Irreversibility

Given a starting microstate, time evolution will result in a trajectory through subsequent microstates. The trajectory is dictated by the rule and may or may not lead back to the starting microstate. Assuming each microstate exists in a starting ensemble of configurations with equal probability of occurrence — again, provided that 0s and 1s have an equal probability of occurring — time evolution will likely merge many configurations

into a few and soon trajectories starting from almost all initial configurations are concentrated onto short cycles, called attractors, containing only a few configurations [35]. The trajectories traced out by the time evolution of several microstates may coalesce, but may never split. For instance, the extremely trivial Rule 0 will turn any starting microstate into the null microstate (all lattice points are 0), so that in one time step, all configurations reduce to just one: this is called a stable fixed point.

Almost all of the rules seen in Section 3.1 are irreversible, which means that although it is possible to identify a unique resulting microstate for each original microstate, it is not possible to identify a unique original microstate for each resulting microstate. In other words, a particular microstate has unique descendants, but does not necessarily have unique ancestors. Rule 204 is the lone exception since it is the identity rule: each microstate is mapped only to itself. For a reversible system, trajectories representing time evolution of different states never intersect or meet. Thus, for reversible systems, the total number of possible configurations will remain constant with time. In the case of an irreversible rule, the total number of possible microstates will be reduced as time evolves. For some systems, all trajectories might reduce to a fixed point solution (a single microstate like in the case of Rule 0), while in other cases, it might reduce to one or a few short cycles or attractors. Figure 3.2 presents all $2^4 = 16$ possible microstates of a 4-site elementary CA arranged around the attractors and fixed points that result from the successive application of a rule. It may be worth noting the work of Voorhees and Beauchemin in [79] where the impact of point mutations (single bit toggling) on the stability of these fixed points and attractors is explored.

3.2.2 Entropy

Irreversible behaviour in CA may be analyzed by considering the evolution of the system's entropy. Entropy is defined, as usual, as the logarithm (in this case base two) of the average number of possible states of a system, that is

$$S(\tau) = \sum_i p_i(\tau) \log_2 \left(\frac{1}{p_i(\tau)} \right) , \tag{3.5}$$

where $p_i(\tau)$ is the probability of obtaining microstate i after τ time steps. $\tau = 0$ corresponds to the starting equiprobable ensemble with all $p_i(0) = 1/2^N$. To understand what this entropy effectively means for an N-cell elementary CA, let us look at an example. Consider a 2-cell elementary CA. The starting equiprobable ensemble will then be composed of

$$\begin{array}{lccl} \text{Microstate 1 at time 0:} & 0 & 0 & \text{with } p_1(0) = 1/4 \\ \text{Microstate 2 at time 0:} & 0 & 1 & \text{with } p_2(0) = 1/4 \\ \text{Microstate 3 at time 0:} & 1 & 0 & \text{with } p_3(0) = 1/4 \\ \text{Microstate 4 at time 0:} & 1 & 1 & \text{with } p_4(0) = 1/4 \end{array} . \tag{3.6}$$

The entropy of this system at time zero, $S(0) = 4 \cdot (1/4) \log_2(4) = 2$, corresponds to the number of cells necessary to represent the 4 possible microstates.

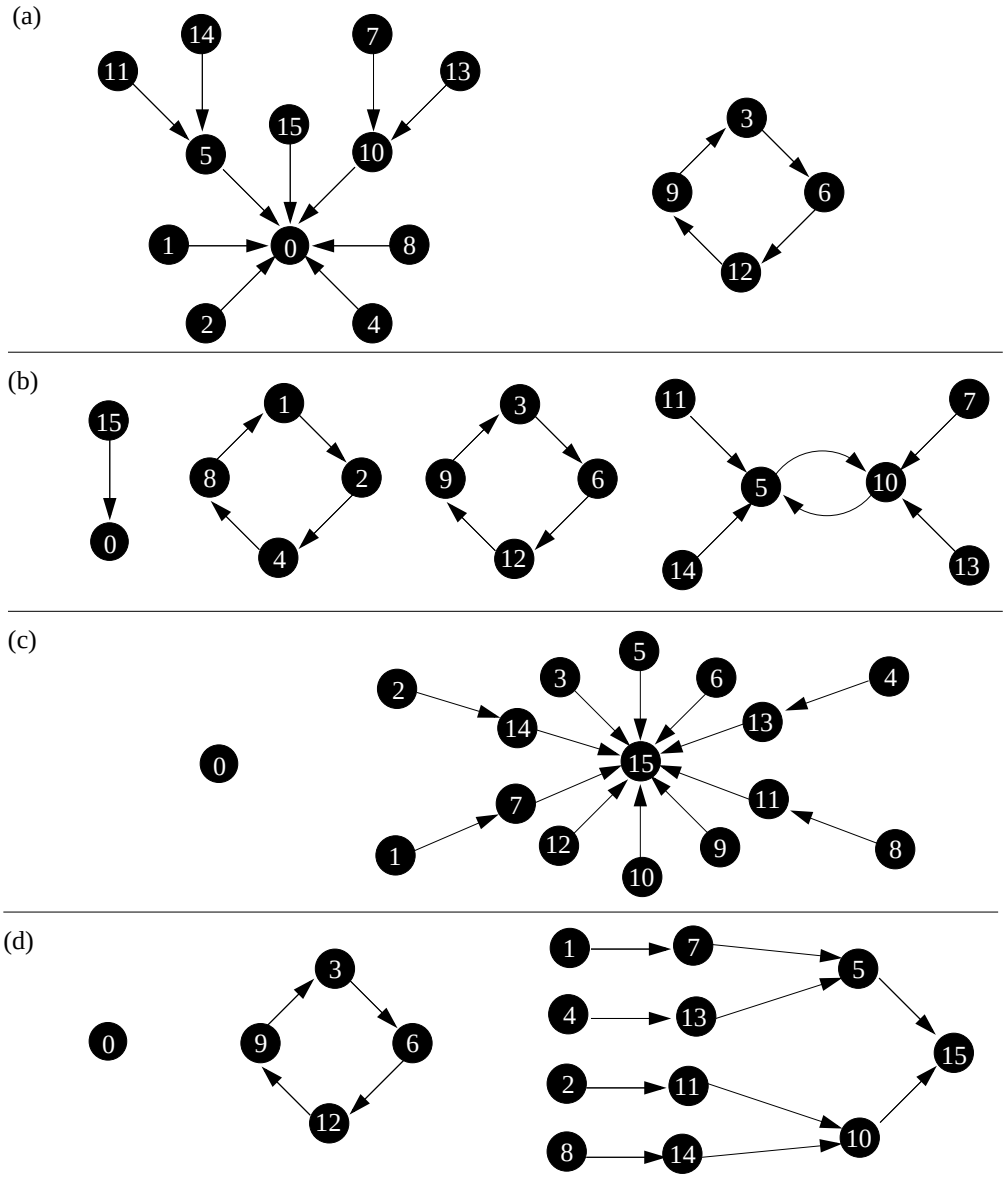


Figure 3.2: The $2^4 = 16$ possible microstates of a 4-site elementary CA arranged around the attractors and fixed points that result from the successive application of (a) Type 1 Rule 72, (b) Type 2 Rule 76, (c) Type 3 Rule 254, and (d) Type 4 Rule 182. Each node represents a microstate (e.g. $14 = 1110$) and the arrows represent the application of the rule to a microstate. In (a), for example, application of Rule 72 to microstate 14 (1110) yields microstate 5 (0101).

If the CA is left to evolve for one time step, i.e. $\tau = 1$, according to Rule 22 (00010110) for example, the microstate ensemble would reduce to

$$\begin{array}{cccc} (0)\ 0\ 0\ (0) & (1)\ 0\ 1\ (0) & (0)\ 1\ 0\ (1) & (1)\ 1\ 1\ (1) \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow \\ 0\ 0 & 0\ 1 & 1\ 0 & 0\ 0 \end{array} , \tag{3.7}$$

where the values in parenthesis are as in (3.4). Already, at time step $\tau = 1$ the system can no longer be found in configuration 11 and only three of the original four microstates remain. The distribution of microstates at $\tau = 1$ is

$$\begin{array}{lcll} \text{Microstate 1 at time 1}: & 0 & 0 & \text{with } p_1(1) = 2/4 \\ \text{Microstate 2 at time 1}: & 0 & 1 & \text{with } p_2(1) = 1/4 \\ \text{Microstate 3 at time 1}: & 1 & 0 & \text{with } p_3(1) = 1/4 \\ \text{Microstate 4 at time 1}: & 1 & 1 & \text{with } p_4(1) = 0/4 \end{array} . \tag{3.8}$$

The entropy of the system at time $\tau = 1$ is given by $S(1) = (2/4)\log_2(4/2)+2\cdot(1/4)\log_2(4/1) = 1.5$. This means that the system would effectively only need 1.5 cells to be fully described instead of the $\log_2(3) \approx 1.58$ cells that one would expect to need to represent a system with 3 microstates. The smaller entropy is due to the microstates not being equally probable — an expression of the system's tendency towards organization.

From this example, it is quite clear that elementary CA are capable of self-organization, tending to reduce in complexity as they evolve. Entropy is typically maximized when a system is completely disorganized, i.e. when all the microstates act independently. Thus, the entropy of an N-site CA takes on its maximal value of N (or $\log_2(2^N)$) for an equiprobable ensemble. For an irreversible system, one can expect the entropy of the system to go down and eventually stabilize to a given value as τ increases, i.e. once all microstates have reduced to a few attractors or fixed points. Figure 3.3 presents a graph of the entropy per site, $S(\tau)/N$ as τ increases, for the rules presented in Figure 3.2. It is interesting to compare the attractor/fixed point structures of the rules in Figure 3.2 to the evolution of their entropy in time. Type 2 Rule 76, for example, has many attractors. This allows it to stably maintain 11 of its $2^4 = 16$ original microstates yielding a high stable entropy. On the other hand, Type 3 Rule 254 condenses the 16 original microstates into only 2 stable fixed points in only 2 time steps, yielding a low stable entropy.

The second law of thermodynamics tells us that isolated microscopically reversible physical systems tend with time to states of maximal entropy and maximal disorder. However, dissipative systems involving microscopic irreversibility, or those open to interactions with their environment, may evolve from disordered to more ordered states. It is the CA's ability to capture the self-organizing nature of irreversible and open systems which makes them such a powerful tool to study such systems. The outlines of snowflakes, the patterns of flow in turbulent fluids, and many biological systems are but a few examples of systems which are well modelled by CA.

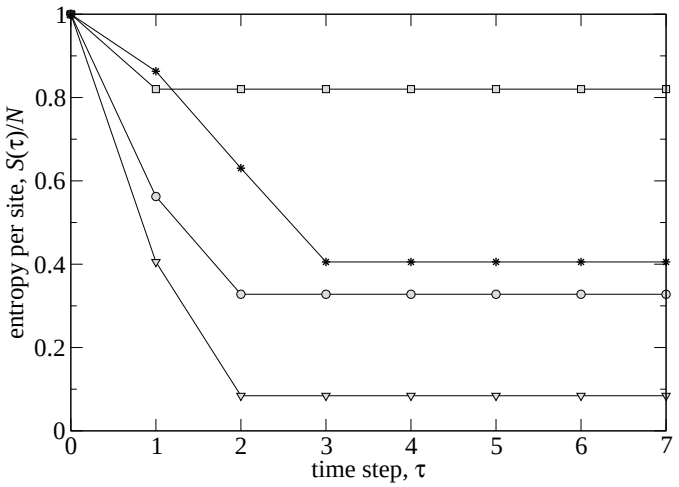


Figure 3.3: Entropy per site $S(\tau)/N$ as τ increases for a 4-site ($N = 4$) elementary one-dimensional CA evolving according to Type 1 Rule 72 (circles), Type 2 Rule 76 (squares), Type 3 Rule 254 (triangles), and Type 4 Rule 182 (stars).

3.3 Beyond One-dimensional Cellular Automata

Since Wolfram's elementary one-dimensional CA, CA models have come a long way. Lattice gas simulations, for example, were proposed as an extension of CA. They are so named because of their original use in modelling fluids as particles which are restricted to moving between discrete positions in the CA lattice. In a lattice gas simulation, the sites are only the discretized space in which the variables are allowed to move and evolve, rather than representing particular states of the system. Among the lattice gas automaton are the famous HPP and FHP models which are discussed briefly below.

3.3.1 The HPP Lattice Gas Model

Although the HPP lattice gas model was developed in the 1970's, it was only in the 1980's that it was finally recognized as a cellular automaton [17]. The HPP model — named after its authors Jean Hardy, Yves Pomeau, and Olivier de Pazzis — consists of particles moving around on a two-dimensional square grid. It defines collision rules for particles and insures conservation of momentum and particle number. Since the HPP model only allows one particle to enter a lattice site with a given direction at any one time, four bits of information at each site are sufficient to define the system. Each bit indicates an incoming particle headed in one of the four possible directions, namely east, north, west, and south. For example, state 1101 at a site would mean that three particles are coming into that site heading in direction 1 (east), 2 (north), and 4 (south).

Each time step is composed of a collision phase and a diffusion phase. The collision phase consists of changing only the states involving the head-on collision of two particles,

namely

$$(0101) \rightarrow (1010) \quad \text{and} \quad (1010) \rightarrow (0101) \ . \tag{3.9}$$

This means that a head-on collision between two particles results in the particles being deflected on the axis perpendicular to the collision. All other collisions, including those involving the collision of three particles (0111, 1101, and 1110), remain unchanged in the collision phase. The diffusion phase involves exporting each bit to the appropriate neighbour, namely the first bit to the east neighbour, the second to the north neighbour, etc.

As a result of its exclusion principle, the HPP model is invariant under time reversal. This means that if one reverses the direction of motion of the particles, one can recover the initial conditions exactly, provided that no error was introduced (e.g. external addition of particles). The HPP model can be extended to model the escape of particles contained in an area through a small aperture. The wall of the container can be modelled by adding an extra bit to each site to mark the presence of the wall and a collision with the wall would result in the particle bouncing back to where it came from [17].

Although the HPP model yields very interesting dynamics, it has an important flaw. Its underlying square grid leads to anisotropy in particle motion which prevents the HPP model from yielding realistic results for the true dispersion of gas particles. The anisotropy of the HPP model was corrected by its descendant, the FHP lattice gas model.

3.3.2 The FHP Lattice Gas Model

The FHP model — also named after its authors Uriel Frish, Brosl Hasslacher, and Yves Pomeau — was introduced in 1986. Its main departure from the HPP model is its implementation on a two-dimensional hexagonal grid rather than a square grid. This corrects the anisotropy flaw of the HPP model (see [17] for the calculation). Of course, the rules also had to be adapted to fit the geometry of the new grid. In its simplest implementation [17], the FHP model is very similar to the HPP model. Each particle moves by one lattice site per time step, and only one particle is allowed to enter a site with a given direction at any one time step. Thus, six bits are sufficient to describe the system at each site, with each bit corresponding to a particle entering that site with a given direction.

As with the HPP model, a time step in the FHP model consists of a collision phase and a diffusion phase. The FHP model's collision rules affect the collision of three particles colliding at 120° angles, and the head-on collision of two particles. The three-particle collision rule is such that

$$(010101) \rightarrow (101010) \quad \text{and} \quad (101010) \rightarrow (010101) \ . \tag{3.10}$$

This means that the collision of three particles at 120° angles results in all three particles

bouncing back to where they came from. The two-particle head-on collisions,

$$(001001) \begin{array}{l} \nearrow (010010) \\ \searrow (100100) \end{array}, \quad (010010) \begin{array}{l} \nearrow (001001) \\ \searrow (100100) \end{array}, \text{ and } (100100) \begin{array}{l} \nearrow (001001) \\ \searrow (010010) \end{array}, \tag{3.11}$$

are such that the particles will both be deflected by 60°, either clockwise or counter-clockwise, with equal probability. The direction of the deflection can be selected randomly or the rules can be kept deterministic by choosing, for example, to deflect all particles in one direction for even time steps, and in the other for odd time steps [17]. Note that only a deterministic rule, like the one proposed by the latter method, will insure the model's invariance under time reversal.

The FHP model was shown by its authors to follow the behaviour described by the Navier-Stokes equation of hydrodynamics, within some limits [17]. However, the high viscosity of the model which is a consequence of the rules, and its discrete nature, which could only be remedied by going to very large scales, made the FHP model impractical to model hydrodynamics. Much work has been put into improving the FHP model and addressing some of its weaknesses (see [17] for a good overview).

Eventually, the lattice Boltzmann method was introduced, which relaxes the requirements for each site of the CA to be represented by bits, and instead proposes to have real numbers represent the probability for a cell to be in a given state. The lattice Boltzmann method has been widely applied to hydrodynamics, pattern formation, and other such problems.

3.3.3 Applying Cellular Automata to the Immune System

Cellular automata are good candidates for studying self-organizing systems as they themselves are self-organizing systems and hence constitute a natural way to represent such systems. A well known feature of CA computations is their ability to produce surprisingly complex behaviour from very simple rules. A CA can produce a range of periodic, chaotic and generally very complex behaviours with intricate spatial and temporal patterns [84]. Nonlinearities or time-delays are not intrinsically difficult to treat since they are either a consequence of the dynamics rather than a cause or are otherwise trivial to include in the context of the CA local rules. Another great advantage of CA is that their computational structure is inherently parallel and they can therefore be run on parallel computers very efficiently.

The earliest CA of the immune system neglected the microscopic details of the immune system's behaviour. Typically, a set of characteristic states like "in rest" or "infected" would be represented by a one-dimensional CA. The evolution of this system would be based on simplistic rules, such as those seen above for the one-dimensional Wolfram CA, which defined how the CA switches from one of those states to another (see [66], Section V for a brief overview of the most prevalent models of this type).

The first models to employ lattice Boltzmann-type CA to model the various components of the immune system were born in the early 90's. Among these are the bit-string

models which are covered in the next chapter.

Chapter 4

Mathematical Models in Immunology and Virology

Computational modelling is of growing interest in a number of fields. The growing attraction to such modelling techniques is clear: computers are getting evermore efficient and fast while the systems we are dealing with are getting evermore complex and diversified. Although computational biology has lately mainly concentrated on data-mining, notably in the context of genomic research, biological simulations have always been around and have recently been gaining momentum. Here, a few of the efforts made to model the immune system using cellular automata will be explored. In particular, the very complete bit-string models will be discussed. Then, we will turn our attention to simpler spatial cellular automaton models of viral infections which is the subject of the research presented herein.

4.1 Bit-string Models

As discussed above, in a real immune system, binding between, for example, a B cell receptor and an antigen's epitope will take place through the matching of the amino acid strings composing both the receptor and the epitope, and binding will occur only if a sufficient match between both amino acid strings is met. It is possible to represent the amino acid string using an alphabet of m characters where each character corresponds to a given amino acid. Then, any of a number of string-matching algorithms can be used to determine the degree of complementarity. For a string of length N, the repertoire size is given by m^N such that the greater m is, the larger the repertoire size will be. This clever idea for representing a receptor's or epitope's idiotype was first introduced by Farmer et al. [23] as a way to perform calculations for determining molecular complementarity and predicting the optimal size of an epitope — see [66] for a short overview of this calculation. But the greatest impact of the string representation introduced by Farmer et al. was its application to lattice gas simulations of the immune system.

Models combining lattice gas simulation with the string representation of the immune system's diversity offer great analogies with the real system. They are developed with

the goal to create a general immune simulator which could be used to run experiments *in silico*. The lattice gas simulation, which describes the movement of individual immune cells in space, takes into consideration the localized nature of events such as encounters, recognition events, and proliferation. The string representation permits a simple implementation of the diversity of the immune system and, within the framework of localized interactions, offers the opportunity to investigate processes such as the spatial spread of mutating viral strains, and local clonal selection.

For simplicity, typical lattice gas simulations making use of the string representation choose to build the epitope and receptor strings using an alphabet containing only two characters, namely 0 and 1. Consequently, those models are referred to as "bit-string" models. Since the receptors and epitopes of bit-string models are represented by strings of bits, these models have a repertoire size of 2^N, where N is the length of the bit string.

The first bit-string model of the immune system was introduced in 1992 by Seiden and Celada [13, 71]. They called their model IMMSIM. More accurately, IMMSIM23 is the original version of IMMSIM. It was written in the APL2 language by Philip E. Seiden and requires the IBM APL2 runtime environment. In its original version, IMMSIM contains T cells, B cells, and antigen presenting cells (APC) other than B cells, as well as antigen (Ag) and antibody (Ab) molecules and is concerned with the humoral response only [71]. The later version of the model, IMMSIM3, was made to include cytotoxic and helper T cells as well as epithelial cells in order to add the cellular response to complement the already existing humoral response [10, 12]. Then, Bernaschi and Castiglione [7] proposed a parallel version of IMMSIM coded in C which they named ParImm, and later CIMMSIM. Finally, a C++ tutorial version of IMMSIM based on CIMMSIM was developed by Steven Kleinstein.

4.1.1 Representing Receptor-specific Interactions

In bit-string models, much like in the real immune system, the immune components are characterized by their receptors. Here, B cells are composed of a B cell receptor and a class II MHC (major histocompatibility complex) molecule. T cells only have a T cell receptor, and antigen presenting cells (APCs) other than B cells only have a class II MHC molecule. Antibody molecules are typically made up of a single B cell receptor. Finally, antigen molecules are made up of segments of two different types: B cell epitopes and presentable peptide strings. In most applications of bit-string models so far, the antigen has been limited to one B cell epitope and one presentable peptide string. Figure 4.1 presents the schematics of those cells and molecules for 8-bit strings.

In a real immune system, B cell and T cell receptors are physiologically different. But because all receptors are implemented the same way in bit-string models, the distinction between various types of receptors is made by restricting what can bind with what through simple binding rules. The various binding rules have been designed as follows. In bit-string models, an antigen has two types of bit-string segments: an epitope which it presents to antibodies and to B cells' surface receptors; and a peptide string which the APC's class II MHC receptor will present to the T cell receptor. When an antigen's epitope is bound by an antibody, an antibody-antigen complex is formed and both entities

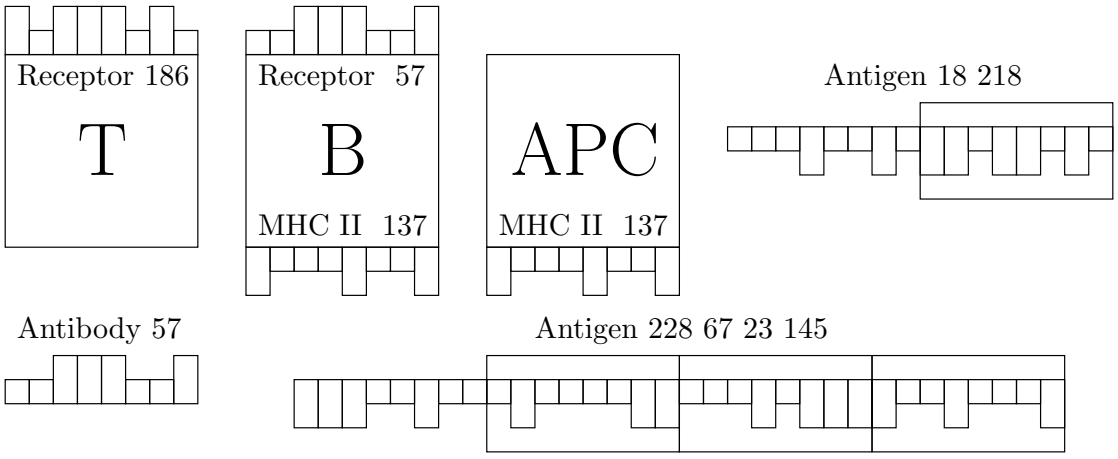


Figure 4.1: Examples of the bit-string models' schematic representation of B cells, T cells, antigen presenting cells (APCs) other than B cells, as well as antibodies and antigen molecules. On antigens, the epitopes are shown exposed while the presentable peptides are boxed. The receptors, MHCs, epitopes and peptides are numbered according to the decimal value of their 8-bit string. For example, the B cell's Receptor 57 is represented as (00111001), with 0s and 1s depicted as short and long blocks respectively. Figure adapted from [71].

are removed from the simulation. When an antigen's epitope is bound by a B cell's surface receptor, the antigen's peptide string is broken in two and whichever half matches the right half of the B cell's class II MHC molecule best will bind it. The other half of the antigen's peptide will be presented to encountered T cells along with the left half of the B cells' bare class II MHC molecule. Figure 4.2 illustrates this binding procedure. APCs other than B cells will bind any antigen with a fixed probability (e.g. 0.002 in [10]) and will present the antigen's peptide on their class II MHC molecule to encountered T cells in the same fashion B cells do. Finally, when a T cell encounters a class II MHC-peptide complex presented by an APC, depending on the probability of interaction, the T cell will bind with the presenting APC and will be activated. If the APC is a B cell, the B cell will also be activated. Activation results in differentiation at a constant rate over a fixed number of time steps. In the real immune system, an individual will have less that 10 different clones of class II MHC molecules. In the Seiden-Celada model, the simulations usually consider only 1 or 2 [10].

For simplicity, in most bit-string models, notably in the Celada-Seiden model [71], the function used to define a "match" between a pair of bits is simply an exclusive `OR` function (`XOR`), where 0 matches 1 and 1 matches 0. The `XOR` function is presented in Figure 4.3. From the number of bit mismatches for a given pair of strings, the probability of interaction is typically calculated as follows, although it does change from model to model depending on the question addressed and the author's preferences.

Interaction probability is characterized by 3 parameters, namely the minimum match,

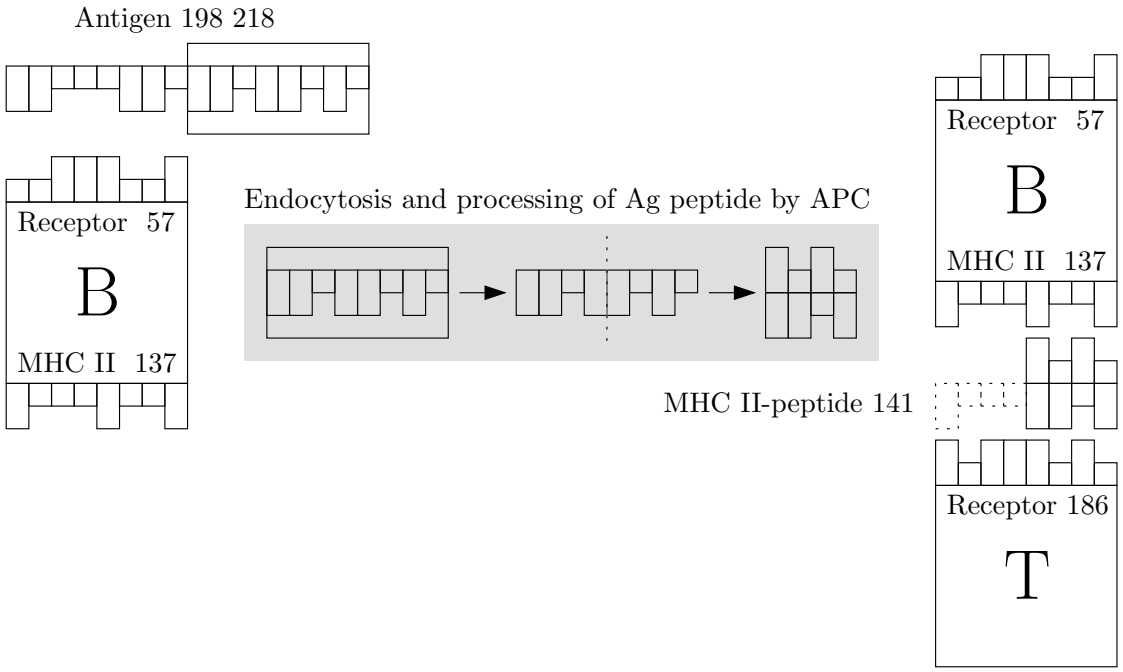


Figure 4.2: Schematic representation of the bit-string models' algorithm for the binding of an antigen to an antigen presenting cell (APC) and its presentation to a helper T cell. An antigen's epitope is bound by a B cell's surface receptor (left). The APC (in this case the B cell) endocytoses the antigen (Ag) peptide and breaks it into two halves (centre). The right half of the cell's class II MHC binds the peptide's half it is most compatible with (top right). The left half of the cell's class II MHC along with the remaining half of the peptide are presented to encountered T cells (bottom right). Figure adapted from [71].

0	XOR	0	=	0
0	XOR	1	=	1
1	XOR	0	=	1
1	XOR	1	=	0

	0	0	1	0	1	0	0	0	1	0	0	1	0
XOR	1	0	0	1	0	1	0	1	0	0	1	0	1
	1	0	1	1	1	1	0	1	1	0	1	1	1

Figure 4.3: (Left) the definition of the XOR rule. (Right) the XOR rule applied to a pair of 10-bit strings resulting in a 3-bit mismatch (3 null characters) between the two strings.

the minimum affinity and the affinity increase. The minimum match is the threshold of matching bits below which the interaction will not take place. The minimum affinity is the probability of interaction when the number of matching bits is exactly the minimum match, P(minimum match). Finally, the affinity increase determines the magnitude of the increase in affinity from an n-bit match to an $n+1$-bit match, and can be defined in a number of ways. In [43], for example, the affinity increase is such that

$$P(n+1) = \text{affinity increase} \times \frac{P(n) \times \left(\begin{array}{c} N \\ n \end{array} \right)}{\left(\begin{array}{c} N \\ n+1 \end{array} \right)} , \tag{4.1}$$

where N is the length of the bit-strings, n is the number of matching bits, $P(n)$ is the probability of interaction for n matching bits, and

$$\left(\begin{array}{c} N \\ n \end{array} \right) = \frac{N!}{n!(N-n)!} . \tag{4.2}$$

4.1.2 The Rules of the Bit-string Models

To give an idea of the computational scale of the simulation, it is interesting to mention that most simulations were run on relatively small grids, typically a 15x15 hexagonal grid[1] and the size of the bit-string, N, is typically chosen to be 8 yielding a clonal diversity of $2^8 = 256$. The time scale of a simulation is not always specified, but in [43, page 74] it is suggested that the relationship between a time step of the model and real time is such that one time step in the model corresponds to one B cell division. It is however not clear whether this is the standard way in which bit-string models define the time scale of the simulations, or if other authors have chosen different conventions. At this point, it is useful to describe what a time step consists of. This is slightly difficult because it may vary across the various bit-string model publications and it is not always specified by the authors. Thus, let us give an example by describing a time step as defined in [10].

Interactions only occur between entities that are allowed to interact and are located in the same grid site. Since at any given step, there will likely be more than one possible interaction for a given entity in a given grid site, all possible interactions are considered. Each interaction's probability is calculated and compared with a random number to determine whether the interaction is successful. An entity can have at most one successful interaction at any one time step such that only one of the successful interactions can take place. One can choose the interaction that had the highest probability or choose one of the successful interactions randomly. Once all interactions have been determined, they are allowed to take place. Then, entities are allowed to die, stimulated cells divide, new cells are born, and antibodies are generated. Finally, the entities diffuse to a randomly

[1] Note that in most articles, the authors mistakenly claim they were using a triangular grid, namely in [14, 43, 71]

chosen neighbouring grid site and that concludes one time step. From what the various publications seem to suggest, cells do not die after a given number of time steps, but rather die with a probability defined by their half life, as is the case in differential equation models.

Cell division will occur in B cells and T cells that have been activated. B cells can divide into memory cells with a longer half life and plasma cells that have a shorter half life but will generate great amounts of antibodies of the same idiotype as the B cell they originated from. When new B and T cells are "born" they are added at random locations on the grid and newly born T cells will have to undergo thymus selection. Thymus selection in the simulations is done through the positive and negative selection of T cells. A T cell whose receptor's left half recognizes a class II MHC molecule's left half with at most one mismatch, and whose receptor's right half recognizes antigen peptides, are positively selected. A T cell whose receptor's left half does not recognize the left half of the bare class II MHC molecules is negatively selected. This results in the B cell repertoire being complete, i.e. 2^N, while the T cell repertoire starts off complete but is then tailored through the thymus selection process.

The removal of antigen can be the result of various scenarios. When an antigen-antibody complex forms, both entities are removed from the simulation. Also, when an antigen has been recognized by a B cell and successfully presented to a T cell, resulting in the activation of both the B and T cells, the antigen is removed and the B and T cells are made to divide. The ratio of plasma B cells to memory B cells resulting from a division is a parameter of the simulation, typically set to 0.5. B cells require the help of T helper cells to activate. Finally, when an APC — other than a B cell — presenting an antigen peptide on its class II MHC molecule is bound by a T cell, only the T cell divides and the antigen is removed.

The immune system simulations described above consider only the humoral branch of the immune system (T cell is to be understood as T helper cell) and the antigen is not considered infective. Infectivity had to be included in order to represent the cell-mediated branch since this branch's major contribution is through the actions of cytotoxic T cells destroying infected cells. In [10], infectivity was included by allowing antigens to infect APCs such that no target cells had to be added. Infection is described by three parameters: the probability of infection when an antigen encounters an APC, the rate of replication of the antigen inside the infected a cell, and the maximum number of antigens that an infected cell can contain before exploding and releasing all the antigens it contains into the simulation as a viral burst. In this early implementation [10], cytotoxic T cells are non-specific cells, i.e. they have no receptor. Instead, they interact with any infected cell with a fixed probability and destroy the cell. In a later paper [12], Celada and Seiden describe a more complex implementation of the cellular and humoral branch which adds epithelial target cells to the simulations, as well as danger signals and cytokines. The simulation also included T cell maturity stages, class I MHC molecules and specific cytotoxic T cells. However, their description of the structures and functions of the new entities is extremely brief at best and one would need access to their computer code to understand how these new "features" have been implemented.

4.1.3 Usage of the Bit-string Models

The bit-string models have been used to investigate various immunological questions among which the process leading to affinity maturation and hypermutation in the humoral branch of the specific immune system [14], the rheumatoid factor paradox [75], the transition process between immune and disease states and the contribution from the humoral only, cellular only, and humoral and cellular branches of the immune system [10, 12], the impact of vaccine efficiency on the humoral branch, the cellular branch and both branches [44], and the dynamics of HIV infection with respect to the selection of escape mutants from immune recognition [8].

Models are typically built for two reasons. They are primarily built as simplifications of the real system. This is done so that the important mechanisms of the more complicated system can be revealed and better understood through the manipulation of the simpler model. Models are also built as complete and detailed representations of the real system. This is done so that hypothesis can be tested on the model rather than on the real system. This is usually desirable when testing an hypothesis on the real system would be too costly, unethical, or otherwise impractical.

The bit-string models, are clearly destined to be of the latter type. The great level of details with which they implement the various aspects of the immune system is impressive and unique. But this author believes their development and use is a little premature. Bit-string models, such as CIMMSIM, contain thousands of parameters. Most of these parameters' value is either not known, or known with very little precision. This makes the complicated bit-string models too unreliable to be used to test scientific hypotheses.

At the moment, there is a need for simpler models of the immune system which permit the investigation of more particular aspects of immune mechanisms. Such models help to further our understanding of particular immunological question, and are particularly useful in extracting the parameter values needed to perfect more complicated models such as CIMMSIM. The next section offers an overview of such models.

4.2 Spatiotemporal Models of Viral Infections

Here, a brief overview of spatiotemporal models is presented. These models were created to investigate the effects of the spatial distribution of cells on the dynamics and evolution of a host-pathogen system, which is the main topic of the research presented herein.

4.2.1 Generic Spatiotemporal Models

The Louzoun et al. Model [48]

In [48], Louzoun et al. set out to determine what type of modelling is most appropriate to explain the emergence of immune complexity out of chemical simplicity. Their investigation consist of identifying how the dynamics of a simple lymphocyte-antigen system varies depending on the type of model chosen to represent the system. To accomplish this, Louzoun et al. introduce a simple ODE system describing the interaction

of a population of lymphocytes, L, and antigens, A, on a discretized grid as

$$\frac{\mathrm{d}L_i}{\mathrm{d}t} = \tau A_i L_i - d_L L_i + \mu_L \nabla^2 L_i \ , \tag{4.3}$$

$$\frac{\mathrm{d}A_i}{\mathrm{d}t} = \lambda - d_A A_i + \mu_A \nabla^2 A_i \ , \tag{4.4}$$

where the i subscript designates the lattice site. λ is the production rate of antigen, τ is the division rate of lymphocytes upon encountering the antigen, d_L and d_A are the death rates of lymphocytes and antigens, respectively, and the last term of each equation represents the diffusion of antigens and lymphocytes to neighbouring lattice sites. Spatially averaging (4.3)–(4.4) yields

$$\frac{\mathrm{d}L}{\mathrm{d}t} = \tau \langle AL \rangle - d_L L \ , \tag{4.5}$$

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \lambda - d_A A \ , \tag{4.6}$$

where L and A are the average over all lattice sites of L_i and A_i respectively, $\langle AL \rangle$ is the average of $A_i \cdot L_i$ over all lattice sites, and the diffusion terms' averages are 0. If L and A are independent variables, i.e. if the concentration of lymphocytes at one point is independent of the number of antigen at that point ($\langle AL \rangle = A \cdot L$), (4.5)–(4.6) simplifies to a spatially-independent ODE with solution

$$\bar{L} = \bar{L}(0)\ \mathrm{e}^{\left(\frac{\tau\lambda}{d_A} - d_L\right)t} \ , \tag{4.7}$$

$$\bar{A} = \lambda / d_A \ . \tag{4.8}$$

This results in exponential growth of the lymphocyte population if $\frac{\tau\lambda}{d_A} > d_L$ or exponential decay if $\frac{\tau\lambda}{d_A} < d_L$.

Using a cellular automaton model of the system described by (4.3)–(4.4), Louzoun et al. find that because each point in their spatial grid contains a different concentration of antigens, the lymphocyte population will undergo exponential growth at sites where $A_i\tau > d_L$ and exponential decay where $A_i\tau < d_L$. Over time, the contributions from sites with decaying lymphocyte population will become negligible and only the points with the highest growth rate will dominate the system average. Thus, the average concentration of cells will grow exponentially, but the growth rate will be proportional to the peaks of the local concentration of antigen, and not to the average concentration throughout the system [48].

With this, Louzoun et al. have shown that if in fact the concentration of antigen and lymphocyte is not independent, the ODE modelling approach is flawed as it will lead to misleading prediction of the amount of antigen in the system or the growth rate of the lymphocyte population [48]. Louzoun et al. explain that L and A are not independent in the real immune system because the lymphocyte concentration is high precisely at the sites where the antigen concentration is high. Thus, models of the antigen-lymphocyte system should take the spatial distribution of these agents into account.

The Funk et al. Model [29]

In [29], Funk et al. investigate the implications on viral infection dynamics of the well-mixed and homogeneous assumptions made by the basic ODE model (1.1)–(1.3). This is done similarly to what is done in [48], i.e. by constructing a spatially extended version of the basic viral infection model, namely

$$\frac{\mathrm{d}T_{i,j}}{\mathrm{d}t} = b - dT_{i,j} - eT_{i,j}V_{i,j} \ , \tag{4.9}$$

$$\frac{\mathrm{d}Y_{i,j}}{\mathrm{d}t} = eT_{i,j}V_{i,j} - uY_{i,j} \ , \tag{4.10}$$

$$\frac{\mathrm{d}V_{i,j}}{\mathrm{d}t} = pY_{i,j} - cV_{i,j} - \frac{m_V}{8}\sum_{i_0=i-1}^{i+1}\sum_{j_0=j-1}^{j+1}\left[V_{i,j} - V_{i_0,j_0}\right] \ . \tag{4.11}$$

$T_{i,j}$, $Y_{i,j}$, and $V_{i,j}$ represent the abundance at site (i, j) of target cell, infected cell, and virions, respectively. Additionally, b is the production rate of target cell, d and u are the death rate of target and infected cells, respectively, e is the infection rate of target cells by virions, and p and c are the production and clearance rate of virions. Note that in this model, the target cells are fixed and only the virions are allowed to diffuse. The term m_V is the diffusion rate of free virions to the 8 adjacent sites. Infection is started by inoculating 10 viral units at the centre of their 21x21 square grid. Funk et al. found that the population equilibrium is the same for both the basic ODE model and its spatial extension. However, they remark that

> "... in line with experimental results, the spatial model indicates a biphasic up-slope of the virus load curve while the non-spatial model lacks to reproduce this. The subdued viral growth rate is due to the fact that the infection settles to its equilibrium earlier at those sites where it started than in the periphery. Averaging out over all sites eventually results in a biphasic up-slope." [29]

Funk et al. conclude that a spatially averaged simulation at the whole-body level serves as a good approximation close to equilibrium, but underestimates the true local infection dynamics in vivo.

Funk et al. go further than in [48] by investigating the effects of heterogeneity of the rules. This is done by allowing each parameter to vary randomly across the grid sites within a certain range from a uniform distribution. This causes the emergence of infection sources and sinks, i.e. sites where the basic reproduction ratio, R_0, is greater or less than 1, respectively. Note that in the context of a viral infection, the basic reproduction ratio, R_0, is defined as the number of secondary infections produced by a single infected cells during its entire period of infectiousness when placed in a population composed entirely of susceptible cells [1, 19]. As a result of the sources and sinks landscape, the viral equilibrium state is very sensitive to the diffusion rate of virions, i.e. parameter m_V in (4.9)–(4.11). This is because in a system of sources and sinks, when sources are outnumbered by sinks, a high diffusion rate will cause the virus to predominantly leave

source sites for sink sites.

Funk et al. also introduce a spatially extended virus-immune system, which adds an immune response in the form of a diffusing immune cell population, $X_{i,j}$, to (4.9)–(4.11), namely

$$\frac{\mathrm{d}T_{i,j}}{\mathrm{d}t} = b - dT_{i,j} - eT_{i,j}V_{i,j} \ , \tag{4.12}$$

$$\frac{\mathrm{d}Y_{i,j}}{\mathrm{d}t} = eT_{i,j}V_{i,j} - uY_{i,j} - kX_{i,j}Y_{i,j} \ , \tag{4.13}$$

$$\frac{\mathrm{d}V_{i,j}}{\mathrm{d}t} = pY_{i,j} - cV_{i,j} - \frac{m_V}{8}\sum_{i_0=i-1}^{i+1}\sum_{j_0=j-1}^{j+1}\left[V_{i,j} - V_{i_0,j_0}\right] \ , \tag{4.14}$$

$$\frac{\mathrm{d}X_{i,j}}{\mathrm{d}t} = aX_{i,j}Y_{i,j} - qX_{i,j} - \frac{m_X}{8}\sum_{i_0=i-1}^{i+1}\sum_{j_0=j-1}^{j+1}\left[X_{i,j} - X_{i_0,j_0}\right] \ . \tag{4.15}$$

The virus-immune system has an additional equilibrium where the pathogen can persist, kept under check by a persisting immune response. In this spatially extended virus-immune system model, under homogeneous rules, the diffusion of virions and immune cells enhances population stability. Equilibrium population size is attained faster in the spatially extended model with fewer oscillations (one initial oscillation instead of > 20) than in the equivalent non-spatial model. Oscillations during the establishment of an infection carry the risk that the pathogen will go extinct. The spatially extended model thus increases the chance that the infection will persist as it reduces the number and amplitude of the viral population oscillations.

The results presented by Funk et al. are interesting because their viral infection and virus-immune models are realistic. Their investigation consisted of explicitly adding a spatial component to the widely used and accepted viral infection models. For this reason, their conclusions are of great relevance to the ongoing modelling efforts of viral infections. Application of their proposed spatial model to specific viral infections would be crucial to validating the way in which space and spatial heterogeneity has been included in these models.

4.2.2 Two-dimensional Cellular Automaton Models of HIV

The Zorzenon dos Santos and Coutinho Model [90]

In [90], Zorzenon dos Santos and Coutinho introduce a simple CA to model the dynamics of HIV. Their motivation is to reproduce the two time scales of an HIV infection: the short time scale (few weeks) associated with the primary response and the long one (few years) associated with the clinical latency period and the onset of AIDS [90]. They suggest that the mean-field ODE models have failed to reproduce the two-scale dynamics of HIV because they do not take into account the local interactions caused by the localization of the initial immune response in the lymphoid organs. The highlight of their spatiotemporal CA model is its ability to reproduce this two-scale dynamics.

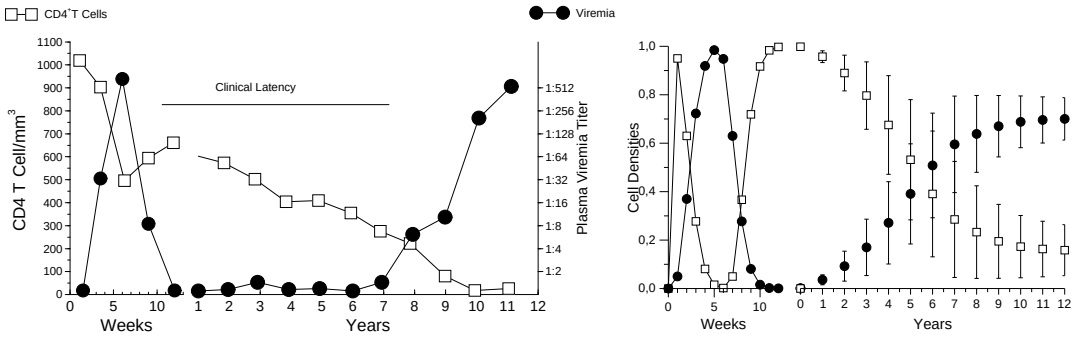


Figure 4.4: Experimental HIV data (left panel) and simulation results from the Zorzenon dos Santos and Coutinho HIV CA model (right panel). In the left panel, the density of $CD4^+$ T cells (open squares) and virus concentration (full circles) are shown. In the right panel, the density of healthy (open squares) and infected (full circles) target cells are shown. Figures reproduced from [90].

Experimental results illustrating the two-scale dynamics of HIV along with results from their CA simulations are presented in Figure 4.4.

In their model, every site in the two-dimensional CA square grid represents a target cell for the HIV, namely a $CD4^+$ T cell or a monocyte. Each target cell can be in any of four states, namely healthy, infected-A1, infected-A2 or dead. A cell infected-A1 corresponds to an infected cell that is free to spread the infection. A cell infected-A2 corresponds to an infected cell in its final stage before apoptosis and can only infect a healthy cell when other infected-A2 cells are present in sufficient concentrations. The rules for the evolution of the CA are the following:

- A healthy cell becomes infected-A1 in the next time step if any of its 8 neighbours are infected-A1 or if at least $2 < R < 8$ of its neighbours are infected-A2.[2]
- An infected-A1 cell becomes infected-A2 after τ time steps.
- An infected-A2 cell becomes a dead cell in the next time step.
- A dead cell is replaced by a healthy cell with probability p_{repl} or otherwise remains dead in the next time step.
- Finally, any new healthy cell can instead be created as an infected-A1 cell with probability p_{infec}, such that the rate at which dead cells are replaced by infected-A1 cells is $p_{\text{newinfec}} = p_{\text{repl}} \times p_{\text{infec}}$.

This last rule is supposed to simulate the introduction of new infected cells in the system, either coming from other compartments of the immune system or resulting from the activation of the latent infected cells.

[2]Unfortunately, the authors give no justification as to why a cell having $R = 8$ A2 infected neighbours would not become infected A1. This is most likely a mistake and should probably have read $2 < R \leq 8$.

In this model, the permanence of the infection is insured by the fact that infected cells are continuously being added at a rate p_{newinfec} per dead cell per time step. These infection seeds lead to the formation of predictable square waves of infection, as seen in Figures 4.5(b,c). In turn, the percolation of these square waves with each other will ultimately results in the formation of a more complex square wave pattern, illustrated in Figure 4.5(d).

This model is the first spatiotemporal model of HIV. Unfortunately, the model does not take into consideration the motility of the target cells. In effect, although the density of cells within lymph nodes is extremely high, this does not prevent the high motility of T cells (see Chapter 7). This is especially true given the fact that the model is used to simulate HIV dynamics on the scale of several weeks, even years. Allowing the cells in this model to move would prevent the formation of the perfect square waves and the emergence of more complicated square pattern. In such conditions, it is unlikely that the model would still be capable of exhibiting the two time scales of the HIV dynamics.

The Strain et al. Model [76]

In [76], Strain et al. propose a more complete spatiotemporal model of HIV which includes the known biophysical properties of HIV. Their motivation is to elucidate whether and how spatial correlations contribute to viral propagation, given that viral propagation is a fundamentally local process.

Strain et al. did a careful calculation of the spatial spread of virus released as a single burst from an infectious cell. From this, they computed the diffusion coefficient for HIV virions, and calculated the probability $P_b(i)$ that a cell i sites away from an infectious cell releasing a burst of b virions will become infected. From this, they computed a basic reproduction ratio (R_0) for HIV which takes into consideration the localized spatial nature of viral bursts. This expression of R_0 predicts that viral propagation will be limited by viral stability at low target cell density, and by geometry (target cell's radius) at high cell density. Additionally, the authors propose a three-dimensional cellular automaton model for HIV viral infections. Their cellular automaton model makes use of the calculations mentioned above for the probability $P_b(i)$ that a cell i sites away from the viral burst of size b released by a dying infected cell will become infected.

Each site of their three-dimensional cubic lattice can be in any of three states, empty (E), infected (I), or target (T) and is updated with the following rules:

- A target cell at site i will become infected with a probability $\sum_{j \neq i} P_b(i-j)$.
- A site containing an infected cell will become empty in the next time step.
- An empty site will contain a target cell in the next time step with probability $\sigma_T + \frac{1}{n}\sum_{nn} \delta_T$, where nn indicates nearest-neighbour lattice sites, and n is the total number of nearest neighbours (6 for the simple cubic lattice used in [76]). The terms δ_T and σ_T are the rate of repopulation of empty lattice site due to the division of immediate neighbours, and influx of cells from peripheral blood or from the thymus, respectively.

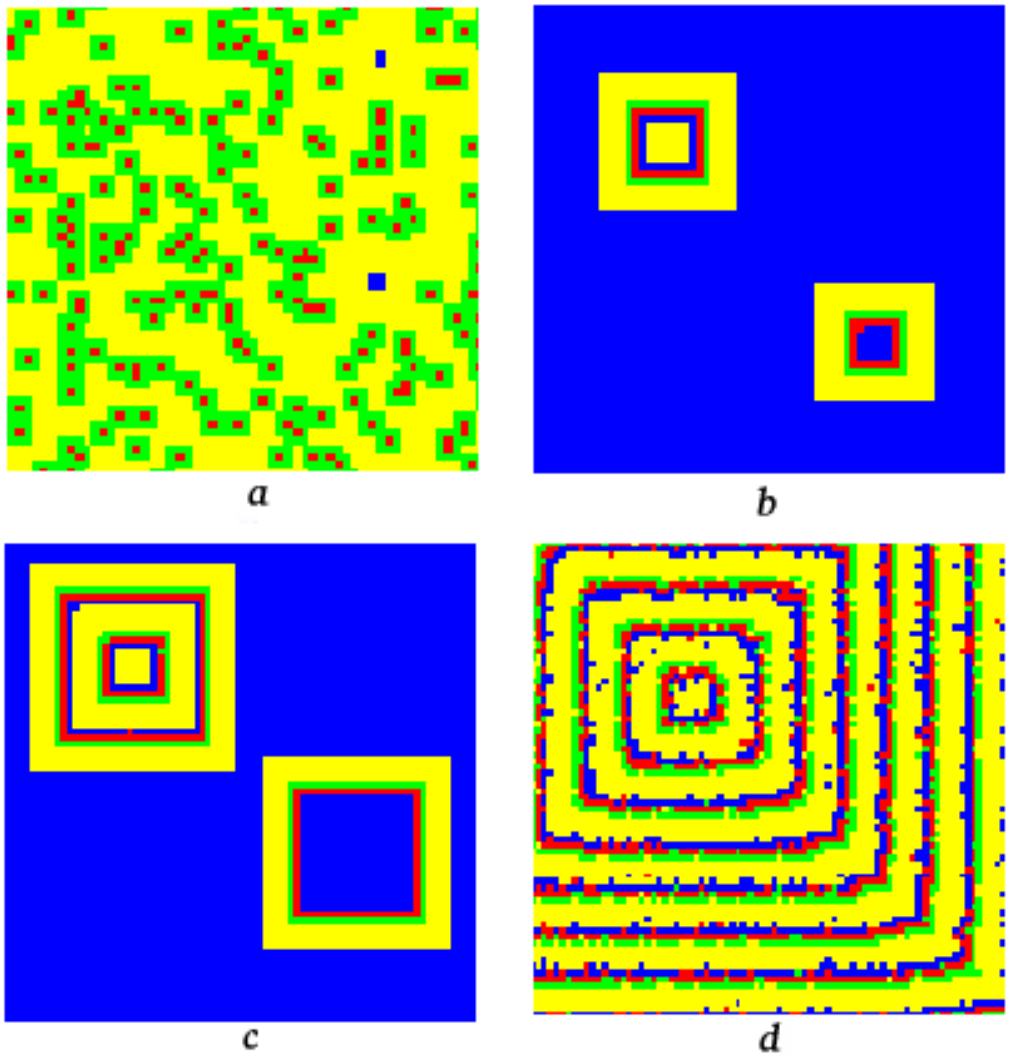


Figure 4.5: Screenshots from the Zorzenon dos Santos and Coutinho CA simulation of HIV infection at (a) 5, (b) 18, (c) 25, and (d) 200 weeks. Colours mark healthy (blue), infected A1 (yellow) infected A2 (green), and dead (red) $CD4^+$ T cells or monocytes, the targets of HIV. Figure reproduced from [90].

Strain et al. found that the spatiotemporal dynamics of their CA model is determined by the propagation efficiency and the recovery rate. If the propagation efficiency is such that the basic reproduction ratio is less than of order 1, the infection does not propagate [76]. The long-term dynamics of their model is determined by the rate of recovery of target cells. Infection, in their model, propagates out as radial wave fronts, leaving a wake of empty cells. If cells recover quickly, virus can diffuse from producer cells in the wave front back across this wake (see the left panels of Figure 4.6). Once the initial pulse has propagated to the edge of their grid, the dynamics of the system settles into a stationary chaotic state in which infected, target, and empty sites coexist [76]. If recovery is slow, the infection propagates transiently as a unidirectional wave (see the right panels of Figure 4.6). In this case, infection can only be sustained if the influx of target cells to random empty sites of the grid is non-zero ($\sigma_T \neq 0$).

Strain et al.'s model is very interesting as it addresses both the contribution of spatial correlations on viral propagation and how this contribution can distort the conclusions arrived at using mean-field approaches. In particular, their finding that the standard ODE model (as proposed by Perelson et al. in [64]) compared to the lattice model overestimates viral infectiousness by more than an order of magnitude, is of particular interest to modellers and experimentalists alike. Unfortunately, they have also assumed the target cells of HIV to be immobile, which is incorrect as will be seen in Chapter 7. It would be interesting to see whether their findings remain valid if one includes the motion of target cells. They also make the additional assumption that virions are released as a single burst upon infected cells' death. It would be interesting to see how a continuous release would affect their result.

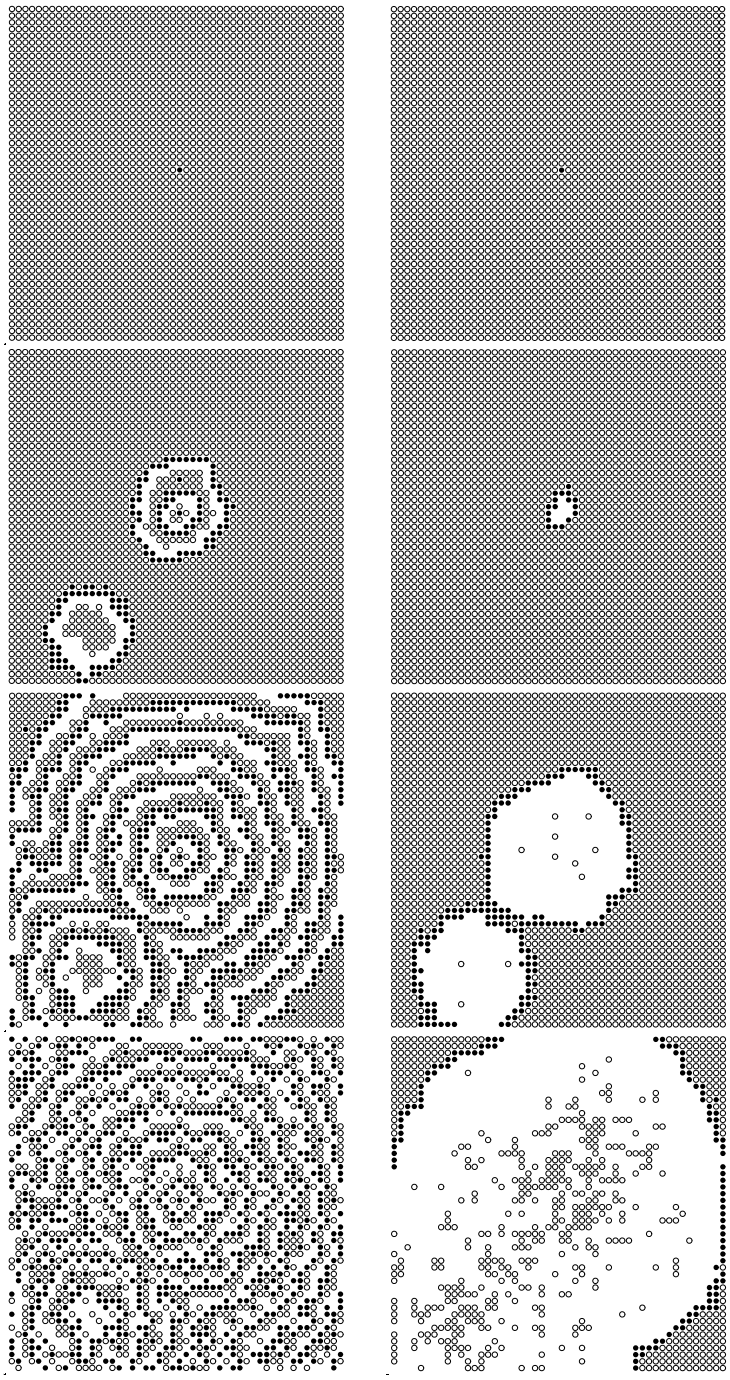


Figure 4.6: Screenshots from the CA simulation of the HIV infection model of Strain et al., for quick (left panels) and slow (right panels) recovery. Healthy cells are denoted by white circles, infected cells by black circles, and white areas denote empty sites. Figures reproduced from [76].

Chapter 5

A Simple Cellular Automaton Model for Influenza A Viral Infections[1]

Here, a cellular automaton model is introduced that will be used later to study the influence of spatial heterogeneities on the dynamical evolution of a viral infection. In this chapter, the CA model is calibrated to a particular viral infection. This is done in order to verify that the model is complex enough to reproduce the general shape of a response to an uncomplicated viral infection, and to see if it gives quantitatively reasonable results when parametrized for a particular viral infection. We chose to calibrate the model to influenza A because we felt that it would be easier to find data for this viral infection. The calibration to influenza A in particular is not crucial as we could have chosen any other uncomplicated viral infection for which data is available. However, the calibration to a particular uncomplicated viral infection, whichever one is chosen, is crucial for assessing whether or not the model is realistic enough to be used as a platform for experimentation with various aspects of the immune system. We have found that it is.

In Section 5.1, we describe the structure and evolution rules of the CA model. In Section 5.2, we present results from the CA simulations and compare them against data from the literature. In Section 5.3, we expand on the biological meaning of the model's parameters, how we have arrived at the values we chose for the model, and examine the model's sensitivity to their values. Finally, in Section 5.4, we take a look at the parameter space around the default parameter values for the parameters to which the model is sensitive.

5.1 The Cellular Automaton Model

Our CA model considers two species: epithelial cells, which are the target of the viral infection, and immune cells, which fight the infection. The virus particles themselves

[1] A version of this chapter has been published in [4].

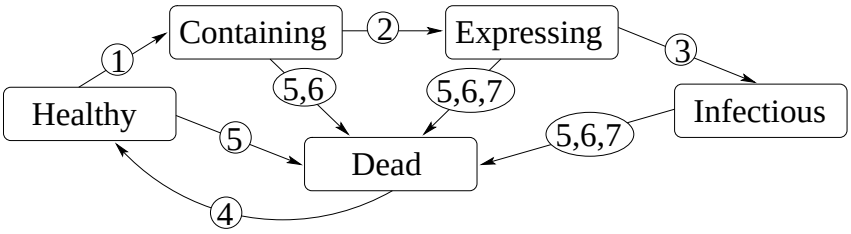
are not explicitly considered, rather the infection is modelled as spreading directly from one epithelial cell to another. The CA is run on a two-dimensional square lattice where each site represents one epithelial cell. Immune cells are mobile, moving from one lattice site to another, and their population size is not constant. The CA lattice is therefore like a tissue of immobile cells which is patrolled by the mobile immune cells. The CA is updated synchronously (i.e. all cells are time-stepped at once rather than one cell at a time). The boundary conditions for both the epithelial and immune cells are toroidal, i.e. an immune cell moving off one edge of the grid are reintroduced at the opposite edge and an infectious epithelial cell at one edge of the grid can infect healthy cells located at the opposite edge. Finally, the neighbourhood of a lattice site is defined as consisting of the 8 closest sites. Infected epithelial cells can only infect their 8 surrounding neighbours and immune cells can move at a speed of 1 site per time step to any of the 8 neighbour sites or remain in place.

An epithelial cell can be in any of five states: healthy, containing, expressing, infectious, or dead. Transitions between epithelial cell states occur as described in Figure 5.1. A simulation is initialized with each epithelial cell being assigned a random age between 0 and δ_H inclusively. All epithelial cells start in the healthy state with the exception of a fraction ρ_C of the total number of epithelial cells which, chosen at random, are set to the containing state.

An immune cell can be in any of two states: unactivated or activated. An unactivated cell is an immune cell that has not yet been activated by an encounter with an expressing or infectious cell. An activated cell is an immune cell that has either already encountered an expressing or infectious cell or has been recruited by another activated immune cell. Transition between the two immune cell states occurs as described in Figure 5.2. A simulation is initialized with a density of ρ_M unactivated immune cells at random locations on the CA lattice, each with a random age between 0 and δ_M inclusively. Additionally, immune cells move randomly on the CA lattice at a speed of one lattice site per time step, and there are $\nu = 6$ time steps/h. B and T cells range from 6 μm in diameter for naive cells who have not yet encountered an antigen, to 15 μm in diameter for effector cells [30], while a lung epithelial cell (lattice site) is typically about 20 μm in diameter [18]. Because B and T lymphocytes are smaller than lung epithelial cells, the CA model does not restrict the number of immune cells which can occupy a given lattice site at any one time step and neglects collisions. All immune cells occupying a given site are allowed to interact with the epithelial cell of that site.

Implicit in this model are the following physiological assumptions:

- Only healthy epithelial cells are able to divide.
- Immune cells cannot be infected and so only ever die of old age.
- No memory immune cells are considered. Memory could be added by setting an extended lifespan to a given fraction of immune cells created during a viral invasion.
- We consider a single viral strain with no mutation. This implies the existence of a single, unique, epitope identifying the infection.



1. Healthy epithelial cells get infected at rate $\beta/8 = 0.25\ \mathrm{h}^{-1}$ per infectious Moore neighbour (8 nearest neighbours), where $\beta = 2\ \mathrm{h}^{-1}$ is the rate of infection of neighbours by infectious cells.

2. A virion-containing cell that has been infected for $\tau_E = 4$ h begins expressing the viral peptide on its epitopes.

3. An expressing cell that has been infected for $\tau_I = 6$ h becomes infectious.

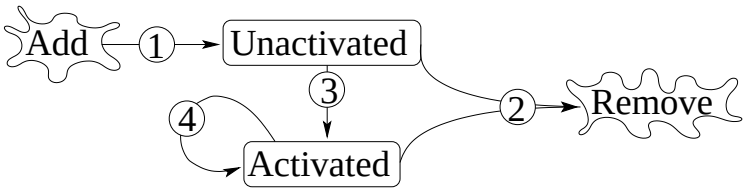
4. A dead cell is replaced by a healthy cell at rate $b^{-1} \times \#\ \text{healthy}/\#\ \text{dead}$, where $b = 12$ h is the division time of an epithelial cell.

5. All cells will die of old age after living for exactly $\delta_H = 380$ h, unless they die earlier because of viral toxicity or immune recognition (see below).

6. Because of viral toxicity, infected cells (i.e. containing + expressing + infectious) will die after having been infected for $\delta_I = 24$ h, unless they die earlier from recognition (see below) or from old age (see above).

7. Finally, expressing and infectious cells die when "recognized" by an activated immune cell.

Figure 5.1: Evolution rules for the epithelial cells in the cellular automaton model.



1. Unactivated immune cells are added at random lattice sites as needed to maintain a minimum density of $\rho_M = 1.5 \times 10^{-4}$ unactivated immune cells per epithelial cell.

2. All immune cells die of old age after living for exactly $\delta_M = 168$ h.

3. An unactivated immune cell becomes activated when it first occupies an expressing or infectious lattice site.

4. If an activated cell is occupying an expressing or infectious lattice site, it kills the epithelial cell and $r_M = 0.25$ new activated immune cells are added at random locations on the grid. The integer part of r_M (if $r_M > 1$) is used to determine the number of immune cells that will be added and the fractional part is taken to be the probability of adding an additional immune cell.

Figure 5.2: Evolution rules for the immune cells in the cellular automaton model.

- Only immune cells with an affinity which exactly matches that of the single viral strain are considered. This implies that the model does not take into consideration cross-reactivity from immune cells capable of responding to the infection with a strength dependant on their affinity to the strain.

5.2 Results

A great deal is known about the influenza virus including viral structure and composition, the replication process, and even some dynamical data regarding the viral and antibody titer over the course of the infection [42, 28, 5]. However, key dynamical information such as the flow rate of immune cells within an infected tertiary lymphoid organ, the clearance rate of viral particles, the lifespan of an infected epithelial cell, is either uncorroborated, unknown, or known with poor precision. In the model presented here, all but two parameters have been taken directly or adapted from [11]. They are presented in Table 5.1 along with their value and description.

Due to the limited dynamical information available for influenza A, there are only a few quantitative characteristics against which the model can be compared. They are:

1. The infection should peak on day 2 [11, 28].
2. Over the course of the infection, the fraction of epithelial cells that are dead should be as follows [11]:
   (a) 10% on day 1;
   (b) 40% on day 2;
   (c) 10% on day 5.
3. From [11], virus concentration should decline to inoculation level on day 6. From [28], experimental data recovered from 8 volunteers indicated that virus shedding persisted for 5 ± 2 d.
4. The number of immune cells should peak anywhere between day 2 (macrophages' peak) and day 7 (cytotoxic T cells' and B cells' peak) [11].
5. At their peak, the number of B cells, helper T cells, and cytotoxic T cells should be 100-fold greater than their normal concentration, while that of plasma cells should be 10^4-fold greater [11].

The simulation results for the default parameters presented in Table 5.1 are shown in Figure 5.3. The infection peaks on day 2, and although there are fewer than 10% of the cells dead on days 1 and 5, there is 40% of the cells dead on day 2. Recovery from the infection is faster than experimental data suggests, but this is probably mainly due to the fact that the immune cells in this model are generic. During a true influenza A infection, the first part of the immune response is dominated by macrophages and T cells while the later part is dominated by the action of antibodies, a subtlety that is missed

Parameter	Value	Description	Biol. Range	Reference
g_w	440	Width of the grid (number of epithelial cells)	N/A	N/A
g_h	280	Height of the grid (number of epithelial cells)	N/A	N/A
ν	6 ts/h	Speed of immune cells (time steps/hour)	2–20 ts/h	[11]
δ_M	168 h	Lifespan of an immune cell	48–480 h	[11]
δ_H	380 h	Lifespan of a healthy epithelial cell	160–600 h	[67]
δ_I	24 h	Lifespan of an infected epithelial cell	?	[11]
ρ_C	0.01	Fraction of initially infected cells	0.001–0.1	[11]
β	2 h^{-1}	Rate of infection of neighbours	N/A	N/A
τ_E	4 h	Delay from containing to expressing viral peptide	?	[11]
τ_I	6 h	Delay from containing to infectious	?	[11]
b	12 h	Duration of an epithelial cell division	7–24 h	[11]
ρ_M	1.5×10^{-4}	Base density of unactivated immune cells per epithelial cell	?	[11, 66, 82]
r_M	0.25	Number of immune cells recruited upon positive recognition	N/A	N/A

Table 5.1: The model's default parameters along with their biological range and reference.

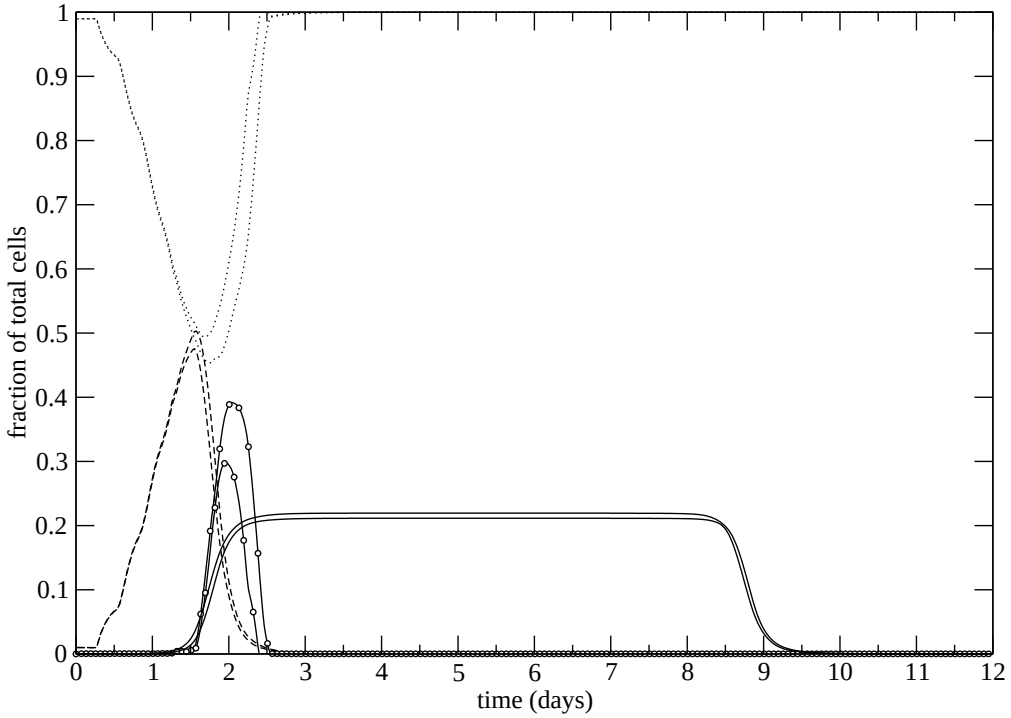


Figure 5.3: Simulation results averaged over 50 simulation runs using the parameter set presented in Table 5.1. The paired lines mark one standard deviation and represent the fraction of epithelial cells that are healthy (dotted line), infected (containing+expressing+infectious) (dashed line), dead (full line with circles), as well as the proportion of immune cells per epithelial cell (full line).

by the model presented here. The number of immune cell does peak between day 2 and 7, with an approximate $1,000$-fold increase from the normal concentration set by ρ_M.

The comparison with the experimental data available for influenza A reveals that the proposed model does in fact have the proper dynamical response when calibrated with biological values from the literature. As mentioned earlier, most of the model's parameter values were taken from [11], where an ODE model for influenza A is presented. The Bocharov et al. model is more complex than the CA model proposed here as it considers a much wider variety of immune cells such as stimulated macrophages, two types of activated T helper cells, activated CTL cells, B cells, plasma cells, and antibodies. However, the Bocharov et al. model, unlike this CA model, does not explicitly have containing, expressing and infectious cells. The immune cell curves of Bocharov et al., like those of this model, agree relatively well with the expected 100-fold increase in the cell population. Bocharov et al.'s dead cell curve is in better agreement than this model's on the first day (10% of cells are dead), but it is a bit low (30%) on day 2 and too high (28%) on day 5. Perhaps one of the biggest difference between the two models is in the shape of the containing+expressing+infectious cells curve (referred to as the infected cells, C_V, in [11]). While at the peak of the infection the model presented here shows approximately 50% of containing+expressing+infectious cells, the Bocharov et al. model has about 70% of infected cells (C_V). Additionally, in the Bocharov et al. model, the fraction of infected cells remains at 70% until day 5 before declining rapidly such that it has dropped to about 10% on day 6.

Overall, the shape of dynamical response given by the Bocharov et al. model does seem to be more accurate than that of the model proposed here, but this is most likely due to the fact that their model considers a wider variety of immune cells. It should be noted that the authors of [11] have also done a fit of their parameters to get a better agreement with experimental data. However, this fit resulted in some of their fitted parameter values lying outside their biological range. For example, the duration of a single division of an epithelial cell during the recovery phase of the disease is about $0.3\ \mathrm{d} \sim 1\ \mathrm{d}$, yielding a regeneration rate of the epithelium of about $1\ \mathrm{d}^{-1} \sim 3\ \mathrm{d}^{-1}$. As a result of the fit, this parameter was assigned a value of $4\ \mathrm{d}^{-1}$. Other parameters, such as the rate constant of epithelial cells' infection by virions, have been assigned values a few orders of magnitude outside their biological range as a result of the fit. This discredits the quality of agreement of the Bocharov et al. model with experimental data.

5.3 Discussion of the Parameters of the Model

Some of the model's parameters have quite a large uncertainty. In some cases this is a reflection of the difficulty of measuring the parameter while in other cases it is a reflection of the variation between individuals. The model also has two free parameters, the recruitment rate of immune cells, r_M, and the rate of infection of neighbours by infectious epithelial cells, β, whose values are not known from physiological data. Here, we will discuss each parameter in detail and investigate the sensitivity of the model to their variation.

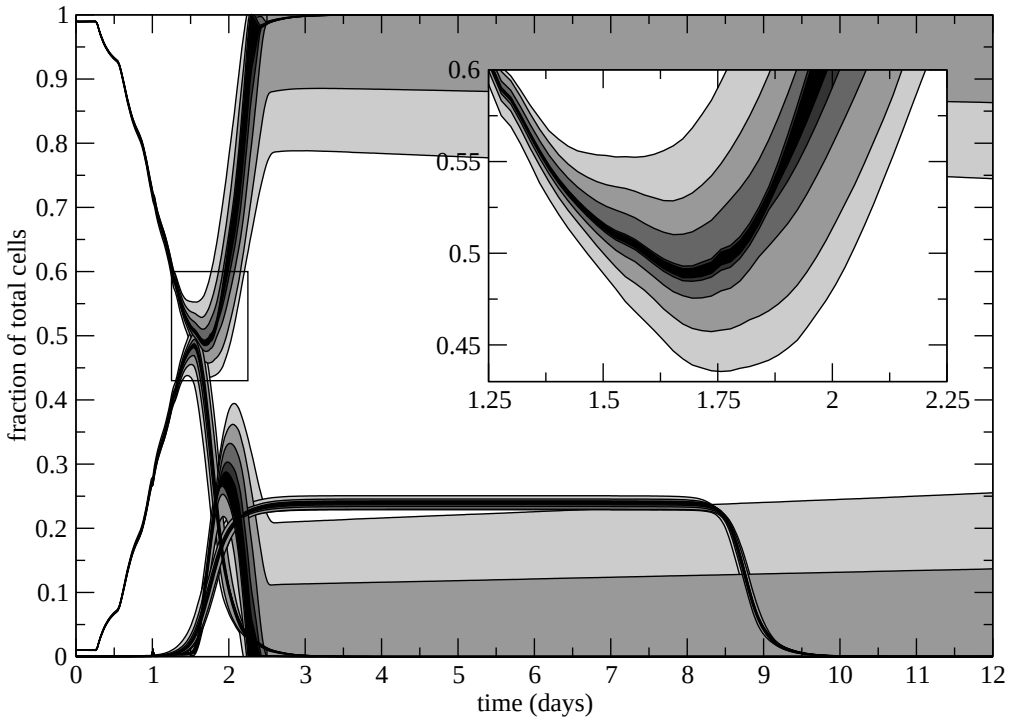


Figure 5.4: Effects of the lattice size on the dynamics of the infection. The greyed areas represent one standard deviation after 10 runs while increasing darkness represents increasing grid sizes of 7,700 cells, 30,800 cells, 123,200 cells, 1,971,200 cells, and 9,432,500 cells. For clarity, the inset shows a detail of the healthy cell curves alone. Increasing the grid size is seen to not affect the average behaviour but decreases the standard deviation.

5.3.1 The Grid's Width and Height, g_w and g_h

Since each lattice site represents an epithelial cell and the width and height (g_w and g_h) of the grid are given in numbers of epithelial cells, it is possible to calculate the physical size of the simulation. The model's default lattice has $g_w \times g_h = 123,200$ cell sites. In uninflamed tissue, there are about $2,200$ ciliated cells per square millimetre of epithelial area [11]. Taking ciliated cells — which account for $60\% \sim 80\%$ of internal bronchial surfaces — to correspond to each lattice site of the CA model, the real-life size of the default lattice is about $123,200 \text{ cells}/2,200 \text{ cells} \cdot \text{mm}^{-2} = 56 \text{ mm}^2$. When inflamed, there are about 700 cells per square millimetre of epithelial tissue area, giving the simulation a real-life area of 176 mm^2.

To test the sensitivity of the model's algorithm to changes in lattice size, i.e. to determine if the finite nature of the simulation influences the results, simulations have been run with lattices ranging from $7,700$ cells to $9,432,500$ cells (the most that could be simulated with the 512 MiB of RAM available in the simulation machine). The results obtained for several lattice sizes in this range are presented in Figure 5.4. What one sees

from this plot is that the average behaviour of the model is unchanged by modifications to the size of the simulation area, but that the standard deviation of the behaviour about the mean decreases as the simulation area is increased. This is consistent with the interpretation of the results from a single simulation run on a large grid as being equivalent to averaged results from several simulations run on smaller grids. This would be the case — that a large simulation is equivalent to an average over several smaller simulations — anytime the simulation size itself does not play a role in the dynamics, and so observing this behaviour strongly suggests that the default lattice size is sufficiently large for its finite nature to be unimportant.

5.3.2 Speed of Immune Cells, ν

Due to the discrete nature of a cellular automaton, immune cells move by increments of one lattice site per time step. Because each lattice site represents one epithelial cell, each time step has to represent the true time required for an immune cell to go from one epithelial cell to the next. Very little is known about the speed at which immune cells patrol tertiary lymphoid organs or by how much the flow will vary depending on whether the area is healthy or infected. This has made the model's time scale difficult to establish.

All time dependent parameters in the model are scaled by the parameter ν which is defined as the number of CA time steps corresponding to 1 h in real time. The duration of the CTL lethal hit process, i.e. the time it takes for a cytotoxic T lymphocyte to kill an infected cell, is about $t_{\text{attack}} = 20$ min ~ 30 min [11]. One can suppose that an immune cell that is not involved with, or in the process of destroying, an infected cell will move from one epithelial cell to the next in less time, say t_{free}. Since on average about 15% of cells are infected over the course of the infection for the default set of parameters chosen for the model (see Figure 5.3), one can compute the average time it takes for an immune cell to go from one epithelial cell to the next as $t_{\text{avg}} = 0.15 \times t_{\text{attack}} + 0.85 \times t_{\text{free}}$. Assuming that t_{free} can be anywhere between negligible, ≈ 0 min, to at most 30 min, then t_{avg} can range from 3 min to 30 min, which corresponds to a range of 2 to 20 time steps per hour for ν. We have chosen to set ν to 6 time steps/h, which means that immune cells move at a speed of one epithelial cell per 10 min. Note that this corresponds to a t_{free} of about 6 min $\sim$ 8 min, but we have found no experimental value in the literature to compare this to.

Figure 5.5 shows the effect of varying ν throughout this range. The faster the immune cells move (the larger ν is), the bigger the advantage they will have over the growth of the infection and thus the less pronounced the infection. This is clearly illustrated in Figure 5.5 where the smallest value of ν results in a lethal infection while the largest value results in a minor infection. The CA model is extremely sensitive to this particular parameter and therefore it would be important for future work to determine this parameter with greater precision.

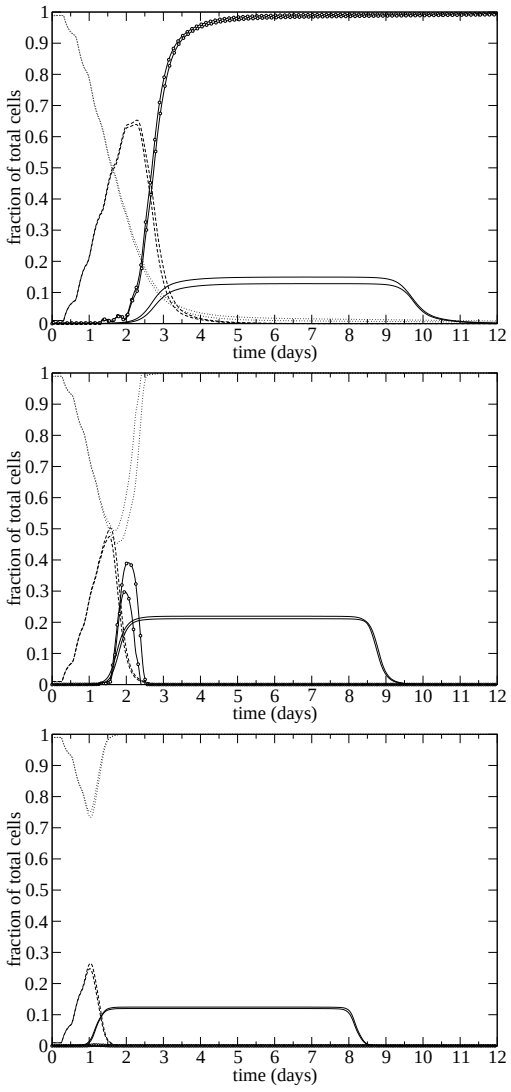


Figure 5.5: The effect of varying the speed of immune cells, ν, on the viral infection's dynamics. From top to bottom, the graphs show the behaviour obtained using ν values of 2, 6, and 20 time steps/h respectively (the physiologically plausible range). The central graph is the same as Figure 5.3. These show that the simulation is sensitive to the parameter ν.

5.3.3 Lifespan of Immune Cells, δ_M

The problem in defining this particular parameter with much precision comes from the fact that the immune cells in the CA model presented here are generic, i.e. they do not represent a particular species of immune cell such as cytotoxic T cells or plasma B cells, rather they represent the combined action of all involved species of immune cells. In [11], the rate constants of natural death for immune cells range from 0.05 d^{-1} for B cells to 0.5 d^{-1} for cytotoxic T cells and plasma B cells. This corresponds to an immune lifespan range of 48 h $\sim$ 480 h. An immune cell lifespan of 168 h was chosen as the default value. In Figure 5.6, the dynamics resulting from the use of a 48 h and 480 h immune cell lifespan is shown. In broadest terms there are two outcomes and which one is observed depends on whether the immune cell lifespan is greater than or less than the time required for the immune cells to completely eliminate the infection. When the immune cell lifespan is less than the time required to clear the infection, all cell populations oscillate, much in the manner of a predator-prey system — a situation which may be referred to as a chronic infection. When the immune cell lifespan is greater than the time required to clear the infection, it does just that.

From Figure 5.6, it can be seen that only at the very bottom end of the physiologically plausible range for the parameter δ_M does its value begin to influence the model's dynamics. Over most of this parameter's range, the infection is killed too early for the immune cell lifespan to be important. Since experimental data indicates that the virion concentration should have declined to inoculation level no later than 144 h (6 d) into the infection, we chose a value for the immune lifespan that is slightly larger than this, namely 168 h (7 d).

5.3.4 Division Time of Epithelial Cells, b

The duration of a single division of an epithelial cell is about 0.3 d $\sim$ 1 d [11]. We chose a value of 12 h (0.5 d) for the model parameter b. Assuming that only healthy cells can undergo successful division, the probability per unit time that any dead cell is revived is given by

$$P(\text{dead} \rightarrow \text{alive}) = \frac{1}{b} \times \frac{\#\text{ healthy cells}}{\#\text{ dead cells}}. \tag{5.1}$$

Figure 5.7 shows the effect of varying the division time of epithelial cells, b, over its physiologically allowed range of 7 h to 24 h. One sees that this parameter can have a significant effect on the dynamics and even on the outcome of the infection. b represents the capacity of the system to regenerate itself: the greater the regenerative power of the system — the smaller b — the better its chances of recuperating after a severe infection. In this model, at the high end of b's range there is a tendency for the influenza A infection to kill the host, while at the low end of its range the epithelial cell regeneration rate is too high to reproduce the level of damage observed experimentally. A default value of 12 h was chosen for the model. With this value, 40% of the epithelial cells are dead at the peak of the infection which is in agreement with the experimental data.

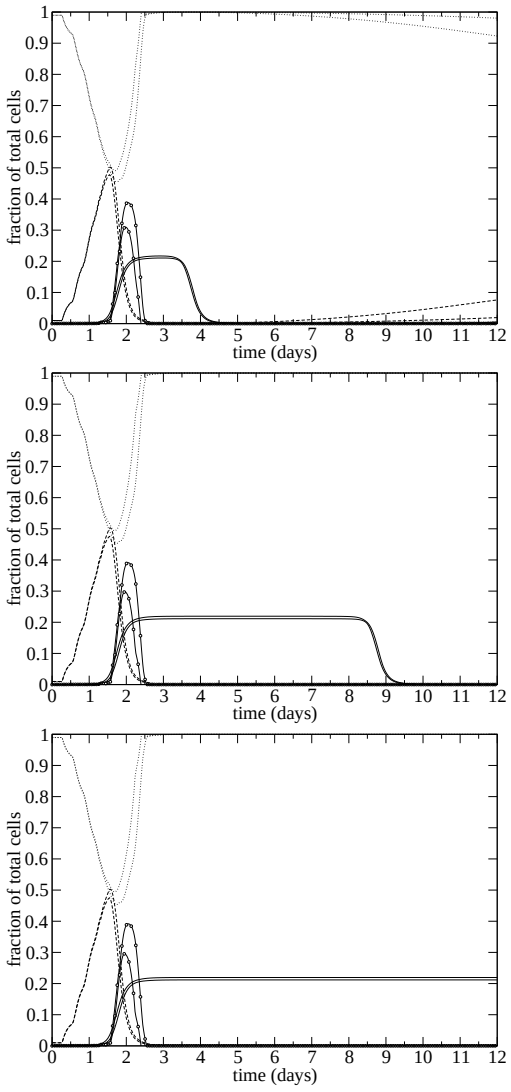


Figure 5.6: The effect of varying the lifespan of immune cells, δ_M, on the viral infection's dynamics. From top to bottom, the graphs show the behaviour obtained using δ_M values of 48 h, 168 h, and 480 h respectively. The central graph is the same as Figure 5.3. These show that the simulation is only sensitive to the parameter δ_M when its value is smaller than the time required for the immune cells to clear the infection.

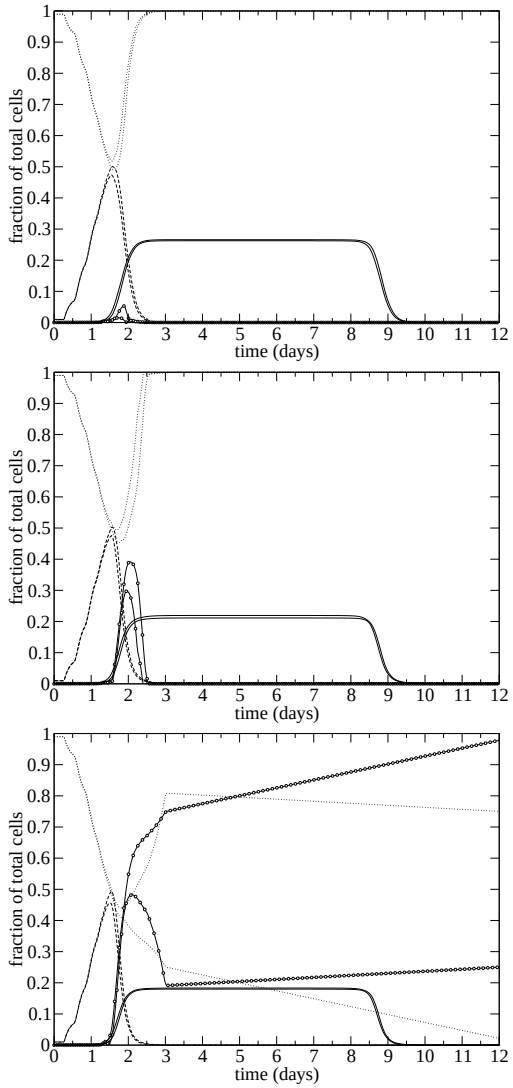


Figure 5.7: The effect of varying the division time of epithelial cells, b, on the viral infection's dynamics. From top to bottom, the graphs show the behaviour obtained using b values of 7 h, 12 h, and 24 h respectively. The central graph is the same as Figure 5.3. These show that the simulation is sensitive to the parameter b.

5.3.5 Lifespan of Epithelial Cells, δ_H

An epithelial cell, in vitro, can go through 20 to 25 population doublings in its lifespan [67]. Each population doubling lasts for b, so the lifespan of an epithelial cell should be about 160 h $\sim$ 600 h. The mid-range value of 380 h was chosen as the default. One impediment to finding more precise information about epithelial cell lifespans is that most in vitro research is done using cell lines that have been made immortal. Additionally, there is much variability in cell lifespan across cell types. In any case, even the lower bound of 160 h is longer than the duration of an uncomplicated influenza A infection and thus the value of δ_H, within the physiologically plausible range, should have little effect on the results. Figure 5.8 shows the dynamics of the viral infection for cell lifespans, δ_H, at the physiologically allowed extrema of 160 h and 600 h as well as at the default mid-range value of 380 h. In the model, one consequence of the en masse killing of epithelial cells by the viral infection is the synchronization of the ages of newly-grown epithelial cells. This is seen as a cell-death "echo" approximately δ_H after the peak of the infection. The only visible difference between simulations run with different values of δ_H is a change in the time at which the cell-death echo occurs.

5.3.6 Duration of the Various Infection Stages (τ_E, τ_I), and Lifespan of Infected Epithelial Cells (δ_I)

Once an epithelial cell has absorbed influenza A virions (e.g. enters the containing state), $\tau_E = 4$ h elapse before it can start expressing viral antigen and thus be recognized by an immune cell [11]. It will take the infected cell an additional 2 h ($\tau_I = 6$ h) to begin releasing virus particles and thus become infectious [11]. Ultimately, influenza A virus being highly cytopathic, the lifespan of an infected epithelial cell is shortened to at most 1 day ($\delta_I = 24$ h) [11].

5.3.7 Proportion of Initially Infected Cells, ρ_C

This parameter represents the magnitude of the initial dose of influenza A virus delivered to test volunteers in viral replication experiments. In particular, in this model, it is the fraction of epithelial cells initially set in the containing state. This parameter's value is not bounded physiologically, rather it is an arbitrary initial condition. In studies of the influenza A virus reported in [11], the typical initial concentration of aerosol-delivered virions is about 10^8 virions/mL. Since the average concentration of epithelial cells in the area typically affected by influenza varies from 10^9 cells/mL to 10^{10} cells/mL, and a single epithelial cell can absorb from 1 to 10 influenza virions, then the typical fraction of cells initially infected is computed to be

$$\rho_C = \frac{10^8 \text{ virions/mL}}{(10^9 \sim 10^{10}) \text{ cells/mL} \times (1 \sim 10) \text{ virions/cell}} \tag{5.2}$$

$$= 0.001 \sim 0.1. \tag{5.3}$$

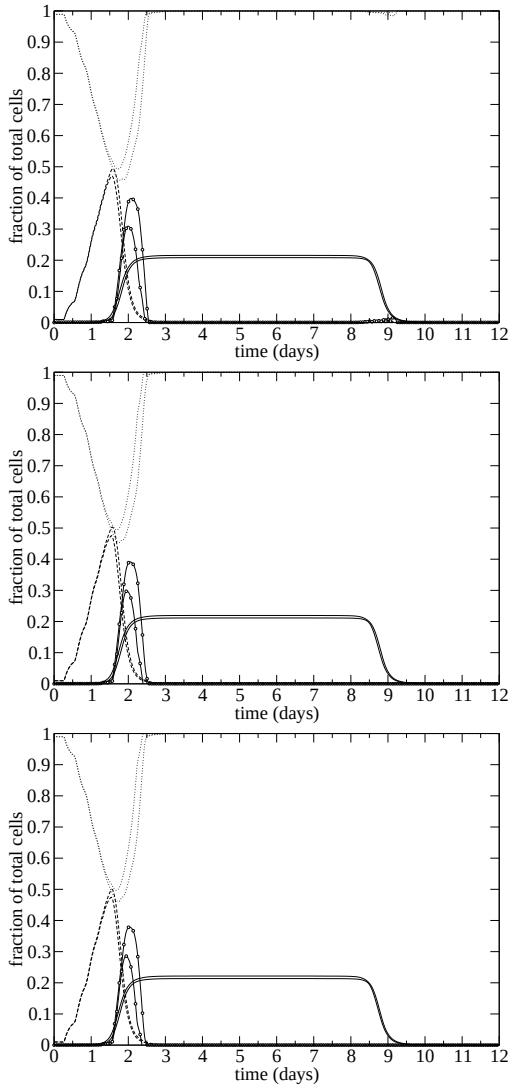


Figure 5.8: The effect of varying the lifespan of epithelial cells, δ_H, on the viral infection's dynamics. From top to bottom, the graphs show the behaviour obtained using δ_H values of 160 h, 380 h, and 600 h respectively. The central graph is the same as Figure 5.3. These show that the simulation is not sensitive to the parameter δ_H.

We have chosen a default value of 0.01 — the logarithmic middle of the range. Figure 5.9 presents the effect on the model's dynamics of varying ρ_C over this range. One sees that the evolution and the outcome of the infection is dependent upon the initial infective dose received. This reflects the competitive nature of the disease and immune system interaction: if one starts with too large a numerical advantage over the other, the latter is bound to loose.

5.3.8 Rate of Infection of Epithelial Cells, β

The growth rate of the infection in the model is determined by the parameter β, which is defined as the number of its neighbours that an infectious cell will infect per hour. This parameter was not found in the literature as it does not correspond directly to the true biological process of infection. In fact, 6 h after a cell has been infected by the influenza A virus, it begins to release influenza A virions (virus particles) into its surrounding and it is those virions that move on to infect neighbouring cells. Instead, the model proposed here does not consider virions at all and rather models the infection process by the direct infection of healthy neighbouring cells by infectious cells. This is computationally simpler and more efficient. Throughout influenza A literature, however, the infection growth is described in terms of the viral titer over the course of the infection rather than as a growth rate. This makes it difficult to establish the parameter β from experimental data.

In [76], Strain et al. suggest a physiologically realistic way of including the virions in a implicit manner for the case of HIV. Although their results may be useful for certain diseases, several factors prevent us from using them in the case of influenza A. We could probably follow a method similar to theirs to find a result that would be suited to the current system. However, since the goal of the research is not to develop an accurate model of influenza A, this is not a crucial issue. Thus, the rate of infection, β, was left as a free parameter and was found to yield best results when set to 2.0 h^{-1}. This means that an infectious cell will, on average, infect 2 of its 8 nearest neighbours per hour. One expects that a smaller value of this parameter will yield a less pronounced infection while larger values will increase the amplitude of the infection and potentially result in the death of the host. This is illustrated in Figure 5.10 where we have explored the effect of using a value $1/2\times$ (1 infected neighbour $\cdot\,\mathrm{h}^{-1}$) and a value $2\times$ (4 infected neighbours $\cdot\,\mathrm{h}^{-1}$) the default.

5.3.9 Base Density of Immune Cells, ρ_M

The viral infection model only concerns itself with immune cells that have a role to play in the viral infection, i.e. those immune cells with a receptor capable of recognizing an epitope of the viral strain present in the simulation. There must always remain a minimum density of cells capable of recognizing and responding to a new outbreak, and this density is set by the parameter ρ_M. The value for ρ_M was calculated based on the following information. It has been determined [82] that intraepithelial lymphocytes have a frequency of about 15 per 100 epithelial cells. Further, the probability that an

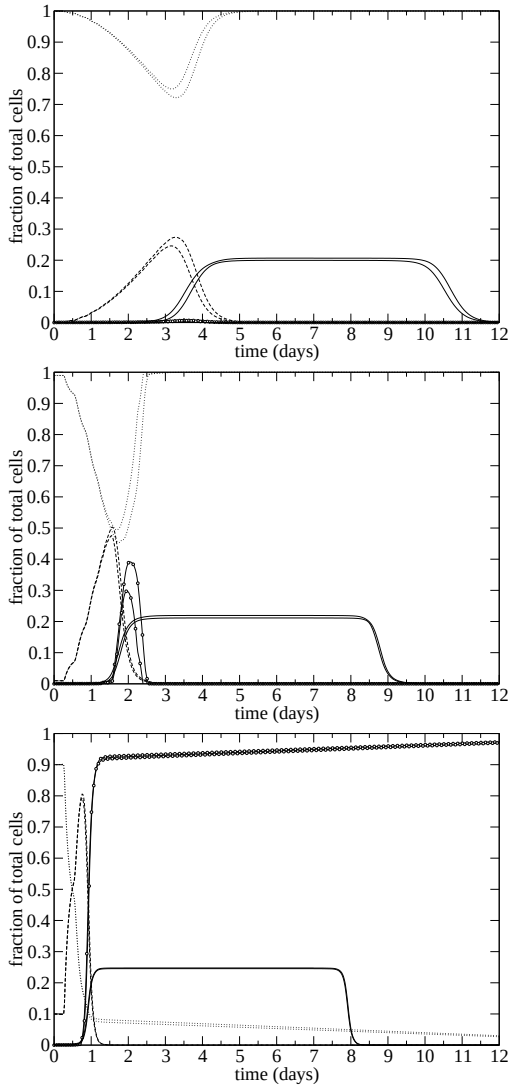


Figure 5.9: The effect of varying the fraction of initially infected epithelial cells, ρ_C, on the viral infection's dynamics. From top to bottom, the graphs show the behaviour obtained using ρ_C values of 0.001, 0.01, and 0.1 respectively. The central graph is the same as Figure 5.3. These show that the simulation is sensitive to the parameter ρ_C.

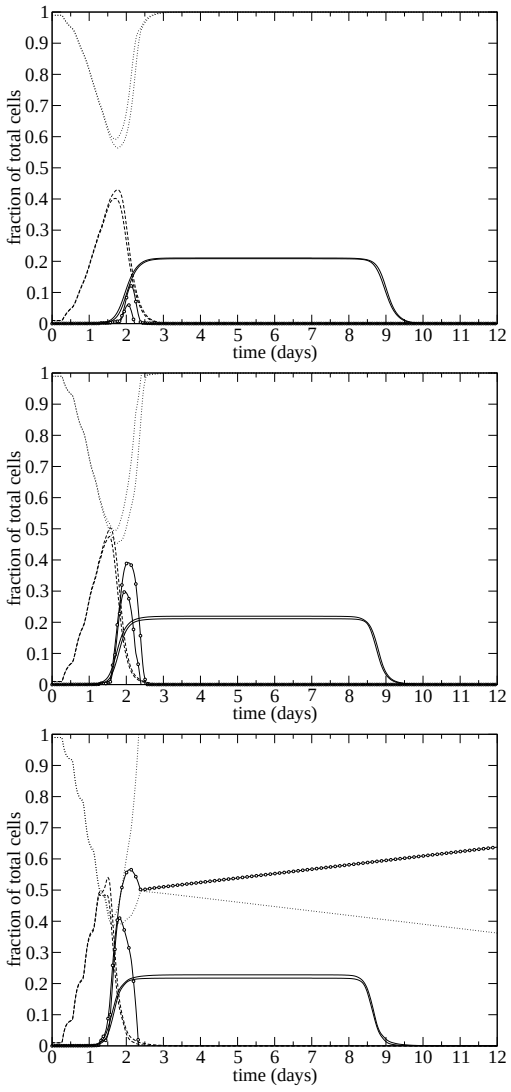


Figure 5.10: The effect of varying the rate of infection of neighbours by infectious epithelial cells, β, on the viral infection's dynamics. From top to bottom, the graphs show the behaviour obtained using β values of 1, 2, and 4 infected neighbour(s) $\cdot\, \mathrm{h}^{-1}$ respectively. The central graph is the same as Figure 5.3. These show that the simulation is sensitive to the parameter β.

immune cell chosen at random will recognize an epitope chosen at random is typically 10^{-5} [66]. This would give a value of $(15/100) \times 10^{-5} = 1.5 \times 10^{-6}$. However, in the case of influenza A, which is commonly contracted, one has to take into account the effect of cross-reactivity and prior immune memory. That is, since an individual is likely to have already been exposed to a strain of influenza A, he or she will have more immune cells ready to respond to the infection than one expects for a less common infection like polio. It is suggested in [11] that the fraction of B or T cells capable of responding to a viral infection with influenza A be increased by 100-fold to account for prior cross-reactivity. Thus, ρ_M is taken to be $1.5 \times 10^{-6} \times 100 = 1.5 \times 10^{-4}$ unactivated immune cells per epithelial cell. Note, however, that this 100-fold increase in the concentration of cells capable of recognizing the antigen accounts for the effect of prior immunity in a purely static manner. At this point, the model does not take into account the dynamical effect of cross-reactivity from immune cells of previous infections who have a small enough antigenic distance from the current strain to take part in the current immune response.

5.3.10 Recruitment Rate of Immune Cells, r_M

When an activated immune cell encounters an infected epithelial cell presenting a viral peptide (expressing or infectious), the activated immune cell will "recruit" a number of new activated immune cells. The number of newly created activated immune cells is set through the parameter r_M. This parameter is one of the two free parameters of the model, and it has no direct biological basis because the immune cells in the model are generic. This parameter represents the efficiency with which the immune system responds to the viral infection. We explored a range of values for r_M, and found that a value of 0.25 recruited immune cells per recognition event, i.e. a 25% chance of recruiting an activated immune cell upon each successful recognition, gives satisfying results.

The typical immune cell division time is about 12 h $\sim$ 24 h [11], which corresponds to a recruitment ranging from 0.08 to 0.16 immune cells per hour. Considering that an immune cell encounters 6 epithelial cells per hour ($\nu = 6$ time steps per hour) and that on average about 15% of those cells are infected, then the model's effective recruitment rate is about $0.25 \frac{\text{immune cell}}{\text{recognition}} \times 0.15 \frac{\text{recognition}}{\text{encounter}} \times 6 \frac{\text{encounter}}{\text{h}} = 0.23$ immune cells recruited per hour, which is higher than the biological range described above. However, that recruitment range is based solely on the cellular divisions of activated B and T cells and does not take into consideration the high production of antibodies by plasma B cells which contributes to reducing the infection, a factor that can account for the higher recruitment rate value required by the model. Simulation runs using an immune cell recruitment rate, r_M, of 0.05 and 1.25 recruited immune cells per recognition event are presented in Figure 5.11. One can see that the parameter r_M has an important effect on the development and the outcome of the infection: too small a value makes the infection fatal, while larger values decrease the amplitude and duration of the infection. It is thus a key parameter and much effort should be invested in finding a way to draw a parallel between it and true biological parameters.

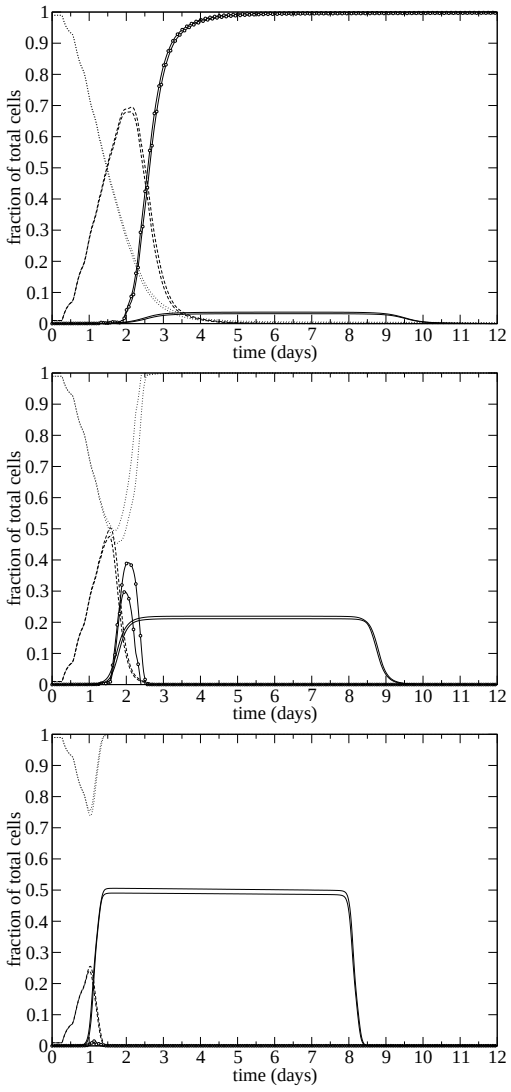


Figure 5.11: The effect of varying the recruitment rate of immune cells, r_M, on the viral infection's dynamics. From top to bottom, the graphs show the behaviour obtained using r_M values of 0.05, 0.25, and 1.25 immune cells per recognition event, respectively. The central graph is the same as Figure 5.3. These show that the simulation is sensitive to the parameter r_M.

5.4 Exploring the Parameter Space

Let us now better characterize the sensitivity of the model to the parameters. This is done for the parameters to which the model was found to be sensitive, namely the speed (ν) and recruitment rate (r_M) of the immune cells, the fraction of initially infected epithelial cells (ρ_C), the infection rate of healthy epithelial cells by infectious ones (β), and the division time of epithelial cells (b).

Within the parameter space considered here, the model always yields one of three outcomes. These outcomes will be referred to as death, chronic infection and remission. The first, death, corresponds to the case where all cells on the grid have died either as a result of the infection or because the damage inflicted on the system by the infection was too great to allow for regeneration. The second, chronic infection, corresponds to the case where infected cells remain at 120 days post infection. The third outcome, remission, corresponds to the case where the infection was successfully cleared and healthy cells repopulated the grid. For a given set of parameter values, not all simulations necessarily result in the same outcome. For simplicity, the parameter space is characterized in terms of the fraction of simulations ending in remission.

Firstly, each parameter to which the model was found to be sensitive was varied over its biological range, keeping all others fixed. Doing so, only 2 of the possible outcomes were observed, namely death and remission. For each set of parameter values, 50 simulations were run to determine the fraction of simulations ending in remission. The results are presented in Figure 5.12 for each of the 5 parameters. Firstly, for the division time of epithelial cells, b, the default parameter value is 12 h, and at 16 h (a 33% increase) we begin to see some simulations ending in death. For the speed of immune cells, $\nu = 6$ ts/h, we begin to see some simulations ending in death at $\nu = 5$ ts/h (a 16% reduction). For the infection rate of immediate neighbours, $\beta = 2\ \mathrm{h}^{-1}$, we begin to see some simulations ending in death at $\beta = 3.3\ \mathrm{h}^{-1}$ (a 65% increase). For the fraction of initially infected cells, $\rho_C = 0.01$, we begin to see some simulations ending in death at $\rho_C = 0.015$ (a 50% increase). Finally, for the recruitment rate of immune cells, $r_M = 0.25$, we begin to see some simulations ending in death at $r_M = 0.2$ (a 20% reduction). Thus the model is more sensitive to changes in the speed (ν) and recruitment rate (r_M) of immune cells as well as the division time of epithelial cells (b) than it is to changes in the fraction of initially infected cells (ρ_C) and the infection rate of healthy epithelial cells by infectious neighbours (β). It would be interesting to find out whether variations of these magnitudes can occur among healthy individuals, as this could reveal whether or not the model is too sensitive to these parameters given that we would not expect healthy individuals to die from an uncomplicated infection with a known strain of influenza A.

Next, we explored two-dimensional slices through the parameter space by varying two parameters simultaneously while keeping the others fixed, for each possible pair of the 5 parameters to which the model was found to be sensitive. Figures 5.13 and 5.14 illustrate the fraction of simulation resulting in remission when varying two parameters simultaneously for the 10 possible combinations of the 5 parameters. It is interesting to see just how close the default parameters sometimes are to outcomes other than remission

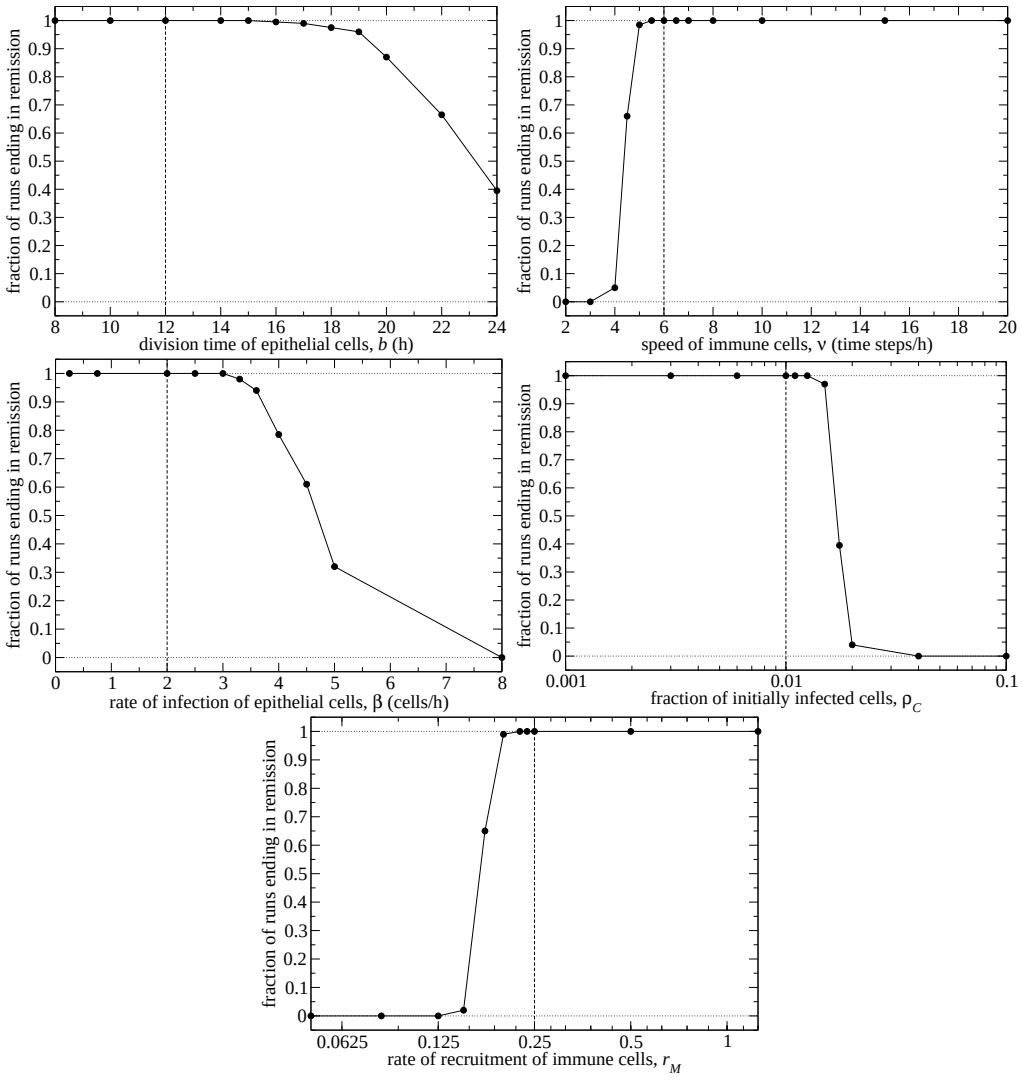


Figure 5.12: The fraction of simulations which end in remission when varying each of the parameters to which the model was found to be sensitive around its default value, marked by the vertical dashed line.

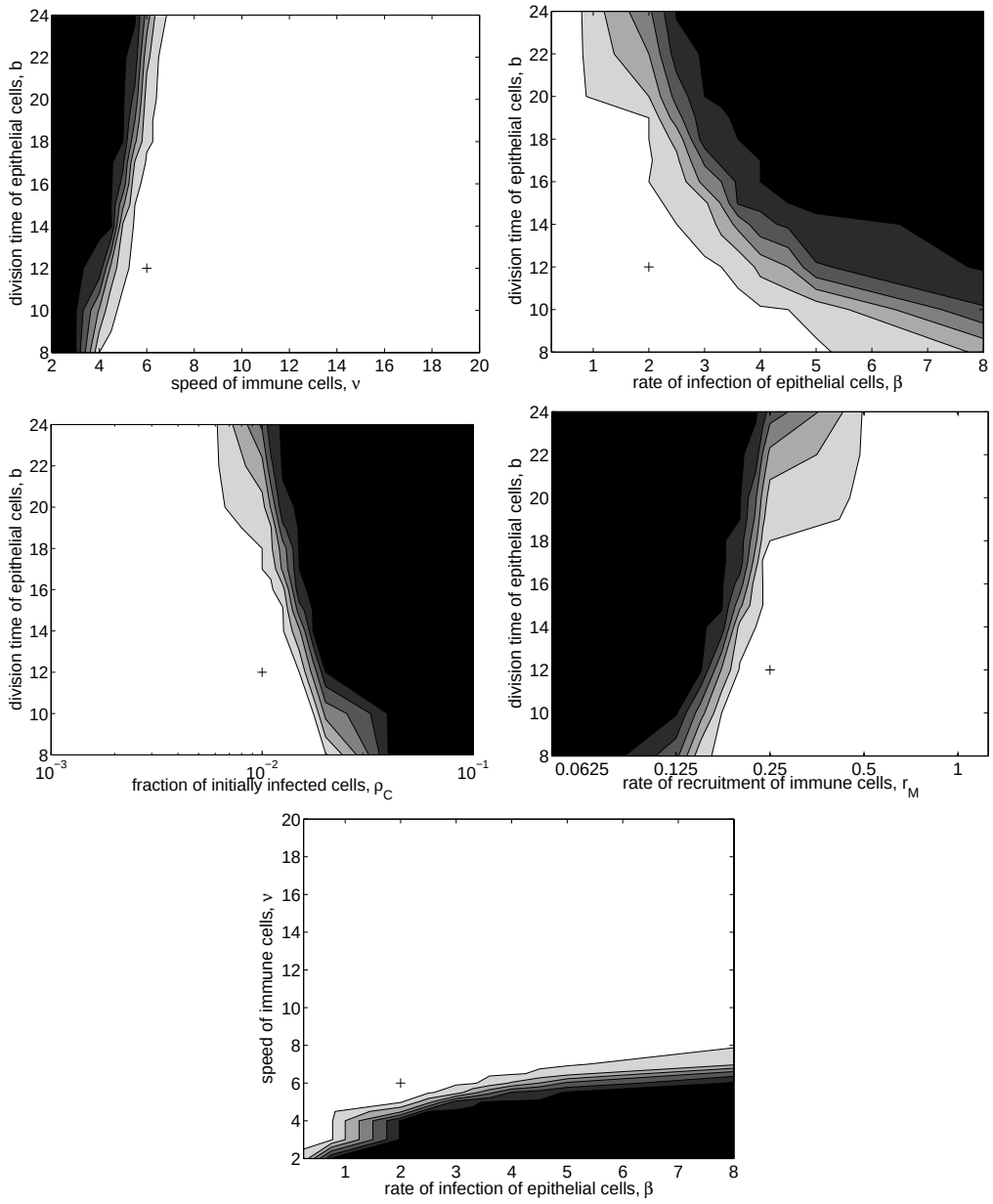


Figure 5.13: Two-dimensional slices through the parameter space for the simultaneous variation of pairs of parameters to which the model was found to be sensitive around their default value. The shading indicates the fraction of simulation resulting in remission, where white indicates that all 50 simulations resulted in remission and black indicates none. The coordinates corresponding to the pair of default values was marked by a cross in each graph.

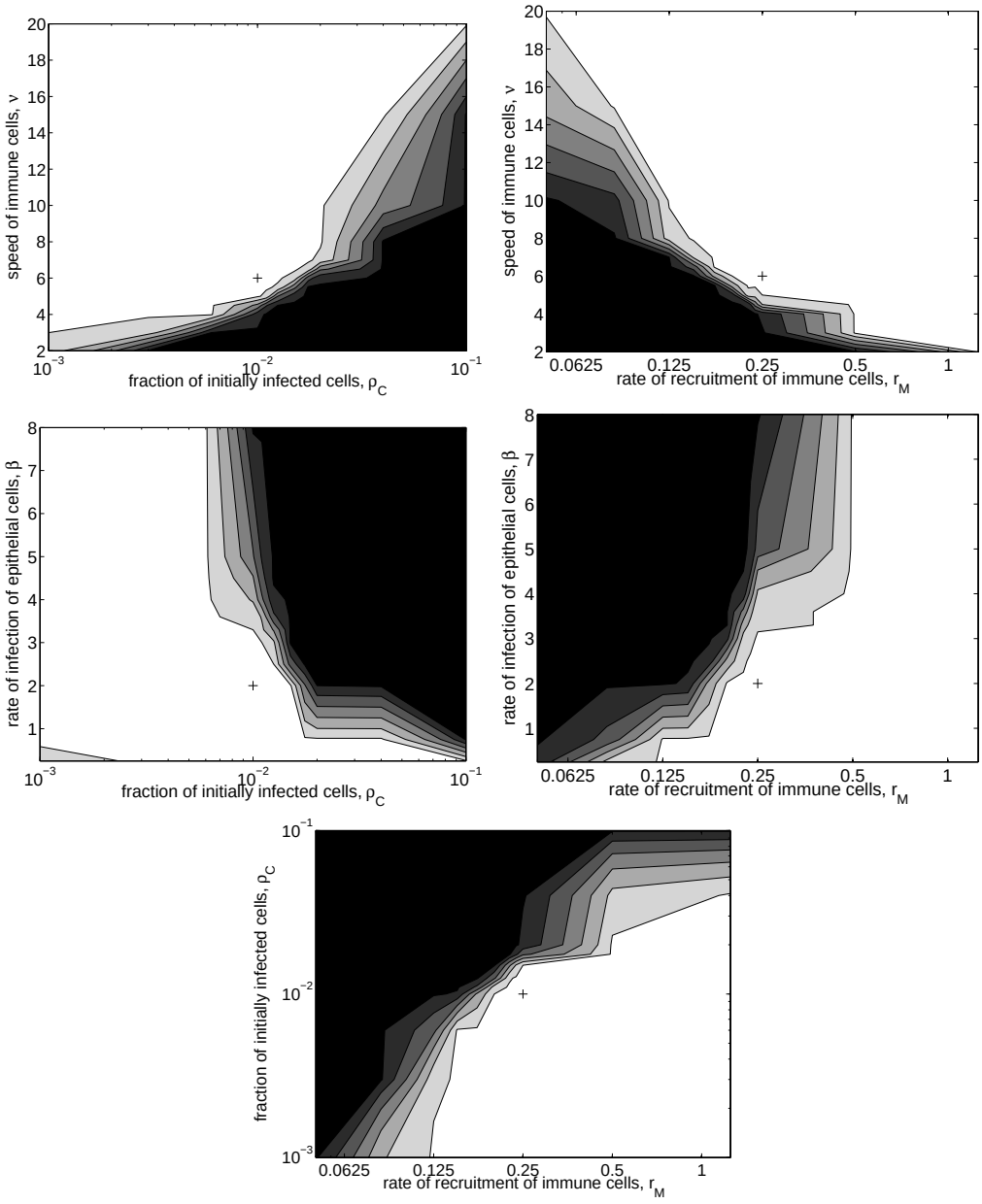


Figure 5.14: Two-dimensional slices through the parameter space for the simultaneous variation of pairs of parameters to which the model was found to be sensitive around their default value. The shading indicates the fraction of simulation resulting in remission, where white indicates that all 50 simulations resulted in remission and black indicates none. The coordinates corresponding to the pair of default values was marked by a cross in each graph.

in Figures 5.13 and 5.14. How close to the default parameters appear to be from non-remission outcomes is of course a consequence of the scale of the graphs. If the ranges explored were smaller, e.g. 8–16 h rather than 8–24 h for the division time of epithelial cells (b), the distance would appear much greater. The choice of the ranges was based on experimental data found in the literature, which are not always strict. Another possible explanation for the closeness of the default parameters to outcomes other than remission is that the immune system's response to pathogens tends towards the optimal solution for responses to all common pathogens rather than the optimal solution for a particular pathogen. In other words, the immune's response to influenza A is not optimal, but may be the optimal or near optimal compromise solution for responding to various common pathogens most effectively.

5.5 Conclusion

Here, we have introduced a CA model for an uncomplicated viral infection. We have shown that once calibrated for the particular case of influenza A, the CA model — which is described by 7 state variables and 11 parameters — is sophisticated enough to reproduce the basic dynamical features of the infection. One expects a 11 parameter model to be able to match 7 dynamical features, however all but 5 parameters of this model are sufficiently-well bound by physiological data that they cannot be used to tune its behaviour. Remarkably, the 7 variable and 11 parameter model's agreement with the experimental dynamical characteristics compares well to that of the 13 variable and 60 parameter model presented in [11]. At the time of publication of [4], the CA model presented here along with the ODE model presented in [11] were the only two existing immunological models for influenza A viral infections in humans. This is quite surprising considering influenza's significance to human health worldwide, particularly its high potential for causing devastating epidemics [87].

The results obtained with the model described herein have surpassed expectations. The goal of this project was to develop a model that would be good enough to be used as a test bench to investigate various theoretical aspects of viral infections. In fact, the influenza A model obtained after calibration behaved impressively well when compared against available experimental data. Most likely, with the inclusion of additional details in the model, such as specific immune cell types, the CA model could be a very promising model of influenza A.

A wide range of behaviours were observed as the model's parameters were varied over their biologically allowed range. It would be valuable to test whether these predicted behaviours can actually be observed in vivo, since if they are not seen then that would falsify the model.

In the next chapter, we will make use of the CA model introduced here to investigate the effects of the well-mixed assumption made by ODE models on viral infection dynamics. Additionally, in future work, we would hope to make use of the CA model presented here to investigate various other theoretical aspects of viral infection dynamics, for example, spatial strain competition and spatial restriction of viral spread by

cytokines.

Chapter 6

Probing the Effects of the Well-mixed Assumption on Viral Infection Dynamics[1]

6.1 Introduction

Here, I explore the effects of spatial structures on the dynamics of a viral infection, whose target cells are fixed in space, using the two-dimensional cellular automaton introduced in Chapter 5. I will explore which kind of effects spatial structures can have on the evolution and outcome of a spatially localized viral infection. I will also show how these spatial structures emerge and by which process they affect the dynamics of the infection.

In Section 6.2, the effect of the distribution of initially infected cells on the progression of the infection is investigated. Section 6.3 compares a local regeneration rule for epithelial cells to a global rule, i.e. the rule for the replacement of dead epithelial cells with healthy cells. In Section 6.4, the effects of the addition of immune cells at random locations versus addition at the site of recruitment are explored. Finally, in Section 6.5, the significance of the spatial effects in the particular case of an uncomplicated influenza A viral infection is discussed.

6.2 Distribution of Initially Infected Cells

In the CA model proposed in Chapter 5, the parameter ρ_C is the fraction of epithelial cells initially set in the infected state, and its default value is 1%. The cells to be initially set to the infected state were picked at random and this resulted in single infected cells as well as groupings or patches of neighbouring infected cells of various sizes. One way to investigate the effect of spatial heterogeneities on the dynamics of the infection is to change the spatial configuration of the epithelial cells that are initially set in the infected

[1] A version of this chapter has been submitted for publication to the Journal of Theoretical Biology on 23 May 2005.

state. To do this, the model was modified to distribute the initially infected cells into groups or patches of fixed size so that the effect of the size of the patches of infected cells on the dynamics of the infection can be investigated.

A new parameter, s, is added to the CA model, being the number of cells that make up a patch of initially infected cells. Since the number of epithelial cells to be initially infected is not necessarily divisible by s, the quotient of that division gives the number of patches to be added to the simulation grid at start up, and the remainder of the division is used to set the probability that an extra patch of size s be added. This means that a fixed initial patch size is enforced at the expense of a fixed fraction of initially infected cells. Each patch of infected cells is individually constructed and is added at a random location on the grid, insuring that no two patches are in contact with each other. The CA model defines the neighbourhood of a site as consisting of the site itself and its eight closest sites (Moore neighbourhood). A patch of s infected cells is constructed by starting with a seed site and growing it by sequentially picking one site at random from the set of sites that neighbour previously-selected sites. Note that this method of forming patches results in patches with densities that decrease with increasing distance from the centre. This characteristic is consistent with a splatter or spray of virions and thus this method was preferred over other patch growing methods such as diffusion-limited aggregation, and random walk additions around a seed.

The results for patches ranging in size from 1 to 1232 infected cells are presented in Figure 6.1. One can see that increasing initial patch sizes result in fewer infected cells and less epithelial damage. This is not surprising since only the cells that make up the perimeter of the patch, i.e. those that have healthy neighbours, can infect other cells. As patches grow, their perimeter to area ratio, namely the fraction of infectious cells that have healthy neighbours, will decrease and so will the effective infection rate.

Let us illustrate this by an example. Consider a system where infected cells infect all of their uninfected Moore neighbours (8 nearest neighbours) in each time step (an infection rate of 100%). The evolution of the system from an initial single seed is illustrated in Figure 6.2. From the relation derived in the table of Figure 6.2, one can compute the effective infection rate, i.e. the number of newly infected cells per infected cell at time step n, to be $8(n+1)/(2n+1)^2 = 4/\sqrt{I} + 4/I$, where $I = (2n+1)^2$ is the number of infected cells in a square patch at time step n. A graph of the effective infection rate as a function of the number of infected cells in a square patch is presented in Figure 6.3. For this toy model, the effective infection rate is proportional to $1/\sqrt{I}$ for $I \gg 1$.

Another interesting feature that can be seen in Figure 6.1 is the increasing standard deviation for increasing initial patch sizes. This is easily explained with the fact that the larger the parameter s, the fewer the sites of infection. In other words, as the initial patch size increases, the 50 simulations are averaging over fewer infection sites. Figure 6.4 presents two example simulations to illustrate the differences that can arise between simulations produced using the same parameter values, when the initial patch size is large. In the case of the example simulations presented in Figure 6.4, early detection made the difference between a small and short infection, and a longer infection resulting in a greater number of infected and dead cells. The larger the initial patch size, the fewer the number of infected patches and thus, the more pronounced this effect will be.

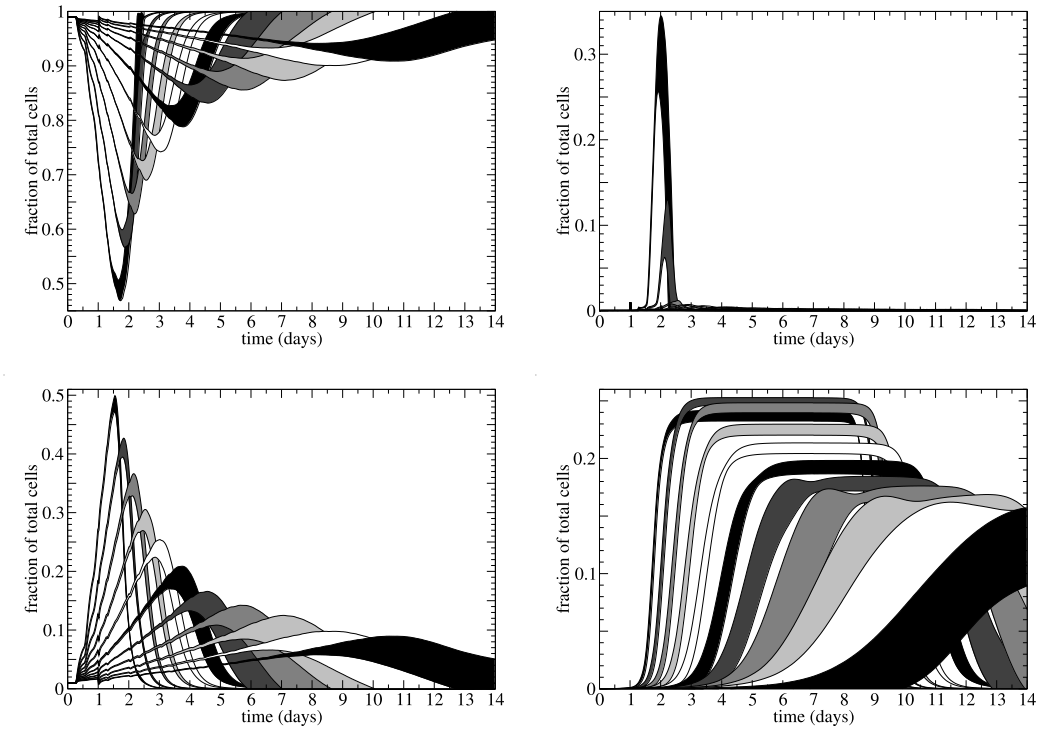


Figure 6.1: The effect of varying the initial patch size, s, on the viral infection's dynamics. The graphs show the time evolution of the populations of healthy (top left), dead (top right), infected (bottom left), and immune cells (bottom right) for s values of 1, 2, 4, 8, 16, 35, 77, 154, 308, 616, and 1232 cells. The greyed areas mark one standard deviation after 50 runs for each initial patch size, with periodically decreasing darkness corresponding to increasing initial patch sizes. In all cases, the black band that peaks first is $s = 1$. The graphs show that the dynamics of the viral infection is sensitive to the spatial organization of the initially infected epithelial cells.

Time	# inf.	# new inf.
0	1	8
1	9	16
2	25	24
3	49	32
⋮	⋮	⋮
n	$(2n+1)^2$	$8(n+1)$

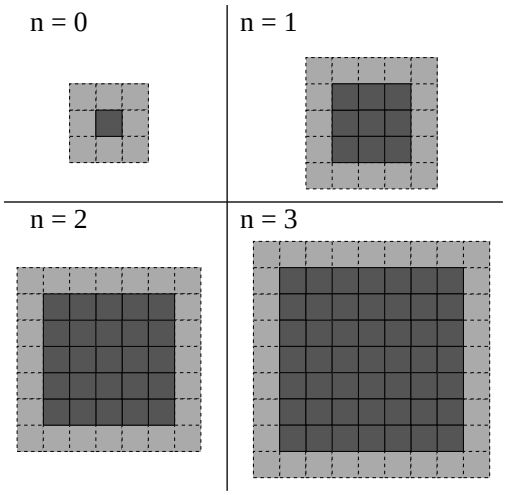


Figure 6.2: Evolution of a simplified system where each infected cell infects all of its uninfected neighbours at each time step, starting from a single infected cell. The table shows the number of infected cells and the number of cells that will become infected in the next time step. The figure illustrates the evolution of the system over the first 4 time steps with infected cells represented in dark grey and the cells which will be infected in the next time step represented in light grey.

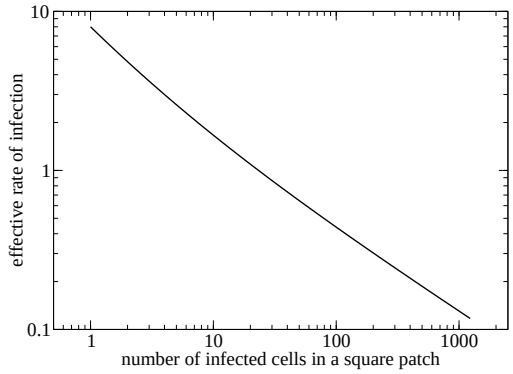


Figure 6.3: The effective rate of infection (newly infected cells per infected cell) as a function of number of infected cells in a square patch for the simplified system presented in Figure 6.2. The effective infection rate is given by $4/\sqrt{I} + 4/I$ where I is the number of infected cells that make up the square patch.

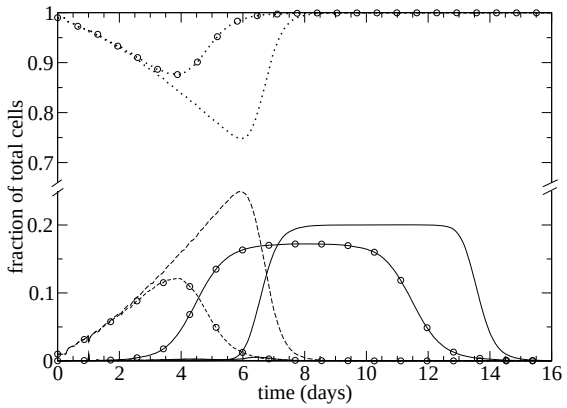


Figure 6.4: Proportion of healthy cells (dotted), infected cells (dashed), and immune cells per epithelial cell (full) for two simulations using an initial patch size of $s = 77$. The simulations, whose only difference is the seed for the random number generator, illustrate the differences that can arise for large values of the initial patch size. In this case, early immune detection (lines with circles) of the infection has allowed minimal damage and early recovery, while late detection (lines without symbols) has resulted in a longer infection with a larger number of infected and dead cells.

This variability for larger values of s can be reduced by averaging simulations with the same number of infection sites (same number of patches) rather than the same absolute number of infected cells (same area).

Finally, it can be seen that there is a decrease in peak immune cell concentration for initial patch sizes below $s = 2$. It is clear that there are two processes at work: one which dominates at small initial patch sizes and one which dominates at large initial patch sizes. As seen in the bottom left of Figure 6.1, the peak number of infected cells decreases monotonically as the initial patch size is increased. The peak concentration of immune cells is, mostly, determined by the peak number of infected cells, and this explains the decrease in the peak concentration of immune cells as the patch size increases. However, I have yet to determine the process responsible for the decrease in peak immune cell concentration at small initial patch sizes.

6.2.1 Not Just a Rescaling Problem

It may be tempting to interpret the effect of the initial patch size on the development and outcome of the infection as a rescaling of the system. In effect, one could imagine that each lump of infected cells represents a single infected cell such that the surface area of one epithelial cell corresponds to s sites of the simulation grid. A grid of area A with an initial patch size of s would be equivalent to a grid of area A/s with an initial patch size of 1. This turns out to be an incorrect interpretation, as seen in Figure 6.5. This figure illustrates that one consequence of increasing the number of simulation sites per epithelial cell is an increase in the number of configurations the simulation can be

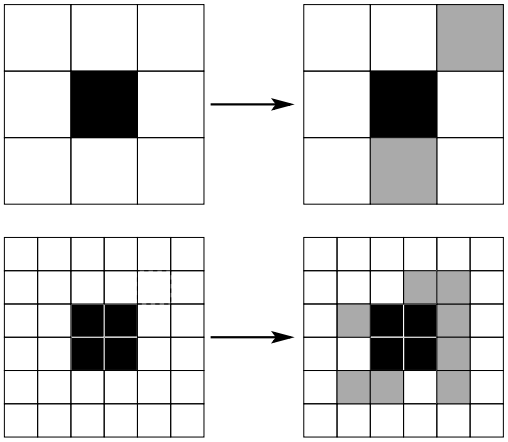


Figure 6.5: The comparison of the infection growth pattern for a simulation where each epithelial cell is represented by: (top) a single grid site; or (bottom) 4 grid sites ($s = 4$). For the infection growth rate to be comparable for the 2 simulations, the fraction of the grid which gets infected needs to be kept constant such that an infection rate β for an initial patch size of 1 becomes $\beta \cdot s$ for an initial patch size of s. Despite this correction, the infection growth pattern is not equivalent because, for example, the radius of the infection increases faster in the former.

in. For example, this causes the radius of infection sites to grow more slowly, even when the cell-to-cell infection rate is increased so that the rate of increase of infected tissue area is kept constant.

6.2.2 Occurrence of Chronic Infection

It is not clear from Figure 6.1, but for initial patch sizes larger than 35, a number of simulations result in chronic infection with the fraction of infected cells stabilizing at 2% in all such cases. The occurrence of chronic infection increases for increasing initial patch sizes. This is illustrated in the top left panel of Figure 6.6. What causes chronic infections in the case of larger initial patch sizes is the lower effective infection rate, which slows the infection dynamics. If the infection growth is slowed down, the infection takes place over a longer period of time and the immune cells start dying off before the infection is fully cleared. Thus, in the CA model, chronic infection arises when the immune cells' lifespan is shorter than the time scale of the infection. Chronic infections can be prevented by choosing a larger value for δ_M, the lifespan of immune cells, for larger values of s, the initial patch size. For $s = 1232$, there are still occurrences of chronic infection with $\delta_M = 300$ h, but the infections are always cleared for $\delta_M = 400$ h (not shown).

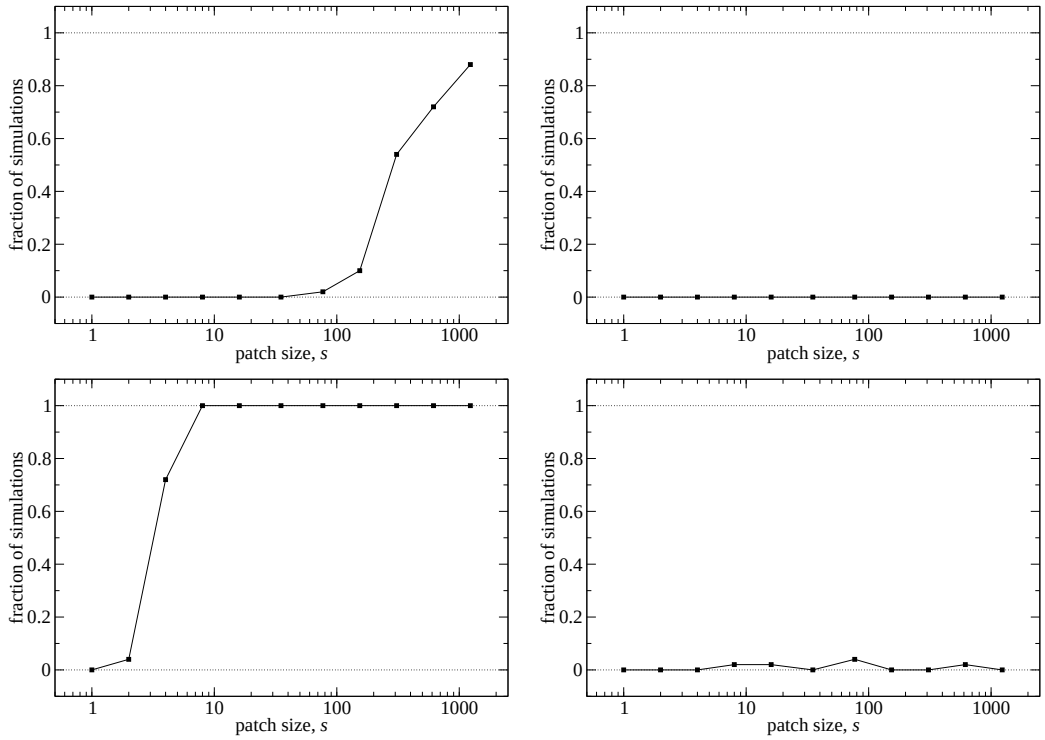


Figure 6.6: Fraction of simulations ending in chronic infection as a function of the initial patch size, using the global (top row) or local (bottom row) epithelial cell regeneration rule, with the addition of immune cells at random locations (left column) or at the site of recruitment (right column). The results were obtained by averaging over 50 simulation runs.

6.3 Global vs Local Epithelial Regeneration

In the model presented in Chapter 5, the regeneration of dead epithelial cells was implemented as a global process rather than a local process, namely, a dead cell is replaced by a healthy cell with probability $b^{-1} \times \#$ healthy/# dead. See rule 4 of Figure 5.1. This epithelial cell regeneration rule was originally chosen to mimic the replacement of dead cells by basal cells or by cells from inferior layers in the context of an influenza A infection. If one, instead, considers an infection taking place in a tissue composed of a monolayer of cells, a local regeneration rule based on the division of immediate neighbours is more appropriate. In this section, the impact of using the local epithelial cell regeneration rule on the dynamics of the infection is investigated. Local regeneration of epithelial cells is modelled by altering rule 4 of Figure 5.1 so that a dead epithelial cell is replaced by a healthy one only if one of its healthy neighbours divides. Note that for both epithelial cell regeneration rules, division or regeneration is simply the process by which a dead cell is replaced by a healthy cell. If there are no dead cells, nothing happens, no regeneration rule is invoked.

The original global regeneration rule is equivalent to assuming that dead and healthy epithelial cells are homogeneously distributed throughout the simulation grid, which is the way in which epithelial regeneration is implemented in simple ODE models. Comparing the two regeneration rules allows us more insight into the effect of the spatial distribution of cells on localized infection dynamics. The results of simulations comparing the global to the local epithelial cell regeneration rules are shown in the left column of Figure 6.7. The top left panel shows the original model with the global epithelial cell regeneration rule, as presented in Chapter 5, and the bottom left panel shows the same model using the local epithelial cell regeneration rule. A typical spatial distribution of cells at day 4 post-infection for both rules is illustrated in the left column of Figure 6.8 as screenshots of the simulation grid, with the panels in the same order as in Figure 6.7. Additionally, the numbers of infected and dead cells at their respective peaks relative to their values in the original CA model introduced in Chapter 5 are presented in Table 6.1 in the two rows labelled "newly recruited immune cells placed at random locations;" the other rows will be discussed in the next section.

One can see that the local epithelial cell regeneration rule results in fewer infected cells and, consequently, in the recruitment of fewer immune cells but in more extensive and longer lasting damage to the epithelium compared to the global regeneration rule.

In the CA model, the infection of epithelial cells spreads locally as infected cells infect their healthy neighbours forming growing patches of infected cells. As the infection progresses, infected cells at the core of these patches die as a result of virus toxicity or immune attacks, and leave behind patches of dead cells surrounded by a perimeter of infected cells. Patches of dead cells can no longer harbour infection and thus serve to limit the growth of the infection. With the global epithelial cell regeneration rule, new healthy cells are allowed to emerge in the middle of the pools of dead cells. This allows the infection to rapidly repopulate the patches of dead cells, thus sustaining a high level of infection with minimal epithelial damage.

With the local epithelial cell regeneration rule, the patches of dead epithelial cells

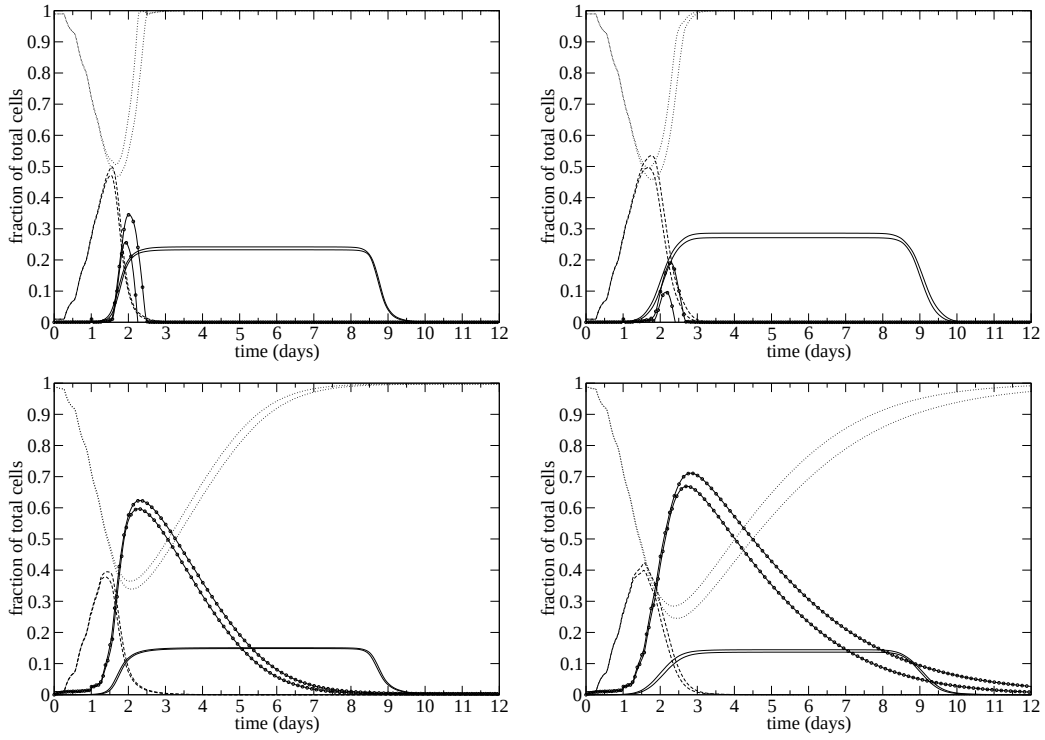


Figure 6.7: The effect of a global (top row) or local (bottom row) epithelial cell regeneration rule with the addition of immune cells at random sites (left column) or at the site of recruitment (right column) on the behaviour of the CA model. Simulation results averaged over 50 simulation runs for an initial patch size of 1. The paired lines mark one standard deviation and represent the fraction of epithelial cells that are healthy (dotted), infected (dashed), dead (full with circles), as well as the proportion of immune cells per epithelial cells (full). The top left panel corresponds to the original model presented in Chapter 5.

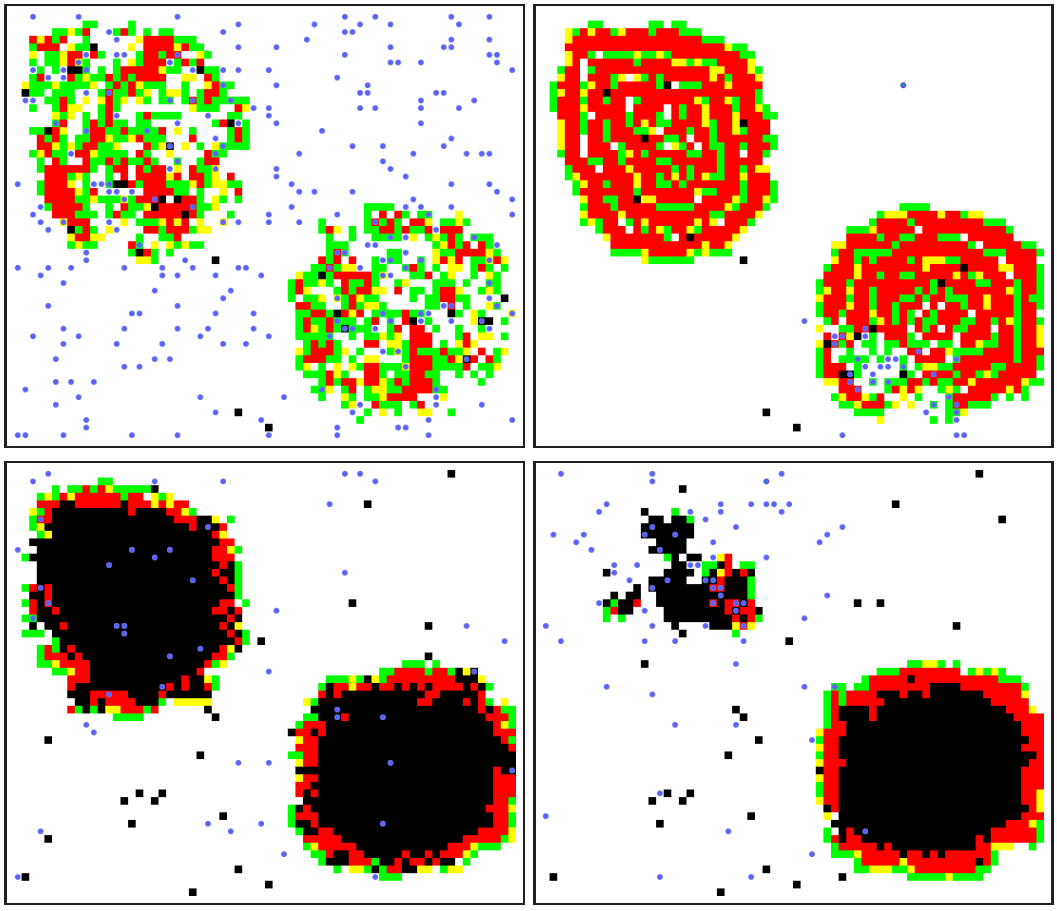


Figure 6.8: Partial screenshots of 4 simulations obtained using the same parameter values and initial cell distribution 3.5 d post-infection with an initial patch size of $s = 25$, using the global (top row) or local (bottom row) epithelial cell regeneration rule, with the addition of immune cells at random locations (left column) or at the site of recruitment (right column). Healthy epithelial cells are white, containing cells are green, expressing cells are yellow, infectious cells are red, and dead cells are black. Immune cells are blue circles. The top left panel corresponds to the original model presented in Chapter 5.

epithelial cell regeneration occurs	newly recruited immune cells placed at	maximum infected cells (relative)	maximum dead cells (relative)
globally	random locations	1.0	1.0
	recruitment site	1.1	0.46
locally	random locations	0.80	2.1
	recruitment site	0.85	2.3

Table 6.1: The effects of the epithelial cell regeneration rules and the immune cell recruitment rules on the number of infected and dead cells at their respective peak. The numbers are relative to their values for the rules of the original model introduced in Chapter 5, namely global epithelial cell regeneration with the addition of immune cells at random locations.

can only be repopulated by healthy cells once the immune cells have begun destroying the rings of infected cells that encircle each patch of dead cells, which otherwise act as a barrier isolating healthy cells from the areas that require regeneration. Thus, the greater accumulation of damage that results from the use of the local regeneration rule is a consequence of the spatial constraints imposed on the regeneration process. This finding is in agreement with that of [76], who reported that for their spatial model of HIV, the infection could only be sustained as a propagating wave when the local rate of cell death was greater than the local regeneration rate, as is the case with the model used here when using the local regeneration rule for epithelial cells.

6.3.1 Occurrence of Chronic Infection

Examination of the results of the local epithelial cell regeneration rules for various initial patch sizes reveals the persistence of infected cells, namely a chronic infection stabilizing at approximately 1% of cells infected, for all but an initial patch size of 1. This is illustrated in Figure 6.6, where the fraction of simulations ending in chronic infection as a function of the initial patch size for the local epithelial cell regeneration rule is presented in the bottom left panel. The smaller number of infected epithelial cells resulting from the use of the local regeneration rule results in the recruitment of fewer immune cells making it harder to fight the viral infection. Additionally, the organization of the infected cells into circular waves makes it harder for the immune cells to target the infected cells' structures. When infected cells are arranged into patches, an immune cell performing a random walk has better chances of landing on multiple infected sites. When infected epithelial cells organize into rings, as is the case with the local regeneration rule, immune cells performing a random walk will often move off the ring structure and "lose sight" of the infection. Consequently, the smaller number of infected cells and their organization into circular waves, facilitates the escape of the infection from immune attacks resulting in a higher incidence of chronic infections than for a global epithelial cell regeneration rule.

6.4 Immune Cells' Proliferation Rule

The proliferation of immune cells in the model presented in Chapter 5 was such that when an activated immune cell moved onto an expressing or infectious cell, new activated immune cells are added at a rate of $r_M = 0.25$ at a random location on the grid. The addition of immune cells at random locations can be justified biologically by the scenario of immune cells being activated and proliferating in the lymph nodes, travelling to the site of infection, and surfacing at random locations throughout the infected tissue. But immune expansion could instead be modelled by adding new activated immune cells on the site where the recruiting activated immune cell is located, hence mimicking immune cell (T cell, macrophages, etc.) division at the infection site. This scenario could correspond to immune cells being activated in the lymph nodes, but travelling to the site of infection while still undergoing their programmed cycles of divisions. In Figure 6.7, the infection dynamics for the addition of immune cells at random locations and at the site of recruitment are compared for the two choices of epithelial cell regeneration rule. A typical spatial distribution of cells at day 4 post-infection under the two immune cell proliferation rules for both epithelial cell regeneration rules are illustrated in Figure 6.8 as screenshots of the simulation grid. Additionally, the numbers of infected and dead epithelial cells at their respective peaks for all rules relative to their values in the original model presented in Chapter 5 are presented in Table 6.1.

Regardless of the epithelial cell regeneration rule, the addition of immune cells at the site of recruitment results in more infected cells at the peak of the infection than addition at random locations. The addition of immune cells at random locations allows recruited immune cells to surface randomly onto a previously unexplored site and efficiently discover new patches of infection. With the addition of immune cells at the site of recruitment, it takes longer for immune cells to discover new sites of infection as they can only find them by diffusion. Thus, although the discovered infection sites are cleared faster and more efficiently with the addition of immune cells at the site of recruitment, the undiscovered infection sites are allowed to grow for longer, resulting in more infected cells overall.

In contrast, the addition of immune cells at the site of recruitment rather than at random locations has a different impact on the number of dead cells at the peak for the two epithelial cell regeneration rules. The addition of immune cells at the site of recruitment results in fewer dead cells when combined with the global epithelial cell regeneration rule, but more dead cells when combined with the local regeneration rule. This discrepancy in the effects of the choice of immune cell addition rule for the two epithelial cell regeneration rules can be explained as follows. For the global epithelial cell regeneration rule, the addition of immune cells at random locations allows the infection to grow almost undisturbed while the immune cells slowly populate the grid randomly through recruitment, mainly landing on healthy sites. But when a sufficient number of immune cells have been added, such that new immune cells tend to be placed on infected sites, the destruction of infected cells by immune cells begins and happens very abruptly. It is this abrupt destruction of infected cells by immune cells that results in the greater number of dead cells seen with the addition of immune cells at random locations rather

than at the site of recognition with the global epithelial cell regeneration rule. This also happens when using the local epithelial cell regeneration rule, but in this case the effect is masked by the large increase in cell destruction at undiscovered infection sites. In fact, with the addition of immune cells at the site of recruitment and the local epithelial cell regeneration rule, the undiscovered site are sometimes allowed to grow to such extent that the infection gets cleared by target-cell limitation in those areas.

It might seem at first that the greater number of infected cells resulting from a non-cytopathic pathogen (as cells are no longer dying from the cytopathic effects of the virus) would result in a very extensive amount of damage at the onset of the abrupt destruction of infected cells. However, with this CA model, this does not happen. The results of setting the infected lifespan to $\delta_I = 4 \times 10^6$ h, which is much longer than the duration of the simulation, were compared with those obtained using the original model presented in Chapter 5 in which $\delta_I = 24$ h. The comparison revealed that lengthening the lifespan of infected epithelial cells does not have a significant effect on the resulting dynamics and does not result in extensive epithelial damage. This is because older infectious cells, regardless of their lifespan, find themselves at the centre of infected patches, and so do not contribute to the infection spread since they do not have healthy neighbours to infect. In the case of short lived infected cells (cytopathic pathogen), for example with $\delta_I = 24$ h, dead infectious cells are replaced with healthy cells which are then re-infected and the configuration of the simulation is essentially unchanged from the case of long lived infectious cells. The only difference is that, in the case of long lived infected cells (non-cytopathic pathogen), the uninterrupted presence of the infectious cells causes slightly more immune cells to get recruited.

6.4.1 Occurrence of Chronic Infection

Examination of the runs in which immune cells are added at the site of recruitment rather than at random locations reveals a dramatic decrease in the fraction of simulations ending in chronic infection. The addition of immune cells at the site of recruitment using the global epithelial cell regeneration rule produced no chronic infection in any of the 50 simulations performed for each initial patch size. Using the local epithelial cell regeneration rule, the addition of immune cells at the site of recruitment produced only a handful of simulations resulting in chronic infection, with the fraction of infected cells stabilizing at approximately 0.1% in all cases. This is illustrated in Figure 6.6.

The reduction in the fraction of simulations resulting in chronic infection when adding immune cells at the site of recruitment rather than at random locations is easily explained. At high infection levels, the addition of immune cells at the site of recruitment increases the efficacy of the response at the site of recruitment but makes it harder for immune cells to find other sites of infection. This results in a greater number of infected cells. But at low infection levels, immune cells added at random locations will rarely be added at an infection site and are likely to die of old age before they can diffuse to an escaped infection foyer. Thus, the addition of immune cells at random locations is the better strategy for high levels of infection allowing rapid detection of the various infection sites, while addition at the recruitment site is the better strategy for low levels

of infection allowing efficient prevention of escape.

6.5 In the Context of Influenza A

Influenza is a good example of a spatially localized viral infection. The infection typically takes place in the upper sixteen generations of the lungs, and the target cells of the infection, the ciliated epithelial cells which cover the respiratory tract, are fixed in place. In Chapter 5, the CA model used here was introduced and successfully calibrated to mimic a viral infection with influenza A. Here, I revisit the CA model to explore how the local epithelial cell regeneration rule and the immune cell addition rule affect the agreement between the CA model and the experimental data cited in Chapter 5 for an uncomplicated influenza A viral infection.

Because the target cells of influenza A are fixed, i.e. do not move around in space, it is ultimately the speed of diffusion of the virions over the epithelial layer which determines whether the population of infected cells grows locally around a productively infected cell, or in a more homogeneous manner as the virions quickly spread out over the target area. But since the lifespan of a productively infected cell, the number of virions it produces, their clearance rate, and their diffusion pattern in the cilia-beaten mucus are not well known in the case of influenza A, it is difficult to estimate how far and how quickly the infection spreads. Consequently, it is hard to assess the extent to which the infection process, as implemented in the model, applies to the particular case of influenza A. For example, in Section 6.2, it was shown that larger patches of infection lead to a decreased effective infection rate. But if occasional jumps in viral spread to previously uninfected areas were to occur in vivo, they could keep the effective infection rate high, by giving the infection access to areas where target cells are still plentiful. Nevertheless, there are still some conclusions to be drawn from the results presented above.

Originally, in Chapter 5, the use of a global epithelial cell regeneration rule seemed appropriate to mimic the replacement of dead cells by basal cells or by cells from inferior epithelial layers. But in the particular case of influenza A, the infection targets the airway epithelium which consists of a single layer of cells everywhere except in the trachea [68]. Thus, it would seem that a local regeneration rule by which a dead epithelial cell is replaced by a healthy cell only if one of its healthy neighbours divides is more appropriate to model cellular regeneration following a viral infection in the lungs. As it turns out, the use of the local epithelial cell regeneration rule does in fact improve the fit of the CA model to available experimental data. Over the course of an influenza infection, there should be about 10% of cells dead on day 1, 40% on day 2 and 10% on day 5 [11]. The global rule results in too fast a regeneration, but the local rule improves the agreement of the number of dead epithelial cells during regeneration.

The local epithelial cell regeneration rule also results in a number of infected cells at the peak of the infection ($\sim 40\%$ of the total) which is smaller than that obtained with the global regeneration rule ($\sim 50\%$ of the total). Unfortunately, there is no data available to assess whether the reduction in the number of infected cells at the peak of the infection constitutes an improvement of the model or not. The other two existing mathematical

models of influenza A, which are ODE models, have arrived at numbers of infected cells at infection peak of 40%–78% [1], and 60%–80% [11] of the total. Experimental data about the fraction of cells infected at the peak of the infection would therefore be invaluable in discriminating between the different models for influenza A and help determine whether spatial heterogeneity plays a role in the development and outcome of the infection.

Finally, it has been suggested in [1] that influenza resolution could be target-cell limited. This means that the infection would die from the lack of new cells to infect, rather than as a result of immune attacks. With the model in its current state, target-cell limitation can occur locally, as seen using the local epithelial cell regeneration rule with the addition of immune cells at the site of recruitment (see Section 6.4).

In the absence of immune cells, target-cell limitation is such that sites of infection grow undisturbed and as the circular waves of infection meet and annihilate, they leave behind nothing but dead cells. Target-cell limited complete resolution of the infection, without the death of all cells, does not occur in the model because as long as the infection wave encircles the dead epithelial cells, segregating them from healthy cells, regeneration cannot be initiated. It is only once immune cells have started attacking the propagating infection wave, creating breaks where dead cells can be in contact with healthy cells, that epithelial cell regeneration can take place.

Target-cell limited resolution could be explored in the absence of immune cells, for example, if the action of cytokines were included in the model. The various cytokines which get produced during an influenza infection are known to hinder viral replication within infected cells, and confer a certain level of protection from infection in surrounding cells [11, 1]. In the CA model, the cytokine response could be modelled by introducing, for example, an inhomogeneous infection rate or an infection rate that would depend on the number of infectious neighbours. This could be the subject of future research.

6.6 Conclusion

Here, the CA model introduced in Chapter 5 was used to investigate the effects of the well-mixed assumption on the dynamics of a localized viral infection. It was shown that the distribution of initially infected cells has a great impact on the dynamics of infection. This is because, in the CA model, infectious cells can only infect their immediate neighbours, and when organized in patches, fewer infectious cells have healthy neighbours.

It was also demonstrated that the regeneration rule chosen for the replacement of dead epithelial cells by healthy ones can have an important impact on infection dynamics. A global epithelial cell regeneration rule, equivalent to simple ODE models, allows areas of dead cells to be replenished by healthy cells even in the local absence of healthy cells. This repopulation, in turn, allows the infection to move back into the newly replenished area it had previously infected, resulting in a greater number of infected cells. On the other hand, the slower local regeneration rule, which requires the local presence of healthy epithelial cells, limits the growth of the infection by starving it of target cells and forces the infection to propagate as a thin circular wave. [76] introduce a spatiotemporal model

for the dynamics of HIV in the spleen. Strain et al. point out that the main differences between their spatial model and a mean field approach such as the basic viral infection ODE model [63, 65], arise from the fact that a viral burst only spreads to nearby cells. They also conclude that in a spatial model, infection sustainability is affected by the recovery rate of destroyed target cells, as local cell destruction limits the spread of the infection which can then only be sustained as a propagating wave. Those findings are in agreement with those presented here.

Then, the choices of whether to add immune cells at random locations on the simulation grid, as is equivalent to simple ODE models, or at the site of recruitment were compared to explore how they affect the dynamics of the infection. It was shown that while addition at random sites permits rapid detection of new infection sites, it makes it harder to avoid infection escape from the immune response. Consequently, random addition of immune cells was found to be a better strategy at high infection levels, while addition at the site of recruitment was the better strategy at low infection levels.

The simulation has also been observed to yield chronic infections for certain rules and patch sizes. It is important to specify that the term "chronic infection" is used here to designate a very small fraction (at most 2%) of infected cells persisting beyond at least 60 days post-infection. At this low level of infection, it is unlikely that patients would be symptomatic. Since the patient's nasal cavities in the absence of symptoms (e.g. runny nose) are dryer, it would be difficult to detect any virus shedding from nasal wash. For this reason, I do not beleive that current experimental data for influenza can rule out the possibility of a low-level persistent infection. Of course, if it were in fact the case that a low-level of infection can persist, this could have very interesting consequences for memory maintenance, and could possibly provide a reservoir for epidemic spread and strain maturation. Much more sensitive tests than those currently in use would need to be performed to rule out or confirm this possibility.

Two spatial models [90, 76] have been suggested for the dynamics of HIV infections. Both models make the assumption that T cells, the target cells of HIV virus, are fixed in space, an assumption that is not realistic given the known patterns of movement of T cells within lymph nodes [55, 54, 53, 52, 51], and may adversely affect the results. Other investigations [29, 48] have chosen to remain more general in their exploration of the effects of the spatial distribution of agents on the evolution and outcome of infections by not considering a particular viral infection. Since the models in [29, 48] have not been calibrated to fit experimental data, it is not known whether they can realistically model any particular infection. In Chapter 5, the model used here was calibrated for influenza A, and was shown to be accurate enough to quantitatively reproduce the response to an uncomplicated infection with this virus. The applicability of the findings presented here follow from that model.

In the present work, the effect of the spatial distribution of infected cells on the dynamics of the infection arises from the fact that the infection can only spread from one infectious cell to its neighbours. The applicability of the findings presented here largely depends on the accuracy of this assumption, namely whether the infection tends to quickly spread over the tissue or grow locally around infected sites. Nonetheless, I have shown in this chapter that a local epithelial cell regeneration rule, where a dead

cell is replaced by a healthy cell when one of its immediate healthy neighbour divides, improves the fit of the CA model to experimental data in the case of an uncomplicated viral infection with influenza A.

Whether or not there exist in vivo virus-host systems where the infection grows locally from neighbour to neighbour, such systems do exist in vitro and are used to address questions such as how viral spread is inhibited by cellular antiviral activities [20, 45]. The team of Dr. John Yin, at the University of Wisconsin-Madison, have introduced a new assay method which consists of a monolayer cell culture covered in an agar solution, which prevents the diffusion of virions at the surface of the cell monolayer such that the infection can only spread to immediate neighbours [20, 45], as is the case in the CA model used here. By complementing these assay experiments with simulations from the CA model used here, significant questions could be addressed. For example, by testing various hypotheses about the production and spread of interferon, and comparing the results of the CA model to that of the experimental assays, it may be possible to discriminate among various potential mechanisms and extract parameters for those mechanisms, such as rate of production/clearance of interferon. The combination of results obtained through such experimental techniques with the flexibility and simplicity offered by spatial *in silico* modelling could lead to great advances in our understanding of host-pathogen interactions.

Chapter 7

A Simple Model for T Cell Movement within Lymph Nodes in the Absence of Antigen

7.1 Introduction

Movement of T cells within lymph nodes facilitates the interactions of T cells with other components of the immune system and enables transmission of information for immune surveillance and response. Often, as in the case of HIV, immune cell interactions also allow the spread of infection within an individual.

Two-photon microscopy has allowed the direct visualization of the movement of T cells within lymph nodes, giving new insights into immune interactions. Recent papers have presented observations of T cell movement that can be used to extract the scheme of movement of T cells within lymph nodes [51, 53, 54, 55]. Remarkably, trajectories of individual T cells obtained from these studies suggest that T cells motion is random and not directed by chemokine gradients over large distances. Immune recognition of antigen within lymph nodes thus appears to occur through the random encounters of T cells with antigen presenting cells.

This chapter assembles and analyzes the available data and proposes a simple model for T cell movement. The model's simulations capture observed T cell trajectories quantitatively and yield estimates of underlying parameters which characterize T cell motion in vivo. The resulting insights will be particularly useful given the growing interest in the development of spatial mathematical models for viral infections [3, 4, 10, 21, 29, 76, 90]. In Section 7.2, a summary of the literature is given. Then, in Section 7.3, the experimental data is analyzed with the aim of identifying a simple description of T cell movement. Finally, in Section 7.4, we propose a model for the movement of T cells within lymph nodes in the absence of antigen.

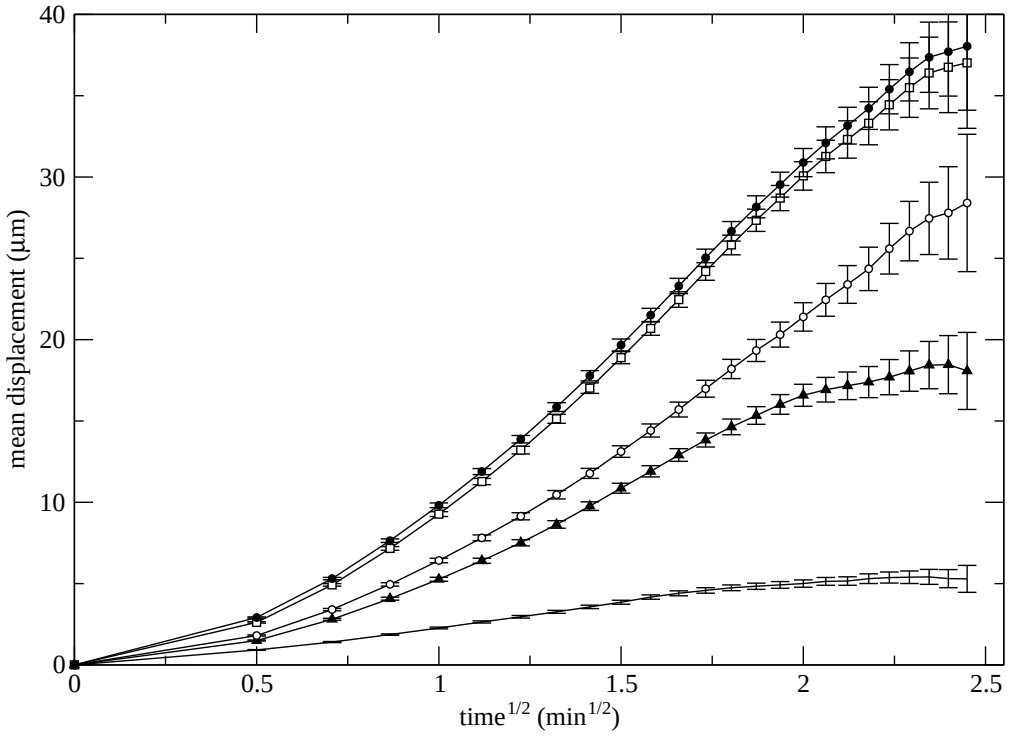


Figure 7.1: Mean displacement as a function of the square root of time in one dimension in x (filled triangles), y (empty circles), z (no symbol), in two dimensions in the xy-plane (empty squares), and in three dimensions (filled circles) for up to 146 $CD8^+$ T cells in mice popliteal lymph nodes in the absence of antigen-loaded activated dendritic cells. Data from [51].

7.2 Summarizing the Literature

The experimental data considered to construct our T cell movement model was taken from 4 papers, namely [51, 53, 54, 55]. These papers were chosen because they presented the mean displacement of T cells as a function of the square root of time, which we used to construct our T cell movement model. The experimental context and the types of cells considered are summarized in Table 7.1 for each publication.

As can be seen from Table 7.1, Miller et al. have projected their mean displacement versus $\sqrt{\text{time}}$ to two dimensions (2d), while Mempel et al. did not. Dr. Mempel was kind enough to provide us with data containing the individual $CD8^+$ T cell tracks (t, x, y, z) that were used to produce the graph of the 3d mean displacement versus $\sqrt{\text{time}}$ presented in Figure 2 of [51]. Figure 7.1 presents the mean displacement of T cells versus $\sqrt{\text{time}}$ in 1d, 2d, and 3d, that we computed from these cell tracks. Closer examination of the 1d mean displacements reveals that it is not consistent across the x, y, and z directions. The 1d mean displacement versus $\sqrt{\text{time}}$ in the z direction is so much smaller than that

reference	cell type	projected to 2d	experimental conditions
Miller et al. 2002 [55]	$CD3^+$ ($CD4^+$ or $CD8^+$)	yes	inguinal and cervical lymph nodes harvested from mice
Miller et al. 2003 [54]	$CD4^+$	yes	inguinal lymph nodes of anaesthetized mice
Miller et al. 2004 [53]	$CD4^+$	yes	cervical or inguinal lymph nodes of mice killed by asphyxiation
Mempel et al. 2004 [51]	$CD8^+$	no	popliteal lymph nodes of anaesthetized mice

Table 7.1: Summary of the literature where two-photon microscopy was used to track T cell movement and a graph of the mean displacement as a function of the square root of time was presented. The first column gives the reference, the second column indicates the type of T cells considered, the third column indicates whether the plot of the mean displacement as a function of the square root of time was projected to two dimensions (2d) or not, and the last column describes the experimental conditions under which the movement of T cells was recorded.

in the x and y directions that the curve of the 3d mean displacement as a function of $\sqrt{\text{time}}$ is indistinguishable within the standard error of the mean from that of the 2d mean displacement in the xy-plane (Figure 7.1).

The smaller mean displacement as a function of $\sqrt{\text{time}}$ in the z direction could be the result of the poorer resolution in the z direction: 6 μm in the z direction compared to ~ 700 nm in the x and y directions [51]. Dr. Mempel has told us that the discrepancy for the mean displacements in the z direction compared to that in the x and y directions was reduced when the resolution in z was increased to 2 μm.[1] But more investigation would be necessary to determine whether the poor resolution in z can fully account for the small mean displacement in the z direction or whether this is the result of $CD8^+$ T cells predominantly moving in the xy-plane.

We have not been able to obtain individual T cell tracks from the Miller et al. group. Probably in order to eliminate a similar discrepancy between the mean displacement in the z direction and that in the x and y directions, Miller et al. give the mean displacement for 2d rather than 3d in their publications [53, 54, 55]. In order to obtain the data points from the 2d mean displacement versus $\sqrt{\text{time}}$ graphs of the Miller et al. publications [53, 54, 55], we digitized the graphs and extracted the data points using the software Engauge Digitizer [57]. Their data can now be compared to the 2d mean displacement we computed from the Mempel et al. data. The 2d mean displacements as a function of $\sqrt{\text{time}}$ for all 4 publications are presented in Figure 7.2. One can see that the 2d mean displacement versus $\sqrt{\text{time}}$ for all 4 publications consistently follows a straight line for times larger than ~ 3 min. As we will see in the next section, this is consistent with T cells performing a random walk.

In addition, the speed of T cells in the absence of antigen was consistent across the 4 publications. Miller et al. found a 2d mean instantaneous velocity of 10.8 ± 0.1 μm/min for $CD3^+$ T cells in [55], 10.9 μm/min for $CD4^+$ T cells in [54], and 9.6 μm/min for $CD4^+$ T cells in [53], while we computed a 2d mean instantaneous velocity of 10.5 ± 0.2 μm/min for $CD8^+$ T cells from the Mempel et al. tracks used in [51].

7.3 Analyzing the Motion of T Cells

Experimental data of the mean T cell displacement versus $\sqrt{\text{time}}$ displays two distinct regimes. At small times, the displacement appears to have a quadratic dependence on $\sqrt{\text{time}}$ (a linear dependence on time), which resembles what one would expect for a particle moving in a straight line at a fixed speed. On the other hand, for larger times, the displacement appears to depend linearly on the square root of time, suggesting that the cells are diffusing. For times smaller than the time scale of a single step of a random walk, a particle performing that random walk will move in a straight line as it takes that step, yielding a quadratic shape at small times. But for larger times, the consecutive steps taken by the particle in random directions yield a linear shape. This is illustrated in Figure 7.3, and can be seen in Figure 7.2 where we present data obtained from [51, 53, 54, 55].

[1] Thorsten Mempel, Personal communication by email, 19 March 2005.

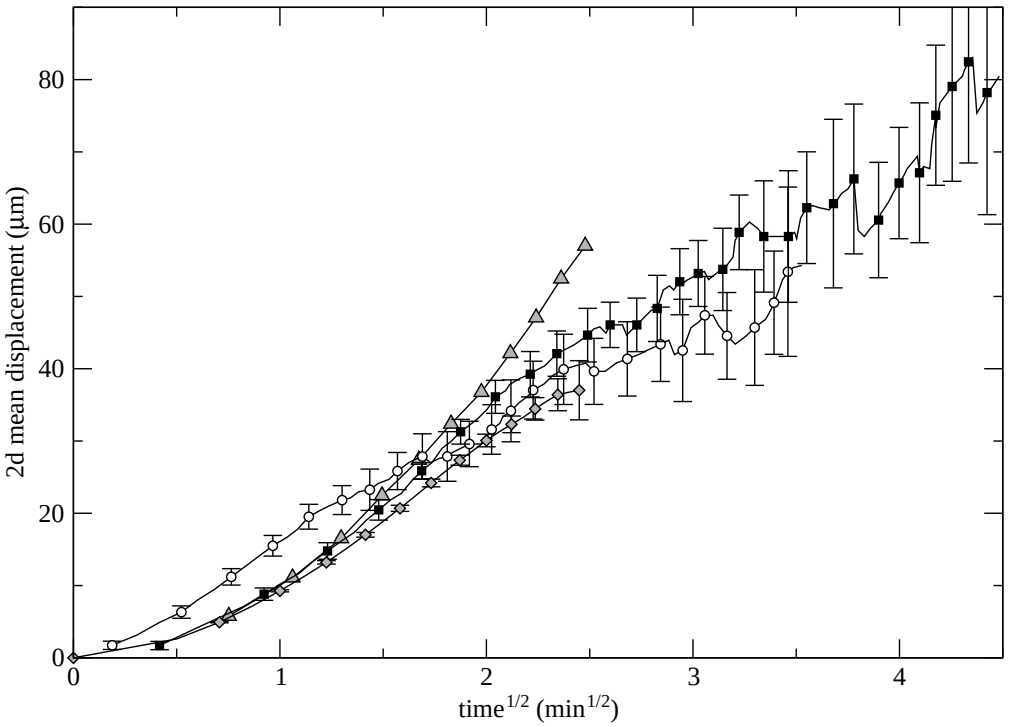


Figure 7.2: Two-dimensional mean displacement of T cells as a function of the square root of time from Miller et al. 2002 [55] (empty circles), Miller et al. 2003 [54] (filled squares), Miller et al. 2004 [53] (grey triangles), and from calculations based on individual T cell tracks used in Mempel et al. 2004 [51] (grey diamonds). Data for [53, 54, 55] was obtained by digitizing the graphs and extracting data points using the software Engauge Digitizer [57]. Standard deviations were only reported in [54, 55].

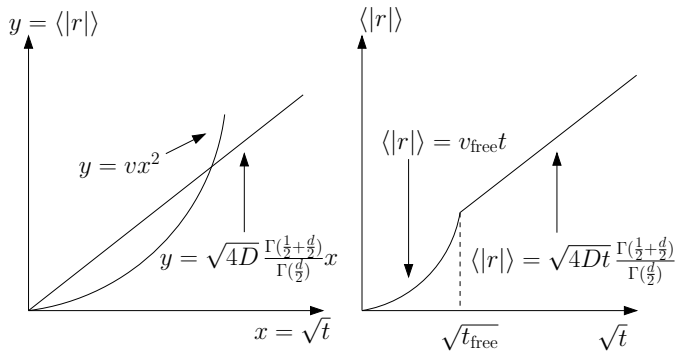


Figure 7.3: (Left) The graph of the mean displacement, $\langle |r| \rangle$, as a function of the square root of time ($\sqrt{t}$) yields a quadratic shape for motion on a straight line at fixed velocity, v, and a linear shape for a random walk or diffusion in d dimensions. (Right) The results from T cell movement experiments are quadratic for small times and linear for larger times.

The two regimes are clearly identified in all data sets except [55], where the short time data differs from the other data sets. This may be because $CD3^+$ T cells were used in [55], while the other publications considered the movement of $CD4^+$ [53, 54] and $CD8^+$ [51] T cells. The transition between the quadratic and the linear regime occurs around 3 min ($\sqrt{t} = 1.8\sqrt{\text{min}}$) at a mean displacement of 30 μm. This suggests that T cells perform a random walk with a mean free time of about 3 min during which T cells move at a 2d speed of about 10 μm/min in a straight line, yielding a 2d mean free path of 30 μm.

As illustrated in Figure 7.3, the slope in the linear regime of the mean displacement, $\langle |r| \rangle$, versus $\sqrt{\text{time}}$ curve provides an estimate of the diffusion coefficient characterizing the random walk of the T cells. Let r be the displacement of a particle at time t given that it started at the origin at time 0. If the particle is undergoing a random walk in d dimensions with diffusion coefficient D, then its mean squared displacement as a function of time is given by

$$\left\langle r^2 \right\rangle = 2dDt \ , \tag{7.1}$$

for t much larger than the time scale of a single time step [6]. This equation is also valid for particles moving in d dimensions who's motion has been projected to $\delta < d$ dimensions. In such a case, the mean squared displacement in δ dimensions is $\langle |r| \rangle = \sqrt{2\delta Dt}$, provided that motion in the x, y, and z directions are statistically independent. This is because $\left\langle r^2 \right\rangle = \left\langle x^2 + y^2 + z^2 \right\rangle = \left\langle x^2 \right\rangle + \left\langle y^2 \right\rangle + \left\langle z^2 \right\rangle$, where x, y, and z are displacements in the individual dimensions, and each term on the right hand side equals $2Dt$ as diffusion in any direction is independent of the others. Thus, we have

$$\left\langle r^2 \right\rangle_{2d} \ = \ \left\langle x^2 \right\rangle + \left\langle y^2 \right\rangle \ = \ 4Dt \ . \tag{7.2}$$

The authors of [51, 53, 54, 55] chose to present the mean displacement, $\langle |r| \rangle$, rather

than the root mean squared displacement, $\sqrt{\langle r^2 \rangle}$, as a function of $\sqrt{\text{time}}$. From the slope of their graph, the authors compute what they call a "motility coefficient", M, which is computed in a manner analogous to that of the diffusion coefficient for particles moving in 2d, namely $M = m^2/4$, where m is the slope of the linear portion of the 2d mean displacement versus $\sqrt{\text{time}}$ curve. Let us see how the motility coefficient, M, relates to the diffusion coefficient, D.

At time t, the displacement in d dimensions of a particle that started at the origin at time 0 is given by

$$|r| = \sqrt{\sum_{i=1}^{d} x_i^2} \ . \tag{7.3}$$

Let us define

$$\alpha_i = \frac{x_i}{\sqrt{2Dt}} \ , \tag{7.4}$$

such that $x_i = \sqrt{2Dt}\alpha_i$, and

$$\left\langle x_i^2 \right\rangle = 2Dt \left\langle \alpha_i^2 \right\rangle \ . \tag{7.5}$$

Since x_i is a Gaussian random variable of mean $\langle x_i \rangle = 0$ and variance $\left\langle x_i^2 \right\rangle = 2Dt$, it follows that α_i is a Gaussian random variable of mean 0 and variance 1. The mean displacement in d dimensions at time t would then be given by

$$\langle |r| \rangle = \left\langle \sqrt{\sum_{i=1}^{d} x_i^2} \right\rangle \tag{7.6}$$

$$= \left\langle \sqrt{\sum_{i=1}^{d} 2Dt\, \alpha_i^2} \right\rangle \tag{7.7}$$

$$= \sqrt{2Dt} \left\langle \sqrt{\sum_{i=1}^{d} \alpha_i^2} \right\rangle \tag{7.8}$$

$$\langle |r| \rangle = \sqrt{2Dt} \left\langle \sqrt{\beta} \right\rangle \ , \tag{7.9}$$

where $\beta = \sum_{i=1}^{d} \alpha_i^2$. The sum of the squares of d Gaussian random variables of mean 0 and variance 1 is a χ^2-distributed random variable with d degrees of freedom [81]. The $\frac{1}{2}^{\text{th}}$ moment of a χ^2-distributed random variable β with d degrees of freedom is given by [81]

$$\left\langle \sqrt{\beta} \right\rangle = \frac{2^{\frac{1}{2}}\, \Gamma\left(\frac{1}{2} + \frac{d}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \ . \tag{7.10}$$

Substituting this back into (7.9), we obtain

$$\langle |r| \rangle = \sqrt{2Dt}\, \frac{\sqrt{2}\, \Gamma\left(\frac{1}{2} + \frac{d}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \ . \tag{7.11}$$

From this, we can work out the mean displacement in one, two, and three dimensions, namely

$$\langle |r| \rangle_{d=1} = \sqrt{\tfrac{4D}{\pi}}\sqrt{t} \ , \tag{7.12}$$

$$\langle |r| \rangle_{d=2} = \sqrt{D\pi}\,\sqrt{t} \ , \tag{7.13}$$

$$\langle |r| \rangle_{d=3} = \sqrt{\tfrac{16D}{\pi}}\sqrt{t} \ . \tag{7.14}$$

Hence, the motility coefficient, $M = m^2/4$, computed from the slope, m, of 2d mean displacement versus $\sqrt{\text{time}}$ graphs is such that $D = 4M/\pi$.

From the combined data sets, we computed a motility coefficient of 77 ± 5 $\mu\text{m}^2/\text{min}$ from the slope of the 2d mean displacement versus $\sqrt{\text{time}}$ graph, for times greater than 4 min (linear portion of the graph). This yields a diffusion coefficient of 98 ± 6 $\mu\text{m}^2/\text{min}$. It is interesting to compare this value with the diffusion coefficient for a sphere of radius R diffusing as a result of thermal energy through a milieu of viscosity η at absolute temperature T. The latter diffusion coefficient is given by the Stokes-Einstein relation,

$$D = \frac{k_B T}{6\pi\eta R} \ , \tag{7.15}$$

where k_B is the Boltzmann constant [6]. For a cell diffusing in vivo, $T = 310$ K is the body temperature, and $\eta = 6.92 \times 10^{-4}\ \text{kg} \cdot \text{s}^{-1} \cdot \text{m}^{-1}$ is the viscosity of water at body temperature. Given that the radius of a T cell is approximately $R = 8$ μm [76], we get a Stokes-Einstein diffusion coefficient for T cells in vivo of 2.5 $\mu\text{m}^2/\text{min}$. The diffusion coefficient obtained from the mean displacement versus $\sqrt{\text{time}}$ plot is more than an order of magnitude larger than the Stokes-Einstein diffusion coefficient for passive thermal diffusion. This is because T cell motion is the result of active crawling on the part of T cells rather than passive motion resulting from thermal energy.

T cells are known to be capable of active motion through crawling. Although the manner in which T cells crawl is not fully understood, many characteristics are thought to be similar to the crawling of amoeba of the lower eukaryote *Dictyostelium discoideum* [26]. Migration is initiated when the T cell adopts a polarized (elongated) shape. This is followed by the formation of a lamellipodium at the leading edge and a uropod at the trailing edge [70] (see Figure 1 in [27]). This shape is referred to as a hand mirror shape or amoeboid movement configuration [69]. To move, the T cell anchors its uropod to the crawling surface, shifts all of its cytoplasm to the front, retracts the uropod, and starts another contraction-retraction cycle (see Figure 1 in [69]).

The mixed quadratic/linear nature of the graphs of the mean displacement of T cells with respect to the square root of time suggest a simple scenario for T cell movement in the absence of antigen. As mentioned above, rapid examination of the T cell mean displacement versus $\sqrt{\text{time}}$ reveals that T cells appear to perform a random walk with a mean free time of ~ 3 min, during which they move with a constant 2d velocity of ~ 10 μm/min. In the next section, we will refine this description by simulating T cell trajectories and comparing the results against experimental data.

7.4 Simulating T Cell Motion

We let a T cell perform a random walk (diffuse) with a mean free time t_{free}, during which it crawls in a straight line at a fixed speed of v_{free}. At the end of each free run, we allow for a pause time, t_{pause}, during which the T cell is immobile as it reorganizes the cellular machinery allowing it to turn. The cell then picks a new random direction, undergoes another free run, and so on. This scheme of movement is somewhat similar to the "run and tumble" motion of certain bacteria such as *E. coli* [6], except that T cells reorient their lamellipodium, rather than tumble, between each free run.

For each value of the parameter triplet (t_{pause}, t_{free}, v_{free}) we simulated the individual trajectories of 10^6 T cells and averaged their 2d displacement every second. The simulations were implemented in C. The run for each individual T cell is computed such that

$$x_{t+\Delta t} = x_t + \Delta t \cdot v \cdot \sin\theta \cdot \cos\phi \tag{7.16}$$

$$y_{t+\Delta t} = y_t + \Delta t \cdot v \cdot \sin\theta \cdot \sin\phi \ , \tag{7.17}$$

where $\phi \in [0, 2\pi]$ is an evenly distributed random variable, $\theta \in [0, \pi]$ is a random variable distributed according to the probability density function $P(\theta) = \frac{1}{2}\sin\theta$, Δt is the step size which we set to 1 s, and v is set to v_{free} for a length of time t_{free} and set to zero for a length of time t_{pause} successively. Note that the z component is omitted because we are only interested in the mean displacement in 2d of the 3d random walk because we want to compare our results with those of Figure 7.2.

At the beginning of each free run, i.e. when v is switched from zero to v_{free}, new values for θ and ϕ are picked at random. For each T cell run, the time for the first pause is chosen at random in the range $[-t_{\text{pause}}+1, t_{\text{free}}]$ to prevent artifacts which would result from the synchronization of pause times across T cell runs. Each T cell run lasts 20 min, and the T cell's squared 2d displacement from the origin at time 0 to its position at time t is computed using

$$r_t = \sqrt{x_t^2 + y_t^2} \ , \tag{7.18}$$

for t from 0 to 20 min. Again, although our algorithm simulates cells moving in 3d, we are interested in their mean displacement in 2d in order to compare our results with those presented in Figure 7.2. We repeat all of the above for each of the 10^6 T cells we wish to simulate. Finally, in order to obtain the average mean displacement, we sum the total displacement after time t for each T cell and divide it by the number of T cells simulated, namely,

$$\langle r_t^2 \rangle = \frac{r_t^{\text{cell\#1}} + r_t^{\text{cell\#2}} + ... + r_t^{\text{cell\#N}}}{N} \ , \tag{7.19}$$

where N is the number of T cell runs simulated.

The parameter values were varied over the ranges $t_{\text{pause}} = [0, 3.5]$ min, $t_{\text{free}} = [0.5, 20]$ min, and $v_{\text{free}} = [5, 50]$ μm/min. For each triplet value, we computed the sum of the squared residuals (SSR) between the experimental data and the simulation results. This was done for each of the experimental data sets of [51, 53, 54, 55] individually

and for a set comprised of all the experimental data. Table 7.2 presents the best triplet values, sorted by increasing SSR, with respect to each experimental set as well as to the set of the combined data.

From Table 7.2, we see that the triplet value of ($t_{\text{pause}} = 0.5$ min, $t_{\text{free}} = 2.0$ min, $v_{\text{free}} = 18.8$ μm/min) minimizes the SSR to the data of the combined sets. If, for simplicity, one only considers fits for which there is no pause time between free runs ($t_{\text{pause}} = 0$ s), the doublet value of ($t_{\text{free}} = 2.0$ min, $v_{\text{free}} = 16.6$ μm/min), the 6^{th} best triplet, minimizes the SSR between the simulation results and the data of the combined sets.

In order to compare the T cell velocities, v_{free}, of the best triplets to those measured during the experiments presented in [51, 53, 54, 55], v_{free} needs to be converted using

$$v_{\text{2d,instantaneous}} = v_{\text{free}} \times \underbrace{\frac{\pi}{4}}_{\text{3d}\to\text{2d}} \times \underbrace{\frac{t_{\text{free}}}{t_{\text{free}} + t_{\text{pause}}}}_{v_{\text{free}}\to v_{\text{instantaneous}}} \quad . \tag{7.20}$$

The term $\pi/4$ is necessary to convert the 3d velocity of our simulation to its 2d equivalent for comparison with the velocities mentioned above for [51, 53, 54, 55]. It comes from the fact that $v_{\text{2d}} = v_{\text{3d}} \cdot \sin\theta$, where θ is the angle from the positive z axis. On average,

$$\langle \sin\theta \rangle = \int_0^\pi P(\theta) \sin\theta \, \mathrm{d}\theta = \int_0^\pi \frac{\sin^2\theta}{2} \, \mathrm{d}\theta = \frac{\pi}{4} \, . \tag{7.21}$$

Additionally, the experimental instantaneous velocity is computed from the distance travelled by a T cell over a short time interval (~ 10 s) [51, 53, 54, 55]. Over this time interval, some T cells are paused, some T cells are moving, and some are in transition between the two states. As a result, the computed instantaneous velocity will not be the true velocity of the T cells but rather a combination of their true velocity and the amount of time they spend moving freely and pausing. The term $t_{\text{free}}/(t_{\text{free}} + t_{\text{pause}})$ accounts for this fact and converts the velocity from our simulation to the T cell's instantaneous velocity that would be measured experimentally. Using this conversion, we obtain $v_{\text{2d,instantaneous}} = 12$ μm/min for our best triplet value ($t_{\text{pause}} = 0.5$ min, $t_{\text{free}} = 2.0$ min, $v_{\text{free}} = 18.8$ μm/min), and $v_{\text{2d,instantaneous}} = 13$ μm/min for our best doublet value ($t_{\text{free}} = 2.0$ min, $v_{\text{free}} = 16.6$ μm/min). These velocities are in reasonable agreement with the experimental value of approximately 10 μm/min [51, 53, 54, 55].

The diffusion coefficients for the 6 best triplets for the combined data set vary between 92 μm^2/min and 95 μm^2/min. These diffusion coefficients are all in agreement with the diffusion coefficient computed from the slope of the combined data set, namely 98 ± 6 μm^2/min.

The discrepancies between the various best triplets for each data set and the set of the combined data are likely the result of the different types of T cells and experimental procedures being followed in each experiment. Miller et al. 2002 studied CD3^+ T cells in explanted inguinal and cervical lymph nodes of mice [55], Miller et al. 2003 studied CD4^+ T cells in mice inguinal lymph nodes [54], Miller et al. 2004 studied CD4^+ T cells

rank	t_{pause} (min)	t_{free} (min)	v_{free} (μm/min)	D (μm^2/min)	SSR (μm^2)
Miller et al. 2002 [55]					
1st	0.50	0.5	42.2	74	337.625
2nd	0.25	0.5	36.6	74	341.566
3rd	0.25	1.0	23.8	75	449.314
4th	0.50	1.0	26.1	75	452.536
5th	0.00	1.0	21.9	80	463.928
Miller et al. 2003 [54]					
1st	2.50	1.5	32.2	97	805.03
2nd	1.75	2.5	20.2	100	830.775
3rd	1.25	2.5	18.8	99	844.728
4th	1.25	2.0	21.9	98	856.171
5th	1.75	2.0	23.8	101	861.45
Miller et al. 2004 [53]					
1st	0.75	14.5	13.3	407	1.13177
2nd	0.75	14.0	13.3	393	1.1513
3rd	0.00	18.5	12.7	497	1.21151
4th	1.00	15.5	13.3	430	1.22341
5th	0.50	13.0	13.3	370	1.23408
Mempel et al. 2004 [51]					
1st	0.50	3.0	14.0	84	1.77569
2nd	1.50	2.5	17.6	81	3.19944
3rd	0.25	3.5	12.7	88	3.26771
4th	1.00	2.5	16.6	82	5.31873
5th	1.00	3.0	14.8	82	5.37939
Set of combined data [51, 53, 54, 55]					
1st	0.50	2.0	18.8	95	3387.32
2nd	0.50	2.5	16.6	95	3421.78
3rd	0.25	2.0	17.6	92	3455.46
4th	1.25	1.5	26.1	93	3488.32
5th	0.75	1.5	23.8	94	3493.76
6th	0.00	2.0	16.6	91	3502.79

Table 7.2: The best triplets for each of the experimental data sets from [51, 53, 54, 55] individually and for a set composed of the combined data. The parameters t_{pause} was varied from 0–3.5 min, t_{free} from 0.5–20 min, and v_{free} from 5–50 μm/min. The triplets (t_{pause}, t_{free}, v_{free}) have been sorted in order of increasing sum of squared residuals, SSR, between the data and the simulation results. The simulation results were produced by averaging the individual displacement of 10^6 T cells for each parameter triplet.

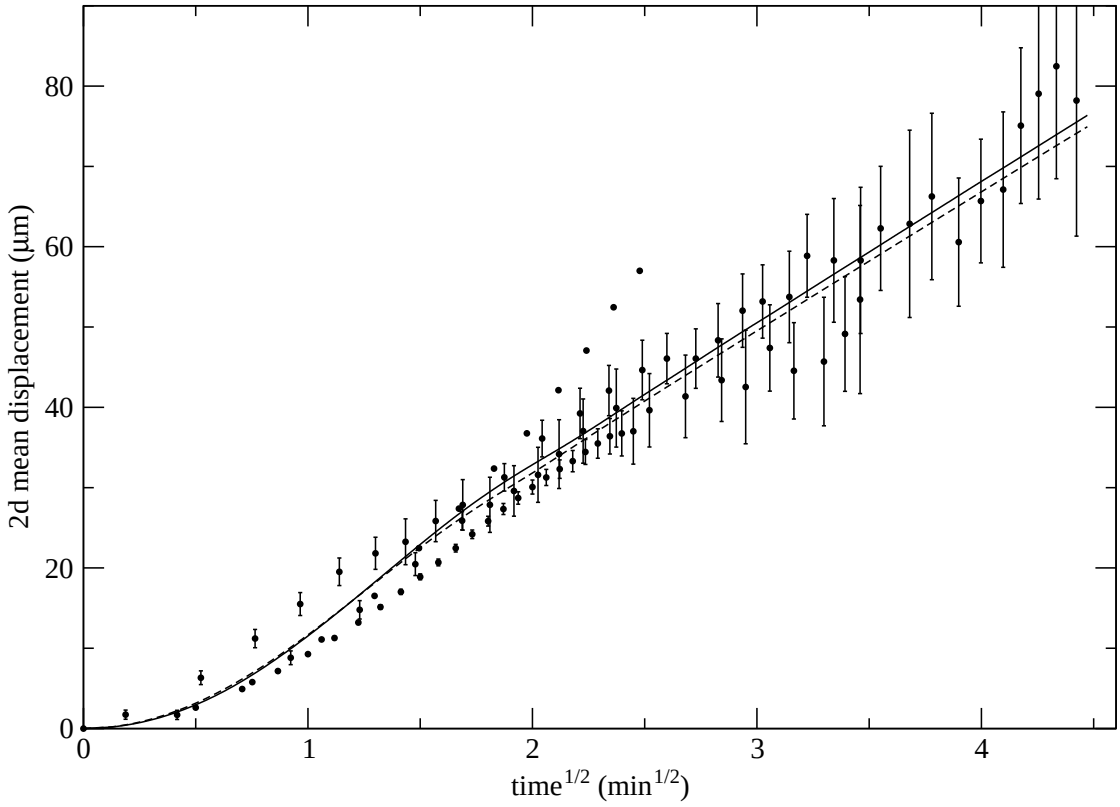


Figure 7.4: The combined experimental data from [51, 53, 54, 55] (full circles) along with the best triplet ($t_{\text{pause}} = 0.5$ min, $t_{\text{free}} = 2$ min, $v_{\text{free}} = 18.8$ μm/min) (full line), and the best doublet or 6^{th} best triplet ($t_{\text{free}} = 2.0$ min, $v_{\text{free}} = 16.6$ μm/min) (dashed line). The simulation results were produced by averaging the individual displacement of 10^6 T cells for each triplet (t_{pause}, t_{free}, v_{free}).

in inguinal and cervical lymph nodes of asphyxiated mice [53], and Mempel et al. 2004 studied $CD8^+$ T cells in popliteal lymph nodes of mice [51]. The data of [51, 53] have the smallest SSR, while that of [54, 55] give worse SSR by two orders of magnitude. This is due to the fact that the experimental data from [51, 53] is quite smooth while that of [54, 55] is noisier, and thus harder to fit.

But in the end, there really isn't much difference between the set of combined experimental data and the simulations obtained with the best doublet or triplet. Figure 7.4 shows the 2d mean displacement versus $\sqrt{\text{time}}$ for the best triplet and the best doublet against the combined experimental data. The differences between the best triplet and the best doublet (6^{th} best triplet) for the set of combined data is not significant given the spread of experimental data. Thus, given the limited amount of available experimental data, it is not possible to narrow the range of acceptable values any further nor to infirm or confirm the length or even the existence of a pause time between successive free runs.

The best triplet and doublet found here should not be regarded as the only solution, i.e. the parameter values that best characterize the motion of T cells in vivo in the absence of antigen. The sixth best triplet ($t_{\text{pause}} = 0$ min, $t_{\text{free}} = 2.0$ min, $v_{\text{free}} = 16.6$ μm/min), for example, has different parameter values from that of the best triplet, but does not have a significantly worse SSR (3502 μm^2 versus 3387 μm^2, a 3% difference). To help in understanding where the best triplets are located with respect to one another in the parameter space of the model, Figure 7.5 presents contour plots of the SSR across the parameter space of the triplets for the combined data set.

From Figure 7.5, one can see that the best triplets appear to occupy a small volume in the parameter space, represented by the white areas on the graphs. Larger quantities of more precise experimental data would be necessary to reduce these areas and constrain the uncertainty on the parameters of the model.

7.5 Conclusion

From the data obtained by two-photon microscopy, the motion of T cells appears to be consistent with the cells performing a random walk for displacements over long times and following straight trajectories over short periods of times. Consequently, we proposed a simple model for the motion of T cells in lymph nodes in the absence of antigen in which T cells move in a straight line at fixed velocity v_{free} for a time t_{free}, pause for a time t_{pause} to reposition their lamellipods and uropod as they randomly pick a new direction to move in, and so on. We have found that this simple model appears to give results which agree best with experimental results for $t_{\text{pause}} = 0.5$ min, $t_{\text{free}} = 2$ min and $v_{\text{free}} = 18.8$ μm/min.

If, for simplicity, one only considers fits for which there is no pause time between free runs ($t_{\text{pause}} = 0$ s), picking $t_{\text{free}} = 2.0$ min, and $v_{\text{free}} = 16.6$ μm/min yields best agreement between the simulation results and the experimental data. Overall, the addition of a pause time to the model does not significantly improve the agreement between the experimental data and the simulation results and as such is not absolutely necessary.

It is important to note that the pause time, which we have added to account for the physiological time necessary for a T cell to turn, should not alternatively be interpreted as the time for contact between a T cell and a dendritic cell. The pause time for our best fit was $t_{\text{pause}} = 30$ s, while dendritic cell-T cell contacts, in the absence of antigen, are consistently reported to last approximately 3 min [51, 52, 53]. Additionally, the mean free path for the best fit, $t_{\text{free}} \times v_{\text{free}} = 38$ μm, is not related to the mean time before a T cell collides with another cell in the lymph nodes because the mean distance between T cells in lymph nodes is less than 5 μm — taking the diameter of a T cell to be 8 μm and the density of T cells within lymph nodes to be 4×10^8 cells/mL [76]. Thus, we have been unable to link the values of the parameters t_{pause}, t_{free} and v_{free} to known time scales of biological processes.

Overall, the experimental data [51, 53, 54, 55] have shown that, at least in lymph nodes, T cell movement appears to be random. This is a good news for spatial mathematical modellers as this makes modelling T cell movement simple. But the experimental

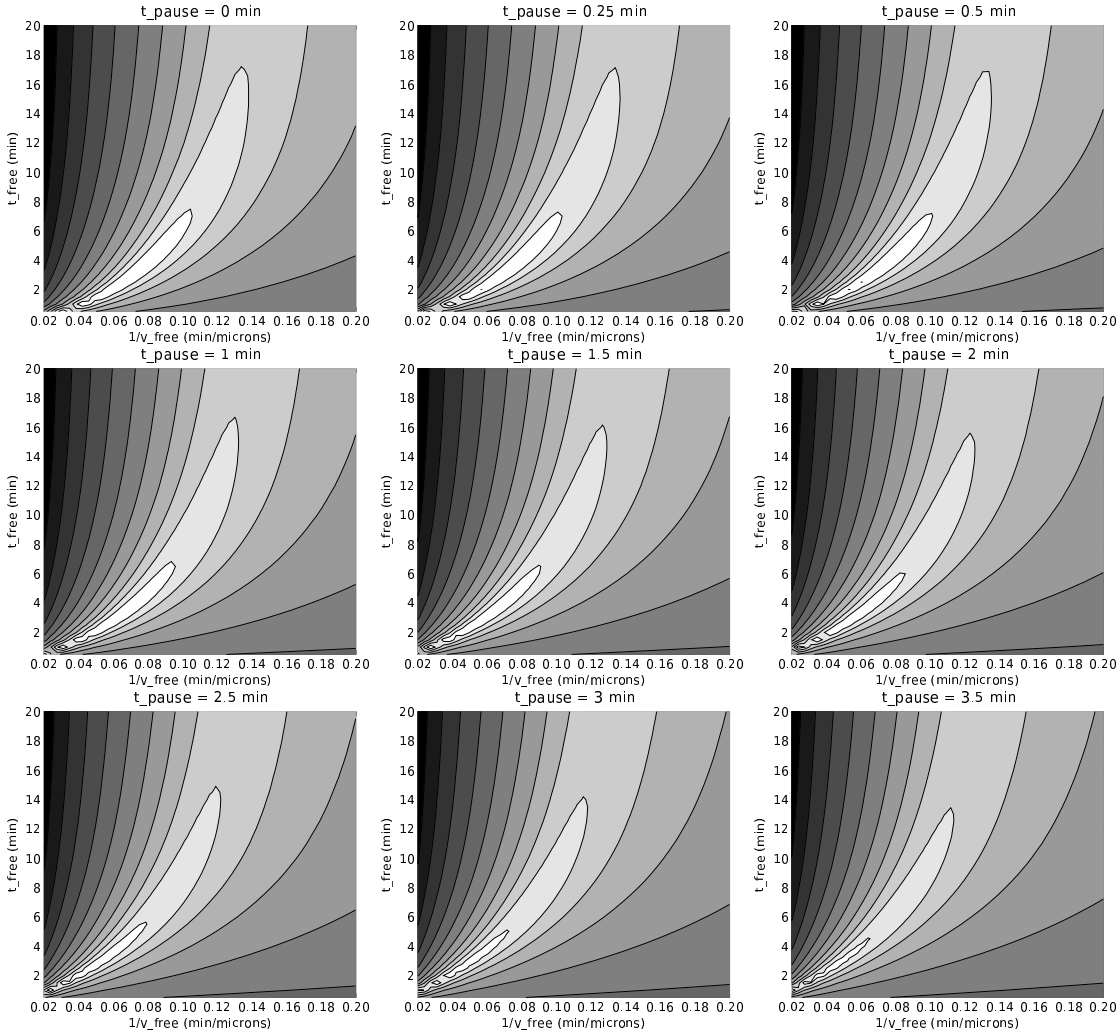


Figure 7.5: Contour plots presenting the natural logarithm of the sum of squared residuals, $\ln(\text{SSR})$, between the combined experimental data from [51, 53, 54, 55] and the simulation results as a function of $1/v_{\text{free}}$ and t_{free} for various values of t_{pause}. Lighter shades of grey represent smaller SSR.

data in [51, 53, 54, 55] does not fully exclude the possibility that chemokine attractants may direct, over short ranges, the motion of T cells. In [40], the possibility of a chemokine gradient guiding the movement of T cells over short ranges is considered and the authors explore how this could be identified through consideration not only of the mean displacement but also higher moments. However, even with higher moments, it is not possible to distinguish between cell pausing due to an encounter with an obstacle and T cell trapping due to chemokine gradients [40]. Hence, it appears as though modelling T cell movement as suggested here is sufficient for incorporating the gross movement of T cells in most spatial models.

In conclusion, we have proposed a simple description of T cell movement that matches experimental data and that can be implemented in a simple and computationally efficient manner. What is now needed is more experimental data on the movement of T cells in the presence of antigen-loaded dendritic cells in order to permit the formulation of a more complete model of T cell movement within lymph nodes, in health and disease.

Chapter 8

Conclusion

8.1 Summary

The main motivation behind the research presented here was the investigation of the effects of spatial structures on the dynamics of viral infections. Throughout this research, various aspects of spatiotemporal modelling of viral infections were covered.

Firstly, a simple two-dimensional cellular automaton was proposed for use in conducting this investigation. The model was shown to give results which compare well with experimental data when calibrated with sensible parameter values for an influenza A infection. In fact, the model proposed here performed better and with parameter values that are more biologically realistic than the model proposed in [11]. At the time of publication of this model in [4], the CA model presented here along with the ODE model presented in [11] were the only two existing mathematical models for influenza A viral infections in humans. The goal of this project was to develop a model that could be used as a test bench to investigate various theoretical aspects of the spatiotemporal spread of viral infections. In fact, the influenza A model obtained after calibration behaved impressively well when compared against available experimental data. Most likely, with the inclusion of additional details in the model, such as explicit consideration of viral particles and the addition of interferon, the CA model could be a very promising model of influenza A.

Using this CA model, I then proceeded to demonstrate specifically how the spatial distribution of agents can affect the dynamics of a viral infection. More specifically, I have looked at how the choice of the initial distribution of infected cells, the regeneration rule for epithelial cells, and the proliferation rule for immune cells, can affect the viral infection dynamics. The spatial distribution of initially infected cells was shown to have an important impact on the dynamics of the infection. This is because, in the CA model, infectious cells can only infect their immediate neighbours, and when organized in patches, fewer infectious cells have healthy neighbours. Additionally, it was found that a global epithelial cell regeneration rule, as is used in simple ODE models, allows areas of dead epithelial cells to be replenished by healthy cells even in the local absence of healthy cell. This repopulation, in turn, allows the infection to move back into the newly

replenished areas it had previously infected, resulting in a greater number of infected epithelial cells. On the other hand, a local epithelial cell regeneration rule, which requires the local presence of healthy epithelial cells, limits the growth of the infection by starving it of target cells and forces the infection to propagate as a thin circular wave. I have also shown how the choice of whether to add immune cells at random locations on the simulation grid, as is equivalent to simple ODE models, or at the site of recruitment can affect the dynamics of the infection. I found that while addition at random sites permits rapid detection of new infection sites, it makes it harder to avoid infection escape from the immune response. Consequently, random addition of immune cells was found to be a better strategy at high infection levels, while addition at the site of recruitment was the better strategy at low infection levels.

Finally, a model for T cell movement within lymph nodes was proposed. So far, spatially-dependent HIV models [76, 90] have chosen to model T cells, the target cells of HIV, as immobile agents, as discussed in Chapter 4 Section 4.2.2. But recent experiments tracking the individual movement of T cells within lymph nodes using two-photon microscopy [51, 52, 54, 55] have provided the necessary data to better characterize T cell movement. The T cell movement model which was proposed here constitutes preliminary work towards the implementation of a more realistic spatial model for HIV infections.

8.2 Future outlook

In 1996, Perelson et al. proposed for the first time the basic ODE model for viral infections (see (1.1)–(1.3)) in its most simple expression [65]. They used it along with experimental HIV-1 patients' viral titers to extract previously unknown or not well-defined dynamical parameters for HIV-1. Most notably, they obtained a viral production rate that was substantially greater than previous minimum estimates, suggesting that HIV dynamics proceeded on a much faster time scale that had been thought. For this reason, this paper got a great deal of press and sparked the interest of immunologists and virologists in mathematical models; an interest which has been growing ever since. For this reason, there is increasing pressure on mathematical modellers within the biophysics and biomathematics communities to produce more realistic models which can make reliable and useful predictions. It is now crucial that new models being developed be as faithful as possible to the true dynamics of the system.

The research presented here has demonstrated how the spatial distribution of agents involved in a viral infection can affect the dynamics of that infection. It was found that use of a spatiotemporal model compared to a well-mixed ODE model can lead to different predictions of the viral dynamics. For example, our spatiotemporal model for influenza A predicts a smaller proportion of infected cells at the peak than the simpler ODE models proposed in [1, 11]. Hopefully, the results presented here will encourage viral infection modellers to first investigate the possible effect of space on the particular viral infection they are studying before approximating its dynamics to that of a well-mixed system. The CA model proposed here could prove very useful for such a task. Its simplicity and useful parameters make it easy to calibrate it for a different viral infection.

Additionally, its modular implementation gives it great flexibility and makes it simple to add additional features. At the moment, I am working on an extension of the simple CA model for influenza A which would explicitly consider virus particles and interferon, and models their diffusion, release, and clearance. This yet more realistic and detailed model of influenza A could lead to new insights into the spatial spread and growth of the infection, and could offer new treatment perspective which would take into account the spatial diversity of the infection.

Other future directions include the implementation of a more realistic spatiotemporal model for HIV dynamics. HIV is particularly interesting from the perspective of spatiotemporal modelling as the vast majority of HIV infection occurs in densely packed lymphoid tissues and correlations between spatial location and viral strain have been observed [76]. Strain et al. have shown that at high cell density, viral propagation is limited by viral stability in the basic well-mixed model while it is limited by geometry in their spatiotemporal model. However, the Strain et al. model assumes that T cells, the target cells of HIV, are fixed in space. In collaboration with Dr. Alan S. Perelson and Dr. Stephanie Forrest, I have begun the development of an HIV CA model which would take into consideration the movement of T cells using the T cell movement model proposed above. This model will certainly provide new insights into the spatial spread and growth of HIV, and will benefit greatly from Dr. Perelson's vast experience at modelling HIV infections.

Ultimately, even though immunology has developed itself faster than most sciences, mainly due to the fact that it is a recent science and as such has benefited from more modern techniques since its foundation, it is still in its infancy. A lot is known about the immune system and viral dynamics, but even more remains to be understood. Given the nature of the pathogen-immune system dynamics, it is certain that modelling efforts will be of great help in shaping the face of modern immunology.

Bibliography

[1] P. Baccam, C. Beauchemin, C. A. Macken, F. G. Hayden, and A. S. Perelson. Kinetics of influenza A virus infection in humans. *Submitted to Journal of Virology*, August 2005.

[2] V. P. Badovinac, B. B. Porter, and J. T. Harty. Programmed contraction of $CD8^+$ T cells after infection. *Nature Immunology*, 3(7), 2002.

[3] C. Beauchemin. Probing the effects of the well-mixed assumption on viral infection dynamics. *Submitted to Journal of Theoretical Biology*, May 2005. Draft available on arXiv:q-bio.CB/0505043.

[4] C. Beauchemin, J. Samuel, and J. Tuszynski. A simple cellular automaton model for influenza A viral infections. *Journal of Theoretical Biology*, 232(2):223–234, 21 January 2005. Draft available on arXiv:q-bio.CB/0402012.

[5] G. T. Belz, D. Wodarz, G. Diaz, M. A. Nowak, and P. C. Doherty. Compromized influenza virus-specific $CD8^+$-T-cell memory in $CD4^+$-T-cell-deficient mice. *Journal of Virology*, 76(23):12388–12393, 2002.

[6] H. C. Berg. *Random Walks in Biology*. Princeton University Press, Princeton, NJ, USA, 1983.

[7] M. Bernaschi and F. Castiglione. Design and implementation of an immune system simulator. *Computers in Biology and Medicine*, 31(5):303–331, 2001.

[8] M. Bernaschi and F. Castiglione. Selection of escape mutants from immune recognition during hiv. *Immunol. Cell Biol.*, 80:307–313, 2002. Special issue devoted to Immuno-informatics.

[9] M. J. Bevan. Helping the $CD8^+$ T-cell response. *Nature Reviews Immunology*, 4:595–602, August 2004.

[10] M. Bezzi, F. Celada, S. Ruffo, and P. E. Seiden. The transition between immune and disease states in a cellular automaton model of clonal immune response. *Physica A*, 245(1–2):145–163, 15 October 1997.

[11] G. A. Bocharov and A. A. Romanyukha. Mathematical model of antiviral immune response III. Influenza A virus infection. *Journal of Theoretical Biology*, 167(4):323–360, 1994.

[12] F. Celada and P. E. Seiden. Modeling immune cognition. *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics, San Diego, CA*, pages 3787–3792, 11–14 October 1998. IEEE Catalog Number 98CH36218.

[13] F. Celada and P. E. Seiden. A computer model of cellular interactions in the immune system. *Immunology Today*, 13(2):56–62, February 1992.

[14] F. Celada and P. E. Seiden. Affinity maturation and hypermutation in a simulation of the humoral response. *European Journal of Immunology*, 26(6):1350–1358, June 1996.

[15] A. Cerwenka, T. M. Morgan, and R. W. Dutton. Naive, effector, and memory CD8 T cells in protection against pulmonary influenza virus infection: Homing properties rather than initial frequencies are crucial. *Journal of Immunology*, 163:5535–5543, 1999.

[16] D. L. Chao, M. P. Davenport, S. Forrest, and A. S. Perelson. Stochastic stage-structured modeling of the adaptive immune system. In *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB 2003)*, pages 124–131, Los Alamitos, California, 2003. IEEE Press.

[17] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, Cambridge, UK, 1998.

[18] R. G. Crystal and J. B. West. *The Lung: Scientific Foundations*. Raven Press Ltd., New York, USA, 1991.

[19] O. Diekmann, J. A. P. Heesterbeek, and J. A. J. Metz. On the definition and the computation of the basic reproduction ratio r_0 in models for infectious diseases in heterogeneous populations. *J. Math. Biol.*, 28(4):365–382, 1990.

[20] K. A. Duca, V. Lam, I. Keren, E. E. Endler, G. J. Letchworth, I. S. Novella, and J. Yin. Quantifying viral propagation *in vitro*: Towards a method for characterization of complex phenotypes. *Biotechnology Progress*, 17(6):1156–1165, December 2001.

[21] R. Durrett. The importance of being discrete (and spatial). *Theoretical Population Biology*, 46(3):363–394, December 1994.

[22] R. Durrett and S. A. Levin. Stochastic spatial models: A user's guide to ecological applications. *Philosophical Transactions of the Royal Society B — Biological Sciences*, 343(1305):329–350, 1994.

[23] J. D. Farmer, N. H. Packard, and A. S. Perelson. The immune system, adaptation, and machine learning. *Physica D*, 22:187–204, 1986.

[24] S. Forrest and S. Hofmeyr. Engineering an immune system. *Graft*, 4(5):5–9, 2001.

[25] S. Forrest, R. E. Smith, B. Javornik, and A. S. Perelson. Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, 1(3):191–211, 1993.

[26] P. Friedl, S. Borgmann, and E.-B. Bröcker. Amoeboid leukocyte crawling through extracellular matrix: lessons from the *Dictyostelium* paradigm of cell movement. *Journal of Leukocyte Biology*, 70(4):491–509, October 2001.

[27] P. Friedl, K. S. Zänker, and E.-B. Bröcker. Cell migration strategies in 3-D extracellular matrix: Differences in morphology, cell matrix interactions, and integrin function. *Microscopy Research and Technique*, 43(5):369–378, 1998.

[28] R. S. Fritz, F. G. Hayden, D. P. Calfee, L. M. R. Cass, A. W. Peng, W. G. Alvord, W. Strober, and S. E. Straus. Nasal cytokine and chemokine response in experimental influenza A virus infection: Results of a placebo-controlled trial of intravenous zanamivir treatment. *Journal of Infectious Diseases*, 180:586–593, 1999.

[29] G. A. Funk, V. A. Jansen, S. Bonhoeffer, and T. Killingback. Spatial models of virus-immune dynamics. *Journal of Theoretical Biology*, 233(2):221–236, 21 March 2005.

[30] R. A. Goldsby, T. J. Kindt, and B. A. Osborne. *Kuby Immunology* 4^{th} *Edition*. W. H. Freeman and Company, New York, 2000.

[31] M. B. Graham and T. J. Braciale. Resistance to and recovery from lethal influenza virus infection in B lymphocyte-deficient mice. *Journal of Experimental Medicine*, 186(12):2063–2068, 15 December 1997.

[32] *GTK+ version 2.0 Reference API*. available at: `http://www.gtk.org/api/`.

[33] *GTK+ version 2.0 Tutorial*. available at: `http://www.gtk.org/tutorial/`.

[34] T. J. Hagenaars, C. A. Donnelly, and N. M. Ferguson. Spatial heterogeneity and the persistence of infectious diseases. *Journal of Theoretical Biology*, 229(3):349–359, 2004.

[35] C. E. Hecht. *Statistical Thermodynamics and Kinetic Theory*. W.H. Freeman and Company, New York, 1990.

[36] R. R. Hightower, S. Forrest, and A. S. Perelson. The evolution of emergent organization in immune system gene libraries. In L. J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 344–350, San Francisco, CA, 1995. Morgan Kaufmann.

[37] R. R. Hightower, S. Forrest, and A. S. Perelson. The Baldwin effect in the immune system: Learning by somatic hypermutation. In R. K. Belew and M. Mitchell, editors, *Adaptive Individuals in Evolving Populations*, pages 159–167. Addison-Wesley, Reading, MA, 1996.

[38] J. Jameson, J. Cruz, M. Terajima, and F. A. Ennis. Human cd8$^+$ and cd4^4 T lymocyte memory to influenza A viruses of swine and avian species. *Journal of Immunology*, 162:7578–7583, 1999.

[39] N. K. Jerne. Towards a network theory of the immune system. *Annales d'Immunologie (Institut Pasteur)*, 125 C:373–389, 1974.

[40] B. Jin, S. Raychaudhuri, R. N. Germain, M. L. Dustin, and A. K. Chakraborty. Analysis of migratory patterns and evaluation of search strategies of T lymphocytes hunting for antigen. FIXME: In preparation.

[41] S. M. Kaech and R. Ahmed. Memory CD8$^+$ T cell differentiation: Initial antigen encounter triggers a developmental program in naïve cells. *Nature Immunology*, 2(5):69–77, May 2001.

[42] E. D. Kilbourne. *Influenza*. Plenum Medical Book Company, New York, 1987.

[43] S. H. Kleinstein and P. E. Seiden. Simulating the immune system. *Computing in Science & Engineering*, pages 69–77, July/August 2000.

[44] B. Kohler, R. Puzone, P. E. Seiden, and F. Celada. A systematic approach to vaccine complexity using an automaton model of the cellular and humoral immune system. i. viral characteristics and polarized responses. *Vaccine*, 19:862–876, 2000.

[45] V. Lam, K. A. Duca, and J. Yin. Arrested spread of vesicular stomatitis virus infections in vitro depends on interferon-mediated antiviral activity. *Biotechnology and Bioengineering*, 90(7):793–804, 30 June 2005.

[46] R. H. Landau and M. J. Pàez. *Computational Physics: Problem Solving with Computers*. Wiley-Interscience, New York, 1997.

[47] A. L. Lloyd and R. M. May. Spatial heterogeneity in epidemic models. *Journal of Theoretical Biology*, 179(1):1–11, 1996.

[48] Y. Louzoun, S. Solomon, H. Atlan, and I. R. Cohen. Modeling complexity in biology. *Physica A*, 297:242–252, 2001.

[49] D. Masopust, S. M. Kaech, E. J. Wherry, and R. Ahmed. The role of programming in memory T-cell development. *Current Opinion in Immunology*, 16:217–225, 2004.

[50] D. Maugeri, P. Panebianco, V. Cilmi, A. Bruno, G. Receputo, A. Salerno, M. Motta, M. G. Napolitano, and M. S. Russo. Alterations of blood and plasma viscosity and erythrocyte filtration in senile osteoporosis. *Archives of Gerontology and Geriatrics*, 19:75–82, 1994.

[51] T. R. Mempel, S. E. Henrickson, and U. H. von Adrian. T-cell priming by dendritic cells in lymph nodes occurs in three distinct phases. *Nature*, 427(6970):154–159, 8 January 2004.

[52] M. J. Miller, A. S. Hejazi, S. H. Wei, M. D. Cahalan, and I. Parker. T cell repertoire scanning is promoted by dynamic dendritic cell behavior and random T cell motility in the lymph node. *Proceedings of the National Academy of Sciences of the United States of America*, 101(4):998–1003, 27 January 2004.

[53] M. J. Miller, O. Safrina, I. Parker, and M. D. Cahalan. Imaging the single cell dynamics of $CD4^+$ T cell activation by dendritic cells in lymph nodes. *Journal of Experimental Medicine*, 200(7):847–856, 2004.

[54] M. J. Miller, S. H. Wei, M. D. Cahalan, and I. Parker. Autonomous T cell trafficking examined *in vivo* with intravital two-photon microscopy. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5):2604–2609, 2003.

[55] M. J. Miller, S. H. Wei, I. Parker, and M. D. Cahalan. Two-photon imaging of lymphocyte motility and antigen response in intact lymph node. *Science*, 296(5574):1869–1873, 7 June 2002.

[56] D. M. Mitchell, A. J. McMichael, and J. R. Lamb. The immunology of influenza. *British Medical Bulletin*, 41(1):80–85, 1985.

[57] M. Mitchell. *Engauge Digitizer*. A free open-source software to extract data points from a graph image. Hosted on SourceForge at: `http://digitizer.sourceforge.net`.

[58] R. Mora, E. Rodriguez-Boulan, P. Palese, and A. García-Sastre. Apical budding of a recombinant influenza A virus expressing a hemagglutinin protein with a basolateral localization signal. *Journal of Virology*, 76(7):3544–3553, April 2002.

[59] D. Moskophidis and D. Kioussis. Contribution of virus-specific $CD8^+$ cytotoxic T cells to virus clearance or pathologic manifestations of influenza virus infection in a T cell receptor transgenic mouse model. *Journal of Experimental Medicine*, 188(2):223–232, 20 July 1998.

[60] H. H. Nguyen, F. W. van Ginkel, H. L. Vu, M. J. Novak, J. R. McGhee, and J. Mestecky. Gamma interferon is not required for mucosal cytotoxic T-lymphocyte responses or heterosubtypic immunity to influenza A virus infection in mice. *Journal of Virology*, 74(12):5495–5501, 2000.

[61] J. S. Patton. Mechanisms of macromolecule absorption by the lungs. *Advanced Drug Delivery Reviews*, 19:3–36, 1996.

[62] J. K. Percus, O. E. Percus, and A. S. Perelson. Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination. *Proceedings of the National Academy of Sciences of the United States of America*, 90:1691–1695, 1993.

[63] A. S. Perelson. Modelling viral and immune system dynamics. *Nature Reviews Immunology*, 2(1):28–36, 2002.

[64] A. S. Perelson, P. Essunger, Y. Cao, M. Vesanen, A. Hurley, K. Saksela, M. Markowitz, and D. D. Ho. Decay characteristics of HIV-1 infected compartments during combination therapy. *Nature*, 387(6629):123–124, May 1997.

[65] A. S. Perelson, A. Neumann, M. Markowitz, J. Leonard, and D. Ho. HIV-1 dynamics *in vivo*: virion clearance rate, infected cell life-span, and viral generation time. *Science*, 271:1582–1586, 1996.

[66] A. S. Perelson and G. Weisbuch. Immunology for physicists. *Reviews of Modern Physics*, 69(4):1219–1267, Oct. 1997.

[67] C. Q. Piao, Y. L. Zhao, L. Liu, and T. K. Hei. Immortalization of primary human bronchial epithelial cells by overexpression of human telomerase catalytic subunits (htert). Technical report, Center for Radiological Research, 2001. Available online at `http://cpmcnet.columbia.edu/dept/radoncology/crr/reports2001/c4.htm`.

[68] C. W. Potter. Influenza. In A. J. Zuckerman, J. E. Banatvala, J. R. Pattison, P. D. Griffiths, and B. D. Schoub, editors, *Principles and Practice of Clinical Virology*, chapter 5, pages 271–297. John Wiley & Sons, Ltd., 5th edition, 2004.

[69] L. Rydgren, B. Norberg, C. Håkansson, C. von Mecklenburg, and N. Söderström. Lymphocyte locomotion. I. the initiation, velocity, pattern, and path of locomotion *in vitro*. *Lymphology*, 9(3):89–96, September 1976.

[70] F. Sànchez-Madrid and M. A. del Pozo. Leukocyte polarization in cell migration and immune interactions. *EMBO Journal*, 18(3):501–511, 1999.

[71] P. E. Seiden and F. Celada. A model for simulating cognate recognition and response in the immune system. *Journal of Theoretical Biology*, 158(3):329–357, October 1992.

[72] D. J. Smith, S. Forrest, D. H. Ackley, and A. S. Perelson. Variable efficacy of repeated annual influenza vaccination. *Proceedings of the National Academy of Sciences of the United States of America*, 96:14001–14006, 1999.

[73] D. J. Smith, S. Forrest, and A. S. Perelson. Immunological memory is associative. In *Workshop Notes, Workshop 4: Immunity Based Systems*, pages 62–70, Kyoto, Japan, 1996. Intnl. Conf. on Multiagent Systems.

[74] D. J. Smith, A. S. Lapedes, S. Forrest, J. C. deJong, A. D. M. E. Osterhaus, R. A. M. Fouchier, N. J. Cox, and A. S. Perelson. Modeling the effects of updating the influenza vaccine on the efficacy of repeated vaccination. In A. D. M. E. Osterhaus, N. J. Cox, and A. W. Hampson, editors, *Options for the Control of Influenza Virus IV*, volume 1219 of *International Congress Series*, pages 655–660, Amsterdam, 2001. Excerpta Medica (Elsevier). Proceedings of the World Congress on Options for the Control of Influenza Virus IV, Crete, Greece, 23–28 September 2000.

[75] J. J. Stewart, H. Agosto, S. Litwin, J. D. Welsh, M. Shlomchik, M. Weigert, and P. E. Seiden. A solution to the rheumatoid factor paradox: Pathologic rheumatoid factors can be tolerized by competition with natural rheumatoid factors. *J. Immunol.*, 159(4):1728–1738, 1997.

[76] M. C. Strain, D. D. Richman, J. K. Wong, and H. Levine. Spatiotemporal dynamics of HIV propagation. *Journal of Theoretical Biology*, 218(1):85–96, 7 September 2002.

[77] S. H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Perseus Books, Cambridge, MA, 1994.

[78] M. J. B. van Stipdonk, E. E. Lemmens, and S. P. Schoenberger. Naïve CTLs require a single brief period of antigenic stimulation for clonal expansion and differentiation. *Nature Immunology*, 2(5):423–429, 2001.

[79] B. Voorhees and C. Beauchemin. Point mutations and transitions between cellular automata attractor basins. *Complex Systems*, 15(1), 2004. Draft available on arXiv:nlin.CG/0306033.

[80] J. Watmough and L. Edelstein-Keshet. Modelling the formation of trail networks by foraging ants. *Journal of Theoretical Biology*, 176:357–371, 1995.

[81] E. W. Weisstein. Chi-squared distribution. From MathWorld — A Wolfram Web Resource. `http://mathworld.wolfram.com/Chi-SquaredDistribution.html`.

[82] J. Westermann and R. Pabst. Distribution of lymphocyte subsets and natural killer cells in the human body. *Clinical Investigator*, 70(7):539–544, 1992.

[83] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55, 1983. available at: `http://www.stephenwolfram.com/publications/articles/ca/83-statistical/`.

[84] S. Wolfram. Cellular automata as models of complexity. *Nature*, 311, 1984. available at: `http://www.stephenwolfram.com/publications/articles/ca/84-cellular/`.

[85] S. Wolfram. Cellular automata. *Los Alamos Science*, 9:2–21, Fall 1983. available at: `http://www.stephenwolfram.com/publications/articles/ca/83-cellular/`.

[86] World Health Organization. Avian influenza. Fact sheet, Department of Communicable Disease Surveillance and Response (CSR), World Health Organization, Revised January 2004. Available online at: `http://www.who.int/csr/don/2004_01_15/en/`.

[87] World Health Organization. Influenza. Fact Sheet 277, World Health Organization, Revised March 2003. Available online at: `http://www.who.int/mediacentre/factsheets/2003/fs211/en/`.

[88] W. R. Young, A. J. Roberts, and G. Stuhne. Reproductive pair correlations and the clustering of organisms. *Nature*, 412:328–331, 19 July 2001.

[89] R. M. Zorzenon dos Santos and A. T. Bernardes. Immunization and aging: A learning process in the immune network. *Physical Review Letters*, 81(14):3034–3037, 5 October 1998.

[90] R. M. Zorzenon dos Santos and S. Coutinho. Dynamics of HIV infection: A cellular automata approach. *Physical Review Letters*, 87(16), 15 October 2001.

Appendix A

MASyV: A Multi-agent System Visualization Program

A.1 Introduction

The goal of this project was to implement a platform that would allow one to study, through visualization, a wide array of cellular automaton models of multi-agent self-organizing systems. To this end, I developed a package called MASyV which consists of a graphical user interface (GUI) accompanied by a message passing library. This framework allows the user to write a simple cellular automaton client program in C, create the desired accompanying images for the agents with a paint program of his/her choice (e.g. GIMP), and get their simulation (client) to talk to the GUI (server) through a Unix socket easily by using the message passing library. The flexible GUI of MASyV is not only capable of data logging and visualization, but also offers the user the possibility to record the simulations to a wide range of compressed video formats, through the use of `transcode`, for maximum portability in the sharing of results.

Here, I will describe in details the implementation of MASyV. Firstly, in Section A.2, I will present the structure of the GUI and describe its usage. Then, in Section A.3, I will illustrate how to use the server programs, `masyv` and `logmasyv` to run a client. Finally, in Section A.4, I will present the architecture of the message passing library and illustrate its usage with a simple "Hello World" client example.

A.2 MASyV's Graphical User Interface

The GUI of MASyV is built using GTK+ widgets and functions. For better graphics performance, the display screen widget, which displays the client simulation, makes use of GtkGLExt's OpenGL extension which provides an additional application programming interface (API) enabling GTK+ widgets to render three-dimensional scenes using OpenGL. For more information about GTK+, its libraries and its widgets, please consult [33, 32] below. Here, the discussion will be limited to describing the overall capabilities of MASyV's GUI rather than the GTK+ structures from which it was built. Figure A.1

shows a series of screenshots of MASyV's GUI.

MASyV's GUI is mainly composed of a notebook widget containing three tabs. The first tab, "Simulation", displays the visual representation of the client simulation. The buttons, "Pause", "Advance", and "Play", can respectively be used to pause, advance by one time step, or by several sequential time steps, the client simulation. The "Screenshot" button captures the display screen to a PNG file and brings up a file selection dialog which allows the user to chose the name and location of the file to be saved. The "Save animation" check box is only enabled when all the correct information has been entered in the capture tab. Checking the enabled check box will initiate a call to `transcode`, which will begin encoding the simulation to a compressed movie file using the codec specified by the user in the "Capture" tab. Finally, a "Step #" label indicates the current time step of the client simulation being displayed on the display screen.

In the second tab, "Statistics", MASyV posts the statistics sent by the client simulation. The "Log to file" check button, the text entry box and the "Browse..." button enables the user to save the statistics from the simulation to a file of his/her choice. When the check button is enabled and checked, the file whose name has been entered in the text entry box is opened and the statistics start being written to the file. When the check button is unchecked, the statistics stop being written and the file is closed. Clicking again on the check button will re-open the file and append the new statistics as they arrive.

Finally, the third tab, "Capture", offers the user the possibility to control many aspects of the capture of the simulation to a compressed movie file. At the moment, MASyV only supports "raw" capture and capture using the "xvid" codec. Capturing the content of the display screen to a compressed movie file is made through a call to `transcode`, which provides a list of utilities to transcode various video, audio, and container formats.

A.3 Running MASyV

Both server programs, `masyv` and `logmasyv`, can be called from the command-line. The command-line options available for `masyv` are as follows.

```
MASyV v0.7 - A Multi-Agent System Visualization Program.
   Copyright (C) 2002 Catherine Beauchemin

The accepted command line options for MASyV are:
-s : simulation to be run
-f : output file for the statistics
-i : time increments
-t : total time (-1 is infinity)
-r : rescaling factor for simulation area
-x : clipping factor for simulation area
-c : client parameters (when supported by client)
-h : displays options and exits
```

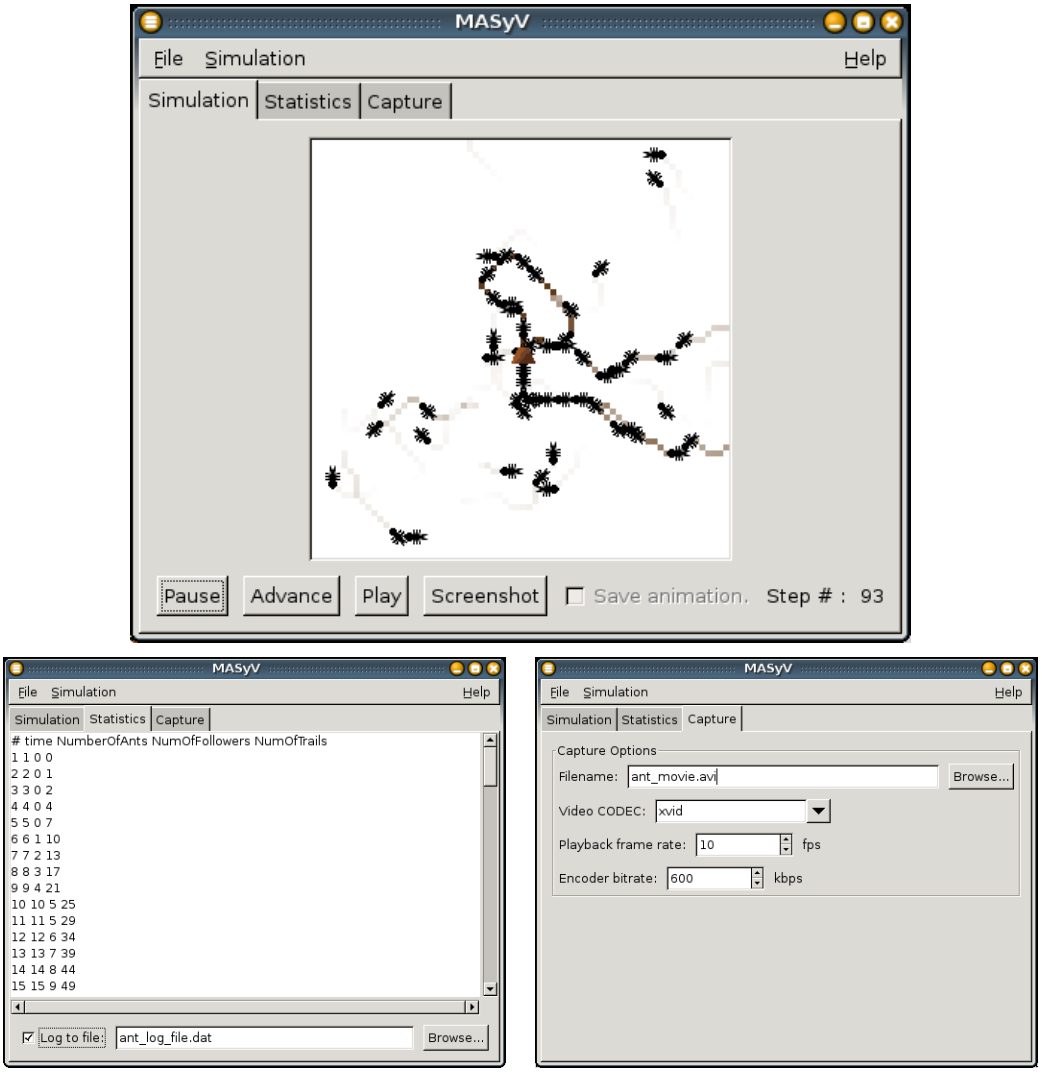


Figure A.1: MASyV's GUI with its Simulation tab (top), Statistics tab (middle), and Capture tab (bottom). The Simulation tab presents MASyV's GUI running the client module `ma_sqr_ant` which models ant depositing and following pheromone trails.

The options are self-explanatory except perhaps for the `-c` option which will be discussed in the next section. The command-line options for `logmasyv` are the same but exclude the `-f`, `-r`, and `-x` options. The `-f` option is not needed by `logmasyv` since the program's only output, the statistics from the client, is sent to the standard output stream, `stdout`, which can be redirected to any file. Let's illustrate `masyv`'s command-line usage by an example.

```
prompt> masyv -s ma_client -f stats.dat -t 200
```

This command would start up `masyv`, run the client `ma_client` for 200 steps and log the statistics to the file `stats.dat`. Note that `masyv` will not log the statistics to `stats.dat` until the "Log to file" check box on the "Statistics" tab has been checked. The command to run `logmasyv` with the same options is

```
prompt> logmasyv -s ma_client -t 200 > stats.dat
```

A.4 Usage of the Message Passing Library

Since the goal of this project was to develop a platform capable of representing any two-dimensional cellular automaton, special thought had to be put into designing the GUI so that it remains generic. The message passing library was implemented to enforce standards in the way the client simulations should communicate with the GUI server, but also to save one the trouble of having to duplicate and maintain the code to speak with the GUI (server) in each client simulation.

The software employs a client-server model with the server providing I/O and supervisory services to the client simulation. This means that all the components of the MASyV package are separate stand-alone programs or modules. At the moment, those modules consist of the GUI server, `masyv`, an additional server program, `logmasyv`, which allows the user to run the client without a GUI interface, and several optional cellular automaton clients which I developed over time for my research. To standardize the communication between client and server, a library for message passing was developed which consists of `ma_message.c`, the actual message passing functions, and `ma_message.h`, which contains the prototypes of the message passing functions and defines a few macros for use by the clients. The client communicates with the GUI through a Unix domain socket stream using the message types provided by the message passing library.

The message passing library defines the following message types to be used by the client:

- `create_client_socket_file(char *program_name, char *address );`
- `ma_send_pixtablespec(int socket_fd, float grid_unit_cell[2][2], int num_layers, int *num_images);`
- `ma_send_arena_size(int socket_fd, int width, int height);`
- `ma_send_texels(int socket_fd, int layer_num, int image_num, int width, int height, const unsigned char *pixels);`

- `ma_send_start_step(int socket_fd, int total_time, int clear);`
- `ma_send_stop_sequence(int socket_fd);`
- `ma_send_agent(int socket_fd, int x, int y, int layer, int image, int opacity);`
- `ma_send_stats(int socket_fd, char *stats);`
- `ma_send_terminate_signal(int socket_fd);`
- `ma_receive_msg(int client_socket_fd, struct ma_receive_ops_t *ops, void *data);`

All but `create_client_socket_file` and `ma_receive_msg` are messages that are typically sent by the client to the GUI server. Messages `ma_send_pixtablespec`, `ma_send_arena_size`, and `ma_send_texels` are used by the client at initialization time to prepare the GUI, while `ma_send_start_step`, `ma_send_stop_sequence`, `ma_send_agent`, and `ma_send_stats` are messages sent by the client during the simulation to tell the GUI how to update the simulation grid and the statistics produced by the client. Finally, the message `ma_send_terminate_signal` can optionally be used by the client to signal to the server that the simulation has terminated.

To demonstrate usage of the message passing library by the client, let us look at the "Hello World" example of a client simulation, `ma_hello`, which consists of a single grey square moving around randomly on a square grid.

```
#include <unistd.h>
#include <stdio.h>
#include "ma_message.h"
#include "particle.c"   /* The image of the square */

#define PROGRAM_NAME "ma_hello"
#define GRID_SPACING 4
#define NUM_LAYERS 1    /* Number of layers */
#define LAYER 0         /* The name of the only layer */
#define NUM_IMAGES 1    /* Number of images in LAYER */

char *ui_socket_address = NULL;
struct ma_receive_ops_t msg_ops;
int client_socket_fd;
int grid_width = 10;
int grid_height = 10;
int total_time = 0;

void do_time_advance( int steps, void *data )
{
```

```
   int pos;

   ma_send_start_step( client_socket_fd, total_time, CLEAR_GRID );
   while( steps-- ) {
      pos = random_a_b(0,grid_width*grid_height-1);
      ma_send_agent( client_socket_fd, pos%grid_width, pos/grid_width,
         LAYER, 1, OPAQUE );
      total_time++;
   }
   ma_send_stop_sequence( client_socket_fd );
}

void configure( void *data )
{
   int num_images[] = { NUM_IMAGES+1 };
   float grid_unit_cell[2][2] = SQUARE_UNIT_CELL( GRID_SPACING );

   ma_send_pixtablespec( client_socket_fd, grid_unit_cell, NUM_LAYERS,
      num_images );
   ma_send_arena_size( client_socket_fd, grid_width, grid_height );
   ma_send_texels( client_socket_fd, LAYER, 1, particle.width,
      particle.height, particle.pixel_data );
   ma_send_stop_sequence( client_socket_fd );
}

void receive_terminate_signal( void *data ) {  exit( 0 );  }

int main( int argc, char *argv[] ) {
   msg_ops = (struct ma_receive_ops_t) {
      ma_receive_req_config: configure,
      ma_receive_req_time_advance: do_time_advance,
      ma_receive_terminate_signal: receive_terminate_signal
   };
   int key;
   char tmp_msg[200];

   while( (key = getopt(argc, argv, "s:")) != EOF )
      switch(key) {
         case 's':
            ui_socket_address = optarg;
         break;
         default:
            exit(0);
         break;
```

```
      }

   if( ui_socket_address == NULL ) {
      sprintf( tmp_msg, "%s: Problems receiving socket address " \
         "from server.\n", PROGRAM_NAME );
      perror( tmp_msg );
      exit(1);
   }

   client_socket_fd = create_client_socket_file( PROGRAM_NAME,
      ui_socket_address );

   while(1)
      ma_receive_msg( client_socket_fd, &msg_ops, NULL );

   exit(0);
}
```

First, let us look at the `main` function. Because it is the server, either `masyv` or `logmasyv`, which runs the client, client options have to be given to the server who passes them on to the client. Two options are typically passed to the client by the server: one is optional, the other is not. The latter, the `-s` option, is used by the server to pass its socket address, `ui_socket_address`, as an argument to the client so that the client may connect to it by creating a socket using the `ma_message` library function `create_client_socket_file`. This `-s` option should not be confused with the server's `-s` option which is used to pass the name of the client simulation to be run. The other option that is typically passed to the client by the server is the optional '`-c`' option. It takes a single argument and can be used to set parameter values for the client. The single argument can be provided in any form since it is up to the client to parse it to extract the parameter values. Let us illustrate usage of the '`-c` option with a simple example where the '`-c`' option is used to set the client `ma_client` grid's width and height. The command would look like

```
prompt> masyv -s ma_client -c 110:70
```

and to parse the argument, the following code could be added to the client's `switch` statement.

```
      switch(key) {
         ...
         case 'c':
            sscanf(optarg, "%d:%n", &grid_width, &count);
            sscanf(optarg += count, "%d", &grid_height);
         break;
         ...
      }
```

After initialization, the client puts itself in a perpetual receiving mode with the command `ma_receive_msg`. The `msg_ops` structure defines which client function will handle each of the server's request messages. The `ma_receive_req_config` message is sent once by the server, as the first message, to requests the configuration information from the client. The configuration messages sent by the client as a response to the server's request consist in transmitting: the specification of the structure to store the client's images (`ma_pixtablespec`), the dimensions of the grid (`ma_send_arena_size`), and each image to be used by the client (with repeated calls to `ma_send_texels`). Then, the client will send a `ma_send_stop_sequence` message to the server to signal that all the configuration data has been sent.

Then, the server will begin requesting steps. The client will send the current time step and tell the server whether to clear the grid from the previous step or not using `ma_send_start_step`, it will send each agent by specifying which image to use and where on the grid it should appear using `ma_send_agent`, will send the statistics for that time step using `ma_send_stats`, and will indicate to the server that all the data for this step has been sent with a call to `ma_send_stop_sequence`.

Finally, the server can terminate the client when it itself receives a terminate signal from the user, for example, by sending a terminate signal to the client. The client should know how to handle such a signal. In the above example, the `ma_hello` client's function `receive_terminate_signal` handled the terminate signal by a simple call to `exit(0)`.

A.5 Conclusion

On April 21, 2003, MASyV was registered on SourceForge, a popular open-source software repository, where it is still hosted. I continue to maintain MASyV and release new versions approximately every year. At the time of writing, MASyV had 95 downloads worldwide. It always generates a great deal of interest at conferences where it is presented.

Currently, the MASyV package comes with several client simulations which I have developed as part of the research I did during my PhD. The available clients are:

`ma_sqr_ants` An ant simulation based on an article by Watmough and Edelstein-Keshet [80]. It illustrates the early formation of trail networks by a colony of pheromone laying ants;

`ma_hex_ants` Implementation on an hexagonal grid of `ma_sqr_ants`;

`ma_ca` A simple one-dimensional cellular automaton as studied extensively by Wolfram. A great example of a simple simulation from which to learn how to use the message passing library to talk to the GUI;

`ma_gas_diff` A lattice gas simulation of gas diffusion. Literally: little grey squares moving randomly on a grid with absorbing boundary conditions. Also a great example of a simple simulation from which to learn how to use the message passing library to talk to the GUI;

`ma_grid_hiv` An implementation of a spatiotemporal HIV model by Maria Zorzenon dos Santos presented in [90];

`ma_ising` The one-dimensional Ising spin model;

`ma_tubule` A simulation of the organization of electrons within the β-tubulin that make up a microtubule. This module will be used to assess whether quantum effects can cause the electrons to self-organize; and

`ma_immune` A simulation of a viral infection taking place in an organ where the target cells are tightly packed and do not move such that each sit of the CA grid represents an organ cell. Immune cells then move randomly on the grid, patrolling the tissue, and take action when they encounter an expressing or infectious cell. This particular model was used in [4] to model a viral infection with influenza A. As such, the model as it is has been calibrated for the specific case of influenza A and as such each site of the grid represents a lung epithelial cell.

This last client module, `ma_immune`, is discussed in great length in Chapter 5.