CA and Monte Carlo Models of HIV Infection

Rachel Mannion,BSc.

Submitted in fulfilment of the requirements for the degree of Master of Science in Computer Applications

Dublin City University
Dr. H.J. Ruskin (supervisor)
Faculty of Computing and Mathematical Sciences
December 2000

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master Of Science in Computer Applications is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Rachel Manion Id No. 98970461

Date: 1/2/01

Acknowledgements

I would like to thank my supervisor Dr. Heather Ruskin for all her time, advice and especially for her patience. I would also like to thank, Prof. Ras Pandey for all his help, input and his hospitality, and Dr. Alistair Sutherland for his comments and criticism.

Thanks also to all the postgrads who have helped me through-out my studies, especially Owen, Jer, Tony and Tom and my bay buddies Noel and Darragh. Also, my parents and my sisters Eunice and Eleanor, for their encouragement and support. All my friends for their continuing patience especially, Catherine, Sharon and Jean.

Abstract

The models presented are discrete Monte Carlo(MC) and Cellular Automata(CA) representations of the interaction of HIV with the immune system. HIV is characterised by the depletion of Helper T cells in the body. Helper T cells are essential to the correct regulation of the immune system. Their degradation leaves the body incapable of defending itself, even against what is usually an unharmful infection. The models consider just four cell types the Macrophage, M, the helper T cell, H , the cytotoxic Killer cell, C and the virus, V. Each cell type can either be in high concentration (1) or low concentration (0). An update of a site consists of nearest-neighbour interaction followed by intra-site interactions. The nearest-neighbour interaction represents the influence of a site's surroundings on it. The intra-site interactions are Boolean equations which represent a succinct interpretation of HIV infection and its effect on the host immune system. Mutation is considered via a probabilistic parameter P_{mut}. Each cell type has inherent mobility due to the nearest-neighbour interactions, explicit mobility is explored by a probabilistic parameter P_{mob}. The MC and CA simulations differ in their updating, with CA updating is synchronous and with MC it is asynchronous. MC is explored as an alternative to the CA model form. Due to the Boolean concentrations of the cell types, synchronous (CA) updating leads to overshooting, there is either complete viral dominance or immune dominance and no intermediate state. Asynchronous (MC) updating smoothes these extremes; intermediate states between immuno-dominance and immuno-deficiency exist. These intermediate states offer new insight into the dynamics of HIV and the immune system. Asynchronous updating gives clearly defined growth patterns and this enables the exploration of critical points. One such critical point is the value of P_{mut} for which the cross-over between immune dominance and deficiency occurs. Also characteristics of the disease progression such as latency can be investigated.

Contents

1 Introduction

The research presented here investigates a number of discrete models of Human Immuno-Deficiency Syndrome (HIV) infection. HIV is a virus which leads to a condition called AIDS -acquired immuno-deficency syndrome. The models are built with the intention to simulate in part some real-life behaviour of the disease. This can be achieved by focusing on the behaviour and interactions of the infection and encorporating this information into a model. If the model does indeed emulate some specific real-life behaviour we can then use the model to gain insight into the mechanisms and behaviour of HIV/AIDS. The assumptions underlying each model are also investigated, because although a model may mimic known behaviour of the infection, it is useless without sound assumptions.

Each of the models we discuss are formulated with Boolean algebra and a computer simulation of the model is then performed. The purpose of the computer simulation is to enable us to perform computer experiments on the model. Since the advent of computer simulations, the line between theory and experiment has become more blurred. Yes, these models are theoretical, because the system is not dealt with directly but rather approximated by a series of mathematical equations and postulates. On the other hand, while there is not a Bunsen burner in sight, these models are also experimental since valid experiments can be performed using a computer simulation. Parameters in a simulation like parameters in traditional experimentation can be varied and effect of these alterations measured. Therefore, it is better to think of a computer simulation falling somewhere between theory and experiment.

Mathematics has been used in biological research to great success. Winfree (Levin, 1999) used mathematics to establish that purposeless oscillations of the heart (ventricular fibrillation) were a major cause of cardiac misfunction. Computer simulations have also been used to study the blood flow in the heart (Levin, 1999). Karle and Hauptman received the Nobel Prize in 1987 for developing algorithms to reveal structures from x-ray data. Such successes in the field of theoretical biology gives credence to what can be achieved by theoretical immunology. The use of mathematics to describe the workings of the immune system is quite recent but provides an exciting application for study using a hybrid of mathematics and computer science.

Using computer simulations, a theoretical immunologist can explore almost any facet of the immune system and disease. Often experiments may be impos-

sible to do in real life, whereas a parameter of a simulation can be altered as required. There can be great cost incurred in traditional experiments where suitable candidates and tissues have to be found. A theory that is simulated can be investigated in a systematic, efficient and productive way. The outcome of this analysis can be measured against how well the model has emulated behaviour found in the real system. If the outcome is favourable, one may be on the right track. In the case of theoretical immunology this may enable the prediction of the efficacy of some drug treatment or the pathogenesis of some disease. This favourable outcome might lead to some more traditional experimentation and concrete results.

A simulation tries to mimic a particular situation with the aim of identifying the causes for certain system behaviour. The results or outcome from a simulation are used to predict what will happen to the system on *a priori* grounds. If behaviour x, was found to occur in a simulation with parameter set $\{P\}$, then it can be deduced the $\{P\}$ will cause behaviour x in the real system. This of course depends on having a "good" simulation. A "good" simulation is one where the assumptions made are sound and the rules are well-developed, but, ultimately, a "good" simulation is proved only when it mimics reality well. Computers have for the length of their lifetime been an alternative to traditional experimentation. Computer simulations are now a part of everyday life, with everything from weather forecasting to the stock market to medicine all utilising computer simulations in some way.

1.1 Immune System Models

The IMMSIMM model is a discrete computer simulation of the immune system which was developed by Seiden and Celada(1992). A discrete model is one where the components of the system such as time and cell-concentration take on discrete values. IMMSIMM aims to provide a *comprehensive* model of the interaction of the immune system with infection. This model is used to perform experiments *in machina*. Rheumatoid factor production,(Lefevre et al. 1996), vaccines,(Kohler, 1999) thymic functioning (Morpugo et al, 1995) are among the conditions and treatments which have been investigated using this model. Other successful models have been developed by the Los Alamos National Laboratory, one of the leading theoretical immunology groups in the world. The group has had much success exploring the dynamics and treatment of viral diseases such

as HIV, (Leitner and Albert, 1999), influenza (Smith et al 1998) and hepatitis (Lamb et al, 1997). These models mostly consist of differential equations and so are classed as continuous models as opposed to discrete. While theoretical immunology still has to prove itself vis-a-vis predicting traditional experimentation via its investigations, we can learn a lot more about the immune system and its behaviour. Modelling viral dynamics allows predictions to be made on the pathogenesis of the disease, its response to therapy and maybe even the design of a suitable vaccine.

1.2 What is HIV infection - AIDS ?

Acquired Immune Deficiency Syndrome, (AIDS), is not a single disease itself, rather it is a collection of diseases caused by a seriously damaged immune system. The collapsed immune system is due to the depletion of key immune cells, by the retrovirus, HIV - Human Immuno-deficiency Virus. The immunology of HIV/AIDS will be discussed in detail in the following chapter.

AIDS was first reported in the USA in 1981 and is now a worldwide epidemic. AIDS is now a bigger killer than war in Africa,(WHO, 2000); infecting 1 in 10 of the population in the sub-Sahara countries,(NIAID, 2000). The genesis of the disease is still unclear but from when it was first noticed in the early 1980's there have been 34 million people worldwide infected with HIV/AIDS and there have been 19 million recorded deaths(WHO,2000).

AIDS is defined by a depletion of certain cells of the immune system, mainly cells which are called Helper cells and also by 26 infections which are characteristic of advanced HIV in an individual. Many of these AIDS defining infections could be easily warded off by a fit immune system, but HIV leaves a body unable to defend itself. HIV infection is spread in three main ways , as a sexually transmitted disease, through infected blood(this is mainly through intravenous drug addicts who use infected needles) and from pregnant mother to unborn child. There is a long progression between first infection with HIV and AIDS defining illnesses. Because of this an infected person can appear healthy, thus increasing the risk of the disease being spread unknowingly. The mean time to develop AIDS is 7-10 years, with some people developing the disease before this time and a small number (< 50 people) having had HIV for longer than 10 years and having not yet developed AIDS like symptoms (JAMA, 1999).

There are treatments for AIDS which involve cocktails of many drugs (these

will be discussed in the next chapter). The aim of these drugs is to ameliorate the symptoms rather than cure the disease. These treatments have also proved costly and have severe side effects. Recently the top 5 pharmaceutical companies in the US announced that they would sell their treatments to African countries at 90% below the American price, but many parts of Africa lack the administration and facilities to make this effective in combating the epidemic there (CDC,2000). The medical focus has shifted from one of finding a cure to one of finding a vaccine.

Vaccines have in the past proved successful in annihilating many viruses such as smallpox and at the moment a vaccine would seem like the best hope for putting an end to the epidemic in Africa and elsewhere. Only recently have the pharmaceutical company Merck expressed optimism about their experimental AIDS vaccine (Reuters Health, 2000), but the vaccine is still on trial and has yet to prove itself to the medical world. A vaccine works by exposing small quantities of the virus to an un-infected individual. This small quantity while not enough to cause complete infection is enough to establish an immune response to the infection. Therefore, if the infection ever occurs again the immune system is primed to fight it. However, as HIV is a disease of the immune system, the construction of a suitable vaccine would seem very complicated. Given that HIV can destroy a person's immune system administering a small amount of the infection, which is required for a vaccine, would seem dangerous. Therefore increasing the knowledge about dynamics of HIV is crucial in helping construct a vaccine. Theoretical modelling is one way of looking more closely at the behaviour of the virus and determining ways to fight it.

1.3 Why model diseases

Diseases are modelled by mathematical means for a variety of reasons. It gives the scientist an opportunity to investigate any facet of the disease, or indeed any proposed vaccine/cure for the disease. Theories can be given sufficient weight so that classical "in vivo" experimentation can take place. Likewise, theories can also be discredited and research can continue whilst taking these negative findings into account.

The theoretical study of HIV evolution is performed at two distinct levels. Firstly, we have the macroscopic evolution as regards disease spread and the global diversity of HIV (Colgate et al, 1989). This gives much-needed back-

ground information on the epidemiology of the disease and the history of the pandemic. Secondly, there is the microscopic level of a single host. Concentrating on the microscopic enables us to learn about viral-immuno interactions and viral kinetics. It also allows us to identify key crossover points in the struggle between HIV and the immune system. The microscopic is the focus of the research presented here.

The contemporary nature of the disease and the fact that a lot is still unknown about the mechanics of HIV/AIDS has contributed towards the interest in modelling it. Many hypotheses exist about HIV infection and the transition to AIDS. It is a lot easier and cheaper to try and validate these hypotheses via computer simulations, governed by mathematical equations, than to use animals or other means of "in vivo" experimentation. Using mathematical models, we hope to gain some understanding of the mechanisms of HIV and therefore help in some way towards unravelling the complexity of the disease progression.

1.4 Scope of thesis

The primary goal of this thesis is to describe some discrete models of HIV infection. We also investigate the kinetics and results of simulations based upon these models. The next chapter will present a brief introduction to immunology and, specifically, its interaction with HIV. Following this, in Chapter 3, we will discuss a method of discrete modelling of complex systems, namely Cellular Automata (CA) and subsequently we outline, in Chapter 4, some past CA models of HIV infection. Our CA model is then presented in Chapter 5, followed by an introduction to the Monte Carlo method in Chapter 6. Our Monte Carlo discrete model is then investigated in Chapter 7 and we present our conclusions in Chapter 8.

1.5 New directions and contributions

In our research we have investigated discrete models of HIV's infection of the immune system. We use asynchronous Monte Carlo updating of discrete states in a lattice and is the first discrete model of HIV to do so. Synchronous updating causes "overshoot" in the system, with either immune or viral dominance and no middle ground. Asynchronous updating allows the existence of intermediate states which do not exist with synchronous updating. The status quo between immune and viral dominance is of much interest as we can investigate what

causes this to cease, with resulting immune or viral dominance. Furthermore, these intermediate states give us insight into the mechanics of the infection. Mutation, probably the most influential characteristic of HIV is incorporated into the model. As this is a computer simulation minute variations to the mutation parameter can be applied and the effect of this can be immediately seen. Using this information we can investigate what mutation rate corresponds to what stage of the disease. Also we have seen from our computer experiments that the latency period of the disease is inversely proportional to the mutation of the virus. This model now represents a framework whereby many characteristics of the disease can be investigated. Using this framework we have investigated the growth of viral and helper cells. The virus we have seen grows exponentially at the beginning and decreases to a constant level. A logarithmic function best represents helper cell growth. We have also explored the half-life of the virus, which is the time taken for the virus to half in concentration and so is indicative of the strength of the immune response. We have seen that this half-life is dependent upon the stage of infection, with progressed infection having a longer half-life. The critical time of recovery for the immune system was also investigated and it was found to be proportional to the mutation rate, with the immune system taking only a short time to recover with low mutation rates. Future studies could use and build on this framework to explore many other facets of the disease. Our models only consider fixed mutation rates. The investigation of HIV infection with a mutation rate varying throughout the simulation, will provide a rich source of exploration and research.

2 Basic Immunology

The immune system is one of the most complex, intricate, and interesting biological systems known. The job of the immune system is to protect an individual against unwanted intrusions from viruses, bacteria and other foreign invaders. The workings of the immune system consists of varying responses all working in a co-ordinated fashion. The combination of immune responses and their ability to regulate each other is fascinating and will continually be a source of fact, theory and hypothesis. Below, we present a very brief introduction to the field and especially the interaction with the Human Immuno-Deficiency Virus (HIV). Most of the description of immunology presented below is a summary of a more in-depth introduction given by (Roitt, 1994) and (Kuby,1992).

2.1 Cell-mediated and Humoral arms

The lymphoid tissues and the organs comprise the lymphoid system which is a core component of the immunological system. Lymphocytes (white blood cells) are the predominant cells of the lymphoid system which also include macrophages (a type of white blood cell) and plasma cells (antibody producing cells). Lymphocytes circulate around the body via the blood system and also the lymphatic channels. The lymphatic channels are used to transport lymph to the successive lymph nodes. At a lymph node impurities in the lymph are filtered by lymphocytes. The immune system is able to recognise foreign invaders(antigen). The recognition of an invader stimulates two main defences, the humoral and the cell-mediated defence (see below).

Cell-Mediated Arm

The cell-mediated arm is concerned with cells which are infected by an intruder (viral infected cells). The cell-mediated response consists of the activation and proliferation of T8 cytotoxic killer cells (CD8+ T cells). T cells are lymphocytes, which mature in the thymus. T8 are T cells which have CD8 protein on their surface. Infected cells are killed by cytotoxic killer T cells . A brief overview of the activation of these killer cells is as follows. Macrophages, (and related antigen presenting cells), which are the body's first line of defence, circulate and ingest any free antigen present in the body. Macrophages have proteins called Class II Major Histocompatibility Complex, (MHC), a complex of genes encoding cell-surface molecules, on their surface. After they ingest antigen,

they display protein fragments of the antigen on their surface along with the MHC protein. Helper T4 cells (a subclass of T cells, which are lymphocytes which mature in the thymus and have CD4 protein (see Section 2.2.2) on their surface) peruse Macrophages. When they encounter the combination of MHC and antigenic protein, they secrete cytokines which stimulate the growth of Killer,T8 cells. Cytokines are proteins which are used for communication by the cells of the immune system. This immune response is very specific, Helper T4 cells have receptors with the ability to recognise a *single epitope* (epitopes are what identifies an antigen) and they will only stimulate a killer response if they encounter this specific epitope on the surface of the macrophage. The cell-mediated response also has a switching-off mechanism. This mechanism consists of another subset of T8 cells, called suppressor cells, whose function it is to secrete cytokines which shut-down the killer cells.

Humoral Arm

The humoral arm is concerned with free antigen (free viral particles) and producing antibodies which neutralise these free virions. The humoral response consists of the maturation and proliferation of B cells (a class of lymphocyte which mature in bone marrow). They differentiate into memory and plasma cells. The memory cells guard against future infection; where as plasma cells secrete antibodies. Antibodies mark free antigens and inactivate them. The humoral response is activated in much the same way as the cell-mediated response, with Helper cells releasing cytokines to stimulate the Humoural arm.

As can be seen from their brief description above, both the Cell-Mediated and Humoral arms are reliant on the correct functioning of T helper cells (CD4+ T cells). T4 Helper cells can be sub-divided further into two T4 cell sub-groups, namely Th1 and Th2, with Th1 controlling the cell mediated response while Th2 controls the humoral response. Th1 and Th2 are *self-regulating* if one is switched on, the other is switched off. Helper cells are central to the correction regulation of the immune response and any impairment of their functionality would result in a mal-functioning immune system.

In addition to both immune responses mentioned above, there are the natural killer (NK) cells, (lymphocytes which are neither B nor T cells) which are active at the first stages of infection and which decrease, as the cell-mediated response becomes dominant. They kill viral cells independently of the MHC complex and any stimulation. Their role in a viral infection is important as they kill cells

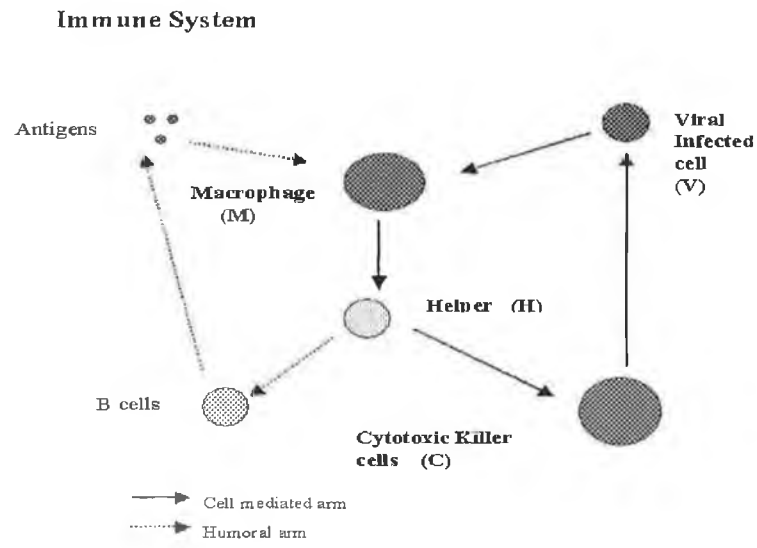


Figure 2.1 The Humoral and Cell-Mediated arms of the Immune System

with growth and surface properties inappropriate for normal cell functions. As viral-infected cells are abnormal they would be recognised as such by NK cells. The cytokine, (IL2-the same cytokine that activates T8 killer cells), is thought to cause an increase in the cytotoxicity of the NK cells and their proliferation. NK cells are thought to decrease once the cell mediated response is activated (it is said that this is due to the competition from T8 killer cells for IL2).

2.2 AIDS/HIV and the Immune System

AIDS (Acquired Immune Deficiency Syndrome) is characterised by a depletion in certain cells of the immune system namely, T4, T8, and B cells. The most notable decline is in the T4 helper cell population. In a healthy person there are 800-1,200 T4 per $\mu Litre$, while at the onset of AIDS an HIV-infected individual would have < 200 per $\mu Litre$. There is usually a long progression period from the first infection with the HIV virus to full blown AIDS. In the first stages of infection, the immune system seems able to cope, but eventually HIV gets the upper hand and leads to the total collapse of the body's defences. Viruses which have a *long interval* between initial infection and onset of serious symptoms are called lentiviruses.

Viruses are noncellular and consist of nucleoid acid surrounded by a protein

coat. They spread by infecting cells and using the host cell to help it propagate. Influenza and Chicken Pox are examples of common ailments caused by viral infection. HIV, like all viruses can only replicate inside cells. The virus infects a cell and commandeers the cell's machinery to reproduce.

2.2.1 HIV - A Retrovirus

HIV is a retrovirus, the genes of which are encoded in single stranded RNA. The normal flow of genetic information is from DNA,(which is double stranded), to RNA,(single stranded). However, with a retrovirus this flow is reversed, single stranded RNA is converted into double stranded DNA. This double stranded DNA, is then integrated into the chromosome of the host, where it directs the production of more viral RNA. The control of this conversion from RNA to DNA is a viral gene called reverse transcriptase. The whole process is called reverse transcription and is incredibly error prone (Nowak and McMichael, 1995). These errors then cause mutations in the genetic makeup of the virus. A virus which rapidly mutates is extremely difficult to fight because the enemy as such is continously changing guise. Being a retrovirus, HIV has a high mutation rate and it is in fact the most highly mutating virus known, (Nowak and McMichael,1995). In any given individual there can be genetic differences of *up to 25%* in the viral particles. The high mutation rate is incredibly advantageous to the virus as it increases the probability that a chance mutation will yield an advantage for the virus. Following one of the fundamental rules of evolution theory, the survival of the fittest, cells with this advantage will become abundant. The mutations would lead to epitopes,(an antigenic determinant), unrecognisable by the host cells, and when they eventually respond to it, the virus could have mutated again. When the virus is integrated into the chromosome of the host it is referred to as a provirus. Then every time the cell divides the virus is also duplicated.

2.2.2 Why HIV targets Helper cells

HIV targets cells which have a receptor molecule called cluster designation 4(CD4) on their surface. Cells which have this molecule are referred to as CD4 cells. gp120 is a protein which resides above and below HIV's surface and which influences the types of cells which HIV infects. gp120 and CD4 are complementary, therefore making CD4 cells a target for HIV. Therefore, HIV has an affinity for T4 helper cells, as they display CD4 on their surface. As should

be obvious from above, T4 helper cells are *central to the correct orchestration of immune defences*, so their depletion weakens the immune system significantly.

The infection of T4 cells leads to the destruction of the T4 cells (either by the virus itself or by T8 killer cells as they are now viral-infected cells) or in some cases the T4 cells return to a resting state, (now infected with HIV). This resting state is a normal immunological response as it represents immunological memory.

Originally the period in which the immune system responded well to the virus was thought of as some sort of latency period for the virus. This idea of latency has since been altered, as instead of the virus ceasing reproduction the immune system is just about keeping it under control(Blakeslee 1998). However, we will continue to reference it under this nomenclature as it is common in literature.

As mentioned before, Helper cells can be subdivided into Th1 and Th2, with their responses being self-regulatory. This is necessary for the proper functioning of the immune system, as a Th2 type response is important at the beginning of infection and a Th1 response is important for progressed infections. In an infection like HIV where the cell-mediated response is more important, a dominant humoral response could be disastrous, as there are far more viral-infected cells than free virions. Indeed, it has been found that the Th2 response seems to dominate over the Th1 response in the course of HIV infection. Researchers think that in the beginning of infection there is a Th1 response which then switches later to a Th2 response (Clerici, 1993). The high mutation rate of the virus then leads it to escape these antibody (Th2) attacks. Recent findings have shown that Th2-type cytokines are dominant in the saliva of HIV-infected individuals(Leigh, 1998). It has been said that Th1 cytokines can be associated with resistance to HIV infection and Th2 cytokines with the progressive phase of the disease (Clerici 1993). The protein gp120 also needs co-receptors, proteins on the outside of cells, to attach to a cell. Two proteins CXCR4 and CCR5 are co-receptors and so along with gp120 are necessary for HIV to attach to a cell, (Cohen and Fauci, 1998). Further, CCR5 is a preferred chemokine receptor for Th1 type cell (Blakeslee, 1998) thus making it more of a target for HIV than Th2 type cells, therefore hindering the cell-mediated attack. It should be noted at this stage that Th1/Th2 differentiation is not incorporated into any of our models at this stage. A framework for a model should use well established immunological fact and when the model proves itself, it can then be built upon to include more contemporary theories.

2.3 HIV/AIDS Treatment

HIV treatment is targeted at different steps in the replication cycle of HIV. Some drugs target the reverse transcriptase enyzme which is responsible for converting RNA into DNA. These drugs are called *nucleosides* and *non-nucleoside* reverse transcriptase inhibitors. Nucleosides act by incorporating themselves into the DNA of virus, thereby stopping the building process; the resulting DNA is incomplete and cannot create new virus. Non-nucleosides stop HIV production by binding directly onto reverse transcriptase and preventing conversion of RNA to DNA. There are also drugs called protease inhibitors which work at the last stage of the viral reproduction cycle; preventing HIV from being successfully assembled and released from infected CD4+ cells.

HAART (High-Active Retroviral Therapy) consists of combinations of these three drug types and has proved successful in prolonging the lives of HIV-infected individuals. HAART and all other HIV treatment is expensive, $\approx \$30,000$ per year (this is a conservative estimate), and patients can suffer severe side effects. Arising from the study of this treatment is the finding that the complete eradication of HIV from an infected person is *impossible*. The reason for this is the population of resting T4 cells. A resting T4 memory cell state represents a reservoir for the virus (see Section 2.2), as a virus in this state is hidden from the immune system and provokes no reaction (Blakeslee, 1997). Almost all current anti-HIV drugs cannot attack HIV in this resting state (Siliciano,1998) . Normally, these cells can be thought of as T memory cells, which will become stimulated again when the *right epitope* is encountered. However, in the case of HIV these cells are infected with HIV and HIV will start replicating again when the T4 cell is stimulated. It is unclear how big a threat these resting memory T cells infected with HIV are, as they only make up 1 in 10,0000 of all T lymphocytes(Blakeslee, 1997). Therefore, it can be seen that although much progress has been made, a great deal more about the pathogenesis of AIDS must be learned, before an effective treatment/vaccine is found.

3 Cellular Automata

3.1 Introduction

Cellular Automata (CA) are a class of discrete models. They aim to mimic complex behaviour by building a system up from simple local interactions. This is in contrast to the global perspective taken by more traditional modelling approaches. Before discussing the history,background and applications of Cellular Automata(CA), let us first discuss what we mean by cellular automata. All CA models share a number of basic characteristics .

3.1.1 The Cell

The basic building block of any CA is the cell. A cell can be representative of a microscopic entity, e.g. a plant cell in a model of plant growth, or can represent a macroscopic unit, e.g. a tree in a model of forest fire spread. All cells act in a homogenous way and their actions are governed by rules (see below). A cell can be in a discrete number of states. The interpretation of what the differing states mean depends on the application. For instance, if discussing a living organism, a state with value 1 may represent life while the state with value 0 may represent death. In a traffic flow simulation, however, the discrete states may represent the velocity of the car, so state 0, represents a stopped car, while state, n, would mean a car travelling n mph (Schreckenberg & Schadschneider, 1996).

3.1.2 The Lattice (or Grid)

The cells sit on a lattice. The lattice is usually employed to take into account the spatial relationship between cells. If we refer again to traffic flow, a car, x, will be in a traffic jam if there are 'cars' at each cell on the lattice adjacent to the cell that x occupies. The spatial relationship between entities is very important when considering many physical systems and cellular automata's innate ability to incorporate this, is one of its many strengths.

The lattice can, however, represent a relationship other than a spatial one. The BSP model (de Boer et al, 1992) is an immunological CA which simulates the interaction between B cells and other B cells via their complementary receptors. This is incorporated by a cell's mirror image on the lattice representing a perfect fit between two B cells while a cell's neighbour's image represents a less exact

fit. Therefore, the lattice in this situation is used as a means of establishing the "lock and key" relationship between complementary receptors on B cells.

If one is modelling more than one interacting entity then alternatively it is better to think of sites rather than cells. Each site can hold the n differing entities. Each site then behaves as an n-dimensional cell with the entities interacting together at the site. The lattice can be of any dimension depending on what suits the application best, e.g. where traffic flow can be represented in 1-D , simulating fluid flow may be best represented in 2-D so that both horizontal and vertical flows are considered.

Alternatively, a lattice may not be considered at all and this is the mean-field approach. In this approach one may consider a string of n bits, the rules of the CA are then applied continuously to these n bits until a fixed point or cycle is found.

3.1.3 Neighbourhoods

Another strength of Cellular Automata in modelling natural phenomena is that the influence of a cell's neighbours is considered. A sense of locality is very important in many physical systems and CA offers a way to incorporate this. A nearest neighbourhood is defined in CA for every cell (or site). This nearest neighbourhood is made up of the cells close enough to x on the lattice to have an influence on it. This introduction of locality means that every interaction has a specific location and range of effect. The range of effect being its nearest neighbourhood. There are different neighbourhoods and they are defined below (Wolfram, 1986) (the definitions are given in 2-D but can easily be extended to higher dimensions).

von Neumann neighbourhood - Five cells consisting of the cell itself, the cell above and below and the cell right and left, i.e. the cells that share a common edge with the central cell.

Moore neighbourhood - This is similar to the von Neumann neighbourhood but contains the cells at its diagonals as well, i.e. the cells that share a common node with the central cell.

Extended Moore neighbourhood - This is an extension of the Moore neighbourhood to the next adjacent cells.

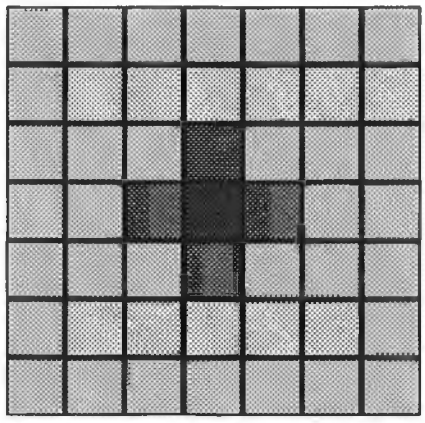


Figure 3.1 von Neumann neighbourhood

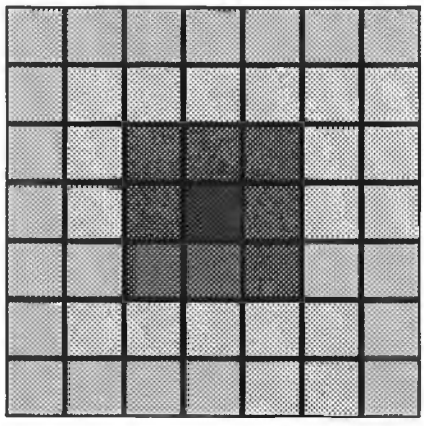


Figure 3.2 Moore neighbourhood

The neighbourhood one would use in a specific application depends on the connectivity of the environment one is modelling. The more connected the environment the larger the range of effect, therefore the definition of neighbourhood in a highly connected system would include many neighbours.

3.1.4 The "rules"

The influence of a cell's neighbourhood is governed by the nearest-neighbour interactions of the model. These determine a cell's state at the next timestep. In CA, rules can be deterministic or probabilistic; if the latter, the model is correctly termed stochastic CA(SCA) or probabilistic CA(PCA). The rules are dependent on the application. Conway's Game of Life aimed to model living

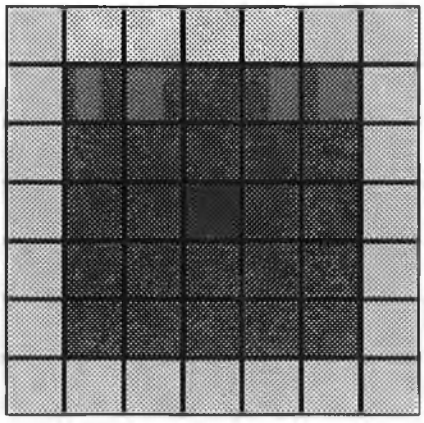


Figure 3.3 Extended Moore neighbourhood

organisms. Each cell was capable of holding a single organism, the state of the cell was dead(0) or alive (1) and the neighbourhood was a Moore one. The rules which governed the life or death of an organism were based solely on the number of "alive" neighbours a cell, x, had. A cell died if it was overcrowded (this was defined has having >3 alive neighbours). A cell also died if it was lonely, ($\leq$1 alive neighbour). The condition conducive to having a cell come alive was having exactly 3 alive neighbours. A status quo was achieved by having exactly 2 alive nearest neighbours, thereby the state of the cell would remain the same. The above example is used to illustrate how the rules, based on the system being modelled, are the mechanics behind the evolution of a CA model. If more than one entity is being modelled and all entities can occupy a cell together, than additional inter-cell interactions can be specified.

3.1.5 Timesteps

The nearest neighbour interaction followed by any other additional local interaction determines a single timestep. Therefore, the timestep is set by the nature of the process being simulated. In a forest fire, fire spread is slow at the beginning and then gains momentum and so spreads more quickly. A single timestep in a CA model of forest fire spread would represent more real time at the start of the simulation, (when the fire spread is slow), than later on, (when the fire is spreading faster). Therefore, a CA timestep has a tenuous link with real-time and caution is advised when trying to establish a relationship between the two.

3.2 Background and Applications

Cellular Automata theory was introduced by von Neumann and Ulam in the 1940s , where von Neumann was principally concerned with describing elementary units which are capable of reproducing themselves. Von Nuemann described an automaton with 29 different states and which he hoped was a Turing machine i.e. capable of performing any calculation possible(Preston & Duff, 1984). Ulam formalised von Neumann's idea of automatons by considering these elements positioned on a lattice and including nearest neighbour interactions. The lattice is important to CA because with out it one cannot establish nearest neighbours for a cell. Without nearest neighbours CA have no sense of locality. The concept of locality is central to CA as it permits the location of an entity to be influential and also the interactions on the entity to have a range of effect. CA were popularised by Conway's Game of Life via Gardner's "Mathematical Games" column in Scientific American (1971). This coinciding with the emerging popularity of computer simulations led to many physical systems being modelled through CA.

In the 1980's Wolfram produced a body of work on the subject most of which is collected in (Wolfram, 1986). Wolfram's systematic studying of CA behaviours led him to develop four classes of CA grouped by their phenomenological behaviour. The first class of CA is characterised by a spatially homogenous global state (see Figure 3.4) , periodic stable structures are characteristic of the second class (see Figure 3.5), chaotic behaviour is exhibited in the third class while in the fourth class there are some localised static complex structures with other structures moving around the cellular space. See figures 3.4-3.7, which were constructed from examples given in Wolfram (1986). Wolfram reasoned, (Wolfram, 1984), that because of the small number of classes, universality existed in the complex behaviour CA exhibits. Furthermore, because of this universality he proposed that many of the details involved in creating a CA are irrelevant w.r.t determining their qualitative behaviour. This he surmised could imply that complex physical and biological systems may too, fall into one of the above four classes. Therefore the study of CA may find some general results that can be applied to these complex systems. The above classification of CA is phenotypic as the CA which exhibit similar behaviours are grouped into the same class. There is also a genotypic classification, i.e. it is derived from the mechanics which drive the simulation. This classification uses the parameter, λ, developed by Langton, λ, is a measure of the state transitions in the rule space

Figure 3.4

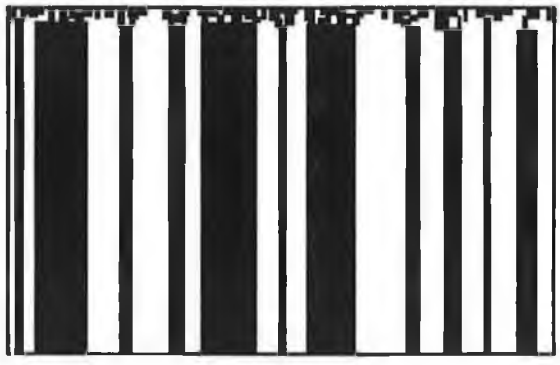


Figure 3.5

of a CA (Gutowitz, 1996, Li et al, 1990). However, Wolfram's phenomenological classification remains the most popular. This is probably due to its intuitive nature and its examination of behaviour, looking at the output which a CA produces, one can easily deduce to which Wolfram class it belongs. The science of complexity is very much based on observed behaviours of systems.

CAs can produce quite complex global behaviour from simple local interactions. A real-life example would be a Mexican wave at a football game. Watching from a distance it looks quite complex and magnificent, however, at a local level it consists of a simple nearest neighbour interaction. That interaction being a cell, (spectator), looking at its next nearest neighbour and when they have changed state, (i.e. waved), then that cell changes state too. Cellular Automata are counterpoint to partial differentiation equations, because instead of trying to simulate complex behaviour from "above" using complex equations, they seek

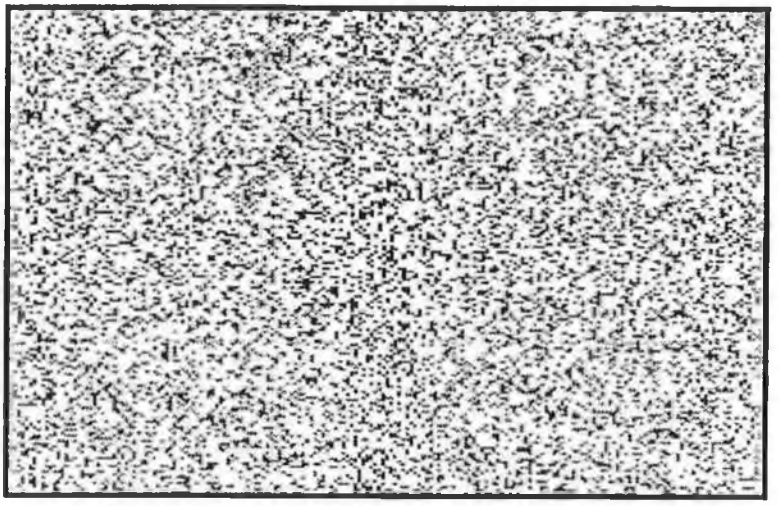


Figure 3.6

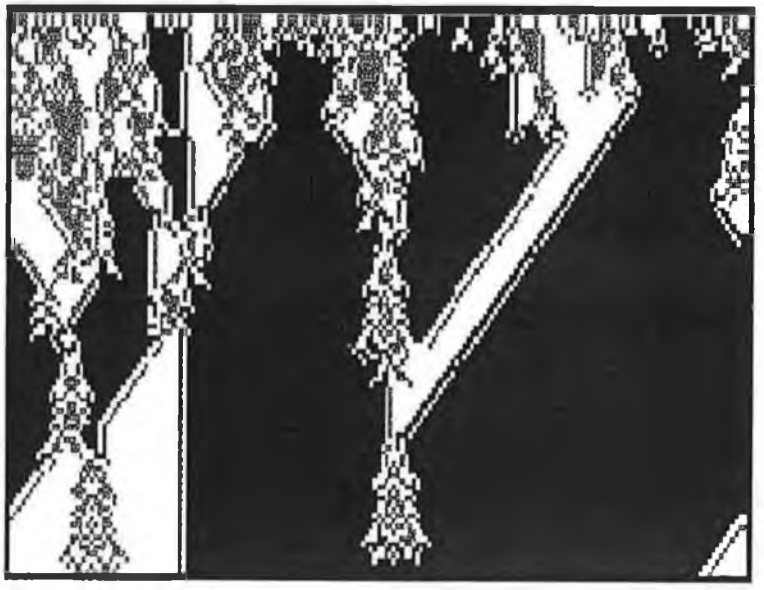


Figure 3.7

to generate complex behaviour at a macroscopic level, by simple rules at a local level, i.e. a bottom-up rather than a top-down approach.

CAs have been used as extremely simple models of differential equations in physics, examples include heat and wave equations (Toffoli, 1984)and the Navier-Stokes equation(Frisch et al, 1986) There are many CA models of biological systems, such as tumour growth (Smolle & Stettner, 1993) and ecological systems, e.g. forest fire spread , (Green et al, 1990), starfish outburst (Hogeweg &Hesper 1990) in existence. CA models have also been used in immunology (de Boer et al, 1992, Deffner, 1993, Zorzenon dos Santos 1993, Seiden and Celada, 1992) with Kaufman et al (1985) developing the first immunological CA model .

With CA, as with differential equations, it is possible to alter the parameters of the system being modelled and establish the consequence of this variation on the system. However, CA holds many advantages over differential equations. CA rules in general are "intuitive", simple and easily modified. Differential equations are often complex involving many parameters and *nth* derivatives. Due to the fact that CA model directly the basic elements of the system, one has tighter control over the manipulation of the model at a microscopic level i.e. at an individual element level and any parameters effecting it. Using CA one can hypothesise about behaviour at the microscopic level and see what the macroscopic behaviour appears like. If the system level behaviour mimics known behaviour then the CA model provides a much richer picture as insight is gained into the microscopic as well as the macroscopic. CA lets behaviour emerge from the model while differential equations attempt to describe behaviour. Of course differential equations have advantages over CA as they are based on rigorously defined mathematical concepts. Differential equations have been the de-facto modelling tool for physical sytems for centuries and have been found to model them very effectively. Many tenets of a system have a known differential part, a simple example would be the velocity of an object. CA does not have such well established resuable characteristics to draw on. CA, however can help one to learn about emergent behaviour and how the basic elements at the microscopic level can influence the overall behaviour of the system. Due to CA's deterministic rules, the individual elements can be *completely controlled.* Also, because the models are made up of simple building blocks for which interactions locally are quite simple but whose overall macroscopic behaviour is quite complicated, they are analogous with nature.

4 CA Models of HIV Infection

4.1 Introduction

The formation of any model involves making assumptions, simplifying relationships and behaviour, and distinguishing the important elements to be included. Therefore, any model is always open to improvements, from new knowledge acquired and from the lessons learned from previous models. No model can be called definitive and can be taken only in context of its current knowledge base and what has been garnered from previous models. Both theoretical and immunology CA modelling are relatively new fields so the archive of knowledge and expertise is not as extensive as it would be if one were formulating a model in a traditional discipline e.g. physics, using traditional methods, such as differential equations.

4.2 The Model Basics

In this section, we present different models of HIV infection and its interaction with the immune system. The different models focus on different aspects of the infection and include differing cell types. They do, however, share a number of basic characteristics and we present these here. All models we discuss in this section are simulated on a 2-D or 3-D regular lattice of length, L with L^2 or L^3 sites. Each site on the lattice is capable of holding at *most* one of each cell type. Each cell type considered is in one of two concentrations, high concentration (1), or low concentration (0). It is important to note at this stage that sites contain n distinct cell types and that sites are involved in two different types of interactions. The first type of interaction is interior to the site, intra-site (inter-cell) interactions, where cell types interact with different cell types at a site. These are Boolean equations and have to be a succinct interpretation of the cell-type's behaviour in the system. Sites also interact with their neighbouring sites and these interactions are the nearest-neighbour interactions. These are restricted between cells of the same type and so are termed, intra-cell, or inter-site interactions.

The immune system is a highly connected environment; location and range of influence are important factors in correctly simulating it. Nearest-neighbour interactions are used to mimic this connectivity by taking into account the

influence of a site's nearest neighbours. The neighbourhood considered is von Neumann, (for definition see Chapter 3), and in all the models we discuss, the nearest-neighbour interaction consists of a logical-OR between the nearest neighbours. Thus, if a cell-type is in high concentration at any one of its nearest neighbours then it will be in high concentration at that given site too. Eqn (4.1) shows the code for the nearest-neighbour interaction for $c(i,j,k)$,where $c(i,j,k)$ is a representation of a cell-type, it is either 1 or 0 , i,k is its position on a $2-d$ lattice and k is the cell -type

$$c(i,j,k) = c(i,j,k) \;||\; c(i+1,j,k) \;||\; c(i-1,j,k) \;||\; c(i,j+1,k) \;||\; c(i,j-1,k) \tag{4.1}$$

Updating is synchronous with Cellular Automata; all interactions take place at all sites simultaneously. At each timestep, each site on the lattice is updated by the nearest-neighbour interactions followed by intra-site interactions.

4.2.1 What is the subject of these models ?

A question which needs to be answered before formulating any model is: what exactly is it that one wants to model ? Once this has been established, it is important to ask, what is necessary to include, so as to model this effectively ? The aim of the models presented below, is to try and simulate HIV and its infection of the body and also the immune system's response to this infection. The immune system's interaction with any invader, no matter how small and threatening, is a very complex one. Modelling an infection such as HIV, where a lot is still unknown, is daunting. HIVs invasion triggers both a humoral and cell-mediated response. Each of these defences consist of various cell types and cell-mediators (lymphokines, cytokines and other regulatory molecules). The majority of the models decide to concentrate on one type of immune defence, cell-mediated being the preferred choice. Why ? Because HIV is a viral infection and the cell-mediated defence targets viral-infected cells and also it simplifies the formation of the model. If this is the case, the answer to the first question, "what is being modelled ?", is the interaction of HIV and the cell-mediated response to it.

Once the subject of the simulation has been decided upon, another question arises. How ? The modelling tool used for all the models presented below, is Cellular Automata. What restrictions does this imply ? Firstly, cell types have to be in discrete concentrations, Boolean concentrations, so they require

Boolean equations. Boolean equations require that very basic generalities about the cell-types need to be abstracted. Therefore, the way in which immune system, or particularly the cell-mediated defence, tries to defeat HIV needs to be investigated. This has to be done with an eye for choosing generalities about each cell type involved and about the virus itself.

One also needs to consider the number of cell types to include. The more cell-types we include, the more complicated the model is made and the harder it is to abstract these generalities. At the same time, one does not want to leave out some vital part of the system. It is a delicate balance between including too little information, with the result that differing cell-types are over-reliant on each other or including too many, so that there is much redundancy and many components are basically doing the same job. One might consider the activation of a killer cell as being very important to include. One would investigate what is clinically known and realise it is brought about by the interaction of an antigen presenting cell, helper cell and various lymphokines and cytokines. One may just pare this down to the generalities; a viral invader encounters an antigen presenting cell which stimulates the growth of helper cells which in turn, stimulates the growth of killer cells. Therefore, the activation of killer cells has been simplified to four cell types: APC, killer, helper and viral cells. Boolean equations can then be constructed which encompass the basic traits and interactions of these cell types.

Another consideration is the hardware of the computer, this imposes physical limits on the model. This point can be illustrated by considering a model with a great number of cell types. If instead one modelled killer cell activation with a 10-cell model which included the four cell types mentioned in the previous paragraph along with free virions, NK cells, lymphokines, suppressor T cells,B cells, and antibodies. If a modest 100x100, 2-d lattice was employed for the simulation, one would have to store for each i,j co-ordinate on the lattice 10 bits of information. Each timestep would generate 100,000 bits of information each timestep which has be stored. If the simulation is run for a 1000 timesteps and 50 runs, this entails a computer dealing with at least 500 million calculations. However, hardware considerations are not as important as they once were as computing capacity is continuing to increase and will accommodate such strains on current resources. This will allow computer experimentalists in the future to formulate models without having to overly consider hardware constraints.

Many of the models discussed below include varying numbers of cell types from

the immune system and virus. They have differing intra-site interactions governing their propagation. The differences exist in an effort to explore alternative views of what is happening, incorporate some new finding or because of weaknesses exposed in assumptions or in the structure of a previous model.

4.3 The Models

4.3.1 PS1

The first CA model, PS1, to tackle the interaction between the immune system and HIV, is that of Pandey and Stauffer (1989). The authors considered three cell types. These consisting of the Helper cell (H) , the cytotoxic Killer cell, (C), and the viral infected cell (V). The immune cells, H and C, represented the cell mediated arm of the immune system. Each of these cell types was represented by Boolean expressions ruling their behaviour, with the right-hand side and left-hand side of these expressions representing the concentrations (high(1) or low (0)) at time, t and $t+1$ respectively. The Boolean equations representing the behaviours of these immune cells are

$$H = H \ and \ (not \ (V)) \tag{4.2}$$

$$C = H \ or \ C \tag{4.3}$$

which means that an H will self- propagate at the next timestep, if V is not present at the current timestep. This incorporates the basic principle that HIV kills Helper cells. Nothing in this model triggers the formation of an H and so it relies on the spread from the initial concentration of H-seeded sites via nearest-neighbour interactions. The killer cell (C) can also self-propagate or can be triggered by the Helper cell. There is no suppression factor on C so once a killer cell is present at a site, it will remain ad infinitum. There are two separate equations governing the behaviour of the virus.

$$V = (H \ and \ V) \ and \ (not \ (C)) \tag{4.4}$$

$$V = not \ (H \ and \ V \ and \ C) \tag{4.5}$$

Eqn (4.5) is more virulent than Eqn(4.4), as V will always propagate except when all three cell types are present at a site, while Eqn(4.4) needs both the presence of an H and a V and the absence of Killer cells to propagate. With Eqn(4.4) a unique combination of cell type concentrations creates a V (H=V=1, C=0) , Eqn(4.5) has seven possible arrangements of cell type concentrations to

make a V ((i)H=V=C=0, (ii)V=C=1, H=0, (iii)V=H=1, C=0, (iv) H=C=1, V=0, (v)H=V=0, C=1,(vi) H=C=0, V=1, (vii),V=C=1, H=0)

Equation (4.4) is chosen to occur at a given site with probability, B, and Eqn. (4.5) with probability, $(1-B)$, so this is a probabilistic cellular automata model (PCA) model. To incorporate this probability into the model, two different types of random mixing of Eqn(4.4) and Eqn(4.5) were employed. Quenched and annealed were the mixing methods used. Quenched and annealed represent different viewpoints. One view is that the same equation should always be used at the same site, so that it becomes an attribute of the site. This is the quenched approach. At the start of the simulation $B * L^2$ sites are assigned Eqn(4.4) for the length of the simulation and $1 - B * L^2$ sites are assigned, Eqn(4.5). On the other hand, with annealed mixing, at each time step and at each site, one of the two viral equations is chosen. The two approaches are conceptually different. In the quenched approach, the sites allocated with Eqn(4.5) for the entire simulation can be thought of as representing something akin to static immune weaknesses, a frozen attribute. This is because Eqn(4.5) would be representative of the immune system's inability to handle the invader, while Eqn(4.4) would be an attribute of competent immune system response. On the other hand, annealed mixing, represents a mobile virulent virus or a transient immune system deficiency. Two equations represented the virus, because at that time not much about the mechanics of the virus was known, the virus was known to be virulent in some stages and less virulent at others (i.e latency period).

PS1 was simulated on a 3-d lattice of length 60, (216,000 sites), with an initial concentration $p = 0.0005$ of each cell type ($\approx$108 of each cell type). The results of the model showed that as the probability of the more virulent virus, i.e Eqn (4.5), increased the helper cell population decreased. Therefore, the less virulent virus may represent early infection with the more virulent one representing the collapse of the immune system. One may ask what changes the behaviour of the virus from Eqn.(4.4) to Eqn(4.5) ? An answer may be the continuing mutation of the virus. As PS1 was an exploratory model not too much weight was attached to the reasoning for the transition from Eqn(4.4) type viral behaviour to Eqn(4.5) type behaviour.

4.3.2 P1

Pandey built further on PS1 with P1 (Pandey, 1989) and introduced a second alternate equation for the Helper cell, (H), which has the general effect of enhancing Helper growth. This, as we understand it, was introduced to complement the enhanced viral proliferation brought about by Eqn(4.5)

$$H = not\ (H\ and\ V) \tag{4.6}$$

Using this equation for H, it can be seen that the cell-mediated response can be triggered by the virus, i.e. V=1 and H=0 leads to H=1 at time $t+1$,which is biologically sound. If H and V were present together than the virus would kill the helper cell. However, an H can be generated from both V and H being absent which does not make sense. Two interaction sets were used, interaction set (1), consisted of equations (4.2),(4.3),(4.4) and interaction set (2) comprised of (4.3),(4.5),(4.6). Interestingly (4.5) and (4.6) are in the same interaction set which leads to H and V appearing simultaneously at sites. This results in their populations oscillating together. However, this would not make sense as the virus kills helper cells and they should not realistically be growing together. The interactions sets (1) and (2) were mixed randomly using both quenched and annealed mixing. As the probability of using interaction set (2) increases, the immune system weakens, which is predictable as interaction set (2) represents a very strong viral attack and a weakened immune response.

4.3.3 KS1

Kougias and Schulte (1990) formulated a model, with again the same complement of three cell types: Helper, Killer and Viral infected. In this model each cell type has only one rule governing its intra-site interactions. There is no random mixing as before, the authors found "no biological justification" for it. This comment signifies a difference in approach between PS1, P1 and KS1. The previous models, PS1, P1, were exploratory with different equations for the virus and helper cell. These equations were mixed as a method of investigating different immune responses and alternative viral attacks. The approach by KS1 is to determine what equation best suits the cell type and not alter that. This approach means that the outcome is better understood, as the rules are completely deterministic. The simulation takes place on a 2-D lattice with $L = 512$, (262,144 sites), and an initial concentration for each cell type of $p = 0.001$, ($\approx$ 262 sites). The Boolean equation representing the Helper cells is

$$H = H \; xor \; V \tag{4.7}$$

This means that V can trigger an H response if no H is already present, or it will kill an H if it is already present. The exclusive-OR means that if both H and V are in low concentration then no H response is triggered. Killer cells were represented by

$$C = (V \; and \; H) or \; C \tag{4.8}$$

An activated helper (a helper in the presence of a viral infected cell) stimulates killer cell growth, or if the killer-cell itself is already present it will self-propagate. This means that once a killer cell is generated, it will keep proliferating as there is no suppression on its propagation.

The viral infected cell's behaviour was described by

$$V = (V \; or \; H) \; and \; (not \; C) \tag{4.9}$$

Therefore an H or a V can generate more virus only in the absence of C, which would kill the virus if it was present. These three equations represent a succinct view of the interaction between the immune system and HIV. The authors showed that an increase in viral population results in a corresponding decay in helper cell population. While this is a basic result, it is still representative of clinical findings, i.e. Helper cell counts decrease as HIV cell counts increase.

4.3.4 P2

Pandey(1991) formulated a model P2 that included the humoral response along with the cell-mediated response. It was an eight cell model that included free virions, viral infected cells, macrophages, lymphokines, killer cells, helper cells, antibodies and suppressor cells. This was a departure from previous models, which concentrated only on the cell-mediated response. The previous models showed, that while the 3-cell model was easy to implement, more cell types were needed so that cell types were not over dependent on each other. A 3-D lattice of length 64, (262,144 sites) was employed with an initial concentration of cell types, $p = 0.000005$, ($\approx$ 1 of each cell type). A model of this type, with many different cell types, can easily contain redundancy with over complicated expressions involving many cell types where one would have sufficed. The eight different cell types included, C_1, free antigen, C_2, antigen expressed in a particular form, C_3, macrophages, C_4 helper cells, C_5 lymphokines,C_6 killer cells, C_7 suppressor cells,C_8 antibodies.

$$C_1 = (C_1 \ or \ C_4) \ and \ (not \ (C_8)) \tag{4.10}$$

Free virions will propagate in the absence of antibodies (which would kill them) if free virions or helper cells are already present. This indicates that the infected Helper cells will eventually explode and release free virions.

$$C_2 = (C_1 \ and \ C_3) \ and \ (not \ (C_6)) \tag{4.11}$$

C_2 represents antigen expressed in a particular form, e.g on a macrophage. These will propagate in the absence of killer cells (as killer cells kill viral infected cells) and if both free antigens and macrophages are present.

$$C_3 = C_3 \ or \ C_1 \tag{4.12}$$

C_3 represents macrophages. These self-propagate or are triggered by free antigens (virions), as they are the first line of the defence against the viral invader.

$$C_4 = (C_4 \ or \ C_2) \ and \ (not \ (C_1)) \tag{4.13}$$

C_4 represents helper cells, which will be killed by free antigens but in the absence of free antigen will self-propagate or be stimulated by antigen on a macrophage.

$$C_5 = (C_5 \ or \ C_4) \ and \ (not \ (C_7)) \tag{4.14}$$

C_5 are lymphokines. In the absence of suppressor cells, they will self-propagate or be propagated by the presence of helper cells.

$$C_6 = C_6 \ or \ C_5 \tag{4.15}$$

C_6 represents killer cells. These will self-propagate or be stimulated by lymphokines

$$C_7 = C_7 \ or \ C_5 \tag{4.16}$$

C_7 represents suppressor cells. These will self-propagate or be stimulated by lymphokines

$$C_8 = C_8 \ or \ C_5 \tag{4.17}$$

C_8 represents antibodies. These too will either self-propagate or be stimulated by lymphokines

Once lymphokines are formed, they will trigger killer and suppressor cells and antibodies. These are all self-propagating and have no suppression factor built in. Therefore the role of lymphokines is influential only at the beginning of the simulation. The role of suppressor cells is now redundant, as their purpose is to suppress or halt the cell-mediated and humoral defences, which in this model continue ad infinitum when stimulated . This is a characteristic weakness of a model involving many different cell types, where one cell type, namely lymphokines, manage to obliterate indirectly, the influence of another cell type, namely the suppressor cells.

Once more, there is an alternative interaction for the virus, and as we are dealing with two different guises of virus, i.e free antigen and viral infected cells, each of these has an alternative interaction. These alternative interactions are as follows

$$C_1 = (C_1 \; or \; C_4) \; and \; (not \; (C_8 \; and \; C_5)) \tag{4.18}$$

$$C_2 = (C_1 \; and \; C_3) \; and \; (not \; (C_6 \; and \; C_5)) \tag{4.19}$$

In Eqn(4.19) , killer cells will only be effective in the presence of lymphokines and this is true also for antibodies in Eqn(4.18). This enhances the viral proliferation by putting stricter criteria on the proper functioning of the killer cells and antibodies. Therefore, in these alternative equations, lymphokines play an important role. So the proliferation of suppressor cells, (which suppress lymphokines) are all the more critical in this set of interactions. The first set of viral interactions is chosen with probability, f, and the second set with probability $1 - f$. For $f \geq 0.8$, the viral population grows and then decays with a corresponding increase in helper cell population; for $f < 0.8$, we see viral domination. This result is not surprising when one considers the additional requirements imposed on killer cells and antibodies to do their job.

The eight cell model, P2 , incorporated two lines of defence, humoral and cell-mediated and two lines of attack, free antigen and viral-infected cells. As a model attempting to incorporate more than the usual number of cell types, it suffered a little from trying to include too much information. Lymphokines could only actively participate at the beginning of the simulation while suppressor cells could not function correctly. On the whole, it is interesting in what it tries to achieve; the unification of two important immune responses. Its weakness and strengths should be used as a starting point for future development of a CA model incorporating both humoral and cell-mediated immune responses.

4.3.5 PS2

A characteristic of HIV infection is the long latency period between initial infection and full blown AIDS. This latency period was explored by Pandey and Stauffer (1990) and was the first model to explore a specific facet of the disease. This model PS2, consisted of 5 cell types, the Macrophage, the Helper cell, the killer cell, interleukin and virus. Interleukin is the name of the cytokine produced by activated Helper cells. The authors assumed that macrophages would be present throughout the simulation and so developed no equation for them. This assumption could be interpreted as the supply of macrophages will always be replenished completely so HIV has no negative effect on this replenishment. It may be naive to assume that the population of macrophages does not decline with HIV infection. The following are the intra-site equations

$$V = H \; and \; (not \; (C)) \tag{4.20}$$

Here the virus will propagate only in the presence of Helper cells and only where killer cells are absent.

$$H = I \; or \; (not \; (V)) \tag{4.21}$$

The Helper cells will be triggered by interleukin but only in the absence of Viral cells

$$C = I \tag{4.22}$$

Interleukin triggers the production of killer cells and notice there is no suppression factor on their production.

$$I = H \tag{4.23}$$

Interleukin is triggered by Helper cells.

Eqns (4.20), (4.21), (4.22), (4.23) are an interesting set of equations, where Interleukin and Helper cells have a mutual dependence. The authors introduced a probability, p, that interleukin concentration will be equal to zero. This could represent defective killer cell receptors or more probably a lack of interleukin production caused by impaired Helper cell functioning. They found that for all non-zero values of p a fixed state of viral dominance was found. As the authors were investigating the latency period, they were interested in how many timesteps it took to reach this state of viral dominance. The timesteps were of the order $\frac{1}{p}$. This would mean that effective interleukin production and proper

Helper cell functioning contribute towards a lengthening of the latency period, while any malfunctioning of these elements cause a shorter latency period.

4.3.6 Other CA immune models

Above, we have presented a brief history of CA models of HIV infection. Other models exist such as that of Sieburg et al (1990). In this model, S1, the authors considered a cellular device machine (CDM). CDM was basically a cellular automaton, where the cell types change state based on their current state and some pattern element. The pattern elements represent the binding sites on a cell's receptors. The CDM was modelled on a two dimensional lattice. The authors classed the evolution of the infection as AIDS for a number of reasons. There was a lasting depletion of Helper cells and this depletion was preceded by a long latency period.

The IMMSIM model is a general model of the humoral arm of the immune system and is similar in construct to S1. It too, is a CA approach but it goes into greater microscopic detail. Antigens, T cells, B cells, APCs are all considered along with their receptors, peptides and epitopes. As stated it is a general model and as of yet has not investigated HIV infection specifically.

Another CA model to explore HIV was presented by Stauffer and Chowdary (1990). In this approach, SC1, the authors considered auto-immune response and normal immune response, along with HIV. The interactions between the cell types were governed by Boolean equations. In this case, the authors didn't utilise a lattice but opted instead for the mean-field approach. In this approach one considers a string of 1's and 0's, each binary digit in the string represents the concentration of a specific cell type and the length of the string is the number of cell types modelled. In this case, six cell types were considered HIV, suppressor cells, killer cells, helper cells, non-HIV specific antibody and B cells. If we take the above cell types in the order given then the binary string ,000000, represents absence of all cell types, while 100010, would represent high concentrations of HIV and non-HIV antibody and absence of all other cell types. In the mean-field approach as a lattice is not used, we have no nearest neighbour interactions. The Boolean equations alone govern each cell type's behaviour. As we have six cell types, each of which can be in one of two concentrations, we have 2^6 possible starting configurations. Each of these starting configurations is then iterated using the Boolean equations and iterated until a fixed point or a limit cycle

is found. The fixed points of the above model in the presence of HIV were a virgin state (000000), a low dose state (100000), a high dose state (100100) and non-responsive state (000100). The mean field approach is interesting as it explores the internal evolution of the equations without any outside influence i.e. nearest neighbour interactions. However, the influence of the nearest neighbours when modelling a system as connected as the immune response is imperative to include.

Mielke and Pandey (1998) developed a fuzzy CA model of HIV infections, MP1 . The authors considered ten different sets of immune response and at each site at each timestep one of the ten interactions sets was randomly chosen to govern the states of the cell types in that site. As the immune system responses can be in some sense random the authors considered this approach viable. However, the genesis of the interaction sets and associated probabilities are not explained in detail. When trying to mimic the random behaviour of a system, one always has to employ artificial means, here a deterministic interaction set is given an associated probability of occurring at a time-step. The use of artificial means runs the risk of the simulation seeming contrived, especially when the interaction sets are not well derived and are not representative of reality. The authors also considered mutation (discussed in Chapter 5). It may have been a better approach to investigate mutation with deterministic rules, where the effect of mutation would have been more explicit, rather than with randomised "fuzzy" interaction sets, where its effect can be clouded by the randomness of the interaction sets. Fuzzy simulations are best employed when known behaviours of the system are incorporated into interactions and not when guesses are being made about behaviours. When guesses are being made about system behaviour the risk is run of the interaction sets being tweaked until the desired results are achieved. On the other hand, if known system behaviour is incorporated into the interaction sets, then real analysis can be done on the results.

5 Current Discrete Models

5.1 Common Feature of MC and CA approaches

In this chapter, we introduce the basic formulations for a CA model (Stauffer & Pandey,1992, Pandey, 1998) and its Monte Carlo (MC) counterpart (Mannion et al 2000a). The Monte Carlo method is described in more detail in Chapter 6. Firstly, both MC and CA, share the same nearest-neighbour interactions. They also share the same intra-site interactions. Both also take into account mutation and mobility. All simulations take place on a 2-D or 3-D regular square lattice. Sites on the edges of the lattice will be missing some nearest neighbours, for this reason we use helical boundaries. Helical or periodical boundaries "wrap" the lattice around so that each site has the appropriate number of nearest neighbours. An example of this would be, on a 2-d lattice the site in the left hand bottom corner of the lattice, (see Figure 5.1, where it is shaded the darkest). This site is missing a southerly nearest neighbour and a nearest neighbour on its left-hand side. Using helical boundaries its mirror image on the top row of the lattice becomes its southerly nearest neighbour and the site in the bottom right corner of the lattice becomes its left nearest neighbour (these and its other nearest neighbours are shaded grey in Fig. 5.1).

5.2 Nearest-neighbour and Intra-Site Interactions

The nearest-neighbour interactions consist of a logical-ORing between the same cell types in their von Neumann neighbourhood (see Figure 5.2 for example). The code implements the nearest-neighbour interaction as follows; for each cell type, the concentration of that cell type in the n nearest neighbours are added together, this is then added to $n-1$ and divided by n. The following code (5.1), is taken from MC1(Mannion et al,2000). Where, im1 is the macrophage concentration, ih1, helper concentration, ic1, killer cell concentration and iv1, viral concentration, ic(k,i,j) is a site where i,j are the lattice co-ordinates (2-D lattice) and k the cell type, ib(i) determines the appropriate nearest-neighbour using helical boundaries.

(5.1)

```
im1 = (4 + ic(1,i,j) + ic(1, ib(i-1),j) + ic(1, ib(i+1),j) + ic(1, i,ib(j-1)) + ic(1,
i, ib(j+1)))/5
ih1 = (4 + ic(2,i,j) + ic(2,ib(i-1),j) + ic(2,ib(i+1),j) + ic(2,i,ib(j-1)) + ic(2,i,
ib(j+1)))/5
```

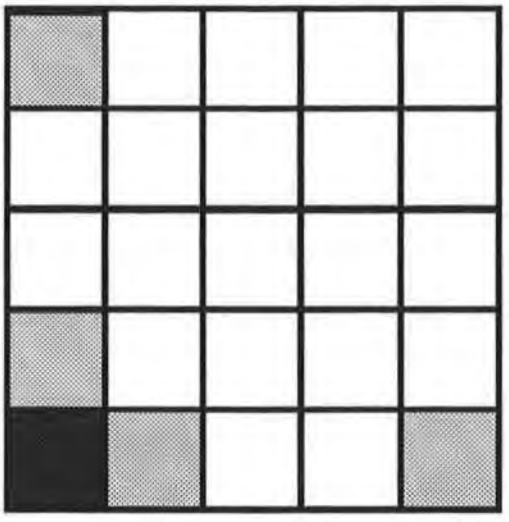


Figure 5.1 The left-hand corner site(shaded darkest) and its nearest neighbours(shaded grey) by use of helical boundaries

ic1 = (4 + ic(3,i,j) + ic(3,ib(i-1),j) + ic(3,ib(i+1),j) + ic(3,i,ib(j-1)) + ic(3,i, ib(j+1)))/5
iv1 = (4 + ic(4,i,j) + ic(4,ib(i-1),j) + ic(4,ib(i+1),j) + ic(4,i,ib(j-1)) + ic(4,i, ib(j+1)))/5

The inter-cell(intra-site) equations that make up our current simulation first appeared in a paper by Stauffer and Pandey, SP1, (1992). These equations are

$$M = M \ or \ V \tag{5.2}$$
$$H = (M \ or \ H) \ and \ (not \ (V)) \tag{5.3}$$
$$C = H \ and \ M \ and \ V \tag{5.4}$$
$$V = (V \ or \ M \ or \ H) \ and \ (not \ (C)) \tag{5.5}$$

This model also focuses on the cell-mediated arm of the immune system, dealing with Macrophages, Helper cells, Killer cells and Viral Infected cells. Macrophages, M, are the new cell type considered in this model compared with previous models PS1, P1 and P2 . Their role is to act as an intermediary, as they propagate both immune and viral cells, and this enables the model to evolve in a way a 3-cell model could not. Eqn(5.2) shows that the Macrophages are self-propagating or can be stimulated by a viral-infected cell. So, if a V appeared at a site, it would induce an M at the site. This can be explained as macrophages are the

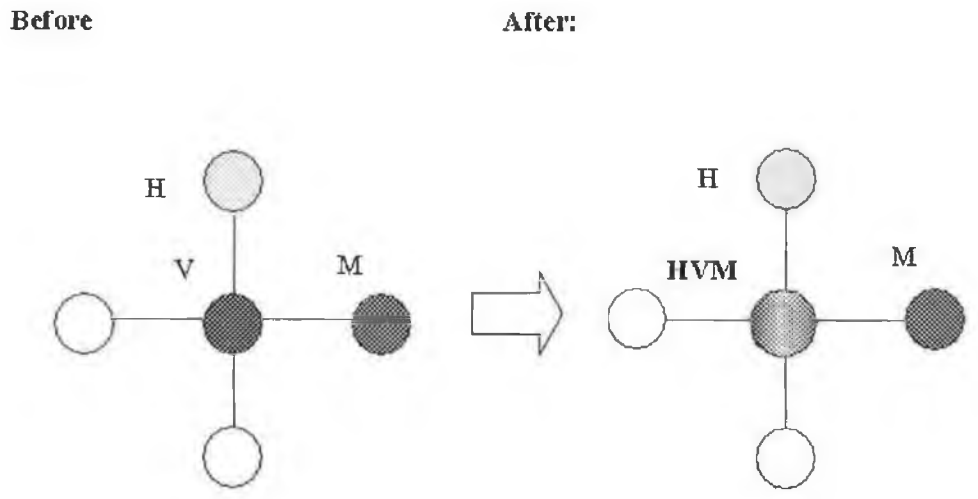


Figure 5.2 An example of the nearest-neighbour mechanism

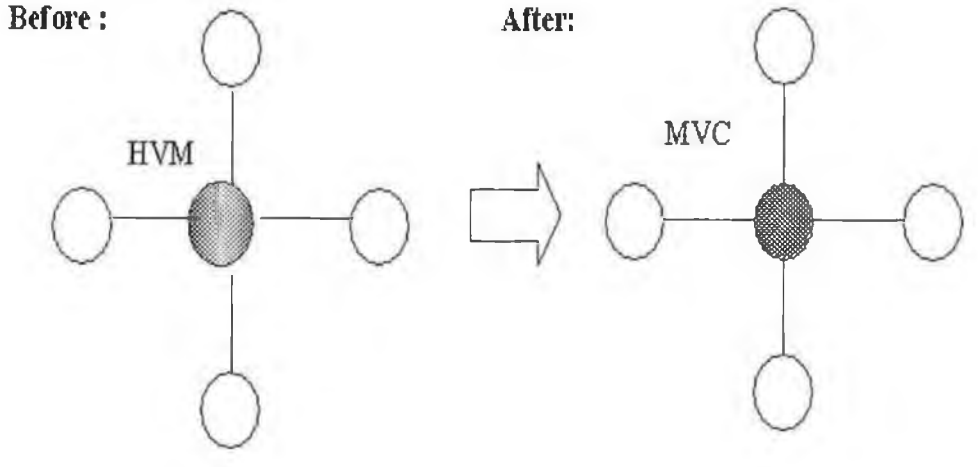


Figure 5.3 An example of the intra-site(inter-cell) interaction

first line of defence. Helper cells are self-propagating or can be induced by an M. Macrophages signal to helper cells to proliferate. V kills Helper cells. The Killer cells, C, are only stimulated when an activated M, i.e. an M which has encountered a V, has induced an H response. An H and an M contribute towards V's proliferation, as both cell types can be virally infected. V cells are also self-propagating. Figure 5.3 shows an example of a intra-site (inter-cell) interaction. Appendix 1 contains the code for a simulation with these intra-site (inter-cell) interactions. An update of a site in this model consists of the nearest-neighbour interactions followed by the intra-site interactions.

M	**H**	**C**	**V**
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	1

Figure 5.4 All possible states for a given site with the valid configurations shaded in grey.

Due to the fact that we are dealing with four cell types in 2 possible states, each site can be in 16 different configurations see Figure 5.4. Due to the intra-site (inter-cell equations){(5.2)-(5.5)} only 4 of these configurations are valid, as these Boolean equations will not give rise to any other states. These valid states are shaded in grey in Figure 5.2. Each of the valid configurations (1000),(1001),(1011),(1101)} are reversible in the sense that each has a distinct

predecessor,

(1000)->(1101)
(1001)->(1001)
(1011)->(1000)
(1101)->(1011)

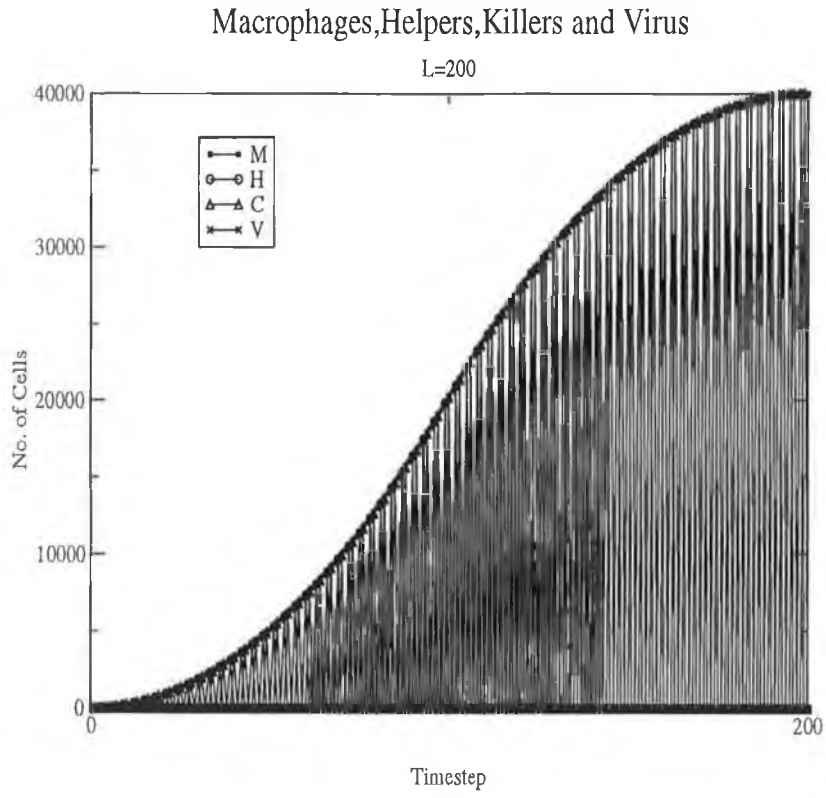


Figure 5.5 Macrophages, Helper,Killer and Viral cells, indistinguishable because of the large oscillations

As nearest-neighbours interactions precede intra-site interactions, they ensure thatthe states of the site are not restricted to a cycle. The nearest-neighbour interactions also cause the system as a whole to be irreversible; given any site at a particular timestep it is impossible to determine the states of the cell types at the site at the previous timestep. This irreversibility is characteristic of a

self-organising system (Wolfram, 1986). Irreversibility can cause a system to evolve from a disordered initial state to an ordered state.

Typical results of such a CA model, simulated on a 2-d lattice of length $l = 200$ and with initial random configuration of one of each cell type, are shown in figure 5.5 . One can see large oscillations in the populations of Helper, Killer and Viral Infected cells while the macrophages stay constant at system size (40,000 sites). As the lattice reaches saturation point, with every site on the lattice containing at least one cell type, (this occurs at timestep ≈200 in Fig 5.5)

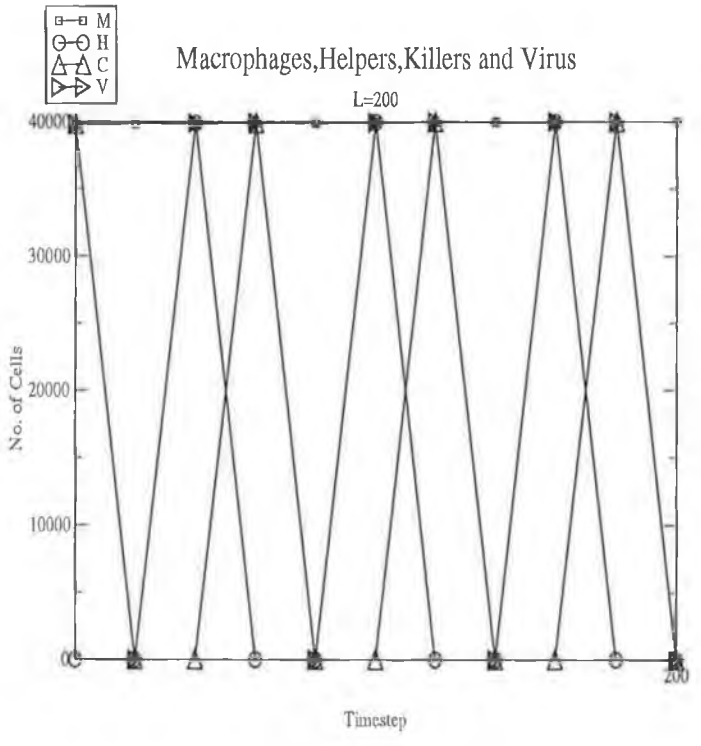


Figure 5.6 This is Figure 5.5 zoomed in on timesteps between 190 and 200 to show oscillations

a cycle is evident. The cycle occurs when the lattice is at saturation point so the nearest-neighbour interactions no longer have an effect on the intra-site interactions. interactions no longer have an effect. The cycle is as follows, all the sites being in state (1000), (a Macrophage at every site), then state (1101),

(M,H,V at every site), followed by state (1011), (M, C and V at every site), and back to the start of the cycle. Figure 5.6 shows this cycle as it shows the cell concentrations between timesteps 190 and 200. This cycle continues indefinitely. Because of this cycle it is impossible to determine a definite outcome for the simulation, as the cycle goes from immune dominance to viral dominance. This cycle is due to the completely deterministic nature of the Boolean equations. Also each cell type is limited to one of two concentrations, low (0) and high(1) and as no intermediate concentration exists this leads to the extremes in the oscillations.

5.3 Mutation

HIV has the highest mutation rate of all known retroviruses (Nowak & Michael, 1991), see section 2.2. Many hypotheses suggest that this mutating rate is one of the main reasons that HIV is so deft at evading the immune response (Nowak & Michael, 1991). It is therefore, vital that a model of HIV should include some mechanism for introducing mutation. HIV mutates because the process of reverse transcriptase is extremely error prone. It is generally thought, (Blakelee 1996), that one mutation occurs every time a DNA copy is made of the viral RNA genome. That is, every time that the virus enters a host cell and integrates itself into the DNA of the host cell. Although mutations will change the epitopes of the virus, (i.e. change the pattern, by which the immune system recognises them), not all mutations will be advantageous to the virus, as the immune system may already be primed for that mutation.

To incorporate mutation, a mutation probability, P_{mut}, is introduced, for the code implementation see Appendix 1.

Defn: P_{mut} , is the probability that the virus has mutated and that this mutation is advantageous to the virus.

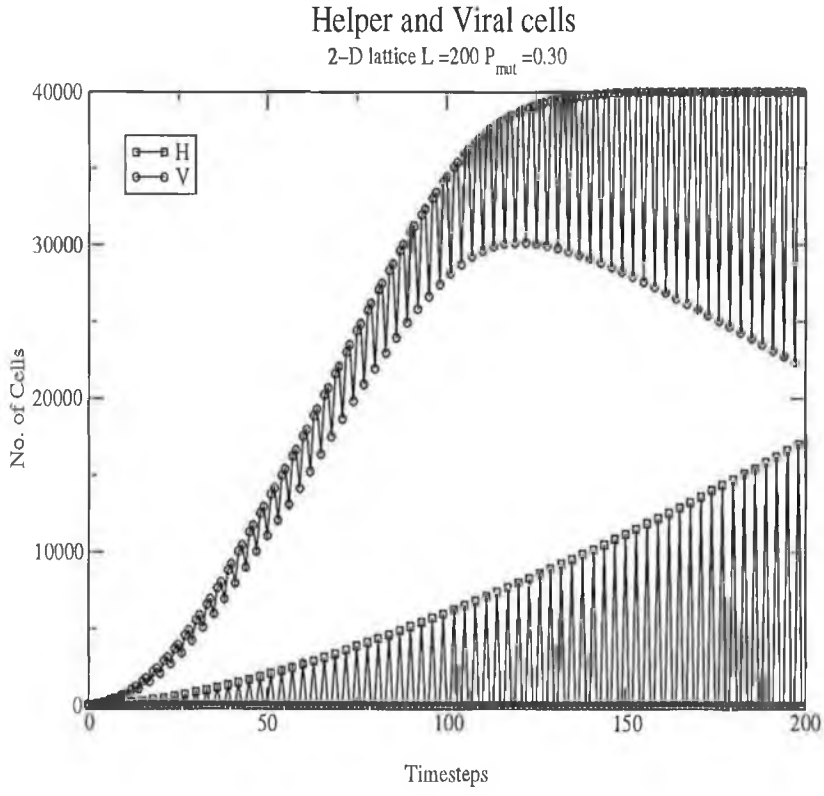


Figure 5.7 With mutation introduced, $P_{mut} = 0.30$ we see a reduction in the oscillations and in the Helper population

An accurate estimate of P_{mut} is impossible, as the mutation rate changes from individual to individual with HIV (Walker &Basgoz, 1998). This of course raises a question with respect to our modelling of HIV infection, as we are concerned with a general model which utilises commonalities across people. Therefore investigating a range of P_{mut} and its effect on the infection is important.

Mutation is incorporated as follows as in Pandey (1998): with probability P_{mut}, V is set to zero in Eqns. 5.2 and 5.4. The CA model is now a probabilistic CA or a stochastic CA. Thus if V is mutated, an M or a C cannot recognise it and so elicit an effective response, while an H can still be killed by a mutated V, even if it does not recognise it. With mutation, the state (1001), an M and a V has more than one predecessor namely

(1001)->(1001)
(1011)->(1001)
(1101)->(1001)

This means now that the inter-cell (intra-site) interactions like the nearest-neighbour interactions are irreversible and this increases the irreversibility of the system as a whole.

Introducing mutation into the CA model therefore affects the population of the immune cells, as can be seen in Figure 5.7, where P_{mut}= 0.30, s.t. the maximum level of Helper cells is reduced and cell population oscillations are somewhat dampened. Due to large oscillations in the cell populations it is difficult to pinpoint exactly at what level of P_{mut} the crossover from immune to viral dominance occurs.

5.4 Mobility

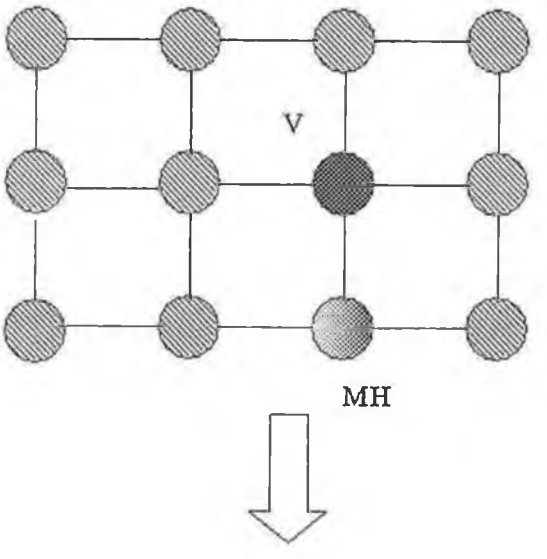
Nearest-neighbour interactions result in intrinsic mobility in the model. Nearest-neighbour interactions cause a cell type to fan out from its central site to all its nearest neighbours. More explicitly, mobility has been considered by introducing a probabilistic parameter, P_{mob}, which is governed by certain rules. This mobility can be thought of as chemotaxis. Chemotaxis is the directed migration of cells, in chemotaxis cells move towards other cells which release certain cytokines. Therefore this explicit mobility we incorporate into our models can be it can be thought of as chemotaxis, as the mobility is *directed.* This type of mobility was introduced by Pandey (1998). The rules governing the mobility parameter P_{mob} (Pandey 1998) are

1. For any cell type to move, it must not already be present at the intended site .

2. For a macrophage or cytotoxic cell to move to a neighbouring site, a viral infected cell must be present at the intended site.

3. For a viral infected cell to move, a macrophage or a helper cell must be at the target site.

Before Mobility



After

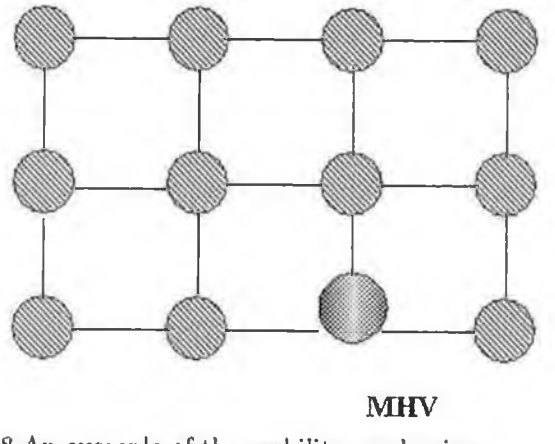


Figure 5.8 An example of the mobility mechanism

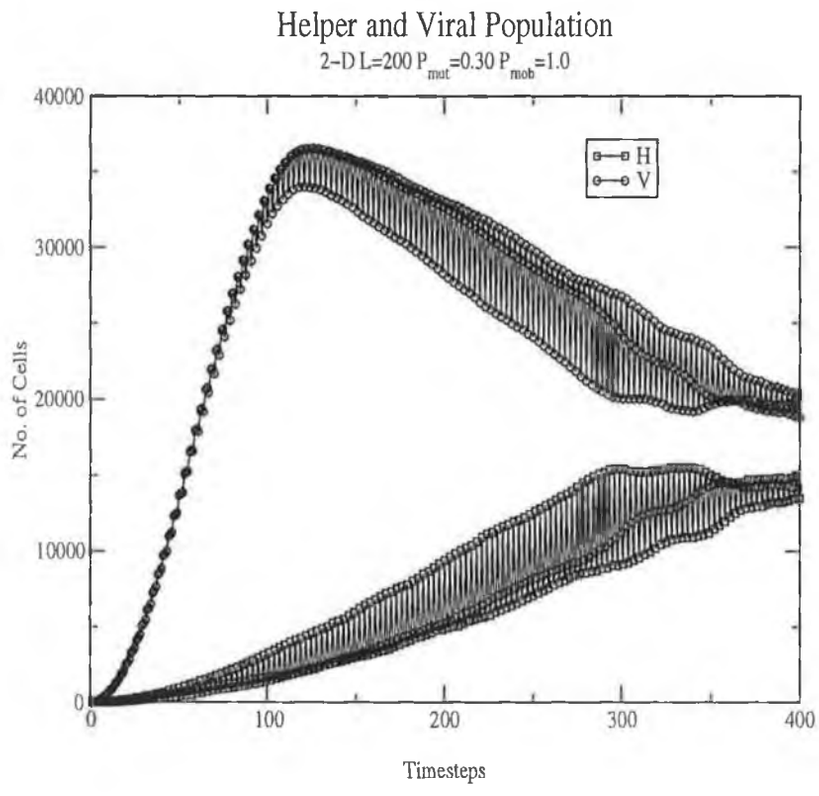


Figure 5.9 We see with extreme mobility, $P_{mob} = 1$ that the cell populations oscillations die down

See Figure 5.8 for an example of the mobility mechanism and for code implementation see Appendix 1.

Mobility dampens the oscillations, (see Figure 5.9), in each cell-type's population, (excluding M which are always at system size). This is due to mobility contributing randomness to what was a previously deterministic system in most cases. Also, the rules governing mobility are such that the cell types are being better employed. The immune system cells, M and C only move if the virus is present at the neighbouring site, i.e. they will only move in order to fight against the virus instead of randomly moving to any site. Also the virus will only move if a M or H is present, meaning the virus will only move if there are cells for it to infect and will stay put otherwise.

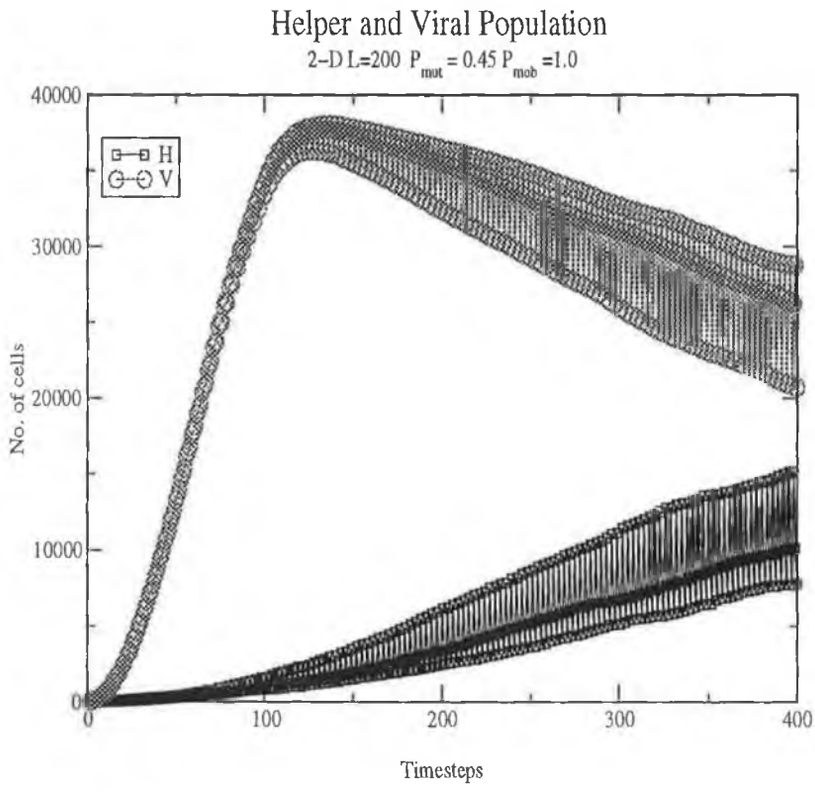


Fig 5.9 We see the virsus beginning to dominate for $P_{mut} = 0.45$ and notice oscillation getting larger eventhough $P_{mob} = 1.0$

Given that mobility dampens oscillations, it makes it easier to track at what level of P_{mut} , the transition from an immuno-dominant system to an immuno-deficient system takes place. With extreme mobility ($P_{mob} = 1$) the transition occurs at $\approx P_{mut}$=0.40, see Fig 5.9. Also, with extreme mobility an increasing P_{mut} results in increasing oscillations in cell populations, see Fig 5.9. Therefore the effect of mutation dominates over the effect of mobility.

In this Chapter, we provided a brief overview of the CA version of our model. The CA model incorporates mutation and mobility using probabilistic parameters. Increasing mutation weakens the immune response. Mobility brings directed migration to the cell types. Even with the dampened oscillations in the cell populations, it is still difficult to define clearly any phase transition between

immune dominance and viral dominance. It is also difficult to determine the critical value of P_{mut} below which we find immune dominance and above which the virus dominates. In the next Chapter we discuss the Monte Carlo approach and will see that this dampens the oscillations sufficiently to allow us to explore cellular dynamics in greater detail.

6 Chapter 6 Monte Carlo Methods

6.1 Background

The Monte Carlo method uses random numbers to establish approximate solutions to various problems. The problem may be intrinsically probabilistic but the Monte Carlo method is also used for problems that are not. The Monte Carlo method was principally developed by von Neumann, Ulam and Metropolis while working on the Manhattan project during World War II, where problems like the random neutron diffusion in fissile material were of concern to them (Hammersley and Handscomb, 1964, Jain,1992). They simulated this randomness by probabilistic methods. The method is named after Monte Carlo in Monaco, which is synonymous with gambling, because of the similarity between games of chance and statistical simulation. The term Monte Carlo method is very general, so any method that utilises random numbers in some way is termed a Monte Carlo method. Prior to von Neumann et al. pioneering work, Monte Carlo methods were used in isolated cases. The most popular of these involved the determination of π by throwing a needle onto a board with parallel lines and observing the number of intersections between the line and the needle, this was termed Buffon's needle (Jain,1992). By 1950, Fermi, Metropolis and Ulam derived estimates for the eigenvalues of the Schrödinger Equations using the Monte Carlo method (Hammersley and Handscomb, 1964). The Ising model, a simple model of ferromagnetism, for which no analytical 3-D solution has been found, was solved numerically in 3-D by an algorithm by Metropolis using the Monte Carlo technique (Hammersley and Handscomb, 1964, Jain, 1992). The advent of computing led to the immense popularity of Monte Carlo method as a simulation tool, with applications far and wide ranging from stock market predictions (Fox, 1999) to modelling subnuclear processes in high energy physics (Jadach et al, 1992).

Monte Carlo simulations are experiments using random numbers. The random numbers may be used to solve a probabilistic or deterministic problem. A simple probabilistic experiment might involve the generation of random numbers to simulate some random physical process, the random numbers would typically generated be in such a way that they directly emulate this random behaviour. An example of this would be stock markets forecasting, where a simple Monte Carlo model may look at the last 36 months of stock price to ascertain associated probabilities for percentage increases and decreases of the stock price. These

probabilities could then be used in a Monte Carlo simulation of the stock price fluctuations for the coming 6 months. A random number,z, ($0 \leq z < 1$) would be generated and checked against a associated probability, P, e.g P = probability of stock price increasing 10% , if $z < P$, then the stock price would increase 10% if $z \geq P$ then it would not. Deterministic problems are not as intuitive to model by the Monte Carlo method as they have no obvious random behaviour. However, sometimes the theory of the deterministic problem reveals an underlying structure that is analogous with some known random process. Monte Carlo experiments can then be performed and the problem solved numerically.

It is interesting to note that the founding fathers of CA methods, von Neumann and Ulam, were also fundamental in the development of Monte Carlo methods. Monte Carlo methods much like CA methods can be contrasted with the traditional methods of problem solving such as partial differential equations. Often a theory is too general or too abstract to allow for numerical interpretation or the formation of equations; in these types of situations CA and MC methods come to the fore. These methods simulate physical processes *directly* so that there is no need to establish differential equations to describe the behaviour of the system. For instance if behaviour in some system is observed as being random it is better to emulate this using some random number generator rather than formulating equations to feign randomness. Likewise, if modelling the immune system by differential equations, parameters such as clearing-rate, renewal rate etc. have to be determined. Many of these parameters are at best educated guesses. With CA such parameters do not have to be determined when the model is formulated; rather than determining the system they are determined *from* the system . This is one of the great advantages of using CA methods. Rather than fixing such parameters at the start they simply come into existence once the simulation is begun and evolve with it.

6.2 Synchronous and Asynchronous Updating

6.2.1 Synchronous

In a CA model, sites are updated synchronously, which means that at each timestep, all interactions are taken to happen simultaneously at all sites, i.e. in strict parallel. No content of a site is altered until its is checked against all its neighbours. This is usually achieved by the programmer maintaining two copies of the lattice, an "old" copy and a "new" copy. When a site is being updated the

program looks at the states of the nearest neighbours of the sites in the "old" lattice and the updated state of this site is then put in the "new" lattice. After each site has been updated the "new" lattice becomes the "old" lattice. With synchronous updating, in a single timestep the order in which sites are updated makes no difference. Synchronous updating has its advantages, the outcome of the system is deterministic and therefore controlled but it is also a very idealised view of a system. As can be seen from Figure 5.5 in Chapter 5, synchronous updating of our model leads to a uniform state, throughout the lattice, with the Boolean equations causing a cycle of uniform states. Synchronous updating causes the system to overshoot, either complete viral dominance or complete immune dominance and no intermediate states; this overshooting occurs for two mains reasons.

1. The concentrations of the cell types can only ever be high,1, or low, 0. No gradient or different levels of concentration are allowed. Because of this Boolean equations have to be employed to describe the system, which leaves no room for middleground.

2. The timelag for updating is too long. Each site has to wait until every other site has been updated before it can update, as is the nature of synchronous updating. This does not allow for any site to have an "advantage" over any other, but also is not representative of physical reality where such "advantages" are commonplace.

6.2.2 Asynchronous

For these reasons we consider an alternative, i.e use of asynchronous updating. In this approach, a site is chosen at random and then *immediately* updated and another site is then chosen at random. The software for asynchronous updating, therefore only requires one copy of the lattice, a "current" one, which reduces the memory overhead of the program. The order in which the sites are chosen can alter the outcome of the simulation, (see Figures 6.1 and 6.2). In each figure, the starting configuration is identical, the only difference is that in Figure 6.1 the left central neighbour is chosen for updating first. Updating consists of the nearest-neighbour interaction, followed by interactions (5.2)-(5.5), (see Chapter 5). This leaves V, an M and a C in the left central site, and an H and an M in the right central site. In the second figure, 6.2, the left central site is chosen

first which results in nothing in the left central site and an M, and H and a V in the right central site.

We investigated asynchronous updating for a number of reasons. With asynchronous updating one does not get a dominant uniform state, but rather pockets of viral and immune dominance co-existing. Therefore, while the concentrations of the cell types can still only be in one of two concentrations, the overall macroscopic picture has different gradients of immune and viral dominance. Asynchronous updating massages the extremes found in synchronous updating towards an average. With synchronous updating, due to nearest-neighbour interactions and intra-site interactions, sites can only can exist in certain states, these being, (1001), an M and a V, (1101), an M, an H and a V and (1011), an M a C and a V. Asynchronous updating increases the number of states that can exist. Along with the above states, the following can exist, (1000), an M, (1010),an M and a C. This increases the interaction between the cell types and also the realism of the model. Due to asynchronous updating the simulation does not get locked into a cycle of uniform states and therefore intermediate states, between viral and immune control, can exist. The growth patterns of cell types can also be distinguished. Figure 6.3 shows such an intermediate state. It is intermediate states of the simulation which are of interest, as one can explore the critical point of transition between immune dominance and immune deficiency. Critical points are of interest in physical systems as they are the point at which the system changes phase i.e. a phase transition of the system. A common example of a phase transition, is that of the density of water when changing from a liquid to a gas, the critical point would be $\approx 100°$ celsius. Critical points in theoretical immunology are key to understanding the mechanics behind how the immune system defends against an invader and what causes the "phase transisition" from a healthy immune system to a weakened one. Once these critical points have been established they can be used as a gauge for how effective a theoretical treatment could be, i.e. does the critical point increase in favour of the immune system or decrease and weaken it or is the phase transistion "smoothened" ?

Asynchronous Monte Carlo updating provides a probabilistic aspect to the system, but rather than changing the core equations this probabilistic aspect is in the updating of the sites. With synchronous updating the core Boolean intra-site equations dictate the way the system behaves while with asynchronous updating these intra-site equations are the way the system "tends" to behave and so allows

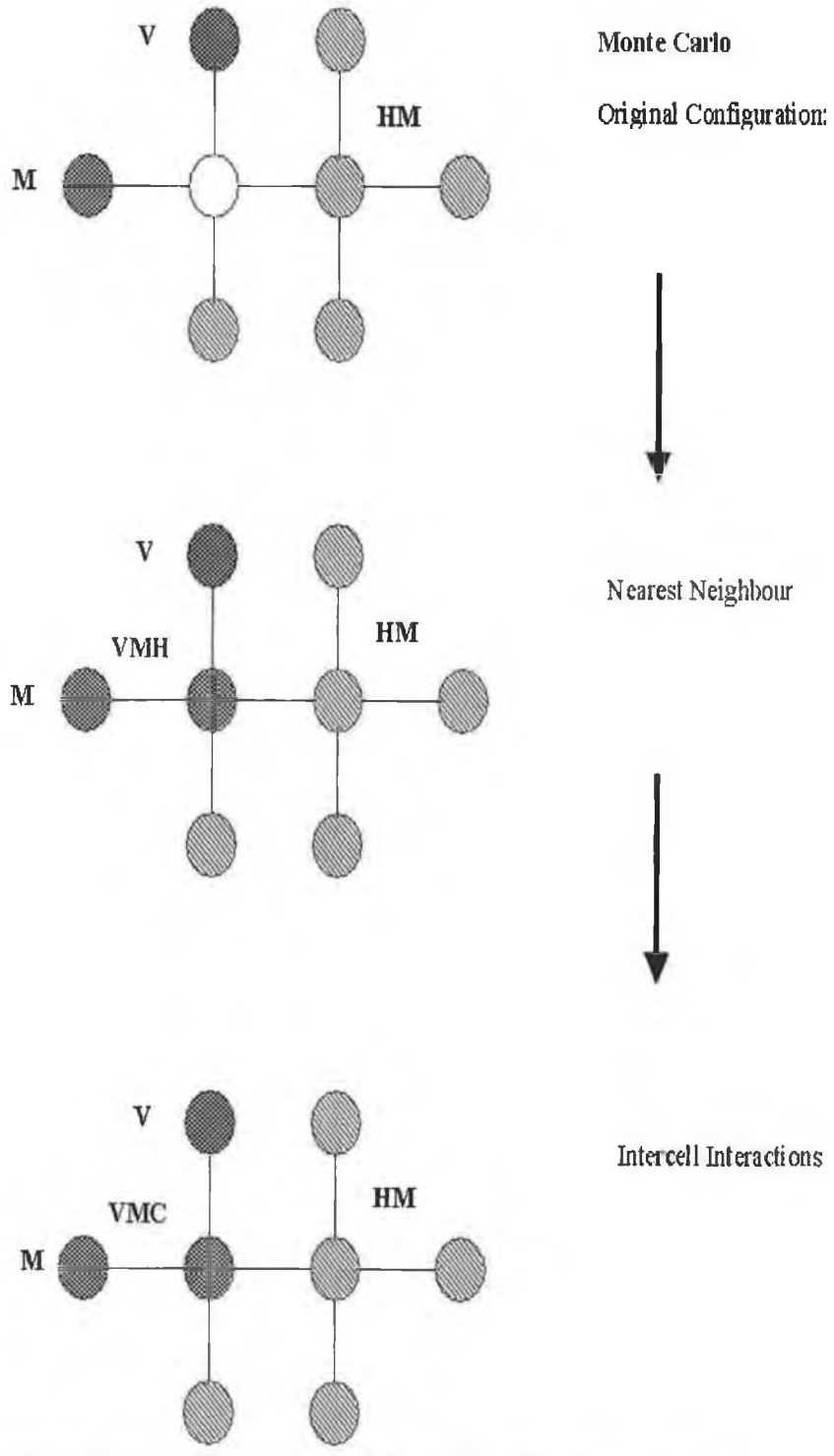


Fig 6.1 Outcome if the left central site gets chosen for updating

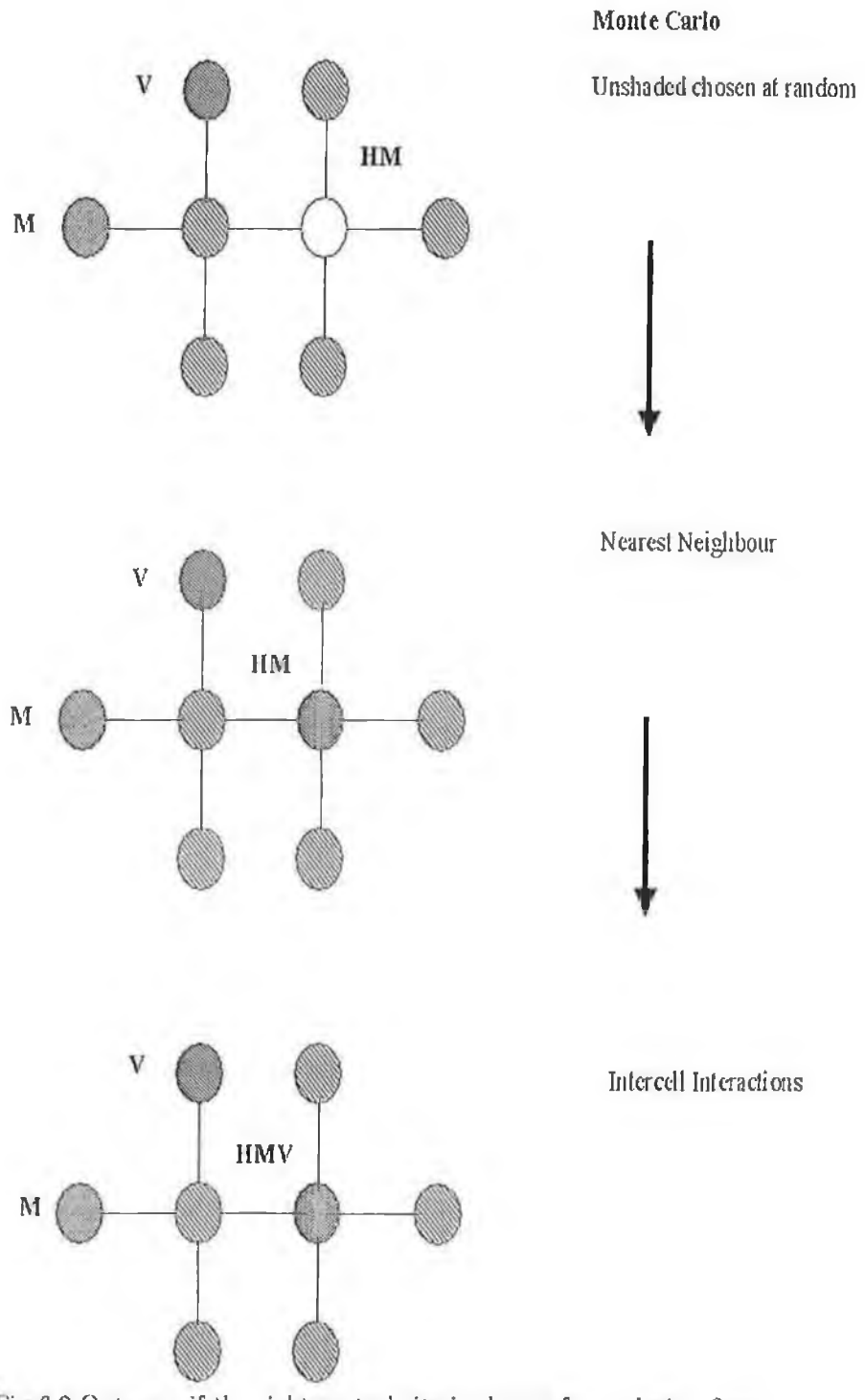


Fig 6.2 Outcome if the right central site is chosen for updating first

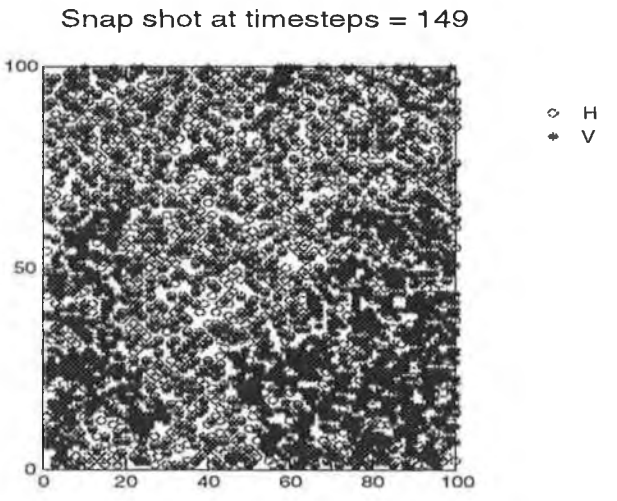


Figure 6.3 Snapshot of Helper and Virus on a 100x100 lattice with $P_{mut} = 0.90$ after 149 MC timesteps

a certain degree of "fuzziness". Also, the timelag in the updating is no longer uniform and deterministic, rather it is distributed probabilistically throughout the lattice at each timestep. So in an effort to mimic a physical process, we are replacing what we do not know with randomness. We can justify our use of Monte Carlo updating on either of two grounds

1. Randomness plays a part in the order in which interactions occur.

2. The order is deterministic but we do not know enough about the system (we have a model of only four cell types) to apply the rules deterministically so we use randomness to replace our lack of knowledge.

At each timestep, N random numbers are generated, where N is the number of sites on the lattice. Each of these random numbers corresponds to a site on the lattice, where some sites can be chosen more than once during a timestep and others may not be chosen at all. In MC models many runs of a simulation are performed so that statistical fluctuations are accounted for, the results reported are an average of the results for each run. Even though the updating is asynchronous in the case where two or more sites are updated before any of their nearest neighbours, the interactions at those sites can be thought of as having

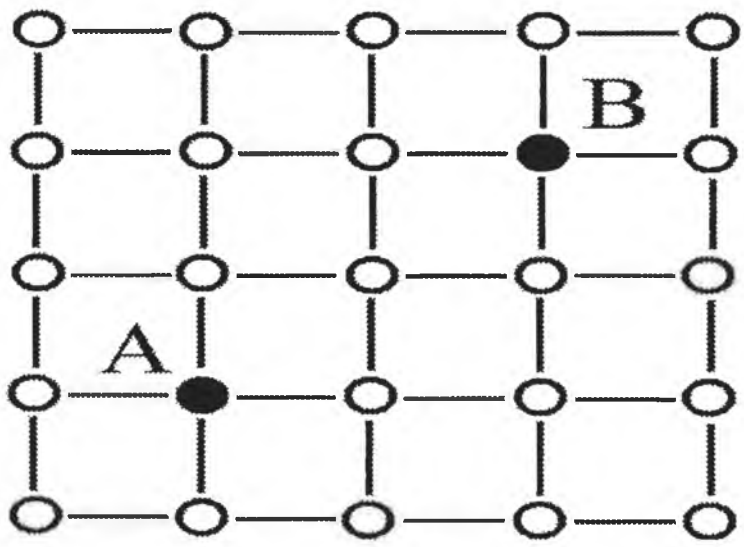


Figure 6.4 If site A and B were updating asynchronously before their nearest neighbours than the interactions at A and B can be said to have happened simultaneously

happened simultaneously. This is because the nearest-neighbours propagate the results of the asynchronous updating throughout the lattice. If a site, a, is updated before its nearest-neighbours and another site, b, is updated before its nearest-neighbours then the interactions at a and b can be considered as having happened simultaneously (see Figure 6.4). With asynchronous updating it is very important to ensure that the sites are chosen in a truly random fashion, with no obvious or repeating pattern in the sites chosen. This means that it is essential to have "good" random numbers .

6.3 Pseudo-Random Numbers

Good Monte Carlo simulation relies on a quality pseudo-random number generator. A pseudo-random number generator is one where a "random" number is generated via an algorithm which is carefully constructed so that the output seems genuinely random. A good random number generator should satisfy a number of basic requirements:

1. A long cycle so that its does not repeat itself too often. The length of the cycle should be longer than the total number of random numbers that

need to be generated.

2. There should be no distinguishable mode i.e no one number should occur more frequently than the other numbers.

The random number generator used for all our MC models can be found in Stauffer(1988). This generator falls into the general category of Multiplicative Linear Congrential Generators and this type of generator is one of the oldest and most widely used. The following is the algorithm for generating a pseudo-random number using this generator .

*ibm = ibm * 16807*
abm = 0.5(float(ibm)+2147483647.)*
rand = abm/2147483647.

ibm is generated originally from a seed (e.g *iseed* =7893). The first time a random number is required we use this algorithm to generate it with ibm=7893, the next time we require a random number we use the same algorithm but this time $ibm = 7893 * 16807$ and so on . This is a good pseudo-random number generator; as can be seen from Figure 6.5 there appears to be no distinguishable mode and as the graph changes shape for the different number of runs, the cycle would appear to be quite long. If the cycle was short, then the overall pattern of the graph would not change from 5 million runs to 10 million runs. The maximum length of the cycle is *2147483647-1.*

6.4 Hamiltonian

The Monte Carlo method has an associated "Hamilitonian" (Stauffer, 1988). In a generic system that uses the Monte Carlo method to change states, a rule is needed to govern these changes. This rule can take the form of an equation based on a key attribute of the system or a phenomenological aspect of the system. This rule is referred to as a Hamiltonian and in effect determines the nature of the simulation. A Hamiltonian, in the strictly physical interpretation, is a function which evaluates the energy of a system in a particular state. We use the term here to refer to a function which determines the "energy" of a system and given that information decides whether or not to alter the state of the system. The interpretation of "energy" depends on the system being modelled. An example of a quantitative Hamiltonian is used in the Ising model. The Ising model mimics the behaviour of interacting molecules of a liquid or dense gas

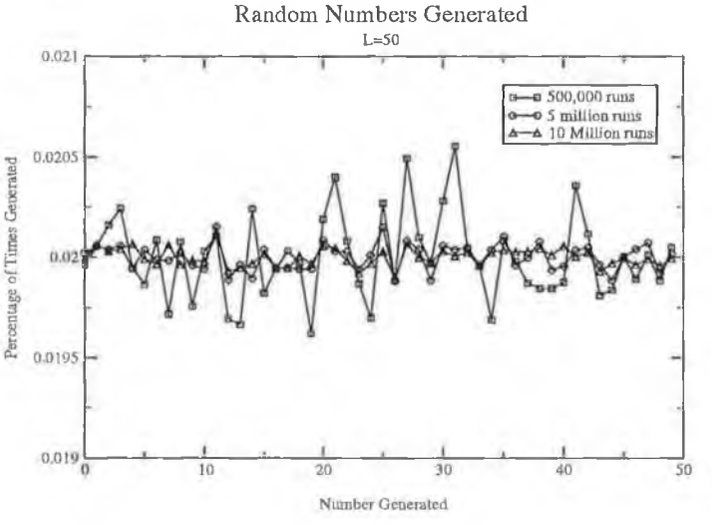


Figure 6.5 Random number generated versus frequency

confined to a chamber. Metropolis' algorithm for the Ising model (Hammersley and Handscomb, 1964, Stauffer, 1988) has $e^{-\Delta E/k_B T}$ as its Hamiltonian, where ΔE is an energy change and k_B is Boltzmann's constant and T is absolute temperature. This simulation is based on a lattice (which represents the chamber) with each site on the lattice having two possible states, 1, occupied and -1, empty. A state in this model is referred to as a "spin". An "up" spin refers to a molecule occupying a site and a "down" spin represents a vacuum at the site. The algorithm for the simulation involves calculating the energy change ΔE and then generating a random number, z, $(0 \leq z < 1)$, and z is then compared with the Hamiltonian. If the random number, z, is less than the Hamiltonian, the spin is flipped otherwise it remains the same.

In our simulation it is impossible to establish a quantitative Hamiltonian. What determines the state of a site at a given timestep is the nearest-neighbour interactions followed by the intra-site interactions. To draw an analogy with the Ising model, the energy change here is the interactions occurring at a site. The state change of a given cell type is a function of the interactions themselves. So, it is better to think of the Hamiltonian in our simulation as being a phenomenological one i.e. it is representative of a phenomenon occurring in the system. This phenomenon is the interaction of HIV and the immune system as we have described in Eqn. (5.1-5.5) and these determine the states of the

cell types. If we investigate this Hamiltonian, we see that the state of a site is very susceptible to change, that is given a site on the lattice the state of the site is more than likely to change after the nearest-neighbour followed by the intra-site interaction. One of the reasons for adopting the asynchronous updating approach is due to the low threshold of the Hamiltonian. This results in large oscillations in cell populations with synchronous updating. As stated previously these large oscillations hide the intermediate states that are visible with asynchronous updating. With synchronous updating all sites are engaged in simultaneous exchange of "energy" (the results of the interactions) across the lattice within a single time-step. With asynchronous updating, the "energy" is not exchanged simultaneously but rather dissipates through the lattice according to the order in which the sites are chosen for updating.

In this chapter, we have discussed the history and theory behind the Monte Carlo method. Any Monte Carlo simulation requires a good pseudo-random number generator. We presented the multiplicative Linear Congruential Generator we utilised in our simulation and we discussed its merits. Asynchronous updating was introduced and its advantages over synchronous updating were discussed. The phenomenological Hamiltonian of the system was also explored .

7 Viral and Immune Dynamics

Asynchronous updating can present underlying features of a model that are hidden by the extremity of synchronous updating. Therefore, asynchronous updating complements rather than contradicts the results found with synchronous updating (see results and figures presented in Chapter 5). One of the features of the simulation that is hidden under the extreme oscillations of the cell populations, which occur with synchronous updating, is the growth pattern of cell types. Asynchronous updating allows the model to be in an intermediate state, poised somewhere between immune dominance and deficiency. This enables the investigation and definition of, phase transitions and critical points where the virus begins to get the upper hand on the immune system. These occur when the system changes from a state of immune dominance to immune deficiency and are essential when exploring the cellular dynamics of HIV infection.

7.1 Cellular Dynamics

The main effect of asynchronous updating on our model, MC1, (Mannion et al; 2000, see Appendix 2 for code implementation) is that there are no longer large oscillations and cell populations attain intermediate values. These allows us to investigate the competition between the immune cells and virus. The growth patterns of the cell types are clearly visible, (see Figure 7.1). Helper cells grow to an equilibrium value. The viral growth pattern sees the virus population reach a peak and then decrease to an equilibrium level. Both equilibrium levels of Helper and Viral cell populations are a function of the corresponding mutation probability, P_{mut}.

It is also easy to approximate these growth patterns by mathematical means. In MC1 we consider a 2-dimensional lattice, (100x100, lattice with 10,000 sites), with nearest-neighbour and intra-site interactions as described in Chapter 5, subsection 5.2. The only difference between this and our CA model is *asynchronous* updating. An update in MC1 comprises of nearest-neighbour interactions followed by intra-site interactions which then results in the *immediate* updating of that site. At the beginning of the simulation the initial population of each cell type is 1, with this single cell being randomly spatially allocated a single site.

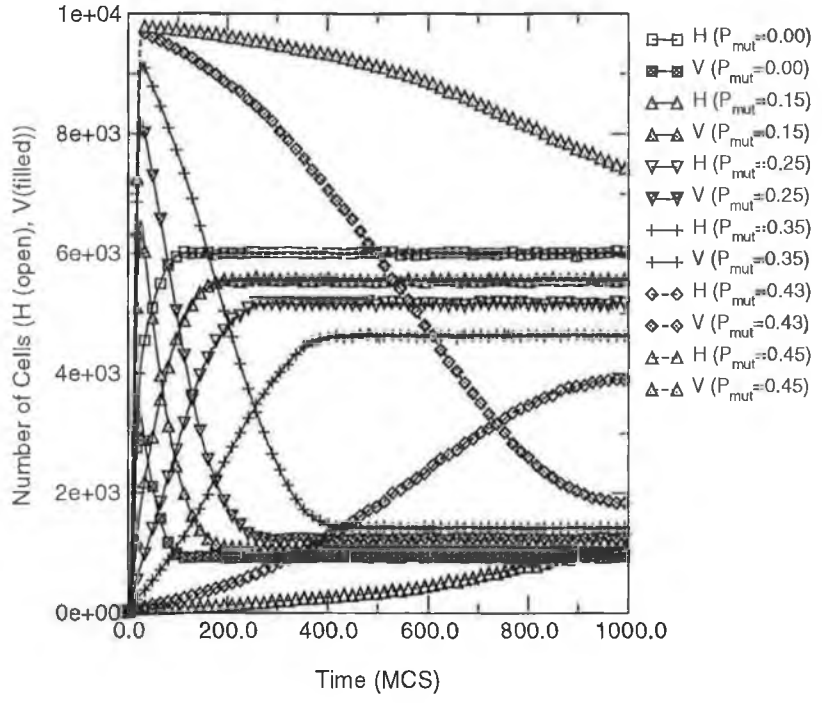


Fig 7.1 MC1 model: Helper and Viral growth patterns for various P_{mut}

7.1.1 Viral Growth

Focusing on Figure 7.2 , one can see that the growth of the viral populations has two phases. The viral population grows to an initial peak; this represents the initial infection catching the immune system unawares. As the immune system mounts its response we then observe a decay in the viral population to a very slightly oscillating equilibrium. This decay is not as dramatic as $P_{mut} \rightarrow P_{crit}$.

Defn: P_{crit} , is the least value of P_{mut} for which viral dominance occurs. This is the critical point for the transition between the phase of immune dominance to the phase of viral dominance. In MC1 P_{crit} was found to be $P_{crit} \approx 0.44$.

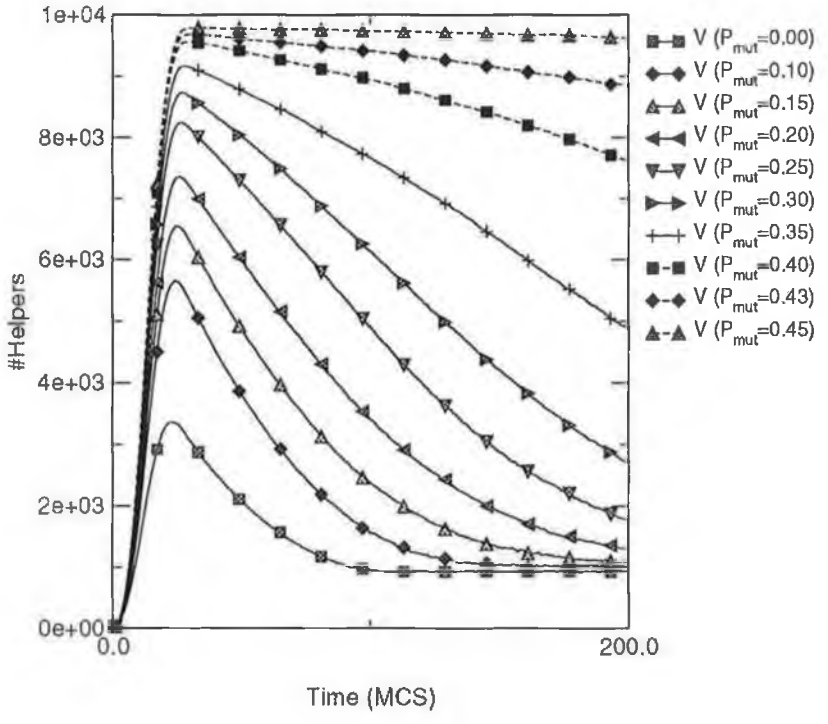


Fig 7.2 MC1 model:Viral growth patterns for various P_{mut}

Exponential Decline of Viral Cells

If we investigate the pattern of decline of the viral population after the initial viral peak we find that an exponential regression line fits the viral data well for $P_{mut} < P_{crit}$

$$V = V_0 e^{-m} \tag{7.1}$$

with V_0 being the peak viral population before decline. The slope, m, is dependent upon P_{mut}, and ranges from -0.018 for $P_{mut} = 0$ to -0.005 for $P_{mut} = 0.35$ (see Figure 7.3). This would mean with that a low mutation probability the virus decays quicker due to strong immune defences and as the P_{mut} increases, the immune system responses are weaker causing slower viral decay.

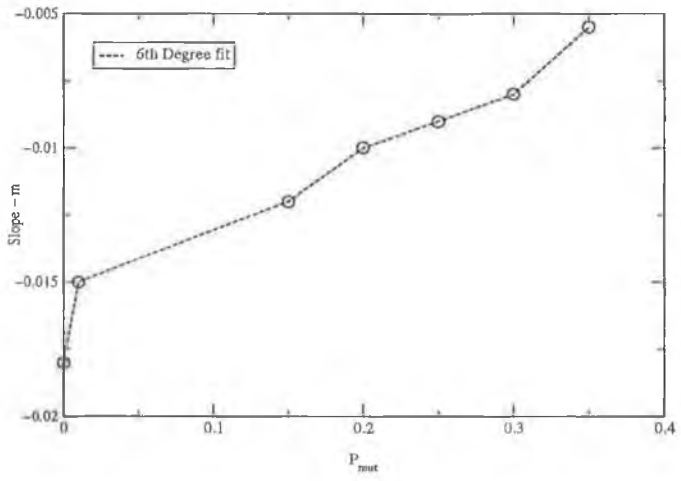


Figure 7.3 P_{mut} versus m, the slope of viral decay , the above is fitted to a 6^{th} degree polynomial

For $P_{mut} \rightarrow P_{crit}$ the exponential approximation becomes less exact. Ho et al (1995) having monitoring 20 HIV-infected individuals, found viral decay to follow an exponential decline, with exponents ranging from m = -.53 to m = -0.21, depending on the mutation probability of the virus. The patients were treated with a protease inhibitor. This decreases the rate of infection of new cells by the virus (see Chapter 2, subsection 2.3). Though our model MC1 does not explicitly incorporate protease inhibitors, we observe for $P_{mut} < P_{crit}$ the immune system dominating over the viral invader. This strong immune response could be representative of the effect of treatments like protease inhibitors. It has to be pointed out that the results of Ho et al were based on *clinical data* from 20 patients and their slope m is based on actual days while our slope m is based on a Monte Carlo Timestep. It would be naive of us to compare actual real time with an artificial Monte Carlo timestep.

7.1.2 Half-Life of Virus

Using Eqn (7.1) we can measure the half life of the virus in this initial rapid decline. This tells us how rapidly the virus halves its concentration, which is an indication of the strength of the immune response. $t_{1/2}$ ranges from 38.5 timesteps to 126.1 timesteps. This variation in half-lives is to be expected,

because as virus mutation levels increase it takes longer for the immune system to kill the virus.

The rate of viral decay is dependent on the initial viral load, m varies 2-fold corresponding with initial viral load varying 3-fold. This could be interpreted as viral decay being dependent upon the stage of infection. High viral load represents progressed infection and lower viral loads would represent the latent infection period. It has been found that viral load and viral clearance rates are independent(Perelson et al, 1997) . The investigations here have been based on a fixed P_{mut}. The above finding may indicate that our mutation probability P_{mut} might best be utilised as variable, with P_{mut} varying throughout the length of the simulation, rather than fixed, where P_{mut} is set at the start of the simulation and not altered.

The equilibrium level of the virus population does not vary greatly for any $P_{mut} < P_{crit}$, with the equilibrium level ranging from $\approx$925 sites out of 10,000 total sites (.0925 concentration) to $\approx$1600 sites out of 10,000 (.16 concentration). Even at zero mutation the immune system does not eliminate all of the virus. This means that even in the presence of a strong immune response, pockets of viral activity still exist. These pockets could easily extend if the mutation probability increased even slightly.

7.2 Helper Growth

We now turn our attention to studying the growth of the Helper cells. Helper cells grow to a steady equilibrium, (see Fig 7.4). The growth of the Helper cells, up to the oscillating equilibrium, is represented well by a logarithmic function, $H \propto \ln(t)$, but as $P_{mut} \rightarrow P_{crit}$ the approximation becomes less exact. An exponential increase in Helper cell population is consistent with proliferation of Helper cells in secondary lymphoid organs such as the lymph node, whilst a linear increase is consistent with procution from a precursor source such as the thymus (Mitchie, 1992). This logarithmic growth could represent a combination of both proliferations which is viable as Helpers in MC1 have no defined precursor. As expected the Helper cell equilibrium is dependent on P_{mut} with its population decreasing with increasing P_{mut} .

The variation in the equilibrium levels of Helper cells, is much more than for the virus, with the population decreasing from $H \approx 6000$ (0.6 concentration) for low mutation levels to $H \approx 4000$ (0.4 concentration) for high mutation levels.

This indicates that viral mutation has a larger effect on the Helper population, than it does on the viral one. Interpreting this, one could say that the decrease in the Helper population, is primarily caused by a higher proportion of mutated virions, rather than an actual increase in viral population.

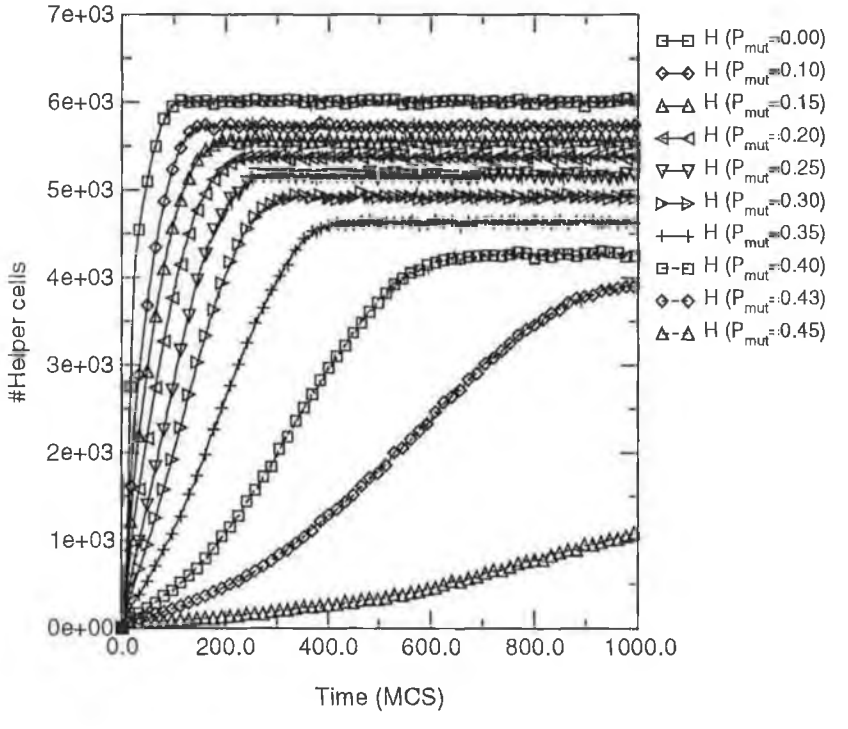


Fig 7.4 MC1 model: Helper Cell Growth Pattern for various P_{mut}

In Figure 7.5, we present the variation of equilibrium cell density with mutation rate, in order to see the relative progression of cell counts. The low mutation regime could be interpreted as the latent period of the infection, where the host cells are able to control the recognisable virus. The cross-over regime relates to the prolonged period of infection where the competition between the host cells and the virus becomes intense. The viral explosion at P_{crit} corresponds to a very advanced stage of infection. A very slight change in the mutation

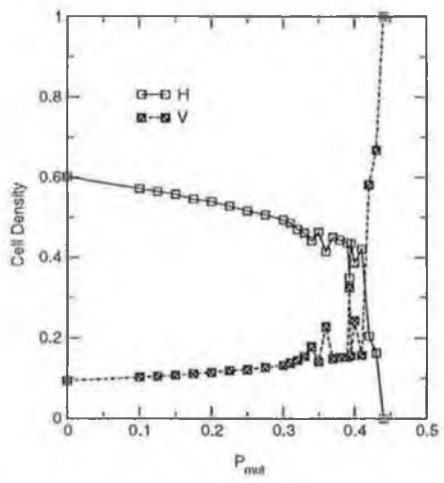


Figure 7.5 Above is the Cell Density of Viral and Helper cells virus P_{mut}

level could result in the transition from a latent stage of the disease to a very advanced one.

7.3 Critical Recovery Time and Crossover

Defn. A *critical recovery time* is the number of timesteps it takes for the Helper population to become larger than the viral population.

This is an important parameter to study as it tells us if and when the immune system can defeat the virus. This critical time, t_{crit}, is dependent on P_{mut}.

Figure 7.6 shows t_{crit} versus P_{mut} on a normal-log scale, which suggests exponential dependence

$$t_{crit} \simeq Ae^{\alpha P_{mut}}$$

with $\alpha = 4.52 \pm 0.29$ in low mutation regime and $\alpha = 15.21 \pm 1.41$ in the high mutation regime. Thus, there is a crossover from a relatively slow recovery time period to a collapse regime at around $P_{mut} \sim 0.4$ when t_{crit} rises much faster. Therefore any anti-retroviral therapy would have to be administered before this crossover in order to sustain some recovery.

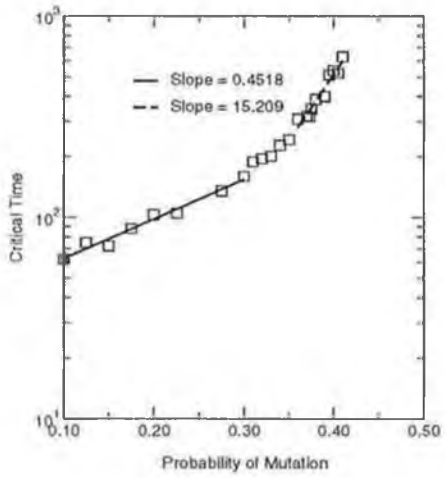


Figure 7.6 P_{mut} versus Critical recovery time

It is obvious from the above, that the Monte Carlo method, has enabled us to learn a lot more about the population dynamics of the cell types. The critical mutation probability, P_{crit}, is not significantly different from P_{crit} obtained previously using parallel updating, so it could be said that Monte Carlo, does not alter the overall results, but just presents us with a richer picture.

7.4 Enhanced Nearest-Neighbour Interactions

We also considered enhancing the nearest-neighbour interactions in MC2 (Mannion et al, 2000b). These were enhanced for a number of reasons. Firstly, to increase the intrinsic mobility of the system, so that the effect of cytokines and other "messenger" molecules would be stronger. Therefore we can ascertain, or at least hypothesise what effect a strong immune response would have on a HIV invader. The enhanced nearest-neighbour interactions are, (see Appendix 2 for code implementation)

$$M'' = M' \text{ or } V' \tag{7.3}$$

$$H'' = M' \text{ or } H' \tag{7.4}$$

$$C'' = C' \text{ or } H' \tag{7.5}$$

Where, $M^{'}$, $H^{'}$ and $C^{'}$, are obtained by the regular nearest-neighbour interactions as before. Therefore Macrophage development at a site can be stimulated by another Macrophage or viral-infected cell at one of its nearest neighbours. A Helper cell can be propagated if there is a Macrophage or a Helper cell at a site in its neighbourhood and a killer cell can appear at a site if one of its nearest neighbours has either a killer cell or helper cell in high concentration. To offset complete immune dominance, we also enhance the nearest-neighbour interaction of the Viral infected cell.

$$V^{''} = V^{'} \ or \ H^{'}$$

Therefore a viral infected cell or a helper cell at a neighbouring site can propagate a viral-infected cell at the central site. $M^{''}$, $H^{''}$,$C^{''}$, and $V^{''}$ are then used for the inter-site interactions. In MC2, nearest-neighbour interactions followed by the enhanced nearest-neighbour interactions followed by the intra-site interactions define an update of a site.

As might be expected with such a strong immune response, the critical mutation probability is quite high, $P_{crit} \simeq 0.88$, when compared with that which occurred previously in MC1. The MC2 growth patterns for Helper and Viral cells can be seen in Fig 7.7. This shows us how the immune system can benefit from an increase in inducers such as cytokines. An increase in these inducers makes it more difficult for the viral invaders to overwhelm the immune system. It should be stated however that only the positive inducing characteristics of cytokines were considered and their negative suppression factor was not taken into account. A further development of MC2 might include these suppression factors.

The growth patterns between MC1 and MC2 are also quite different (compare Fig 7.1 (MC1) with Fig 7.7(MC2)). Without the enhanced nearest-neighbour interactions in MC1 we saw the virus grow to a peak at the beginning, representing the initial viral attack and then the viral population decrease to an equilibrium. With the enhanced interactions we see the viral population reach its population equilibrium much like the helper growth pattern in the MC1. The Helper growth in this model has two patterns. The first for $P_{mut} \leq P_{crit}$ shows a pattern like the viral growth pattern. The second, for $P_{mut} > P_{crit}$ sees the Helper cells grow to a peak and then decrease to an equilibrium value. This second growth pattern for the Helper cells is much like the viral growth pattern in MC1. This indicates a shift in the method in which the virus attacks. Pre-

viously without the enhanced interactions, in MC1, the Helper cell population grew to an equilibrium value that was either greater or less than the viral population. Therefore *without* enhanced nearest-neighbour interactions the virus dominated by slowing the helper growth rate until the equilibrium helper population was less than the viral one. With the enhanced interactions in MC2 the helper population grows quickly in the beginning to try and halt the viral invasion but after this initial increase, it decreases due to the rapid growth of the virus.

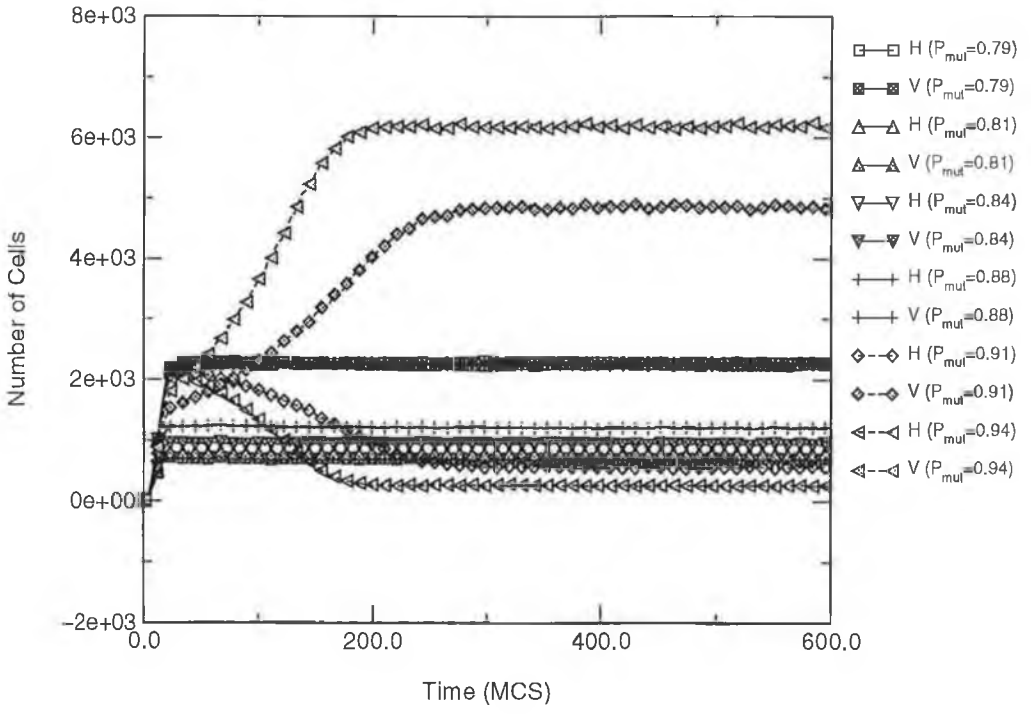


Fig 7.7 Helper and Viral growth patterns in MC2 model

Mobility affects the cell population growth patterns by speeding up the rate of growth for both viral and helper cells, (Pandey, 1998, Mannion et al 2000b) .Both cells reach their equilibrium populations more rapidly with extreme mo-

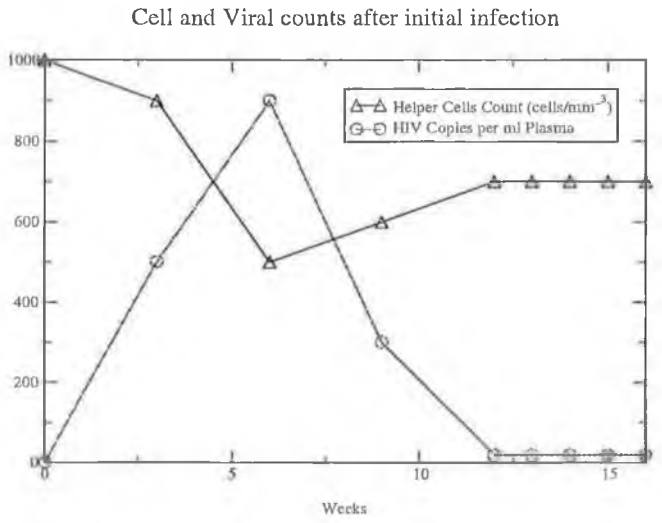


Figure 7.8 Helper and Viral cell counts just after initial infection (constructed from data from Fauci et al, 1996)

bility, $P_{mob} = 1$. This is probably due to mobility increasing the interactions between cell types. Also the rules governing mobility enable the cell types to be used to better effect.

7.4.1 Investigating Latency

One of the characteristics of HIV is the latency period, which is the length of time it takes from the original infection to full blown AIDS. This time varies from between 2-3 years (rapid progressors) , 7-11 years (typical progressors) to beyond 20 years (longterm non-progressors). In Figure 7.8 we see the helper cell count and viral count after the initial infection, in Figure7.9 we see what these cell counts would be like in a typical progressor after 10 years.

Factors determining how long this latency period lasts have been the subject of much research. Using MC2 with enhanced nearest-neighbour interactions, we investigated how the mutation of the virus may influence this latency period. For this investigation we used a 3-D lattice of length 50 (see Appendix 3 for code implementation). We looked at values of $P_{mut} \geq P_{crit}$, ($P_{mut} < P_{crit}$, led to Helper cell dominance) and then calculated how many timestep it took for the viral population to exceed that of the helper population. We found the

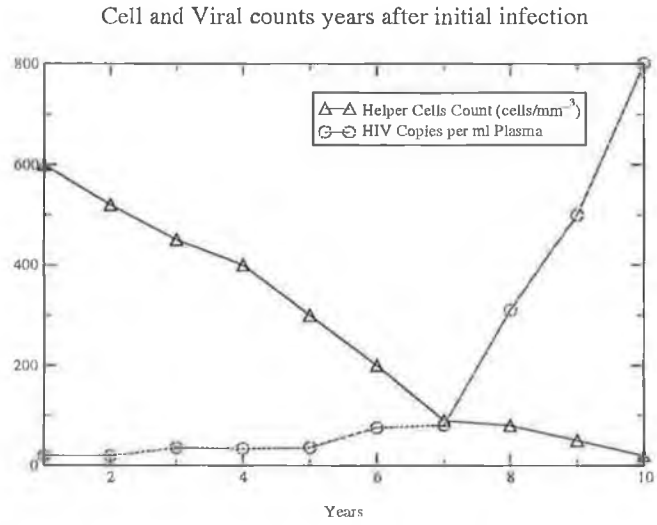


Figure 7.9 Helper and Viral cell counts in a typical progressor years after infection (constructed from data from Fauci et al, 1996)

following relationship with ΔP=P_{crit}−P_{mut}

$$MCS = \left(\frac{\Delta P}{3.58}\right)^{-0.73} \tag{7.7}$$

As is evident in the Eqn(7.7), small increases in P_{mut} lead to significant decreases in the latency time. A small increase in the mutation probability associated with one latency category can lead to a change to a different latency category of shorter timespan. Therefore we conclude that in relation to our model it is the mutation of the virus which determines the length of the latency time and a slight increase in that mutation can dramatically decrease the associated latency time. It must be stressed that the latency period we are discussing is that of our simulation, MC2, with time in units of Monte Carlo timesteps which we do not attempt to equate with real-time.

In this chapter we have presented two Monte Carlo models, MC1 and MC2, which utilise asynchronous updating. Asynchronous updating enables the investigation of cellular dynamics and phase transitions. Both viral and helper growth patterns were discussed, viral decay follows an exponential decline and helper cells grow in a logarithmic fashion. For levels of $P_{mut} < P_{crit}$ the time taken for the Helper cells to defeat the viral invaders was considered and an exponential dependence on P_{mut} was illustrated. We found that different categories of latency periods were dependent on P_{mut} and this dependency was very

sensitive. What this suggests is that the category of progressor a HIV-infected individual falls into, is dependent on the mutation rate of their HIV infection.

8 Conclusions and Future Work

We have presented discrete models of the immune system's interaction with HIV, models which used both synchronous (Cellular Automata) and asynchronous (Monte Carlo) updatings. Neither updating procedure can be said to be right nor wrong, or indeed one more correct than the other. They can only be used to complement each other. Some structures and patterns may be hidden using one updating procedure and may come to light when using the other. Because of the "overshooting" which occurs with synchronous updating the intermediate state of the system when it is poised between immune and viral dominance can not be investigated. With asynchronous updating the oscillations of cell populations were dampened sufficiently to allow us to explore this intermediate state in detail and let us investigate cell dynamics and growth patterns. Conversely, the deterministic aspect of synchronous updating allows us to see the dynamics of individual sites at a magnified level . Therefore the asynchronous updating shows us a "macroscopic" picture while the synchronous updating provides the "microscopic" equivalent.

Modelling an infection like HIV, using discrete methods is never going to be an easy task. The pathogenesis of AIDS and the internal dynamics of HIV are still not well understood (this of course being a reason to explore it using a mathematical model/computer simulation). Therefore assumptions have to be made on various aspects of the infection/disease that is not yet known. Also there is the other difficulty of scaling down this immense system to a few components. One is always going to face certain difficulties when trying to reduce the number of active components in a system with a large number of interacting elements. Choices have to be made. Which components are essential to the model ? Which are superfluous ? This choice of course can be wrong and is limited by the current knowledge base and our assumptions.

In our simulations we reduced the number of components to four cell types, the Macrophage,the Helper cell, the Killer cell and the Viral infected cells, out of a vast number of different cell types comprising the immune system. Therefore we have made a lot of assumptions, some which may be correct and some which certainly may not prove to be so. We have included probabilistic parameters to incorporate both the mutation of the virus and its explicit mobility. The mutation was necessary to include as it is the overwhelming characteristic of

the virus. Our results reflect this, we have shown that the mutation rate of the virus may be the determining factor of the latency period of the disease. Also the mobility parameter is restricted by rules, which could be rethought to better reflect chemotaxis, i.e the increased directional migration of cells.

Future work may rethink this model, in terms of increasing the number of cell types. This should be done in a systematic way without overly increasing the complexity of the model while ensuring that biological fact is never compromised. Theories that could be explored include the cross-regulation of Th1\Th2 Helper T cells, which was recently explored by continuous means (Fishman & Perelson, 1994). Mutation also could be re-thought; perhaps it could better explored as a variable parameter rather than fixed. A reason for exploring variable mutation was indicated by our results, (see Section 7.3.1), where we showed that the different stages of infection corresponded to different fixed mutation rates. If the mutation rate of HIV was to increase steadily throughout its infection, the rate of this increase would then determine the latency period, with a rapid increase leading to rapid progression into AIDS and a very slow or stationary rate leading to non-progression. Therefore treatment might focus on keeping the mutation rate below a critical point rather than on its eradication.

The purpose of any simulation is insight and not hard data and results. Theoretical immunology enables "clean" experiments to be performed with little cost. We have shown that asynchronous updating dampens cell population oscillations so that the actual mechanics of the cell-types can be determined, however, we still maintain Boolean expressions to govern their behaviour. Therefore we have all the benefits of a CA model, the interactions are completely deterministic without any of the loss of information caused by the "overshooting" which occurs with synchronous updating. Our asynchronous models can be used as a framework on which to build on in the future, changes can easily be encorporated and parameters varied. The field of theoretical immunology is growing, as computer resources increase and are capable of performing very large simulations for diverse models. The theoretical modelling of HIV should provide the traditional experimentalists with some fresh ideas for their research. Likewise, findings from traditional experiments should be incorporated into development of theoretical models. The complementary efforts of both fields is necessary for the successful future of each of them.

9 Glossary

Acquired Immune Deficiency Syndrome (AIDS) - A disease caused by the retrovirus HIV, which results in a dramatic decline in the T4 cell population, leading to a diminished immune system (see cell types below).

Antibodies- Proteins that bind to antigens and aid in their removal and destruction

Antigen-A molecule recognised as foreign by the immune system

Antigen Presenting Cells (APCs) - cells that can process and present antigen peptides(protein fragments) on their surface in association with Class II MHC molecules,(MHC defined below). A Macrophages is an APC.

B cells- B cells are lymphocytes that mature in the bone marrow; they are a source of antibodies

Epitope-An antigenic determinant present on an antigenic molecule, it interacts with an antibody or a T-cell receptor.

CD4- A protein on a cell surface, that recognises MHC II molecules on a APC.

CD8- A protein on a cell surface, that recognises MHC I molecules on a target cell.

Cytokines - Proteins that regulate the intensity and length of an immune response, by stimulating and inhibiting the proliferation of various immune cells and antibodies.

Cytotoxicity- The ability to kill cells.

gp120 - A protein which sits above and below HIV's surface which enable HIV to recognise, attach to and penetrate certain types of cells.

Helper cells- A subset of T4 cells, which when activated by APCs, differentiate and release cytokines to stimulate both cell mediated, (T8 killer cells) and humoral, (antibodies) responses.

Human immunodeficency virus (HIV)- A retrovirus, which infects T4 cells.

Inducers- T cells that trigger the maturation of other T cells

Leucoctyes- white blood cells

Lymph - Fluid surrounding the cells in the lymphatic system which provides a path through which nutrients, gases, and wastes can travel.

Lymphatic system- A collections of glands and vessels that drain lymph from body tissues and return it to the circulatory system (i.e. the blood system).

Lymphocytes- White blood cells central to the immune system, they can be categorised in three ways: T cells, B cells, and null cells

Lymhokines-A generic name for cytokines

Macrophage - A scavenger cell which specialises in ingesting and processing of antigen

MHC - Major Histocompatibility complex, a complex of genes encoding cell-surface molecules. Class 1 MHC molecules are on almost all nucleated cells. Class II MHC molecules are mainly expressed on APCs.

Interleukins- cytokines secreted by leukocytes

Null cells - Lymphocytes that do not express the membrane molecules that characterise T and B cells

Natural Killer Cells -Lymphocytes (null cells) that have cytotoxicity ability and are not MHC restricted.

Protease- Proteins which degrade other proteins, by splitting the peptide bonds, leading to the destruction or dramatic alternation of the target protein.

Retrovirus- A virus where the flow of genetic information is converted from RNA to DNA (which is the reverse of normal genetic information flow which converts DNA to RNA). The virus' genetic information is then integrated into the host cell's DNA, so that each time the cell multiplies so does the virus.

Reverse Transcriptase: A viral gene that carries out the conversion of RNA to DNA.

T cells- T cells are lymphocytes that mature in the thymus. They can be further sub divided into four subsets on the basis of function (Helper, Inducer, Killer and Suppressor) and only two sub classes on the basis of biochemical markers, namely T4 (with CD4 as a surface marker) and T8 (with CD8 as a surface marker) cells.

T4 cells- T lymphocytes that recognise antigen in the context of MHC II proteins. T4 cells have two functions . They function as Helpers and Inducers

T8- T cells that recognise antigen in the context of MHC I protein. T8 cells perform 2 main functions, killing and suppressing.

T8 killer cells - Activated by helper cells, these destroy target cells by antibody dependent cytotoxicity (ADCC).

T8 suppressor cells- T8 cells that suppress the immune response.

Th1-A subpopulation of the T4 Helper cell, which is defined by the cytokines it produces upon stimulation, (namely IL-2 and IFN) and mainly augments cell-mediated immunity

Th2-A subpopulation of the T4 Helper cell which is defined by cytokines it produces, (IL-4 ,IL5,Il-6 and IL-10) upon stimulation, which mainly augments humoral immunity

Tropism-HIV's attraction for T4 Helper cells, is due to the protein on the surface of the virus, gp120, and gp120's affinity for the CD4 protein on the surface of the Helper cell.

References

[1] 12th World AIDS Conference. Daily summary sessions. [online].
http://www.aids98.ch/archive/30_TUESDAY/300698_summarysession_tracka.html
http://www.aids98.ch/archive/1_WEDNESDAY/010798_summarysession_tracka.html
,
http://www.aids98.ch/archive/2_THURSDAY/020798_summarysession_tracka.html
,
http://www.aids98.ch/archive/3_FRIDAY/030798_summarysession_tracka.html

[2] Blakeslee D., 1996. The Journal of the American Medical Association, HIV/AIDS Information Center. Combination Therapy:The Rationale[online] available from http:// www.ama-ssn.org/special/hiv/newsline/briefing/comb.htm#note3 [accessed May 1999]

[3] BlakesleeD.,1996. Retroviruses: An Introduction [online] available http://www.ama-assn.org/special/hiv/newsline/briefing/retro.htm [accessed May 1999]

[4] Blakeslee D. 1996. Aid:The End of Latency. available http://www.ama-assn.org/special/hiv/newsline/briefing/latency.htm

[5] Blakeslee D. 1997. Latent HIV In Resting Memory Cells The Next Target For Eradication. [online] available. http://www.ama-assn.org/special/hiv/newsline/conferen/icaac/ic97per1.htm

[6] Celada, F., & Sieden, P.E., 1992. *A COMPUTER-MODEL OF CELLULAR INTERACTIONS IN THE IMMUNE-SYSTEM. Immunology Today.* 13(2) 56-62.

[7] CDC National Center for HIV, STD and TB Prevention Daily News Update: May 15, 2000 General Media. Cheaper drugs to Combat AIDS [online] availabale from http://www.ama-assn.org/special/hiv/newsline/cdc/051500g4.htm [accessed May 2000]

[8] Clerici, M. 1993. Cell-Mediated Immunity in HIV infection. AIDS. 7(suppl.1): S135-S140.

[9] Cohen, O.J., & Fauci, A.S., 1998. The Journal of the American Medical Association, HIV/AIDS Information Center. HIV/AIDS in 1998-Gaining the

Upper Hand ? [online] available from http://www.ama-assn.org/special/hiv/library/readroom/hiv98/ed80050x.htm [accessed March 1999].

[10] Colgate, S.A., Stanley, E.A., Hyman, J.M.,Quails, C.R., and Layne, S.P. Aids and a Risk-Based Model. Los Alamos Science, Number 18.

[11] Dasgupta, S., 1992. Monte Carlo simulation of the shape space model of immunology. Physica A, 189, 403-410.

[12] De Boer, R. J., Segel, L. A. and Perelson, A. S. (1992). Pattern formation in One and Two-dimensional Shape-space models of the Immune System. Journal of Theoretical Biology. 155, 295-333

[13] Deffner, M.,1993. Computer simulations of immunological cellular automata: the Tsallis model. Physica A, **195**, 279-285. dimensional shape space models of the immune system. J. Theoret. Biol. 155, 295-333.

[14] Fox, G. 1999. Stock Market Example of Multiple Monte Carlos[online], accessed June 1999. available from http://www.npac.syr.edu/users/gcf/CPS615NI95/node48.html

[15] Fishman, M.A. and Perelson, A.S. (1994)Th1/Th2 cross regulation. Journal of Theoretical Biology. **170, 25-26**

[16] Frisch, U., Hasslacher B., & Pomeau Y. Lattice Gas for the Navier-Stokes Equation. Physical Review letters. 56. 1505-1508.

[17] Gardner M.,1971. "Mathematical Games". Scientific Americam. **224**, February.

[18] Green, D.G., Tridgell, A. and Gill, A.M. (1990). Interactive simulation of bushfire spread in heterogeneous fuel. Mathematical and Computer Modelling 13(12), 57-66.

[19] Gutowitz, H. (1996). Cellular Automata and The Sciences of Complexity, Part I H. Complexity, **1**, 5

[20] Hammersley, J.M. and Handscomb, D.C. 1964. Monte Carlo Methods. Chapman and Hall, New York.

[21] Ho, D., Neumann, A.U., Perelson, A.S.,Chen, W., Leonard, M. & Markowitz, M. Rapid turnover of plasma virions and CD4 lymphocytes in HIV-1 infection, Nature. 373. 123-127.

[22] Hogeweg, P. and Hesper, B., 1990. Crowns crowding: an individual oriented model of the Acanthaster phenomenon. Pp. 169-188 in Acanthaster and the Coral Reef: A Theoretical Perspective ed by R.H. Bradbury. Springer-Verlag: Berlin.

[23] Jadach S., Richter-Was, E., Skrzypek, M., Ward, B.F.L, and Was, Z. (1992) Monte Carlo Program BHLUMI2.01 for Bhabha Scattering at Low Angles with Yennie-Frautschi-Suura Exponentiation . Computer Physics Communications 70 (1992), 305

[24] Jain. S., 1992. Monte Carlo Simulations of Disordered Systems. World Scientific.

[25] The Journal of the American Medical Association, HIV/AIDS Information Center. HIV infection and AIDS[online] available from http://www.ama-assn.org/special/hiv/support/infect.htm[accessed May 2000].

[26] Kaufman, M.,Urbain, J. &Thomas,R. 1985. Towards a logical analysis of the immune response. J Theor Biol 21;114(4):527-61

[27] Kohler, B. 1999. Vaccine Modelling via IMMSIM. Paper delivered at Theory in Immunology conference at ZiF, Bielefeld Germany, October 1999.

[28] Kougias, Ch. F. & Schulte, J., 1990. Simulating the Immune Response to the HIV-1 Virus with Cellular Automata . Journal of Statistical Physics , 60, 263-273.

[29] Kuby, J., 1992, Immunology, New York Freeman.

[30] Langton C.G., 1990,Computation at the edge of chaos: Phase transitions and emergent computation. Emergent Computation, page 12. North-Holland, Amsterdam, 1990. C. G. Langton, Computation at the edge of chaos: Phase transitions and emergent computation, In S. Forrest, editor, Emergent Computation, pa

[31] Laurence, J., 1985. The Immune System in AIDS. Scientific American, 253, 70-79.

[32] Lefèvre, O., Seiden, P.E. an Celada F. 1996. Insights into Rheumatoid Factor Production using a Cellular Automaton Model of the Immune System. Advances in Mathematical Modeling of Biological Processes, Ed. D. Kirschner, Int. J. Appl. Sci. Computation 3, 32-47 .

[33] Levin et al. The Impact of Biology on Mathematics. [online]http://www.bis.med.jhmi.edu/Dan/mathbio/T.html, accessed June 2000

[34] Leigh JE, Steele C, Wormley FL Jr., Luo W, Clark RA, Gallaher W, Fidel PL Jr. Th1/Th2 cytokine expression in saliva of HIV-positive and HIV-negative individuals: a pilot study in HIV-positive individuals with oropharyngeal candidiasis. J Acquir Immune Defic Syndr Hum Retrovirol 1998; 19(4):373-80.

[35] Leitner, T & Albert, J. The molecular clock of HIV-1 unveiled through analysis of a known transmission history. Proc. Natl. Acad. Sci. USA, 96,10752-10757.

[36] Li,W., Packard,.N.H., & Langton,C.G., 1990, Transisition Phenomena in Cellular Automata Rule Space. Physica D. **45**. 77-94.

[37] Mannion, R., Pandey, R.B., & Ruskin, H.J. 2000. Effect of Mutation on Helper T cells and Viral Population: A computer Simulation Model for HIV. *Theor. Biosci.* **119** 10 .

[38] Mannion, R., Pandey, R.B., & Ruskin, H.J. (2000).A Monte Carlo Approach to Population Dynamics of Cells in an HIV Immune Response Model. *Theor. Biosci.* **119** 145.

[39] Mielke, A. & Pandey, R.B., 1998. A Computer Simulation Study of Cell Population in a Fuzzy Interaction Model for Mutating HIV. Physica A, 251, 430-438.

[40] Morpugo, D., Serentha, R., Seiden, P.E., & Celada, F. Modelling Thymic Function in a Cellular Automaton. Int. Immunology 7, 505-516

[41] National Insitute for Infectious Diseases, 1998. Factsheet-HIV/AIDS.

[42] Nowak, M.A. & McMichael A.J., 1995. How HIV Defeats the Immune System. Scientific American, August, 58-65

[55] Smolle, J. and H. Stettner. (1993). Computer simulation of tumour cell invasion by a stochastic growth model. J. Theor. Biol. 160: 63-72. , 10758-10763.

[56] Smith, D. 1999. Variable efficacy of repeated annual influenza vaccination. Delivered at Theory in Immunology conference, ZiF, Bielefeld, October 2000.

[57] Schreckenberg M. & Schadschneider A. Modelling and Simulation of Traffic Flow with Cellular Automata. 1996 Conference on Scientific Computing in Europe September 2-4,1996 Dublin Ireland. (Edited H.J. Ruskin & Y. Feng)

[58] Sieburg, H. B., McCuthchen, J. A., Clay, O. K., Cabalerro, L. & Ostlund, J.J 1990, Simulation of HIV infection in artificial immune systems . Physica D, 45, 208-277.

[59] Sites, D.P., Stobo, J.D. & Fudenberg , H. H. & Wells, J.V., 1984. Basic & Clinical Immunology,5th Edition, LangeMedical Publications.

[60] Stauffer, D. 1988. Monte Carlo Simulations in Statisical Physics , in Computer Simulation and Computer Algebra. (ed Winkelmann and Zabolitzky). Springer-Verlag,Berlin.

[61] Stauffer, D. & Pandey, R.B., 1992. Immunologically Motivated Simulations of Cellular Automata, Computer Simulations . 6(4), 404-410

[62] Stauffer, D., 1992. Localisation Versus Damage Spreading in Immunological Shape Space : A Vectorised Algorithm. International Journal of Modern Physics C, 3(6), 1189-1194.

[63] Wolfram, S. 1986. Theory and Applications of Cellular Automata. World Scientific Publishing.

[64] Wolfram, S. 1984. Universality and Complexity in Cellular Automata. Physica D. 1-35.

[65] World Health Organisition, 2000. Report on the Global HIV/AIDS epidemic. June

[66] Zorzenon dos Santos, R. M., 1993. Transient times and periods in the high-dimensional shape-space model for immune systems, Physica D, 196, 12-20.

10 Appendices

Appendix 1

MC1 Fortran Programme 2-d Lattice with Asynchronous updating, mutation and mobility

```
      program mc2d11

c ****
c     MC/Stochastic simulations for immune response
c     Four cell-types are considered: 1->Macrophages
c     2 -> Helper (T4), 3->Cytotoxic (T8), 4->Virus (V)
c ****
c     Method: Distribute cells of concentration p(i=1,4)
c           on a lattice randomly with no more than one
c           cell of one type at a site.
c           Select a site (i,j) randomly.
c       Use the neighboring (inter-site) interaction rule

      implicit none

      integer   i, j, k, l, lp1, ls, it, jt, ir
      integer   i1, j1, i2, j2, nsite
      integer   ncell, nn, nbr
      integer   ic1, ih1, im1, iv1, mutv
      integer   nrun, maxt, mint, mintf
      integer   ibin, iseed

      real      pmut, pmob, per
      real      abm, rand
c     real      t1, t2, second, dum, ranf

      parameter (l = 100, ls = l*l)
      parameter (maxt = 1000,mint = 1, mintf = ls)
      parameter (lp1 = l + 1, iseed = 7893, nrun = 10)
      parameter (pmut = 0.43, pmob=0.0)
      parameter (ncell = 4, nn = 4)

      integer   icell(ncell), icsum(ncell), isum(ncell)
      integer   mx(nn), my(nn)
      integer   ib(0:lp1)
      integer   ic(ncell,l,l)
      integer   ix(ncell), iy(ncell)

      real      p(ncell)
      real      x(ncell, maxt), y(ncell, maxt)
      real      r(ncell, maxt)
```

```
      real      ac(ncell, maxt)
      real      ac0(ncell)
      real      per_cell(ncell)

c     call srand(iseed)
c     call ranset(iseed)

c     t1 = second(dum)
      ibm = 2*iseed - 1

      write(6,*)' mc2d11.f: l, maxt,mint,mintf, nrun ='
      write(6, 1000) l, maxt,mint,mintf, nrun
1000   format(5i7)
      write(6, *) ' pmut =', pmut,' pmob =',pmob
      write(6,*)' iseed =',iseed

      p(1) = 0.0001
      p(2) = 0.000
      p(3) = 0.000
      p(4) = 0.0001

      do 5 k = 1, ncell
        icell(k) = 1 + p(k)*ls
5     continue

      write(6,*)' Initial conc. of cells:1-ncell:'
      write(6,1001) p(1), p(2), p(3), p(4)
1001   format(4f12.8)
      write(6,*)' Initial cells # : 1-ncell:',
     &          icell(1), icell(2), icell(3), icell(4)

      per  = 1./float(nrun)

      mx(1) =  1
      mx(2) =  0
      mx(3) = -1
      mx(4) =  0
      my(1) =  0
      my(2) =  1
      my(3) =  0
      my(4) = -1

      do 10 k = 1, ncell
        ac0(k) = 0.0
      do 10 j = 1, maxt
```

```
          ac(k,j) = 0.0
          x(k,j) = 0.0
          y(k,j) = 0.0
          r(k,j) = 0.0

10       continue

        do 20 i = 1, l
          ib(i) = i
20       continue
          ib(lp1) = 1
          ib(0) = l

c ****
c       outer (nrun) loop
c ****

        do 80 ir = 1, nrun

c ****
c       initialize the cellular state
c ****

          do 30 j = 1, l
          do 30 i = 1, l
          do 30 k = 1, ncell
            ic(k, i,j) = 0
30         continue

c ****
c          distribute the cells randomly in the lattice
c ****

          do 34 k = 1, ncell

            if (icell(k).gt.0) then

              isum(k) = 0

32             continue

              ibm = ibm * 16807
              abm = 0.5*(float(ibm)+2147483647.)
              rand = abm/2147483647.
              i1 = 1 + rand*l
              ibm = ibm * 16807
```

```
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                j1 = 1 + rand*l
                if (ic(k,i1,j1).lt.1) then
                  ic(k,i1,j1) = 1
                  isum(k) = isum(k) + 1
                endif

                if (isum(k).lt.icell(k)) go to 32

              endif

34             continue

c ****
c       calculate the initial number of each cell type
c       initialize their displacements
c ****

            do 45 k = 1, ncell
              icsum(k) = 0

            do 40 j = 1, l
            do 40 i = 1, l
              icsum(k) = icsum(k)+ ic(k,i,j)
40           continue

              ac0(k) = ac0(k) + float(icsum(k))
              ix(k) = 0
              iy(k) = 0
45           continue

            do 70 it = 1, maxt

              do 62 jt = 1, mint

              do 50 nsite = 1, mintf


c             do 50 nsite = 1, ls

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                i = 1 + rand*l
                ibm = ibm * 16807
```

```
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                j = 1 + rand*l

                im1 = (4 + ic(1,i,j)
     :                  + ic(1, ib(i-1),j) + ic(1, ib(i+1),j)
     :                  + ic(1, i,ib(j-1)) + ic(1, i, ib(j+1)))/5
                ih1 = (4 + ic(2,i,j)
     :                  + ic(2,ib(i-1),j) + ic(2,ib(i+1),j)
     :                  + ic(2,i,ib(j-1)) + ic(2,i, ib(j+1)))/5
                ic1 = (4 + ic(3,i,j)
     :                  + ic(3,ib(i-1),j) + ic(3,ib(i+1),j)
     :                  + ic(3,i,ib(j-1)) + ic(3,i, ib(j+1)))/5
                iv1 = (4 + ic(4,i,j)
     :                  + ic(4,ib(i-1),j) + ic(4,ib(i+1),j)
     :                  + ic(4,i,ib(j-1)) + ic(4,i, ib(j+1)))/5


                mutv = iv1

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                if (rand.le.pmut) mutv = 0

                ic(1,i,j) = (im1 + mutv + 1)/2
                ic(2, i,j) = (1 - iv1)*(ih1 + im1 +1)/2
                ic(3, i,j) = ih1 * im1 * mutv
                ic(4,i,j) = (1 - ic1) * (ih1 + im1 + iv1 + 2)/3

50          continue

c ****
c       attempt to move each cell
c ****

            ibm = ibm * 16807
            abm = 0.5*(float(ibm)+2147483647.)
            rand = abm/2147483647.
            if (rand.le.pmob) then

              do 60 j = 1, l
              do 60 i = 1, l

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
```

```
rand = abm/2147483647.
i1 = 1 + rand*l

ibm = ibm * 16807
abm = 0.5*(float(ibm)+2147483647.)
rand = abm/2147483647.
j1 = 1 + rand*l

ibm = ibm * 16807
abm = 0.5*(float(ibm)+2147483647.)
rand = abm/2147483647.
k = 1 + rand*ncell

if (ic( k,i1,j1 ) .eq. 1) then

   ibm = ibm * 16807
   abm = 0.5*(float(ibm)+2147483647.)
   rand = abm/2147483647.
   nbr = 1 + rand*nn

   i2 = ib(i1 + nnx(nbr))
   j2 = ib(j1 + nny(nbr))

   if ( ic(k,i2,j2) .eq. 0 ) then

     if ( (k.eq.1).and. (ic(4,i2,j2).eq.1)) then

       ic(k, i2, j2) = ic(k, i1,j1)
       ic(k, i1, j1) = 0

       ix(k) = ix(k) + nnx(nbr)
       iy(k) = iy(k) + nny(nbr)

     elseif (k.eq.2) then

       ic(k, i2, j2) = ic(k, i1,j1)
       ic(k, i1, j1) = 0

       ix(k) = ix(k) + nnx(nbr)
       iy(k) = iy(k) + nny(nbr)

     elseif ( (k.eq.3) .and.
              ic(4,i2,j2).eq.1) then

       ic(k, i2, j2) = ic(k, i1,j1)
       ic(k, i1, j1) = 0
```

```
                    ix(k) = ix(k) + mx(nbr)
                    iy(k) = iy(k) + my(nbr)

                  elseif ( (k.eq.4) .and.
     :                    ( ic(2,i2,j2) .eq. 1 .or.
     :                      ic(1,i2,j2) .eq. 1) ) then

                    ic(k, i2, j2) = ic(k, i1,j1)
                    ic(k, i1, j1) = 0

                    ix(k) = ix(k) + mx(nbr)
                    iy(k) = iy(k) + my(nbr)

                  endif

                endif

              endif

60            continue

            endif

62          continue

c ****
c           find the number of each cell type
c ****
            do 66 k = 1,ncell
              icsum(k) = 0

            do 64 j = 1,l
            do 64 i = 1,l
c             ic(k,i,j) = icn(k,i,j)
              icsum(k) = icsum(k) + ic(k, i,j)
64          continue
              per_cell(k) = 1./float(icsum(k) + 1)
              ac(k, it) = ac(k, it) + float(icsum(k))

66          continue


            do 68 k = 1, ncell

              x(k,it) = x(k,it) + float(ix(k))
```

```
                y(k,it) = y(k,it) + float(iy(k))
c               x(k,it) = x(k,it) + per_cell(k)*float(ix(k))
c               y(k,it) = y(k,it) + per_cell(k)*float(iy(k))

68            continue


70          continue

c ****
c           end of the time loop
c ****

80          continue

c ****
c           end of the run loop
c ****

              do 90 k = 1, ncell
                ac0(k) = ac0(k)*per

              do 90 i = 1, maxt

                ac(k,i) = ac(k,i)*per

                if (pmob.gt.0.0) then

                  x(k,i) = x(k,i)*per
                  y(k,i) = y(k,i)*per
                  r(k,i) = sqrt(x(k,i)*x(k,i) + y(k,i)*y(k,i))

                endif

90            continue

c ****
c           write the average no. cells
c ****

              write(6,*)' initial number of cell types:'
              write(6,2000) (ac0(i), i = 1, ncell)
2000           format(4f12.3)

              write(6,*)' i, ac(k, i), k = 1, ncell:'
c             write(6,3000)( i*mint, (ac(k,i), k = 1, ncell),
```

```
c     :                  i = 1, maxt)

            do it = 1, maxt
              write(6,3000) it*mint, (ac(k,it), k = 1, ncell)
            enddo

3000         format(i7, 4f12.3)

            if (pmob.gt.0.0) then

              write(6,*)' '
              write(6,*)' it*mint, r(k,it), k = 1, ncell:'

              do it = 1, maxt
                write(6, 4000) it*mint, (r(k,it), k = 1,ncell)
              enddo

              write(6,*)' '
              write(6,*)' it*mint, x(k,it), k = 1, ncell:'

              do it = 1, maxt
                write(6, 4000) it, (x(k,it), k = 1,ncell)
              enddo

              write(6,*)' '
              write(6,*)' it*mint, y(k,it), k = 1, ncell:'

              do it = 1, maxt
                write(6, 4000) it*mint, (y(k,it), k = 1,ncell)
              enddo

            endif

4000         format(i7, 4f12.3)
c           t2 = second(dum) - t1
c           write(6,*)' CPU =',t2,'Seconds'
            stop
            end
```

Appendix 2

MC2 Fortran Programme 2-d Lattice with Asynchronous Updating, Enhanced Nearest Neighbour Interactions, Mutation and Mobility

```
      program  im11

c ****
c     MC/Stochastic CA for immune response
c     Cray-version
c     Evaluate the difference between helper and viral population
c ****

      implicit none

      integer   i, j, k, l, lp1, ls, it, jt, ir
      integer   i1, j1, i2, j2, nsite
      integer   ncell, nn, nbr, kcell
      integer   ic1, ih1, im1, iv1, mutv
      integer   nc1, nh1, nm1, nv1
      integer   nrun, maxt, mint, mintf
      integer   ibm, iseed

      real      pmut, pmob, per
      real      abm, rand
c     real      t1, t2, second, dum, ranf

      parameter (l = 100, ls = l*l)
      parameter (maxt = 500,mint = 1, mintf = ls)
      parameter (lp1 = l + 1, iseed = 7893, nrun = 50)
      parameter (pmut = 0.894, pmob=0.0)
      parameter (ncell = 4, nn = 4)

      integer   icell(ncell), icsum(ncell), isum(ncell)
      integer   nnx(nn), nny(nn)
      integer   ib(0:lp1)
      integer   ic(ncell,l,l)
      integer   ix(ncell), iy(ncell)

      real      p(ncell)
      real      x(ncell, maxt), y(ncell, maxt)
      real      r(ncell, maxt)
      real      ac(ncell, maxt)
      real      vhdiff(maxt)
      real      ac0(ncell)
```

```
      real       per_cell(ncell)

c       call srand(iseed)
c       call ranset(iseed)

c       t1 = second(dum)
        ibm = 2*iseed - 1

        write(6,*)' mc2d21.f: l, maxt,mint,mintf, nrun ='
        write(6, 1000) l, maxt,mint,mintf, nrun
1000     format(5i7)
        write(6, *) ' pmut =', pmut,' pmob =',pmob
        write(6,*)' iseed =',iseed

        p(1) = 0.0001
        p(2) = 0.000
        p(3) = 0.000
        p(4) = 0.0001

        do 5 k = 1, ncell
          icell(k) = 1 + p(k)*ls
5       continue

        write(6,*)' Initial conc. of cells:1-ncell:'
        write(6,1001) p(1), p(2), p(3), p(4)
1001     format(4f12.8)
        write(6,*)' Initial cells # : 1-ncell:',
     &              icell(1), icell(2), icell(3), icell(4)

        per  = 1./float(nrun)

        nnx(1) =  1
        nnx(2) =  0
        nnx(3) = -1
        nnx(4) =  0
        nny(1) =  0
        nny(2) =  1
        nny(3) =  0
        nny(4) = -1

        do 10 k = 1, ncell
          ac0(k) = 0.0
        do 10 j = 1, maxt

          ac(k,j) = 0.0
          x(k,j) = 0.0
```

```
            y(k,j) = 0.0
            r(k,j) = 0.0

10        continue

          do 20 i = 1, l
            ib(i) = i
20        continue
            ib(lp1) = 1
            ib(0) = l

c ****
c         outer (nrun) loop
c ****

          do 80 ir = 1, nrun

c ****
c         initialize the cellular state
c ****

            do 30 j = 1, l
            do 30 i = 1, l
            do 30 k = 1, ncell
              ic(k, i,j) = 0
30          continue

c ****
c           distribute the cells randomly in the lattice
c ****

            do 34 k = 1, ncell

              if (icell(k).gt.0) then

                isum(k) = 0

32               continue

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                i1 = 1 + rand*l
                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
```

```
                j1 = 1 + rand*l
                if (ic(k,i1,j1).lt.1) then
                  ic(k,i1,j1) = 1
                  isum(k) = isum(k) + 1
                endif

                if (isum(k).lt.icell(k)) go to 32

              endif

34             continue

c ****
c       calculate the initial number of each cell type
c       initialize their displacements
c ****

            do 45 k = 1, ncell
              icsum(k) = 0

            do 40 j = 1, l
            do 40 i = 1, l
              icsum(k) = icsum(k)+ ic(k,i,j)
40           continue

              ac0(k) = ac0(k) + float(icsum(k))
              ix(k) = 0
              iy(k) = 0
45           continue

            do 70 it = 1, maxt

              do 62 jt = 1, mint

              do 50 nsite = 1, mintf


c             do 50 nsite = 1, ls

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                i = 1 + rand*l
                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
```

```
      j = 1 + rand*l

      nm1 =      ic(1,i,j)
     :          + ic(1, ib(i-1),j) + ic(1, ib(i+1),j)
     :          + ic(1, i,ib(j-1)) + ic(1, i, ib(j+1))
      nh1 =      ic(2,i,j)
     :          + ic(2,ib(i-1),j) + ic(2,ib(i+1),j)
     :          + ic(2,i,ib(j-1)) + ic(2,i, ib(j+1))
      nc1 =      ic(3,i,j)
     :          + ic(3,ib(i-1),j) + ic(3,ib(i+1),j)
     :          + ic(3,i,ib(j-1)) + ic(3,i, ib(j+1))
      nv1 =      ic(4,i,j)
     :          + ic(4,ib(i-1),j) + ic(4,ib(i+1),j)
     :          + ic(4,i,ib(j-1)) + ic(4,i, ib(j+1))

      im1 = (9 + nm1 + nv1)/10
      ih1 = (9 + nh1 + nm1)/10
      ic1 = (9 + nc1 + nh1)/10
      iv1 = (9 + nv1 + nh1)/10

c     im1 = (4 + ic(1,i,j)
c    :          + ic(1, ib(i-1),j) + ic(1, ib(i+1),j)
c    :          + ic(1, i,ib(j-1)) + ic(1, i, ib(j+1)))/5
c     ih1 = (4 + ic(2,i,j)
c    :          + ic(2,ib(i-1),j) + ic(2,ib(i+1),j)
c    :          + ic(2,i,ib(j-1)) + ic(2,i, ib(j+1)))/5
c     ic1 = (4 + ic(3,i,j)
c    :          + ic(3,ib(i-1),j) + ic(3,ib(i+1),j)
c    :          + ic(3,i,ib(j-1)) + ic(3,i, ib(j+1)))/5
c     iv1 = (4 + ic(4,i,j)
c    :          + ic(4,ib(i-1),j) + ic(4,ib(i+1),j)
c    :          + ic(4,i,ib(j-1)) + ic(4,i, ib(j+1)))/5


      mutv = iv1

      ibm = ibm * 16807
      abm = 0.5*(float(ibm)+2147483647.)
      rand = abm/2147483647.
      if (rand.le.pmut) mutv = 0

      ic(1,i,j) = (im1 + mutv + 1)/2
      ic(2, i,j) = (1 - iv1)*(ih1 + im1 +1)/2
      ic(3, i,j) = ih1 * im1 * mutv
      ic(4,i,j) = (1 - ic1) * (ih1 + im1 + iv1 + 2)/3
```

```
50          continue

c ****
c           attempt to move each cell
c ****

            ibm = ibm * 16807
            abm = 0.5*(float(ibm)+2147483647.)
            rand = abm/2147483647.
            if (rand.le.pmob) then

              do 60 j = 1, l
              do 60 i = 1, l
              do 60 kcell = 1, ncell

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                i1 = 1 + rand*l

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                j1 = 1 + rand*l

                ibm = ibm * 16807
                abm = 0.5*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                k = 1 + rand*ncell

                if (ic( k,i1,j1 ) .eq. 1) then

                  ibm = ibm * 16807
                  abm = 0.5*(float(ibm)+2147483647.)
                  rand = abm/2147483647.
                  nbr = 1 + rand*nn

                  i2 = ib(i1 + nnx(nbr))
                  j2 = ib(j1 + nny(nbr))

                  if ( ic(k,i2,j2) .eq. 0 ) then

                    if ( (k.eq.1).and. (ic(4,i2,j2).eq.1)) then

                      ic(k, i2, j2) = ic(k, i1,j1)
                      ic(k, i1, j1) = 0
```

```
                ix(k) = ix(k) + nnx(nbr)
                iy(k) = iy(k) + nny(nbr)

              elseif (k.eq.2) then

                ic(k, i2, j2) = ic(k, i1,j1)
                ic(k, i1, j1) = 0

                ix(k) = ix(k) + nnx(nbr)
                iy(k) = iy(k) + nny(nbr)

              elseif ( (k.eq.3) .and.
     :                 ic(4,i2,j2).eq.1) then

                ic(k, i2, j2) = ic(k, i1,j1)
                ic(k, i1, j1) = 0

                ix(k) = ix(k) + nnx(nbr)
                iy(k) = iy(k) + nny(nbr)

              elseif ( (k.eq.4) .and.
     :                 ( ic(2,i2,j2) .eq. 1 .or.
     :                   ic(1,i2,j2) .eq. 1) ) then

                ic(k, i2, j2) = ic(k, i1,j1)
                ic(k, i1, j1) = 0

                ix(k) = ix(k) + nnx(nbr)
                iy(k) = iy(k) + nny(nbr)

              endif

            endif

          endif

60        continue

        endif

62      continue

c ****
c       find the number of each cell type
c ****
```

```
                do 66 k = 1,ncell
                  icsum(k) = 0

                do 64 j = 1,l
                do 64 i = 1,l
c                 ic(k,i,j) = icn(k,i,j)
                  icsum(k) = icsum(k) + ic(k, i,j)
64               continue
                  per_cell(k) = 1./float(icsum(k) + 1)
                  ac(k, it) = ac(k, it) + float(icsum(k))

66               continue


                 do 68 k = 1, ncell

                   x(k,it) = x(k,it) + float(ix(k))
                   y(k,it) = y(k,it) + float(iy(k))
c                  x(k,it) = x(k,it) + per_cell(k)*float(ix(k))
c                  y(k,it) = y(k,it) + per_cell(k)*float(iy(k))

68               continue


70              continue

c ****
c               end of the time loop
c ****

80              continue

c ****
c               end of the run loop
c ****

                do 90 k = 1, ncell
                  ac0(k) = ac0(k)*per

                do 90 i = 1, maxt

                  ac(k,i) = ac(k,i)*per

                  if (pnob.gt.0.0) then

                    x(k,i) = x(k,i)*per
```

```
                y(k,i) = y(k,i)*per
                r(k,i) = sqrt(x(k,i)*x(k,i) + y(k,i)*y(k,i))

              endif

90          continue

c ****
c           write the average no. cells
c ****

            write(6,*)' initial number of cell types:'
            write(6,2000) (ac0(i), i = 1, ncell)
2000          format(4f12.3)

            write(6,*)' i, ac(k, i), k = 1, ncell:'
c           write(6,3000)( i*mint, (ac(k,i), k = 1, ncell),
c    :                    i = 1, maxt)

            do it = 1, maxt
              vhdiff(it) = ac(4,it) - ac(2,it)
            enddo

            do it = 1, maxt
              write(6,3000) it*mint, (ac(k,it), k = 1, ncell),
     :                    vhdiff(it)
            enddo

3000          format(i7, 5f12.3)

            if (pnob.gt.0.0) then

              write(6,*)' '
              write(6,*)' it*mint, r(k,it), k = 1, ncell:'

              do it = 1, maxt
                write(6, 4000) it*mint, (r(k,it), k = 1,ncell)
              enddo

              write(6,*)' '
              write(6,*)' it*mint, x(k,it), k = 1, ncell:'

              do it = 1, maxt
                write(6, 4000) it, (x(k,it), k = 1,ncell)
              enddo
```

```
      write(6,*)' '
      write(6,*)' it*mint, y(k,it), k = 1, ncell:'

      do it = 1, maxt
        write(6, 4000) it*mint, (y(k,it), k = 1,ncell)
      enddo

    endif

4000   format(i7, 4f12.3)
c     t2 = second(dum) - t1
c     write(6,*)' CPU =',t2,'Seconds'
      stop
      end
```

Appendix 3

MC2 Fortran Programme 3-d Lattice with Asynchronous updating Enhanced Nearest-Neighbour Interactions,Mutation and Mobility.

```
      program  im3d11

c ****
c       MC/Stochastic CA for immune response
c       Cray-version
c       Evaluate the difference between helper and viral population
c ****

        implicit none

        integer    i, j, k, l, lp1, lll, it, jt, ir
        integer    i1, j1, k1, i2, j2, k2, nsite
        integer    kc, ncell, nn, nbr, kcell
        integer    ic1, ih1, im1, iv1, mutv
        integer    nc1, nh1, nm1, nv1
        integer    nrun, maxt, mint, mintf
        integer    ibm, iseed

        real       pmut, pmob, per
        real       abm, rand
c       real       t1, t2, second, dum, ranf

        parameter (l = 30, lll = l*l*l)
        parameter (maxt = 200,mint = 1, mintf = lll)
        parameter (lp1 = l + 1, iseed = 7893, nrun = 50)
        parameter (pmut = 0.851, pmob=1.0)
        parameter (ncell = 4, nn = 6)

        integer    icell(ncell), icsum(ncell), isum(ncell)
        integer    nnx(nn), nny(nn), nnz(nn)
        integer    ib(0:lp1)
        integer    ic(ncell,l,l,l)
        integer    ix(ncell), iy(ncell), iz(ncell)

        real       p(ncell)
        real       x(ncell, maxt), y(ncell, maxt), z(ncell, maxt)
        real       r(ncell, maxt)
        real       ac(ncell, maxt)
        real       vldiff(maxt)
        real       ac0(ncell)
```

```
      real      per_cell(ncell)
      real      persite, sum, avh, vhmax, vhmin, err

c     call srand(iseed)
c     call ranset(iseed)

c     t1 = second(dum)
      ibm = 2*iseed - 1

      write(6,*)' in3d11.f: l, maxt,mint,mintf, nrun ='
      write(6, 1000) l, maxt,mint,mintf, nrun
1000   format(5i7)
      write(6, *) ' pmut =', pmut,' pmob =',pmob
      write(6,*)' iseed =',iseed

      p(1) = 0.00025
      p(2) = 0.000
      p(3) = 0.000
      p(4) = 0.00025

      do 5 kc = 1, ncell
        icell(kc) = 1 + p(kc)*lll
5     continue

      write(6,*)' Initial conc. of cells:1-ncell:'
      write(6,1001) p(1), p(2), p(3), p(4)
1001   format(4f12.8)
      write(6,*)' Initial cells # : 1-ncell:',
     &          icell(1), icell(2), icell(3), icell(4)

      per = 1./float(nrun)

      nmx(1) =  1
      nmx(2) =  0
      nmx(3) = -1
      nmx(4) =  0
      nmx(5) =  0
      nmx(6) =  0
      nmy(1) =  0
      nmy(2) =  1
      nmy(3) =  0
      nmy(4) = -1
      nmy(5) =  0
      nmy(6) =  0
      nmz(1) =  0
      nmz(2) =  0
```

```
      nnz(3) =  0
      nnz(4) =  0
      nnz(5) =  1
      nnz(6) = -1

      do 10 kc = 1, ncell
        ac0(kc) = 0.0
      do 10 j = 1, maxt

        ac(kc,j) = 0.0
        x(kc,j) = 0.0
        y(kc,j) = 0.0
        z(kc,j) = 0.0
        r(kc,j) = 0.0

10     continue

      do 20 i = 1, l
        ib(i) = i
20     continue
        ib(lp1) = 1
        ib(0) = l

c ****
c       outer (nrun) loop
c ****

      do 80 ir = 1, nrun

c ****
c       initialize the cellular state
c ****

        do 30 k = 1, l
        do 30 j = 1, l
        do 30 i = 1, l
        do 30 kc = 1, ncell
          ic(kc, i,j,k) = 0
30       continue

c ****
c       distribute the cells randomly in the lattice
c ****

        do 34 kc = 1, ncell
```

```
            if (icell(kc).gt.0) then

              isum(kc) = 0

32             continue

              ibm = ibm * 16807
              abm = 0.4999*(float(ibm)+2147483647.)
              rand = abm/2147483647.
              i1 = 1 + rand*l
              ibm = ibm * 16807
              abm = 0.4999*(float(ibm)+2147483647.)
              rand = abm/2147483647.
              j1 = 1 + rand*l
              ibm = ibm * 16807
              abm = 0.4999*(float(ibm)+2147483647.)
              rand = abm/2147483647.
              k1 = 1 + rand*l
              if (ic(kc,i1,j1,k1).lt.1) then
                ic(kc,i1,j1,k1) = 1
                isum(kc) = isum(kc) + 1
              endif

              if (isum(kc).lt.icell(kc)) go to 32

            endif

34           continue

c ****
c       calculate the initial number of each cell type
c       initialize their displacements
c ****

            do 45 kc = 1, ncell
              icsum(kc) = 0

            do 40 k = 1, l
            do 40 j = 1, l
            do 40 i = 1, l
              icsum(kc) = icsum(kc)+ ic(kc,i,j,k)
40          continue

              ac0(kc) = ac0(kc) + float(icsum(kc))
              ix(kc) = 0
              iy(kc) = 0
```

```
            iz(kc) = 0
45        continue

          do 70 it = 1, maxt

            do 62 jt = 1, mint

            do 50 nsite = 1, mintf

              ibm = ibm * 16807
              abm = 0.4999*(float(ibm)+2147483647.)
              rand = abm/2147483647.
              i = 1 + rand*l
              ibm = ibm * 16807
              abm = 0.4999*(float(ibm)+2147483647.)
              rand = abm/2147483647.
              j = 1 + rand*l
              ibm = ibm * 16807
              abm = 0.4999*(float(ibm)+2147483647.)
              rand = abm/2147483647.
              k = 1 + rand*l

              nm1 =      ic(1,i,j,k)
             :         + ic(1,ib(i-1),j,k) + ic(1,ib(i+1),j,k)
             :         + ic(1,i,ib(j-1),k) + ic(1,i,ib(j+1),k)
             :         + ic(1,i,j,ib(k-1)) + ic(1,i,j,ib(k+1))
              nh1 =      ic(2,i,j,k)
             :         + ic(2,ib(i-1),j,k) + ic(2,ib(i+1),j,k)
             :         + ic(2,i,ib(j-1),k) + ic(2,i,ib(j+1),k)
             :         + ic(2,i,j,ib(k-1)) + ic(2,i,j,ib(k+1))
              nc1 =      ic(3,i,j,k)
             :         + ic(3,ib(i-1),j,k) + ic(3,ib(i+1),j,k)
             :         + ic(3,i,ib(j-1),k) + ic(3,i,ib(j+1),k)
             :         + ic(3,i,j,ib(k-1)) + ic(3,i,j,ib(k+1))
              nv1 =      ic(4,i,j,k)
             :         + ic(4,ib(i-1),j,k) + ic(4,ib(i+1),j,k)
             :         + ic(4,i,ib(j-1),k) + ic(4,i,ib(j+1),k)
             :         + ic(4,i,j,ib(k-1)) + ic(4,i,j,ib(k+1))

              im1 = (13 + nm1 + nv1)/14
              ih1 = (13 + nh1 + nm1)/14
              ic1 = (13 + nc1 + nh1)/14
              iv1 = (13 + nv1 + nh1)/14

              mutv = iv1
```

```
                ibm = ibm * 16807
                abm = 0.4999*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                if (rand.le.pmut) mutv = 0

                ic(1,i,j,k) = (im1 + mutv + 1)/2
                ic(2,i,j,k) = (1 - iv1)*(ih1 + im1 +1)/2
                ic(3,i,j,k) = ih1 * im1 * mutv
                ic(4,i,j,k) = (1 - ic1) * (ih1 + im1 + iv1 + 2)/3

50          continue

c ****
c       attempt to move each cell
c ****

            ibm = ibm * 16807
            abm = 0.4999*(float(ibm)+2147483647.)
            rand = abm/2147483647.
            if (rand.le.pmob) then

              do 60 k = 1, l
              do 60 j = 1, l
              do 60 i = 1, l
              do 60 kcell = 1, ncell-1

                ibm = ibm * 16807
                abm = 0.4999*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                i1 = 1 + rand*l

                ibm = ibm * 16807
                abm = 0.4999*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                j1 = 1 + rand*l

                ibm = ibm * 16807
                abm = 0.4999*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                k1 = 1 + rand*l

                ibm = ibm * 16807
                abm = 0.4999*(float(ibm)+2147483647.)
                rand = abm/2147483647.
                kc = 1 + rand*ncell
```

```
      if (ic( kc,i1,j1,k1 ) .eq. 1) then

        ibm = ibm * 16807
        abm = 0.4999*(float(ibm)+2147483647.)
        rand = abm/2147483647.
        nbr = 1 + rand*nn

        i2 = ib(i1 + nnx(nbr))
        j2 = ib(j1 + nny(nbr))
        k2 = ib(k1 + nnz(nbr))

        if ( ic(kc,i2,j2,k2) .eq. 0 ) then

          if ( (kc.eq.1).and. (ic(4,i2,j2,k2).eq.1)) then

            ic(kc, i2, j2,k2) = ic(kc, i1,j1,k1)
            ic(kc, i1, j1,k1) = 0

            ix(kc) = ix(kc) + nnx(nbr)
            iy(kc) = iy(kc) + nny(nbr)
            iz(kc) = iy(kc) + nnz(nbr)

          elseif (kc.eq.2) then

            ic(kc, i2, j2,k2) = ic(kc, i1,j1,k1)
            ic(kc, i1, j1,k1) = 0

            ix(kc) = ix(kc) + nnx(nbr)
            iy(kc) = iy(kc) + nny(nbr)
            iz(kc) = iz(kc) + nnz(nbr)

          elseif ( (kc.eq.3) .and.
     :             ic(4,i2,j2,k2).eq.1) then

            ic(kc, i2, j2,k2) = ic(kc, i1,j1,k1)
            ic(kc, i1, j1,k1) = 0

            ix(kc) = ix(kc) + nnx(nbr)
            iy(kc) = iy(kc) + nny(nbr)
            iz(kc) = iz(kc) + nnz(nbr)

          elseif ( (kc.eq.4) .and.
     :             ( ic(2,i2,j2,k2) .eq. 1 .or.
     :               ic(1,i2,j2,k2) .eq. 1) ) then

            ic(kc, i2, j2,k2) = ic(kc, i1,j1,k1)
```

```
                    ic(kc, i1, j1,k1) = 0

                    ix(kc) = ix(kc) + nnx(nbr)
                    iy(kc) = iy(kc) + nny(nbr)
                    iz(kc) = iz(kc) + nnz(nbr)

                  endif

                endif

              endif

60          continue

          endif

62          continue

c ****
c         find the number of each cell type
c ****
          do 66 kc = 1,ncell
            icsum(kc) = 0

          do 64 k = 1,l
          do 64 j = 1,l
          do 64 i = 1,l
            icsum(kc) = icsum(kc) + ic(kc, i,j,k)
64        continue
            per_cell(kc) = 1./float(icsum(kc) + 1)
            ac(kc, it) = ac(kc, it) + float(icsum(kc))

66        continue


          do 68 kc = 1, ncell

            x(kc,it) = x(kc,it) + float(ix(kc))
            y(kc,it) = y(kc,it) + float(iy(kc))
            z(kc,it) = z(kc,it) + float(iz(kc))

68        continue


70        continue
```

```
c ****
c          end of the time loop
c ****

80          continue

c ****
c          end of the run loop
c ****

            do 90 kc = 1, ncell
              ac0(kc) = ac0(kc)*per

            do 90 i = 1, maxt

              ac(kc,i) = ac(kc,i)*per

              if (pmob.gt.0.0) then

                x(kc,i) = x(kc,i)*per
                y(kc,i) = y(kc,i)*per
                z(kc,i) = z(kc,i)*per
                r(kc,i) = sqrt(x(kc,i)*x(kc,i) + y(kc,i)*y(kc,i)
     :                                         + z(kc,i)*z(kc,i))

              endif

90          continue

c ****
c          write the average no. cells
c ****

            write(6,*)' initial number of cell types:'
            write(6,2000) (ac0(i), i = 1, ncell)
2000         format(4f12.3)


            vhmax = 0.0
            vhmin = 0.0
            sum = 0.0
            do it = 1, maxt
              vhdiff(it) = ac(4,it) - ac(2,it)
              if (it.ge.maxt/2) then
                sum = sum + vhdiff(it)
                if (vhdiff(it).lt. vhmin) vhmin = vhdiff(it)
```

```
            if (vhdiff(it).gt. vhmax) vhmax = vhdiff(it)
          endif
        enddo

        avh = 2.0*sum/float(maxt)
        err = vhmax - vhmin

        write(6,*)' '
        print *,' avh =',avh
        print *,' vhmax =',vhmax,' vhmin = ',vhmin,' error =',err
        write(6,*)' '
        write(6,*)' i, ac(kc, i), kc = 1, ncell, v-h:'
        write(6,*)' '
        do it = 1, maxt
          write(6,3000) it*mint, (ac(kc,it), kc = 1, ncell),
     :                  vhdiff(it)
        enddo

3000     format(i7, 5f12.3)

        write(6,*)' '
        write(6,*)' i, Cell density: 1, ncell, v-h:'
        persite = 1./float(lll)

        do it = 1, maxt
          write(6,3000) it*mint, (ac(kc,it)*persite, kc = 1,
     :                  ncell), vhdiff(it)*persite
        enddo

        if (pmob.gt.0.0) then

          write(6,*)'  '
          write(6,*)' it*mint, r(kc,it), kc = 1, ncell:'

          do it = 1, maxt
            write(6, 4000) it*mint, (r(kc,it), kc = 1,ncell)
          enddo

        endif

4000     format(i7, 4f12.3)
c       t2 = second(dum) - t1
c       write(6,*)' CPU =',t2,'Seconds'
        stop
        end
```

Appendix 4

Ancillary publications on the thesis material

Theory Biosci. (2000) 119: 145–155

http: //www.urbanfischer.de/journals/theorybiosc

A Monte Carlo Approach to Population Dynamics of Cells in a HIV Immune Response Model

Rachel Mannion[1], Heather J. Ruskin[1] and R. B. Pandey[2]

[1]School of Computer Applications Dublin City University, Dublin 9, Ireland
[2]Department of Physics and Astronomy, University of Southern Mississippi, Hattiesburg, MS 39406–5046

Address for correspondence: Rachel Mannion, School of Computer Applications, Dublin City University, Dublin 9, Ireland

Received: August 7, 1999; accepted: October 25, 1999

Key words: asynchronous updating, Monte Carlo, HIV immune response, mutation, discrete model

Summary: Using a direct Monte Carlo simulation, population growth of helper T-cells (N_H) and viral cells (N_V) is studied for an immune response model with an enhanced spatial inter-cellular interaction relevant to HIV as a function of viral mutation. In the absence of cellular mobility ($P_{mob} = 0$), the helper T-cells grow nonmonotonically before reaching saturation and the viral population grows monotonically before reaching a constant equilibrium. Cellular mobility ($P_{mob} = 1$) enhances the viral growth and reduces the stimulative T-cell growth. Below a mutation threshold (P_c), the steady-state density of helper T-cell (ρ_H) is larger than that of the Virus (ρ_V); the density difference $\Delta\rho_o (= \rho_V - \rho_H)$ remains a constant at $P_{mob} = 1$ while $-\Delta\rho_o \to 0$ as $P_{mut} \to P_c$ at $P_{mob} = 0$. Above the mutation threshold, the difference $\Delta\rho_o$ in cell density, grows with $\Delta P = P_{mut} - P_c$ monotonically: $\Delta\rho_o \propto (\Delta P)^{\beta}$ with $\beta \simeq 0.574 \pm 0.016$ in absence of mobility, while $\Delta\rho_o \simeq 6(\Delta P)$ with $P_{mob} = 1$.

Introduction

Rule-based interactions are frequently used in computational modeling of cellular populations particularly for Cellular Automata (CA) methods and variants, i. e., probabilistic CA (PCA) and stochastic CA (SCA) [1–13]. In most CA approaches [2–5], part of the rule-based intersite cellular interactions are implemented simultaneously, where the mechanics to achieve this simultaneity involve visiting each site of the lattice synchronously in order to assign each cell its intermediate state. A set of inter-cellular interaction rules is then used with these intermediate states at each lattice site to update their states at the next time step [8]. The stochastic mobility of cells was recently introduced in a stochastic CA (SCA) [13] in order to take

into account cellular motility, together with synchronous update with independent rules, as in traditional CA. Very recently, we introduced a computer simulation method [14] in which updates of the cellular states, in contrast to synchronous updates, occur randomly and one at a time, with subsequent mobility of cells. We extend this direct Monte Carlo (DMC) approach here to incorporate the effects of stimuli factors via inter-cellular inter-site interaction by including more cellular elements from the neighboring sites. We find that extending intercellular interaction to neighboring sites appropriate for HIV response, leads to different growth patterns than was found for the intracellular interaction alone, with the effect of viral mutation on cellular growth more pronounced.

Model

The model is very similar to that of our first MC simulation [14] of cell population in immune response, except for the inter-site interaction (see below). As before, we consider a cell mediated immune response with four cell types: macrophages (M), helper T-cells (H), cytotoxic T-cells (C), and antigen/virion (or virus carrying cells) (V) each with a binary cellular state to represent their high ("1") and low ("0") concentrations. A set of boolean expressions can be used to describe the intra-site cellular interactions [8],

$$M(t+1) = M(t).or.V(t), \tag{1(a)}$$

$$H(t+1) = [M(t).or.H(t)].and.[notV(t)], \tag{1(b)}$$

$$C(t+1) = M(t).and.H(t).and.V(t), \tag{1(c)}$$

$$V(t+1) = [H(t).or.M(t).or.V(t)].and.[notC(t)], \tag{1(d)}$$

where the states of the four cell types at time $t+1$ evolve from their states at time t. This interaction set, (Eq. 1) has been previously analyzed in detail with a mean field approach [8]. This analysis found that starting with a random configuration (out of sixteen) and applying the above equations leads to a flow diagram with two fixed points, which have a cycle of period two. The fixed points represent states of "immunocompetency" and "immunodeficiency", while the cycle includes infected, severely infected, and susceptible states.

In order to incorporate the effects of mediators, growth factors, effectors etc. [15, 16] via local interactions and to study the population of cells, we consider a discrete lattice of size $L \times L$. We typically start with one or two cells of each cell type distributed randomly among a fraction p of the lattice sites. A site can be occupied by four different cell types, with never more than one cell of each type at a site. A site i is referred to as occupied

by a cell type *ic* if the state of the cell type *ic* is 1 (high concentration); the cellular state '0' (low concentration), is referred to as an empty site for the cell type. The cell populations change as we implement the cellular interactions and update their states using the following steps:

(A) Random sequential update of cellular state:

(*i*) select a site *i* randomly.

(*ii*) find the sum $s_i(ic)$ of each cell type (*ic*) over their four neighbors and the cell at site *i*: if $s_i(ic) \geq 1$ then assign an intermediate state $s_i(ic)' = 1$ otherwise $s_i(ic)' = 0$.

(*iii*) with these intermediate states ($s_i(ic)' = M'$, H', C', V' for $ic = 1$, 2, 3, and 4 respectively), evaluate the corresponding stimulated states ($s_i(ic)'' = M''$, H'', C'', H''), using the following relations,

$$M'' = M'.or.V' \qquad (2(a))$$

$$H'' = H'.or.M' \qquad (2(b))$$

$$C'' = C'.or.H' \qquad (2(c))$$

$$V'' = V'.or.H' \qquad (2(c))$$

Note that this set of interactions is implemented in addition to interaction (1) in the following step (iv) to take into account IL2, cytokines and other effectors to enhance the reaction.

(*iv*) Using the current state $s_i(ic)''$ of cells at site *i*, implement the intercellular interaction (Eq. (1)) among different cell types, to update their state, i. e., *M*, *H*, *C* and *V* at site *i*.

(*v*) Repeat steps (*i*)–(*iv*) L^2 times.

(B) Mutation: viral mutation is considered probabilistically in the preceding step *A*(*iv*) and where inter-cell interactions (Eq. (1)) are implemented. With probability (P_{mut}), a virus is mutated such that the host cells no longer recognize it; then $V = 0$ in Eq. (1(a), 1(c)).

(C) Random sequential move with immunological mobility: we set the mobility rate P_{mob} in the beginning of simulation, $0 \leq P_{mob} \leq 1$ where $P_{mob} = 0$ means no mobility and $P_{mob} = 1$ describes highest mobility. The following steps (*i–viii*) are implemented with probability P_{mob}:

(*i*) select a site *i* randomly,

(*ii*) select a cell type (*ic*) at the site *i* randomly.

(*iii*) **IF** the cell type *ic* is present at site *i*, **THEN**

(*iv*) select one of the nearest neighbor sites *j*;

(*v*) **IF** the cell type *ic* is absent at site *j*, **THEN**

(*vi*) With probability P_{mob} attempt to move the cell type *ic* from site *i* to site *j*. In order to accept the move further specific criteria [16] must be satisfied. For example, site *j* must have a viral infected cell, i. e., $V = 1$ for macrophages and cytotoxic cells to move. On the other hand, a virion can move to site *j* if either macrophage or helper cell or both cell types are present, i. e., $(M.or.H) = 1$ at site *j*.

(*vii*) Go to step (*i*) in case any of the above "IF" conditions fail.
(*viii*) Repeat steps (*i*)–(*vii*) $4 \times L^2$ times.

The above procedures, (*A*–*C*), carried out sequentially, define a unit Monte Carlo step (MCS). We perform the simulation for a fixed number of time steps with a number of independent runs for each mutation probability with different mobility rates. Note that this procedure is nearly the same as in our first MC study, except in step *A*(*iii*) where the cellular states are stimulated by their inter-site-inter-cell interactions, Eq. (2(a)–(d)), as well as by the separate inter-cell and inter-site interactions. Including this stimulus, produces a considerable change in the growth pattern as we see below. Further, the interactions, cellular mobility, along with the mutation mechanism adopted here are specific to this model and may depend on the type of immune response.

Results and Discussion

Simulations are performed mostly on 2-dimensional 100×100 and 200×200 lattices with a very low initial concentration of each cell type, typically one of each distributed randomly. We have also used different sample sizes to check for severe finite size effects and the qualitative results are independent of the lattice sizes. The mutation rate, $0 \leq P_{mut} \leq 1$, is varied and the data presented here are mostly in the range of $P_{mut} \simeq 0.75 - 1.00$, where significant changes in the growth pattern of virus and helper T-cells are observed for mobility $P_{mob} = 0$ and 1. It must be pointed out that the range of numerical value of P_{mut} is relatively large and should not be compared with clinical mutation rates – it must be scaled in order to make it clinically realistic; also a variable mutation rate may prove to be more characteristic of the virus. Up to 50 independent samples are used to find the average number of cells. As before, we monitor the population of each cell type with time steps as a function of mutation rate. We focus primarily on the populations of helper T-cells and viral cells since the population of macrophages reaches its maximum constant value in a rather short series of time steps.

Since the simulation is performed on a square lattice it is easy to inspect visually as the number of cells grow. Figure 1 shows a typical evolution of cells at different time steps. For viral mutation $P_{mut} = 0.90$, we see that both helper T-cell and viral populations are low but $N_H > N_V$ at the initial stage of growth (Fig. 1(a)). The viral population grows faster and dominates the helper cells over long time periods (Fig. 1). Attempts are made to quantify such growth patterns in the following.

Growth of cell populations with time is presented in Figure 2 for $P_{mob} = 0$. At relatively low mutation, ($P_{mut} < 0.90$), both helper T-cell and virus grow very fast to their constant values ($N_H \to N_{HS}$, $N_V \to N_{VS}$). While

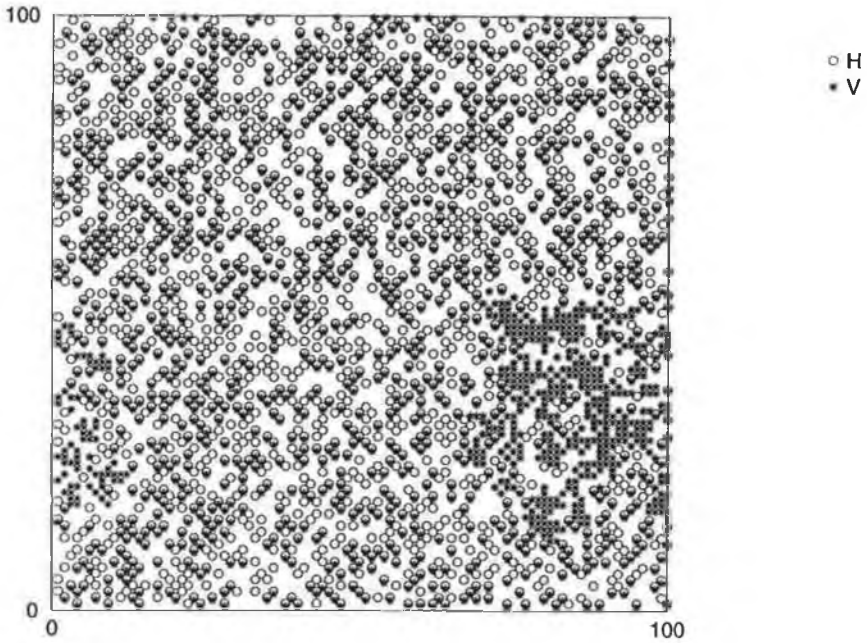


Figure 1 a

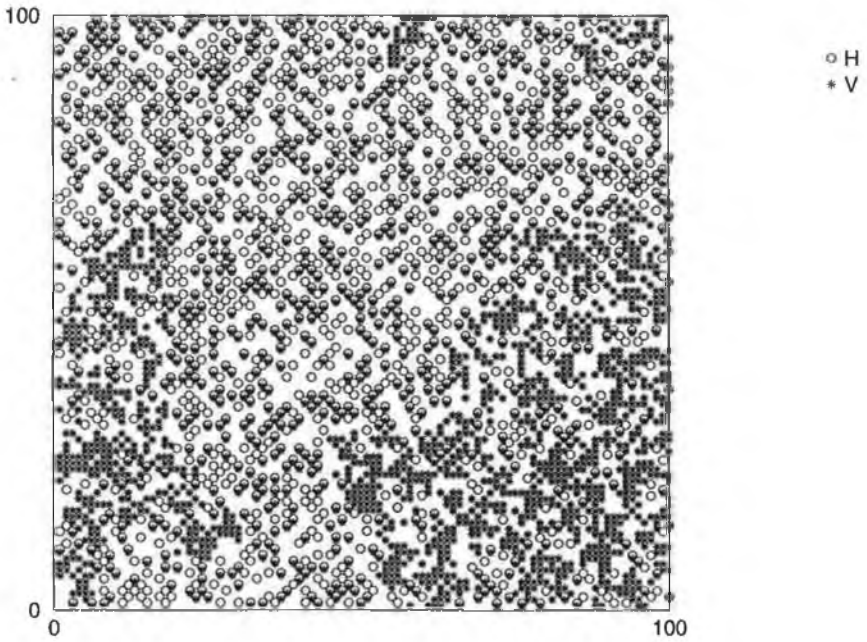


Figure 1 b

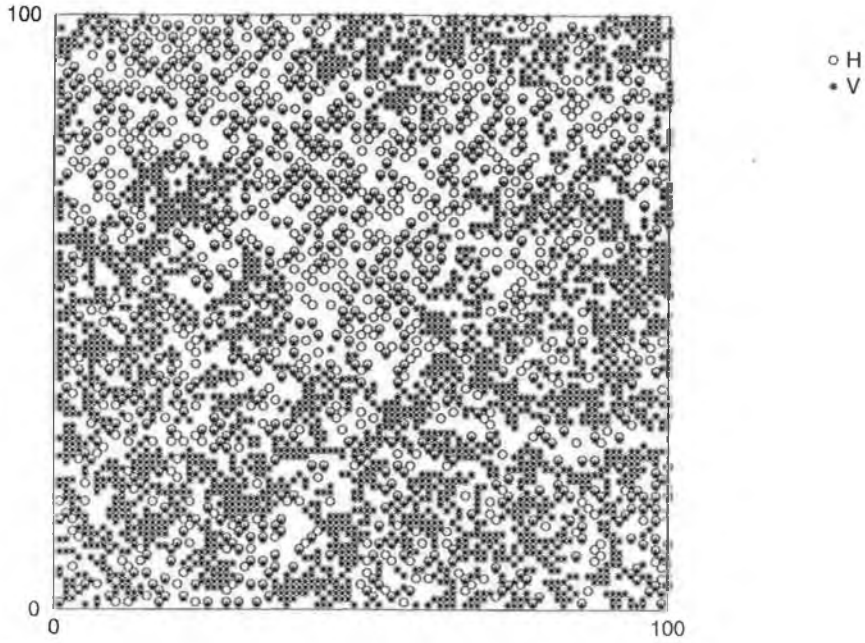


Fig. 1 c

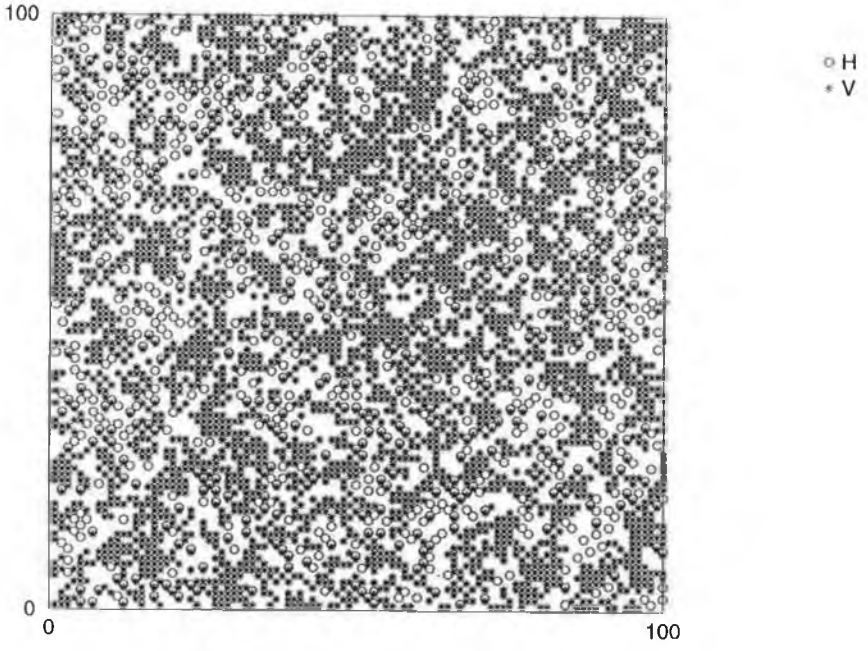


Fig. 1 d

Fig. 1. Typical snapshots of helper T-cells (square) and virus (star) at time steps $t = 75(a)$, $149(b)$, $223(c)$, $297(d)$ for $P_{mut} = 0.90$ in absence of cell mobility on a 100×100 lattice.

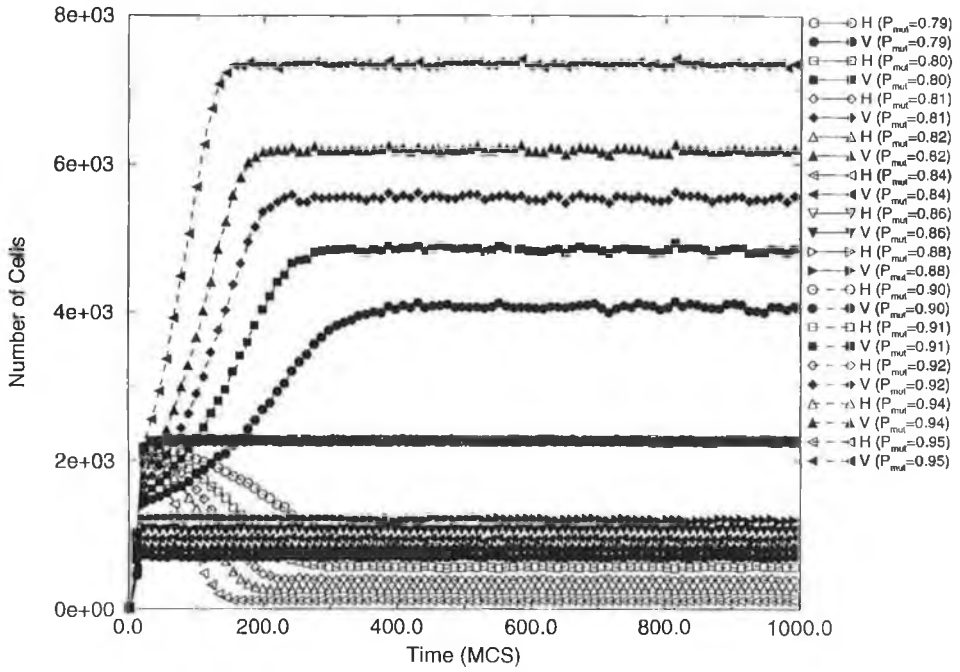


Fig. 2 a

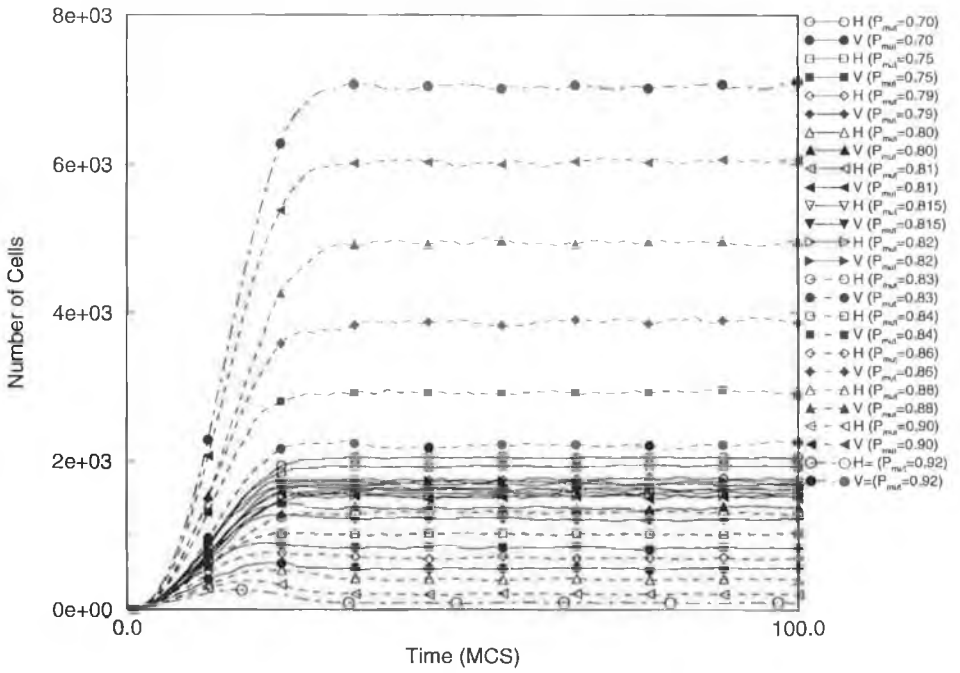


Fig. 2 b

Fig. 2. Number of cells (helper T-cells and virus) versus time step at various mutation rates P_{mut} with the cellular mobility $P_{mob} = 0.0(a), 1.0(b)$. Sample of size 100×100 is used with up to 50 independent runs.

N_{VS} increases very slowly, N_{HS} remains unchanged on increasing P_{mut} from 0.79 to 0.88. The helper cells dominate over the viral population. In the high mutation rate regime, on the other hand, the viral population increases much faster to a constant value while the population of helper T-cell decays systematically on increasing the mutation rate. Thus, the viral population dominates over the helper cell ($N_{VS} > N_{HS}$). With cell mobility ($P_{mob} = 1$), both helper cells and viral cells show a speedy increase to constant values and the viral population dominates over the helper cells at $P_{mut} \geq 0.81$ (see fig. 2(b)). Thus, cellular mobility *enhances* the viral effect. The results for the less-restricted mobility are presented here and differ from the restricted case only for $P_{mut} \geq P_{crit}$, leading to higher equilibrium population value for V. We would like to emphasise that the cellular growth pattern is different to that of our recent MC study [14] which lacked inter-cellular inter-site interaction as in *A(iii)*. Compared with our previous MC model [14], the cell populations are significantly decreased while the critical mutation rate is significantly higher. Also the viral population in this study never exceeds the helper population below the critical mutation rate, even at the initial stage of infection. Thus, the specificity of the local medium as considered here, (interaction *A(iii)*), is important in controlling the growth pattern of cells. Further, oscillation in cellular populations as found for stochastic cellular automata (SCA) [13] does not appear here as in our first MC study [14].

In order to provide an estimate of an MC time step with the order of magnitude in real time, we have to look at the clinical data such as the variation of HIV RNA copies in plasma, culturable plasma viremia, and CD4 cell (Helper cell) counts with weeks and years reported by Fauci et al. [15]. The growth and decay of helper cells with high viral mutation crudely resembles that of clinical data. Accordingly, about 300 MC timesteps corresponds to the order of about 10 years in clinical data. We would, however, like to caution that this comparison should not be taken literally since the growth and decay time depends on the size of the lattice.

The variation of equilibrium cell density with the mutation rate may provide an estimate of immune progression. Figure 3 shows such a variation. We see that the viral density increases while the helper cell density decays monotonically beyond a critical value, P_c. In absence of cellular mobility ($P_{mob} = 0.0$), increase in viral density above $P_c = 0.884$ shows a continuous transition to a progression of infection with P_{mut}. The decay of helper cell density, likewise, describes a continuous depletion of immuno-competence. For $P_{mut} \leq P_c$, helper cell density (ρ_H) is larger than the viral density (ρ_V) and the density difference, $\Delta\rho = \rho_H - \rho_V$, remains constant. One may interpret the region $P_{mut} \leq P_c$ as a latent state while $P_{mut} \geq P_c$ indicates a state of HIV with a continuous progression of infection above the critical threshold. Note that the cell mobility changes the progression considerably (Fig. 3). Below the critical mutation rate ($P_c \simeq 0.820$), we see a con-

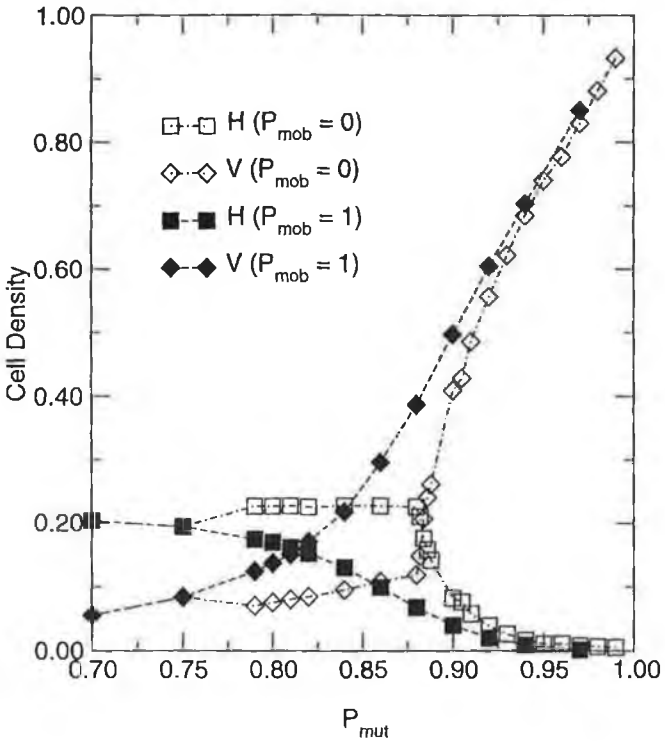


Fig. 3. Equilibrium densities of helper T-cells (ρ_H) and virus (ρ_V) versus mutation rate (P_{mut}). (Statistics are as for Figure 2).

stant decay of immuno-competence as $\Delta\rho$ decreases with P_{mut}. Above the threshold mutation P_c, $\Delta\rho$ increases with P_{mut}, illustrating the advance to complete collapse of the immune system which characterizes AIDS.

From the variation of the cell density in Figure 3, it is interesting to see the contrast in the growth and decay of virus and helper T-cell density count, with ($P_{mob} = 1$) and without ($P_{mob} = 0$) the cell mobility. One may treat the cell densities or their density difference, $\Delta\rho_o = \rho_V - \rho_H$, as an order parameter to analyze the type of transition at the threshold mutation. A close examination of the variation (Figure 4), suggests a continuous phase transition for $P_{mob} = 0$, i. e.,

$$\Delta\rho_o \simeq A\Delta P^{\beta}, \tag{3}$$

where $\Delta P = P_{mut} - P_c$ and A a constant. In presence of cell mobility ($P_{mob} = 1$), the phase transition is smeared out (see Fig. 4). From a log-log plot (Figure 5) of the variation of the order parameter ($\Delta\rho_o$) with ΔP, we estimate the exponent $\beta \simeq 0.574 \pm 0.016$. In the presence of mobility we find $\Delta\rho_o \simeq (5.976 \pm 0.013)\,\Delta P$.

In summary, effects of a local mediator for cellular interaction are considered to enhance the stimuli factors in an HIV immune response model, in-

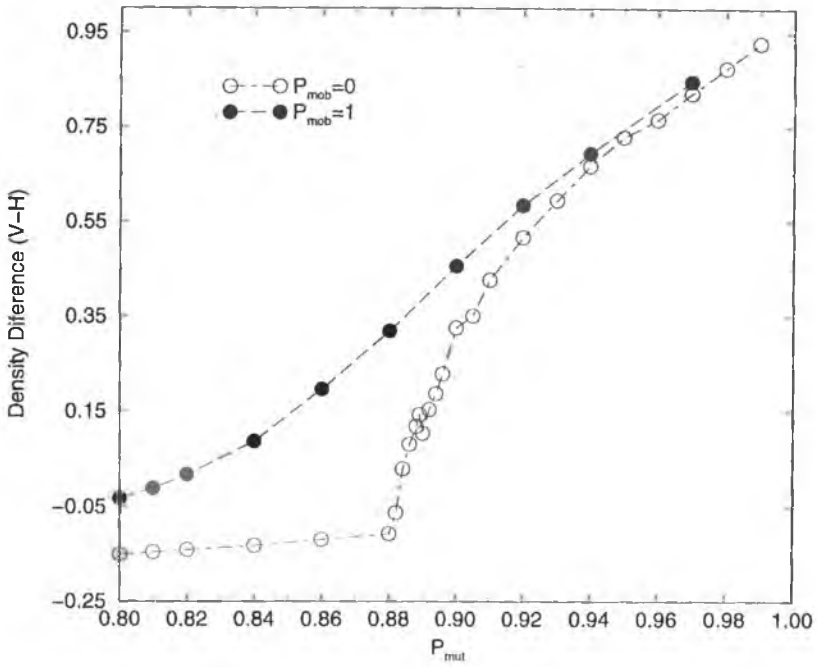


Fig. 4. Variation of the difference in cell density ($\Delta\rho = \rho_V - \rho_H$) with the mutation rate. (Statistics are as for Fig. 2 and 3).

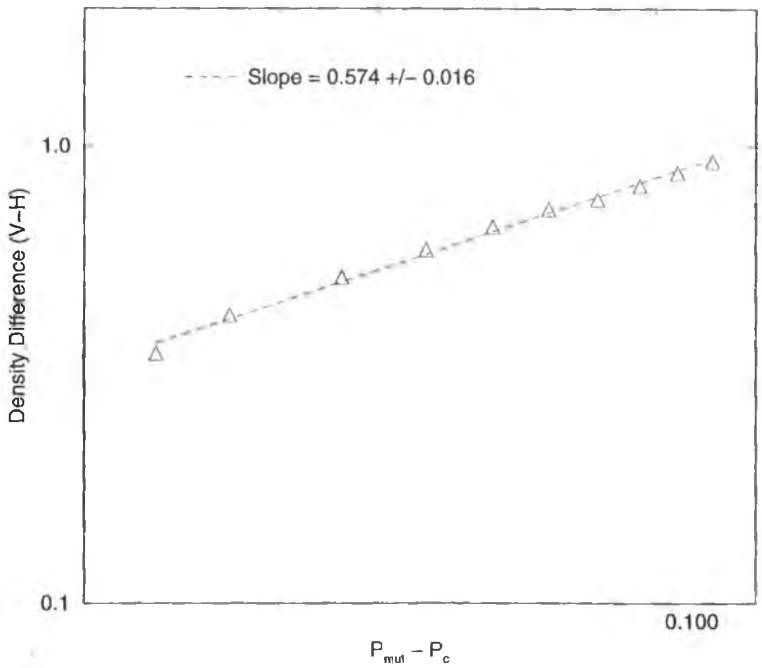


Fig. 5. Immuno-deficient order parameter ($\Delta\rho_o$) versus ($P_{mut} - P_c$) above the mutation threshold ($P_c = 0.884$) on a log-log scale for $P_{mob} = 0$. (Statistics are as for previous Figs).

vestigated by an MC simulation. Enhanced local interaction changes the pattern of infection progression. Growth of cell populations are studied as a function of viral mutation rate for mobile ($P_{mob} = 1$) and immobile ($P_{mob} = 0$) cells. For mutation below the threshold (P_c), the helper cells control the immune system, while viral growth weakens the immune response above P_c. The transition from immuno-competent state to immuno-deficient state and its progression depends on P_{mob}. In the absence of mobility, $P_{mob} = 0$, $\Delta\rho = \rho_H - \rho_V$ is positive and constant (immuno-competent) at P_{mut} below the threshold ($P_c = 0.884$). Viral density continues to grow and helper T-cell density count continues to decay above the threshold mutation. The transition from immuno-competent to immuno-deficient state is less extreme than for $P_{mob} = 1$ (Fig. 3) and is characterized by an exponent $\beta \simeq 0.574 \pm 0.016$. In the presence of mobility, on the other hand, the difference in helper cell and viral densities ($\Delta\rho$) continues to decrease with mutation below the thresold mutation ($P_c = 0.82$). The immuno-deficient order parameter ($\Delta\rho_o$) increases linearly with the mutation above the threshold. Thus, cell mobility is important in controlling the threshold and the growth pattern of cells, i. e., the progression of infection.

Acknowledgements

The authors would like to thank D. Stauffer for discussion. This work is also a part of a DOE-EPSCoR grant; support from Alexander von Humboldt foundation is acknowledged by RBP. Support from EI/SC/1999/104 grant is also acknowledged by HIR.

References

1. A. S. Perelson and G. Weisbuch, Rev. Mod. Phys. 69, 1219 (1997).
2. I. Dayan, D. Stauffer, and S. Havlin, J. Phys. A 21, 2473 (1988); A. U. Neumann, Physica A 162, 1 (1989).
3. S. Moss de Oliveira, P. M. C. de Oliveira, and D. Stauffer, "Evolution, Money, War and Computers", Teubner (Stuttgart and Leipzig, 1999) Chapter 3.
4. R. B. Pandey, in "Scientific Computing in Europe, SCE 1996", Sept. 2–4, 1996, Dublin, Ireland, edited by H. J. Ruskin, R. O'Connor, and Y. Feng.
5. R. Z. dos Santos, Ann. Rev. Comp. Phys. vol. VI, edited by D. Stauffer, World Scientific Singapore, 1999, p. 159.
6. D. Chowdhury, M. Sahimi, and D. Stauffer, J. Theor. Bio. 152, 263 (1991).
7. D. Stauffer and R. B. Pandey, Computers in Physics 6, 404 (1992).
8. R. B. Pandey, Physica A 179, 442 (1991).
9. F. Castiglione, M. Bernaschi, and S. Succi, Int. J. Mod. Phys. C 8, 527 (1997).
10. E. Ahmed, J. Stat. Phys. 85, 291 (1996).
11. K. Kaneko, Physica D 103, 505 (1997) and references therein.
12. A. Mielke and R. B. Pandey, Physica A 251, 430 (1998).
13. R. B. Pandey, Theor. Biosci. 117, 32 (1998).
14. R. Mannion, H. Ruskin, and R. B. Pandey, Theor. Biosci. (accepted for publication, 1999).
15. A. S. Fauci, G. Pantaleo, S. Stanley and D. Weissmann, Ann. Internal Medic. 124, 654 (1996).
16. P. D. Greenberg and S. R. Riddell, Science 285, 546 (1999).

Figure Captions:

Figure 1: Typical snapshots of helper T-cells (square) and virus (star) at time steps $t = 75(a), 149(b), 223(c), 297(d)$ for $P_{mut} = 0.90$ in absence of cell mobility on a 100×100 lattice.

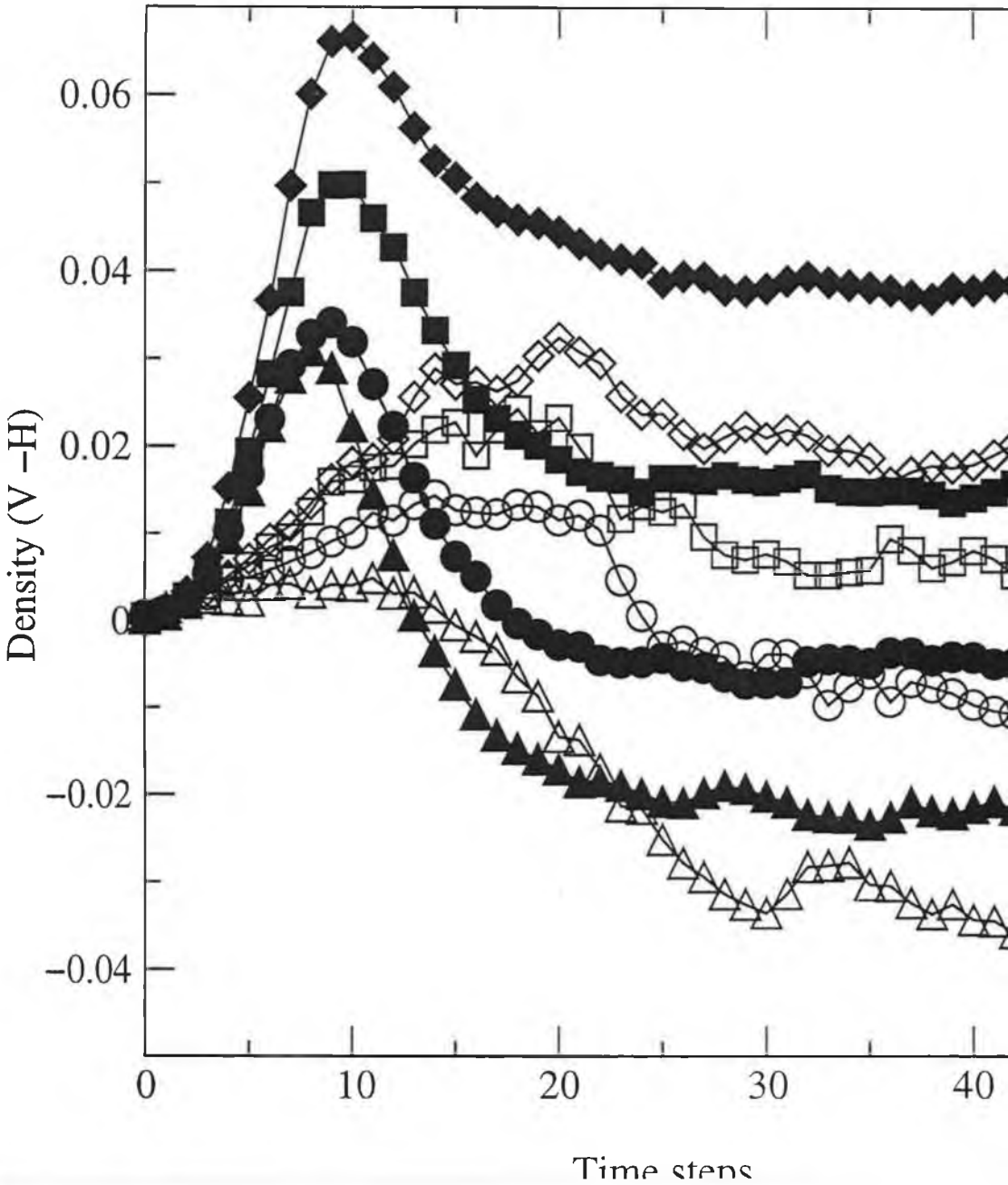
Figure 2: Number of cells (helper T-cells and virus) versus time step at various mutation rates P_{mut} with the cellular mobility $P_{mob} = 0.0(a), 1.0(b)$. Sample of size 100×100 is used with up to 50 independent runs.

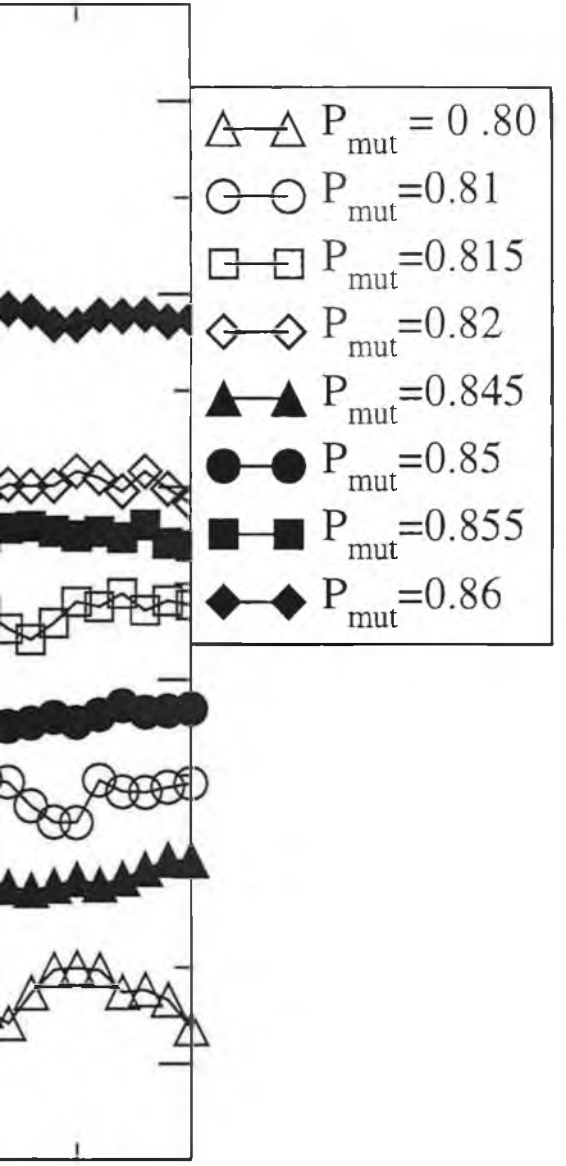
Figure 3: Equilibrium densities of helper T-cells (ρ_H) and virus (ρ_V) versus mutation rate (P_{mut}). (Statistics are as for Figure 2).

Figure 4: Variation of the difference in cell density ($\Delta\rho = \rho_V - \rho_H$) with the mutation rate. (Statistics are as for Fig. 2 and 3).

Figure 5: Immuno-deficient order parameter ($\Delta\rho_o$) versus ($P_{mut} - P_c$) above the mutation threshold ($P_c = 0.884$) on a log-log scale for $P_{mob} = 0$. (Statistics are as for previous Figs).

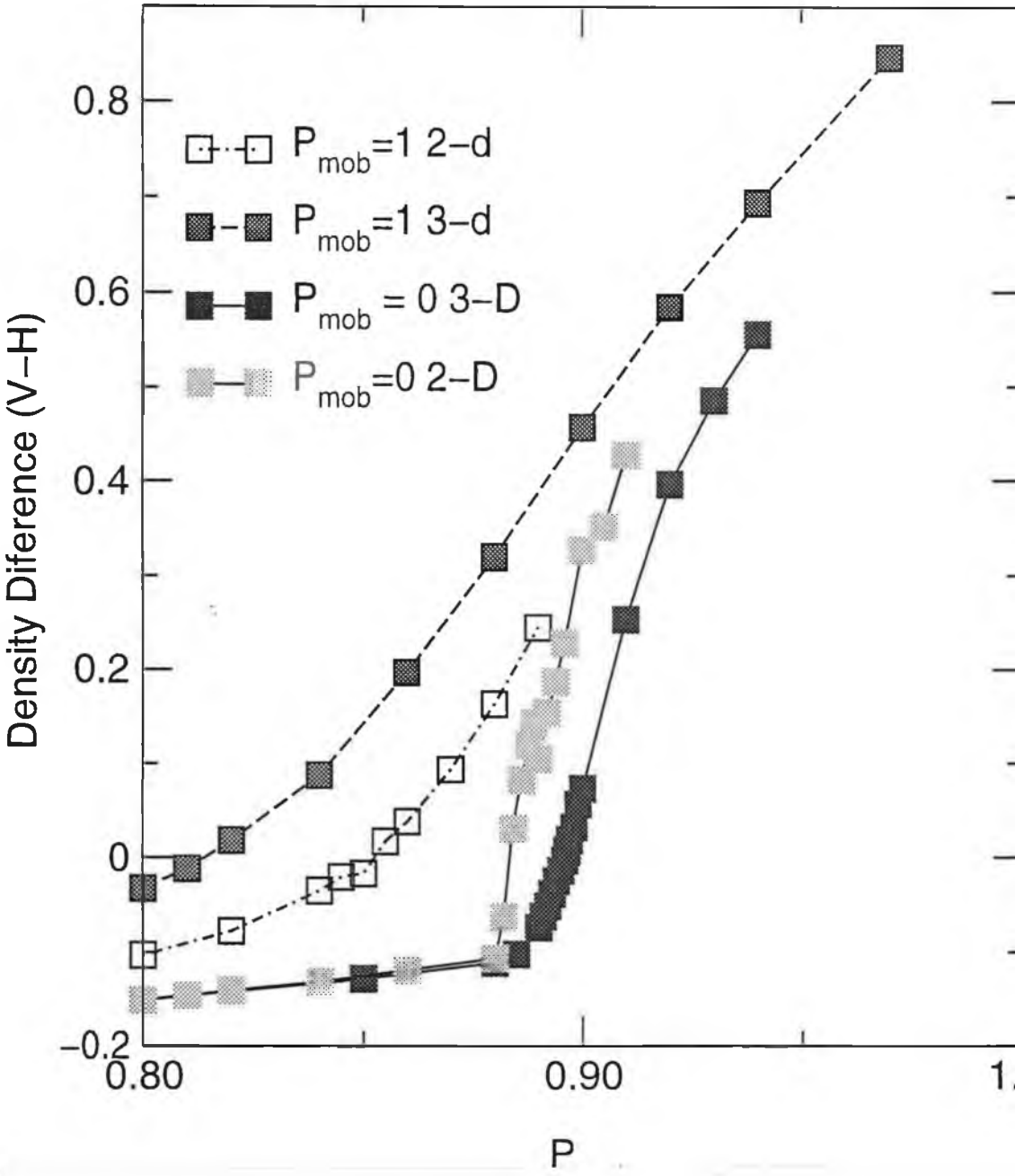
Figure 1




P mut = 0 .80
P mut =0.81
P mut =0.815
P mut =0.82
P mut =0.845
P mut =0.85
P mut =0.855
P mut =0.86

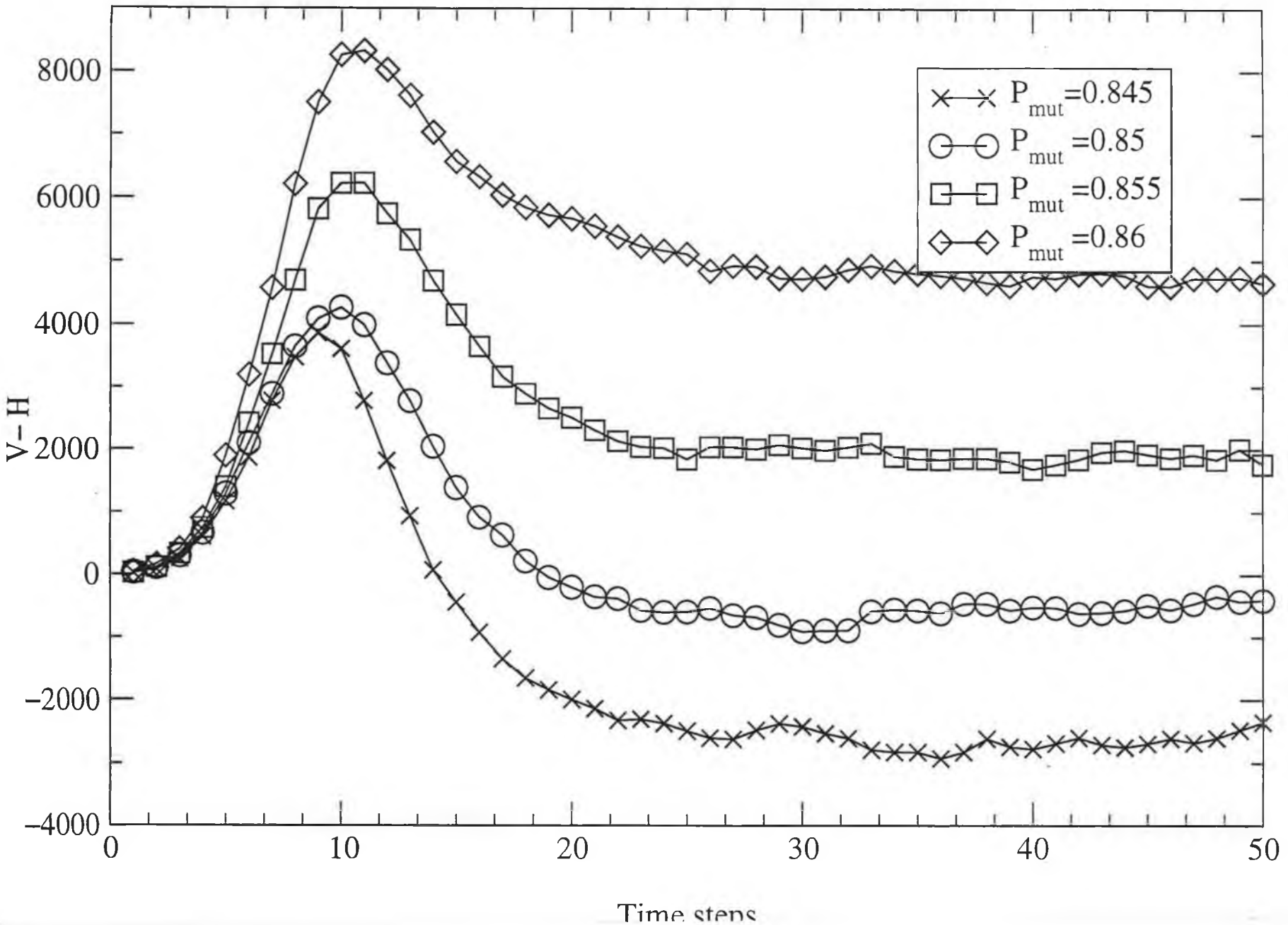
50

Figure 2


0.8
0.6
0.4
0.2
0
−0.2
0.80
0.90
P
Density Diference (V−H)
P_mob=1 2−d
P_mob=1 3−d
P_mob = 0 3−D
P_mob=0 2−D

.00

Figure 3


8000
6000
4000
2000
0
−2000
−4000
V − H
0
10
20
30
40
50
Time steps
P mut =0.845
P mut =0.85
P mut =0.855
P mut =0.86

Theory Biosci. (2000) 119: 10–19

http://www.urbanfischer.de/journals/theorybiosc

Effect of Mutation on Helper T-Cells and Viral Population: A Computer Simulation Model for HIV

Rachel Mannion[1], Heather Ruskin[1] and R. B. Pandey[2]

[1]School of Computer Application, Dublin City University, Dublin, Ireland
[2]Department of Physics and Astronomy, University of Southern Mississippi, Hattiesburg, MS 39406–5046

Address for correspondence: Rachel Mannion, School of Computer Application, Dublin City University, Dublin 9, Ireland

Received: August 17, 1999; accepted: Oktober 25, 1999

Key words: Monte Carlo simulation, Immune Response, HIV, viral matation, intra/inter cell interactions

Summary: A Monte Carlo simulation is proposed to study the dynamics of helper T-cells (N_H) and viral (N_V) populations in an immune response model relevant to HIV. Cellular states are binary variables and the interactions are described by logical expressions. Viral population shows a nonmonotonic growth before reaching a constant value while helper T-cells grow to a constant after a relaxation/reaction time. Initially, the population of helper cells grows with time with a power-law, $N_H \sim t^{\beta}$, before reaching the steady-state; the growth exponent β increases systematically ($\beta \simeq 1-2$) with the mutation rate ($P_{mut} \simeq 0.1-0.4$). The critical recovery time (t_c) increases exponentially with the viral mutation, $t_c \simeq Ae^{\alpha P_{mut}}$, with $\alpha = 4.52 \pm 0.29$ in low mutation regime and $\alpha = 15.21 \pm 1.41$ in high mutation regime. The equilibrium population of helper T-cell declines slowly with P_{mut} and collapses at ~ 0.40; the viral population exhibits a reverse trend, i. e., a slow increase before the burst around the same mutation regime.

Introduction and Motivation

A considerable interest has been recently directed toward modeling the population dynamics of cells in an immune response to HIV. In computational modeling of population dynamics, two approaches have emerged in last decade: continuum [1–4] and discrete methods [5–16]. The discrete lattice methods have become increasingly popular due to their simplicity in direct implementations particularly with the rule-based cellular interactions. Cellular Automata (CA) [6–8] approach has been frequently used in recent years to study the population dynamics of cells in a variety of immune response models [10]. In most CA approaches [7, 10, 15], parts of

the rule based intersite cellular interactions are implemented simultaneously to all cells synchronously in order to assign their temporary intermediate states which are then followed by a set of inter-cellular interaction rules at each lattice site to update their states at the next time step [11]. Recently, the mobility of cells are considered stochastically in a stochastic CA (SCA) [17] in order to take into account cellular motility along with synchronous update with independent rules as in traditional CA. In this article we present a computer simulation method in which the cellular states resulting from the cellular interaction are updated stochastically and randomly with one update at a step (see below). Cell's mobility is also considered stochastically with a motility criteria as before [17]. The direct Monte Carlo (MC) simulation presented here is, thus, an alternate approach to CA methods and is a viable approach to study population growth in systems as complex as HIV immune response [18, 19]. In the following; the model is described with a simple interaction already used before [17] to study the effect of viral mutation on the population of helper cells.

Model

We consider a cell mediated immune response with four cell types: macrophages (M), helper T-cells (H), cytotoxic T-cells (C), and antigen/virion (or virus carrying cells) (V). The cellular states are described by a binary variable, i.e., their high concentration by "1" and low concentration by "0". As before [11], a set of boolean expressions can be used to describe the cellular interaction,

$$M(t+1) = M(t).or.V(t), \tag{1(a)}$$

$$H(t+1) = [M(t).or.H(t)].and.[notV(t)], \tag{1(b)}$$

$$C(t+1) = M(t).and.H(t).and.V(t), \tag{1(c)}$$

$$V(t+1) = [H(t).or.M(t).or.V(t)].and.[notC(t)], \tag{1(d)}$$

where the states of the four cell types at time $t+1$ are evolved from their states at time t. Equation (1(a)) describes the growth of macrophages which will be in their high concentration ($M(t+1) = 1$) state at time step $t+1$ if they were already in this state ($M(t) = 1$) (a self-propagating interaction) or if a viral infected cell was present ($V(t) = 1$) or both macrophages and virus are present at time t. Other equations (1(b–d)) refers similar growth conditions for M, H and C [11].

The four cell types lead to sixteen configurations. This interaction set was analyzed in detail [11] with a mean field approach where all cells of each cell type behave in the same way regardless of their location in space, an infinite range interacting system [5]. Iterating the above equation from a

random configuration leads to a flow diagram. One may easily check [11] that there are two fixed points, an immunocompetent (absence of all activated cell types) and an immunodeficient (presence of macrophages and virus, and absence of helper and cytotoxic T-cells), and a cycle of period two among the "infected" (presence of macrophages, helper cells, and virus, and absence of cytotoxic cells), "severely infected" (presence of all but the helper T-cells), and "susceptible" (absence of all but the activated macrophages) configurations. Thus, this interaction captures some general characteristics of immune response in HIV infection [18, 19].

We consider a discrete lattice of size $L \times L$ to incorporate the effects of mediators, growth factors, effectors etc. [18, 19] via local interactions. Initially, a small number of each cell type (typically one of each), are randomly distributed among a fraction p of the lattice sites. A site can be occupied by four different cell types, however, more than one cell of one type is not allowed at a site. A site i is referred as occupied by a cell type c if the state of the cell type c is 1 (high concentration); the cellular state '0' is referred as an empty site for the cell type. The number of cells grow and decay as we implement the cellular interactions and update their states using the following steps:

(A) Random sequential update of cellular state:
- (*i*) select a site i randomly.
- (*ii*) find the sum $s_i(c)$ of each cell type (c) over their four neighbors and the cell at site i: if $s_i(c)' \geq 1$ then assign an intermediate state $s_i(c)' = 1$ otherwise $s_i(c)' = 0$.
- (*iii*) Using the current state $s_i(c)'$ of cells at site i, implement the inter-cellular interaction (eq. (1)) among different cell types, to update their state, i. e., M, H, C and V at site i.
- (*iv*) Repeat steps (*i*)–(*iii*) L^2 times.

(B) Mutation: viral mutation is considered probabilistically in the preceding step A(*iii*) where inter-cell interactions (eq. (1)) are implemented. Because a virus cannot be recognized by the host cells, the viral state is set to $V = 0$ with the mutation probability P_{mut} in eq. 1(a) and (c). This enhances the advantage of viral action over the host cells.

(C) Random sequential move with immunological motility criteria: we set the mobility rate P_{mob} in the beginning of simulation, $0 \leq P_{mob} \leq 1$ where $P_{mob} = 0$ means no mobility and $P_{mob} = 1$ describes highest mobility. The following steps (*i–viii*) are performed with probability P_{mob} to implement the cell mobility [17]:
- (*i*) select a site, say, i randomly,
- (*ii*) select a cell type (c) at the site i randomly.
- (*iii*) IF the cell type c is present at site i, **THEN**
- (*iv*) select one of the nearest neighbor site j;

(*v*) IF the cell type c is absent at site j, **THEN**
(*vi*) attempt to move the cell type c from site i to site j. In order to accept the move further specific motility criterion [17] must be satisfied, namely, site j must have a viral infected cell, i. e., $V = 1$ for macrophages and cytotoxic cells to move while either macrophages or helper cells must be present (M or V), (note this is a less restrictive criteria than in [IF]) at site j for virion to move to site j.
(*vii*) Go to step (*i*) in case any of the above "IF" fails.
(*viii*) Repeat steps (*i*)–(*vii*) $4 \times L^2$ times.

The above procedures (*A*–*C*) carried out sequentially defines a unit Monte Carlo step (MCS). Simulation is performed for a fixed number of time steps with a number of independent runs for each mutation probability with different mobility rates. The hopping procedure and mutation mechanism adopted here are specific to this model and may depend on the type of immune response. For example, helper T-cells move to their neighboring empty sites with and without presence of other cell types. It is envisaged that the helper cells have more mobility since they play a key role in orchestrating the immune response.

Results and Discussion

Simulations are performed mostly on a 100×100 lattice with a very low initial concentration of each cell type, typically one of each distributed randomly. Different sample sizes are used to check for severe finite size effects. The qualitative results presented here are independent of the lattice sizes within the statistical fluctuations. We vary the mutation rate, $P_{mut} = 0.0 - 0.50$ for mobility $P_{mob} = 0$ and 1. Up to 50 independent samples are used to measure the average number of cells. We monitor the population of each cell type with time steps as a function of mutation rate. Population of macrophages reaches its maximum constant value in a rather short time steps, therefore, we focus mainly on the population of helper T-cells and virus.

Figure 1 shows the evolution of cells (H, V) population as a function of viral mutation rate. In absence of viral mutation ($P_{mut} = 0.0$), we see that the population of helper cells (N_H) increases rather fast to a constant value (N_{HS}). Let us define the recovery response period (a relaxation time) as the number of time steps (τ) needed to approach the equilibrium population value. Increasing the viral mutation increases this relaxation time while reducing the equilibrium population of helper cells (N_{HS}). Viral population, on the other hand, grows nonmonotonically, a rapid increase is followed by a decay before reaching a constant equilibrium value (N_{VS}). The equilibrium viral population (N_{VS}) increases, though slowly, with the

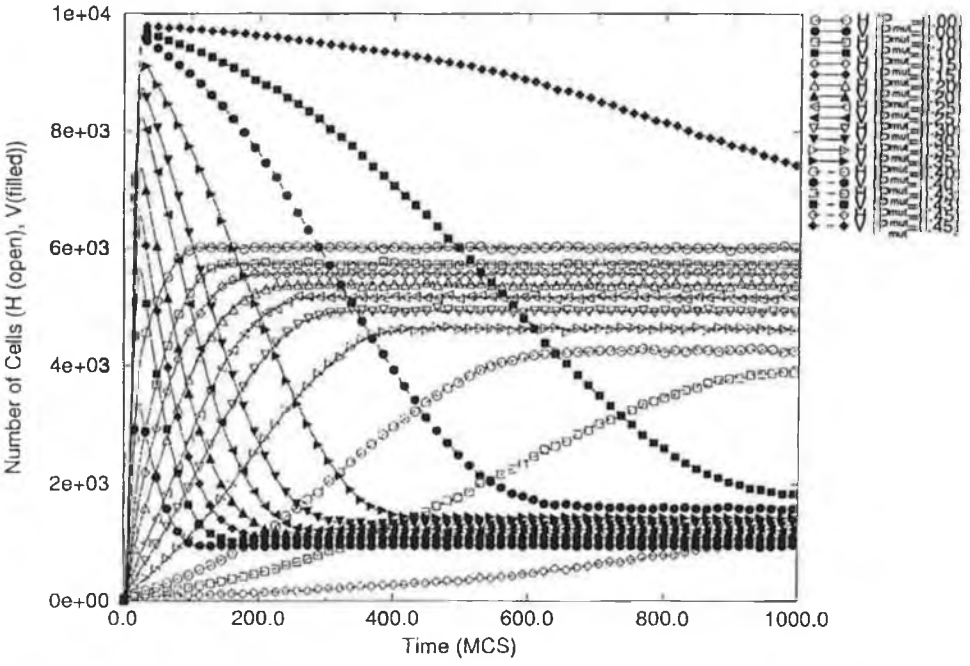


Figure 1 a

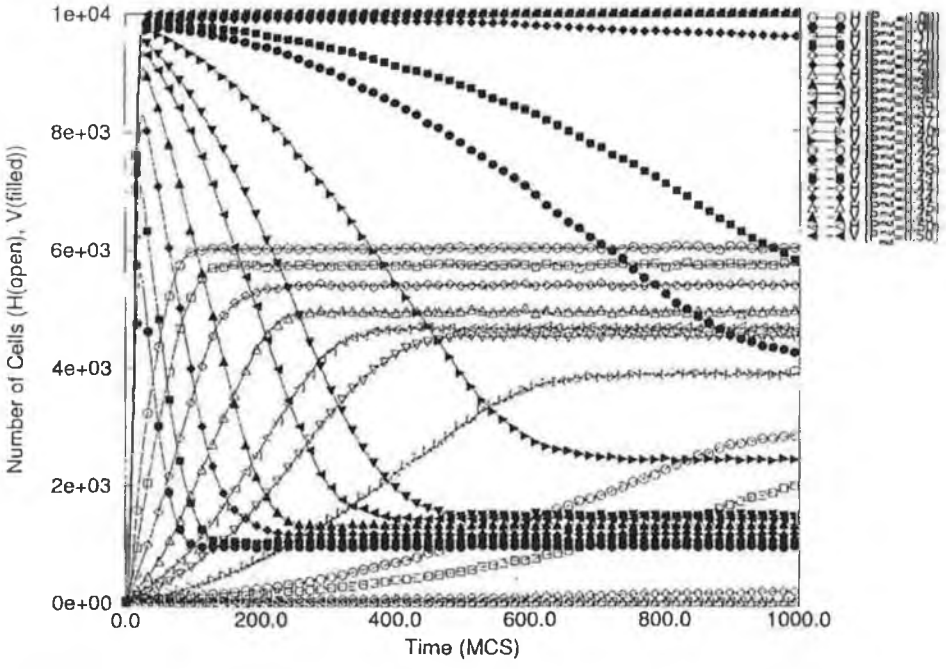


Figure 1 b

Fig. 1. Number of cells (helper T-cells and virus) versus time step at various mutation rate $P_{mut} = 0.0 - 0.5$ with the cellular mobility $p_{mob} = 0.0(a)$, $1.0(b)$. Sample of size 100×100 is used with 10 independent runs.

mutation rate. Beyond a certain threshold of mutation rate ($P_{cmut} \sim 0.4$), the viral population grows much faster while the helper T-cell populations deplete. Although, the qualitative growth pattern of cell populations remain unchanged by incorporating the mobility (fig 1(b)), the growth of viral population is further enhanced by mobility while the growth of T-cells population is somewhat reduced. It is worth pointing that unlike recent results of an stochastic cellular automata (SCA) approach [17], we do not observe oscillation in cell populations. It is rather easy to identify the equilibrium value of the cell populations in our MC approach. The variation of the equilibrium values with the mutation rate may be useful in understanding better the response and growth process as follows.

One of the most difficult and frequently asked questions in such simulations is, how does the number of time steps is related with the real time. It is tempting to make a crude estimate via qualitative comparison with the clinical data. From the variation of HIV RNA copies in plasma, culturable plasma viremia, and CD4 cell counts with weeks and years reported by Fauci et al. [17], it appears that the non-monotonic growth pattern of virus population in figure 1 could be compared with the variation of HIV RNA copies or viramia. With this assumption, 12 weeks could be of the order of 300 MCS time i. e., one time step is equivalent to six hours. We should caution however, that the viral mutation rate, an independent (constant) should

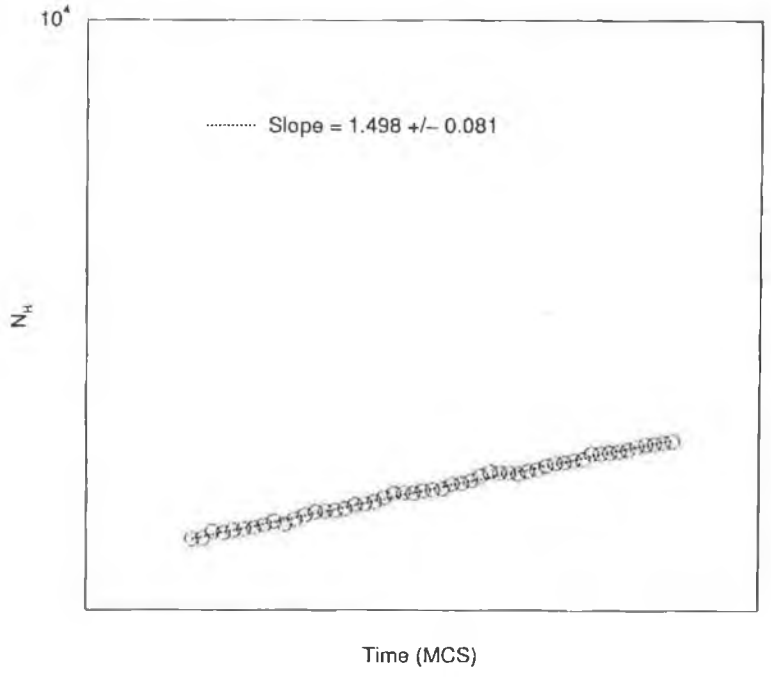


Fig. 2. Typical variation of the helper T-cell population with time steps in early stage of response (data from fig. 1(b) for $P_{mut} = 0.37$ in the range of 200–350 MCS) on a log-log scale.

be factored in making a better estimate of time. At present we do not know, how the viral mutation used here depends on time in clinical finding. Therefore, it is difficult to compare the growth rate with time step alone without considering the mutation rate as we see below.

Rate of growth of activated helper T-cells would be an interesting quantity to monitor during the progression of initial infection. We can study the growth rate of helper cells by analyzing the data in a range where population shows a well defined dependence. Figure 2 shows the variation of the helper cell population with time step less than τ on a log-log scale. The linear variation of the data in the intermediate time regime suggests the possibility of a power-law growth of the T-cell population (N_H) with time (t), i. e.,

$$N_h \sim t^{\beta} \tag{2}$$

We estimate the values of the growth exponent β at various mutation rate (see table 1). Note that the exponent increases systematically ($\beta \simeq 1-2$) with the mutation rate ($P_{mut} \simeq 0.10 - 0.43$). This implies that the helper cells grow faster as the virus mutates, but levels off faster as they conquered by the virus.

Table 1 Growth exponent β

P_{mut}	β
0.10	1.00
0.20	1.21
0.30	1.40
0.35	1.44
0.37	1.50
0.40	1.63
0.43	1.93

From figure 1, we see that the viral population is larger than the helper T-cells initially, but the helper cells grow and overtakes the viral population at $P_{mut} \simeq 0.0 - 0.40$. Let us define a critical recovery time (t_c) in which the population of helper cells (N_H) becomes larger than the viral population (N_V). We find that the critical time, t_c, depends on the mutation rate. Figure 3 shows a t_c versus P_{mut} on a normal-log scale, which suggests exponential dependence,

$$t_c \simeq A e^{\alpha P_{mut}}$$

with $\alpha = 4.52 \pm 0.29$ in low mutation regime and $\alpha = 15.21 \pm 1.41$ in high mutation regime. Thus, there is a crossover from a relatively slow recoverable time period to a collapse regime at around $P_{mut} \sim 0.4$ when t_c rises much faster. This means that the highly active anti-retroviral therapy (HAART) has to be administered before the crossover mutation rate develops in order to sustain the recovery.

In order to see the relative progression of cell counts, we present the variation of equilibriun cell density with the mutation rate in figure 4. We see a slow decline in T-cell density and increase in viral density for P_{mut} up to 0.40 beyond which the viral density explodes while the T-cell density collapses. Around this mutation threshold, the population of T-cells and viruses fluctuates which may be a region where opportunistic infections may occur. Thus the range of low mutation rate ($P_{mut} \leq 0.33$) could

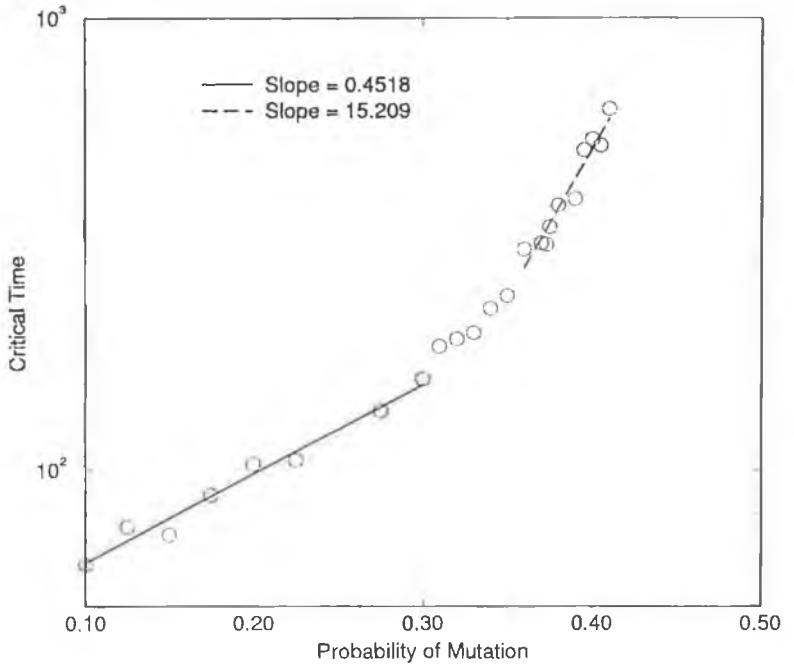


Fig. 3. Critical recovery time step versus viral mutation rate on a semi-log scale. Statistics is the same as in fig. 1(b).

be interpreted as the latent period of virions infection where host immune cells are able to control the recognizable virions (fig. 3 and 4). The crossover regime ($P_{mut} \simeq 0.35 - 0.41$) corresponds to prolonged period of infection where competition between host cells and virus becomes intense. This regime, though appears narrow in the mutation range, may corresponds to a long period (order of 5–8 years) in real time [18]. The range of mutation above the threshold ($P_{mut} \geq 0.40$) relates to an advanced stage [18] leading to death. We would like to emphasize that even a slight increase in mutation rate results in a change from a latent to an advanced state of disease.

In summary we have presented a MC simulation to study the growth of cellular elements in a cell-mediated immune response relevant to HIV infection. In contrast to cellular automata approaches, the oscillation in cellular population is vanished. We find that the viral mutation rate is very important in orchestrating the growth rates of cells and their equilibrium density. From the plot of recovery time with the mutation rate, we are able to provide an exponential growth pattern (eq. 3) which shows a crossover from a slow progression of infection to a rapid advance leading to collapse. These observations are consistent with our analysis of relative cell density count with the mutation rate – a slow decline of T-cell counts with some fluctuation before collapse with an opposite trend in viral density.

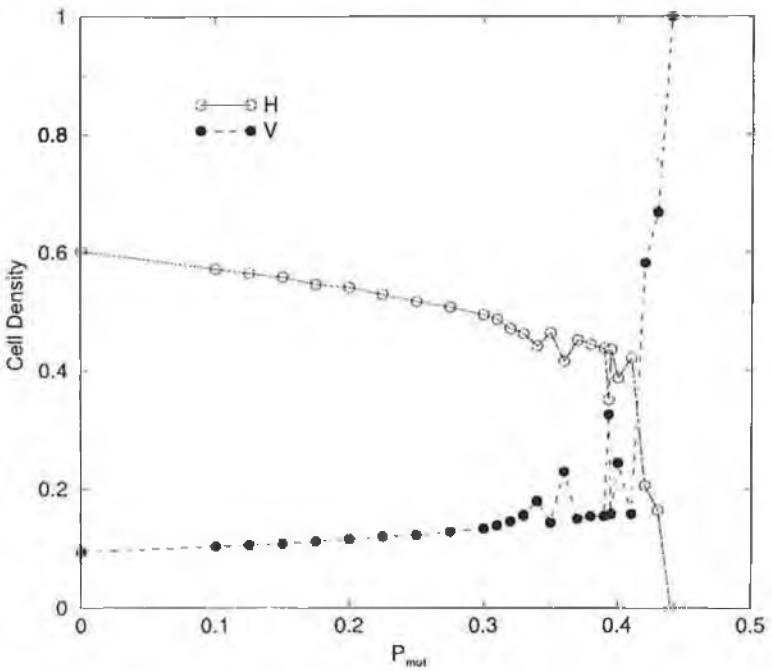


Fig. 4. Equilibrium density of helper T-cells and virus versus mutation rate with the same statistics as in fig. 1(b).

Acknowledgements

Author would like to thank D. Stauffer for useful discussion.

References

1. A. S. Perelson and G. Weisbuch, Rev. Mod. Phys. (1997).
2. K. Lippert and U. Behn, Ann. Rev. Comp. Phys. vol. V, edited by D. Stauffer, World Scientific Singapore, 1997, p. 287.
3. M. A. Nowak and C. R. M. Bangham, Science 272, 74 (1996); M. A. Nowak and A. J. McMichael, Sci. Am. 273, 58 (1995).
4. G. W. Hoffmann and M. D. Grant, in "Issues on Mathematical Modelling of AIDS, Lecture Notes in Biomathematics" edited by C. Castillo-Chavez (Springer, Berlin, 1989).
5. M. Kaufman, J. Urbain, and R. Thomas, J. Theor. Bio. 114, 527 (1985); G. Weisbuch and H. Atlan, J. Phys. A 21, L 189 (1988).
6. I. Dayan, D. Stauffer, and S. Havlin, J. Phys. A 21, 2473 (1988); A. U. Neumann, Physica A 162, 1 (1989).
7. S. Moss de Oliveira, P. M. C. de Oliveira, and D. Stauffer, "Evolution, Money, War and Computers", Springer Verlag (Heidelberg and Berlin, 1998) Chapter 3.
8. R. B. Pandey, in "Scientific Computing in Europe, SCE 1996", Sept. 2–4, 1996, Dublin, Ireland, edited by H. J. Ruskin, R. O'Connor, and Y. Feng.
9. D. Chowdhury, M. Sahimi, and D. Stauffer, J. Theor. Bio. 152, 263 (1991).
10. D. Stauffer and R. B. Pandey, Computers in Physics 6, 404 (1992).
11. R. B. Pandey, Physica A 179, 442 (1991).

12. F. Castiglione, M. Bernaschi, and S. Succi, Int. J. Mod. Phys. C 8, 527 (1997).
13. E. Ahmed, J. Stat. Phys. 85, 291 (1996).
14. K. Kaneko, Physica D 103, 505 (1997) and references therein.
15. R. Z. dos Sansos, Ann. Rev. Comp. Phys. vol. VI, edited by D. Stauffer, World Scientific Singapore, 1999, p. 159.
16. A. Mielke and R. B. Pandey, Physica A 251, 430 (1998).
17. R. B. Pandey, Theor. Biosci. 117, 32 (1998).
18. A. S. Fauci, G. Partako, S. Starley, and D. Weissmann, Ann. Internal Medic. 124, 654 (1996).
19. P. D. Greenberg and S. R. Riddell, Science 285, 546 (1999).

Effects of Viral Mutation on Cellular Dynamics in a Monte Carlo simulation of HIV immune response model in Three Dimensions

R. Mannion and H. J. Ruskin

School of Computer Applications
Dublin City University, Dublin 9, Ireland

R.B. Pandey

Department of Physics and Astronomy
University of Southern Mississippi, Hattiesburg, MS 39406-5046

Abstract: The cellular dynamics of HIV interaction with the immune system is explored in 3-dimensions using a direct Monte Carlo simulation. Viral mutation with probability, P_{mut}, is considered with immobile and mobile cells. With immobile cells, viral population becomes larger than that of the helper cells beyond a latency period T_{crit} and above a mutation threshold P_{crit}. That is at $P_{mut} \geq P_{crit}$, $T_{crit} \propto (P_{mut} - P_{crit})^{-\gamma}$, with $\gamma \simeq 0.73$ in three dimensions and $\gamma \simeq 0.88$ in 2D. Very little difference in P_{crit} is observed between two and three dimensions. With mobile cells, no power-law is observed for the period of latency, but the difference in P_{crit} between two and three dimensions is increased. The time-dependency of the density difference between Viral and Helper cell populations $(\rho_V - \rho_H)$ is explored and follows the basic pattern of an immune response to infection. This is markedly more defined than in the 2-D case, where no clear pattern emerges.

Introduction: There have been many models of cellular populations using rule-based interactions, in particular those of Cellular Automata(CA), probabilistic CA (PCA), and stochastic CA (SCA), (Perelson & Weisbuch, 1997; Dayan et al, 1988; de Oliveira et al, 1999; Pandey, 1996; Chowdury et al , 1991; Stauffer & Pandey , 1992; Pandey, 1991; Castilglione, 1997; Ahmed, 1996 ; Kaneko, 1997; Mielke & Pandey ,1998; Pandey, 1998) . Most CA approaches (Dayan et al, 1988; de Oliveira et al, 1999; Pandey, 1996; dos Sansos, 1999) use synchronous updating, thereby ensuring that the cellular interactions at each site are implemented simultaneously. The mobility of cells was recently considered via an SCA (Pandey, 1998) model with traditional CA synchronous updating. Other models have also recently been investigated, where asynchronous rather than synchronous updates occur (Mannion et al 1999a; Mannion et al 1999b). The sites in this approach are chosen randomly and cellular interactions are then implemented with immobile and mobile cells. Very recently the effects of stimuli were enhanced by increasing the interaction among neighbouring cells (Mannion et al; 1999b). However, this study was performed in two dimensions, whereas the host space, (the lymph node and other secondary lymphoid organs, where antigen driven responses occur), is closer to three dimensions. We therefore extend the earlier study here to three dimensions. Although we do not find a dramatic change in cellular growth patterns in going from 2-D to 3-D, the fluctuations in data (for cell populations) are reduced. The model is presented below, with interaction sets described, which incorporate probabilistic mechanisms for mobility and mutation, described. The cell populations are discussed as functions of mutation and mobility and also comparitively with the previous 2-D findings.

Model:

As before, we consider a cell mediated immune response with four cell types: macrophages (M), helper T-cells (H), cytotoxic T-cells (C), and antigen/virion (or virus carrying cells) (V) each with a binary cellular state to represent their high ("1") and low ("0") concentrations. A set of boolean expressions can be used to describe the intra-site cellular interactions [8],

$$M(t+1) = M(t).or.V(t), \tag{1(a)}$$

$$H(t+1) = [M(t).or.H(t)].and.[notV(t)], \tag{1(b)}$$

$$C(t+1) = M(t).and.H(t).and.V(t), \tag{1(c)}$$

$$V(t+1) = [H(t).or.M(t).or.V(t)].and.[notC(t)], \tag{1(d)}$$

where the states of the four cell types at time $t+1$ evolve from their states at time t. This interaction set, (Eq. 1) has been previously analyzed in detail with a mean field approach (Pandey, 1991). This analysis found that, starting with any random configuration of the sixteen possible configurations and applying the above equations, leads to a flow diagram

with two fixed points, which have a cycle of period two. The fixed points represent states of "immunocompetency" and "immunodeficiency", while the cycle includes infected, severely infected, and susceptible states.

In order to incorporate the effects of mediators, growth factors, effectors (Mannion et al 1999a, Mannion et al 1999b) via local interactions and to study the population of cells, we consider a discrete simple cubic lattice of size $L \times L \times L$. The model is initiated with a random distribution of a single M and a single V. A site can be occupied by four different cell types, with a maximum of one cell of each type possible at a site. A site i is referred to as occupied by a cell type ic if the state of the cell type ic is 1 (high concentration); the cellular state '0', (low concentration), is referred to as an empty site for the cell type. The cell populations change as we implement the cellular interactions and update their states using the following steps:

(A) Random sequential update of cellular state:

(i) select a site i randomly.

(ii) find the sum $s_i(ic)$ of each cell type (ic) over the neighboring sites (six neighbors for 3-D) and the cell at site i: if $s_i(ic) \geq 1$ then assign an intermediate state $s_i(ic)' = 1$ otherwise $s_i(ic)' = 0$.

(iii) with these intermediate states ($s_i(ic)' = M', H', C', V'$ for $ic = 1, 2, 3$, and 4 respectively), evaluate the corresponding stimulated states ($s_i(ic)'' = M'', H'', C'', H''$), using the following relations,

$$M'' = M'.or.V' \qquad (2(a))$$

$$H'' = H'.or.M' \qquad (2(b))$$

$$C'' = C'.or.H' \qquad (2(c))$$

$$V'' = V'.or.H' \qquad (2(c))$$

Note that this set of interactions is implemented in addition to interaction (1), in the following step (iv), to take into account IL2, cytokines and other effectors which enhance the reaction.

(iv) Using the current state $s_i(ic)''$ of cells at site i, implement the inter-cellular interaction (Eq. (1)) among different cell types, to update their state, i.e., M, H, C and V at site i.

(v) Repeat steps $(i) - (iv) L^3$ times.

(B) Mutation: viral mutation is considered probabilistically in the preceding step $A(iii)$ where inter-cell interactions (Eq. (1)) are implemented. With probability (P_{mut}), a virus mutates such that the host cells no longer recognize it; then V=0 in Eq (1(a), 1(c)).

(C) Random sequential move with immunological mobility: we set the mobility rate P_{mob} in the beginning of simulation, $0 \leq P_{mob} \leq 1$ where $P_{mob} = 0$ represents no mobility and $P_{mob} = 1$ describes highest mobility. The following steps $(i-viii)$ are implemented with probability P_{mob}:

(i) select a site i randomly,

(ii) select a cell type (ic) at the site i randomly.

(iii) **IF** the cell type ic is present at site i, **THEN**

(iv) select one of the nearest neighbor sites j;

(v) **IF** the cell type ic is absent at site j, **THEN**

(vi) With probability P_{mob} attempt to move the cell type ic from site i to site j. In order to accept the move, further specific criteria [16] must be satisfied, namely, site j must have a viral infected cell, i.e., $V = 1$ for macrophages and cytotoxic cells to move, while both macrophages as well as helper cells must be present $(M = H = 1)$ at site j for virion to move to site j.

(vii) Go to step (i) in case any of the above "IF" conditions fail.

$(viii)$ Repeat steps $(i) - (vii)$ $4 \times L^3$ times.

The above procedures, $(A - C)$, carried out sequentially, define a unit Monte Carlo step (MCS). We perform the simulation for a fixed number of time steps with a number of independent runs for each mutation probability and for maximum and zero mobility.

Results and Discussion: Simulations are performed mostly on a 3-dimensional $50 \times 50 \times 50$ lattice with a very low initial concentration of M and V, typically one of each distributed randomly. Different lattice sizes i.e 20^3 and 30^3 are also used to check for finite size effects. The qualitative results presented here are independent of sample size, unless specified. The growth patterns of the Helper and Viral Cells in three dimensions are similar to patterns in 2-D simulations (see Fig. 1) . The time-dependent growth of the density difference between Viral and Helper populations, i.e. $\rho_V - \rho_H$, was investigated. This represents the fight for control between the immune system and the viral invader. A typical variation with timestep is presented in Figure 1 and is illustrated for $P_{mob} = 1$ which exhibits the more interesting features. The virus reaches a peak approximately ten timesteps after the start of initial infection. This corresponds to the early infection period, where the invader exploits the immune system's lack of awareness of infection to gain the upper hand. This peak is then followed by a decrease in viral dominance to an oscillating equilibrium, which represents the immune system's mounting response. This oscillating equilibrium is governed , as expected, by P_{mut}. Below a certain mutation level P_{crit} , this equilibrium is negative, representing Helper dominance. Mutation levels larger than P_{crit} result in a positive equilibrium, representative of viral dominance.

We see, for increasing P_{mut}, an increase in peak amplitude in the initial viral attack together with a longer time lag to reach the equilibrium state of either immune or viral control. This agrees with what is known clinically about infection at an early stage (Fauci et al, 1996). The time-delay for the immune response, evident in 3-D, is not as clear in 2-D and the oscillations in the equilibrium density difference are larger. The 3-D curves have a characteristic well-defined shape compared to 2-D , due to averaging over additional neighbours . In the absence of mobility in three dimensions, $P_{crit} = 0.887$ (slightly increased from the 2-D value of 0.884) and with extreme mobility $P_{crit} = 0.815$ (compared with 0.852 in 2 dimensions). The phase transitions between immune-competency and immune-deficiency are very similar in both dimensions, with the 3-D transitions being smoother, as evident from Figure 2.

We also consider the measurement of the "latency" period as predicted by the simulation. Latency is the length of time taken for the initial HIV infection to develop into AIDS. While the term does not accurately reflect cell behaviour, (as the virus is replicating and mutating) it is used to describe the phase of the disease progression for which no macroscopic effects are evident. Latency is a characteristic of every HIV infection, whether of length 2-3 years (rapid progressors), 7-11 years (typical progressors) or more than 20 years(longterm non-progressors) (Fauci et al, 1996). We attempt to measure the latency period of our simulation by counting the number of timesteps taken by the viral population to dominate the helper population (see Figure 3). We define a parameter P_{crit} - P_{mut} (P_Δ),which we plot against Monte Carlo timesteps(t_{mc}), where $P_{mut} \geq P_{crit}$, (as $P_{mut} < P_{crit}$ leads to Helper cell dominance). We first investigate this parameter in the absence of mobility. For $P_{mut} \geq P_{crit}$, the latency period decreases and can be described in three dimensions by an inverse power law i.e.

$$t_{mc} = \left(\frac{P_\Delta}{3.58}\right)^{-0.73}$$

A similar law also exists for 2-D simulation, with the corresponding magnitude of the exponent higher (=-0.88)

It is worth pointing out that small increases in P_Δ lead to dramatic decreases in the latency period. An increase of just 0.007 in P_Δ, from P_Δ =0, gives a corresponding 50% decrease in latency time. This magnitude of a decrease in real-time may result in the transition from one progression category to a progression category of shorter timespan . The range of P_{mut} corresponding to the latency period is 0.0425 (compared with 0.065 in 2-D) Three-dimensional results have consistently smaller latency times than those found in 2-D. This may be due to additional nearest neighbours in 3-D causing the dominant pattern of intra-site interactions to spread more quickly. For $P_\Delta \geq 0.0425$, ($P_\Delta \geq 0.065$ in the 2-D case), there is no latency period with the virus dominating throughout the simulation. This could represent extreme rapid progressors or a mutation probability that is too large

to represent that of HIV's accurately. For $P_{mut} < P_{crit}$ the latency time is infinite i.e. the immune system retains control over HIV. These two extremes of latency period, may not be representative of the disease progression. For this reason we focus on the region between these extremes and hypothesise that it contains timesteps representative of latency times corresponding to typical and rapid progressors and longterm non-progressors. This would imply that the mutation probabilities leading to the different rates of progression of disease are very specific with corresponding small ranges. The range of P_Δ is more sensitive at low values of P_Δ, e.g P_Δ =0.005, corresponding to longer latency times than at values of P_Δ, e.g P_Δ=0.04, corresponding to shorter latency times.

Exploring latency is not as straightforward when mobility is introduced, as the transistion from an immuno-competent system to an immuno-deficent system is not a clear one. The transistion is "fuzzy" with oscillations between viral and immune dominance eventually leading to complete viral dominance above P_{crit}. Given these oscillations it is impossible to define a timestep for when the viral population resurges. This "fuzzy" transistion exists because mobility increases the interactions between cells and also increases the stochasticity of the system, thereby decreasing its reliance on the core deterministic interaction sets. This "fuzzy" transition is also responsible for the almost continous phase transistion for mobile cells evident in Figure 2 compared with the very obvious transition for immobile cells. This would suggest that our treatment of mobility maybe somewhat unsophisicated and that future studies might usefully review the mechanism used.

In summary, the extension of our previous 2-D model to 3-D, has little effect on overall cell-growth patterns or the value of P_{crit} for both mobile and immobile cells. This could be because our extension to 3-D, which includes just two extra nearest neighbours is too simplistic and that a true extension should be more sophisticated, possibly including next-nearest neighbours and variable mobility. The latency period, predicted by our simulations, was investigated and followed a power law for both 2-D and 3-D, with a decrease in the magnitude of the exponent from 0.88 in 2-dimensions to 0.73 in 3-D. It is clear that viable latency times correspond to a small P_{mut} range only and are significantly affected by small increases in the mutation probability. The pattern of the immune system's fight for control over the virus was explored for extreme mobility and corresponds well to clinical findings, i.e. initial advantage for the virus, followed by a decrease in its population due to the mounting immune response. The amplitude of the peak of infection, as measured by difference between viral and helper densities, and the time taken for the immune response are functions of P_{mut}.

Acknowledgements: Hospitality received at DCU is warmly acknowledged by RBP.

References:

PERELSON,A. S. & WEISBUCH, G. (1997). Immunology for Physicists. *Rev. Mod. Phys.* **69(4)**, 1219-1267 .

DAYAN, I., STAUFFER, D. & HAVLIN, S. (1988). Cellular Automata Generalization of the Weisbuch-Atlan Model for Immune Response *J. Phys. A* **21**, 2473.

DE OLIVEIRA, S., DE OLIVEIRA, P.M.C. & STAUFFER, D. (1999) "Evolution, Money, War and Computers", Teubner (Stuggart and Leipzig) Chapter 3.

PANDEY,R.B. (1996) . A Cellular Automata Approach to Modelling the Immune Response: HIV, in "Scientific Computing in Europe, SCE 1996",Sept. 2-4, 1996, Dublin, Ireland, edited by H.J. Ruskin, R. O'Connor, and Y. Feng.

DOS SANSOS, R.Z.(1999) Immune Responses: Getting close to experimental results with cellular automata models. *Ann. Rev. Comp. Phys. vol. VI*, edited by D. Stauffer,World Scientific Singapore, p. 159.

CHOWDHURY, D., SAHIMI, M. & STAUFFER, D. (1991) A discrete model for immune surveillance, tumor immunity, and cancer *J. Theor. Bio.* **152**, 263.

STAUFFER, D. & PANDEY, R.B.(1992). Immunologically motivated simulation of cellular automata. *Computers in Physics* **6** , 404 .

PANDEY, R.B. (1991) Cellular automata approach to interacting cellular network models for the dynamics of cell population in an early HIV infection. *Physica A* **179** , 442 (1991).

CASTIGLIONE, F., BERNASCHI, M. & SUCCI, S. (1997). Simulating the Immune Response on a Distributed Parallel Computer. *Int. J. Mod. Phys. C* **8**, 527 (1997).

AHMED, E. (1996). Fuzzy Cellular Automata Models in Immunology.*J. Stat. Phys* **85**, 291.

KANEKO, K., (1997) . Coupled maps with growth and death: An approach to cell differentiation . *Physica D* **103**, 505 and references therein.

MIELKE, A. & PANDEY, R.B. (1998). A Computer Simulation Study of Cell Population in a Fuzzy Interaction Model for Mutating HIV. *Physica A* **251**, 430 (1998).

PANDEY, R.B.(1998) A Stochastic Cellular Automata Approach to Cellular Dynamics for HIV :Effect of Viral Mutation. *Theor. Biosci.* **117**, 32 (1998).

MANNION, R., RUSKIN, H.J. & PANDEY, R.B. (1999). Effect of Mutation on Helper T cells and Viral Population: A computer Simulation Model for HIV. *Theor. Biosci.* (accepted for publication) .

MANNION, R., RUSKIN , H.J., & PANDEY, R.B. (preprint). A Monte Carlo

Approach to Population Dynamics of Cells in an HIV Immune Response Model. *Theor. Biosci.*
FAUCI, A.S., PANTALEO, G., STANLEY, S. & WEISSMANN D. (1996). Immunopathogenic mechanisms of HIV infection. *Ann. Internal Medic.* **124**, 654 (1996).

Figure Captions:

Figure 1: Progression of $\rho_V - \rho_H$ with time, for various P_{mut} . 3-D filled shapes, 2-D

Figure 2: Variation of the difference in cell density ($\Delta\rho = \rho_V - \rho_H$) with mutation rate P_{mut}, for 3-D and 2-D.

Figure 3: Variation of latency period with $P_{mut} - P_{crit}$. Circles representing 2-D , squares representing 3-D , with $P_{mob} = 0$