Analysis and Dynamic Active Subspaces for a Long Term Model of HIV

by

Tyson S. Loudon

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Applied Mathematics & Statistics).

Golden, Colorado

Date _____

Signed: _____
Tyson S. Loudon

Signed: _____
Dr. Stephen Pankavich
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Willy Hereman
Professor and Head
Department of Applied Mathematics and Statistics

ABSTRACT

The Human Immunodeficiency Virus (HIV) disables many components of the body's immune system and, without antiretroviral treatment, leads to the onset of Acquired Immune Deficiency Syndrome (AIDS) and subsequently death. The infection progresses through three stages: initial or acute infection, an asymptomatic or latent period, and finally AIDS. Modeling the entire time course of HIV within the body can be difficult as many models have oversimplified its biological dynamics in the effort to gain mathematical insight but fail to capture the three stages of infection. Only one HIV model has been able to describe the entire time course of the infection, but this model is large and is expensive to simulate. In this paper, we'll show there are two viral free steady states and conduct a stability analysis of one of the steady states. Then, we'll present a reduced order model for the T-cell count 1700 days after initial infection using active subspace methods. Building on the previous results, we'll create a global in time approximation of the T-cell count at any time using dynamic active subspaces.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The human immune system is a complex network of cells, cell products, and cell-forming tissues that protect the body from harmful bacteria, viruses, and other harmful substances. A key component of the complex network of cells that protects the body is the T-cell. T-cells are a type of white blood cell that respond to harmful pathogens. One particular type of T-cell is the $CD4^+$ T-cell, or helper T-cell, which responds to harmful pathogens by directing other varieties of T-cells to destroy the pathogens. The $CD4^+$ T-cells are created in the bone marrow, and then undergo a maturation process in the thymus. These 'mature' helper T-cells are called immunocompetent T-cells and now have the ability to fight infections. Immunocompetent T-cells lie dormant until a pathogen presents itself on the surface of an antigen-preserving cell. Healthy individuals typically have 1000 healthy $CD4^+$ T-cells per cubic millimeter of blood. Because of the important role $CD4^+$ T-cells play in the body it is necessary for individuals to maintain a population density at or around this level.

The basic structure of a virus includes a nucleus which contains nucleic acids and a virus specific enzyme. The enzyme is coated with a layer of protein, called a capsid. In addition, there is an outer layer comprised of carbohydrates, lipids, or proteins. A virus, like a parasite, requires a host cell to reproduce. The virus attaches itself to a host cell and fuses itself to the host cell's membrane. However, a virus will not attach itself to any cell. Viruses have a method of recognition which is used so that the virus will only attach itself to an accommodating host cell. After attaching itself to the correct host cell, the envelope of the virus and the host cell will merge. At this time, the virus releases its contents into the host cell. These contents include viral nucleic acids and viral specific enzymes. If all of the necessary enzymes are present, then the virus replicates and new virions, which bud from the cell membrane, are released. Then, the new virions repeat the process.

The human immunodeficiency virus (HIV) recognizes the $CD4^+$ T-cells as an accommodating host cell. HIV is different than a standard virus, it is called a retrovirus. Retroviruses replicate within a host via a process called reverse transcription. Reverse transcription is a process by which an enzyme of the HIV virus, called reverse transcriptase, creates a complementary strand of DNA from RNA. The newly created HIV DNA is called a provirus. When the provirus is created, the $CD4^+$ T-cell ceases to have immunological function and the host cell continues to create HIV

virions. The process by which the HIV virus transcribes DNA from RNA is highly error prone. The errors in the transcription process create mutations, which allow the virus to continue to elude the immune system. In some cases, the HIV enters a $CD4^+$ T-cell but is unable to replicate for some time. The $CD4^+$ T-cell in this case is called a latently infected T-cell. Eventually, the latently infected T-cells become infected T-cells and the replication process begins.

The HIV infection is characterized by three distinct stages: the acute infection, the chronic infection, and the transition to Acquired Immune Deficiency Syndrome (AIDS). The first stage, the acute infection, takes place within about the first 10 weeks of being introduced to the virus and is characterized by rapid fluctuations in the T-cell and virus population. With respect to the T-cell population there is initially a rapid decrease, and then a rapid increase. Symptoms during this phase of the infection include fever, swollen glands, fatigue, rash, and sore throat. The next stage of the infection is the chronic infection, also known as the latency period. This phase lasts for seven to ten years. During the latency period the T-cell population and the virus population remain at relatively constant levels with the T-cell population decreasing at a slow rate. The third stage of the infection, the transition to AIDS occurs when the T-cell population reaches a density lower than 200 cells per cubic millimeter.

Many models have been proposed for the HIV infection. However, many of these models are overly simplistic and can only accurately capture the first stage of the infection. These simplified models show the T-cell count asymptotically approaching a nonzero limit in the second phase of the infection, which experimentally and biologically we know to not be the case. Only one model has been able to accurately capture all three stages of the HIV infection. This model, proposed by Hadjiandreou et al in [3], is a system of seven nonlinear autonomous coupled differential equations with twenty-seven parameters. In this paper we will analyze this model and use methods from sensitivity analysis to approximate solutions of the T-cell count as a function of time.

CHAPTER 2
HIV MODEL AND ANALYSIS

This paper is concerned with analysis of the following long term model examined and created in [3]:

$$\left.\begin{aligned}
\frac{dT}{dt} &= s_1 + \frac{p_1}{V+C_1}VT - \delta_1 T - (K_1 V + K_2 M_1)T \\
\frac{dT_I}{dt} &= \psi(K_1 V + K_2 M_I)T + \alpha_1 T_L - \delta_2 T_I - K_3 T_I CTL \\
\frac{dT_L}{dt} &= (1-\psi)(K_1 V + K_2 M_I)T - \alpha_1 T_L - \delta_3 T_L \\
\frac{dM}{dt} &= s_2 + K_4 VM - K_5 VM - \delta_4 M \\
\frac{dM_I}{dt} &= K_5 VM - \delta_5 M_I - K_6 M_I CTL \\
\frac{dCTL}{dt} &= s_3 + (K_7 T_I + K_8 M_I)CTL - \delta_6 CTL \\
\frac{dV}{dt} &= K_9 T_I + K_{10} M_I - K_{11} VT - (K_{12} + K_{13})VM - \delta_7 V
\end{aligned}\right\} \tag{2.1}$$

In this long term model, a variety of cells in the immune system are considered; however, $CD4^+$ *T-cells* rank as one of the most critical components in determining the body's response to HIV infection. In (2.1) the T-cell (T) population is increased by a standard source, s_1, which produces such cells at a constant rate, and a nonlinear generation term $\frac{p_1}{V+C_1}VT$. Hence, we assume that the body will provide a steady rate of T-cell production; accordingly, its associated rate is constant.

In contrast, a natural death term $\delta_1 T$ removes T-cells at a proportion depending on the T-cells count and the average T-cell life span, providing negative feedback to growth; lastly an "infected" term, $(K_1 V + K_2 M_1)T$, represents the T-cell's infection by a HIV *virion* or *infected macrophage*, dependent upon the infectious particle's infection rate. Then, some proportion ψ of these newly infected T-cells are converted into active *infected T-cells*, while the remainder are sent into temporary dormancy as *latent T-cells*.

The infected T-cells (T_I) receive, in addition to the number created by the virions and macrophages, a supply of "activated" latent T-cells, at a rate $\alpha_1 T_L$. A natural death, as in the T-cell case, is provided by $\delta_2 T_1$, while a more interesting term, $K_3 T_1 CTL$, represents the extermination of infected

T-cells by *cytotoxic lymphocytes*, one of the many attacker cells the immune system employs.

The proportion $(1-\psi)$ of latent T-cells created by the virions and macrophages remain as sleeper cells for some time; however at a rate $\alpha_1 T_L$, these dormant cells activate into the actively hostile infected T-cells. Not surprisingly, latent T-cells suffer from a natural death rate $\delta_3 T_L$.

Macrophages (M) possess a natural birth rate, s_2, as well as a natural death rate, δ_4. In addition, they are created in response to HIV infection, with rate $K_4 V M$. Once these macrophages are created, they attempt to eliminate the virions. The virions fight back by infecting the macrophages, transforming them into infected macrophages, which then serve to infect T-cells.

These infected macrophages (M_I) die at a certain rate δ_5 due to natural death, and are also hunted by cytotoxic lymphocytes at a rate K_6. Once infected, these macrophages assist the infected T-cells in producing virions, which infect more T-cells and macrophages, which produce more virions. This viscous circle provides positive feedback for the infection, allowing for massive amounts of virions to be produced and flow rampant throughout the body. In addition, these infected macrophages don't attack the virions, their uninfected counterparts do.

The main defender of the body against infected cells is the cytotoxic lymphocyte (CTL), which seeks to destroy the renegade body cells that HIV has infected and altered, namely T_I and M_I (these attacks are carried out alongside the uninfected macrophages). These CTL, also known as *killer T-cells*, are lymphocytes like the helper T-cells, T; similarly, these CTL are produced at a constant rate s_3 by the bone marrow. Furtively, the original model contains an "adjustment factor" $K_7 T_I + K_8 M_I$, which is "fitted" to make the model fit clinical data.

Lastly, and potentially most critically, the growth of virions V depends on a variety of parameters. The viruses are continually produced by the infected T-cells and infected macrophages at rates K_9 and K_{10}, respectively. In addition, the virions are lost at some rates K_{12} and K_{13}, proportional to the infection of T-cells and macrophages by virions, respectively. Also, macrophages, in accordance with their bodily function, kill the virions at a rate K_{13}. Finally, the virion particles die at some natural death rate δ_7.

Note that all parameter values in (2.1) are positive. Typical values and ranges for the parameters taken from [3] can be seen below in Table 1.

Table 1: Parameter values and ranges

Number	Notation	Value	Range	Value taken from:	Units
1	s_1	10	5 - 36	[4]	$mm^{-3}d^{-1}$
2	s_2	0.15	0.03 - 0.15	[4]	$mm^{-3}d^{-1}$
3	s_3	5	-	Fitted	$mm^{-3}d^{-1}$
4	p_1	0.2	0.01 - 0.5	Fitted	d^{-1}
5	C_1	55.6	1 - 188	Fitted	mm^{-3}
6	K_1	3.87 x 10^{-3}	10^{-8} - 10^{-2}	Fitted	mm^3d^{-1}
7	K_2	10^{-6}	10^{-6}	[4]	mm^3d^{-1}
8	K_3	4.5 x 10^{-4}	10^{-4} - 1	Fitted	mm^3d^{-1}
9	K_4	7.45 x 10^{-4}	-	Fitted	mm^3d^{-1}
10	K_5	5.22 x 10^{-4}	4.7 x 10^{-9} - 10^{-3}	Fitted	mm^3d^{-1}
11	K_6	3 x 10^{-6}	-	Fitted	mm^3d^{-1}
12	K_7	3.3 x 10^{-4}	10^{-6} - 10^{-3}	Fitted	mm^3d^{-1}
13	K_8	6 x 10^{-9}	-	Fitted	mm^3d^{-1}
14	K_9	0.537	0.24 - 500	Fitted	d^{-1}
15	K_{10}	0.285	0.005 - 300	Fitted	d^{-1}
16	K_{11}	7.79 x 10^{-6}	-	Fitted	mm^3d^{-1}
17	K_{12}	10^{-6}	-	Fitted	mm^3d^{-1}
18	K_{13}	4 x 10^{-5}	-	Fitted	mm^3d^{-1}
19	δ_1	0.01	0.01 - 0.02	Fitted	d^{-1}
20	δ_2	0.28	0.24 - 0.7	Fitted	d^{-1}
21	δ_3	0.05	0.02 - 0.069	Fitted	d^{-1}
22	δ_4	0.005	0.005	[4]	d^{-1}
23	δ_5	0.005	0.005	[4]	d^{-1}
24	δ_6	0.015	0.015 - 0.05	[6]	d^{-1}
25	δ_7	2.39	2.39 - 13	[4]	d^{-1}
26	α_1	3 x 10^{-4}	-	Fitted	d^{-1}
27	ψ	0.97	0.93 - 0.98	Fitted	-

2.1 Mathematical Analysis of (2.1)

Though the system (2.1) possesses a large number of steady states, the authors discovered at least ten using standard parameter values and a computational root finder, one is often most interested in the disease-free equilibrium. In this section, we investigate the stability properties of the disease-free equilibrium state.

Our first result demonstrates that only one such equilibrium state exists when all populations of (2.1) are positive.

Theorem 2.1. *The only biological relevant steady state of (2.1) satisfying* $V \equiv 0$ *is*

$$E_{NI} := \left(\frac{s_1}{\delta_1}, 0, 0, \frac{s_2}{\delta_4}, 0, \frac{s_3}{\delta_6}, 0\right)$$

Hence, the only guarantee of viral clearance as $t \to \infty$ occurs when actively and latently infected populations are also eradicated, resulting in healthy cell and macrophage populations tending asymptotically to background values.

If we do not assume that all of the populations in (2.1) are positive, then there exists another virus free steady state.

Theorem 2.2. *A steady state of the system* (2.1) *satisfying* $V \equiv 0$ *is given by*

$$E := \left(\frac{s_1}{\delta_1 - \omega K_2 K_9}, \omega K_{10}, \frac{\omega K_{10}\xi}{K_6(\alpha_1 + \delta_3\psi)}, \frac{s_2}{\delta_4}, -\omega K_9, -\frac{\delta_5}{K_6}, 0\right)$$

where

$$\omega = \frac{s_3 K_6 + \delta_5\delta_6}{\delta_5(K_7 K_{10} - K_8 K_9)} \quad \textit{and} \quad \xi = (1-\psi)(\delta_2 K_6 - \delta_5 K_3)$$

The proofs of Theorem 2.1 and 2.2 are contained in Appendix A. Plugging in the standard parameter values from Table 1 we get the following values for the populations:

$$T = 1010.39\,\mathrm{mm}^{-3}$$

$$T_I = 54.57\,\mathrm{mm}^{-3}$$

$$T_L = -15.77\,\mathrm{mm}^{-3}$$

$$M = 30\,\mathrm{mm}^{-3}$$

$$M_I = -102.83\,\mathrm{mm}^{-3}$$

$$CTL = -1666.67\,\mathrm{mm}^{-3}$$

$$V = 0\,\mathrm{mm}^{-3}$$

The proof of Theorems 2.1 and 2.2 guarantee that E_{NI} and E are the only two viral free steady states. Also, assuming the parameter values in (2.1) are all positive, the steady state E given in Theorem 2.2 is guaranteed to have a negative cytotoxic T-lymphocyte population. So, under no parameter regime will the steady state E be biologically relevant. Additionally, it would not make sense to set $\delta_5 = 0$ (even though this would take care of the problem) since this would imply that the infected macrophages do not have a natural death rate, and can only be eliminated by the cytotoxic T-lymphocytes.

2.2 Stability Analysis

Next, we provide necessary and sufficient conditions which guarantee the local asymptotic stability of the disease-free equilibrium E_{NI}.

Theorem 2.3. *The equilibrium state E_{NI} is locally asymptotically stable if and only if $R_0 \leq 1$, where*

$$R_0 = \max\{R_1, R_2, R_3\}$$

and

$$R_1 = \frac{K_1 K_9}{\delta_2 K_{11}}$$

$$R_2 = \frac{K_5 K_{10}}{(K_{12} + K_{13})\delta_5}$$

$$R_3 = \frac{K_2 K_5 K_9 s_1 s_2}{\delta_1\delta_2\delta_4\delta_5\delta_7 + \delta_4\delta_5 K_1 K_9 s_1 + \delta_1\delta_2 K_5 K_{10} s_2}.$$

The proof of Theorem 2.3 is also contained within Appendix A. Computing the basic reproduction number of Theorem 2.3 by using the standard parameter values given in Table 1, we find that $R_0 = 953 >> 1$. Hence, as expected, the non-infective steady state E_{NI} is not locally asymptotically stable.

Upon further inspection, Theorem 2.3 implies that slight perturbations in parameter values (perturbations which will obviously occur naturally, as the body is an inherently stochastic mechanism) may result in drastically different outcomes. In addition, the lack of asymptotic stability implies that there is no "disk of convergence" for each parameter value; unless the body's inner workings reduce to parameter values which precisely mimic the values in [3], there is no guarantee that the virus' prominence will follow the predicted path of [3].

Note that in Theorem 2.3, the local asymptotic stability of the virus free steady state E_{NI} depends only on 15 of the 27 parameters in (2.1). This indicates that parameter reduction may be possible. In Chapters 3 and 4 we will see a method by which to accomplish parameter reduction.

CHAPTER 3
ACTIVE SUBSPACE MODELING

In this chapter we will use active subspaces in order to approximate the T-cell count at a specific time given the parameter values in (2.1).

3.1 Defining An Active Subspace

The theory behind active subspaces begins with the matrix C defined as

$$C = \int (\nabla_x f)(\nabla_x f)^T \rho \, d\mathbf{x} \tag{3.1}$$

where f is the quantity of interest in a given computational model, the gradients of f are taken with respect to the model parameters, and ρ is a probability density. The matrix C is the average of the outer product of the gradient of f with itself and has some useful properties that will allow us to deduce information about f.

Looking at the entries of the matrix C

$$C_{ij} = \int \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} \rho \, d\mathbf{x}$$

we can see that it is a symmetric matrix. Since it is a symmetric matrix, it permits the eigendecompostion

$$C = W \Lambda W^T, \quad \text{where} \quad \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_1 \geq \ldots \geq \lambda_m \geq 0 \tag{3.2}$$

and W is an orthogonal matrix whose columns are the orthonormal eigenvectors $\mathbf{w}_i$, $i = 1, \ldots, m$.

From (3.2) we can solve for the eigenvalues of the C matrix. They are given by

$$\lambda_i = \int \left((\nabla_x f)^T \mathbf{w}_i \right)^2 \rho \, d\mathbf{x} \quad i = 1, \ldots, m. \tag{3.3}$$

From (3.3) we can see that the eigenvalues of the C matrix are the mean squared directional

derivatives of f, in the direction of the corresponding eigenvector. Thus, the eigenvalues of C give us useful information about our quantity of interest because, for instance, if an eigenvalue is small then (3.3) tells us that, on average, the quantity of interest f does not change significantly in the direction of the corresponding eigenvector. Conversely, if the eigenvalue under consideration is large, then we know that f changes considerably in the direction of the corresponding eigenvector and therefore, we need to investigate what happens in that direction.

Then, once we have the eigendecomposition (3.2) we can separate the eigenvalues and eigenvectors in the following way

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \quad W = \begin{bmatrix} W_1 & W_2 \end{bmatrix}. \tag{3.4}$$

In (3.4), Λ_1 contains the 'large' eigenvalues, Λ_2 contains the 'small' eigenvalues, W_1 contains the eigenvectors associated with the 'large' eigenvalues, and W_2 contains the eigenvectors associated with the 'small' eigenvalues.

An easy way to differentiate between the 'large' and 'small' eigenvalues is to plot the eigenvalues on a log plot from greatest to least and look for gaps. Gaps in the plot will correspond to differences of an order of magnitude. Put all of the eigenvalues before the gap into Λ_1 and the rest in Λ_2. A more systematic method of choosing how many eigenvalues to put in Λ_1 will be given in section 3.3.

With the decomposition (3.4), we can represent any element $\mathbf{x}$ in the parameter space in the following way

$$\mathbf{x} = \underbrace{WW^T}_{\mathrm{I}}\mathbf{x} = W_1 \underbrace{W_1^T\mathbf{x}}_{\mathbf{y}} + W_2 \underbrace{W_2^T\mathbf{x}}_{\mathbf{z}} = W_1\mathbf{y} + W_2\mathbf{z}. \tag{3.5}$$

Now, when we evaluate our quantity of interest at a specific value in the parameter space $\mathbf{x}$, this is the same as evaluating the quantity of interest at the point $W_1\mathbf{y} + W_2\mathbf{z}$, i.e.

$$f(\mathbf{x}) = f(W_1\mathbf{y} + W_2\mathbf{z}).$$

Because of the way we defined W_1 and W_2 we know that small perturbations in $\mathbf{y}$ will not change f much on average. But, small perturbations in $\mathbf{z}$ will, on average, change f significantly. For this reason we define the active subspace to be the range of W_1 and the inactive subspace to be the range of W_2.

3.2 Approximating C

In this section we will be approximating the eigenvalues and eigenvectors of the matrix C defined by (3.1) using a random sampling algorithm. The algorithm that we will be using is outlined in [1] (Algorithm 3.1) and [2] and is given as follows:

Algorithm 3.1

1. Draw N samples $\{\mathbf{x}_j\}$ independently according to the density function ρ.
2. For each parameter sample $\mathbf{x}_j$, compute $\nabla_x f_j = \nabla_x f(\mathbf{x}_j)$.
3. Approximate

$$C \approx \hat{C} = \frac{1}{N}\sum_{j=1}^{N}(\nabla_x f_j)(\nabla_x f_j)^T$$

4. Compute the eigendecompositions $\hat{C} = \hat{W}\hat{\Lambda}\hat{W}^T$.

The last step is equivalent to computing the singular value decomposition of the matrix

$$\frac{1}{\sqrt{N}}[\nabla_x f_1 \dots \nabla_x f_N] = \hat{W}\sqrt{\hat{\Lambda}}\hat{V}, \tag{3.6}$$

where it can be shown that the singular values are the square roots of the eigenvalues of $\hat{C}$ and the left singular vectors are the eigenvectors of $\hat{C}$. The singular value decomposition method of approximating $\hat{C}$ was developed by Russi in his PhD thesis [5].

3.3 Approximating the T-cell Population After 1700 Days

Now we will apply Algorithm 3.1 to the HIV model (2.1) with our quantity of interest being the T-cell count 1700 days after initial infection. We will choose each sample so that every element is uniformly distributed between -1 and 1, i.e. $\mathbf{x_j} \sim [-1, 1]^{27}$. The T-cell count population was chosen to be the quantity of interest because it is a good indicator of a patient's overall health and 1700 days was chosen because typically, regardless of parameter values, the patient's T-cell count will not be zero after 1700 days. However, if a time later than 1700 days is chosen, then the patient's

T-cell count might be zero before the final time is reached.

Since we do not have an explicit function for the T-cell count after 1700 days as a function of our parameters we can not explicitly compute the gradients required in step 3 of Algorithm 3.1. Instead we will approximate the gradients using a forward finite difference with a step size of 10^{-6}.

More specifically, for each of the random samples $\mathbf{x_j} \sim [-1,1]^{27}$ we use the linear mapping

$$\mathbf{x} = \frac{1}{2}\big(\text{diag}(\mathbf{x_u} - \mathbf{x_l})\mathbf{x_j} + (\mathbf{x_u} + \mathbf{x_l})\big),$$

where $\mathbf{x_u}$ and $\mathbf{x_l}$ are the upper and lower bounds of the parameters respectively, to map the normalized parameters into their biologically relevant range. Here $\mathbf{x}$ are the parameter values inputted into the simulation. Then, a stiff differential equations solver in MATLAB, ode23s, is used to calculate the T-cell count after 1700 days. Then, one by one we will perturb each of the 27 parameters by 10^{-6} and again calculate the T-cell count after 1700 days. With these two values we then use a forward finite difference to approximate the gradient of the T-cell count after 1700 days with respect to the model parameters.

For the above mapping, $\mathbf{x_u}$ and $\mathbf{x_l}$ are taken to be 2.5% above and below the typical values given in Table 1. The reason the ranges in Table 1 were not used is twofold. First, not all of the parameters are given ranges. Secondly, when the entire ranges were used the results of the simulation appeared to be very different than what is expected biologically. Highly oscillatory solutions were found for the T-cell count as a function of time. Because of this, we chose to limit the parameter ranges to 2.5% above and below the typical value. For parameter values within this range, all solutions of the T-cell count as a function of time were shown to exhibit all three stages of the HIV infection.

Figure 1 below shows the approximate eigenvalues of the matrix C using the T-cell count after 1700 days as the quantity of interest. From this figure we can see that there is a gap between the first and second eigenvalues.

In order to more accurately test the best decomposition of Λ we can use the follow measure of separation

$$\hat{\lambda}_k = \frac{\lambda_k - \lambda_{k+1}}{\lambda_1}, \quad k = 1, 2, \ldots, 26. \tag{3.7}$$

Then, the dimension of the active subspace, i.e. the number of eigenvalues put into Λ_1, will be

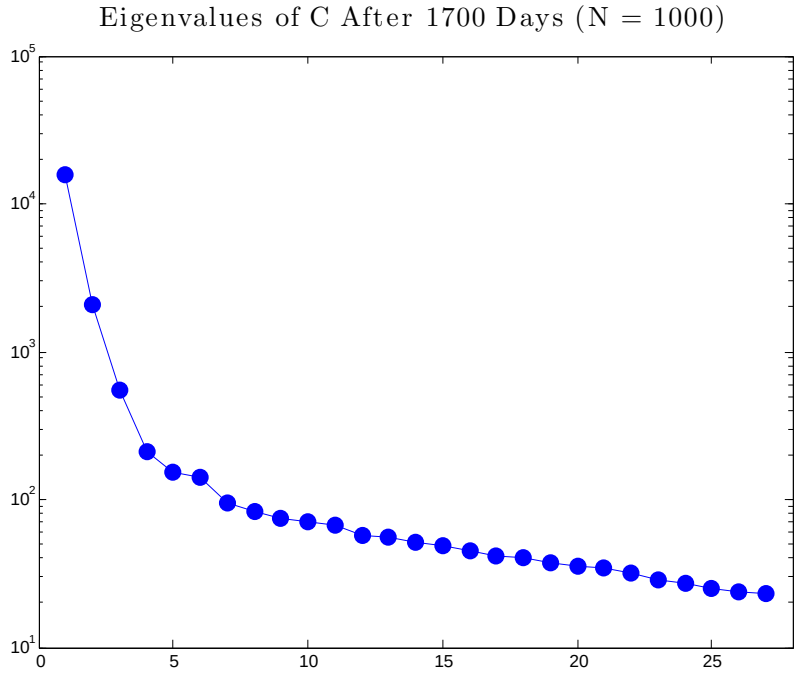


Figure 1: Approximation of eigenvalues of C using 1000 random samples.

given by

$$\text{dim} = \underset{\text{k}=1,\ldots,26}{\text{argmax}}\, \hat{\lambda}_{\text{k}} \tag{3.8}$$

While the index of the largest value of $\hat{\lambda}_k$ tells us where the largest gap in the eigenvalues of C is located, it is most convenient to only look at the first two values $\hat{\lambda}_1$ and $\hat{\lambda}_2$. By doing this, we limit the dimension of the active subspace to one and two respectively, which allows us to easily plot the quantity of interest as a function of the active subspace and allows us to fit a curve or surface to the data. Plotting the the values of $\hat{\lambda}_k$ results in Figure 2 below.

Clearly, with the measure of separation given by (3.7), the best choice for the dimension of the active subspace is one. Consequently, we put λ_1 in the matrix Λ_1 and the rest of the eigenvalues λ_i, $i = 2, \ldots, 27$, along the diagonal of the matrix Λ_2. From the definition of the active subspace given in section 3.1 the active subspace will be the span of the first eigenvector.

Figure 3 shows the eigenvector corresponding to the largest eigenvalue shown in Figure 1. We can see that there are three parameters with weights greater than 0.3. These are parameters 9, 22,

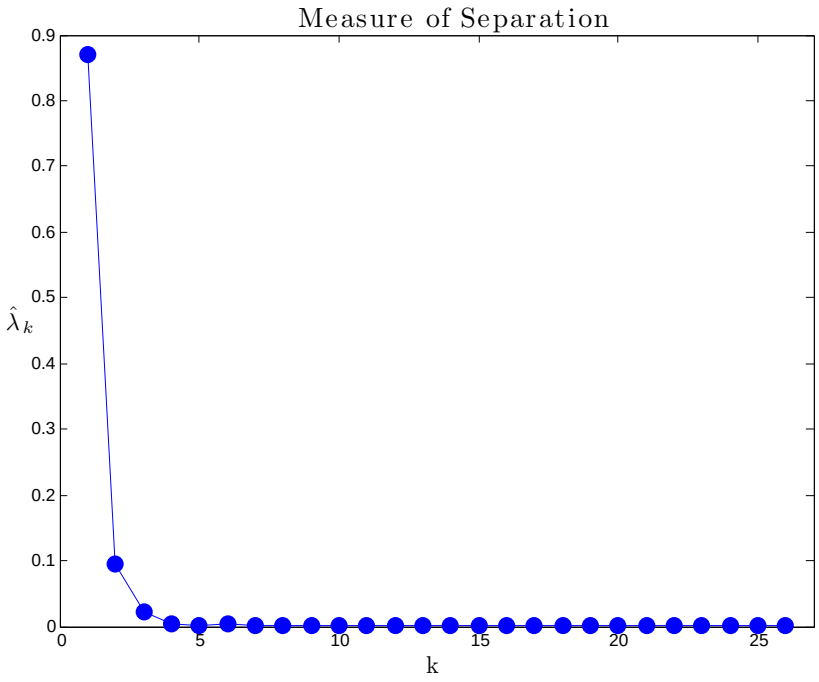


Figure 2: Measure of separation for eigenvalues of C.

and 25. From Table 1 we can see that these parameters are K_4, δ_4, and δ_7 which represent the increase in macrophage population due to the immune system, the death rate of the macrophage population, and the death rate of the virus population respectively. So, small perturbations in these parameters will change the T-cell count after 1700 days because they are the most heavily weighted. Whereas, changing the parameters whose weights are at or near zero will not change the T-cell count after 1700 days significantly.

Looking at the T-cell count along the active subspace results in Figure 4. We will call plots of the quantity of interest along the active subspace sufficient summary plots, the term used for these plots in [1]. From Figure 4 (left) we can see a trend, namely that as you increase the value of the active variable, the inner product of the weight vector with the normalized parameter values, then the T-cell count decreases.

So, in order to be able to approximate the T-cell count after 1700 days we will fit a four parameter arctangent function to the data. We did this by using the MATLAB command lsqcurvefit which minimizes the residual, in the least squares sense, of the difference in the data and the

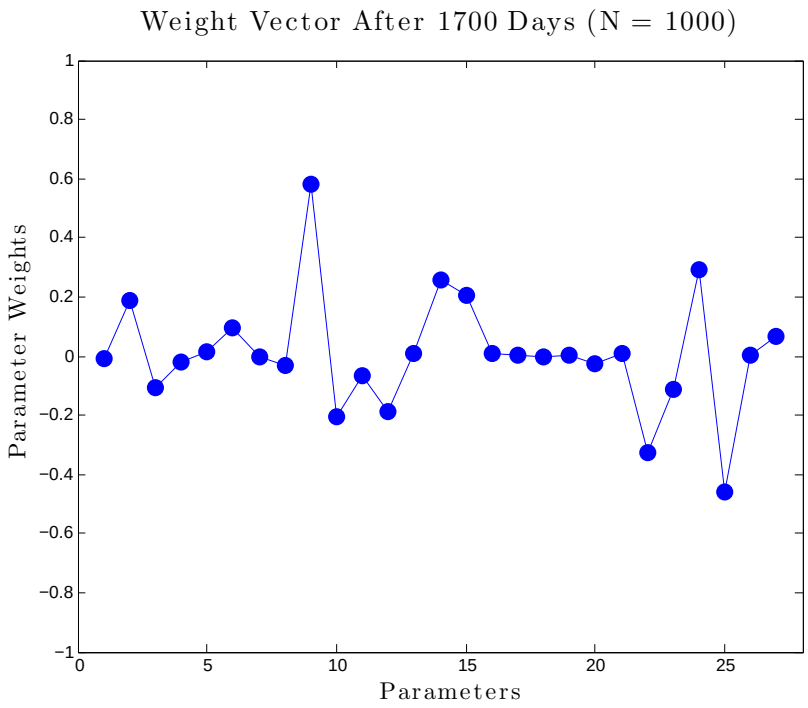


Figure 3: Approximation of the 1st eigenvector of C using 1000 random samples.

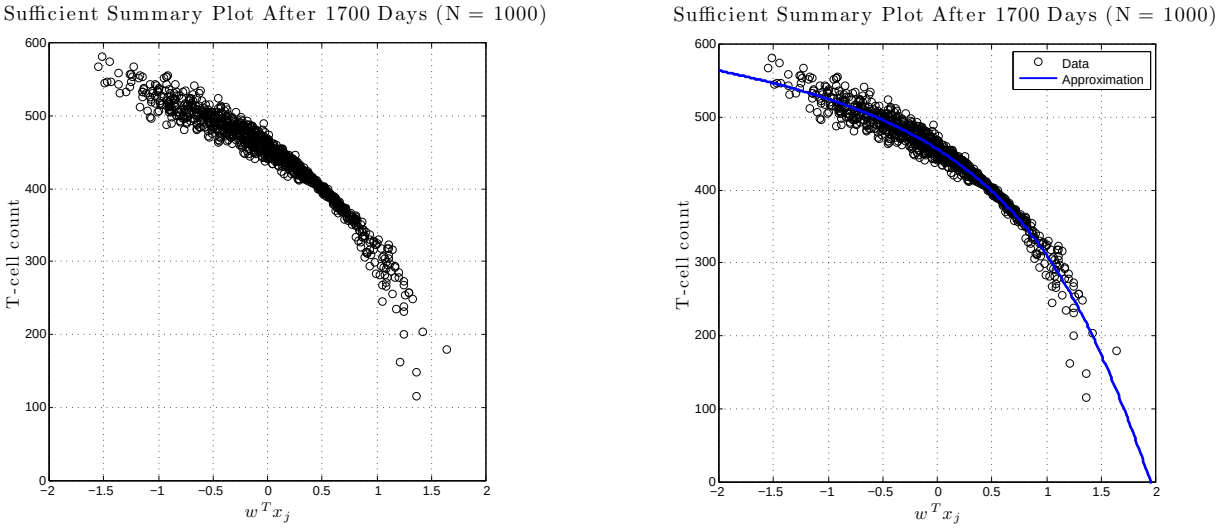


Figure 4: Sufficient summary plot after 1700 days (left). Approximation to the T-cell count after 1700 days (right).

approximation. Using this we find that the data is best fit by the function

$$T(x) = -79.2532 - 492.5680\tan^{-1}(0.8933\text{x} - 1.9069).$$

The result of plotting the above approximation of the T-cell count after 1700 days on top of the simulation data can be seen in Figure 4 (right).

In order to test the efficacy of the above approximation we ran 100 simulations and computed the relative error in the approximation. The results of this can be seen in Figure 5 below. Figure 5 shows that at most the approximation is slightly less than 10% off from the value given by solving (2.1) using a stiff ordinary differential equations solver. Furthermore, for all but seven of the simulations, the approximation was less than 5% off.

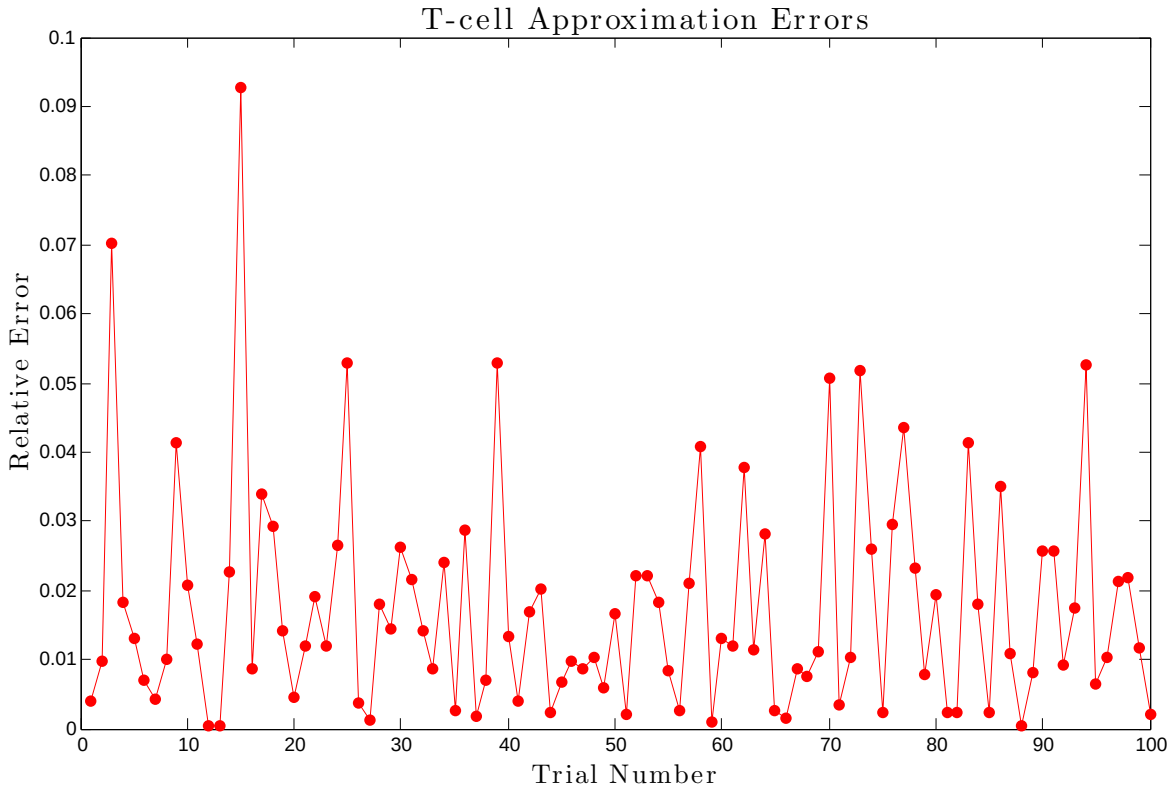


Figure 5: Relative errors in the approximation of the T-cell count after 1700 days.

CHAPTER 4

Dynamic Active Subspaces

Now that we have calculated the approximation for the T-cell count at 1700 days along the active subspace, we can repeat the methods used in the previous section for many different time values in order to create a global in time T-cell approximation. In this section we will go over methods for doing this.

4.1 Method

First, it is necessary to choose what time values to approximate the T-cell count at. We chose to break up the time from 0 to 3400 days into 55 non-uniform intervals. The time discretization can be found in Appendix B. After partitioning the time interval from 0 to 3400 days into non-uniform partitions we then computed the eigenvalues and eigenvectors at each time step.

Next, we need to orient the eigenvectors to be in approximately the same direction so that they transition smoothly from one time step to the next. By this we mean that from one time step to the next, the magnitude of the components of the consecutive weight vectors differ only slightly, but they have different signs. So, we multiply certain weight vectors by -1 so that they are all oriented the same. The active subspace method given in section 3.1 and 3.2 gives the active subspace up to a plus or minus sign. This occurs because the active subspace is based on the eigenvectors of the matrix C, and multiplying an eigenvector by a constant results in another eigenvector and multiplying an orthonormal vector by negative one preserves the orthonormality of the vector.

After orienting the weight vectors, we computed and fitted curves to the sufficient summary plots. For each sufficient summary plot we used one of three approximations. The three approximations used were linear approximations, arctangent approximations, or arctangent approximations multiplied by a heaviside step function. The latter approximation was used for later times when for a certain value of the active variable all T-cell count values to the right of the active variable were zero, and for the T-cell count values to the left of the active variable the fit resembled an arctangent function. For example, if we look at Figure 6 we can see that for values of the active variable greater than about 0.5 the T-cell count is zero, but for values of the active variable greater than 0.5 T-cell count trend resembles an arctangent function.

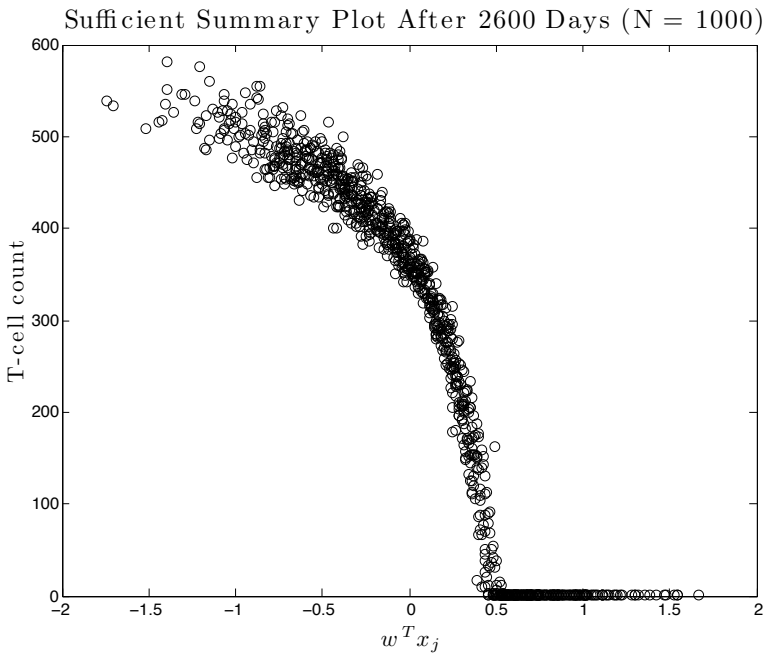


Figure 6: Sufficient summary plot after 2600 days using 1000 trials.

We used the arctangent function multiplied by a heaviside function approximation for time values greater than or equal to 1800 days. To compute the fits for these sufficient summary plots, we remove the data points corresponding to a zero T-cell value and fit a four parameter arctangent function to the leftover data points. Then, we approximate the zero of the resulting arctangent function in MATLAB using the fzero function. Next, we multiply the arctangent approximation by the heaviside function with the argument of the heaviside function being the zero of the arctangent approximation minus the active variable. The result of this process on the sufficient summary plot after 2600 days can be seen below in Figure 7.

The next step is to piece together the approximations using linear basis functions, i.e.

$$T(\mathbf{x}, t) \approx \sum_{i=1}^{55} T_{t_i}(\mathbf{x})\phi_{t_i}(t) \tag{4.1}$$

where $\mathbf{x}$ is the active variable, $T_{t_i}(\mathbf{x})$ is the approximation to the T-cell count in the i^{th} time

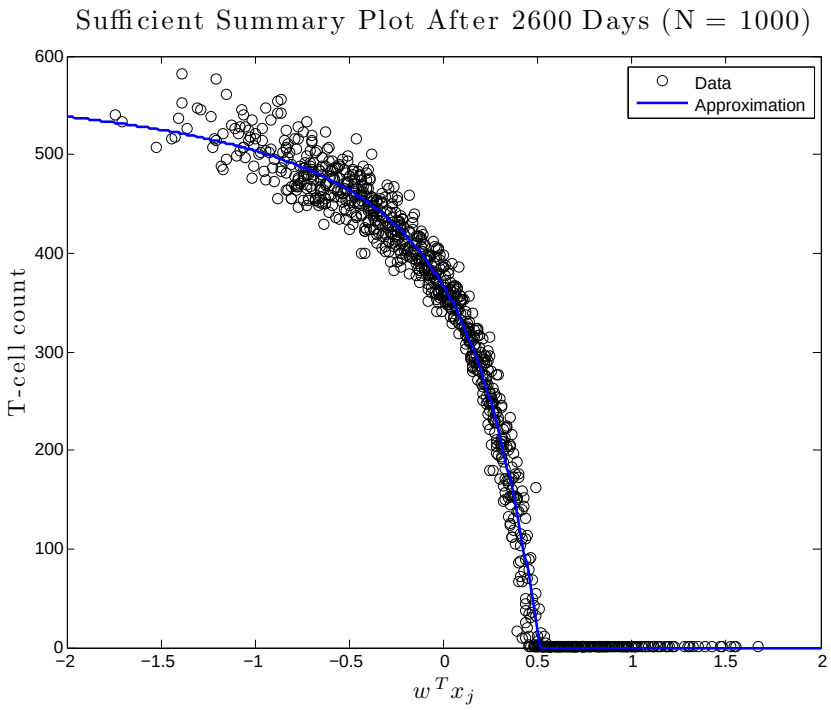


Figure 7: Sufficient summary plot after 2600 days with the approximation.

interval, and $\phi_{t_i}(t)$ is the linear basis function given by

$$\phi_{t_i}(t) = \begin{cases} \dfrac{t - t_{i-1}}{t_i - t_{i-1}} : & t_{i-1} \leq t \leq t_i \\ \dfrac{t_{i+1} - t}{t_{i+1} - t_i} : & t_i \leq t \leq t_{i+1} \\ 0 \quad : & t \notin [t_{i-1}, t_{i+1}] \end{cases}$$

4.2 An Example

To see an example of how dynamic active subspaces works, suppose we have the weight vectors and an approximation to the sufficient summary plot for T-cell counts at 60 and 65 days and we want to approximate what the T-cell count after 62 days will be.

The sufficient summary plots and weight vectors for the T-cell count after 60 and 65 days are shown below in Figures 8 and 9.

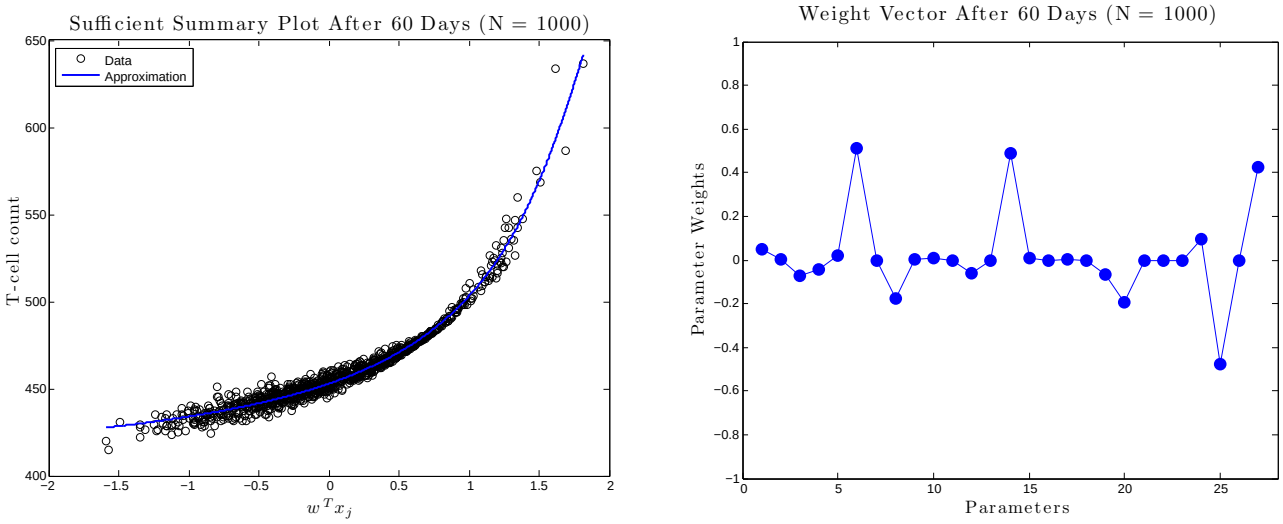


Figure 8: T-cell approximation after 60 days (left). First eigenvector of C with the quantity of interest being the T-cell count after 60 days (right).

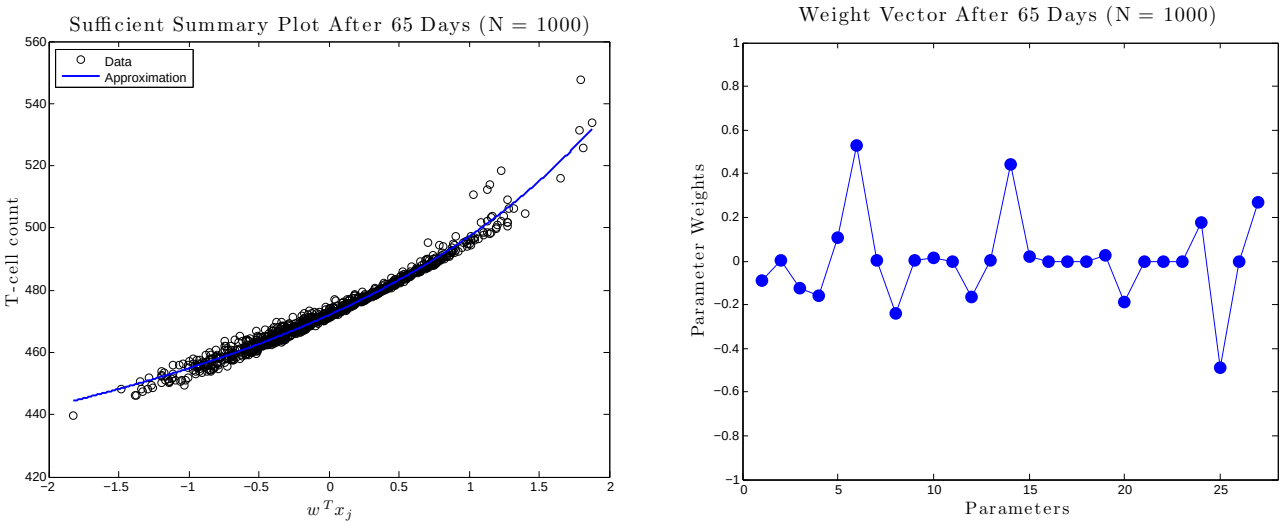


Figure 9: T-cell approximation after 65 days (left). First eigenvector of C with the quantity of interest being the T-cell count after 65 days (right).

The T-cell count for any time t between 60 and 65 is given by

$$T(\mathbf{x}, t) \approx T_{60}(\mathbf{x})\phi_{60}(t) + T_{65}(\mathbf{x})\phi_{65}(t), \quad 60 \leq t \leq 65.$$

Plugging in $t = 62$ into the above approximation gives the results shown below in Figure 10.

Figure 10 shows that the approximation is in good agreement with the data points.

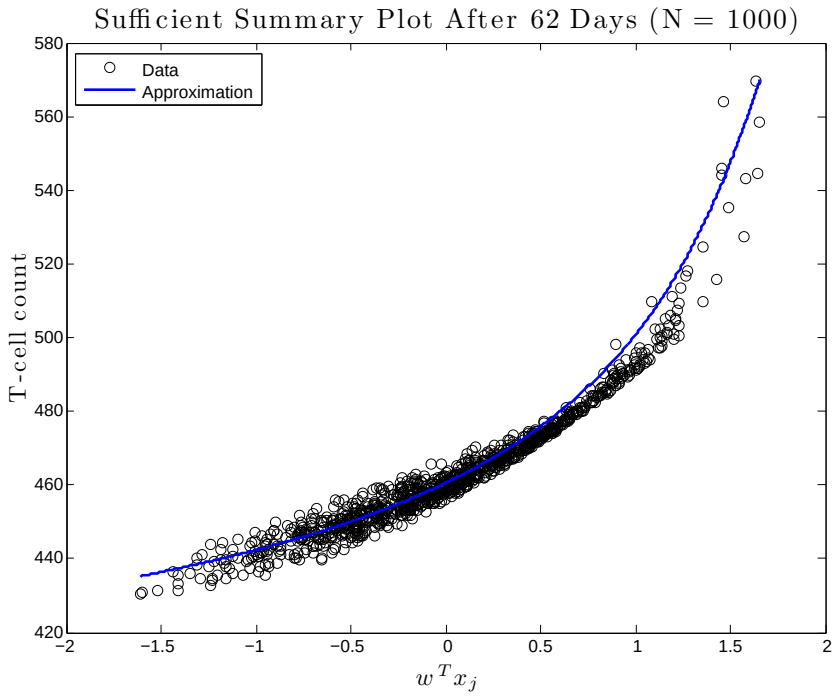


Figure 10: Sufficient summary plot after 62 days with the approximation.

Also, the approximation to the weight vector at 62 days is a linear interpolation between the weight vectors at 60 days and 65 days. Figure 11 below shows the absolute error of the approximation to the weight vector and the actual weight vector computed using algorithm 3.1.

4.3 Results

The results of computing the active subspace for 55 time steps can be seen below in Figure 12. From Figure 12 we can see that the trends in the sufficient summary plot transition smoothly from one time step to the next. Also, we can see the transitions back and forth from linear trends to arctangent trends. Lastly, for the times greater than or equal to 1800 days we can see the trends resemble arctangent function multiplied by heaviside step functions.

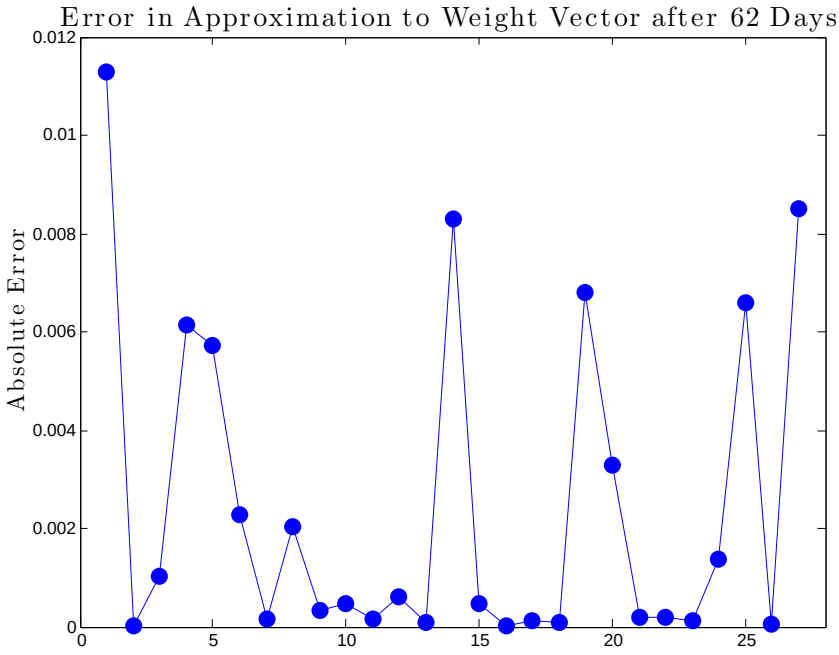
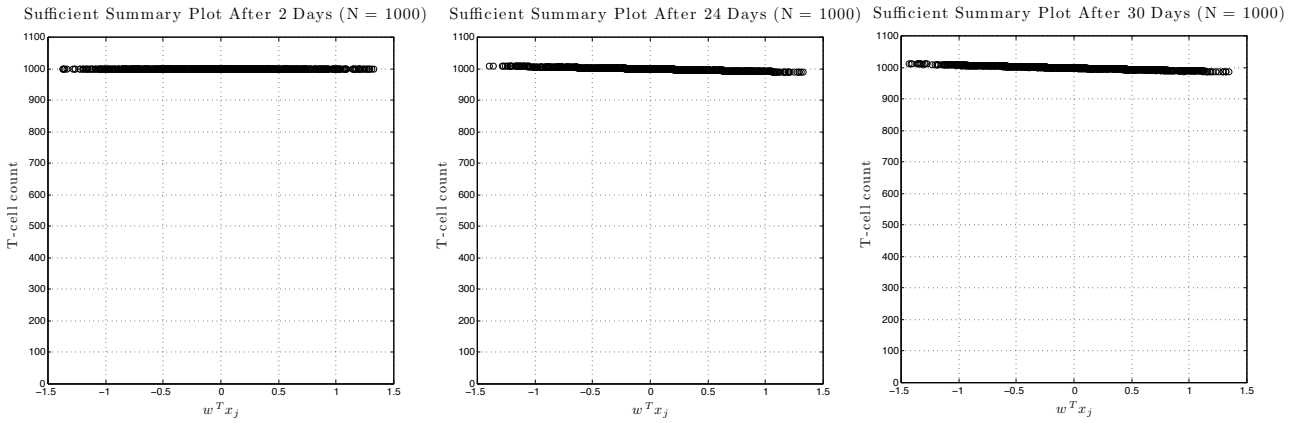
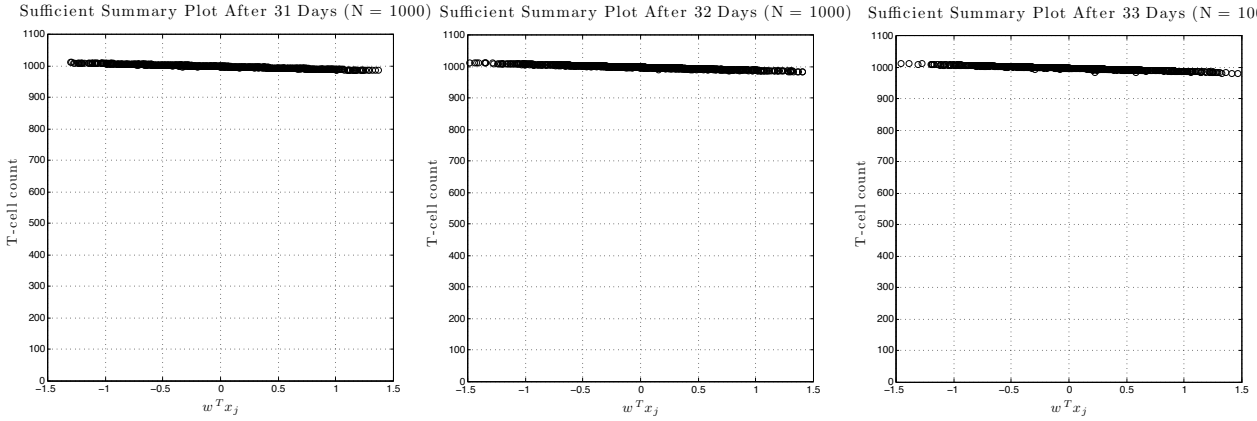
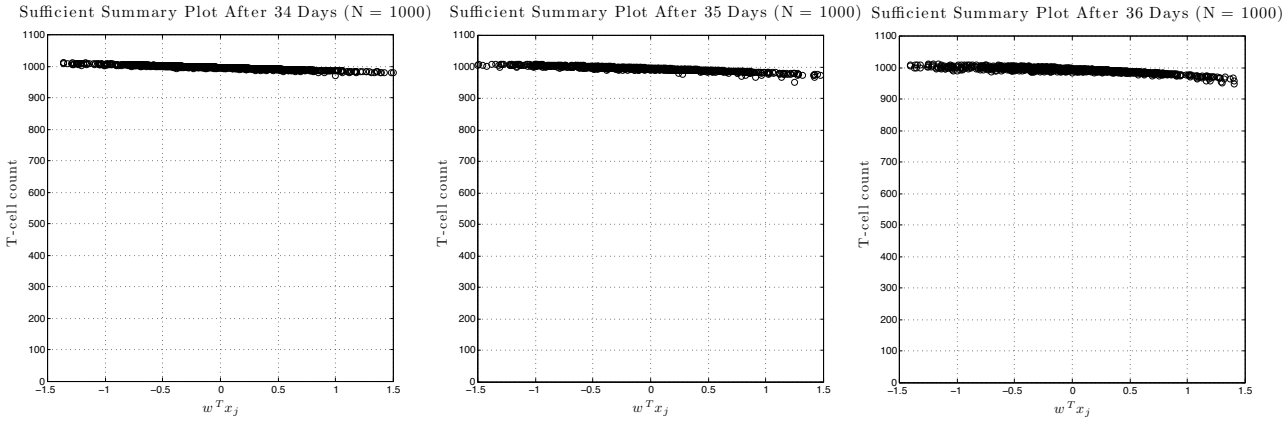
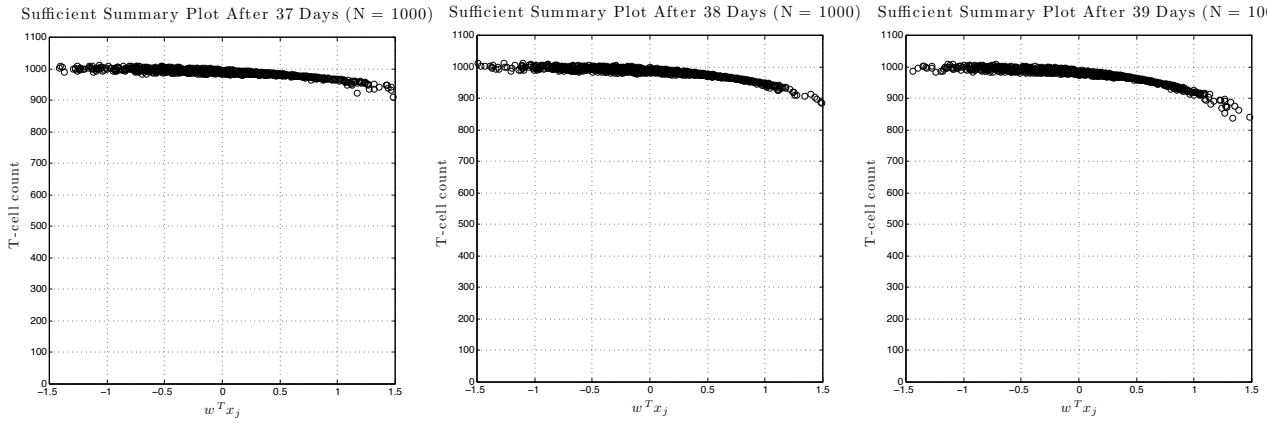


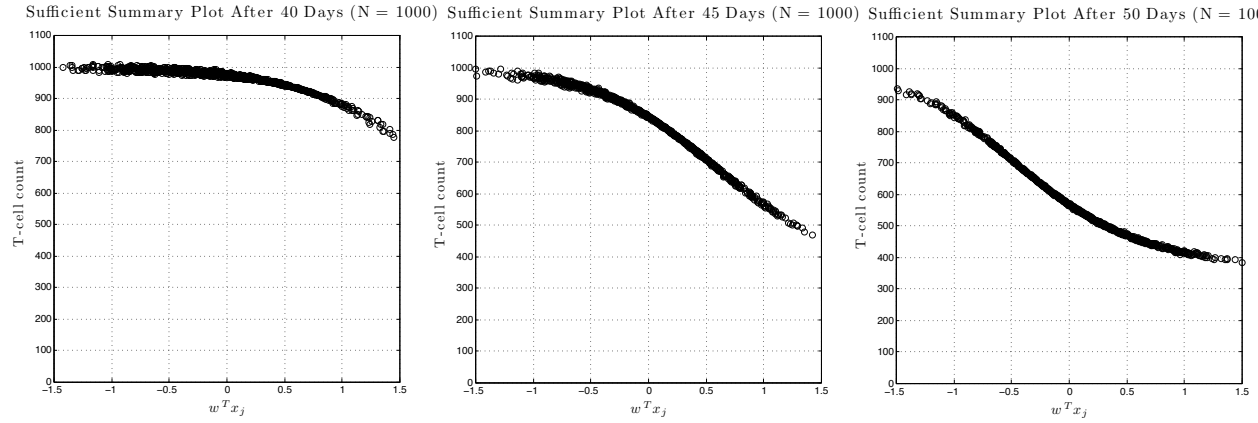
Figure 11: Absolute error in the approximation to the weight vector after 62 days.



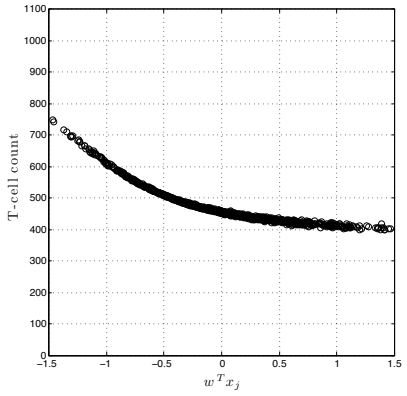

Sufficient Summary Plot After 31 Days (N = 1000)
Sufficient Summary Plot After 32 Days (N = 1000)
Sufficient Summary Plot After 33 Days (N = 1000)
T-cell count
$w^T x_j$


Sufficient Summary Plot After 34 Days (N = 1000)
Sufficient Summary Plot After 35 Days (N = 1000)
Sufficient Summary Plot After 36 Days (N = 1000)
T-cell count
$w^T x_j$

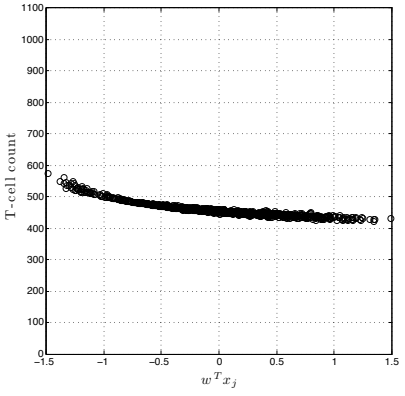

Sufficient Summary Plot After 37 Days (N = 1000)
Sufficient Summary Plot After 38 Days (N = 1000)
Sufficient Summary Plot After 39 Days (N = 1000)
T-cell count
$w^T x_j$


Sufficient Summary Plot After 40 Days (N = 1000)
Sufficient Summary Plot After 45 Days (N = 1000)
Sufficient Summary Plot After 50 Days (N = 1000)
T-cell count
$w^T x_j$

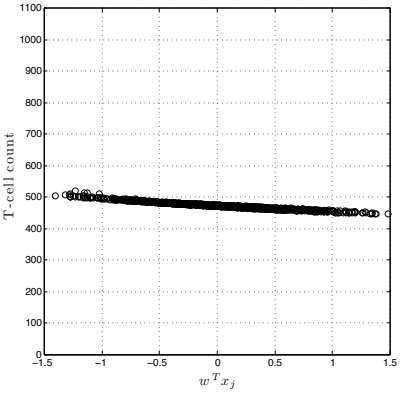
Sufficient Summary Plot After 55 Days (N = 1000)



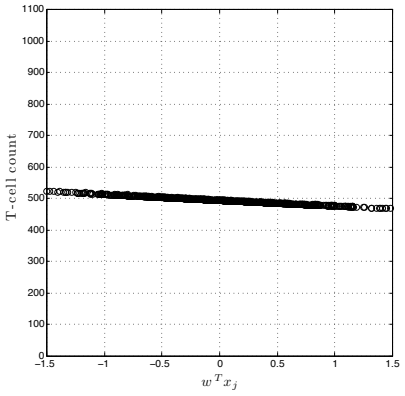
Sufficient Summary Plot After 60 Days (N = 1000)



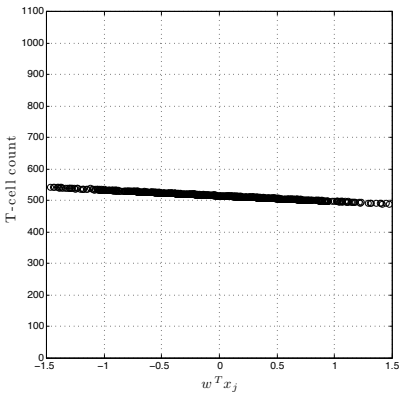
Sufficient Summary Plot After 65 Days (N = 1000)



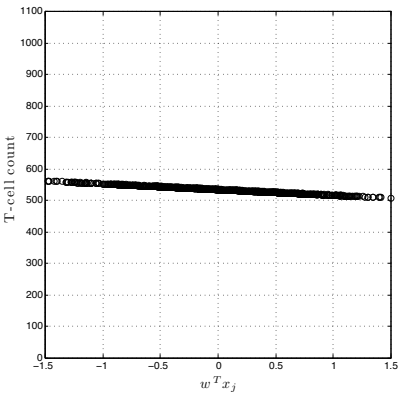
Sufficient Summary Plot After 70 Days (N = 1000)



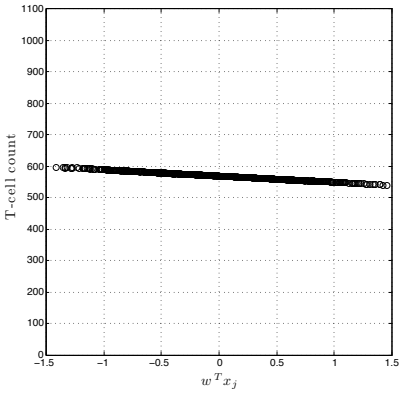
Sufficient Summary Plot After 75 Days (N = 1000)



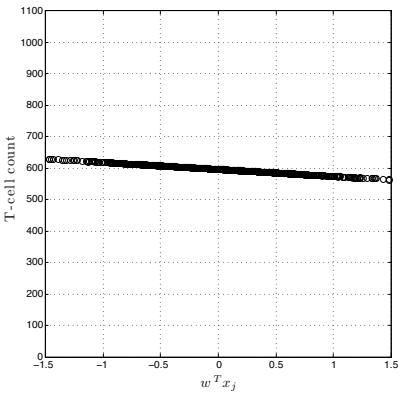
Sufficient Summary Plot After 80 Days (N = 1000)



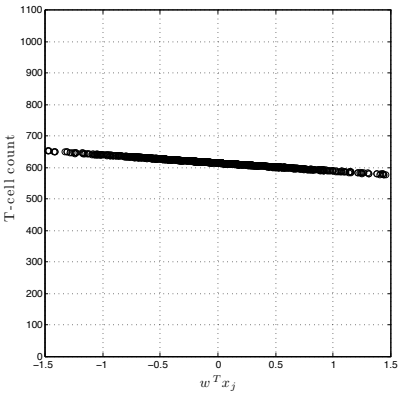
Sufficient Summary Plot After 90 Days (N = 1000)



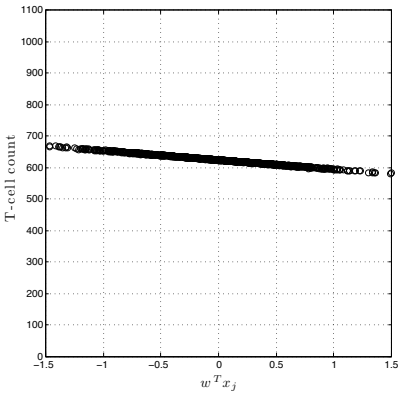
Sufficient Summary Plot After 100 Days (N = 1000)



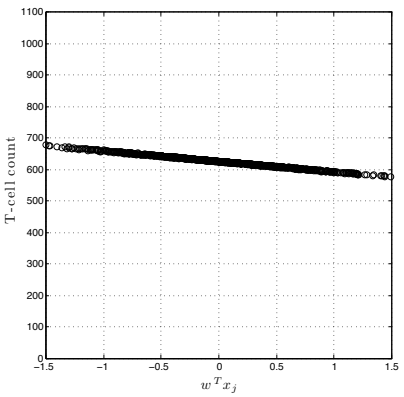
Sufficient Summary Plot After 110 Days (N = 1000)



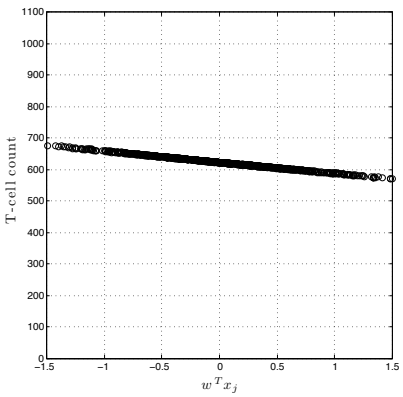
Sufficient Summary Plot After 120 Days (N = 1000)



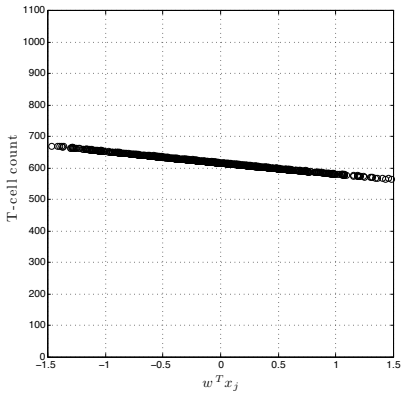
Sufficient Summary Plot After 130 Days (N = 1000)



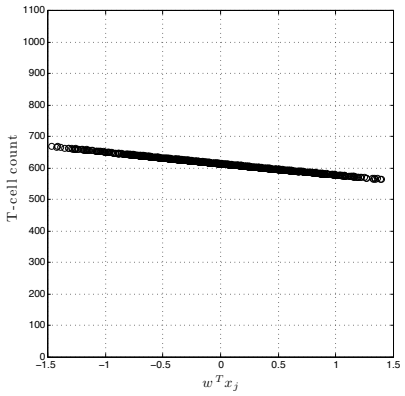
Sufficient Summary Plot After 140 Days (N = 1000)



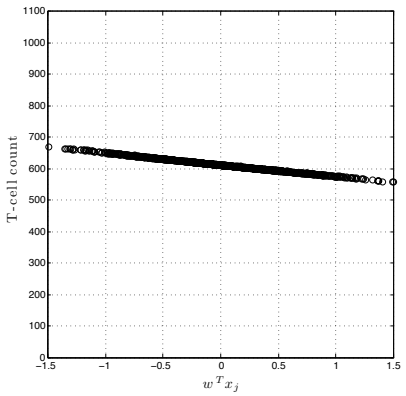
Sufficient Summary Plot After 160 Days (N = 1000)



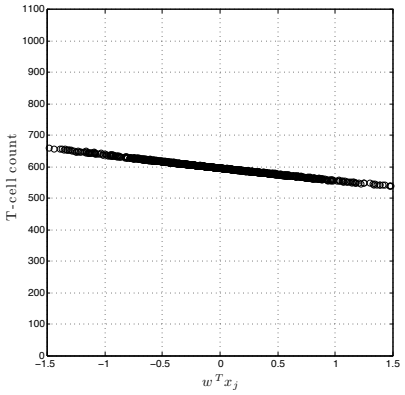
Sufficient Summary Plot After 180 Days (N = 1000)



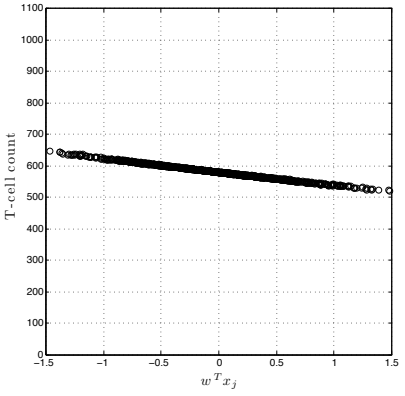
Sufficient Summary Plot After 200 Days (N = 1000)



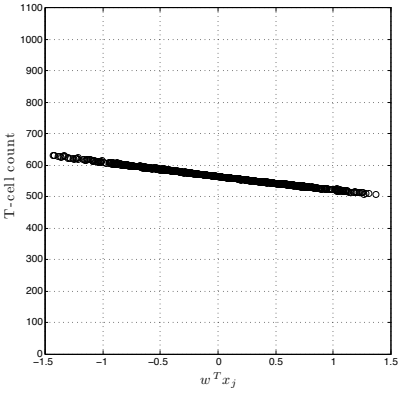
Sufficient Summary Plot After 300 Days (N = 1000)



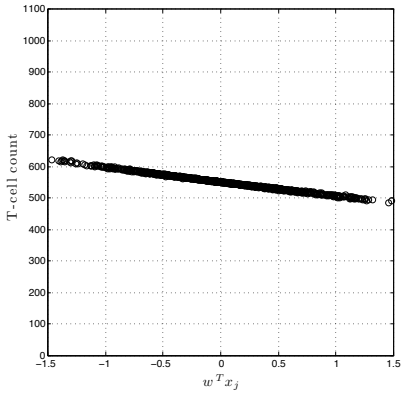
Sufficient Summary Plot After 400 Days (N = 1000)



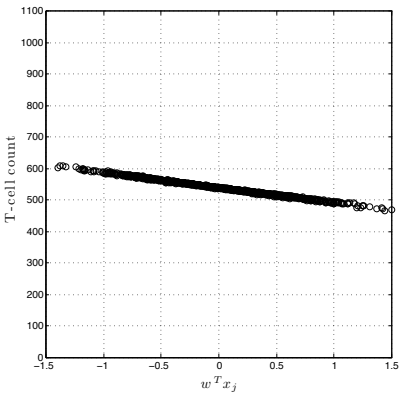
Sufficient Summary Plot After 500 Days (N = 1000)



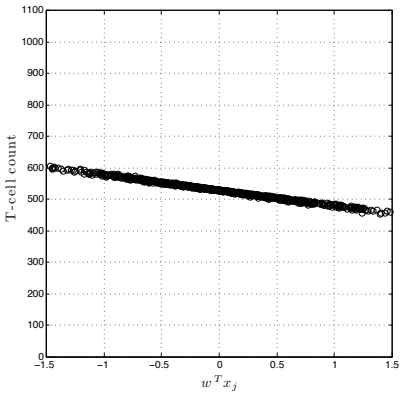
Sufficient Summary Plot After 600 Days (N = 1000)



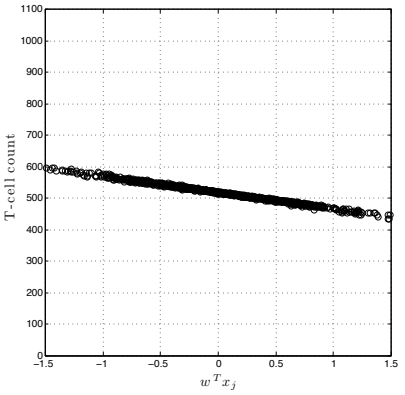
Sufficient Summary Plot After 700 Days (N = 1000)



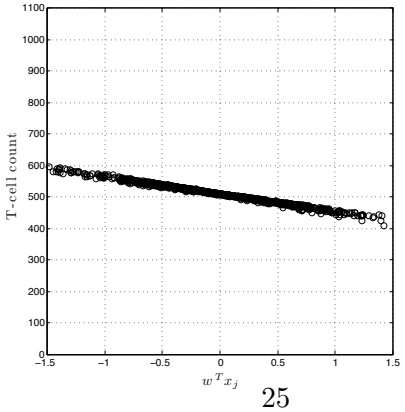
Sufficient Summary Plot After 800 Days (N = 1000)



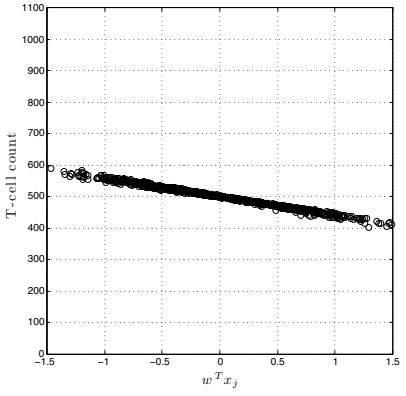
Sufficient Summary Plot After 900 Days (N = 1000)



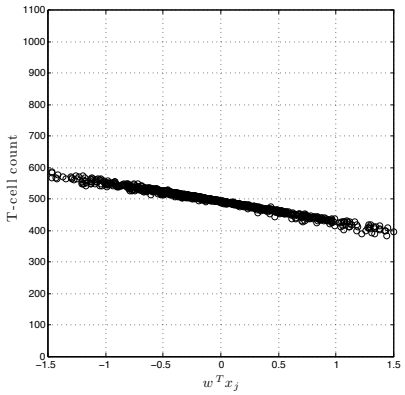
Sufficient Summary Plot After 1000 Days (N = 1000)



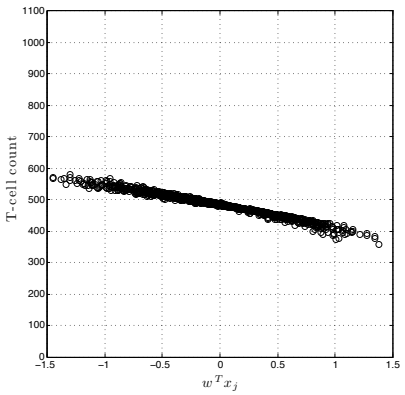
Sufficient Summary Plot After 1100 Days (N = 1000)



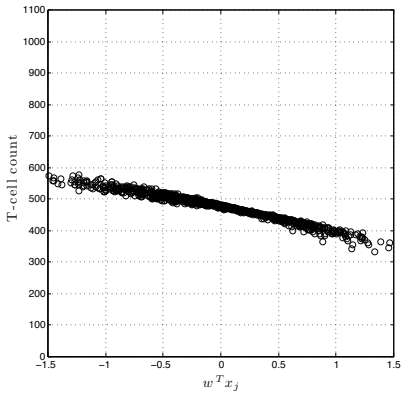
Sufficient Summary Plot After 1200 Days (N = 1000)



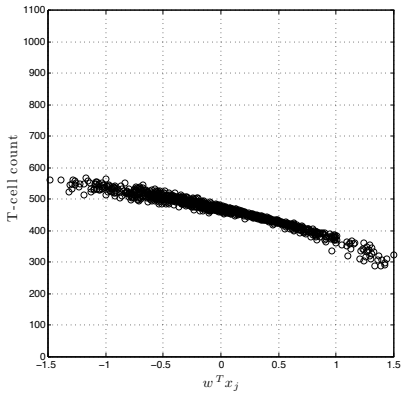
Sufficient Summary Plot After 1300 Days (N = 1000)



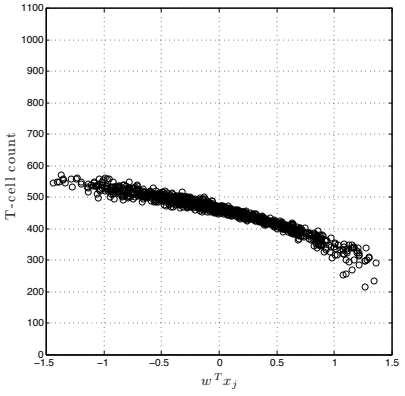
Sufficient Summary Plot After 1400 Days (N = 1000)



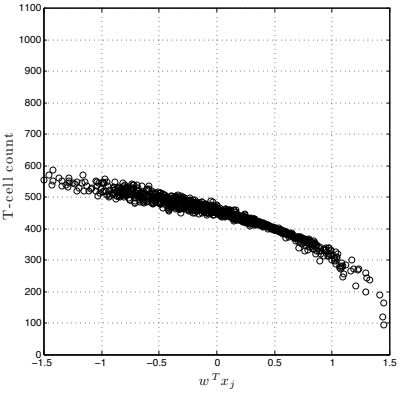
Sufficient Summary Plot After 1500 Days (N = 1000)



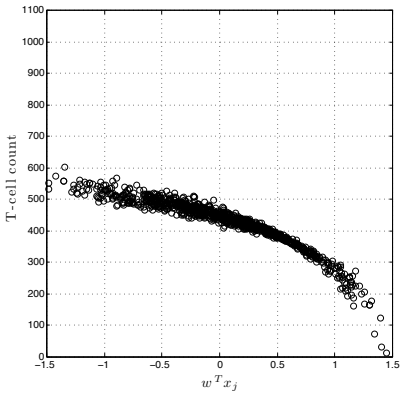
Sufficient Summary Plot After 1600 Days (N = 1000)



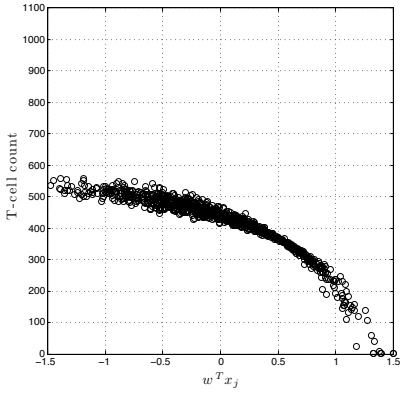
Sufficient Summary Plot After 1700 Days (N = 1000)



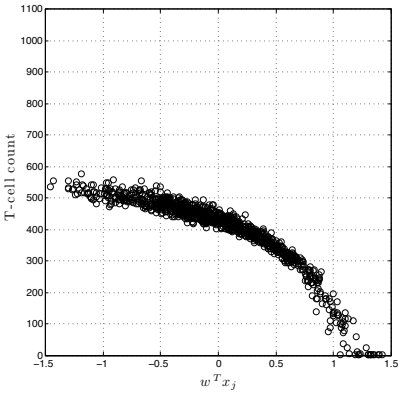
Sufficient Summary Plot After 1800 Days (N = 1000)



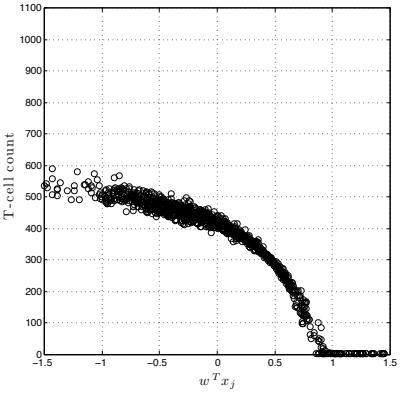
Sufficient Summary Plot After 1900 Days (N = 1000)



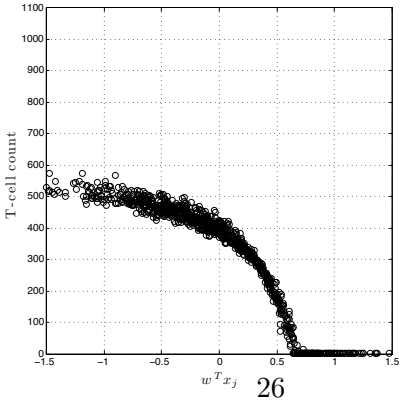
Sufficient Summary Plot After 2000 Days (N = 1000)



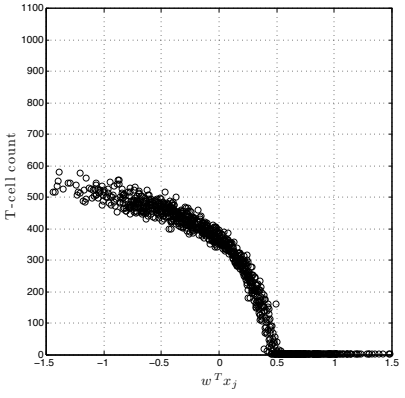
Sufficient Summary Plot After 2200 Days (N = 1000)



Sufficient Summary Plot After 2400 Days (N = 1000)



Sufficient Summary Plot After 2600 Days (N = 1000)



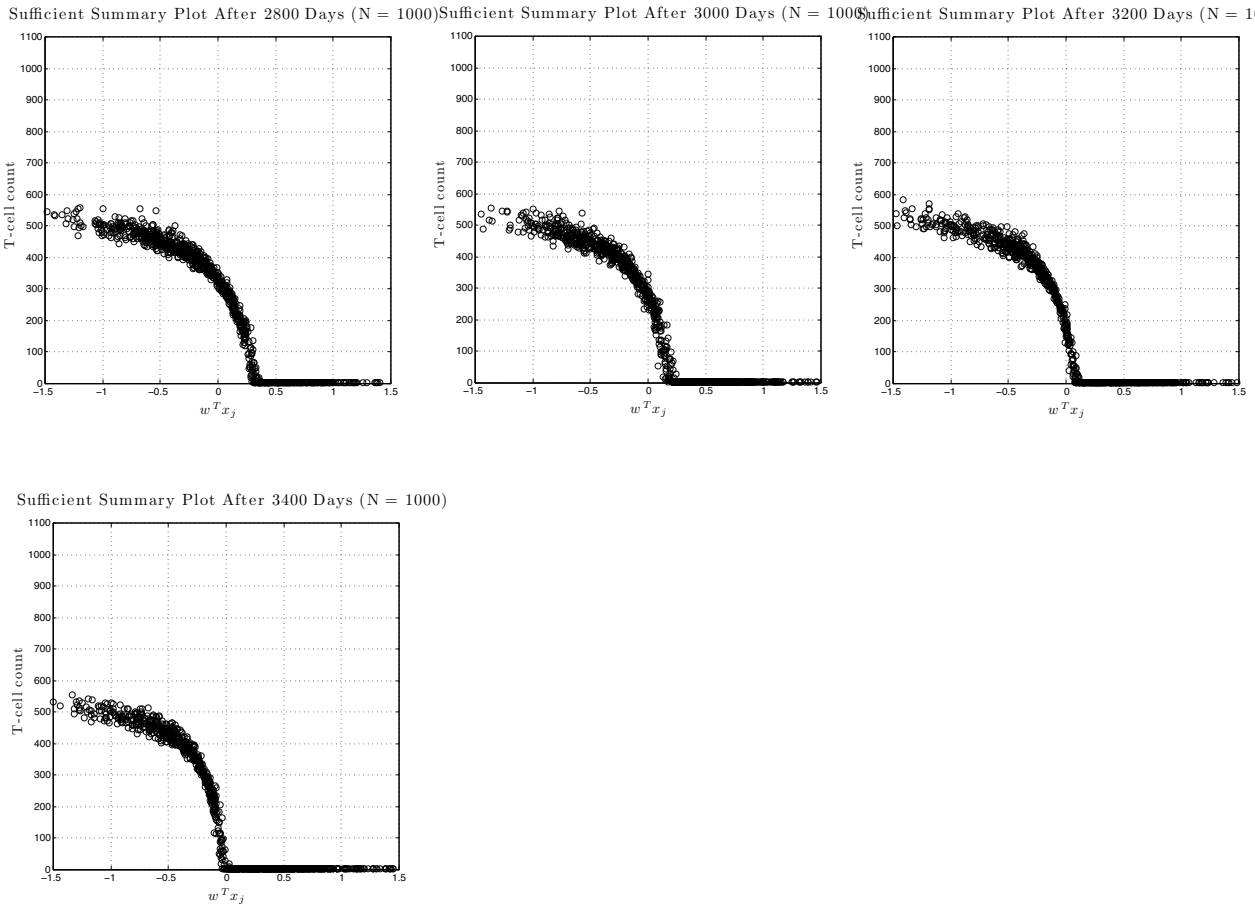


Figure 12: Sufficient summary plots throughout the course of the infection

When computing the measure of separation for all 55 time steps, Figure 13 (left) below shows the best dimension of the active subspace at each time step using (3.8)

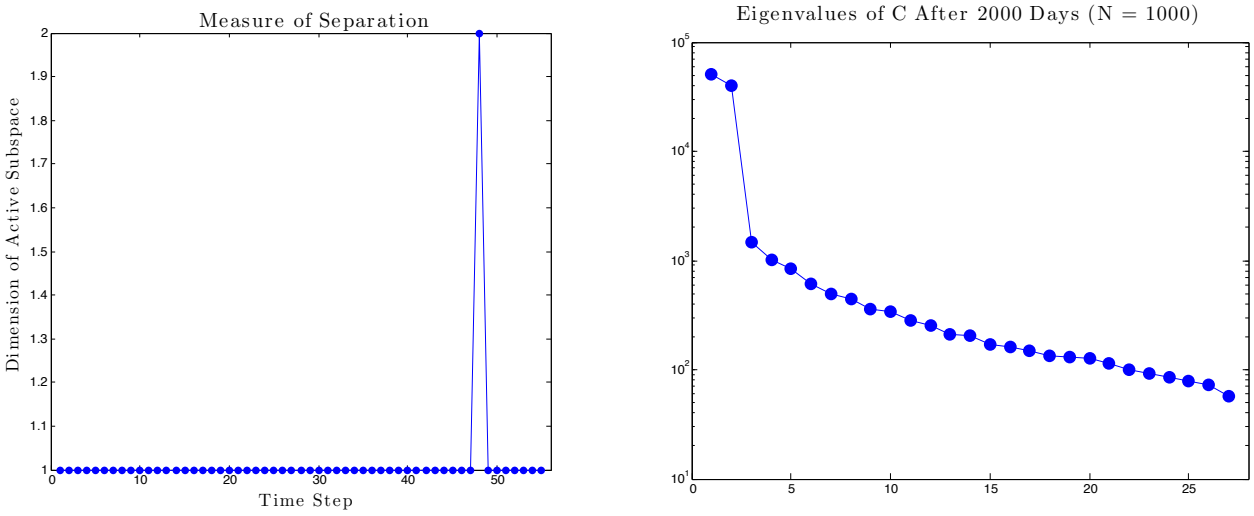


Figure 13: Dimension of the active subspace for each time step (left). Eigenvalues of the matrix C after 2000 days (right).

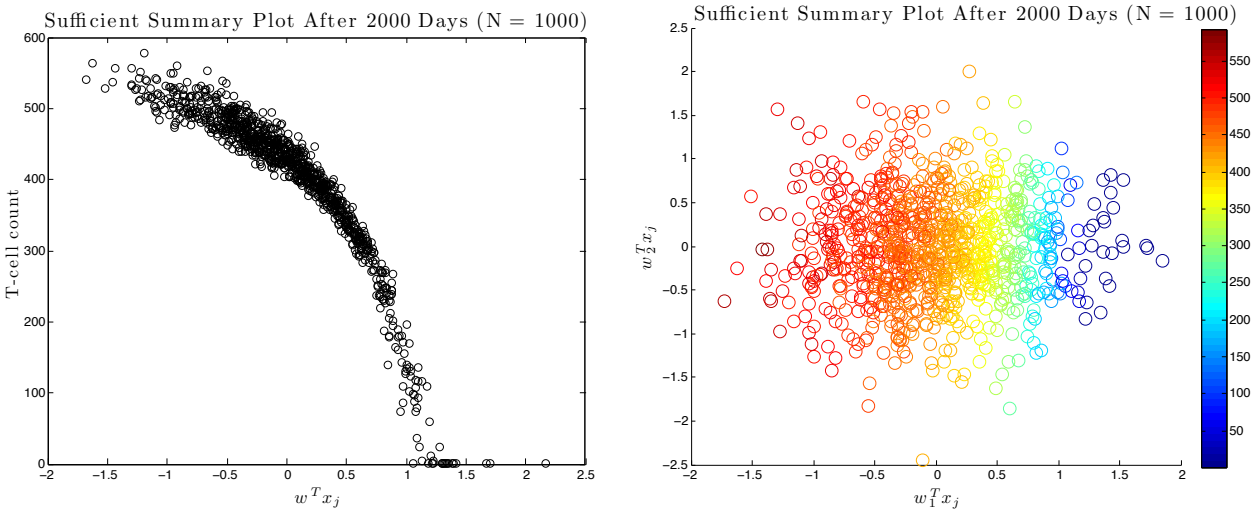


Figure 14: T-cell approximation after 65 days (left). First eigenvector of C with the quantity of interest being the T-cell count after 65 days (right).

From Figure 13 (left) we can see that at time step 48, which corresponds to 2000 days after infection, the best choice for the dimension of the active subspace is two. The plot of the eigenvalues of the matrix C can be seen in Figure 13 (right).

Clearly, the largest gap in the eigenvalues occurs between the second and third eigenvalues. Figure 14 shows the one and two dimensional sufficient summary plots respectively for the T-cell count after 2000 days.

Looking at the two dimensional sufficient summary plot in Figure 14 (right), we can see that there is not much variation in the T-cell count in the vertical direction. All of the variation in the T-cell count appears to be in the horizontal direction. Also, in Figure 14 (left) we can see that the one dimensional sufficient summary plot clearly shows a distinct trend. For these reasons, and also because of the simplicity in fitting a curve, rather than a surface, we chose to just use the one dimensional trend.

After following the method outlined in Section 4.1, Figure 15 below show the plots of the T-cell count with our approximation to the T-cell count using dynamic active subspaces. All of the data fits and the time discretization can be found in Appendix B.

From Figure 15 we can see that the approximation is not so good after 2000 days. This is because from 2000 days to 3400 days (or until the T-cell count hits zero) the step size in our global approximation is 200 days. In order to get a more accurate approximation we could decrease this

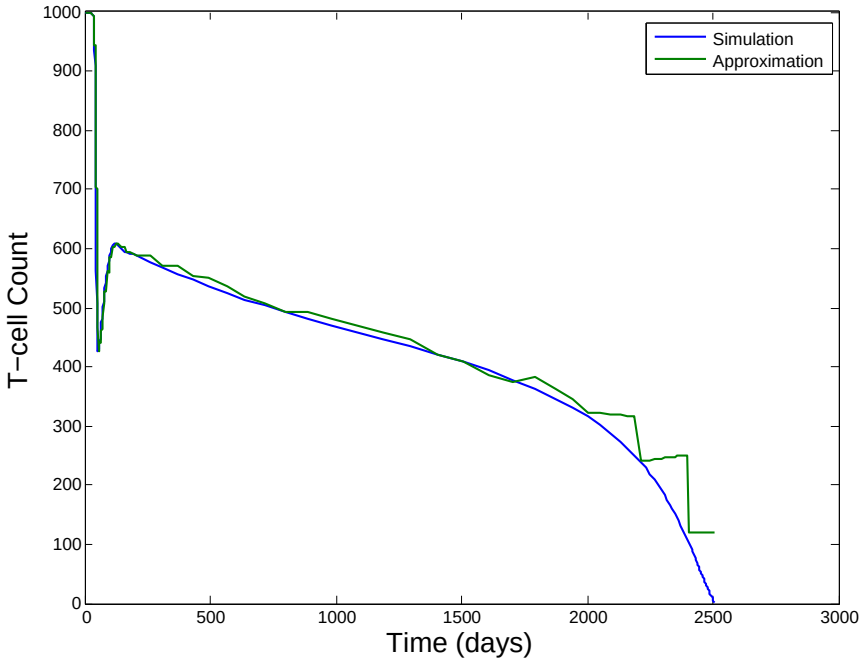


Figure 15: Global approximation of the T-cell count.

step size.

In order to test the accuracy of our analytical approximation to solutions of (2.1), we ran 100 simulations and computed the relative error in our approximation. For these error calculations, we randomly picked a time uniformly between 0 and 1500 days, and also randomly picked the parameter values uniformly within their ranges. The results can be seen below in Figure 16. From Figure 16 we can see that for all but four simulations our approximation was less than 5 percent off from the value given by the stiff differential equation solver, ode23s, in MATLAB.

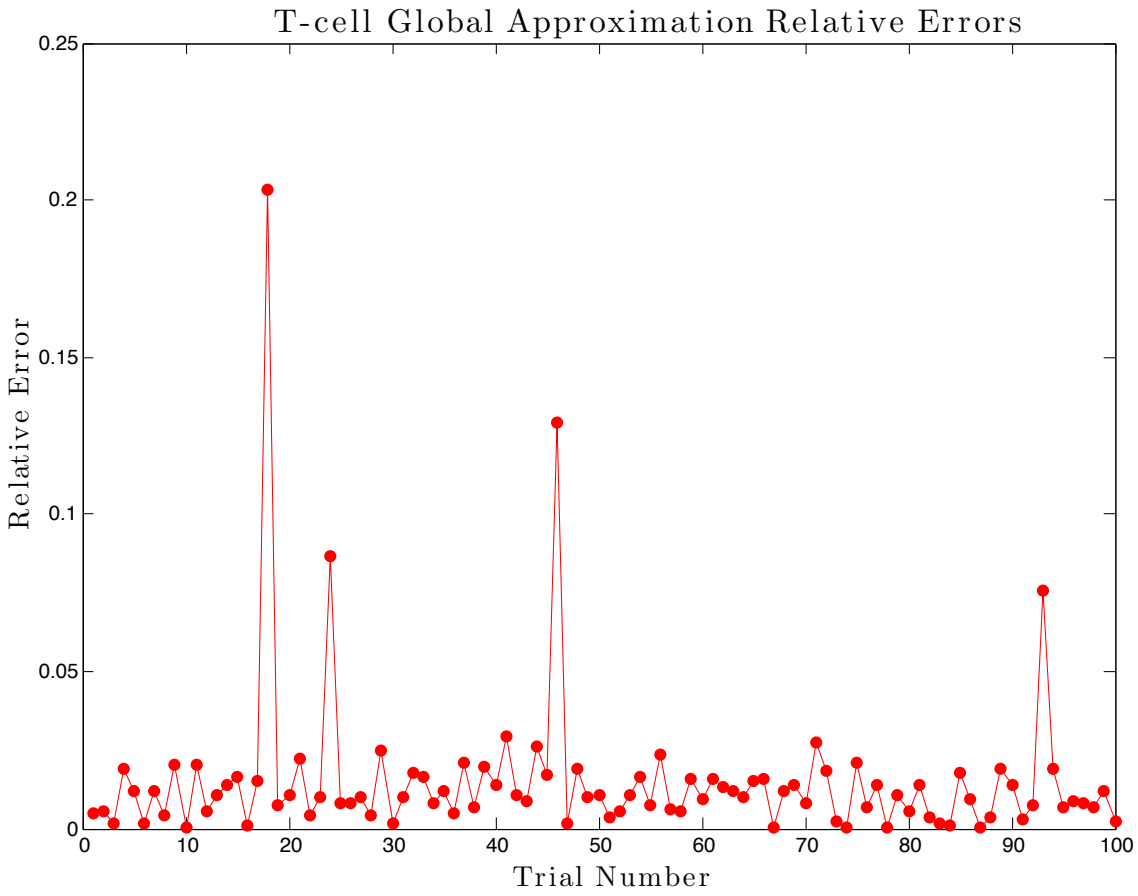


Figure 16: Relative error in the global approximation of the T-cell count.

CHAPTER 5

Conclusion

In this paper we considered the system (2.1) which is one of the first to accurately represent all three stages of the HIV infection. We showed that only one biologically feasible virus free steady state exists and gave conditions which guarantee the asymptotic stability of this steady state. Then, by using dynamic active subspaces, we reduced the computational model (2.1) into an analytic model (4.1). The efficacy of the model (4.1) was investigated by calculating the relative error compared to the system (2.1) solved with a stiff differential equations solver.

Going forward, we can now look at reducing the dimension of the system (2.1). This can be accomplished by looking at the weight vectors throughout the entire course of the infection and seeing which parameter weights were at or near zero the whole time. Then, by eliminating these interactions from (2.1) we hope to be able to derive a simpler system that still accurately predicts all three stages of the HIV infection.

Another direction to look in will be computing errors and convergence results for the dynamic active subspaces. The methods used in Chapters 3 and 4 can be used for any system of differential equations. So, by examining how the error, L^2 or L^∞, decreases with the step size $h = \max\{t_{i+1} - t_i \,|\, i = 0, \ldots, N-1\}$ we can determine a rate of convergence for this scheme.

REFERENCES CITED

[1] P.G. Constantine, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM 2015.

[2] P. Constantine and David Gleich, *Computing Active Subspaces* arXiv: 1408.0545

[3] M. Hadjiandreou, Raul Conejeros, Vassilis S. Vassiliadis, *Towards a Long-Term Model Construction for the Dynamic Simulation of HIV Infection*, Mathematical Biosciences and Engineering, **4** (2007) 489-504.

[4] D. E. Kirschner and A. S. Perelson, *A Model for the Immune Response to HIV: AZT Treatment Studies*

[5] T. M. Russi, *Uncertainty Quantification with Experimental Data and Complex System Models*, Ph.D. thesis, UC Berkeley, 2010.

[6] E. Vergu, A. Mallet, and J. Golmard, *A Modeling Approach to the Impact of HIV Mutations on the Immune System*

APPENDIX A
PROOFS OF THEOREMS 2.1, 2.2, and 2.3

In the appendix, we outline the proofs of the theorems given in Chapter 2.

Proof of Theorem 2.1. Beginning with (2.1), we search for steady states by assuming that all time derivatives are zero within the equations, and attempt to solve for the constant states $(T, T_I, T_L, M, M_I, CTL, V)$. Assuming $V = 0$ within the system of ODEs provides a significant reduction in the complexity of the system. The M equation implies $M = \frac{s_2}{\delta_4}$. Using this within the equation for M_I implies that $M_I = 0$, and it follows from the equation for V that $T_I = 0$ as well. Collecting these terms in the CTL differential equation implies that $CTL = \frac{s_3}{\delta_6}$. The equations for T_I and T_L together imply $T_L = 0$, and finally, with the remaining populations determined, the first equation implies $T = \frac{s_1}{\delta_1}$. Hence, the only non-infective steady state is

$$E_{NI} := \left(\frac{s_1}{\delta_1}, 0, 0, \frac{s_2}{\delta_4}, 0, \frac{s_3}{\delta_6}, 0\right).$$

□

Proof of Theorem 2.2. Beginning the same way as in the above proof, we set all time derivatives equal to zero and also assume $V = 0$. Once again the M equation implies $M = \frac{s_2}{\delta_4}$. Using this within the equation for M_I implies that either $M_I = 0$ or $CTL = -\frac{\delta_5}{K_6}$. For this steady state we choose the latter. Then, multiplying the T_I equation by $(1 - \psi)$ and the T_L equation by ψ and adding gives

$$0 = (\alpha_1 + \psi\delta_3)T_L + \left(\frac{(1-\psi)K_3\delta_5}{K_6} - (1-\psi)\delta_2\right)T_I \tag{A.1}$$

Creating a linear system with (A.1), the CTL equation, and the V equation and then solving for T_I, T_L, and M_I gives

$$\begin{aligned} T_I &= K_{10}\omega \\ T_L &= \frac{K_{10}\xi\omega}{K_6(\alpha_1 + \delta_3\psi)} \\ M_I &= -K_9\omega \end{aligned}$$

where

$$\omega = \frac{s_3K_6 + \delta_5\delta_6}{\delta_5(K_7K_{10} - K_8K_9)} \quad \text{and} \quad \xi = (1-\psi)(\delta_2 \mathrm{K}_6 - \delta_5 \mathrm{K}_3)$$

Lastly, plugging in the value of M_I into the T equation and solving for T gives

$$T = \frac{s_1}{\delta_1 - \omega K_2 K_9}$$

□

Finally, we sketch the proof of the asymptotic stability result, which utilizes a standard method from the theory of dynamical systems (i.e. the Hartman-Grobman and Routh-Hurtwitx theorems) to determine the qualitative behavior of (2.1).

Proof of Theorem 2.3. We begin by computing the Jacobian of (2.1) evaluated at the steady states E_{NI}

$$J(E_{NI}) = \begin{pmatrix} -\delta_1 & 0 & 0 & 0 & -\frac{K_2 s_1}{\delta_1} & 0 & \frac{(p_1 - c_1 K_1)s_1}{c_1 \delta_1} \\ 0 & -\frac{\delta_2\delta_6 + K_3 s_3}{\delta_6} & a_1 & 0 & \frac{pK_2 s_1}{\delta_1} & 0 & \frac{pK_1 s_1}{\delta_1} \\ 0 & 0 & -a_1 - \delta_3 & 0 & -\frac{(p-1)K_2 s_1}{\delta_1} & 0 & -\frac{(p-1)K_1 s_1}{\delta_1} \\ 0 & 0 & 0 & -\delta_4 & 0 & 0 & \frac{(K_4 - K_5)s_2}{\delta_4} \\ 0 & 0 & 0 & 0 & -\frac{\delta_5\delta_6 + K_6 s_3}{\delta_6} & 0 & \frac{K_5 s_2}{\delta_4} \\ 0 & \frac{K_7 s_3}{\delta_6} & 0 & 0 & \frac{K_8 s_3}{\delta_6} & -\delta_6 & 0 \\ 0 & K_9 & 0 & 0 & K_{10} & 0 & -\delta_7 - \frac{K_{11} s_1}{\delta_1} - \frac{(K_{12} + K_{13})s_2}{\delta_4} \end{pmatrix}$$

From this, we can see that three eigenvalues are certainly real and negative

$$\lambda_1 = -\delta_1, \quad \lambda_2 = -\delta_4, \quad \lambda_3 = -\delta_6.$$

The remaining four eigenvalues are more difficult to identify as they are determined by the quartic equation

$$a_4\lambda^4 + a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$

where

$$a_4 = \delta_1\delta_4\delta_6^2 > 0$$

$$\begin{aligned} a_3 &= \alpha_1\delta_1\delta_4\delta_6^2 + \delta_4\delta_6^2 K_{11}s_1 + \delta_1\delta_6^2 K_{12}s_2 + \delta_1\delta_6^2 K_{13}s_2 + \delta_1\delta_4\delta_6 K_3 s_3 \\ &\quad +\delta_1\delta_4\delta_6 K_6 s_3 + \delta_1\delta_2\delta_4\delta_6^2 + \delta_1\delta_3\delta_4\delta_6^2 + \delta_1\delta_4\delta_5\delta_6^2 + \delta_1\delta_4\delta_7\delta_6^2 \\ &> 0 \end{aligned}$$

and a_0, a_1, and a_2 are much longer and not necessarily positive.

Instead, we must impose conditions on each term to guarantee positivity, which is needed for the roots of the quartic to possess negative real part by the Routh-Hurwitz criteria. In particular, the two negative terms in a_2 are dominated by the remaining positive terms if and only if

$$K_1 K_9 \leq \delta_2 K_{11}$$

and

$$K_5 K_{10} \leq (K_{12} + K_{13})\delta_5.$$

The same conditions imply the positivity of a_1. For a_0, the negative terms are dominated by positive terms if and only if these two conditions hold and

$$K_2 K_5 K_9 s_1 s_2 \leq \delta_1\delta_2\delta_4\delta_5\delta_7 + \delta_4\delta_5 K_1 K_4 s_1 + \delta_1\delta_2 K_5 K_{10} s_2.$$

The final inequalities of the Routh-Hurwitz criteria are also implied by these conditions. Hence, defining

$$R_1 = \frac{K_1 K_9}{\delta_2 K_{11}}$$

$$R_2 = \frac{K_5 K_{10}}{(K_{12} + K_{13})\delta_5}$$

$$R_3 = \frac{K_2 K_5 K_9 s_1 s_2}{\delta_1\delta_2\delta_4\delta_5\delta_7 + \delta_4\delta_5 K_1 K_9 s_1 + \delta_1\delta_2 K_5 K_{10} s_2}$$

and

$$R_0 = \max\{R_1, R_2, R_3\}$$

we see that the equilibrium is locally asymptotically stable if and only if $R_0 \leq 1$.

□

APPENDIX B
TIME DISCRETIZATION AND CURVE FITS

Throughout this paper we used three different curve fits. They are given as follows:

$$T(\mathbf{X}, x) = X_1 x + X_2 \tag{B.1}$$

$$T(\mathbf{X}, x) = X_1 + X_2 \mathrm{arctan}\big(\mathrm{X_3 x} - \mathrm{X_4}\big) \tag{B.2}$$

$$T(\mathbf{X}, x) = U\big(X_5 - x\big)\Big(X_1 + X_2 \mathrm{arctan}\big(\mathrm{X_3 x} - \mathrm{X_4}\big)\Big) \tag{B.3}$$

In (B.3), $U(x)$ represents the heaviside step function. In the table below, if only X_1 and X_2 are given then fit (B.1) was used. If only X_1, X_2, X_3, and X_4 are given the fit (B.2) was used. Lastly, if all of X_1, X_2, X_3, X_4, and X_5 are specified, then fit (B.3) was used.

Step	Time (days)	X_1	X_2	X_3	X_4	X_5
1	2	-0.7001	999.9998	-	-	-
2	24	-7.5385	999.8561	-	-	-
3	30	-9.1399	999.1159	-	-	-
4	31	-9.4223	998.7555	-	-	-
5	32	-9.7547	998.2877	-	-	-
6	33	-10.0848	997.5888	-	-	-
7	34	-10.8047	996.5409	-	-	-
8	35	-11.8506	994.9286	-	-	-
9	36	409.5503	-391.4344	5.1000	12.9873	-
10	37	920.4146	-62.0121	1.3540	2.2269	-
11	38	831.2210	-123.0287	1.8728	3.2643	-
12	39	883.6065	-87.2336	1.7616	2.1759	-
13	40	822.9661	-132.5512	1.7187	2.1786	-
14	45	690.1227	-264.6748	1.1813	0.6638	-
15	50	700.7747	-280.6336	1.1062	-0.5182	-
16	55	666.5974	-202.0994	1.4524	-1.7229	-
17	60	676.6785	-179.1918	1.5455	-3.0025	-

18	65	851.8952	-306.5000	0.6299	-2.9093	-
19	70	-18.5318	494.0200	-	-	-
20	75	-18.1534	514.7776	-	-	-
21	80	-18.3724	534.3451	-	-	-
22	90	-19.8289	568.7354	-	-	-
23	100	-22.3309	595.8887	-	-	-
24	110	-25.6259	614.6021	-	-	-
25	120	-29.7189	624.2096	-	-	-
26	130	-33.4046	625.7675	-	-	-
27	140	-35.7335	622.7802	-	-	-
28	160	-36.2728	616.0632	-	-	-
29	180	-36.4516	613.5536	-	-	-
30	200	-37.4263	611.6527	-	-	-
31	300	-41.0519	596.0775	-	-	-
32	400	-42.9225	579.6616	-	-	-
33	500	-44.1441	564.7162	-	-	-
34	600	-45.8963	551.3280	-	-	-
35	700	-47.6779	539.2665	-	-	-
36	800	-49.4288	528.3012	-	-	-
37	900	-51.8465	518.2273	-	-	-
38	1000	-55.4417	508.8040	-	-	-
39	1100	-59.1658	499.9343	-	-	-
40	1200	-62.0424	491.0517	-	-	-
41	1300	-68.1405	482.0779	-	-	-
42	1400	-7.0270 x 10^4	-4.5331 x 10^4	0.0016 x 10^4	0.0099 x 10^4	-
43	1500	-6.9171 x 10^4	-4.4546 x 10^4	0.0032 x 10^4	0.0134 x 10^4	-
44	1600	-6.9172 x 10^4	-4.4544 x 10^4	0.0034 x 10^4	0.0133 x 10^4	-
45	1700	-6.8164 x 10^4	-4.3859 x 10^4	0.0056 x 10^4	0.0161 x 10^4	-
46	1800	-364.8267	-664.2612	1.3581	2.7775	1.5945
47	1900	-314.2482	-612.6825	1.6793	2.8535	1.3639

48	2000	57.2136	-354.2093	1.5486	1.8115	1.2794
49	2200	-82.1383	-442.8312	2.0436	2.0987	0.9351
50	2400	-178.0935	-497.9389	2.7357	2.2393	0.6819
51	2600	-70.6040	-423.2509	2.9292	1.6720	0.5133
52	2800	-134.7672	-459.1508	3.6991	1.5891	0.3479
53	3000	-17.5600	-375.5894	4.0446	0.9448	0.2220
54	3200	-214.5754	-504.5574	5.3670	0.9768	0.0976
55	3400	-17.1234	-365.0256	5.7169	0.0915	0.0078

APPENDIX C

CODES

This is the code used to calculate the weight vector for the one dimensional active subspace.

```
%X(1) --> T   = Healthy T-cells
%X(2) --> T_I = Infected T-cells
%X(3) --> T_L = Latently-infected T-cells
%X(4) --> M   = Healthy macrophages
%X(5) --> M_I = Infected macrophages
%X(6) --> C   = Cytotoxic T-lymphocytes population
%X(7) --> V   = HIV population

%Initialize algorithm parameters
N = 1000;   %Number of samples for each time step
time = [];  %Time values
h = 1e-6;   %Finite difference step size
trial = 1;  %Trial number (used when saving figures)

%Pre-allocate memory
q = zeros(N,numel(time));          %Output of interest (T-cell count
    after time(ii) days)
qplus = zeros(27,1);               %Perturbed output of interest
gradq = zeros(27,N,numel(time)); %Gradient of output of interest
Xs = zeros(N,27,numel(time));      %To save the normalized paramters
w = zeros(27,numel(time));         %Weight vectors
evalues = zeros(27,numel(time)); %Eigenvalues of the C matrix
diff = zeros(numel(time),1);       %Differences in largest and smallest
    element of gradq
I = eye(27);                       %27x27 identity matrix

%Set upper and lower bounds for paramters
xl = .975*[10;.15;5;.2;55.6;3.87e-3;1e-6;4.5e-4;7.45e-4;5.22e-4;3e
```

```
    -6;3.3e-4;6e-9;5.37e-1;2.85e-1;7.79e-6;1e-6;4e
    -5;.01;.28;.05;.005;.005;.015;2.39;3e-4;.97];
xu = 1.025*[10;.15;5;.2;55.6;3.87e-3;1e-6;4.5e-4;7.45e-4;5.22e-4;3e
    -6;3.3e-4;6e-9;5.37e-1;2.85e-1;7.79e-6;1e-6;4e
    -5;.01;.28;.05;.005;.005;.015;2.39;3e-4;.97];

%Set initial conditions
IC = [1000,0,0,30,0,500,1e-3];

%Run simulation
for ii = 1:numel(time)

    for jj = 1:N

        %Randomly sample parameters within acceptable ranges
        Xs(jj,:,ii) = 2*rand(1,27) - 1;
        params = 1/2*(diag(xu - xl)*Xs(jj,:,ii)' + (xu + xl));

        %Create function handles
        f1 = @(X) params(1) + params(4)/(X(7) + params(5))*X(7)*X(1) -
            params(19)*X(1) - (params(6)*X(7) + params(7)*X(5))*X(1);
        f2 = @(X) params(27)*(params(6)*X(7) + params(7)*X(5))*X(1) +
            params(26)*X(3) - params(20)*X(2) - params(8)*X(2)*X(6);
        f3 = @(X) (1 - params(27))*(params(6)*X(7) + params(7)*X(5))*X
            (1) - params(26)*X(3) - params(21)*X(3);
        f4 = @(X) params(2) + params(9)*X(7)*X(4) - params(10)*X(7)*X
            (4) - params(22)*X(4);
        f5 = @(X) params(10)*X(7)*X(4) - params(23)*X(5) - params(11)*X
            (5)*X(6);
        f6 = @(X) params(3) + (params(12)*X(2) + params(13)*X(5))*X(6)
            - params(24)*X(6);
        f7 = @(X) params(14)*X(2) + params(15)*X(5) - params(16)*X(7)*X
```

```
    (1) - (params(17) + params(18))*X(7)*X(4) - params(25)*X(7);
f = @(t,X) [f1(X),f2(X),f3(X),f4(X),f5(X),f6(X),f7(X)]';

%Numerically solve system of ODEs
options = odeset('events', @totalZero);
[tout,fout] = ode23s(f,[0,time(ii)],IC,options);
q(jj,ii) = fout(end,1);

for kk = 1:27

    %Numerically solve perturbed system of ODEs
    xplus = Xs(jj,:,ii)' + h*I(:,kk);
    paramsplus = 1/2*(diag(xu - xl)*xplus + (xu + xl));
    f1 = @(X) paramsplus(1) + paramsplus(4)/(X(7) + paramsplus
        (5))*X(7)*X(1) - paramsplus(19)*X(1) - (paramsplus(6)*X
        (7) + paramsplus(7)*X(5))*X(1);
    f2 = @(X) paramsplus(27)*(paramsplus(6)*X(7) + paramsplus
        (7)*X(5))*X(1) + paramsplus(26)*X(3) - paramsplus(20)*X
        (2) - paramsplus(8)*X(2)*X(6);
    f3 = @(X) (1 - paramsplus(27))*(paramsplus(6)*X(7) +
        paramsplus(7)*X(5))*X(1) - paramsplus(26)*X(3) -
        paramsplus(21)*X(3);
    f4 = @(X) paramsplus(2) + paramsplus(9)*X(7)*X(4) -
        paramsplus(10)*X(7)*X(4) - paramsplus(22)*X(4);
    f5 = @(X) paramsplus(10)*X(7)*X(4) - paramsplus(23)*X(5) -
        paramsplus(11)*X(5)*X(6);
    f6 = @(X) paramsplus(3) + (paramsplus(12)*X(2) + paramsplus
        (13)*X(5))*X(6) - paramsplus(24)*X(6);
    f7 = @(X) paramsplus(14)*X(2) + paramsplus(15)*X(5) -
        paramsplus(16)*X(7)*X(1) - (paramsplus(17) + paramsplus
        (18))*X(7)*X(4) - paramsplus(25)*X(7);
    f = @(t,X) [f1(X),f2(X),f3(X),f4(X),f5(X),f6(X),f7(X)]';
```

```
69              options = odeset('events', @totalZero);
70              [tout,fout] = ode23s(f,[0,time(ii)],IC,options);
71              qplus(kk) = fout(end,1);
72
73          end
74
75          %Calculate the gradients using finite differences
76          gradq(:,jj,ii) = (qplus - q(jj,ii))/h;
77
78      end
79  end
80
81  %Compute the weights, eigenvalues, and plot results
82  close all
83  for nn = 1:numel(time)
84
85      %Compute the singular value decomposition of C
86      [U,S,V] = svd(1/sqrt(N)*gradq(:,:,nn));
87      w(:,nn) = U(:,1);
88      w2 = U(:,2);
89
90      %Compute the eigenvalues of C
91      evalues(:,nn) = diag(S.^2);
92
93      %Plot the eigenvalues of C on a log plot
94      fig = figure;
95      semilogy(1:27,evalues(:,nn),'.-b','MarkerSize',30)
96      title(['Eigenvalues of C After ' int2str(time(nn)) ' Days (N = '
            int2str(N) ')'],'Interpreter','latex','Fontsize',16,'FontWeight'
            ,'bold','Position',[12.5 180 0])
97      xlim([0,28])
98      set(get(gca,'Title'),'Units','Normalized','Position',[.45,1.04])
```

```
99      set(fig,'PaperUnits','inches','PaperSize',[10 8])
100     hgexport(fig, ['Evalues' int2str(trial) '_' int2str(time(nn)) '.pdf
            '], hgexport('factorystyle'), 'Format', 'pdf');
101
102     %Plot the weight vector
103     fig = figure;
104     plot(1:27,w(:,nn),'.-b','MarkerSize',30)
105     title(['Weight Vector After ' int2str(time(nn)) ' Days (N = '
            int2str(N) ')'],'Interpreter','latex','Fontsize',16,'FontWeight'
            ,'bold','Position',[12.5 1.05 0])
106     xlabel('Parameters','Interpreter','latex','Fontsize',14)
107     ylabel('Parameter Weights','Interpreter','latex','Fontsize',14)
108     set(get(gca,'Title'),'Units','Normalized','Position',[.45,1.04])
109     xlim([0,28])
110     ylim([-1,1])
111     set(fig,'PaperUnits','inches','PaperSize',[10 8])
112     hgexport(fig, ['WV' int2str(trial) '_' int2str(time(nn)) '.pdf'],
            hgexport('factorystyle'), 'Format', 'pdf');
113
114     %Plot the absolute value of weight vector components
115     fig = figure;
116     plot(1:27,abs(w(:,nn)),'.-b','Markersize',30)
117     title(['Parameter Weight After ' int2str(time(nn)) ' Days (N = '
            int2str(N) ')'],'Interpreter','latex','Fontsize',16,'Fontweight'
            ,'bold','Position',[12.5 1.05 0])
118     xlabel('Parameters','Interpreter','latex','Fontsize',14)
119     ylabel('Magnitude of Weight','Interpreter','latex','Fontsize',14)
120     xlim([0,28])
121     ylim([0,1])
122     set(fig,'PaperUnits','inches','PaperSize',[10 8])
123     hgexport(fig, ['WVmag' int2str(trial) '_' int2str(time(nn)) '.pdf'
            ], hgexport('factorystyle'), 'Format', 'pdf');
```

```
124
125     %Sufficient summary plot
126     fig = figure;
127     plot(Xs(:,:,nn)*w(:,nn),q(:,nn),'ko');
128     title(['Sufficient Summary Plot After ' int2str(time(nn)) ' Days (N
            = ' int2str(N) ')'],'Interpreter','latex','Fontsize',16,'
           FontWeight','bold','Position',[-.15 1130 0])
129     xlabel('$w^T x_j$','Interpreter','latex','FontSize',14)
130     ylabel('T-cell count','Interpreter','latex','FontSize',14)
131     set(get(gca,'Title'),'Units','Normalized','Position',[.45,1.04])
132     xlim([-2,2])
133     ylim([0,1100])
134     axis square;
135     grid on;
136     set(fig,'PaperUnits','inches','PaperSize',[10 8])
137     hgexport(fig, ['SSP' int2str(trial) '_' int2str(time(nn)) '.pdf'],
           hgexport('factorystyle'), 'Format', 'pdf');
138
139     %Find the difference of max and min gradq to check for errors
140     diff(nn) = max(max(gradq(:,:,nn))) - min(min(gradq(:,:,nn)));
141
142 end
143
144 %Save the trial data
145 save(['trial' int2str(trial) 'data.mat'],'evalues','Xs','q','w','diff',
       'time')
```

This code is used to stop the stiff differential equations solver when the T-cell count is almost at zero.

```
1 function [val, isterm, dir] = totalZero(~,y)
2
3 val = min((y(7) - 1e-16),(y(1) - 1));
```

```
4 isterm = 1;
5 dir = -1;
```