MATHEMATICAL MODELING OF CHEMICAL SIGNALS IN INFLAMMATORY PATHWAYS

by

Ian Price

B.A., University of Florida, 2004

Submitted to the Graduate Faculty of

The Department of Mathematics in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH

MATHEMATICS DEPARTMENT

This dissertation was presented

by

Ian Price

It was defended on

April 01, 2011

and approved by

Dr G Bard Ermentrout, University Professor, Department of Mathematics

Dr Giles Clermont, MD, Department of Critical Care Medicine, UPMC

Dr Ivan Yotov, Professor and Chair, Department of Mathematics

Dissertation Advisor: David Swigon, Asst Professor, Department of Mathematics

Mathematical Modeling of Chemical Signals in Inflammatory Pathways

Ian Price, PhD

University of Pittsburgh, 2011

Mechanistic, autonomous, ordinary differential equations represent a powerful way to crystalize and reproduce the dynamics of complex, nonlinear interactions. Design and calibration of these models, however, represent a challenge to the construction of fully validated models. Various parameter techniques are employed, evaluated and improved upon for the purpose of fitting in a nonlinear setting. Cells communicate with other cells and their environment by producing and receiving chemical signals. In the context of pathogen response, these signals regulate how the collective of cells reacts. One such undifferentiated response to signal is known as inflammation, and it is an important mediator of pathogen clearance as well as tissue healing; however, it also has the potential to damage the surrounding tissue when regulatory mechanisms break down. Models are built using the mechanisms of these interactions to produce a high level effect, and to predict what measures can be taken, as in influenza, to prevent dysregulation. The models developed for inflammatory response first take into consideration the production and reception of immune factors, cytokines, and then put these mechanisms into the context of tissue level response to external signals and internal signals in the form of system damage. This is incorporated into a nonlinear model of immune response to Influenza A Virus, with innate, adaptive, and humoral immunity components. The model is calibrated against data for both sublethal and lethal initial dosages. A model of mosquito response to exogenous cytokine as immune stimulation is also explored. Successful model fitting using Metropolis-Hastings methods yields multi-objective results for nonlinear deterministic models.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would first like to thank my committee:

- Bard Ermentrout—the man who helped me a graduate student.
- David Swigon—my advisor who has been continually supportive of my research and encouraging in my studies.
- Gilles Clermont—a man who continues to innovate and encourage others to do the same.
- Ivan Yotov—the professor who brought into computer mathematics, something without which I would be lost today.

To my other professors:

- Jon Rubin—who always holds his students to a higher standard
- Juan Manfredi—who teaches an infectious pleasure of math.

To my family, without whom I would not be here today: my father Rick, my mother Christine, my brothers Wyatt and Coulton, my uncle Bob, and my aunt Dolores.

1.0 INTRODUCTION

Mathematical models come in a variety and uses, the common feature among them being the abstraction of a complex physical system into the components considered most pertinent. Causal models, as opposed to correlative models, further represent the physical system as a series of causes and effects; the mechanistic ordinary differential equation models used in this research rely on sufficient information being available to construct a model with sufficient detail to reproduce the causes and effects observed in physical experiments. The added advantage to a causal model over a correlative model is the added confidence in interpolating and extrapolating the effects of unmeasured inputs.

The availability of such models provides scientists with important tools for *in silico* experiments of several types. First, with a validated model, one can design a new physical experiment and test with the model before proceeding with an expensive experiment that may not succeed in its objectives. But perhaps as important, one can hypothesize and test within the model system different types of mechanisms, the relative contributions of mechanisms, and complementary as well as overlapping effects.

As technology advances, data collection becomes more robust, and computing power increases: modeling is more available and perhaps more necessary to give meaning to this robust data. And so, techniques continue to be developed to address issues of modeling within fields of study. In biology, the issues of multi-scale effects, nonlinear dynamics, complexity, and signal

response to exogenous signal each present challenges to the modeling community; however, it is not enough to use increased computing power as brute force without consistently questioning how models should be constructed and what information and dynamics we can gain by re-gearing models with more subtle dynamics to better describe the physical systems modeled.

This work primary focuses on the effects of exogenous chemical signalling to a system in the context of inflammatory response. The models developed examine the effects of chemical on the production of other chemical signals, the effects of all those signals on cells and cell movement, and finally the cumulative effect on the tissue. To achieve this, rather than build large linear models, these studies use simplification and reduction of space to create smaller, more intuitive, though less tractable, non-linear models. From here, techniques of parameter fitting are explored and refined, and the global dynamics and potential behaviors of these models are fleshed out.

The work begins by focusing on extending Michaelis-Menten regulation to achieve a saturating inhibition often seen in biology. This saturating inhibition is then used in the dynamics of Immune-cell production and inhibition of cytokine. This is then placed in the greater context of tissue and inflammation, which in turn is used as a compartment in an adaptive immune response model to influenza A virus. In another model of system response to exogenous signal, a model of a negative feedback mechanism is developed so that a stable limit cycle is a potential outcome, and this limit cycle is fit to data.

While many types of modeling continue in development, e.g. partial differential equations, stochastic differential equations, and agent based models, the non-linear models of ordinary differential equations develop in tandem. And as computational power continues to

develop, the theory behind different modeling techniques becomes even more important as application grows.

2.0 CHEMICAL SIGNALLING AND INFLAMMATION

Chemical signalling is a rich field of modeling interest and has been since the beginning of quantitative chemistry. For the simplest chemical interaction, models using Mass-Action kinetics describe and predict outcomes well. In the 1920s, Michaelis and Menten found that Mass Action did not fully describe chemical reactions in the presence of a catalyst (Michaelis), and their research yielded the Michaelis-Menten formula used today. In biological systems, the catalyst is typically an enzyme or other factor on the same chemical scale; mass-action models with a quasi-steady state assumption applied produce a Michaelis-Menten formula as output (Edelstein-Keshet). However, the interaction of a chemical with a cell is a multi-scale problem, and requires more care. This problem is usually approached using Mass-Action Kinetics, Michaelis-Menten, or a Hill type function, with the function chosen by best fit to experimental data. Various other models exist also approaching this problem (Fall), each with its strengths and weaknesses.

The first goal of this project is to model the interaction of an innate immune cell, nominally the macrophage, with three classes of chemicals: pro-inflammatory cytokines, anti-inflammatory cytokines, and chemokines. This abstraction is sufficient for the level of accuracy sought in the problem. We assume for the sake of the model that pro-inflammatory cytokines induce the macrophage to produce all three classes of chemicals, that anti-inflammatory cytokines inhibit the production of all three, and that chemokines increase the local number of

macrophages and other cell types by inward migration. From here, we engage the biological knowledge of how these components interact, and produce a model for their activity.

The activity and synergistic interaction of these chemicals and cells account for the bulk of inflammatory activity in tissue. In an idealized environment, the macrophages and chemicals in a system are at rest and non-reactive in the absence of external stimuli; however, in the presence of an external stimulus such as particulates or bacteria, pro-inflammatory chemicals drive this system so that the external stimulus can be remedied by the macrophages. But, these mechanisms can also react to stimuli for which clearance is not possible, and when the immune activity does not de-activate, cause damage to the surrounding environment, the tissue. Understanding these mechanisms, so to have an accurate and predictive model of this small system is useful in describing larger systems where auto-inflammatory damage is an issue.

2.1 INTRACELLULAR KINETICS

2.1.1 Michaelis-Menten Kinetics

As a point of reference, I will briefly review the concept of Michaelis-Menten kinetics. Assume that we have two chemical species, A and B, where A converts naturally to B with the aid of an enzyme, and the rate of production can be described by:

$$B' = \frac{V_{max} A}{V_{sub} + A} \tag{0.1}$$

V_{max} is the maximal production rate, so that for any arbitrarily large amount of A, the production rate cannot exceed V_{max}. V_{sub} is the substrate affinity of A for the enzyme; in the

most common definition, it is the amount of A for which half the maximal production rate is achieved. In chemical terms, it describes how well the substrate binds to the enzyme, with a larger V_{sub} corresponding to less binding. A competitive inhibitor is one that acts to substitute itself to the receptor in lieu of the substrate, so when one is introduced it effectively increases V_{sub}, and hence the fraction of maximal production is smaller for a fixed level of substrate. By contrast, an uncompetitive inhibitor lowers V_{max} by delaying the final step in production, thereby directly lowering the maximal production rate. Whereas in the presence of a competitive inhibitor, the maximal production rate can still be reached by saturating the system with substrate, with an uncompetitive inhibitor no amount of saturation will lead back to maximal production. An example of inhibition that simultaneously affects both terms V_{max} and V_{sub} is allosteric, or a type of mixed, inhibition that functions by binding externally to the receptor.

In the model that we produce, competitive and uncompetitive inhibition will be nonlinear functions of the inhibitor.

2.1.2 Saturation Kinetics

For a cell to begin producing chemicals, the external chemical stimuli and pro-inflammatory cytokines must start a chemical chain to activate the section of the cell's DNA that produces the desired chemicals. After the DNA is transcribed, producing messenger RNA (mRNA), the RNA must be translated into a chain of amino acids, which are folded into the correct conformation to be the chemical product exported by the cell.

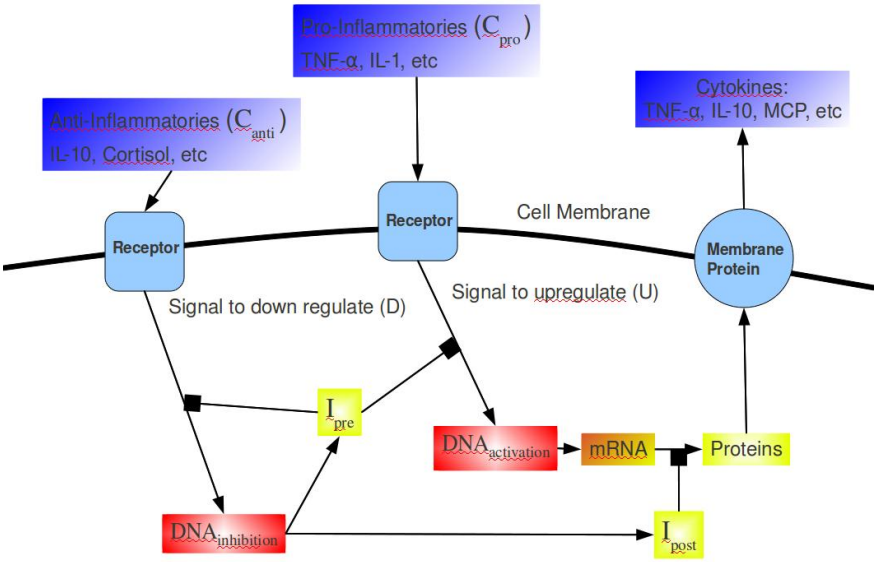

Figure 1. Schematic of Macrophage:

We consider the macrophage to have two types of receptors, one for pro and one for anti-inflammatory cytokines. Receptor activation causes an intracellular cascade to up-regulate DNA transcription. The products of the transcription and translation affect both the production of cytokines, and the efficacy of the intracellular processes.

There are hence many steps in the intracellular production of chemical products where an inhibitor may act. For our purposes, we will consider pre-transcription and post-transcriptional inhibition. We consider then, three cases: where there is only pre-transcriptional inhibition, only post-transcriptional, or both working together. To see how this works, we consider a cell that is exposed to pro-inflammatory cytokines (C_{pro}) and anti-inflammatory cytokines (C_{anti}). C_{pro} and C_{anti} activate the external cell receptors corresponding to up-regulation (U_{reg}) and down-regulate (D_{reg}), respectively. D_{reg} activates the inhibition we seek to model (I_{pre}, I_{post}, or both). I_{pre} is the inhibitor on pre-transcriptional inhibition, and I_{post} is the inhibitor for post-transcriptional inhibition. U_{reg} activates DNA transcription to RNA (R_{NA}), and our final product (P) come from translation of the RNA into proteins. We adopt a quasi-steady state assumption for all equations

except for P, since under normative circumstances there are more substrates than receptors to bind to.

$$
\begin{aligned}
\frac{dD_{reg}}{dt} &= \alpha C_{anti} - \mu_1 D_{reg} \\
\frac{dU_{reg}}{dt} &= \beta C_{pro} - \mu_2 U_{reg} \\
\frac{dI_{pre}}{dt} &= \frac{\gamma D_{reg}}{D_{reg} + \eta_1 I_{pre} + \nu_1} - \mu_3 I_{pre} \\
\frac{dI_{post}}{dt} &= \begin{cases} \delta D_{reg} - \mu_4 I_{post}, & \text{without } I_{pre} \text{ present} \\ \dfrac{\delta D_{reg}}{D_{reg} + \eta_2 I_{pre} + \nu_2} - \mu_4 I_{post}, & \text{with } I_{pre} \text{ present} \end{cases} \\
\frac{dR_{NA}}{dt} &= \begin{cases} \varepsilon U_{reg} - \mu_5 R_{NA}, & \text{without } I_{pre} \text{ present} \\ \dfrac{\varepsilon U_{reg}}{U_{reg} + \eta_3 I_{pre} + \nu_3} - \mu_5 R_{NA}, & \text{with } I_{pre} \text{ present} \end{cases} \\
\frac{dP}{dt} &= \begin{cases} \varphi R_{NA} - \mu_6 P, & \text{without } I_{post} \text{ present} \\ \dfrac{\varphi R_{NA}}{R_{NA} + \eta_4 I_{post} + \nu_4} - \mu_6 P, & \text{with } I_{post} \text{ present} \end{cases}
\end{aligned}
\tag{0.2}
$$

Post-transcription inhibition alone leads to competitive inhibition of the final product in the traditional sense.

$$
\frac{dP}{dt} = \frac{V_{max} C_{pro}}{V_{sub} + V_I \, C_{anti} + C_{pro}} - \mu P \tag{0.3}
$$

Notice how, in opposition to the normal Michaelis-Menten function, the constant V_{sub} is replaced by the linear function in of the inhibitor C_{anti}: $V_{sub} + V_I \, C_{anti}$, where V_I is an algebraic combination of parameters in (0.3).

However, pre-transcription inhibition alone leads to the competitive term being replaced by a Michaelis-Menten function of the inhibitor:

$$\frac{dP}{dt} = \frac{V_{max} C_{pro}}{V_s + \left(\frac{K_{max} C_{anti}}{K_s + C_{anti}} \right) + C_{pro}} - \mu P \tag{0.4}$$

The algebraic formulae for all terms are available as a supplement. Additionally, some approximations were made in computing the formula, which are available.

Instead of the substrate affinity being replaced by a linear function, as in the case of competitive inhibition, it is now replaced by the nonlinear function of the form: $V_s + K_{\max} C_{anti} / K_s + C_{anti}$. This function is also now monotone increasing in C_{anti} and bounded by non-zero values. Since the function is bounded, inhibition cannot drive production to exponentially small values. Biologically, the affinity of the chemical protagonist must be bounded by the interval $V_s, V_s + K_{max}$, which may be more relevant and practical to modeling real systems than infinitely increasing inhibition. In the end result, the effect of the inhibitor saturates, so that flooding the system with it has the same effect as introducing an optimal level of inhibitor.

Intuitively, the case where both pre- and post-transcriptional are applied gives the most complicated terms. The best comparison to traditional dynamics is to allosteric inhibition, which combines competitive and uncompetitive terms. As with the pre-transcriptional case, both the competitive and uncompetitive terms are saturating functions, giving the form:

$$\frac{dP}{dt} = \frac{V_{max} C_{pro}}{C_{pro}\left(1 + \frac{K_{11} C_{anti} + K_{12}}{C_{anti} + K_{22}} \right) + V_s \left(1 + \frac{K_{11} C_{anti} + K_{12}}{C_{anti} + K_{22}} \right)\left(\frac{C_{anti} + J_{12}}{C_{anti} + J_{22}} \right)} - \mu P \tag{0.5}$$

The parameters are algebraic combination of the parameters in (0.2), and can be found in the appendix.

Also, as in the pre-transcriptional case, the functions replacing the competitive and uncompetitive terms are bounded and monotone increasing. While the double saturation term in the competitive function seems somewhat over-complicated, it can give us a sense of what properties we might expect from a model such as this. But in the final analysis, the message of the derivation should be that double inhibition leads to a saturating allosteric inhibition.

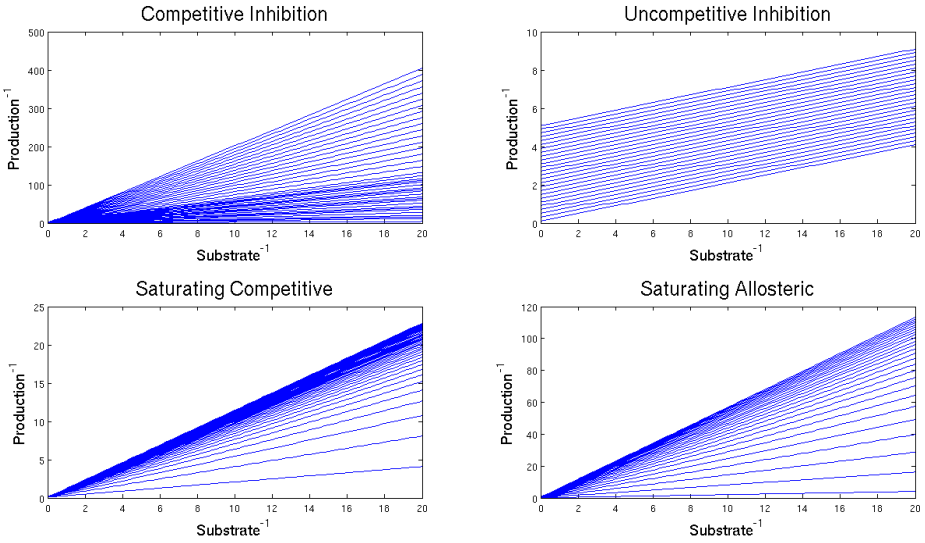

Figure 2: Graphs of Lineweaver-Burk Equations:

Increasing levels of inhibitor demonstrate the outcome in four different inhibition regimes: competitive and uncompetitive (traditional), and saturating competitive and saturating allosteric (proposed).

In Figure 2, we compare competitive and uncompetitive saturation, with the proposed saturating competitive and saturating allosteric inhibition. In each graph, the concentration of inhibitor is discretely increased. We can see immediately from the double reciprocal graph that competitive and uncompetitive increase in their saturation without bounds, while the saturating functions are distinctly bounded. As expected, saturating allosteric inhibits to a greater magnitude than the saturating competitive. Moreover, it is important to note from the graphs, that for fixed inhibitor, the double reciprocal graph is linear, indicating a function of the form of Michaelis-Menten. While the proposed functions for inhibition on the multi-scale level between

the cytokine and cell dynamics are a model, we find that by replacing traditional terms with bounded monotone functions produces observable, and potentially measurable, bound inhibition.

The equations (0.4) and (0.5), written with L and T respectively, can be interpreted as an open-loop regulatory system with T and L as inputs and their production as outputs, which was made into a closed system by assuming that the newly produced T and L act on the input of the system, thereby closing the loop. We shall make use of this point of view in later chapters, when we replace the variable T in the production term of each of the equation in by the summation Σ of all pro-inflammatory signals. For now, we can rewrite the system as:

$$\mathbf{X} = \begin{pmatrix} T \\ L \end{pmatrix}, \qquad \dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}) - \boldsymbol{\mu}\mathbf{X} \tag{0.6}$$

F(X) is the response function of the intracellular regulatory system. Since T is the activator of the system, an increase in T leads to an increase in **F**. Similarly, as L represents an inhibitor, an increase in L leads to a decrease in **F**. We have verified that for all sets of positive parameter values in (0.6) derived from system (0.2) that the partial derivatives act as desired. The derivations and parameters can be found in Appendix A.1.

$$\frac{\partial F_i}{\partial T}(\mathbf{X}) > 0, \text{ and } \frac{\partial F_i}{\partial L}(\mathbf{X}) < 0, \text{ for i=1, 2.} \tag{0.7}$$

Taking the limits of the inhibitor in (0.6) gives more precise insight into equation (0.8) and Figure 2. Superscripts are used to differentiate parameters where appropriate.

$$\lim_{L\to 0}\mathbf{F}(\mathbf{X}) = \begin{pmatrix} \dfrac{V_{max}^{(t)}T}{1+{K_{12}}/{K_{22}}\;\; T+V_s^{(t)}\cdot {J_{12}}/{J_{22}}} \\ \dfrac{V_{max}^{(l)}T}{V_s^{(l)}+T} \end{pmatrix}, \quad \lim_{L\to\infty}\mathbf{F}(\mathbf{X}) = \begin{pmatrix} \dfrac{V_{max}^{(t)}T}{(1+K_{11})(T+V_s^{(t)})} \\ \dfrac{V_{max}^{(l)}T}{\left(V_s^{(l)}+{K_{max}}/{K_s}\right)+T} \end{pmatrix} \tag{0.8}$$

It can be seen from the derivation in Appendix A.1 that the following relationships are true:

$$\begin{aligned} K_{12}/K_{22} &< K_{11} \\ J_{12}/J_{22} &< 1 \end{aligned} \tag{0.9}$$

A priori, for both components, the output of **F(X)** as L approaches zero should be greater than the output as L approaches infinity. This is guaranteed for a parameter set strictly derived from the original system; for a naïve parameter set, guaranteeing that the relationship in (0.9) holds implies that (0.7) will be true, from the algebraic representation of the partial derivatives, not shown here.

We see in equation (0.8) that **F**(X) is always bounded above and below by Michaelis-Menten expressions. Moreover, we see that inhibition can never be complete; inspecting the case of L tending toward infinity, we see that the system will not tend towards zero production for T>0. This confirms the saturation effect seen in the double reciprocal plots in Figure 2, that all saturating functions are bounded between the two extreme cases. We see from the graph that beyond the extremal cases, any fixed level of anti-inflammatory also gives Michaelis-Menten kinetics. For standard forms of inhibition, such as competitive, uncompetitive, or Hill-type, there is no upper bound on the double-reciprocal graph, implying that large amounts of inhibition drive the system arbitrarily close to zero. Hence, the proposed inhibition exhibits a desirable property that we do not replicate by replacing it with a more standard inhibition type listed above.

Furthermore, the proposed equations are ultimately intuitive to Michaelis-Menten kinetics, just as with the standard types of inhibition. From the limiting formulae, we confirm a parameter set for when the maximal production rate for the inhibited case is less than the rate for

the uninhibited case. We also determine the level of pro-inflammatory substrate necessary to achieve the half maximal reaction rate (Lehninger, Nelson, and Cox) and can find when the half-reaction constant for the inhibited case is greater than the uninhibited case. We also have closed forms, in both formulae, for the maximal production and substrate affinity of uninhibited and inhibited systems, that we can approximate from physical experiments or conversely propose for a given parameter set.

These dynamics were developed with the pro-inflammatory cytokine TNF-α and the anti-inflammatory cytokine IL-10, in mind. In the next sections, we shall explore the two dimensional dynamics of these pro- and anti-inflammatory cytokines, first for a single cell and then for multiple cells.

2.1.3 Dynamics of Saturation Kinetics

We now consider a static immune cell population of size M that reacts only to TNF-α (T) and IL-10 (L). We assume that T is inhibited both pre- and post- transcriptionally, and that L is inhibited only pre- transcriptionally, giving the two dimensional system:

$$\begin{aligned} T' &= \frac{b_t T}{T + \left[T + \frac{g_1 L + g_2}{L + d_2}\right]\left[\frac{k_1 L + k_2}{L + d_1}\right]} M - \mu_t T \\ L' &= \frac{b_l T}{T + \frac{g_1 L + g_2}{L + d_2}} M - \mu_l L \end{aligned} \tag{0.10}$$

In the biology, the desired reaction of this system to small stimuli is a rise in all species followed by a peak, then return to baseline; this is referred to as excitability. Biologically, excitability is desirable so that systems of this nature can respond to a short-term threat such as a

pathogen or irritant and return to the baseline state without causing damage to the medium in which this system functions. For this system to be excitable, we want to impose for initial value of TNF, T_0, that the derivative of T with respect to time zero be positive. Note that here we are taking the derivative with respect to time, not the partial of the function with respect to the variable as in (0.7).

$$\left.\frac{dT}{dt}\right|_{(T,L)=(T_0,0)} > 0 \tag{0.11}$$

For this system to be excitable, we want there to be some threshold value of T so that $T_0 > T_{threshold} > 0$ makes (0.11) true. However, this condition is best evaluated in terms of the number of immune cells, M. It is equivalent to the statement

$$M > \frac{\mu_t}{b_t}\left(T_0 + \left[T_0 + \frac{g_2}{d_2}\right]\left[\frac{k_2}{d_1 + k_2}\right]\right) \tag{0.12}$$

Hence, as we introduce higher levels of TNF, the system requires a larger ambient number of immune cells to see an excitable response to an initial non-zero level of TNF. Otherwise, T will decrease monotonically to its stable fixed point. This seems un-intuitive at first, as many system dynamics might cause a larger (or attracting) spike as more stimulus is introduced. However, from the quasi-steady state assumption applied to the system, limited receptor numbers mean that potential input to a fixed number of immune cells saturates, and so output saturates. Hence, this result is consistent with the assumptions we made upon the system. And for high levels of TNF, substrate decays faster than new substrate is produced, causing the system to return to steady-state.

Next, the system is investigated for stability. The system is formulated so that (0,0) is always an equilibrium point of the system. Linearization of the system yields that one eigenvalue that is always negative, $\lambda_1=-\mu_l$, and the other eigenvalue positive or negative dependent on parameters. Thus stability of the system depends on the condition:

$$M < \frac{\mu_t}{b_t}\left[\frac{g_2}{d_2}\right]\left[\frac{k_2}{d_1+k_2}\right] \tag{0.13}$$

But, this condition contradicts (0.12) for every value $T_0>0$, and is (0,0) is stable for fixed M, then TNF will not spike. Further, as M increases, the stability of (0,0) changes from stable to unstable when equality holds. Numerically, we can see that this is transcritical bifurcation, and as M increases the upper increasing branch stabilizes.

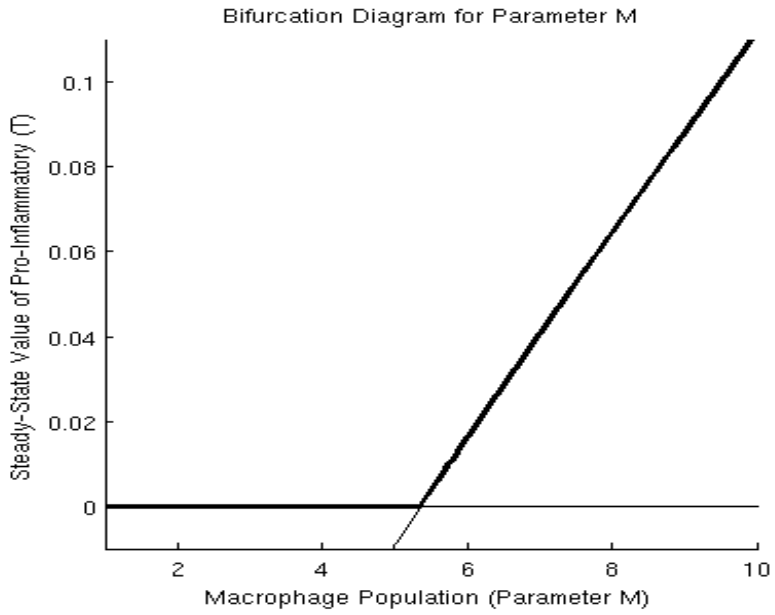

Figure 3. Bifurcation Diagram for Static Macrophage Population:

As the parameter M grows larger, the quiescent state loses stability in a transcritical bifurcation, and an elevated branch becomes stable.

The system, then, will only have an excitable response for a dynamic number of immune cells that can cross above this threshold temporarily and then decrease to below threshold. In the

biology, when there is an inflammatory stimulus, immune cell numbers increase, consistent with this model. We explore this further in the following sections.

2.2 MACROPHAGE, TNF-A, IL-10, AND CHEMOKINES

In this section, we consider a population M of macrophages all producing and sensing cytokines in accord with the autocrine dynamics. Extending this to the interaction of multiple macrophages, we admit a dynamical macrophage population M. To analyze effective inhibition in this model, we reduce the system to a three-variable model by treating L (anti-inflammatory cytokines) as a parameter; we explore the parameter space that corresponds to excitatory dynamics. We analyze the stability of the four-variable system and explore variations in key parameters, and conclude by examining the stability of the system in space.

2.2.1 Macrophage Production of Cytokines

The production of cytokines and other signalling molecules plays a central role in the inflammatory response of tissue to insult. Tissue macrophages, the cells that produce the bulk of these cytokines, are in turn influenced in an autocrine fashion by the cytokines they produce (Abbas and Lichtman;Janeway). These cells have surface receptors which initiate an intracellular signal to up-regulate or down-regulate production of a given effector molecule (Janeway). The goal of this effort is to develop and present an intuitive model of the activation and inhibition of cytokine-production, based on extracellular and intracellular signalling mechanisms.

The model assumes that the production of cytokines for a given macrophage depends on two types of signals: (i) a signal to down-regulate and (ii) a signal to up-regulate production. The signal to down-regulate, represented by the variable D, is conveyed through the IL-10 signalling pathway, and the signal to up-regulate, U, is conveyed through the NF-κB pathway (Meisel et al. 1580-86). The complexities of intracellular signalling for both pathways are simplified in this model, as represented in Figure 1, and written out:

$$\begin{aligned}
\frac{dD}{dt} &= \alpha L - \mu_1 D \\
\frac{dU}{dt} &= \beta T - \mu_2 U \\
\frac{dP_1}{dt} &= \frac{\gamma D}{D + \eta_1 P_1 + \nu_1} - \mu_3 P_1 \\
\frac{dP_2}{dt} &= \frac{\delta D}{D + \eta_2 P_1 + \nu_2} - \mu_4 P_2 \\
\frac{dR}{dt} &= \frac{\varepsilon U}{U + \eta_3 P_1 + \nu_3} - \mu_5 R \\
\frac{dT}{dt} &= \frac{b_t R}{R + \eta_4 P_2 + \nu_4} - \mu_t T \\
\frac{dL}{dt} &= \varphi R - \mu_l L \\
\frac{dC}{dt} &= \theta R - \mu_c C
\end{aligned} \tag{0.14}$$

The signal to down-regulate (D) is triggered when the extracellular anti-inflammatory cytokines (L) such as IL-10 bind to receptor proteins (Kontoyiannis et al. 3760-70;Mocellin et al. 36-43;Moore et al. 165-90;Moore et al. 683-765). The strength of the signal decays at a constant rate as bound receptors are internalized and degraded as part of the physiologic turnover process. The intracellular portion of the bound receptor initiates the IL-10 signalling pathway (Abbas and Lichtman;Janeway;Moore et al. 683-765). This pathway up-regulates the production of at least

two types of inhibitory proteins, denoted by P_1 and P_2. Protein P_1 acts to inhibit the pre-transcription signalling pathways for both D and U, by means of competitive inhibition in the production terms for P_1, P_2, and mRNA (R). Protein P_2 is a post-translational inhibitor, that acts competitively in the production term for pro-inflammatory cytokines (T) (Meisel et al. 1580-86).

The signal to up-regulate is triggered when extracellular pro-inflammatory cytokines (T) such as TNF-α or IL-1 bind to receptor proteins(Clark 335-43;Zhou et al. 945-53) . From here, intracellular signal results in the production of various types of mRNA, that will in turn be translated into the proteins which will become cytokines released outside the cell (Abbas and Lichtman), in the equations for T, L, and C. As mentioned above, this process is inhibited pre-transcriptionally by a protein product of the IL-10 pathway. For anti-inflammatory cytokines such as IL-10 or chemokines (C) such as RANTES, mRNA is fully translated, and proteins products are released outside the cell where they are consumed, and the mRNA decays. The production of pro-inflammatory cytokines from mRNA is, however, inhibited by P_2. This process thus impedes the production of inflammatory agents at two levels. This allows for greater potential to modulate the inhibition of the pro-inflammatory cytokines production compared to that of the anti-inflammatory cytokines. (Edelstein-Keshet;Keener and Sneyd;Murray).

Following the same quasi-steady state assumption followed in chapter 1, we derive differential equations in T, L, and C, shown in system (0.15).

2.2.2 Intercellular Signaling and Dynamics

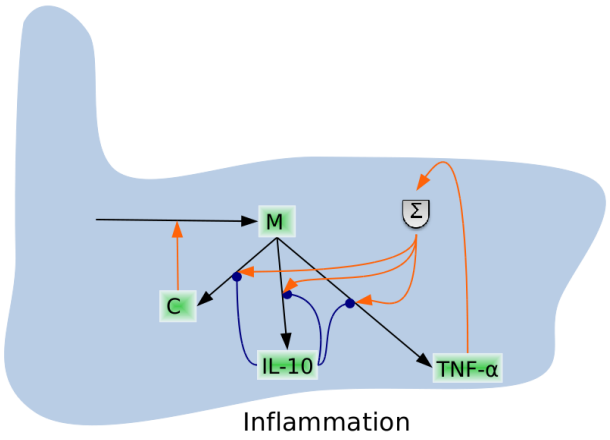

Figure 4. Intercellular Signalling:

The dynamics of macrophage mediated cytokine production and macrophage recruitment is modeled and schematically displayed.

$$\Sigma = a_1 T$$

$$M' = \frac{b_{mc} C^p}{a_{mc}^p + C^p} - \mu_m (M - b_m)$$

$$T' = \frac{b_t M \Sigma}{\Sigma + \left[\Sigma + \frac{g_1 L + g_2}{L + d_2}\right]\left[\frac{k_1 L + k_2}{L + f_0}\right]} - \mu_t T$$

$$\tau\, L' = \frac{b_l M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_l L$$

$$C' = \frac{b_c M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_c C \qquad (0.15)$$

The recruitment of macrophages from the blood occurs due to chemokine signalling, and derecruitment due to deactivation and homeostatic forces (Fernandez and Lolis 469-99;Janeway;Murdoch and Finn 3032-43). The equation M is constructed so that there exists a homeostatic level of macrophages b_m: when the macrophage level rises above b_m, they are

derecruited; likewise, for M < b_m, they are recruited to tissue. The exact mechanisms that determine the set point b_m for a specific tissue is unclear. A rise in chemokine (C) concentration triggers an influx of macrophages into the system. The Hill term reflects the concentration gradient of chemokines necessary for inward migration to occur. A macrophage is derecruited from tissue at a rate provided by the parameter μ_m.

We assume the variable M represents both activated, cytokine producing macrophages, and inactive macrophages. As cytokine production by macrophages only occurs in the presence of non-zero pro-inflammatory effector (T), one can consider the macrophages in this model to be activated if they are in contact with a stimulating signal, e.g., T>0. We shall generalize the signal (Σ) to be any stimulus which causes macrophage production of cytokines to up-regulate. For the four-variable system, the signal term can be written as $\Sigma = a_1 T$, with any further sources of stimulation included additively (Reynolds et al. 220-36;Day et al. 237-56).

Activated macrophages are customarily divided into two populations, M_1 and M_2 (Abbas and Lichtman;Janeway). This designation differentiates between the early stage of activation when pro-inflammatory is more heavily produced, M_1, and the later stage when anti-inflammatory is more heavily produced, M_2. Both M_1 and M_2 are considered to be activated macrophages. This division of macrophages into two phases can also be thought of as a continuum of cytokine production, where anti-inflammatory cytokine levels remains near quiescence while pro-inflammatory launches, then anti-inflammatory launches as pro-inflammatory cytokine levels approach quiescence. We seek to encapsulate this phenomenon into the model by considering two time scales; pro-inflammatory cytokines are produced and decay in normal time, and anti-inflammatory cytokines are produced and decay in slow time. We accomplish this by multiplying L' by $\tau > 1$, making L a slow time variable.

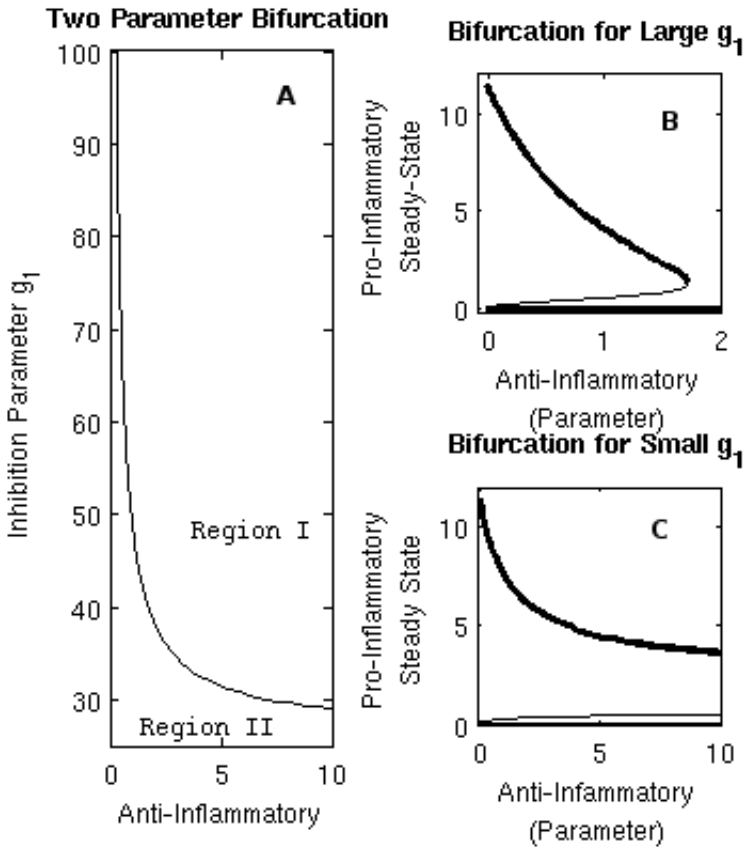

Figure 5. Bifurcation Diagrams for Fixed Levels of Anti-Inflammatory Effectors:

(A) Two parameter bifurcation diagram of Anti-Inflammatory against inhibition parameter g1. In Region I, there is one stable equilibrium point. In Region II, there are two stable equilibrium points. (B) For g1 large, the upper branch loses stability in a saddle-node bifurcation, and any trajectory will go to the lower branch for high anti-inflammatory effectors. (C) For g1 small, the upper branch remains stable, and there is no guarantee that a given trajectory will be in the basin of attraction for the lower branch.

In order to explore the slow-time dynamics of the system, we consider the *Fast-Slow Reduction* subsystem in which L'=0. Thus, we treat L as a parameter in the other three equations. For L=0, we choose a parameter set that results in a bistable system, with one stable fixed point corresponding to the rest state and a second stable fixed point corresponding to the excited state. Then, we compute the bifurcation diagram in L (Figure 5). Either the bistability persists (Figure 5C), or the excited branch loses stability in a saddle-node bifurcation (Figure

5B). Varying the inhibition parameter g_1, we see in the two parameter bifurcation diagram (Figure 5A) a line of saddle nodes dividing Region I and Region II. In Region I, there is only one stable attracting state, while in Region II there are two. Above a threshold value of g_1, varying L from 0 to 10 moves it from Region II into Region I, eliminating the bistability. Below that critical value of the inhibition parameter g_1 the bistable region persists for arbitrarily large values of inhibitor L (Perko;Kuznetsov). The one and two parameter bifurcation diagrams are computer numerically using AUTO (Doedel 265-84) with their trajectories being viewed in the frontend software, XPPAUT (Ermentrout).

We wish our parameter set to be in situation with the saddle-node bifurcation, so that when an immune response is initiated with anti-inflammatory effector concentration (L) near zero, the system can come into the basin of attraction for the excited arm, and as anti-inflammatory increases the excited arm loses stability so that the system becomes attracted to the rest state. This system would thus be excitable. Hence, by considering the dynamics of the *Fast-Slow Reduction* subsystem, we succeed in using the single variable M to describe the qualitative properties of a model that explicitly uses both phases M_1 and M_2.

2.2.3 Excitability and Stability of Four Variable Intercellular Model

In this section, we examine the stability and the properties of the four-variable system with no additional assumptions made about the dynamics of individual variables. We expect that it will behave as an excitable system if we choose the parameters to be the same as for the *Fast-Slow Reduction* subsystem, and τ sufficiently large. Figure 6 shows a trajectory of the system in the (T,L) plane when given an initial stimulus of pro-inflammatory effector(T). As desired

from the stimulus, T peaks first, followed by a peak in L. With no further stimulus, the system proceeds to the quiescent state. This numerically confirms excitability, as we expect analytically.

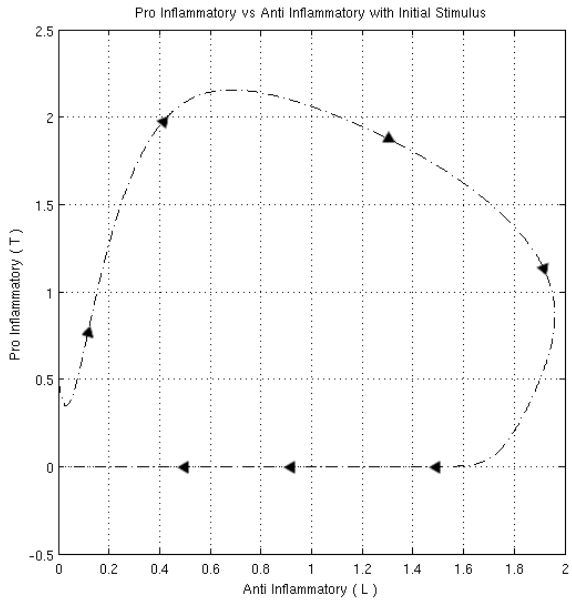

Figure 6. Regulation of Pro by Anti-Inflammatory Effectors:

For the four-variable system (3.4), an initial pro-Inflammatory signal is given. For each time point, Pro-inflammatory effectors are graphed against anti-inflammatory with respect to time. As anti-inflammatory effectors rise, pro-inflammatory effectors fall until they reaches zero and the graph approaches the origin.

Starting from steady-state in all variables except the initial pro-inflammatory effector, which has initial value one, the level of pro-inflammatory effector peaks within 5 hours and then decays (Figure 7). There is an initial downward slope, caused by a macrophage population insufficient to sustain pro-inflammatory growth as was present in the two-dimensional system. However, the macrophage population increases from rest, and once it reaches threshold, the derivative of the graph for pro-inflammatory changes sign. While the pro-inflammatory effector and chemokines peak at about the same time, the macrophage population peaks later at about 10 hours, and finally the anti-inflammatory effector peaks at about 15 hours. The production of pro-

inflammatory effector and chemokines are inhibited by the anti-inflammatory signal, which continues to grow after the pro-inflammatory signal has halved from its peak. With chemokine production inhibited, the induction of macrophages is eventually overtaken by derecruitment. Persistent levels of anti-inflammatory cytokines past its peak continue to inhibit cytokine production and allow the macrophage population to fall below threshold and return to rest at which point the anti-inflammatory also returns to rest. We note that since this experiment included only an initial signal rather than a persistent external signal, e.g. a bacterial population, cell damage, etc, we expect the system to return to the rest state. However, varying parameters sufficiently, the behavior bifurcates, and an initial stimulus can reset the system into a state of chronic inflammation.

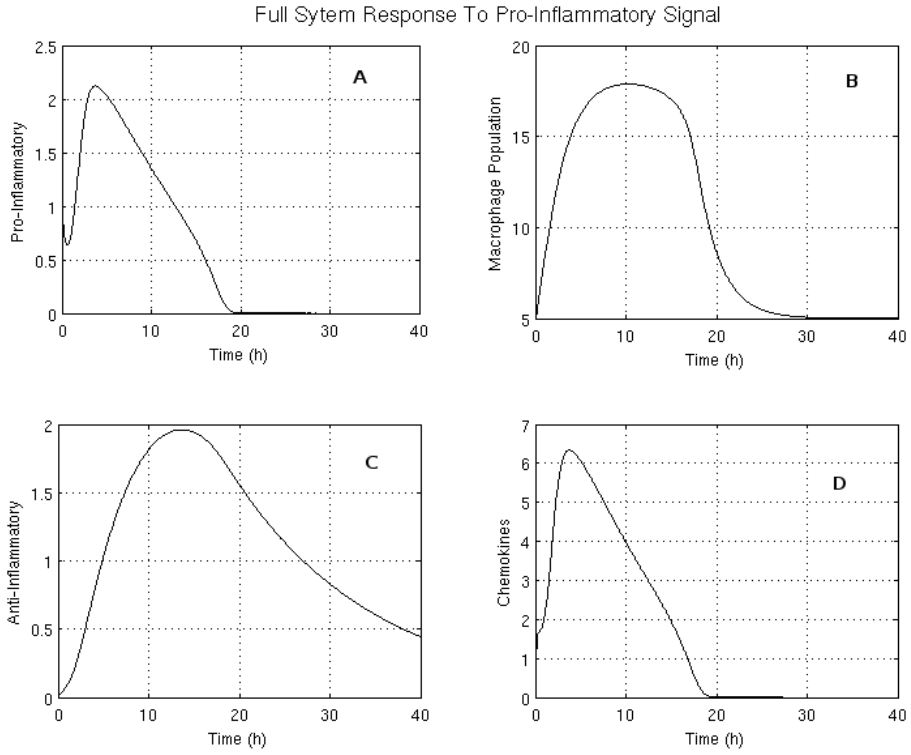

Figure 7. Inflammatory Response to Exogenous Pro-Inflammatory Signal:

For the four-variable system, an initial pro-inflammatory signal T(0)=1 is given at time zero, and the four time courses (A) pro-inflammatory effectors, (B) macrophages, (C) anti-inflammatory effectors, and (D) chemokines each spike before going to rest.

Parameter	Significance	Value
a_1	Signalling parameter for pro-inflammatory effector	1.0
a_2	Signalling parameter for dead tissue	0.7
g_1	Inhibition parameter for signalling	40.0 conc
g_2	Inhibition parameter for signalling	10.3 $conc^2$
d_2	Inhibition parameter for signalling	1.0 conc
b_{mc}	Rate of macrophage recruitment by chemokines	5.5 conc $time^{-1}$
a_{mc}	Michaelis-Menten parameter for macrophage recruitment	1.0 $conc^2$
p	Hill-coefficient for macrophage recruitment	2.0
μ_m	Rate of macrophage derecruitment	0.4 $time^{-1}$
b_m	Equilibrium value of macrophage population	5.0 conc
b_t	Rate of pro-inflammatory effector production	10.0 $time^{-1}$
μ_t	Rate of pro-inflammatory effector decay	6.5 $time^{-1}$
k_1	Inhibition parameter for pro-inflammatory effector	0.95 conc
k_2	Inhibition parameter for pro-inflammatory effector	2.0 $conc^2$
f_0	Inhibition parameter for pro-inflammatory effector	2.5 conc
τ	Time parameter for anti-inflammatory effector	80.0
b_l	Rate of anti-inflammatory production effector	20.0 $time^{-1}$
μ_l	Rate of anti-inflammatory effector decay	5.0 $time^{-1}$
b_c	Rate of chemokine production	40.0 $time^{-1}$
μ_c	Rate of chemokine decay	8.0 $time^{-1}$

Table 1. Parameters for Macrophage Model

While the equations for the anti-inflammatory effector and chemokines share similar trajectories, the time re-scaling of the dynamics of the anti-inflammatory effector results in different behaviors. The profile of the chemokines behaves more like that of the pro-inflammatory effector, despite different inhibition mechanisms.

The unperturbed state where cytokine production is zero we refer to as the quiescent state. The quiescent state (M, T, L, C) = (b_m, 0, 0, 0) is stable. To prove stability, we linearize about the fixed point to get the matrix (0.16) and characteristic polynomial (0.17):

$$\begin{bmatrix} -\mu_m & 0 & 0 & \chi \\ 0 & a_1 b_t b_m \dfrac{d_2}{g_2}\dfrac{f_0}{k_2} - \mu_t & 0 & 0 \\ 0 & \dfrac{b_m b_l}{\tau}\dfrac{d_2}{g_2} & -\mu_l/\tau & 0 \\ 0 & b_m b_c \dfrac{d_2}{g_2} & 0 & -\mu_c \end{bmatrix} \tag{0.16}$$

$$(\lambda + \mu_m)(\lambda + \mu_c)(\tau\lambda + \mu_l)(\lambda + \mu_t - a_1 b_t b_m d_2 f_0/(g_2 k_2)) = 0 \tag{0.17}$$

Here χ depends on the value of the exponent p. The first three solutions of λ are negative. The fourth depends on parameter values, and for $b_m < a_1(\mu_t g_2 f_0)/(b_t d_2 k_2)$, this eigenvalue is negative. If we let $a_1 = 1$ then this is the same condition as the (T, L) subsystem. Thus, this system combines the stability condition derived from the (T, L) subsystem with the excitability derived from the *Fast-Slow Reduction* subsystem. We additionally see from Figure 8, that this fixed point is locally attracting.

2.2.4 Pro-Inflammatory Production and Bifurcation Analysis

Next, we examine the effect of varying b_t and b_m on the fixed points, which is captured by the bifurcation diagrams in Figure 8 and Figure 9, respectively, obtained by using the AUTO continuation software (Doedel 265-84). These parameters come as natural choices to vary, given their role in the stability described above. Furthermore, altering their values corresponds to disorders in immune-regulation: varying b_t corresponds to varying production of protein from mRNA, and varying b_m to varying the steady-state population of macrophages in tissue.

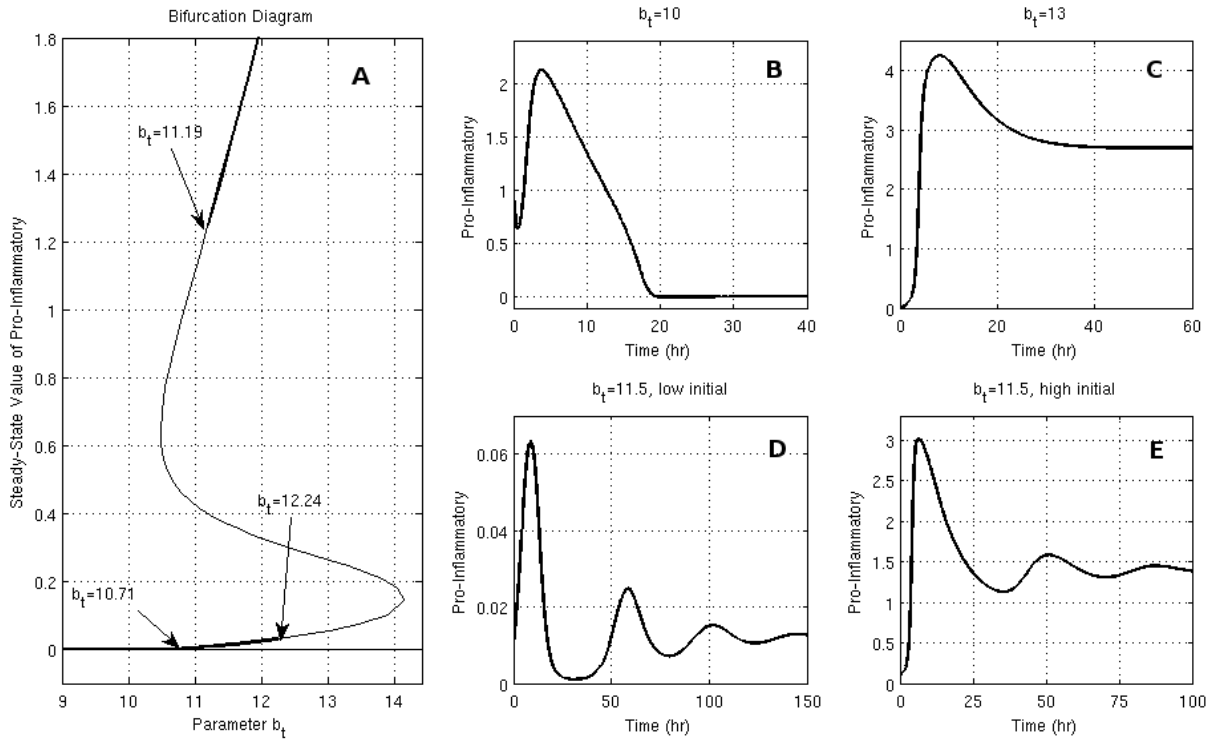

Figure 8. Analysis for Behavior of Parameter for Pro-Inflammatory Production:

(A) Bifurcation diagram of the system for b_t. . Trajectory of pro-inflammatory effector for selected values of b_t: (B) b_t is set less than the transcritical value, and the system goes to rest. (C) b_t is set after the Hopf bifurcation value, and the system becomes elevated. (D) b_t is set in the bistable region and given a small amount of pro-inflammatory effector T(0)=0.01. (E) b_t is set in the bistable region and given a large amount of pro-inflammatory effector T(0)=0.1.

As in Figure 3, the quiescent steady state is lost in a transcritical bifurcation in both Figures 8 andFigure 9; however, unlike Figure 3, stability of the upper branch, the active state, is lost in an Andronov-Hopf (AH) bifurcation, creating unstable periodic orbits and a branch of saddle-nodes, which folds over twice and becomes once again stable in another AH bifurcation. For b_t (Figure 8), this creates a bi-stable region with the lower stable state close to quiescence and a chronically inflamed state. The chronically inflamed state is a stable fixed point where the macrophage population is elevated, and production of all cytokines continues indefinitely. As

the parameter b_t increases, the basin of attraction for the lower state branch shrinks and it becomes more likely the system will jump towards the higher state. The bi-stability is lost at the lower AH bifurcation, and all trajectories tend towards the chronically inflamed state.

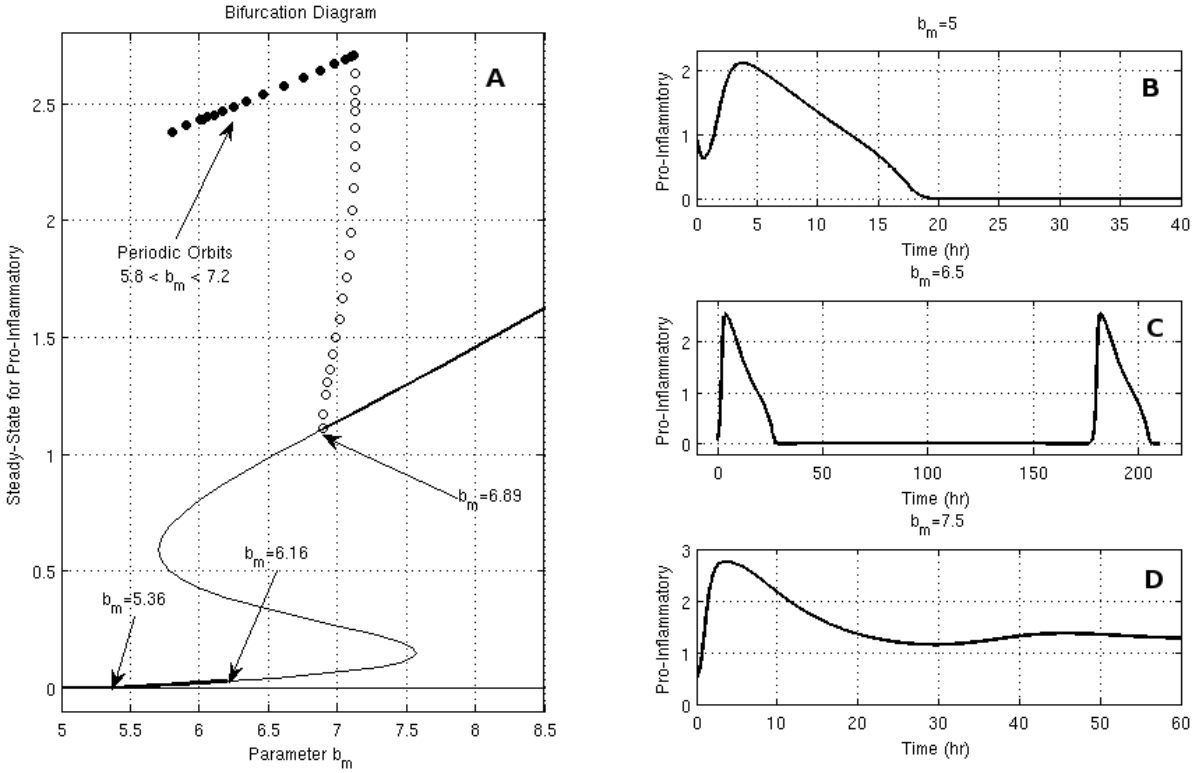

Figure 9. Analysis for Behavior of Parameter for Macrophage Population:

(A) Bifurcation diagram of the system for b_m. Trajectory of pro-inflammatory effector for selected values of b_t: (B) b_m is set less than the transcritcal value, and the system spikes then goes to rest. (C) b_m is set between the value of the two Hopf birfurcation, and the system stabilizes in a periodic orbit. (D) b_m is set greater than the second Hopf bifurcation, and the system becomes elevated given an initial stimulus.

For b_m (Figure 9), rather than a bistable region there exists a region with no stable fixed point, rather a periodic attractor that lies along the quiescent state for the majority of its trajectory and producing periodic peaks in all variables. The bifurcation diagrams for b_t and b_m, despite similar structures, reveal qualitatively different behavior that reflects their different roles in the biology they model. Increasing the production rate of pro-inflammatory effectors (Figure

8) very quickly produces serious and unabated inflammation. While in the bifurcation diagram, we see a marked jump from the lower state to the higher, state, comparing the cases where b_t is 10, 11.5, and 13 we see a continuum of initial peaks given an initial insult. However, as b_t rises, those peaks lose the ability to return to a quiescent (or near quiescent) equilibrium. In the course of a given disease, a mechanism to boost the production rate of pro-inflammatory effectors would cause significant damage; a modeled intervention would need to address production as well as remove excess cytokines from the tissue.

In the model, we assume self-regulation of macrophages to a steady-state population at quiescence. By varying the parameter b_m, we see how the model treats a scenario where a disorder raises the number of ambient macrophages floating in the system at rest. Change in the steady-state value is more gradual, but is interrupted by an intermediate state with stable periodic orbits (Figure 9). The periodic spiking arises from a bifurcation in the unstable periodic orbits produced from the lower AH bifurcation, and likewise die in a collision with the unstable periodic orbits produced from the higher AH bifurcation. It is unclear if there is biological significance to the periodic peaks of pro-inflammatory effectors, or if it is an analytic anomaly in the model, but the phenomenon of oscillation in production of NF-κB, an up-regulatory mechanism for pro-inflammatory effector, has been observed experimentally (Covert et al. 1854-57). As the parameter b_m increases, the period between spikes shortens and the height of the spikes rises until the stability of the oscillations is lost and the system becomes attracted to an elevated steady state of pro-inflammatory.

2.2.5 Model in Space, Transcritical Bifurcation and Pattern Formation.

The ODE that we examine in the bifurcation analysis, by their nature, assume a well-mixed property. This homogeneity disappears when we examine the equations and their bifurcation structure in a spatial model. To the equations (0.15), we add diffusion across space, and a chemotactic gradient in the macrophage equation. By engaging in a PDE variant of the model, we wish to see what potential behaviors that dysregulated systems, such as shown in Figure 8, present spatially.

$$
\begin{aligned}
&\Sigma = a_1 T \\
&\frac{\partial M}{\partial t} = -\mu_M (M - b_m) + D_m \frac{\partial^2 M}{\partial x^2} - \chi \frac{\partial}{\partial x}(M \frac{\partial C}{\partial x}) \\
&\frac{\partial T}{\partial t} = \frac{b_t M \Sigma}{\Sigma + (\Sigma + \frac{g_1 L + g_2}{L + d_2})(\frac{k_1 L + k_2}{L + f_1})} - \mu_l T + D_t \frac{\partial^2 T}{\partial x^2} \\
&\frac{\partial L}{\partial t} = \frac{b_l M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_l L + D_l \frac{\partial^2 L}{\partial x^2} \\
&\frac{\partial C}{\partial t} = \frac{b_c M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_c C + D_c \frac{\partial^2 C}{\partial x^2}
\end{aligned}
\tag{0.18}
$$

For the original parameter values, the quiescent state (M_0, 0, 0, 0) is asymptotically stable. However, when the parameter for macrophage levels, b_m, is raised, the quiescent state loses stability in a transcritical bifurcation. This new elevated branch stays close to the quiescent branch. Analytically, we want to be able to predict when a pattern forming instability will arise. Looking at the linearization matrix (0.16), we can expect for the quiescent state, that the chemokine gradient will not affect the stability, and that we will expect uniform stability.

However, for the elevated branch, many of the zero terms in the matrix become non-zero, and the chemokine gradient will make an impact on the over-all stability of the system.

Following the common procedure to find pattern forming instabilities in space(Murray), we linearize about the fixed point using the perturbation $X^i = X^i_0 + \varepsilon X^i_1 \exp(\lambda t + ikx)$ so we can look and look at the eigenvalues λ of the first order ε terms which are given by the linearized matrix. Unlike the ODE system, we will consider the eigenvalues as a function of the wave number k. The off-diagonal chemotactic term in the diffusion matrix is what makes the pattern forming possible, as it is positive in the linearized matrix and all the other diffusion term make the diagonal terms more negative. As the determinant is the product of the eigenvalues, we plot the determinant in Figure 10 as a function of k, to find for which values of k the system has a positive eigenvalue.

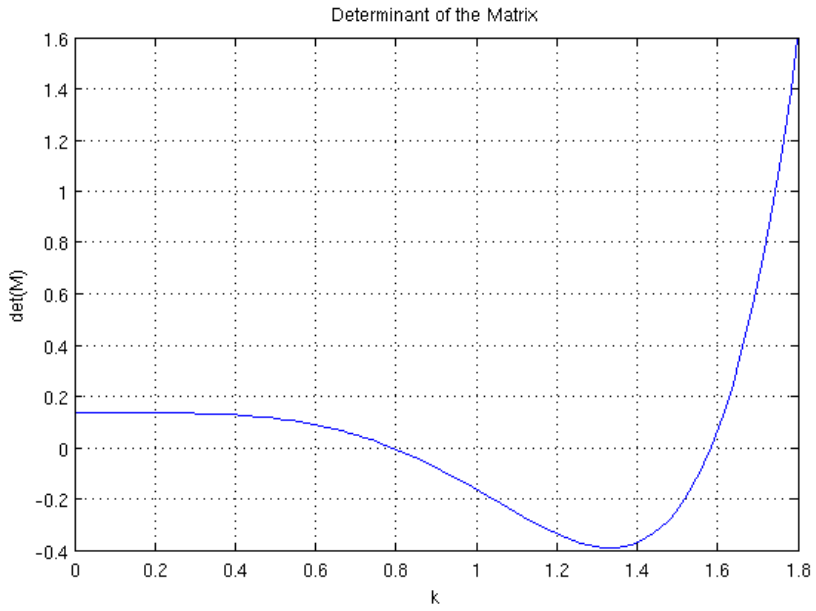

Figure 10. Determinant of the Linearized Spatial System:

The determinant of the linearized matrix of the elevated branch is plotted as a function of the wave number k, taken from the linear perturbation. For k in [0.8, 1.6], the determinant changes signs, implying a positive eigenvalue for those values of k.

The determinant as a function of k predicts a pattern forming wave instability with wave number k=1. The determinant changes signs at k=.8 and k=1.6. As we see in Figure 11, these changes in sign are the result of a single eigenvalue first becoming positive and then becoming negative as k increases. Hence, the system is linearly stable for all other wave numbers.

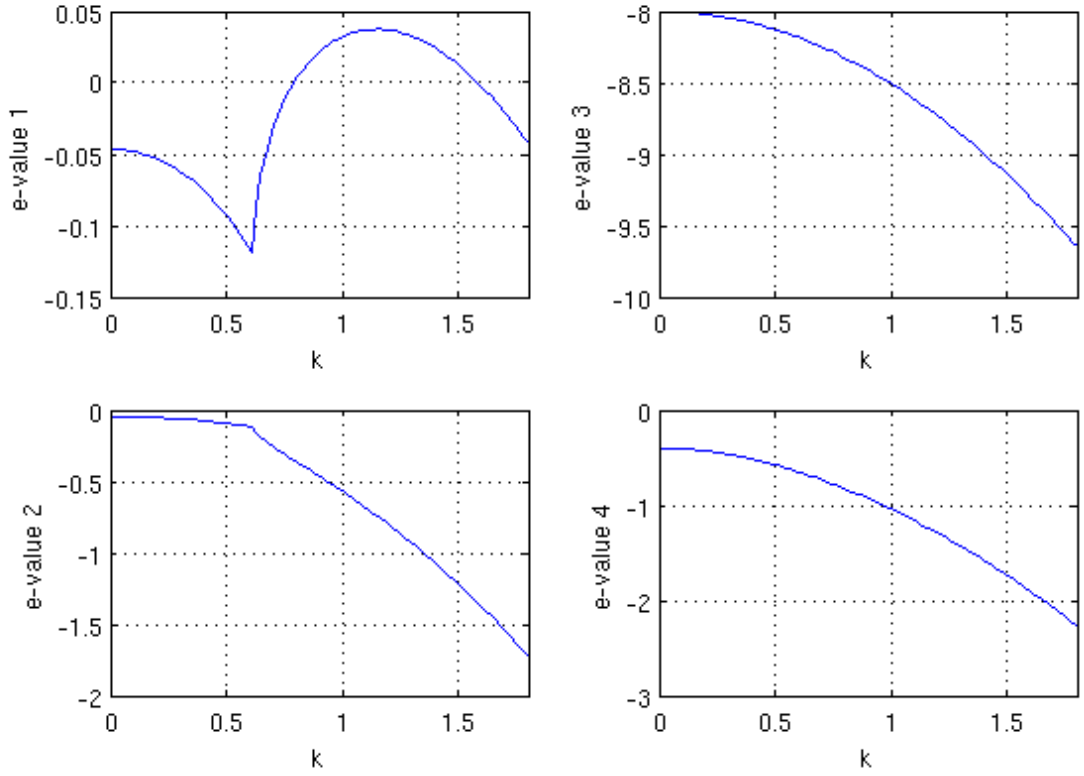

Figure 11. Eigenvalues of Linearized Spatial System:

The individual eigenvalues of the system all start negative. However, eigenvalues 1 and 2 are conjugates of each other until their imaginary parts become zero at the bifurcation point near k=0.6, and their real parts diverge.

All eigenvalues start negative and end negative. The first eigenvalue is the only one to change signs, confirming the pattern forming instability is the result of a single eigenvalue. Also, eigenvalue 1 and eigenvalue 2 are conjugates for k in [0, .6), so the real parts are the same; the real parts diverge when the imaginary parts go to zero.

The values of the diffusion terms and the chemotaxis term were chosen to illustrate the capacity of the model to produce a pattern forming instability and do not necessarily reflect biologically relevant values. The pattern forming instability is driven by strong macrophage chemotaxis.

In one-dimensional space and time, the pattern in space remains stationary rather than move across space as seen in Figure 12. The PDE is numerically approximated using a finite difference method (Hall and Porsching), and executed in the XPPAUT integration environment. The space-time graphs for the pro-inflammatory and anti-inflammatory cytokines are shown. Even though the stripes for the variable L are thicker, they are centered at the same position as the variable T. These locations are the clusters of production of cytokines. The IL-10 anti-inflammatory cytokine, for this parameter choice, is much more diffuse than the pro-inflammatory cytokine, TNF-α. This containing effect of a more diffuse IL-10 and a localized TNF- α may be one component to the stability of this pattern, in addition to strong chemotaxis that allows macrophages to cluster. Alternately, the containing effect of IL-10 may be biologically relevant to the formation of self-contained areas of damage such as granulomata, rather than a region of damage that expands without boundaries.

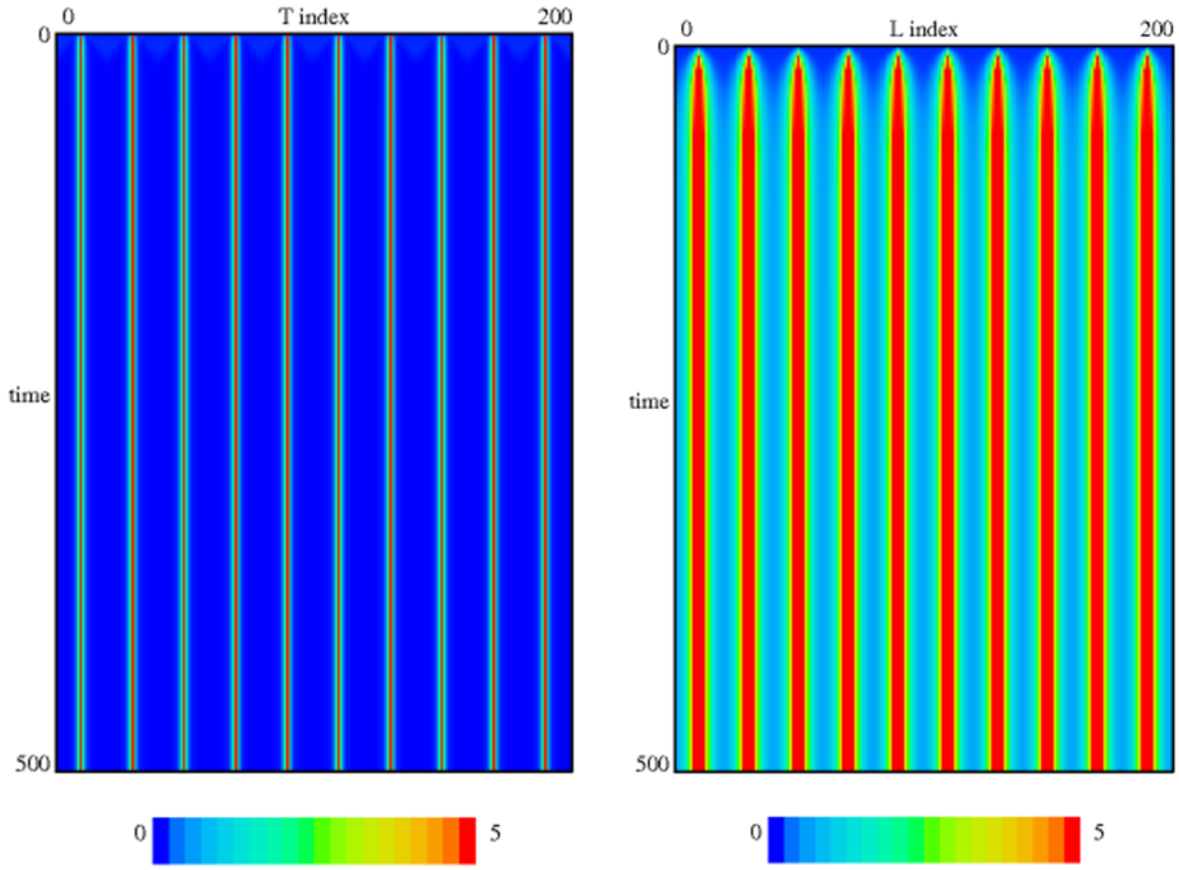

Figure 12. Patterns forming across space:

Even though the indices of the stripes match, the anti-inflammatory effectors (right) have thicker stripes then the pro-inflammatory effectors (left) and more concentration between stripes. The diffusion parameters for them are identical.

A non-intuitive difference between this model and the ODE model, that we see a posteriori, is that this model has a chemotactic effect on the immediate area, whereas the ODE model has a chemotactic effect designed to act on a far-reaching area, such as from nearby arteries. This leads to the PDE producing an effect where macrophages deplete entirely from certain regions, and cluster in others, i.e. there is no net change in the number of macrophages, only re-positioning. This relevant to certain biological circumstances and short-term disease

courses; however, it is notable that this assumption departs from that of the model presented above.

2.2.6 Discussion and Conclusions

This excitable system demonstrates a variety of responses, and a range of behaviors for a sufficiently altered parameter set. It has many desirable qualities such as excitability, stability of the fixed point, and the necessity of macrophage quorums to produce cytokines en masse. It also has the important observable that IL-10 peaks later and longer, despite a simplification in the treatment of macrophage activation.

As often occurs in the process of modeling biological systems, several practical abstractions and simplifications were made; however, we take care that these simplifications have not detracted from the overall validity of the model. The first such example is that while there exist dozens of cytokines and related proteins, we present only three classes of cytokines: pro-inflammatory cytokines, anti-inflammatory cytokines, and chemokines. The division into pro-inflammatory and anti-inflammatory cytokines is common, and we generalize this modelling process by including the chemokine classification. Further complicating the modelling, not all cytokines can be universally classified as pro-inflammatory or anti-inflammatory; often cytokines have both properties and this classification depends on the context in which we consider a given cytokine. Using this system, we can account for the basic behavior. It also allows us to proffer a conceptual model, although the full dynamics of intracellular and intercellular signalling are still being researched. It also allows us to keep the model intellectually and computationally tractable.

This model on a high level resembles several other models of inflammation, including Reynolds and Day (Reynolds et al. 220-36;Reynolds et al. 220-36). The difference comes in the conceptual construction of the model, with a different approach to mechanism. The dynamics of pro- and anti-inflammatory production and effects on immune cells are derived through a variant of chemical reaction kinetics adapted for the multi-scale nature of the interactions. The dynamics arising from this model also allow for auto-inflammatory disorders which are of increasing interest.

Finally, we consider the drawbacks to not explicitly including the process of macrophage activation; activated macrophages have distinct difference from naïve ones such as longer life and increased activity. Since we assume a large source of naïve macrophages that only activate, we consider them to be macrophages extent when there is no signal. Because macrophages have no effect on the system when there is no signal, in practical terms we can disregard their differences from activated macrophages.

At the end of the day, the purpose of the system is to serve as a subsystem to a larger environment, where these properties of the subsystem are important.

2.3 INFLAMMATION MODEL

The acute inflammatory response affects all tissues and plays a protective role in the early systemic response to stimuli such as pathogenic bacteria, foreign cells, and particulates. Acute inflammation also mediates more complex forms of immunity and defends compromised tissue from secondary infections. When dysregulated, acute inflammation may also seriously impede tissue healing and may contribute to adverse outcomes such as organ failure and death (Takala et

al. 529-38). Thus, a synthesis of the various interactions of the acute inflammatory response aids in understanding how various effectors interact and affect tissue (Clermont et al. 2061-70;Chow et al. 74-84;Reynolds et al. 220-36).

A basic, yet robust, inflammation model is key to model more complex immune reactions, such as acute lung injury and influenza A virus infection (Goodman et al. 523-35). Inflammation acts to stimulate cellular immune effectors such as dendritic cells, macrophages, natural killer cells, and neutrophils (Janeway). However, much like the case for severe bacterial infections, excessive inflammation may also result in undesirable collateral damage such as rapidly progressive organ failure and death; therefore, proper modeling of the inflammatory response and intervention will help the understanding of disease course and treatment.

Several models have successively improved on the understanding of acute inflammation and sepsis (Day et al. 237-56;Reynolds et al. 220-36;Daun et al. 843-53;Kumar et al. 145-55). These models primarily consider intercellular signalling between general classes of immune effectors, as well as a specific source of pathogen. The model presented here builds on past models but is derived from biological interactions and focuses on molecular signalling both inside and outside of the cell as the main source of dynamics. The model takes into account an expanded set of effectors and cell types. In addition, a novel approach to inhibition by anti-inflammatory effectors is derived and incorporated into the model.

To model the inflammation in the context of tissue, we append to the system in Chapter 2 equations describing the dynamics of neutrophils, neutrophil product (such as iNOS), and tissue. We then explore the dynamic response to signal initiated both by damage and by exogenous signal, such as LPS. Finally, we observe that the system is bistable between the quiescent state and the damaged state.

2.3.1 Equations for Tissue

To put the role of the above described inflammation in the context of tissue, we append four new equations in (0.19) that describe tissue and other immune factors in relation to the already described four variable cytokines production compartment. The four equations represent neutrophils in blood $(\hat{N})$, neutrophils in tissue (N), neutrophil produced tissue toxin (X), and tissue health (H). Previous models used neutrophils rather than macrophages as the main source of pro-inflammatory (Reynolds et al. 220-36;Day et al. 237-56;Kumar et al. 145-55); however, macrophages produce pro-inflammatory in several orders of magnitude greater than neutrophils (Janeway). In accord with previous models, though, neutrophils inflict damage to tissue, here via production of proteins such as elastase and iNOS , a key enzyme associated with free radical production (Abbas and Lichtman;Janeway). Damage to tissue then feeds back signaling the up-regulation of cytokine production (Prince et al. 407-17).

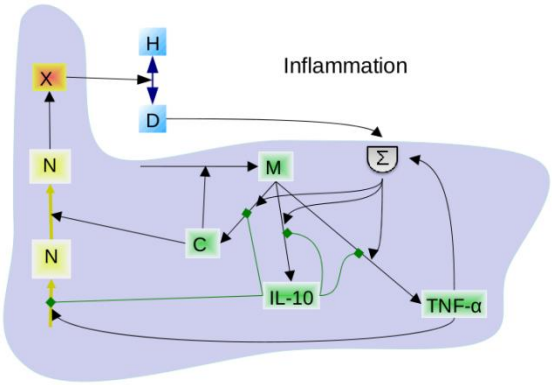

Figure 13. Inflammation Schematic:

The system of inflammation equations and relationships are displayed schematically, and show the potential for feedback from epithelial cell damage to drive inflammatory damage.

$$
\begin{aligned}
\Sigma &:= a_1 T + a_2 (H_0 - H) \\
\frac{dM}{dt} &= \frac{b_{mc} C^p}{a_{mc}^p + C^p} - \mu_m (M - b_m) \\
\frac{dT}{dt} &= \frac{b_t M \Sigma}{\Sigma + \left[\Sigma + \frac{g_1 L + g_2}{L + d_2} \right] \left[\frac{k_1 L + k_2}{L + f_0} \right]} - \mu_t T \\
\tau \frac{dL}{dt} &= \frac{b_l M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_l L \\
\frac{dC}{dt} &= \frac{b_c M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_c C \\
\frac{d\hat{N}}{dt} &= \frac{b_{nt} T}{a_{nt} + a_{nl} L + T} - \frac{r_{nc} NC}{a_{nc} + C} - \mu_{\hat{n}} \hat{N} \\
\frac{dN}{dt} &= \frac{r_{nc} NC}{a_{nc} + C} - \mu_n N \\
\frac{dX}{dt} &= \frac{b_{xn} N}{a_{xn} + N} + b_{xm} M \Sigma - g_{xh} HX - \mu_x X \\
\frac{dH}{dt} &= b_h H (H_0 - H) \left(\frac{H - \theta}{H_0} \right) - \frac{g_{hx} H X^q}{a_{hx}^q + X^q}
\end{aligned}
\qquad (0.19)
$$

Neutrophil activation occurs in the blood, and is up-regulated by the presence of the pro-inflammatory effector, which diffuses into the blood in amounts proportional to those present in tissue. This activation is inhibited by anti-inflammatory effector activity. Activated neutrophils rapidly decay in the blood, unless they are adducted chemotactically into the tissue. Hence not all activated neutrophils will find their way into the tissue. Once in the tissue, neutrophils also decay at a constant rate.

Neutrophils in tissue have an undifferentiated response, and hence act the same whether there is benefit to the system or not. As mentioned above, this response mediates the production of toxic substances. We assume neutrophils produce these substances in a Michaelis-Menten manner; also, we assume small amounts are produced by macrophages in response to signal. These toxins are either degraded or consumed through their interaction with tissue. Conversely, we model tissue damage with Hill dynamics; this is a simple way to reflect resilience of tissue to small levels of toxin (Keener and Sneyd). Total tissue, in a general setting, is represented by a total number of cells H_0; in this model the number of target cells is normalized to be $H_0=1$, so that we can represent H as the percentage of available tissue. Tissue regenerates proportionately to the interaction of the active tissue in the system, H, with the amount of dead tissue in the system, (H_0-H). Hence, when there is no dead tissue, there is no growth; and there is no growth when there is no active tissue. Furthermore, we assume that tissue growth has an Allee threshold, θ, a minimal amount of tissue needed for any growth to occur, and so the growth term is multiplied by $(H-\theta)/H_0$ (Stephens and Sutherland 401-05). For active tissue less than θ, tissue growth is negative; this corresponds to necrotic tissue sending a signal for remaining cells to apoptose (Hitomi et al. 1311-23). A priori, we set θ close to zero.

Finally, the algebraic equation for the signal received b y macrophages, Σ, is redefined to be the summation of the signal received from the pro-inflammatory effector and the signal received from cellular debris, proportional to the amount of dead tissue (H_0-H) (Eigenbrod et al. 8194-98).

Parameter	Significance	Value
a_1	Signalling parameter for pro-inflammatory effector	1.0
a_2	Signalling parameter for dead tissue	0.7
g_1	Inhibition parameter for signalling	40.0 conc
g_2	Inhibition parameter for signalling	10.3 $conc^2$
d_2	Inhibition parameter for signalling	1.0 conc
b_{mc}	Rate of macrophage recruitment by chemokines	5.5 conc $time^{-1}$
a_{mc}	Michaelis-Menten parameter for macrophage recruitment	1.0 $conc^2$
p	Hill-coefficient for macrophage recruitment	2.0
μ_m	Rate of macrophage derecruitment	0.4 $time^{-1}$
b_m	Equilibrium value of macrophage population	5.0 conc
b_t	Rate of pro-inflammatory effector production	10.0 $time^{-1}$
μ_t	Rate of pro-inflammatory effector decay	6.5 $time^{-1}$
k_1	Inhibition parameter for pro-inflammatory effector	0.95 conc
k_2	Inhibition parameter for pro-inflammatory effector	2.0 $conc^2$
f_0	Inhibition parameter for pro-inflammatory effector	2.5 conc
τ	Time parameter for anti-inflammatory effector	80.0
b_l	Rate of anti-inflammatory production effector	20.0 $time^{-1}$
μ_l	Rate of anti-inflammatory effector decay	5.0 $time^{-1}$
b_c	Rate of chemokine production	40.0 $time^{-1}$
μ_c	Rate of chemokine decay	8.0 $time^{-1}$
b_{nt}	Rate of neutrophil activation	5.0 conc $time^{-1}$
a_{nt}	Michaelis-Menten parameter for neutrophil activation	15.0 conc
a_{nl}	Inhibition parameter for neutrophil activation	40.0
μ_{nb}	Rate of neutrophil deactivation	0.1 $time^{-1}$
r_{nc}	Rate of neutrophil recruitment	10.0 $time^{-1}$
a_{nc}	Michaelis-Menten parameter for neutrophil recruitment	10.0 conc
μ_{nt}	Rate of neutrophil derecruitment	0.03 $time^{-1}$
b_{xn}	Rate of toxin production by neutrophils	50.0 conc $time^{-1}$
a_{xn}	Michaelis-Menten constant for toxin production	1.5 conc
b_{xm}	Rate of toxin production by macrophages	0.01 $conc^{-1}time^{-1}$
μ_{xh}	Rate of toxin decay	8.0 $time^{-1}$
g_{xh}	Rate of toxin destruction interacting with tissue	35.0 $conc^{-1}time^{-1}$
b_h	Rate of tissue regeneration	0.15 $conc^{-2}time^{-1}$
θ	Threshold for tissue	0.05 conc
g_{hx}	Rate of tissue destruction interacting with toxin	9.0 $time^{-1}$
q	Hill-coefficient for tissue destruction	2.0
a_{hx}	Michaelis-Menten coefficient for tissue destruction	17.3 $conc^2$
τ	Slow time coefficient	80.0

Table 2: Parameter values used for inflammation system

2.3.2 Tissue Response to Insult

As with the Inflammation subsystem, we can model the response of the larger system to the presence of an initial amount of endotoxin. Furthermore, we can model the response of this system to tissue damage and the healing process from that damage (Figure 14, Figure 15, Figure 16); trajectories are produced using XPPAUT (Ermentrout). Following immune stimulation by a substance such as endotoxin, neutrophils in blood and tissue as well as toxin spike initially along with the inflammatory subsystem (Figure 14). Blood neutrophils have a secondary spike from continued activation without outward migration. Yet, there is limited excursion into tissue in view of the relatively weak chemokine signal. Hence, neutrophils accumulate for some time in the circulation before deactivation brings them to rest. The tissue takes a minor blow from the introduction of endotoxin, but promptly heals.

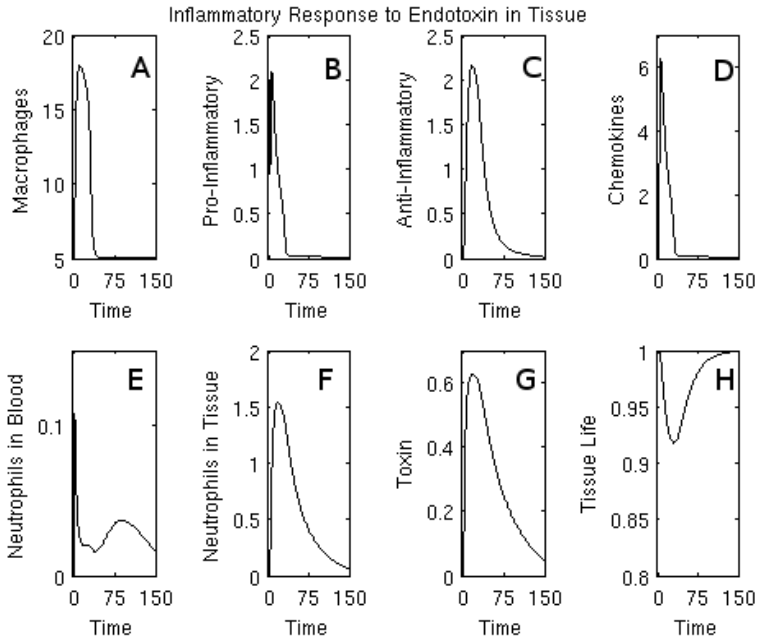

Figure 14. Tissue Level Inflammatory Response to Endotoxin:

The system response to an initial stimulus of pro-inflammatory T(0)=1. Here, as in **Figure 15** and**Figure 16** below, the graphs represent time courses for (A) macrophages, (B) pro-inflammatory effectors, (C) anti-inflammatory effectors, (D) chemokines, (E) neutrophils in blood, (F) neutrophils in tissue, (G) toxin, and (H) tissue life.

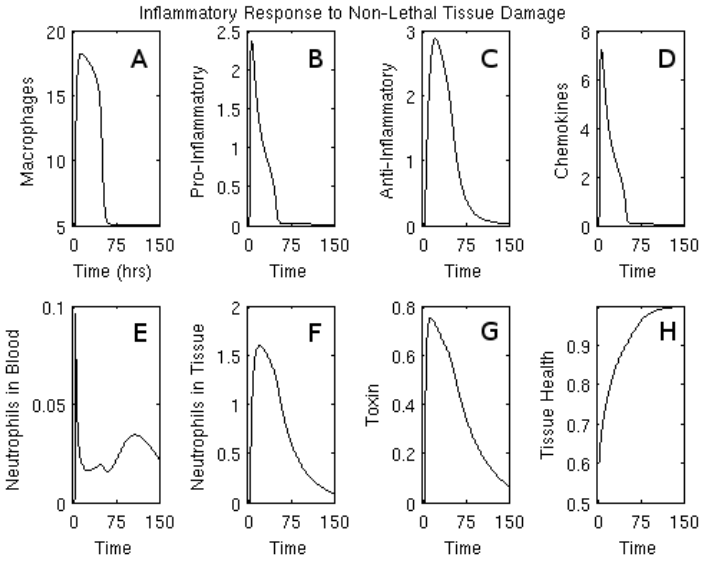

Figure 15. Nonlethal Tissue Level Inflammatory Response to Damage:

The system response to 40% of tissue being damaged.

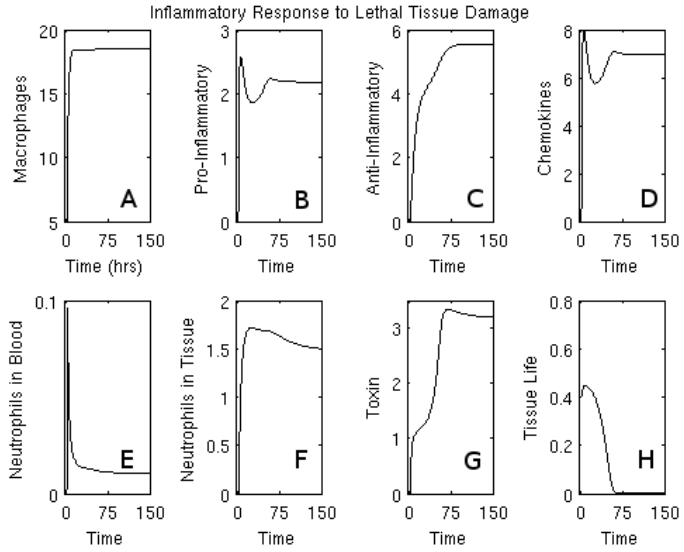

Figure 16. Lethal Tissue Level Inflammatory Response to Damage

The system response to 60% of tissue being damaged.

Consider now the infliction of direct tissue damage (Figure 15). If damage is modest (40%), one observes restoration of quiescence in the inflammation subsystem as in the endotoxin case and response in the inflammation subsystem. However, the response is of greater magnitude as damage contributed to the pro-inflammatory response. In the tissue diagram, notice that the graph changes convexity. When damage inflicted is heavy (60%), one observes an initial attempt at reducing damage (healing) within the first 12 hours (Figure 16). However, collateral damage from activated neutrophils overwhelms the tissue growth rate, and the tissue is dead within 72 hours. All other equations remain activated by the signal from the dead tissue. Since θ is set to 0.05 and the initial active tissue (H) initiates at 0.40, the growth term in the H equation continues to be positive for the first 72 hours. This shows that the threshold at which tissue death may occur happens as a product of the model, and not as the direct result of a pre-set parameter θ. Similar behavior occurs for θ set to 0. Hence, the positive feedback mechanism of dead tissue activating neutrophils which cause more damage results in the death of the system (Figure 16). However, this positive feedback loop is regulated (Figure 15). As the anti-inflammatory effector inhibits the activation of neutrophils, we hypothesize a non-homeostatic initial level of anti-inflammatory effector will aid in the regulation of higher levels of damage.

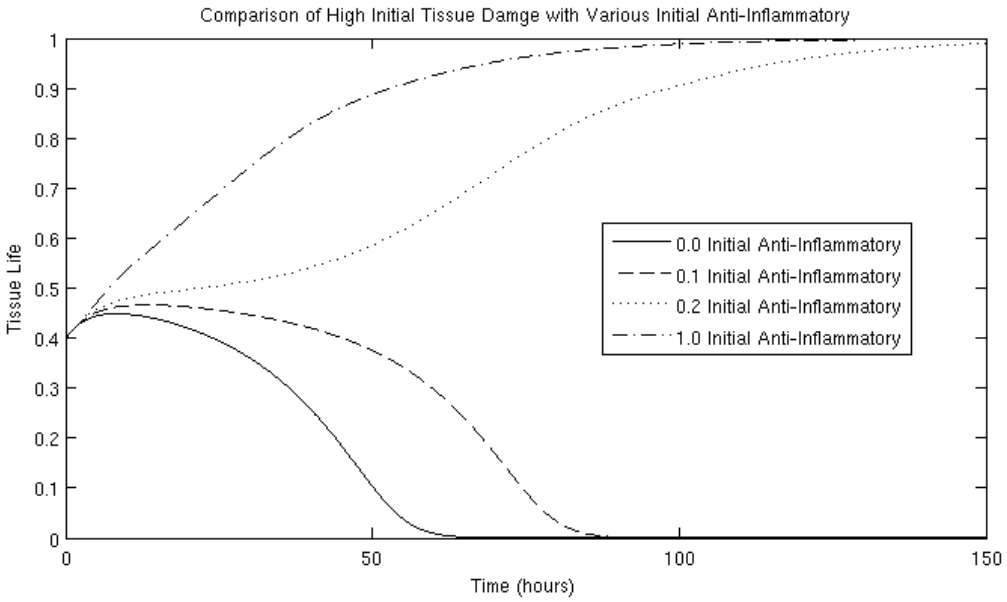

Figure 17. Comparison of Tissue Healing for Various Initial Levels of Anti-Inflammatory Effectors

Several time courses of Tissue Life starting from 60% damage plotted together, each with a different amount of initial anti-inflammatory.

We pre-condition the system with increasing amounts of initial anti-inflammatory effector along with 60% initial damage and evaluate whether damage is controlled (Figure 17) . A small amount increases the life-span of the tissue, but does not allow for healing; a moderately large amount allows for a slow recovery; a very large amount of anti-inflammatory effector allows for a swift recovery. We note that introducing anti-inflammatory cytokines is not a panacea for all levels of damage. At levels of damage exceeding 80% damage, no amount of anti-inflammatory effector will recover the system.

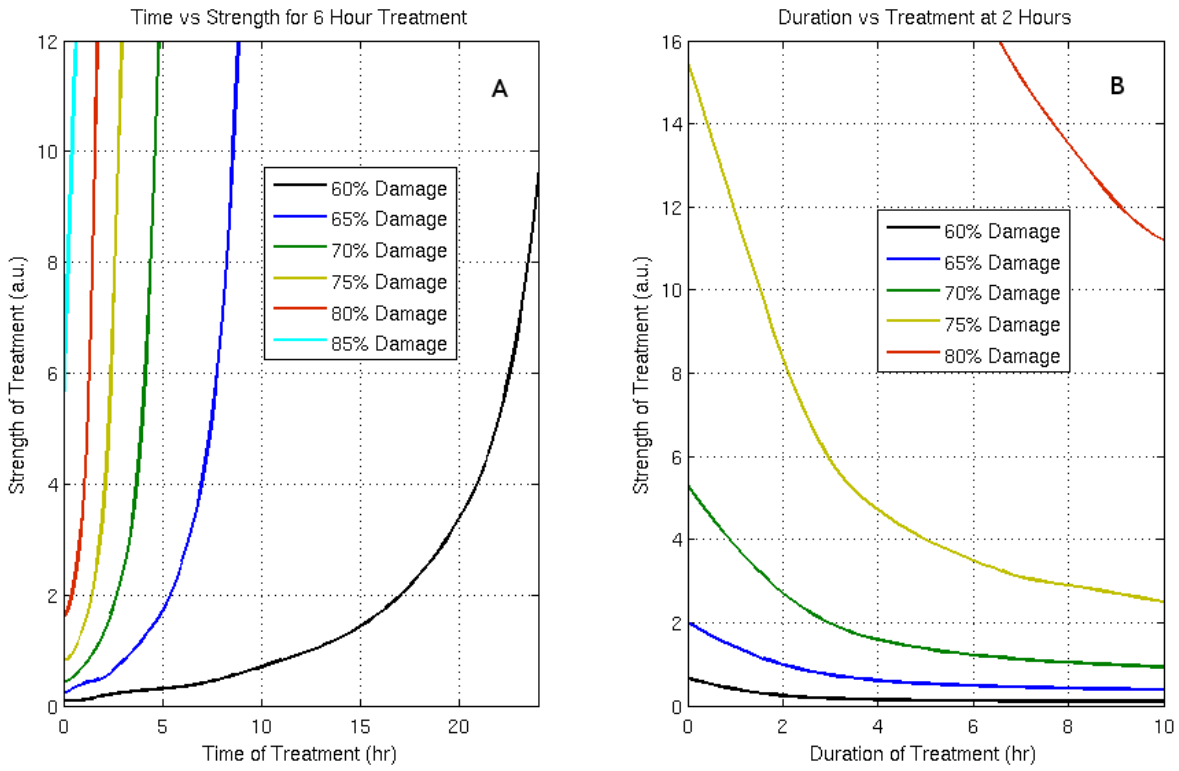

Figure 18. Minimal Interventions Required for Healing

(A) We assume a 6 hour intervention of anti-inflammatory effectors. The x-axis represents the time of treatment after the initial trauma, and the lines represent the minimal strength of the intervention to ensure survival given that level of damage. (B) We assume there is treatment 2 hours after the initial trauma. The x-axis represents the duration of treatment with anti-inflammatory effectors, and each line represents the minimum strength necessary for survival given that level of damage.

To explore more fully the space of opportunity of an anti-inflammatory effector as an intervention to recover from tissue damage, we consider three axes of intervention: (1) the time of treatment after the initial incident, (2) the duration, (3) and the dose of the treatment (Figure 18, panels A and B). In both figures we view several lines corresponding to the initial damage. The area above the line represents a dose that will result in recovery, although recovery time may take as long as two weeks. Treatment within two hours, administered for six hours, will recover tissue with up to 75% damage. If treatment is delayed, fewer cases can feasibly be recovered.

Because of saturation effects, introducing arbitrarily large amount of anti-inflammatory effector does not result in better outcomes. We examine the relationship between the duration of intervention and the minimal dose of intervention to ensure recovery, assuming intervention is initiated at two hours (Figure 18 B). A high dose of anti-inflammatory effector can recover 70% damage if administered for three hours and 75% damage if administered for eight hours. Further increase in duration will not recover higher levels of damage. These scenarios can apply to burn patients or blunt trauma victims. However, large amounts of anti-inflammatory effectors, such as corticosteroids have also been shown to favor infection and may worsen outcome, by virtue of their immunosuppressive effect (Sprung et al. 1137-43;Sun et al. 76-84).

2.3.3 Stability of Tissue and Inflammation Model

We choose parameters in the tissue module on the same scale and the inflammation module with the restriction that H=0 and H=1 are stable, attracting, fixed points. We observe, however, the theoretical existence of a third stable fixed point for some parameter sets (Figure 19). Increasing parameter a_2, the proportionality constant of the damage signal, in the continuation program AUTO (Doedel 265-84), we see the a saddle node bifurcation introduce two new branches of fixed points, whose lower branch becomes stable in an Andronov-Hopf bifurcation. For the purposes of this model, we set a_2 below the critical value, so that there are only two stable fixed points.

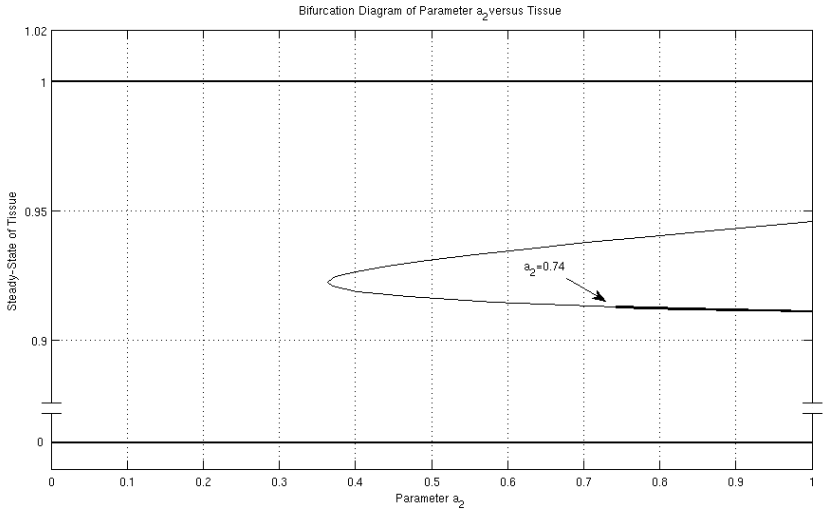

Figure 19. Bifurcation Diagram for Strength of Damage Signal, a_2

Bifurcation Diagram for a_2, the parameter for the strength of the damage signal. For a_2 large, a positive feedback loop in the inflammatory system forms and a third branch of fixed point becomes stable. This third branch (in addition to H=1, health and H=0 death) represents a chronically inflamed state.

However, the third stable arm is theoretically interesting as it represents incomplete healing close to complete tissue integrity. The biological correlate would be the persistence of damaged tissue maintained by ongoing influx of neutrophils that damage tissue at the same rate at which it is being repaired. Whether this mechanism could form the basis of chronic inflammatory or autoimmune conditions, such as rheumatoid arthritis, is an interesting hypothesis. Chronic anti-inflammatory interventions have shown benefit in such afflictions.

2.3.4 Propagation of Damage in Space

In Figure 19, we see that by varying the damage feedback parameter, a_2 that the bistable system becomes tri-stable for increasing values. As in 2.2.5, we use this alternate fixed point as a starting point in looking for symmetry breaking instabilities. Diffusion and chemotaxis are added to the system to make a system of partial differential equations in one dimension of space,

using the perimeter of a circle, S^1. Perturbation techniques were employed to find which sets of spatial parameters allow for patterns to form. From prior distributions of values, parameter sets were produced randomly and tested for positive eigenvalues for wave numbers one through ten. Depending on the spatial parameter set, symmetry breaking bifurcations arise in space with various resulting wave numbers between k = [2, 9]. The basic formula for linear perturbation is given in section Section 2.2.5.

The diffusion was discretized using central differences with the chemotaxis term discretized with an upwinding scheme.

$$
\begin{aligned}
&\Sigma := a_1 T + a_2 (H_0 - H) \\
&\frac{dM}{dt} = \frac{b_{mc} C^p}{a_{mc}^p + C^p} - \mu_m (M - b_m) + D_M \frac{\partial^2 M}{\partial x^2} - \chi_1 \frac{\partial}{\partial x}\left(M \frac{\partial C}{\partial x} \right) \\
&\frac{dT}{dt} = \frac{b_t M \Sigma}{\Sigma + \left[\Sigma + \frac{g_1 L + g_2}{L + d_2} \right] \left[\frac{k_1 L + k_2}{L + f_0} \right]} - \mu_t T + D_T \frac{\partial^2 T}{\partial x^2} \\
&\frac{dL}{dt} = \frac{b_l M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_l L + D_L \frac{\partial^2 L}{\partial x^2} \\
&\frac{dC}{dt} = \frac{b_c M \Sigma}{\Sigma + \frac{g_1 L + g_2}{L + d_2}} - \mu_c C + D_C \frac{\partial^2 C}{\partial x^2} \\
&\frac{d\hat{N}}{dt} = \frac{b_{nt} T}{a_{nt} + a_{nl} L + T} - \frac{r_{nc} N C}{a_{nc} + C} - \mu_{\hat{n}} \hat{N} + D_{\hat{N}} \frac{\partial^2 \hat{N}}{\partial x^2} - \chi_2 \frac{\partial}{\partial x}\left(M \frac{\partial \hat{N}}{\partial x} \right) \\
&\frac{dN}{dt} = \frac{r_{nc} N C}{a_{nc} + C} - \mu_n N + D_N \frac{\partial^2 N}{\partial x^2} - \chi_2 \frac{\partial}{\partial x}\left(M \frac{\partial N}{\partial x} \right) \\
&\frac{dX}{dt} = \frac{b_{xn} N}{a_{xn} + N} + b_{xm} M \Sigma - g_{xh} H X - \mu_x X + D_X \frac{\partial^2 X}{\partial x^2} \\
&\frac{dH}{dt} = b_h H (H_0 - H)\left(\frac{H - \theta}{H_0} \right) - \frac{g_{hx} H X^q}{a_{hx}^q + X^q} + D_H \frac{\partial^2 H}{\partial x^2}
\end{aligned}
\tag{0.20}
$$

	Figure 20	Figure 21	Figure 22
D_M	0.007	0.0058	0.0086
D_T	0.0081	0.0031	0.0063
D_L	0.179	0.131	0.0914
D_C	0.0914	0.0965	0.0662
D_{Nb}	0.0034	0.0032	0.0067
D_N	0.0037	0.009	0.0068
D_X	0.265	0.28	0.253
D_H	0.000663	0.00797	0.00065
X_1	0.000602	0.000984	0.0012
X_2	0.000101	0.000164	0.00015

Table 3: Examples of Diffusion and Chemotaxis Parameters that produce patterns

Localized damage was then applied to simulate a wound. Parameter sets producing positive eigenvalues were introduced and simulations run; the spatial parameters are labeled by the values of the wave number k for which the eigenvalue is positive. Unlike the pattern with macrophages only, the damage stays localized and does not propagate across space.

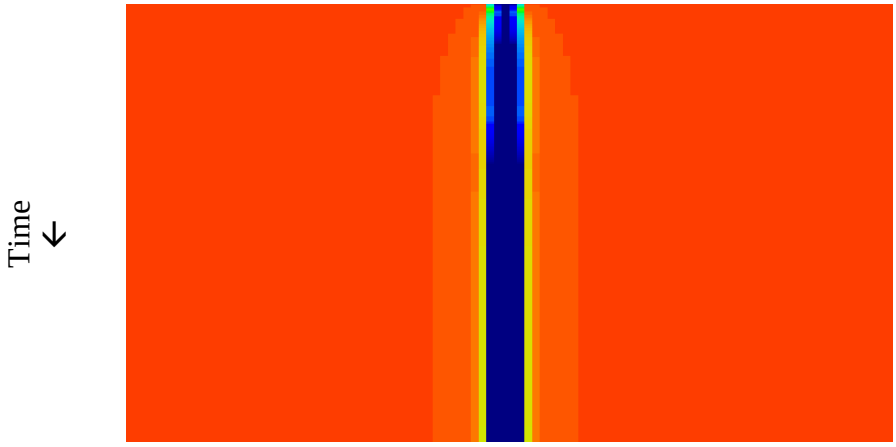

Figure 20. Damage with Wave Number 2

Diagram of Epithelial cells versus Time, with red representing H=1.0 and blue representing H=0.8.

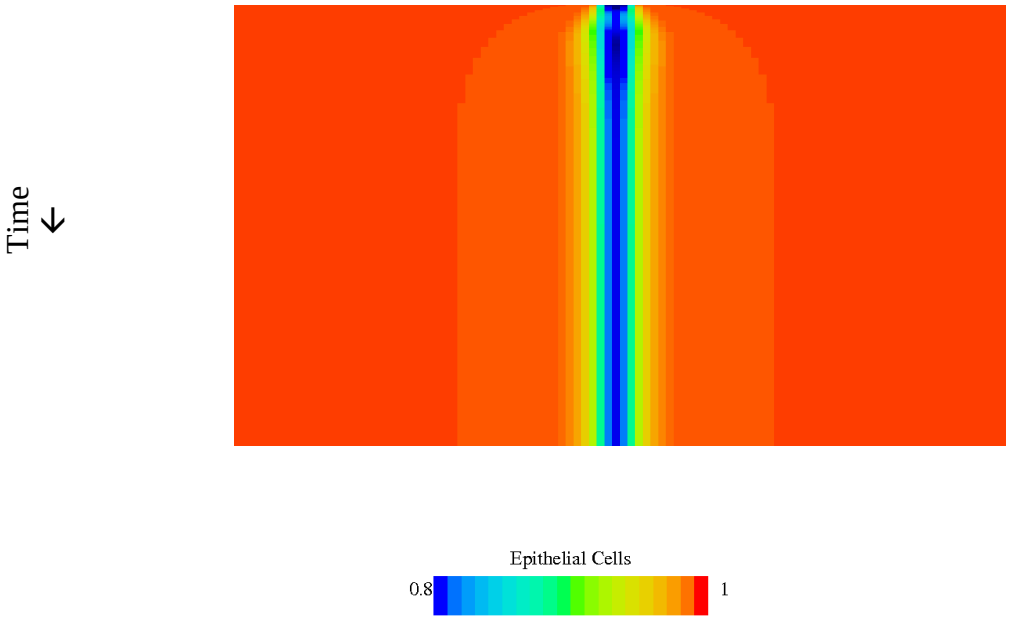

Figure 21. Damage with Wave Number 6

Diagram of Epithelial cells versus Time, with red representing H=1.0 and blue representing H=0.8.

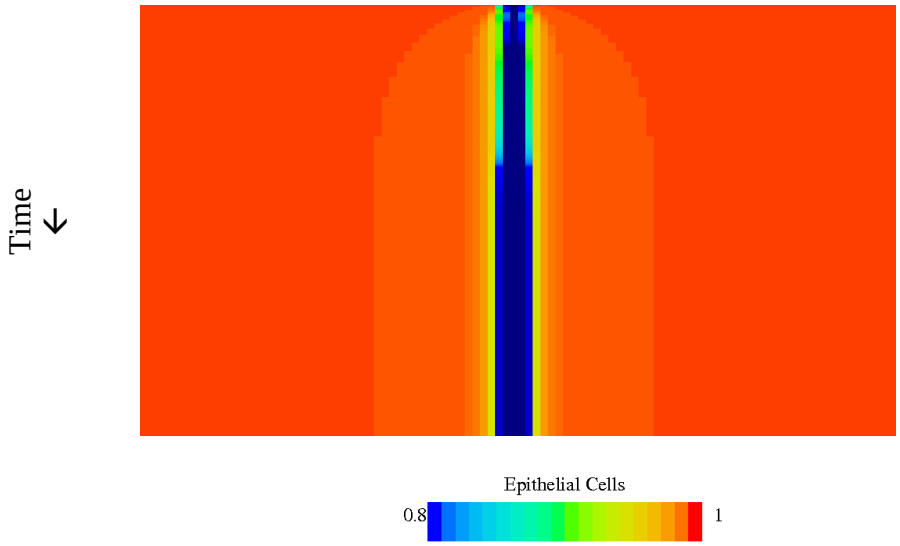

Figure 22. Damage with Wave Number 9

Diagram of Epithelial cells versus Time, with red representing H=1.0 and blue representing H=0.8.

Very often, in a well-mixed system, a parameter set representing a dysregulated or chronically inflamed system can stabilize additional fixed points. In a well-mixed system, the totality of the inflammation and damage is represented as a whole, whereas this may not be the case when space is introduced. In biological systems, repeat exposure to irritants can cause the aggradation of immune cells and the localized loss of tissue, such as the case with granulomata. Both the formation and response of these conglomerates of damage, especially in the lungs pose an important modeling problem as the rise of inflammatory diseases, such as Chronic Obstructive Pulmonary Disease (COPD). Modeling the response of these damaged systems to further pathogens such as pneumonia and virus also becomes important in understanding the disease processes when these co-morbidities are present.

2.3.5 Discussion and Conclusions

We developed a mechanistic model of the host response to local tissue stress that extends our previous work (Chow et al. 74-84;Kumar et al. 145-55;Reynolds et al. 220-36) by including more biologically plausible cytokine dynamics and cell recruitment mechanisms. This extension is mandated by the growing availability of biological data that such mathematical models should assimilate, and on which prediction performance should be assessed. The model presented herein is designed to preserve the ability to describe supra-cellular dynamics, yet construct the input-output mapping that cells represent in a less heuristic, more biologically relevant fashion. In response to an inflammatory stress, which is instantiated by activation of local surveillance cells, typically tissue macrophages and antigen presenting cells, the model produces a characteristic spike of pro-inflammatory effectors closely followed by a spike of anti-inflammatory effectors, a dynamic that is observed both in data and other models of acute inflammation. The model presented herein has several structural similarities to its predecessors such as an additive signal term, nonlinear production and inhibition terms, and feedback pro-inflammatory inflammation originating from tissue damage. Several new features are included in this model such as (1) separation of roles for pro-inflammatory cytokines and chemokines, (2) the distinction of the primary role of neutrophils in defense from the primary role of macrophages in cytokine production, and (3) an inhibition term by anti-inflammatory effectors derived from described signaling interactions and intuitive from a perspective of Michaelis-Menten kinetics. Specifically, we intended to represent a simplified biology of IL-10 down-regulation of the NK-kB signaling pathway, or of TNF and IL-1 in particular. Accordingly, our model would be particularly well suited to assimilate datasets that include time series tissue level and/or serum measurements of these analytes, in addition to markers of tissue injury.

As presented, the model does not include an explicit infectious trigger, such as local pathogen. It was primarily designed to focus on propagation of the immune response as seen in aseptic tissue damage or auto-immune damage. In the absence of a signal from pathogen, we use the model to critically examine endogenous signals and their effects. We exploit the fact that acute inflammation acts locally. This model distinguishes itself in its treatment of biological signalling on intracellular and intercellular levels to achieve a generic initial inflammatory response.

Among the features of the model, macrophage and neutrophil migration depend upon chemokine production and decay, rather than upon pro-inflammatory cytokines. While the biological motivation of this representation seems to be obvious, it may not seem immediately clear from a modeling perspective that such a representation is desirable. These effects are lumped together in previous models. As a result, assimilation of cytokine time series has required the introduction of specific time delay, or biologically implausible Hill exponents in the cellular response function to accommodate observed biological timing (Chow et al. 74-84;Daun et al. 843-53). Yet, from the literature we know that the production, effects, and diffusion of chemokines differ widely from those of canonical pro-inflammatory cytokines, such as TNF and IL-1. The model reflects that their regulation and behavior are different. While the two curves may appear similar in simulation scenarios presented, this is not a rule for the system and for all valid parameter spaces. In other parameter spaces, their behavior can be quite different, and hence the model can be calibrated against a wider variety of data sets.

Not every parameter set produces dynamics that are in accord with known information of the inflammatory mechanism. There are several explicit constraints on admissible regions of parameter spaces for this model. At various points, one-parameter and two-parameter

bifurcations show limitations on parameters in relation to each other. Taking all these into consideration, we surmise that upon calibration to a dataset, identifiability and sensitivity analyses are likely to lead to reductions in the number of free parameters of the system (Daun et al. 843-53).

We believe that such a reformulation of the model was important in achieving the goal of producing a versatile, predictive model of systemic immune response to both aseptic and septic forms of tissue damage. Not all pathogens or irritants themselves cause significant damage, but still induce this generic, self-amplifying, immune response. This response does not always aids in combating pathogens, and may cause collateral damage through exaggerated acute response. Finally, an initial event could trigger a self-sustaining response. This model allows the possibility of a chronic inflammatory state stabilized by a positive feedback loop of damage and signaling. Ultimately, this model fills a need to account for inflammatory damage based on immune signaling, rather than the direct presence of pathogen. The model also allows the examination of a wider range of perturbations, disruptions or 'treatments' to the system than simpler versions.

We are specifically interested in examining the immune response to viruses of exceptional virulence. This is particularly useful, as in such illnesses the acute inflammatory response has been documented to play a major role in inflicting a large amount of damage, and be a direct cause of death. This is certainly the case for pandemic influenza and viruses associated with hemorrhagic fever (Kobasa et al. 319-23;Kash et al. 9499-511). However, this immune mobilization is required to initiate a signalling cascade that recruits the adaptive immune response and a response that can eliminate the virus and provide future immunity.

Therefore, descriptive, predictive and intervention models of severe viral inflictions, such as Influenza A, model of IAV must extend beyond the classic target-limited formulation and include inflammation as a key component, as it becomes clear that successful intervention has to be early and include a significant immuno-modulatory component. Indeed, modeling an intervention for aggressive viral infections (or other pathogens as well) will likely point out a requirement of minimizing inflammation-inflicted damage without compromising the advantages of the acute inflammatory response.

3.0 THE INFLUENZA MODEL

3.1 OVERVIEW

The original intent of the project was to merge the Hancioglu model of innate, adaptive and humoral immunity with a variation of the Reynolds and Day model (Reynolds et al. 220-36;Day et al. 237-56) of inflammatory response of tissue to endotoxin and pathogen. However, rather than re-use the model proposed by Reynolds and Day, we developed the inflammatory model as proposed in Chapter 1.3. The Hancioglu model(Hancioglu, Swigon, and Clermont 70-86) is a derivative of the Bocharov model(Bocharov and Romanyukha 323-60), and provides our starting point. At the start of the modeling process, the only term shared by the inflammatory model and the Hancioglu model are target cells, H; and while there are terms in the inflammatory compartment that affect that influence the immunity compartment, the only mediator from the influenza compartment to the inflammatory compartment are dead cells.

Simultaneously, our work with Franklin Toapanta and Ted Ross (Toapanta and Ross 112) gave us access to a large array of data. With data, several new problems came to the floor. First and foremost, now that data exists, how do we correlate theoretical biological classes within the model to very specific data. Data was collected by optical density through Luminex or Elisa, else through Flow Cytometry. Especially with cells, there exists a continuum of phenotypes that have the fluid, similar basic properties but different functionalities as represented by the proteins

present on the surface of the cell. For example, a "macrophage" and an "antigen presenting cell" could be considered the same item, as they both are monocytes that have many of the same properties, share some surface proteins, and overlap in some unspecialized tasks; however, they perform different functionalities in the model. Second, given that we can be specific about which cell and cytokine does what, how specific do we become in the model formulation and the level of abstraction we keep. These considerations stay with us throughout the modeling process.

The Hancioglu model has as state variables: Target Cells (H), Infected Cells (I), Resistant Cells (R), Virus (V), Antigen Presenting Cells/Macrophages (M), Interferon (F), Plasma Cells (P), Effector Cells (E), Antibodies (A), and Antigenic Distance (S). In the process of developing the model in response to data: we remove variable (S), divide (F) into Type I (F) and Type II (G) interferon, divide (E) into NK cells (K) and NKT cells (E), reformulate our representation (R), parse Antigen Presenting cells into APCs (P) and macrophages (M), replace Plasma Cells by B-cells (B), and introduce T_H1 Cells (O) and IL-12 (W). Combined with the inflammatory model, we present at the end of the process a model in 20 variables and 90 parameters, in contrast to the Hancioglu model in 10 variables and 27 parameters.

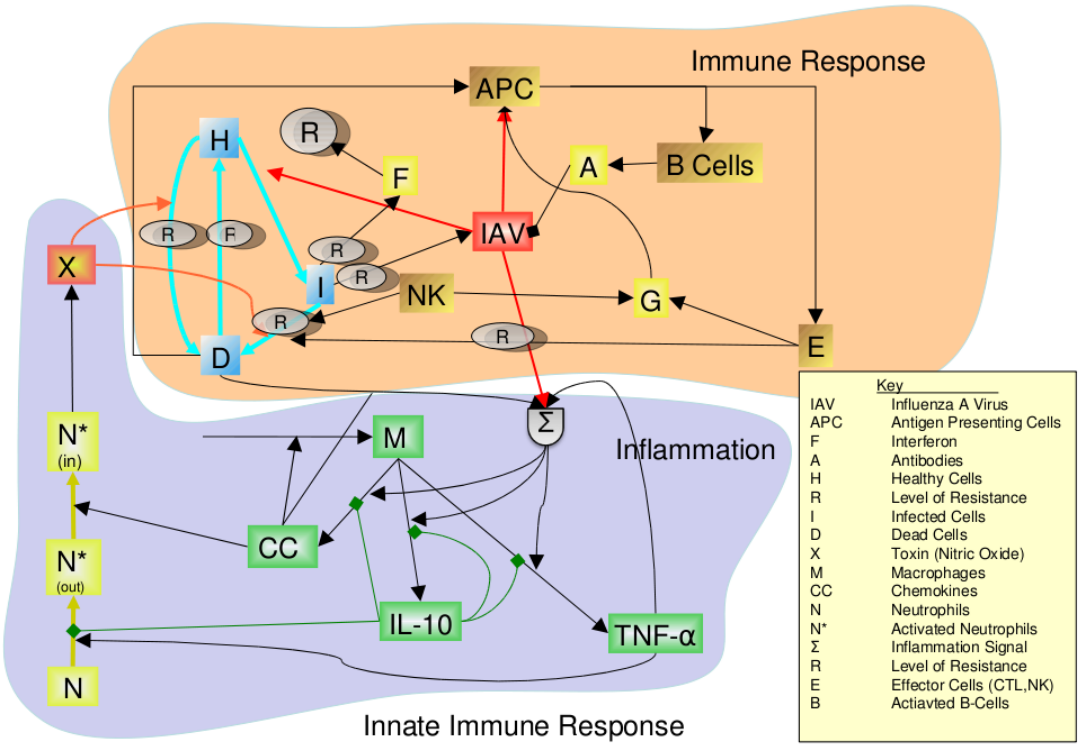

Figure 23. Model Schematics for Immune Response to Influenza A Virus

3.1.1 Modeling Process

The first iteration of the "new" influenza model, was to add Virus to the Tissue model. This also involved adding infected cells, as well as the target cells. A distinction is now made between cells that die from inflammatory activity (DH) and cells that die from viral apoptosis (DI). An equation is also added for DH, and DI is represented algebraically as the number of epithelial cells remains constant. Thus, in the model, the only mediation of the immune system is neutrophil activity to eliminate infected cells. This is a very blunt tool, and can clear very small infections, but trying to augment neutrophil activity to clear all infections

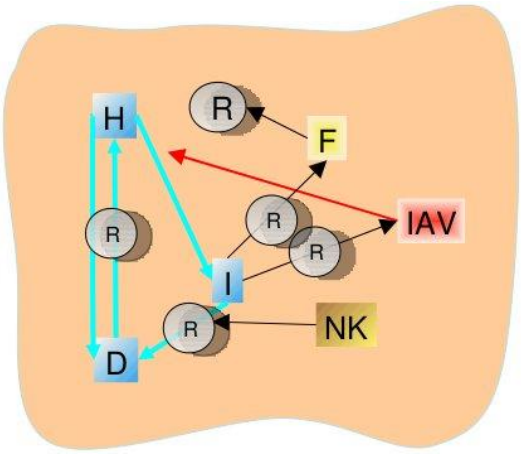

Figure 24. Model Schematics for Innate Immunity

The second iteration was more involved. Primarily, interferons which has been a single entity in the Hangioglu model were divided into Type I interferons (alpha and beta) and Type II interferons (gamma) (Pestka, Krause, and Walter 8-32). The cell class known as Resistant was stricken from the model; exposure to Type I Interferon does not remove the receptor to which the influenza haemagglutinin binds. Mostly produced by infected cells, these interferons acts paracrinely to halt cellular machinery in viral production, to prevent viral apoptosis, and to signal to effector cells using MHC cell surface presentation for the cell to be lysed (Smith, Lombardi, and Foster 869-77;Carayannopoulos and Yokoyama 26-33). Type II interferons act later in the system to aid in the maturation of adaptive cell types (Schroder et al. 163-89;Chesler and Reiss 441-54). A new algebraic quantity, also named R, as a function of interferon, is introduced to the system to represent the percentage of epithelial cells, both target and infected, affected by Type I interferons. R is bound by 0 and 1, and the terms (1-R) and (R) are multiplied by the pertinent terms as described by the lines above. Additionally, the function R(F) inhibits mitosis, so that new epithelial cells are not produced. Production of interferon, because it halts cellular production, is modeled to be self-inhibiting; as R approaches 1, fewer infected cells properly

produce interferons, and so production falls. This form of innate immunity is therefore by no means perfect, and practically the value of R rarely rises above 0.8 for any parameter choice.

The other aspect of innate immunity that was next added, are Natural Killer (NK) cells(French and Yokoyama 45-51). In the Hancioglu model, the class of Effector Cells acted upon the system by lysing infected cells. Within the data, and also conceptually, this role can be divided into two cell types, NK Cells and T-Cells, also called NKT-Cells (Yokoyama, Kim, and French 405-29;Yokoyama 363-65). While NKT-cells are highly specific as well as highly efficient in the destruction of infected cells, NK cells are effective in clearance of infected cells but do not need time to mature and appear earlier in response to irregularities in the cells caused by infection. In the model, NK cells simply need a chemokine gradient to enter the system (Walzer et al. 1337-44). In the model, NK cells are also one of the initial sources for the production of Type II interferon (Carayannopoulos et al. 404-14), becoming an important producer of the interferon before the adaptive immune system becomes fully active. In the absence of an adapative and humoral immune system, this model becomes tri-stable in numerical trajectories run: return to baseline, death, and viral fixation. In the previous iterations, only baseline and death were explicitly stable fixed points, with most trajectories going to death when virus is introduced. But, with the innate immunity functioning, it is possible for the immune system to target a sufficient number of infected cells so that the virus and target cells stay at stable, non-zero levels.

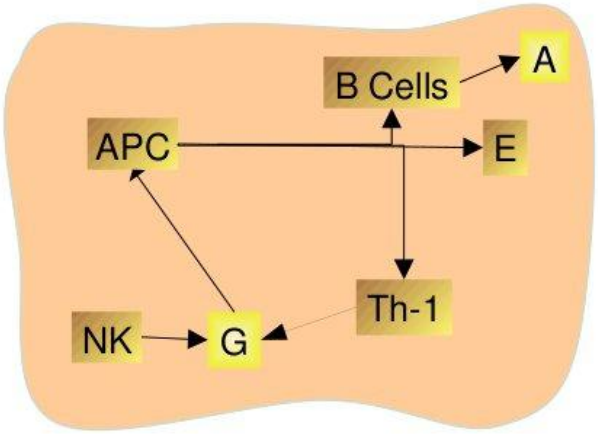

Figure 25. Model Schematics for Adaptive Immunity

The structure for the Antigen Presenting Cells (APCs) is modeled from the presentation in Hancioglu et al (Hancioglu, Swigon, and Clermont 70-86), where APC activation happens as the product of the reservoir of unactivated APC and a signal from either virus or dead cells. These cells are thought of as being dendritic cells, that perform MHC II presentation of viral components to the greater immune system. In the first iteration of the model, this activation is explicit, with a naïve population of APCs and an activated population. Further, since one of the roles of Type II interferon is the maturation of dendritic cells, the activation term for the APCs is multiplied by a function of Type II interferon. Type II interferon is not explicitly needed for the activation, as the function is comprised of a constant term added to a saturating function of Type II interferon. For the activation term itself, the signal for viral particles comes from either free virions or dead infected cells; recall the model differentiates between cells killed by inflammation and by virus. Because the data contains a high pfu count for virus at its peak and because we assume receptor saturation for free virus, we use a saturating function of virus rather than a linear function. The function of dead infected cells, on the other hand, is bound by the total number of epithelial cells, which is an impracticality to reach, so is modeled locally by a linear function.

Following the Hancioglu model, the original production of NKT-cells is proportional to the number of activated APCs, has a steady state level, and de-recruits modestly. A term is also added as a fatigue term, given that NKT-cells deactivate after eliminating a finite number of infected cells; this term is multiplied by the function R(F), since NKT-cells target only infected cells with the presentation of MHC markers, as happens to cells in the presence of Type I interferon.

T_H1 cells, or helper T-cells enter the system to aid in the development and maturation of the cells in the adaptive immune component of the model. Production works similarly to that of NKT-cells, but there is no degradation due to lysis of infected cells. T_H1 cells produce Type II interferon, which as stated is an important component to the activation of APCs; this is the first sense in which T_H1 cells help create a positive feedback loop. They also interact with APCs to produce the cytokine IL-12.

IL-12 is an important co-factor to the production of Type II interferon in NK cells and in T_H1 cells. Therefore, activated APCs trigger the activation of T_H1 cells, which promote the activation of APCs, creating a loop of positive growth that can only be slowed by the elimination of the signal from virus. The production of IL-12 occurs when T_H1 cells bind to APCs, and so we use a Michaelis-Menten term of the T_H1 to represent receptor saturation of these two cells.

B-cell maturation is originally modeled based off Hancioglu, as being proportional to the number of APCs; however, as IL-12 is important to the maturation of B-cells, the maturation term of B-cells is multiplied by a saturating function of IL-12. B-cells are modeled here, as data for Plasma cells was unavailable.

As in the Hancioglu model, Antibodies are modeled as being produced linearly from B-cells, decaying at a certain rate, and being eliminated at a certain rate when they encounter free

virions. Unlike the Hancioglu model, antigenic distance is not invoked. The data shows virus being cleared in the sublethal cases in 5-7 days. Since it takes over a week for antigenic distance to change, we assume the antigenic distance is approximately constant in the time frame of interest. As in the Hancioglu model, it is the model use of antibodies, or humoural immunity, that ensures that virus is cleared rather than becomes fixated or rebounds. The other sources of immunity: inflammation, innate, and adaptive, all focus on the cellular level. The antibodies work on the level of macro-molecules to bind to individual free virions, thereby eliminating the source of future infections rather than treating infection ex post facto. From a practical level of modeling, an initial viral trajectory will not have a positive slope if there is a large number of steady state antibodies present; instead the trajectory will have a negative slope in time towards the non-infected state.

This is the model as originally envisaged, complete with all the cell types and interactions we thought necessary through research and conceptual and qualitative modeling of the system. Quantitative modeling and parameter fitting brings several new aspects of the system interactions into play, and the model is changed accordingly as we will elaborate further.

3.1.2 Model Refinement

As the process of parameter estimation unfolded, and the trajectories compared against data, iteratively changes were made to the initial model described in 2.1.1 to better suit the theory and the experimental data. This process of critically analyzing the data and the equations has yielded a model better capable of reflecting biological knowledge and reproducing the data.

The first major change is to reflect the role of the macrophage as a sentry in the epithelium against pathogens and irritants. Alveolar macrophages, the baseline population of

macrophages in the system, are lodged randomly between target cells and have a proportional probability of being exposed to virus when virus first enters the system. On this initial exposure to virus, these macrophages undergo various behaviors, and may become infected with virus, although they are not a major source of future virions. Importantly among the reactions of alveolar macrophages to virus early on, is the production of TNF-α. In the data we see an early rise in this cytokine that does not correspond to the signal either from dead cells or other pro-inflammatory cytokines. So, the signal term is changed so that there are now two signal terms, Σ_1 and Σ_2. The first signal term is identical to the signal term originally envisioned, and is used for anti-inflammatory cytokines and chemokines. The second signal term is the first with an added term for virus. The viral term saturates for a very low level of free virus in the system, corresponding to its role in detecting small initial amounts of virus.

Next, we establish the mechanism for a baseline production of IL-10 in the system. For several cytokines and chemokines, the basal level of the cytokine can be approximated by zero in the system, as it may represent much less than 1% at the peak of the system. However, it is known that *in vivo* Target Epithelial Cells produce some IL-10 as an anti-inflammatory mediator for the respiratory tissue. In the data, this basal level of cytokine may represent between 20 and 50% of the peak value of IL-10 in the system. So, a term for baseline production of IL-10 by Target Epithelial Cells is introduced, with production modified by resistance term R which stops the cells from conducting their normal activities.

Another species for which there is a large proportion of total at baseline is Type I Interferon. Typically, these are produced by infected cells as a paracrine warning by infected cells; however, APCs produce it systemically. Further, when infected cells go to zero, interferon production continues, which as an innate immune factor insulates the system from virus

production from the remaining infected cells. Hence, a term for production of interferon proportional to the number of APCs added to the equation. Since APCs in the data peak after Type I interferon, this produces a secondary minor peak in the trajectory, so we experiment with terms for interferon use and depletion that might compensate for this. These depletion terms, though, do not work out well, and we omit them from the final model.

Along the same vein as Target Cells producing IL-10 and APCs producing interferon, we experiment with having the NKT cells produce various cytokines as occurs biologically. This did not significantly contribute to the model, and merely served to raise the number of parameters and increase the intricacy of the model. Therefore, this change was omitted in future iterations.

The model as originally formulated contains two terms for APCs, inactive and active. However, the inactive serve no other purpose other than to be activated. Further, the dynamics of the inactive in the model remain close to constant. Since there is no empirical data for the inactive APCs, and no important mechanisms to be included, the explicit activation is removed from the model and the inactive population is replaced by a constant.

The empirical data comes from the lungs of mice, only providing a small picture of the organism as a whole. While the lung is the most important organ in this disease process, it is not entirely isolated, even at baseline. We know at baseline there exist basal levels of immune cells and antibodies that diffuse freely through the lung, produced elsewhere in the organism. To account for this, we add constant terms to the production of antibodies and other cell types to represent outside sources of immune factors.

The most substantive changes made to the model come in the production of T_H1, NKT, and B cells. All three are triggered to be produced by APCs, and done as a recruitment and

maturation proportional to the number of APCs, following the preceding model. However, through our parameter searches and parameter estimation techniques, we find that these production terms do not produce the changes in scale that are needed to describe the data. Since the change in scale of APCs does not correspond to the change in scale of T_H1 or NKT cells, we instead describe the recruitment and maturation as Hill functions of APCs. This type of model works because a certain number of APCs are needed before proper MHC II presentation to the lymph nodes can result in proper maturation of T-cells; however, the hill term is a highly simplified mathematical expression for this process. Instead of expressing production of B-Cells as being proportional, in any sense, to the number APCs, we express the maturation of B-Cells as a mass action dynamic between APCs and naïve B-cells that have not been exposed to the antigens carried by the APCs. This process is then upregulated by the presence of IL-12. After the re-evaluation of the production terms for NKT and T_H1 cells, all other hill terms are re-evaluated to ensure that the hill coefficient allows for the change in scale between the causal agent and its product.

The model of 20 differential equations and 90 parameters, as described, is presented in the following section.

3.1.3 Influenza Model Equations

$$
\begin{aligned}
\Sigma_1 &= a_{11}T + a_{12}D \\
\Sigma_2 &= a_{11}T + a_{12}D + \frac{a_{21}V}{1 + a_{22}V} \\
M' &= \frac{b_{mc}C^{hill}}{a_{mc}^{hill} + C^{hill}} - \mu_M (M - b_m) \\
T' &= \frac{b_l M \Sigma_2}{\Sigma_2 + (\Sigma_2 + \frac{g_1 L + g_2}{L + d_2})(\frac{k_1 L + k_2}{L + f_1})} - \mu_l T \\
L' &= \frac{b_l M \Sigma_1}{\Sigma_1 + \frac{g_1 L + g_2}{L + d_2}} - \mu_l (L - b_{lh}(1 - R)H) \\
C' &= \frac{b_c M \Sigma_1}{\Sigma_1 + \frac{g_1 L + g_2}{L + d_2}} - \mu_c C
\end{aligned}
\tag{0.21}
$$

$$
\begin{aligned}
\hat{N}' &= \frac{b_{nt}T}{a_{nt} + a_{nl}L + T} - \mu_n \hat{N} - \frac{r_{nc}\hat{N}C}{a_{nc} + C} \\
N' &= \frac{r_{nc}\hat{N}C}{a_{nc} + C} - \mu_n N \\
X' &= \frac{b_{xn}N}{a_{xn} + N} - \mu_x X - g_{xh}HX \\
H' &= b_h D H\left(\frac{H - \theta}{H_0}\right) - \frac{g_{hx}HX^{hill}}{a_{hx}^{hill} + X^{hill}} \\
D_h' &= \frac{g_{hx}HX^{hill}}{a_{hx}^{hill} + X^{hill}} - b_h D_h H\left(\frac{H - \theta}{H_0}\right) \\
D &= D_h + D_i
\end{aligned}
$$

$$V' = g_{vi}(1-R)I - g_{vh}HV - \mu_v V - \frac{a_{v1}V}{1+a_{v2}V} - g_{va}AV$$

$$I' = g_{HV}VH - g_{iv}(1-R)I - \frac{g_{ix}IX^{hill}}{a_{ix}^{hill} + X^{hill}} - b_{ik}RIK - b_{ie}RIE$$

$$F' = b_{fi}(1-R)I - \mu_f F + b_{fp}P$$

$$K' = \frac{b_{kc}C^{hill}}{a_{kc}^{hill} + C^{hill}} - \mu_k K - b_{ki}RIK$$

$$G' = \frac{b_{go}W}{a_{go}+W}O + \frac{b_{gk}W}{a_{gk}+W}K - \mu_g G$$

$$R = \frac{F}{a_{rf}+F}$$

$$P' = P_0\left(\frac{g_{pv}V}{a_{pv}+V} + g_{pd}D_I\right)\left(g_p + \frac{b_{pg}G}{a_{pg}+G}\right) - \mu_p(P - b_p)$$

$$E' = \frac{b_{ep}P^{hill}}{a_{ep}^{hill} + P^{hill}} - b_{ei}RIE - \mu_e E$$

$$O' = \frac{b_{op}P^{hill}}{a_{op}^{hill} + P^{hill}} - \mu_o O$$

$$W' = \frac{b_{wo}O}{a_{wo}+O}P - \mu_w W$$

$$B' = b_b + b_{bp}WP(B_0 - B) - \mu_b B$$

$$A' = b_a + b_{ab}B - g_{av}AV - \mu_a A$$

3.2 MODEL RESPONSE TO PARAMETER FITTING

Whereas various proposed models for influenza infections have described qualitative behaviors of the system and used heuristics alongside viral titers to fit parameters, this parameter estimation regime is more ambitious in that it seeks to fit a model that was conceived qualitatively according to theoretical biological knowledge, and fit it to a large body of empirical data. The adult sublethal itself has data in 15 variables over 8 time points, making for 120 data sets from which the parameter estimation seeks to infer 90 parameters. The adult lethal has data in 12 variables over 6 time points, making for 72. In a parameter estimation regime that attempts to fit the model to both sublethal and lethal data simultaneously, this makes 192 data points to fit to—nearly double the number of parameters in the model. The abundance of knowledge to which we have access introduces issues unlike fitting to scarcity of data. Several types of parameter estimation approaches are used, sometimes iteratively between one and another, to slowly move towards a system that describes the qualitative and quantitative features of the system.

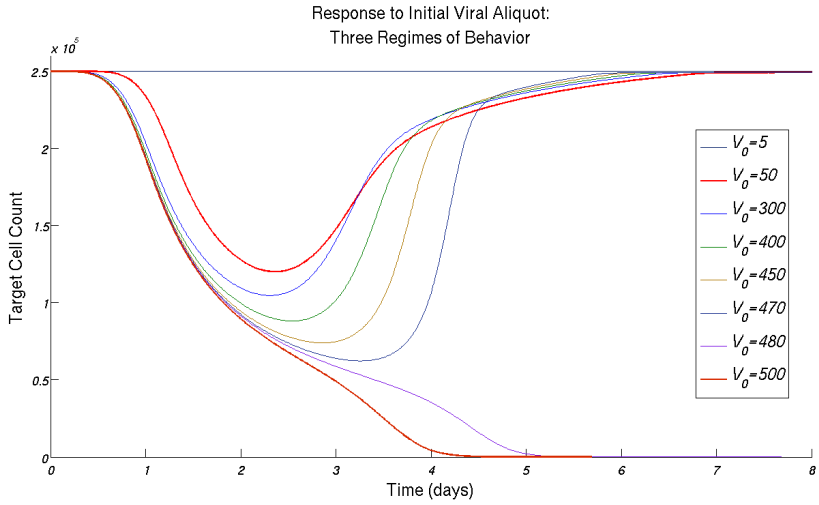

Figure 26. Model Response to Viral Aliquots

Qualitatively, the model needs to respond to initial aliquot in three different ways. For low initial virus, there should be little or no response. For the sublethal dose, 50 pfu, the population of Target Cells falls to between 50% and 25% of the total before clearing the virus and returning to baseline. For the lethal dose, 500 pfu, the first few days of the trajectory should be similar to the sublethal, but cross a threshold so as to fall into the basin of attraction for the dead state.

3.2.1 Initial Hand-Fitting

The influenza model as taken from Hancioglu et al (Hancioglu, Swigon, and Clermont 70-86) provided an initial reference point for several parameters and a suggestion of ranges. These parameters were normalized and the empirical data is not. So, an initial iteration in the parameter fitting was to rescale the parameters in the Hancioglu model to fit the peaks in the data, thus undoing any normalization. The number of epithelial cells was rescaled to be closer to the number of cells in the lungs in the experimental animals. Similarly, the trajectories taken from the inflammation model were re-scaled to fit the data. Various parameters and parameter ranges were additionally taken from the macrophage activation model presented in Day et al (Day, Friedman, and Schlesinger 11246-51). Additionally, many parameters can be directly inferred from the empirical data, for example the values of various cell and cytokine populations at the non-infected fixed point. When data is available, the substrate affinity terms are taken to be between the median and the mean of the data for the substrate they modify.

From the XPP interface (Ermentrout), we manually adjust various parameters in order to divine parameter sets that exhibit the qualitative behavior in the cells, cytokines, virus and cells that we desire. The non-infected point needs to be stable, which we get from the XPP interface. For the sublethal dose, all species are expected to rise, reach a maximum, and fall while the

target epithelial cells fall then return to the non-infected state. Importantly, for the lethal dose, the epithelial cells follow a path close to the trajectory for the sublethal dose; but where the sublethal trajectory finds its minimum, the derivative of the lethal trajectory stays negative until all target cells are depleted. As various changes to the model are employed, these qualities are examined in XPP and changes to parameters made manually to compensate for new features.

3.2.2 Genetic Algorithm and MADS

In their most general form, genetic algorithms are programs that seek to find the minimum of the likelihood estimator by producing a sequence of parameter sets by using a type of natural selection on the parameter sets. The next generation of the sequence is produced by a "parent,' where many new parameter sets are produced as "offspring" and the offspring that has the best value, where best is subjective to the algorithm, becomes the next point is the sequence. There exist many types of genetic algorithms that use various rules for producing new generations and selecting among the new generation of parameter sets. A subspecies of the genetic algorithm is the "direct search," which produces a several parameter sets near the current iteration, and chooses among all parameter sets the one that produces the lowest values. For the algorithm to evaluate parameter sets, a functional referred to as an objective function needs to be defined and entered into the program; the objective function takes the parameter values as an argument, and from that argument produces a single positive numerical value that indicates how well the parameter set describes the data. As the objective value moves closer to zero, it indicates that the parameter set better describes the data.

Thus, as the direct search proceeds, the value of the objective function is non-increasing. In MADS (mesh adaptive direct search), the mesh of parameter sets created about the current

parameter set, is produced randomly unlike the related Generalized Pattern Search (Audet and J.E.Dennis 889-903;Audet and J.E.Dennis 188-217;Abramson et al. 948-66). The strength to MADS and to other direct search methods, is that no derivative information is needed, and the objective function to optimize need not be continuous. However, since the objective function is necessarily non-increasing, the algorithm problematically becomes attracted to individual energy wells that may not solve the entire system.

Experimentally, while the convergence of the method remains as the parameter number increases, computation time and accuracy become problematic. The method was prototyped and solved on small systems as a preliminary experiment to judge the feasibility of the algorithm. It produced very good results on systems with four parameters and white noise added to the data. It produced good results, but with less accuracy on ten parameter system. For the system with 90 parameters in 20 variables, the output trajectories of the MADS parameter searches were, as a rule, constant valued and insensitive to initial conditions. Hence, MADS proved to be ill-suited to the parameter estimation problem of this model, especially since the strengths that the method brings are not important to the problem. The system is continuous and multi-modal, thus it is not a good fit.

3.2.3 Tangents and Method of Splines

In an effort to exploit the relative abundance of data, 20 variables with data for 15 in adult sublethal, we explore techniques that assume a large amount of data. For a given differential equation taken by itself from the system, if all variables in that equation have experimental data, then we should be able to fit as many parameters in that equation as we have

number of time points—which for sublethal adults is 8; this also assumes that the parameters appear only in the given equation. The missing component for this formulation is an approximation of the derivative at each measured time point. We attempt to approximate the value of the derivative at these discrete time points by fitting the data to cubic spline interpolations and using the value of the tangent of the spline at the time point for the derivative (Ahlberg, Nilson, and Walsh;Wahba). However, the spline interpolation is found using the means of the data points, and does not a priori give the variance in the derivative information. In fitting noisy data with a degree of uncertainty, it does not behoove us to assume our approximation of the derivative is more certain than empirically collected data. Ranges of tangent values were constructed by randomly picking points from the time points before and after the desired time points, using a gaussian distribution constructed from the mean and standard deviation of the experiment. In the end, this did not guarantee the same sign for all values of the tangent, let alone construct a productive range of tangent values. Lacking a scaling factor for the derivatives, the full implementation of this method could not be achieved.

But, it is assumed that the derivatives at time point zero are zero, since the system starts from the fixed point. Additionally, there is some clarity as to the time and position of the highest value for the trajectory from the experimental data, which we may assume has derivative zero, or close to zero since we do not know from the data the precise time point for the maximum. Since this method fits to data in only two points, only two parameters can be estimated from this data. Further using that the derivative on the left hand side of the equation is zero, we divide through by the decay rate μ, and fit the other normalized parameters. Half lives for various cell activations and macromolecules are typically available in the literature, so when we undo the normalization, we will multiply the approximated values by the range of the half lives. In

several of the terms for which this method is viable, there is only production and decay, so this allows us to approximate production terms, and hill terms where applicable. This technique was applied in some variant to: Macrophages (M), Natural Killer Cells (K), NK T-Cells (E), B-Cells (B), IL-12 (W), and T_H1 Cells (O). Since the solutions are algebraic, a single value was possible to obtain and use as the center for a range of possible parameter values. In some instances, the values produced algebraically were negative, in which case the equation terms were re-formatted. Hill terms were not chosen algebraically, rather by which hill term produced the lowest error when the algebraic values were used to plot the production curve against the data values for the two fit points; in general, every equation involving a Hill term had discrete values 1 through 5 separately produced. For production terms involving a Michaelis-Menten or Hill term, it's worth noting that the substrate affinities were almost uniform, and the difference came in the production term, and how steep a slope the hill coefficient produced. The lowest Hill coefficient to describe the data was chosen.

This analysis of equations and data yielded several benefits. First, it provided an a priori value of several parameters. Second, it created a frame work for evaluating the viability of the terms used within the equations. Third, it allowed us to choose the lowest Hill coefficient that describes the data. While the original frame work of fitting all equations and experimental time points to tangent values from the splines did not work in this context, being able to approximate two parameters per equation for several equations accumulated many more trustworthy priors.

3.2.4 Metropolis-Hastings

We next consider an ensemble-based approach to parameter fitting; we assume that there exists a continuum of parameter sets that fit the data equally well. We would want to sample

from this continuum, this distribution of the parameter space, using Bayesian inference. Using a Metropolis Hastings method and a uniform prior distribution over the parameter space, we sample from the posterior distribution. An existing parameter set is perturbed and then its objective function is evaluated and compared to the previous objective function; the choice to accept or reject the new parameter set is made randomly. A value u is chosen from a random uniform distribution over 0 to 1 and used in the formula:

$$\text{For u in [0,1], if the ratio } \frac{e^{-\beta\varphi_{new}}}{e^{-\beta\varphi_{old}}} \begin{cases} > \text{u, accept new set.} \\ < \text{u, repeat old set.} \end{cases} \tag{0.22}$$

This process is repeatedly over hundreds of thousands to millions of iterations. Typically, the size of the perturbations between successive parameter sets should be chosen so that the average acceptance rate is between 25% and 35%.

In an attempt to sample from the whole parameter space and not a local subspace, we use a variant of Metropolis-Hastings known as parallel tempering. Several chains, in our program 12, are run simultaneously and swap their parameter sets at scheduled intervals. Swap intervals between 10 and 100 were used during this project. The difference between the chains is a multiplier to the difference in the energies, the inverse of this multiplier being the chain's temperature. The baseline chain has temperature value one, and the temperatures rise monotonically for each additional chain; β_i is the inverse of a temperature T_i so that these values are between 0 and 1. The difference in β affects the rate at which the chains swap their values. A value u is chosen from a random uniform distribution on 0 to 1 and used to randomly choose to swap between adjacent chains.

$$\text{For u in [0,1], if the ratio } \frac{e^{-(\beta_i-\beta_{i+1})\varphi_{i+1}}}{e^{-(\beta_i-\beta_{i+1})\varphi_i}} \begin{cases} > \text{u, swap chains i and i+1} \\ < \text{u, continue chains} \end{cases} \tag{0.23}$$

Temperature differences were chosen to have a swap rate approximating one half between consecutive chains. As temperatures rise, the free energy of the method, and the level of error the likelihood estimator accepts rises, so that isolated regions of the parameter space can potentially be reached. The highest temperature should be chosen so to replicate a random sampling of the parameter space. More frequent swapping can lead to better exploration of the space, but at the same time lead to parameter subspaces being inadequately explored.

Parameter values are explored in log space. New parameter sets are produced by the exponential of a random normal distribution using the old parameter values as the means and the step-size epsilon as the standard deviations. More, the step-size is normalized for the size of the space for the individual parameter. The final model has 90 parameters, only 40 of which we have empirical or experimental knowledge of; these parameters tend to have relatively smaller prior spaces to sample from. For the remaining parameters, previous parameter fitting regimes provide a mean value that we use as the center of a parameter distribution going from $1/10^{th}$ of this mean value to 10 times, i.e. one $\log_{10}$ scale down and up. The normalization of the step size allows for very large and very small distributions to be sampled from evenly. Drawing from the distribution on a log scale, has the added benefit of allowing very large and very small values of a given parameter to be chosen on an equal basis.

Once a posterior distribution has been sample, it can happen that the mean of the posterior be statistically different from the mean of our prior distribution. Especially when the prior of the distribution is chosen teleologically and from prior fits rather than literature or experimental data, it becomes prudent to shift the center of the prior distribution and to re-sample the space. This was done many times as the space was iteratively sampled.

A practical concern in the design of the objective function was the change in scale of the experimental data. While some data change over one or two $\log_{10}$ scales, other data can change by three or four scales. If we consider the summation of data minus trajectory, then divided by standard deviation, this change in scales is going to weigh unevenly on the variables that change very little and the variables with large changes. Further than only talking about unevenly weighing between variables with large and small changes, if we consider that standard deviations are proportional to their respective means—data for very low means will be weighed more heavily than data for very large means. Hence, a minority of the total data produces the majority of the error in the objective function and the parameter fitting algorithm focuses on minimizing these errors rather than all data errors evenly. To compensate the problem of different scales, we transform the terms of the objective function to evaluate the trajectory and mean data on a log scale before subtracting the two. This term is then divided by the ratio of the standard deviation to the mean, rather than the standard deviation; this is the coefficient of variation, and it expresses the variation of the data as a relative to the mean rather than as an absolute measure (Johnson, Miller, and Freund).

$$\varphi(p) = \frac{1}{2\pi} \sum_{i=1}^{\#\text{ data points}} \frac{log_{10}(trajectory_i(p)) - log_{10}(mean\ data_i)}{(standard\ error_i)/(mean\ data_i)} + heuristics \quad (0.24)$$

Finally, the problem comes to us of finding an ensemble of parameters that achieves a multi-objective fit, that is, ones that works for both doses of initial virus, 50 pfu (sublethal) and 500 pfu (lethal). The first strategy is to fit to both single objectives, and to sub-select from each ensemble a collection of parameter sets that most closely fit to the other objective. This strategy, however, is not guaranteed to produce any viable sub-sets from the ensembles. The alternative strategy is to produce a multi-objective likelihood function, derived from the values of the

likelihood function of the sublethal data set and the likelihood function of the lethal data set. The two functions we used were the ℓ_2-norm of the two likelihood functions and the product of the likelihood functions. In theory, the ℓ_2-norm, once equilibrated, should explore the space of the two likelihood outputs as a semi-circle, as a semi-circle will provide conservation of energy as a new parameter fit that is equally likely shifts the energy from one likelihood function to the other. The downside to this formulation is that the equilibrated energy function is concave and could lead to parameter regimes that fit both objective functions equally badly. On the other hand, the product produces a slant hyperbola that will explore equal energies; this formulation is convex, so there is theoretically a point along the slant hyperbola closest to the origin. However, if the space of energy outputs that we sample from does not produce a convex set (which we do not know a priori), then the function may explore extremals rather than along the continuum from sublethal fits to lethal fits. A third consideration for functions of the two likelihoods is the max function, which would use the maximum of the two objective functions; this is a more severe version of the slant hyperbola, which could produce a better output if the parameter sets exist as a convex set to be explored, but is more likely if the set is not convex to become trapped on one side of the extremals. Should the slant hyperbola function demonstrate that a convex set of energy values does exist, this function would be useful to further explore the space.

3.3 MODEL AND DATA

In the qualitative fitting of the model, the focus was on a few features of the model that would make it amenable to expected data. For the sublethal, trajectories reach their peak then proceed back to a stable baseline state. For lethal, the trajectory of target cells proceeds to zero

and leads the system to a stable attractor. From here, the model must be fit to data, and the inverse problem addressed formally with the goal of obtaining an ensemble of parameter sets that quantitatively and qualitatively describe the system.

Data was collected from balb/c mice, aged 12-16 months, inoculated with influenza A/8/PR/34. For the mice given a sublethal dosage of the inoculum, data was collected for 15 or our 20 variables over a course of 19 days with a minimum of three mice harvested per time point. Data was collected on days: 0, 1, 2, 3, 5, 7, 9, 11, 15, and 19; because the viral infection clears before day 9 and is gone by day 11, days 15 and 19 are not included in the fitting.

For the mice given a lethal dosage of the inoculum, data was collected for 12 or our 20 variables over a course of 7 days with a minimum of three mice harvested per time point. Data was collected on days: 0, 1, 2, 3, 5, and 7. After day 7 there was only one surviving mouse which was close to death; the high mortality made data collection difficult. The lethal data lacks Antibodies, Macrophages, and Antigen Presenting cells, which are available in the sublethal. Information about the variables and their corresponding measurables is summarized in Table 4.

For the purposes of fitting qualitative features, some heuristics were added to the likelihood function. For the sublethal, we require that all Target Cell trajectories stay above 20% of the total cells count; correspondingly, for the lethal trajectory is required to have all Target Cell trajectories be below 20% for days 5 and beyond. Additionally, the lethal trajectory is integrated to 11 days, and all days 5 and beyond are required to be below this threshold by a smooth heaviside function, and be monotone decreasing. Since the dead state is attracting, a parameter set that abides by these conditions should go this attractor. For the lethal trajectory, a heuristic is placed on the antibodies to be suppressed for the first three days, which corresponds to the sublethal data.

Label	Variable	Units	Measurable	Measurement
TNF-α	T	pg/mL	TNF-α	Luminex
IL-10	L	pg/mL	IL-10	Luminex
Chemokines	C	pg/mL	MCP and MIP-1β	Luminex
Macrophages	M	cell count	CD11c- CD11b+ CD40+ Gr-1	Flow
Blood Neutrophils	Ñ	cells count	None	N/A
Tissue Neutrophils	N	cell count	Gr-1+ (high) CD11b+ (high)	Flow
NOS	X	pg/mL	None	N/A
Target Epithelial Cells	H	cell count	Heuristic	Heuristic
Infected Epithelial Cells	I	cell count	None	N/A
Damaged Epithelial Cells	DH	cell count	None	Heuristic
Virus	V	pfu/mL	Influenza A/PR/8/34	Plaque Assay
Type I Interferon	F	pg/mL	IFN-α and IFN-β	ELISA
Type II Interferon	G	pg/mL	IFN-γ	Luminex
Natural Killer Cells	K	cell count	CD49b(DX5)+ CD69+	Flow
Antigen Presenting Cells	P	cell count	CD11c+ CD11b+ CD40+ Gr-1	Flow
B-Cells	B	cell count	CD19+ CD69+	Flow
NKT-Cells	E	cell count	CD3+ CD8+ CD69+	Flow
IL-12	W	pg/mL	IL-12	Luminex
T_H1 Helper Cells	O	cell count	CD3+ CD4+ CD69+	Flow
Antibodies	A	pg/mL	I_GM Antibodies	HAI

Table 4. Variables and Measurables.

We summarize the biological labels we use, the corresponding variable, the units of measurement, the precise measurable, and what kind of test was used to collect the data.

For both trajectories, a heuristic is placed on Damaged Target cells to be proportional to the logarithm of the Neutrophil data; Neutrophils are the main agent in the inflammatory damage, and it is their saturation that causes inflammatory damage, hence the logarithmic term.

3.4 METROPOLIS-HASTINGS

The model employs many non-linear terms, which are necessary for describing saturation phenomenon of chemical signalling, precluding many commonly used parameter estimation methods. Thus, the parameter fitting regime that we choose, a Metropolis-Hastings method, takes into account only the objective value produced by a given parameter set to evaluate its efficacy, and thus side-steps the difficulties typically encountered when fitting to nonlinear terms. The Metropolis-Hastings method produces an ensemble that can then be used to examine the posterior distribution of parameters. For a sufficiently large number of iterations, the distribution of values for each parameter theoretically approximates the marginal probability distribution of the parameter. Each parameter set represents a variation of the model that is equally likely in a statistical sense, as each parameter produces approximately the same energy once the Markov-Chain has equilibrated.

The Metropolis-Hastings method is computationally intense, especially to run a sufficient number of samples so as to properly explore the whole range. As the equations will be solved several million times, we employ every feasible technique to speed up computation. Additionally, the influenza ODE system is very stiff and requires a stiff solver. We use CVODE to solve the system, as provided through SUNDIALS (Hindmarsh et al. 363-96), using a Backwards Difference schema of order 5 with a Newton Iteration and user-provided Jacobian (Ascher and Petzold). To speed up the computation, the system is re-ordered to have minimum possible bandwidth; LAPack Libraries are employed for the inversion of the Jacobian.

However, for practical considerations for the Metropolis-Hastings method, bounds on the parameters must be chosen, temperatures for the parallel chains chosen, and step-sizes chosen. The temperatures should be that the swap rate between chains be close to 50%, and the step-sizes

chosen so that equilibration the acceptance rate is between 25% and 35%. Initial parameter sets are chosen randomly, then allowed to run fixed length, and the results evaluated so that the temperatures and step-sizes are updated in accord with the results of the statistical data for these short runs. The distributions of these short runs were used to update the parameter ranges for those for which there is not data from the literature; the final ranges of the parameters used is summarized in Table 5. New parameters are chosen from a pseudo-random uniform distribution, with the random numbers seeded by the time of the start of the algorithm. The choice of the new parameters occurs on a log scale, so that small values are chosen with equal probability as large values. For parameters with unknown bounds, the mean of prior distributions is used as a center, and the space explored one log10 scale up and one down. Since the priors are not evenly spaced, to ensure even travel through parameter space, we multiply the step-size by the ratio of the upper-bound to the lower bound for individual parameters. In general, only the step-size for the lowest temperature chain was chosen by hand, and the step-sizes for the higher temperatures chain proceed as the step-size of the previous temperature chain times the square root of the ratio of the previous temperature to the current one.

Successive burn-in chains are run from the last parameter set until three successive runs have the desired swap and acceptance rates. This is done for a likelihood function measuring trajectories against the sublethal data, the result of which is referred to as the sublethal ensemble. And the result of the likelihood function measuring to lethal data is referred to as the lethal ensemble. To search for parameter sets that fit against both sets of data equally well, we consider a function of the two likelihoods; the result of using the ℓ_2-norm of the two likelihood function we refer to as the multi-objective ensemble. For each ensemble, we run 4 instances, i.e. repetitions, of the Metropolis Hastings method, each with 5 million iterations, swapping between

adjacent temperature chains every 100 iterations. So, each ensemble contains 20 million parameter sets.

Name	Units	Description	Range	Figure 26
α_{11}	signal cytokine^{-1}	Signal induced to Macrophages by TNF-α	[1.1E-4, 1.1E-2]	0.000317
α_{12}	signal cell^{-1}	Signal induced to Macrophages by Damaged Epithelial Cells	[8E-6, 8E-4]	3.64e-05
α_{21}	signal	Maximal signal induced to Macrophages by Virus	[3.20, 320]	226
α_{22}	virus	Substrate affinity for signal induced by Virus	[3E+3, 3E+5]	33600
γ_1	signal	Inhibitory term for effect of IL-10 on cytokine production	[0.75, 75]	5.72
γ_2	signal cytokines	Inhibitory term for effect of IL-10 on cytokine production	[670, 67000]	1100
δ_2	cytokines	Inhibitory term for effect of IL-10 on cytokine production	[50, 200]	152
β_{mc}	cells day^{-1}	Maximal chemotactic adduction of Macrophages	[1500, 20500]	8930
α_{mc}	cytokines	Substrate affinity for adduction of Macrophages	[400, 1000]	414
μ_m	day^{-1}	Decay/removal of macrophages	[0.02, .25]	0.0225
β_m	cells	Baseline number of Macrophages	[400, 5600]	4750
β_t	cytokines day^{-1}	Maximal production rate of TNF-α	[0.052, 5.2]	0.809
ξ_1	dimensionless	Inhibitory term for pro-inflammatory cytokine production	[0.5, 10]	0.638
ξ_2	cytokines	Inhibitory term for pro-inflammatory cytokine production	[50, 200]	75.8
δ_1	cytokines	Inhibitory term for pro-inflammatory cytokine production	[50, 200]	134
μ_t	day^{-1}	Decay/removal of TNF	[50, 360]	253
β_l	cytokines day^{-1}	Maximal production rate of IL-10 by Macrophages	[0.375, 37.5]	0.782
μ_l	day^{-1}	Decay/removal of IL-10	[3.5, 6.2]	5.66
β_{lh}	cytokines cell^{-1}	Production rate of IL-10 by Target Epithelial Cells	[1e-4, 5e-4]	0.000307
β_c	cytokines day^{-1}	Maximal production rate of Chemokines	[25, 2500]	31.8
μ_c	day^{-1}	Decay/removal of Chemokines	[30, 90]	33.8
β_{nt}	cells day^{-1}	Maximal activation rate of Neutrophils by TNF-α	[1E+4, 1E+6]	745000
α_{nt}	cytokines	Substrate affinity for activation of Neutrophils	[10 , 80]	49.7
α_{nl}	dimensionless	Inhibitory effect of IL-10 on the activation of Neutrophils	[0.05, 0.5]	0.25
γ_{nc}	day^{-1}	Maximal chemotactic adduction of Neutrophils	[42, 4200]	297
α_{nc}	cytokines	Substrate affinity for adduction of Neutrophils	[35, 3500]	101
μ_n	day^{-1}	Decay/removal of Neutrophils	[0.2, 1.5]	1.01

Table 5. Influenza System Parameter Table

Name	Units	Description	Range	Figure 26
β_{xn}	signal day^{-1}	Maximal production rate of NOS	[3.00, 300]	26.1
α_{xn}	cells	Substrate affinity for production of NOS by Neutrophils	[40, 4000]	1530
γ_{xi}	$cell^{-1}$ day^{-1}	Removal of NOS during Infected Cell destruction	[3E-7, 3E-5]	4.16e-06
γ_{xh}	$cell^{-1}$ day^{-1}	Removal of NOS during Epithelial Cell destruction	[3.0E-7, 3.0E-5]	5.65e-07
μ_x	day^{-1}	Decay/removal of NOS	[1, 60]	8.18
β_h	day^{-1} $cell^{-1}$	Replication rate of Epithelial Cells	[0.001, 0.1]	0.0053
θ	cells	Strong Allee term for the replication of Epithelial Cells	[12500, 37500]	25500
γ_{hv}	$virus^{-1}$ day^{-1}	Viral infection rate	[3E-7, 3E-5]	3.85e-06
γ_{hx}	cells day^{-1}	Maximal destruction rate of Epithelial Cells by NOS	[0.5, 50]	14.9
α_{hx}	signal	Substrate affinity for destruction of Epithelial Cells	[0.2, 20]	0.384
γ_{ix}	cells day^{-1}	Maximal destruction rate of Infected Cells by NOS	[0.5, 50]	0.821
α_{ix}	signal	Substrate affinity for destruction of Infected Cells	[0.15, 15]	0.76
γ_{ik}	$cell^{-1}$ day^{-1}	Removal rate of Infected Cells by NK Cells	[5.0E-6, 5.0E-4]	1.21e-05
γ_{ie}	$cell^{-1}$ day^{-1}	Removal rate of Infected Cells by Effector Cells	[1.0e-5, 1.0e-3]	9.76e-05
μ_i	day^{-1}	Decay/removal of Infected Cells	[0.5, 2]	1.9
γ_{vi}	virus $cell^{-1}$ day^{-1}	Production rate of Virus by Infected Cells	[14, 1400]	137
γ_{vh}	$cell^{-1}$ day-1	Removal of Virus during infection of Epithelial Cells	[2.7E-7, 2.7E-5]	2.11e-05
γ_{va}	$antibody^{-1}$ day^{-1}	Elimination of virus due to Antibody neutralization	[1.5E-4, 1.5E-2]	0.00063
γ_v	day^{-1}	Removal rate of sub-threshold Viral quantities	[0.90, 90]	8.88
α_v	$virus^{-1}$	Inverse of lowest viral level capable of infection	[0.00425, 0.425]	0.02
μ_v	day^{-1}	Decay/removal of Virus	[1, 6]	2.04
β_{fi}	interferons $cell^{-1}$	Production rate of IFN-α/β by Infected Cells	[0.054, 5.4]	2.14
β_{fp}	interferons $cell^{-1}$	Production rate of IFN-α/β by APC	[9.3E-4, 9.3E-2]	0.0477
γ_{fi}	$cell^{-1}$ day^{-1}	Excess absorption rate of IFN-α/β by Infected Cells	[6.35E-4, 0.0635	0.0103
μ_f	day^{-1}	Decay/removal of IFN α/β	[2, 40]	32.4
α_{rf}	interferons	Substrate affinity of epithelial cells to IFN-α/β	[20, 70]	24.9

Table 5 (continued) Influenza System Parameter Table

Name	Units	Description	Range	Figure 26
β_k	cells	Baseline number of NK Cells	[220, 2300]	1380
β_{kc}	cells day^{-1}	Maximal chemotactic adduction of NK Cells	[6E+3, 6E+4]	43300
α_{kc}	cytokines	Substrate affinity for adduction of NK Cells	[400, 1000]	966
γ_{ki}	cell^{-1} day^{-1}	Removal of NK Cells during Infected Cell elimination	[1.9E-8, 1.9E-6]	1.1e-07
μ_k	day^{-1}	Decay/removal of NK Cells	[0.07, 0.6]	0.489
β_{go}	interferons cell^{-1}	Maximal production rate of IFN-γ by Th1-Cells	[0.0014, 0.14]	0.0024
α_{go}	cytokines	Substrate affinity in production of IFN-γ by Th1-Cells	[1.5, 150]	7.98
β_{gk}	interferons cell^{-1}	Maximal production rate of IFN-γ by NK Cells	[0.29, 29]	0.42
α_{gk}	cytokines	Substrate affinity in production of IFN-γ by NK-Cells	[1.5, 150]	92.5
μ_g	day^{-1}	Decay/removal of IFN-γ	[2, 40]	27.7
P_0	cells day^{-1} signal^{-1}	Inactive APC available for activation	[2800, 2.8E5]	20300
γ_{pv}	signal virus^{-1}	Maximal signal for APC from Virus	[0.2, 20]	1.25
α_{pv}	virus	Substrate affinity for signal from Virus	[3000, 300000]	3960
γ_{pi}	signal cell^{-1}	Signal for APC from Dead Infected Cells	[9.5E-6, 9.5E-4]	8.75e-05
γ_p	activation	Non-specific activation rate of APC	[0.005, 0.5]	0.0493
β_{pg}	activation interferon^{-}	Maximal activation rate of APC by IFN-γ	[0.34, 34]	0.643
α_{pg}	interferons	Substrate affinity for activation of APC by IFN-γ	[90, 550]	164
μ_p	day^{-1}	Decay/removal of APC	[0.1, 0.45]	0.194
β_p	cells	Baseline number of activated APC	[200, 1200]	972
β_{ep}	cells day^{-1}	Maximal activation rate of Effector Cells	[2E+4, 2E+5]	59900
α_{ep}	cells	Substrate affinity in the activation of Effector Cells	[3E+3, 3E+4]	6630
β_{ei}	cell^{-1} day^{-1}	Removal of Effector Cells during Infected Cell elimination	[1.5E-7, 1.5E-5]	6.29e-06
μ_e	day^{-1}	Decay/removal of Effector Cells	[0.2, 0.35]	0.239
β_{op}	cells day^{-1}	Maximal activation rate of Th1 Cells	[5E+4, 1.5E+5]	109000
α_{op}	cells	Substrate affinity in the activation of Th1 Cells	[3E+3, 3E+4]	4360
μ_o	day^{-1}	Decay/removal of Th1 Cells	[0.2, 0.35]	0.263

Table 5. (continued) Influenza System Parameter Table

Name	Units	Description	Range	Figure 26
β_{wo}	cytokines cell^{-1} day^{-1}	Maximal production rate of IL-12	[0.0075, 0.075]	0.0676
α_{wo}	cells	Substrate affinity in the production of IL-12	[1.5E+4, 1.5E+5]	16400
μ_w	day^{-1}	Decay/removal of IL-12	[1.0, 5.0]	4.73
β_b	cells day^{-1}	Non-specific activation of B-Cells	[10, 1000]	94.9
β_{bp}	cell^{-1} cytokine^{-1} day^{-1}	APC induced activation of B-Cells	[1.15E-7, 1.15E-5	2.04e-6
B_0	cells	Reservoir Number of unactivated B-Cells	[3E+4, 1.75E+5]	53000
μ_b	day^{-1}	Decay/removal of B Cells	[0.1, 0.4]	0.276
β_a	antibodies day^{-1}	Non-specific production of Antibodies	[0.0023, 0.23]	0.0035
β_{ab}	antibodies cell^{-1} day^{-1}	B-Cell production of Antibodies	[0.002, 0.2]	0.0131
γ_{av}	Virus^{-1} day^{-1}	Removal of Antibodies during Virus elimination	[3.25E-6, 3.25E-4	1.0e-05
μ_a	day^{-1}	Decay/removal of Antibodies	[0.04, 2]	0.627
H_0	cells	Total number of Epithelial Cells	2.50E+005	2.50E+005
h_m	dimensionless	Hill coefficient for the recruitment of Macrophages	3	3
h_x	dimensionless	Hill coefficient for the inflammatory removal of epithelial cells	2	2
h_e	dimensionless	Hill coefficient for the maturation of NKT-cells	2	2
h_o	dimensionless	Hill coefficient for the maturation of T_H1 cells	2	2

Table 5. (continued) Influenza System Parameter Table

3.5 METROPOLIS ENSEMBLE RESULTS

From the Metropolis-Hastings, we obtain three ensembles, each with four instances of 5 million iterations each that sample from the posterior distribution. We now evaluate the convergence of these distributions, ostensibly by showing that the Metrpolis method failed to diverge. For the sublethal and multi-objective, each instance failed to diverge by the Geweke test, while a few distributions in the lethal instances did diverge. We use the Gelman-Rubin test

to evaluate if the four instances we used each converge to the same distribution. Again, sublethal and multi-objective failed to diverge, while lethal diverged in nearly half of the parameters.

The histograms in Figure 27, Figure 28, and Figure 29 show the posterior distributions that the Metropolis method sampled from. The sublethal ensemble appears almost ideal, with many of the statistically sampled distributions approximating a smooth function, the marginal distribution parameter wise; further, there are many broad, clearly defined peaks, indicating the system is not overly sensitive to parameter perturbation. The lethal histograms, by contrast, has very sharp defined peaks and rough boundaries to its distributions, suggesting that it was perhaps not sampled thoroughly enough; however, an additional 8 million iterations were run but did not improve convergence or the histogram. The multi-objective histogram shows near uniform distributions, with only a few parameters showing statistical bias towards a certain value, indicating that these parameters are generally insensitive to change.

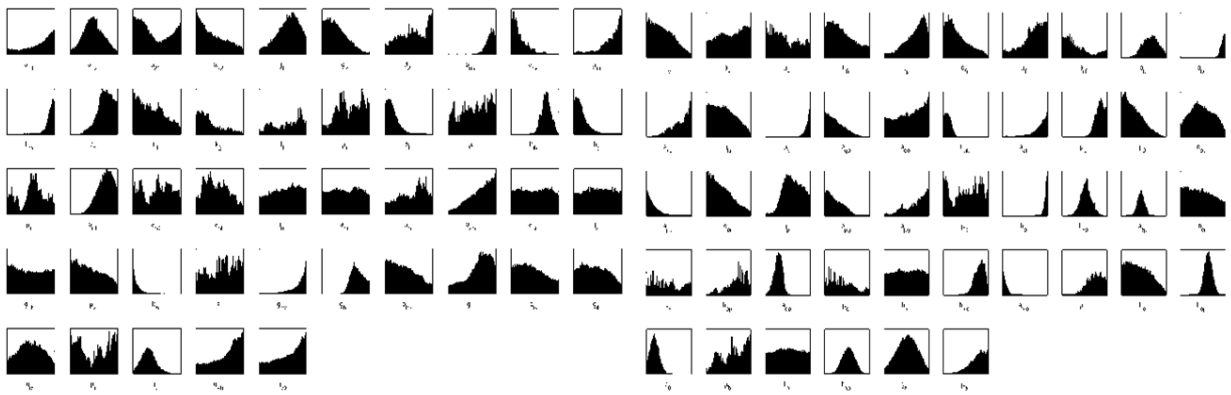

Figure 27. Histograms of Sublethal Ensemble

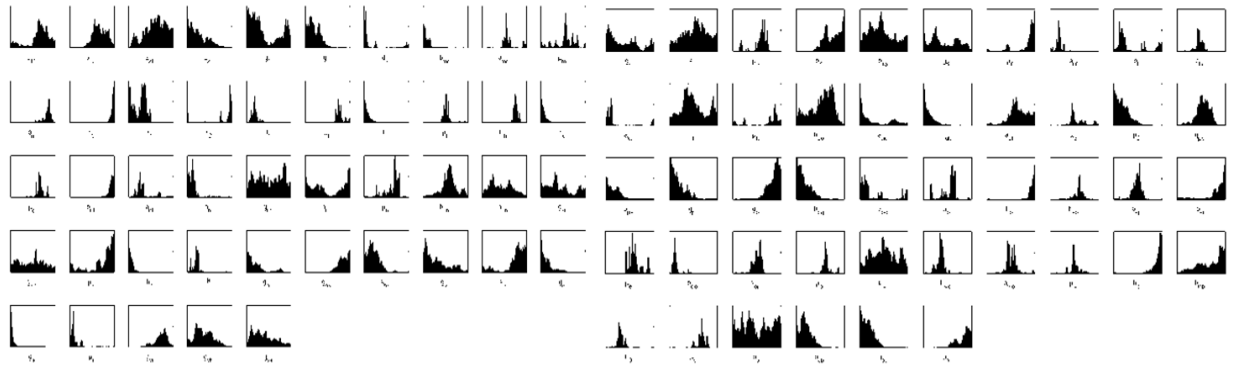

Figure 28. Histograms of Lethal Ensemble

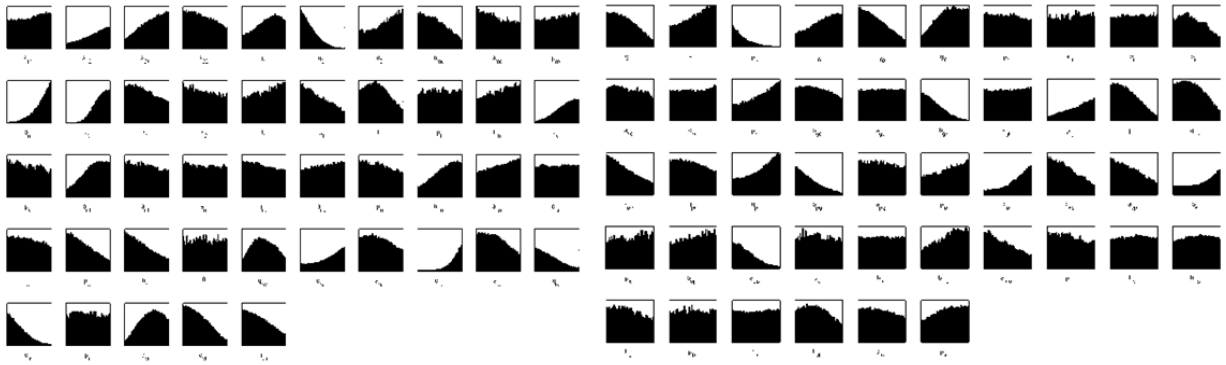

Figure 29. Histograms of Multi-Objective, l_2-norm, Ensemble

We next look at the trajectories produced by the ensembles compared against data. Since each ensemble contains 20 million trajectories, rather than present them simultaneously or arbitrarily pick the best trajectory, we represent the ensemble as a whole by presenting the quartiles of the trajectories. The solid black line represents the median value of all trajectories at a given time point, while the dark grey encompasses the central 50% of all trajectories and the light grey the central 90% of all trajectories. The blue error bars represent the data that we fit to, using the mean and standard error.

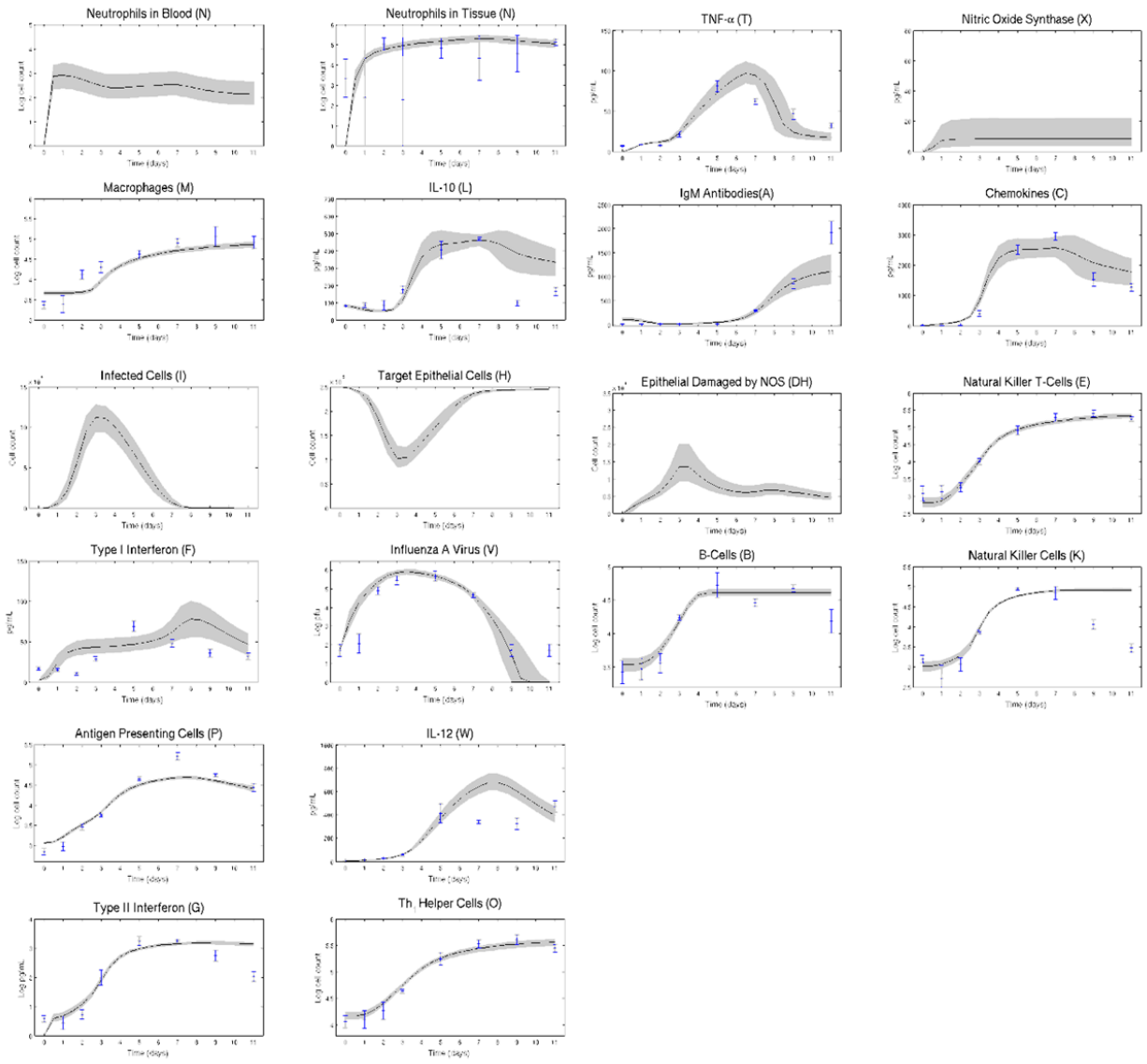

Figure 30. Quartiles of Sublethal Ensemble

As seen in Figure 30 and Figure 31, both the sublethal and lethal ensembles produce graphs that fit well to their respective data. With the exception of Type I interferon and Natural Killer Cells, the sublethal fits to the data very well. Despite problems with convergence of the distribution, the lethal also fits very well, with the exception of IL-10. From these greater ensembles, we choose to sub-select ensembles that we consider to be valid by certain metrics.

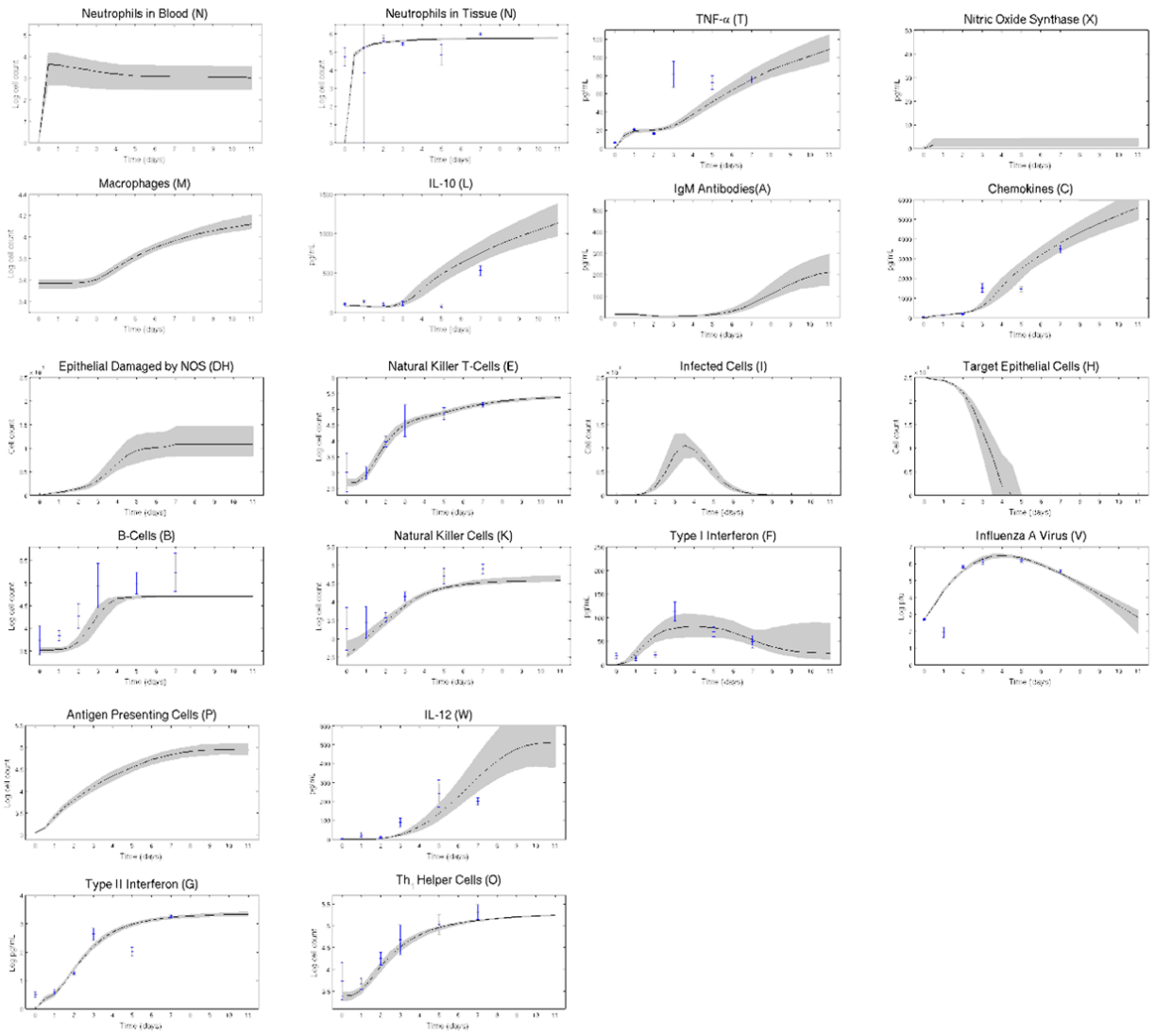

Figure 31. Quartiles of Lethal Ensemble

One criterion by which we choose to validate a parameter set is to evaluate if the baseline, uninfected state is asymptotically stable. So, the Jacobian for each parameter set around the fixed point is computed and tested. From the results, we see that very few, approx 0.3%, of the sublethal are valid by this measure, however they produce quartile plots nearly identical to those of the whole ensemble and have a distribution of likelihood values statistically similar to the entire ensemble. Results are similar for the lethal ensemble, with 0.5% stable at baseline. However, the structure of the model does allow for some parameter sets to stabilize

fixed points near the baseline state, and thus return to some basal state where small amounts of pro-inflammatory cytokines are produced. Hence, we acknowledge that most parameter sets do not give baseline stability, but we our validation only requires that the attractor is arbitrarily close to the baseline if not the baseline state itself. More importantly, the goal to find parameter sets that produce the unstable saddle that separates sublethal trajectories from lethal trajectories in the target cell population drives our inquiries of validity. So, in Figure 32, we show the quartiles for Target Cells and Virus when the lethal ensemble is given sublethal initial conditions and data, and mutis mutandis for the sublethal ensemble.

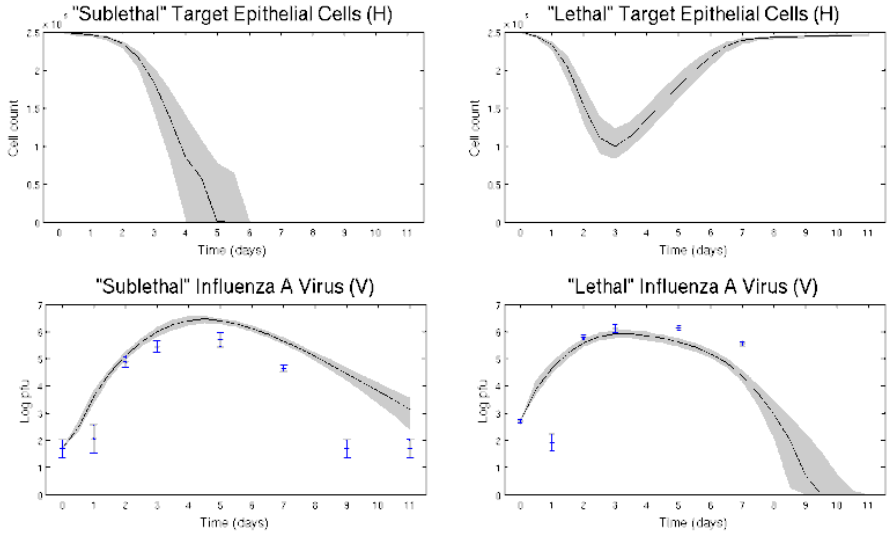

Figure 32. Comparing Sublethal and Lethal:

In the left hand column, the lethal ensemble is graphed with the sublethal initial conditions and data. In the right hand column, the sublethal ensemble is graphs with the lethal initial conditions and data. The sublethal dies and the lethal survives, which is counter-indicative, and shows we are unlikely to find a valid subset of parameter sets that contains the qualitative features of both from either ensemble.

The results shown in Figure 32 do not bode well for finding parameter sets that work qualitatively well for both sublethal and lethal trajectories. The lethal ensemble shows more

variation in its response to sublethal initial conditions, which may be useful in further iterations of parameter fitting.

As the sublethal and lethal ensembles do not give us the qualitative features we seek, we investigate the results of the multi-objective ensemble. In Figure 33, we see the ensemble plotted in Target Cells and Virus, as in Figure 32, with sublethal and lethal initial conditions on the left and right, respectively; but this time using the same ensemble for both columns.

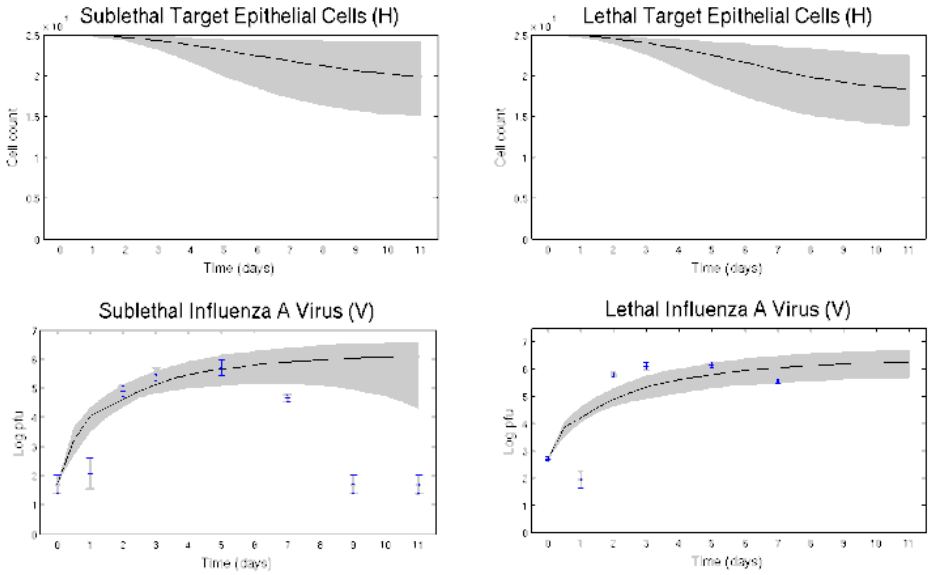

Figure 33. l_2 Multi-Objective Ensemble Results.

Target cells and Virus from the multi-objective ensemble are shown here with initial conditions and data for the sublethal on the left and lethal on the right. The differences between the two sets are marginal. The lethal does not go to the death state, and in the sublethal most trajectories do not show viral clearance.

And from the graphs we see the qualitative features of the model are not reproduced for either sublethal or lethal initial conditions. In both, it seems plausible that an intermediate fixed point stabilizes and that virus becomes fixed in the system. The sublethal does not return to health, and the lethal does not die. In the sublethal, less than 25% even show viral clearance.

The norm was chosen to represent the energies so that the graphs fit equally well, and the result seems to be an ensemble where the trajectories fit equally poorly.

To gain insight into this phenomenon, for each parameter set in all three ensembles we associate an ordered pair, with the first value being that of the sublethal likelihood function and the second value being that of the lethal likelihood function. We then plot all these ordered pairs on a loglog plot, with the sublethal ensemble in red, the lethal ensemble in green, and the multi-objective ensemble in yellow, as shown in Figure 34.

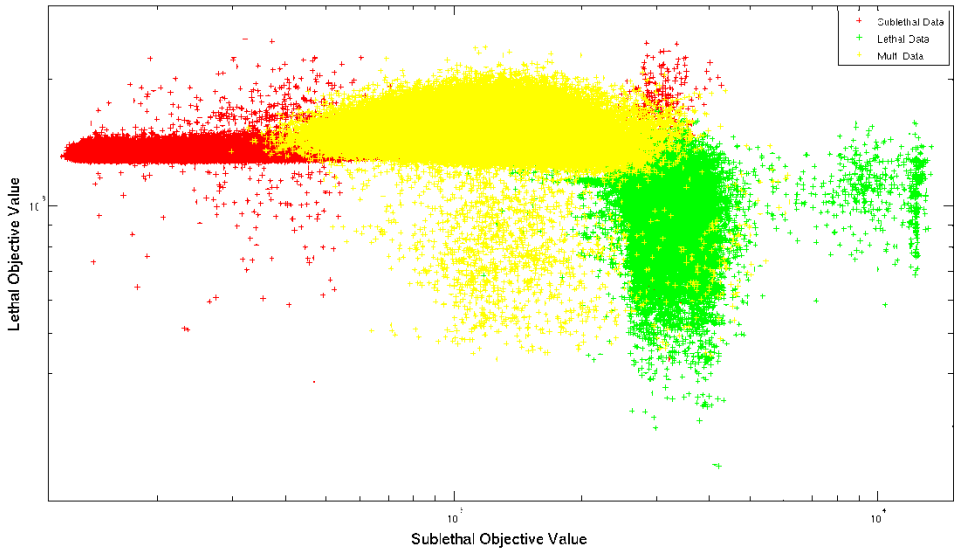

Figure 34. Log Scale Comparison of Objective Function Values, l_2 norm

As we might anticipate from the comparison of quartiles, the sublethal has a robust region of values parallel to its axis, but a very thin region of values for the lethal trajectories. The lethal ensemble is similar, with somewhat greater variation for sublethal values. Interestingly, the l_2 multi-objective ensemble does not cluster in the area directly between them, although there are some points, rather it connects the two regions on the outside.

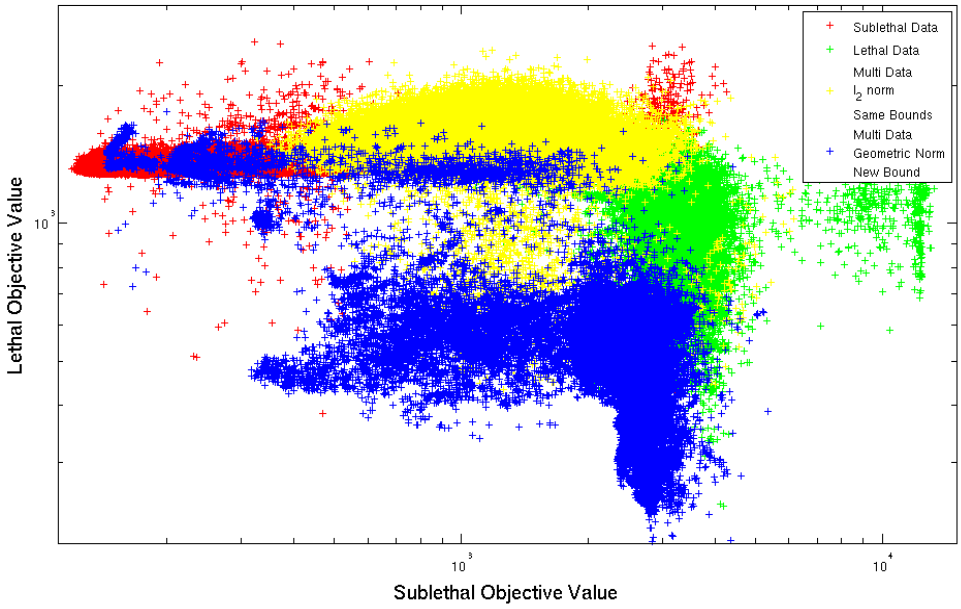

Figure 35. Log Scale Comparison of Objective Function Values, Geometric mean

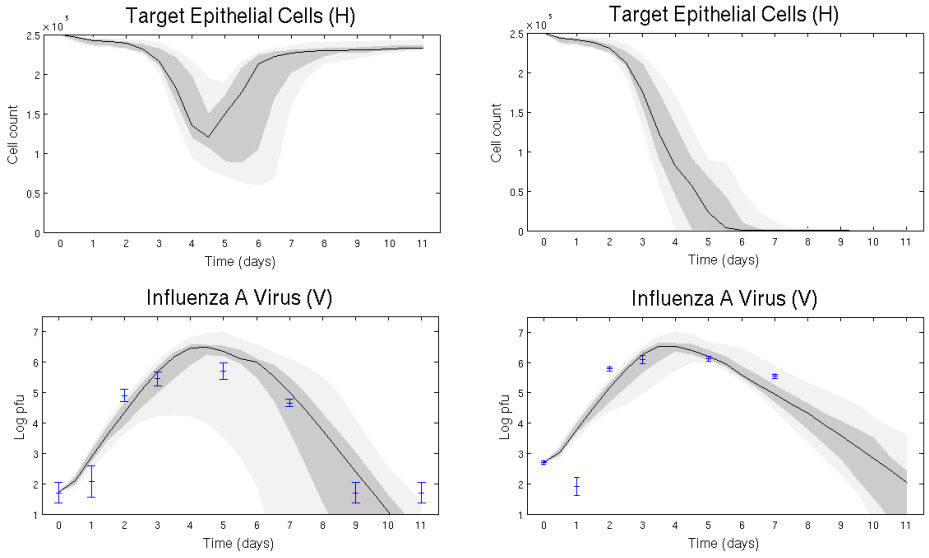

Figure 36. Geometric Multi-Objective Ensemble Results.

The results of the geometric mean give the quantitative response we want to see in terms of data with the virus, and the qualitative results as well in terms of recovery for the sublethal (left-hand column) and death for the lethal (right hand column).

n=number of time-data points for obj fxn 1

m=number of time-data points for obj fxn 2

$$l_2(\varphi_1,\varphi_2)=\sqrt{\left(\frac{\varphi_1}{n}\right)^2+\left(\frac{\varphi_2}{m}\right)^2} \qquad (0.25)$$

$$G(\varphi_1,\varphi_2)=\sqrt{\varphi_1^{\frac{\max(n,m)}{n}}\varphi_2^{\frac{\max(n,m)}{m}}}$$

In the yellow cloud of Figure 34, we see trajectories that fit neither scenario and a large unexplored white space between the sublethal and lethal clouds. At equilibrium, a metropolis-hastings method stays at its baseline energy as it explores parameter space, meaning this can be treated like a constant. The l_2-norm was thus set equal to a constant, which in the upper right hand quadrant graphs an ellipse; which is essentially the result we see in that figure. However, we want a continuous graph that takes values from the red to the green clouds; hence, we choose a slant hyperbola for its shape. This is morally equivalent to the geometric mean. The mean is weight according to number of data points in each objective function, and then run in place of the prior multi-objective function to produce Figure 35, which while more disorderly by nature yields data points substantially closer to the origin. After validating the results, we find substantially improved fits for the trajectories, as seen in Figure 36.

3.6 DISCUSSION

Parameter estimation, model identification, and validation are iterative steps in the mathematical modeling process that address the ill-posed problem of modeling living systems by mathematical constructs. In this model, we sought to begin from a position of qualitative

features and move, both through model evaluation and parameter estimation, to a position of quantitative accuracy and validity. While we have successfully fit both the sublethal and lethal scenarios, we have yet to fit both simultaneously. More worrying, in improving our quantitative fits to data, we lose the qualitative features that allows dose dependent responses. But we now know, that in the parameter space, such parameter sets that allow for the unstable saddle separating responses of sublethal and lethal aliquots are sparse, and that adjustment of the objective function is necessary to weigh more heavily towards parameter sets that produce these results.

Interestingly, while the sublethal and multi-objective ensembles passed the Gelman-Rubin test and the Geweke test, they gave much less interesting results than the lethal ensemble, which had a small subset of parameters that failed both tests. While the lethal ensemble fit its native scenario well, it proved to be robust in its response to the alternate scenario. We also see in Figure 34 that the lethal ensemble (green) has a wider range of values to the sublethal objective function than vice versa. This highlights one of the downsides to statistical sampling. The sampling for the lethal ensemble diverged by the available tests, but provided a more robust set of parameters and system responses for us to examine. The method is sampling likelihoods, not giving a robust picture of the various behaviors the model is capable of. For that, we would need a global sensitivity analysis. However, we can surmise that had the lethal ensemble resulted in a distribution that failed to diverge, we would see more of the rigidness and lack of response to change in initial conditions that we see in the sublethal and multi-objective ensembles.

Another downside of the Metropolis method is demonstrated in the multi-objective ensemble. Even though it is apparent that there are lower energy values the multi-objective

function can take, that is, closer to the origin in Figure 34, the method travels mainly along the ellipse formed by the ℓ_2-norm. This is because, unlike the genetic algorithm, this method seeks out the lowest average energies that it can travel along, not the lowest global energy. We try to account for possible disconnections between parameter regions by employing the parallel tempering. Theoretically, the method should have traveled along the lowest energy ellipse that was available to it, but instead it ran along the ellipse connecting the sublethal and lethal ensembles. Part of what makes the magnitude of the ellipse so large is the increased uncertainty that comes from joining the uncertainty of the measurements from the sublethal data and the lethal data. We see in the multi-objective histograms that most marginal distributions are near uniform, even though the acceptance rate of the method is ideal (about 30%). With the free energy so high, the selection pressure eases and the whole space is sampled near uniformly because a change in a certain parameter can be offset easily by other changes (parameter correlations) without making a major difference to the level of equilibrated energy. Experimentally showing the importance of the functional form of the multi-objective function, and taking the sublethal and lethal ensembles as possible constrains of this function, future objective functions should be chosen that are convex, or at least not concave. Employing a weighted geometric mean, we are able to see the energy of the Metropolis method travel along a trajectory more amenable to our purposes. This characteristic to follow the graph is intuitive though non-obvious.

A final, practical issue for the use of the Metropolis method: When equilibrated, the step-size of the ensemble for lethal is an order of 10 less than the step-size for sublethal. This speaks to the sensitivity of the lethal objective function in response to parameter value changes. This introduces difficulty into the multi-objective likelihood function however, because the

component of the function derived from the lethal function will be more sensitive to changes than the sublethal function. Thus, if a parameter perturbation causes dramatically more energy change in the lethal component than the sublethal component, then that change will likely be rejected, thus giving less exploration to the lethal space. This could lead to the multi-objective function better fitting the sublethal data.

3.7 CONCLUSIONS:

While there are many models available for influenza (Baccam et al. 7590-99;Hancioglu, Swigon, and Clermont 70-86;Miao et al. 6687-98;Saenz et al. 3974-83;Smith and Perelson), very few so ambitiously attempt to fit to data as this model—both due to the complex biology of the disease process which is continuing to be understood and due to a scarcity of data. It is very rare that a scenario arises that in a mathematical model for a biological system, as many as 15 for 20 variables are associated with data. This brings forwards many opportunities for growth and increasing validity of the model, but also difficulties in parameter fitting regimes where uncertainty and measurement errors propagate. Through this process of fitting, we come to a greater understanding of the model, its merits, and its place among other models for influenza.

3.7.1 Virus Trajectories

In terms of model reaction to parameter sets, there are two basic regimes for viral trajectories. First, the peak load of virus saturates at a level independent of dose, and the dosage determines the time of the peak; this is a common phenomenon in viral models (Nowak and

May;Hancioglu, Swigon, and Clermont 70-86). Second, the peak load of virus rises and falls as a continuous output of the initial dosage. In both regimes, there is a lower threshold of virus for which virus clears from the system without the viral trajectory achieving a positive slope. This model seeks to conform to the latter; however, parameter sets exist, and are common, that conform to the former. We have seen in iterative versions of the model as it has developed, that a parameter set that allows the peak level of virus to depend on dose will display the dynamic behavior that we want. Surely, in the data itself, the viral load for the lethal is over a log scale higher than in the sublethal. Some parameter sets in the ensembles avoid needing to deal dynamically by having a static peak in between the two sets of data.

Further in the viral trajectories, we notice a fall in the data on day one but no fall in the trajectory. Because of the well mixing property of the system, and the lack of a formal eclipse period written in the equations, the viral trajectories are not going to reasonably pass through the experimentally recorded data point on day one, since the value is so low. The model assumes that as soon as infected cells exist, they will apoptose and release virus, where this process takes approximately two days after the initial infection to complete. Once the virus has infected the system for two days; however, we can reasonably assume the well-mixed property that at any time an average number of infected target cells will be apoptosing and releasing new virions into the system. One of the goals for this system was that it be autonomous, i.e. that the right hand side be free of time-dependent terms. Using time dependent terms for these parameters, though, provide instructive fits to low initial time points, and are well documented in the literature (Baccam et al. 7590-99).

3.7.2 Target Limitation

This model deviates from the use of SIR dynamics in its description of the target cell population. Since there is no *removed* category of cell, there is no repository of cells to repopulate the system once the naïve *susceptible* population has been complete converted to the *infected* category. Many models rely on substrate depletion, i.e. the elimination of the *susceptible* cells, to motivate the virus to peak and fall in its trajectory (Nowak and May;Beauchemin 464-77;Smith and Perelson). The use of substrate depletion comes from epidemiology, and is widely used in modeling HIV and T-Cells (Baccam et al. 7590-99). While for influenza, substrate depletion may be a viable modeling technique on a local level, practically having the number of viable epithelial cells fall to zero would result in acute respiratory failure. Therefore, an individual that recovers should maintain more than approximately 40% of their cells available for respiration. In the model, the sublethal trajectories recover without falling below this threshold, showing that virus is eliminated from the system without depleting the stock of naïve cells. Correspondingly, the lethal doses do deplete the population of target cells, which eventually corresponds to falling virus after the individual has died. While there is no removed class, there is an algebraic expression for the effects of Type I interferon; however, these interferons do not prevent target cells from becoming infected, they simply affect what percentage of the infected cells are able to produce new virions. It is noteworthy that in the model the interferons do not achieve 100% efficacy.

3.7.3 Qualitative vs. Quantitative

The use of various parameter fitting techniques and the results of the ensemble model call certain questions to the forefront when practicing model fitting in a data rich environment. First and foremost is the balance of quantitative terms in the objective function and heuristic terms. The heuristic terms represent the qualitative features that we wish the model to represent, while the quantitative terms represent the data. In data poor environments, there is not an issue of over-fitting to the data or the accumulation of uncertainty from the data; however, in a data rich environment there must be a level of conscious consideration over how to weigh the heuristic terms in relation to the data-driven terms. Otherwise, as seen in the results, the qualitative features may evaporate completely in favor of a better quantitative fit. Another consideration in the data rich environment is keeping a level of skepticism about the quality of the data in terms of similar data sets. For the 15 sets of data collected, there are subsamples that come about empirically that do not have rational explanations. However, this can be from either unexplained biology or from faulty measuring techniques. In lieu of throwing an unexplained measurement out, it may be more reasonable to weight this measurement in relationship to the accuracy of the data.

In total, we see in fitting in a data rich environment, how important a well-considered objective function becomes. The posterior distributions that we sample exist in relation to the objective function and change in relation to how we re-balance or re-write the objective function. The areas of parameter space sampled also depend on the form of the objective function, as we see with the multi-objective function. Therefore, the problem of parameter fitting is not simply a matter of the algorithms we employ, but depends very highly on the heuristic interpretation of the nature of the model and the nature of the data.

4.0 MODELING OF EXOGENOUS CYTOKINES IN MOSQUITOES

4.1 BIOLOGICAL BACKGROUND

Malaria affects 350-500 million and kills 2 million people each year [1]. *Plasmodium falciparum*, the most important human malaria parasite, is transmitted by female *Anopheles* mosquitoes. Parasite development in the mosquito begins with the ingestion of blood containing sexual-stage gametocytes. Mobile ookinetes penetrate the mosquito midgut epithelium 24–36 h later and transform into midgut-bound oocysts within the open circulatory system. Oocysts grow and develop for 10–12 days and then release thousands of sporozoites, which invade the salivary glands and are released during later blood feeding.

The mosquito is not a neutral vector of transmission, but rather ingests blood components including immune-modulating factors from the infected mammalian host [2]. The impact of this "immunological cross-talk" on parasite transmission requires an understanding the complex blood-feeding interface of the mammalian host, the mosquito and the malaria parasites that utilize both of these hosts for biological development.

One of these cross-talking factors is the cytokine transforming growth factor-β1 (TGF-β1). In mammals, TGF-β1 is produced as an inactive precursor that is activated following dissociation of inhibitory proteins, a process that can be promoted by reaction products of the free radical nitric oxide (NO). Nitric oxide synthesis is catalyzed by isoforms of NO synthase

(NOS), among which the inducible isoform (iNOS) is commonly associated with host defense responses. In some immune cells,TGF-β1 reduces iNOS activity, expression, mRNA stability, translation, and protein stability, while in other cell types TGF-β1 can upregulate iNOS expression and enzyme activity [3–7]. This upregulation results in continued synthesis of NO [5], which in turn has the potential to induce synthesis of additional TGF-β1 [8–10], thereby yielding a cycle of positive feedback regulation.

TGF-β1 plays a pivotal role during malaria parasite infection and therefore may maintain an "immunological balance" during parasite infection [11]. The regulation of iNOS by TGF-β1 is central to this balance and to the effects of TGF-β1 on parasite infection [7]. During *Plasmodium* infection in humans and mice, NO can inactivate sporozoites and circulating gametocytes [12,13], and various studies have reported both beneficial and detrimental effects of NO on the pathology of infection [14,15].

The action of TGF-β1 in the context of parasite infection may be potentiated by the *A. stephensi* TGF-β ortholog As60A. *As60A* expression is induced in the mosquito midgut epithelium in response to *Plasmodium* infection – this response is parasite-load dependent and correlated with periods of parasite motility and growth [16,17].

Taken together, the interactions within the TGF-β1-NOS-NO axis in the context of malaria parasite infection in *A. stephensi* include complexities of timing, dose-dependent effects, feedback regulation, multiple inducers of single targets, and multiple targets of single regulatory factors. We hypothesized that computational simulation of this system may yield novel basic and translational insights into the biology, pathology, and ecology of malaria transmission.

Accordingly, we created a simple mathematical model that accounts for some of the main drivers of the response to ingested TGF-β1. Our model accounts for the oscillatory

expression of *AsNOS* and affirms that this oscillation requires the TGF-β1-induced expression of *AsNOS*.

4.2 MODEL OF INTER-SPECIES IMMUNE CROSS-TALK

In this study, we focused on the bloodfeeding interface between the mammalian host and *A. stephensi*, with the assumption that elevated AsNOS acts as a surrogate for parasite killing. Among the plethora of proteins present in the mammalian host's circulation, the cytokine TGF-β1 is present at high levels, though >99% of it is in a biologically latent form [18–20]. When *A. stephensi* takes a bloodmeal, it therefore ingests latent TGF-β1, which we have shown is converted rapidly to active TGF-β1 [21,22]. In mice and humans, TGF-β1 often suppresses the expression of iNOS [6]. In turn, iNOS-derived NO can lead to the activation of latent TGF-β1 [7,21–23].

As has been observed in mammals, the upregulation of *AsNOS* expression [22,24] and AsNOS catalytic activity [25] is involved in the activation of ingested latent TGF-β1 in the mosquito midgut [7,21,22]. Interestingly, low levels of active TGF-β1 upregulate the expression of *AsNOS*, thereby forming a positive feedback loop [21,22].

From here, we hypothesized the existence of an inhibitor for AsNOS. We based this hypothesis on two biological facts. iNOS, the mammalian analogue of AsNOS, is self-inhibitory [26]. Though it is currently not known if AsNOS is also self-inhibitory, we have shown that the induction of *AsNOS* expression is enhanced by L-NAME, a non-selective inhibitor of NOS [22], thereby supporting the hypothesis that NO feeds back negatively on *AsNOS* expression.

Moreover, biological systems that upregulate often have a mechanism for down-regulation. Thus, we introduce an inhibitor which is most likely MEK/ERK.

In modeling the mosquito-mammal immune cross-talk, we first depicted the relevant biochemical interactions graphically Figure 37. Next, to turn this model into a system of ordinary differential equations, the up-regulation arrows were written in the form of Michaelis-Menten kinetics, decay arrows were written in the form of mass-action kinetics, and down-regulation was written in the form of competitive inhibition [27,28].

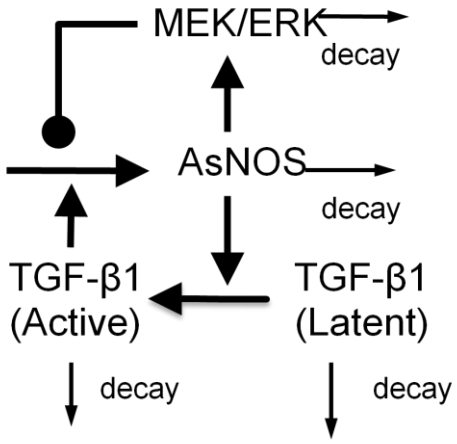

Figure 37. Schematic for the primary bio-chemical interactions in the mosquito midgut.

From here, we write equations describing these processes. We let *L* represent Latent TGF-β1, *A* represent Active TGF-β1, *N* represent *AsNOS*, and *X* represent the hypothetical Inhibitor. The functional terms are written as Michaelis-Menten kinetics or Hill-type functions. We assume that *L* is converted to its active form via NO, here represented by the expression of *AsNOS*. There is a small decay of *A* as well. We assume a constant background level, b_n, of *AsNOS* with a constant decay rate. The main positive feedback loop in our model is that *A* is able to induce *AsNOS* production [21,22] but that this production can be inhibited by *X*. Finally, *X* is itself produced by *N* and has an intrinsic decay. With these assumptions, we obtain the following set of simple relations:

$$\begin{aligned}
\frac{dL}{dt} &= -L\ f_1(N) \\
\frac{dA}{dt} &= L\ f_1(N) - \mu_A A \\
\frac{dN}{dt} &= \mu_N\ (b_n - N) + \frac{f_2(A)}{f_3(A, X)} \\
\frac{dX}{dt} &= -\mu_X X + f_4(N)
\end{aligned} \tag{0.26}$$

4.2.1 Model Reduction

In terms of experimental data, we have one state variable, *N*, to model, and three variables without data. For *N*, we have 6 time points for each trajectory, and 4 dosages, additionally making this problem multi-objective in addition to being under-determined. To deal with the issue of dimensionality, we conscientiously reduce the dimensions of the system. The project was originally motivated by the hypothesis that exogenous cytokine directly drives the behavior of the system for the entirety of the experimental study; from this we start with the assumptions that exogenous cytokine exists in abundance and that the reaction is happening in relation to a static up-regulation rate, making production of *N* roughly proportional to current levels.

Thus, we can write the third formula from the equation purely in terms of *X* and *N*. The two dimensional system is an activator-inhibitor system, using competitive inhibition.

$$\frac{dN}{dt} = \mu_N (b_N - N) + \frac{\alpha N^p}{N^p + a_{NX} X^q + a_N^p}$$
$$\frac{dX}{dt} = -\mu_X X + \frac{b_{XN} N}{a_{XN} + N} \tag{0.27}$$

Hence, we have reduced a four dimensional system with data in one dimension into a two dimensional system with data in one dimension.

Parameter estimation for this problem presents issues on various fronts. First and foremost, this system is not well studied in the literature and there are few values that we can draw or approximate. So, our parameter search will be solely in terms of finding an a priori parameter set that fits the data. Second, we need to understand the structure of the model and how it interacts globally with the parameter sets; varying parameter sets introduces an Andronov-Hopf bifurcation, which should not be surprising since this is an activator-inhibitor system. The data suggest some up and down motion, in 3 of the 4 data sets the mean moves up and down alternately between each data point. But we are limited by the amount of available data; there are 6 time-data points for each trajectory which can yield a maximum of 5 motions in the data, whereas we would like to see three full oscillations, i.e. 6 motions, before substantively claiming oscillations. Due to the motion of the data, we seek at first to fit this data to a set of parameters on or near a stable limit cycle; even if the system is not itself oscillatory, a stable limit may be the best way to encapsulate the up and down motions observed in the data. We use AUTO [29] to numerically confirm that a supercritical Andronov-Hopf bifurcation of this system exists for p at approximately 1.9 for our initial parameter set.

To fit the parameters to the data, we introduced a likelihood function of the data and parameters [30] , and use a Metropolis-Hastings method to sample the distributions of parameters

that fit to this model. Working from the parameter distribution produced by this Metropolis Hastings method, for each separate data set a Levenberg-Marquadt algorithm is applied to choose a specific value for each parameter [31]. The centroid of the parameter distribution is used as the initial guess. To avoid convergence failure, the parameters for the exponents p and q are held constant. Several runs of the Levenberg-Marquadt method are then run for various sets of p and q. The parameter set with the lowest normed difference from experimental data is chosen.

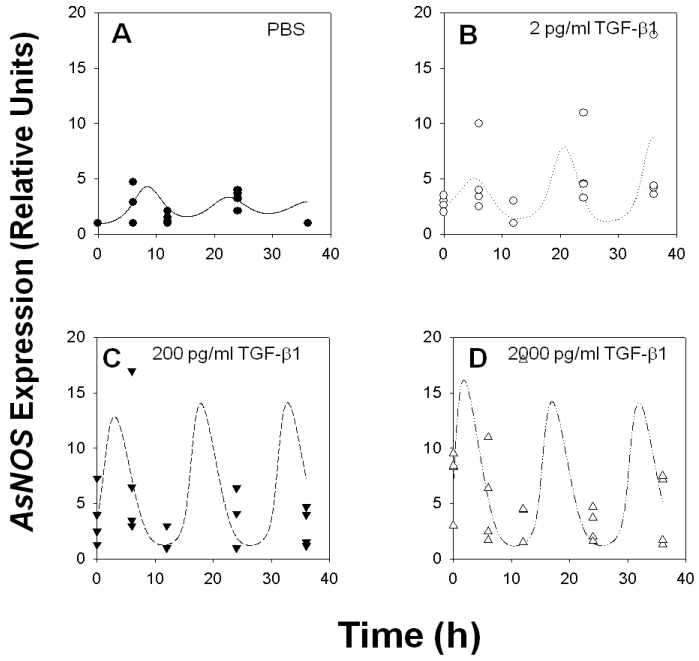

Figure 38. A single trajectory for each dosage of exogenous TGF-β1

(A) PBS, (B) 2, (C) 200, (D) 2000 pg/ml

These parameter values then produce the trajectories seen in Figure 38 A-D. By construction, the only parameter that should vary from scenario to scenario is the value α, which represents the level of TGF-β1 in the system. With the exception of a_n, the other values remain

consistent, regardless that the parameter values for each scenario are derived independently of one another.

Parameter	Value Figure 38	MCMC Figure 40	Parameter/ I.C.	Value Figure 38	MCMC Figure 40
μ_n	1.6	[1.4, 1.8]	α_{PBS}	35	[38, 42]
b_n	0.5	[0.45, 0.55]	α_2	42.5	[40, 45]
a_{nx}	1.3	[1.1, 1.4]	α_{200}	49	[44, 50]
a_n	5.5	[5.1, 5.8]	α_{2000}	50	[48, 52]
p	2	[1.8, 2.2]	N^0_{PBS}	1	[0.8, 1.2]
q	1.7	[1.5, 2.0]	N^0_2	2.5	[1.0, 2.0]
μ_x	0.66	[0.6, 0.75]	N^0_{200}	3.75	[1.2, 3.0]
b_{xn}	28	[26, 30]	N^0_{2000}	7	[3.0, 6.0]
a_{xn}	10	[9.5, 11]	X^0_{PBS}	7.6	[6, 9]
			X^0_2	7.5	[6, 9]
			X^0_{200}	8	[6, 9]
			X^0_{2000}	7.9	[6, 9]

Table 6: Parameters and Initial Conditions for 2D Mosquito System

For a_n, only the value for the PBS scenario differed while the other three remain consistent with one another. The quasi-steady state assumption requires that the equation go quickly to equilibrium. While this assumption bears fruit for the other three scenarios which have higher initial levels of TGF-β1 and thus drive the equation more quickly to equilibrium, the assumption appears to be weaker in the case of control levels of TGF-β1. However, oscillations persist for the control case; so we examine a two-parameter bifurcation varying both α and a_n, which yields in Figure 39 a robust region of oscillations which terminates in a Takens-Bogdanov bifurcation outside the experimentally relevant region [32].

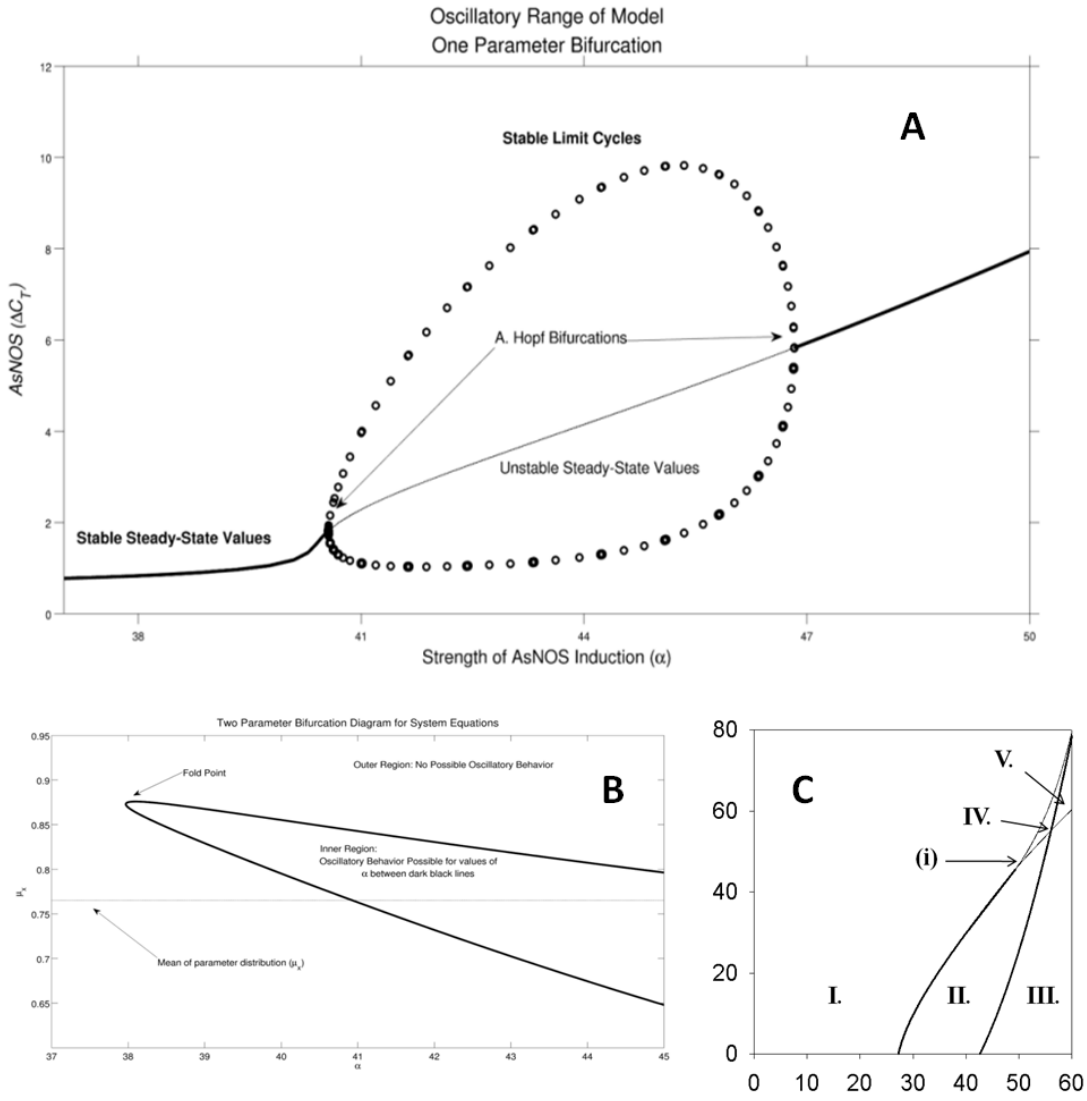

Figure 39. Bifurcations of the Two Var Mosquito Model

A. 1 Parameter Bifurcation showing Hopf bifurcation B. 2 Parameter Fold bifurcation

C. 2 Parameter Takens Bogdanov bifurcation.

4.2.2 Ensemble Modeling

While a single parameter set can be representative of the types of trajectories the model produces, often it is useful to explore many parameter sets that fit the data equally. To do this,

we revisit the 2 variable model of TGF-β1, *AsNOS,* and MEK/ERK inhibitor from the context of ensemble modeling (i.e., creating multiple, equally plausible, biologically similar models that fit the data). A Metropolis-Hastings algorithm employing Markov-Chain Monte-Carlo, takes a given parameter set and perturbs it to produce either a statistically similar parameter fit, or repeats the perturbations until an equally plausible parameter set arises. Every parameter set is designed to simultaneously fit the four dosages for which we have data. Additionally, there are parameters individually selected for separate trajectories in the categories of the first initial condition, the second initial condition, and the degree of self-activation, making a total of 21 parameters. Each parameter is randomly selected from a range of values. Parallel tempering is performed over 6 chains at different temperatures, with swapping every 25 iterations, for a total of 2,500,000 iterations of parameter selections for each chain.

Results were filtered by two criteria. First, the parameter for activation corresponds to each dosage, so since the dosages monotonically increase, the parameters alpha for activation we require to monotonically increase. Second, we require that the baseline value of activation parameter alpha (corresponding to PBS) produces a stable limit cycle and unstable fixed point. This second requirement we impose because experimental data exists implying levels of AsNOS that rises and falls repeatedly in the long term without exogenous stimulation.

Figure 40 shows the output of the ensemble, and shows us how the baseline parameter set reacts to rising activation in the model. The graphs present various trajectories that fit the experimental data equally well and demonstrate the robustness of the model to changes in parameter values. The black lines represent the mean of all possible trajectories. The dark grey represents the variability of the central 50% of trajectories, and the light grey represents the variability of the central 90% of trajectories.

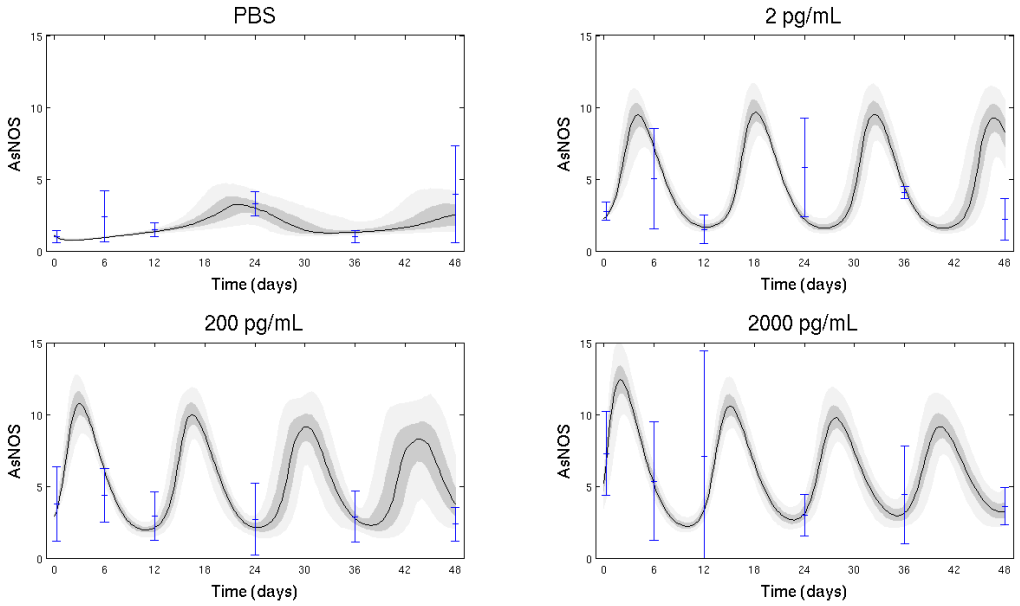

Figure 40: Ensemble Model of Multi-Objective Function.

Predicted *AsNOS* dynamics at four doses of TGF-β1 in the bloodmeal of *Anopheles stephensi*, fit with an ensemble of parameter sets. The means and standard deviations are shown as error bars.

The predicted trajectories give a good representation of the data. With the exception of the 48 hour time point for dose 2 pg/ml, all trajectories lie close to one standard deviation of the mean. The data points for the 200 pg/ml are all relatively low, but the parameter set is slave to the fit for 2 pg/ml and 2000 pg/ml, which both have much larger apparent data values. Further, the activation is selected to be between the activation for these two. The model thus predicts that there exist much larger values outside the times that the data was collected.

The PBS panel suggests a baseline oscillation of the system that has an approximate period of 24 hours; it is not apparent if this is linked to the circadian rhythm of Anopheles stephensi. Treatment with increasing doses of TGF-β1 gradually increases the amplitude of the oscillations and shorten the period to almost 12 hours in the highest dose. In all trajectories, there is low variability in the start of the oscillation and high variability in the tail.

Observationally, there is more variability in how the parameter sets fit the 200 pg/ml dose, compared to all the other doses.

Having used data on 0, 2, 200, and 2000 pg/mL TGF-β1 as input, we proceed to predict how a 6,000 pg/mL dose will affect the ensemble of parameters.

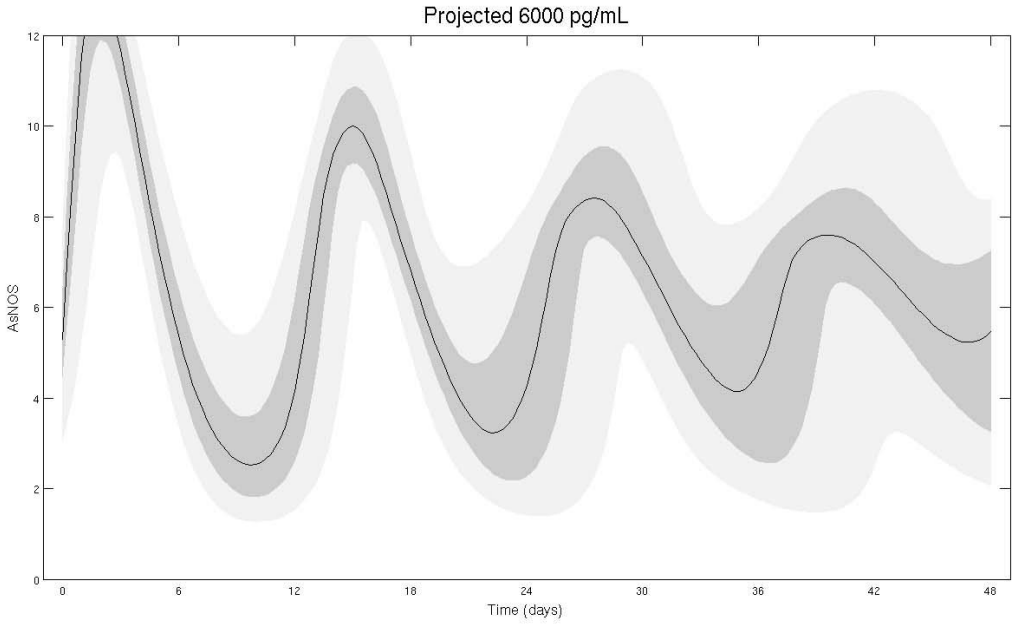

Figure 41. **Predicted *AsNOS* dynamics at 6000 pg/mL TGF-β1.**

The ensemble of parameter sets when given initial conditions and TGF-β1 parameter altered for a dose outside fitted range.

Prediction of the response to 6000 pg/ml requires extrapolation of three parameters: activation parameter α, and two initial conditions. Further, given that this is an ensemble of parameters, we require a range of values for these three unknowns. Therefore, means and standard deviations are extrapolated rather than individual values. For the first initial condition and activation α, means and standard deviations were extrapolated from the other parameter sets. For the second initial condition, mean and standard deviation was interpolated. Then, for each

parameter set in the ensemble, we use the mean and standard deviation to randomly draw 20 sets of these values from a normal distribution.

The simulation was then run as before, and the quartiles are depicted in the previous graph. We predict that at 6000 pg/ml, the time evolution of *AsNOS* will be much more variable towards the tail, suggesting a much less uniform response (and, likely, lack of statistically significant difference in *AsNOS* vs. PBS alone) as compared to 2, 200, and 2000 pg/ml. An initial analysis of these findings suggest that some of the fits preserve the stable limit cycle seen in lower doses, but some parameter sets show they have crossed a Hopf bifurcation, destabilizing the limit cycle and making the fixed point stable. Thus, the *AsNOS* in the system is either attracted to a stable oscillation else dampens to a fixed level of AsNOS.

4.2.3 Discussion

In experiments from which we obtain data, the level of TGF-β1 is successively increased. The model represents the effects of the TGF-β1 by a parameter α; this parameter controls the degree of activation in the model. Normalizing by the baseline value of the activator (in PBS) for a given ensemble, we find that the averaged increase in activation of 2, 200, and 2000 pg/ml are 9%, 12%, and 14%, respectively. We can infer that the majority of the achieved effect occurs at very low levels of external TGF-β1. We see in Figure 42 that the activation corresponding to each dose saturates as the dose increases. The source of this saturation is unclear; it may be due to receptor saturation in response to increasingly large doses of TGF-β1. The data point for 6000 pg/ml was extrapolated from the previous points, using a linear extrapolation on a log scale, and its extension is represented by a dotted line.

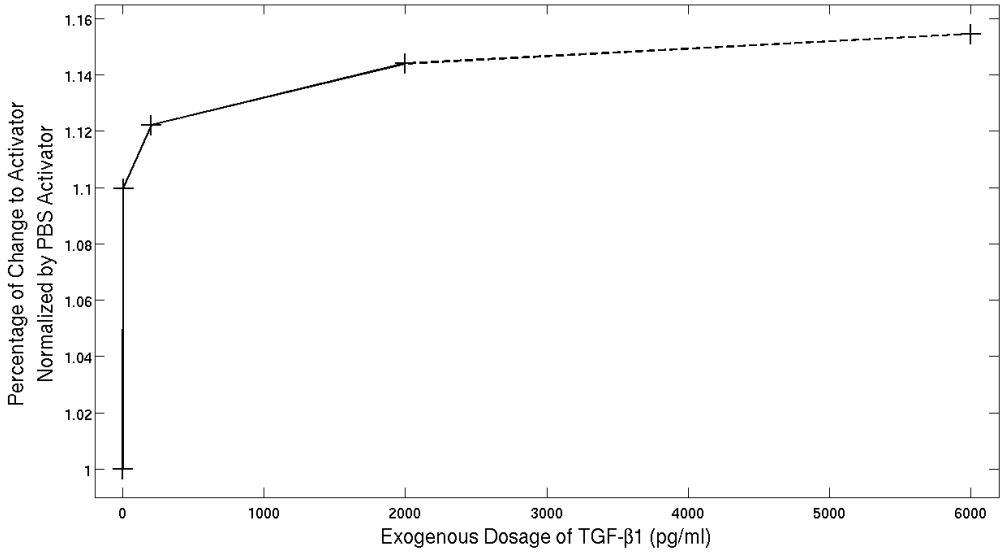

Figure 42. Saturation of activation parameter α

Dosage levels 0, 2, 200, and 2000 pg/ml. An extrapolated data point for 6000 pg/ml is shown.

From the global sensitivity analysis of the model, we observe three primary behaviors for the *AsNOS* trajectory. These behaviors depend on parameter choice, where the parameter sets exist in relation to the manifold Andronov-Hopf bifurcations and the Takens-Bodanov bifurcation as seen in Figure 39. The first behavior goes quickly to a steady state between all the time points, giving an equally bad fit to all points. The second behavior is excitable, giving an initial spike in *AsNOS* before going to steady state. The final behavior gives parameter fits that result in a stable limit cycle, that oscillates and approximates many of the data points. The data, by its nature, goes up and down in a non-intuitive way, and one should not expect true oscillations; however, the oscillatory graphs best describe the data when compared to the system going quickly to steady state.

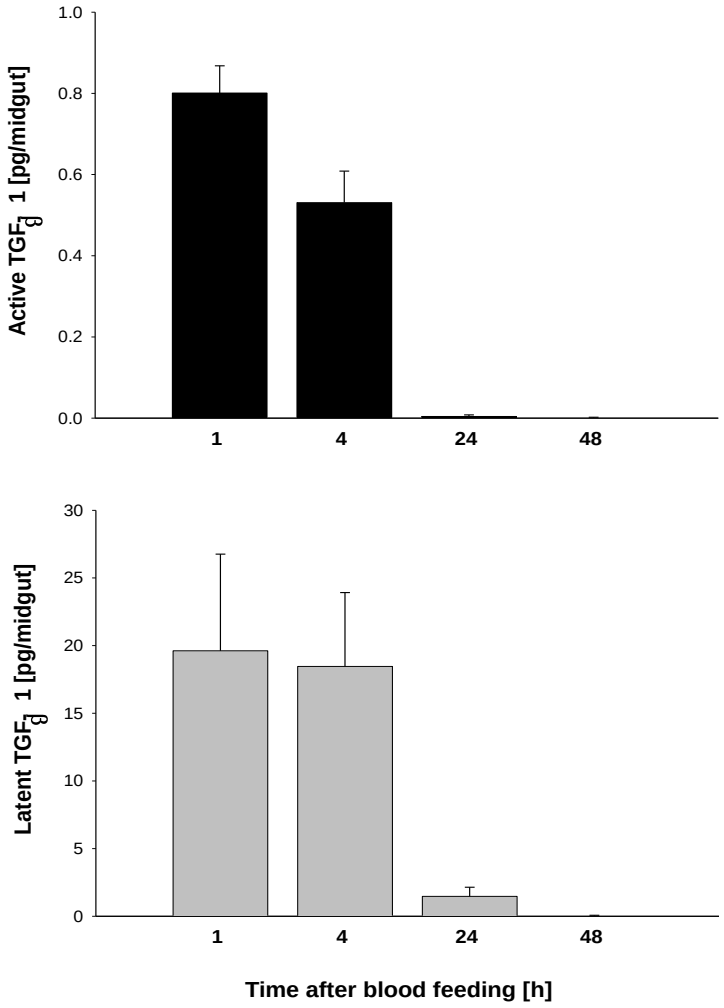

Figure 43. Degradation of TGF-β1, (A) Latent and (B) Active, in the midgut.

The natural next step, would be to reverse the dimensionality reduction to come back to the higher dimensional model. However, this proves problematic as even slow decay of TGF-β1 forces the system to rest. This is an insurmountable issue in the model, as it was later determined experimentally that TGF-β1 has a very short half-life in the *Anopheles* midgut (Figure 43) [21,22], and therefore the assumption of a constant level of TGF-β1 or TGF-β1-like activity would appear to be unjustified. Accordingly, we made assumptions that allowed for an oscillatory fit: we hypothesized that the amount of TGF-β1-like activity remained present at a constant value for the experimental time frame. This procedure of modeling and successful fit

implies there is a quality of this assumption that needed to be further investigated. This assumption can be justified by including *AsNOS* induction of the *Anopheles stephensi* homolog of TGF-β1, As60A (Crampton et al., 2001). In support of this hypothesis, we found that indeed TGF-β1 induces As60A and that this induction is regulated negatively by NO.

The As60A hypothesis eloquently explains why the numerically-derived values of α saturate rather than multiply. As there exists a finite number of receptors that the TGF-β1 can bind to and activate, only the maximal amount of As60A, as limited by the number of receptors, can be produced. The remainder of the TGF-β1 then decays naturally before binding to cell receptors. This can be observed as 200 pg/ml of TGF-β1 corresponding to a homeostatic level of 1.12% activation while 2000 pg/ml corresponds to 1.14% activation.

The parameters associated with TGF-β1 in those two equations may be better used in describing the homeostatic effects of the Anopheles homologue. Taking into account that α better represents the homeostatic level of As60A, we hypothesize that the experiments with higher initial levels of TGF-β1 reach this homeostatic level quickly, as the parameters vary little between the two experiments. On the other hand, since the control experiment varies the most from the other parameter sets, we can hypothesize that a homeostatic level of As60A is not reached, else is not reached quickly else does not approach a steady state value.

4.2.4 Conclusions

This model described herein provides a first look at the immune processes in the mosquito midgut. The model predicts that an endogenous version of the same cytokine would be necessary to drive oscillations in *AsNOS*. Further, in modeling the properties of TGF-β1, we have a first iteration in modeling its ortholog, As60A. The model also provides a first iteration

in modeling the inhibition system within the mosquito. However, further modeling may provide insight into whether the inhibition is on the transcriptional, translational, or post-translational levels.

From here, a proposed TGF-β1 ortholog such as As60A should be added into the system, to model the dynamics when the mammalian cytokine decays sufficiently so as to cease signaling to the system. Then, *P. falciparum* will be added to the system to determine the dynamics of AsNOS in its immune role. The ultimate goal is to understand the mosquito as not simply a vector of disease transmission, but as a potential tool to limit the spread of a disease-causing parasite.

4.3 FIVE EQUATION MOSQUITO MODEL

The 4 variable model proposed in the previous section was driven by exogenous cytokine TGF-β1. Attempts to model the 4 variable model to the data ultimately fail because we now have experimental data confirming that the cytokine decays so that little is left after 24 hours. Without this exogenous driver, the system proceeds to a stable fixed point very quickly. With the constraint that decay must be so that this is true in the model, we cannot proceed. However, there does exist an endogenous driver in the biological system, As60A, which can account for the baseline oscillations. We hypothesize that this cytokine, along with AsNOS and the MEK/ERK pathway, continuously oscillate at baseline, and becomes excited in the presence of exogenous cytokine with decaying oscillations proceeding back to baseline.

4.3.1 Model Equations

We model the mosquito midgut with a system of 5 equations after the original model (0.26) representing: Latent TGF-β1 (L), Active TGF-β1 (A), *AsNOS* (N), As60A (S), and MEK/ERK (X). Unlike the previous model, we divide this model into two components, the exogenous system (L) and (A), and the endogenous system (N), (S) and (X).

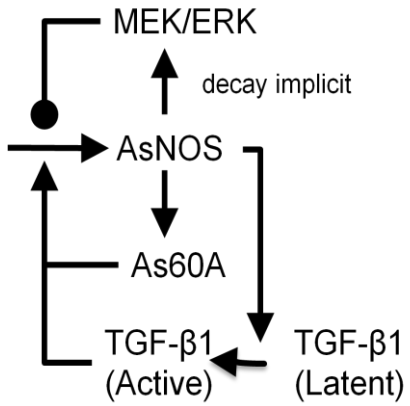

Figure 44. Schematic of relevant chemical interactions, updated from **Figure 37**

The endogenous system, in the absence of exogenous stimulus, oscillates at a low baseline level. AsNOS activates production of both activator As60A and inhibitor MEK/ERK, producing a dynamic structure similar to an activator/inhibitor system like the 2 variable model. In equation (0.28), Hill-coefficients r_1 and r_2 are chosen for the activator and inhibitor so that the limit cycle is stable. Otherwise, the endogenous system is structurally similar to the 2 variable model.

The exogenous compartment contains a combination of the decay of both species of TGF-β1 and the induction of latent to active by a double saturation of latent and AsNOS. Decay was added to latent because high initial doses did not decay in accord with Figure 43. Various formulae were tested for the induction of latent to active. Since the exact mechanism of induction is unknown, we qualitatively tested various regimes including mass action and

individual saturation of both species in the induction. The exogenous and endogenous compartments are linked through the orthologs AsNOS and As60A. Their combined effected is represented as Σ. The correlation coefficient c_r represents the efficacy As60A has on the endogenous system in relation to its ortholog.

$$\begin{aligned}
\frac{dL}{dt} &= -b_L\left(\frac{L}{a_L+L}\right)\left(\frac{N}{a_{LN}+N}\right)-\mu_L L \\
\frac{dA}{dt} &= b_L\left(\frac{L}{a_L+L}\right)\left(\frac{N}{a_{LN}+N}\right)-\mu_A A \\
\Sigma &= S + c_r A; \\
\frac{dN}{dt} &= \mu_N(b_N - N) + \frac{b_{NS}\Sigma^p}{\Sigma^p + a_{NX}X^q + a_{NS}^p} \\
\frac{dX}{dt} &= \frac{b_{XN}N^{r_1}}{a_{XN}^{r_1}+N^{r_1}} - \mu_X X \\
\frac{dS}{dt} &= \frac{b_{SN}N^{r_2}}{a_{SN}^{r_2}+N^{r_2}} - \mu_S S
\end{aligned} \tag{0.28}$$

Parameters for the exogenous compartment are derived from exploring a parameter space that gives output consistent with available data, such as in Figure 40 and Figure 43. Parameters in the endogenous compartment are based on the parameter ranges in the 2 variable model, and rescaled in time to give an 8 hour oscillation at baseline. As of now, the parameter set derived from searches is qualitative rather than quantitative; we wish to explore the increased robustness to external stimulation the system yields that was eliminated when we reduced the number of dimensions in the 2 variable model.

4.3.2 Qualitative Model Trajectories

In the trajectories in Fig. 43, the panel showing the response to PBS in the top left shows a stable 8 hour oscillation, as per the design of the model and parameter selection. This period was chosen due to its best ability to fit data. The other three panels show response of the system to 2, 200, and 2000 pg/ml of TGF-β1. For an initial non-zero level of TGF-β1, the system shows two basic reactions. First, stimulation produces a higher than baseline peak or sequences of peaks. Second, stimulation produces a phase resetting, where the subsequent peaks occur earlier.

Parameter	Value	Parameter	Value
b_l	1	μ_x	0.97
a_{ln}	2	b_{xn}	55
a_l	15	a_{xn}	9.9
μ_l	0.4	r_1	2
μ_a	0.4	μ_s	6.4
c_r	0.4	b_{sn}	120
μ_n	2.4	a_{sn}	5.6
b_n	0.63	r_2	3
b_{ns}	190	L_0	{0,2,200,2000}
a_{ns}	16	A_0	0
a_n	8	N_0	1.1
p	1	X_0	0.7
q	2	S_0	0.14

Table 7: **Parameters and Initial Conditions for 5 variable Mosquito Model**

However, as seen in Figure 45, the two larger initial doses of TGF-β1 (200 and 2000) also have another defining trait: *AsNOS* levels stay moderately elevated while peaking very little. This is due to corresponding high accumulation of inhibitor, presumed to be activated MEK/ERK, following the large initial peaks of *AsNOS* from stimulation. In the 2000 pg/ml case,

the trajectory does not start to undampen until nearly 48 hours and requires even more time to return to the stable attractor as seen in the PBS.

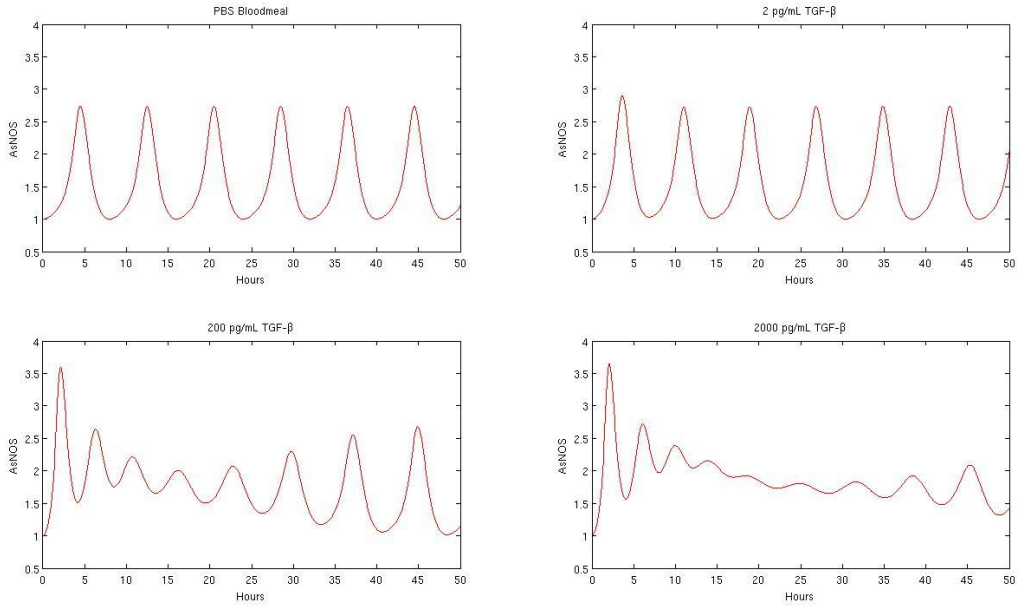

Figure 45. Trajectories for Qualitative 5 variable model

For the four scenarios: PBS, 2, 200, and 2000 pg/ml TGF added at time 0

In the model, we do not include parasites which will produce their own set of pro-inflammatory stimulants, so it is important that in the model we can hypothesize how the system will respond to further stimulation and possibly a glut of stimulant at later time points. In the 2 pg/ml case, we see only a modest peak and a phase resetting. This may not be much, but a marginally elevated level of AsNOS early in the infection may help in the long term infection course. In the short term, the parasite does not produce stimulants for the system to respond to, which is what makes the presence of exogenous stimulus in the form of TGF-β1 so important. However, the system needs to be able to respond dynamically once the parasite does start inducing immuno-stimulants. And if there exists a glut of inhibitor preventing the system from responding, it may hamper the ability of the system to properly respond to an ongoing infection.

In the 200 pg/ml case, there is a high early AsNOS peak and reasonable time frame for recovery of oscillations; inhibitor MEK/ERK is elevated initially but returns to the stable limit cycle attractor. For 2000 pg/ml TGF-β1, there is also the high early *AsNOS* peak, but also a highly dampened response of AsNOS in the long term, due to persistent MEK/ERK activation. We therefore hypothesize that it will be difficult to stimulate a response to parasites or parasite-derived immuno-stimulants in this case, as it takes more than 48 h to return to return to the trajectory regime that we know is responsive.

4.3.3 Testing Additional Stimulation Hypotheses in silico

Based on the mathematical model, we could predict that the 2000 pg/mL case will be less responsive to any stimulus that induces *AsNOS* as compared to the 2 or 200 pg/mL cases. We tested this hypothesis *in silico*, and show the results in Figure 46. From simulated time point 18 h until 22 h, we use a smooth heaviside function added to Σ to continuously add a moderate amount of *AsNOS* equivalent to simulate increases in AsNOS induced by stimuli not currently in our mathematical model (e.g. parasite-derived immunostimulants). The original trajectoriess as seen in Figure 45 are repeated in black, with the change from the additional input graphed in red.

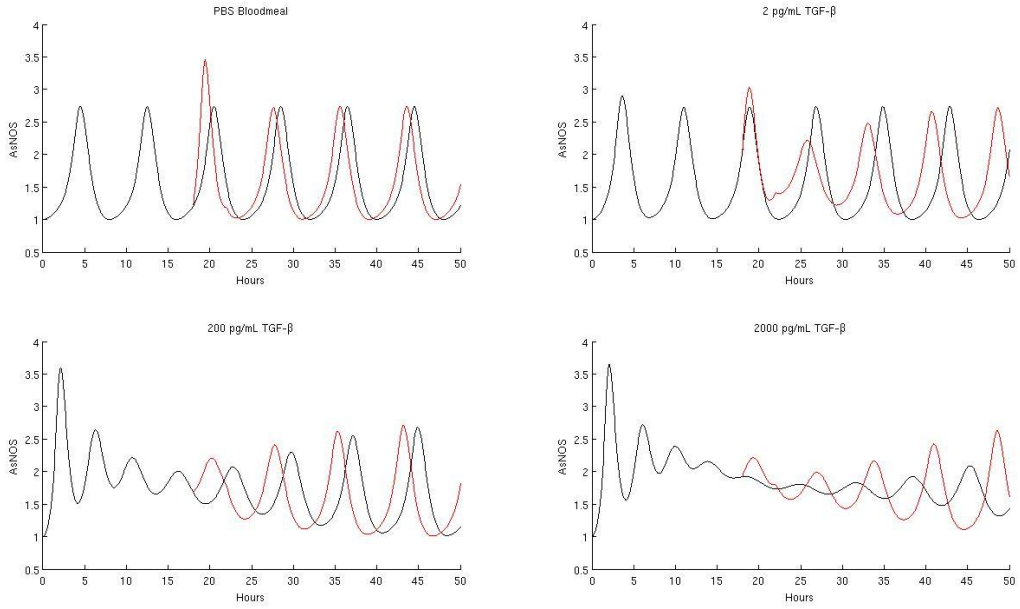

Figure 46. Stimulation at 18 hours

As in **Figure 45**, the four scenarios are plotted. At time 18 hours, an additional stimulus of moderate level is given for 4 h until time 22 hours.

The results fall in line with what we know about stimulation being added in the form of TGF-β1. In PBS, a moderate spike occurs, similar to the original 2 pg/mL case, followed by phase resetting. In 2 pg/ml, a reduce spike occurs, followed by a phase resetting and a trajectory that quickly return to baseline. This phenomenon was seen in the original 200 pg/ml case, where a glut of inhibitor must decay before full spiking returns. In 200 pg/ml, the mild oscillatory behavior is augmented and the system returns to true oscillations more quickly than without additional stimulation. Full oscillations seem to return by 28 h, rather than 36 h. In 2000 pg/ml, the near monotone original curve begins to show some oscillatory and peaking behavior. It begins to approach baseline oscillations by 40 h as opposed to the original > 48 h. While the first three scenarios produce pronounced peaks, the fourth responds sluggishly to stimuli, although better than nothing at all.

These trajectories seemingly confirm our hypothesis regarding the winged response of the higher dosages of TGF-β1 to further stimulation. From these trajectories, we seek to investigate further using the mathematical model. We compare the responsiveness of the system to further immunostimulants versus that to inhibitors of MEK/ERK. We hypothesize that time is important, so we also computed output for addition of stimulation for time point 0 h until 4 hr, but it showed no substantive difference to the original graphs shown in Figure 46, indicating that the mechanisms for production and inhibition are already saturated during that time frame by the initial TGF-β1 dosages.

Whereas adding immuno-stimulation at time 18 hours has a pronounced effect on all trajectories, adding further immune stimulation early on affects only the size of the peak and not the long term outcome of the trajectory nor its long term competitiveness. In these in silico experiments, we have focused on the effects of further immuno-stimulation in the context of how the system would respond to a parasite driven regime. However, in the long term treatment of parasite, we wish to develop a smart way of approaching the control and elimination of parasite in host. Since our main obstacle has been the glut of inhibitor induced by an over activated AsNOS regime, we test the response of introducing competitive inhibition to the production of the modeled inhibitor MEK/ERK. We introduced a smooth heaviside for the competitive inhibitor to MEK/ERK at time 0 hours until 4 hours in Figure 47, and 18 hours until time 22 hours in Figure 48.

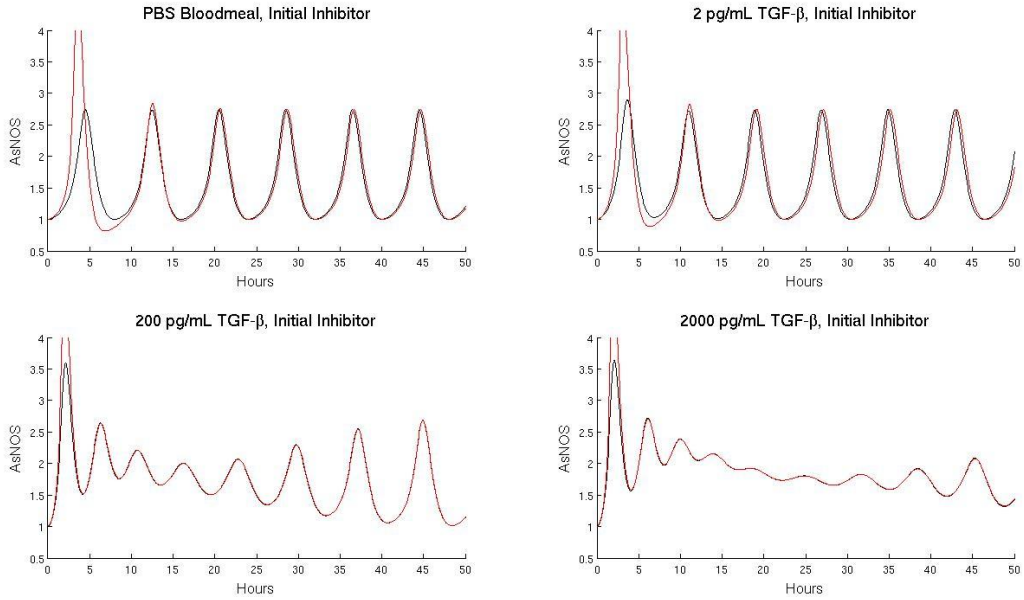

Figure 47. Addition of inhibitor at time zero.

The four scenarios are plotted. At time 0 hours, a competitive inhibitor affecting production of MEK/ERK is given for 4 h until time 4 hours. The new simulation is plotted in red. The curves from Figure 45 are repeated in black. The two are almost identical.

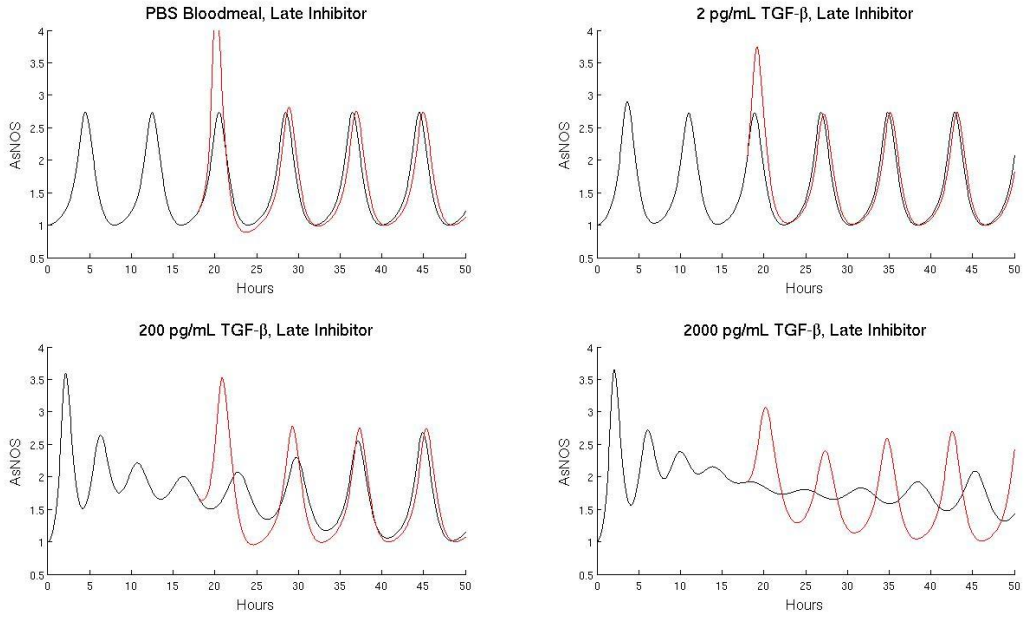

Figure 48. Addition of inhibitor at time 18.

The four scenarios are plotted. At time 18 hours, a competitive inhibitor affecting production of MEK/ERK is given for 4 h until time 22 hours. The new simulation is plotted in red. The curves from Figure 45 are repeated in black.

In the PBS, we see a moderate spike occur in both regimes, similar to the original 2 pg/ml case, however with little or no phase resetting. In the 2 pg/ml cases, for the initial inhibition, a slightly larger initial spike occurs, but no long term effects are seen. For the later inhibition, a moderate spike occurs without many long term differences. The more interesting effects start to be seen in the higher dosages. In the 200 pg/ml case, the initial inhibition causes a larger primary spike but no long term changes; however, the later inhibition resets the trajectory to the baseline oscillation. In the 2000 pg/ml, the results are similar to the 200 pg/ml case but more dramatic. The return to baseline oscillations in the later inhibition case occurs very rapidly as opposed to the addition of immuno-stimulants in Figure 46.

4.3.4 Discussion

The oscillatory behavior of the mosquito midgut as proposed by the model is dynamic and moreover robust. It explains the dynamic responses seen in data in response to various bloodmeal experiments, and may explain why 24-48 h time responses to PBS do not appear dissimilar to other data. This model matches the times of the peaks and troughs seen in the data, but not their magnitudes, so as not to be a quantitative fit. While this model is currently a qualitative fit, it provides insights into the machinery of the chemical interactions that the 2 variable model did not. For example, because this system is robust in its response to external stimulation, we demonstrate how over-burdening the system with TGF-beta can result in detrimental effects. A heavy rise in inhibitor can make it difficult for additional stimulation later

to produce any effects. Additionally, prolonged exposure of native tissue to elevated TGF can be deleterious to the host and provide a compromised barrier for parasite to exploit.

The absence of oscillatory behavior in the 2000 pg/mL case is indicative of a loss of robustness to response in the system. This is confirmed with the weak response to stimulation in Figure 46. It has previously been shown that 2 and 200 pg/mL inoculations with TGF decrease parasite load over baseline while 2000 does not in highly virulent strains of parasite. We can hypothesize from this model that this effect may be the result of 2 and 200 having a strong initialized response followed by a continued dynamic response, while 2000 does not emit a dynamic response after its initial strong response.

Our *in silico* experimentation leads us to hypothesize that a MEK/ERK inhibitor that is effective until 24 hours might improve the responsiveness of the host system to parasite-derived immuno-stimulants, especially when the initial does of TGF-β1 is high.

5.0 CONCLUSION:

The project confronted several challenges in its endeavor to produce valid, tractable, intuitive models of the biological systems being described. In retrospect, producing the equations for a nonlinear model with the desired balance of abstraction and biological fidelity was the simple component of the projects. Issues of data availability, high variation, and methods of collection all plague the quality of the biological information against which we compare and validate the models. That much of the biology itself is a matter of debate creates a scenario where several models must be vetted and compared against available data before a local optimum among models of similar types can be chosen. Computation and numerical integration create further barriers to building large, complex, non-linear models. For higher dimensions, efficient exploration of parameter space also creates a need for innovation and testing of new techniques.

In the milieu of these issues, inter alia, the results of these studies show that there is room in the ensemble of canonical ODE models for more varied, more subtle approaches. The 4 variable model of immune-cell reaction to pro-inflammatory effectors shows a nuanced, qualitatively accurate representation of a local inflammatory response; the bifurcation analysis further shows the different variations in response for various parameter regimes. The 8 variable tissue level model demonstrates higher level effects of this compartment, with stable health, decay, and in certain parameter regimes a stable auto-inflammatory chronic state. This chronic

state is shown spatially as potentially being an isolated, unhealed wound. This tissue model then placed into a 20 variable, 90 parameter system of viral response shows how inflammation in a viral response can reduce survival as the inflammatory paths overwhelm the system before the adaptive immune system clears pathogen. While much simpler in principle, the mosquito model takes onto itself the task of fitting a stable limit cycle to data with a multi-objective function taken from 4 sets of data; practical issues surrounding the fitting of a limit cycle to data make it an interesting question to itself. The metropolis-hastings method with parallel tempering developed over the course of this project proved to be an efficacious method of exploring possible solutions of both these endeavors.

At the heart of this project were two separate but complementary problems—a forward problem of defining a system of ODEs that describes the biology and an inverse problem of identifying parameters for that system to replicate available data. Chooses of modeling included a smaller non-linear system over a large linear system; indeed using non-linear terms and understanding their effects was a prerogative of this project. Other systems of modeling argue against use of nonlinearities. Another prerogative of the project was the focus on mechanistic terms over phenomenological terms. Although there were several iterations of increasing accuracy in the biology and fitting with successive models to find a locally best model, uncertainty remains as to whether better models of either the same type or other types exist. The overhead of changing the model is the re-calculation of the inverse problem, which itself has high levels of uncertainty.

In the course of working on multiple projects, slowly the methodologies developed through the course of research have moved from being ad hoc methods to generalized techniques that can be applied in multiple situations. Parameter searches in high dimensions is an increasing

necessity as models grow in complexity. The Metropolis method with parallel tempering for the multi-objective functions is a versatile and useful tool in approaching the inverse problem for parameter fitting. Genetic Algorithms, another widely used tool reviewed in this research, rarely exceed capacity to search 30 dimensional dimensionality let alone 90 dimensions. That we have, from very rough fits, converged on a high dimensional distribution of acceptable parameter fits is a great accomplishment to itself. That we have further refined the model and re-fit numerous times increasing accuracy of the fit in successive iteration points to its strength. That it could be extended to the setting of the mosquito model with a four dosage multi-objective ensemble with ease demonstrates its versatility. That there is also a tool developed for parameter fitting when the model is highly non-linear opens up the possibility for more non-linear models to be developed. Increased tractability in nonlinear and large models coming from increased efficiency in computation and computational tools allows us to best access and use effectively gains in processing power.

APPENDIX

A.1 DERIVATION OF SATURATION KINETICS

To derive the equations used for saturating inhibition, we start from the initial equations (0.2) and set them in quasi-steady:

$$\frac{dD_{reg}}{dt} = \alpha C_{anti} - \mu_1 D_{reg}$$

$$\frac{dU_{reg}}{dt} = \beta C_{pro} - \mu_2 U_{reg}$$

$$\frac{dI_{pre}}{dt} = \frac{\gamma D_{reg}}{D_{reg} + \eta_1 I_{pre} + \nu_1} - \mu_3 I_{pre}$$

$$\frac{dI_{post}}{dt} = \begin{cases} \delta D_{reg} - \mu_4 I_{post}, & \text{without } I_{pre} \text{ present} \\ \dfrac{\delta D_{reg}}{D_{reg} + \eta_2 I_{pre} + \nu_2} - \mu_4 I_{post}, & \text{with } I_{pre} \text{ present} \end{cases}$$

$$\frac{dR_{NA}}{dt} = \begin{cases} \varepsilon U_{reg} - \mu_5 R_{NA}, & \text{without } I_{pre} \text{ present} \\ \dfrac{\varepsilon U_{reg}}{U_{reg} + \eta_3 I_{pre} + \nu_3} - \mu_5 R_{NA}, & \text{with } I_{pre} \text{ present} \end{cases}$$

$$\frac{dP}{dt} = \begin{cases} \varphi R_{NA} - \mu_6 P, & \text{without } I_{post} \text{ present} \\ \dfrac{\varphi R_{NA}}{R_{NA} + \eta_4 I_{post} + \nu_4} - \mu_6 P, & \text{with } I_{post} \text{ present} \end{cases}$$

And put the intracellular signalling into quasi-steady state:

$$D_{reg} = \frac{\alpha}{\mu_1} C_{anti}$$

$$U_{reg} = \frac{\beta}{\mu_2} C_{pro}$$

$$I_{pre} = \frac{\nu_1 + D_{reg}}{2\eta_1}\left[-1+\left(1+\frac{4\gamma\eta_1 D_{reg}}{\mu_3(\nu_1 + D_{reg})^2}\right)^{1/2}\right] \approx \frac{\gamma/\mu_3 D_{reg}}{\nu_1 + D_{reg}}$$

$$I_{post} = \begin{cases} \delta/\mu_4 D_{reg}, & \text{without } I_{pre} \text{ present} \\ \dfrac{\delta/\mu_4 D_{reg}}{D_{reg} + \eta_2 I_{pre} + \nu_2}, & \text{with } I_{pre} \text{ present} \end{cases}$$

$$R_{NA} = \begin{cases} \varepsilon/\mu_5 U_{reg}, & \text{without } I_{pre} \text{ present} \\ \dfrac{\varepsilon/\mu_5 U_{reg}}{U_{reg} + \eta_3 I_{pre} + \nu_3}, & \text{with } I_{pre} \text{ present} \end{cases}$$

$$\frac{dP}{dt} = \begin{cases} \varphi/\mu_6 R_{NA}, & \text{without } I_{post} \text{ present} \\ \dfrac{\varphi/\mu_6 R_{NA}}{R_{NA} + \eta_4 I_{post} + \nu_4}, & \text{with } I_{post} \text{ present} \end{cases}$$

Note the use of approximation: $(1+\varepsilon)^{1/2} = 1 + \varepsilon/2$. Since the denominator is greater than the square of the numerator, we claim this approximation valid for all values.

In the simplest case, we assume post transcriptional but no pre-transcriptional inhibition, the algebra works out easily and we get a formula of the form:

$$\frac{dP}{dt} = \frac{V_{max} C_{pro}}{V_{sub} + V_I\, C_{anti} + C_{pro}} - \mu P$$

$$V_{MAX} = \varphi/\mu_6$$

$$V_{SUB} = \nu_4\left(\frac{\mu_2\,\mu_5}{\beta\,\epsilon}\right)$$

$$V_I = \eta_4\left(\frac{\alpha\,\delta}{\beta\,\epsilon}\right)\left(\frac{\mu_2\,\mu_5}{\mu_1\,\mu_4}\right)$$

Alternately, if we look at pre-transcriptional inhibition without post transcriptional, more algebra becomes involved. The formula for I_{PRE} is inserted into the equation for R_{NA}, then constants are multipled out.

$$R_{NA} = \frac{\frac{\varepsilon}{\mu_5} C_{PRO}}{C_{PRO} + \left(\frac{\frac{\mu_2 \gamma}{\mu_3 \beta} C_{ANTI}}{C_{ANTI} + \nu_1 \frac{\mu_1}{\alpha}} \right) + \nu_3 \frac{\mu_2}{\beta}}$$

Production of our product P is directly proportional to the amount of R_{NA} since there is no most transcriptional inhibition.

$$\frac{dP}{dt} = \frac{V_{max} C_{pro}}{V_s + \left(\frac{K_{max} C_{anti}}{K_s + C_{anti}} \right) + C_{pro}} - \mu P$$

$$V_{MAX} = \varphi \varepsilon \Big/ \mu_5 \mu_6$$

$$V_S = \nu_3 \frac{\mu_2}{\beta}$$

$$K_{MAX} = \eta_3 \left(\gamma \mu_2 \Big/ \beta \mu_3 \right)$$

$$K_S = \nu_1 \frac{\mu_1}{\alpha}$$

When both pre- and post-transcriptional inhibitions are involved, more algebraic manipulation and approximations are needed to come to a tractable equation. Simplification of terms is used for I_{POST} so that the final expression becomes a Michaelis-Menten function of C_{ANTI}. We can safely assume that $\mu_1 << \alpha$ so that their ratio is much less than one. When that

ratio is squared, it approximates zero. These terms we set to zero, and then factor out terms of C_{ANTI} which divide by each other to 1. When this has been accomplished we get the result:

$$I_{POST} \approx \frac{\frac{\mu_1 \delta}{\mu_4 \alpha} C_{ANTI}}{C_{ANTI} + \frac{\mu_1}{\alpha} \frac{\gamma}{\mu_3} + \nu_1}$$

Using this equation for I_{POST} and the equation for R_{NA} used above in the equation for product, after (at this point trivial) grouping of like terms, yields:

$$\frac{dP}{dt} = \frac{V_{max} C_{pro}}{C_{pro}\left(1 + \frac{K_{11} C_{anti} + K_{12}}{C_{anti} + K_{22}}\right) + V_s\left(\frac{K_{11} C_{anti} + K_{12}}{C_{anti} + K_{22}}\right)\left(\frac{C_{anti} + J_{12}}{C_{anti} + J_{22}}\right)} - \mu P$$

PROJECT CODE

The equation files were written in txt files with extension .ode, while the parameter estimation programming was written in C++.

A.2 MACROPHAGE_MODEL.ODE

```
#macrophage_model.ode

# Ian Price
# 25 September, 2008
# Revised December 4th, 2008

# This work has been a collaboration of Ian Price, David Swigon, G. Bard
# Ermentrout and Gilles Clermont-- at the University of Pittsburgh.

#####################################################################
# Inflammation Subsystem                                            #
#####################################################################
```

```
# Signalling equation

Sigma=a_1*TT

aux signal=Sigma

# Inhibition equation

Phi(Sigma,Inhib)= Sigma + (g_1*Inhib + g_2)/(Inhib + d_2)

par a_1=1, g_1=40, g_2=10.3, d_2=1

# Macrophages (M)

M'= b_mc*CT^m_hill/(a_mc + CT^m_hill) - mu_m*(M-b_m)

par b_mc=5.5, m_hill=2, a_mc=1, mu_m=0.4, b_m=5

# TNF-alpha in tissue (TT)

TT'= b_t*M*Sigma/(Sigma+Phi(Sigma,LT)*(k_1*LT+k_2)/(LT+f_0)) - mu_t*TT

par b_t=10, mu_t=6.5, k_1=.95, k_2=2, f_0=2.5

# Interleukin-10 in tissue (LT)

LT'=1/tau*(b_l*M*Sigma/Phi(Sigma,LT) - mu_l*LT)
```

```
par tau=80, b_l=20, mu_l=5

# Chemokines in tissue (CT)

CT'=b_c*M*Sigma/Phi(Sigma,LT) - mu_c*CT

par b_c=40, mu_c=8

#####################################################################
# Initial Conditions                                                #
#####################################################################

init M=5, TT=0, LT=0, CT=0

#####################################################################
# Numerical Methods - Call using Stiff Solver                       #
#####################################################################

@ maxstor=5000000
@ method=cvode, tol=1e-9, atol=1e-9
@ bounds=1000000
@ total=100

done
```

A.3 INFLAMMATION_TISSUE.ODE

```
#inflammation_tissue.ode

# Ian Price

# 05 December, 2008


# This file includes:

# (*) The inflammation subsystem

# (*) The neutrophil pathway

# (*) The relation to tissue


# This file is the final of a sequence of files

# 1. inflammation_basic.ode

# 2. inflammation_neutrophil.ode

# 3. inflammation_tissue.ode


# This work has been a collaboration of Ian Price, David Swigon, G. Bard

# Ermentrout and Gilles Clermont-- at the University of Pittsburgh.


#########################################################################

# Inflammation Subsystem                                                #

#########################################################################


# Signalling equation


Sigma=a_1*TT+a_2*(1-H)


par a_1=1, a_2=.7
```

```
aux signal=Sigma

# Inhibition equation

Phi(Sigma,Inhib)= Sigma + (g_1*Inhib + g_2)/(Inhib + d_2)

par g_1=40, g_2=10.3, d_2=1

# Macrophages (M)

M'= b_mc*CT^m_hill/(a_mc + CT^m_hill) - mu_m*(M-b_m)

par b_mc=5.5, m_hill=2, a_mc=1, mu_m=0.4, b_m=5

# TNF-alpha in tissue (TT)

TT'= b_t*M*Sigma/(Sigma+Phi(Sigma,LT)*(k_1*LT+k_2)/(LT+f_0)) - mu_t*TT

par b_t=10, mu_t=6.5, k_1=0.95, k_2=2, f_0=2.5

# Interleukin-10 in tissue (LT)

LT'=1/tau*(b_l*M*Sigma/Phi(Sigma,LT) - mu_l*LT)

par tau=80, b_l=20, mu_l=5

# Chemokines in tissue (CT)
```

```
CT'=b_c*M*Sigma/Phi(Sigma,LT) - mu_c*CT

par b_c=40, mu_c=8

######################################################################
# Neutrophil Pathway                                                 #
######################################################################

# Neutrophils in blood (NB)

NB'= b_nt*TT/(a_nt + a_nl*LT + TT) - mu_nb*NB - r_nc*NB*CT/(a_nc + CT)

par b_nt=5, a_nt=15, a_nl=40, mu_nb=.1, r_nc=10, a_nc=10

# Neutrophils in tissue (NT)

NT'= r_nc*NB*CT/(a_nc + CT) - mu_nt*NT

par mu_nt=0.03

# Free Radicals (X)

X'= b_xn*NT/(a_xn + NT) + b_xm*M*Sigma - mu_x*X - g_xh*H*X

par b_xn=50, a_xn=1.5, b_xm=.01, mu_x=8, g_xh=35
```

```
#####################################################################
# Tissue Equations                                                  #
#####################################################################

# Uninfected/undamaged/Target Tissue (H)

H'= b_h*H*(1-H)*(H-theta) - g_hx*H*X^x_hill/(a_hx + X^x_hill)

par b_h=.15, theta=0.05, g_hx=9, x_hill=2, a_hx=300

# Damaged Tissue (D)

aux D=1-H

#####################################################################
# Initial Conditions                                                #
#####################################################################

init M=5, TT=0, LT=0, CT=0

init NB=0, NT=0, X=0

init H=1

#####################################################################
# Numerical Methods - Call using Stiff Solver                       #
#####################################################################
```

```
@ method=cvode, tol=1e-9, atol=1e-9

#####################################################################
# Settings                                                          #
#####################################################################

@ xp=t, yp=h, xlo=0, xhi=100, ylo=-.1, yhi=1.1
@ maxstor=5000000
@ bounds=1000000
@ total=100

done
```

A.4 PATTERN FORMATION ODE FILE

```
# pattern_tissue.ode

# Declare user-defined functions

Sigma(TT,H)=a_1*TT+a_2*(1-H)

par a_1=1, a_2=.9

Phi(TT,L)= TT+(g_1*L+g_2)/(L+d_2)

par g_1=40, g_2=10.3, d_2=1
```

f1(M,C)=b_mc*C^h_m/(a_mc^h_m + C^h_m) - mu_m*(M-b_m)

par b_mc=5.5, h_m=2, a_mc=1, mu_m=0.4, b_m=5

f2(M,TT,L,H)= b_t*M*Sigma(TT,H)/(Sigma(TT,H)+Phi(Sigma(TT,H),L)*(k_1*L+k_2)/(L+f_0)) - mu_t*TT

par b_t=10, mu_t=6.5, k_1=0.95, k_2=2, f_0=2.5

f3(M,TT,L,H)=b_l*M*Sigma(TT,H)/Phi(Sigma(TT,H),L) - mu_l*L

par b_l=0.25, mu_l=0.0625

f4(M,TT,L,C,H)=b_c*M*Sigma(TT,H)/Phi(Sigma(TT,H),L) - mu_c*C

par b_c=40, mu_c=8

f5(TT,L,C,NB)= b_nt*TT/(a_nt + a_nl*L + TT) - mu_nb*NB - g_nc*NB*C/(a_nc + C)

par b_nt=5, a_nt=15, a_nl=40, mu_nb=.1, g_nc=10, a_nc=10

f6(C,NB,N)= g_nc*NB*C/(a_nc + C) - mu_n*N

par mu_n=0.03

f7(M,TT,N,X,H)= b_xn*N/(a_xn + N) + b_xm*M*Sigma(TT,H) - mu_x*X - g_xh*H*X

par b_xn=50, a_xn=1.5, b_xm=.01, mu_x=8, g_xh=35

f8(X,H)= b_h*H*(1-H)*(H-theta) - g_hx*H*X^h_x/(a_hx^h_x + X^h_x)

par b_h=.15, theta=0.05, g_hx=9, h_x=2, a_hx=17.3

periodic array conditions

M0=M100

M101=M1

TT0=TT100

TT101=TT1

LT0=LT100

LT101=LT1

CT0=CT100

CT101=CT1

NB0=NB100

NB101=NB1

NT0=NT100

NT101=NT1

X0=X100

X101=X1

H0=H100

H101=H1

par dx=1,

par Dm=.007, Dtt=.0081, Dl=.179, Dc=.0914, Dnb=.0034, Dn=.0037, Dxt=.265, Dh=6.63e-4

par chi_1=-8.02e-4, chi_2=-1.01e-4

Array Equations

M[1...100]'=f1(M[j],CT[j])+Dm/dx^2*(M[j+1]-2*M[j]+M[j-1])-chi_1/dx^2*(((M[j+1]-M[j])*heav(CT[j]-CT[j+1])+M[j])*(CT[j]-CT[j+1])+((M[j-1]-M[j])*heav(CT[j]-CT[j-1])+M[j])*(CT[j]-CT[j-1]))

TT[1...100]'=f2(M[j],TT[j],LT[j],H[j])+Dtt/dx^2*(TT[j+1]-2*TT[j]+TT[j-1])

LT[1...100]'=f3(M[j],TT[j],LT[j],H[j])+Dl/dx^2*(LT[j+1]-2*LT[j]+LT[j-1])

CT[1...100]'=f4(M[j],TT[j],LT[j],CT[j],H[j])+Dc/dx^2*(CT[j+1]-2*CT[j]+CT[j-1])

NB[1...100]'=f5(TT[j],LT[j],CT[j],NB[j])+Dnb/dx^2*(NB[j+1]-2*NB[j]+NB[j-1])

NT[1...100]'=f6(CT[j],NB[j],NT[j])+Dn/dx^2*(NT[j+1]-2*NT[j]+NT[j-1]) -chi_2/dx^2*(((NT[j+1]-NT[j])*heav(CT[j]-CT[j+1])+NT[j])*(CT[j]-CT[j+1])+((NT[j-1]-NT[j])*heav(CT[j]-CT[j-1])+NT[j])*(CT[j]-CT[j-1]))

X[1...100]'=f7(M[j],TT[j],NT[j],X[j],H[j])+Dxt/dx^2*(X[j+1]-2*X[j]+X[j-1])

H[1...100]'=f8(X[j],H[j])+Dh/dx^2*(H[j+1]-2*H[j]+H[j-1])

Initial Conditions

M[1...100](0)=5

TT[1...100](0)=0

LT[1...100](0)=0

CT[1...100](0)=0

NB[1...100](0)=0

NT[1...100](0)=0

X[1...100](0)=0

H[1...100](0)=1

Numerics

@ meth=rk, tol=1e-7

@ dt=.005

@ total=100, nout=40

@ maxstor=500000

done

A.5 INFLUENZA ODE FILE

```
# influenza_2011_03.ode

# Ian Price
# Updated 01 March, 2011

# This file includes:
# (*) The inflammation subsystem
# (*) The neutrophil pathway
# (*) The relation to undamaged/uninfected/healthy tissue
# (*) Virus and infected tissue
# (*) Interferon a/B and cell resistance
# (*) Natural Killer Cells and Interferon Gamma
# (*) Antigen Presenting Cells and Effector Cells
# (*) B-Cells and Antibodies

# This file is an update of the fifth of a sequence of files
# 1. influenza_basic.ode
# 2. influenza_resist.ode
# 3. influenza_nk.ode
# 4. influenza_ctl.ode
```

```
# 5. influenza_complete.ode

# This work has been a collaboration of Ian Price, David Swigon, G. Bard
# Ermentrout and Gilles Clermont-- at the University of Pittsburgh.

##############################################################
# User Defined Functions                                     #
##############################################################

# Signalling equations

Sigma1=a_11*TT + a_12*D

par a_11=6.3e-3, a_12=1.4e-4

Sigma2=Sigma1+(a_21*V)/(a_22+V)

par a_21=318, a_22=3960

aux signal1=Sigma1
aux signal2=Sigma2

# Inhibition equation

Phi(Sigma,Inhib)= Sigma + (g_1*Inhib + g_2)/(Inhib + d_2)

par g_1=32.4, g_2=4930, d_2=86.8
```

```
# Resistance (R)

R=F/(a_rf+F)

# Michaelis-Menten Function

MM(x,b,a)=b*x/(x+a)

# Hill Function

Hill(x,a,h)=x^h/(x^h+a^h)

# Most parameters are letter followed by and underscore and other letters:
# 'b' are production terms and always in the numerator,
# 'a' terms stay in the denominator,
# 'g' represent transfer from one category to another else destruction,
# 'h' are hill coefficients, and
# 'mu' is used for decay terms.
# Parameters followed by numbers have more complicated uses.

############################################################
# Inflammation Subsystem                                   #
############################################################

# Macrophages (M)
# Depends on M, CT

M'= b_mc*Hill(CT,a_mc,h_m) - mu_m*(M-b_m)
```

par b_mc=1.80e4, a_mc=487, mu_m=0.0852, b_m=5470

TNF-alpha in tissue (TT)

Depends on M, TT, LT, H, I, V

TT'= b_t*M*Sigma2/(Sigma2+Phi(Sigma2,LT)*(k_1*LT+k_2)/(LT+d_1)) - mu_t*TT

par b_t=0.414, k_1=1.97, k_2=83.8, d_1=123, mu_t=112

Interleukin-10 in tissue (LT)

Depends on M, TT, LT, H, I, F

LT'=b_l*M*Sigma1/Phi(Sigma1,LT) - mu_l*(LT - b_lh*(1-R)*H)

par b_l=0.764, mu_l=4.92, b_lh=3.64e-4

Chemokines in tissue (CT)

Depends on M, TT, LT, CT, H, I, DH

CT'=b_c*M*Sigma1/Phi(Sigma1,LT) - mu_c*CT

par b_c=35.5, mu_c=33.5

##

Neutrophil Pathway

##

```
# Neutrophils in blood (NB)

# Depends on TT, LT, NB, CT

NB'= b_nt*TT/(a_nt + a_nl*LT + TT) - NB*MM(CT,g_nc,a_nc) - mu_n*NB

par b_nt=8.55e5, a_nt=35.2, a_nl=0.201, g_nc=80.6, a_nc=106, mu_nb=0.277

# Neutrophils in tissue (NT)

# Depends on CT, NB, NT

NT'= NB*MM(CT,g_nc,a_nc) - mu_n*NT

par mu_n=1.26

# Free Radicals (X)

# Depends on NT, X, H, I

X'= MM(NT,b_xn,a_xn)- g_xi*I*X - g_xh*H*X  - mu_x*X

par b_xn=7.68e1, a_xn=3.13e3, g_xi=9.03e-7, g_xh=7.15e-6, mu_x=34.1

############################################################
# Tissue Equations                                         #
############################################################

# Uninfected/Undamaged Tissue (H)

# Depends on F, H, I, DH, V, X
```

H'= b_h*(1-R)*H*D*(H-theta)/tot_cell - g_hv*V*H - g_hx*H*Hill(X,a_hx,h_x)

par b_h=6.7e-03, theta=3.13e4, g_hv=3.83e-06, g_hx=49.5, a_hx=4.07

Uninfected/Undamagedd Tissue Death (DH)

Depends H, X, F, DH

DH'= g_hx*H*Hill(X,a_hx,h_x) - b_h*(1-R)*H*DH*(H-theta)/tot_cell

Infected Tissue (I)

Depends on V, H F, I, X, K, E

I'= g_hv*V*H - g_ix*I*Hill(X,a_ix,h_x) - g_ik*R*I*K -g_ie*R*I*E - mu_i*(1-R)*I

par g_ix=5.15, a_ix=1.34, g_ik=6.44e-05, g_ie=4.92e-5, mu_i=1.52

Infected Tissue Death (DI)

DI=tot_cell-H-I-DH

aux dmgi=DI

Damaged Tissue (D)

D=tot_cell-H-I

aux dmg=D

```
############################################################
# Virus                                                    #
############################################################

# Depends on V, I, F, H, A

V'= g_vi*(1-R)*I - g_vh*H*V - g_va*V*A - g_v*V/(1+a_v*V) - mu_v*V

par g_vi=226, g_vh=2.02e-05, g_va=1.4e-3, g_v=1.25, a_v=0.301, mu_v=1.51

aux logv=log10(V+eps)

#g_vi/g_iv is the burst size.  increasing g_vi effectively makes
# a more deadly virus

############################################################
# Interferon(alpha/beta) and Resistance                    #
############################################################

# Interferon alpha/beta (F)
# Depends on F, I, PA

F'= b_fi*(1-R)*I + b_fp*P - g_fi*I*F - mu_f*F

par b_fi=3.01, b_fp=1.18e-2, g_fi=1.05e-2, mu_f=8.04

aux logF=log10(F+eps)
```

```
aux resist=R

par a_rf=36

############################################################
# Natural Killer Cells and Interferon-gamma                #
############################################################

# NK Cells (K)
# Depends on CT, K, I, F

K'=b_kc*Hill(CT,a_kc,h_k) - g_ki*R*I*K - mu_k*(K-b_k)

par b_k=1090, b_kc=5.39e4, a_kc=866, g_ki=2.66e-8, mu_k=0.588

# Interferon-gamma (G)
# Depends on W, O, K, G

G'=MM(W,b_go,a_go)*O + MM(W,b_gk,a_gk)*K - mu_g*G

par b_go=2.2e-3, g_go=8.25e-5, a_go=16.5, b_gk=5.38e-1, g_gk=1.7e-5, a_gk=1.18e2, mu_g=2.06e1

aux logG = log10(G+eps)

############################################################
# Dendritic Cells and CTL                                  #
############################################################
```

```
# Activated Dendritic Cells (PA)
# Depends on V, H, I, DH, G, P

P'= p_0*(MM(V,g_pv,a_pv)+g_pi*DI)*(g_p + MM(G,b_pg,a_pg)) - mu_p*(P - b_p)

par p_0=1.41e4, g_pv=0.306, a_pv=3.48e3, g_pi=3.52e-05, g_p=0.483, b_pg=2.04, a_pg=213
par mu_p=1.98e-1, b_p=1.11e3

aux logP = log10(P+eps)

#Effector Cells/CTL (E)
# Depends on P, F, I, E

E'=b_ep*Hill(P,a_ep,h_e) - b_ei*R*I*E - mu_e*E

par  b_ep=1.07e5, a_ep=1.01e4, b_ei=2.33e-06, mu_e=2.73e-1

aux logE=log10(E+eps)

############################################################
# B-Cells and Antibodies                                   #
############################################################

# Th-1 Helper Cells (O)
# Depends on PA, O

O'=b_op*Hill(P,a_op,h_o) - mu_o*O
```

par b_op=1.43e5, a_op=7.37e3, mu_o=2.92e-1

aux logO=log10(O+eps)

IL-12 (W)

Depends on O, P, W

W'=MM(O,b_wo,a_wo)*P - mu_w*W

par b_w=1.39e-10, b_wo=6.80e-2, a_wo=1.65e4, mu_w=4.56

Mature B-Cells

Depends on W, P, B

B'=b_b + b_bp*W*P*(b_0-B) - mu_b*B

par b_b=42.4, b_bp=1.77e-6, b_0=4.05e4, mu_b=1.37e-1

aux logb=log10(B+eps)

Antibodies (A)

Depends on B, A, V

A'=b_a + b_ab*B - g_av*A*V - mu_a*A

par b_a=4.46e-2, b_ab=1.30e-2, g_av=8.37e-06, mu_a=0.665

```
#############################################################
# Hill Terms                                                #
#############################################################

# Because Hill terms do not vary continuously in the real line,
# they are collected here and varied statically.

par h_m=3, h_x=2, h_k=2, h_e=3, h_o=2

# This is strictly a term of convenience for plotting logs
par eps=1e-8

# Number of epithelial cells in the system
par tot_cell=250000

#############################################################
# Initial Conditions                                        #
#############################################################

init M=0, TT=0, LT=0, CT=0

global 0 t {M=b_m; LT=b_lh*H}

init NB=0, NT=0, X=0

init H=250000, I=0, DH=0, V=50

init F=0, K=0
```

```
global 0 t {F=b_fp*b_p/mu_f; K=b_k}

init P=0, E=0

global 0 t {P=b_p; E=b_ep/mu_e*Hill(b_p,a_ep,h_e)}

# G is placed out of sequence because it depends on W, O, and K.
init O=0, W=0, G=0

global 0 t {O=b_op/mu_o*Hill(b_p,a_op,h_o); W=b_p/mu_w*b_wo*O/(a_wo+O)}

global 0 t {G= O/mu_g*MM(W,b_go,a_go) + K/mu_g*MM(W,b_gk,a_gk)}

init B=0 A=0

global 0 t {B=(b_b+b_bp*W*b_p*b_0)/(b_bp*W*b_p+mu_b); A=(b_a+b_ab*B)/mu_a}

############################################################
# Numerical Methods - Call using Stiff Solver              #
############################################################

@ dt=.01
@ method=cvode, tol=1e-9, atol=1e-9, mxstep=1e5
@ bandup=9, bandlo=9

@ maxstor=5000000
@ bounds=100000000
```

```
@ total=14

done
```

A.6 MOSQUITO MODEL 2D

```
# Mosquito model (competitive inhibition)
# Ian Price
# 2011, Feb 07

par mu_n=1.6, b_n=0.5, p=2,a_n=5.5, a_nx=1.28, q=1.7
par mu_x=0.67, b_xn=27.6, a_xn=10.35
par a=42.5

g(N,X)=a*N^p/(a_n^p + a_nx*X^q + N^p)

N'=mu_n*(b_n-N) + g(N,X)

X'=-mu_x*X + b_xn*N/(a_xn + N)

init N=1, X=7

@ maxstor=50000000
@ method=cvode, tol=1e-9, atol=1e-9
@ bounds=1000000
@ total=40
```

done

A.7 MOSQUITO 5D ODE FILE

```
# LAXNS_v2.ode

# 2010 - 08 - 04


# Ian Price

# in conjunction with Bard Ermentrout, Yoram Vodovotz, and Shirley Luckhart


# L-A-X-N-S system, time is in hours


# The model describes the effect of exogenous TGF-B entering the

# midgut of the mosquito, and the effects on the endogenous system

# of regulating AsNOS levels


# Period of oscillation is adjustable

# parameter period is the user defined period of oscillation for the endogenous subsystem


par period=8


# Latent TGF-B (L)


L'=-b_l*(N/(a_ln+N))*(L/(a_l + L)) - mu_l*L


par b_l=0.2, a_ln=2, a_l=50, mu_l=.3
```

```
# Active TGF-B (A)

A'= b_l*(N/(a_ln+N))*L/(a_l + L) - mu_a*A

par mu_a=.25

# Combined signal of As60A and Active TGF-B
# c_r is the affinity of TGF-B for the same receptors as As60A

Sigma= S + c_r*A

par c_r=.25

# AsNOS (N)

N'=(mu_n*(b_n-N) + f(Sigma,X))/period

par mu_n=19.3, b_n=.628

# Inhibitor MEK/ERK (X)

X'=(b_xn*N^r1/(a_xn^r1 + N^r1) - mu_x*X)/period

par b_xn=437.9, a_xn=9.93, r1=2 , mu_x=7.73

# As60A

S'=(b_sn*N^r2/(a_sn^r2 + N^r2) - mu_s*S)/period
```

par b_sn=966 , a_sn=5.58 , r2=3 , mu_s=51.5

function for production of AsNOS

f(S,X)= b_ns*S^p1/(S^p1 + a_ns^p1 + a_nx^p1*X^q1)

par b_ns=1515, p1=1, a_ns=16, a_nx=8 , q1=2

init L=2, A=0, N=1.001, X=0.9259, S=0.1090

@ total=50

@ bounds=3000

@ method=cvode, tol=1e-9, atol=1e-9

@ maxstor=500000

done

A.8 INFLUENZA PARALLEL TEMPERING CODE

```
/* logtemper_multi_mpi.cpp
   mpic++ logtemper_multi_mpi.cpp -w -llapack -lblas -lsundials_cvode -lsundials_nvecserial -o mpi_temper.x

   24 March, 2011
   This file uses a Metropolis Hastings Method-- Markov Chain Monte Carlo --to explore the
   parameter space.  Random movements are selected probabilistically based on an objective
   function.  The objective function is fit to measured data points.  The measurements are
```

```
  compared to trajectories on a log scale.
 */

// Usual pre-processor stuff:
#include <iostream>
#include <cmath>
#include <cstdio>
#include <ctime>
#include <cstdlib>
#include <cstring>
#include <exception>
#include <algorithm>
#include <fstream>
#include "mpi.h"
using namespace std;

// Define global parameters:
#define inter    10   // runs between swaps
#define totpar   95    // total number of parameters
#define nchain   12    // total number of temperature chains
#define stor     1     // number of iterations between save points
#define maxstor  500  // maximum storage capacity
#define HMIN     5.0e4
#define VMIN     50.0

// CVODE pre-processor stuff:
#include <cvode/cvode.h>            // prototypes for CVODE fcts. and consts
#include <cvode/cvode_lapack.h>     // prototype for CVLapackDense
```

```
#include <nvector/nvector_serial.h>  // serial N_Vector types, fcts, and macros
#define Ith(v,i)    NV_Ith_S(v,i-1)  // Ith numbers components 1..NEQ
#define IJth(C,i,j) BAND_COL_ELEM(C,i-1,j-1)

// CVODE Constants:
#define NEQ   20               // number of equations
#define RTOL  RCONST(1.0e-08)    // scalar relative tolerance
#define ATOL  RCONST(1.0e-08)    // assume use same abs tol for each var
#define T0    RCONST(0.0)       // initial time
#define T1    RCONST(0.5)
#define TADD  RCONST(0.5)
#define TEND  RCONST(11)
#define ZERO  RCONST(0.0)
#define MAXSTEPS RCONST(1e6)
#define mupper 8
#define mlower 8
int NOUT=(int)(TEND/TADD)+1;

// these are specific to sublethal/lethal
#define vars_sublethal 15    // number of variables for which we have data
#define time_sublethal 8     // number of time points for which we use data
double  pnts_sublethal=(double)(vars_sublethal*time_sublethal);
#define vars_lethal    12
#define time_lethal    6
double  pnts_lethal=(double)(vars_lethal*time_lethal);

// Parallel Tempering Prototypes:
int obj_sublethal(double *, double*, double[][vars_sublethal], double[][vars_sublethal]);
```

```
int obj_lethal(double *, double*, double[][vars_lethal], double[][vars_lethal]);
int inf_traj(double y[][NEQ]);
void newpoint(double*, double*, double, double[], double[]);
int read_in(char[], double[][vars_sublethal], double[][vars_sublethal]);
int read_in(char[], double[][vars_lethal], double[][vars_lethal]);
double geom_mean(double, double);
int heav(realtype arg);
double sheav(double, double);
double slog10(double);
double Hill(double, double, int);

// obj_sublethal computes an objective function value for a par set called by the main program
// inf_traj computes trajectories and is called by obj_sublethal
// newpoint computes a new par set from a previous one and is called by the main program
// read_in is a subroutine that reads the data files, called in the main program
// heav returns the heaviside function value
// slog10 returns log10 restricted to finite return values
// Hill returns the hill function value

// CVODE Function Prototypes:
static int f(realtype t, N_Vector y, N_Vector ydot, void *user_data);

static int Jac(int N, int mu, int ml,
               realtype t, N_Vector y, N_Vector fy,
           DlsMat J, void *user_data,
           N_Vector tmp1, N_Vector tmp2, N_Vector tmp3);

static int check_flag(void *flagvalue, char *funcname, int opt);
```

```
// Global parameters:
void set_parameters(double*);
// the set_parameters prototype is a little hacky, as well as how parameters are called
// a parameter set needs to be simultaneously accesed by inf_traj, f, and Jac
// the best way to do this escapes me, thus the current set up
realtype v0, a_11, a_12, a_21, a_22, g_1, g_2, d_2;
realtype b_mc, a_mc, mu_m, b_m;
realtype b_t, k_1, k_2, d_1, mu_t;
realtype b_l, mu_l, b_lh, b_c, mu_c;
realtype b_nt, a_nt, a_nl, g_nc, a_nc, mu_nb, mu_n;
realtype b_xn, a_xn, g_xi, g_xh, mu_x;
realtype b_h, theta, g_hv, g_hx, a_hx;
realtype g_ix, a_ix, g_ik, g_ie, mu_i;
realtype g_vi, g_vh, g_va, g_v, a_v, mu_v;
realtype b_fi, b_fp, g_fi, mu_f, a_rf;
realtype b_k, b_kc, a_kc, g_ki, mu_k;
realtype b_go, g_go, a_go, b_gk, g_gk, a_gk, mu_g;
realtype p_0, g_pv, a_pv, g_pi, g_p;
realtype b_pg, a_pg, mu_p, b_p;
realtype b_ep, a_ep, g_ei, mu_e;
realtype b_op, a_op, mu_o;
realtype b_w, b_wo, a_wo, mu_w;
realtype b_b, b_bp, b_0, mu_b;
realtype b_a, b_ab, g_av, mu_a;

int h_m=3, h_x=2, h_k=2, h_e=3, h_o=2, tot_cell=2.5e5;
```

```
////////////////////
// MAIN FUNCTION //
////////////////////



int main()
{
  // clock_t init, final;
  // init=clock();

  // --------------------------------------------------------------------
  //    MPI Initialize commands:
  // --------------------------------------------------------------------
  MPI::Status status;
  int i, j, numsent, sender, sendee;
  int anstype;

  MPI::Init();
  int myid = MPI::COMM_WORLD.Get_rank();
  int numprocs = MPI::COMM_WORLD.Get_size();

  int master = 0;  // Master is the 0'th processor
  double starttime, endtime;

  // For sending tuplets of info:
  int extra = 20;
  double xc[totpar+1+(2*inter*totpar+extra)];
```

```
double xcout[totpar + 3];
int accept, reject;

double randxc[2*inter*totpar+extra];
double xnewvect[totpar];

// ---------------------------------------------------------------------
//    Read in data etc.:
// ---------------------------------------------------------------------
// Pointers to mount files:
FILE *f_in, *fpt;

// The following should be read from init.dat:
int N; //N is the number of runs:
double beta[nchain];
double eps[nchain];
double prior[totpar]; //this is for our specific obj fxn
int burn; // number of burn-in runs

// The basic values should be read from init.dat:
if((f_in=fopen("init.dat","r"))==NULL){
  printf("\nERROR - Need file init.dat\n");
  return(0);}
else {
  fscanf(f_in, "%d", &N);
  for(int a=0; a<nchain;a++){
    fscanf(f_in, "%lg", &beta[a]);}
  for(int a=0; a<nchain; a++){
```

```
    fscanf(f_in, "%lg", &eps[a]);}
  for(int a=0; a<totpar; a++){
    fscanf(f_in, "%lg", &prior[a]);}
  fscanf(f_in, "%d", &burn);
  fclose(f_in);
}

// Scan in data:

double dat_sublethal[time_sublethal][vars_sublethal];
double std_sublethal[time_sublethal][vars_sublethal];

double dat_lethal[time_lethal][vars_lethal];
double std_lethal[time_lethal][vars_lethal];

int k;
k=read_in("sublethal",dat_sublethal,std_sublethal);
if (k==0) {
  cout << "Trouble in data read-in" << endl;
  return(0);}
k=read_in("lethal",dat_lethal,std_lethal);
if (k==0) {
  cout << "Trouble in data read-in" << endl;
  return(0);
}

// Scan in bounds on parameters:
```

```
double lbounds[totpar];
double ubounds[totpar];

if((f_in=fopen("bounds.dat","r"))==NULL){
  printf("\nERROR - Need file bounds.dat\n");
  return(0);}
else {
  for(int a=0; a<totpar; a++) {
    fscanf(f_in, "%lg", &lbounds[a]);}
  for(int a=0; a<totpar; a++) {
    fscanf(f_in, "%lg", &ubounds[a]);}
  fclose(f_in);
}

////////////////////////////////////
// ESTABLISH STORAGE SPACE FOR FILE //
////////////////////////////////////

// establish parameters for the routine
int flag1=0;
int flag2=0;

// space for accepted/rejected
int accepted[nchain]; // stores the number of accepted ones
int rejected[nchain]; // stores the number of rejected ones

// space for swap accepted/rejected
int swapAccepted[nchain-1];
```

```
int swapRejected[nchain-1];

// space for parameter values
double xold[nchain][totpar];
double xnew[nchain][totpar];

double parVecLo[maxstor/stor+1][totpar]; //store the lowest temperature chain
double parVecSk[maxstor/stor+1][totpar]; //store a middle temperature chain
double parVecHi[maxstor/stor+1][totpar]; //store the highest temperature chain

// space for energy values
double energyOld[nchain];  // energy for previous parameter
double energyNew;
double energy_sublethal, energy_lethal;

// space for  the energy value for each chain
double energyVec[maxstor/stor+1][nchain];

// error checking values
double err=0, errin=0;

// space for rates
double rate[nchain];
double swapRate[nchain-1];

// ----------------------------------------------------------
//          MASTER PROCESS
// ----------------------------------------------------------
```

```
if (myid ==0){
  double starttime, endtime;
  starttime = MPI::Wtime();

  //initialise values for accepted/rejected
  for(int a=0; a<nchain; ++a){
    accepted[a]=1;
   rejected[a]=0;
  }

  // initialise values for swap accepted/rejected
  for( int b=0; b < (nchain-1); ++b){
    swapAccepted[b]=0;
    swapRejected[b]=0;
  }

  // Instead of an initial values being given, choose it randomly.
  // Pick points randomly about the prior.

  // Randomly seed pseudo-random number generator using clock time
  unsigned int seed = time(NULL);
  srand48(seed);

  while(1){
    for( int c=0; c<nchain; c++){
          newpoint(*xold+c*totpar, prior, eps[c], lbounds, ubounds);
          flag1=obj_sublethal(&energy_sublethal,*xold+c*totpar,dat_sublethal,std_sublethal);
```

```
        flag2=obj_lethal(&energy_lethal,*xold+c*totpar,dat_lethal,std_lethal);
        // the objective function is the geometric mean with bias based on ratio of data points
        energyOld[c]=geom_mean( energy_sublethal, energy_lethal );
        if ( max(flag1,flag2)==1 ) {continue;}
    }
   break;
  }

  for( int d=0; d<totpar; d++){
    parVecLo[0][d]=xold[0][d];
    parVecSk[0][d]=xold[2][d];
    parVecHi[0][d]=xold[nchain-1][d];
  }

  for(int e=0; e<nchain; e++){
    energyVec[0][e] = energyOld[e]; // assign initial energy to first row
  }

  cout << "Iteration 0" << " of " << N << " completed" <<endl;

} // end myid ==0

//////////////////////////
// ENTER MAIN ALGORITHM //
//////////////////////////

for (int i=1; i<= N; i++){
```

```
// ------------------------------------------------------------
//          MASTER PROCESS
// ------------------------------------------------------------
if (myid == 0){

  // 1. Run each chain separately.  Possible place to parallelise
  for (int k = 0; k < nchain; k++){

        // Here, we concatenate all the info we want for a chain into one big vector xc:
        for (int pp=0; pp<totpar; pp++){
          xc[pp] = xold[k][pp];  // **Math check 1.  Is this right? (k*totpar stuff)
        }
        xc[totpar] = energyOld[k];

    // Generate random numbers starting at totpar+1+pp = info_num + pp, where pp is new index over rands
        for (int pp=0; pp < 2*inter*totpar+extra; pp++){
          xc[totpar+1 + pp] = drand48();
        }
        // Sends info vector xc to processor k+1 (since k=0 is master) and tags it with k, the chain index
        MPI::COMM_WORLD.Send(xc, totpar+1+2*inter*totpar+extra, MPI::DOUBLE, k+1, k);
  }

  for (int k = 0; k < nchain; k++){
        MPI::COMM_WORLD.Recv(xcout, totpar+3, MPI::DOUBLE, MPI::ANY_SOURCE, MPI::ANY_TAG,
status);
        anstype = status.Get_tag();
```

```
        // To be deleted, but informative for now while loops are small:

        cout << "receiving chain " << anstype << " info at iteration " << i << " from proc " << status.Get_source()
<< endl;



        for (int pp=0; pp<totpar; pp++){

          xold[anstype][pp] = xcout[pp];

        }

        energyOld[anstype] = xcout[totpar];

        accepted[anstype] = accepted[anstype] + xcout[totpar+1]; // **Check math 3: are acceptance/rejections
computed correctly?

        rejected[anstype] = rejected[anstype] + xcout[totpar+2];



    } // End 1.



    // 2. store and print information and statistics about run esp every 5th:

    if ((i>burn)&&(i%stor==0)){

        for( int a=0; a<totpar; a++){

          parVecLo[(i%maxstor)/stor][a]=xold[0][a];

          parVecSk[(i%maxstor)/stor][a]=xold[2][a];

          parVecHi[(i%maxstor)/stor][a]=xold[nchain-1][a];}

        for(int b=0; b<nchain; b++){

          energyVec[(i%maxstor)/stor][b]=energyOld[b];}

        cout << "Iteration " << i << " of " << N << " completed" <<endl;

    } // End 2.



    // ---------------------------------------------------------------------

    //     SWAPPING BETWEEN CHAINS:

    // ---------------------------------------------------------------------

    // 3. swap chains:
```

```
for (int kk = (nchain-1); kk>0; kk--){
        double r=drand48();
        if ((beta[kk]-beta[kk-1])*(energyOld[kk]-energyOld[kk-1]) > log(r)){
          // Exchange Energies:
          energyNew = energyOld[kk];
          energyOld[kk] = energyOld[kk-1];
          energyOld[kk-1] = energyNew;
          // Exchange vectors
          for (int j=0; j<totpar; j++){
            xnew[kk][j] = xold[kk][j];
            xold[kk][j] = xold[kk-1][j];
            xold[kk-1][j] = xnew[kk][j];}
          if (i>=burn)
            ++swapAccepted[kk-1];
        }
        else {
          if (i>=burn)
            ++swapRejected[kk-1];
        }
} // End 3.



// ------------------------------------------------------------------
//    Write to file:
// ------------------------------------------------------------------
// 4.  Every maxstor steps, print file
if ((i>burn)&&(i%maxstor==0)){
        char nrg_str[32], par_str1[32], par_str2[32], par_str3[32];
```

```
int mods = (int)(i/maxstor);

// Add zeros to numbers to sort automatically in files
if (mods<10){
  sprintf(par_str1, "parameters_low_00%d.dat", mods);
  sprintf(par_str2, "parameters_skew_00%d.dat", mods);
  sprintf(par_str3, "parameters_high_00%d.dat", mods);
  sprintf(nrg_str, "energies_00%d.dat", mods); }
else if ((mods>=10)&&(mods<100)) {
  sprintf(par_str1, "parameters_low_0%d.dat", mods);
  sprintf(par_str2, "parameters_skew_0%d.dat",mods);
  sprintf(par_str3, "parameters_high_0%d.dat", mods);
  sprintf(nrg_str, "energies_0%d.dat", mods); }
else {
  sprintf(par_str1, "parameters_low_%d.dat", mods);
  sprintf(par_str2, "parameters_skew_%d.dat",mods);
  sprintf(par_str3, "parameters_high_%d.dat", mods);
  sprintf(nrg_str, "energies_%d.dat", mods);
}


// low energy parameter file
fpt=fopen(par_str1,"w");
fprintf(fpt, "Parameters:\n");
for(int a=0; a<totpar; a++){
  fprintf(fpt, "#%d\t", a);}
fprintf(fpt,"\n");
```

```
for(int a=0; a<maxstor/stor; a++){
  for(int b=0; b<totpar; b++){
    fprintf(fpt,"%.2e\t",parVecLo[a][b]);}
  fprintf(fpt,"\n");}
fclose(fpt);

// skew energy parameter file
fpt=fopen(par_str2,"w");
fprintf(fpt, "Parameters:\n");
for(int a=0; a<totpar; a++){
  fprintf(fpt, "#%d\t", a);}
fprintf(fpt,"\n");
for(int a=0; a<maxstor/stor; a++){
  for(int b=0; b<totpar; b++){
    fprintf(fpt,"%.2e\t",parVecSk[a][b]);}
  fprintf(fpt,"\n");}
fclose(fpt);

// high energy parameter file
fpt=fopen(par_str3,"w");
fprintf(fpt, "Parameters:\n");
for(int a=0; a<totpar; a++){
  fprintf(fpt, "#%d\t", a);}
fprintf(fpt,"\n");
for(int a=0; a<maxstor/stor; a++){
  for(int b=0; b<totpar; b++){
    fprintf(fpt,"%.2e\t",parVecHi[a][b]);}
  fprintf(fpt,"\n");}
```

```
        fclose(fpt);


        // energy file
        fpt=fopen(nrg_str,"w");
        fprintf(fpt, "Energy for Chains:\n");
        for(int a=0; a<nchain; a++){
          fprintf(fpt, "#%d\t", a+1);}
        fprintf(fpt,"\n");
        for (int a=0; a<maxstor/stor; a++){
          for(int b=0; b<nchain; b++){
            fprintf(fpt, "%.2f\t", energyVec[a][b]);}
          fprintf(fpt,"\n");}
        fclose(fpt);
    } // End 4.



  } // end myid == 0


  // ------------------------------------------------------------
  //          SLAVE PROCESSES
  // ------------------------------------------------------------


  else{


    // ------------------------------------------------------------------------------------------
    MPI::COMM_WORLD.Recv(xc, totpar+1+2*inter*totpar+extra, MPI::DOUBLE, master, MPI::ANY_TAG,
status);
    k = status.Get_tag();    // specifies tag of the message received - here's it's chain index
```

```
for (int pp =0; pp<totpar; pp++){
        xold[k][pp] = xc[pp];
}
energyOld[k] = xc[totpar];

for (int pp=0; pp<2*inter*totpar+extra; pp++){
        randxc[pp] = xc[totpar+1+pp];
}
int randind = 0;
int accept_here = 0;
int reject_here = 0;

// randnums = xc[totpar+1:end]
// ---------------------------------------------------------------------------------------------

// 1a. Run chain
for (int m = 0; m < inter; m++){  //swap each interim # of steps
        // for (int m = 0; m < 2; m++){  //swap each interim # of steps

        // A. Produce parameter sets that do not crash the obj fxn
        err = 0; // delele
        errin = err;
        //while (err - errin < 500){
        while (err - errin < extra){

          // i. Generate new set of parameters
```

```
  // newpoint(*xnew+k*totpar, *xold+k*totpar, eps[k], lbounds, ubounds);
  // ------------------------------------------------------------------------------------------
  //   New point function copied here (for random number generation):

 for(int i=1; i<totpar; i++){ // **Check math 4.  How does this work for the indixes?

  //xnew[i]=xc[i]*exp(2*(drand48()-.5)*eps*ubounds[i]/lbounds[i]);
  // Random number from xc[]:
  //xnew[k][i]= xold[k][i]*exp(2*(xc[totpar+1+2*i*m]-.5)*eps[k]*ubounds[i]/lbounds[i]);

  xnew[k][i]= xold[k][i]*exp(2*(randxc[randind]-.5)*eps[k]*ubounds[i]/lbounds[i]);
  ++randind;

  if(fabs(xnew[k][i])>ubounds[i])
    xnew[k][i]=lbounds[i]/ubounds[i]*fabs(xnew[k][i]);
  else if (fabs(xnew[k][i])<lbounds[i])
    xnew[k][i]=ubounds[i]/lbounds[i]*fabs(xnew[k][i]);
 }
// ------------------------------------------------------------------------------------------
 for (int i=0;i<totpar; i++){
  xnewvect[i] = xnew[k][i];
 }

 // ii. Filter out parameter sets that don't integrate
 // flag = obj_sublethal(&energyNew, *xnew+k*totpar, dat_sublethal, std_sublethal);
 flag1 = obj_sublethal(&energy_sublethal, xnewvect, dat_sublethal, std_sublethal);
 flag2 = obj_lethal(&energy_lethal, xnewvect, dat_lethal, std_lethal);
 // the objective function is the geometric mean with bias based on ratio of data points
```

```
energyNew=geom_mean( energy_sublethal , energy_lethal );
if ( max(flag1,flag2)==1 ) {
  ++err; //
  // provide new random number set
  continue; //sets that fail disregarded, return to while loop
}

// iii. Successful parameter set, proceeds to test probability
else {break;};
}  // end while loop

// Only allow limited number of parameter sets to fail before exiting
if (err - errin >= extra){
  //something's wrong, error msg, then exit program
  cout <<"Infinite loop in new parameter selection"<<endl;
  cout << "in process " << myid << " and randin = " << randind << endl;
  return(0);}

// B. Select old or new parameter set
//double r=drand48();
//double r = xc[totpar+1+2*i*m+1];
double r = randxc[randind];
++randind;

if ( beta[k]*(energyOld[k]-energyNew) > log(r) ) {
  for (int j=0; j<totpar; j++){
    xold[k][j]=xnew[k][j];}
  energyOld[k] = energyNew;
```

```
            if (i>=burn) { //only accept/reject after burn-in
              ++accept_here;}}
           else {
            if (i>=burn) {//only accept/reject after burn-in
              ++reject_here;}}


   } // end 1a;

   // ---------------------------------------------------------------------------------------------
   for (int pp =0; pp<totpar; pp++){
         xcout[pp] = xold[k][pp];
   }
   xcout[totpar] = energyNew;
   xcout[totpar+1] = accept_here;
   xcout[totpar+2] = reject_here;
   MPI::COMM_WORLD.Send(xcout, totpar+3 , MPI::DOUBLE, master, k);
   // ---------------------------------------------------------------------------------------------

  } // end else

} // end loop over swaps N



//////////////////////////
// EXIT MAIN ALGORITHM //
//////////////////////////
```

```
// ------------------------------------------------------------
//          MASTER PROCESS
// ------------------------------------------------------------
if (myid == 0){

  // Compute rates:
  for (int j=0; j<nchain; j++){
    rate[j] = (double) accepted[j]/(accepted[j]+rejected[j]);
    if (j<nchain-1)
          swapRate[j]=(double) swapAccepted[j]/(swapAccepted[j]+swapRejected[j]);
  }

  // Finish time:
  //final=clock()-init;
  //double tot_time= double(final/CLOCKS_PER_SEC);
  endtime = MPI::Wtime();
  //  std::cout << "total time = " << endtime-starttime << std::endl;
  double tot_time = endtime-starttime;

  ////////////////////////////
  // CREATE TERMINAL OUTPUT //
  ////////////////////////////

  cout <<"---------------------------------------"<<endl;
  for (int j=0; j<nchain; j++){
    cout<<"chain " << j+1 << " accepted " << accepted[j] << ", rejected "
```

```
          << rejected[j] << " Rate: " << rate[j] <<endl;}


for (int j=0; j<nchain-1; j++){
  cout << "chains " << j+1 << " and " << j+2 << " SwapAccepted "
            << swapAccepted[j] << " SwapRejected " << swapRejected[j]
            << " Rate: " << swapRate[j]<< endl;
}
cout << "Total time is " << tot_time << " seconds for "
          << N << " iterations." << endl;


//////////////////////
// WRITE DATA TO FILE //
//////////////////////


// final statistics
fpt=fopen("statistics.dat","w");
fprintf(fpt, "Run Statistics:\n");
for (int j=0; j<nchain; j++){
  fprintf(fpt,"Chain %d: accepted %d, rejected %d. Rate: %f\n",         \
              j+1,accepted[j],rejected[j],rate[j]);}
for (int j=0; j<nchain-1; j++){
  fprintf(fpt,"Chains %d and %d: Swap Accepted %d, Swap Rejected %d. Rate: %f\n", \
              j+1,j+2, swapAccepted[j], swapRejected[j], swapRate[j]);}
fprintf(fpt,"Total time is %d seconds for %d iterations", tot_time, N);
fclose(fpt);



/*
```

```
// low energy parameter file
fpt=fopen("parameters_low.dat","w");
fprintf(fpt, "Parameters:\n");
for(int a=0; a<totpar; a++){
fprintf(fpt, "#%d\t", a);}
fprintf(fpt,"\n");
for(int a=0; a<min(N,maxstor)/stor; a++){
for(int b=0; b<totpar; b++){
fprintf(fpt,"%.2e\t",parVecLo[a][b]);}
fprintf(fpt,"\n");}
fclose(fpt);

// high energy parameter file
fpt=fopen("parameters_high.dat","w");
fprintf(fpt, "Parameters:\n");
for(int a=0; a<totpar; a++){
fprintf(fpt, "#%d\t", a);}
fprintf(fpt,"\n");
for(int a=0; a<min(N,maxstor)/stor; a++){
for(int b=0; b<totpar; b++){
fprintf(fpt,"%.2e\t",parVecHi[a][b]);}
fprintf(fpt,"\n");}
fclose(fpt);

// energy file
fpt=fopen("energies.dat","w");
fprintf(fpt, "Energy for Chains:\n");
for(int a=0; a<nchain; a++){
```

```
    fprintf(fpt, "#%d\t", a+1);}
    fprintf(fpt,"\n");
    for (int a=0; a<min(N,maxstor)/stor; a++){
    for(int b=0; b<nchain; b++){
    fprintf(fpt, "%.2f\t", energyVec[a][b]);}
    fprintf(fpt,"\n");}
    fclose(fpt); */

  } // end if myid==0


  MPI::Finalize();

}

//////////////////
// END PROGRAM //
//////////////////
// ----------------------------------------------------------


//////////////////
// Subroutines //
//////////////////



////////////////////////
// New Point Function //
////////////////////////
```

```
void newpoint(double* vnew, double* vold, double eps, double lb[], double ub[])
{
  // From parameter set vold, we obtain parameter set vnew
  // The perturbation from vold to vnew occur randomly on a log scale with
  // variation eps.
  // The parameter values occur between lower bound lb and upper bound ub.

  for(int i=1; i<totpar; i++){
    vnew[i]=vold[i]*exp(2*(drand48()-.5)*eps*ub[i]/lb[i]);
    if(fabs(vnew[i])>ub[i])
      vnew[i]=lb[i]/ub[i]*fabs(vnew[i]);
    else if (fabs(vnew[i])<lb[i])
      vnew[i]=ub[i]/lb[i]*fabs(vnew[i]);
  }
  return;
}

// Functions for reading in data
int read_in(char s0[], double data[][vars_sublethal], double std[][vars_sublethal]) {
  double temp1[time_sublethal][vars_sublethal];
  double temp2[time_sublethal][vars_sublethal];
  FILE* f_in;
  char s1[32]="data_";
  strcat(s1,s0);
  strcat(s1,".dat");
  char s2[32]="std_";
  strcat(s2,s0);
```

```
strcat(s2,".dat");
if((f_in=fopen(s1,"r"))==NULL){
  cout << "\nERROR - Need file " << s1 << endl;
  return(0);}
else {
  for(int a=0; a<time_sublethal; a++){
    for(int b=0; b<vars_sublethal; b++){
          fscanf(f_in, "%lg", &temp1[a][b]);
    }
  }
  fclose(f_in);
}

if((f_in=fopen(s2,"r"))==NULL){
  cout << "\nERROR - Need file " << s2 << endl;
  return(0);}
else {
  for(int a=0; a<time_sublethal; a++){
    for(int b=0; b<vars_sublethal; b++){
          fscanf(f_in, "%lg", &temp2[a][b]);
    }
  }
  fclose(f_in);
}

for (int a = 0; a<time_sublethal; ++a) {
  for (int b = 0; b<vars_sublethal; ++b) {
    data[a][b] = log10(temp1[a][b]);
```

```
      std[a][b]=temp2[a][b]/temp1[a][b];
    }
  }
  return(1);
}

int read_in(char s0[], double data[][vars_lethal], double std[][vars_lethal]) {
  double temp1[time_lethal][vars_lethal];
  double temp2[time_lethal][vars_lethal];
  FILE* f_in;
  char s1[32]="data_";
  strcat(s1,s0);
  strcat(s1,".dat");
  char s2[32]="std_";
  strcat(s2,s0);
  strcat(s2,".dat");
  if((f_in=fopen(s1,"r"))==NULL){
    cout << "\nERROR - Need file " << s1 << endl;
    return(0);}
  else {
    for(int a=0; a<time_lethal; a++){
      for(int b=0; b<vars_lethal; b++){
            fscanf(f_in, "%lg", &temp1[a][b]);
      }
    }
    fclose(f_in);
  }
```

```
 if((f_in=fopen(s2,"r"))==NULL){
   cout << "\nERROR - Need file " << s2 << endl;
   return(0);}
 else {
   for(int a=0; a<time_lethal; a++){
     for(int b=0; b<vars_lethal; b++){
           fscanf(f_in, "%lg", &temp2[a][b]);
     }
   }
   fclose(f_in);
 }

 for (int a = 0; a<time_lethal; ++a) {
   for (int b = 0; b<vars_lethal; ++b) {
     data[a][b] = log10(temp1[a][b]);
     std[a][b]=temp2[a][b]/temp1[a][b];
   }
 }
 return(1);
}

// Smart log
double slog10(double arg)
{
 if (arg>1)
   return(log10(arg));
 else
   return(0);
```

```
}

// Heaviside Function
int heav(realtype arg)
{
 if (arg>0)
    return(1);
 else
    return(0);
}

// Geometric Mean
double geom_mean(double arg1, double arg2)
{
  double r=1.6;

  return(  exp( log( arg1 * exp(r*log(arg2)) )/(1+r) )  );

}




// Smooth Heaviside Function
double sheav(double arg1, double arg2)
{
  return((tanh(2*(arg1-arg2)/arg1)+1)/2);
}
```

```
// Hill function
double Hill(double X, double a, int h){
  return pow(X,h)/(pow(X,h)+pow(a,h));
}

////////////////////////////////////////
// OBJECTIVE FUNCTION FOR INFLUENZA //
////////////////////////////////////////

int obj_sublethal(double *obj, double par[], double data[][vars_sublethal], double std[][vars_sublethal])
{
  int i,j,k;
  realtype yout[NOUT][NEQ];
  *obj=0;

  par[0]=50;
  set_parameters(par);

  k=inf_traj(yout);
  if (k==1)
    {return(1);}

  // index of the data locations in memory
  int s[time_sublethal]={0,2,4,6,10,14,18,22};

  // Evaluate Objective Function

  for (i=0; i<time_sublethal; ++i) // time points 0,1,2,3,5,7,9,11
```

```
  {
    // data
    *obj+=pow((data[i][0]-slog10(yout[s[i]][1]))/std[i][0],2);     //N
    *obj+=pow((data[i][1]-slog10(yout[s[i]][2]))/std[i][1],2);     //M
    *obj+=pow((data[i][2]-slog10(yout[s[i]][3]))/std[i][2],2);     //L
    *obj+=pow((data[i][3]-slog10(yout[s[i]][4]))/std[i][3],2);     //T
    *obj+=pow((data[i][4]-slog10(yout[s[i]][6]))/std[i][4],2);     //A
    *obj+=pow((data[i][5]-slog10(yout[s[i]][7]))/std[i][5],2);     //C
    *obj+=pow((data[i][6]-slog10(yout[s[i]][10]))/std[i][6],2);    //F
    *obj+=pow((data[i][7]-slog10(max(VMIN,yout[s[i]][11])))/std[i][7],2); //V
    *obj+=pow((data[i][8]-slog10(yout[s[i]][13]))/std[i][8],2);    //E
    *obj+=pow((data[i][9]-slog10(yout[s[i]][14]))/std[i][9],2);    //B
    *obj+=pow((data[i][10]-slog10(yout[s[i]][15]))/std[i][10],2);    //K
    *obj+=pow((data[i][11]-slog10(yout[s[i]][16]))/std[i][11],2);  //P
    *obj+=pow((data[i][12]-slog10(yout[s[i]][17]))/std[i][12],2);  //W
    *obj+=pow((data[i][13]-slog10(yout[s[i]][18]))/std[i][13],2);  //G
    *obj+=pow((data[i][14]-slog10(yout[s[i]][19]))/std[i][14],2);  //O

    // Heuristic on Dead Healthy Cells
    *obj+=5*sheav(500*data[i][0], yout[s[i]][12]); // DH

  }

// FURTHER HEURISTICS
for (i=1; i<NOUT; i++) {

    // We assume survival, so the number of epithelial cells should not
    // penetrate below 20% of the original cell count
```

```
    *obj+=500*sheav(HMIN, yout[i][9]); //H

    // Viral load should decrease monotonically past day 5
    if(i>11) {
          *obj+=100*heav(yout[i][11] - yout[i-1][11]);} //V

 }


 return(0);
}

int obj_lethal(double *obj, double par[], double data[][vars_lethal], double std[][vars_lethal])
{
 int i,j,k;
 realtype yout[NOUT][NEQ];
 *obj=0;

 par[0]=500;
 set_parameters(par);

 k=inf_traj(yout);
 if (k==1)
   {return(1);}

 // index of the data locations in memory
 int s[time_lethal]={0,2,4,6,10,14};
```

```
// Evaluate Objective Function


for (i=0; i<time_lethal; ++i) // time points 0,1,2,3,5,7
  {
    // data
    *obj+=pow((data[i][0]-slog10(yout[s[i]][1]))/std[i][0],2);     //N
    *obj+=pow((data[i][1]-slog10(yout[s[i]][3]))/std[i][1],2);     //L
    *obj+=pow((data[i][2]-slog10(yout[s[i]][4]))/std[i][2],2);     //T
    *obj+=pow((data[i][3]-slog10(yout[s[i]][7]))/std[i][3],2);     //C
    *obj+=pow((data[i][4]-slog10(yout[s[i]][10]))/std[i][4],2);    //F
    *obj+=pow((data[i][5]-slog10(max(VMIN,yout[s[i]][11])))/std[i][5],2); //V
    *obj+=pow((data[i][6]-slog10(yout[s[i]][13]))/std[i][6],2);    //E
    *obj+=pow((data[i][7]-slog10(yout[s[i]][14]))/std[i][7],2);    //B
    *obj+=pow((data[i][8]-slog10(yout[s[i]][15]))/std[i][8],2);    //K
    *obj+=pow((data[i][9]-slog10(yout[s[i]][17]))/std[i][9],2);  //W
    *obj+=pow((data[i][10]-slog10(yout[s[i]][18]))/std[i][10],2);  //G
    *obj+=pow((data[i][11]-slog10(yout[s[i]][19]))/std[i][11],2);  //O


    // Heuristic on Antibodies:
    // We also assume on days 0,1,2, and 3 that antibodies are low
    // since they do not really mature until 7 days typically
    if (i<4) {
          *obj+=pow(2*(1-slog10(yout[s[i]][6])),2);  }//A


    // Heuristic on Dead Target Cells
    *obj+=5*sheav(500*data[i][0], yout[s[i]][12]); //DH
  }
```

```
  // FURTHER HEURISTICS
  for (i=1; i<NOUT; i++) {

    // Viral load should decrease monotonically past day 5
    if(i>11) {
      *obj+=100*heav(yout[i][11] - yout[i-1][11]); } //V

    // Heuristic that lethal trajectories for cells are monotone decreasing
    *obj+=100*heav(yout[i][9] - yout[i-1][9]); //H

    // We assume the number of epithelial cells should penetrate
    // below HMIN (20% of the original cell count) by day 5
    if (i>11) {
      *obj+=100*sheav(yout[i][9], HMIN); }  //H

  }
  return(0);
}

////////////////////////////////
//  Trajectory Calling Module //
////////////////////////////////

int inf_traj(double ytraj[][NEQ]){

  realtype reltol, t, tout;
  N_Vector y, abstol;
  void *cvode_mem;
```

```
int flag, flagr, iout;
int mxsteps, eqn;

y = abstol = NULL;
cvode_mem = NULL;
mxsteps = MAXSTEPS;

// Create serial vector of length NEQ for I.C. and abstol:

y = N_VNew_Serial(NEQ);
if (check_flag((void *)y, (char *)"N_VNew_Serial", 0)) return(1);
abstol = N_VNew_Serial(NEQ);
if (check_flag((void *)abstol, (char *)"N_VNew_Serial", 0)) return(1);

// Initial conditions:
realtype V0=v0, H0=tot_cell, M0=b_m, L0=b_lh*H0, P0=b_p;
realtype K0=b_k, F0=b_fp*b_p/mu_f;
realtype E0=b_ep/mu_e*Hill(b_p,a_ep,h_e);
realtype O0=b_op/mu_o*Hill(b_p,a_op,h_o);
realtype W0=b_p/mu_w*O0*b_wo/(O0+a_wo);
realtype G0=O0/mu_g*b_go*W0/(W0+a_go) + K0/mu_g*b_gk*W0/(W0+a_gk);
realtype B0=(b_b+b_bp*b_p*W0*b_0)/(b_bp*b_p*W0+mu_b);
realtype A0=(b_a + b_ab*B0)/mu_a;

// order is {NB,N,M,L,T,X,A,C,I,H,F,V,DH,E,B,K,P,W,G,O}
realtype init[NEQ]={0, 0, M0, L0, 0, 0, A0, 0, 0, H0, F0, V0, 0, E0, B0, K0, P0, W0, G0, O0};

// Initialize y:
```

```
for (eqn = 0; eqn < NEQ; ++eqn)
  {
    Ith(y,eqn+1) = ytraj[0][eqn] = init[eqn];
  }

/* Set the scalar relative tolerance */
reltol = RTOL;

/* Set the vector absolute tolerance */
for (eqn = 1; eqn <= NEQ; ++eqn)
  {
    Ith(abstol,eqn) = ATOL;
  }

/* Call CVodeCreate to create the solver memory and specify the
 * Backward Differentiation Formula and the use of a Newton iteration */
cvode_mem = CVodeCreate(CV_BDF, CV_NEWTON);
if (check_flag((void *)cvode_mem, (char *)"CVodeCreate", 0)) return(1);

/* Call CVodeInit to initialize the integrator memory and specify the
 * user's right hand side function in y'=f(t,y), the inital time T0, and
 * the initial dependent variable vector y. */
flag = CVodeInit(cvode_mem, f, T0, y);
if (check_flag(&flag, (char *)"CVodeInit", 1)) return(1);

/* Call CVodeSVtolerances to specify the scalar relative tolerance
 * and vector absolute tolerances */
flag = CVodeSVtolerances(cvode_mem, reltol, abstol);
```

```
if (check_flag(&flag, (char *)"CVodeSVtolerances", 1)) return(1);

/* Call CVLapackDense to specify the LAPACK dense linear solver */
flag = CVLapackBand(cvode_mem, NEQ, mupper, mlower);
if (check_flag(&flag, (char *)"CVLapackBand", 1)) return(1);

/* Set the Jacobian routine to Jac (user-supplied) */
flag = CVDlsSetBandJacFn(cvode_mem, Jac ); /* */
if (check_flag(&flag, (char *)"CVDlsSetBandJacFn", 1)) return(1);

/*Set the number of maximum steps*/
flag=CVodeSetMaxNumSteps(cvode_mem, mxsteps);
if (check_flag(&flag, (char *)"CVodeSetMaxNumSteps",1)) return(1);

/*Set the parameters for f_data*/
//  flag = CVodeSetUserData(cvode_mem, par);
//  if (check_flag(&flag, (char *)"CVodeSetUserData",1)) return(1);

iout = 1;  tout = T1;

while(1) {
  flag = CVode(cvode_mem, tout, y, &t, CV_NORMAL);

  /* write vector ytraj*/
  for (eqn = 0; eqn < NEQ; ++eqn)
    {
          ytraj[iout][eqn] = Ith(y, eqn+1);
    }
```

```
    //need to return flag to main program instead of break
    if (flag<0)
      { return(1); }

    if (flag == CV_SUCCESS) {
      iout++;
      tout += TADD;
    }

    if (iout == NOUT) break;

  }

  /* Free y vector */
  N_VDestroy_Serial(y);
  N_VDestroy_Serial(abstol);

  // Free integrator memory
  CVodeFree(&cvode_mem);

  return(0);
}

//////////////////////////
// Function Evaluation //
//////////////////////////
```

```
static int f(realtype t, N_Vector y, N_Vector ydot, void *user_data)
{
  // store variables used in subroutine:
  realtype M, T, L, C;
  realtype NB, N, X, H, DH;
  realtype V, I, F, K, G;
  realtype P, E, O, W, B, A;
  realtype Sigma1, Sigma2, DI, D, R;
  double *par=(double*)user_data;

  // define variables:
  M = Ith(y,3); T = Ith(y,5); L = Ith(y,4); C = Ith(y,8); NB = Ith(y,1);
  N = Ith(y,2); X = Ith(y,6); H = Ith(y,10); DH = Ith(y,13); I = Ith(y,9);
  V = Ith(y,12); F = Ith(y,11); K = Ith(y,16); G = Ith(y,19); P = Ith(y,17);
  E = Ith(y,14); O = Ith(y,20); W = Ith(y,18); B = Ith(y,15); A = Ith(y,7);

  D=tot_cell-H-I;
  Sigma1=a_11*T+a_12*D;
  Sigma2=Sigma1+(a_21*V)/(a_22+V);
  DI=tot_cell-H-I-DH;
  R=F/(a_rf+F);

  Ith(ydot,3) = b_mc*Hill(C,a_mc,h_m) - mu_m*(M-b_m);
  Ith(ydot,5) = b_t*M*Sigma2/(Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)) - mu_t*T;
  Ith(ydot,4) = b_l*M*Sigma1/(Sigma1+(g_1*L+g_2)/(L+d_2)) - mu_l*(L - b_lh*(1-R)*H);
  Ith(ydot,8) = b_c*M*Sigma1/(Sigma1+(g_1*L+g_2)/(L+d_2)) - mu_c*C;
  Ith(ydot,1) = b_nt*T/(a_nt + a_nl*L + T) - NB*C*g_nc/(C +a_nc) - mu_n*NB;
  Ith(ydot,2) = NB*C*g_nc/(C + a_nc) - mu_n*N;
```

```
  Ith(ydot,6) = b_xn*N/(N+a_xn) - g_xi*I*X - g_xh*H*X - mu_x*X;
  Ith(ydot,10) = b_h*(1-R)*H*D*(H-theta)/tot_cell - g_hv*V*H - g_hx*H*Hill(X,a_hx,h_x);
  Ith(ydot,13) = g_hx*H*Hill(X,a_hx,h_x) - b_h*(1-R)*H*DH*(H-theta)/tot_cell;
  Ith(ydot,9) = g_hv*V*H - g_ix*I*Hill(X,a_ix,h_x) - g_ik*R*I*K -g_ie*R*I*E - mu_i*(1-R)*I;
  Ith(ydot,12) = g_vi*(1-R)*I - g_vh*H*V - g_va*V*A - g_v*V/(1+a_v*V) - mu_v*V;
  Ith(ydot,11) = b_fi*(1-R)*I + b_fp*P - g_fi*I*F - mu_f*F;
  Ith(ydot,16) = b_kc*Hill(C,a_kc,h_k) - g_ki*R*I*K - mu_k*(K - b_k);
  Ith(ydot,19) =b_go*W/(a_go+ W)*O + b_gk*W/(a_gk+W)*K - mu_g*G;
  Ith(ydot,17) = p_0*(g_pv*V/(a_pv+V)+g_pi*DI)*(g_p + b_pg*G/(a_pg+G)) - mu_p*(P-b_p);
  Ith(ydot,14) = b_ep*Hill(P,a_ep,h_e) - g_ei*R*I*E - mu_e*E;
  Ith(ydot,20) = b_op*Hill(P,a_op,h_o) - mu_o*O;
  Ith(ydot,18) = b_wo*O/(a_wo+O)*P - mu_w*W;
  Ith(ydot,15) = b_b + b_bp*W*P*(b_0-B) - mu_b*B;
  Ith(ydot,7) = b_a + b_ab*B - g_av*A*V - mu_a*A;


  return(0);
}


static int Jac(int N, int mu, int ml,
                   realtype t, N_Vector y, N_Vector fy,
               DlsMat J, void *user_data,
               N_Vector tmp1, N_Vector tmp2, N_Vector tmp3)
{
  // store variables for the subroutine:
  realtype M, T, L, C;
  realtype NB, NT, X, H, DH;
  realtype V, I, F, K, G;
  realtype P, E, O, W, B, A;
```

```
realtype Sigma1, Sigma2, DI, D, R, dRdF;
double *par=(double*)user_data;

// define variables:
M = Ith(y,3); T = Ith(y,5); L = Ith(y,4); C = Ith(y,8); NB = Ith(y,1);
N = Ith(y,2); X = Ith(y,6); H = Ith(y,10); DH = Ith(y,13); I = Ith(y,9);
V = Ith(y,12); F = Ith(y,11); K = Ith(y,16); G = Ith(y,19); P = Ith(y,17);
E = Ith(y,14); O = Ith(y,20); W = Ith(y,18); B = Ith(y,15); A = Ith(y,7);

D=tot_cell-H-I;
Sigma1=a_11*T+a_12*D;
Sigma2=Sigma1+(a_21*V)/(a_22+V);
DI=tot_cell-H-I-DH;
R=F/(a_rf+F);
dRdF=a_rf/pow(a_rf + F,2);

// set column vector pointers:
realtype *Col_1, *Col_2, *Col_3, *Col_4, *Col_5, *Col_6;
realtype *Col_7, *Col_8, *Col_9, *Col_10, *Col_11, *Col_12, *Col_13;
realtype *Col_14, *Col_15, *Col_16, *Col_17, *Col_18, *Col_19, *Col_20;

Col_1 = BAND_COL(J,0); Col_2 = BAND_COL(J,1);
Col_3 = BAND_COL(J,2); Col_4 = BAND_COL(J,3);
Col_5 = BAND_COL(J,4); Col_6 = BAND_COL(J,5);
Col_7 = BAND_COL(J,6); Col_8 = BAND_COL(J,7);
Col_9 = BAND_COL(J,8); Col_10 = BAND_COL(J,9);
Col_11 = BAND_COL(J,10); Col_12 = BAND_COL(J,11);
Col_13 = BAND_COL(J,12); Col_14 = BAND_COL(J,13);
```

```
  Col_15 = BAND_COL(J,14); Col_16 = BAND_COL(J,15);

  Col_17 = BAND_COL(J,16); Col_18 = BAND_COL(J,17);

  Col_19 = BAND_COL(J,18); Col_20 = BAND_COL(J,19);


  // define jacobian matrix:


  // Row for M:

  IJth(Col_3,3,3) = -mu_m;

  IJth(Col_8,3,8) = b_mc*pow(a_mc,h_m)*pow(C,h_m-1)/pow( pow(a_mc,h_m) + pow(C,h_m) ,2);


  // Row for T:

  IJth(Col_3,5,3) = b_t*Sigma2/(Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1));

  IJth(Col_5,5,5) = a_11*b_t*M/(Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)) -
b_t*M*Sigma2/pow((Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)),2)*(a_11*(L*(k_1+1)+k_2
+d_1)/(L+d_1)) - mu_t;


  IJth(Col_4,5,4) = -
b_t*M*Sigma2/pow((Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)),2)*((Sigma2+(g_1*L+g_2)
/(L+d_2))*((d_1*k_1-k_2)/pow(L+d_1,2))+((k_1*L+k_2)/(L+d_1))*((d_2*g_1-g_2)/pow(L+d_2,2)));


  IJth(Col_10,5,10) =  -a_12*b_t*M/(Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)) +
b_t*M*Sigma2/pow((Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)),2)*a_12*(L*(k_1+1)+k_2
+d_1)/(L+d_1);


  IJth(Col_9,5,9) = IJth(Col_10,5,10);


  IJth(Col_12,5,12) =
a_21*a_22/pow(a_22+V,2)*b_t*M*(1/(Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)) -
Sigma2/pow((Sigma2+(Sigma2+(g_1*L+g_2)/(L+d_2))*(k_1*L+k_2)/(L+d_1)),2)*(L*(k_1+1)+k_2+d_1)/(L+d_1)
);


  // Row for L:
```

```
  IJth(Col_3,4,3) = Sigma1*b_l/(Sigma1+(g_1*L+g_2)/(L+d_2));

  IJth(Col_5,4,5) = a_11*b_l*M/(Sigma1+(g_1*L+g_2)/(L+d_2)) -
a_11*Sigma1*b_l*M/pow(Sigma1+(g_1*L+g_2)/(L+d_2),2);

  IJth(Col_4,4,4) = -Sigma1*b_l*M/pow(Sigma1+(g_1*L+g_2)/(L+d_2),2)*(g_1*d_2-g_2)/pow(L+d_2,2)-mu_l;

  IJth(Col_10,4,10) = -a_12*b_l*M/(Sigma1+(g_1*L+g_2)/(L+d_2)) +
a_12*Sigma1*b_l*M/pow(Sigma1+(g_1*L+g_2)/(L+d_2),2)+b_lh*mu_l*(1-R);

  IJth(Col_9,4,9) = -a_12*b_l*M/(Sigma1+(g_1*L+g_2)/(L+d_2)) +
a_12*Sigma1*b_l*M/pow(Sigma1+(g_1*L+g_2)/(L+d_2),2);

  IJth(Col_11,4,11) = -b_lh*mu_l*H*dRdF;



  // Row for C:

  IJth(Col_3,8,3) = b_c*Sigma1/(Sigma1+(g_1*L+g_2)/(L+d_2));

  IJth(Col_5,8,5) = a_11*b_c*M/(Sigma1+(g_1*L+g_2)/(L+d_2)) -
a_11*Sigma1*b_c*M/pow(Sigma1+(g_1*L+g_2)/(L+d_2),2);

  IJth(Col_4,8,4) = -Sigma1*M*b_c/pow(Sigma1+(g_1*L+g_2)/(L+d_2),2)*(g_1*d_2-g_2)/pow(L+d_2,2);

  IJth(Col_8,8,8) = -mu_c;

  IJth(Col_10,8,10) =  -a_12*b_c*M/(Sigma1+(g_1*L+g_2)/(L+d_2)) +
a_12*Sigma1*b_c*M/pow(Sigma1+(g_1*L+g_2)/(L+d_2),2);

  IJth(Col_9,8,9) = IJth(Col_10,8,10);



  // Row for NB:

  IJth(Col_5,1,5) = b_nt*(a_nt+a_nl*L)/pow(a_nt+ a_nl*L + T,2);

  IJth(Col_4,1,4) = b_nt*T*a_nl/pow(a_nt+a_nl*L+T,2);

  IJth(Col_8,1,8) = -NB*g_nc*a_nc/pow(a_nc+C,2);

  IJth(Col_1,1,1) = -mu_n - g_nc*C/(a_nc+C);



  // Row for N:

  IJth(Col_8,2,8) = -IJth(Col_8,1,8);

  IJth(Col_1,2,1) = g_nc*C/(C+g_nc);

  IJth(Col_2,2,2) = -mu_n;
```

```
// Row for X:

IJth(Col_1,6,1) = b_xn*a_xn/pow(a_xn+NT,2);

IJth(Col_6,6,6) = -mu_x -g_xi*I - g_xh*H;

IJth(Col_10,6,10) = - g_xh*X;

IJth(Col_9,6,9) = - g_xi*X;


// Row for H:

IJth(Col_6,10,6) = -g_hx*H*pow(a_hx,h_x)*h_x*pow(X,h_x-
1)/((pow(a_hx,h_x)+pow(X,h_x))*(pow(a_hx,h_x)+pow(X,h_x)));

IJth(Col_10,10,10) = b_h*(1-R)*((tot_cell-H-I)*(2*H-theta)-H*(H-theta))/tot_cell - g_hv*V -g_hx*pow(X,
h_x)/(pow(a_hx,h_x)+pow(X,h_x));

IJth(Col_9,10,9) = -b_h*(1-R)*H*(H-theta)/tot_cell;

IJth(Col_12,10,12) = -g_hv*H;

IJth(Col_11,10,11) = -dRdF*b_h*H*D*(H-theta)/tot_cell;


// Row for DH:

IJth(Col_6,13,6) = -IJth(Col_6,10,6);

IJth(Col_10,13,10) = h_x*pow(X,h_x)/(pow(a_hx,h_x)+pow(X,h_x)) - b_h*(1-R)*DH*(2*H-theta)/tot_cell;

IJth(Col_13,13,13) = -b_h*(1-R)*H*(H-theta)/tot_cell;

IJth(Col_11,13,11) = -dRdF*b_h*H*DH*(H-theta)/tot_cell;


// Row for I:

IJth(Col_6,9,6) = -g_ix*I*h_x*pow(X,h_x-1)/((pow(a_ix,h_x)+pow(X,h_x))*(pow(a_ix,h_x)+pow(X,h_x)));

IJth(Col_10,9,10) = g_hv*V;

IJth(Col_9,9,9) = -mu_i*(1-R) - g_ix*pow(X,h_x)/(pow(a_ix,h_x)+pow(X,h_x))-g_ik*R*K - g_ie*R*E;

IJth(Col_12,9,12) = g_hv*H;

IJth(Col_11,9,11) = -(mu_i*I+g_ik*I*K + g_ie*I*E)*dRdF;

IJth(Col_16,9,16) = -g_ik*R*I;
```

```
IJth(Col_14,9,14) = -g_ie*R*I;


// Row for V:

IJth(Col_10,12,10) = -g_vh*V;

IJth(Col_9,12,9) = g_vi*(1-R);

IJth(Col_12,12,12) = -g_vh*H - mu_v - g_va*A- g_v/pow(1+a_v*V,2);

IJth(Col_11,12,11) = -g_vi*I*dRdF;

IJth(Col_7,12,7) = -g_va*V;


// Row for F:

IJth(Col_9,11,9) = b_fi*(1-R) - g_fi*F;

IJth(Col_11,11,11) = -b_fi*dRdF - g_fi*I - mu_f;

IJth(Col_17,11,17) = b_fp;


// Row for K:

IJth(Col_8,16,8) = pow(a_kc, h_k)*h_k*b_kc*pow(C, h_k-
1)/((pow(a_kc,h_k)+pow(C,h_k))*(pow(a_kc,h_k)+pow(C,h_k)));

IJth(Col_9,16,9) = -g_ki*R*K;

IJth(Col_11,16,11) = -g_ki*dRdF*I*K;

IJth(Col_16,16,16) = -mu_k - g_ki*R*I;


// Row for G:

IJth(Col_16,19,16) = b_gk*W/(W + a_gk);

IJth(Col_19,19,19) = -mu_g;

IJth(Col_20,19,20) = b_go*W/(W + a_go);

IJth(Col_18,19,18) = b_go*a_go/pow(W + a_go,2)*O+b_gk*a_gk/pow(W + a_gk,2)*K;


// Row for P:
```

```
  IJth(Col_10,17,10) = -g_pi*p_0*(b_pg*G/(G + a_pg) + g_p);

  IJth(Col_13,17,13) = IJth(Col_10,17,10);

  IJth(Col_9,17,9) = IJth(Col_10,17,10);

  IJth(Col_12,17,12) = -g_pv*a_pv/pow(V + a_pv,2)*p_0*(b_pg*G/(G + a_pg) + g_p);

  IJth(Col_19,17,19) = -b_pg*a_pg/pow(G + a_pg,2)*p_0*(g_pv*V/(V + a_pv) + g_pi*(tot_cell-H-I-DH));

  IJth(Col_17,17,17) = -mu_p;


  // Row for E:

  IJth(Col_9,14,9) = -g_ei*R*E ;

  IJth(Col_11,14,11) = -g_ei*I*E*dRdF;

  IJth(Col_17,14,17) = b_ep*h_e*a_ep*pow(P,h_e-
1)/((pow(a_ep,h_e)+pow(P,h_e))*(pow(a_ep,h_e)+pow(P,h_e)));

  IJth(Col_14,14,14) = -g_ei*R*I - mu_e;


  // Row for O:

  IJth(Col_17,20,17) = b_op*h_o*a_op*pow(P,h_o-
1)/((pow(a_op,h_o)+pow(P,h_o))*(pow(a_op,h_o)+pow(P,h_o)));

  IJth(Col_20,20,20) = -mu_o;


  // Row for W:

  IJth(Col_17,18,17) = b_wo*O/(O+a_wo);

  IJth(Col_20,18,20) = b_wo*a_wo/pow(O + a_wo,2);

  IJth(Col_18,18,18) = -mu_w;


  // Row for B:

  IJth(Col_17,15,17) = b_bp*W*(b_0-B);

  IJth(Col_18,15,18) = b_bp*P*(b_0-B);

  IJth(Col_15,15,15) = -b_bp*W*P - mu_b;
```

```
  // Row for A:
  IJth(Col_12,7,12) = -g_av*A;
  IJth(Col_15,7,15) = b_ab;
  IJth(Col_7,7,7) = -g_av*V - mu_a;

  return(0);
}

void set_parameters(double* par)
{
  v0=par[0], a_11=par[1], a_12=par[2], a_21=par[3], a_22=par[4];
  g_1=par[5], g_2=par[6], d_2=par[7];
  b_mc=par[8], a_mc=par[9], mu_m=par[10], b_m=par[11];
  b_t=par[12], k_1=par[13], k_2=par[14], d_1=par[15], mu_t=par[16];
  b_l=par[17], mu_l=par[18], b_lh=par[19];
  b_c=par[20], mu_c=par[21];
  b_nt=par[22], a_nt=par[23], a_nl=par[24], g_nc=par[25], a_nc=par[26], mu_nb=par[27];
  mu_n=par[28];
  b_xn=par[29], a_xn=par[30], g_xi=par[31], g_xh=par[32], mu_x=par[33];
  b_h=par[34], theta=par[35], g_hv=par[36], g_hx=par[37], a_hx=par[38];
  g_ix=par[39], a_ix=par[40], g_ik=par[41], g_ie=par[42], mu_i=par[43];
  g_vi=par[44], g_vh=par[45], g_va=par[46], g_v=par[47], a_v=par[48], mu_v=par[49];
  b_fi=par[50], b_fp=par[51], g_fi=par[52], mu_f=par[53], a_rf=par[54];
  b_k=par[55], b_kc=par[56], a_kc=par[57], g_ki=par[58], mu_k=par[59];
  b_go=par[60], g_go=par[61], a_go=par[62], b_gk=par[63], g_gk=par[64];
  a_gk=par[65], mu_g=par[66];
  p_0=par[67], g_pv=par[68], a_pv=par[69], g_pi=par[70], g_p=par[71];
  b_pg=par[72], a_pg=par[73], mu_p=par[74], b_p=par[75];
```

```
  b_ep=par[76], a_ep=par[77], g_ei=par[78], mu_e=par[79];
  b_op=par[80], a_op=par[81], mu_o=par[82];
  b_w=par[83], b_wo=par[84], a_wo=par[85], mu_w=par[86];
  b_b=par[87], b_bp=par[88], b_0=par[89], mu_b=par[90];
  b_a=par[91], b_ab=par[92], g_av=par[93], mu_a=par[94];
  return;
}

static int check_flag(void *flagvalue, char *funcname, int opt)
{
  int *errflag;

  // Check if SUNDIALS function returned NULL pointer - no memory allocated:
  if (opt == 0 && flagvalue == NULL) {
    fprintf(stderr, "\nSUNDIALS_ERROR: %s() failed - returned NULL pointer\n\n",
                funcname);
    return(1); }

  /* Check if flag < 0 */
  else if (opt == 1) {
    errflag = (int *) flagvalue;
    if (*errflag < 0) {
      fprintf(stderr, "\nSUNDIALS_ERROR: %s() failed with flag = %d\n\n",
                 funcname, *errflag);
      return(1); }}

  /* Check if function returned NULL pointer - no memory allocated */
  else if (opt == 2 && flagvalue == NULL) {
```

```
    fprintf(stderr, "\nMEMORY_ERROR: %s() failed - returned NULL pointer\n\n",
                funcname);
    return(1); }


  return(0);
}
```

Appendix section's first paragraph.

Second paragraph.

Appendix subsection

This is a subsection (level-3 division) of appendix A.

BIBLIOGRAPHY

Reference List

Abbas, Abul K. and Andrew H. Lichtman. Cellular and Molecular Immunology. 5th ed., updated ed. 564 p vols. Philadelphia, PA: Saunders, 2005.

Abramson, Mark A, et al. "ORTHOMADS: A deterministic MADS instance with orthogonal directions." SIAM Journal on Optimization 20.2 (2008): 948-66.

Ahlberg, J. H., Edwin N. Nilson, and J. L. Walsh. The Theory of Splines and Their Applications. xi, 284 p vols. New York: Academic Press, 1967.

Ascher, U. M. and Linda Ruth Petzold. Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. xvii, 314 p vols. Philadelphia: Society for Industrial and Applied Mathematics, 1998.

Audet, Charles and Jr J.E.Dennis. "Analysis of Generalized Pattern Searches." SIAM Journal on Optimization 13.3 (2002): 889-903.

---. "Mesh Adaptive Direct Search Algorithms for Constrained Optimization." SIAM Journal on Optimization 17.1 (2004): 188-217.

Baccam, P., et al. "Kinetics of influenza A virus infection in humans." J.Virol. 80.15 (2006): 7590-99.

Beauchemin, C. "Probing the effects of the well-mixed assumption on viral infection dynamics." J.Theor.Biol. 242.2 (2006): 464-77.

Bocharov, G. A. and A. A. Romanyukha. "Mathematical model of antiviral immune response. III. Influenza A virus infection." J.Theor.Biol. 167.4 (1994): 323-60.

Carayannopoulos, L. N., et al. "Murine trophoblast cells induce NK cell interferon-gamma production through KLRK1." Biol.Reprod. 83.3 (2010): 404-14.

Carayannopoulos, L. N. and W. M. Yokoyama. "Recognition of infected cells by natural killer cells." Curr.Opin.Immunol. 16.1 (2004): 26-33.

Chesler, D. A. and C. S. Reiss. "The role of IFN-gamma in immune responses to viral infections of the central nervous system." Cytokine Growth Factor Rev. 13.6 (2002): 441-54.

Chow, C. C., et al. "The acute inflammatory response in diverse shock states." Shock 24.1 (2005): 74-84.

Clark, I. A. "How TNF was recognized as a key mechanism of disease." Cytokine Growth Factor Rev. 18.3-4 (2007): 335-43.

Clermont, G., et al. "In silico design of clinical trials: a method coming of age." Crit Care Med. 32.10 (2004): 2061-70.

Covert, M. W., et al. "Achieving stability of lipopolysaccharide-induced NF-kappaB activation." Science 309.5742 (2005): 1854-57.

Daun, S., et al. "An ensemble of models of the acute inflammatory response to bacterial lipopolysaccharide in rats: results from parameter space reduction." J.Theor.Biol. 253.4 (2008): 843-53.

Day, J., A. Friedman, and L. S. Schlesinger. "Modeling the immune rheostat of macrophages in the lung in response to infection." Proc.Natl.Acad.Sci.U.S.A 106.27 (2009): 11246-51.

Day, J., et al. "A reduced mathematical model of the acute inflammatory response II. Capturing scenarios of repeated endotoxin administration." J.Theor.Biol. 242.1 (2006): 237-56.

---. "A reduced mathematical model of the acute inflammatory response II. Capturing scenarios of repeated endotoxin administration." J.Theor.Biol. 242.1 (2006): 237-56.

Doedel, E. J. "AUTO: A Program for the Automatic Bifurcation Analysis of Autonomous Systems". Tenth Manitoba Conference on Numerical Mathematics and Computing, September 1, 1980 Winnipeg, Canada: 1981.V 265-84.

Edelstein-Keshet, Leah. Mathematical Models in Biology. xliii, 586 p vols. Philadelphia: Society for Industrial and Applied Mathematics, 2005.

---. Mathematical Models in Biology. xliii, 586 p vols. Philadelphia: Society for Industrial and Applied Mathematics, 2005.

Eigenbrod, T., et al. "Cutting edge: critical role for mesothelial cells in necrosis-induced inflammation through the recognition of IL-1 alpha released from dying cells." J.Immunol. 181.12 (2008): 8194-98.

Ermentrout, Bard. Simulating, Analyzing, and Animating Dynamical Systems

a Guide to XPPAUT for Researchers and Students. xiv, 290 p vols. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

Fall, Christopher P. Computational Cell Biology. xx, 468 p vols. New York: Springer, 2002.

Fernandez, E. J. and E. Lolis. "Structure, function, and inhibition of chemokines." Annu.Rev.Pharmacol.Toxicol. 42 (2002): 469-99.

French, A. R. and W. M. Yokoyama. "Natural killer cells and viral infections." Curr.Opin.Immunol. 15.1 (2003): 45-51.

Goodman, R. B., et al. "Cytokine-mediated inflammation in acute lung injury." Cytokine Growth Factor Rev. 14.6 (2003): 523-35.

Hall, Charles A. and Thomas A. Porsching. Numerical Analysis of Partial Differential Equations. xv, 319 p vols. Englewood Cliffs, N.J: Prentice Hall, 1990.

Hancioglu, B., D. Swigon, and G. Clermont. "A dynamical model of human immune response to influenza A virus infection." J.Theor.Biol. 246.1 (2007): 70-86.

---. "A dynamical model of human immune response to influenza A virus infection." J.Theor.Biol. 246.1 (2007): 70-86.

Hindmarsh, Alan, et al. "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers." ACM Transactions on Mathematical Software 31.3 (2005): 363-96.

Hitomi, J., et al. "Identification of a molecular signaling network that regulates a cellular necrotic cell death pathway." Cell 135.7 (2008): 1311-23.

Janeway, Charles. Immunobiology
the Immune System in Health and Disease. 6th ed ed. xxiii, 823 p vols. New York: Garland Science, 2005.

Johnson, Richard Arnold, Irwin Miller, and John E. Freund. Miller & Freund's Probability and Statistics for Engineers. 8th ed ed. viii, 544 p vols. Boston: Prentice Hall, 2011.

Kash, J. C., et al. "Global host immune response: pathogenesis and transcriptional profiling of type A influenza viruses expressing the hemagglutinin and neuraminidase genes from the 1918 pandemic virus." J.Virol. 78.17 (2004): 9499-511.

Keener, James P. and James Sneyd. Mathematical Physiology. viii, 766 p vols. New York: Springer, 1998.

---. Mathematical Physiology. viii, 766 p vols. New York: Springer, 1998.

Kobasa, D., et al. "Aberrant innate immune response in lethal infection of macaques with the 1918 influenza virus." Nature 445.7125 (2007): 319-23.

Kontoyiannis, D., et al. "Interleukin-10 targets p38 MAPK to modulate ARE-dependent TNF mRNA translation and limit intestinal pathology." EMBO J. 20.14 (2001): 3760-70.

Kumar, R., et al. "The dynamics of acute inflammation." J.Theor.Biol. 230.2 (2004): 145-55.

Kuznetsov, Yuri. Elements of Applied Bifurcation Theory. 2nd ed ed. xix, 591 p vols. New York: Springer, 1998.

Lehninger, Albert L., David L. Nelson, and Michael M. Cox. Lehninger Principles of Biochemistry. 5th ed ed. 1 v. (various pagings) vols. New York: W.H. Freeman, 2008.

Meisel, C., et al. "Differential regulation of monocytic tumor necrosis factor-alpha and interleukin-10 expression." Eur.J.Immunol. 26.7 (1996): 1580-86.

Miao, H., et al. "Quantifying the early immune response and adaptive immune response kinetics in mice infected with influenza A virus." J.Virol. 84.13 (2010): 6687-98.

Michaelis, Leonor. Einführung in Die Mathematik Für Biologen Und Chemiker. 2. erweiterte und verb. aufl. mit 117 textabbildungen ed. vi, 318 p vols. Berlin: J. Springer, 1922.

Mocellin, S., et al. "The dual role of IL-10." Trends Immunol. 24.1 (2003): 36-43.

Moore, K. W., et al. "Interleukin-10 and the interleukin-10 receptor." Annu.Rev.Immunol. 19 (2001): 683-765.

Moore, K. W., et al. "Interleukin-10." Annu.Rev.Immunol. 11 (1993): 165-90.

Murdoch, C. and A. Finn. "Chemokine receptors and their role in inflammation and infectious diseases." Blood 95.10 (2000): 3032-43.

Murray, J. D. Mathematical Biology. 3rd ed ed. 2 v vols. New York: Springer, 2002.

Nowak, M. A. and Robert M. May. Virus Dynamics

Mathematical Principles of Immunology and Virology. xii, 237 p vols. Oxford: Oxford University Press, 2000.

Perko, Lawrence. Differential Equations and Dynamical Systems. 3rd ed ed. xiv, 553 p vols. New York: Springer, 2001.

Pestka, S., C. D. Krause, and M. R. Walter. "Interferons, interferon-like cytokines, and their receptors." Immunol.Rev. 202 (2004): 8-32.

Prince, J. M., et al. "Toll-like receptor-4 signaling mediates hepatic injury and systemic inflammation in hemorrhagic shock." J.Am.Coll.Surg. 202.3 (2006): 407-17.

Reynolds, A., et al. "A reduced mathematical model of the acute inflammatory response: I. Derivation of model and analysis of anti-inflammation." J.Theor.Biol. 242.1 (2006): 220-36.

Saenz, R. A., et al. "Dynamics of influenza virus infection and pathology." J.Virol. 84.8 (2010): 3974-83.

Schroder, K., et al. "Interferon-gamma: an overview of signals, mechanisms and functions." J.Leukoc.Biol. 75.2 (2004): 163-89.

Smith, A. M. and A. S. Perelson. "Influenza A virus infection kinetics: quantitative data and models." Wiley.Interdiscip.Rev.Syst.Biol.Med. (2010).

Smith, P. L., G. Lombardi, and G. R. Foster. "Type I interferons and the innate immune response--more than just antiviral cytokines." Mol.Immunol. 42.8 (2005): 869-77.

Sprung, C. L., et al. "The effects of high-dose corticosteroids in patients with septic shock. A prospective, controlled study." N.Engl.J.Med. 311.18 (1984): 1137-43.

Stephens, P. A. and W. J. Sutherland. "Consequences of the Allee effect for behaviour, ecology and conservation." Trends Ecol.Evol. 14.10 (1999): 401-05.

Sun, L., et al. "Effect of IL-10 on neutrophil recruitment and survival after Pseudomonas aeruginosa challenge." Am.J.Respir.Cell Mol.Biol. 41.1 (2009): 76-84.

Takala, A., et al. "Markers of systemic inflammation predicting organ failure in community-acquired septic shock." Clin.Sci.(Lond) 97.5 (1999): 529-38.

Toapanta, F. R. and T. M. Ross. "Impaired immune responses in the lungs of aged mice following influenza infection." Respir.Res. 10 (2009): 112.

Wahba, Grace. Spline Models for Observational Data. xii, 169 p vols. Philadelphia, Pa: Society for Industrial and Applied Mathematics, 1990.

Walzer, T., et al. "Natural killer cell trafficking in vivo requires a dedicated sphingosine 1-phosphate receptor." Nat.Immunol. 8.12 (2007): 1337-44.

Yokoyama, W. M. "Betting on NKT and NK cells." Immunity. 20.4 (2004): 363-65.

Yokoyama, W. M., S. Kim, and A. R. French. "The dynamic life of natural killer cells." Annu.Rev.Immunol. 22 (2004): 405-29.

Zhou, J., et al. "Functional tumor necrosis factor-related apoptosis-inducing ligand production by avian influenza virus-infected macrophages." J.Infect.Dis. 193.7 (2006): 945-53.