ATSC 409 Project

COMPARING TWO MODELS OF HIV POPULATION DYNAMICS

Ronald Monillas

42469114

I. INTRODUCTION

The human immunodeficiency virus or HIV is the world's leading infectious killer. Since the first cases were reported in 1981, an estimated 39 million people have died from HIV infection. HIV is transmitted from infected to susceptible individuals by the exchange of bodily fluids. Once inside the body, HIV particles infect white blood cells by attaching to the CD4 protein embedded in the cell membranes of helper T cells, macrophages, and dendritic cells. The genome of the virus, which is made up of RNA, then enters these cells and is reverse transcribed into DNA, which is subsequently incorporated into the genome of the host. The virus may then remain latent within the genome of the host cell or become active, in which case it is transcribed to produce both the proteins necessary to replicate and daughter RNA particles. When actively replicating, HIV can produce hundreds of daughter viruses per day per host cell, often killing the host cell in the process. These virus particles then bud off and go on to infect other CD4-bearing cells, repeating the process. Eventually, without treatment, the population of CD4+ helper T cells declines dramatically, signaling the onset of AIDS. Normally CD4 lymphocytes are activated by a specific binding mechanism to release cytokines that stimulate both attack and antibody-producing cells. By destroying activated CD4 lymphocytes, HIV can eliminate the very cells that recognize and fight other infections.

After an individual is infected by HIV, the number of virions within the bloodstream skyrockets and then plummets again in a period known as the acute phase. The rapid rise in virus particles reflects the infection of CD4 cells and the replication of HIV within actively infected host cells whereas the decline in virus particles is thought to have been caused by the suppression of the viral infection by the immune system. Phillips (1996), however, suggested that the number of virions might decline because most of the susceptible CD4 cells have already been infected and thus there are fewer host cells to infect. Phillips developed a model to assess the dynamics of the CD4 cells and the virus under the assumption that the body does not get better at eliminating infected cells or virus particles over time. He thus came to the conclusion that "the reduction in virus concentration during acute infection may not reflect the ability of the HIV-specific immune response to control the virus replication." Phillip's model proved that an HIV-specific immune response is not necessary to explain the reduction in virus concentration.

Phillips' model is adapted from the model presented by McLean et al (1991), which actually was not used to study acute HIV infection. McLean's group reviewed the population dynamics of HIV within an individual after treatment with zidovudine, a drug that prevents reverse transcription or the formation of viral DNA. Two mechanisms – the development of drug resistance and the existence of a viral sanctuary – have been proposed to trigger a second rise in the levels of plasma HIV after a significant short-term reduction in disease progression and in the amount of HIV in plasma observed post-zidovudine therapy. However, McLean's group proposed a third possible mechanism – that the observed patterns of viral abundance simply reflect the natural dynamics of a parasite (HIV) after therapeutic protection of its host (CD4 cells). Both Phillips and McLean's group acknowledge that the changes they have observed are simply a result of population dynamics.

In this project we attempt to (1) emulate Andrew Phillips' model in his 1996 study and (2) compare it to that of McLean's group in their 1991 study. The two investigations both examine the population dynamics of HIV infection within an individual but are motivated by different phenomena. McLean's group aims to predict HIV patterns within an individual following treatment with a specific medication while Phillips' interest lies in validating the causal relation between an HIV-specific immune response and a reduction in virus concentration during acute infection. The two mathematical models defined by Phillips and by McLean's group are very similar in that the former is adapted from the latter. Although the main focus of the project is to reproduce Phillips' research and findings, it would be interesting to see the similarities and differences in long-term behavior of the cell populations between Phillips' model and McLean's group's model. Disclaimer: My background is in mathematics, statistics, and computer science. The reason why I chose to model a biological process for my project is because I wanted to do something that neither I nor the graders are too familiar with.

II. THE MODEL

We illustrate Phillips' (1996) and McLean's group's (1991) models in the form a flow diagram. Activated uninfected CD4 lymphocytes (R) arise at some rate ω and are withdrawn either by HIV-independent death at rate μ or by infection at rate βV. In the earlier model by McLean et al, the rate ω is described to be dependent on the amount of free virus V (i.e. $\omega = \frac{\Gamma}{1+\chi V}$) whereas in the later model by Phillips, the rate ω is defined as a purely constant rate (i.e. $\omega = \Gamma\tau$). In both models, the parameter Γ is given as the rate at which new uninfected CD4 cells emerge. McLean's group reasons that uninfected CD4 cells become activated at a rate of $\frac{1}{1+\chi V}$ that decreases with a growing burden of free virus where χ is the rate at which free virus destroy emerging CD4 cells. In other words, McLean's group presumes that out of all the Γ new CD4 lymphocytes generated every day, only a tiny fraction $\frac{1}{1+\chi V}R$ of them live on to become activated while the rest $\frac{\chi V}{1+\chi V}R$ are destroyed by free virus. On the other hand, Phillips thinks that the influx rate of activated uninfected CD4 lymphocytes depends on a constant activation rate τ. Interestingly, Phillips mimics McLean's group's entire model except for the parameter ω. Upon infection or encounter with a virus, CD4 cells either develop a latent infection with chance p or immediately enter a phase of active virus production with chance 1 – p. The latently infected cells (L) are then removed either by HIV-independent cell death at rate μ (same as uninfected cells) or by activation at rate α. Actively infected cells (E) that produce virus are generated immediately after infection (with chance 1 – p) or from the activation of latently infected cells at rate α before they die at rate δ. Free virions (V) are produced at rate π by actively infected cells and are cleared at rate σ. Note the dashed arrow from E(t) to V(t) – this represents the notion that the viral production by budding of actively infected cells does not directly eliminate the infected cells. Here are the flow diagram and the table of variables and parameters.

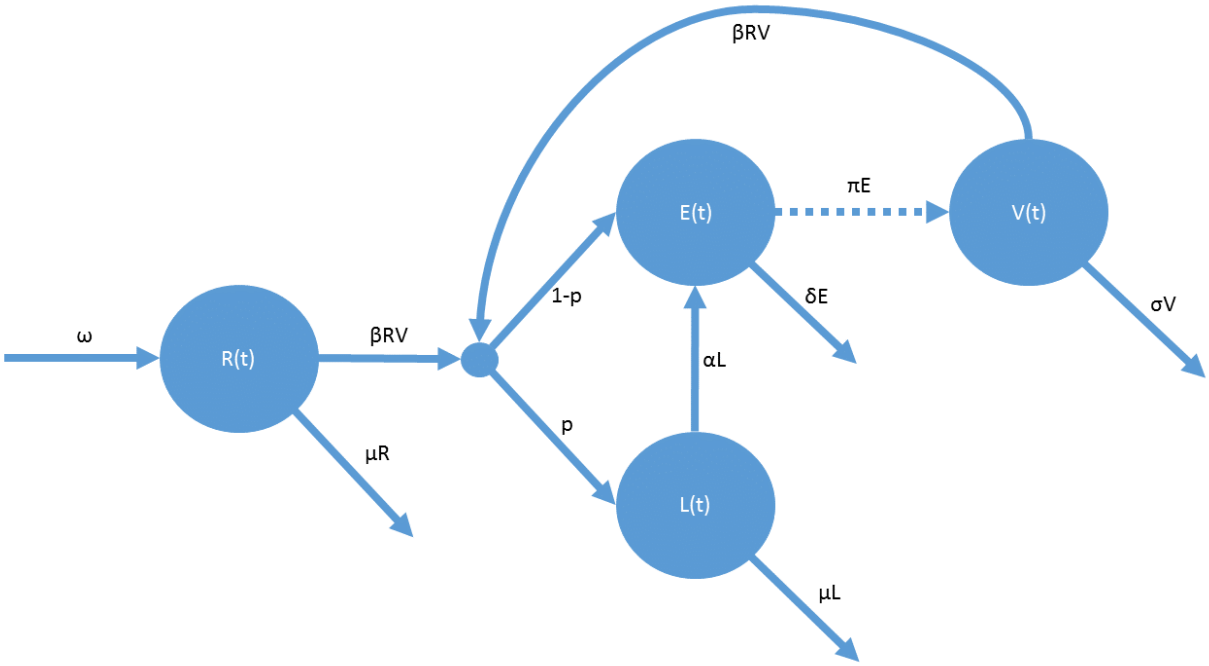

Figure 1: Flow diagram for a simple HIV model based on Phillips (1996) and McLean et al. (1991). See **Table 1** for the description of the variables and the parameters. The circles represent the number of entities experiencing the different stages of the infection (except for V, which is not a stage of the infection but is a representation of the population of the pathogen). The arrows connecting these circles represent the rate per day at which one compartment leads to another, where the total flow rate is written beside each arrow. When two arrows meet at a circle, this represents an interaction that must occur between two compartments to give rise to another compartment (e.g. an uninfected cell must encounter a virus to become infected).

Variable	Interpretation	Comments
R	Number of activated, uninfected CD4 lymphocytes	These are the susceptible (S) entities in an SEIR model.
L	Number of latently infected CD4 lymphocytes	These are cells that are infected but are not producing any free virus. These are the exposed (E) entities in an SEIR model.
E	Number of actively infected CD4 lymphocytes	These are cells that are infected and are producing virus. These are the infectious (I) entities in an SEIR model.
V	Number of cell-free virions in the blood stream	These are normally not included in epidemic models.
Parameter	**Interpretation**	**Comments**
ω	Arrival rate of activated, uninfected CD4 cells per day	Phillips and McLean's group have opposing opinions on the nature of this parameter.
Γ	Rate at which uninfected CD4 cells arise per day	A value of 1.36 is equivalent to 3.4×10^8 cells per day in the whole body.
τ	Proportion of CD4 cells activated per day	This is used by Phillips in his assumption that the activation rate is constant.
χ	Rate of destruction of precursors to CD4 cells in the bone marrow per day	This is used by McLean's group in their assumption that the activation rate is dependent on the amount of virus.
μ	HIV-independent removal rate of uninfected CD4 cells per day	The average lifespan of a CD4 cell in the absence of HIV is 2 years so a typical value is 1.36×10^{-3}.
β	Rate of infection of CD4 cells per virion per day	This parameter cannot be directly estimated.

p	Proportion of cells becoming latently infected upon infection per day	This usually ranges from 0.1 to 0.9.
α	Activation rate of latently infected cells per day	A CD4 cell has a 1 in 25 chance of being activated and producing virus. A typical value is 0.04.
δ	Removal rate of actively infected cells per day	The average life span of an actively infected cell is 3 days so a typical value is 0.33.
π	Rate of production of virions by an actively infected cell per day	A single actively infected cell produces 100 virions on average per day.
σ	Removal rate of cell-free HIV from circulation per day	The average life span of a virus is half a day so a typical value is 2.

Table 1: Biological interpretations and typical values of the model variables and parameters used in **Figure 1**. All values are for a small quantity of tissue that normally contains 1000 CD4 lymphocytes. All interpretations and commentary except for those for χ and p are based on Phillips (1996).

Both Phillips' model and McLean's group's model consist of four ordinary differential equations describing the dynamics of four cell populations: activated, uninfected CD4 lymphocytes (R), latently infected cells (L), actively infected cells (E), and cell-free virus called virions (V). In the flow diagram, each variable represented by a circle changes at a rate equal to the sum of all of the arrows entering the circle minus all of the arrows exiting the circle. Note that all variables and parameters are assumed to be nonnegative. The equations are as follows.

$$\frac{dR}{dt} = \omega_i - \mu R - \beta RV, \; i = P, M \tag{1}$$

$$\frac{dL}{dt} = p\beta RV - (\alpha + \mu)L \tag{2}$$

$$\frac{dE}{dt} = (1-p)\beta RV + \alpha L - \delta E \tag{3}$$

$$\frac{dV}{dt} = \pi E - \sigma V \tag{4}$$

where $\omega_P = \Gamma\tau$ (Phillips) and $\omega_M = \frac{\Gamma}{1+\chi V}$ (McLean's group). (5)

At first glance, the set of equations resembles a compartmental epidemic model with four stages, and it certainly is! While our model is a variation of the typical SEIR model, it doesn't have an expression for the R in SEIR or the recovered entities. The only way for cells to escape the system is to die either by natural causes or by HIV replication, which explains why HIV infection is so problematic. Another aspect of our model that differentiates it from other epidemic models is the fact that virions or the V population are technically not the same thing as the CD4 cells, which make up the R, L and E populations. As seen in the figure above, CD4 cells and virions interact with each other and virus are produced from infected CD4 host cells, but CD4 cells do not become virions and vice versa. An intrinsic assumption of compartmental models in epidemiology is that only one type of population is of interest that becomes stratified into various stages of the infectious disease. Even though virions are not a condition or health state of the

population of interest, the CD4 cell population, they are included in our model so that we can monitor the temporal and behavioral dynamics of the virus population along with that of the CD4 cell population. If we are to classify our model as an epidemic model, we would treat it as a variation of the SEIR model where the recovered stage is left out and the pathogen's population is taken in.

Two other things that stand out from our model and our equations are the potential associations between the parameters δ and π and between the pair ω_p and ω_M. In the flow diagram, we depict both δ and π as exit rates of actively infected CD4 cells to the rest of the system with a solid arrow and a dashed arrow respectively, whereas in the equations, only the term δE is subtracted from the L population. Since δ is described as the removal rate of actively infected CD4 cells and π is described as the rate of virion production by an actively infected cell, there must be a direct relationship between the two where $\delta \geq \pi$. One possible scenario is $\delta = \pi + c$ where c is the death rate of actively infected cells by natural causes and the rate of virion production by an actively infected cell π is assumed to be equal to the death rate of actively infected cells by HIV-related causes. If that is the case, then it is assumed that the production of x virions results in the death of the same number x actively infected cells. There's also the possible scenario of $\delta = \pi$ but this is unlikely due to the fact that the production of virus in an infected host cell doesn't automatically lead to the host cell's death. These are all hypotheses that only a virology expert could provide answers to; for now, we stick to the model that we have aimed to emulate. Another interestingly potential link between parameters is that between ω_p and ω_M, the entry rates of new CD4 cells into the system. Saving further discussion for later, we know so far that both Phillip and McLean's group believe in a immigration rate Γ of new CD4 cells but they differ in their assumptions regarding the activation process – at a constant rate τ for Phillip, at a rate $\frac{1}{1+\chi V}$ that's dependent on the V population for McLean's group. It shall be interesting to witness how the difference in these assumptions will affect the population dynamics in the two models.

III. THE ANALYSIS

We begin our analysis of the model by looking at the boundedness of the solution to the differential equations. The equations in our model determine the evolution of a biological process described by four time-dependent state variables R(t), L(t), E(t), and V(t). Recall that each of the variables represents a population experiencing a specific stage of HIV infection. The state in HIV infection at any time t can be given by a single point (R, L, E, V) in phase space. As time varies, this point moves around in the phase space and traces out a trajectory. A common and easy way of examining the boundedness of the solution is to treat the motion of the solution in phase space (R, L, E, V) as the flow of a fluid with velocity (R*, L*, E*, V*) where $R^* = \frac{dR}{dt}$, $L^* = \frac{dL}{dt}$, $E^* = \frac{dE}{dt}$ and $V^* = \frac{dV}{dt}$. The divergence of this flow measures how the volume of a fluid particle changes and is given by $\frac{\partial R^*}{\partial t} + \frac{\partial L^*}{\partial t} + \frac{\partial V^*}{\partial t} + \frac{\partial E^*}{\partial t}$. A positive divergence means that the fluid volume is increasing locally, a negative divergence means that the fluid volume is

shrinking locally, and zero divergence means the fluid volume remains constant. Looking back to our equations, we get

$$\frac{\partial R^*}{\partial t} + \frac{\partial L^*}{\partial t} + \frac{\partial V^*}{\partial t} + \frac{\partial E^*}{\partial t} = -(2\mu + \beta V + \alpha + \delta + \sigma). \quad (6)$$

Given that all variables and parameters are nonnegative, the divergence has a negative value for both Phillips' model and McLean's group's model. Therefore, each small volume shrinks to zero as the t → ∞ at a rate independent of R, L or E but dependent on V. The consequence for the solution (R, L, E, V) is that every trajectory in phase space is eventually confined to a region of zero volume. In other words, none of the physical variables R, L, E or V will blow up and the solution will remain bounded in time for both models.

Next, we identify the steady-state solutions by solving the equilibrium conditions $\frac{dR}{dt} = 0$, $\frac{dL}{dt} = 0$, $\frac{dE}{dt} = 0$, and $\frac{dV}{dt} = 0$ simultaneously for R_{eq}, L_{eq}, E_{eq} and V_{eq}. A steady state of a system, which is sometimes called as an equilibrium or a stationary point, is a point in phase space from which the system will not change in time once that state has been reached. Since we have two models, we find the equilibria for each model separately. We aim to obtain all possible solutions for Phillips' model R_{eqP}, L_{eqP}, E_{eqP} and V_{eqP} and for McLean's group's model R_{eqM}, L_{eqM}, E_{eqM} and V_{eqM}. We use the popular and powerful computation program Mathematica to solve the equilibrium conditions for each model and obtain the following equilibria.

P: $\{R_{eqP} \to \frac{\Gamma}{\mu}\tau, L_{eqP} \to 0, E_{eqP} \to 0, V_{eqP} \to 0\}$ (7)

$$\{R_{eqP} \to \frac{\sigma}{\pi}\frac{\delta}{y_0}\frac{x_0}{\beta}, L_{eqP} \to \frac{\sigma}{\pi}\frac{\delta}{y_0}p * V_{eqP}, E_{eqP} \to \frac{\sigma}{\pi} * V_{eqP}, V_{eqP} \to \frac{-x_1+y_2}{x_4}\} \quad (8)$$

M: $\{R_{eqM} \to \frac{\Gamma}{\mu}, L_{eqM} \to 0, E_{eqM} \to 0, V_{eqM} \to 0\}$ (9)

$$\{R_{eqM} \to \frac{\sigma}{\pi}\frac{\delta}{y_0}\frac{x_0}{\beta}, L_{eqM} \to \frac{\sigma}{\pi}\frac{\delta}{y_0}p * V_{eqM}, E_{eqM} \to \frac{\sigma}{\pi} * V_{eqM}, V_{eqM} \to \frac{-(x_4+x_5+q)}{2x_6}\} \quad (10)$$

$$\{R_{eqM} \to \frac{\sigma}{\pi}\frac{\delta}{y_0}\frac{x_0}{\beta}, L_{eqM} \to \frac{\sigma}{\pi}\frac{\delta}{y_0}p * V_{eqM}, E_{eqM} \to \frac{\sigma}{\pi} * V_{eqM}, V_{eqM} \to \frac{-(x_4+x_5-q)}{2x_6}\} \quad (11)$$

where $x_0 = \alpha + \mu$, $y_0 = \alpha + \mu - p\mu$, $x_1 = \delta\mu\sigma(x_0)$, $x_2 = \pi\beta\Gamma\tau(x_0)$, $x_3 = \pi\beta\Gamma(x_0)$, $x_4 = \beta\delta\sigma(x_0)$, $x_5 = \delta\mu\sigma\chi(x_0)$, $x_6 = \beta\delta\sigma\chi(x_0)$, y_i is the same thing as x_i but with y_0 inside the parentheses instead of x_0, and $q = \sqrt{-4x_7(x_1 - y_3) + (x_4 + x_5)^2}$.

Before moving on to stability analysis, let's note a couple of things about the steady-state solutions. First, it is certainly very difficult to discuss the interpretation of a combination of multiple parameters. While one or two combinations have straightforward interpretations (e.g. $x_0 = \alpha + \mu$ is the total rate at which latently infected cells exit the system either by death or by becoming active) and some

can be described intuitively (e.g. $\frac{\sigma}{\pi}$ is like a survival rate for virus where a higher value means a higher chance of getting removed from the system; $\frac{\Gamma}{\mu}$ is like a survival rate for uninfected CD4 cells where a higher value means a higher chance of producing more cells to enter the system), majority of the combinations cannot be understood easily. As a result, we can only infer so much from the steady-state solutions. What we can say as truth though is that there are underlying relationships between the variables in their steady states. We know that regardless of which model we look at, the number of latently infected cells and the number of actively infected cells are correlated to the number of free virus. As seen in (7) through (11) above, an increase in the number of virus leads to an increase in the number of both latently infected cells and actively infected cells, although the other way around is not necessarily true. If V = 0, then L = E = 0 and R = some constant, which makes sense since the absence of HIV automatically denotes the absence of infected cells and the constant production of new CD4 cells. Interestingly, all the non-virus variables involve the ratio $\frac{\sigma}{\pi}$ in their steady-state solutions. Described earlier as the survival rate for virus in the system, the ratio $\frac{\sigma}{\pi}$ is directly proportional to variables R, L and E and high values of $\frac{\sigma}{\pi}$, which imply that the exit rate of virus is much greater than its entry rate, tend toward high values for R, L and E. It is also apparent that Phillips' model has two equilibria while McLean's group has three, two of which are conjugates. We will see how these sets of steady-state solutions compare to each other once we've performed stability analysis.

A steady state tells us the behavior of the solution at a single point. What is more meaningful to us is the long-term asymptotic behavior of the solution near the point. To get a better understanding of this behavior, we calculate the Jacobian matrix of our system of equations and compute its eigenvalues from which we can infer local stability. The Jacobian matrix, evaluated at an equilibrium, provides a linear approximation of the non-linear model near that equilibrium. In the vicinity of an equilibrium, the trajectories predicted using the model will be very nearly the same as those of the linear approximation. First, we analyze the stability properties of the equilibria in Phillips' model. The relevant Jacobian matrix is given as follows.

$$J_P = \begin{pmatrix} -\mu - \beta V & 0 & 0 & -\beta R \\ p\beta V & -\alpha - \mu & 0 & p\beta R \\ (1-p)\beta V & \alpha & -\delta & (1-p)\beta R \\ 0 & 0 & \pi & -\sigma \end{pmatrix}. \tag{12}$$

Evaluating the Jacobian matrix at the first equilibrium in (7) gives us

$$J_{P1} = \begin{pmatrix} -\mu & 0 & 0 & \frac{-\Gamma\tau\beta}{\mu} \\ 0 & -\alpha - \mu & 0 & \frac{p\Gamma\tau\beta}{\mu} \\ 0 & \alpha & -\delta & \frac{(1-p)\Gamma\tau\beta}{\mu} \\ 0 & 0 & \pi & -\sigma \end{pmatrix}. \tag{13}$$

We then attempt to obtain eigenvalues from the characteristic polynomial $\det(J_{P1} - \lambda I) = 0$. Unfortunately, identifying the eigenvalues from such fourth-order polynomial equations is difficult unless the polynomial factors. From this point on, we may not be able to fully reach a conclusion on the stability properties of our equilibria due to this difficulty. We will try to get as far as we can, but only so much can be said about the steady-state solutions at this point. With regards to this equilibrium, the characteristic polynomial can be factored into two terms. We get

$$(\lambda + \mu)\left(\lambda^3 + [x_0 + \delta + \sigma]\lambda^2 + \left[x_0(\delta + \sigma) + \delta\sigma - \frac{(1-p)\beta\Gamma\tau\pi}{\mu}\right]\lambda + \left[\delta\sigma x_0 - \frac{y_0(1-p)\beta\Gamma\tau\pi}{\mu}\right]\right) = 0$$

which is extremely complicated and almost impossible to derive eigenvalues from. One technique to deduce stability for this equilibrium is the use of the Routh-Hurwitz criterion, which will not be discussed further in this project. The Routh-Hurwitz is an important concept in control system theory and is mentioned here merely as a suggestion on future directions for the project. For now, we can only say that one of the eigenvalues of J_{P1} is $-\mu$, which is a negative value while the other eigenvalues are unmanageably hard to define explicit algebraic expressions for. Sadly, we cannot draw conclusions about the local stability of an equilibrium without knowledge of all the eigenvalues from its Jacobian matrix. The only method left for us to try to infer stability conditions is by running and analyzing multiple numerical and graphical simulations with varying values for the parameters and initial conditions, which will be done in the next section. Note that this method doesn't guarantee a fully correct assessment of the stability of an equilibrium. For now, we keep going and move on to the next equilibrium. The Jacobian matrix evaluated at the second equilibrium in (8) is

$$J_{P2} = \begin{pmatrix} -\mu - \frac{x_1 - y_2}{\delta\sigma x_0} & 0 & 0 & -\frac{\delta\sigma x_0}{\pi y_0} \\ p\frac{-x_1 + y_2}{\delta\sigma x_0} & -\alpha - \mu & 0 & p\frac{\delta\sigma x_0}{\pi y_0} \\ (1-p)\frac{-x_1 + y_2}{\delta\sigma x_0} & \alpha & -\delta & (1-p)\frac{\delta\sigma x_0}{\pi y_0} \\ 0 & 0 & \pi & -\sigma \end{pmatrix}. \tag{14}$$

Again, it is almost impossible to get the eigenvalues of the matrix above. We will examine the local stability of this equilibrium by graphical analysis.

We now attempt to perform stability analysis on McLean's group's model. The relevant Jacobian matrix is given as follows.

$$J_M = \begin{pmatrix} -\mu - \beta V & 0 & 0 & -\frac{\Gamma\chi}{(1+\chi V)^2} - \beta R \\ p\beta V & -\alpha - \mu & 0 & p\beta R \\ (1-p)\beta V & \alpha & -\delta & (1-p)\beta R \\ 0 & 0 & \pi & -\sigma \end{pmatrix}. \tag{15}$$

Notice that the only difference between J_P and J_M is the term in the first row and fourth column. An additional term $-\frac{\Gamma\chi}{(1+\chi V)^2}$ is added to the expression to account for the different expression for ω in McLean's group's model. Because there is barely any difference between the Jacobian matrices and the expressions for the stationary points are a bit more complicated in McLean's group's model than those in Phillips', we expect to face the same struggle and difficulty of obtaining explicit expressions for the eigenvalues. We will most likely investigate stability properties for these equilibria using graphical simulations as well. Evaluating the Jacobian matrix at the zero steady-state solutions in (9) gives us

$$J_{M1} = \begin{pmatrix} -\mu & 0 & 0 & -\Gamma\chi - \frac{\Gamma\tau\beta}{\mu} \\ 0 & -\alpha-\mu & 0 & \frac{p\Gamma\tau\beta}{\mu} \\ 0 & \alpha & -\delta & \frac{(1-p)\Gamma\tau\beta}{\mu} \\ 0 & 0 & \pi & -\sigma \end{pmatrix}. \tag{16}$$

The eigenvalues of J_{M1} are essentially the same as the eigenvalues of J_{P1} and therefore require little discussion. Reference to the Routh-Hurwitz condition is again necessary to understand the asymptotic behavior of the equilibrium. We proceed with computing the Jacobian matrices for the conjugate equilibria in (10) and (11) so we get

$$J_{M2} = \begin{pmatrix} -\mu + \frac{x_4+x_5+q}{2\delta\sigma\chi x_0} & 0 & 0 & -\frac{\Gamma\chi}{(1+\chi V)^2} - \frac{\delta\sigma x_0}{\pi y_0} \\ -p\frac{x_4+x_5+q}{2\delta\sigma\chi x_0} & -\alpha-\mu & 0 & p\frac{\delta\sigma x_0}{\pi y_0} \\ (p-1)\frac{x_4+x_5+q}{2\delta\sigma\chi x_0} & \alpha & -\delta & (1-p)\frac{\delta\sigma x_0}{\pi y_0} \\ 0 & 0 & \pi & -\sigma \end{pmatrix} \tag{17}$$

and

$$J_{M2} = \begin{pmatrix} -\mu + \frac{x_4+x_5-q}{2\delta\sigma\chi x_0} & 0 & 0 & -\frac{\Gamma\chi}{(1+\chi V)^2} - \frac{\delta\sigma x_0}{\pi y_0} \\ -p\frac{x_4+x_5-q}{2\delta\sigma\chi x_0} & -\alpha-\mu & 0 & p\frac{\delta\sigma x_0}{\pi y_0} \\ (p-1)\frac{x_4+x_5-q}{2\delta\sigma\chi x_0} & \alpha & -\delta & (1-p)\frac{\delta\sigma x_0}{\pi y_0} \\ 0 & 0 & \pi & -\sigma \end{pmatrix}. \tag{18}$$

Similarly, we lack the computing ability to obtain the eigenvalues and thus figure out the stability properties of these equilibria.

So far we have found two equilibria for Phillips' model and three equilibria for McLean's group's model and tried to perform stability analysis on these equilibria. The eigenvalues for the zero steady-state solutions are the same for both models, signifying that behavior near zero populations is the same for both models. Because of the mass-action assumption in the course of infection (the term βRV), the equations are non-linear and are therefore challenging to find a general solution to. For most non-linear models involving multiple variables, there is no general solution that can be used to predict the state of the system at any future point in time. It can be nearly impossible to get an overall picture of the dynamics

of the system from numerical simulations and stability analyses alone. Some advanced techniques that may be used to get an overall picture of the dynamics of the system are proportional transformation and separation of time scales, which will not be done in this project.

IV. THE IMPLEMENTATION

Numerical simulation of these equations use the Cash-Karp method, which is a fifth-order Runge-Kutta algorithm, with a fixed time step. We choose to do so because of several reasons – it is the RK algorithm that we are most familiar with; it is fifth-order accurate; the original papers used closely similar methods; and we already have code that implements it that we would simply need to modify. Implementations of algorithms from labs 5 and 6 are adapted to fit the problem at hand. A single input file in yaml format is created to contain the parameters and initial conditions for both models. An integrator subclass is made for each model to which derivatives are passed on. All the code is written in Python and is best viewed on IPython Notebook. Below is the code used to create the following simulations.

```
##############################################################################
# ATSC 409 Project
##############################################################################

from collections import namedtuple
import numpy as np
from matplotlib import pyplot as plt
plt.style.use('ggplot')
import warnings
warnings.simplefilter(action = "ignore", category = FutureWarning)
import yaml

def rkck_init():
    # % initialize the Cash-Karp coefficients
    # % defined in the tableau in lab 4,
    # % section "Embedded Runge Kutta"
    a = np.array([0.2, 0.3, 0.6, 1.0, 0.875])
  # c1 coefficients for the fifth order scheme
    c1 = np.array([37.0 / 378.0, 0.0, 250.0 / 621.0,
                   125.0 / 594.0, 0.0, 512.0 / 1771.0])
  # c2=c* coefficients for the fourth order scheme
    c2 = np.array([2825.0 / 27648.0, 0.0, 18575.0 / 48384.0,
                   13525.0 / 55296.0, 277.0 / 14336.0, .25])
    b = np.empty([5, 5], 'float')
  # the following line is ci - ci* in lab4, \Delta_est equationl
  # this is used to calculate \Delta_est = estError for the embedded
  # Runge Kutta  \sum_^6 (c_i -c_i^*)
    c2 = c1 - c2
  # this sets b values for same tableau
    b[0, 0] = 0.2
    b[1, 0] = 3.0 / 40.0
    b[1, 1] = 9.0 / 40.0
    b[2, 0] = 0.3
    b[2, 1] = -0.9
    b[2, 2] = 1.2
    b[3, 0] = -11.0 / 54.0
    b[3, 1] = 2.5
    b[3, 2] = -70.0 / 27.0
    b[3, 3] = 35.0 / 27.0
    b[4, 0] = 1631.0 / 55296.0
    b[4, 1] = 175.0 / 512.0
    b[4, 2] = 575.0 / 13824.0
    b[4, 3] = 44275.0 / 110592.0
```

```
    b[4, 4] = 253.0 / 4096.0
    return (a, c1, c2, b)

class Integrator:

    def set_yinit(self,initvars,uservars,timevars):
        ### uservars: parameters
        if uservars:
            self.config['uservars'].update(uservars)
        uservars = namedtuple('uservars', self.config['uservars'].keys())
        self.uservars = uservars(**self.config['uservars'])
        ### initvars: x y z a
        if initvars:
            self.config['initvars'].update(initvars)
        initvars = namedtuple('initvars', self.config['initvars'].keys())
        self.initvars = initvars(**self.config['initvars'])
        ### timevars is not really used here
        if timevars:
            self.config['timevars'].update(timevars)
        timevars = namedtuple('timevars', self.config['timevars'].keys())
        self.timevars = timevars(**self.config['timevars'])
        self.yinit = np.array([self.initvars.x, self.initvars.y, self.initvars.z, self.initvars.a])
        self.nvars = len(self.yinit)

    def __init__(self, coeffFileName):
        with open(coeffFileName, 'rb') as f:
            config = yaml.load(f)
        self.config = config
        # read in dt tstart tend
        timevars = namedtuple('timevars',config['timevars'].keys())
        self.timevars = timevars(**config['timevars'])
        # read in dtpassmin dtpassmax dtfailmin dtfailmax s rtol atol maxsteps maxfail
        adaptvars = namedtuple('adaptvars', config['adaptvars'].keys())
        self.adaptvars = adaptvars(**config['adaptvars'])
        self.rkckConsts = rkck_init()

    def __str__(self):
        return 'integrator instance with attributes initvars, timevars,uservars,adaptvars'

    def derivs(self, y, t):
        raise ValueError('derivs needs to be overridden in the derived class')
        return None

    def rkckODE(self, yold, timeStep, deltaT):
        # initialize the Cash-Karp coefficients
        # defined in the tableau in lab 4,
        a, c1, c2, b = self.rkckConsts
        i = self.initvars
        # set up array to hold k values in lab4
        derivArray = np.empty([6, self.nvars], 'float')
        ynext = np.zeros_like(yold)
        bsum = np.zeros_like(yold)
        estError = np.zeros_like(yold)
        # vector k1 in lab4 equation 3.9
        derivArray[0, :] = self.derivs(yold, timeStep)[:]
        # calculate step
        # c1=c_i in lab 4 notation, but c2=c_i - c^*_i
        y = yold
        for i in np.arange(5):
            bsum = 0.
            for j in np.arange(i + 1):
                bsum = bsum + b[i, j] * derivArray[j, :]
            # vectors k2 through k6 in lab4
            # pdb.set_trace()
            derivArray[i + 1, :] = self.derivs(y + deltaT * bsum, timeStep + a[i] * deltaT)[:]
            # partial sum of error in lab4 \Delta_est
            #  sum the error term
            estError = estError + c2[i] * derivArray[i, :]
            # print "estError: ",estError
            # 5th order estimate y_{n+1}
            ynext = ynext + c1[i] * derivArray[i, :]
        # final fifth order anser
        y = y + deltaT * (ynext + c1[5] * derivArray[5, :])
        # final 4th order estimate estimate
        estError = deltaT * (estError + c2[5] * derivArray[5, :])
        # print "estError final: ",estError
        timeStep = timeStep + deltaT
        # pdb.set_trace()
        return (y, estError, timeStep)
```

```
    def timeloopfixed(self):
        """fixed time step with estimated errors"""
        t = self.timevars
        yold = self.yinit
        yError = np.zeros_like(yold)
        yvals = [yold]
        errorList = [yError]
        timeSteps = np.arange(t.tstart, t.tend, t.dt)
        for theTime in timeSteps[:-1]:
            yold, yError, newTime = self.rkckODE(yold, theTime, t.dt)
            yvals.append(yold)
            errorList.append(yError)
        yvals = np.array(yvals).squeeze()
        errorVals = np.array(errorList).squeeze()
        return (timeSteps, yvals, errorVals)

class Integ(Integrator):
    def __init__(self, coeff_file_name,initvars=None,uservars=None,timevars=None):
        super().__init__(coeff_file_name)
        self.set_yinit(initvars,uservars,timevars)

##### Create an integrator subclass for Phillips' model.
class IntegP(Integ):
    def derivs(self, coords, t):
        x,y,z,a = coords
        u = self.uservars
        f = np.empty_like(coords)
        f[0] = u.gamma/(1 + u.chi*a) - u.mu*x - u.beta*x*a
        f[1] = u.prob*x*a - (u.alpha + u.mu)*y
        f[2] = (1 - u.prob)*u.beta*x*a + u.alpha*y - u.delta*z
        f[3] = u.pi*z - u.sigma*a
        return f

##### Create an integrator subclass for McLean's group's model.
class IntegM(Integ):
    def derivs(self, coords, t):
        x,y,z,a = coords
        u = self.uservars
        f = np.empty_like(coords)
        f[0] = u.gamma*u.tau - u.mu*x - u.beta*x*a
        f[1] = u.prob*x*a - (u.alpha + u.mu)*y
        f[2] = (1 - u.prob)*u.beta*x*a + u.alpha*y - u.delta*z
        f[3] = u.pi*z - u.sigma*a
        return f

##### Define the parameter values and initial conditions.
timevars=dict(tstart=0,tend=100,dt=0.01)
uservars=dict(gamma=0.5, tau=0.5, mu=0, beta=0.5, prob=0.5, alpha=0.5, delta=0.5, pi=0.5, sigma=0.5, chi=0.5)
initvars=dict(x=0.5,y=0.5,z=0.5,a=0.5)
params=dict(timevars=timevars,uservars=uservars,initvars=initvars)

##### Numerically solve Phillips' equations.
theSolver = IntegP('project.yaml',**params)
timevals, coords, errorlist = theSolver.timeloopfixed()
xvals,yvals,zvals,avals=coords[:,0],coords[:,1],coords[:,2],coords[:,3]

##### Numerically solve McLean's group's solutions.
theSolver2 = IntegM('project.yaml',**params)
timevals2, coords2, errorlist2 = theSolver2.timeloopfixed()
xvals2,yvals2,zvals2,avals2=coords2[:,0],coords2[:,1],coords2[:,2],coords2[:,3]

##### Create a plot for Phillips' model.
fig,ax = plt.subplots(1,1,figsize=(8,6))
ax.plot(timevals,xvals,label='R')
ax.plot(timevals,yvals,label='L')
ax.plot(timevals,zvals,label='E')
ax.plot(timevals,avals,label='V')
ax.set(title="Phillips' model",xlabel='time',ylabel='population count')
out=ax.legend()

##### Create a plot for McLean's group's model.
fig2,ax2 = plt.subplots(1,1,figsize=(8,6))
ax2.plot(timevals2,xvals2,label='R')
ax2.plot(timevals2,yvals2,label='L')
ax2.plot(timevals2,zvals2,label='E')
ax2.plot(timevals2,avals2,label='V')
ax2.set(title="McLean's group's model",xlabel='time',ylabel='population count')
out=ax2.legend()

################################################################################
```

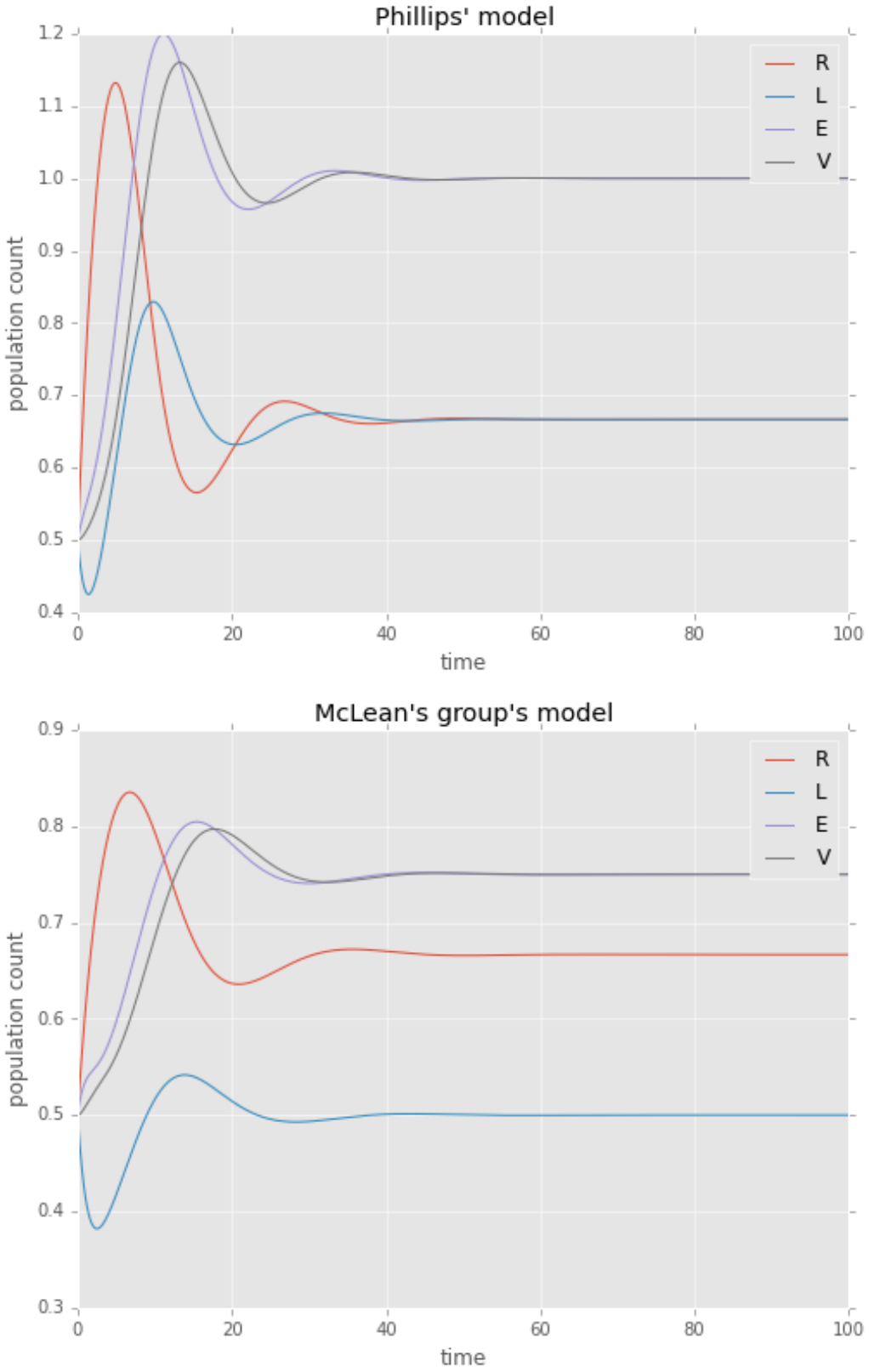

Figure 2: Simulations of (a) Phillips' model and (b) McLean's group's model with all parameters and initial conditions set to 0.5. Notice the huge difference in the range of the two plots: (a) 0 to 12 and (b) 0 to 1. Also, the rise and decline that occur before t=30 are much more dramatic in (a) than those in (b).

We now discuss the results of several numerical simulations and stability analyses. First, we acknowledge the unfortunate fact that it is essentially infeasible to pinpoint the exact stability properties of a stationary point in a non-linear model with multiple variables using numerical simulations alone. Given that our models have 9 parameters in 4 equations and assuming that all values are multiples of 0.1, it would take an incredibly long time to try out all the possible combinations of parameter values and initial conditions, especially since our parameters vary in their range of possible values. With that being said, we cannot say much about the stability properties of the equilibria that we have identified in (7) through (11) without using advanced techniques in control system and chaos theory. Based on a couple of numerical simulations, however, we postulate that the system always converges towards one of the steady states, which is determined by the parameter values and the initial conditions. Even though it is unknown which equilibrium the system will converge to, we know at the very least that the system will first experience oscillatory behavior and eventually reach some stability at a stationary point. Without obtaining the general solution to the equations or the exact stability properties of the equilibria, we are able to get an idea of the system's long-term population dynamics.

Similarities and differences between simulations of Phillips' model and McLean's group's model exemplify the similarities and differences between the two models. As seen in Figure 2 above, the main difference between the two models is that the range of values in the variables in Phillips' model is wider than that in McLean's group's model and it's all due to the difference in the expression for ω. Because McLean's group's model restricts the entry of new uninfected CD4 cells according to the virus population, fewer CD4 cells enter the system in this model, thus making the range of possible values smaller than in the other model. In terms of behavior, however, the two models demonstrate typically the same phenomenon: a series of oscillations or dramatic ups-and-downs with decreasing amplitude that flats out into a horizontal line. In addition, in cases where one model yields zero populations for L, E and V, the other model does as well. That means that otherwise, Phillips' model reaches stability with the equilibrium in (8) while McLean's group's model reaches stability with either the equilibrium in (10) or the equilibrium in (11).

V. CONCLUSIONS

- For both models, the solutions remain within a bounded region, which means that the solution will always be physically reasonable. Since the divergence depends on the number of free virus, an increase in the initial amount of free virus leads to an increase in the rate at which the solution tends toward a steady state.
- Phillips' model has two equilibria while McLean's group's model has three equilibria. Regardless of the parameter values and the initial conditions, the system always oscillates first and converges to one of these equilibria eventually. Therefore it is possible to predict the solution at a far enough future time.

- Phillips' model and McLean's group's model behave very similarly. The only difference is the range at which all possible values of the population count lie within.
- Advanced mathematical tests are required to fully determine the stability properties of the equilibria. Trying these out is a good next step for future directions.
- The results from Phillips (1996) and McLean et al (1991) are reproducible. It turns out that the oscillatory behavior of the cell populations during the acute phase is truly due to the natural population dynamics. Also, the implementation of the Cash-Karp method works well for this problem and is a reliable algorithm for numerically solving ordinary differential equations.

VI. REFERENCES

Austin, P., and Susan Allen. Numerical Methods. http://clouds.eos.ubc.ca/~phil/numeric/index.html. Web. 21 Nov. 2015.

"Global HIV/AIDS Overview." *Global HIV/AIDS Overview*. N.p., n.d. Web. 21 Nov. 2015.

McLean, Angela R., Vincent C. Emery, Alison Webster, and Paul D. Griffiths. "Population Dynamics of HIV within an Individual after Treatment with Zidovudine." *Aids* 5.5 (1991): 485-90. Web.

Otto, Sarah P., and Troy Day. *A Biologist's Guide to Mathematical Modeling in Ecology and Evolution*. Princeton: Princeton UP, 2007. Print.

Phillips, A. N. "Reduction of HIV Concentration During Acute Infection: Independence from a Specific Immune Response." *Science* 271.5248 (1996): 497-99. Web.