

# Modeling the Effects of Malaria Preventative Measures

Monroe P. Griffin

Wofford College

griffinmp@email.wofford.edu

Advisor: Dr. Anne J. Catlla

Mathematics Department

Wofford College

catllaaj@wofford.edu

## Abstract

Malaria is a serious and sometimes fatal epidemic affecting nearly half of the world's population and is the 5th leading cause of death by infectious disease world-wide. There are currently two recommended methods for prevention and eradication of the disease: insecticide-treated mosquito nets (ITNs) and indoor residual spraying (IRS), but efficacies and compliance vary from region to region. In this project, we look at the effects of ITNs and IRS as methods for eradication of malaria. To compare these methods, we develop a differential equation model and apply the next generation matrix method to determine the basic reproductive number. The differential equation model builds on classical SIR epidemiological models, with added constraints for the two preventative measures. Analysis shows that the effects of ITNs and IRS can help eradicate the disease. We find that the effect of ITNs is significantly greater than the effect of IRS. We conclude that the combination of compliance and efficacy for ITNs needs to be at least 61% and that there is no such percentage for IRS alone that will eradicate the disease. At a minimum, in combination, compliance and efficacy for ITNs needs to be at least 60% and compliance and efficacy for IRS needs to be at least 60%.

## 1 Introduction

Malaria is a serious and sometimes fatal epidemic affecting nearly half of the world's population. World-wide, malaria is the 5th leading cause of death from infectious diseases, but in Africa it is the 2nd leading cause [7]. In the United States, about 1500 new cases of malaria occur each year, mainly in travelers or immigrants [7]. In 2010, the World Health Organization (WHO) estimated 149 - 274 million cases of malaria occurred, with approximately 655,000 deaths [12].

Malaria is a parasitic disease transmitted by the bite of an infected mosquito or, more infrequently, by blood transfusion (surgery, needle sharing, birth) [7]. There are 4 different species that cause malaria in humans: *Plasmodium falciparum*, *P. vivax*, *P. ovale*, *P. malariae* [7]. Of the four, *P. falciparum* and *P. vivax* are the most common while *P. falciparum* is the most deadly [12]. In humans, the parasites multiply in the liver before infecting the red blood cells. Once the parasites reach the blood system, symptoms such as high fever, headache and vomiting occur, usually within 10 to 15 days after the mosquito bite [7, 12]. Also, once the parasites reach the human blood cells, the female *Anopheles* mosquito can become infected by taking a blood meal. It is important to note that the *Anopheles* mosquito feeds between dusk and dawn, often inside. Other than the preferred vertebrate human host, the parasites can survive in the female *Anopheles* mosquito, in fact, most of the parasites' growth cycle occurs in the mosquito [7].

Agencies have been working on eradication and prevention of malaria since the 1950s [7]. However, over the years, mosquitoes and parasites have developed resistance to different insecticides and drugs, respectively [7]. Currently, the WHO recommends the use of insecticide-treated mosquito nets (ITNs) and indoor spraying with residual insecticides (IRS) as forms of malaria control [12]. ITNs are simply nets with small apertures that are treated with approved insecticides that are non-toxic to humans and other animals. With ITNs, re-treatment of the net must occur after a short period of time (6 – 12 months). Over time, the insecticides degrade due to exposure to the sun and through washing [7]. Longer lasting insecticides are in development and are known as Long Lasting Insecticide-treated Nets (LLINs). ITNs are placed over beds to protect the users and are commonly known as insecticide-treated bed nets [7]. IRS makes use of the fact that mosquitoes tend to stay inside houses after taking their blood meal. As a result, approved insecticides are sprayed and applied to the walls of residences to kill mosquitoes. IRS has become a costly endeavor because of the expensive non-toxic insecticides [7].

ITNs and IRS are methods of prevention whereas antimalarial drugs are used in treatment of the disease. Antimalarial drugs are also recommended in combination to help prevent infection. Due to the resistance of the parasite to some medications in some areas, each country has a recommended list of antimalarial medications [7]. It is also recommended that antimalarial medications be used in combination with artemisinin derivatives, creating another class of effective antimalarial drugs. The combinations with artemisinin derivatives are called artemisinin-based combination therapies (ACTs), and are the most effective treatment for malaria, with a 95% cure rate against the *falciparum* strain of malaria, the most common strain [12]. Although there are methods for treatment of the disease, malaria is still a deadly disease because of the parasites' developed resistance to a number of malaria medications and because of the severe symptoms that develop quickly in humans [12].

To address biological issues and questions such as how to eradicate malaria, we employ mathematical modeling. For this project, we start from the classical epidemiological ODE Susceptible - Infected - Recovered (SIR) model. In the SIR model, different fractions of the population are classified as Susceptible to infection,  $S(t)$ , Infected,  $I(t)$ , or Recovered from infection,  $R(t)$ , where  $t$  represents time. The size of each group can change based on different parameters and the current values of  $S(t)$ ,  $I(t)$ , and  $R(t)$ , but the total population will remain constant. The SIR model is a simple model useful for modeling diseases such as chicken pox and influenza [15]. Modern epidemiology ODE models are based on these same SIR concepts of populations flowing between states [1, 3, 5, 11, 13]. For our model, we will apply the basic concepts of the SIR model to the vector-borne disease malaria. The term vector refers to the living organism that transmits a disease from one host to another, in our case, the mosquito.

This project focuses on the effectiveness of malaria preventative measures, specifically ITNs and IRS. We focus on these methods because their functions and effects are well known, and they are the only purely preventative measures recommended for possible malaria eradication. In section 2, we describe the derivation of our model from the classic SIR model and show the model solution, using the numerical computing program MATLAB. In section 3, we present the idea of the basic reproductive number and describe how we determine it for malaria. In section 4, we describe the results from our model and form a conclusion. And, finally, in section 5 we provide a summary.

## 2 The Model

In this section, we describe the formulation of our model and describe the solution. The solution is determined numerically using MATLAB to solve the system of equations.

### 2.1 Model Formation

We develop a differential equation model, which is useful in studying the overall system because we can look at the flow dynamics of individuals as well as the number of people at different stages of the disease cycle. In fact, we can determine the equilibrium state, the state in which the overall system does not change, which can be either endemic or not, to aid in analysis of the eradication of the disease.

For our model, we augment the basic SIR model to include preventative measures as seen previously by other researchers [3, 4, 5]. We include the effects of ITNs and IRS preventative measures based on the fact that both methods are recommended for the prevention of malaria by the WHO. We do not focus on other factors such as vaccination and partial immunity, since currently, there is no approved vaccination against malaria recommended by health agencies. There are vaccines being developed, but none have finished clinical testing to be approved for use [7].

In our model, the human and mosquito populations are functions of time,  $t$ . To distinguish between the human population and the mosquito population, we use the subscripts  $h$  and  $m$ , respectively. The human population was developed using a Susceptible - Exposed - Infected - Recovered - Susceptible (SEIRS) model, adapted from the classical SIR model. The different populations are denoted as follows: susceptible humans,  $S_h$ , exposed humans,  $E_h$ , infected humans,  $I_h$ , and recovered humans,  $R_h$ . Note that the entire human population can be expressed as the sum of these subpopulations:  $N_h = S_h + E_h + I_h + R_h$ . The “Exposed” category models the incubation period before a human becomes infectious, contrasting with the “Infected” category in which individuals are infected and can infect susceptible mosquitoes. In addition, the “Recovered” category is included as a way to deal with the issue of partial immunity after recovering from infection, a simplifying assumption of the model. Biological evidence exists for partial immunity of humans over long periods of exposure and treatment for malaria [4, 5, 7]. We allow for humans, after being treated for malaria, to have a partial immunity to the parasite for some time after treatment of the disease results in recovery. We do not focus on the long-term aspects of partial immunity that can develop over prolonged exposure to the disease.

Similarly, the mosquito population was developed using a Susceptible - Exposed - Infected (SEI) model. The different populations are similarly denoted as follows: susceptible mosquitoes,  $S_m$ , exposed mosquitoes,  $E_m$ , and infected mosquitoes,  $I_m$ . Also, note that the entire mosquito population can be expressed as the sum of these subpopulations:  $N_m = S_m + E_m + I_m$ . Since the lifespan of a mosquito is relatively small compared to the lifespan of humans, our model assumes that the mosquitoes will die from the infection. There is some data to suggest that the malaria parasite shortens the mosquito’s lifespan as well, confirming our choice of the SEI model for the mosquito population [3, 5, 9].

As time progresses, individuals move from one class to another at different rates based on the current state of the infection. In our model, humans and mosquitoes enter the populations at specific birth rates,  $\Lambda_h$  and  $\Lambda_m$ , respectively; die from natural causes at specific rates,  $\mu_h$  and  $\mu_m$ , respectively; and die from disease induced death at specific rates,  $\alpha_h$  and  $\alpha_m$ , respectively. We assume, in a similar manner as Esteva et al., that vertical transmission is not allowed, meaning

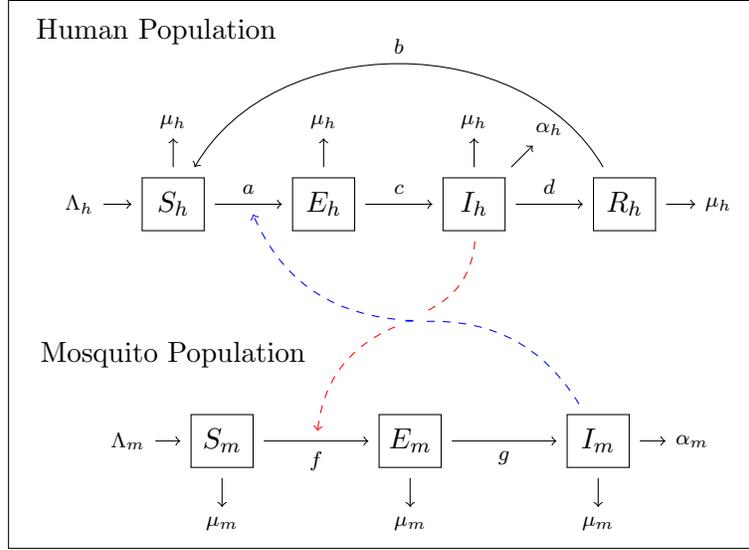


Figure 1: This schematic diagram shows the interactions between the human and mosquito populations as well as the flow of individuals between classes in the compartmental model. The solid arrows show the movement of individuals from one class to another. The dotted arrows show the interactions, through vector transmission, between the human and mosquito populations. The constants used in the differential equations appear with the associated arrow in the diagram. In addition, birth rates ( $\Lambda_h$  and  $\Lambda_m$ ), natural death rates ( $\mu_h$  and  $\mu_m$ ), and disease induced death rates ( $\alpha_h$  and  $\alpha_m$ ) are included before and after their respective arrows.

that individuals (i.e. new borns) enter the susceptible population instead of entering other classes (exposed, infected, or recovered) [11]. We allow individuals to move from the susceptible human population ( $S_h$ ) to the exposed human population at a rate proportional to both the size of the susceptible human population and the infected mosquito population and inversely proportional to the total human population,  $aS_hI_m/N_h$ . The inversely proportional term is used to make the model parameters independent of  $N_h$ . This follows from the true-mass-action assumption, which provides that  $N_h$  does not need to remain constant if there are a small number of cases of infection compared to the total population [8]. Members of the exposed class ( $E_h$ ) move to the infected humans class at a rate proportional to the number of individuals in the exposed human population,  $cE_h$ . Individuals move from the infected human population ( $I_h$ ) at a rate proportional to the size of the infected human populations,  $dI_h$ . Lastly, individuals move from the recovered human class ( $R_h$ ) to the susceptible human population at a rate proportional to the recovered human population size,  $bR_h$ . For the mosquito populations, mosquitoes move from the susceptible mosquito population ( $S_m$ ) to the exposed mosquito population at a rate proportional to both the size of the susceptible mosquito population and the size of the infected human population and inversely proportional to total mosquito population,  $fS_mI_h/N_m$ . This term also uses the true-mass-action assumption. Finally, mosquitoes move from the exposed mosquito population ( $E_m$ ) at a rate proportional to the number of individuals in the exposed mosquito population,  $gE_m$ .

In addition to considering the various stages of the disease, we model the effects of malaria preventative measures. The symbols *ITN* and *IRS* are used respectively for ITN and IRS pro-

tection. Incorporated in each parameter is the efficacy of the method and the percent compliance of the method ( $ITN = ITN_{eff} \cdot ITN_{comp}$  and  $IRS = IRS_{eff} \cdot IRS_{comp}$ ). In the literature, there are no common values accepted for the efficacies of ITNs or IRS. Comprehensive studies of these methods conclude wide-ranging values for the efficacies because the studies were conducted over different time spans and in different regions [2, 6]. In order to incorporate these two prevention methods into the model, we look closely at how each method is employed and the effect it has on the spread of the disease. The effect of using ITNs is two-fold: (1) ITNs can reduce the number of bites from mosquitoes as they physically provide a barrier between the mosquito and the human, and (2) ITNs reduce the population of the mosquitoes by killing them after they land on the treated net. In contrast, the effect of IRS is the killing of the mosquito population as they interact with the sprayed walls; however, IRS is not able to provide a mechanism for preventing the mosquito bites that spread the disease. The effects of ITNs are incorporated in the rates of transmission from the susceptible human class to the exposed human class and from the susceptible mosquito class to the exposed mosquito class with a parameter measured as a percent,  $(1 - ITN)$ , and also a separate death rate in each of the mosquito subpopulations:  $S_m$ ,  $E_m$ , and  $I_m$ . When  $ITN = 1$ , there is no movement from the susceptible human class to the exposed human class as well as no movement from the susceptible mosquito class to the exposed mosquito class. When  $ITN = 0$ , the nets have no effect and the disease spreads as such. The effect of IRS is incorporated solely in the death of the mosquito populations:  $S_m$ ,  $E_m$ , and  $I_m$ . The parameters  $h$  and  $j$  are respective rates for the removal of mosquitoes from the different classes associated with ITNs and IRS; they have units of per day ( $\text{day}^{-1}$ ). The values for  $ITN$  and  $IRS$  range from 0 to 1 inclusive as we increment each constant to account for a wide range of efficacies and compliance applicable.

A schematic depiction of the model is given in Figure 1. Our assumptions, as discussed, lead to the development of the following differential equations:

$$\frac{dS_h}{dt} = -a \frac{I_m S_h (1 - ITN)}{S_h + E_h + I_h + R_h} + bR_h + \Lambda_h - \mu_h S_h \quad (1a)$$

$$\frac{dE_h}{dt} = a \frac{I_m S_h (1 - ITN)}{S_h + E_h + I_h + R_h} - cE_h - \mu_h E_h \quad (1b)$$

$$\frac{dI_h}{dt} = cE_h - dI_h - \mu_h I_h - \alpha_h I_h \quad (1c)$$

$$\frac{dR_h}{dt} = dI_h - bR_h - \mu_h R_h \quad (1d)$$

$$\frac{dS_m}{dt} = -f \frac{I_h S_m (1 - ITN)}{S_m + E_m + I_m} + \Lambda_m - \mu_m S_m - h \cdot ITN \cdot S_m - j \cdot IRS \cdot S_m \quad (1e)$$

$$\frac{dE_m}{dt} = f \frac{I_h S_m (1 - ITN)}{S_m + E_m + I_m} - gE_m - \mu_m E_m - h \cdot ITN \cdot E_m - j \cdot IRS \cdot E_m \quad (1f)$$

$$\frac{dI_m}{dt} = gE_m - \mu_m I_m - \alpha_m I_m - h \cdot ITN \cdot I_m - j \cdot IRS \cdot I_m. \quad (1g)$$

## 2.2 Model Solution

We employ numerical methods to solve Equations (1), using the parameter values in Table 1. These simulations are performed using MATLAB's built-in `ode45` function. The `ode45` function evaluates the differential equations using an explicit 4<sup>th</sup> order Runge-Kutta method for solving ODEs. In these simulations, initial population sizes are chosen based on populations used in similar studies

Description	Symbol	Value	Source
Human Infection rate	$a$	$.25^I$	Estimated
Human Re-susceptibility rate	$b$	$.04^{II}$	Estimated
1/(latent period for humans)	$c$	$1/14$	[3]
Human Recovery rate	$d$	$.205^{III}$	Estimated
Mosquito infection rate	$f$	$.415^{IV}$	Estimated
1/(latent period for mosquitoes)	$g$	$1/12$	[3]
Parameter for ITN rate	$h$	$1/365^V$	Estimated
Parameter for IRS rate	$j$	$1/365^V$	Estimated
Human birth rate	$\Lambda_h$	$.028$	[3]
Natural human death rate	$\mu_h$	$.0000391$	[3]
Human death rate due to the disease	$\alpha_h$	$.0004$	[3]
Mosquito birth rate	$\Lambda_m$	$6^{VI}$	Estimated
Natural mosquito death rate	$\mu_m$	$.04$	[3]
Mosquito death rate due to disease	$\alpha_m$	$.01$	[3]

<sup>I</sup> product of human transmission probability (.5) and biting rate (.5), from [3]

<sup>II</sup> product of recovery rate (.005) and modified recovery rate (8.04), from [3]

<sup>III</sup> sum of treatment rate (.2) and recovery rate (.005), from [3]

<sup>IV</sup> product of mosquito transmission probability (.83) and biting rate (.5), from [3]

<sup>V</sup> division by 365 allows for the rate to have the correct units of days<sup>-1</sup>

<sup>VI</sup> mosquito birth rate, 12 eggs/day/mosquito, where half are assumed to be male and the other half female

Table 1: This table gives the constants and parameters used in the model.

[4, 3, 5]. The initial conditions used are  $S_h = 300$ ,  $E_h = 0$ ,  $I_h = 1$ ,  $R_h = 0$ ,  $S_m = 300$ ,  $E_m = 0$ , and  $I_m = 0$ . Note that these initial populations are nondimensional. We find that the initial condition does not affect our analysis. We start with high susceptible human and mosquito populations compared to the other states. In addition, we must have a source of initial infection, which we represent with the initial condition of  $I_h = 1$ .

Figure 2 presents typical solution plots for the human population and the mosquito population. As the individuals leave one class for another, the curves change accordingly. Once the curves level off, we stop the simulation as the populations reached equilibrium. Overall, without any intervention strategy ( $ITN = 0$  and  $IRS = 0$ ), the final infected human and exposed human populations are non-zero; however, they are smaller than the non-infectious states (susceptible and recovered classes). Since the populations are non-zero, malaria has not been eradicated at the equilibrium state. For the mosquito classes, notice that the equilibrium state contains almost an equal number of infected mosquitoes as susceptible mosquitoes. It is important to note that, here, the built-in MATLAB solver will find a stable solution, which may have a dependence on initial conditions, if the system has more than one steady state. However, we later confirm with our analysis, that there is at most one positively steady state solution, indicating that there is not a dependence on the initial conditions chosen.

Figure 3 gives another example of plots of the populations over time, factoring in some intervention strategies ( $ITN = .5$  and  $IRS = .5$ ). Unlike Figure 2, the different populations and classes do not go to an equilibrium. In this case, with some intervention strategies, the susceptible human population appears to continue growing although the disease has not been eradicated. This

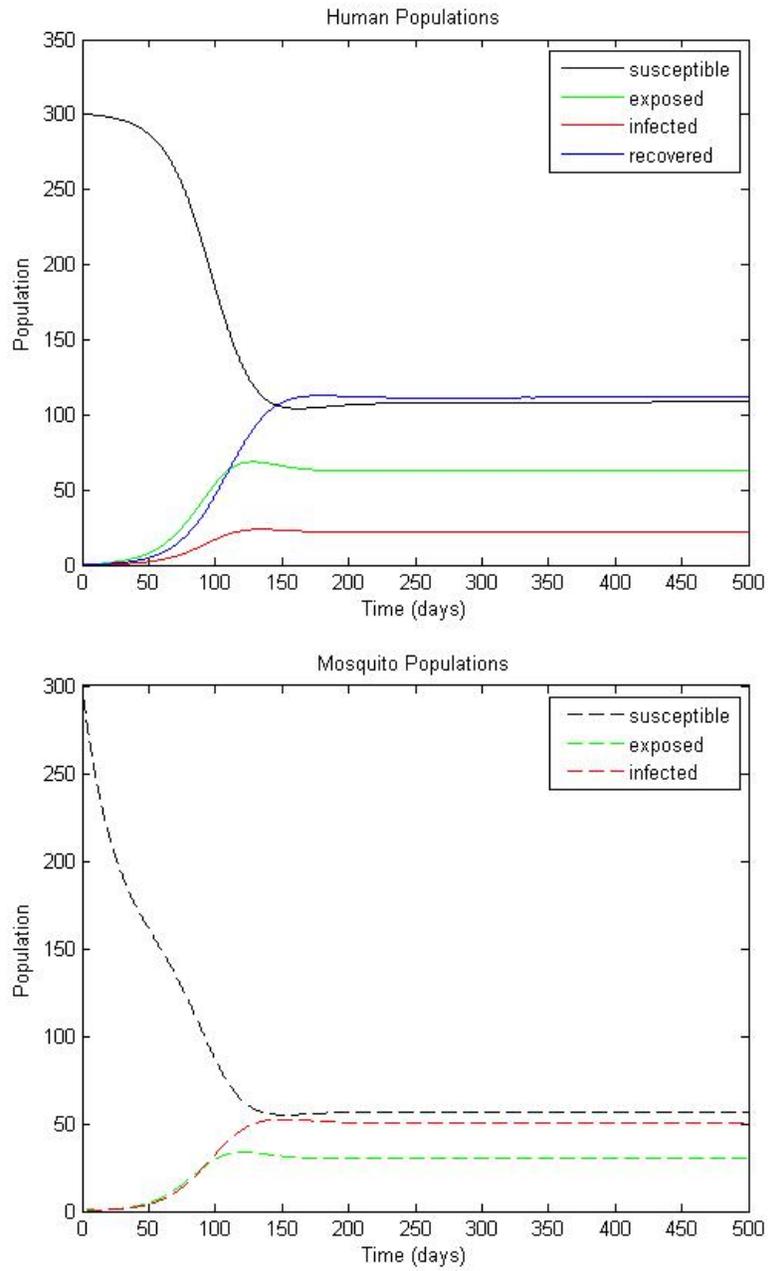


Figure 2: These are plots of the human population (top) and the mosquito population (bottom) as functions of time. Note that both reach a non-zero equilibrium. This figure was created using initial conditions from the text, using the parameter values in Table 1,  $ITN = 0$ , and  $IRS = 0$ . The equations were solved in MATLAB using the built-in `ode45` solver.

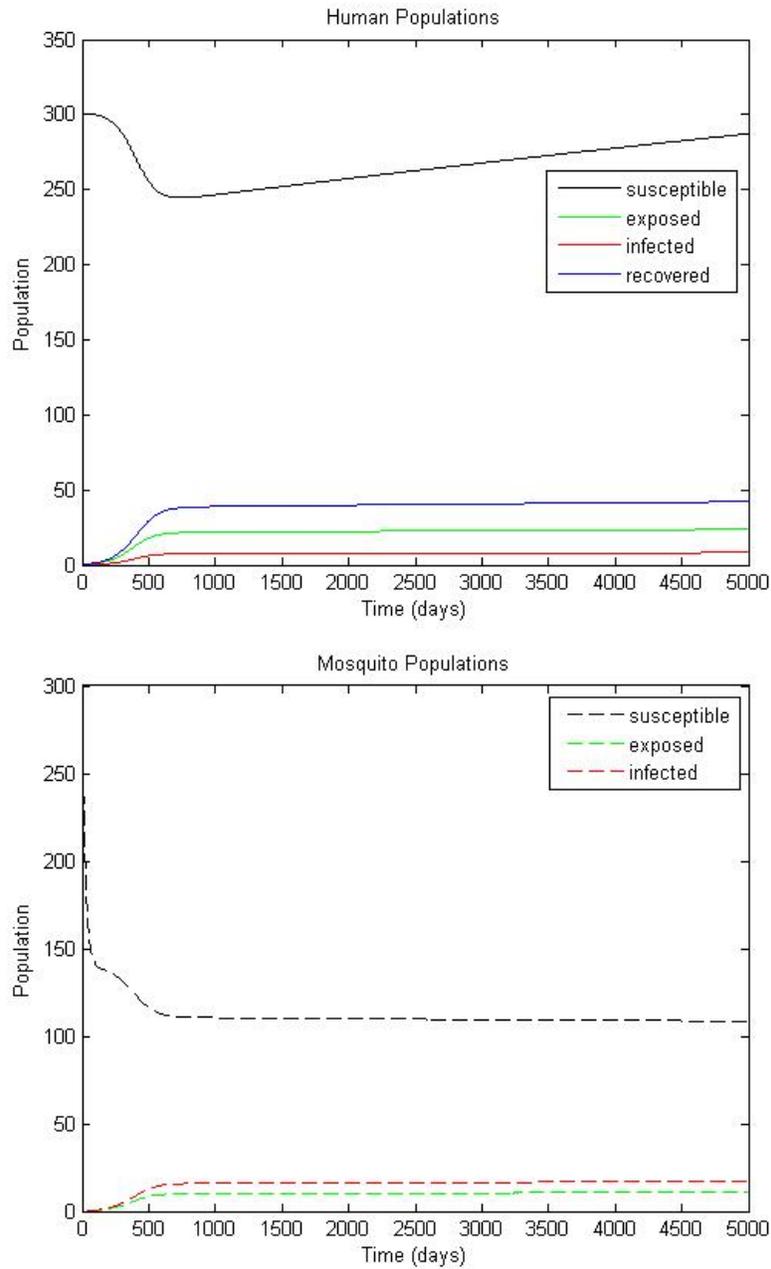


Figure 3: These plots show the human population (top) and the mosquito population (bottom) as a function of time. deleted statement about equilibrium. This figure was created using initial conditions from the text, using the parameter values in Table 1,  $ITN = .5$ , and  $IRS = .5$ . The equations were solved in MATLAB using the built-in `ode45` solver.

phenomenon is due to the fact that the mosquito population becomes so small that the birth rate of humans overwhelms the death rate.

### 3 Reproductive Number

In epidemiology, the reproductive number is often used to determine the seriousness of the disease in question. It is defined as “the expected number of secondary cases produced by a single (typical) infection in a completely susceptible population.” [10] In other words, it is a measure of how fast a disease will spread through a population. As defined, a reproductive number less than one will cause the disease to die out; whereas, a reproductive number greater than one classifies the disease as endemic [3].

Mathematically, there are a number of different methods to compute a reproductive number. The most common method to determine the reproductive number of a system of differential equations is to find the dominant eigenvalue of the steady-state Jacobian matrix for the linearized system. For a general discussion of linear stability analysis, see [14]. However, in our study, we focus on using another type of linear analysis, the next generation method, to find the reproductive number and compare it to the results from a linear stability analysis. Details of the linear stability analysis can be found in Appendix A. In order to determine the reproductive number for our system, since our model does not allow for an analytic expression to be determined, we use MATLAB to symbolically and numerically solve the system and determine the necessary eigenvalue and reproductive numbers. See Appendices A & B for the annotated MATLAB code.

In the next generation method, we look at a smaller set of our original equations [10]. The next generation matrix,  $\mathbf{G}$ , is a square matrix where each element is the expected number of secondary infections from one infected individual for each type of subpopulation. In other words, each element is a reproductive number; however, the basic reproductive number for the disease is the dominant eigenvalue of  $\mathbf{G}$ . Note that the rate of new infection,  $\mathbf{F}$ , is given by the product of  $\mathbf{G}$  with the rates of transfer of infection,  $\mathbf{V}$ . So, to compute  $\mathbf{G}$  we use  $\mathbf{G} = \mathbf{F}\mathbf{V}^{-1}$ . Mathematically,  $\mathbf{F}$  and  $\mathbf{V}$  are the Jacobian matrices,

$$\mathbf{F} = \left[ \frac{\partial F_i(x_o)}{\partial x_j} \right] \text{ and } \mathbf{V} = \left[ \frac{\partial V_i(x_o)}{\partial x_j} \right],$$

where  $F_i$  are new infections,  $V_i$  is the transfer of infections from one compartment to another, and  $x_o$  is the disease-free equilibrium state [10].

To develop  $\mathbf{F}$  and  $\mathbf{V}$  for our model, we modify the original differential equations to reflect only the creation of new infections or the transfer of current infections. To obtain these equations, we narrow our focus to the classes of “infected” individuals:  $\frac{dE_h}{dt}$ ,  $\frac{dI_h}{dt}$ ,  $\frac{dE_m}{dt}$ , and  $\frac{dI_m}{dt}$ . For the equations later used in the  $\mathbf{F}$  matrix, we focus on the aspects of the equations that pertain to causing new

infections, denoted using the squiggle,  $\tilde{X}$ . These equations are given by:

$$\frac{d\tilde{E}_h}{dt} = a \frac{I_m S_h (1 - ITN)}{S_h + E_h + I_h + R_h} \quad (2a)$$

$$\frac{d\tilde{I}_h}{dt} = 0 \quad (2b)$$

$$\frac{d\tilde{E}_m}{dt} = f \frac{I_h S_m (1 - ITN)}{S_m + E_m + I_m} \quad (2c)$$

$$\frac{d\tilde{I}_m}{dt} = 0. \quad (2d)$$

Focusing on the aspects of the original equations that relate to the transfer of infection, we develop equations later used in the  $\mathbf{V}$  matrix. Note the sign change for the equations as we describe the rate of infection instead of the flow of individuals. We note these equations using the hat,  $\hat{X}$ , and they are given by:

$$\frac{d\hat{E}_h}{dt} = cE_h + \mu_h E_h \quad (3a)$$

$$\frac{d\hat{I}_h}{dt} = -cE_h + dI_h + \mu_h I_h + \alpha_h I_h \quad (3b)$$

$$\frac{d\hat{E}_m}{dt} = gE_m + \mu_m E_m + h \cdot ITN \cdot E_m + j \cdot IRS \cdot E_m \quad (3c)$$

$$\frac{d\hat{I}_m}{dt} = -gE_m + \mu_m I_m + \alpha_m I_m + h \cdot ITN \cdot I_m + j \cdot IRS \cdot I_m. \quad (3d)$$

To find  $\mathbf{F}$ , we compute the Jacobian of Equations (2) evaluating at the disease-free state,  $S_h = N_h$ ,  $E_h = 0$ ,  $I_h = 0$ ,  $R_h = 0$ ,  $S_m = N_m$ ,  $E_m = 0$ , and  $I_m = 0$ . To find  $\mathbf{V}$ , we compute the Jacobian of Equations (3) also evaluating at the disease-free state. Afterwards, matrix multiplication and eigenvalue calculations lead to the reproductive number. In solving for the dominant eigenvalue of  $\mathbf{G}$ , specific values for  $ITN$  and  $IRS$  are used. These computations are done in MATLAB using the appropriate built-in functions, as described in Appendices A & B.

## 4 Results and Discussion

In our analysis of the preventative measures of malaria, we ran a number of simulations changing the amount of the two different preventative measure strategies, ITNs and IRS, ranging from 0 to 1 ( $ITN \in [0, 1]$  and  $IRS \in [0, 1]$ ). In total, 121 different combinations were used, using increments of .1 for each strategy. For each combination, the basic reproductive number was found using the next generation method as described in Section 3. Figure 4 displays the final data graphically, plotting the reproductive number as a function of  $ITN$  and  $IRS$ . In general, we see that as the use of preventative measures increases, the reproductive number gets closer to 1, eventually falling below that critical value. We note that this value occurs for some value of  $ITN \approx .60$ .

Figure 5a shows the effect of the reproductive number as the value of  $ITN$  increases, without the effect of IRS ( $IRS = 0$ ), using the next generation method. As the value of  $ITN$  increases, the reproductive number decreases eventually below 1. From the plot, we see that a moderately

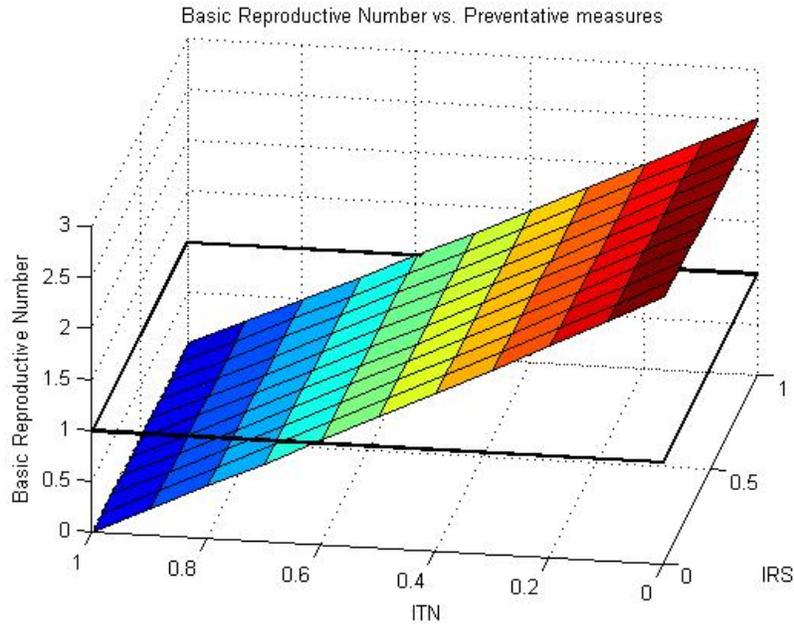


Figure 4: This plot shows the basic reproductive number as a function of  $ITN$  and  $IRS$  and was created using the next generation method data. Notice that the reproductive number drops below one near  $ITN = .60$ ; this occurs near the border of the cyan and green shaded regions. This plot was made in MATLAB using the `surf` built-in function.

high value for  $ITN$  is necessary to cause the disease to go from endemic to dying out. We estimate the critical value for  $ITN$  to be .61, by looking closely at the region  $ITN \in [.55, .65]$ . Figure 5b shows the trend of the reproductive number as  $ITN$  increments between .55 and .65. This value for  $ITN$  means that it is possible to eradicate the disease with only using ITNs, but it would require a moderately large compliance as well as a moderately large efficacy for ITNs to do so. We need to remember that the variable  $ITN$  represents the combination of compliance and efficacy of ITNs. In fact, there is a limited range of combinations that would allow for the combination of compliance and efficacy to be at least 61%. In the literature, we do not see values for efficacy of ITNs near that range; they are consistently below 61% [6].

In Figure 6, we see the effects of IRS without any effects from ITNs ( $ITN = 0$ ). As the value of  $IRS$  increases, the reproductive number does decrease; however, not much across the whole interval  $IRS \in [0, 1]$ . In fact, with  $ITN = 0$ , we see that the reproductive number decreases by .0966, from 2.6117 to 2.5151. The effect of IRS is not enough alone to cause the reproductive number to go below 1, causing the disease to die out.

However, IRS can be somewhat effective in combination with ITNs. When combined, the reproductive number falls below one when  $ITN = .60$  and  $IRS = .60$ . In combination, any increase in these parameters above these critical points will cause the basic reproductive number to continue decreasing below one, helping to eradicate the disease.



## 5 Summary

In this project, we created a system of differential equations to model the spread of malaria, including the effects of two major preventative measures: ITNs and IRS. Our equations work off the assumption that individuals move between classes, different for the human and mosquito populations, depending on the size of those classes. In our model, the two populations interact via the bite from the malaria vector (mosquito). We determine the basic reproductive number of the disease by using the next generation method, using numerical methods and techniques to show how the reproductive number varies with *ITN* and *IRS*.

From the simulation data, we see that both prevention strategies are able to reduce the reproductive number for malaria. However, the effect of ITNs is much greater than the effect of IRS. In combination, use of both ITNs and IRS, the reproductive number is lowered more than only using one of the strategies. Using solely ITNs, the reproductive number drops below zero at a value of .61. That number is slightly lowered with the effects of IRS. Although the disease can be eradicated with ITNs and IRS, there needs to be a high combination of compliance among users of the strategies and high efficacies of each strategy. At some point, if the efficacies are not high enough, even 100% compliance will not cause *IRS* and *ITN* to be large enough to make the reproductive number to be below one.

With this project, there are areas for future work. There are other epidemiological models that could obtain the basic reproductive number, but use a different approach. Those methods include network models and cellular automata models. It would be helpful to compare the results of those models with the results obtained with our differential equation model. Other suggestions would be to obtain precise efficacies for the different strategies. In the literature, an accepted efficacy value for either method does not currently exist, hence our reasoning for sampling all possible values in this study. However, using the correct efficacies, when determined, compliance can be calculated in order to help areas try to eradicate malaria. As a next step, a sensitivity analysis of this model could be useful to understand the effect of small perturbations in these values as agencies work to narrow down correct efficacies. In addition, there are other preventative measures in development, such as ACTs and vaccinations, that could be added to the model as one more aspect in trying to identify the reproductive number and aid in fighting this disease.

## 6 Acknowledgments

The author would like to thank Dr. Anne J. Catlla for her continued support, guidance, and knowledge as adviser to this project. He would also like to thank Drs. A. Shiflet and G. Shiflet for their time and contributions as committee members in the defense. In addition, he would like to acknowledge Wofford College for providing necessary computational and research resources in completing this project.

## A Linear Stability Analysis

### A.1 Discussion and Results

In order to find the transition point where the disease goes from endemic to dying out, we conduct a linear stability analysis. We start by inputting the differential equations and constants as symbolic

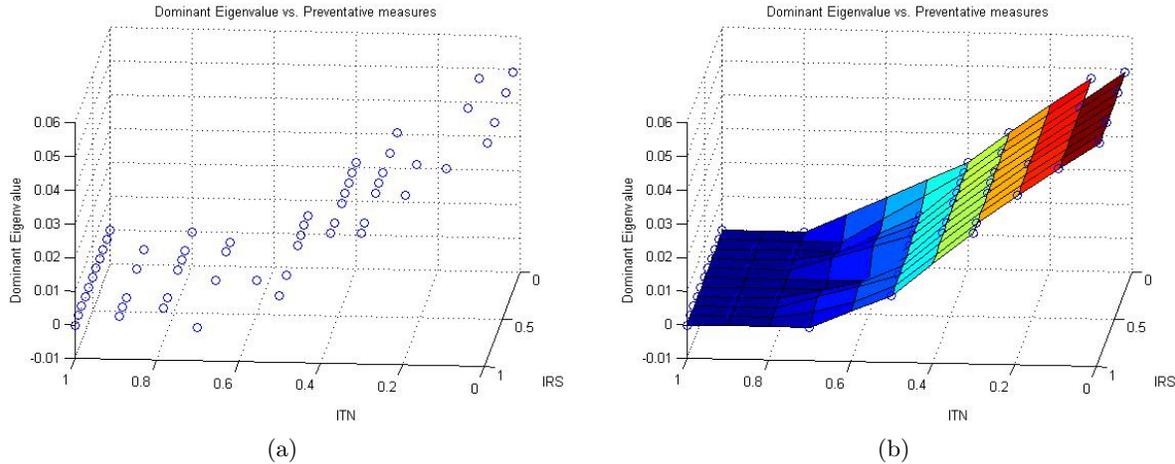


Figure 7: These plots show the dominant eigenvalue as a function of  $ITN$  and  $IRS$  and were created using the linear stability analysis data. It is important to note that not all  $ITN/IRS$  combinations yield a dominant eigenvalue due to the lack of an equilibrium solution. Plot (a) shows the data as a scatter plot while Plot (b) shows the data with an interpolated fit. Notice that the dominant eigenvalue drops below zero for  $ITN \approx .6$ ; this region is blue shaded. Both plots were made in MATLAB: Plot (a) was made using the `scatter3` built-in function and Plot (b) was made using the `griddata`, `surf`, and `plot3` built-in functions.

equations and variables. Using MATLAB's built-in `solve` function, we solve the system of equations, each set to zero, to find the fixed points of the system. In order to continue the analysis, we convert the matrix containing the symbolic fixed points to a numeric fixed point matrix. Next, we must eliminate all of the negative and complex fixed points. Physically, a population can never be negative, and similarly, populations with imaginary parts are physically impossible. After finding the positive real fixed points, we substitute each of them individually into the Jacobian matrix. Using the built-in function `jacobian`, MATLAB symbolically finds the Jacobian matrix for the system. Subsequently, we substitute the fixed points individually into the Jacobian before finding the eigenvalues. The built-in `eig` function can be used to find the eigenvalues of a given matrix. At this point in the analysis, we are only concerned with the real parts of the eigenvalue and removing the complex part does not adversely affect the analysis. After we obtain only positive and real eigenvalues for the system, we must find the dominant eigenvalue using the `max` function. Because this codes runs slowly, we also used an alternative approach to find the actual basic reproductive number.

Using the results of the linear stability analysis, we find that the dominant eigenvalue falls below zero for the same value of  $ITN$  and  $IRS$  as the next generation method analysis. Figure ?? gives the overall trend of the eigenvalue for different values of  $ITN$  and  $IRS$ . From this plot we initially see the effect of both  $ITN$  and  $IRS$ . We note that for some critical value for  $ITN$ , the eigenvalue falls below zero ( $ITN \approx .6$ ). This effect of the eigenvalue falling below zero translates to the reproductive number falling below one.

## A.2 MATLAB Code

The following MATLAB code automates the linear stability analysis on the ODE model. This code frequently switches between symbolic, numerical, and decimal variables to accommodate the requirements of the different functions and processes used. The author would like to point out the use of `clearvars -except` as a method of clearing all variables except those that follow the command, which is necessary for our overall counter and the variables used in saving. The `vpa` function converts variables from symbolic or numeric to a decimal form. It is important to note that this code allows for data to be saved during the simulation in case the program is accidentally terminated in the middle of the execution. Please note that this code runs very slowly. We also note that not all ITN/IRS combinations run with this code. For the situations where the system does not reach an equilibrium, the `solve` command fails. In another function, we implemented the `try` and `catch` command to deal with the failed attempts. The details of that code are not given here.

```
1 %% This code is useful for finding the dominant eigenvalue of the linearized system
2 %% We use it in the first part of the reproductive number analysis.
3
4 %% Note: This simulation runs slowly because of lines 66 & 67. The solve function
5 %% runs slowly in this situation and may fail if an equilibrium solution
6 %% cannot be determined.
7
8 clear all
9 format long
10 tInitial = tic; % Used to determine run time of simulation
11
12 % ITN values are in the first row, IRS values are in the second row, this segment
13 %% is used for running the whole simulation for different values of ITN and IRS.
14 IRSITNmat = [zeros(1,11) ones(1,11)*.1 ones(1,11)*.2 ones(1,11)*.3 ones(1,11)*.4 ...
15             ones(1,11)*.5 ones(1,11)*.6 ones(1,11)*.7 ones(1,11)*.8 ones(1,11)*.9 ...
16             ones(1,11);0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1 ...
17             0:.1:1 0:.1:1];
18
19 % Used in the saving process
20 IRStoSave = []; ITNtoSave = []; scriptRstoSave = []; ShtoSave = []; EhtoSave = []; ...
21     IhtoSave = []; RhtoSave = []; SmttoSave = []; EmtoSave = []; ImtoSave = [];
22
23 %% Each run of the for loop determines the dominant eigenvalue of system,
24 %% using the specified ITN and IRS values
25 for overall = 1:size(IRSITNmat,2)
26     clearvars -except overall IRSITNmat tInitial IRStoSave ITNtoSave ...
27         scriptRstoSave ShtoSave EhtoSave IhtoSave RhtoSave SmttoSave ...
28         EmtoSave ImtoSave %% Clears all variable except the save matrices
29
30     %% Define functions
31     syms Sh Eh Ih Rh Sm Em Im
32     %% Constants
33     a = .25;
34     b = .04;
35     c = 1/14;
36     d = .205;
37     f = .83*.5;
```

```

38     g = 1/12;
39     h = 1/365;
40     j = 1/365;
41     Lambdah = .028;
42     muh = .0000391;
43     alphah = .0004;
44     Lambdam = 6;
45     mum = .04;
46     alphas = .01;
47     % Changes ITN and IRS value with each run of the for loop
48     ITN = IRSITNmat(1,overall); % ITN values are stored in the first row
49     IRS = IRSITNmat(2,overall); % IRS values are stores in the second row
50
51     %% Equations
52     Shdot = -a*Im*Sh*(1-ITN)/(Sh+Eh+Ih+Rh)+b*Rh+Lambdah-muh*Sh;
53     Ehdot = a*Im*Sh*(1-ITN)/(Sh+Eh+Ih+Rh)-c*Eh-muh*Eh;
54     Ihdot = c*Eh-d*Ih-muh*Ih-alphah*Ih;
55     Rhdot = d*Ih-b*Rh-muh*Rh;
56     Smdot = -f*Ih*Sm*(1-ITN)/(Sm+Em+Im)+Lambdam-mum*Sm-h*ITN*Sm-j*IRS*Sm;
57     Emdot = f*Ih*Sm*(1-ITN)/(Sm+Em+Im)-g*Em-mum*Em-h*ITN*Em-j*IRS*Em;
58     Imdot = g*Em-mum*Im-alphas*Im-h*ITN*Im-j*IRS*Im;
59
60     %% Find fixed points — Solves the system of equations. The default is
61     %% setting the equations equal to zero
62     [Shstar, Ehstar, Ihstar, Rhstar, Smstar, Emstar, Imstar] = ...
63     solve(Shdot, Ehdot, Ihdot, Rhdot, Smdot, Emdot, Imdot);
64     W = vpa([Shstar, Ehstar, Ihstar, Rhstar, Smstar, Emstar, Imstar]);
65     % Assigns the correct values for later determining the fixed points.
66     % Based on how the solve function outputs data alphabetically
67     Shstar = W(:,6); Ehstar = W(:,1); Ihstar = W(:,3); Rhstar = W(:,5); ...
68     Smstar = W(:,7); Emstar = W(:,2); Imstar = W(:,4);
69
70     %% Creates fixedPoints matrix including all POSSIBLE fixed points not
71     %% just physically meaningful fixed points
72     fixedPoints = [];
73     for fp=1:length(Shstar) % uses length of Shstar, all have same length
74         % Note: weird 'z' value, unsure what it is. Could not find answer
75         % in the literature. Apparently equals 122
76         if double(findsym([Shstar(fp) Ehstar(fp) Ihstar(fp)...
77             Rhstar(fp) Smstar(fp) Emstar(fp) Imstar(fp)])) == 122
78             continue % Skips over the fixed points with that unknown 'z'
79         end
80         % Creates the fixedPoints matrix
81         fixedPoints(end + 1, :) = [Shstar(fp) Ehstar(fp) Ihstar(fp)...
82             Rhstar(fp) Smstar(fp) Emstar(fp) Imstar(fp)];
83     end
84     fixedPoints = double(fixedPoints); % To work with removing fixed points
85
86     delete = 0;
87     %% Goes backwards because rows will be removed and will cause an error
88     %% if they are removed from the top first
89     for i = size(fixedPoints,1):-1:1
90         for j = 1:size(fixedPoints,2)
91             % removes all negative and imaginary fixed points
92             if (fixedPoints(i,j) < 0 ) || (imag(fixedPoints(i,j)) ~=0)
93                 delete = 1;

```

```

94         end
95     end
96     if delete == 1
97         %% deletes the row/fixed point once it has cycled through
98         %% all the columns in a row
99         fixedPoints(i,:) = [];
100        delete = 0;
101    end
102 end
103
104 %% This section of code removes fixed points that are trivial, in other words,
105 %% either the human population OR the mosquito population goes to zero
106 deletel = 1;
107 delete2 = 1;
108 for i = size(fixedPoints,1):-1:1
109     for j = 1:4
110         fixedPoints(i,j);
111         if fixedPoints(i,j) > 0
112             deletel = 0;
113         end
114         deletel;
115     end
116     for j = 5:7
117         if fixedPoints(i,j) > 0
118             delete2 = 0;
119         end
120     end
121     %% fixed point has to have values in both Human AND mosquito
122     %% sections not to be deleted
123     if deletel == 1 || delete2 == 1
124         fixedPoints(i,:) = [];
125     end
126     deletel = 1;
127     delete2 = 1;
128 end
129
130 %% To help with evaluation of code to make sure all possible solutions are
131 %% determined. The rest of the code might assume only one fixed point in the
132 %% matrix. We wanted to know if there was more than one to help with this fact
133 if size(fixedPoints, 1) > 1
134     disp('FIXEDPOINTS MATRIX TOO LARGE')
135 end
136
137 %% Compute the Jacobian
138 J = jacobian([Shdot, Ehdot, Ihdot, Rhdot, Smdot, Emdot, Imdot],...
139             [Sh Eh Ih Rh Sm Em Im]);
140
141 for k=1:size(fixedPoints,1)
142     %% Evaluates the jacobian for the fixed point by substitution
143     Jeval = subs(J, [Sh Eh Ih Rh Sm Em Im], [fixedPoints(k,1), ...
144         fixedPoints(k,2), fixedPoints(k,3), fixedPoints(k,4), ...
145         fixedPoints(k,5), fixedPoints(k,6), fixedPoints(k,7)]);
146     eigens = eig(Jeval); % computest the eigenvalues
147     eigens = real(eigens); % replaces complex number with its real part
148     scriptR = max(eigens); % determines the dominant eigenvalue
149

```

```

150     % displays the values for ITN, IRS and the dominant eigenvalue
151     disp([num2str(overall), ':', ' ITN = ', num2str(ITN), ', IRS = ', ...
152           num2str(IRS), ' >>>> dominant eigenvalue: ', num2str(scriptR)])
153     % Useful in the saving process
154     IRStoSave(end + 1) = IRS;
155     ITNtoSave(end + 1) = ITN;
156     scriptRstoSave(end + 1) = scriptR;
157     ShtoSave(end + 1) = fixedPoints(k,1);
158     EhtoSave(end + 1) = fixedPoints(k,2);
159     IhtoSave(end + 1) = fixedPoints(k,3);
160     RhtoSave(end + 1) = fixedPoints(k,4);
161     SmttoSave(end + 1) = fixedPoints(k,5);
162     EmtoSave(end + 1) = fixedPoints(k,6);
163     ImtoSave(end + 1) = fixedPoints(k,7);
164     % Saves data to file for later use
165     save('Analysis.mat', 'IRStoSave', 'ITNtoSave', 'scriptRstoSave', 'ShtoSave', ...
166          'EhtoSave', 'IhtoSave', 'RhtoSave', 'SmttoSave', 'EmtoSave', 'ImtoSave')
167     end
168 end % End of analysis
169 toverall = toc(tInitial); % determines overall run time of code
170 averageTime = toverall/size(IRSITNmat,2) % calculates average run time

```

## B Next Generation Matrix Method - MATLAB Code

The following MATLAB code runs the next generation matrix method. A description of this analysis is given in Section 3. Note that this method can be used for all of the simulations or to just focus on a set of parameter values, because of its fast run time.

```

1  %% This code runs the Next Generation Matrix Method for finding the
2  %% Basic Reproductive number
3
4  clear all
5  format long
6  %%% The following matrix will run all (121) combinations
7  IRSITNmat = [zeros(1,11) ones(1,11)*.1 ones(1,11)*.2 ones(1,11)*.3 ...
8              ones(1,11)*.4 ones(1,11)*.5 ones(1,11)*.6 ones(1,11)*.7 ones(1,11)*.8 ...
9              ones(1,11)*.9 ones(1,11);0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1 ...
10             0:.1:1 0:.1:1 0:.1:1 0:.1:1 0:.1:1];
11 Rnaught2tosave = []; ITNtoSave = []; IRStoSave = []; %% Used for saving
12
13 for overall = 1:size(IRSITNmat,2)
14     % Clears all variable except the save matrices
15     clearvars -except overall IRSITNmat Rnaught2tosave ITNtoSave IRStoSave
16     % Define functions
17     syms Sh Eh Ih Rh Sm Em Im Nm Nh
18     % Constants
19     a = .25;
20     b = .04;
21     c = 1/14;
22     d = .205;
23     f = .83*.5;
24     g = 1/12;
25     h = 1/365;

```

```

26     j = 1/365;
27     Lambdah = .028;
28     muh = .0000391;
29     alphah = .0004;
30     Lambdam = 6;
31     mum = .04;
32     alphas = .01;
33     % Changes ITN and IRS value with each run of the for loop
34     ITN = IRSITNmat(1,overall); % ITN values are stored in the first row
35     IRS = IRSITNmat(2,overall); % IRS values are stores in the second row
36
37     %% Full Differential Equations
38     Shdot = -a*Im*Sh*(1-ITN)/(Sh+Eh+Ih+Rh)+b*Rh+Lambdah-muh*Sh;
39     Ehdot = a*Im*Sh*(1-ITN)/(Sh+Eh+Ih+Rh)-c*Eh-muh*Eh;
40     Ihdot = c*Eh-d*Ih-muh*Ih-alphah*Ih;
41     Rhdot = d*Ih-b*Rh-muh*Rh;
42     Smdot = -f*Ih*Sm*(1-ITN)/(Sm+Em+Im)+Lambdam-mum*Sm-h*ITN*Sm-j*IRS*Sm;
43     Emdot = f*Ih*Sm*(1-ITN)/(Sm+Em+Im)-g*Em-mum*Em-h*ITN*Em-j*IRS*Em;
44     Imdot = g*Em-mum*Im-alphas*Im-h*ITN*Im-j*IRS*Im;
45
46     %% "squiggle" equations are for the F matrix (see paper, section 3)
47     Ehsquiggle = a*Im*Sh*(1-ITN)/(Sh+Eh+Ih+Rh);
48     Ihsquiggle = 0;
49     Emsquiggle = f*Ih*Sm*(1-ITN)/(Sm+Em+Im);
50     Imsquiggle = 0;
51     F = jacobian([Ehsquiggle Ihsquiggle Emsquiggle Imsquiggle], [Eh Ih Em Im]);
52
53     %% "hat" equations are for the V matrix (see paper, section 3)
54     Ehhat = c*Eh+muh*Eh;
55     Ihhat = -c*Eh+d*Ih+muh*Ih+alphah*Ih;
56     Emhat = g*Em+mum*Em+h*ITN*Em+j*IRS*Em;
57     Imhat = -g*Em+mum*Im+alphas*Im+h*ITN*Im+j*IRS*Im;
58     V = jacobian([Ehhat Ihhat Emhat Imhat],[Eh Ih Em Im]);
59
60     G = F*V^(-1); %% The next generation matrix
61     G2 = subs(G, [Sh Eh Ih Rh Sm Em Im], [Nh, 0, 0, 0, Nm, 0, 0]);
62     %% Determines the dominant eigenvalue of the next generation matrix
63     eigenvalue = max(double(eig(G2)));
64     disp([num2str(overall), ':', ' ITN = ', num2str(ITN), ', IRS = ', ...
65         num2str(IRS), ' >>>> dominant eigenvalue: ', num2str(eigenvalue)])
66     Rnaught2tosave(end + 1) = eigenvalue; ITNtoSave(end + 1) = ITN; ...
67     IRStoSave(end + 1) = IRS;
68 end
69 %% Saves the Data in a .mat file
70 save('RnaughtMethod2_Data121.mat', 'ITNtoSave', 'IRStoSave', 'Rnaught2tosave')

```

## References

- [1] Shiflet A.B. and Shiflet G.W. *Introduction to Computational Science: Modeling and Simulation for the Sciences*. Princeton University Press, 2006.
- [2] Pluess B., Tanser F.C., Lengeler C., and Sharp B.L. Indoor residual spraying for preventing malaria (review). *The Cochrane Library*, Issue 4, 2010.
- [3] Chiyaka C., Tchuente J.M., Garira W., and Dube S. A mathematical analysis of the effects of control strategies on the transmission dynamics of malaria. *Appl. Math. Comput.*, 195(2):641 – 662, 2008.
- [4] Chiyaka C., Garira W., and Dube S. Transmission model of endemic human malaria in a partially immune population. *Math. Comput. Modelling*, 46(5-6):806 – 822, 2007.
- [5] Chiyaka C., Garira W., and Dube S. Effects of treatment and drug resistance on the transmission dynamics of malaria in endemic areas. *Theoretical Population Biology*, 75(1):14 – 29, 2009.
- [6] Lengeler C. Insecticide-treated bed nets and curtains for preventing malaria (review). *The Cochrane Library*, Issue 2, 2009.
- [7] Centers for Disease Control and Prevention. CDC - Malaria (accessed August 8, 2011) <http://www.cdc.gov/MALARIA>.
- [8] Chowell G., Hengartner N.W., Castillo-Chavez C., Fenimore P.W., and Hyman J.M. The basic reproductive number of ebola and the effects of public health measures: the cases of congo and uganda. *J. Theoret. Biol.*, 229(1):119 – 126, 2004.
- [9] Ferguson H.M. and Read A.F. Why is the effect of malaria parasites on mosquito survival still unresolved? *Trends in Parasitology*, 18(6):256 – 261, 2002.
- [10] Jones J.H. Notes on  $R_0$ . online handout, May 1 2007. [jhj1@stanford.edu](mailto:jhj1@stanford.edu).
- [11] Esteva L., Gumel A.B., and de Len C.V. Qualitative study of transmission dynamics of drug-resistant malaria. *Math. Comput. Modelling*, 50(3-4):611 – 630, 2009.
- [12] World Health Organization. WHO Malaria (accessed August 8, 2011) <http://www.who.int/topics/malaria/en/>.
- [13] Sokolow S.H., Foley P., Foley J.E., Hastings A., and Richardson L.L. Disease dynamics in marine metapopulations: Modelling infectious diseases on coral reefs. *Journal of Applied Ecology*, 46(3):pp. 621–631, 2009.
- [14] Strogatz S.H. *Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering*. Studies in Nonlinearity. Westview Press, 1994.
- [15] Tassier T. SIR model of epidemics, 2005.