MODEL-BASED ANALYSIS OF TWO FIGHTING WORMS

A Thesis submitted to Kent State
University in partial fulfillment of the
requirements for the Degree of Master of
Science

By

Zakiya M. Tamimi

August, 2004

Thesis written by

Zakiya M. Tamimi

B.S., Palestine Polytechnic University, 1999

M.S., Kent State University, 2004

Approved by

___________________________________ , Advisor

___________________________________ , Chair, Department of Computer Science

___________________________________ , Dean, College of Arts and Sciences

ABSTRACT

Self-replicating malicious codes (worms) are striking the Internet vigorously. A particularly interesting trend is the emergence of mutually agnostic virus. In this work we study such combating virus system. Like other virus models, we also present an adapted version of Lotka-Volterra equation model to show the interaction dynamics of mutually antagonistic self-spreading codes. There are few novel findings of this enhanced model. Such as the prediction of oscillatory behavior of interacting worms population conforming to existing biological systems.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The one person who I can never thank enough for making this work possible is my advisor Dr Javed Khan, who is the best to guide, advise, encourage, and criticize. I also want to thank the defense committee members Dr Hassan Peyravi, and Dr Kazim Khan, who provided quite helpful advices on mathematical issues.

I am very grateful to Dr Chuck Gartland who checked the mathematical correctness of our model and provided helpful advices regarding the numerical solutions. My thanks go also to my mathematician friend Jeffery Anderson, M.S. who always offered helpful suggestions and solutions.

Chapter 1

INTRODUCTION

Computer viruses are increasingly becoming a major source of productivity drain for internet operations. While the potency of most viruses varies at the mercy of the virus writers, but even the relatively harmless viruses are becoming a major productivity drain because of their increasingly sophisticated spreading mechanism. When tens of millions of computers are affected even a simple reboot can bring part of the internet to its knee. On top of these the incessant adaptive mutation launched by their writers are also evident now.

A particularly sophisticated recent introduction is the killer worm (also called counter-worm, predator worm, or good will mobile code). This is a new phenomenon that has made headlines recently. These worms are out there fighting malicious codes (Code-Red, MS-Blast, and Sasser) spread by rival virus writer groups. Besides malicious combating worms already there is the concept of virus like predator codes.

There is an interesting digital culture which is helping the emergence of these predator worms. For example, one worm's authors fight another group to expand their peer-to-peer networks, which are formed by Trojan horses. Later, these P2P networks are used to launch new worm, generate Denial of Service attack, or circulate spam anonymously. In addition, some predator worms scavenge on other worms as means to spread their code. For example, the predator worm spread through a flaw or backdoor of another worm. In addition, predator worms can have non-malicious applications. They could the necessary proactive countermeasure to fight zero-day worms. Some researchers have already defined predators as “good will” mobile codes that go after malicious worms and kill them.

The goal of this research is to mathematically model the behavior of combating worms. This thesis models prey-predator dynamics under several interesting combat scenarios. This includes the presence of antivirus software, effect of security patches, and combating worm’s capability to block or use each others backdoors. Following the steps of previous researchers, we use Lotka-Volterra equations and expand it for our models. We present the detail model equations, and corresponding scenario analysis based on their numerical solutions.

1.1. Related work

While modeling worms is not totally new, there’s only very few in literature about killer virus (predator worm). Two papers are in the same line as our work. Toyoizumi and Kara used Lotka-Voltera equations to model and analyze the interaction between predator worms and traditional worms [1]. They define predators as “good will mobile codes” that kill malicious viruses. Also, they discuss how to minimize predators main size effect (their number) without losing their effectiveness. Nicol and Lilijenstam define active defenses, as techniques that “take the battle to the worm” [4]. They model four active defenses, two of them “nullifying and “sniper” are predator worms. They use the number of protected hosts, total consumed bandwidth, and peak scanning rate the network as metrics of effectiveness.

Staniford was the first to attempt to model random scanning Internet worms [5]. His model is a quantitative theory that explains Code-Red spread. The theoretical data generated by his equation fairly matched with available Code-Red data. Later Zou et al provided an enhanced model of Code-Red that considers the effects worm countermeasures and routers congestion [6]. They base their “two-factor worm model” on Kermack-Mckendrick. Their simulations and numerical solutions are better match Code-Red data.

The thesis is organized in the following way. First we present the basics of our model. We present a classification of potential combating worms, and the description of the base environment. The Chapter 3 presents the simplest of the scenarios. In Chapter 4 we study the effect of anti-virus- which we call harvesting effect on the combat. Chapter 5 contains general analysis of patching prey worm against predator worm. Chapter 6 introduces the effect of employing system patches on the prey-predator system. Finally we conclude and propose future work.

Chapter 2

MODEL BASIS

2.1. Virus Types

Although the terminologies have not been firmly established in literature here we will consider Virus to be a super set of self-replicating malicious codes. *Worm* a subset of viruses that are network aware (use network protocols and parameters to spread). Worms can be fully automated (port scanning) or partially human-dependent like (email) or can be combination of different attack techniques.

Traditional ways to defend against worms are called *defensive techniques* (countermeasures) based on prevent, detect, clean. *Antivirus* programs are all traditional software programs that scan for, detect, and clean worm infections. *System patches* are software made available by operating system authors that can fix existing security hole of bug. A system patch does not remove a virus but will block the spreading mechanism of a new virus or even an existing virus.

Some worms can attack already infected machine, clean existing worm, and takeover that machine. We refer to such killing (or cleaning) worm as *predator worm*, e.g. Code-Green, Welchi, Netsky. On the other hand, *prey worms* are the victims of predator worms, e.g. MS-Blast, Bagle, and Sasser. Worms that attack other worms are predators. Figure 2.1 explains the classification. Internet worms are primarily classified according to prey and predator role.

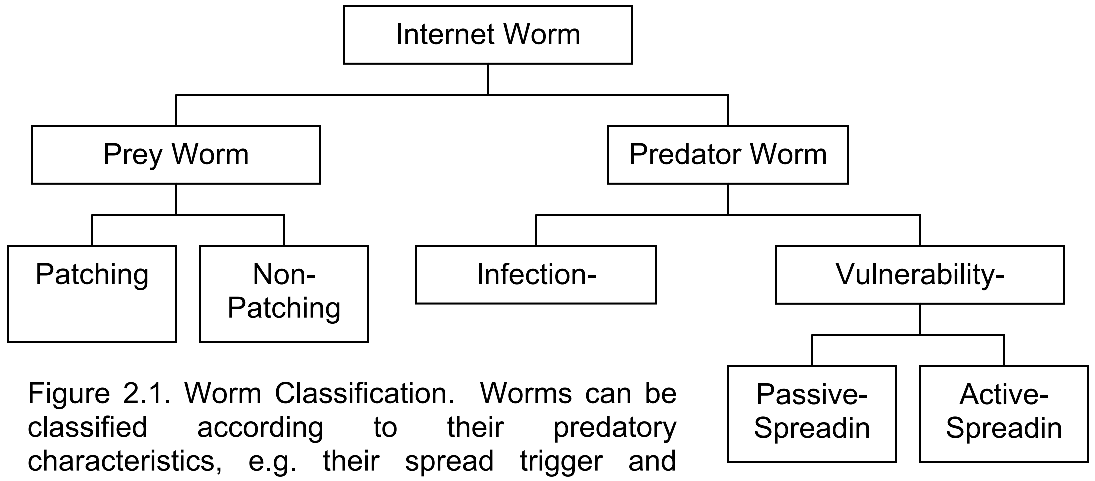


Figure 2.1. Worm Classification. Worms can be classified according to their predatory characteristics, e.g. their spread trigger and scanning technique.

Most predators spread by finding and attacking already infected machine. However, other predators, in addition, can find and attack clean machines. An *infection-driven* predator, exclusively, attack infected machines. *Vulnerability-driven* predator, on the other hand, can attack both clean and infected machines. A predator worm can find victim prey worms by actively propping for prey-infected machines (called *active-spreading*). A prey worm can "fall into predator trap" if it unknowingly scans a predator-infected machine. A predator that depends on prey to scanning, is called *passive-spreading* predator, e.g. CR-Clean. Figure 2.1 shows that predator worms are first classified based on their spreading trigger-infection driven and vulnerability driven. The later is further classified based on their scanning technique-passive and active spreading.

As shown in Figure 2.1, prey worms can be patching or non-patching. Prey worms may protect themselves from their predators by closing the security hole through which they penetrated, thus preventing predator from getting in. We call such prey worms a *patching worm* otherwise they are *non-patching* prey worms.

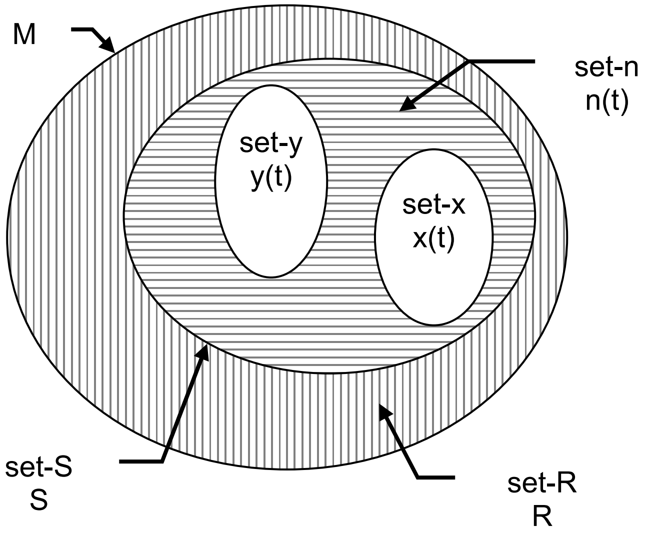
2.2. Environment

We assume The Internet size as fixed number of machines M . Two reasons for this assumption: small infection period, and huge Internet size. Internet worms spread in minutes/hours, while a major change take much longer. The day and night time effect is compensated for by the wide spread of the Internet around the world. The other reason is the huge size of the Internet, that even adding or removing thousands of machines at once doesn't have significant effect of ten of millions total number. For a specific worm, one machine is either susceptible to that infection (called *vulnerable*), or immune (called *removed*). Vulnerable machines can be penetrated by a worm, and once infected they spread the infection on their own. Removed machines cannot be infected by a worm for some reason, such as: being patched, having platform, behind firewall, etc.

Usually, there are many worms spreading over the Internet at the same time. This research is concerned with the interaction of two prey and predator worms, and thus we ignore all other existing worms. We also assume that set of vulnerable machines (set-S) for the prey worm is the same for the predator worm. Same applies to the removed machines set (set-R). The vulnerable and removed machines sets are complements of each others. If number of vulnerable machines is S, and number of vulnerable ones is R, $S + R = M$, the total number of machines. Figure 2.2 shows the two main sets set-S and set-R. Usually, vulnerable and removed machines don't switch back and forth. However, in cases when system patch is used, a vulnerable machine can become removed.

A vulnerable machine that is infected by a worm is called *infectious*. All other vulnerable machines that are not compromised fall in the *clean* machines set (set-n) of size $n(t)$. Machines can change state from clean to infectious, or infectious to clean. An infectious machine is infected one worm: prey or predator worm. We assume that no machine has multiple worm infection. The reason for this are, when a prey meets a predator worms, the predator would takeover the prey worm, leaving only the predators infection. Infectious machines that are infected by a prey worm (worm-x) are called set-x, which has cardinality $x(t)$. Infectious machines that are infected by predator worm (worm-y) fall in set-y with size $y(t)$. Figure 2.2 shows the two infectious sets and their relation to the clean set. Machines in set-x can change state and move into set-y. The cardinalities of set-n, set-x, and set-y, are variable functions of time, where the total sum $S = n(t) + x(t) + y(t)$ is the size of vulnerable machines set.

Figure 2.2 Machines Sets
Internet: M
Vulnerable machines: S
Removed machines: R
M=S+R
Clean machines: n(t)
Prey (Infectious): x(t)
Predator (Infectous: y(t)
S=n(t)+x(t)+y(t)



Chapter 3

PREY, PREDATOR MODEL

Two combating worms: worm-x and worm-y, spread over a network. Worm-x is a traditional prey worm, which spreads by infecting clean machines and has no patching feature, e.g. Slammer. Worm-y is a predator worm that can only takeover worm-x infected machines; thus considered infection-driven. Two possible variations of worm-y are: passive-spreading and active-spreading. Set-x is the set of all worm-x members and has cardinality of $x(t)$ at any time t. Set-y, of size $y(t)$, contains all instances of worm-y.

The number of clean machines that are infected, by worm-x, at any time is dependent on both available clean and worm-x infected machines. This infection increase rate (clean-set) decease rate) equals axn, where a a positive fraction. Encounters between worm-x and worm-y result in increase in set-y (decreases in set-x) which is proportional to number of available worm-x infected and worm-y infected machines. This change rate equals bxy, where b is a positive number. Figure 3.1 describes the transition between sets in this scenario. The link signifies the transfer rate of members from one set to the other. The Value on the arrow is the transition rate.

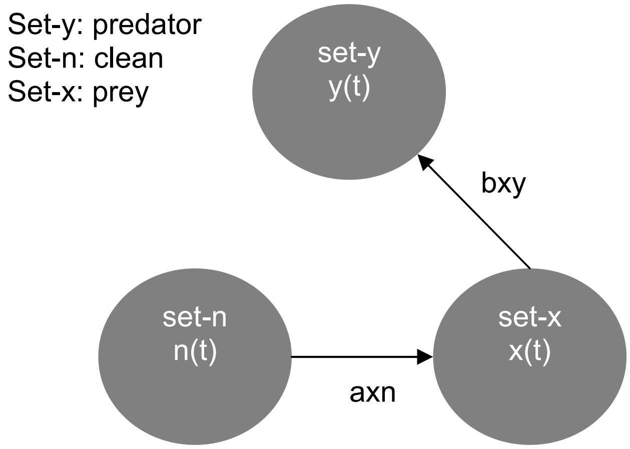


Figure 3.1. Machines transition diagram for prey-predator scenario. The circles represent machines' sets and arrows are transitions' direction and rate.

The infection rate of worm-x is the first derivative of $x(t)$. The same applies to worm-y infection rate, and clean machines change rate. The dynamic of the system can be described by equations 3.1, 3.2, 3.3, and 3.4.

$$\frac{dx}{dt} = axn - bxy \tag{3.1}$$

$$\frac{dy}{dt} = bxy \tag{3.2}$$

$$\frac{dn}{dt} = -axn \tag{3.3}$$

$$x(0) = x_0, y(0) = y_0, n(0) = n_0 \qquad (3.4)$$

The value of parameters a and b are dependent on worms scanning rate and network size. Following, we discuss the value derivation of a and b for two predator variations: passive-spreading and active-spreading.

3.1. Parameters Computations

Passive-Spreading Predator. Assume that worm-y is passive-spreading predator; thus does no scanning on its own to find infected machines. Let worm-x *scanning rate* be r, where r is the number of unique scans generated by the worm per a unit of time. The total number of scans by all members in set-x then is rx. Since R+x(t)+y(t)+n(t)=M, the value of rx is the sum of all scans by worm-y of all machine sets, as in equation 3.5

$$rx = \frac{rxR + rx^2 + rxy + rxn}{M} \qquad (3.5)$$

If each time that worm-x scans a clean machine results in a new infection, then parameter a is given by equation 3.6

$$a = \frac{r}{M} \qquad (3.6)$$

Likewise, if every encounter between y-worm and worm-x infected machine results in takeover by worm-y, then parameter b is given by equation 3.7

$$b = \frac{r}{M} \qquad (3.7)$$

Active-Spreading Predator. Active-spreading worm-y does its own scanning in order to find worm-x infected machines. Assuming worm-y has scanning rate be v, the total number of scanning by members in set-y is vy satisfies equation 3.8

$$vy = \frac{vyR + vyx + vy^2 + vyn}{M} \qquad (3.8)$$

The encounters between worm-x and worm-y are result of scans by both. Thus, parameter b can be described by equation 3.9

$$b = \frac{v + r}{M} \qquad (3.9)$$

Other Factors. We assumed that an encounter between x-infected and y-infected machine will always result in worm y taking over. Practically, worm-y can have some success probability g. That case the new value of b will be computed as gb. This applies in the same way to parameter a. If worm-x has h percent of its scans successful, the new value of a would be ha.

3.1. Analysis

The following solution curves were obtained using Maple numerical solution.

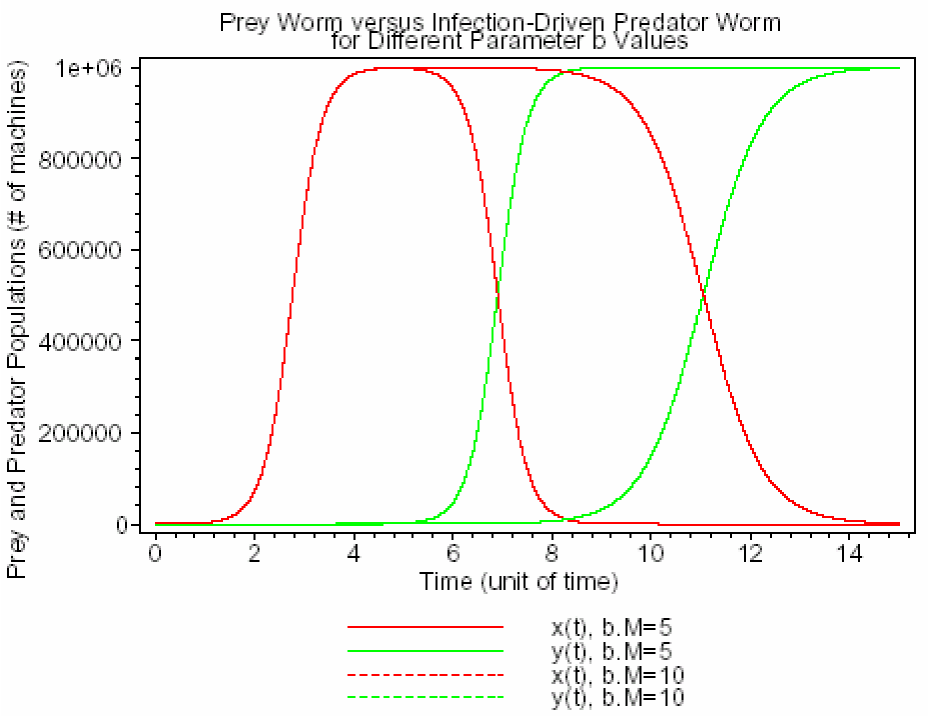


Figure 3.2. a.M=10, M=3000000, x0=100, y0=1, n0=1000000

In figure 3.2 and 3.3, two curves are plotted $x(t)$ in red and $y(t)$ in green for different values of $a:b$. The general behavior described here shows that initially worm-x increase exponentially as it would without worm-y existence. Worm-y increase proportional to worm-x increase (food of worm-y). The increase in worm-y population results in decrease in worm-x population (as worm-y eats worm-x). The $x(t)$ curve stops increasing (where it hits it maximum) and starts declining. Curve $y(t)$ continues to increase until it uses up all available worm-x members, where it hits its maximum and freeze thereafter. The system reaches steady state when both infection rates are zero. This occurs when all worm-x infected machines are re-infected by worm-y.

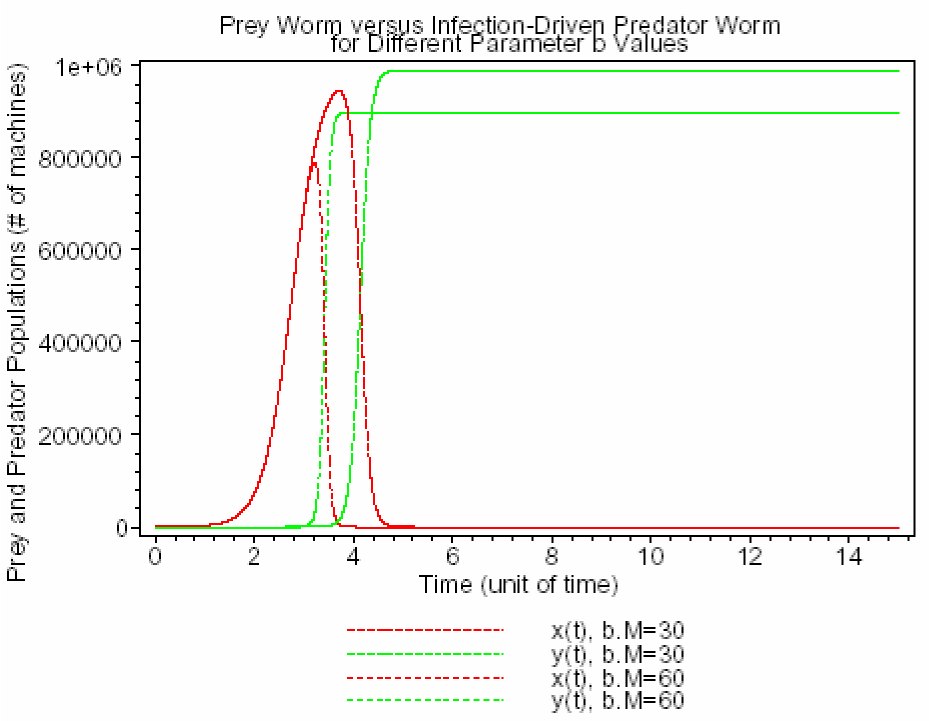


Figure 3.3. a.M=10, M=3000000, x0=100, y0=1, n0=1000000

Breakout Condition. By studying figure 3.2, we can observe that $\max(x) = \max(y) = S$. We name this condition as *Prey-breakout condition* since it occurs as result of faster growth in prey population than predator population ($b \leq a$).The time when breakout is taking place is called *breakout interval.* If $k = a - b$ and if $k \geq 0$, than increasing k will increase the breakout interval. It's important to note that whenever there's prey-breakout, there's predator breakout too.

Contamination Condition. In figure 3.3 $\max(x) \le \max(y)$. This condition is called *prey-contamination condition*, which occurs when the predator population is faster than the prey ($b > a$). This condition is the opposite of prey-breakout condition. However, predator population breakout could happen in parallel with prey-contamination.

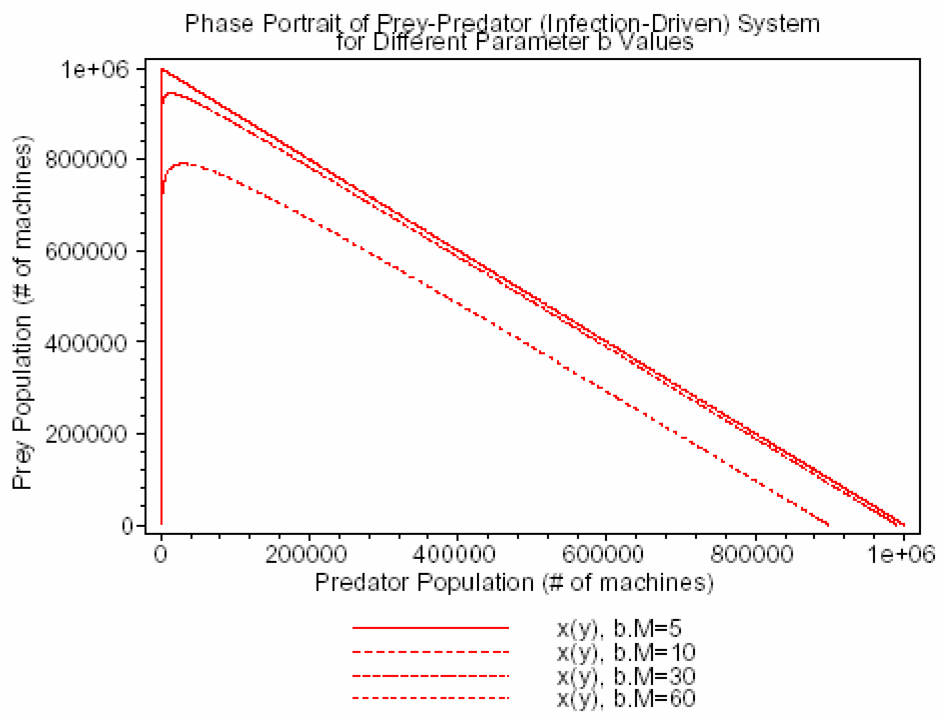


Figure 3.4. a.M=10, M=3000000, x0=100, y0=1, n0=1000000.

Figure 3.4 is plot of $x(y)$ for different values of the ratio $a:b$. Increasing b value results in increasing $y(t)$ which means more effective worm-y that can catch worm-x even faster. The larger is the maximum of $x(t)$, the larger $y(t)$ maximum needed to stabilize the system. Thus, larger b result in smaller maximum for $x(t)$, $y(t)$, and $x(y)$.

3.2. Initial Population

Initial population is the number of worm-x infected machines at time zero $x(0)$, where *time-zero* is the point when worm-y is launched. *Response time* is the time between worm-x start and worm-y start, or the time duration when worm-x and worm-y didn't (before time zero). It is clean then that response time and initial population are direct linear computations of each others, thus can be thought of as one thing.

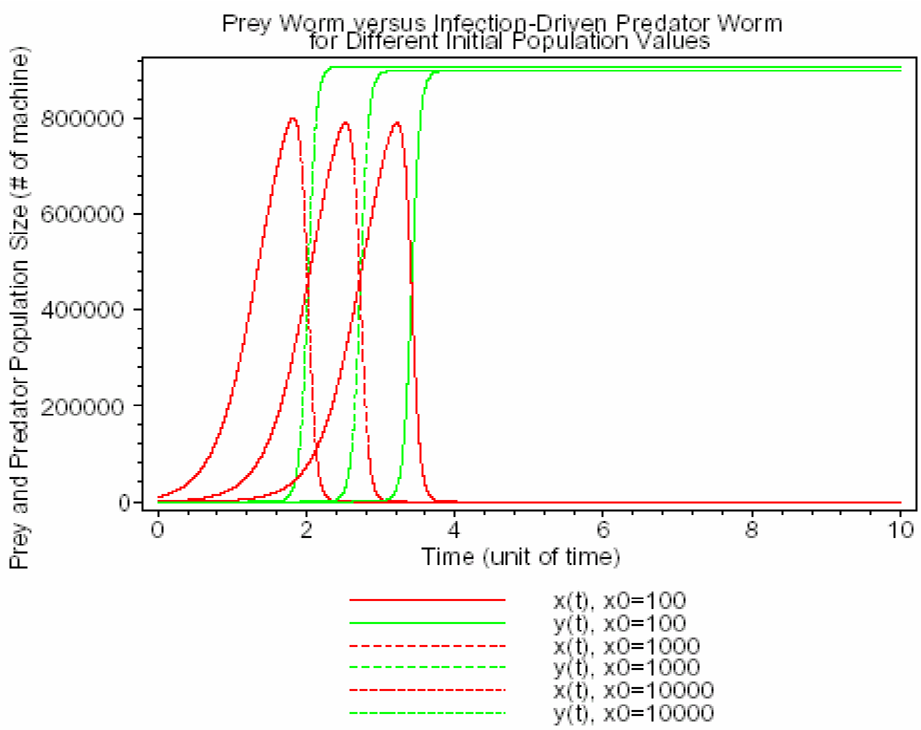


Figure 3.5. a.M=10, b.M=60, M=3000000, y_0=1, n_0=1000000

Figure 3.5 is a plot of $x(t)$ and $y(t)$ for different initial population values. The graph explains the small effect of $x(0)$ on maximum values; even increasing $x(0)$ exponentially increases the maximum linearly. The phase portrait of the two curves is shown in figure 3.6. It's clear that changing the initial population causes almost no difference on the resulting populations.

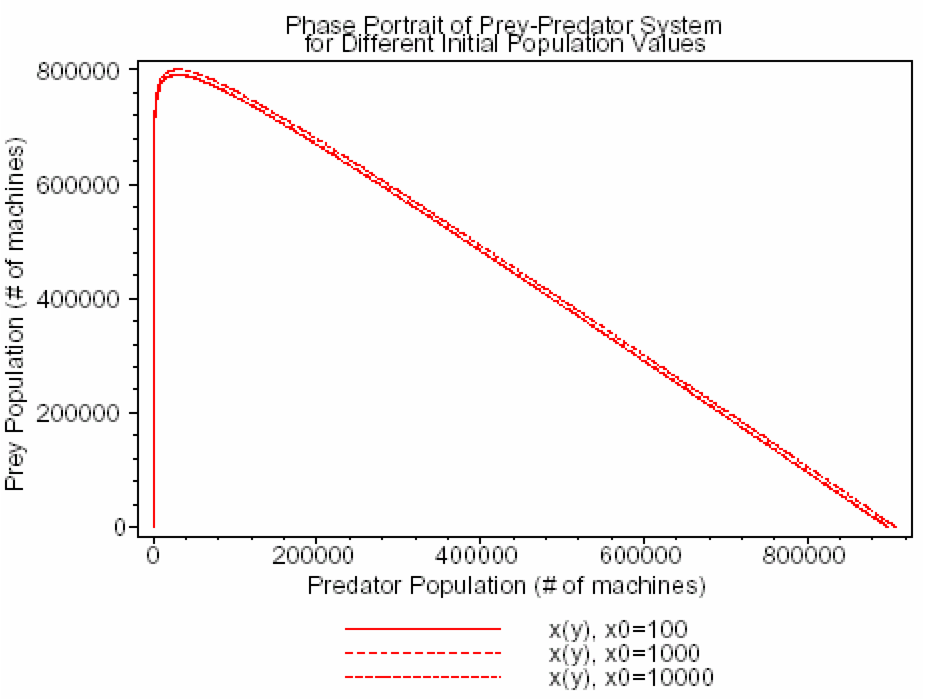


Figure 3.6. a.M=10, b.M=60, M=3000000, y_0=1, n_0=1000000

3.3. Case Expansion

We expand this case by considering vulnerability-driven type of predator. So, worm-y infects both clean and worm-x infected machines. Figure 3.7 describes the transitions between the machines sets. Worm-x increases as the basic scenario, by infecting clean machines at rate axn. Worm-y increases by targeting clean machines at rate cyn (c is positive), and clean machines decreases at the same rate. The system dynamics can be described in equations 3.10, 3.11, 3.12 and 3.13.

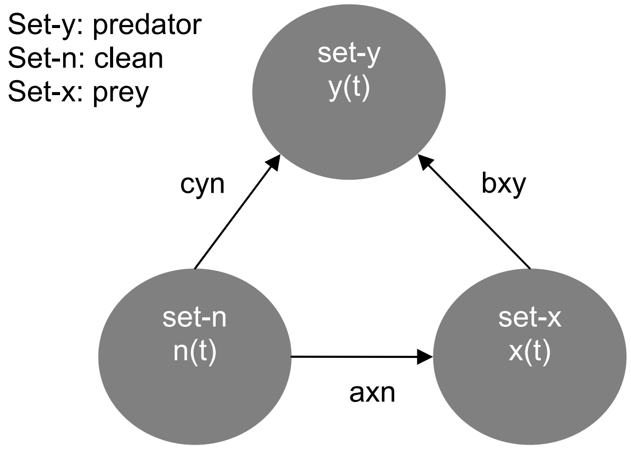


Figure 3.7. Machines transition diagram for prey-predator expansion scenario. The circles represent machines' sets and arrows are transitions' direction and rate.

$$\frac{dx}{dt} = axn - bxy \tag{3.10}$$

$$\frac{dy}{dt} = cyn + bxy \tag{3.11}$$

$$\frac{dn}{dt} = -axn - cyn \tag{3.12}$$

$$x(0) = x_0, y(0) = y_0, n(0) = n_0 \tag{3.13}$$

Parameters a, b and c are dependent on worms scanning rate and network size. Parameter a and b values can be derived as before in the basic scenario. The value of c is dependent of worm-y scanning rate and is described in equation 3.14

$$c = \frac{v}{M} \tag{3.14}$$

When predator is infection-driven ($c = 0$) prey and predator are totally coupled in a closed loop. Increasing the prey population increases the predator population. Increasing the predator population decreases prey population. Decreasing prey population decreases predator population. Finally decreasing predator population increases the prey population.

In case of vulnerability-driven predator ($c > 0$) the predator has more than one option to spread. The prey-predator loop is not totally closed. Increasing prey population increases the predator population. Increasing the predator population decreases prey population. Decreasing prey population does NOT decrease predator population.

Figure 3.8 shows the plot of $n(t)$, $x(t)$ and $y(t)$. Compared with figure 3.2 and 3.3, the behavior is similar with two exceptions: the prey-breakout condition didn't take place, while $y(t)$ reaches the maximum environment capacity. Figure 3.8 shows that prey-contamination meanwhile it shows a predator-breakout increase condition.

Breakout Condition. Observing figure 3.8, we notice that $\max(y) = S$, which we call as *predator-breakout condition*. This occurs because predator growth is not limited by prey population but environment capacity ($c > 0$).

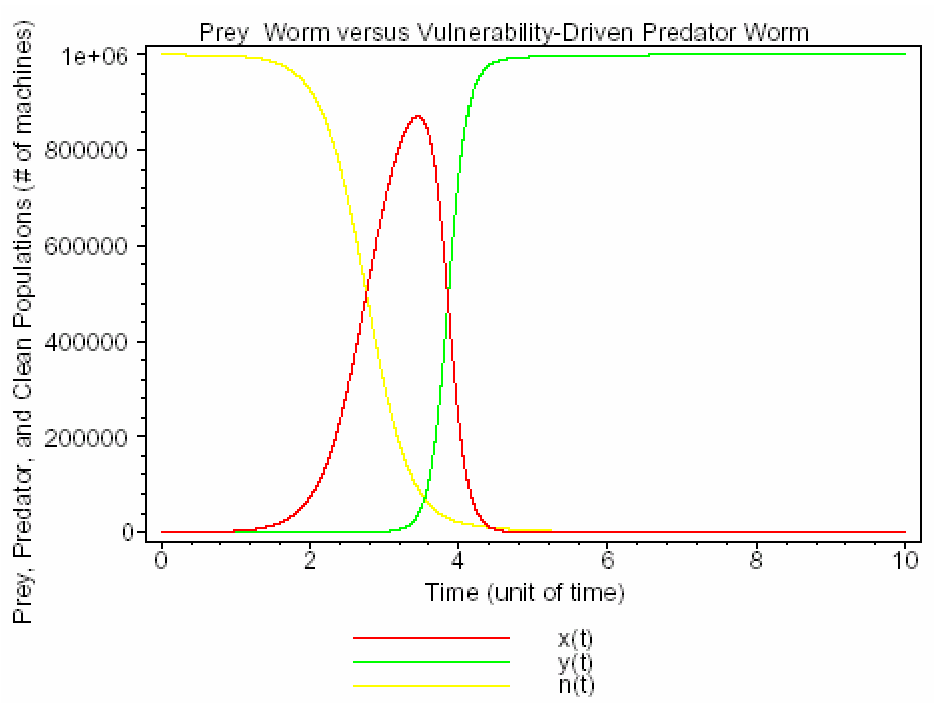


Figure 3.8. a.M=10, b.M=25, c.M=5, M=3000000, x_0=100, y_0=1, n_0=1000000

Chapter 4

PREY, PREDADTOR, AND ANTIVIRUS MODEL

Worm-x and worm-y are prey and predator worms that are competing over an environment. Worm-y is vulnerability-driven predator. Some machines on the network run antivirus software that can detect and clean both worms' infections. This scenario is analogous to harvesting (spraying, or fishing) phenomena in biological systems, where some third-party eliminates members of both combating populations. We assume that as people become more aware of an epidemic, they download, install, and update antivirus software at increasing rate.

We define the number of machines with antivirus as an increasing function of time. The functions $z_x(t)$ and $z_y(t)$ are the ratios of worm-x and worm-y, respectively, infected machines that are cleaned by the antivirus software. For example, $yz_y(t)$ is the number of worm-y infected machines that become clean of infection at time t. We define $z_x(t)$ and $z_y(t)$ in equations 4.1 and 4.2. The constants d_x and d_y are fraction numbers that determines the antivirus capacity to clean worm-x and worm-y, respectively. If d_x is 0.5 and d_y is 0.3, the antivirus can clean 50% of worm-x machines, and 30% of worm-y on the long run.

$$z_y(t) = d_y t/(t+1) \tag{4.1}$$

$$z_x(t) = d_x t/(t+1) \tag{4.2}$$

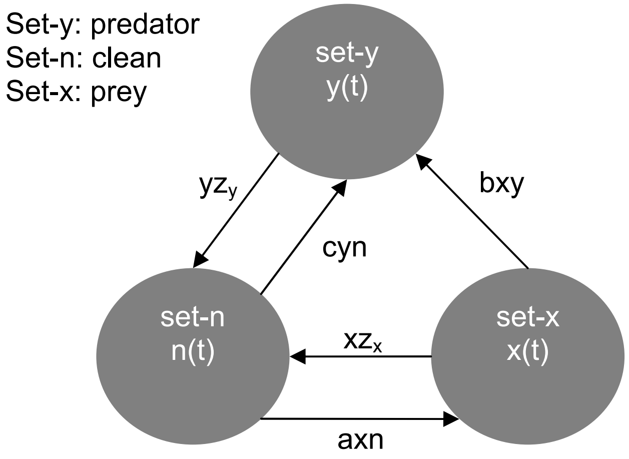


Figure 4.1. Machines transition diagram for prey-predator and antivirus scenario. The circles represent machines' sets and arrows are transitions' direction and rate.

Figure 4.1 describes the transition of members between machines' sets as a result of the two worms and antivirus reactions. Worm-x increase on count of clean machines set (set-n) at rate axn. Meanwhile, set-n gains worm-x machines back at rate xz_x, once cleaned by the an antivirus. On the other hand, worm-y increase on count of both clean and worm-x machines at rate $cyn + bxy$. In contrary of all previous scenarios, set-y decreases at rate $yz_y(t)$, as result of antivirus effect. The system behavior is described by equations 4.3, 4.4, 4.5, and 4.6

$$\frac{dx}{dt} = axn - bxy - xz_x \tag{4.3}$$

$$\frac{dy}{dt} = cyn + bxy - yz_y \tag{4.4}$$

$$\frac{dn}{dt} = -axn - cyn + xz_x + yz_y \tag{4.5}$$

$$x(0) = x_0, y(0) = y_0, n(0) = n_0 \tag{4.6}$$

4.1. Analysis

In this subsection, we assume that both worm-x and worm-y are affected equally by the antivirus effect. Thus $z_y = z_x = z$.

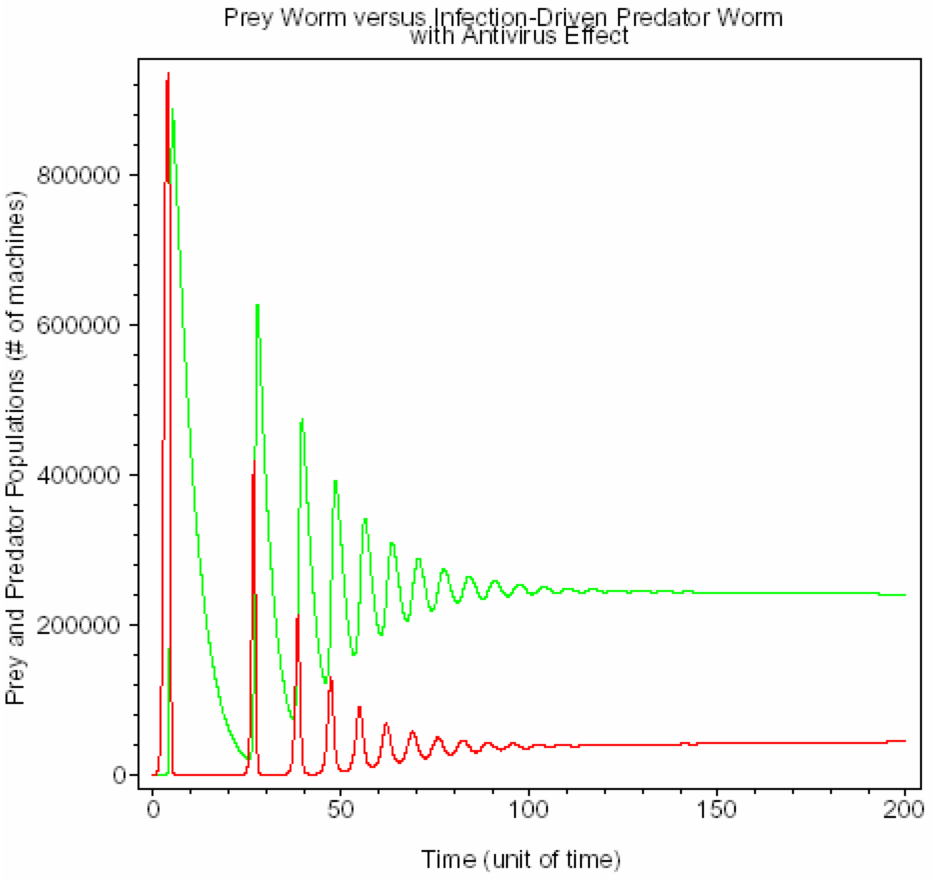


Figure 4.2. c.M=0 (Infection-Driven). a.M=10, b.M=25, d=0.4, x_0=100, y0=1, n_0=1000000, M=3000000

Figure 4.2 shows a new type of behavior, both curves $x(t)$ and $y(t)$ oscillate for a while as they gradually become constant lines. This phenomenon is a result of introducing the antivirus effect, which kills predators as will as prey infections. Originally, the increase in predators population causes degrade in prey population, and this is what is initially happening in this case. However, as the antivirus cleans some predator infections causing its population to drop, more prey infections will have chance to survive, and thus prey population increases again. Increasing prey population results in increasing predator population. However the second peek is lower than the first once since the antivirus is continuously reducing both populations. This periodical behavior repeats itself each time with lower maximum values. The oscillation turns into straight lines with some vibration, which eventually diminishes, resulting into two constant lines. At this stage the system reaches its steady state or equilibrium point, where the change rate is zero for all variables (increases and decreases in both populations cancel each others)

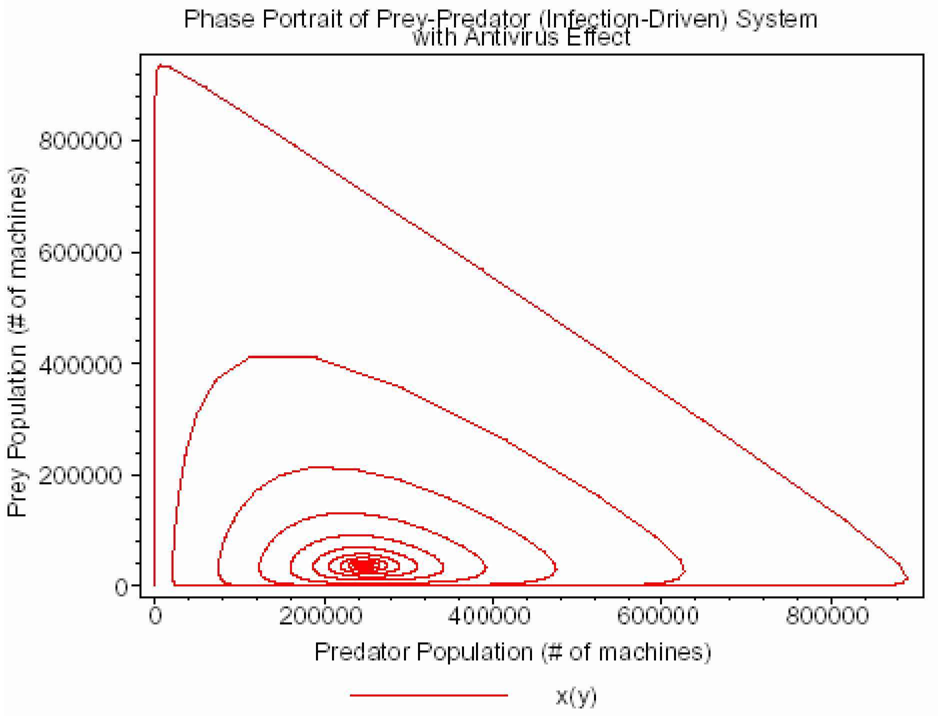


Figure 4.3. c.M=0 (Infection-Driven). a.M=10, b.M=25, d=0.4, x_0=100, y0=1, n_0=1000000, M=3000000

Figure 4.3 shows the system's oscillation behavior in a different way. The $x(y)$ curve, winds inwards until it reaches the equilibrium point, which is not zero. In other words, both infections will continue to exist forever.

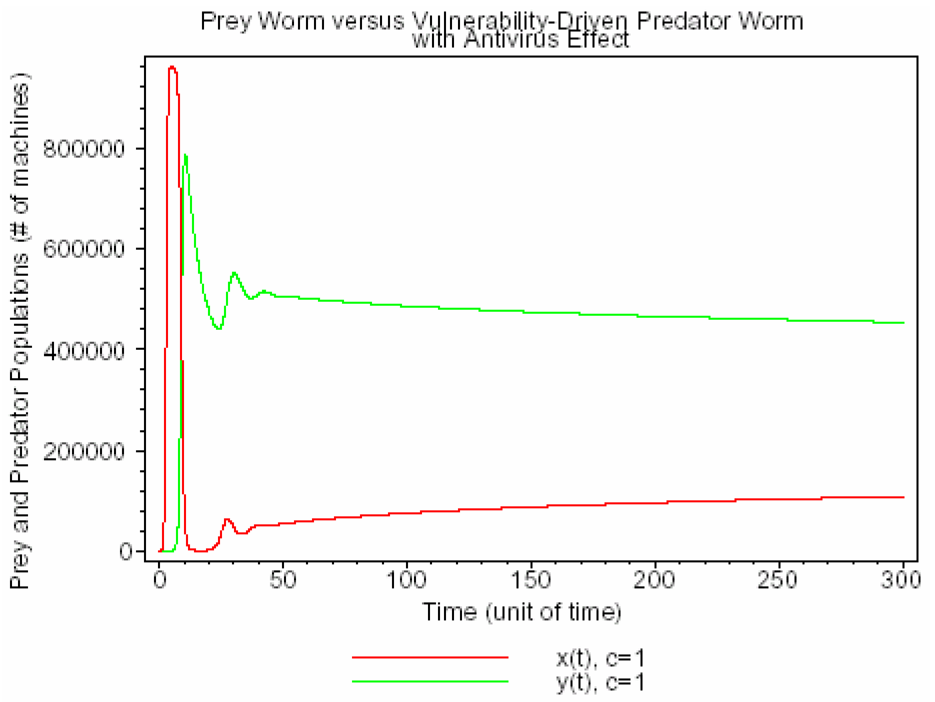


Figure 4.4. a.M=10, b.M=7, d=0.4, x_0=100, y_0=1, n_0=1000000, M=3000000

In the both figure 4.2 and 4.3, we ignored the effect of parameter c. Now we study the effect of parameter c using figure 4.4 and 4.5. Both graphs have the same values, except that in the $c.M = 1$ in the first and $c.M = 2$ in the other. It's noticeable that increasing the value of c reduces oscillation.

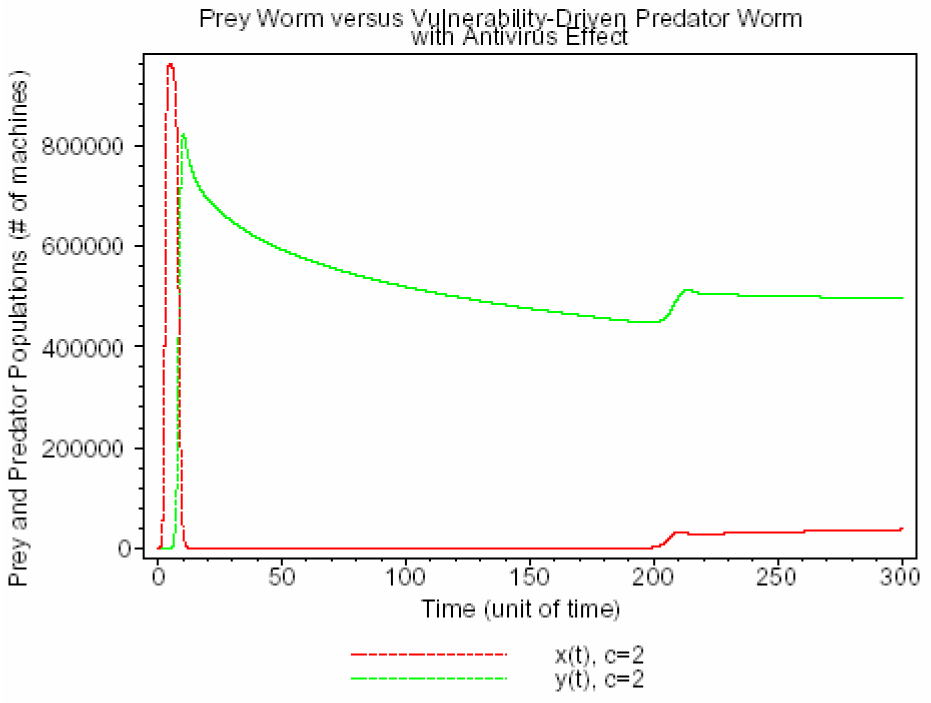


Figure 4.5. a.M=10, b.M=7, d=0.4, x_0=100, y_0=1, n_0=1000000, M=3000000.

Parameter c decreases oscillation as it loosens the coupling loop between prey and predator behaviors. As we mentioned in section 3.4 when c is zero, the system is fully tied. In the case when c is not zero, increasing the prey increases the predator, and increasing the predator decreases the prey. However, decreasing prey doesn't decrease the predator as much. As a result the peak values in both curves are smaller for. Thus oscillation vanishes sooner for larger c.

Chapter 5

PATCHING-PREY, PREDADTOR MODEL

A patching worm is a prey worm tries to protect itself from a predator by closing the backdoor through which it penetrated the system. Different instances of the same worm may or may not succeed in patching a machine, for reasons like the machine platform or configuration. We define p for be the presentation of prey-infected machines that become immune to predator infection. On the other hand $q = 1 - p$ of prey worms fail to patch some infected machines.

The prey members fall into two sets: successful and unsuccessful. The successful patching worm subset is called set-x_p, and has cardinality $x_p(t)$. The complement subset is the unsuccessful patching worms, called set-x_n, and of cardinality $x_n(t)$. The behavior of this patching prey-predator system is described in figure 5.1. The clean machines set losses member machines to both set-x_p and set-x_n, at proportional to their ratios p and q. On the other hand, only set-x_n would loss members to predators set (set-y). Equations 5.1 through 5.5 describe the dynamics of the system.

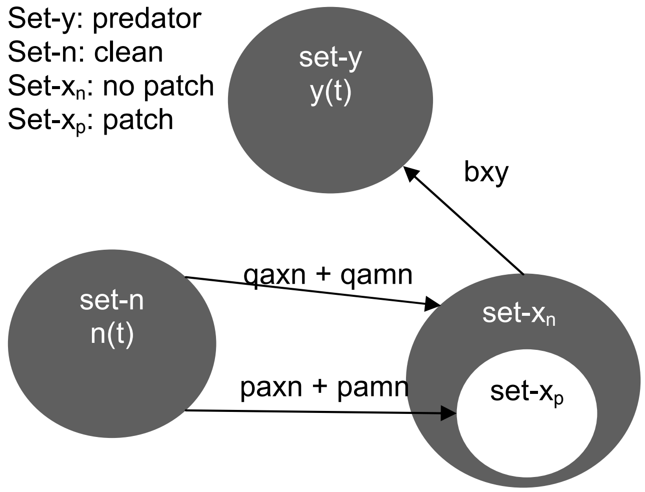


Figure 5.1. Machines transition diagram for patching prey versus predator scenario. The circles represent machines' sets and arrows are transitions' direction and rate.

$$\frac{dx_n}{dt} = q(ax_n n + ax_p n) - bx_n y \tag{5.1}$$

$$\frac{dx_p}{dt} = p(ax_n + ax_p n) \tag{5.2}$$

$$\frac{dy}{dt} = bx_n y \tag{5.3}$$

$$\frac{dn}{dt} = -ax_n n - ax_p n \tag{5.4}$$

$$
\begin{aligned}
&x_n(0) = x_{n0}, x_p(0) = x_{p0} \\
&y(0) = y_0, n(0) = n_0
\end{aligned}
\qquad (5.5)
$$

This scenario has the basis of mutant worm scenario, where a group of one worm population gains extra feature that makes them more resistant to predator worm, more infectious, etc. More interesting, the featured members can bread both featured and ordinary members, same applies to the ordinary members' breading. In this section we considered the case where sub-group of prey worms mutate and thus become immune to predator infection.

5.1. Analysis

Figure 5.2 is a plot of the prey population $x_n(t) + x_p(t)$ in red and the predator population $y(t)$ in green. Prey population grows exponentially, as long as the predator population is insignificant (around zero). Once $y(t)$ picks up in value, the prey population $x_n(t) + x_p(t)$ starts to decrease. The predator manages to take over q percent of the maximum prey infections, as the p percent are immune. Thus $x_n(t) + x_p(t)$ stops at level $p.\max(x_n + x_p)$, while $y(t)$ stops at the complement level, and the system becomes stable. The yellow curve is the plot of $n(t)$, which is the number of machines is set-n (clean vulnerable machines). This curve reaches zero as all clean machines in the system become infected. Thus the change rate of all sets in the system reaches zero at its steady state.

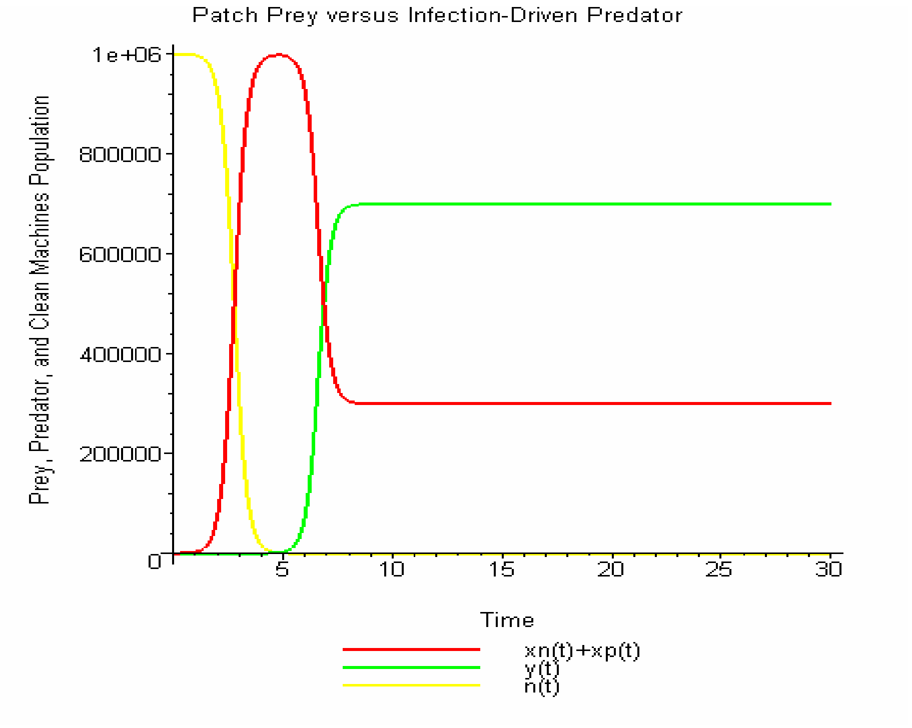


Figure 5.2. a.M=10, b.M=15, x_{n0}=100, x_{p0}=0, y0=1, n0=1000000, M=3000000, p=0.3, q=0.7

Figure 5.3 clearly shows that the equilibrium point of the system is not zero. Though the prey population reaches the environment capacity, we don't call this a prey-breakout condition. The reason for this is prey-breakout should always results in predator breakout. Instead we call this *maximum-infection condition*, when all vulnerable machines are infected at stability state.

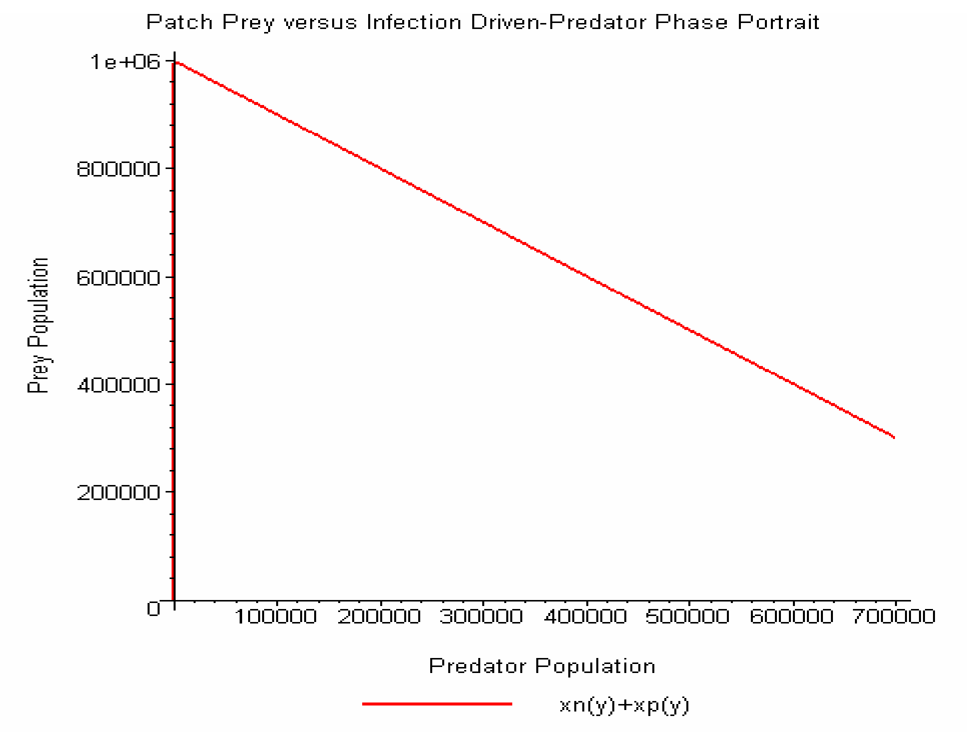


Figure 5.3. a.M=10, b.M=15, x_{n0}=100, x_{p0}=0, y0=1, n0=1000000, M=3000000, p=0.3, q=0.7

Chapter 6

PREY, PREDADTOR, AND SYSTEM PATCH MODEL

In all previously scenarios the number of vulnerable machines on the network was fixed S. In this scenario we consider the system patching factor, which immune clean vulnerable machines in set-n. Thus the size of set-S shrinks and set-R expands as result of system patches. We assume that patching a machine makes it immune to both prey and predator infections. We define the number of system patched machines as increasing function of time $w(t)$, as defined in equation 6.1. The constant k determines the capability of the patching. For example if $k = 0.4$, it means on the long run this system patch will immune 40% of the vulnerable clean machines.

$$w(t) = kt/(t+1) \tag{6.1}$$

Figure 6.1 describes the machines transitions between the different sets as result of interaction between the two worms and the system patch. Every time a clean machine is system patched, that machine is moved from set-S to the removed machines set (set-n). Members are lost by set-n and gained by set-R at a rate proportional to both, $n(t)$ and $w(t)$. The interaction between set-n, set-x, and set-y is the same, except that as set-n is shrinking, both prey and predator will have fewer machines to infect.

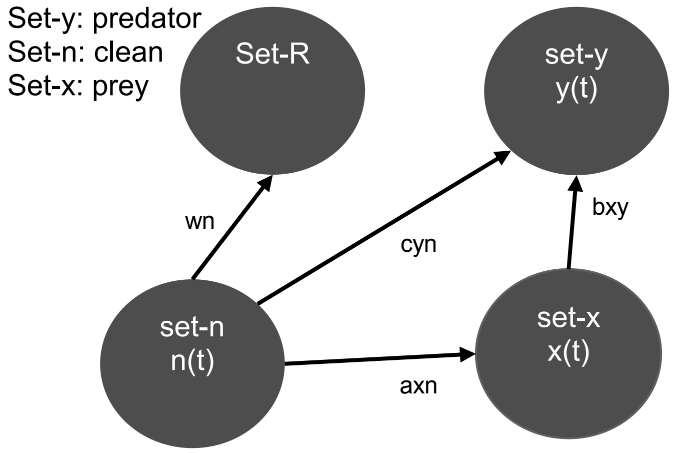


Figure 6.1. Machines transition diagram for prey-predator and system-patch scenario. The circles represent machines' sets and arrows are transitions' direction and rate.

Equations 6.2 through 6.5 are similar to equations 3.10- 3.13, except for the term wn in equation 6.4.

$$\frac{dx}{dt} = axn - bxy \tag{6.2}$$

$$\frac{dy}{dt} = cyn + bxy \tag{6.3}$$

$$\frac{dn}{dt} = -axn - cyn - wn \tag{6.4}$$

$$x(0) = x_0, y(0) = y_0, n(0) = n_0 \tag{6.5}$$

6.1. Analysis

Figure 6.2 shows how increasing system patch effectiveness could affect the prey-predator system. Increasing k results in decreasing the maximum of both $x(t)$ and $y(t)$ curves. It's clear that system patching would prevent infections breakout. Also, highly effective system patch ($k > 0.5$) can even contaminate a super-speed infection cost in few hundreds of infected machines.

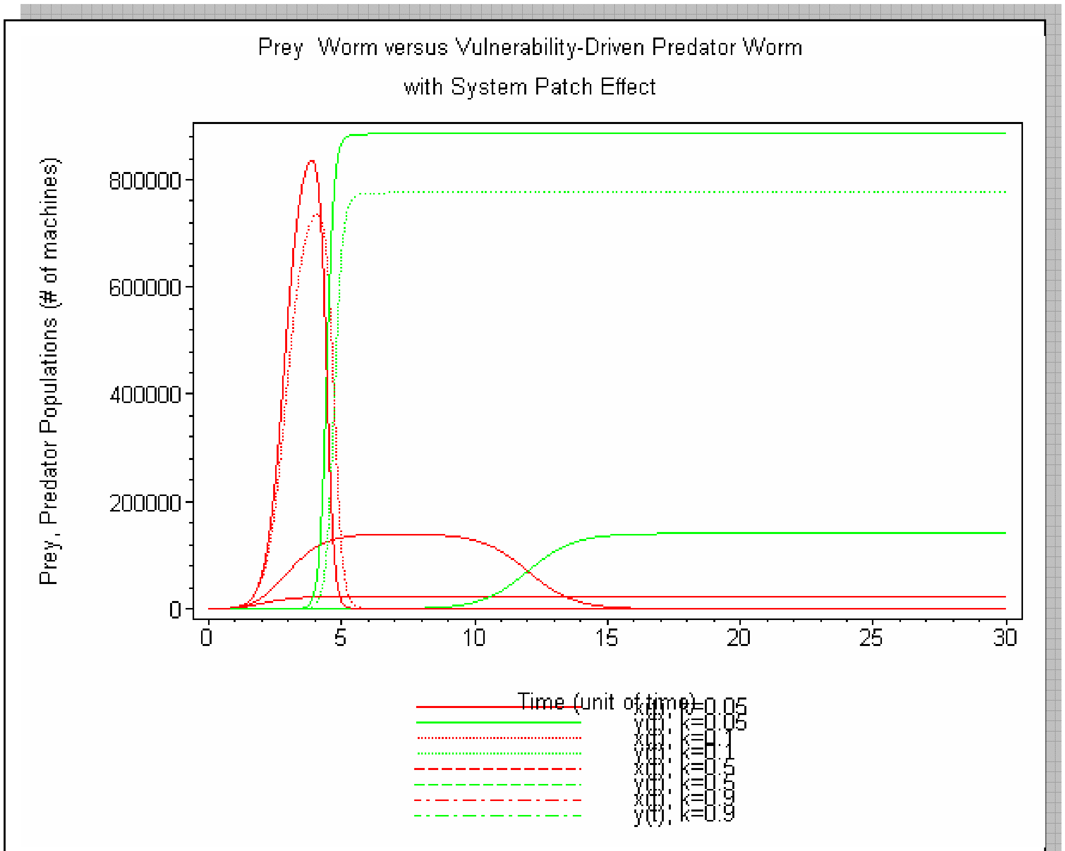


Figure 6.2. a.M=10, b.M=15, x_0=100, y_0=1, n_0=1000000, M=3000000

Figure 6.3 shows the effect of increasing k on the system equilibrium point equilibrium.

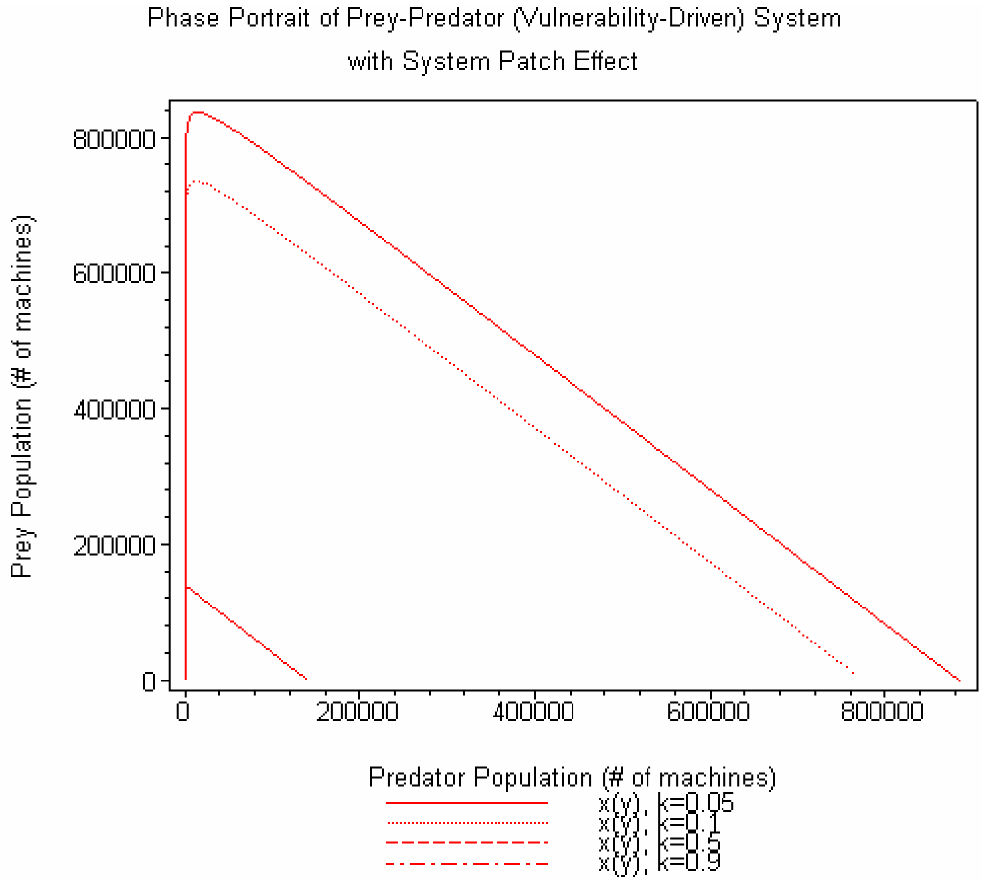


Figure 6.3. a.M=10, b.M=15, x_0=100, y_0=1, n_0=1000000, M=3000000

SUMMERY

In this paper we have presented several scenarios of virus-virus warfare. We classify worm types according to their predatory characteristics. We study and analyze the prey and predator interaction, and investigate the parameters values. We expand our deterministic time-continuous model and use numerical solutions to study several advanced scenarios, including antivirus effect on prey-predator system, patching prey versus predator, and using system patches effect. Since the beginning of this work co-incidentally several ware-fare has been reported in real Internet. However, we must warn this work does not model the specific warfare.

FUTURE WORK

There are actually additional scenarios which can be potentially modeled. One example is **Cascade Chain Worms (Wave Worm).** Many worms have more than one version. New versions are meant to update old ones. However, existence of old versions can increase or decrease the new version spread rate. Another example is **Multi-Infection scenario.** Our current model considers the number infected machines to be the worm population size. This is true as long as each machine single infection. A particular challenge in current computer virology research- including this work is very limited scope to verify model results with real world data due to absence of any estimation mechanism in the Internet. Techniques such as Internet telescopes May in near future provide such verifiability [7].

Up to date, all existing models, including those in this paper, are based on random network model. Meanwhile, the Internet is a scale-free network that has power low degree of distribution. Worms will soon hit new spread records, if they are designed to take advantage of massively connected Internet nodes, also called hubs in [9]. More important, scale-free networks can helpful to the spread worms' vaccines. [8].

REFERENCES

[1] H. Toyoizumi, A. Kara. Predators: Good Will Mobile Codes Combat against Computer Viruses. Proc. of the 2002 New Security Paradigms Workshop, 2002

[2] T.A. Burton. Volterra Integral and Differential Equations. New York: Academic Press, 1983

[3] S. Staniford, V. Paxson, N. Weaver. How to 0wn the Internet in Your Spare Time. In Proc. of the 11th USENIX Security Symposium, 2002.

[4] D. Nicol, M. Lilijenstam. Models of Active Worm Defenses. IPSI, 2004

[5] S. Staniford. Analysis of Spread of July Infestation of the Code Red Worm. http://www.silicondefense.com/cr/july.html, 2001

[6] C. Zou, W. Gong, D. Towsley. Code Red Worm Propagation Modeling and Analysis, CCS, 2002

[7] D. Moore. Network Telescopes: Observing Small or Distant Security Events. Invited presentation at the 11th USENIX Security Symposium (Security '02), 2002

[8] J. Balthrop, S. Forrest, M. Newman, M. Williamson. Technological Networks and the Spread of Computer Viruses. Science Magazine, Vol 304, Pag 527-9, 2004

[9] R. Albert, H. Jeong, A. Barabàsi. Internet: Diameter of the World-Wide Web, Nature, Vol 401, Pag 130-1, 1999

[10] B. McWilliams. New Worms Seek and Destroy Code Red. http://www.commoncriteria.org/news/newsarchive/Sept01/sept02.htm, September 2001

[11] P. Ferrie, F. Perriot, P Ször. Worm Wars. Symantec Security Response. http://ww.virusbtn.com, 2003

[12] D. Kienzle, M. Elder. Recent Worms: A Survey and Trends. Proc. of the 2003 ACM Workshop on Rapid Malcode, 2003

[13] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver. Inside the Slammer Worm. IEEE Security & Privacy, 2003

[14] R. Lemos. Worm Feeds on Sasser-Infected Computers. CNET News.com. http://zdnet.com.com/2100-1105-5212284.html, May 2004

[15] R. Vamosi. Could You Get Caught in a Virus Gang War? CNET Reviews. http://reviews.cnet.com/4520-3513_7-5125006-1.html, March 2004

[16] F-Secure Corporation. The Year of the Worm. White Paper. http://www.f-secure.com/2003, 2003

[17] R. Lemos. New Viruses Feed on MyDoom Infections. CNET News.com. http://zdnet.com.com/2100-1104-5156105.html, February 2004

[18] M. Kotadia. Bagle Spreads New Threat. ZDNet(UK). http://zdnet.com.com/2100-1105-5175172.html, March 2004

Appendix A

MAPLE COMMANDS

I. Prey-Predator Basic Scenario

```
> restart;
> with(DEtools);
> with(plots);
> with(linalg);
> with(student);
> f:= (x,y,n) -> a*x*n/M - b*x*y/M;
```

$$f := (x, y, n) \to \frac{a\,x\,n}{M} - \frac{b\,x\,y}{M}$$

```
> g:= (x,y,n) -> b*x*y/M;
```

$$g := (x, y, n) \to \frac{b\,x\,y}{M}$$

```
> h:= (x,y,n) -> -a*x*n/M;
```

$$h := (x, y, n) \to -\frac{a\,x\,n}{M}$$

```
> U:= [x(t),y(t),n(t)];
```

$$U := [\mathrm{x}(t), \mathrm{y}(t), \mathrm{n}(t)]$$

```
> sys1:= diff(x(t),t)=f(x(t),y(t),n(t)),
diff(y(t),t)=g(x(t),y(t),n(t)),
diff(n(t),t)=h(x(t),y(t),n(t));
```

$sys1 :=$

$$\frac{d}{dt}\mathrm{x}(t) = \frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M}, \frac{d}{dt}\mathrm{y}(t) = \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M}, \frac{d}{dt}\mathrm{n}(t) = -\frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M}$$

```
> ic1:= x(0)=x[0], y(0)=y[0], n(0)=n[0];
```

$$ic1 := \mathrm{x}(0) = x_0, \mathrm{y}(0) = y_0, \mathrm{n}(0) = n_0$$

```
> b_v:= [5, 10, 30, 60];
```

$$b\_v := [5, 10, 30, 60]$$

```
> prey_string:= ["x(t), b.M=5", "x(t), b.M=10", "x(t),
b.M=30", "x(t), b.M=60"];
```

prey_string := ["x(t), b.M=5" , "x(t), b.M=10" , "x(t), b.M=30" , "x(t), b.M=60"]

```
> predator_string:= ["y(t), b.M=5", "y(t), b.M=10", "y(t),
b.M=30", "y(t), b.M=60"];
```

predator_string := ["y(t), b.M=5" , "y(t), b.M=10" , "y(t), b.M=30" , "y(t), b.M=60"]

```
> xy_string:= ["x(y), b.M=5", "x(y), b.M=10", "x(y),
b.M=30", "x(y), b.M=60"];
```

xy_string := ["x(y), b.M=5" , "x(y), b.M=10" , "x(y), b.M=30" , "x(y), b.M=60"]

```
> param2:= [a=10, b=b_v[j], M=3000000, x[0]=100, y[0]=1,
N=1000000, n[0]=1000000]$j=1..4;
```

$$\mathit{param2} := [a = 10, b = 5, M = 3000000, x_0 = 100, y_0 = 1, N = 1000000, n_0 = 1000000],$$
$$[a = 10, b = 10, M = 3000000, x_0 = 100, y_0 = 1, N = 1000000, n_0 = 1000000],$$
$$[a = 10, b = 30, M = 3000000, x_0 = 100, y_0 = 1, N = 1000000, n_0 = 1000000],$$
$$[a = 10, b = 60, M = 3000000, x_0 = 100, y_0 = 1, N = 1000000, n_0 = 1000000]$$

```
> for i from 1 by 1 to 4 do
 > sol2[i]:= dsolve(eval({sys1,ic1},param2[i]), U,
 type=numeric);
 > g3[i]:= odeplot(sol2[i], [[t,x(t)],[t,y(t)]], 0..15,
 numpoints=1000, thickness=2, legend=[prey_string[i],
 predator_string[i]], title="Prey Worm versus Infection-
 Driven Predator Worm\nfor Different Parameter b Values",
 linestyle=i, labeldirections=[HORIZONTAL,VERTICAL],
 labels=["Time (unit of time)", "Prey and Predator
 Populations (# of machines)"], axes=BOXED);
 > g4[i]:= odeplot(sol2[i], [y(t),x(t)],
 0..20,numpoints=1000, thickness=2, linestyle=i,
 legend=xy_string[i], title="Phase Portrait of Prey-Predator
 (Infection-Driven) System\nfor Different Parameter b
 Values", labeldirections=[HORIZONTAL,VERTICAL],
 labels=["Predator Population (# of machines)","Prey
 Population (# of machines)"], axes=BOXED);
> end do;

> display (g3[1],g3[2]);
> display (g3[3],g3[4]);
> display (seq(g4[i],i=1..4));
```

II. Initial-Population of Prey-Predator Scenario

```
> restart;
> with(DEtools);
> with(plots);
> with(linalg);
> with(student);
> f:= (x,y,n) -> a*x*n/M - b*x*y/M;
```

$$f := (x, y, n) \rightarrow \frac{a\,x\,n}{M} - \frac{b\,x\,y}{M}$$

```
> g:= (x,y,n) -> b*x*y/M;
```

$$g := (x, y, n) \rightarrow \frac{b\,x\,y}{M}$$

```
> h:= (x,y,n) -> -a*x*n/M;
```

$$h := (x, y, n) \rightarrow -\frac{a\,x\,n}{M}$$

```
> U:= [x(t),y(t),n(t)];
```

$$U := [\mathrm{x}(t), \mathrm{y}(t), \mathrm{n}(t)]$$

```
> sys1:= diff(x(t),t)=f(x(t),y(t),n(t)),
diff(y(t),t)=g(x(t),y(t),n(t)),
diff(n(t),t)=h(x(t),y(t),n(t));
```

$sys1 :=$

$$\frac{d}{dt}\mathrm{x}(t) = \frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M},\ \frac{d}{dt}\mathrm{y}(t) = \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M},\ \frac{d}{dt}\mathrm{n}(t) = -\frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M}$$

```
> ic1:= x(0)=x[0], y(0)=y[0], n(0)=n[0];
```

$$ic1 := \mathrm{x}(0) = x_0,\ \mathrm{y}(0) = y_0,\ \mathrm{n}(0) = n_0$$

```
> x_v:= [100, 1000, 10000];
```

$$x\_v := [100, 1000, 10000]$$

```
> prey_string:=["x(t), x0=100", "x(t), x0=1000", "x(t),
x0=10000"];
```

prey_string := ["x(t), x0=100" , "x(t), x0=1000" , "x(t), x0=10000"]

```
> predator_string:=["y(t), x0=100", "y(t), x0=1000", "y(t),
x0=10000"];
```

predator_string := ["y(t), x0=100", "y(t), x0=1000", "y(t), x0=10000"]

```
> xy_string:=["x(y), x0=100", "x(y), x0=1000", "x(y),
x0=10000"];
```

xy_string := ["x(y), x0=100", "x(y), x0=1000", "x(y), x0=10000"]

```
> param2:= [a=10, b=60, M=3000000, x[0]=x_v[j], y[0]=1,
N=1000000, n[0]=1000000]$j=1..3;
```

$$param2 := [a = 10, b = 60, M = 3000000, x_0 = 100, y_0 = 1, N = 1000000, n_0 = 1000000],$$
$$[a = 10, b = 60, M = 3000000, x_0 = 1000, y_0 = 1, N = 1000000, n_0 = 1000000],$$
$$[a = 10, b = 60, M = 3000000, x_0 = 10000, y_0 = 1, N = 1000000, n_0 = 1000000]$$

```
> for i from 1 by 1 to 3 do
 > sol2[i]:= dsolve(eval({sys1,ic1},param2[i]), U,
 type=numeric);
 > g3[i]:= odeplot(sol2[i], [[t,x(t)],[t,y(t)]], 0..10,
 numpoints=1000, thickness=2, legend=[prey_string[i],
 predator_string[i]], title="Prey Worm versus Infection-
 Driven Predator Worm\nfor Different Initial Population
 Values", linestyle=i,
 labeldirections=[HORIZONTAL,VERTICAL], labels=["Time (unit
 of time)","Prey and Predator Population Size (# of
 machine)"], axes=BOXED);
 > g4[i]:= odeplot(sol2[i], [y(t),x(t)], 0..20,
 numpoints=1000, thickness=2, linestyle=i,
 legend=xy_string[i], title="Phase Portrait of Prey-Predator
 System\nfor Different Initial Population Values",
 labeldirections=[HORIZONTAL,VERTICAL],labels=["Predator
 Population (# of machines)","Prey Population (# of
 machines)"], axes=BOXED);
> end do;
```

```
> display(seq(g3[i],i=1..3));
> display(seq(g4[i],i=1..3));
```

III. Prey-Predator Expansion Case

```
> restart;
> with(DEtools);
> with(plots);
> with(linalg);
> with(student);
> f:= (x,y,n) -> a*x*n/M - b*x*y/M;
```

$$f := (x, y, n) \rightarrow \frac{a\,x\,n}{M} - \frac{b\,x\,y}{M}$$

```
> g:= (x,y,n) -> c*y*n/M + b*x*y/M;
```

$$g := (x, y, n) \rightarrow \frac{c\,y\,n}{M} + \frac{b\,x\,y}{M}$$

```
> h:= (x,y,n) -> -a*x*n/M - c*y*n/M;
```

$$h := (x, y, n) \rightarrow -\frac{a\,x\,n}{M} - \frac{c\,y\,n}{M}$$

```
> U:= [x(t),y(t),n(t)];
```

$$U := [\mathrm{x}(t), \mathrm{y}(t), \mathrm{n}(t)]$$

```
> sys1:= diff(x(t),t)=f(x(t),y(t),n(t)),
diff(y(t),t)=g(x(t),y(t),n(t)),
diff(n(t),t)=h(x(t),y(t),n(t));
```

$$sys1 := \frac{d}{dt}\,\mathrm{x}(t) = \frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M}, \frac{d}{dt}\,\mathrm{y}(t) = \frac{c\,\mathrm{y}(t)\,\mathrm{n}(t)}{M} + \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M},$$
$$\frac{d}{dt}\,\mathrm{n}(t) = -\frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{c\,\mathrm{y}(t)\,\mathrm{n}(t)}{M}$$

```
> ic1:= x(0)=x[0], y(0)=y[0], n(0)=n[0];
```

$$ic1 := \mathrm{x}(0) = x_0, \mathrm{y}(0) = y_0, \mathrm{n}(0) = n_0$$

```
> param2:= [a=10, b=25, c=5, M=3000000, x[0]=100, y[0]=1,
n[0]=1000000];
```

$$param2 := [a = 10, b = 25, c = 5, M = 3000000, x_0 = 100, y_0 = 1, n_0 = 1000000]$$

```
> sol2:= dsolve(eval({sys1,ic1},param2), U, type=numeric);
> g3:= odeplot(sol2, [[t,x(t)],[t,y(t)],[t,n(t)]], 0..10,
numpoints=1000, thickness=2, legend=["x(t)", "y(t)",
"n(t)"], title="Prey  Worm versus Vulnerability-Driven
Predator Worm", style=LINE,
labeldirections=[HORIZONTAL,VERTICAL], labels=["Time (unit
of time)","Prey, Predator, and Clean Populations (# of
machines)"], axes=BOXED);
```

```
>g4:= odeplot(sol2, [y(t),x(t)], 0..20, numpoints=1000,
thickness=2, style=LINE, legend="x(y)", title="Phase
Portrait of Prey-Predator (Vulnerability-Driven) System",
labeldirections=[HORIZONTAL,VERTICAL], labels=["Predator
Population (# of machines)","Prey Population (# of
machines)"], axes=boxed);

>display(g3);

>display(g4);
```

IV. Prey-Predator and Antivirus Scenario

```
> restart;
> with(DEtools);
> with(plots);
> with(linalg);
> with(student);
>f:= (x,y,n,z) -> a*x*n/M - b*x*y/M - z*x;
```

$$f := (x, y, n, z) \rightarrow \frac{a\,x\,n}{M} - \frac{b\,x\,y}{M} - z\,x$$

```
>g:= (x,y,n,z) -> b*x*y/M - z*y;
```

$$g := (x, y, n, z) \rightarrow \frac{b\,x\,y}{M} - z\,y$$

```
>h:= (x,y,n,z) -> -a*x*n/M + z*x + z*y;
```

$$h := (x, y, n, z) \rightarrow -\frac{a\,x\,n}{M} + z\,x + z\,y$$

```
>k:= (t) -> d/((t+1)^2);
```

$$k := t \rightarrow \frac{d}{(t+1)^2}$$

```
>sys1:= diff(x(t),t)=f(x(t),y(t),n(t),z(t)),
diff(y(t),t)=g(x(t),y(t),n(t),z(t)),
diff(n(t),t)=h(x(t),y(t),n(t),z(t)), diff(z(t),t)=k(t);
```

$$sys1 := \frac{d}{dt}\mathrm{x}(t) = \frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M} - \mathrm{z}(t)\,\mathrm{x}(t), \frac{d}{dt}\mathrm{y}(t) = \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M} - \mathrm{z}(t)\,\mathrm{y}(t),$$
$$\frac{d}{dt}\mathrm{n}(t) = -\frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} + \mathrm{z}(t)\,\mathrm{x}(t) + \mathrm{z}(t)\,\mathrm{y}(t), \frac{d}{dt}\mathrm{z}(t) = \frac{d}{(t+1)^2}$$

```
> ic1:= x(0)=x[0], y(0)=y[0], n(0)=n[0], z(0)=z[0];
```

$$ic1 := \mathrm{x}(0) = x_0, \mathrm{y}(0) = y_0, \mathrm{n}(0) = n_0, \mathrm{z}(0) = z_0$$

```
> U:= [x(t),y(t),n(t),z(t)];
```

$$U := [\mathrm{x}(t), \mathrm{y}(t), \mathrm{n}(t), \mathrm{z}(t)]$$

```
> param:= [a=10, b=25, d=0.4, x[0]=100, y[0]=1, z[0]=0,
n[0]=1000000, M=3000000];
```

$$param := [a = 10, b = 25, d = 0.4, x_0 = 100, y_0 = 1, z_0 = 0, n_0 = 1000000, M = 3000000]$$

```
> sol1:= dsolve(eval({sys1,ic1},param), U, type=numeric);
```

$$sol1 := \mathbf{proc}\ (x\_rkf45)\ \ldots\ \mathbf{end\ proc}$$

```
> odeplot(sol1, [[t,x(t)],[t,y(t)]], 0..200, numpoints=1000,
labels=["Time (unit of time)","Prey and Predator
Populations (# of machines)"],
labeldirections=[HORIZONTAL,VERTICAL], axes=BOXED,
thickness=2, title="Prey Worm versus Infection-Driven
Predator Worm\nwith Antivirus Effect",
legend=["x(t),y(t)"]);
> odeplot(sol1, [y(t),x(t)], 0..300, numpoints=2000,
labels=["Predator Population (# of machines)","Prey
Population (# of machines)"],
labeldirections=[HORIZONTAL,VERTICAL], axes=BOXED,
thickness=2, title="Phase Portrait of Prey-Predator
(Infection-Driven) System\nwith Antivirus Effect",
legend="x(y)");


> restart;
> with(DEtools);
> with(plots);
> with(linalg);
> with(student);
> f:= (x,y,n,z) -> a*x*n/M - b*x*y/M - z*x;
```

$$f := (x, y, n, z) \to \frac{a\,x\,n}{M} - \frac{b\,x\,y}{M} - z\,x$$

```
> g:= (x,y,n,z) -> c*y*n/M + b*x*y/M - z*y;
```

$$g := (x, y, n, z) \to \frac{c\,y\,n}{M} + \frac{b\,x\,y}{M} - z\,y$$

```
> h:= (x,y,n,z) -> -a*x*n/M -c*y*n/M + z*x + z*y;
```

$$h := (x, y, n, z) \rightarrow -\frac{a\,x\,n}{M} - \frac{c\,y\,n}{M} + z\,x + z\,y$$

```
> k:= (t) -> d/((t+1)^2);
```

$$k := t \rightarrow \frac{d}{(t+1)^2}$$

```
> sys1:= diff(x(t),t)=f(x(t),y(t),n(t),z(t)),
diff(y(t),t)=g(x(t),y(t),n(t),z(t)),
diff(n(t),t)=h(x(t),y(t),n(t),z(t)), diff(z(t),t)=k(t);
```

$$sys1 := \frac{d}{dt}\,\mathrm{x}(t) = \frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M} - \mathrm{z}(t)\,\mathrm{x}(t),$$

$$\frac{d}{dt}\,\mathrm{y}(t) = \frac{c\,\mathrm{y}(t)\,\mathrm{n}(t)}{M} + \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M} - \mathrm{z}(t)\,\mathrm{y}(t),$$

$$\frac{d}{dt}\,\mathrm{n}(t) = -\frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{c\,\mathrm{y}(t)\,\mathrm{n}(t)}{M} + \mathrm{z}(t)\,\mathrm{x}(t) + \mathrm{z}(t)\,\mathrm{y}(t),\ \frac{d}{dt}\,\mathrm{z}(t) = \frac{d}{(t+1)^2}$$

```
> ic1:= x(0)=x[0], y(0)=y[0], n(0)=n[0], z(0)=z[0];
```

$$ic1 := \mathrm{x}(0) = x_0,\ \mathrm{y}(0) = y_0,\ \mathrm{n}(0) = n_0,\ \mathrm{z}(0) = z_0$$

```
> U:= [x(t),y(t),n(t),z(t)];
```

$$U := [\mathrm{x}(t), \mathrm{y}(t), \mathrm{n}(t), \mathrm{z}(t)]$$

```
> c_v:= [1,2];
```

$$c\_v := [1, 2]$$

```
> param:= [a=10, b=7, c=c_v[i], d=0.4, x[0]=100, y[0]=1,
z[0]=0, n[0]=1000000, M=3000000]$j=1..2;
```

$$param := [$$

$$a = 10, b = 7, c = [1, 2]_i, d = 0.4, x_0 = 100, y_0 = 1, z_0 = 0, n_0 = 1000000, M = 3000000$$

$$], [$$

$$a = 10, b = 7, c = [1, 2]_i, d = 0.4, x_0 = 100, y_0 = 1, z_0 = 0, n_0 = 1000000, M = 3000000$$

$$]$$

```
> prey_string:=["x(t), c=1", "x(t), c=2"];
```

prey_string := ["x(t), c=1" , "x(t), c=2"]

```
> predator_string:=["y(t), c=1", "y(t), c=2"];
```

predator_string := ["y(t), c=1", "y(t), c=2"]

```
> sty:= [1, 3];
```

$$sty := [1, 3]$$

```
> xy_string:=["x(y), c=1", "x(y), c=2"];
```

$$xy\_string := [\text{"x(y), c=1"}, \text{"x(y), c=2"}]$$

```
> for i from 1 by 1 to 2 do
 > sol[i]:= dsolve(eval({sys1,ic1},param[i]), U,
 type=numeric);
 > g1[i]:=odeplot(sol[i], [[t,x(t)],[t,y(t)]], 0..300,
 numpoints=1000, labels=["Time (unit of time)","Prey and
 Predator Populations (# of machines)"],
 labeldirections=[HORIZONTAL,VERTICAL], axes=BOXED,
 thickness=2, title="Prey Worm versus Vulnerability-Driven
 Predator Worm\nwith Antivirus Effect",
 legend=[prey_string[i],predator_string[i]],
 linestyle=sty[i]);
 > g2[i]:= odeplot(sol[i], [y(t),x(t)], 0..300,
 numpoints=2000, labels=["Predator Population (# of
 machines)","Prey Population (# of machines)"],
 labeldirections=[HORIZONTAL,VERTICAL], axes=BOXED,
 thickness=2, title="Phase Portrait of Prey-Predator
 (Vulnerability-Driven) System\nwith Antivirus Effect",
 legend=xy_string[i], linestyle=sty[i]);
> end do;

> display(g1[1]);
> display(g1[2]);
```

V. Patching Prey versus Predator Scenario

```
> restart;
> with(DEtools);
> with(plots);
> with(linalg);
> with(student);
> f:= (x,y,n,m) -> q*a*x*n/M + q*a*m*n/M - b*x*y/M;
```

$$f := (x, y, n, m) \to \frac{q\,a\,x\,n}{M} + \frac{q\,a\,m\,n}{M} - \frac{b\,x\,y}{M}$$

```
> f1:=(x,y,n,m) -> p*a*x*n/M + p*a*m*n/M;
```

$$f1 := (x, y, n, m) \to \frac{p\,a\,x\,n}{M} + \frac{p\,a\,m\,n}{M}$$

```
> g:= (x,y,n,m) -> b*x*y/M;
```

$$g := (x, y, n, m) \to \frac{b\,x\,y}{M}$$

```
> h:= (x,y,n,m) -> -a*x*n/M - a*m*n/M;
```

$$h := (x, y, n, m) \to -\frac{a\,x\,n}{M} - \frac{a\,m\,n}{M}$$

```
> sys1:= diff(x(t),t)=f(x(t),y(t),n(t),m(t)),
diff(y(t),t)=g(x(t),y(t),n(t),m(t)),
diff(n(t),t)=h(x(t),y(t),n(t),m(t)),
diff(m(t),t)=f1(x(t),y(t),n(t),m(t));
```

$$sys1 := \frac{d}{dt}\,\mathrm{x}(t) = \frac{q\,a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} + \frac{q\,a\,\mathrm{m}(t)\,\mathrm{n}(t)}{M} - \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M},\ \frac{d}{dt}\,\mathrm{y}(t) = \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M},$$

$$\frac{d}{dt}\,\mathrm{n}(t) = -\frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{a\,\mathrm{m}(t)\,\mathrm{n}(t)}{M},\ \frac{d}{dt}\,\mathrm{m}(t) = \frac{p\,a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} + \frac{p\,a\,\mathrm{m}(t)\,\mathrm{n}(t)}{M}$$

```
> U:= [x(t),y(t),n(t),m(t)];
```

$$U := [\,\mathrm{x}(t), \mathrm{y}(t), \mathrm{n}(t), \mathrm{m}(t)\,]$$

```
> ic1:= x(0)=x[0], y(0)=y[0], n(0)=n[0], m(0)=m[0];
```

$$ic1 := \mathrm{x}(0) = x_0,\ \mathrm{y}(0) = y_0,\ \mathrm{n}(0) = n_0,\ \mathrm{m}(0) = m_0$$

```
> param1:= [a=10, b=15, x[0]=100, m[0]=0, y[0]=1,
n[0]=1000000, M=3000000, p=0.3, q=0.7];
```

$$param1 := [$$

$$a = 10,\ b = 15,\ x_0 = 100,\ m_0 = 0,\ y_0 = 1,\ n_0 = 1000000,\ M = 3000000,\ p = 0.3,\ q = 0.7\,]$$

```
> sol1:= dsolve(eval({sys1,ic1},param1), U, type=numeric);
```

sol1 := proc (*x_rkf45*) ... end proc

```
> odeplot(sol1, [[t,x(t)+m(t)],[t,y(t)],[t,n(t)]], 0..30,
numpoints=300, legend=["xn(t)+xp(t)","y(t)","n(t)"],
labeldirections=[HORIZONTAL,VERTICAL], labels=["Time",
"Prey, Predator, and Clean Machines Population"],
title="Patch Prey versus Infection Driven Predator",
thickness=2);
> odeplot(sol1, [y(t),x(t)+m(t)], 0..30, numpoints=300,
legend="xn(y)+xp(y)",
labeldirections=[HORIZONTAL,VERTICAL], labels=["Predator
Population", "Prey Population"], title="Patch Prey versus
Infection Driven-Predator Phase Portrait", thickness=2);
```

VI. Prey-Predator and System Patch Scenario

```
> restart;
> with(DEtools);
> with(plots);
> with(linalg);
> with(student);
> f:= (x,y,n,w) -> a*x*n/M - b*x*y/M;
```

$$f := (x, y, n, w) \to \frac{a\,x\,n}{M} - \frac{b\,x\,y}{M}$$

```
> g:= (x,y,n,w) -> c*y*n/M + b*x*y/M;
```

$$g := (x, y, n, w) \to \frac{c\,y\,n}{M} + \frac{b\,x\,y}{M}$$

```
> h:= (x,y,n,w) -> -a*x*n/M - c*y*n/M - w*n;
```

$$h := (x, y, n, w) \to -\frac{a\,x\,n}{M} - \frac{c\,y\,n}{M} - w\,n$$

```
> k:= (t) -> g/(t+1);
```

$$k := t \to \frac{g}{t+1}$$

```
> U:= [x(t),y(t),n(t),w(t)];
```

$$U := [\mathrm{x}(t), \mathrm{y}(t), \mathrm{n}(t), \mathrm{w}(t)]$$

```
> sys1:= diff(x(t),t)=f(x(t),y(t),n(t),w(t)),
diff(y(t),t)=g(x(t),y(t),n(t),w(t)),
diff(n(t),t)=h(x(t),y(t),n(t),w(t)), diff(w(t),t)=k(t);
```

$$sys1 := \frac{d}{dt}\,\mathrm{x}(t) = \frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M},\ \frac{d}{dt}\,\mathrm{y}(t) = \frac{c\,\mathrm{y}(t)\,\mathrm{n}(t)}{M} + \frac{b\,\mathrm{x}(t)\,\mathrm{y}(t)}{M},$$

$$\frac{d}{dt}\,\mathrm{n}(t) = -\frac{a\,\mathrm{x}(t)\,\mathrm{n}(t)}{M} - \frac{c\,\mathrm{y}(t)\,\mathrm{n}(t)}{M} - \mathrm{w}(t)\,\mathrm{n}(t),\ \frac{d}{dt}\,\mathrm{w}(t) = \frac{g}{t+1}$$

```
> ic1:= x(0)=x[0], y(0)=y[0], n(0)=n[0], w(0)=w[0];
```

$$ic1 := \mathrm{x}(0) = x_0,\ \mathrm{y}(0) = y_0,\ \mathrm{n}(0) = n_0,\ \mathrm{w}(0) = w_0$$

```
> g_v:= [0.05, 0.1, 0.5, 0.9];
```

$$g\_v := [0.05, 0.1, 0.5, 0.9]$$

```
> param2:= [a=10, b=25, c=2, g=g_v[j], M=3000000, x[0]=100,
y[0]=1, n[0]=1000000, w[0]=0]$j=1..4;
```

$$param2 := [$$
$$a = 10, b = 25, c = 2, g = 0.05, M = 3000000, x_0 = 100, y_0 = 1, n_0 = 1000000, w_0 = 0],$$
$$[a = 10, b = 25, c = 2, g = 0.1, M = 3000000, x_0 = 100, y_0 = 1, n_0 = 1000000, w_0 = 0],$$
$$[a = 10, b = 25, c = 2, g = 0.5, M = 3000000, x_0 = 100, y_0 = 1, n_0 = 1000000, w_0 = 0],$$
$$[a = 10, b = 25, c = 2, g = 0.9, M = 3000000, x_0 = 100, y_0 = 1, n_0 = 1000000, w_0 = 0]$$

```
> x_s:=["x(t), k=0.05", "x(t), k=0.1", "x(t), k=0.5",
"x(t), k=0.9"];
```

x_s := ["x(t), k=0.05" , "x(t), k=0.1" , "x(t), k=0.5" , "x(t), k=0.9"]

```
> y_s:=["y(t), k=0.05", "y(t), k=0.1", "y(t), k=0.5",
"y(t), k=0.9"];
```

y_s := ["y(t), k=0.05" , "y(t), k=0.1" , "y(t), k=0.5" , "y(t), k=0.9"]

```
> xy_s:=["x(y), k=0.05", "x(y), k=0.1", "x(y), k=0.5",
"x(y), k=0.9"];
```

xy_s := ["x(y), k=0.05" , "x(y), k=0.1" , "x(y), k=0.5" , "x(y), k=0.9"]

```
> for i from 1 by 1 to 4 do
 > sol2[i]:= dsolve(eval({sys1,ic1},param2[i]), U,
 type=numeric);
 > g3[i]:= odeplot(sol2[i], [[t,x(t)],[t,y(t)]], 0..30,
 numpoints=300, thickness=1, legend=[x_s[i], y_s[i]],
 title="Prey  Worm versus Vulnerability-Driven Predator
 Worm\nwith System Patch Effect", linestyle=i,
 labeldirections=[HORIZONTAL,VERTICAL], labels=["Time (unit
 of time)","Prey, Predator Populations (# of machines)"],
 axes=BOXED);
 > g4[i]:= odeplot(sol2[i], [y(t),x(t)], 0..20,
 numpoints=300, thickness=1, linestyle=i, legend=xy_s[i],
 title="Phase Portrait of Prey-Predator (Vulnerability-
 Driven) System\nwith System Patch Effect",
 labeldirections=[HORIZONTAL,VERTICAL], labels=["Predator
 Population (# of machines)","Prey Population (# of
 machines)"], axes=boxed);
> end do;

> display (seq(g3[i],i=1..4));
> display (seq(g4[i],i=1..4));
```