**Advanced Placement Biology**

**Research Project**

**Chris Crutchfield**

**Brian Richardson**

**4/24/2001**

**Abstract**

Simulations of populations are a very useful tool to the field of science. Instead of having to manually track and count populations of organisms, scientists can now simulate the interactions of organisms in the environment and predict an ecosystem’s future. An application of such a technology would be useful to simulate, for instance, the Arroyo Del Valle creek and predict whether or not there are dire consequences for it in the future.

This project’s aim is to simulate accurately the population dynamics of an ecosystem involving a producer species, a herbivore species, and a consumer species in a computer program. We expect to see in this predator-prey model a demonstration of the interactions between predators and prey in the ecosystem.

For these means, we constructed the computer program in the C++ programming language that, with the means of equations, calculated and predicted the population sizes of the three species.

Our results showed that, yes, the population dynamics could be simulated by a computer program, but not nearly accurately enough to predict the future very well with the algorithms we had.

**Introduction**

**What sort of interactions go on between a community of organisms in an ecosystem?**

# Well, to answer that, we must first ask, what is a community and what is an ecosystem?

A community is all the organisms that inhabit a common environment and interact with each other. And example of such a community is the variety of organisms living within the Arroyo Del Valle creek. These organisms interact with each other. Algae produces oxygen for the myriad water creatures that inhabit the lake, and various animals hunt and eat other organisms in the creek. These interactions between different organisms form the community of the area.

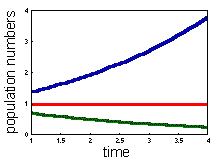
The community inhabits an ecosystem, which is the interacting system of both the abiotic and biotic parts of the environment. The abiotic conditions of an environment are very important, because they affect the organisms living in the ecosystem immensely. If such cycles as the nitrogen cycle are inhibited somehow, plants in the ecosystem will be severely hampered in any endeavors to grow.

## **How do the organisms grow in the ecosystem?**

There are two polar extremes in organism growth. Some organisms, the r-strategists, grow exponentially, and quickly breach the carrying capacity. This overproduction of offspring is usually soon followed by a sharp drop in population because the ecosystem cannot support such a large amount of organisms. These r-strategists follow an exponential equation of growth. The exponential model of growth was presented by the famous British economist, Thomas Robert Malthus (1766-1834), who stated that populations tended to increase much faster than the supply of food. The exponential equation for growth of a population is:

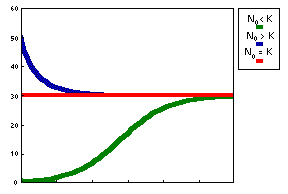
dP/dt = (b – m)\*P = r\*P

dP/dt is the change in population with respect to time. “b” stands for the amount of births in the time interval, “m” stands for the amount of deaths in the time interval, and P is the population size. (b – m) can be abbreviated by r, which is the rate of the exponential. There are three cases in an exponential function: the blue is where r is positive, and the population is growing, the red is where r is zero and the population stays the same size, and the green is where r is negative and the population is decreasing in size.



The second case of growth for organisms is the K-strategist model. This model, so called because K stands for the Carrying Capacity of the system, was developed by the Belgian Mathematician Pierre Verhulst in 1838 who brought forth the concept that the rate at which the population is growing may be limited by the density of the population. The so-called Logistic model is defined by the differential equation:

dP/dt = r0\*P\*(1-P/K)



In this model, there are three outcomes. The blue is where the initial population, P0 is greater than the Carrying Capacity. It swiftly drops down the population size to reach the carrying capacity. The second outcome is where P0 equals the carrying capacity. In this case, nothing happens, because any rise or fall in population is going away from the carrying capacity, which the equation wants to converge to. The final outcome, in the green, is where P0 is less than the carrying capacity, and it moves upwards to reach the carrying capacity level. The logistic model has two equilibria in its system: when P is equal to 0 and when P is equal to the carrying capacity. Once it reaches either, it will stay there unless the population is upset by, say, a natural disaster.

**Predator-Prey Interaction**

Organisms in an ecosystem interact with each other, usually in a predator-prey interaction. In a food chain, the consumers on top rely on their prey below them to stay alive. When a lower level organism is removed from the food chain, this causes drastic consequences for all organisms above them on the food chain. For example, if there were huge runoffs of sewage or fertilizer into the Arroyo Del Valle, an algae bloom would likely occur in the water due to the increased levels of nitrates and phosphates. Since there is a huge growth in the population of algae, the population sizes of the organisms that feed on them grow in turn because of the overabundance of food. This ripple effect continues down the line, as each predator population gains size because of the abundance of the prey population that it feeds on. However, when the algae bloom ends and the algae population returns to normal, the organisms that feed on the algae find themselves starving for lack of food. They die in turn. This also ripples down the populations of organisms in the creek causing many of them to die. This effect is one of the interactions we hope to see in our simulation.

##### **Hypothesis/Prediction**

**Hypothesis:** The population dynamics of an ecosystem containing an herbivore, a carnivore, and a producer species can be predicted accurately by a computer program

**Prediction**: If the population dynamics of an ecosystem containing an herbivore, a carnivore, and a producer species can be predicted accurately by a computer program, then we will see a reasonable simulation of the population levels of an ecosystem.

**Experiment**

Due to the unusual nature of our project (it involves programming rather than a lab experiment), the procedure will discuss the nature of the program that we wrote and how it works, rather than a step-by-step detail of what we did to make it.

***Procedure:***

After extensive research into the subject, a series of equations were found/developed to simulate the interactions between the producer, the herbivore, and the carnivore. Obviously, this model could not be very detailed, as a lifelike model of an ecosystem is way outside the realm of mere high school students. Thus we limited the simulation to just the biotic interactions of the three species.

First, we created a basic model of the interactions using word equations based on a single ecosystem, with no migration. The equations are in bold, my commentary is underneath.

**Carnivore(t) = Carnivore(t – dt) + (Births – Deaths) \* dt**

*Simply put, the amount of carnivores at time t is equal to the previous number of carnivores (dt is the Calculus term for the increment) plus the births, minus the deaths.*

**Births = Birthrate\*(Carnivore – Deaths)**

*Births are equal to the birthrate times the number of surviving individuals*

**Deaths = Carnivore – Consumed/ConsumptionRate**

*Deaths are equal to the amount of carnivores who starved*



**Herbivore(t) = Herbivore(t – dt) + (Births – Deaths) \* dt**

*Like above, the amount of herbivores at time t is equal to the previous amount of herbivores plus the amount of births, minus the amount of deaths.*

**Births = Birthrate\*(Herbivore – Deaths)**

*Births are equal to the birthrate times the number of surviving individuals*

**Deaths = Herbivore – Consumed/ConsumptionRate**

*Deaths are equal to the amount of herbivores who starved*

*[Note: We could not get the equation to work based on the Herbivores as both predators and prey. The numbers of herbivores would drop alarmingly during every run.]*



**Producer(t) = Producer(t – dt) + (Births – Deaths) \* dt**

*Like above, the amount of producers at time t is equal to the previous amount of producers plus the amount of births, minus the amount of deaths.*

**Births = (Producer – Deaths)\*Birthrate\*(1 – Producers/CarryingCapacity)**

*The number of births are equal to the amount of surviving producers times the birthrate times one minus the producers divided by carrying capacity. This is the logistic model of growth defined by the equation:*

dP/dt = r0\*P\*(1-P/K)

**Deaths = Consumption**

*In this model, producers die when they are consumed by herbivores.*



**Consumption = min(prey, max(0, consumptionrate\*predator))**

*Consumption is the minimum value between the number of prey and the number of predators times the consumption rate. The max function is there to prevent negative consumption from occurring.*

***Constants:***

*For this model, we specified some constants for simplicity’s sake:*

**Consumption Rate = 1**

*In one iteration, one predator eats one prey*

**Producer Birth Rate = 2**

**Herbivore Birth Rate = .2**

**Carnivore Birth Rate = .02**

**Carrying capacity = 100000**

**Producer(0) = 10000**

**Herbivore(0) = 1000**

**Carnivore(0) = 100**



This model has a tendency to “crash” (all the species die) and so was not really an accurate model of an ecosystem. After some more research, we stumbled upon a different method of modeling. Basically, instead of one big “bubble” of an ecosystem, this method divided the ecosystem up into smaller “bubbles” with immigration and emigration of individuals in between. So, if conditions were not favorable for herbivores in bubble number 1, a certain amount of them would migrate towards bubble number 2. My model has only 4 bubbles, which made it much more stable, but still able to crash. The equations were revised to this:

[Note: I have simplified them due to over redundancy. The complete equations can be seen in the source code.]



















Levels of migration are related directly to the amount of deaths in the quadrant



**Organism1(t) = Organism1(t – dt) + (Births1 + Immigration1 – Emigration1 – Death1**

*The number of organisms at time t in quadrant 1 is equal to the previous number of organisms in quadrant 1 plus the amount of births in quadrant 1, plus the amount of immigration to quadrant 1 minus the amount of emigration from quadrant 1 minus the amount of deaths in quadrant 1.*

**Births1 = Birthrate\*(Organism1 – Deaths1)**

*Births are equal to the birthrate times the number of surviving organisms in quadrant 1.*

# **Immigration1 = OrganismsFrom2 + OrganismsFrom4**

*Immigration to quadrant 1 is equal to the number of organisms from quadrants two and four.*

**Emigration1 = min(Organisms1 – Deaths1, migrationRate\*Deaths)**

*The amount of emigration from quadrant one is the minimum value between the amount of organims in quadrant one minus the amount that died in quadrant one, and the rate of migration times the amount of deaths. Basically, the amount of organisms that migrate is a fraction of those who die. Dying organisms signify bad conditions for that particular “bubble”*

# **Deaths1 = Organisms1 – Consumption/ConsumptionRate**

*The amount of deaths are equal to the amount of organisms that starved.*



*Several of the other constants were specified, such as the migration rates of the Producers (.1) the herbivores(.05) and the carnivores(.01).*

*This model produced a more stable simulation, which was much harder to crash the ecosystem than the previous model.*

**Source Code:**

This is the source code for the first model:

*#include <iostream.h>*

*#include <fstream.h>*

*#include <apstring.h>*

*#include <stdlib.h>*

*#define CARNIVORE\_BIRTH\_RATE .02*

*#define HERBIVORE\_BIRTH\_RATE .2*

*#define PRODUCER\_BIRTH\_RATE 2*

*#define PRODUCER\_CARRYING\_CAPACITY 100000*

*double min(double a, double b);*

*double max(double a, double b);*

*void main()*

*{*

*double producer, herbivore, carnivore;*

*int time;*

*apstring fileName = "F:\\Documents and Settings\\Administrator\\Desktop\\results\\";*

*apstring temp;*

*cout << "Input initial producer population size (negative for default): ";*

*cin >> producer;*

*cout << "Input initial herbivore population size (negative for default): ";*

*cin >> herbivore;*

*cout << "Input initial carnivore population size (negative for default): ";*

*cin >> carnivore;*

*cout << "Input amount of iterations: ";*

*cin >> time;*

*cout << "Input the file name to be stored to (do not add file extension)" << endl << "(type 'none' to have it output to console)" << endl << fileName;*

*cin >> temp;*

*cin.get();*

*cout << endl << endl;*

*ofstream infile;*

*if(temp != "none")*

*{*

*fileName += temp;*

*fileName += ".txt";*

*infile.open(fileName.c\_str());*

*if(infile.fail())*

*{*

*cerr << "ERROR: Unable to open file " << fileName << endl;*

*abort();*

*}*

*}*

*if(producer < 0)*

*producer = 10000;*

*if(herbivore < 0)*

*herbivore = 1000;*

*if(carnivore < 0)*

*carnivore = 100;*

*if(temp != "none")*

*infile << producer << " " << herbivore << " " << carnivore << endl;*

*for(int t=1;t<=time;t++)*

*{*

*carnivore += (CARNIVORE\_BIRTH\_RATE\*min(herbivore, max(0, 1\*carnivore)/1) - carnivore + min(herbivore, max(0, 1\*carnivore)/1));*

*herbivore += (HERBIVORE\_BIRTH\_RATE\*min(producer, max(0, 1\*herbivore)/1) - herbivore + min(producer, max(0, 1\*herbivore)/1));*

*producer += (((producer - min(producer, max(0, 1\*herbivore)))\*PRODUCER\_BIRTH\_RATE\*(1-producer/PRODUCER\_CARRYING\_CAPACITY)) - min(producer, max(0, 1\*herbivore)/1));*

*if(temp == "none")*

*{*

*cout << "Iteration[" << t << "]" << endl;*

*cout << "Producer population size: " << producer << endl;*

*cout << "Herbivore population size: " << herbivore << endl;*

*cout << "Carnivore population size: " << carnivore << endl;*

*cin.get();*

*cout << endl << endl << endl;*

*}*

*else*

*infile << producer << " " << herbivore << " " << carnivore << endl;*

*}*

*}*

*double min(double a, double b)*

*{*

*if(a < b)*

*return a;*

*return b;*

*}*

*double max(double a, double b)*

*{*

*if(a > b)*

*return a;*

*return b;*

*}*

**This is the source code for the second model:**

*#include <iostream.h>*

*#include <iomanip.h>*

*#include <stdlib.h>*

*#include <time.h>*

*#include <stdio.h>*

*#include <fstream.h>*

*#include <apstring.h>*

*#define CARNIVORE\_BIRTH\_RATE .02*

*#define HERBIVORE\_BIRTH\_RATE .2*

*#define PRODUCER\_BIRTH\_RATE 2*

*#define PRODUCER\_CARRYING\_CAPACITY 10000*

*#define HERBIVORE\_CARRYING\_CAPACITY 1000*

*#define MIGRATION\_RATE\_PRODUCER .1*

*#define MIGRATION\_RATE\_HERBIVORE .05*

*#define MIGRATION\_RATE\_CARNIVORE .01*

*double min(double a, double b);*

*double max(double a, double b);*

*void main()*

*{*

*srand( (unsigned)time( NULL ) );*

*double producer, herbivore, carnivore;*

*int time;*

*double carnivore1, carnivore2, carnivore3, carnivore4;*

*double herbivore1, herbivore2, herbivore3, herbivore4;*

*double producer1, producer2, producer3, producer4;*

*double pprodD1, pprodR1, pprodL2, pprodD2, pprodU3, pprodR3, pprodL4, pprodU4;*

*double pherbD1, pherbR1, pherbL2, pherbD2, pherbU3, pherbR3, pherbL4, pherbU4;*

*double pcarnD1, pcarnR1, pcarnL2, pcarnD2, pcarnU3, pcarnR3, pcarnL4, pcarnU4;*

*double prodD1, prodR1, prodL2, prodD2, prodU3, prodR3, prodL4, prodU4;*

*double herbD1, herbR1, herbL2, herbD2, herbU3, herbR3, herbL4, herbU4;*

*double carnD1, carnR1, carnL2, carnD2, carnU3, carnR3, carnL4, carnU4;*

*apstring fileName = "F:\\Documents and Settings\\Administrator\\Desktop\\results\\";*

*apstring temp;*

*cout << "Input initial producer population size (negative for default): ";*

*cin >> producer;*

*cout << "Input initial herbivore population size (negative for default): ";*

*cin >> herbivore;*

*cout << "Input initial carnivore population size (negative for default): ";*

*cin >> carnivore;*

*cout << "Input amount of iterations: ";*

*cin >> time;*

*cout << "Input the file name to be stored to (do not add file extension)" << endl << "(type 'none' to have it output to console)" << endl << fileName;*

*cin >> temp;*

*cin.get();*

*cout << endl << endl;*

*ofstream infile;*

*if(temp != "none")*

*{*

*fileName += temp;*

*fileName += ".txt";*

*infile.open(fileName.c\_str());*

*if(infile.fail())*

*{*

*cerr << "ERROR: Unable to open file " << fileName << endl;*

*abort();*

*}*

*}*

*if(producer < 0)*

*producer = 10000;*

*if(herbivore < 0)*

*herbivore = 1000;*

*if(carnivore < 0)*

*carnivore = 100;*

*carnivore1 = carnivore2 = carnivore3 = carnivore4 = carnivore/4;*

*herbivore1 = herbivore2 = herbivore3 = herbivore4 = herbivore/4;*

*producer1 = producer2 = producer3 = producer4 = producer/4;*

*if(temp != "none")*

*infile << (producer1 + producer2 + producer3 + producer4) << " " << (herbivore1 + herbivore2 + herbivore3 + herbivore4) << " " << (carnivore1 + carnivore2 + carnivore3 + carnivore4) << endl;*

*for(int t=1;t<=time;t++)*

*{*

*pprodD1 = double(rand())/32767;*

*pprodR1 = double(rand())/32767;*

*pprodL2 = double(rand())/32767;*

*pprodD2 = double(rand())/32767;*

*pprodU3 = double(rand())/32767;*

*pprodR3 = double(rand())/32767;*

*pprodL4 = double(rand())/32767;*

*pprodU4 = double(rand())/32767;*

*pherbD1 = double(rand())/32767;*

*pherbR1 = double(rand())/32767;*

*pherbL2 = double(rand())/32767;*

*pherbD2 = double(rand())/32767;*

*pherbU3 = double(rand())/32767;*

*pherbR3 = double(rand())/32767;*

*pherbL4 = double(rand())/32767;*

*pherbU4 = double(rand())/32767;*

*pcarnD1 = double(rand())/32767;*

*pcarnR1 = double(rand())/32767;*

*pcarnL2 = double(rand())/32767;*

*pcarnD2 = double(rand())/32767;*

*pcarnU3 = double(rand())/32767;*

*pcarnR3 = double(rand())/32767;*

*pcarnL4 = double(rand())/32767;*

*pcarnU4 = double(rand())/32767;*

*prodD1 = MIGRATION\_RATE\_PRODUCER\*(producer1 - min(producer1,max(0,1\*herbivore1)))\*pprodD1/(pprodD1 + pprodR1);*

*prodR1 = MIGRATION\_RATE\_PRODUCER\*(producer1 - min(producer1,max(0,1\*herbivore1)))\*pprodR1/(pprodD1 + pprodR1);*

*prodL2 = MIGRATION\_RATE\_PRODUCER\*(producer2 - min(producer2,max(0,1\*herbivore2)))\*pprodL2/(pprodD2 + pprodL2);*

*prodD2 = MIGRATION\_RATE\_PRODUCER\*(producer2 - min(producer2,max(0,1\*herbivore2)))\*pprodD2/(pprodD2 + pprodL2);*

*prodU3 = MIGRATION\_RATE\_PRODUCER\*(producer3 - min(producer3,max(0,1\*herbivore3)))\*pprodU3/(pprodU3 + pprodR3);*

*prodR3 = MIGRATION\_RATE\_PRODUCER\*(producer3 - min(producer3,max(0,1\*herbivore3)))\*pprodR3/(pprodU3 + pprodR3);*

*prodL4 = MIGRATION\_RATE\_PRODUCER\*(producer4 - min(producer4,max(0,1\*herbivore4)))\*pprodL4/(pprodL4 + pprodU4);*

*prodU4 = MIGRATION\_RATE\_PRODUCER\*(producer4 - min(producer4,max(0,1\*herbivore4)))\*pprodU4/(pprodL4 + pprodU4);*

*herbD1 = MIGRATION\_RATE\_HERBIVORE\*(herbivore1 - min(herbivore1,max(0,1\*carnivore1)))\*pherbD1/(pherbD1 + pherbR1);*

*herbR1 = MIGRATION\_RATE\_HERBIVORE\*(herbivore1 - min(herbivore1,max(0,1\*carnivore1)))\*pherbR1/(pherbD1 + pherbR1);*

*herbL2 = MIGRATION\_RATE\_HERBIVORE\*(herbivore2 - min(herbivore2,max(0,1\*carnivore2)))\*pherbL2/(pherbL2 + pherbD2);*

*herbD2 = MIGRATION\_RATE\_HERBIVORE\*(herbivore2 - min(herbivore2,max(0,1\*carnivore2)))\*pherbD2/(pherbL2 + pherbD2);*

*herbU3 = MIGRATION\_RATE\_HERBIVORE\*(herbivore3 - min(herbivore3,max(0,1\*carnivore3)))\*pherbU3/(pherbU3 + pherbR3);*

*herbR3 = MIGRATION\_RATE\_HERBIVORE\*(herbivore3 - min(herbivore3,max(0,1\*carnivore3)))\*pherbR3/(pherbU3 + pherbR3);*

*herbL4 = MIGRATION\_RATE\_HERBIVORE\*(herbivore4 - min(herbivore4,max(0,1\*carnivore4)))\*pherbL4/(pherbL4 + pherbU4);*

*herbU4 = MIGRATION\_RATE\_HERBIVORE\*(herbivore4 - min(herbivore4,max(0,1\*carnivore4)))\*pherbU4/(pherbL4 + pherbU4);*

*carnD1 = min(min(carnivore1,max(0,1\*carnivore1)), MIGRATION\_RATE\_CARNIVORE\*(carnivore1-min(carnivore1,max(0,1\*carnivore1))))\*pcarnD1/(pcarnD1 + pcarnR1);*

*carnR1 = min(min(carnivore1,max(0,1\*carnivore1)), MIGRATION\_RATE\_CARNIVORE\*(carnivore1-min(carnivore1,max(0,1\*carnivore1))))\*pcarnR1/(pcarnD1 + pcarnR1);*

*carnL2 = min(min(carnivore2,max(0,1\*carnivore2)), MIGRATION\_RATE\_CARNIVORE\*(carnivore2-min(carnivore2,max(0,1\*carnivore2))))\*pcarnL2/(pcarnL2 + pcarnD2);*

*carnD2 = min(min(carnivore2,max(0,1\*carnivore2)), MIGRATION\_RATE\_CARNIVORE\*(carnivore2-min(carnivore2,max(0,1\*carnivore2))))\*pcarnD2/(pcarnL2 + pcarnD2);*

*carnU3 = min(min(carnivore3,max(0,1\*carnivore3)), MIGRATION\_RATE\_CARNIVORE\*(carnivore3-min(carnivore3,max(0,1\*carnivore3))))\*pcarnU3/(pcarnU3 + pcarnR3);*

*carnR3 = min(min(carnivore3,max(0,1\*carnivore3)), MIGRATION\_RATE\_CARNIVORE\*(carnivore3-min(carnivore3,max(0,1\*carnivore3))))\*pcarnR3/(pcarnU3 + pcarnR3);*

*carnL4 = min(min(carnivore4,max(0,1\*carnivore4)), MIGRATION\_RATE\_CARNIVORE\*(carnivore4-min(carnivore4,max(0,1\*carnivore4))))\*pcarnL4/(pcarnL4 + pcarnU4);*

*carnU4 = min(min(carnivore4,max(0,1\*carnivore4)), MIGRATION\_RATE\_CARNIVORE\*(carnivore4-min(carnivore4,max(0,1\*carnivore4))))\*pcarnU4/(pcarnL4 + pcarnU4);*

*carnivore1 += (CARNIVORE\_BIRTH\_RATE\*(min(herbivore1,max(0,1\*carnivore1))/1) + carnL2 + carnU3 - min(min(carnivore1,max(0,1\*carnivore1)), MIGRATION\_RATE\_CARNIVORE\*(carnivore1-min(carnivore1,max(0,1\*carnivore1)))) - carnivore1 + min(herbivore1,max(0,1\*carnivore1))/1);*

*carnivore2 += (CARNIVORE\_BIRTH\_RATE\*(min(herbivore2,max(0,1\*carnivore2))/1) + carnR1 + carnU4 - min(min(carnivore2,max(0,1\*carnivore2)), MIGRATION\_RATE\_CARNIVORE\*(carnivore2-min(carnivore2,max(0,1\*carnivore2)))) - carnivore2 + min(herbivore2,max(0,1\*carnivore2))/1);*

*carnivore3 += (CARNIVORE\_BIRTH\_RATE\*(min(herbivore3,max(0,1\*carnivore3))/1) + carnD1 + carnL4 - min(min(carnivore3,max(0,1\*carnivore3)), MIGRATION\_RATE\_CARNIVORE\*(carnivore3-min(carnivore3,max(0,1\*carnivore3)))) - carnivore3 + min(herbivore3,max(0,1\*carnivore3))/1);*

*carnivore4 += (CARNIVORE\_BIRTH\_RATE\*(min(herbivore4,max(0,1\*carnivore4))/1) + carnD2 + carnR3 - min(min(carnivore4,max(0,1\*carnivore4)), MIGRATION\_RATE\_CARNIVORE\*(carnivore4-min(carnivore4,max(0,1\*carnivore4)))) - carnivore4 + min(herbivore4,max(0,1\*carnivore4))/1);*

*herbivore1 += ((1 - herbivore1/HERBIVORE\_CARRYING\_CAPACITY)\*HERBIVORE\_BIRTH\_RATE\*(min(producer1,max(0,1\*herbivore1))/1) + herbL2 + herbU3 - min(min(herbivore1,max(0,1\*herbivore1)), MIGRATION\_RATE\_HERBIVORE\*(herbivore1-min(herbivore1,max(0,1\*herbivore1)))) - herbivore1 + min(producer1,max(0,1\*herbivore1))/1);*

*herbivore2 += ((1 - herbivore2/HERBIVORE\_CARRYING\_CAPACITY)\*HERBIVORE\_BIRTH\_RATE\*(min(producer2,max(0,1\*herbivore2))/1) + herbR1 + herbU4 - min(min(herbivore2,max(0,1\*herbivore2)), MIGRATION\_RATE\_HERBIVORE\*(herbivore2-min(herbivore2,max(0,1\*herbivore2)))) - herbivore2 + min(producer2,max(0,1\*herbivore2))/1);*

*herbivore3 += ((1 - herbivore3/HERBIVORE\_CARRYING\_CAPACITY)\*HERBIVORE\_BIRTH\_RATE\*(min(producer3,max(0,1\*herbivore3))/1) + herbD1 + herbL4 - min(min(herbivore3,max(0,1\*herbivore3)), MIGRATION\_RATE\_HERBIVORE\*(herbivore3-min(herbivore3,max(0,1\*herbivore3)))) - herbivore3 + min(producer3,max(0,1\*herbivore3))/1);*

*herbivore4 += ((1 - herbivore4/HERBIVORE\_CARRYING\_CAPACITY)\*HERBIVORE\_BIRTH\_RATE\*(min(producer4,max(0,1\*herbivore4))/1) + herbD2 + herbR3 - min(min(herbivore4,max(0,1\*herbivore4)), MIGRATION\_RATE\_HERBIVORE\*(herbivore4-min(herbivore4,max(0,1\*herbivore4)))) - herbivore4 + min(producer4,max(0,1\*herbivore4))/1);*

*producer1 += ((producer1 - min(producer1 ,max(0,1\*herbivore1)))\*PRODUCER\_BIRTH\_RATE\*(1 - producer1/PRODUCER\_CARRYING\_CAPACITY) + prodL2 + prodU3 - min(producer1,max(0,1\*herbivore1)) - MIGRATION\_RATE\_PRODUCER\*(producer1-min(producer1,max(0,1\*herbivore1))));*

*producer2 += ((producer2 - min(producer2 ,max(0,1\*herbivore2)))\*PRODUCER\_BIRTH\_RATE\*(1 - producer2/PRODUCER\_CARRYING\_CAPACITY) + prodR1 + prodU4 - min(producer2,max(0,1\*herbivore2)) - MIGRATION\_RATE\_PRODUCER\*(producer2-min(producer2,max(0,1\*herbivore2))));*

*producer3 += ((producer3 - min(producer3 ,max(0,1\*herbivore3)))\*PRODUCER\_BIRTH\_RATE\*(1 - producer3/PRODUCER\_CARRYING\_CAPACITY) + prodD1 + prodL4 - min(producer3,max(0,1\*herbivore3)) - MIGRATION\_RATE\_PRODUCER\*(producer3-min(producer3,max(0,1\*herbivore3))));*

*producer4 += ((producer4 - min(producer4 ,max(0,1\*herbivore4)))\*PRODUCER\_BIRTH\_RATE\*(1 - producer4/PRODUCER\_CARRYING\_CAPACITY) + prodD2 + prodR3 - min(producer4,max(0,1\*herbivore4)) - MIGRATION\_RATE\_PRODUCER\*(producer4-min(producer4,max(0,1\*herbivore4))));*

*if(temp == "none")*

*{*

*cout << "Iteration[" << t << "] -Sectors-" << endl;*

*cout << setw(12) << "Organism" << setw(6) << " 1" << setw(6) << " 2" << setw(6) << " 3" << setw(6) << " 4" << setw(6) << " TOTAL" << endl;*

*cout << setw(12) << "Producers" << setw(6) << int(producer1) << setw(6) << int(producer2) << setw(6) << int(producer3) << setw(6) << int(producer4) << setw(6) << int(producer1 + producer2 + producer3 + producer4) << endl;*

*cout << setw(12) << "Herbivores" << setw(6) << int(herbivore1) << setw(6) << int(herbivore2) << setw(6) << int(herbivore3) << setw(6) << int(herbivore4) << setw(6) << int(herbivore1 + herbivore2 + herbivore3 + herbivore4) << endl;*

*cout << setw(12) << "Carnivores" << setw(6) << int(carnivore1) << setw(6) << int(carnivore2) << setw(6) << int(carnivore3) << setw(6) << int(carnivore4) << setw(6) << int(carnivore1 + carnivore2 + carnivore3 + carnivore4) << endl;*

*cout << "Press Enter to continue";*

*cin.get();*

*cout << endl << endl << endl;*

*}*

*else*

*infile << (producer1 + producer2 + producer3 + producer4) << " " << (herbivore1 + herbivore2 + herbivore3 + herbivore4) << " " << (carnivore1 + carnivore2 + carnivore3 + carnivore4) << endl;*

*}*

*}*

*double min(double a, double b)*

*{*

*if(a < b)*

*return a;*

*return b;*

*}*

*double max(double a, double b)*

*{*

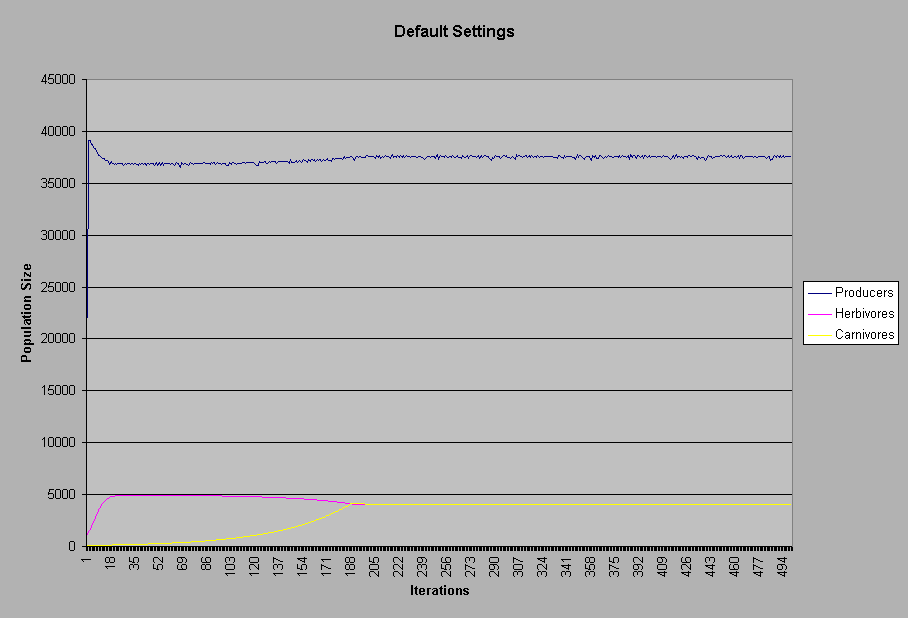
*if(a > b)*

*return a;*

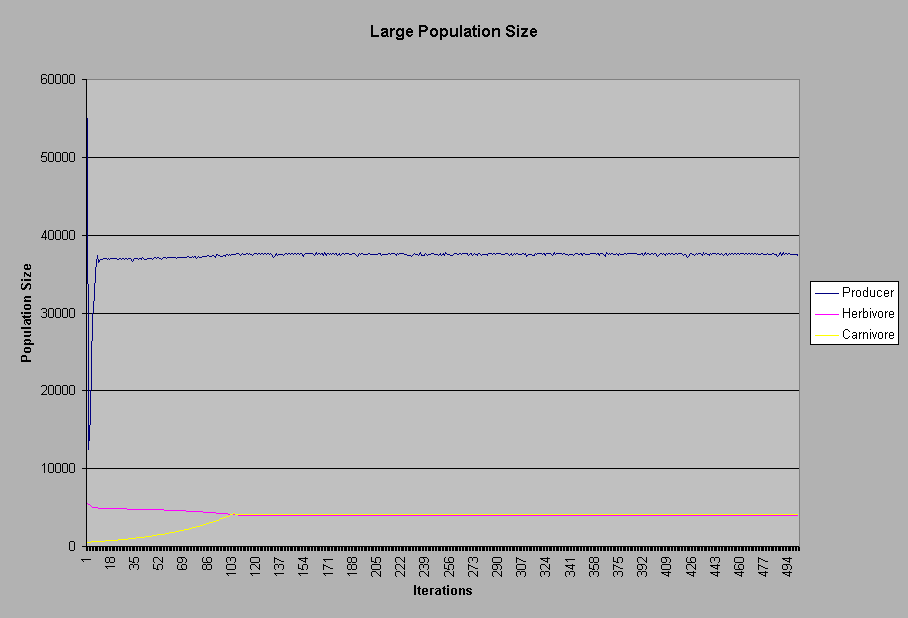
*return b;*

*}*

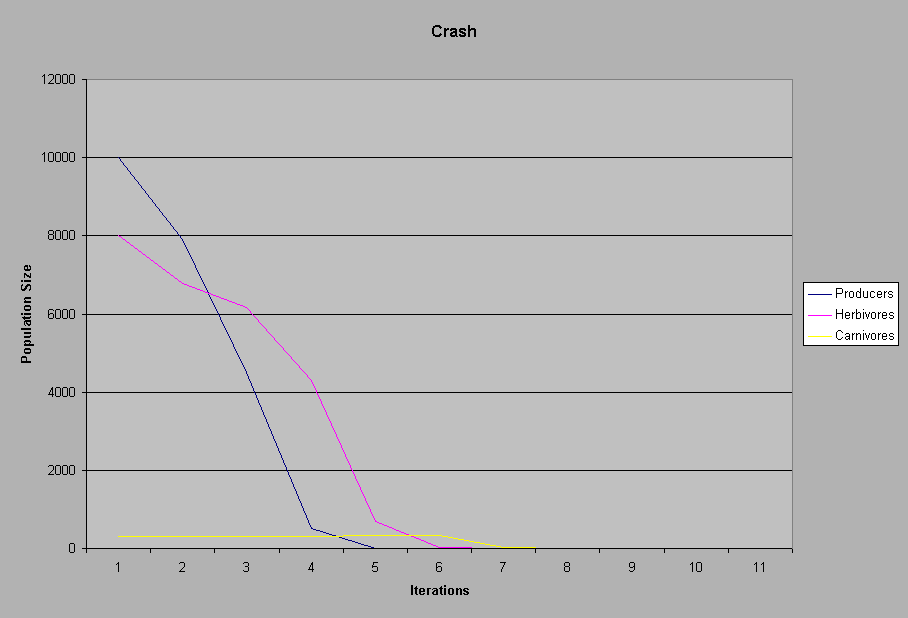
**Data and Observations**



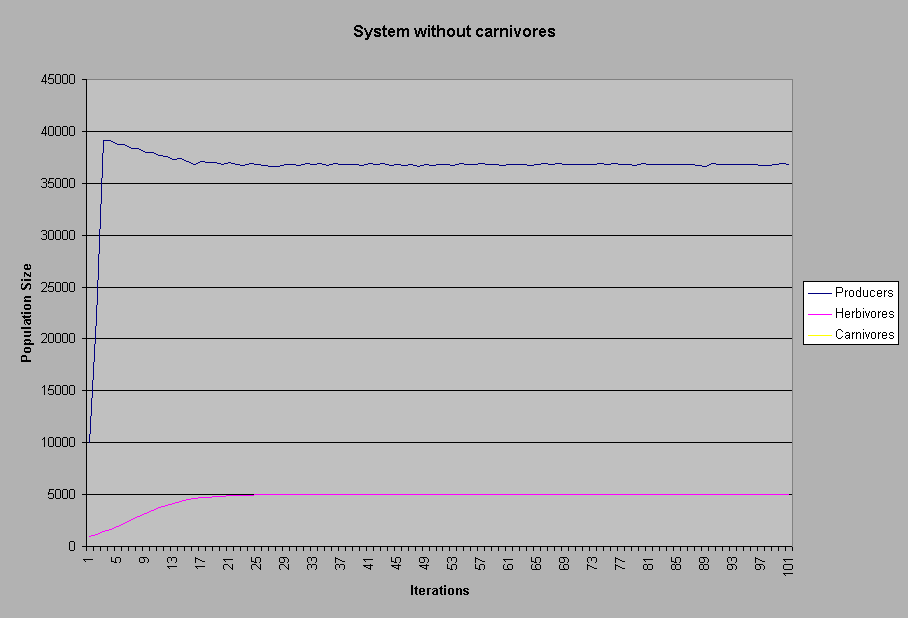
With default settings (Producers at 10,000, Herbivores at 1,000, and Carnivores at 100) the producers made a sharp increase towards its equilibrium point, while the herbivores moved in what looks like an exponential growth pattern followed by a slow recession. The carnivores appear to be growing in a logistic pattern.



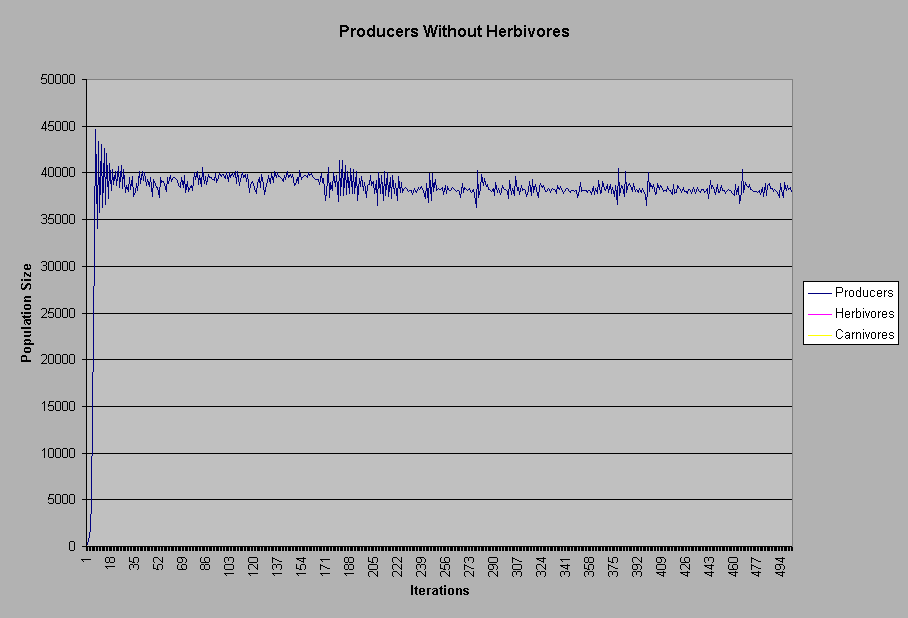
Starting out with a large population size yielded a sharp drop in producers, while herbivores and carnivores managed to stay rather constant.



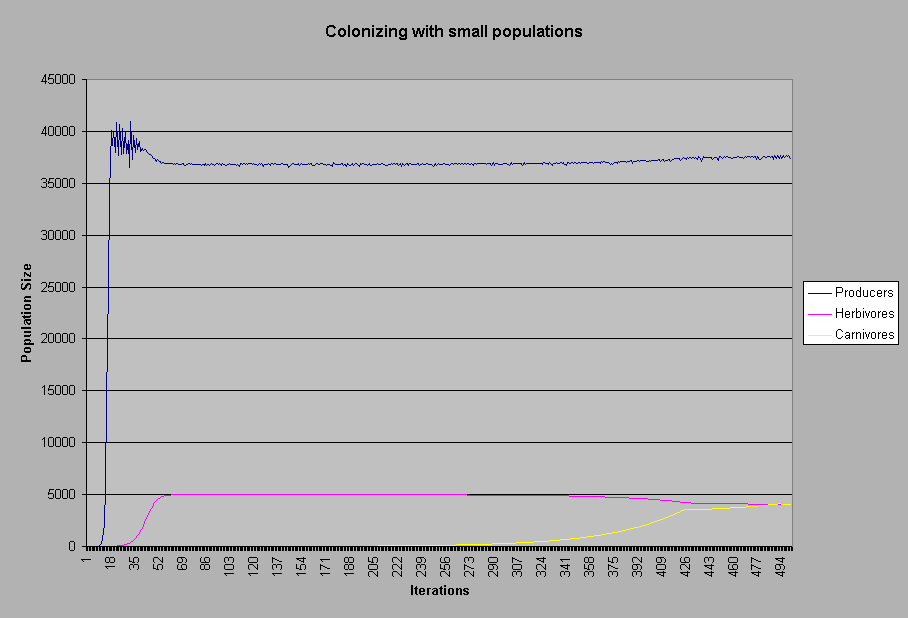
By setting Producer population levels at 10,000, Herbivores at 8,000, and carnivores at 200, I was able to get the 4 “bubble” ecosystem to “crash”. Notice how the producer population drops below the herbivores before the herbivore population starts to decrease as well. This is accurate as to what a real ecosystem would yield.



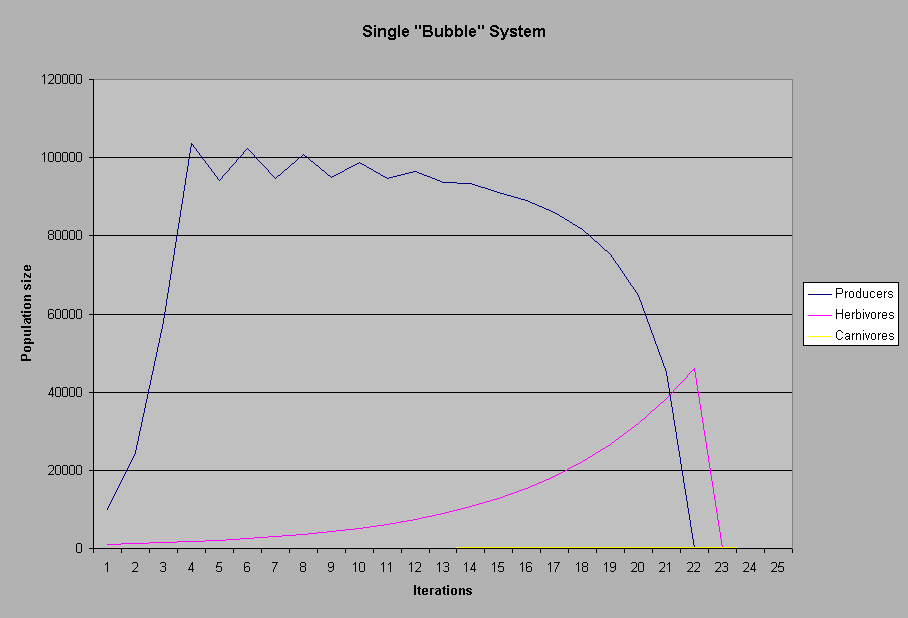
In this model, I tried to see what would happen if carnivores were taken out of the picture. Apparently, when carnivores are set to 0, the herbivores follow a logistic regression growth pattern while the producers grow exponentially past the carrying capacity, then fall down to meet it.



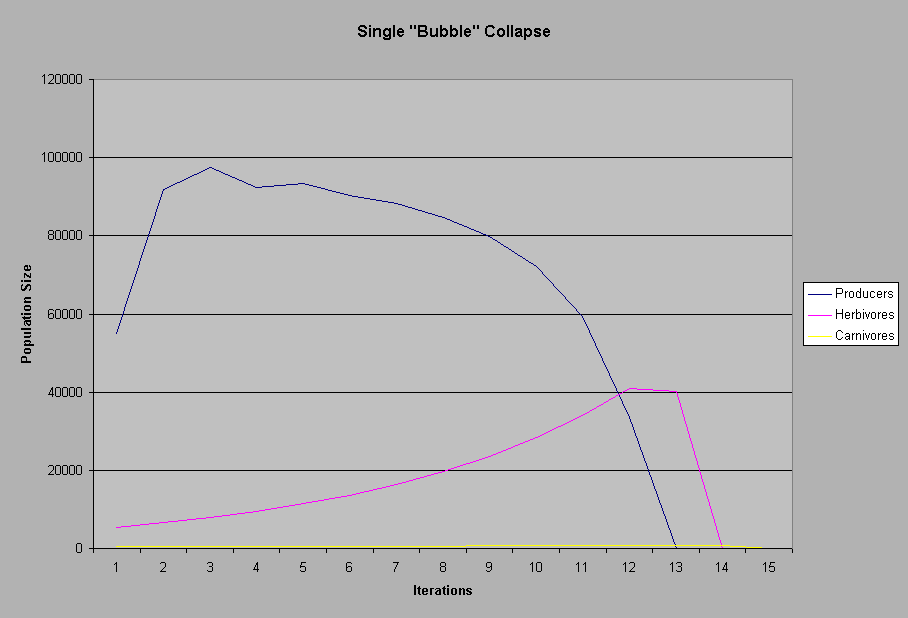
In this dramatic example, herbivores and, hence, carnivores, were removed from the population to see how producers would grow without being culled by the herbivores. Apparently, they grow with a lot more oscillations, as the population shoots way above the carrying capacity within a few iterations, then falls below it, then shoots back up.



In this example, I started the populations out very small to mimic what it would be like for the three species to colonize a new area with no other life. Producers were set at 2, herbivores at 1, and carnivores at 1. The producers and the herbivores reached their carrying capacity relatively quickly, while the carnivores struggled to reach the carrying capacity, eventually reaching it after nearly 500 iterations!



This model shows the general instability of the single “bubble” system. This is the single bubble system set at its default values. It is near impossible to get the single bubble system to reach an equilibrium. After only 25 iterations, the ecosystem crashed.



With this single bubble system I tried to get it to reach an equilibrium by increasing the amount of producers while keeping herbivore and carnivore populations rather low. However, the general instability of the single bubble system resulted in this collapse of the population, which ironically lasted only 14 iterations, much shorter than the previous default values.

**Analysis and Conclusion**

**Analysis:** The graphs clearly show that the population dynamics are relatively accurate. However, they are nowhere near precise, as predicting the true population is not feasible.

**Conclusion:** Based on the graphs of the data, I can say that the population dynamics can be predicted, however not very precisely. The computer program only created a reasonable approximation of the population sizes of the producers, the herbivores, and the carnivores.

**Recommendations**

For the benefit of future students, we will provide the following recommendations for those who would be willing to follow in our footsteps and create a computer simulation. Hopefully a student proficient enough in Java or C++ can finish where we left off and create a more accurate model.

**Where we ran into some trouble**

* **Equations**- In order to accurately simulate an ecosystem, equations are vital. In order to find these equations, it took a lot of research.
* **Be Reasonable-** Do not bite off more than you can chew. It is not very feasible to find equations to predict how abiotic factors will interact with biotic factors. We tried this. We were forced to give up that idea.
* **Where to get started?** - Pinpoint a very specific topic to model. Do not try and model something that is very broad.
* **Time is your friend** – Ultimately, manage your time. Unless you want to be working until 4 AM the night before the project is due, you must remember time is of the essence. The more time spent on the project, the better it will be.

**Ideas on expanding are experiment**

* Simulating the creek. Unfortunately we were not able to because of the difficulty of simulating abiotic factors. However, it may be possible just looking at biotic factors.
* Add more factors to the population model. The more organisms, the more realistic it is. Our equations had a problem where the herbivore death rate was somewhat off. This can give you misleading results.

***Information that will help you either (1) help you get an idea of what to simulate in a computer program or (2) expand on are program on the abiotic factors in an environment.***

* **Two URLs to take a look at:** <http://www.mathstat.usouthal.edu/~hitt/courses/590/population/population.html>

<http://rulbii.leidenuniv.nl/wwwkim/popdyn.html>

**Daily Log**

**April 17-** Created hypothesis and prediction for this project after the previous one was found to be not feasible.

**April 19-** Research concluded. Results showed that we were in big trouble. Began developing equations for the program

**April 20-** Talked with professor from UCI through Janell Richardson. Gained some insight about the evils of procrastination and also some equation ideas.

**April 23-** Much work was done. Program was completed and debugged. Brian wrote the introduction and Chris gathered data from several test runs.

**April 24-** Work concluded. Report finished.

Bibliography

Farrell, Mike. Global Carbon Cycle. (Online) available

http://www.esd.ornl.gov/iab/iab2-3.htm 22 April 2001

Hannon, Bruce. Spacial Predator-Prey Models (Online) Available

http://web.aces.uiuc.edu/aim/model/day3text2.htm 24 April 2001

Hitt, Richard. Population of dynamics. (Online) Available

http://www.mathstat.usouthal.edu/~hitt/courses/590/population/population.html

22 April 2001

International Baccalaureate Organization. energy in ecological systems. (Online)

Available

http://www.racerocks.com/pearson/ensy/syllabus/ensybooklet4/ensybooklet4.htm

20 April 2001

Kay, James. Ecosystem Intergrity. (Online) Available

http://www.fes.uwaterloo.ca/u/jjkay/pubs/envmgmt/index.html#t2

21 April 2001

McNaughton, S.J., Oesterhek, M., Frank D.A., & Williams, K.J. Ecosystem-

Productivity (Online) Available

http://syllabus.syr.edu/bio/mcnaughton/ecosyslvl.html 20 April 2001

Odum, E.P. Food Chains. (Online) Available

http://www.ultranet.com/~jkimball/BiologyPages/F/FoodChains.html 21 April 2001

Pidwirny, Michael. Ecosystem Productivity. (Online) Available

http://geog.ouc.bc.ca/conted/onlinecourses/geog\_210/contents/210~2~2~6.html

19 April 2001

Predenergast, John & Anderson, Rob. Global Population Dynamics Database.

(Online) Available http://cpbnts1.bio.ic.ac.uk/gpdd/ 15 April 2001

Schawartz, Elizabeth. Water Cycle. (Online) Available

http://mbgnet.mobot.org/fresh/cycle/index.htm 20 April 2001

Senvenster, Dr. J.G. Population Dynamics Models interactive. (Online) Available

http://rulbii.leidenuniv.nl/wwwkim/popdyn.html 16 April 2001

Sharov, Alexei. Exponential Model. (Online) Available

http://www.gypsymoth.ento.vt.edu/~sharov/PopEcol/lec5/exp.html

22 April 2001

Sharov, Alexei. Logistic Model. (Online) Available

http://www.gypsymoth.ento.vt.edu/~sharov/PopEcol/lec5/logist.html 23 April 2001

Thiel, Eric. Creek Watch Home Page. (Online) Available

http://www.pleasanton.k12.ca.us/amador/Creek/AP98/AP98.html 10 April 2001