Modeling of Predator-Prey Relationships

New Mexico

Supercomputing Challenge

Final Report

April 4, 2012

57

La Cueva

Team Members

Robert McDaniels

Dennis Huang

Eli Echt-Wilson

Teacher

Sam Smith

Executive Summary:

In nature, predator-prey relationships are enormously important to maintaining the balance of populations. Predators regulate the prey's population, which tend to be large in numbers, and prey limit the predators when over predation reduces their numbers enough that the predators can no longer sustain population growth due to starvation.

Problem Definition:

The goal of this project is to find the optimal environment for a predator-prey food chain where there is a perfect balance between the predator, prey, and food source. This kind of model can be applied to any situation in which an organism is killed in a predictable way, such as deforestation and poaching, and can be used to determine, as an example, what parameters to change to prevent extinction of an endangered animal.

Problem Solution:

The simulation will use standard exponential growth models with an additional death rate, $dP/dt=kP(1-P/K)-D$, where dP/dt is the rate of population growth/decline, k is some constant, P is the population, K is the carrying capacity, and D is the death rate. The "constants" in this equation will be adjusted for each iteration based on the populations of each organism (plant, prey, and predator). To determine if the model has reached stability, the simulation will record the populations over time and stop when it has reached the median of population eight consecutive times.

Expected Results:

After programming the model and iterating through the possible parameters for the system, a set of optimal parameters to sustain a healthy population can be determined. These can be used to determine how to save endangered animals, sustain a good economic "ecosystem", and can be applied to anything in which two different entities are pitted against each other and cause the other's success to be limited.

Conclusion:

Predator prey relationships define how many ecosystems function. Endangered animals are crucial to many environments around the world, and are often keystone species, meaning the ecosystem they live in becomes unstable without them. If applied to global markets, a simple model simulating a healthy economy could be established. This could be used to determine what “growth” and “death” rates can be used to sustain an ideal economy, and economic policies could be adjusted based off of these rates.

Code:

```
import java.util.*;

public class Actor{
    /* for now, Actor is just an empty class because there are no graphics - fill in later */
    public Actor(){}
}

public class Animal extends Actor{}

public class Grass extends Actor{
public int energy;

public void die(){
energy=0;
}
}

public class Sheep extends Animal{
    //Note: These “constants” should NOT be final so they can be changed for each test
public static double EAT_RATE;
public static double COPY_RATE;
public int energy;

public void eat(Grass grass){
    energy+=grass.energy;
    grass.die();
}
}
```

```java
public class Wolf extends Animal{
    public static double EAT_RATE;
    public static double COPY_RATE;
public int energy;

public void eat(Sheep sheep){
    energy+=sheep.energy;
    sheep.die();
}
}

public class Experiment{
    private ArrayList<Sheep> sheep;
    private ArrayList<Wolf> wolves;
    private ArrayList<Grass> grass;

    public Experiment(int nsheep,int nwolf,int ngrass){
        sheep=new ArrayList<Sheep>(nsheep);
        for(int i=0;i<sheep.length;++i){
            sheep.set(i,new Sheep());
        }
        wolves=new ArrayList<Wolf>(nwolf);
        for(int i=0;i<wolves.length;++i){
            wolves.set(i,new Wolf());
        }
        grass=new ArrayList<Grass>(ngrass);
        for(int i=0;i<grass.length;++i){
            grass.set(i,new Grass());
        }
    }

    public static int randrange(int maxval){
        return (int)(Math.random()*maxval);
    }
```

```
public void run_iteration(){
    int x=0;
    for(int i=0;i<sheep.length;++i){
        if(Math.random()<=Sheep.EAT_RATE){
            x=randrange(grass.length)
            sheep.get(i).eat(grass.get(x));
        }
        if(Math.random()<=Sheep.COPY_RATE){
            sheep.add(new Sheep());
            sheep.get(i).energy-=Sheep.COPY_ENERGY;
        }
    }
    for(int i=0;i<wolves.length;++i){
        if(Math.random()<=Wolf.EAT_RATE){
            x=randrange(sheep.length)
            wolves.get(i).eat(sheep.get(x));
            sheep.remove(x);
        }
        if(Math.random()<=Wolf.COPY_RATE){
            wolves.add(new Wolf());
            wolves.get(i).energy-=Wolf.COPY_ENERGY;
        }
    }
    for(int i=0;i<grass.length;++i){
        grass.get(i).grow();
    }
}

public void run(){
    while(true){
        run_iteration();
    }
```

```
    }
}
```

Most Significant Achievement:

We wrote code!

References:

- http://home.messiah.edu/~deroos/csc171/PredPrey/PPIntro.htm
- http://www.globalchange.umich.edu/globalchange1/current/lectures/predation/predation.html
- http://www.stolaf.edu/people/mckelvey/envision.dir/lotka-volt.html
- http://serc.carleton.edu/quantskills/methods/quantlit/popgrowth.html
- http://ugrad.math.ubc.ca/coursedoc/math100/notes/mordifeqs/logistic.html