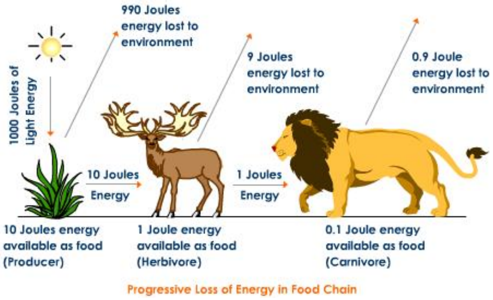
Computational Physics Group Project: Ecosystem: predator and prey

David Hicks
Weiyao Ke
Shagun Maheshwari
Fan Zhang

April 13, 2015

Population interaction of predator and prey in eco-system



1000 Joules of Light Energy
990 Joules energy lost to environment
9 Joules energy lost to environment
0.9 Joule energy lost to environment
10 Joules Energy
1 Joules Energy
10 Joules energy available as food (Producer)
1 Joule energy available as food (Herbivore)
0.1 Joule energy available as food (Carnivore)
Progressive Loss of Energy in Food Chain

Figure : default

Population interaction of predator and prey in eco-system

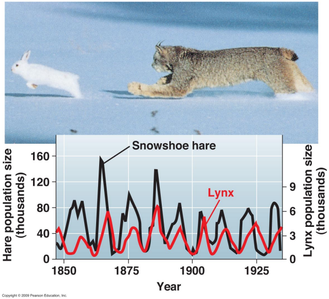


Figure : http://www.anselm.edu/homepage/jpitocch/genbi101/ecology1intropops.html

A simplified determinsitic mode: L-V equation

Simulation of a eco-system with predator and prey

A simulation keep the essential nature of the interaction between and within the species, and predict the evolution of population step by step.

- Both predator and prey reproduces when they reach the age of reproduction

Simulation of a eco-system with predator and prey

A simulation keep the essential nature of the interaction between and within the species, and predict the evolution of population step by step.

- Both predator and prey reproduces when they reach the age of reproduction
- Predator feeds on prey.

Simulation of a eco-system with predator and prey

A simulation keep the essential nature of the interaction between and within the species, and predict the evolution of population step by step.

- Both predator and prey reproduces when they reach the age of reproduction
- Predator feeds on prey.
- Predator and prey will die out if maximum age is reached or starved for enough long time

Simulation of a eco-system with predator and prey

A simulation keep the essential nature of the interaction between and within the species, and predict the evolution of population step by step.

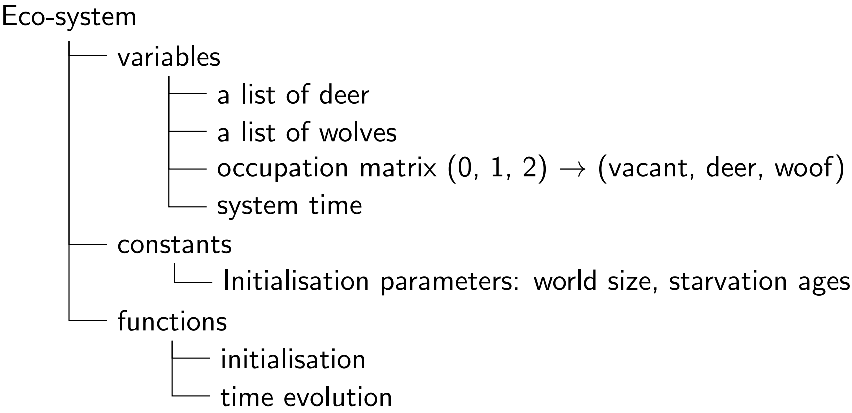
- Both predator and prey reproduces when they reach the age of reproduction
- Predator feeds on prey.
- Predator and prey will die out if maximum age is reached or starved for enough long time
- However, simulation is a random process and change the deterministic nature of LV equation (more realistic).

Structural setup

Animal Class → Deer/ Wolf

- variables
  - presentation position
  - previous position
  - reproducing age variable
  - starving age variable
- constants
  - starvation age
  - reproduction age
- functions
  - check status: live/dead
  - check maturity: procreate/not

Structural setup



- Eco-system
  - variables
    - a list of deer
    - a list of wolves
    - occupation matrix (0, 1, 2) → (vacant, deer, woof)
    - system time
  - constants
    - Initialisation parameters: world size, starvation ages
  - functions
    - initialisation
    - time evolution

Initialisation

A sanity simulation requires several constrains on the initialisation of parameters.

- Reproduction age of predators must be larger than their starvation age. (Or else wolf can sustain themselves ...)

Initialisation

A sanity simulation requires several constrains on the initialisation of parameters.

- Reproduction age of predators must be larger than their starvation age. (Or else wolf can sustain themselves ...)
- Starvation age of the deer is extremely large. (Always enough plants!)

time evolution (All things are done serially!)

- Step1: loop over wolves and deer and increase ages by 1; then check if they are alive. Only live animals are kept in list for operations below.

time evolution (All things are done serially!)

- Step1: loop over wolves and deer and increase ages by 1; then check if they are alive. Only live animals are kept in list for operations below.
- Step2: loop over wolves and check neighbours. If deer nearby, capture a random one and move to that position. If no deer but vacancies around, move to a random location. If a mature wolf's present location differs from previous location, deposits a new-born at its previous location.

time evolution (All things are done serially!)

- Step1: loop over wolves and deer and increase ages by 1; then check if they are alive. Only live animals are kept in list for operations below.
- Step2: loop over wolves and check neighbours. If deer nearby, capture a random one and move to that position. If no deer but vacancies around, move to a random location. If a mature wolf's present location differs from previous location, deposits a new-born at its previous location.
- Step3: loop over deer and delete those whose location are now occupied by wolves (via the occupation matrix)

time evolution (All things are done serially!)

- Step1: loop over wolves and deer and increase ages by 1; then check if they are alive. Only live animals are kept in list for operations below.
- Step2: loop over wolves and check neighbours. If deer nearby, capture a random one and move to that position. If no deer but vacancies around, move to a random location. If a mature wolf's present location differs from previous location, deposits a new-born at its previous location.
- Step3: loop over deer and delete those whose location are now occupied by wolves (via the occupation matrix)
- Step4: loop over deer and check neighbours. If vacancies around, move to a random location. If a mature deer's present location differs from previous location, deposits a new-born at its previous location.

time evolution (All things are done serially!)

- Step1: loop over wolves and deer and increase ages by 1; then check if they are alive. Only live animals are kept in list for operations below.
- Step2: loop over wolves and check neighbours. If deer nearby, capture a random one and move to that position. If no deer but vacancies around, move to a random location. If a mature wolf's present location differs from previous location, deposits a new-born at its previous location.
- Step3: loop over deer and delete those whose location are now occupied by wolves (via the occupation matrix)
- Step4: loop over deer and check neighbours. If vacancies around, move to a random location. If a mature deer's present location differs from previous location, deposits a new-born at its previous location.
- Step5: go back to Step1 and increase system time by 1

parameter scanning

Parameter Search

5 parameters to test (5-D parameter space)

- **Initial population of deer**
- **Initial population of wolves**
- Reproduction age of deer
- Reproduction age of wolf
- Starvation "age" of wolf

Reduce to 4 dimensions (4-D)

- **Ratio of initial populations : Size of point**
- Reproduction age of deer : x-axis
- Reproduction age of wolf : y-axis
- Starvation "age" of wolf : z-axis

Results of Full Parameter Search

Results of Restricted Parameter Search

Fix initial population ratios

Ecosystem at Equilibrium

Parameters used: ....