Computational Physics Group Project: Ecosystem: predator and prey

David Hicks
Weiyao Ke
Shagun Maheshwari
Fan Zhang

April 13, 2015

Introduction to eco-system modelling

Implementation of the simulation

Results and discussion

Population interaction of predator and prey in eco-system

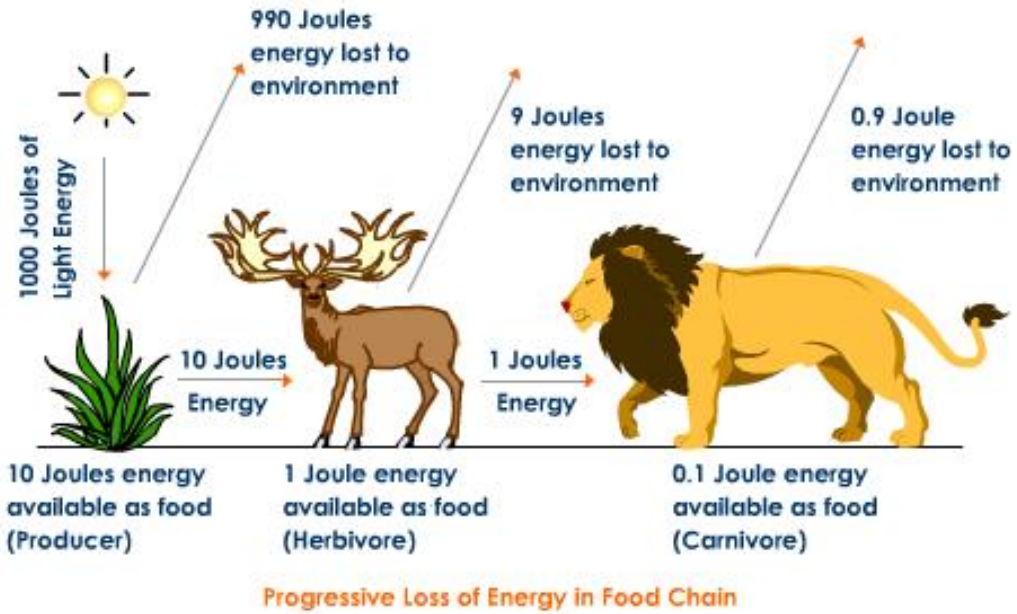


Figure: default

Population interaction of predator and prey in eco-system

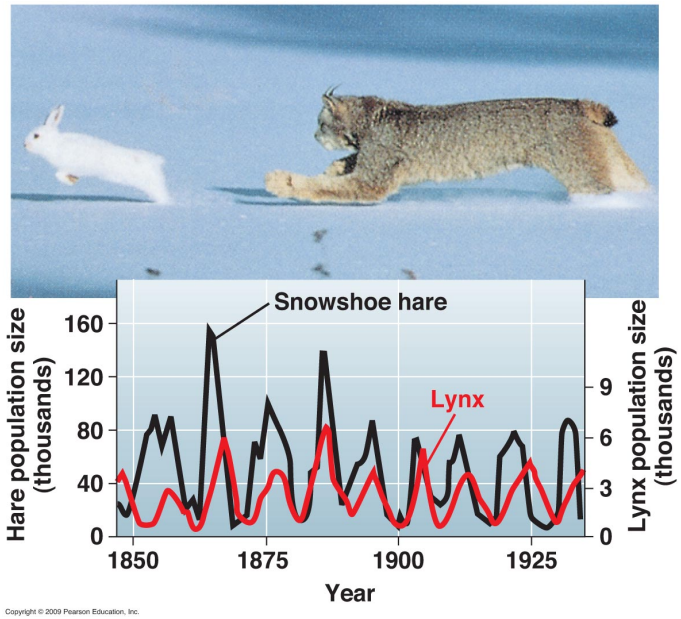


Figure:
http://www.anselm.edu/homepage/jpitocch/genbi101/ecology1intropops.html

A simplified determinsitic mode: L-V equation

The dynamics of biological systems consist of one predator and one prey can be described by Lotka-Volterra equations:

$$\begin{aligned} \frac{dx}{dt} &= \alpha x - \beta xy = x(\alpha - \beta y) \\ \frac{dy}{dt} &= -\gamma y + \delta xy = -y(\gamma - \delta x) \end{aligned}$$

Where, x is the number of prey, y is the number of predator, $\frac{dx}{dt}$ and $\frac{dy}{dt}$ represent the growth rates of two populations, and α, β, γ and δ are parameters describing the interaction of two species. When the biological system has reached eco-equilibrium, the number of predator and prey are supposed to be either situation below.

$$x = 0, y = 0$$

$$\gamma \qquad \alpha$$

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) $\rightarrow$ (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!)

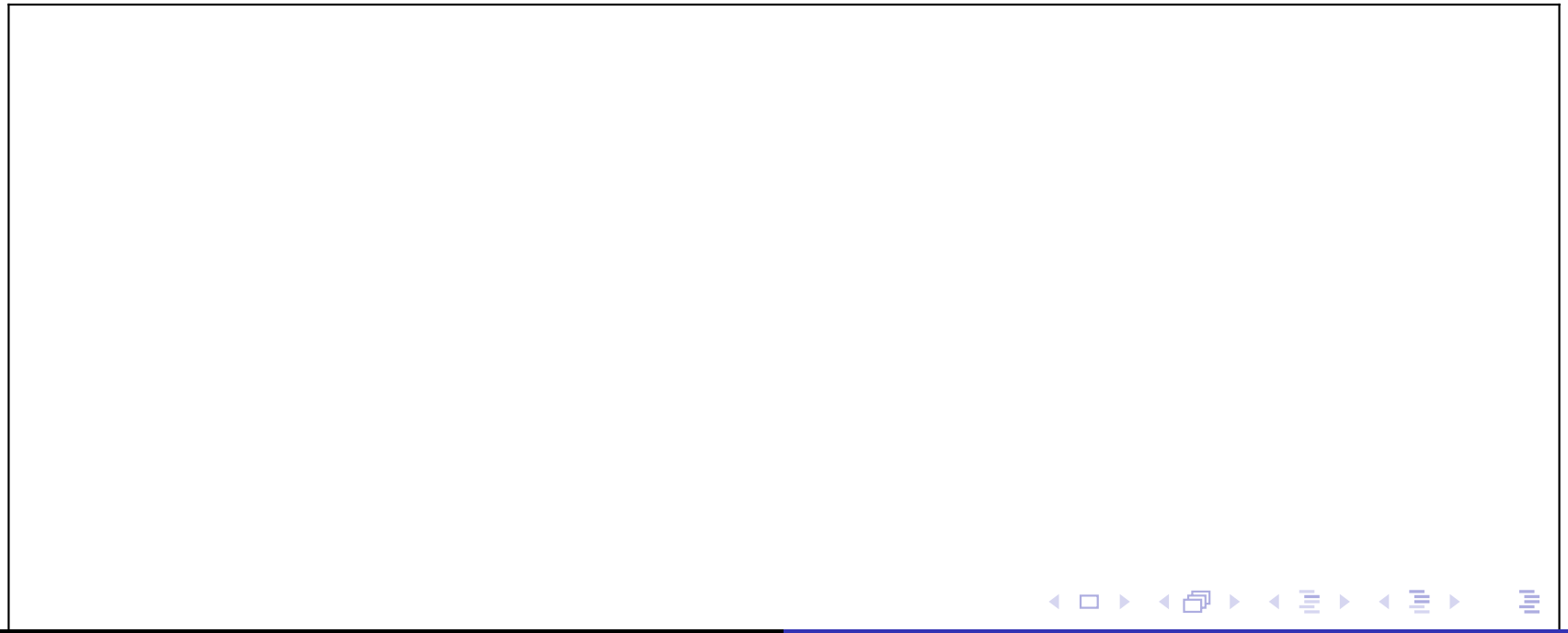
- Step1: loop over wolves and deer and increase ages by 1; then check if they are alive. Only live animals are kept in list for operations below.
- Step2: loop over wolves and check neighbours. If deer nearby, capture a random one and move to that position. If no deer but vacancies around, move to a 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.

Simulation of a eco-system with predator and prey

We set up a $N \times N$ grid and simulate the eco-system with L-V equation.
Evolution Step 1: check wolf and deer population, increase its age, and see whether a single animal has starved to death.
Evolution step 2: evolution of wolves:



Evolution step 3

Evolution of deers:

- 1. Delete all unfortunate deers.

Evolution step 3

Evolution of deers:

- 1. Delete all unfortunate deers.
- 2. Evolution of live deers.

parameter scanning

Parameter Search

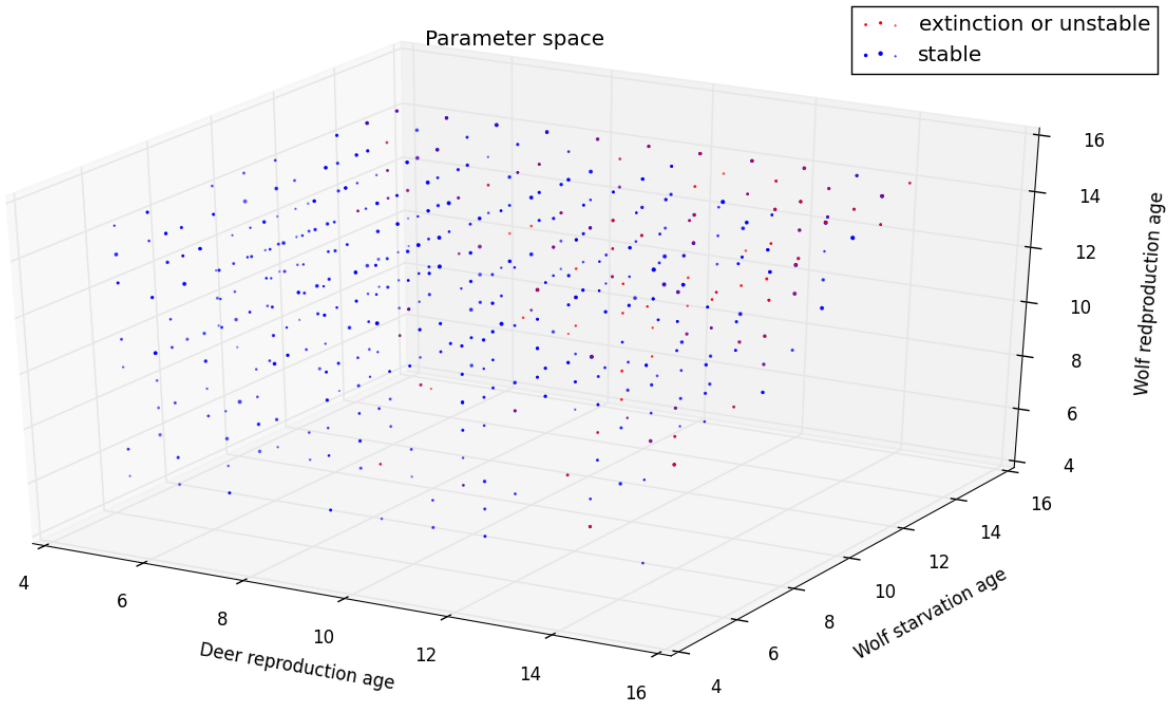
5 parameters to test (5-D parameter space)

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

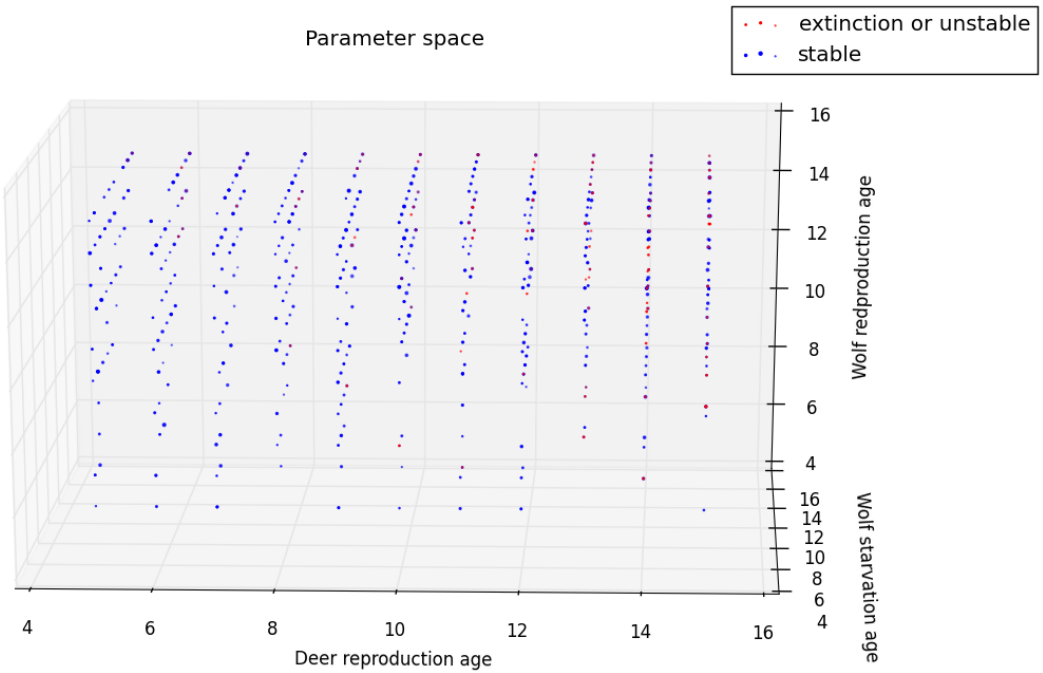
Reduce to 4 dimensions (4-D)

- **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

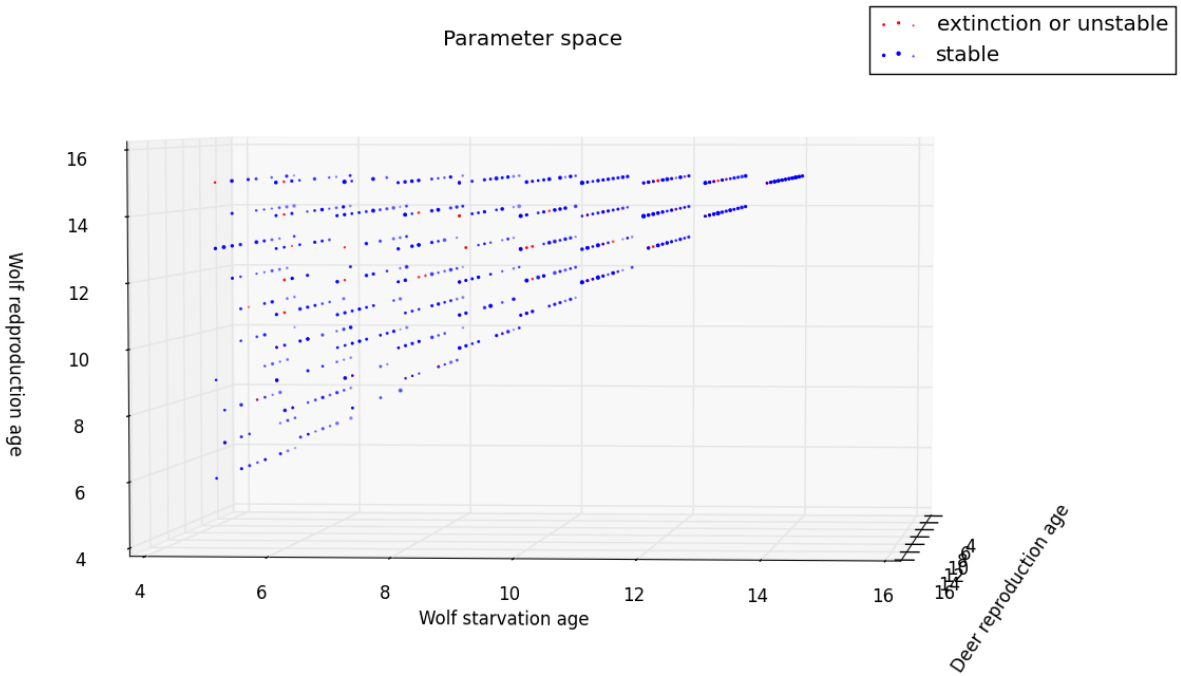
Results of Full Parameter Search



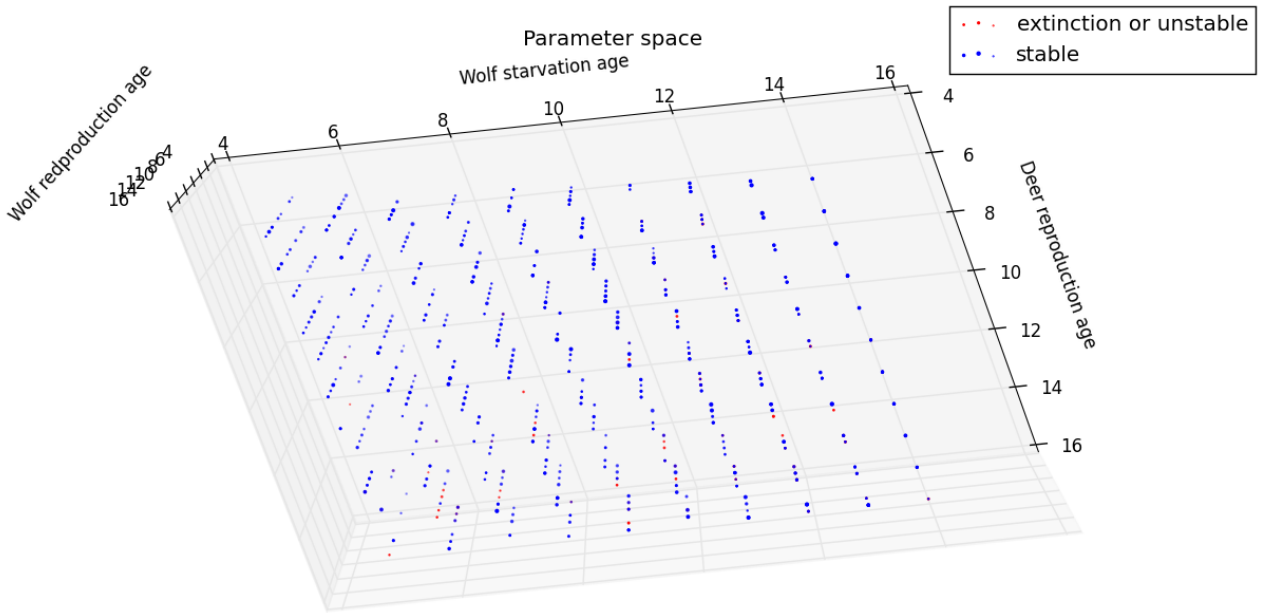
Results of Full Parameter Search



Results of Full Parameter Search

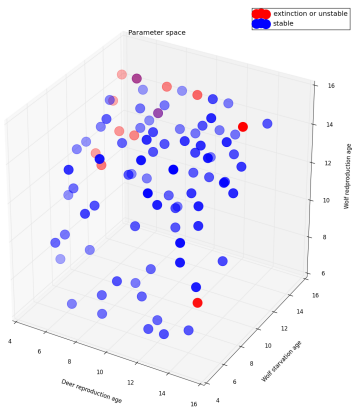


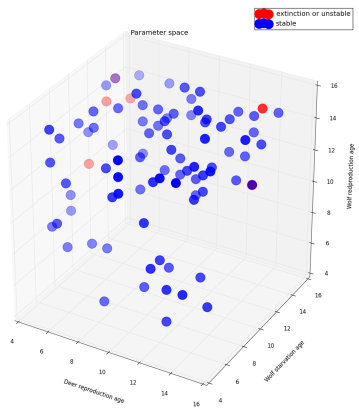
Results of Full Parameter Search

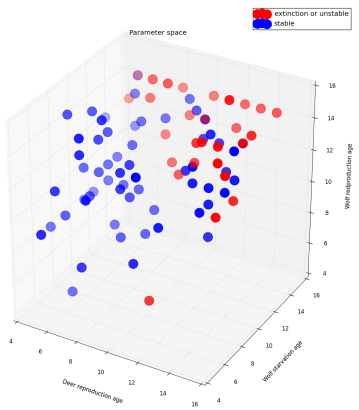


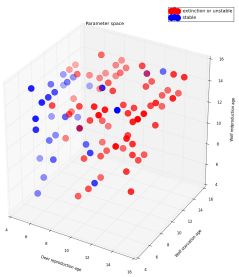
Results of Restricted Parameter Search

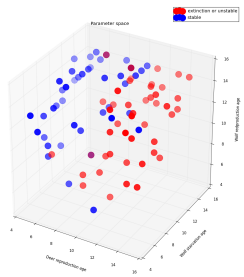
Fix initial population ratios











Ecosystem at Equilibrium

Parameters used:

- Initial number of deer: 2,500
- Initial number of wolves: 250
- Deer reproduction rate: 5
- Wolf reproduction rate: 14
- Wolf starvation rate: 11

Animation Time!