

Computational Physics Group Assignment 2

Predator-Prey Model

Christopher Flower, Long Li, Shen Yan, Wenzhe Yu

April 3, 2016

1 Introduction

In this work, the ecosystem of shark and fish is simulated by a predator-prey model using Python code.

2 Method

The predator-prey model is used to simulate the dynamics of biological systems in which two species, e.g. shark and fish, interact. Mathematically, the population of the prey and the predator can be described by non-linear first order differential equations, the Lotka-Volterra equations:

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

where x and y are the number of prey and the number of predators. The prey growth term αx leads to an exponential growth of prey population in absence of predators. The predator loss term $-\gamma y$ leads to an exponential decay of predator population in absence of prey. The prey loss term $\beta y x$ and the predator growth term $\delta x y$ depend both on the number of predators and the number of prey.

The Lotka-Volterra equations have periodic solutions with the maxima of predator population shifted 90° compared to prey population. At any given time the system is in a limit cycle and/or will reach a stable periodic solution. Extinction of species occurs when the population becomes too small to recover.

In the Monte-Carlo implementation of the predator-prey model, the ecosystem of shark and fish is set up with a 2-dimensional array with periodic boundary conditions, i.e. a torus world is considered. The parameters of this system include initial number of shark, initial number of fish, number of time steps for fish and shark to procreate, respectively, and the number of time steps for shark to die of starvation.

At each time step, shark and fish behave following several rules:

Fish: swim and breed

- for each fish make a list of available neighboring positions; pick one at random; move fish there; increment age.
- if fish reaches breeding age, put a new fish at (i,j) and set both fish age 0.

Shark: hunt and breed

- for each shark make a list of adjacent fish; pick one at random; move shark there and erase that fish; reset starving time; increment age.
- if no fish around, make a list of available neighboring positions; pick one at random; move shark there; increment age.
- if shark reaches breeding age, put a new shark at (i,j) and reset age to 0 for both.
- if shark reaches starvation age, erase shark.

The following arrays are used to keep track with the behavior of shark and fish:

Fish(i,j): represents presence or absence of a fish at (i,j). -1 if no fish; otherwise the age of the fish.

Shark(i,j): represents presence or absence of a shark at (i,j), similar to **Fish**(i,j).

FishMove(i,j): holds a record whether fish at (i,j) has been moved in the current time step.

SharkMove(i,j): holds a record whether shark at (i,j) has been moved in the current time step.

SharkStarve(i,j): stores the time since shark at (i,j) last ate.

3 Results and Discussions

4 Conclusions

The source code of this work is downloadable online at github.com/vyu16/PHY566-DUKE.

References

- [1] Notes of *Computational Physics* by Prof. S.A. Bass.
- [2] A.K. Dewdney, Sharks and fish wage an ecological war on the toroidal planet Wa-Tor, 1984.