PHY566 Group Project 2b
Predator Prey Model

Team 2
Chris Flower, Long Li, Shen Yan, and Wenzhe Yu

04/08/2016

Introduction

Here we are asked to simulate a shark/fish ecosystem via predator prey model

Formally, predator prey models are those that obey the Lotka-Volterra equations:

$$\frac{dx}{dt} = x(\alpha - \beta y)$$

$$\frac{dy}{dt} = -y(\gamma - \delta x)$$

Here, x represents prey, y represents predators

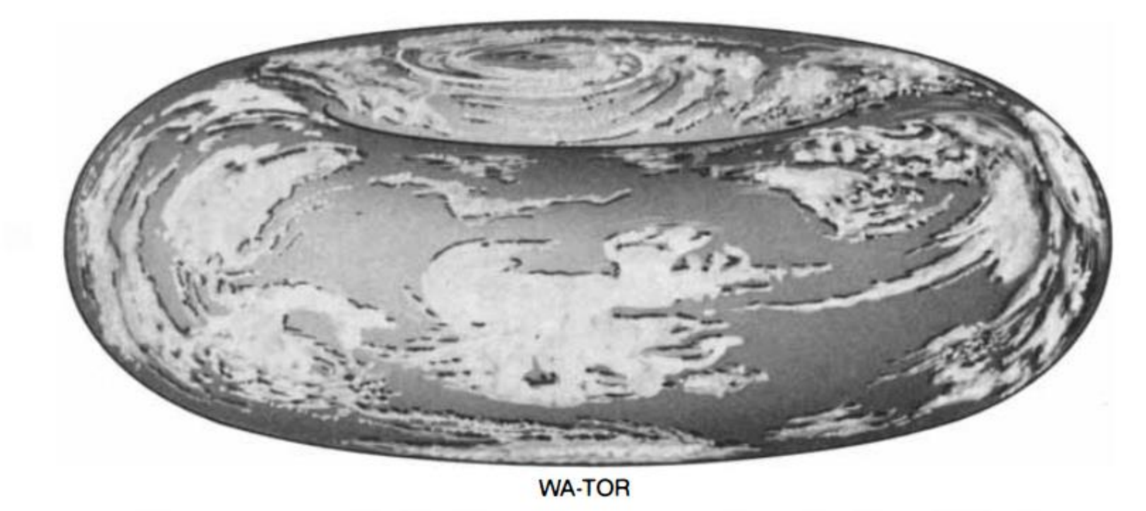
Note that the prey loss term and predator growth term

both depend on number of prey and predators.

Introduction

- These equations have periodic solutions (as expected) with population peaks shifted 90 degrees out of phase. (Note: extinction can of course occur.)

- To implement this model, a toroidal, 2 dimensional world was constructed.



A.K. Dewdney, Sharks and fish wage an ecological war on the toroidal planet Wa-Tor, 1984.

Introduction

The following rules governed the dynamics of the program:

Fish:

Move randomly to available adjacent spaces one at a time.

If fish reach breeding age, a new fish is born and age is reset.

Sharks:

If a fish is next to a shark, shark moves there and eats it. Otherwise, sharks move randomly to unoccupied locations.

If breeding age, sharks reproduce. If a shark has not eaten in a certain number of steps however, that shark starves and dies.

Algorithm

Five parameters: nfish(3000), nsharks(300), fbreed(4), sbreed(10) and sstarve(4). From Wa-Tor.

Three matrices: fish[i,j] (its value tells the age of fish. 0 means no fish), shark[i,j] and starve[i,j]. i in range(L), j in range(W)

Algorithm

In each time step, perform one move( ) function.

Fish move first: tmp=copy(fish), tmp will be looped so that every fish moves only once. However, only fish[i,j] will be modified.

checkandgetrandomdirection_fish(i,j) returns a position [inew,jnew] that fish can move in. Avoid sharks and fish. 2D random walk.

If fish age > fbreed, a new fish is born at [i,j] and renew both fish's ages.

Algorithm

Sharks move next: tmp=copy(shark)

checkandgetrandomdirection_stof(i,j) returns a position [inew,jnew] that has a fish. No fish will return [-1,-1] and if so, starve[i,j] will be checked. If starve[i,j] < shark starve age, do the same thing as the fish--checkandgetrandomdirection_shark(i,j). Otherwise the shark will be deleted.

If there is a fish, eat it and renew starve[inew,jnew]. Also check breeding age.

move( ) function ends.

‘for loop’ can see the change of fish[i,j] and shark[i,j].

Results

Initial parameters:

Grid size = 100x80

fish = 3000

sharks = 300

fishbreed = 4

sharkbreed = 10

sharkstarve = 4

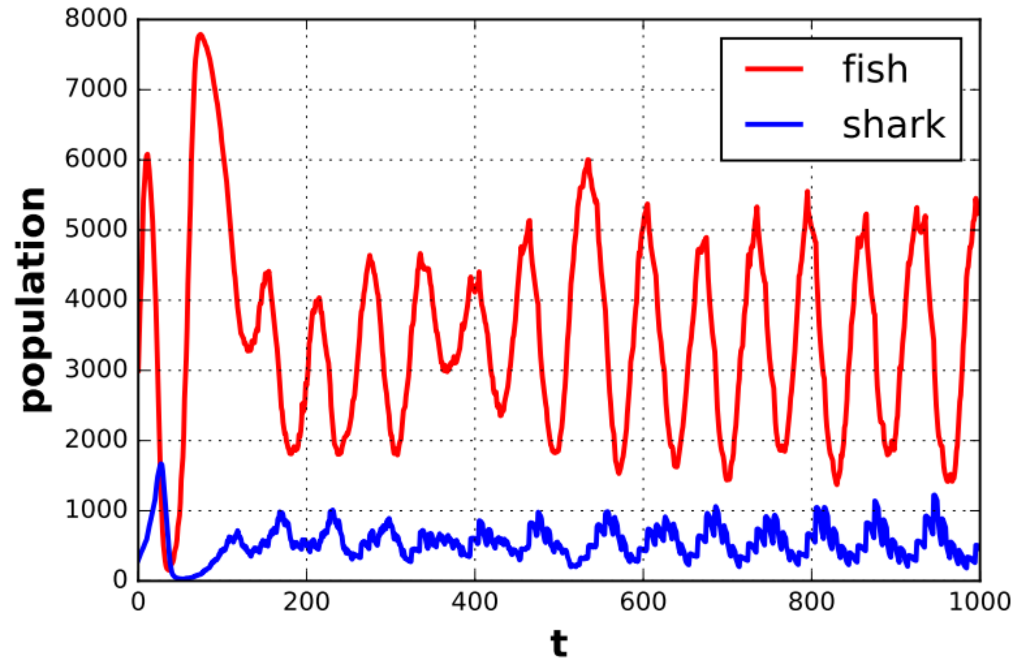
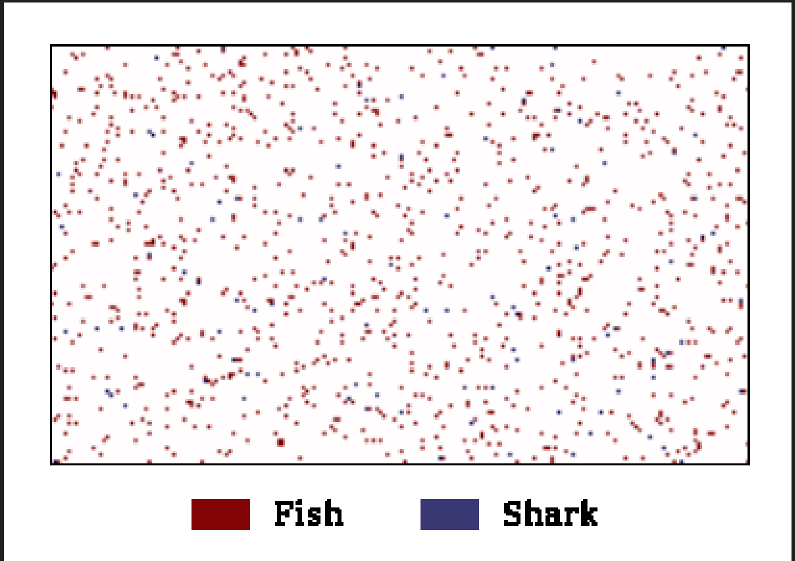
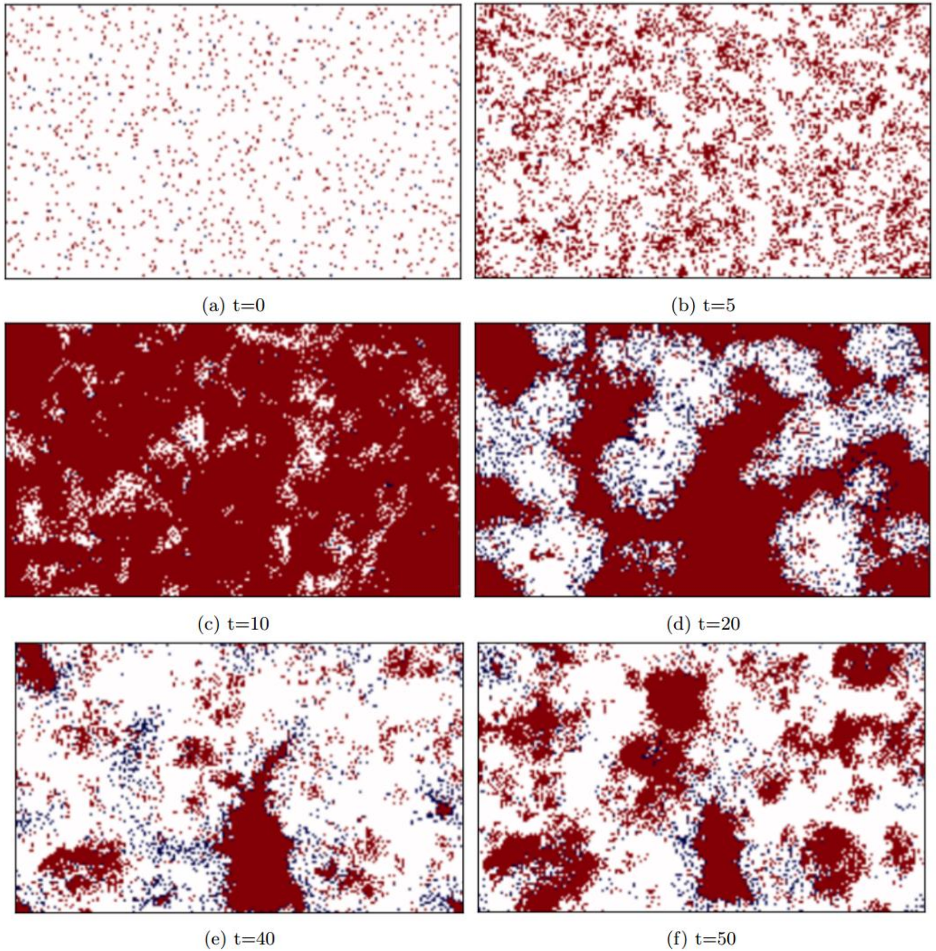


Figure 1: Fish and Shark population as a function of time.

Results

Snapshots at various t




(a) t=0

(b) t=5

(c) t=10

(d) t=20

(e) t=40

(f) t=50

Figure 2: Configurations of the 2D ecosystem. Red: fish. Blue: Shark.

Thanks!

Source code:

github.com/vyu16/PHY556-DUKE

Questions?