

# ELEC 1908 2019

## Project 1: The game of life

Due Sunday Feb. 24th at midnight by email to [tjs@doe.carleton.ca](mailto:tjs@doe.carleton.ca)

The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, live or dead. Every cell interacts with its eight neighbors, which are the cells that are directly horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

- Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.
- Any live cell with more than three live neighbours dies, as if by overcrowding.
- Any live cell with two or three live neighbours lives on to the next generation.
- Any dead cell with exactly three live neighbours becomes a live cell.

More info see wiki “Conway’s Game of Life”

## Part I

- You are given two matlab programs that implement a basic “Game of Life”. One is a fairly straightforward implementation and the other an attempt to make it as small as possible. Both implement the following:
  - Uses a  $n \times n$  grid (say 50x50) – start small when debugging
  - Initial random distribution of cells – uses the “rand” function!
  - Periodic boundary conditions to make it “infinite”
  - Uses matrices to represent the grid.
  - Loops over the life time – number of iterations
  - Uses the “spy” function to plot (replot) the matrix every iteration.
- Form groups of 2-3.
- Make a “.m” file of both programs and keep them! Run, Test and understand both programs.
- Create a commented version of the small program that explains how each line works.

**This is the small program:**

```
function game(n,m,t)

B = round(rand(n,m));
for time=1:t
A = B;
for i=1:n
for j=1:m
cola = [i-1,i,i+1];
cola(cola<1) = n;
cola(cola>n) = 1;
colb = [j-1,j,j+1];
colb(colb<1) = m;
colb(colb>m) = 1;
R=sum(sum(A(cola,colb)));

B(i,j) = ((R==3 || R==4) && A(i,j)) || ( R==3 && ~B(i,j));
end
end
spy(B);
pause(0.05);
end
```

Both programs can be found on the web page.

## Part II: Statistics

Pretend you are a biologist and obtain some information about your fauna.

- Modify the program so you can run multiple simulations. (You will want to turn off the “spy()” so that program runs quickly).
- Collect data on each simulation. (Number of live cells as function of time – population)
- Plot population as a function of time for a set of single simulations.
- For multiple simulations obtain the averages for the population and standard deviations. Look for the appropriate Matlab commands. Plot them as a function of time.
- Try varying inputs such as size, length of simulation. How does this effect your results.
- Any conclusions that you can make?
- Code in the ability to have “sources” – cells that are always alive – and “sinks” – cells that are always dead. Show some interesting situations. How does this change your statistics?
- What happens to the statistics if you remove your “periodic boundaries”?
- What happens to the statistics if you change the “rules”

## Part III: Populations and evolution

- Modify the program so that there are two species and look at:
  - The effect of whether the two species can co-exist on the same spot.
  - Add in aggressive tendencies ie if a individual of species 1 has more then N neighbours of species 2 he dies. Vary N and see what happens to the population statistics.
  - Investigate the effect of asymmetric aggression.
  - Try random genetic variation during a simulation. You will need to set the magnitude of the variation and rate. Look at variation of:
    - \* Aggressiveness.
    - \* The original rules of growth.

## Part IV: Bonus!

If you are ambitious you can try to explore more options. For example 3D or “smart” animals.

## Report

Each group should email me the code, a description of how the implementation is done. Plots of some population statistics and discussion of what they mean. Aprox. 3 pages of text and bunch of plots. **All in a single PDF document**

Example graphs - for 20X20 over 100 timesteps

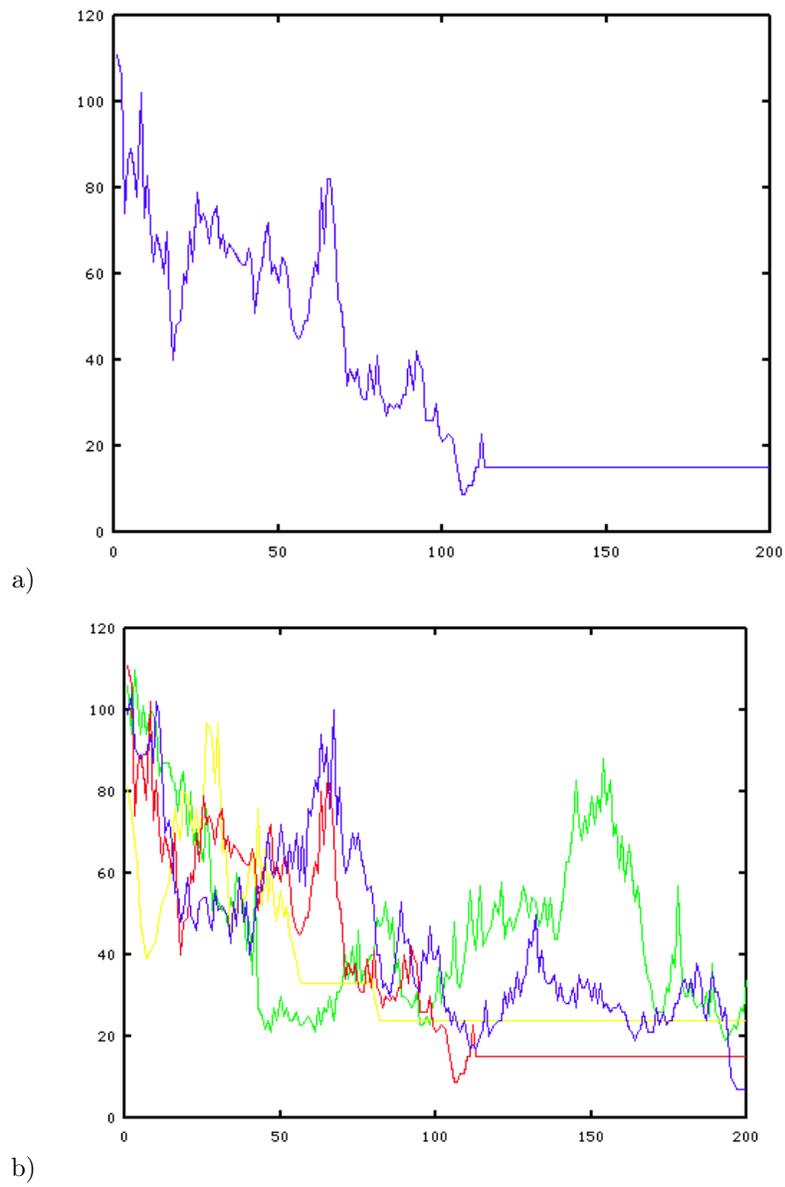
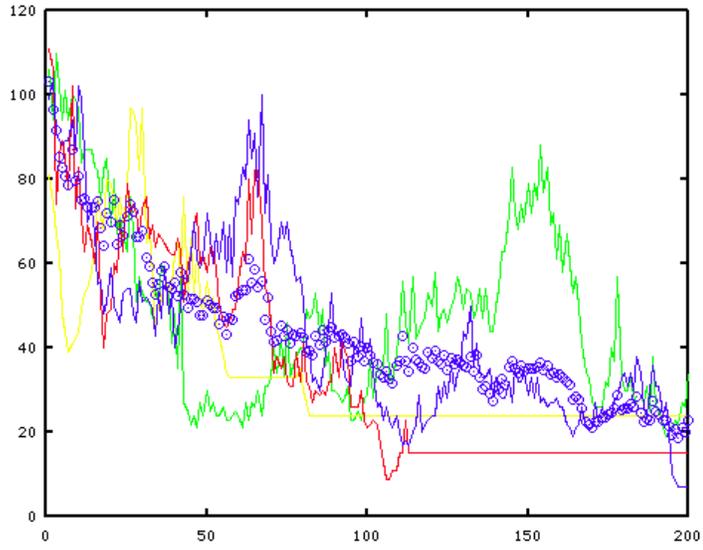
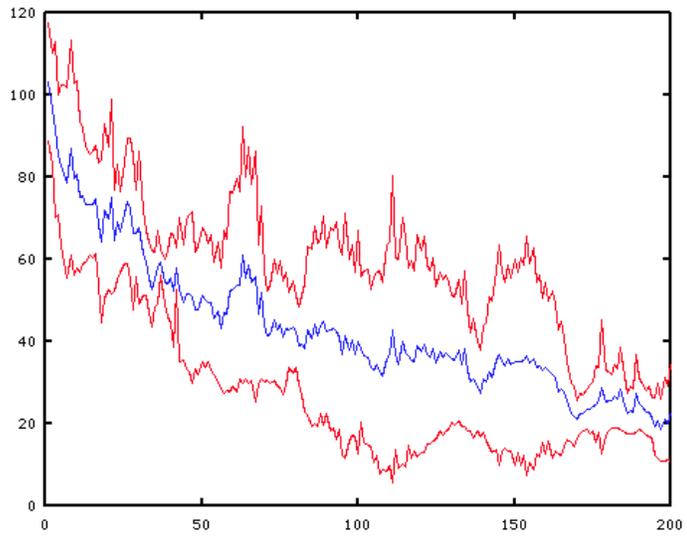


Figure 1: a) Single population density simulation. b) Multiple population simulations



a)



b)

Figure 2: Multiple population simulations and average. b) average and  $\pm$  std dev