Modelling Complex Systems: Project Sheet 2

May 19, 2011

The modelling complex systems course is assessed on the basis of two exercise sheets, of which this is the second one. Each exercise sheet has three 'projects', each consisting of a series of questions. After each question is a number of points associated with the question. The total points over all the questions is 100. To pass the course (grade 3) you must correctly answer questions amounting to at least half the points. In order to get grade 4 you must correctly answer 75% of the questions. In order to get a grade 5 you must correctly answer some of the questions labelled *grade 5 work* and have answered at least 75% correct over all the questions.

The lab sessions of this course are intended as a learning experience and we will provide extensive help with the exercises. To get the most out of this course you should attend these sessions where we will give valuable tips and advice on the problems. You are encouraged to work together in groups but each person should submit their own final set of answers to the questions. It is preferred that simulations should be implemented in Matlab, but it is acceptable to use other languages. We will try to help with programming problems in all languages, but 'guarantee' help only in Matlab.

David Sumpter (Å5401) has an office hour Wednesdays 11-12.

The deadline for the second exercise sheet is Thursday 9th of June 2011. Please email hand-ins to **david.sumpter@math.uu.se**. Written hand-ins can also be handed directly to David or Qi, or left in our mail boxes in the mathematics department. All code should be submitted as an appendix.

1 Cellular automata

Go to
http://ccl.northwestern.edu/netlogo/models/CA1DElementary
and try running the one dimensional cellular automata with a variety of rules. Find rules which are always the same independent of starting conditions; rules which produce periodic patterns repeating indefinitely; rules which produce random or chaotic patterns; and rules that produce 'pretty' patterns (you don't need to report your findings).

Now answer the following questions

1. Implement a program which calculates the box counting dimension of a matrix of 1's and 0's. Find the box counting dimension of: a chessboard matrix of alternating 1's and 0's; a matrix of random numbers; and of the output of CA90 (available on the course website). **(5)**

2. Implement your own elementary cellular automata simulator. The input should be a vector of length 8 or a number between 0 and 255, indicating the different rules. The output should be the change in cells through time in a two dimensional array. Simulate and provide examples of each of the four complexity classes discussed in the lectures. Find the rules which produce fractal like shapes. **(5)**

3. Calculate Box counting dimension, Entropy and Lyapunov exponents for the output of every one of the 256 rules. Can you use these measurements to characterize different cellular automata? The problem here is finding what you should calculate Entropy and Lyapunov exponent on. Discuss different approaches. *(grade 5 work)* **(5)**

2 Forest fires

The forest fire model is as follows. On time period 1 the forest is empty. On each time step there is first a growth phase where trees spontaneously appear on empty sites with probability g per site per time step. After growth there is a "lightening season" where there is a probability f per site per time step that it is struck by lightening. If the struck site contains a tree it burns down. Furthermore any adjoining trees burn down and so on until there are no more adjoining trees. The surviving trees pass to the next time step.

1. Implement the forest fire model in one dimension. If the growth rate is too high lightening burns all the trees down whenever it strikes, but if it is too low very little grows. Set $f = 1/L$ where L is the number of sites (should be taken to be around 1000). Now by systematically changing g find the growth rate which optimises the average number of trees alive at the end of each time step in the long term. **(5)**

2. Implement the forest fire model in two dimensions on a square lattice. Now neighbours are defined to be the four adjacent cells on the lattice. Simulate the model's behaviour for a grid of size L by L sites with $f = 0.1/L^2$ per site per time step and for $g = 1000f$, $g = 100f$ and $g = 10f$. Start with a lattice in which half the sites have trees and run the simulation for $(g/f)L^2$ time steps. Every time there is a fire store how many trees were burnt down in a vector U. Plot the frequency of fires of a size larger than u, $F(U > u)$, against fire size u on a log-log plot. Compare it to a power law, i.e. $F(U > u) = ku^\alpha$. Start by performing simulations for small L (e.g. 20) and then try larger values (the time taken to run grows quickly with L). **(10)**

3 Self-propelled particle models

Start by looking at the three simulation models presented in the lectures. Try simulating them for different parameter values and look at the code to understand how they work.

In these models we have n individuals moving in a two dimensional world with size L times L .

1. Implement the following self-propelled particle model of flocking in two dimensions with both attraction and alignment terms. An individual has position $(x_i(t), y_i(t))$ and a directional vector determined by angle $\theta_i(t)$ and fixed absolute distance traveled per time step δ. On each time step an individual picks at random one neighbour j within radius r. Then with probability p it changes its directional angle to face directly towards neighbour j, with probability q it takes the directional angle of its neighbour θ_j and with probability $1-p-q$ it maintains its previous direction θ_i. A random variable ϵ with range $[-\nu/2, \nu/2]$ is then added to the directional angle. Provide output f your simulation for parameter values: $p = q = 0.3$; $p = 0$ & $q = 0.6$; $p = 0.6$ & $q = 0$. Briefly discuss the kinds of dynamics you see for different numbers of individuals with these and other parameter values. **(8)**

2. For fixed $p = 0.1$ investigate how the alignment measure

$$\frac{1}{n}\sqrt{\left(\sum_{i=1}^{n}\cos(\theta_i(t)\right)^2 + \left(\sum_{i=1}^{n}\sin(\theta_i(t))\right)^2}$$

   changes as a function of q. In particular show the phase transition as you increase q. **(5)**

3. Develop a measure of aggregation for your model. The aggregation measure should capture how close together group members are. Can you find parameter values where large groups form an move together without losing group members? *(grade 5 work)* **(7)**