Modelling Complex Systems: Project Sheet 2 2015

April 16, 2015

The deadline for this exercise sheet is midnight May 15th.

Please submit hand-ins on Studentportalen. All code should be submitted as an appendix and not as part of the answer to the hand-ins. Please feel free to submit videos illustrating your results where appropriate, via Studentportalen or uploaded elsewhere.

1 Percolation & Phase Transitions

This exercise will be covered in lab session on Wednesday April 22nd. Please attempt to answer the questions before you come to the lab session.
Questions about this section can be directed to Ernest: `yu.liu@math.uu.se`
Note: papers referred to are available on the course page.

Percolation theory provides one of the mechanisms of how power laws come about. To get the basic ideas, we look at a simple version of the theory, namely 2D percolation.

Consider a $L \times L$ square lattice, and each 1×1 square will be colored in red with independent probability p, otherwise, it is white. The area of the whole lattice is $A \equiv L^2$, and the area of each square is 1. Percolation theory deals with various properties of *clusters* that form, i.e., a group of nearest neighboring colored squares (here, 'nearest squares' means the 4 squares adjacent to it). The area of cluster i is denoted as s_i, which is obviously always equal or larger than 1.

1. When L and p are given, we could ask what the expected area $\langle s \rangle$ is of the cluster to which a randomly chosen square belongs, i.e.,

$$\langle s \rangle = \sum_i (P(s_i) \cdot s_i) = \sum_i (\frac{s_i}{A} \cdot s_i)$$

which we expect to be an increasing function of p. For L=20, plot the normalized mean cluster area $\langle s \rangle / A$ vs p, with p varying from 0 to 1. On the same figure, plot the same quantities for $L = 50, 100, 200, 300$. Can you see a 'sharp change', that is, a 'phase transition' generally speaking? For which L is the transition the sharpest? Your figure should clearly show that the critical p is around $p_c = 0.5927462...$, which is called the 'percolation threshold'. **(4 points)**

Hints: You may find the Matlab function bwlabel from the Image Processing Toolbox useful. Your script may take some time to run, so in the plot of $\langle s \rangle$, just show sufficient interesting points to demonstrate the main character of this plot (no need to be equally varying). Around the transition if there is one, show more points.

2. Now instead of just considering the mean area, we consider the distribution of cluster area. Let $Q(s)$ be the number of clusters with area s that form. Draw $Q(s)$ when $p = 0.3$, 0.592 and 0.8 respectively; draw the corresponding cumulative distributions $Q_c(s)$, i.e. the number of clusters with area s or greater; draw the corresponding cumulative distributions $Q_c(s)$ in log-log scale (log base 10). Since your script may take some time to run, test your code first for $L = 50$ and then make final plots for $L = 800$. Do all of them follow a power law? **(4 points)**

3. Only when $L \to \infty$ and $p = p_c$ exactly do the distributions follow a rigid power law, which can be shown analytically (see Newman 2005). Now run the simulation multiple times for the case $L = 800$ and $p = 0.592$ as the question 2. Plot the log-log plot of the average $Q(s)$ and average $Q_c(s)$, respectively. Since $L = 800$ rather than infinite, the tail of the curve will deviate from a power-law, which is known as a finite-size effect. You would see this effect in both of your log-log plots, but do not worry about that too much. Use the method for discrete data in Clauset's paper (2007) to determine the exponent of this power law and the standard error from $Q(s)$, and then write down the distribution of cluster area in the form of $Q(s) = C \cdot s^{-\alpha}$. Plot the fitted power law over the points obtained from the simulation **(4 points)**.

4. Percolations generate power laws only under very specific circumstances (say, $p = p_c$, $L \to \infty$), so it seems an unlikely explanation for power law phenomena which appears widely in physics, biology, economics, etc. However, surprisingly, many dynamical systems tend to arrange themselves so that they always sit around the critical point. This is called self-organized criticality.

   One classic example is the forest fire model, as follows (Newman 2005). 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 "lightning season" where there is a probability f per site per time step that it is struck by lightning. 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.

Implement the forest fire model, referring to the percolation model above. Set $L = 300, f = 0.1/L^2, g = 10000f$, and plot the evolving of p vs. time step, namely the average tree density, corresponding to the p in the percolation model. Does p sit around p_c finally? Try to explain intuitively why p evolves in the way you saw. Write down your guess about the possible behavior of p if $L \to \infty$. **(5 points)**.

2 Self-Propelled Particle Models

Questions about this section can be directed to Alex:
`alexander.szorkovszky@math.uu.se`

This exercise will be covered in lab session on Monday April 27th. Please attempt to answer the questions before you come to the lab session.

First run the `Align2D.m` model provided on the webpage. This implements a Vicsek alignment model. We now develop a version of the model where interactions are only with neighbours travelling in a similar direction.

1. Implement a new rule so that a particle only aligns with neighbours already aligned within an angle α of its own direction. Use number of particles $N = 40$, domain size $L = 10$, radius of interaction $r = 1$, and angular noise $e = 0.5$. Run this for $J = 500$ time steps, at each time calculating the *polar order*

$$\phi_p(t) = \frac{1}{N}\left|\sum_{k=1}^{N} e^{i\theta_k(t)}\right| \tag{1}$$

where θ_k is the orientation of particle k and i is the imaginary unit. Also calculate the *apolar order*

$$\phi_a(t) = \frac{1}{N}\left|\sum_{k=1}^{N} e^{2i\theta_k(t)}\right| \tag{2}$$

Run the simulation for $\alpha = 0$, $\pi/3$ and $2\pi/3$. Plot both of these measurements as a function of time. **(4 points)**

2. Now set the radius of interaction $r = 2$ and $J = 200$. Run several simulations with various combinations of the noise parameter e and angle α, varying both between 0 and π, and calculating the two order parameters at the end of each simulation. Make two dimensional heat maps for the average polar and apolar order, as a function of e and α. In what regime is $\phi_a > \phi_p$ and what does the motion look like? **(5 points)**

3. Since real animals are not points, introduce a rule to reduce the likelihood of "collisions", when particles are too close together. This could be done, for example, by varying the particle speed with time, or by steering away from incoming particles, or both. Does this qualitatively change the flocking behaviour, compared to without the new rule? **(5 points)**

3 Genetic Algorithms

Questions about this section can be directed to Alex:
`alexander.szorkovszky@math.uu.se`

This exercise will be covered in lab session on Wednesday May 6th. Please attempt to answer the questions before you come to the lab session.

For this problem we will imagine we have a painter robot similar to the cellular automaton from the first problem sheet. We will use this robot to paint the floor of a room. To make it interesting, the painter starts at a random place in the room, and paints continuously. We will also imagine that there is exactly enough paint to cover the floor. This means that it is wasteful to visit the same spot more than once.
To see if there is a optimal set of rules for the painter to follow, you will create a genetic algorithm. Download the MATLAB function `painter_play.m` from the course webpage. As inputs, this function receives

1. A chromosome: A 1x54 array of numbers between 0 and 3 that shows how to respond (0:go straight, 1:turn left, 2:turn right, 3: random turn

left/right) in each of the 54 possible states (empty/occupied/painted in squares forward/left/right = $3^3 = 27$, current square empty/painted previously = 2).

2. An environment: A 2D array representing a rectangular room. Empty (paintable) space is represented by a zero, while furniture or extra walls are represented by ones. Outside walls are automatically created by `painter_play()`.

The function `painter_play()` then uses the rule set to guide a painter, initially placed in the room with a random position and direction, until the paint can is empty. Note that the painter does not move when it tries to walk into a wall or furniture. The efficiency (total fraction of paintable space covered) is then given as an output, as well as the X-Y trajectory of the painter.
To see that the painter works, you can try passing it an empty room for an environment and a trivial chromosome. For example, a chromosome consisting of all 3s produces a kind of random walk.

Now do the following:

1. Think of a simple strategy for the painter to cover a lot of space in an empty room. Describe this strategy in a few words or sketch it, but do not try to encode it in the chromosome. **(2 points)**

2. Create 50 random chromosomes in a 50x54 matrix, as well as a 20x40 empty room. Create a genetic algorithm to evolve this population over 200 generations, playing each chromosome several times and storing the chromosome's average efficiency as the fitness. Use single-point crossover with a mutation rate of 0.002 per locus per generation. Plot the final set of chromosomes using `imagesc()`. Play one of the more successful ones, and plot the X-Y trajectory. Is this what you expected? **(5 points)**

3. Plot the average fitness in the population vs generation. You will likely see large sudden "jumps" in fitness, corresponding to strategic innovations. In your own words, write down two possible examples of an innovation that would increase fitness. **(3 points)**

4. Add some furniture to the empty room (about 100 square metres in total) and use one of your highly evolved chromosomes, and plot the X-Y trajectory. How does the efficiency compare to that in an empty room? If the strategy fails, how does it fail? Now try running the genetic algorithm with your new furnished room from the start. How does the strategy compare to the empty room strategy? **(4 points)**