2. Cellular automata and epidemics spreading

The SIRS model comprises agents (i.e., people or other organisms), i, arranged on a two-dimensional lattice. The agents may contract some infection (e.g., the annual flu), and can be in one of three states:

- S: susceptible to the infection;
- I: infected;
- R: recovered from the infection (and hence immune to it).

The SIRS model is called like this because agents go through these states cyclically: S changes to I when a susceptible agent comes into contact with an infected one; recovery changes the agent from I to R; eventually, in the absence of contact with infected agents, the immune response wears off and R changes back to S.

More precisely, when a site i is updated:

- $S \to I$ with probability p_1 if at least one neighbour of i is I; otherwise site i is unchanged.
- $I \to R$ with probability p_2.
- $R \to S$ with probability p_3.

1. One way to simulate this model is with a fully parallel updating scheme. In one timestep, every lattice site is updated, and the state of site i at timestep t+1 is determined by the state of the lattice at time t (i.e., before any of the updates are made). Does it matter what order the sites are visited for update in this scheme? To convince yourself of the answer, you can take for instance $p_1 = p_2 = 1$ and $p_3 = 0$.

2. An alternative approach is a random sequential updating scheme, which you should use in the remainder of this Checkpoint. Here, only one site is updated in each timestep, and the site to update is chosen randomly. If there are N sites in the system, N such updates is called a Monte Carlo sweep. Therefore on average each site may be updated more than once. (If you like, you can use your programs, or some maths, to find out how many times on average a site is updated.)

3. Write a Java program to simulate the SIRS model on the 2d square lattice with periodic boundary conditions. Your program should allow the user to choose the system size, probabilities p_1, p_2 and p_3, and use the random sequential updating scheme. It should also be able to show a visualisation of the state of the lattice as it is running.

4. Find some parameter sets which leads to (i) an absorbing state with all sites susceptible to the infection; (ii) a dynamic equilibrium between S, I and R; (iii) a case with cyclic wave of infections through the lattice. Visualise the dynamics in each case.

5. Now set $p_2 = 0.5$. The average number of infected sites, $\langle I \rangle$, is a quantity that provides a quantitative measure of the different possible behaviours (just as the magnetisation in the Ising model). Using this measure, obtain the phase diagram of the system in the $p_1 - p_3$ plane (i.e., a diagram showing the regions where qualitatively different behaviour emerges). To do this, you should plot, in the same $p_1 - p_3$ plane, the value of the average fraction of infected sites $\langle I \rangle / N$, e.g. as a contour or colour plot. To do the averaging, for each value of (p_1, p_3), you can use a very long simulation after equilibration, or a number of shorter simulations (for each you should still wait for equilibration).

6. Where are the waves in your phase diagram? To pinpoint these, a good plot is a countour plot of the variance of the number of infected sites $\frac{\langle I^2 \rangle - \langle I \rangle^2}{N}$ as a function of p_1 and p_3 (still for $p_2 = 0.5$). For one parameter set which leads to cyclic spreading of waves, plot I/N, the fraction of infected sites, as a function of time (in units of Monte Carlo sweeps), and comment on your graph. Use your graph to estimate the recurrence period of the wave.

7. Modify your program so that some fraction of agents are permanently immune to the infection (i.e., in the R state). For $p_2 = p_3 = 0.5$, what fraction of immunity is require to prevent the spreading of the infection, as a function of p_1?