AMH5 Assignment 1

Due Thursday 14 September 2017 (week 7) by 5pm.

Please email me a pdf of your answers and your MATLAB code and instructions on how to run it. (Alternatively, you can hand me a paper document, but please still email the MATLAB code and how to run it.)

Problem 1: Simulating a random walk

[3 marks]

Write a simulation (in MATLAB) of a 1-D random walk of N particles starting with an even distribution in the interval $[-1, 1]$ and moving with spatial step size Δx at each time step Δt. At each time step, a particle moves left with probability $p/2$, right with probability $p/2$ and stays still with probability $1 - p$. For the following parts, let us assume that $p = 0.9$.

To make convergence easier, we will assume that random walkers represent blocks filling the intervals $(n\Delta x, (n + 1)\Delta x]$ for integers n, rather than point masses sitting only at $n\Delta x$.

In addition, assume that the random walkers can walk over the range $[-10, 10]$. If they reach the boundary and try to move beyond it, they just stay in place. This assumption means you do not have to simulate over an infinite space.

(a) Start with $N = 1000$ and $\Delta x = 1$ and $\Delta t = 1$. You will have to set the appropriate initial distribution of random walkers, so that they evenly cover the interval $[-1, 1]$. Plot the population distributions at times $t = 20$.

(b) Take $\Delta x = 0.5$ and the appropriate Δt such that the random walk will "diffuse" at the same rate. Plot the population distributions at times $t = 20$.

(c) Take $\Delta x = 0.2$ and the appropriate Δt such that the random walk will "diffuse" at the same rate. Plot the population distributions at times $t = 20$.

For comparison, plot all three population distributions on the same figure and label them. To do a proper comparison, you will have to scale the populations from (a), (b), and (c) by a different constant factor so that the initial distribution is the same.

Problem 2: Simulating the PDE

[3 marks]

(a) Write the (deterministic) PDE along with initial and boundary conditions that correspond to the dynamics described in the system above if we were to take $\Delta t \to 0$ and scale Δx appropriately so that the system converges.

(b) Use the 'pdepe' function in MATLAB to simulate the PDE from part (a) to time $t = 20$ and compare the result with the plots obtained from Problem 1 by plotting them on the same figure. Depending on how you scaled the plots in Problem 1, you will have to set the initial distribution and population scaling of the PDE appropriately to compare the PDE solution with the random walk simulations above.

Problem 3: Simulating a random walk off the lattice

[4 marks]

Can you take the 1-D random walk off the x-lattice? For the random walk in Problem 1, we assumed that the walking particles can only be at integer multiples of Δx. What if we keep the time step discrete, but we assume that space is continuous, i.e., at times $t = n\Delta t$, particles can have any real x value, not just integer multiples of Δx? Mathematically, what is the correct distribution of step sizes $x \in \mathbf{R}$, such that the off-lattice random walk (probabilistically) reproduces the behaviour of the diffusion equation? Use this distribution in the following part.

Suppose we start with $N = 1000$ particles moving with time step $\Delta t = 1$. Suppose they all start at $x = 0$, so that they fall in the interval [-1, 1]. At every time step they move to a new point, but this point does not have to fall on a lattice made up of discrete intervals. Also, for this problem, assume that there is no boundary. Write a MATLAB simulation of how these particles will scatter.

In your pdf document, please explain the reasoning behind how you attacked the problem and wrote your simulation.

Some of this, especially Problem 3, might be tricky, so feel free to ask me questions by email, in class, or Skype (email me to let me know if you want to do this).

MATLAB has several built-in random number generators, including 'rand' (uniformly distributed between 0 and 1), 'randn' (normally distributed), and 'binorand' (binomially distributed). You can find documentation about this in MATLAB help. You are not required to use all three or any of these examples, and you are free to use other MATLAB built-in random number generators. I don't think it has a built-in random walk program yet.