MXB261 - Modelling and Simulation Science

Assignment 1 - Problem Solving Task - 15% of final grade

Due: 23:59 on Friday of Week 6 (29 August 2025)

Part 1 - A Biased Random Walk - 10 marks

We want to simulate how individuals might move from one side of a large room to the other side. In a way, trying to capture how a large crowd of people might cross a large room. We want to let each individual walk across the room until they either reach the other wall or they bump into someone else who has walked ahead of them and is waiting at the other wall. We can simulate where our population of individuals may end up by thinking of them as a population of particles undergoing a random walk.

Suppose we have a 2D simulation domain, 99×99 units. One at a time, particles will start at a nominated position along the top row of the domain, and will follow a biased walk "under gravity" towards the bottom row. Here is what you need to consider when coding the random walk:

1. a particle can move in three directions: South, West or East (or down, left or right).
2. If the particle is to move West or East but the location is occupied, generate a new direction to move until the particle can move.
3. If the particle is to move South but the location is occupied (or the particle has reached the bottom row in the domain), the particle stops - this is the end of the particle's walk.
4. Assume the side boundaries wrap around (cyclic), so that if a particle were to try to go west at the left-hand boundary, it appears at the right-hand boundary.

The biased walk will allow a choice of direction at each step of either South, West or East, with probabilities s, w, e, respectively, that are described by four different cases:

(i) $s = w = e = \frac{1}{3}$

(ii) $s = \frac{2}{3}$, $w = \frac{1}{6}$, $e = \frac{1}{6}$

(iii) $s = \frac{3}{5}$, $w = \frac{3}{10}$, $e = \frac{1}{10}$

(iv) $s = \frac{3}{5}$, $w = \frac{1}{10}$, $e = \frac{3}{10}$

Simulate the biased walks of N particles. When all particles have completed their walks, calculate the height of all the resulting growths building up, in each column, from the bottom row of the domain, and produce a histogram of the distributions of these heights.

For each probability 'case' (i) - (iv) above, you are to compare the results for two different start positions ($P = 1$, and 'rand'); for 1 start position, all the particles will start in column 50 in the top row; for 'rand' start positions, each particle will start from a random location along the top row with equal likelihood of any position in the top row being chosen. For each of the possible cases (probability 'case' and start position), compare the results for $N = 100$ and $N = 200$ particles.

- Submit a code which simulations the above particle biased walk "under gravity". Your code must be in MATLAB, and must include a function that accepts input parameters N (the number of particles), P (the start position), and s, w, e (the probabilities).
- Plot figures in MATLAB showing the distribution of heights for 1, and 'rand' start positions, and the two different N values, for each of the probability cases (i) - (iv). Create the histograms with 99 bins (each bin representing a different column). There will be 4 figures in total, with the cases (i) - (iv) being displayed in a 2×2 grid of subplots within the figure, for each P and N combination In other words:
  - Figure 1 - 2x2 plots for $P = 1, N = 100$ which each subfigure one of the biases in (i)-(iv)
  - Figure 2 - 2x2 plots for $P = 1, N = 200$ which each subfigure one of the biases in (i)-(iv)
  - Figure 3 - 2x2 plots for $P = rand, N = 100$ which each subfigure one of the biases in (i)-(iv)
  - Figure 4 - 2x2 plots for $P = rand, N = 200$ which each subfigure one of the biases in (i)-(iv)

  Be sure to correctly label your figures: xlabel axis, ylabel axis, legend (if neccessary) title and large enough font size.
- Submit a report summarising your findings for the above particle walk. In your report include the four figures mentioned above along with captions and a concise discussion which addresses the following topics:
  1. compare the effect of having a single starting point or random starting points on the distribution of column heights.
  2. discuss how adding more particles effects the distribution of column heights.
  3. mention whether the distributions for the particles final position looks like any other distributions you've encountered in MXB261.

Part 2 - Understanding KL Divergence Through Sampling - 5 marks

In this part, you will explore how the Kullback-Leibler (KL) divergence varies as sample size increases when estimating a probability distribution from finite samples.

Task: A discrete random variable K follows a Poisson distribution with parameter $\lambda = 4$. Your task is to investigate how well empirical distributions approximate this true distribution as sample size increases.

- **Setup:** Define the true Poisson PMF with $\lambda = 4$. Truncate the PMF to $k \in \{0, 1, ..., 15\}$ for practical computation and re-normalise so it sums to 1.
- **Sampling Implementation:** Implement inverse transform sampling to generate samples from your truncated Poisson PMF. Generate samples for 6 sample sizes: 10, 25, 50, 100, 175, and 250.
- **KL Divergence Calculation:** For each sample size, repeat the sampling 100 times. For each experiment:
  - Calculate the empirical PMF from the samples
  - To avoid numerical issues with log(0), use `max(empirical_pmf, 1e-10)`
  - Compute the KL divergence between the true and empirical distributions
- **Visualisation:** Create 2 figures with proper labels, titles, and legends:
  1. A plot of mean KL divergence versus sample size with error bars. Use `errorbar()` with standard error SE = std(KL values)/sqrt(nbr of experiments)
  2. A figure with 2×3 subplots, one for each sample size, showing bar charts comparing true PMF to one example empirical PMF
- **Analysis:** In your report (100-300 words), address:
  1. What does KL divergence = 0 mean in this context?
  2. How and why does KL divergence change as sample size increases?

Submission:

This assignment is to be submitted via Canvas.

- The MATLAB source codes for implementing this assignment should be submitted on Canvas as completely runnable .m files. If these scripts require any files other than the provided .mat files, these must also be submitted
- Your report (in pdf format) containing the Figures and the Discussions should be uploaded to Canvas. At the end of the report please include a copy of your code. You can do this by either publishing your code to a PDF and combining this PDF with the report PDF, or by copying and pasting your code into word and saving as a PDF. Please make sure the Figures, Discussion and Code is in one PDF.

Guide to the Marking Schedule

- Part 1: (total: 10 marks)
  - **1 mark:** Code well-structured and well-documented.
  - **5 marks:** Code functionality
    - Input parameters are specified.
    - Start positions P is coded correctly.
    - Number of particles is coded correctly
    - Probabilities s, e, w (for 4 different cases) are coded correctly.
    - wrap-around boundary is coded correctly.
  - **4 marks:** The discussion shows insight, accurate results, and correct interpretation of the results, with reference to the figures (that should include title, and axes labels).
- Part 2: (total: 5 marks)
  - **3 marks:** Correct implementation of truncated Poisson PMF and inverse transform sampling. Correct KL calculation, including direction and handling of numerical issues.
  - **2 marks:** Clear, well-labelled figures. Correct explanation of links between KL divergence, sample size, and approximation quality.