University of Hull

Report to Analyse the Simulation of Random Growth of Cells

By

Matthew Humble

202209974

Programming Assignment

Computational Science 600093

Dr Chandrasekhar Kambhampati

Due: 7th January 2025

Word Count:

**Contents**

Contents…………………………………………………………………………………...…1

Figures………………………………………………………………………………………..

Introduction…………………………………………………………………………………..

Task 1.1…………………………………………………………………………

Task 1.2……………………………………………………………………………

Task 2.1……………………………………………………………………………

Task 2.2………………………………………………………………………………

Conclusion………………………………………………………………………………………

Bibliography…………………………………………………………………………………..

**Figures**

**Introduction**

Computational science is the discipline within computer science that advances knowledge in various scientific fields through complex computational analysis, modelling, simulation, and numerical approximation (Plonus, 2007). As a result, it can provide innovative solutions to problems by using powerful computers to replace or complement traditional modelling and experimental methods.

This study aims to understand the different complexities that encapsulate growth simulations and how they can be applied to real-world scenarios. We shall build upon various random grid walk models to understand random movement, which can be further applied to random cell growth, simulating a disease such as a cancer tumour. In addition, the Gompertz model will support this simulation by applying more realistic characteristics, such as time, cell capacity, and growth rate. Combining these methods shall form a basic simulation of how a cancer tumour can start to grow and multiply over time in a living organism. Thus, aims to provide more insight into cell growth that can be applied to real-world situations.

**Task 1.1**

The first steps to create the simulation include creating a random grid walk function. The goal for this task is for a cell to start in the middle of a one hundred by one hundred grid and move randomly: left, right, up and down one grid place for one hundred steps.

This task relies on two random number generators and a simple binary algorithm to decide at random whether the cell should move in any of the four directions. This algorithm can be looped one hundred times to simulate the movement for one hundred steps.

However, one issue with this algorithm is the implementation of two random number generators. As a computer cannot be biologically truly random, we must implement pseudo-random number generators to simulate randomness. A pseudo-random number generator is an algorithm that ‘produces a deterministic stream of seemingly random numbers’ (Reeds & Walkenhorst, 2011). Using this function, a distribution type must be applied to determine the range and type of outputs to expect from the function. The two main types considered for this project include: uniform and normal distribution.

A comparison of a graph

Description automatically generated with medium confidence**A comparison of a graph

Description automatically generated with medium confidence**‘Normal distribution is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data from the mean’ (Chen, 2024). For example, if a normal distribution function had a mean of 0 and a standard deviation of 1, most results would be in the range -1 and 1, primarily around 0, and values out of this range would rarely be selected in comparison. Resulting in a ‘bell curve’ when graphed, as shown in Figure 1. Normal distributions can be used in several circumstances. Often, normal distribution can be found to occur naturally. ‘For example, the average height of a human is roughly 5’ 9”’(Chen, 2024). Taller and shorter people exist but decrease in frequency within the population. Thus, normal distribution would be helpful in simulations relating to natural trends such as population height.

**Figure 1: Normal Distribution Graph**

Alternatively, ‘uniform distribution refers to a type of probability distribution in which all outcomes are equally as likely to occur’ (Chen, 2024). Real-world similarities between uniform distribution can be considered a fair coin toss because the odds of getting either heads or tails are even. This can also be plotted on a graph, as shown in Figure 2. Ideally, the graph will be a straight horizontal line. However, this is not always the case on small sample sizes due to variance, but this is less noticeable within a large dataset.

**Figure 2: Normal Distribution Graph**

Concerning the problem set out in this task, a uniform distribution shall be implemented as a random distribution, as there is no bias towards one result, and we can ensure even possibilities for all results. Similar to the coin toss analogy, two uniform distribution random number generators will be used to provide a binary result of either one or zero.

**A grid with numbers and letters

Description automatically generated**Once implemented, the random numbers must be assigned to a direction. This is done using a simple algorithm to determine the result of two binary values. Thus, (0 0)2 move right by one, (0 1)2 move left by one, (1 0)2 move down by one and (1 1)2 move up by one. This formula covers all random outcome possibilities and all possible directions of movement. Once this algorithm is iterated through one hundred times to represent one hundred steps, the results can be seen in Figure 3.

**Figure 3: Random Walk on Grid for 100 Steps**

Figure 3 depicts the task's results, with the point moving up twenty-six times, down twenty-seven times, left nineteen times, and right twenty-eight times. As uniform distribution was implemented, it would be expected that the point would move evenly twenty-five times in all directions. However, due to variance in a small sample size, known as statistical fluctuation, the results are slightly off what was predicted as the point moved left six times less than expected. However, it is expected by the ‘law of large numbers’ that a larger sample will represent a population mean (Taboga, 2021). Thus, as greater step size is applied to the algorithm, results will be expected to align with uniform distribution.

**Task 1.2**

Developing on from the previous task

**References**

1. Plonus, M. (May 2007) The Digital Computer, Electronics and Communications for Scientists and Engineers. Available online: <https://www.sciencedirect.com/science/article/abs/pii/B9780125330848500197> [Accessed: 13/12/2024]
2. Rees, S & Walkenhorst, J. (November 2011) Large-Scale Credit Risk Loss Simulation, GPU Computing Gems Jade Edition. Available online: <https://www.sciencedirect.com/science/article/abs/pii/B9780123859631000241> [Accessed: 13/12/2024]
3. Chen, J (March 2024) Normal Distribution: What It Is, Uses, and Formula. Available online: <https://www.investopedia.com/terms/n/normaldistribution.asp> [Accessed 13/12/2024]
4. Chen, J (June 2024) Uniform Distribution: Definition, How It Works, and Example. Available online: <https://www.investopedia.com/terms/u/uniform-distribution.asp> [Accessed: 13/12/2024]
5. Taboga, M (2021) "Law of Large Numbers", Lectures on probability theory and mathematical statistics. Available online: <https://www.statlect.com/asymptotic-theory/law-of-large-numbers> [Accessed: 14/12/2024]