Second Midterm

Scientific Computing II
Fundación Universitaria Konrad Lorenz
October 17, 2023

- This is an **individual exam**. Therefore, do not provide or receive help from anyone for the completion of this midterm.
- The exam will be available from October 17th, 2023, 6.15pm. From that time on you will have 2 hours and 15 minutes to complete the midterm exam.
- The solution must be turned in only through the Aula virtual. Midterms sent over email will not be graded.
- Full score will only be given to correct and completely justified answers. Miraculous, obtuse and unnecessarily complex solutions will receive partial or null score.
- You can ask any question of this assignment during class or through email: juliano.jimenezc@konradlorenz.edu.co.

Game of Life

Conway's Game of Life is a cellular automaton devised by John Conway in 1970. It is a zero-player game that operates on a grid of cells. Each cell can be either alive or dead, and it follows simple rules to evolve through generations.

The game is played on a grid, where each cell can be in one of two states: alive or dead. The grid evolves over discrete time steps (generations). The rules for cell evolution are as follows:

- Any live cell with 2 or 3 live neighbors survives to the next generation.
- Any dead cell with exactly 3 live neighbors becomes a live cell.
- All other live cells die, and all other dead cells remain dead.

The game continues to evolve through generations, creating fascinating patterns and structures. It has applications in various fields, including mathematics, computer science, and biology. As a mathematician, your mission is to simulate the game of life in *C++* for different densities in parallel and plot your results.

These are your tasks:

1. (0/10) Create a function that initializes an $N \times N$ array ($N = 200$) based on an initial density ρ, which determines the quantity of alive cells at the beginning of the simulation. You are expected to distribute uniformly the alive cells across the grid.

2. (0/10) Create a function that evolves the previous array for a total of 500 iterations by applying the rules of the game of life. You can define another function with the sole purpose of evolving the array a single period.

3. (0/20) Parallelize your code to run the simulations concurrently for different values of ρ. Specifically, you are expected to partition the unit interval $[0, 1]$ in 50, and choose the values of this equally spaced partition as input densities. Use at least 10 processes for this job.

FIGURE 1. Plot of a configuration of the game of life after a certain amount of iterations.

4. (0/10) Do a report in Jupyter Notebook showing your results as a plot, where the x-axis represents the initial density and the y-axis represents the final density after 500 iterations. What can you conclude? Is there a critical density for which the final density is maximized?