ENGR3: Introduction to Programming

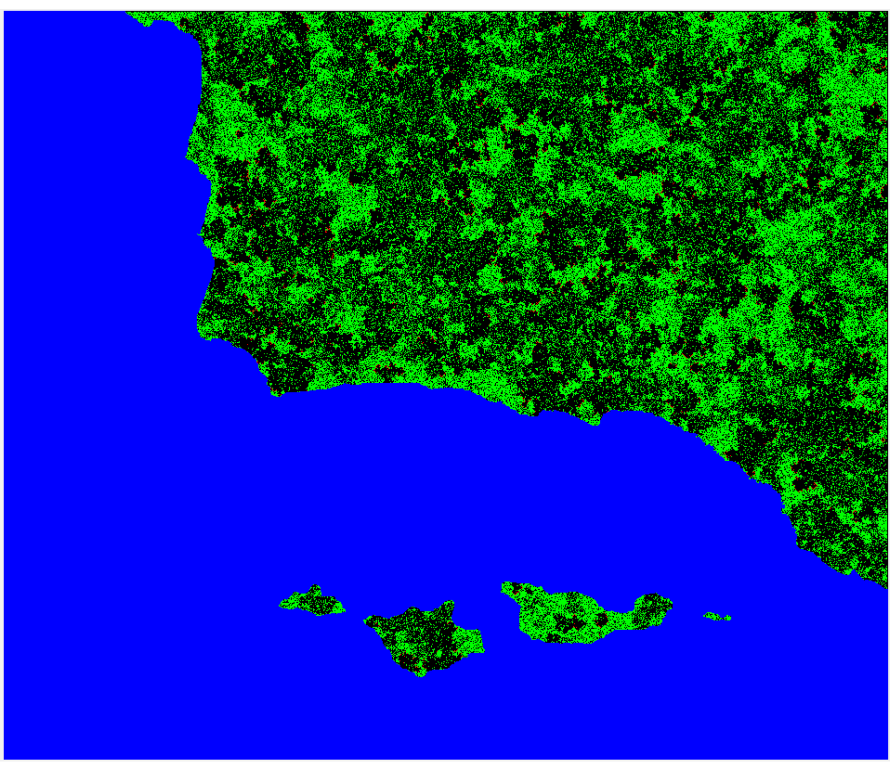
2024 Midterm Project

Due: Sunday November 3 at 11:55pm

There are 5 Tasks. Each Task is worth 20 points.

Be sure to comment your code!

Due to the prolonged threat of California fires, Santa Barbara and Ventura counties have decided to work together to create a numerical simulation for forest fire dynamics. Since you are a computational scientist with expertise in creating computer simulations, they have enlisted your help. They would like to use the model to understand how fire plays a role in their ecosystem, fire's effect on urban development, make predictions for air quality, understand how climate change will impact future fires, and develop responsible ways for handling future fires. Before you can do any of that, you need to develop a proof of concept to show the Next-Generation Fire Monitoring System Committee.



Model Overview:

The simulation that you will be creating falls under a class of modeling called cellular automata. In cellular automata, there is a grid of cells with states that evolve over time and space by following a set of rules. In your simulation, you can think of each cell as a region of land on a map that either has water, trees, fire, or is empty. The fire dynamics are modeled using a set of rules that describe when a cell can grow trees, when a fire can start, and how fire can spread to other trees. Our simulation will have the following rules:

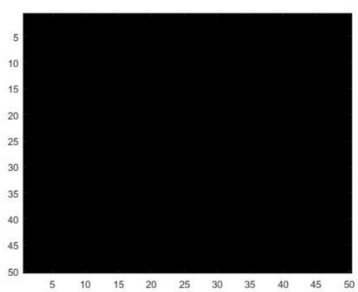
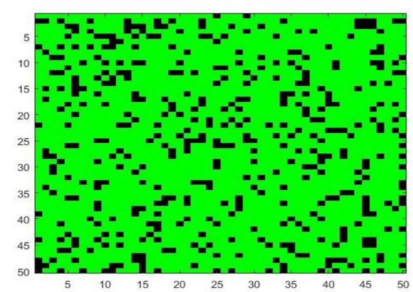
1. Trees grow on empty cells with probability ptree
2. A fire can start on a cell with a tree with probability pfire
3. A fire can spread to adjacent cells above, below, or beside it (no diagonals) if they have trees on them with probability pspread
4. A fire cell turns into an empty cell one time step after it has started (all trees burned)
5. A water cell cannot have trees or fires on it
6. A cell can only be occupied by fire, trees, water, or nothing at any particular time.

You will code these rules in parts and then combine them together to make your final simulation.

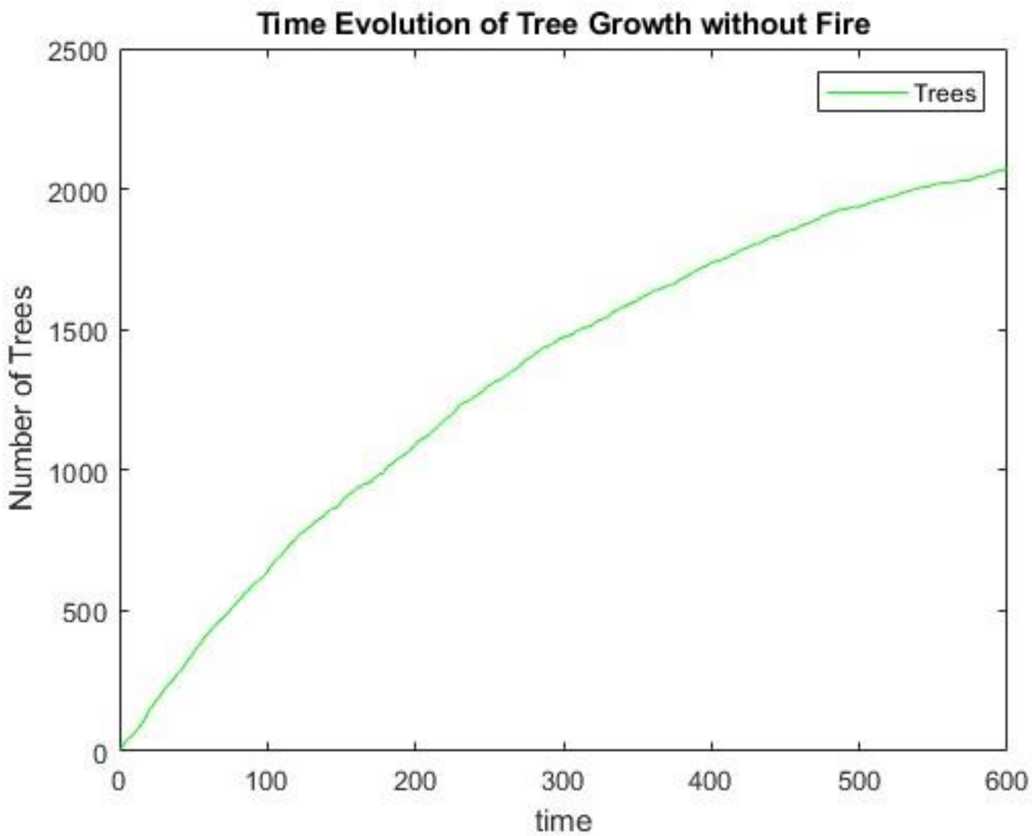
Task 1: Growing trees on the lattice

Write a piece of code that first creates an empty grid of cells of size n x n (black color) and grows trees on the empty cells with a probability of ptree. For each time step, loop over the elements of the grid one by one and change that cell to green with a probability of ptree. For each cell in the grid, you will need to generate a random number with rand(). If rand(1) is less than ptree, then you can grow a tree on an empty cell by changing the cell from black to green. Otherwise, the cell stays black. Show the trees growing over time with image(RGB_img) or imshow(RGB_img) and pausing afterwards for 0.01 with pause(0.01). Submit your code with n = 50, ptree=0.003, and timesteps=600

The beginning and end of your animation should resemble the figure below. Since it's a stochastic (random) process, your final image will look slightly different from this one.

Starting: t=0	Ending: t=600
	

For each of the time steps, calculate the total number of trees at any given time. Then duplicate this figure below by plotting the total number of trees vs time. Match the title, axis labels, legend, and curve color as seen below. Play around with the linewidth and choose one that you think looks the best (up to you to decide). Note that since the process is random your graph will look slightly different each time and might look slightly different than the one below. This is okay as long as it looks similar.



Task 1 Deliverables:

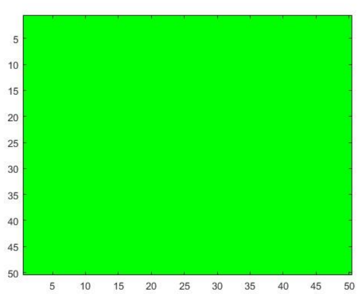
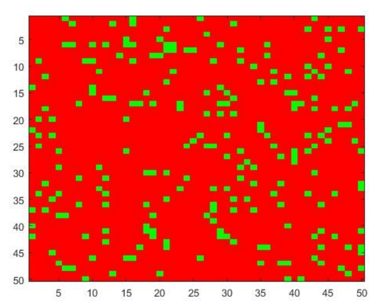
1. **Animation**
2. **Time evolution figure**

Task 1 Filename: part_1.m

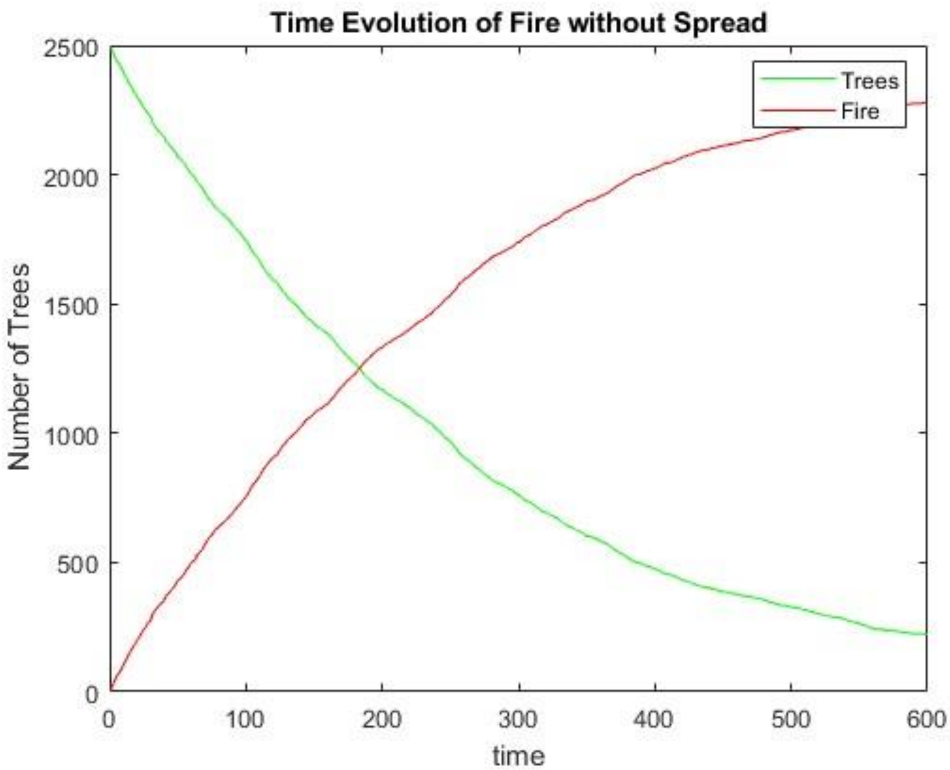
Task 2: Starting fires on the lattice

Write a piece of code that first creates a completely green grid of cells of size n x n and starts fires on the green cells with a probability of pfire. For this task, we will violate rule 4 and keep the cell on fire after it has started. We will fix this issue later on. For each time step, loop over the elements of the grid one by one and change that cell from green to red with a probability of pfire. For each cell in the grid, you will need to generate a random number with rand(). If rand(1) is less than pfire, then you can start a fire on a tree cell by changing the cell from green to red. Otherwise, the cell stays green. Show the evolution of trees and fires starting over time with image(RGB_img) or imshow(RGB_img) and pausing afterwards for 0.01 with pause(0.01). Submit your code with n = 50, pfire=0.004, and timesteps=600

The beginning and end of your animation should resemble the figure below. Since it's a stochastic (random) process, your final image will look slightly different from this one.

Starting: t=0	Ending: t=600
	

Similar to task 1, plot the time evolution of the total number of trees and fires vs time. As before, label the axes and make the title the same as before. Have the legend show both the total number of trees and fires with red curves for the fires and green curves for the trees. Remember that the curve on your graph might look slightly different each time:



Task 2 Deliverables:

1. **Animation**
2. **Time evolution figure**

Task 2 Filename: part_2.m

Task 3: Spreading fires on the lattice

Fires can only spread to green cells. A green cell catches on fire if it has an adjacent cell that is on fire with probability pspread. An adjacent cell is a cell above, below, left, or right of the cell. Diagonal cells are not considered adjacent.

The figures below show how fire can spread to adjacent cells when pspread=1. In the grid below, the cell at the coordinates (i,j) and time t catches on fire since the cell at coordinates (i,j+1) is on fire. Unlike in task 2, a cell that is on fire at time step t turns to an empty cell (black) at the next time step given by t+1.

t

	A(i-1, j)	
A(I, j-1)	A(I,j)	A(I, j+1)
	A(i+1, j)	

t+1

	A(i-1, j)	
A(I, j-1)	A(I,j)	A(I, j+1)
	A(i+1, j)	

t+2

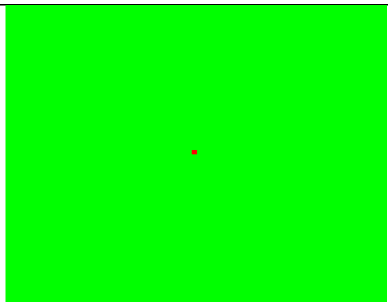
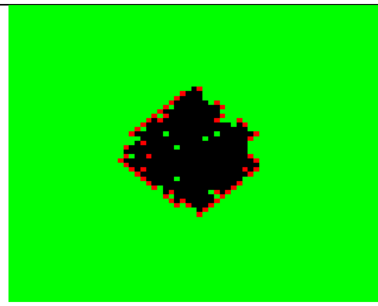
	A(i-1, j)	
A(I, j-1)	A(I,j)	A(I, j+1)
	A(i+1, j)	

t+3

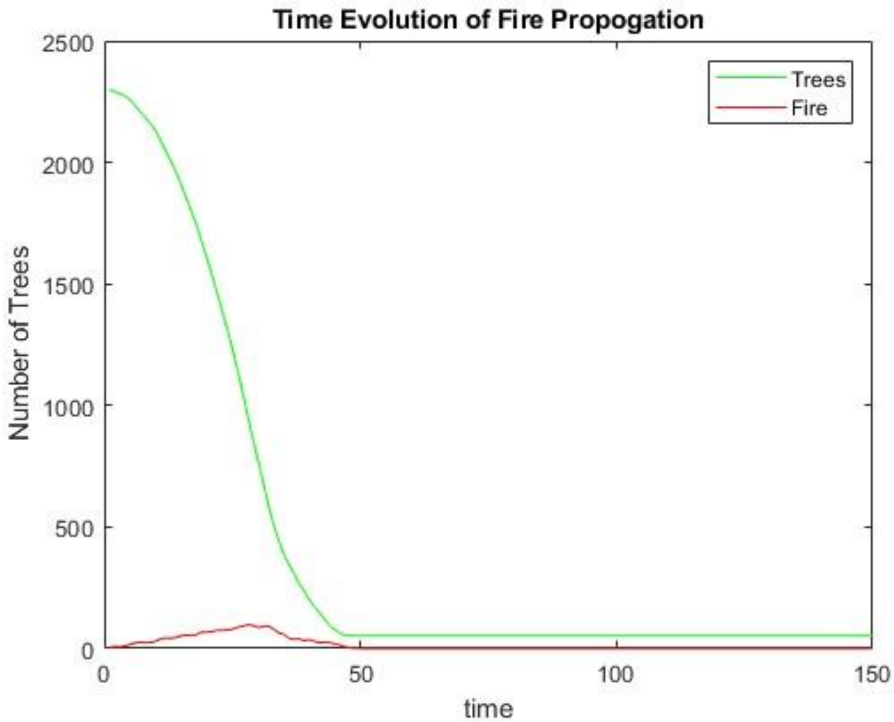
	A(i-1, j)	
A(I, j-1)	A(I,j)	A(I, j+1)
	A(i+1, j)	

Write a piece of code that first creates a completely green grid of cells of size n x n with one red cell in the very center (floor(n/2), floor(n/2)) and a black edge around all sides. The black edge allows us to ignore the edges of the grid, which simplifies the coding. Progress the simulation forward in time by catching a green cell on fire with probability pspread if its adjacent cell is also on fire. Extinguish the old fire cells at the next time step. Show the evolution of trees and fires over time with image(RGB_img) or imshow(RGB_img) and pausing afterwards for 0.05. Submit your code with n = 50, pspread=0.8, and timesteps=150

Your animation should resemble the figures below. Since it's a stochastic (random) process, your final image will look slightly different from this one.

Starting: t=0	During simulation
	

For each timestep, count the total number of trees and fires using the same approach you used to Task 1 and 2. Duplicate the legend, titles, and axes labels as seen below. Remember that the curve on your graph might look slightly different each time:



Deliverables:

1. Animation
2. Time evolution figure

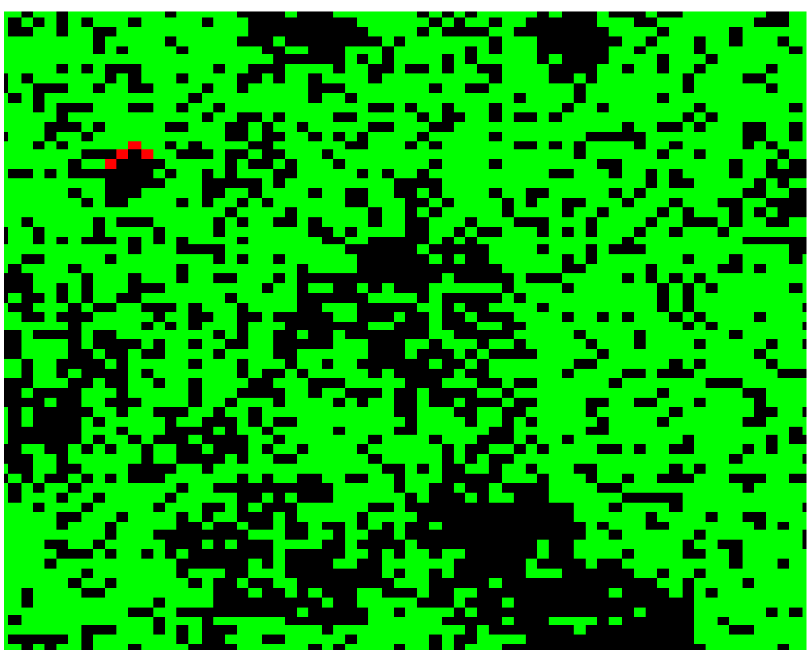
Task 3 Filename: part_3.m

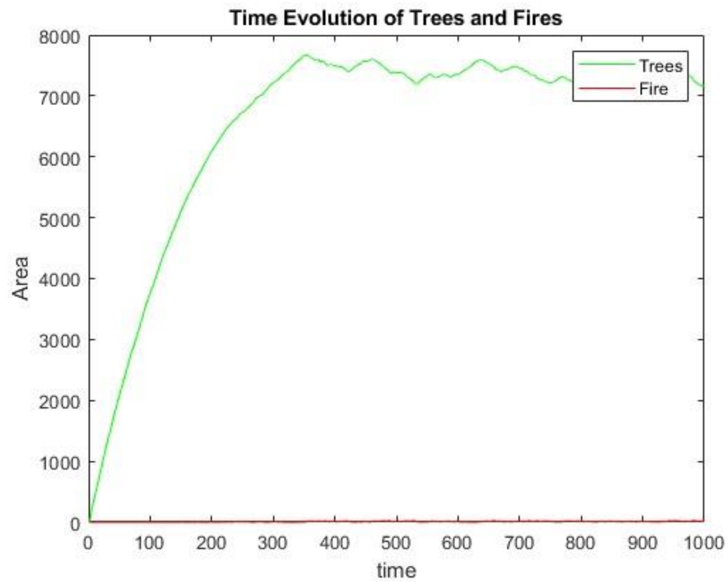
Task 4: Combining steps

Combine your code from Task 1-3 to create the complete fire model.

Complete Model steps:

1. Make a completely empty (black) grid of cells of size n x n. Set n=100
2. Loop over time for 1000 time steps
    a. Loop over the cells in the grid (excluding the edges)
        i. Spread fire with probability pspread=0.5 using the approach from Task 3.
        ii. Remove old fire cells from the previous time step
        iii. Start fires on green cells with probability pfire= 0.00005 using the approach from Task 2
        iv. Grow trees on empty cells with probability ptree=0.005 using the approach from Task 1
        v. Count the total number of trees and fires
        vi. Show the current grid of cells, pausing for 0.005 afterwards
3. Plot the total number of trees and fires as a function of time.





Deliverables:

1. Animation
2. Time evolution figure

Task 4 Filename: part_4.m

Task 5: Adding the ocean

Now that you have a working model of the fire dynamics, it's time to overlay the model on top of a map of the Santa Barbara and Ventura coastline. Load the 'SB_map.mat' file. The file contains a matrix called B0, which is a matrix with 1s where the ocean is and 0s where there is land (see the below image). B0 will serve as the blue values for the simulation's animation. Create 2 additional matrices of zeros of the same size as B0 to serve as the fires and trees (red and green pixels).

Remember that ocean cells cannot have trees or fires on them (rule 5). Copy and modify your code from Task 4 to check if there is ocean at a given pixel. If there is not ocean, then your code should continue as it does in Task 4.

Deliverables:

1. Animation

Task 5 Filename: part_5.m

Summary of Deliverables:

Task 1:
Filename: part_1.m
This file should create an animation and create a figure showing number of trees vs time

Task 2:
Filename: part_2.m
This file should create an animation and create a figure showing number fires vs time

Task 3:
Filename: part_3.m
This file should create an animation and create a figure showing number trees and fires vs time

Task 4:
Filename: part_4.m
This file should create an animation and create a figure showing number trees and fires vs time

Task 5:
Filename: part_5.m
This file should create an animation

Each part of the project should have a separate file submitted corresponding to it. There should be a total of 5 files submitted.

Warning: the use of ChatGPT is not permitted. We will know if you used ChatGPT