



Midterm Exam

INSTRUCTIONS

Complete the exercises below and upload them to Canvas as a single MATLAB script file (.m) using the naming convention “ENGR131_21F_Midterm_abc123.m”, replacing abc123 with your Case ID, and ## with the two-digit lab number.

For example, if Dr. Williams were submitting this test, it would be ENGR131_21F_Midterm_mrw8.m

For your script, please perform the following:

1. Separate each question into separate, runnable sections using the “%%” comment notation.
 2. You may use the code and notes from class, the textbook, and MATLAB’s documentation.
 3. Use comments as appropriate to indicate your thoughts and how your code works (or is supposed to work).
- This is 10 points (10%) of your grade.

QUESTIONS

There are 3 questions for this exam.

1. SOLVING SYSTEMS OF EQUATIONS (20 PTS)

There's a chill in the air and Fall is here. There are so many neat things that make Fall so special, as shown in Figure 1. Solve for the value of each Fall icon using a systems of equations approach. **DO NOT use the symbolic solver (you will receive 0 points if you do).** You will know you have the right answer as the correct values in the correct order, when rounded, transposed, and converted to characters, should spell out a recognizable word. Return your solution values to the Command Window.

$$\text{Apple} + \text{Apple} + \text{Tree} + \text{Football} + \text{Football} = 378$$

$$\text{Leaves} + \text{Leaves} - 580 + \text{Tree} + \text{Tree} + \text{Leaves} + \text{Football} = - \text{Grapes}$$

$$\text{Pumpkin} + \text{Pumpkin} + \text{Leaves} + \text{Pumpkin} - 401 = - \text{Leaves}$$

$$\text{Tree} + \text{Grapes} + \text{Grapes} + \text{Leaves} + \text{Leaves} + \text{Tree} + \text{Grapes} + \text{Grapes} + \text{Tree} = 740$$

$$\text{Apple} + \text{Apple} + \text{Pumpkin} + \text{Pumpkin} + \text{Football} + \text{Football} + \text{Pumpkin} = 531$$

$$\text{Leaves} + \text{Pumpkin} + \text{Apple} + \text{Tree} + \text{Grapes} + \text{Football} = 474$$

Key:



Apple



Football



Grapes



Leaves



Pumpkin



Tree

Fig. 1. The themes of Fall

2. COMPUTING VOLUMES (20 PTS)

A particular conical red cup is often found at many social gatherings in college filled with apple cider. The dimensions of this cup can be found in Figure 2a. Based on these dimensions, use Matlab to determine the locations at which you would place markings for 5, 9, and 16 oz. of fluid such that the volume at each height computed would match each target volume. Use the method of computing the volume of a revolved solid using a step size of 0.15 in. and iterating the volume calculation (your choice of method) with an increasing height until the correct volume is reached. Report these computed heights to the Command Window and replicate Figure 2b.

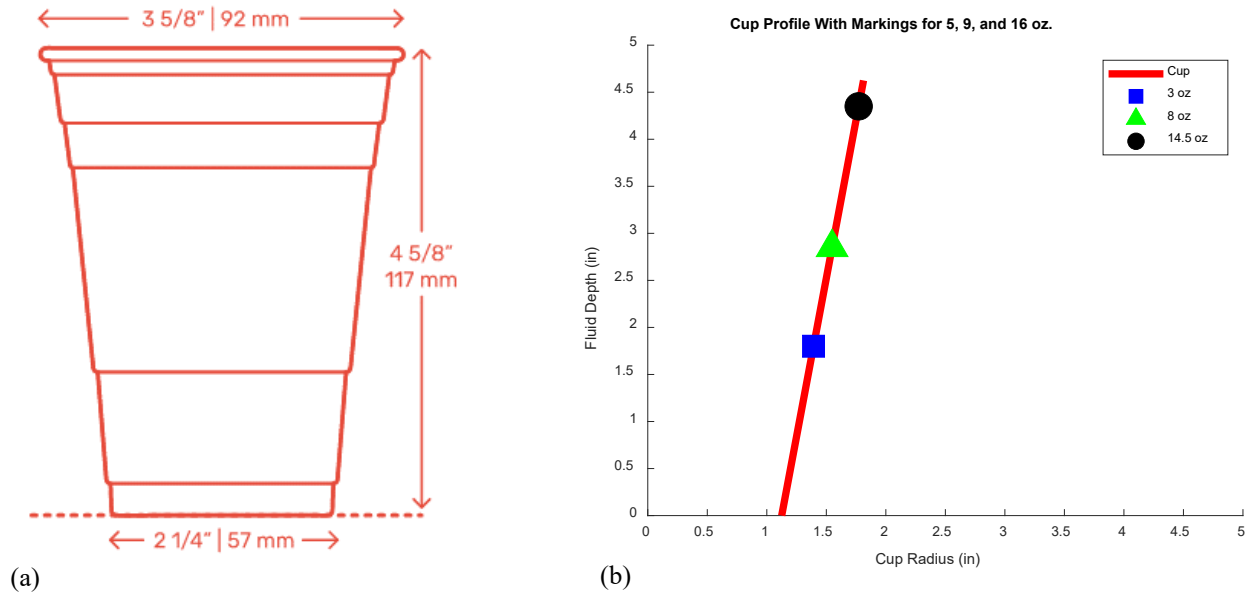


Figure 2. Dimensions of popular party cup (a), and profile of cup with the height of each target volume shown (b).

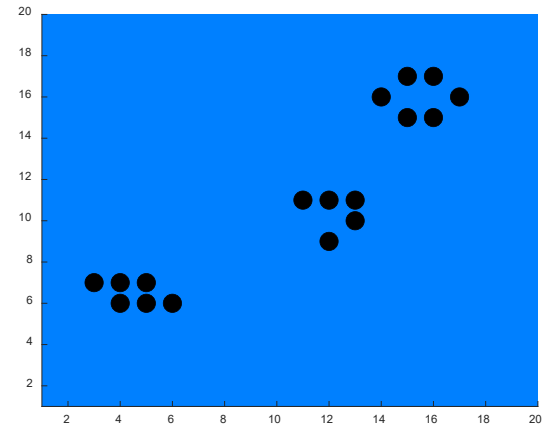
3. CELL SIMULATOR (50 PTS)

Program a basic simulator that uses a few simple rules to govern whether cells in a “dish” persist or die and when new living cells appear. At each step in your simulation it should show those cells that are going to die and those that will become live with red ‘x’s and green stars respectively as shown in Figure 3b before ending the step with the current cells as in Figure 3c. Show living cells as black circles and spaces with no cells as empty as shown. The cell life/death rules are as follows:

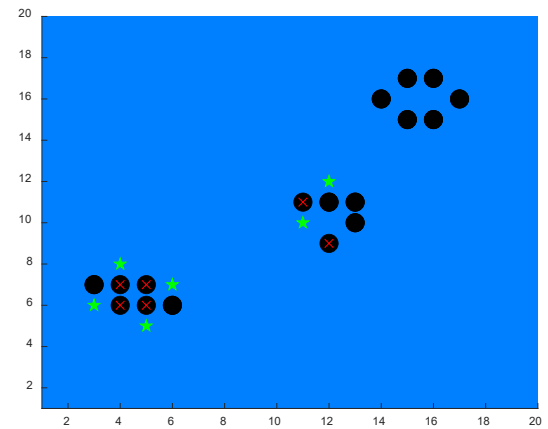
- Any live cells with fewer than two live neighbors die, as if by underpopulation.
- Any live cell with two or three live neighbors lives on to the next generation.
- Any live cell with more than three live neighbors dies, as if by overpopulation.
- Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

Use Matlab to program this simulator by doing the following:

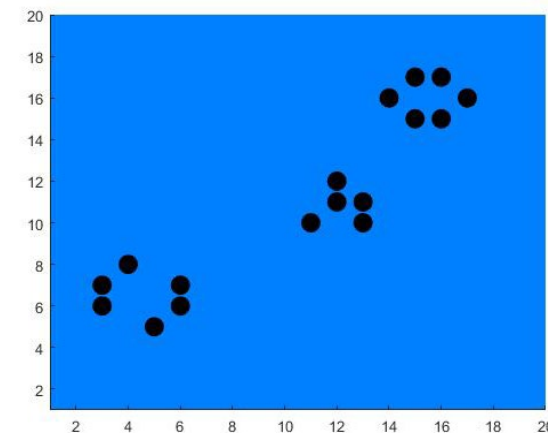
- A) Load the file StartingCells.mat file to load the map of the starting live cell locations (1 means a live cell, 0 is no cell).
- B) Use a loop to get commands from the user with a prompt with instructions to press s or q to step the simulation or quit:
 - a. When the user presses ‘s’, step the simulation forward to show the cells that will be transitioning (dying or appearing), pause for 0.5 seconds, then update the plot with the current cells.
 - b. When the user presses ‘q’, exit the simulation.
 - c. For all other entries, remind the user to enter only the two approved entries.
 - d. The simulation should continue until the user exits.
- C) Use the following functions to keep your code modular:
 - a. A function to implement the rules above to determine the fate of each cell.
 - b. A function to plot the cells and whether they are dying or newly appearing. The background should be bluish, but not the “stock” blue color. The orientation of the plotted cells should look like the data in the map variable.



(a)



(b)



(c)

Fig. 3. First step of the Game of Life for this problem showing starting locations (a), first transitions (b), and new current cells (c).

Hints:

- Don't worry about the edges (first and last rows and columns).
- This may seem daunting but consider mapping out your thought process. Most of the operation is little more than selection statements and a few loops.
- Consider how an empty cell is portrayed – is it plotted as empty, or as a marker that blends into the background such that it only *looks* empty?
- For more interesting behavior, check out the file StartingCells2.mat and StartingCells131.

Revision	Description	Date
A	Original Document	9/23/2021