You must work either on your own or with one partner. If you work with a partner you must first register as a group in CMS and then submit your work as a group. Adhere to the Code of Academic Integrity. For a group, “you” below refers to “your group.” You may discuss background issues and general strategies with others and seek help from the course staff, but the work that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is not OK for you to see or hear another student’s code and it is certainly not OK to copy code from another person or from published/Internet sources. If you feel that you cannot complete the assignment on your own, seek help from the course staff.

Objectives

Completing this project will solidify your understanding of for-loops, while-loops, and nested loops (Chapters 2 and 3 in Insight) and introduce you to the basics of user-defined functions (Chapter 5). You also will continue to explore MATLAB graphics.

Ground Rule

The use of arrays and the break command is not allowed. In fact, do not use the break command in this course.

1 My calendar

Write a script monthCal that prints a one-month calendar. Your script should solicit input for the number of days in the month and the starting day-of-the-week. The output from an example run of the script is shown below (user input is shown in italics):

Number of days: 31
Starting day-of-the-week (1=Mon, 7=Sun): 2

Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

For the user’s convenience when inputting the starting day-of-the-week, Monday is day 1, Tuesday is day 2, …, Sunday is day 7. However, for printing the calendar you must use Sunday as the first day of the week. The output dates should line up neatly as shown above, but the format doesn’t have to be exactly the same.

Hints: (1) Use a for-loop to count from 1 to n where n is the number of days. (2) You need to set an appropriate “test” to determine when to begin a new line.

2 What is the probability of any triangle being acute?

(If one answers this question based on a survey of triangles drawn in mathematics textbooks, one would say “close to 100%!”) You will perform a computational experiment to find the answer. A triangle is acute if each of the three interior angles is strictly less than 90 degrees. In our experiment, we will generate n “random triangles,” determine how many of them are acute, and approximate the probability of acuteness as the number of acute triangles divided by n, where n is a large integer value.
Define a random triangle by generating two of the three angle values, \( a \) and \( b \). The third angle, \( c \), is then \( 180 - a - b \). In order to get truly random (i.e., uniformly random) triangles, \( a \) and \( b \) must be generated independently, each in the open interval \((0,180)\).\(^1\) However, we have to deal with the constraint \( a + b < 180 \). We can visualize the two variables, \( a \) and \( b \), on a 2-dimensional plane where the x-axis represents the possible values for \( a \) and the y-axis represents \( b \), as shown in the figure to the right. The yellow region encompasses the values of \( a \) and \( b \) where \( a + b < 180 \). Generating \( a \) and \( b \) independently in \((0,180)\) gives us values bounded by the magenta square, but we can transform any “point” \((\alpha, \beta)\) outside of the yellow region by reflecting it across the line \( a + b = 180 \) (the diagonal) into the yellow region. The reflected point \((\alpha', \beta')\) is given by \( \alpha' = 180 - \beta \) and \( \beta' = 180 - \alpha \).

2.1 Estimate the probability

Write a script `acuteTriProb` that estimates the probability of any triangle being acute using the method described above with \( n = 10000 \). Display the estimated probability as well as \( n \).

2.2 Histogram for a small number of trials

Complete the script `acuteTriHist` to build a histogram duration simulation. In each step of this simulation, generate a random triangle as described above, classify it as acute or non-acute (right or obtuse), add a color block to the appropriate column of the histogram, and add a pause of .1 seconds (`pause(.1)`) so that the simulation plays like a movie upon execution. The simulation ends when either column of the histogram gets `histMax` blocks or after `nMax` triangles have been evaluated, whichever occurs first. Three snapshots of the figure during an example run of the simulation are shown below.

The given skeleton (incomplete) code sets up the figure window and labels the two bars of the histogram to be drawn. Observe the use of the command `text`, which puts text on the figure. For example, `text(x,y,'hello')` puts the text “hello” on the figure at the position \((x,y)\). By default the text is left-justified. (Later you can read Insight Appendix A.6 to learn about controlling text display.) Use the given function `DrawRect` to draw each block of the histogram.

---

\(^1\) It is tempting to generate \( a \) in \((0,180)\) and \( b \) in \((0,180 - a)\), but this does not work for answering our question. Why?
Using a given function: Download the file **DrawRect.m** from the course website and put it in the same folder as your script **acuteTriHist**; then all the scripts in that folder can call function **DrawRect**. Function **DrawRect** takes five values as arguments: the x-coordinate of the lower-left (LL) corner, the y-coordinate of the LL corner, the width, the height, and the color. For example, to draw a blue rectangle with the LL corner at (3,5) that is 9 units wide and 4 units tall, you can write the command `DrawRect(3,5,9,4,'b')` in your **acuteTriHist** script. Here are the MATLAB predefined color names: 'b' for blue, 'c' for cyan, 'g' for green, 'k' for black, 'm' for magenta, 'r' for red, 'w' for white, and 'y' for yellow.

Each block of the histogram should have height 1 and you can choose the width and color; if necessary change the position of the text that labels each column. For program development and experimentation you can change the values of `histMax` and `nMax`, but change them back to the original values before submission (`histMax` is 10 and `nMax` is 15).

Submit your files **monthCal.m**, **acuteTriProb.m**, and **acuteTriHist.m** on CMS. Part B of Project 2 will appear in a separate document. Part B has the same deadline as Part A.