Matlab tutorial - Math 4450/5450 - Spring 2015 - Zirbel

Matlab is available in computer labs on the BGSU campus. To find which ones, go to When you start Matlab, the **command window** appears, with $\gg$ as the prompt. The commands below should be typed at the prompt. Don't type the line numbers or the comments at the ends of the lines. If you are using Octave

Matlab basics

1: `2*9` Matlab can be used as a calculator.
2: `sin(1)`
3: `2^999`
4: `x=sin(1)` You can store values with variable names.
5: `x` This displays the value of `x`
6: `x=rand(10,1)` `x` is a collection of 10 random numbers, uniformly distributed between 0 and 1. `x` is like a column of cells in a spreadsheet. It is called a **vector** or a 10 by 1 **matrix**.
7: `x+5` Add 5 to each entry of `x`.
8: `100*x` Multiply each entry of `x` by 100.
9: `ceil(100*x)` Round up to the next integer. This is how you generate random numbers uniformly distributed from 1 to 100.
10: `x=rand(10,7)` A 10 by 7 matrix of random numbers. To repeat this, or any other command, hit the up arrow as many times as you want.
11: `x=rand(100,1)` Go ahead, try something larger than 100.
12: `x=rand(100,1);` Use a semicolon to suppress the output.
13: `max(x)`
14: `mean(x)`
15: `sum(x)`
16: `median(x)`
17: `cumsum(x)` A vector of cumulative sums of entries of `x`.

18: `y=sort(rand(10,1))` Sort another 10 random numbers.

Matrices

19: `P = zeros(20,20)` Examine `P` to see what you get.
20: `sum(P)` Calculate the sums of the columns of `P`.
21: `sum(P')` Transpose first to calculate the sums of the rows of `P`.
22: `P(1,2) = 0.5` Set one value in the matrix `P`.
23: `P(1,1:3) = 1/3` Set three values in the matrix `P`.
24: `P(10,:)` Display row 10 of the matrix `P`.

Plotting points and histograms

25: `plot(x)` Plot the values of `x` against numbers 1, 2, ...
26: `hist(x)` Make a histogram of the values of `x`.
27: `hist(x,30)` Use 30 bins instead of the default 10.
28: `y=-log(x)` Take the natural logarithm of the entries of `x`.
29: `hist(y,30)` Most of the values of `y` are near 0, but some are as large as 6. The numbers in `y` have what is called the **exponential distribution**.
30: `x=rand(10000,1)` You typed a command similar to this a while ago. Type `x=` and then the up arrow to bring it back, then edit it to change 100 to 10000.
31: `hist(-log(x),30)` Looks better now.
32: `z=randn(1000,1);` Generate 1000 normally distributed numbers
33: `hist(z,30)` The histogram is bell–shaped. You might want to try this with more than 1000 numbers.
34: `help hist` To get help, type “help” followed by the name of a command. Sometimes
you can guess what you want and then read about it. Or google your question, like “matlab hist”.

Getting some Matlab programs

35: I have posted a number of Matlab programs on Github. To find them, google "Github zirbel random processes" or go to `https://github.com/clzirbel/Random_Processes` and find the Download Zip button or go straight to `https://github` `com/clzirbel/Random_Processes/archive/master.zip` to get the .zip file. Unzip the file and save it in an accessible place.

36: In Matlab, use the dropdown above the command window to set the working directory to the Matlab folder in the folder you downloaded from Github.

37: Open the file `gambler_outcome.m`. It will open in an editor window. It is a good idea to read the commands in it and start to understand what they do. I have added many comments to the commands in the file, separated from the commands by the % character.

38: `pwd` Figure out what Matlab considers to be the working directory. This is where it looks for programs and where it writes out files.

39: `gambler_outcome` Matlab will load the file and run the commands in it. When you are done running the program, press Control-C to break execution.

40: Run the program `gambler_outcome` multiple times, then quit. In the program, find the line that pauses after each step, set the pause time to 0.0, and then run it again. Now each game will take less time to generate and draw, and you can get a better sense of what they look like.