Lab Assignment # 2 – MATLAB Functions

Due: Thurs. Sep. 18, 2008

Directions: In this assignment you'll practice writing MATLAB functions and scripts. You may want to read the lab notes for Thursday Sept. 11. In addition, each problem includes a list of MATLAB built-in functions that may be useful, use the `help` command or MATLAB's help browser to learn about those functions.

You can work with others and discuss the problems, but each student must write his/her own, independent solution. If you are unsure about what i mean by this, please ask!

What to turn in? Each assigned problem specifies sample outputs you should produce and submit. The solution to each problem should include a print out of the function / script file followed by the specified output (*e.g.*, a plot, a vector, a matrix, etc.)

Problem 1. Write a MATLAB function called `stats` that takes as *input* a vector `x`, and returns two *output* values: the mean value of the elements of `x`, `x_bar`, and their standard deviation, `std_dev`, given respectively by

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \qquad \text{and} \qquad \sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}},$$

where n is the lenght of `x`.

Sample output:

1. create a vector `x` of 100 uniformly distributed random numbers (no need to turn this one in), and use the function `stats` to find the mean and standard deviation of `x`
2. create a vector `y` of 100 normally distributed random numbers, and use the function `stats` to find the mean and standard deviations of y
3. plot the vectors `x` and `y` in the same plot (using a different plot line style for each)

Remark: verify your result using the built-in MATLAB functions `mean` and `std`

Suggested MATLAB functions/operators: `.^`, `hold on`, `plot`, `rand`, `randn`, `sum`

Problem 2. Write a MATLAB script that creates a 5×2 uniformly distributed random matrix, A, and uses the function `compass` to plot the rows of A as vectors from the origin to the point $(a_{i1}, a_{i2}),\ i = 1, \ldots, 5$.

Sample output: a random example of your results.

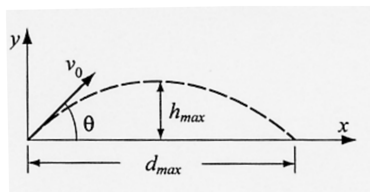
Problem 3. The equations

$$v_x = v_o \cos\theta, \quad \text{and} \quad v_y(t) = v_o \sin\theta - g\,t = v_{yo} - g\,t$$

describe the velocity of an object launched from the point (x_0, y_0) with velocity v_o in the direction given by θ. and

$$x(t) = x_o + v_x\,t, \quad \text{and} \quad y(t) = y_0 + v_{yo}t - \frac{1}{2}g\,t^2$$

its position at time t.



The maximum height reached by the object and the time at which it reaches that height, are given by

$$h_{max} = \frac{v_{yo}^2}{2\,g} \quad \text{and} \quad t_{h_{max}} = \frac{v_{yo}}{g}.$$

(a) Write a MATLAB function, `parabolic`, that takes as *input* v_o and θ and produces as *output* two lines of text displaying the information "`the maximum height reached by the object was h_max`" and a similar one for `t_hmax`, and three plots: (1) one with the position $x(t)$ vs. t, (2) one with the position $y(t)$ vs. t, and (3) one that displays the trajectory of the object (plot of y vs. x), using N points over the time interval $[0, \texttt{t\_final}]$.

(b) Repeat part (a), but using a script file instead of a function file. That is, no input arguments are passed when calling the procedure, the script should prompt the user to enter them.

Sample output: turn in the results obtained with $v_o = 200\,m/s$ and $\theta = 43°$.

Suggestions:

1. Before you start, you should figure out how to calculate `t_final` given v_o and θ.
2. You can use $g = 9.81\,m/s^2$, and $(0, 0)$ as the initial point.

Suggested MATLAB functions/operators: `input`, `linspace`, `plot`, `sprintf`