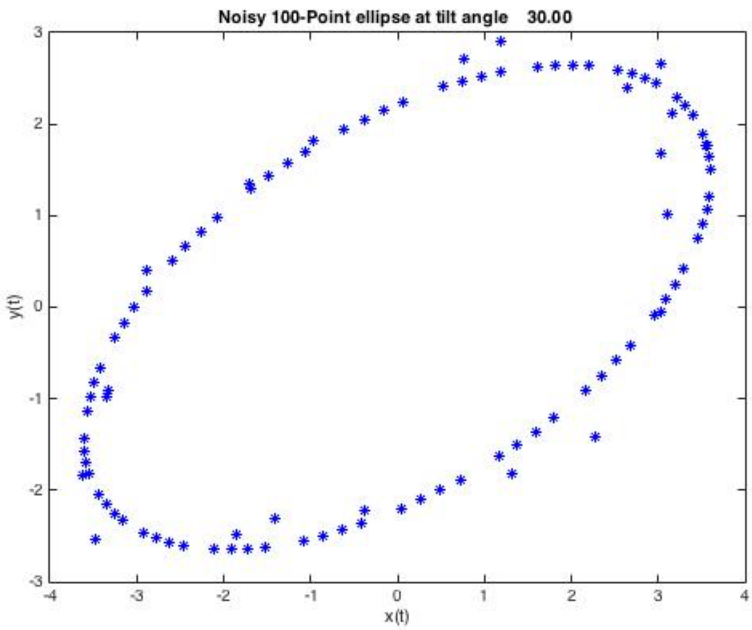
Chem/Stat3240: Homework 3b
Matlab

September 16, 2015

3 Modify the function `myEllipse` written in problem 2 to a function `myEllipseRand[theta_,n_,nNoise]` that generates the x an y coordinates as before, but adds a random x and y perturbation to a coordinate pair at intervals specified by the input nNoise. To do this you will need to incorporate a loop that uses a conditional to detect every $\texttt{nNoise}^{th}$ sample of the loop index variable k (including k=1). When the condition is true, use the `rand` command to add separate uniform random noise (-0.5 to 0.5) terms to the computed x_k and y_k. Create a plot for $\theta = 30, \texttt{nPoints} = 100$, and $\texttt{nNoise} = 5$ with the x-axis labeled '$x(t)$', the y-axis labeled '$y(t)$', and the title 'Noisy n-Point Ellipse with Tilt Angle theta', where n and theta are replaced by their actual values. Use the `sprintf` and `num2str` commands to create a text string for the `title` command to accomplish this. The output of the function `makeEllipseRand` will be a vector of x coordinates and a vector of y coordinates. The code should programmatically save the plot as a pdf file (see `saveas` command) named `myEllipseRand`.

The following figure is what the output plot should look like, including points markers and connecting lines (see plot options).



Upload your completed function to Cody as well as to the collar site, along with the test suite and a saved pdf file of your plotted ellipse.

4 Calculus tells us that for very small positive values of h,

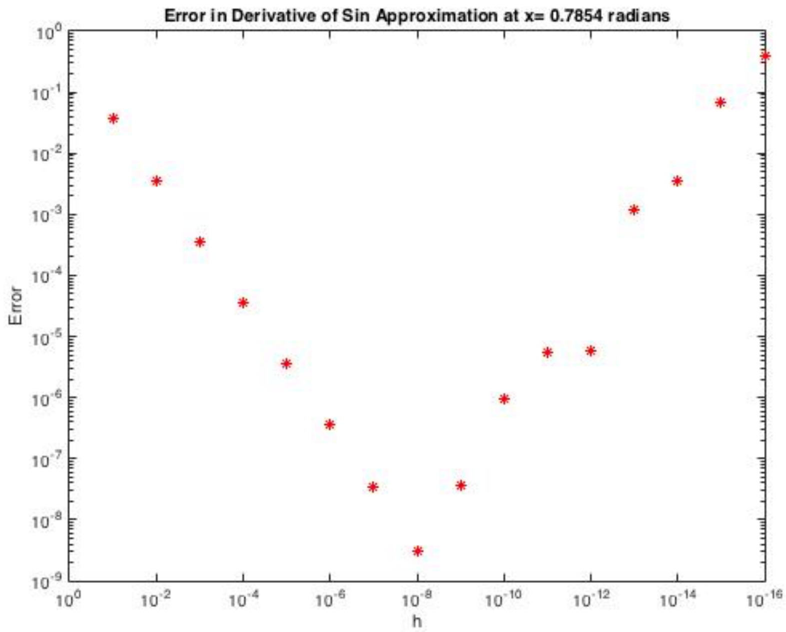
$$e_h(x) = \left| \frac{\sin(x+h) - \sin(x)}{h} - \cos(x) \right| = O(h)$$

That is, the approximation to the derivative of $sin(x)$ approaches the true derivative ($cos(x)$) for small values of h, where the "big-O" notation denotes that the error $e_h(x)$ is on the order of h .

Complete the function template for `sinDerivative(x)` which takes an input x in the range $[0, 2\pi]$ and computes the values of $e_h(x)$ for $h = 1/10, 1/100, \ldots, 1/10^{16}$, and determines the h that minimizes the error e_h (`h_best`) and the associated e_h (`e_best`) for outputs of the function `sinDerivative`. Determining `e_best` and `h_best` should be done with a conditional inside a loop that indexes over all values of h.

Note that in the evaluation of the divided difference, any errors in the evaluation of $\sin(x+h) - \sin(x)$ are magnified by $1/h$. Thus, as h goes to zero, the "calculus" error goes to zero but the roundoff error goes to infinity. Thus, the "optimum" choice of h reflects the need to compromise between these two tendencies.

You could use the `logspace` command and array-based inversion to generate the values for h. Remember to preallocate any vectors before indexing their elements inside a loop. Create a `loglog` plot of the approximation error versus h as show below with labels and titles. To get h to go from high to low values, use the command `set(gca,'XDir','reverse')`. Again use sprintf to create a plot title specific to the input x. Programmatically save the displayed plot to a pdf file (see `saveas`) with file name `derivativeError`. The plot for $x = \frac{\pi}{4}$ is shown below. Submit the pdf file of the plot to collab, as well as the code file. Submit the code to the Cody its as well.



5 Create a function `sinDerivativeVec(x)` that performs the same computation as `sinDerivative(x)`, but does so using vectorized operations and the `min` command, rather than the `for` loop and a conditional. Submit the code file to Cody as well as to the collab site.