Homework Project #1
Math 365, Spring 2013

Due Wednesday February 6th

All assignments are to be done in Matlab. It is recommended that you use the Matlab “Publish” command to present your work. This assignment is out of 100 points, but you will notice that you can only get 90 points if you do not do any of the Bell and Whistle (B&W) problems. Each Bell and Whistle is worth 5 points (unless otherwise stated). The following concepts are covered in this assignment.

- Plotting elementary functions and their composites,
- Modifying various attributes of the plot, including the axis window and the labels.
- Using function handles,
- The for loop
- Writing scripts
- The input, ginput and pause commands

1. (20 points) Using functions available in Matlab and the arithmetic operations we have discussed in class, find at least three different expressions (not already covered in class) that result in NaN values. To get full credit, you must try to explain in words why you believe your expression evaluates to NaN.

Bells and Whistles
(a) For each unique and clever way you come up with that nobody else in the class found, you’ll receive a bonus two B&W points point.

2. (20 points) Search the web to find a non-trivial continued fraction approximation to \(\pi\) using only positive integers in the expression. Using Matlab, evaluate the continued fraction with enough terms so that your expression has at least 12 correct digits. Turn in the Matlab expression you used to evaluate \(\pi\), and your resulting approximation to \(\pi\). Format your answer correctly so that you can see 16 digits of your result.

Bells and Whistles
(a) The Online Encyclopedia of Integer Sequences provides an integer sequence that can be used in a continued fraction expression for \(\pi\). Can you write a script that evaluates \(\pi\) using a user input number \(n\) entries in this sequence? Google OEIS A001203 to get the sequence for \(\pi\). Turn in a printout of your code, along with the results of the first 7 entries in the convergent sequence for \(\pi\).

3. (25 points) Given the two functions \(f(x)\) and \(g(x)\) below, you are going to construct a composite function \(h(x)\) and its derivative.

\[ f(x) = \sin(4x) + 2, \quad g(x) = \cos(e^x), \quad h(x) = f(x)g(x) - 1.25 \]

Write a Matlab script that does the following.

(a) Create function handles \(f\) and \(g\) for \(f(x)\) and \(g(x)\). Use these to create a function handle for \(h(x)\).
(b) Plot \( h(x) \) over the domain \([-1, 2.5]\). Add labels to your plot.

(c) Find (using pencil and paper!) the derivative \( h'(x) \) in terms of the derivatives \( f'(x) \) and \( g'(x) \) and the original functions \( f(x) \) and \( g(x) \). \textbf{Hint:} Use logarithmic differentiation!

(d) Create function handles for \( f'(x) \) and \( g'(x) \) (call them \( \text{fp} \) and \( \text{gp} \), for example). Use these to create a function handle for \( h'(x) \) (call it \( \text{hp} \)).

(e) Use the derivative \( h'(x) \) to create a function \( T(x,a) \) which is the equation of a tangent line to \( h(x) \) at the point \( a \). Plot tangent lines to the original curve \( h(x) \) at three different values of \( a \) in \([-1, 2.5]\).

(f) Plot a symbol (\( * \), \( o \), etc.) on the curve at the three points of tangency.

Turn in a plot of the original function \( h(x) \), with the three tangent lines, and a second plot of the function \( h'(x) \).

\textbf{Bells and Whistles}

(a) Instead of plotting a tangent line over the whole \( x \) domain, choose a subset of the full domain \([-1, 2.5]\) so that the tangent line is of fixed length and it looks more like a skateboard or a ski sliding along the curve. Animate the ski using a \texttt{for} loop to loop over a sequence of points along the curve, plotting the tangent line segment at each point. Turn in three plots showing the ski in three different locations.

(b) Using a graphical means, plot symbols on the graph of \( h(x) \) showing exactly where the graph goes from being concave up to concave down. \textbf{Hint:} Use \( h''(x) \).

4. (25 points) In this problem, you are asked to plot an \( n \) sided irregular polygon and computes its area. Write a script that does the following.

(a) Prompt the user for \( n \), the number of sides in the polygon they wish to construct.

(b) Use the \texttt{axis} command to open a figure window with axis limits \([-2,2,-2,2]\).

(c) In a \texttt{for} loop, prompt the user to click the screen inside of the figure window to input the location of a vertices in counter-clockwise order. Plot a symbol of your choosing at each vertex location as the user clicks the screen. Do this for all \( n \) vertices. Store the vertices in two arrays \( x \) and \( y \). Plot the outline of the polygon.

(d) Use the formula below to compute the area of the polygon.

\[
A = \sum_{i=1}^{n} \frac{(y_{i+1} - y_i)(x_{i+1} + x_i)}{2}
\]

Assume that the vertices “wrap” so that \( x_{n+1} = x_1 \) and \( y_{n+1} = y_1 \).

(e) To see that you are getting the correct answer, compare your results with that Matlab function \texttt{polyarea} (try \texttt{help polyarea} to see how to use this function) You should get exactly the same answer as \texttt{polyarea}.

Turn in the program you wrote to compute the area, a plot of one test problem, and show that you get the same area as that computed by \texttt{polyarea}.

\textbf{Bells and Whistles}

(a) Compute the area of the polar “flower”, given by the parametric equations

\[
x(t) = (6 + 4 \cos(5t)) \cos(t)
\]
\[
y(t) = (6 + 4 \cos(5t)) \sin(t)
\]

Turn in a plot of the polar flower, and the resulting area you get. \textbf{Hint:} Compute \( x \) and \( y \) using \( t = \text{linspace}(0,2\pi,201) \).