ENGR 102
Lab #12

Activity #1: Plotting Curves and Derivatives – to be done as a team

As a team, you are to develop a program that will ask a user for a polynomial (of arbitrary degree) and then will generate a plot of the curve, its derivative curve, and its second derivative curve. All three should be plotted on the same axes so that it is easy to see how the first and second derivatives affect the shape of the curve. You are also to highlight local maxima and minima by drawing points on the plots showing where these occur. You may use code developed in prior weeks if it is helpful (but it will probably be as easy, at this point, to write new code).

- The user should be able to enter the polynomial as a set of coefficients. You will need to be sure to clearly tell the user the format to enter these coefficients in.

- You should determine the derivatives analytically. That is, if you have a list of the coefficients of the original polynomial, you should be able to generate the first and second derivatives exactly.

- The plots can be over a predetermined range of x and y values, but should be sufficiently large that someone entering a polynomial would easily be able to generate polynomials passing through the area covered by the plots. Your plots can be generated by sampling many points on the curves and plotting a line connecting those points.

- You should identify the local maxima and minima in the range you are plotting. You can do this by looking through the points generated for your plot, and finding ones that are local maxima (a point is greater than those just before and after) and local minima (a point is smaller than those just before and after). These points should be plotted on top of the plots of the curves.

- Your plots for the three curves and the maxima/minima should be sufficiently varied (in color/shape) that it is easy to distinguish one from another.

Follow the following process: Complete parts a and b before writing any code!

a) First, you will practice generating a bottom-up design:
    a. As a team, brainstorm what functionality you are likely to need in this program. For each basic task your team comes up with, where all the pieces needed are already known, write down a short function description, along with any input arguments needed, and what return value there will be, if any (i.e. construct a document, which will also be used for the subsequent design descriptions).
    b. Only do this for basic functions for which the steps you will need to follow are well-known or obvious. You can consider functions that will call other functions, as long as you have first designed the other function.
    c. Your goal in this part is not to generate a complete code design, but rather to identify functions to implement that are likely to be helpful in your program.
b) Next, create a top-down design of the program.
    a. Put together a document (can be added to the one from part (a)) outlining the top-down design that your team comes up with.
    b. You should make use of the functions created in the bottom-up “phase”, above, to whatever extent seems helpful.

c) Next, code up your solution.
    a. You should create separate functions for each of the "nodes" in the top-down hierarchy that you created.
    b. You should also create separate functions for the features you identified in the bottom-up phase (a), above.
    c. Note that you might want to split up the coding responsibilities for the different functions among different team members. If you did a good job with design, the individual functions should be able to be coded up independently of one another.
    d. As a team, you will need to decide what variables you will use in the main program
    e. Be sure to include docstrings for all the functions you create

Activity #2: Integrating a function – to be done as a team

This activity is meant to familiarize you with the process of integration by computing area under a curve. As you may have learned by now in calculus, to integrate a function over a range of values, we can compute the area under the curve, when plotted in a graph. To do this, we essentially divide up the area under the curve into a number of simple shapes (like rectangles) and sum up the area of those shapes; as we use more and more of the shapes, we get a more and more accurate estimate of the area. For this example, we will use rectangles and trapezoids.

You are to write a program that computes the area under a function from one value to another. You should allow this to be an arbitrary function.

- Begin by outlining your program. Similar to part 1, you should identify any functions that you want to include (this program should be simpler than the one in activity #1). Put together a document listing the functions you will use in your program.
- You should take in as input a starting and ending value of the range that you want to integrate over.
    - You may, optionally, take in input for the function itself that you want to integrate. (If you choose to do this, see the note on "eval" in Activity #3.)
- You should integrate the function by dividing it into a number of elements, computing the area for each of those elements, and summing up those areas.
- Your program should begin by dividing the range into 10 elements, and then repeatedly increasing the number of elements until the area converges.
    - You may assume convergence if consecutive area calculations differ by less than 10^{-6}.
- For each of the shapes, your program should report both the area computed, and how many iterations it took to achieve convergence.
- You should compute area using 4 different shapes/methods:
    - A rectangle whose height is the value of the function at the beginning of the interval
    - A rectangle whose height is the value of the function at the end of the interval
    - A rectangle whose height is the value of the function at the midpoint of the interval
    - A trapezoid with one parallel edge being the value of the function at the beginning of the interval, and another being the value of the function at the end of the interval.

  For each of these, calculate the area for a particular subinterval –for the rectangles, the width will be the interval width, for the trapezpoid, that will be the "height" of the trapezoid.