Project Proposal: Plot Twist

Ray Boppana

The goal of this project is to create a user-friendly graphing calculator that is more sophisticated than a standard graphing calculator. Specifically, I want my program to have the ability to graph any valid function that a graphing calculator can plot. Furthermore, I want my program to have the ability to return the derivative of any function that is inputted. Furthermore, I want to give the user the ability to graph in polar and parametric easily. I plan on graphing parametric graphs in real time! I also want to approximate integrals of functions with Taylor series approximations and calculate intersections between graphs using Newton’s law of approximation. If there is time, I would like to explore the possibility of adding a regression feature which calculates the linear regression of a set of data points.

In order to accomplish all of this, I need to create classes for all of the elementary math terms (power term, exponential, trig terms, inverse trig terms, log, etc…). Each class will have the ability to return an (x,y) coordinate as a tuple for a given x which will be called multiple times when graphing the function. Each class will also have the ability to return its derivative. These elementary terms can then be used to make more complex functions through multiplication, division, addition and function composition. (for example, sin(x^2) is really a power term inside a sin(x) term). Calculating derivatives of these more complex functions can be done recursively through the chain rule and calling the derivatives of the elementary terms appropriately. A tricky part of the project will be creating the parseFunction(s, var) method which given a string s which represents a valid function and a variable var will return the correct function represented as an object. I will need to spend serious time breaking down the steps of how to parse a function, apply PEMDAS and return the function object. Polar and parametric modes shouldn’t be too hard to code once I have gotten the Cartesian code working. Implementing the regression calculator will require me to look up how a regression is calculated based on a set of points and coding the formula. The same applies for calculating the r, r^2 and the t score of the regression. No modules outside tkinter will be used for the project.

Update1:

No t score for regression will be implemented.

Update2:

* Integrals are not implemented.
* The project now has scrubbing ability for Cartesian and statistical modes
* Program can parse math constants such as pi and e
* User can click on intersection points in Cartesian mode or plotted points in stats mode to see their value.