PROJECT REPORT

Name of the project: advanced scientific calculator

Purpose of the project: It is a mathematical application designed by us for mathematicians, physicists and chemists who quite frequently encounter cumbersome and complicated integrals , solutions to complicated equations ,derivatives etc. Using our project one can do these tasks very easily .Our project is inspired by Wolfram Alpha .

Functionalities:

- Integrator
- Differentiator
- 2D grapher
- Quadratic equation solver
- Cubic equation solver

Algorithm to various functionalities:

1. Integrator-
   We are using Riemann Integration for evaluating definite integral and area. It is found by dividing the given interval into small sized partitions(rectangles).The value of the integral is the sum of the areas(with the correct sign) of all the small rectangles, obtained by multiplying the size of the partition with the average value of the function in the partition. It is positive or negative depending upon the sign of the function. For area whenever the function is negative we take its absolute value. We take a function as an input in the form of string and parse it using "fparser" library. We also take the interval in which the integral or the area is to be evaluated and then compute the integral or the area as explained above. However we are facing problems when the function is approaching infinity.
2. Differentiator-
   We are using the first principle of differentiation that is f'(x) = lim h->0 {f(x+h)-f(x)}/h. RHD=lim h->0{f(x+h)-f(x)}/h LHD=lim h->0{f(x)-f(x-h)}/h
   Here h>0 and function is only differentiable iff RHD=LHD
   We take a function as an input in the form of string and parse it using "fparser" library. We also accept the value of the point at which derivative is to be computed. Here we take h=0.00001 and evaluate the value of the function at a point using eval() function of fparser class. We calculate the RHD and LHD and check if they are equal or not.If they are equal the value of the derivative is

their average, otherwise the function is non-differentiable. Maxima is obtained at a point where right hand derivative (R.H.D) < 0 and left hand derivative (L.H.D) > 0. Minima is obtained at a point where right hand derivative (R.H.D) > 0 and left hand derivative (L.H.D) < 0. Point of inflection is obtained when R.H.D and L.H.D both tend to zero and are of the same sign. We take a function as an input in the form of string and parse it using "fparser" library. We also accept the interval in which the maxima, minima and inflection points have to be located. We traverse the entire interval by dividing the entire interval into small sized partitions and compute the derivative at closely spaced points. Using the above conditions we check whether there exists any point of maxima, minima or point of inflection.

3. 2D grapher-

   To make the graph of a function we input the domain under which the graph is to be drawn. Now we consider points on the x axis with small difference as of the order of 10^-5 . evaluating the values of the function on closely spaced x values we get corresponding points on the canvas. Now the graph is drawn by making small lines joining the consecutive points using the function for line and then the imprint function.

4. Quadratic equation solver-

   The roots of the quadratic equation ax^2 +bx +c are given by the quadratic formula $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$. The roots of the equation are imaginary if the value of the expression b^2-4ac is negative.

   Cubic equation solver- The coefficients of the cubic equation are taken as input from the user. Numerical methods are used. Firstly, the derivative test is used for finding the number of real roots of the cubic equation. Then we locate the real roots of the equation by checking its value on points spaced apart by 1.Wherever there is a change of sign, we stop, and divide the interval into 2 parts, and then checking the value of the function at the mid-points. This process continues till we obtain a point where the value of the function is very close to 0.

5. Cubic equation solver-

   The coefficients of the cubic equation are taken as input from the user. Numerical methods are used. Firstly, the derivative test is used for finding the number of real roots of the cubic equation. Then we locate the real roots of the equation by checking its value on points spaced apart by 1.Wherever there is a change of sign, we stop, and divide the interval into 2 parts, and then checking the value of the function at the mid-points. This process continues till we obtain a point where the value of the function is very close to 0.

TROUBLES FACED:

- We had little a bit of problems while making the code of cubic equation solver. It took time to come up with a perfect code that deals with all the situations.

- We do have difficulties while converting input strings by user to mathematical forms. We found fparser.hh library but could not add it to the codeblocks directory. So we decided to make our own parser.

- We were having difficulty in making the part dealing with the parsing of composite functions. However, the major problems arose while parsing linear combination/products of functions.

- Also we have problems in finding the definite integrals/derivatives/graph plotting of discontinuous functions.

- We also faced a lot problems while making the menu in simple cpp and linking it with the grapher code, integrator code, etc. The main problem was taking input from user inside the simplecpp window because once we returned to main from simplecpp window it was not possible to reinitialise the simplecpp window. So we had to figure out a way to take input and convert inside simplecpp window.

- We also faced a lot of problems while plotting discontinuous functions and functions which tend to infinity at finite poins such as 1/x at 0, tanx at pi/2 etc.The program crashed when the function approached infinity. So we had to develop a code to deal with such situations too.

The coefficients of the cubic equation are taken as input from the user. Numerical methods are used. Firstly, the derivative test is used for finding the number of real roots of the cubic equation. Then we locate the real roots of the equation by checking its value on points spaced apart by 1.Wherever there is a change of sign, we stop, and divide the interval into 2 parts, and then checking the value of the function at the mid-points. This process continues till we obtain a point where the value of the function is very close to 0.