

Integration Assignment

Aniket Pratap

January 16, 2022

Abstract

The main point of this assignment is to create our own small math library containing our approximation definitions of: $\sin(x)$, $\cos(x)$, $\log(x)$, $\exp(x)$ – euler's constant, and \sqrt{x} . Not only that, but the library also contains an integrate function which integrates using Simpson's 1/3 rule. This library is then connected to the main function, `integrate.c`, which takes in user input to integrate a series of functions in a file called `functions.c`. These functions can be anything with respect to the coded math library `mathlib.c`. User input is as follows: the user runs `./integrate`, created by a make file, along with flags from `-a-j` to represent the functions. The flags inputted are the lower and upper bounds specified by `-p` and `-q` and the partitions: `-n`. A typical call would look like this: `./integrate -a -p 0.0 -q 1.0 -n 10`. This means integrate function "a" starting from 0.0-1.0 using 10 partitions. The output should spit back the data the user inputted along with the different partitions and approximation for that number of partitions. This user input system could be done in many ways, but I am inclining towards using an array with false values. Everytime the user inputs a function, the array they changes to true and outputs the function. This array system is done so that multiple functions can be run at once. This program also contains a usage function and error handling for possible errors on the user's part. One way of error handling could be done by setting a default range if the user doesn't input anything for it. Another way of error handling can be done by changing the user's inputs. For example, if he inputs numbers that divides by zero, then zero will be swapped with $1e-14$.