Guillermo Alfaro
CSE13S Winter 2022

Numerical Integration Design Document

Description:

Numerical library using a series of mathematical functions. It will solve using Simpson's Composite ⅓ rule of a range provided n number of times.

Files:

1. functions.c:

   This is a provided file that didn't need editing.

2. functions.h:

   This is a provided header file that didn't need editing.

3. integrate.c:

   The file that has the main() function, is the file that you interact with.

4. mathlib.c:

   Math library has definitions for exponents, sin, cosine, sqrt, log, and integrate.

5. mathlib.h:

   This is a header file to go in mathlib.c.

6. Makefile:

   The file that makes cleans, formats, and compiles all files together into one executable.

7. README.md:

   Tells you how to use the provided files and what arguments they accept.

8. DESIGN.pdf:

   A blueprint on how to recreate this lab given the proper resources.

9. WRITEUP.pdf:

   Graphs with different amounts of partitions with explanations of graphs and anything notable learned on the way.

Pseudocode:

Mathlib.c

```
#include “mathlib.h”
#define EPSILON 1e-14
exp(x)
    Initialize all variables to double for precision purposes
    While trm > EPSILON
        Trm *= abs(x) / k
        Sum + trm
        K += 1
    If x > 0 return sum else return 1/sum
sin(x)
    Initialize all variables to double for precision purposes
    While abs(t) > EPSILON
        Perform sin calculation using taylor series
cos(x)
    Initialize all variables to double for precision purposes
    Compare sin pesdocode to get cosine (v, t = 1.0, 1.0)
    Format is similar to sin
sqrt(x)
    Initialize all variables to double for precision purposes
    Gonna have to improve with factoring
    F = 1 (multiply return value at end by F)
    While x > 1
        X /= 4
        F *= 2
    While abs(y-z) > EPSILON
        Z = y
        Y = .5 * (z + x / z)
    Return y * f
log(x)
    Initialize all variables to double for precision purposes
    F = 0 (add to return at end)
    While x > e
        x  /= e
        F += 1
    While abs(p-x) > EPSILON
        Y = y + (x/p) -1
        P = exp(y)
    Return y+f
integrate()
    Get composite ⅜ rule pseudo-code
    Instead of modulo 3 use 2
    Change sum += to be respective of Simpson’s composite ⅓
    Do a check if (i + a * h) is gonna be 0
        If it is change it to very small decimal 10e-100
```

Integrate.c

```
Include header files
Include <stdbool.h>
main() will be in this file
Use getopt() to have -a - -j, -n, -p, -q, and -H
    Use bools for arguments
Should be integrated using even number partitions

If a - j using || are false then show usage screen
If P or Q are unchanged from their initialization then no new P or W was given,
show usage screen
If H is true show usage screen
Check if all a-j if true then print which function was selected and numbers
provided
Do calculator in for loop starting from 2 to n partitions printing each step
```

Notes Pseudocode:

- Not initializing variables to double can and will cause issues further down the line.
- Integrate needed a patch to work with functions that divide by x.
  - Without the patch, a -nan would output.
  - With the patch that changes a value of 0 to one near 0, it works as intended.

Potential Errors:

- If you include two functions the program will hitch and not run.
  - This goes for if you include two of the same arguments.

Guillermo Alfaro
CSE13S Winter 2022

Citations:

- CODEBLOCK 1:
  - I used this source for how to make mandatory parameters with getopt().
  - The code checks if an argument was changed or not from when it was initialized, if it is the same from being initialized then it wasn't changed so you can return a statement.
  - Original author is Klas Lindbäck.
    - Edited by: Patrick.
  - Raw link https://stackoverflow.com/questions/26976495/mandatory-parameter-getopt-in-c/26977018
- CODEBLOCK 2:
  - I used this source to print a .txt file to bash. I tried using the book for this course but didn't quite get it to work so I utilized outside resources.
  - I used the code almost line for line and do NOT claim it as mine in any way.
  - The code opens a file in read-only mode, and if actually opened correctly it enters a while loop that prints each character as it appears, once reaching the end of file (EOF) the loop stops, the file is closed and it's done.
  - Original author is Alok Singhal.
  - Raw link https://stackoverflow.com/questions/3463426/in-c-how-should-i-read-a-text-file-and-print-all-strings
- I used Eugene Chou's section lecture on 1/14/22 to get my getopt() in integrate.c working.
  - I utilized his tips for bools, not using atoi, and For() statements for incrementing.

- I utilized all the provided pseudo-code in asgn2.pdf to get my mathlib.c math functions working.
- I visited Alec’s section on 1/18/22 at 2:00 PM to get help with integrate().
    - He helped explain Simpson’s composite ⅜ and ⅓ rule.