Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчёт по заданию №6

«Сборка многомодульных программ. Вычисление корней уравнений и неопределённых интегралов.»

Вариант 9 / 5 / 2

Выполнил:
студент 105 группы
Минхузов Д. В.

Преподаватели:
Гуляев А. В.
Русол А. В.

Москва
2024

Содержание

Постановка задачи

Реализовать численный метод, позволяющий вычислить площадь плоской фигуры, ограниченной тремя кривыми, заданных функциями $f_1(x) = \frac{3}{(x-1)^2+1}$, $f_2(x) = \sqrt{x+0.5}$, $f_3(x) = e^{-x}$. Для решения поставленной задачи также реализуются и используются метод трапеций вычисления определенного интеграла; методы хорд и отрезков вычисления корней уравнения на отрезке для нахождения вершин фигуры. При этом необходимо аналитически определить отрезки, на которых производится поиск корней.

Также программа должна иметь модуль на языке Ассемблера, реализующий функции f_1, f_2, f_3 и вычисляющий их в вещественных значениях, используя сопроцессор x87. Функции должны быть реализованы по соглашению `cdecl`.

Математическое обоснование

Для анализа предложенных функций построен график:

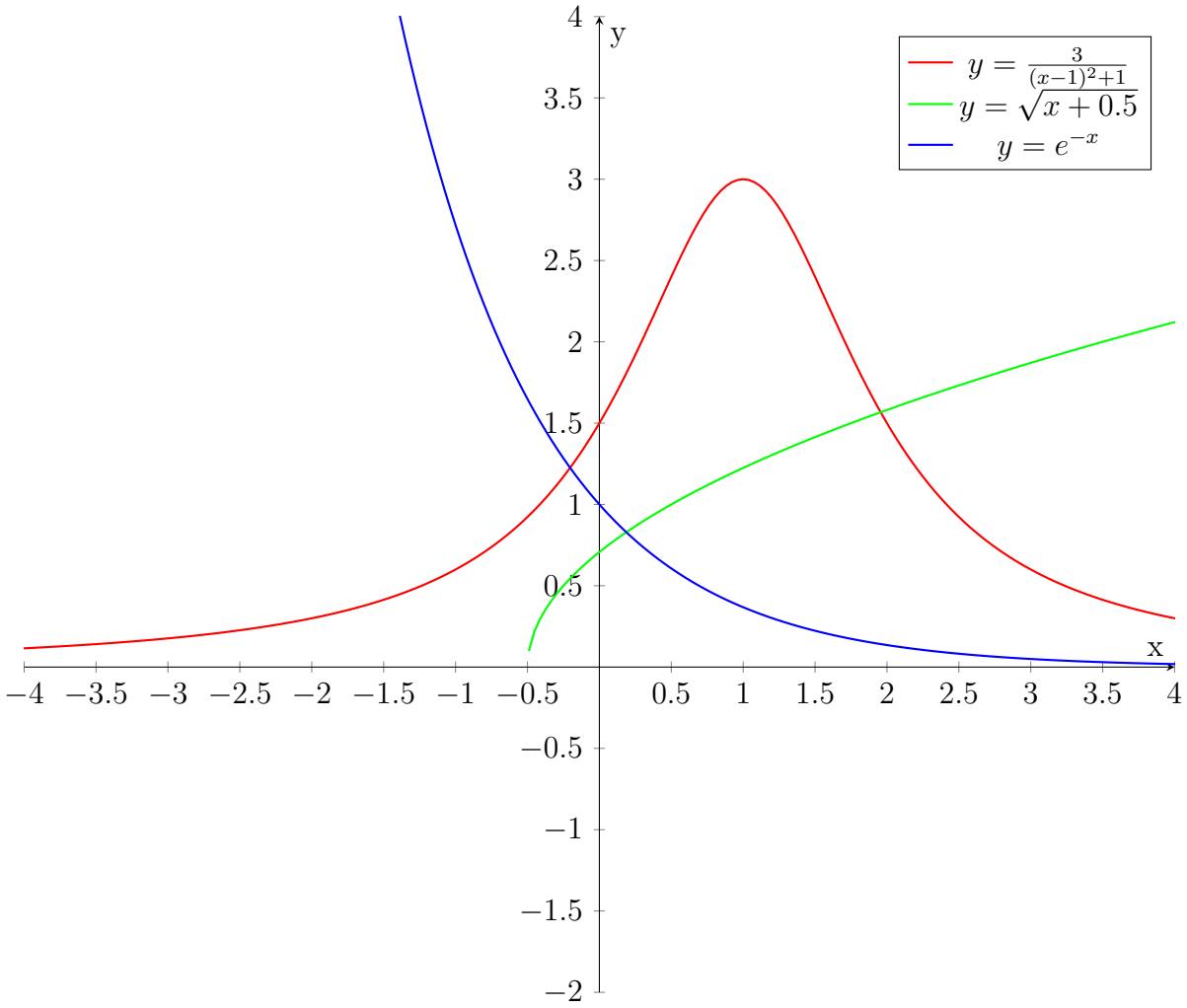

Рис. 1: Плоская фигура, ограниченная графикам заданных функций

Рассмотрим необходимые свойства функции на данном отрезке, чтобы была возможность воспользоваться нашими методами:

1. $f(a)f(b) < 0$;
2. f непрерывно дифференцируема на $[a, b]$;
3. f' монотонна и сохраняет знак на $[a, b]$.

Исходя из этих требований разумно выбрать следующие отрезки для поиска корней:

- $f(x) = f_1(x) - f_2(x) : [1, 2]$ (т.к. $f'(x) = \frac{6(x-1)}{((x-1)^2+1)^2} - \frac{1}{2\sqrt{x+0.5}}$ убывает и < 0);

- $f(x) = f_1(x) - f_3(x) : [-0.262, 0.323]$ (т.к. $f'(x) = e^{-x} - \frac{6(x-1)}{((x-1)^2+1)^2}$ возрастает и > 0);
- $f(x) = f_2(x) - f_3(x) : [0, 1]$ (т.к. $f'(x) = \frac{1}{2\sqrt{x+0.5}} + e^{-x}$ убывает и > 0).

Используя следующую формулу (для n отрезков разбиения на отрезке $[a, b]$), вычислим определенный интеграл методом трапеций: $\int\limits_a^b f(x)\mathrm{d}x = \frac{b-a}{2n}$ $\left(f(a) + f(b) + 2\sum\limits_{k=1}^{n-1} f(x_k)\right) + R$, где $|R| = O(\frac{1}{n^2}) \approx \varepsilon_2$. Посмотрим на правило Рунге для метода трапеций: $\varepsilon_2 \approx \frac{1}{3}(I_{2n} - I_n)$, где I_n - приближённое значение (без остатка) интеграла на отрезке $[a, b]$ с n отрезками разбиения. Для определения площади, нам надо посичтать 3 определенных интеграла (для каждой пары функций), поэтому точность надо увеличить в 3 раза. Тогда заметим, что значения $\varepsilon_1 = \varepsilon_2 = 0.0001$ при $\varepsilon = 0.001$ (по условию задачи) достаточно.

Результаты экспериментов

При помощи программы с точностью $\varepsilon_1 = 0.0001$ найдены координаты вершин фигуры, ограниченной графиками заданных функций (таблица 1).

Кривые	x	y
1 и 2	1.95615	1.56721
1 и 3	-0.20333	0.816009
2 и 3	0.18741	1.206122

Таблица 1: Координаты точек пересечения

Ниже приведен графики функций с закрашенной фигурой.

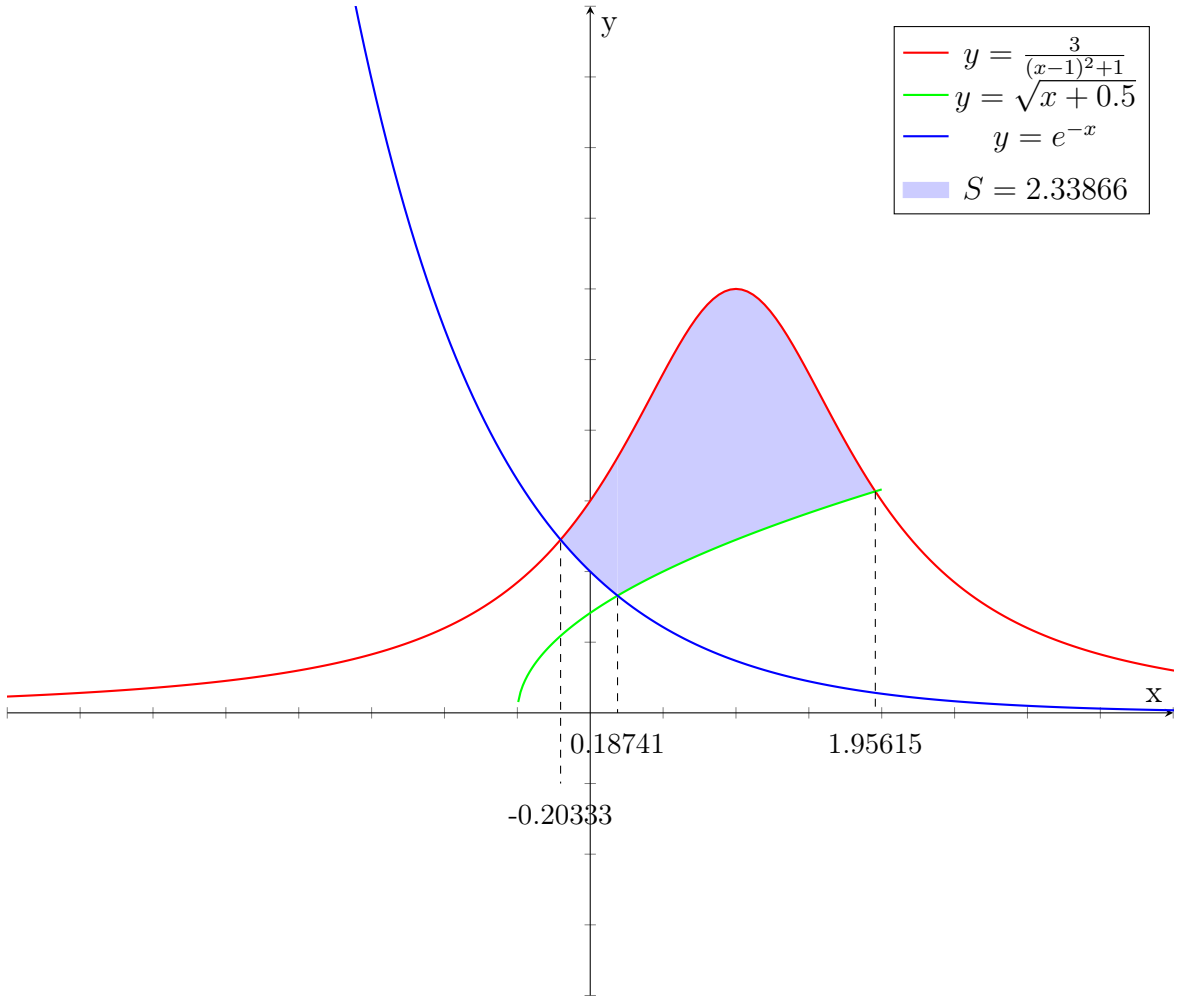

Рис. 2: Плоская фигура, ограниченная графиками заданных функций, ее площадь

Структура программы и спецификация функций

При работе программы используются следующие глобальные константы: ITERS_MAX, INTEGRAL_ITERS_MAX, MAX_INTEGRAL. ITERS_MAX отвечает за максимальное число итераций в цикле поиска корня, т.к. при неоптимальных значениях отрезка поиск может идти слишком долго. INTEGRAL_ITERS_MAX имеет такой же смысл, но используется для остановки при вычислении интеграла. Я сделал ее больше, из-за чего программа будет выполняться дольше при некоторых случаях, но в больших случаях с большей точностью сможет вычислить интеграл. Константа MAX_INTEGRAL отвечает за максимальное значение интеграла, нужна во избрежание переполнений. Также используется глобальный счетчик itersCnt, который нужен для сохранения числа итераций во время вычисления корня.

Функции f1, f2, f3 для вычисления математических выражений написаны на языке NASM под 32-битную архитектуру и импортируются в Си коде из объектного файла funcs.o.

Функция main парсит аргументы командной строки, указывает на ошибки в использовании, выводит справку. В зависимости от аргумента -mode/-M есть 3 варианта работы программы:

1. вызов функции area, которая в свою очередь вызывает функции root и integral для подсчета площади, а также подсчета числа абсцисс точек пересечения, числа операций и их вывода при необходимости;
2. вызов функции root с пользовательскими параметрами и вывод значения корня или сообщения об ошибке;
3. вызов функции integral с пользовательскими параметрами и вывод значения определённого интеграла.

Полный список используемых функций:

- double root(double (*f)(double), double (*g)(double), double a, double b, double eps1) вычисляет корень уравнения $f(x) = g(x)$ на заданном отрезке $[a, b]$ с точностью eps1, выбор между методом хорд или отрезков осуществляется на этапе компиляции, выводит ошибки, если таковые появляются на этапе вычисления;
- double integral(double (*f)(double), double a, double b, double eps2) вычисляет $\int_a^b f(x)\mathrm{d}x$ с точностью eps2 методом трапеций, выводит ошибку если такие появляются на этапе вычисления;
- double area(int flag_inter, flag_iter, double eps) вычисляет площадь фигуры, ограниченной кривыми, заданными функциями f_1, f_2, f_3, с точностью eps. Флаги absMode и iterMode показывают, нужно ли выводить абсциссы вершин фигуры и кол-во итераций, потребовавшихся для их нахождения, соответственно;
- void printHello() выводит базовую справку и предлагает посмотреть полноценную, выполняется при запуске программы без аргументов;
- void printHelp() выводит полную справку по программе.

Сборка программы (Make-файл)

Makefile содержит следующие основные цели:

1. all - сборка программы;
2. clean - удаление всех объектных и исполняемого файлов;
3. arc - создания zip-архива со всеми файлами проекта.

В задании требуется реализовать методы хорд и отрезков, выбор должен осуществляться на этапе компиляции. Это происходит при помощи флага -D, при компиляции пользователь имеет возможность выбора переменной method. Значение SECANT означает, что будет использоваться метод хорд, значение SEGMENT означает метод отрезков, также он является базовым вариантом при компиляции (если введено какое-либо другое значени)

```
1  .PHONY: all clear arc
2
3  all: main.o funcs.o mainProg
4
5  funcs.o:
6          nasm -f elf32 funcs.asm -o funcs.o
7
8  #                                          method:
9  # SEGMENT -
10 # SECANT -
11 main.o:
12         gcc -m32 -c main.c -o main.o -D $(method)
13
14 mainProg:
15         gcc -m32 -lm main.o funcs.o -o main
16
17 clean:
18         rm *.o main
19
20 arc:
21         mkdir -p ARC
22         zip ‘date +%Y.%m.%d_%N‘.zip main.c Makefile funcs
              .asm
23         mv *.zip ARC/
```

Листинг 1: Текст Makefile

Отладка программы, тестирование функций

Тестирование и отладка основного модуля на языке C происходило путем вывода в консоль интересующих значений в определенные моменты работы программы, в том числе повторный вызов функций. В ходе тестирования, полученные программой значения сравнивались с ответами Wolfram Mathematica; использовался сайт desmos.com для построения и первичного анализа кривых.

Программа на Си и на Ассемблере

Исходные тексты программы, модуля на языке Ассемблер и Makefile'а имеются в архиве, который приложен к этому отчету.

Анализ допущенных ошибок

Первые допущенные ошибки были связаны с вычислением функции e^{-x}, а также последующие ошибки в тестировании, которые приводили к заблуждениям о корректности работы программы. .

Также, реализация метода хорд изначально работала некорректно, могла выходить за рамки отрезка, по-разному работала при различных характерах выпуклости. Было допущено несколько ошибок, связанных с проверкой корректности данных, но они отлавливались довольно быстро.

Список литературы

[1] Ильин В. А., Садовничий В. А., Сендов Бл. Х. Математический анализ. Т. 1 -- Москва: Наука, 1985.