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

Отчет по заданию №6

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

Вариант 9 / 3 / 3

Выполнил:
студент 102 группы
Туниянц Б. А.

Преподаватель:
Кулагин А. В.

Москва
2020

Содержание

Постановка задачи 3

Математическое обоснование 4

Результаты экспериментов 6

Структура программы и спецификация функций 7

Сборка программы (Make-файл) 9

Отладка программы, тестирование функций 10

Программа на Си и на Ассемблере 11

Анализ допущенных ошибок 12

Список цитируемой литературы 13

Постановка задачи

Требуется с точностью ε = 0.001 найти площадь плоской фигуры, ограниченной тремя заданными кривыми:

1) $f_1 = \frac{3}{(x-1)^2+1}$

2) $f_2 = \sqrt{x+0.5}$

3) $f_3 = e^{-x}$

В начале требуется найти точки пересечения кривых методом Ньютона, для которого предварительно определить отрезки, на которых будут пересечения. Далее требуется реализовать функцию интеграла, который будет рассчитываться через Формулу Симпсона. С помощью этой функции требуется найти искомую площадь. Обе функции должны быть предварительно протестированы.

Математическое обоснование

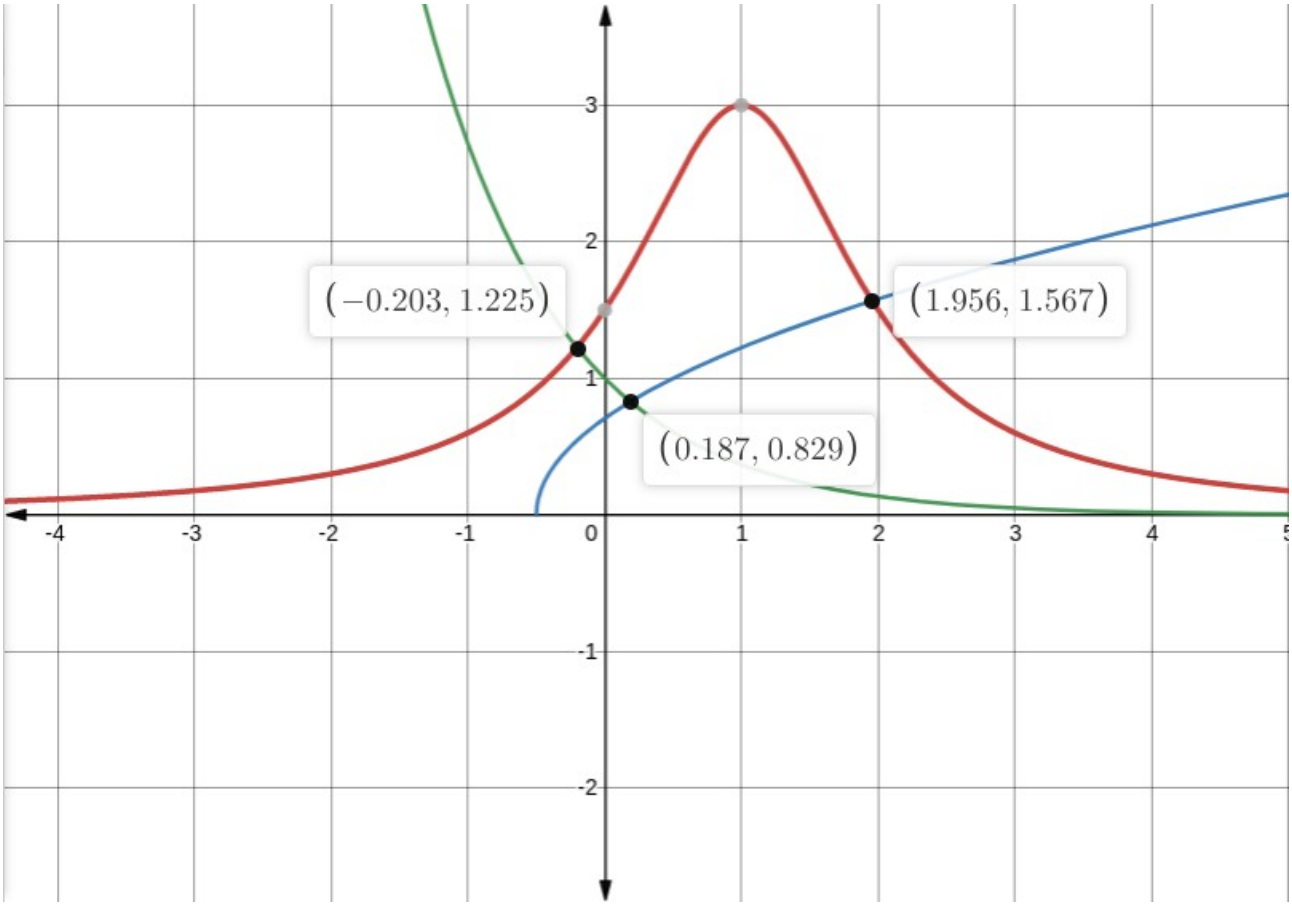

Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

1. Выбор отрезков для поиска точек пересечений

   Нахождение точек пересечений проводилось методом Ньютона с заданной погрешностью $\varepsilon_1 = 0.0001$. $F(x) = f1(x) - f2(x)$ *на* $[a, b]$
   Берем a и b такие, чтобы выполнялось:

   1) f1(x), f2(x) определены и непрерывны на [a, b], из чего следует, что F(x) также непрерывна на [a, b]

   2) F(x) * F(b) < 0

   Заданные функции непрерывны на области определения, следовательно, непрерывна и F(x). Проверяя условия и пользуясь, для наглядности, построенным графиком, определяем отрезки, используемые для определения точек пересечения.

   Такими отрезками будут:
   [1.0, 2.0] для точки пересечения f_1 *и* f_2
   [-1.0, 1.0] для точки пересечения f_2 *и* f_3
   [-1.0, 1.0] для точки пересечения f_1 *и* f_3

2. Интегрирование и выбор ε_1, ε_2

   Интегрирование проводилось при помощи формулы Симпсона:
   $$\int_a^b f(x)dx \approx \frac{b-a}{6} * (f(a) + 4 * f(\frac{a+b}{2}) + f(b))$$
   Для того, чтобы снизить погрешность вычислений, отрезок можно разбить и применить формулу Симпсона к каждому отрезку разбиения, а нужный интеграл будет равен сумме интегралов каждого из отрезков по свойству интегралов.

   Погрешность вычислений при помощи формулы Симпсона составляет
   $$f^{(4)}(x) * \frac{(b-a)^5}{2880} .$$
   Итого, для рассчета количества отрезков, на которое надо разбить исходный отрезок, надо решить неравенство:
   $f^{(4)}(x) * \frac{(b-a)^5}{2880} + 0.0001 < 0.001$, $f^{(4)}(x)$ берем максимально возможную на [a, b]. После решения данного неравенства, получается, что точно хватит 25 отрезков разбиения вычислений с заданной точностью.

Результаты экспериментов

Кривые	x	y
1 и 2	1.956153,	1.567212
2 и 3	0.187411	0.829103
1 и 3	-0.203335	1.225483

Таблица 1: Точки пересечения кривых

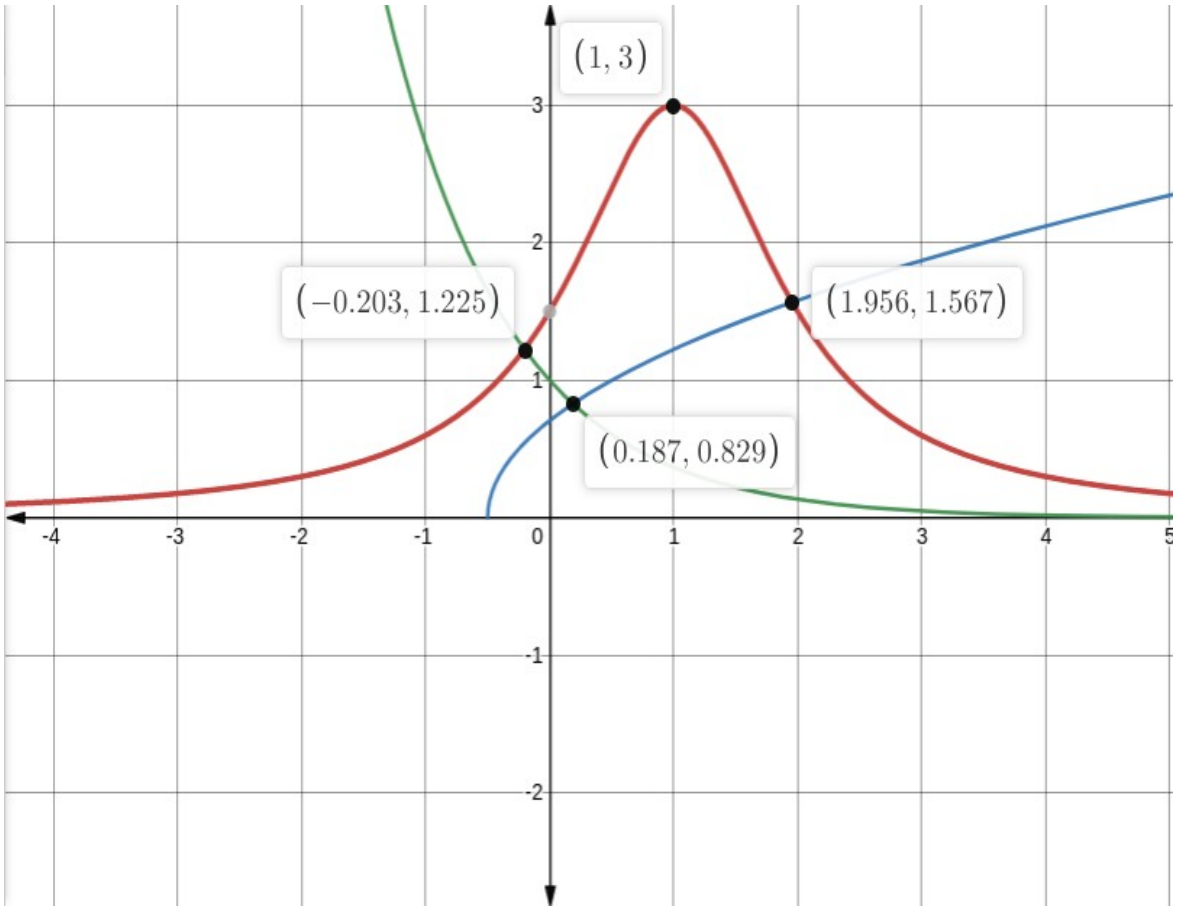

Рис 2: Жирными отмечены точки, ограничивающие искомую площадь

Структура программы и спецификация функций

Программа состоит из 1 модуля на языке C, 1 модуля на Ассемблере.

1. Модуль main.c.

   1) int main(int argc, char const *argv[]): Функция начала программы; принимает на вход аргументы int argc, char const *argv[] из командной строки и обрабатывает их на совпадение с описанными флагами. Задана глобальная переменная iterations, которая выступает в роли счетчика количества итераций в Методе Ньютона.

   2) void test_root(function fx1, function fx2, function fx3, function dfx1, function dfx2, function dfx3): Функция тестирования Метода Ньютона; принимает на вход три функции и соответствующие производные к ним.

   3) void test_intg(function f1, function f2, function f3, double eps): Функция тестирование Формулы Симпсона; принимает на вход 3 функции и требуемую точность вычислений.

   4) double integral(function f, double a, double b, double eps2): Функция подсчета интеграла при помощи формулы Симпсона; принимает на вход функцию, требуемую точность, левую и правую границу отрезка, на котором вычисляется интеграл.

   5) double root(function fx1, function fx2, double a, double b, function dfx1, function dfx2, double eps1): Функция вычисления абсциссы пересечения кривых; принимает на вход 2 функции, левую и правую границу отрезка, на котором существует точка пересечения, а так же соответствующие производные для функций и требуемую точность вычислений.

2. Модуль f_asm.asm.

   1) f1: функция принимает на вход double x, координату, и вычисляет в ней значение функции f_1, описанной ранее в отчете.

   2) f2: функция принимает на вход double x, координату, и вычисляет в ней значение функции f_2, описанной ранее в отчете.

   3) f2: функция принимает на вход double x, координату, и вычисляет в ней

значение функции f_3 , описанной ранее в отчете.

4) df1: функция принимает на вход double x, координату, и вычисляет в ней значение производной функции f_1 , описанной ранее в отчете.

5) df2: функция принимает на вход double x, координату, и вычисляет в ней значение производной функции , f_2 описанной ранее в отчете.

6) df3: функция принимает на вход double x, координату, и вычисляет в ней значение производной функции , f_3 описанной ранее в отчете.

Сборка программы (Make-файл)

Makefile собирает модули в файл programm. Сборка осуществляется по ключу all, а удаление промежуточных файлов — по ключу clean.

```
all:
	nasm -f elf -o f_asm.o f_asm.asm
	gcc -m32 -c main.c -o main.o
	gcc -m32 main.o f_asm.o -o meduzen -lm
clean:
	rm *.o
```

Отладка программы, тестирование функций

Тестирование проводилось при помощи ключей –test-integral и –test-root.

Функция root тестируется непосредственно пользователем: ему предоставляется возможность указать пару из 2 функций, границы отрезка, на котором искать пересечение и точность вычислений. На входе из функции пользователь получит либо точку пересечения, либо сообщение, что точка не найдена на заданном интервале.

Функция integral тестировалась при следующих данных:

Уравнение	Левая граница	Правая граница	Результат
f1	0.000000	10.000000	6.736611
f1	-5.000000	1.000000	4.216943
f1	1.000000	17.000000	4.524961
f2	0.000000	1.000000	0.989043
f2	0.000000	3.000000	4.129565
f2	1.000000	17.000000	47.580416
f3	-2.000000	2.000000	7.253722
f3	-1.000000	7.000000	2.717380
f3	0.000000	1.000000	0.632121

Таблица 2: Результаты работы программы integral при тестировании

Проверка правильности вычисленных интегралов проводилась при помощи сервиса wolframalpha.com.
Проверка правильности вычисленных точек пересечения для заданных функций проводилась при помощи desmos.com.

Программа на Си и на Ассемблере

Тексты программ находятся в архиве meduzen.zip, приложенном к отчету.

Анализ допущенных ошибок

Ошибок не допущено.

Список литературы

[1] AAK, 04/05/2021

[2] Real Analysis: Bruce Blackadar, Department of Mathematics and Statistics, University of Nevada, Reno, 202

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