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 1/ 1/ 8

Выполнил:
студент 104 группы
Воробьев С. Ю.

Преподаватель:
Сенюкова О. В.

Москва
2020

Содержание

Постановка задачи

С заданной точностью ε вычислить площадь плоской фигуры, ограниченной тремя кривыми. Для этого требуется выполнить следующие подзадачи:

- С некоторой точностью ε_1 вычислить абсциссы точек пересечения кривых, используя метод деления отрезка пополам для уравнения $F(x) = 0$.
- Представить площадь заданной фигуры как алгебраическую сумму определенных интегралов и вычислить эти интегралы с некоторой точностью ε_2 по формуле прямоугольников.
- Величины ε_1 и ε_2 подобрать вручную так, чтобы гарантировалось вычисление площади фигуры с точностью ε

Математическое обоснование

Корни:
$(f_1 - f_3)(-3) > 0, (f_1 - f_3)(-2) < 0, (f_1 - f_3)(x) \downarrow \Rightarrow x_1 \in [-3, -2]$
$(f_2 - f_3)(-3) > 0, (f_2 - f_3)(-2) < 0, (f_2 - f_3)(x) \downarrow \Rightarrow x_2 \in [-1, 0]$
$(f_1 - f_2)(1) < 0, (f_1 - f_2)(2) > 0, (f_1 - f_2)(x) \uparrow \Rightarrow x_3 \in [1, 2]$

Для оценки **погрешности** интеграла будем пользоваться формулой Рунге[2]:

$$\Delta_{2n} = \frac{1}{3}(I_{2n} - I_n)$$

Погрешность одного инетграла можно вычислить по формуле:

$$\Delta_{I_i} = \varepsilon_2 + 2f_{imax}\varepsilon_1$$

Суммарная погрешность 3-х интегралов будет:

$$\Delta_I^{\Sigma} = 3\varepsilon_2 + 6f_{max}\varepsilon_1 < \varepsilon$$

Пусть $\varepsilon_2 = k\varepsilon$, тогда:

$$\varepsilon_1 \leq \frac{(1-3k)\varepsilon}{6f_{max}}$$

Заметим, что $k < 1/3$ – пусть $k = 1/5$, а $f_{max} < 10$, тогда можем подставить все это в выражение:

$$\varepsilon_1 \leq \frac{(1-3k)\varepsilon}{6f_{max}} = \frac{\varepsilon}{150}$$

$$\varepsilon_2 \leq \frac{\varepsilon}{5}$$

Результаты экспериментов

Таким образом у меня получились координаты точек пересечения(таблица 1):

Кривые	x	y
1 и 3	-2.3905	2.0916
2 и 3	-0.5495	9.0990
1 и 2	1.2518	5.4965

Таблица 1: Координаты точек пересечения

Проиллюстрируем результаты графиком (рис. 1).

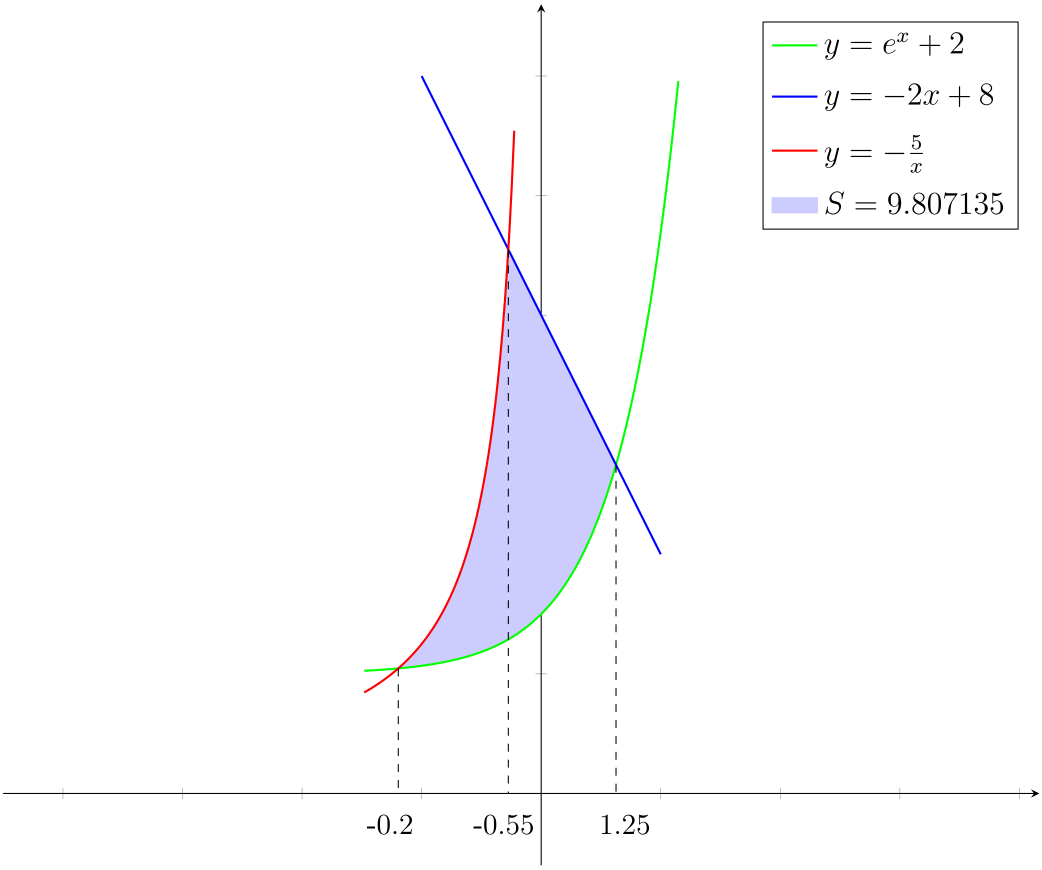

Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Программа построена на двух основных функциях – root и integral

$root(f, g, a, b, eps)$
Возвращаемое значение – корень f - g = 0 (double)
f и g – указатели на функции (double (*)())
a и b – границы интервала (double)
eps – допустимая погрешность вычисления корня (double)

$integral(f, a, b, eps)$
Возвращаемое значение – значение определённого интеграла(double)
f – указатель на подынтегральную функцию (double (*)())
a и b – пределы интегрирования (double)
eps – допустимая погрешность вычисления интеграла(double)

Математические функции реализованы в виде методов
$f1(x), f2(x), f3(x), f4(x), f5(x), f6(x)$
Функции 1-3 реализованы на NASM
Функции 4-6 реализованы на Си и нужны для тестирования работы функций с численными методами
На вход все эти функции принимают x (double) – абсциссу, а возвращают $f_i(x)$ (double) – ординату.

Программа состоит из 4-х файлов:
main.c – главный файл, обрабатывает флаги, вызывает функции посчета
calc.c – файл с функциями, выполняющими подсчеты – integral, root, f_4, f_5, f_6
HEAD.h – файл с заголовочными файлами и прототипами
func.asm – файл с функциями f_1, f_2, f_3 на языке NASM

Сборка программы (Make-файл)

```
all: main clean
main: main.o func.o
gcc -m32 -o main main.o func.o -lm

main.o: main.c
gcc -m32 -std=c99 -c main.c

func.o: func.asm
nasm -felf32 func.asm

clean:
rm *.o
```

Как видно из Make-файла – программа собирается из двух основных частей: main.c и func.asm.
В файле main.c подключен заголовочный файл HEAD.h, который прикрепляет к main.c все необходимое в процессе препроцессирования. Таким образом исходный файл линкуется из двух объектных файлов – main.o и func.o

Отладка программы, тестирование функций

Протестируем наши функции с численными методами на другом наборе функций (таблица 2) и (таблица 3). Во второй колонке значение, посчитанное моей программой, в третьей колонке – посчитанное через Wolfram Alpha.

Входные данные	x	Walpha
$ln(x) + 2$ $\sqrt{x}$ (0, 1]	0.215224	0.2152235
$ln(x) + 2$ $-x + 5$ [2, 3]	2.207940	2.2079400
$-x + 5$ $\sqrt{x}$ [3, 4]	3.208713	3.2087121

Таблица 2: Координаты точек пересечения

Подынтегральная функция	I	Walpha
$ln(x) + 2$ [1, 3]	5.2958	5.29583
$-x + 5$ [-2, 0]	12.0000	12.00000
$\sqrt{x}$ [5, 5.5]	1.1455	1.14554

Таблица 3: Интегралы от других функций

Программа на Си и на Ассемблере

Исходные тексты программы имеются в архиве, который приложен к этому отчету.

Анализ допущенных ошибок

При подключении библиотеки math.h возникли проблемы – она не подключалась. Исправить удалось с помощью использования флага -lm при линовке.

При нахождении корней методом деления отрезка пополам изначально я не учел, что возможны два случая: убывание и возрастание функции. Я рассматривал только один, и это привело к ошибкам в вычислении корней. Исправилось это проверкой и корректировкой входных данных перед непосредственной работой алгоритма.

Список литературы

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

[2] Осокин А. Е. [http://e-lib.gasu.ru/eposobia/metody/R_4_7.html 4.7 Правило Рунге оценки погрешности. Экстраполяция Ричардсона.]