МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени М. В. ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ №6

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

**Вариант 11 / 2 / 2**

Выполнила:

студентка 103 группы

Цыс А. И.

Преподаватель:

Батузов К. А.

Москва

2019

**Содержание**

Постановка задачи - 2

Математическое обоснование - 3

Результаты экспериментов - 5

Структура программы и спецификация функций - 6

Сборка программы (Make-файл) - 7

Программа на Си и на Ассемблере - 8

Список цитируемой литературы - 9

**Постановка задачи**

Требуется реализовать численные методы, позволяющие вычислять площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых y = f1(x), y = f2(x) и y = f3(x) задаются в текстовом виде на этапе сборки программы.

Реализуется метод хорд решения уравнений. Для подсчёта интегралов используется формула трапеций.

Необходимо разработать две программы: основную — для вычисления площади и вспомогательную — для построения исполняемого кода, вычисляющего значения функций.

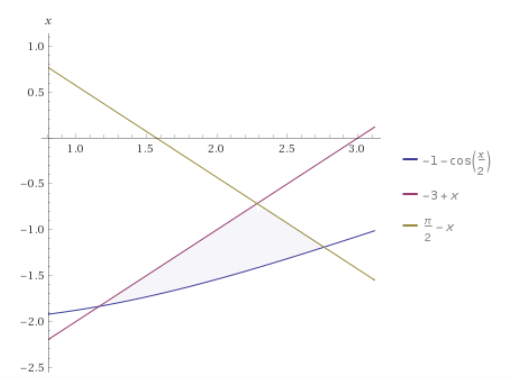
**Математическое обоснование**

Проведён анализ кривых методами матанализа. Использованы вспомогательные средства для построения графиков на компьютере.

В качестве eps1 взято число 0.0001, eps2 – 0.000005

Они подходят, так как считаются три интеграла и три точки пересечения: 3\*e2 + 3\*2\*10\*e1=3\*(e2+20\*e1)=0.0006 <e=0.001, где 10 — значение, меньше которого значения функций на рассматриваемом отрезке.

Графики кривых:



Достаточное условие для существования ровно одного корня на отрезке таково: на концах отрезка функция F(x) = f1(x) – f2(x) имеет разные знаки и на всем отрезке первая и вторая производная функции не меняет знак.

-1 - cos(x/2) – x + 3 = - cos(x/2) – x + 2 > 0 => x > 1.5

- cos(x/2) – x + 2 < 0 => x < 1

(- cos(x/2) – x + 2)’ = 1/2sin(x/2) – 1 < 0 при любом x

(1/2sin(x/2) – 1)’ = 1/4cos(x/2) > 0 => 0 < x < π

* 0 < x < 3

-1 - cos(x/2) + x – π/2 > 0 => x > 3

-1 - cos(x/2) + x – π/2 < 0 => x < 2.5

(-1 - cos(x/2) + x – π/2)’ = 1/2sin(x/2) + 1 > 0 => 1 < x < 2π

(1/2sin(x/2) + 1)’ = 1/4cos(x/2) > 0 => 0 < x < π

* 0 < x < 3

-3 + x – π/2 + x = 2x - 3 – π/2 > 0 => x > 2.5

2x - 3 – π/2 < 0 => x < 2

(2x - 3 – π/2)’ = 2 > 0 при любом x

* 0 < x < 3

Значит, отрезков поиска точек пересечения - [0, 3].

**Результаты экспериментов**

В результате проведенных экспериментов были получены следующие результаты:

|  |  |  |
| --- | --- | --- |
| Кривые | x | y |
| 1 и 2  1 и 3  2 и 3 | 1.1649  2.7603  2.2853 | -1.8351  -1.1895  -0.7147 |

S = 0.578

**Структура программы и спецификация функций**

Программа строится из 3 модулей: ассемблерного functions и main и generation на Си.

Ассемблерный модуль состоит из 3 функций, предназначенных для вычисления значений заданных функций (f1, f2, f3). На вход каждая из функций принимает значение типа double — абсцисса точки, в которой необходимо высчитать значение функции — и возвращает найденное значение в виде числа типа double.

Модуль main на Си содержит функцию main(int argc, char \*argv[]), принимающую на вход параметры из командной строки, определяющие поведение функции и вывод; функцию

double root(double (\*f)(), double (\*g)(), double (\*df)(), double (\*dg)(), double a, double b, double eps1), получающую на вход две функции, точку пересечения которых необходимо найти, их производные, границы отрезка для поиска и значение точности подсчёта; функцию

double integral(double (\*f)(), double a, double b, double eps2), получающую на вход функцию, значение интеграла от которой необходимо найти, границы отрезка интегрирования и значение точности подсчёта; а также отдельные функции, предоставляющие возможности для тестирования функций нахождения корня и интеграла путём ввода с клавиатуры параметров для этих функций вручную.

Модуль generation на Си содержит функцию main(int argc, char \*argv[]), принимающую на вход параметры из командной строки, определяющие поведение функции. Функция выполняет генерацию ассемблерного кода в модуле functions.asm.

**Сборка программы (Make-файл)**

all: functions.asm prog

clean:

    rm -rf prog \*.o

generation.o: generation.c

    gcc generation.c -c -o generation.o -m32

gen: generation.o

    gcc generation.o -o gen -m32

functions.asm: $(SPEC\_FILE) gen

    ./gen $(SPEC\_FILE)

functions.o: functions.asm

    nasm -f elf32 functions.asm -o functions.o

main.o: main.c

    gcc main.c -c -o main.o -m32

prog: main.o functions.o

    gcc functions.o main.o -o prog -m32

**Программа на Си и на Ассемблере**

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

**Список литературы**

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