

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 8 / 1 / 1**

Выполнил:  
студент 105 группы  
Домнин С. В.

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

Москва  
2023

# Содержание

Постановка задачи	2
Математическое обоснование	3
Результаты экспериментов	5
Структура программы и спецификация функций	6
Сборка программы (Make-файл)	8
Отладка программы, тестирование функций	10
Программа на Си и на Ассемблере	11
Анализ допущенных ошибок	12
Список цитируемой литературы	13

## Постановка задачи

В ходе работы было необходимо реализовать численный метод поиска корней уравнения и вычисление площади плоской фигуры, ограниченной тремя кривыми. Требования к заданию:

- Точки пересечения графиков, образующие вершины фигуры, площадь которой нам необходимо вычислить находятся с помощью метода деления отрезка пополам.
- Вычисление площади происходит с помощью метода прямоугольников.
- Отрезок для применения методов наложения корней, вычисляется аналитически.

Задача решается для следующих функций:

1.  $f_1 = e^x + 2$

2.  $f_2 = -2x + 8$

3.  $f_3 = -\frac{5}{x}$

## Математическое обоснование

Рассмотрим ограничения на функции для сходимости метода деления отрезка пополам [1]:  $f(a) * f(b) < 0$ ,  $f(x)$  непрерывно дифференцируема на  $[a, b]$ . И исходя из этих ограничений найдем отрезки, на которых будет осуществляться поиск вершин фигуры.

- $f_{12} = e^x + 2 + 2x - 8$

Функция непрерывно дифференцируема как разность непрерывно дифференцируемых функций  $f_1$  и  $f_2$ .

$$f'_{12} = e^x + 2;$$

$$f_{12}(1) = -1.28;$$

$$f_{12}(1.5) = 3.482;$$

$$f_{12}(1) * f_{12}(2) = -4.457 < 0;$$

Исследуем  $f_{12}$  на  $[1, 1.5]$ .

- $f_{23} = -2x + 8 + \frac{5}{x}$

Функция непрерывно дифференцируема как разность непрерывно дифференцируемых функций  $f_2$  и  $f_3$ .

$$f'_{23} = -2 + \frac{5}{x^2};$$

$$f_{23}(-0.8) = 3.35;$$

$$f_{23}(-0.3) = -7.07;$$

$$f_{23}(-0.8) * f_{23}(-0.3) = -23.68 < 0;$$

Исследуем  $f_{23}$  на  $[-0.8, -0.3]$ .

- $f_{13} = e^x + 2 + \frac{5}{x}$

Функция непрерывно дифференцируема как разность непрерывно дифференцируемых функций  $f_1$  и  $f_3$ .

$$f'_{13} = e^x - \frac{5}{x^2};$$

$$f_{13}(-2.5) = 0.082;$$

$$f_{13}(-2) = -0.36;$$

$$f_{13}(-2.5) * f_{13}(-2) = -0.0295 < 0;$$

Исследуем  $f_{13}$  на  $[-2.5, -2]$ .

Для вычисления интеграла с использованием итерационного метода деления отрезка пополам, необходимо разбить заданный отрезок на равные части. Затем мы вычисляем значение функции в середине каждой части и продолжаем разбивать ту часть отрезка, значения функции на концах которой имеют разные знаки. Деление отрезка продолжается до достижения необходимой точности решения  $\varepsilon_2$ .

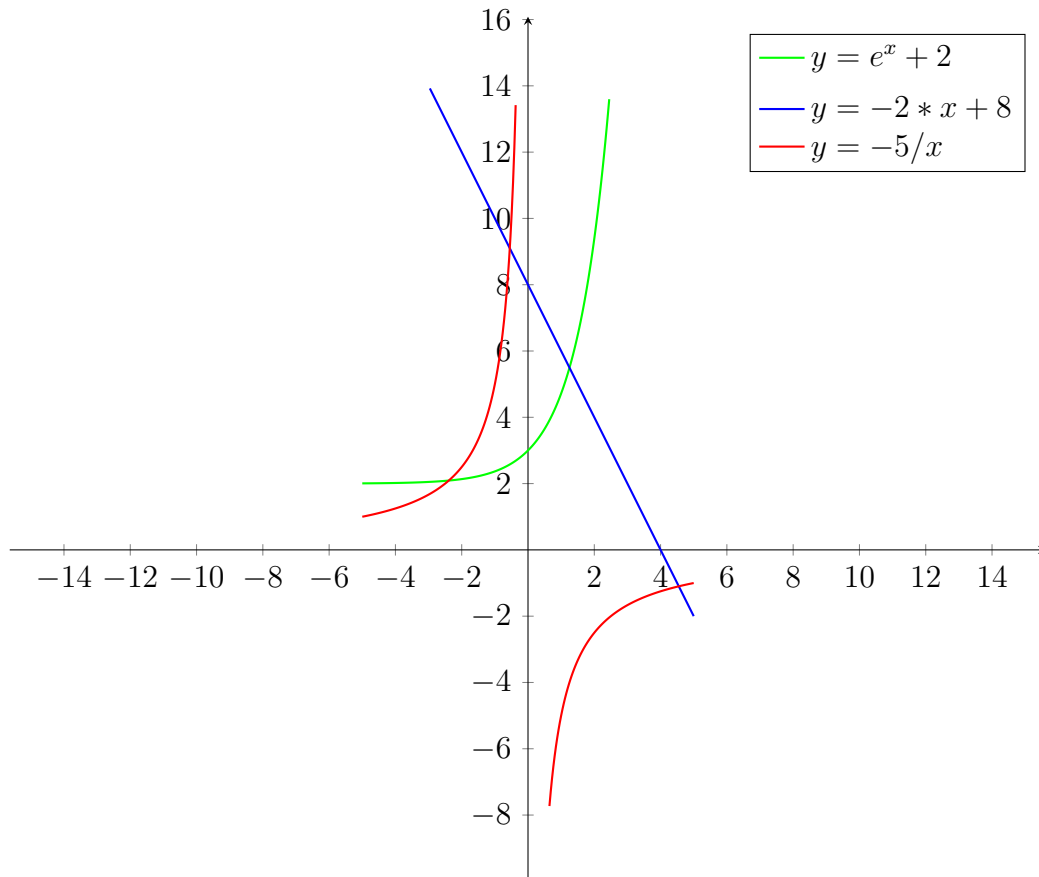


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Вычислим  $\varepsilon_1$  и  $\varepsilon_2$  с помощью которых нам необходимо будет искать площадь, учитывая правило Рунге. Каждая из трех точек пересечения кривых считается с точностью  $\varepsilon_1$ . Площадь фигуры, ограниченной этими кривыми, будет отличаться от правильной не более чем на  $3\varepsilon_1^2$ . Вычисление интеграла производится с погрешностью  $\varepsilon_2$ , поэтому общая точность будет  $3\varepsilon_1^2 + 3\varepsilon_2 < \varepsilon$ .

К примеру, при  $\varepsilon = 0.001$  можно взять  $\varepsilon_1 = \varepsilon_2 = 0.0001$ .

## Результаты экспериментов

В данном разделе необходимо провести результаты проведенных вычислений: координаты точек пересечения (таблица 1) и площадь полученной фигуры.

Кривые	$x$	$y$
1 и 2	1.2518	5.4965
2 и 3	-0.5495	9.0991
1 и 3	-2.3905	2.0916

Таблица 1: Координаты точек пересечения

Результаты можно представить не только в текстовом виде, но и проиллюстрировать графиком (рис. 2).

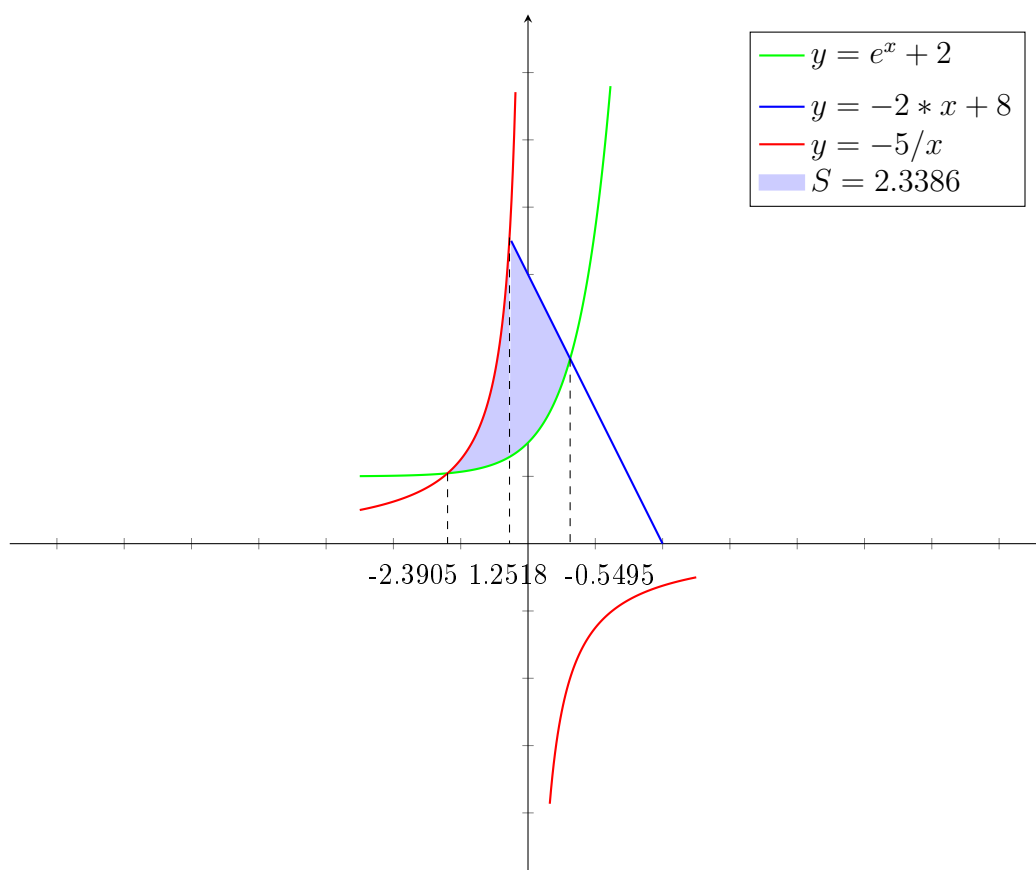
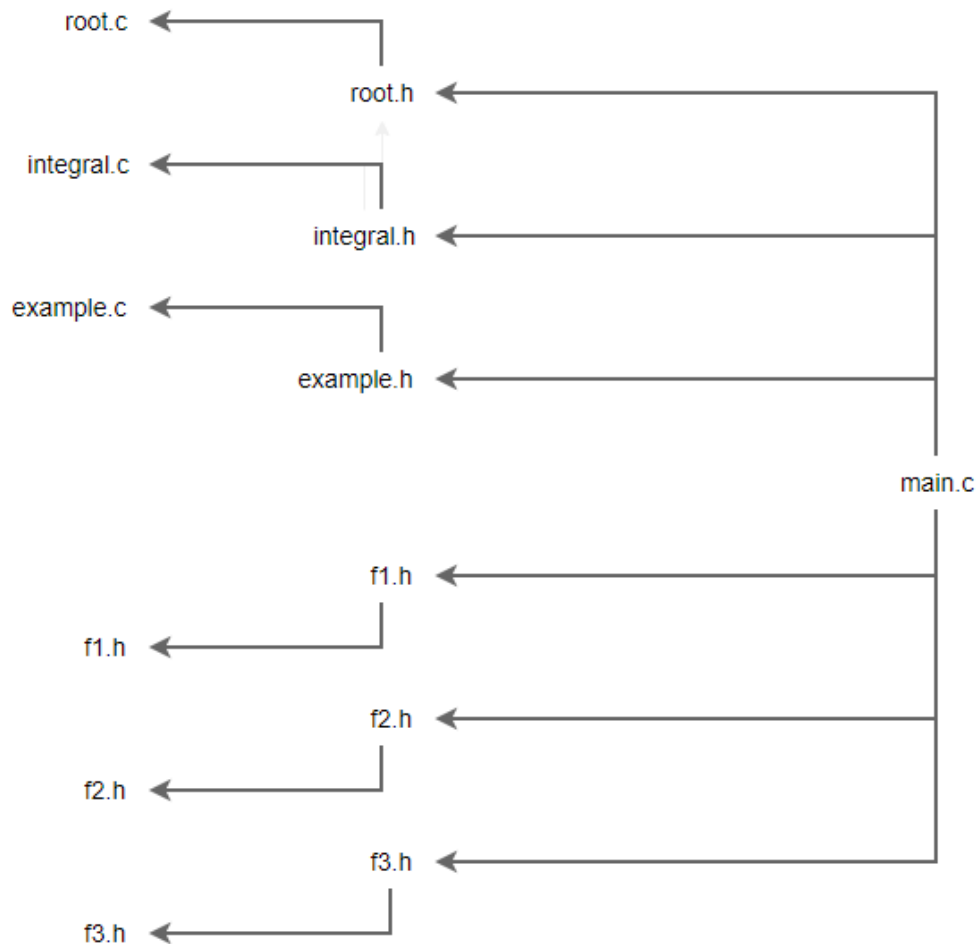


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

# Структура программы и спецификация функций



Моя программа состоит из шести модулей: `main.c` (основная часть программы, обрабатывающая вводимые параметры и управляющая работой программы), `root.c` (функция, реализующая численный метод для нахождения корня с помощью метода деления отрезка пополам), `integral.c` (функция, реализующая численный метод для вычисления интеграла с помощью метода прямоугольников), `f1.asm`, `f2.asm` и `f3.asm` (функции и в соответствии с вариантом задания)

## Модуль `main.c`

Считывает все введенные параметры командной строки и согласно заданному условию выводит необходимые параметры.

```
int main(const int argc , const char *argv [])
```

## Модуль root.c

Находит координату решения уравнения  $f = g$ ,  $f$  и  $g$  - функции, на отрезке  $[a, b]$  с точностью  $\text{eps1}$  с помощью метода деления отрезка пополам.

**double root(double (\*func\_f)(double), double (\*func\_g)(double), double a,**

## Модуль integral.c

Вычисляет значение интеграла функции  $f$  на отрезке  $[a, b]$  с точностью  $\text{eps2}$  с помощью метода прямоугольников.

**double integral(double (\*func\_f)(double), double a, double b, double eps2**

## Модуль f1.asm

Вычисляет значение функции  $f1 = e^x + 2$  в точке  $x$ ;

**На ассемблере: global f1**

**На Си: double f1(double x)**

## Модуль f2.asm

Вычисляет значение функции  $f1 = -2x + 8$  в точке  $x$ ;

**На ассемблере: global f2**

**На Си: double f2(double x)**

## Модуль f3.asm

Вычисляет значение функции  $f1 = -5/x$  в точке  $x$ ;

**На ассемблере: global f3**

**На Си: double f3(double x)**

## Модуль example.c

1. Вычисляет значение функции  $x^2$  в точке  $x$ ;

*doubleex<sub>f</sub>1(doublex).*

2. Вычисляет значение функции  $\sin(x)$  в точке  $x$ ;

*doubleex<sub>f</sub>2(doublex).*

3. Вычисляет значение функции  $\frac{\sqrt{x}}{2}$  в точке  $x$ ;

*doubleex<sub>f</sub>3(doublex).*



## Сборка программы (Make-файл)

Процесс сборки:

- Компиляция: цели main.o, func.o root.o integral.o, example\_func.o;
- Линковка: цель all;
- Удаление объектных файлов: цель clean;
- Тестирование работы функций root.c и integral.c: цель test;
- Создание zip-архива со всеми файлами: цель arc;

Текст makefile:

```
1 .PHONY: all clean arc
2
3 all: integral
4
5 integral: main.o f1.o f2.o f3.o root.o integral.o example.o
6     gcc -m32 $^ -lm -o integral.out -no-pie
7
8 main.o: main.c root.h integral.h f1.h f2.h f3.h example.h
9     gcc -m32 -c $< -lm -o main.o
10
11 f1.o: f1.asm
12     nasm -f elf32 $< -o f1.o
13
14 f2.o: f2.asm
15     nasm -f elf32 $< -o f2.o
16
17 f3.o: f3.asm
18     nasm -f elf32 $< -o f3.o
19
20 %.o: %.c
21     gcc -m32 -c $< -o $@
22
23 clean:
24     -rm *.o integral.out
25
26 test: integral
27     ./integral.out -R 4:5:0.5:1:0.0001:0.8767
28     ./integral.out -R 5:6:0.1:0.5:0.0001:0.25555
29     ./integral.out -R 4:6:0.5:1:0.0001:0.63
30     ./integral.out -I 4:2:10:0.0001:330.666666
31     ./integral.out -I 5:3.5:4:0.0001:-0.282828
32     ./integral.out -I 6:2:6:0.0001:3.9561704
33     mkdir -p ARC
34     zip 'date +%Y.%m.%d_%N'.zip integral.out example.c example
        .h f1.asm f1.h f2.asm f2.h f3.asm f3.h integral.c
        integral.h root.c root.h main.c Makefile
```

```
mv *.zip ARC
```

## Отладка программы, тестирование функций

Для тестирования и отладки численных методов были написаны дополнительные функции на языке си и реализованы ключи командной строки -R и -I. -R.

- -R позволяет протестировать функцию root. Фактические параметры вызова root задаются единственным параметром этой опции в виде F1:F2:A:B:E:R, где F1, F2 - номера используемых функций, A, B, E - значения параметров a, b, eps1 функции root, R - правильный ответ (вычисленный аналитически). Программа вызывает функцию root с указанными параметрами, сравнивает результат с правильным ответом и выводит на экран полученный результат, абсолютную и относительную ошибку.
- -I: Позволяет протестировать функцию integral. Фактические параметры вызова integral задаются единственным параметром этой опции в виде F:A:B:E:R, где F - номер используемой функции, A, B, E - значения параметров a, b, eps2 функции integral, R - правильный ответ (вычисленный аналитически). Программа вызывает функцию integral с указанными параметрами, сравнивает результат с правильным ответом и выводит на экран полученный результат, абсолютную и относительную ошибку.

Исходные функции идут под номерами 1-3 дополнительные под номерами 4-6.

Также для тестирования функции в *makefile* была реализована цель test, которая выполняет программу с ключами -R или -I в соответствии с таблицами:

Функция	Отрезок	Корень	Результат работы программы
$x^2 - \sin(x)$	[0.5, 1]	0.8767	0.8767
$x^2 - \frac{\sqrt{x}}{2}$	[0.1, 0.5]	0.25555	0.25555
$\sin(x) - \frac{\sqrt{x}}{2}$	[0.5, 1]	0.63	0.63

Таблица 2: Тестирование поиска корней

Функция	Интеграл	Ответ	Результат работы программы
$x^2$	$\int_2^{10} x^2 dx$	330.667	330.666666
$\sin(x)$	$\int_{3.5}^4 \sin(x) dx$	-0.2829	-0.282828
$\frac{\sqrt{x}}{2}$	$\int_2^{10} \frac{\sqrt{x}}{2} dx$	3.956	3.9561704

Таблица 3: Тестирование вычисления интегралов

## Программа на Си и на Ассемблере

В данном разделе необходимо написать, что исходные тексты программы имеются в архиве, который приложен к этому отчету.

Наличие приложенного архива с текстами программы является обязательным требованием сдачи отчета. Если почтовая служба по каким-то причинам блокирует электронное письмо с вашим архивом, то используйте зашифрованный архив с паролем «123». Такого пароля достаточно, чтобы почтовые сервисы не заглядывали в архив.

## Анализ допущенных ошибок

## Список литературы

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