

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 3 / 3 / 3**

Выполнила:  
студентка 119 группы  
Голубкова М. С.

Преподаватель:  
Сковорода Н. А.

Москва  
2021

# Содержание

Постановка задачи	2
Математическое обоснование	3
Результаты экспериментов	4
Структура программы и спецификация функций	5
Сборка программы (Make-файл)	7
Отладка программы, тестирование функций	8
Анализ допущенных ошибок	9

## Постановка задачи

В данном задании требовалось реализовать численный метод, позволяющий вычислить площадь плоской фигуры, ограниченной тремя кривыми, с точностью  $\varepsilon = 0.001$ . В этом варианте интегралы вычислялись методом Симпсона (методом парабол), вершины фигуры, абсциссы которых являлись границами интегрирования, вычислялись методом Ньютона (методом касательных). Отрезок, на котором применялся метод нахождения корней, был вычислен аналитически.

## Математическое обоснование

Сначала было определено необходимое значение  $\varepsilon_2$ . Так как искомый интеграл выражается через алгебраическую сумму трёх других интегралов, то их допустимые погрешности складываются при получении точности результата. Соответственно точности в 0.001 мало, интегралы будут вычисляться с точностью  $\varepsilon_2 = 0.0001$

Вычисление  $\varepsilon_1$  происходит автоматически в функции `main`. Оно реализуется через цикл `while`,  $\varepsilon_1$  подбирается так, чтобы максимальная погрешность вычисляемого интеграла не достигала 0.0005. Тогда максимальная погрешность результата не достигнет 0.0015, что удовлетворяет условию задания. Выбор функции для вычисления интегралов был сделан вручную, для  $f_3$  вычислялся самый маленький  $\varepsilon_1$ .

Отрезок  $[\varepsilon_1, 3]$  для поиска корней был найден методом подбора, но выбор был обоснован:

$$(f_1(\varepsilon_1) - f_2(\varepsilon_1)) > 0; (f_1(3) - f_2(3)) < 0$$

$$(f_2(\varepsilon_1) - f_3(\varepsilon_1)) < 0; (f_2(3) - f_3(3)) > 0$$

$$(f_1(\varepsilon_1) - f_3(\varepsilon_1)) < 0; (f_1(3) - f_3(3)) > 0$$

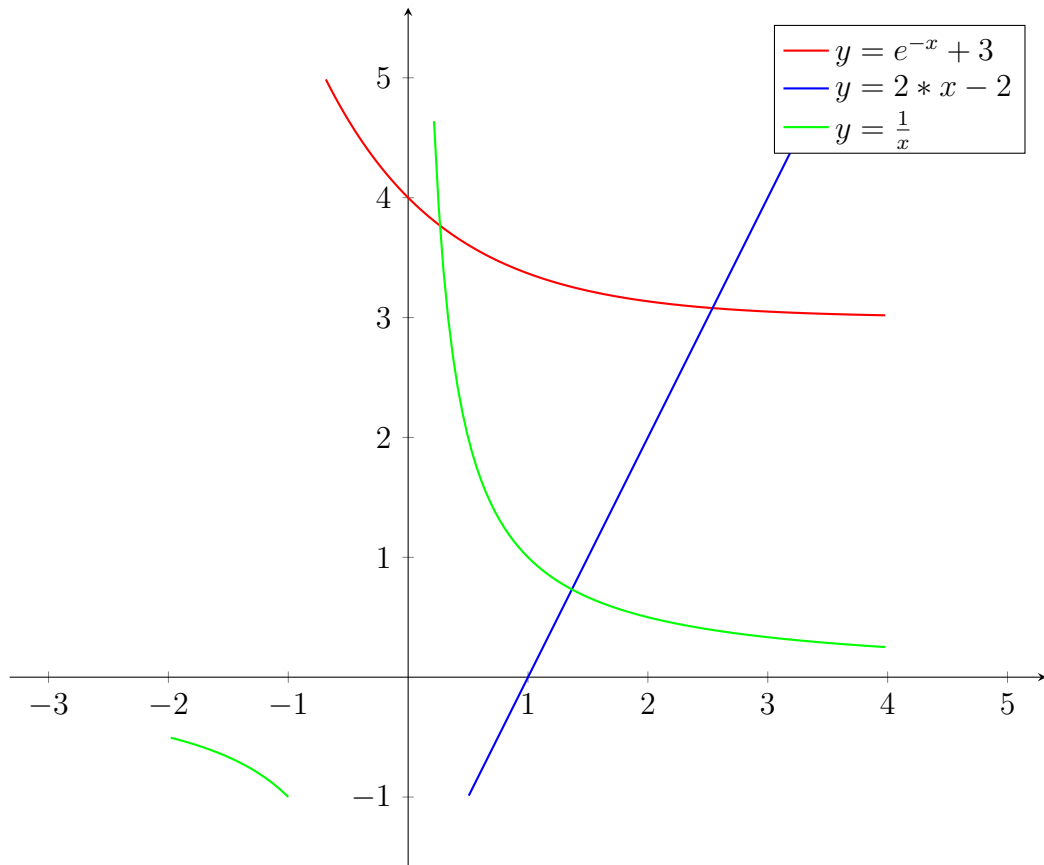


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

## Результаты экспериментов

Кривые	$x$	$y$
1 и 2	2.539455	3.078909
2 и 3	1.366025	0.732051
1 и 3	0.265474	3.766842

Таблица 1: Координаты точек пересечения

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

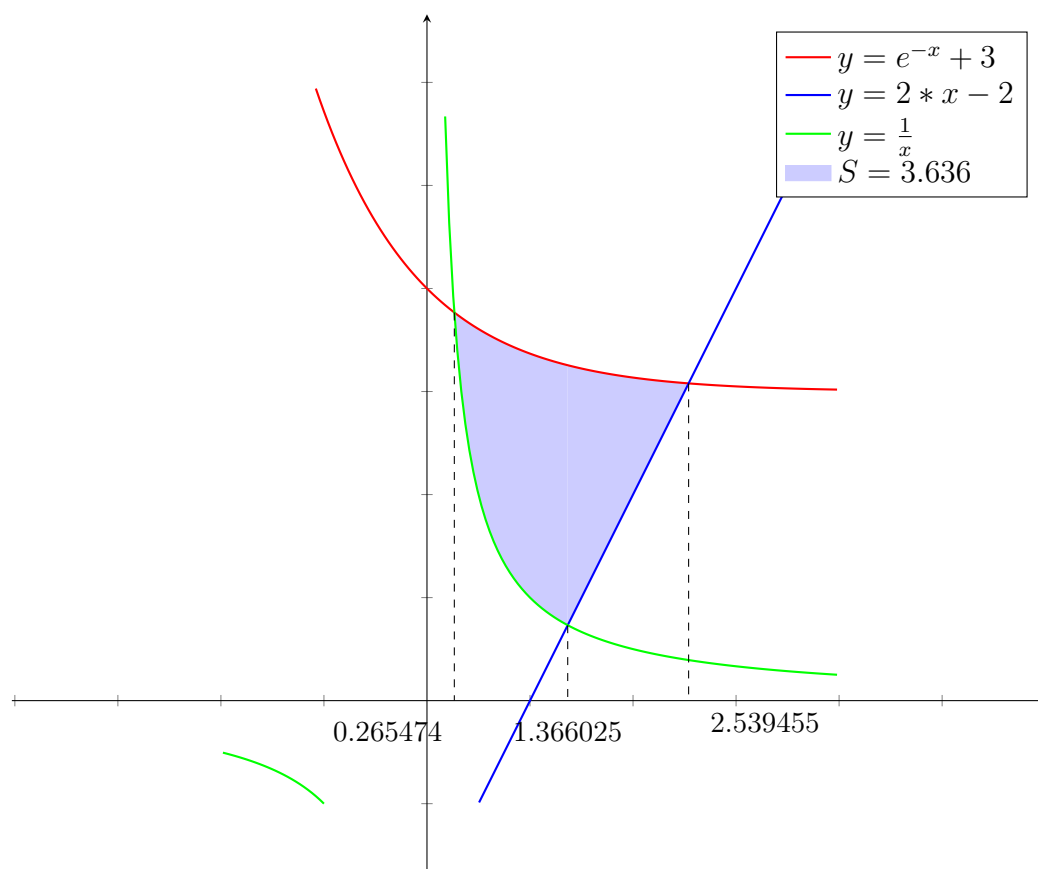


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

# Структура программы и спецификация функций

Полный список функций модуля на языке Си и описание их работы:

- F

Принимает на вход указатели на две функции вида *double f(double x)* и число типа *double*.

Возвращает значение разности этих функций в данной точке, тип *double*.

- root

Принимает на вход указатели на две функции вида *double f(double x)* и *double g(double x)*, два числа типа *double* - границы отрезка, точность вычислений типа *double* и указатели на производные *double ff(double x)* и *double gg(double x)*.

Вычисляет методом касательных значение уравнения  $f(x) = g(y)$  на отрезке  $[a, b]$  с точностью  $\varepsilon_1$ .

Возвращает решение уравнения, тип - *double*.

- integral

Принимает на вход указатель на функцию вида *double f(double x)*, два числа типа *double* - границы интегрирования и точность вычислений типа *double*.

Вычисляет интеграл функции  $f$  методом Симпсона на отрезке  $[a, b]$  с точностью  $\varepsilon_1$ .

Возвращает значение определённого интеграла, тип - *double*.

- testroot

Не принимает аргументы.

Сначала считывает два целых числа - номера функций, сохраняет указатели на них и их производные, затем считывает отрезок для поиска корня и точность вычислений. С этими параметрами и вызывается функция *root*, значение найденного корня сразу выводится.

Ничего не возвращает.

- testintegral

Не принимает аргументы.

Сначала считывает одно целое число - номер функции, сохраняет указатель на неё, затем считывает пределы интегрирования и точность вычислений. С этими параметрами и вызывается функция *integral*, его значение сразу выводится.

Ничего не возвращает.

- main

Главная функция, в которой начинается работа программы.

Может принимать встроенные аргументы.

Вызывает root для нахождения точек пересечения кривых, вызывает integral для нахождения определённых интегралов трёх прямых на отрезках, определённых с помощью root, считает площадь данной плоской фигуры. выполняет некоторые команды при помощи своих аргументов.

Модуль на языке Ассемблера представляет собой 6 функций - 3 функции и их производные. Каждая функция получает на вход число типа double, вычисляет значения в этих точках и возвращает их же.

## Сборка программы (Make-файл)

```
CC1=gcc
CC2=nasm
F1=-f elf32
F2=-c -m32
F3=-m32
all: 6.o 6hw.o
    $(CC1) 6.o 6hw.o $(F3)
6.o: 6.asm
    $(CC2) $(F1) 6.asm
6hw.o: 6hw.c
    $(CC1) 6hw.c $(F2)
clean:
    rm *.o
```



## Отладка программы, тестирование функций

Для тестирования функций `root` и `integral` при запуске программы необходимо написать соответственно `./a.out test root` и `./a.out test integral`.

- Тестирование `root`.

Тест 1:

$$\frac{1}{x} = e^{-x} + 3 \text{ на } [0.1, 4] \text{ при } \varepsilon_1 = 0.01$$

Тест 2:

$$2 * x - 2 = \frac{1}{x} \text{ на } [1, 3] \text{ при } \varepsilon_1 = 0.00001$$

Тест 3:

$$2 * x - 2 = e^{-x} + 3 \text{ на } [-1, 5] \text{ при } \varepsilon_1 = 0.001$$

Тест	<i>prog</i>	<i>Wolfram</i>
1	0.26	0.26
2	1.36603	1.36603
3	2.539	2.539

Таблица 2: Результаты тестов `root`

- Тестирование `integral`.

Тест 1:

$$f = \frac{1}{x} \text{ на } [2, 4] \text{ при } \varepsilon_1 = 0.001$$

Тест 2:

$$f = 2 * x - 2 \text{ на } [-3, 2] \text{ при } \varepsilon_1 = 0.01$$

Тест 3:

$$f = e^{-x} + 3 \text{ на } [0, 5] \text{ при } \varepsilon_1 = 0.00001$$

Тест	<i>prog</i>	<i>Wolfram</i>
1	0.693	0.693
2	-15.00	-15.00
3	15.99328	15.99328

Таблица 3: Результаты тестов `integral`

## Анализ допущенных ошибок

Все ошибки, допущенные при работе над этим заданием, произошли по невнимательности и были своевременно устранены.

Так, например, при реализации формулы Симпсона для вычисления определённого интеграла внутри цикла во время прибавления к суммам очередного значения функции был потерян знак "+" в конструкции "+=". Ошибка была обнаружена не так оперативно, как хотелось бы, но всё же была устранена.