

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 11 / 1-4 / 1-3**

Выполнил:  
студент 102 группы  
Такшин А. И.

Преподаватель:  
Гуляев Д. А.

Москва  
2024

# Содержание

Постановка задачи	2
Математическое обоснование	3
Алгоритм нахождения площади криволинейного треугольника . . . . .	3
Формулы оценки погрешностей методов . . . . .	4
Оценка общей погрешности . . . . .	4
Результаты экспериментов	5
Структура программы и спецификация функций	6
Сборка программы (Make-файл)	7
Отладка программы, тестирование функций	8
Программа на Си и на Ассемблере	9
Анализ допущенных ошибок	10
Список цитируемой литературы	11

## Постановка задачи

- Требуется реализовать программу, которая численными методами вычисляет площадь фигуры, ограниченной тремя кривыми заданными в виде формул  $f : \mathbb{R} \rightarrow \mathbb{R}$ .
- Для вычисления площади реализованы численные методы интегрирования: через формулу прямоугольников, через формулу трапеций и, используемый по умолчанию, через формулу Симпсона.
- Для нахождения вершин криволинейного треугольника реализованы 4 метода нахождения корня функции на заданном отрезке: метод деления отрезка пополам, метод хорд, метод Ньютона и комбинированный метод (метод хорд + метод Ньютона).
- Так как для корректной работы перечисленных методов необходимо, чтобы значения функции на концах имели разные знаки, отрезок для их применения должен быть вычислен заранее аналитически и задан в файле с описанием функций.

# Математическое обоснование

## Алгоритм нахождения площади криволинейного треугольника

Рассмотрим на примере (рис. 1) порядок вычисления площади криволинейного треугольника образованного графиками трёх функций, в предположении, что любые из двух функций  $f_1, f_2, f_3$  пересекаются ровно в одной точке на заданном отрезке.

Упорядочим точки попарных пересечений функций  $\{A, B, C\}$ , тогда для вычисления площади достаточно сложить модули интегралов функций  $g_1(x), g_2(x)$  на отрезках  $[A_x; B_x], [B_x; C_x]$ , где  $g_1(x)$  разность функций, точка пересечения которых является точкой  $A$ , а  $g_2(x)$  разность функций, точка пересечения которых является точкой  $C$ . Так как функции пересекаются в одной точке, знак разности любых двух из них не будет меняться после прохождения точки пересечения, а значит полученная сумма будет корректно определять площадь.

В данном примере, точка  $A$  является точкой пересечения  $f_2(x), f_3(x)$ , поэтому программа изначально вычислит модуль интеграла на отрезке  $[A_x, B_x]$  функции  $g_1(x) = f_2(x) - f_3(x)$ . Точка  $C$  является точкой пересечения  $f_1(x), f_3(x)$ , поэтому вторым слагаемым будет модуль интеграла  $g_2(x) = f_1(x) - f_3(x)$

$$S(\Phi) = \left| \int_{A_x}^{B_x} f_2(x) - f_3(x) dx \right| + \left| \int_{B_x}^{C_x} f_1(x) - f_3(x) dx \right|$$

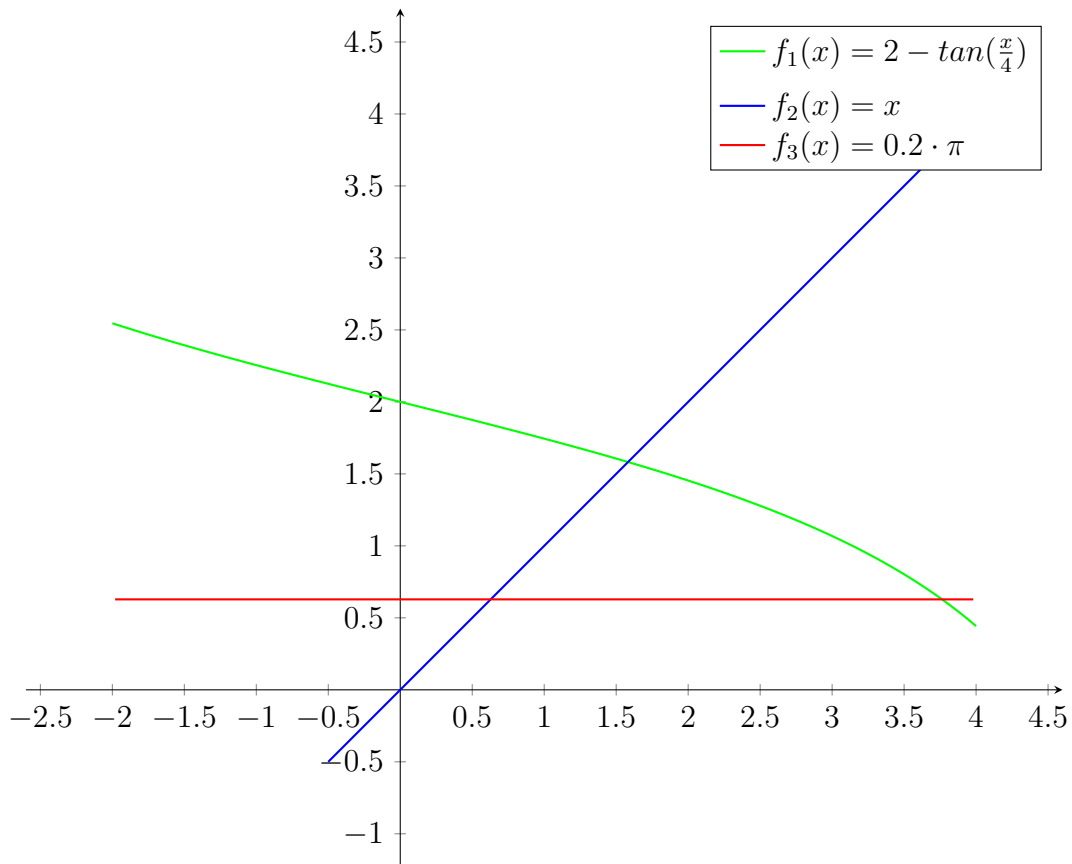


Рис. 1: Пример плоской фигуры, ограниченной графиками заданных уравнений

## Формулы оценки погрешностей методов

Абсолютная погрешность по абсциссе при вычислении точки пересечения  $\varepsilon_1$  задается непосредственно в вычисляющем её методе. Для нахождения значения  $\varepsilon_2$  - абсолютной погрешности при вычислении определенного интеграла использовались известные формулы оценки погрешности [1]:

- $\varepsilon_2 = \frac{f'(\xi)}{2} h^2 (b - a)$ , для формулы прямоугольников
- $\varepsilon_2 = \frac{f''(\xi)}{12} h^2 (b - a)$ , для формулы трапеций
- $\varepsilon_2 = \frac{f^{(4)}(\xi)}{2880} h^4 (b - a)$ , для формулы Симпсона

Учитывая, что  $h = \frac{(b-a)}{n}$  и предполагая  $f^{(n)} \approx 1$ , получаем следующие оценки на число шагов:

- $n = \sqrt[2]{\frac{(b-a)^3}{2\varepsilon_2}}$ , для формулы прямоугольников
- $n = \sqrt[2]{\frac{(b-a)^3}{12\varepsilon_2}}$ , для формулы трапеций
- $n = \sqrt[4]{\frac{(b-a)^5}{2880\varepsilon_2}}$ , для формулы Симпсона

## Оценка общей погрешности

Пусть с помощью описанных выше методом мы получили оценку  $I'$  для интеграла на отрезке  $[a'; b']$ , где  $a' = a + \varepsilon_1$ ,  $b' = b + \varepsilon_1$ , и пусть  $I$  - действительное значение интеграла на отрезке  $[a; b]$ . Тогда из разложения в ряд Тейлора:

$$I' = I + f(a)\varepsilon_1 + f(b)\varepsilon_1 + o(\varepsilon_1)$$

$$I' - I \approx f(a)\varepsilon_1 + f(b)\varepsilon_1$$

Итоговая ошибка вычисления двух интегралов составит:

$$\varepsilon_3 = (f(A_x) + 2f(B_x) + f(C_x))\varepsilon_1$$

Разобьём требуемую наибольшую ошибку  $\varepsilon$  пополам между  $\varepsilon_3$  и  $\varepsilon_2$ . Тогда итоговые оценки для  $\varepsilon_1$  и  $\varepsilon_2$  будут такими:

- $\varepsilon_1 = \frac{\varepsilon}{2}$
- $\varepsilon_2 = \frac{\varepsilon}{2(f(A_x) + 2f(B_x) + f(C_x))}$

## Результаты экспериментов

В результате работы программы для тестового примера (рис. 1) были получены следующие координаты точек пересечения:

Кривые	$x$	$y$
1 и 2	1.5824	1.5824
2 и 3	0.6283	0.6283
1 и 3	3.7634	0.6283

Таблица 1: Координаты точек пересечения

После чего программа корректно нашла искомую площадь криволинейного прямоугольника (рис. 2).

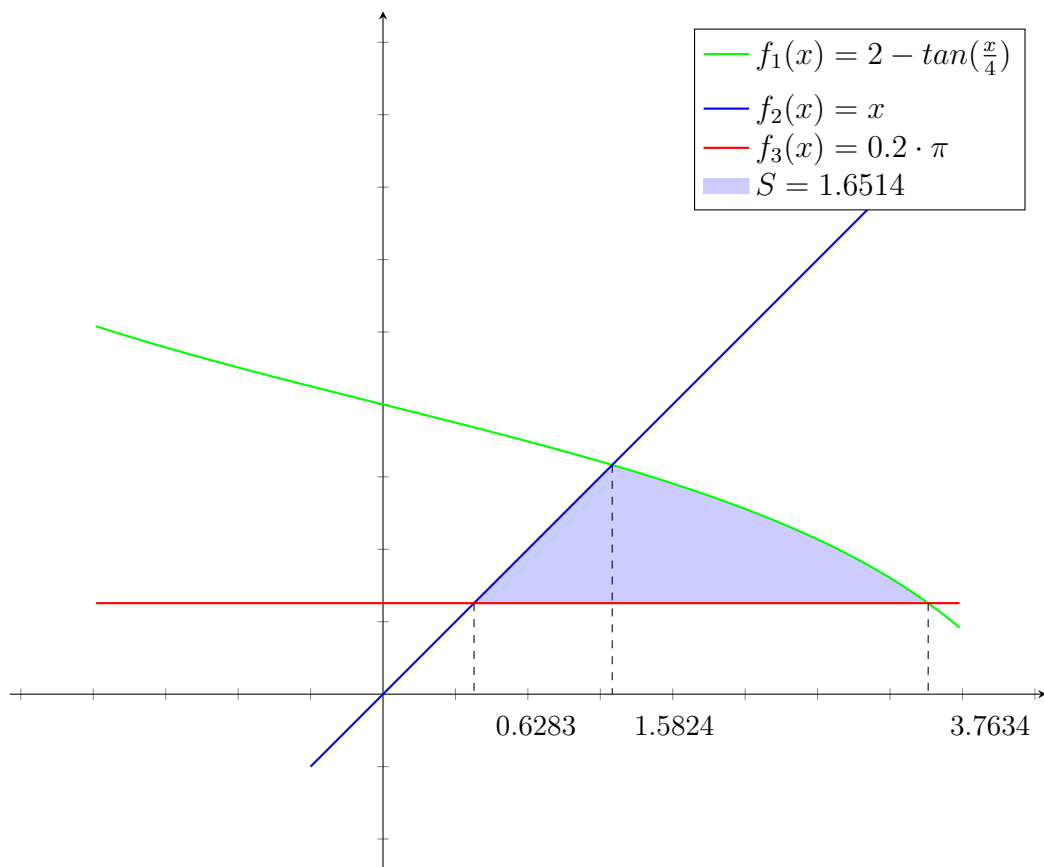


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

# Структура программы и спецификация функций

Программа состоит из следующих файлов

Основные файлы:

- `main.c` - устанавливает порядок запуска функций из других файлов, обрабатывает аргументы командной строки.
- `root.c/h` - содержит функции, вычисляющие пересечение заданной функции с осью абцисс.
- `integral.c/h` - содержит функции, вычисляющие интеграл

Файлы подпрограммы, которая переводит польскую нотацию в `asm` код:

- `function_to_asm.c` - считывает из заданного файла польскую нотацию, вызывает функции из других файлов. На выходе возвращает в `stdin` полученный листинг ассемблера.
- `operation_tree.c/h` - строит дерево, описывающие порядок вычисления выражений. Разрешает проблемы связанные с нехваткой регистров `x87` процессора.
- `constant.h` - содержит константы, необходимые для работы описанных выше функций.

Файлы вспомогательных функций

- `cvector.c/h` - содержит реализацию динамического массива.
- `tools.c/h` - содержит вспомогательные простые функции.
- `macro.h` - содержит макросы препроцессора.

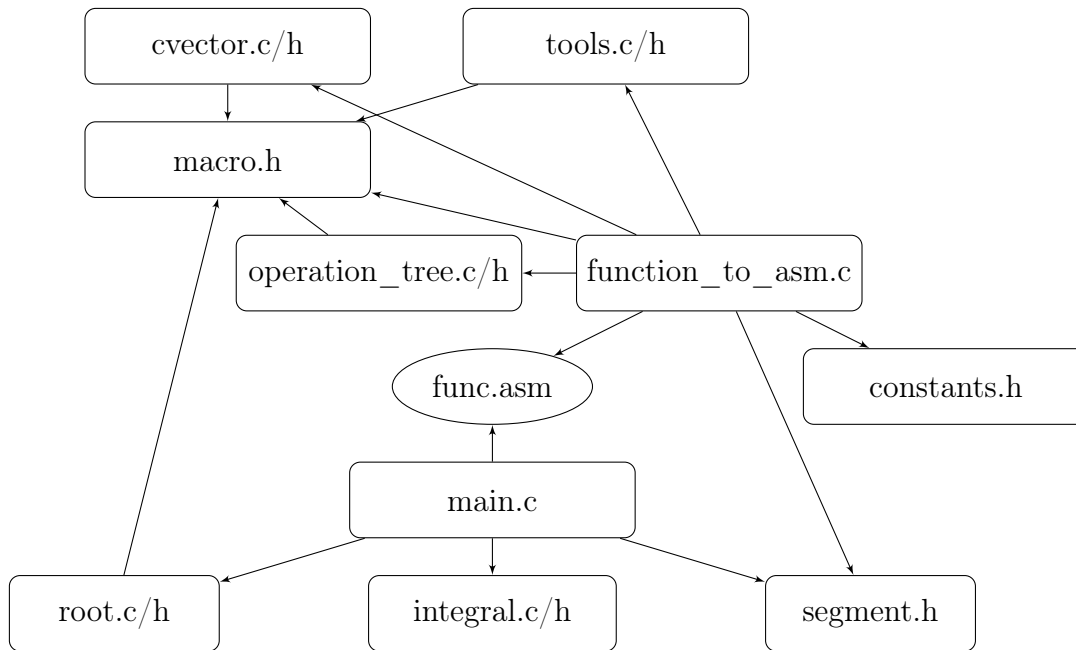


Рис. 3: Графическое представление структуры программы

## Сборка программы (Make-файл)

Зависимости между модулями программы полностью соотносятся с графическим представлением её структуры (рис. refgraph1). Исходный файл Make-файла можно найти в архиве или репозитории (см. Программа на Си и на Ассемблере).

Для задания необходимых методов вычисления точки пересечения и функции, по которым будут производиться вычисления интегралов, нужно использовать аргументы CFLAGS при вызове make all. Например, для компиляции проекта с использованием метода Ньютона и формулы прямоугольников можно использовать следующую команду:

```
make clean CFLAGS=DNEWTON_METHOD-DRECTANGLE_RULE"make
all
```



## Отладка программы, тестирование функций

Для тестирования функций и отладки программы использовался специальный ключ препроцессора (-DDEBUG).

Функции описанные в файле root.c были протестированы на следующих примерах:  $f(x) = x$ ,  $f(x) = x^2 - 2$ ,  $f(x) = \cos(x)$ , для каждого из тестов подбирался соответствующий отрезок для поиска корня:  $[-1; 1]$ ,  $[0; 2]$ ,  $[1; 4]$ . Полученные результаты полностью совпали с вычисленными аналитически.

Функции из файла integral.c тестировались сразу в полной сборке программы. Вот входных данных в spec.txt, задающем кривые ограничивающие криволинейный треугольник, аналитически вычисленные значения площади и полученные результаты в ходе работы программы:

Curves	Segment	Predicted	Stdout
$y_1 = 0, y_2 = x, y_3 = 2 - x$	$[-4; 4]$	1.0000	1.0000
$y_1 = 2 - tg(\frac{\pi}{4}), y_2 = x, y_3 = 0.2 \cdot \pi$	$[0; 4]$	1.6514	1.6516
$y_1 = \cos(x), y_2 = 2x, y_3 = 0$	$[-1; 2]$	0.7900	0.7678

Таблица 2: Результаты тестирования программы

В результате тестирования программы, было обнаружено, что на некоторых данных погрешность получается больше ожидаемой. Данная проблема не исправляется даже при уменьшении  $\varepsilon \leq 10^{-7}$ .

Тем не менее, результаты программы в большинстве случаев сходятся с прогнозируемыми с требуемой точностью.

## Программа на Си и на Ассемблере

Исходные тексты программ имеются в архиве, который приложен к этому отчету. А ещё исходные тексты программ можно найти на репозитории [github](#).

## Анализ допущенных ошибок

## Список литературы

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