

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 6 / 4 / 2

Выполнил:
студент 105 группы
Лабутин А. А.

Преподаватель:
Смирнов А. В.

Москва
2019

Содержание

Постановка задачи	2
Математическое обоснование	3
Результаты экспериментов	5
Структура программы и спецификация функций	7
Сборка программы (Make-файл)	8
Отладка программы, тестирование функций	9
Программа на Си и на Ассемблере	10
Анализ допущенных ошибок	11
Список цитируемой литературы	12

Постановка задачи

Требуется реализовать численный метод, позволяющий вычислять площадь плоской фигуры, ограниченной тремя кривыми:

1) $y = 0.6x + 3$

2) $y = (x - 2)^3 - 1$

3) $y = \frac{3}{x}$

Вершины фигуры определяются как точки пересечения соответствующих графиков функций. Для их нахождения используется комбинированный метод хорд и касательных, вычисляющий корень уравнения $f(x) = g(x)$ на отрезке, определяемом аналитически.

Площадь фигуры есть определенный интеграл. Для его вычисления надо использовать метод трапеций.

Математическое обоснование

Графики заданных функций изображены на рис. 1. Из него видно, что площадь фигуры есть сумма 3-х интегралов [1]:

- 1) от функции $f_1(x) - f_3(x)$ на отрезке $[x_{13}^1; x_{23}^1]$
- 2) от функции $f_1(x) - f_2(x)$ на отрезке $[x_{23}^1; x_{13}^2]$
- 3) от функции $f_3(x) - f_2(x)$ на отрезке $[x_{13}^2; x_{23}^2]$

Здесь x_{ij}^1, x_{ij}^2 - абсциссы точек пересечения графиков функций f_i и f_j .

Для поиска абсцисс точек пересечения используется комбинированный метод хорд и касательных. Его суть состоит в итерационном приближении к корню уравнения $f_i(x) = f_j(x)$ с двух сторон отрезка, определяемого аналитически для каждого случая:

1) $f_1(x) = f_3(x) : x_{13}^1 \in [-7.0; -5.0], x_{13}^2 \in [0.1; 1.5]$

2) $f_2(x) = f_3(x) : x_{23}^1 \in [-1.0; 0.1], x_{23}^2 \in [2.0; 4.0]$

Чтобы гарантировать единственность корня уравнения $F(x) = f_i(x) - f_j(x) = 0$ на соответствующей отрезке, требуется, чтобы производная $F'(x)$ функции $F(x) = f_i(x) - f_j(x) = 0$ имела постоянный знак, а на концах отрезка принимала значения разных знаков: $F'(a)F'(b) < 0$. Формулы для вычисления корня уравнения $F(x) = f_i(x) - f_j(x) = 0$ [1]:

- 1) методом хорд:

$$c_1 = \frac{aF(b) - bF(a)}{F(b) - F(a)}$$

- точка пересечения хорды $(a; F(a)) - (b; F(b))$ с осью абсцисс.

- 2) методом касательных:

$$c_2 = d - \frac{F(d)}{F'(d)}$$

- точка пересечения касательной с осью абсцисс.

Если на $[a; b]$ $F'(x)F''(x) > 0$, то функция $F(x)$ либо возрастает и выпукла вниз, либо убывает и выпукла вверх. В этом случае метод хорд используется для приближения к корню слева, а метод касательных - справа. Тогда в 1-й формуле на очередном шаге берется отрезок $[c; b]$ и уже к нему применяется формула. Во 2-й формуле точка $d = b$ и на очередном шаге строится касательная в точке $(c_2; F(c_2))$, после чего формула применяется снова.

Если же на $[a; b]$ $F'(x)F''(x) < 0$, то функция $F(x)$ либо возрастает и выпукла вверх, либо убывает и выпукла вниз. В этом случае метод хорд используется для приближения к корню справа, а метод касательных - слева. Тогда в 1-й формуле на очередном шаге берется отрезок $[a; c_1]$ и уже к нему применяется формула. Во 2-й формуле точка $d = a$ и на очередном шаге строится касательная в точке $(c_2; F(c_2))$, после чего формула применяется снова.

Корень считается найденным с точностью ε_1 , если $|c_2 - c_1| < \varepsilon_1$.

После нахождения требуемых корней вычисляются три интеграла на соответствующих подотрезках $[a; b]$ методом трапеций с точностью ε_2 . Для этого выбирается начальное число разбиений $n = 10$ и на очередной итерации вычисляются значения интегралов I_n и I_{2n} для числа разбиений n и $2n$ отрезка $[a; b]$ по формуле:

$$I_n \approx h(0.5F_0 + F_1 + \dots + F_{n-1} + 0.5F_n),$$

где $F(x) = f_i(x) - f_j(x)$, $F_i = F(a + ih)$, $h = \frac{b-a}{n}$. Если $p|I_{2n} - I_n| \geq \varepsilon_2$ ($p = \frac{1}{3}$ - константа Рунге для метода трапеций), то число разбиений n увеличивается вдвое и итерация повторяется. Иначе искомый интеграл I есть I_{2n} .

Чтобы площадь фигуры как сумма интегралов на подотрезках была найдена с точностью ε , достаточно, чтобы каждый из трёх интегралов был вычислен с точностью $\varepsilon_2 = \frac{\varepsilon}{3}$ [1].

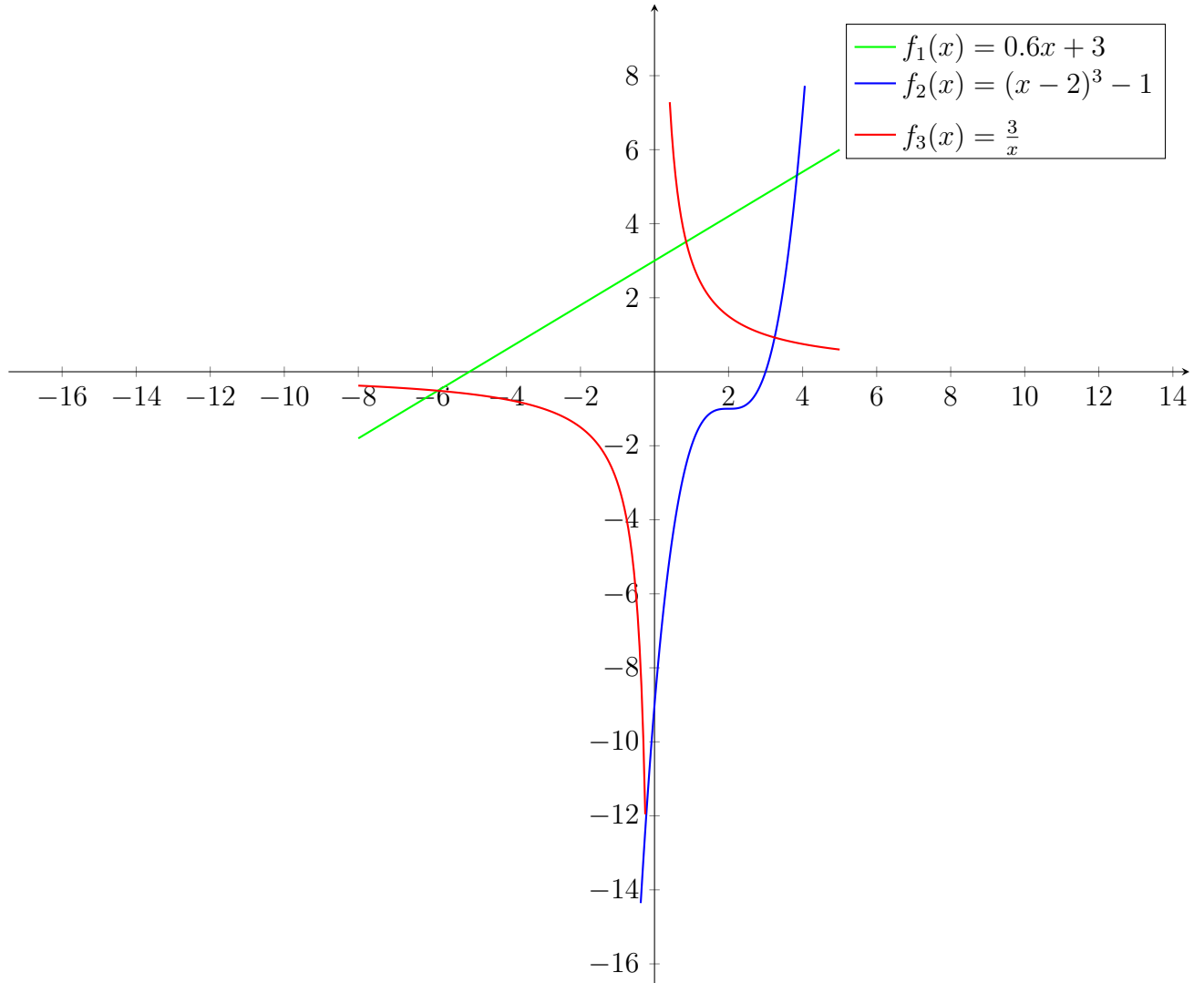


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Результаты экспериментов

Координаты точек пересечения кривых, вычисленные с разной точностью ε_1 , приведены в таблицах 1, 2, 3.

ε_1	x	y
0.1	3.823324	5.293994
0.01	3.842851	5.305711
0.001	3.847328	5.308397
0.0001	3.847722	5.308633
0.00001	3.847757	5.308654
0.000001	3.847760	5.308656

Таблица 1: Координаты точек пересечения кривых 1 и 2, вычисленные с разной точностью ε_1

ε_1	x_1	y_1	x_2	y_2
0.1	-5.854167	-0.512447	0.901598	3.540959
0.01	-5.854319	-0.512442	0.858980	3.515388
0.001	-5.854319	-0.512442	0.854542	3.512725
0.0001	-5.854107	-0.512461	0.854148	3.512489
0.00001	-5.854102	-0.512461	0.854107	3.512464
0.000001	-5.854102	-0.512461	0.854102	3.512461

Таблица 2: Координаты точек пересечения кривых 1 и 3, вычисленные с разной точностью ε_1

ε_1	x_1	y_1	x_2	y_2
0.1	-0.271254	-12.716479	3.213015	0.784836
0.01	-0.247740	-12.356332	3.241830	0.915078
0.001	-0.244210	-12.302917	3.243586	0.923212
0.0001	-0.243969	-12.299273	3.243906	0.924699
0.00001	-0.243932	-12.298713	3.243925	0.924787
0.000001	-0.243929	-12.298674	3.243929	0.924803

Таблица 3: Координаты точек пересечения кривых 2 и 3, вычисленные с разной точностью ε_1

Результаты проиллюстрированы графиком (рис. 2).

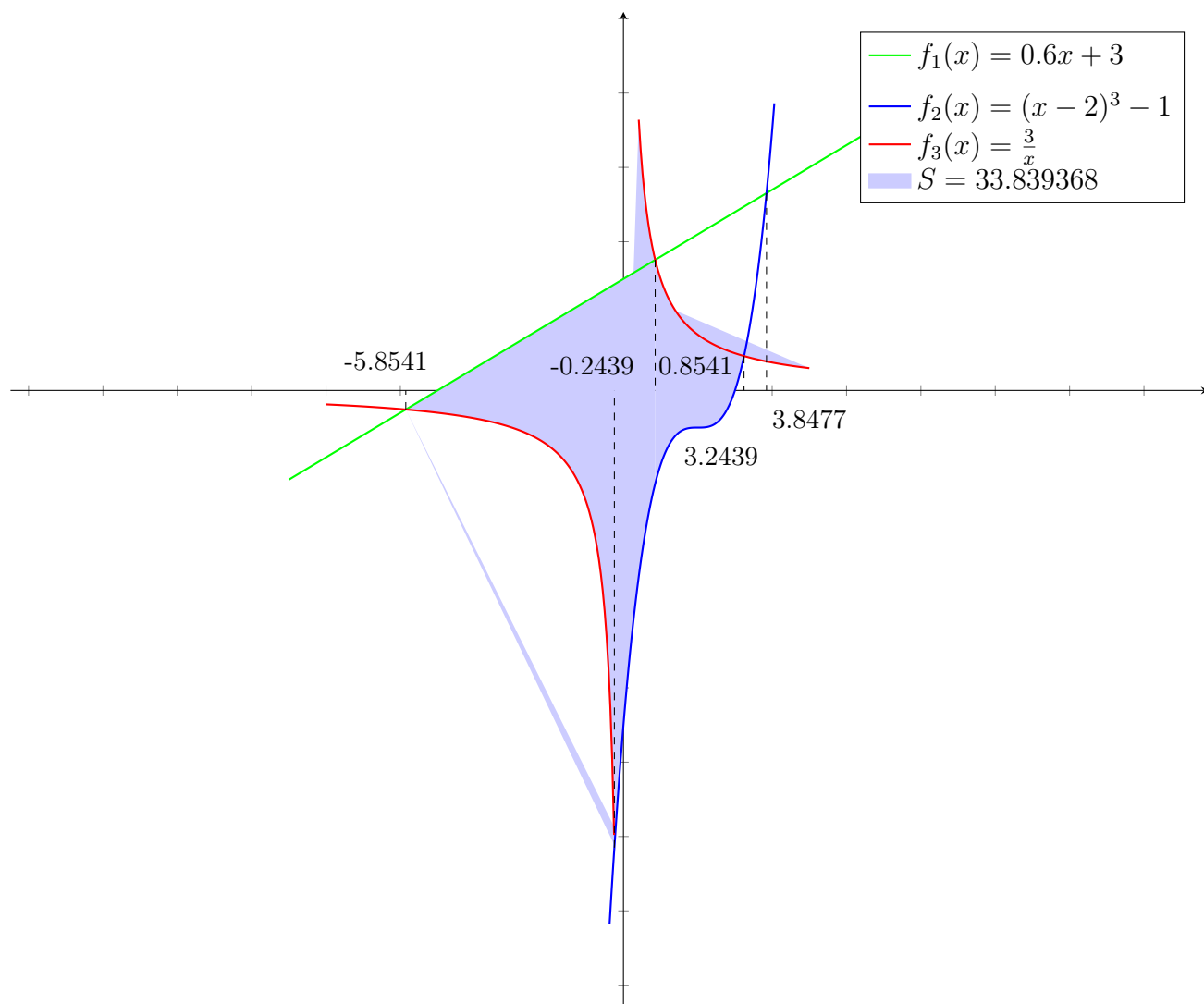


Рис. 2: Плоская фигура, ограниченная графиками заданных функций

Структура программы и спецификация функций

В программе используются следующие модули:

1. main.c - вычисление площади фигуры с помощью функций:
 - main - обработка аргументов командной строки (распознавание ключей и их параметров), вычисление площади фигуры с точностью ε как суммы интегралов на подотрезках;
 - F - возврат значения разности функций $f_i(x) - f_j(x)$ в точке-аргументе;
2. root.c - вычисление корня уравнения $f_i(x) = f_j(x)$ с помощью функций:
 - root - использование комбинированного метода хорд и касательных на заданном отрезке для приближенного вычисления корня с заданной точностью ε_1 ;
3. integral.c - вычисление интеграла на отрезке с помощью функций:
 - integral - использование метода трапеций для приближенного вычисления интеграла от функции $f_i(x) - f_j(x)$ на заданном отрезке с заданной точностью ε_2
4. f1.asm - вычисление значения функции $f_1(x)$ в точке x
5. f2.asm - вычисление значения функции $f_2(x)$ в точке x
6. f3.asm - вычисление значения функции $f_3(x)$ в точке x
7. f1_1.asm - вычисление значения производной функции $f_1(x)$ в точке x
8. f2_1.asm - вычисление значения производной функции $f_2(x)$ в точке x
9. f3_1.asm - вычисление значения производной функции $f_3(x)$ в точке x
10. main.h - объявления функций выше для использования в модуле main.c

Сборка программы (Make-файл)

Makefile:

```
all: prog
main.o: main.c
    gcc -g -o main.o -c main.c -m32
root.o: root.c
    gcc -g -o root.o -c root.c -m32
integral.o: integral.c
    gcc -g -o integral.o -c integral.c -m32
f1.o: f1.asm
    nasm -g -o f1.o f1.asm -f elf32
f2.o: f2.asm
    nasm -g -o f2.o f2.asm -f elf32
f3.o: f3.asm
    nasm -g -o f3.o f3.asm -f elf32
f1_1.o: f1_1.asm
    nasm -g -o f1_1.o f1_1.asm -f elf32
f2_1.o: f2_1.asm
    nasm -g -o f2_1.o f2_1.asm -f elf32
f3_1.o: f3_1.asm
    nasm -g -o f3_1.o f3_1.asm -f elf32
prog: main.o root.o integral.o f1.o f2.o f3.o f1_1.o f2_1.o f3_1.o
    gcc -g -o prog main.o root.o integral.o f1.o f2.o f3.o f1_1.o f2_1.o
    f3_1.o -m32
clean:
    rm -rf *.o prog
```

Как видно из исходного кода, исполняемый файл *prog* зависит от файлов *main.c*, *root.c*, *integral.c*, *f1.asm*, *f2.asm*, *f3.asm*, *f1_1.asm*, *f2_1.asm*, *f3_1.asm*.

Отладка программы, тестирование функций

Тестирование проводилось для следующих модулей:

- root.c - для различных значений ε_1 (см. раздел "Результаты экспериментов")
- integral.c - для различных значений ε_2 (см. таблицу 4)

ε	I_1	I_2	I_3	I
0.1	13.765716	13.869583	6.204069	33.839368
0.01	13.765716	13.85611	6.217544	33.839370
0.001	13.765714	13.847472	6.226182	33.839368
0.0001	13.765714	13.847037	6.226617	33.839368
0.00001	13.765714	13.846966	6.226688	33.839368
0.000001	13.765714	13.846963	6.226691	33.839368

Таблица 4: Вычисление промежуточных интегралов I_1 , I_2 , I_3 и интеграла I с различной точностью ε_1

Программа на Си и на Ассемблере

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

Анализ допущенных ошибок

В основном, все допущенные ошибки были совершены по невнимательности и из-за опечаток в ключевых словах языков C и Assembler. Логических ошибок вычисления корней уравнений, интегралов допущено не было.

Список литературы

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