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 2 / 2 / 2

Выполнил:
студент 103 группы
Иванов М. Ю.

Преподаватель:
Кузьменкова Е. А.

Москва
2019

Содержание

Постановка задачи

Необходимо реализовать многомодульную программу, вычисляющую площадь плоской фигуры, ограниченной графиками трех функций с заданной точностью ε. Для нахождения вершин фигуры использовался **метод хорд**. Отрезок для применения данного метода должен быть вычислен аналитически. Подсчет площади плоской фигуры производился с помощью **метода трапеций**.

Математическое обоснование

Функции

Необходимо было найти площадь между тремя кривыми, заданных функциями:

1. $f_1 = 3 * (\frac{0.5}{(x+1)} + 1)$
2. $f_2 = 2.5 * x - 9.5$
3. $f_3 = \frac{5}{x}$

Ниже приведены графики данных функций (рис. 1).

Поиск корней

Нахождение корней проводилось с помощью **метода хорд** с вычислительной точностью $\varepsilon_1 = 0.001$ на промежутке $[0.5, 7]$. Для использования данного метода необходимо выполнение следующих условий на отрезке $[a, b]$:

1. $F(x) \in C^1[a, b]$;
2. $F(a) * F(b) < 0$;
3. $F'(x)$ монотонна на $[a, b]$;
4. $F'(x)$ сохраняет знак на $[a, b]$. x[1]

Поиск интеграла

Интегрирование проводилось **методом трапеций** с вычислительной точностью $\varepsilon_2 = 0.001$.

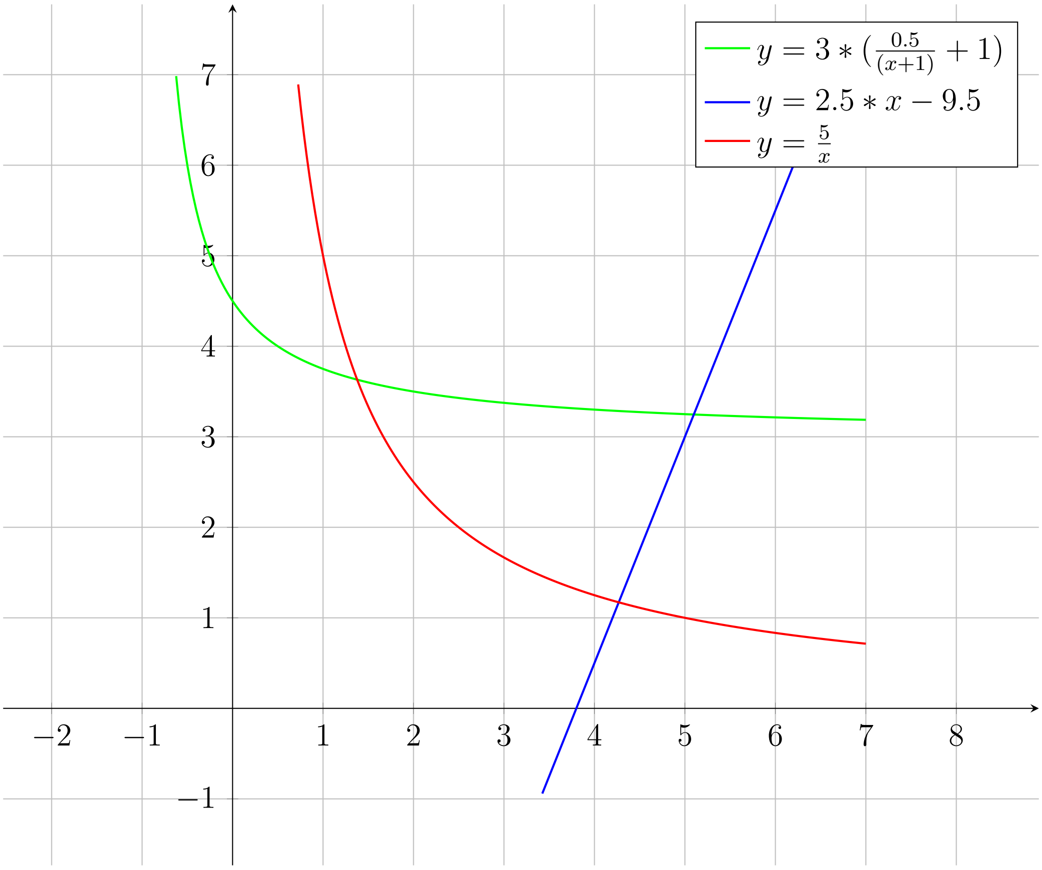

Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Обоснавание выбора промежутка $[a, b]$

Ниже приведено математическое обоснование выполнения условий применимости **метода хорд** поиска корней на промежутке $[0.5, 7]$.

1. $F_{12}(x) = 3 * (\frac{0.5}{(x+1)} + 1) - 2.5 * x + 9.5$

   (a) $F'_{12}(x) = -\frac{3}{2(x+1)^2} - 2.5$
   Производная функции непрерывна на промежутке $[0.5, 7]$.

   (b) $F_{12}(0.5) = 4 + 8.25 = 12.25 > 0$,
   $F_{12}(7) = \frac{51}{16} + 8 = -\frac{77}{17} < 0$
   Тогда $F_{12}(0.5) * F_{12}(7) < 0$.

   (c) $F''_{12}(x) = \frac{3}{(x+1)^3} > 0$ (на промежутке $[0.5, 7]$) => первая производная возрастает на этом промежутке.

   (d) $F'_{12}(x) < 0$ на всем промежутке $[0.5, 7]$.

2. $F_{13}(x) = 3 * (\frac{0.5}{(x+1)} + 1) - \frac{5}{x}$

   (a) $F'_{13}(x) = -\frac{3}{2(x+1)^2} + \frac{5}{x^2}$
   Производная функции непрерывна на промежутке $[0.5, 7]$.

(b) $F_{13}(0.5) = 4 - 10 = -6 < 0$,
$F_{13}(7) = \frac{51}{16} - \frac{5}{7} > 0$
Тогда $F_{13}(0.5) * F_{13}(7) < 0$.

(c) $F''_{13}(x) = \frac{3}{(x+1)^3} - \frac{10}{x^3}$
$F'''_{13}(x) = 3 * (-\frac{3}{(x+1)^4} + \frac{10}{x^4}) = 0$ только при $x < 0$, значит $F''_{13}(x)$ возрастает на промежутке $[a, b]$, так как $F'''_{13}(1) = 3 * (-\frac{3}{16} + 10) > 0$.
При этом $F''_{13}(7) = \frac{3}{8^3} - \frac{10}{7^3} = \frac{3*7^3 - 10*8^3}{56^3} < 0 => F''_{13}(x) < 0$ на всем промежутке $[a, b]$, тогда $F'_{13}(x)$ убывает на данном промежутке.

(d) $F'_{13}(7) = -\frac{3}{128} + \frac{5}{49} > 0$, тогда так как $F'_{13}(x)$ убывает на всем промежутке $[0.5, 7]$, то $F'_{12}(x) > 0$ на данном промежутке.

3. $F_{23}(x) = 2.5 * x - 9.5 - \frac{5}{x}$

(a) $F'_{23}(x) = 2.5 + \frac{5}{x^2}$
Производная функции непрерывна на промежутке $[0.5, 7]$.

(b) $F_{23}(0.5) = -8.25 - 10 < 0$,
$F_{23}(7) = 8 - \frac{5}{8} > 0$
Тогда $F_{23}(0.5) * F_{23}(7) < 0$.

(c) $F''_{23}(x) = -\frac{10}{x^3} < 0$ (на промежутке $[0.5, 7]$) => первая производная убывает на этом промежутке.

(d) $F'_{23}(x) > 0$ на всем промежутке $[0.5, 7]$.

Значения ε_1 и ε_2

При вычислении корней методом хорд имеется погрешность, которая вычисляется по формуле $|x_n - x_{n-1}| < \frac{|F(x_n)|}{m} = \varepsilon_1$, где m - минимальное значение модуля первой производной на сегменте $[a, b]$. [1]

При вычислении площадей методом трапеции имеется погрешность $R = -\frac{F''(\xi)}{12n^2}(b - a)^3 = \varepsilon_2$, где $a \leq \xi \leq b$, n - число разбиений отрезка $[a, b]$ на равные части, a, b - корни уравнений. [1]

Результаты экспериментов

Координаты точек пересечения представлены в таблице (таблица 1) и на графике (рис. 2). **Площадь фигуры, заключенной между кривыми, равна** $S = 5.087$ (рис. 2).

Кривые	x	y
1 и 2	5.078	3.247
2 и 3	1.375	3.632
1 и 3	4.267	1.168

Таблица 1: Координаты точек пересечения

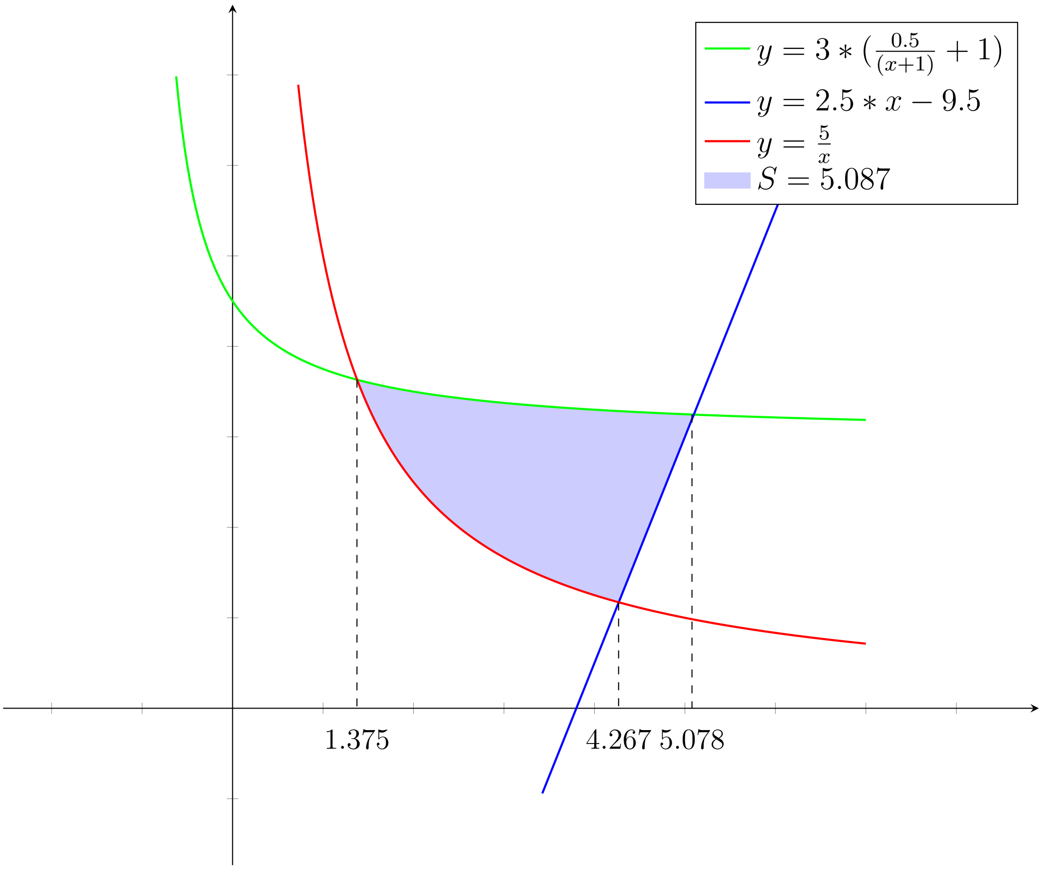

Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Вычислительный модуль

В вычислительном модуле **calc.c** описаны следующие функции:

1. double min(double a, double b)
   Возвращает минимальное из чисел a и b

2. double max(double a, double b)
   Возвращает максимальное из чисел a и b

3. int search_max(double *x, int n)
   Возвращает индекс максимального элемента в массиве вещесвенных чисел из n элементов

4. int search_mid3(double x[])
   Возвращает средний элемент массива из 3 чисел по возрастанию

5. double search_delta(double (*f)(double), double x1, double x2, double eps)
   Возвращает минимальное значение функции f, меньшее 0, или 0 на промежутке от x1 до x2.

6. double integral(double (*f)(double), double a, double b, double eps, double delta)
   Возвращает определенный интеграл функции f на промежутке от a до b с поднятием функции на delta.

7. double root(double (*f)(double), double (*g)(double), double a, double b, double eps)
   Возвращает координату абцисс точки пересечения функций f и g на отрезке от a до b.

Модуль функций

В модуле **func.asm** описаны 3 функции основного задания, на вход каждой из которых подается вещественное число, а на выход - значение функции в заданной точке.

Главный модуль

В главном модуле **main.c** описаны следующие функции:

1. int s_in_a(char *s, char *arr[], int n)
   Проверяет наличие строки s в массиве строк arr размером n и возвращает индекс строки в массиве если она там есть и 0 иначе.

2. double test_f1(double x)
   Тестовая функция 1. Возвращает значение функции в точке x.

3. double test_f2(double x)
   Тестовая функция 2. Возвращает значение функции в точке x.

4. double test_f3(double x)
   Тестовая функция 3. Возвращает значение функции в точке x.

5. int main(int argc, char *argv[])
   Главная функция модуля. В нем происходит подсчет площади и вывод ее на экран.

Библиотеки

Помимо стандартных библиотек языка C, в программе задействована вспомогательная библиотека lib.h, в которую занесены все функции вспомогательных модулей.

Сборка программы (Make-файл)

Итоговый проект square.e собирается из 3 объектных модулей:
main.o func.o calc.o
Они в свою очередь собираются из 3 файлов:
main.c calc.c func.asm
Все фунцкии, использующиеся в программе, описаны в библиотеке: lib.h
Также используется библиотека: io.inc

Все модули находятся в папке SRC.

Итоговый проект состоит из следующих модулей:

- main.c
  Главный модуль. В нем происходит обработка опций, вводимых в командной строке. Также в нем происходит подсчет искомой площади фигуры.

- calc.c
  Вычислительный модуль. В нем находятся функция, вычисляющая корни уравнений, функция, вычисляющая определенный интеграл и другие вспомогательные функции.

- func.asm
  Модуль функций. В нем вычисляются функции, заданные условием задачи.

- lib.h
  Библиотека. Здесь описаны прототипы всех используемых функций.

Для сборки проекта написан makefile, в котором отражены все зависисмости и прописаны все необходимые для компиляции ключи. С текстом makefile можно ознакомиться ниже.

MAKEFILE:

```
TARGET = bin/square.e
PROG_OBJ = obj/main.o obj/func.o obj/calc.o
C_FLAGS = -std=c99 -c -m32 -o
ASM_FLAGS = -f elf32 -DUNIX -o
all: $(TARGET)
run: $(TARGET)
./$(TARGET)
clean:
rm -f $(PROG_OBJ)
$(TARGET): $(PROG_OBJ)
gcc -o $(TARGET) $(PROG_OBJ) -m32
obj/main.o: src/main.c src/lib.h
gcc $(C_FLAGS) obj/main.o src/main.c
obj/%.o: src/%.c
gcc $(C_FLAGS) $@ $<
obj/%.o: src/%.asm
nasm $(ASM_FLAGS) $@ $<
```

Отладка программы, тестирование функций

Тестирование и отладка численных методов производилось на тестовых функций test_f1, test_f2, test_f3.
$f_1 = \frac{2}{x}$
$f_2 = x$
$f_3 = x^2$

Кривые	a	b	x	y
f_1 и f_2	0.5	5	1.413	1.413
f_1 и f_3	0.5	5	1.260	1.587
f_2 и f_3	0.5	5	0.996	0.996

Таблица 2: Координаты точек пересечения тестовых функций

Корни в тестах равны 1.414, 1.260 и 1.0 соответственно и проверяются непосредственно.

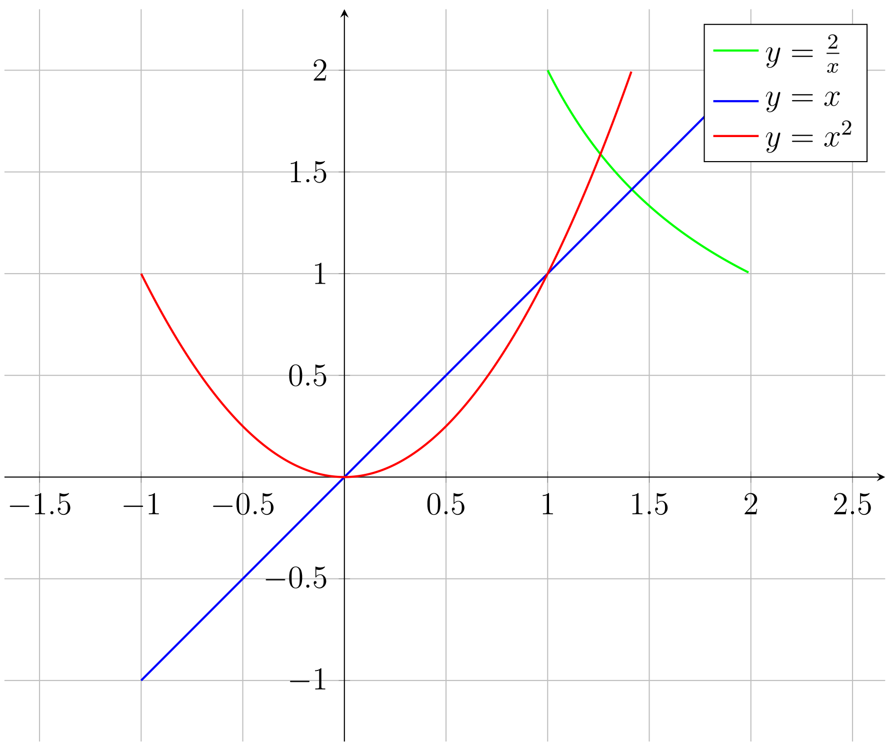

Рис. 3: Графики тестовых функций

Кривая	a	b	Результат
$f_1 = \frac{2}{x}$	1	2	1.386
$f_2 = x$	1	2	1.500
$f_3 = x^2$	1	2	2.333

Таблица 3: Примеры вычисления определенных интегралов

Для заданных тестов первообразными будут соответственно:

1. $\int\limits_1^2 f_1 = 2\ln x + C$

2. $\int\limits_1^2 f_2 = \frac{x^2}{2} + C$

3. $\int\limits_1^2 f_3 = \frac{x^3}{3} + C$

Значения определённых интегралов проверяются непосредственно через формулу Ньютона-Лейбница.

Программа на Си и на Ассемблере

Тексты всех модулей программы, включая библиотеку, имеются в приложенном архиве task6.zip.

Анализ допущенных ошибок

В ходе выполнения задания были допущены некоторые ошибки. В функциях search_mid3 и search_max в качестве параметра принималось int *x вместо double *x, что приводило к некорректной работе программы. Ошибка была допущенна из-за невнимательности и исправлялась изменением типа переменной *x.

Список литературы

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