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 / 3 / 1

Выполнила:
студентка 102 группы
Анисимова Д. В.

Преподаватель:
Сенюкова О. В.

Москва
2020

Содержание

Постановка задачи

Требуется с точностью $\varepsilon = 0.001$ найти площадь плоской фигуры, ограниченной тремя кривыми:

- $y = 3(\frac{0.5}{x+1} + 1)$
- $y = 2.5x - 9.5$
- $y = 5/x, \ x > 0$

Сначала должны быть найдены точки пересечения кривых. Для этого надо реализовать функцию, вычисляющую корень уравнения методом касательных, и аналитически найти интервал, на котором будут найдены корни. Далее нужно реализовать функцию, вычисляющую интеграл методом прямоугольников, и с помощью нее найти искомую площадь. Обе функции должны быть предварительно протестированы.

Математическое обоснование

Сначала найдем первые и вторые производные данных в задании функций.

- $f_1'(x) = -\frac{1.5}{x+1}$
- $f_2'(x) = -2.5$
- $f_3'(x) = -\frac{5}{x^2}$
- $f_1''(x) = \frac{3}{(x+1)^2}$
- $f_2''(x) = 0$
- $f_3''(x) = \frac{10}{x^3}$

Для того, чтобы найти точки пересечения графиков функций $y = f_i(x)$ и $y = f_j(x)$, необходимо рассмотреть функцию $F(x) = f_i(x) - f_j(x)$. Чтобы найти корень уравнения $F(x) = 0$, нужно найти отрезок, на котором первая и вторая производная не меняют знак. Если a и b — концы отрезка, то $F(a) * F(b) < 0$. [2]

1. $F_1(x) = f_1(x) - f_2(x) = 3(\frac{0.5}{x+1} + 1) - 2.5x + 9.5$
   - $F_1'(x) = -2.5 - \frac{1.5}{(x+1)^2} < 0, \ \forall x \neq -1$
   - $F_1''(x) = \frac{3}{(x+1)^3}$
     Как видно из графика, нужный нам корень уравнения $F_1(x)$ — число положительное, значит, можно считать, что $x > 0$ и $F_1''(x) > 0$
   - С учетом графика выберем интервал для поиска корня: $[4; 5,5]$. Проверим, что он подходит:
     $F(4) = 2.875 > 0$
     $F(5.5) \approx -1.019 < 0$

2. $F_2(x) = f_2(x) - f_3(x) = 2,5x - 9.5 - \frac{5}{x}$
   - $F_2'(x) = 2.5 + \frac{5}{x^2} > 0, \ \forall x \neq 0$
   - $F_2''(x) = -\frac{10}{x^3} < 0. \ \forall x \neq 0$
   - С учетом графика выберем интервал для поиска корня: $[3.5; 4.7]$. Проверим, что он подходит:
     $F(3.5) \approx -2.1786 < 0$
     $F(4.7) \approx 1.186 > 0$

3. $F_3(x) = f_1(x) - f_3(x) = 3(\frac{0.5}{x+1} + 1) - \frac{5}{x}$
   - $F_3'(x) = -\frac{1.5}{(x+1)^2} + \frac{5}{x^2} = \frac{3.5x^2+10x+5}{(x+1)^2x^2} > 0, \ \forall x > 0$
   - $F_3''(x) = \frac{3}{(x+1)^3} - \frac{10}{x^3} < 0, \ \forall x > 0$
   - С учетом графика выберем интервал для поиска корня: $[0.5; 2.3]$. Проверим, что он подходит:
     $F_3(0.5) = -6 < 0$
     $F_3(2.3) \approx 1.28$

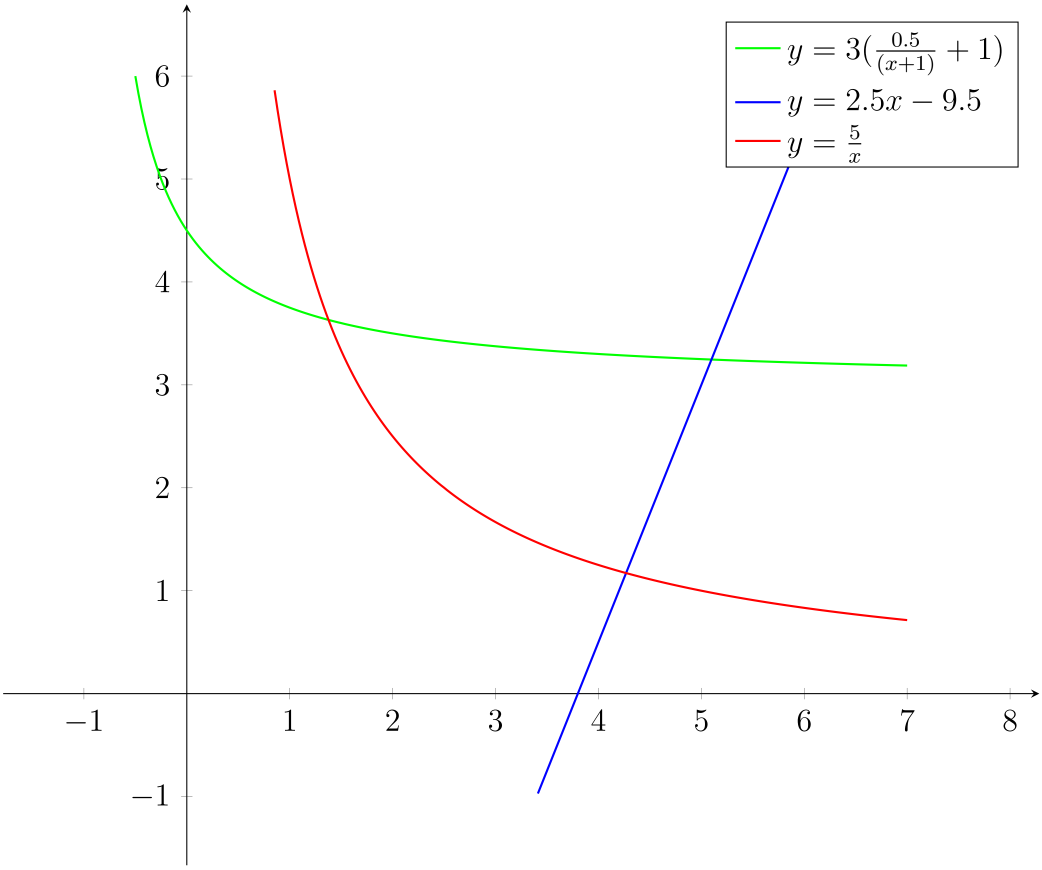

Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Обоснование выбора ε_1 и ε_2.

Известна формула погрешности при подсчете площади методом прямоугольников:. Формулу вычисления интеграла методом прямоугольников можно записать в следующем виде: $\frac{b-a}{n}\sum\limits_{k=1}^{n} f(x_{2k-1}) + R,\ R = \frac{(b-a)^3 f''(\xi)}{24n^2} = \varepsilon_2$. [1].
Сначала заметим, что $(b-a) \geqslant 1$. Скажем сразу, что $\varepsilon_1 < \varepsilon = 0.001$. Тогда с учетом погрешности $\frac{f''(\xi)}{24n^2} = \frac{\varepsilon_2}{(b-a)^3} \leqslant \frac{\varepsilon_2}{0.994} \approx 1.006\varepsilon_2$.
$|\frac{(b-a+2\varepsilon_1)^3 f''(\xi)}{24n^2} - \frac{(b-a)^3 f''(\xi)}{24n^2}| = |\frac{(b-a)^2 \cdot 6\varepsilon_1 f''(\xi)}{24n^2} + \frac{(b-a)\cdot 12\varepsilon_1^2 f''(\xi)}{24n^2} + \frac{8\varepsilon_1^3 f''(\xi)}{24n^2} \leqslant 4 \cdot 6 \cdot 1.006\varepsilon_1\varepsilon_2 + 2 \cdot 12\varepsilon_1^2 \cdot 1.006\varepsilon_2 + 8\varepsilon_1^3 \cdot 1.006\varepsilon_2 < 56.336\varepsilon_1\varepsilon_2$
$|\frac{(b-a+2\varepsilon_1)}{n}\sum\limits_{k=1}^{n} f(x_{2k-1}) - \frac{b-a}{n}\sum\limits_{k=1}^{n} f(x_{2k-1})| = |2\varepsilon_1 f(\xi_1)| < 9\varepsilon_1$
Таким образом, $56.336\varepsilon_1\varepsilon_2 + 9\varepsilon_1 < \varepsilon$. Нетрудно проверить, что $\varepsilon_1 = 0.00001$ и $\varepsilon_2 = 0.00001$ удовлетворяют этому условию.

Результаты экспериментов

Кривые	x	y
1 и 2	5.098387	3.245967
2 и 3	4.268535	1.171338
1 и 3	1.377013	3.631048

Таблица 1: Координаты точек пересечения

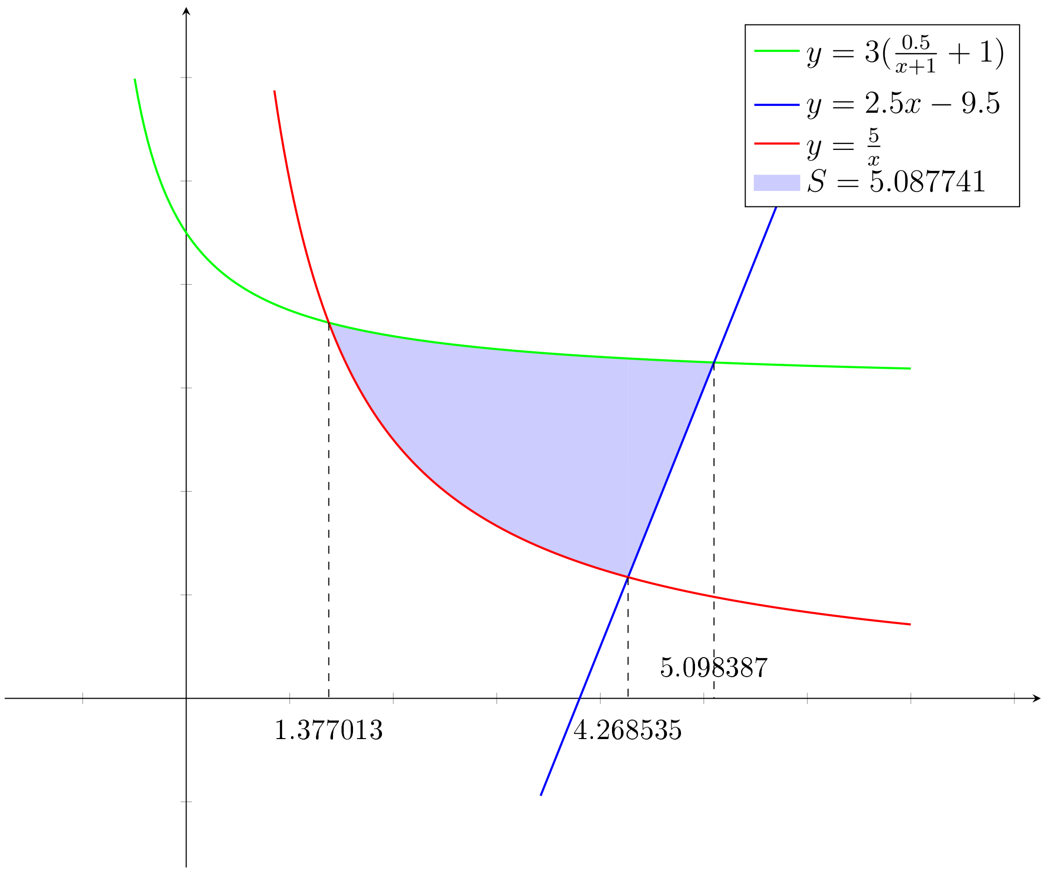

Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

1. Модуль main.c.

   - `void help(void)`

     Ничего не получает на вход, ничего не возващает. Выводит все допустимые ключи командной строки.

   - `root(double (*f)(double), double (*g)(double), double a, double b, double eps1, double (*df)(double), double (*dg)(double))`

     (*f) — указатель на функцию f
     (*g) — указатель на функцию g
     a — левая граница отрезка, на котором функция будет искать корень
     b — правая граница отрезка, на котором функция будет искать корень
     eps1 — ε_1, точность, с которой должно быть получено значение
     (*df) — указатель на функцию df, т.е. на производную функции f
     (*dg) — указатель на функцию dg, т.е. на производную функции g
     Функция возвращает число типа double — корень уравнения f(x)-g(x), найденный методом касательных.

   - `double integral(double (*f)(double), double a, double b, double eps2)`

     (*f) — указатель на функцию f
     a — нижний предел интегрирования
     b — верхний предел интегрирования
     eps2 — ε_2, точность, с которой должно быть получено значение
     Функция возвращает число типа double — $\int\limits_a^b f(x)dx$, вычисленное методом прямоугольников.

2. Модуль Tests.c

   - `void test(int argc, int i, char **argv)`

     argc — количество аргументов командной строки
     i — номер, под которым в командной строке шел ключ test"
     argv — указатель на массив строк в командной строке
     Функция ничего не возвращает. Либо вызывает тестирующую функцию, либо выводит сообщение.

   - `void testroot(void)`

     Ничего не получает на вход, ничего не возвращает. Позволяет тестировать функцию root, находящую корень уравнения методом касательных.

   - `void testintegral(void)`

     Ничего не получает на вход, ничего не возвращает. Позволяет тестировать функцию integral, вычисляющую интеграл методом прямоугольников.

- `int rootit(double (*f)(double), double (*g)(double), double a, double b, double eps1, double (*df)(double), double (*dg)(double))`
  (*f) — указатель на функцию f
  (*g) — указатель на функцию g
  a — левая граница отрезка, на котором функция будет искать корень
  b — правая граница отрезка, на котором функция будет искать корень
  eps1 — ε_1, точность, с которой должно быть получено значение
  (*df) — указатель на функцию df, т.е. на производную функции f
  (*dg) — указатель на функцию dg, т.е. на производную функции g
  Функция возвращает число итераций, потребовавшихся для вычисления корня.

3. Модуль functions.asm

   - `double f1(double);`
     Вычисляет $f_1(x) = 3(\frac{0.5}{x+1} + 1)$
   - `double f2(double);`
     Вычисляет $f_2(x) = 2.5x - 9.5$
   - `double f3(double);`
     Вычисляет $f_3(x) = \frac{5}{x}$
   - `double f4(double);`
     Вычисляет $f_4(x) = 10ln(x)$ (Функция создана для тестирования root и integral)
   - `double f5(double);`
     Вычисляет $f_5(x) = -x^3$ (Функция создана для тестирования root и integral)
   - `double f6(double);`
     Вычисляет $f_6 = -\frac{1}{\sqrt{x+8}}$ (Функция создана для тестирования root и integral)
   - `double df1(double);`
     Вычисляет $f_1'(x)$
   - `double df2(double);`
     Вычисляет $f_2'(x)$
   - `double df3(double);`
     Вычисляет $f_3'(x)$
   - `double df4(double);`
     Вычисляет $f_4'(x)$
   - `double df5(double);`
     Вычисляет $f_5'(x)$
   - `double df6(double);`
     Вычисляет $f_6'(x)$

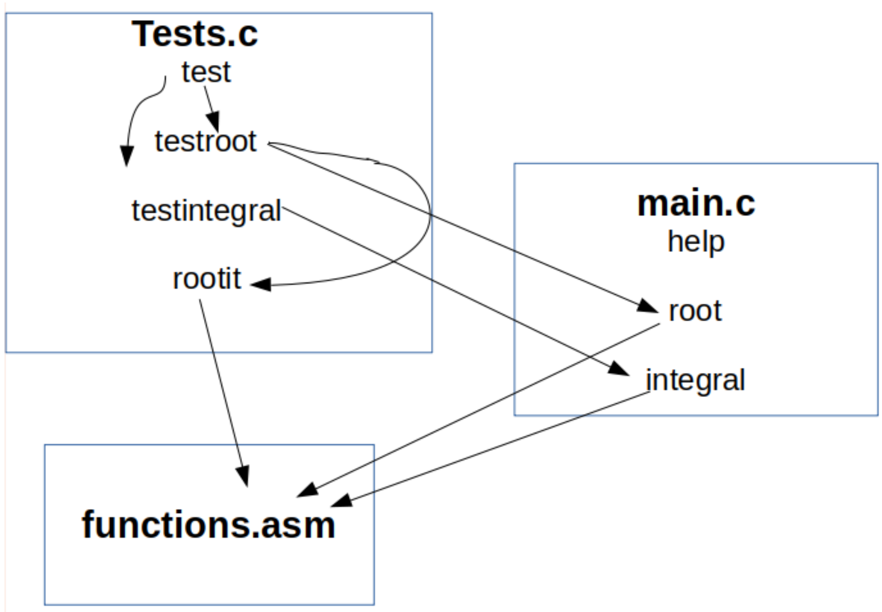

Рис. 3: Схема зависимостей модулей и основных функций

Сборка программы (Make-файл)

Makefile собирает модули в файл program. Сборка осуществляется по ключу all, а удаление промежуточных файлов — по ключу clean.

```
all: program

program: main.o Tests.o functions.o
gcc -m32 -o program main.o Tests.o functions.o

main.o: main.c
gcc -m32  -std=c99 -c main.c

Tests.o: Tests.c
gcc -m32 -std=c99 -c Tests.c

functions.o: functions.asm
nasm -f elf32 functions.asm

clean:
-rm main.o
-rm Tests.o
-rm functions.o
```

Отладка программы, тестирование функций

Тестирование проводилось при помощи ключей -test root и -test integral. Тестирование функции root производилось на следующих уравнениях:

- $10ln(x) + x^3 = 0$
  Выбранный отрезок: [0.5;1.5]
  Результат работы программы: 0.924114

- $10ln(x) + \frac{1}{\sqrt{x+8}} = 0$
  Выбранный отрезок: [0.5;1.5]
  Результат работы программы: 0.967155

- $x^3 - \frac{1}{\sqrt{x+8}} = 0$
  Выбранный отрезок: [0.3;1.5]
  Результат работы программы: 0.697331

Тестирование функции интеграл производилось на следующий функциях:

- $f(x) = 10ln(x)$
  Пределы интегрирования: [1;2]
  Результат работы программы: 3.862967

- $f(x) = -x^3$
  Пределы интегрирования: [4;8]
  Результат работы программы: -960.000023

- $f(x) = -\frac{1}{\sqrt{x+8}}$
  Пределы интегрирования: [-1;0]
  Результат работы программы: -0.365369

Проверка того, что функции действительно выдают верный ответ, производилась с помощью сервисов fooplot.com и wolframalpha.com.

Программа на Си и на Ассемблере

Тексты програм находятся в архиве Program.zip, приложенном к отчету.

Анализ допущенных ошибок

Была допущена ошибка: выбраны слишком маленькие ε_1 и ε_2, в результате чего программа зацикливалась.

Список литературы

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

[2] Трифонов Н.П., Пильщиков В.Н. Задания практикума на ЭВМ (1 курс). Учебное пособие, 2-е исправленное издание. — М.: МГУ, 2001.