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 1 / 4 / 1

Выполнил:
студент 102 группы
Титушин А. Д.

Преподаватель:
Кулагин А. В.

Москва
2022

Содержание

Постановка задачи

Требуется реализовать численный метод, позволяющий с заданной точностью $\varepsilon = 0.001$ вычислить площадь плоской фигуры, ограниченной тремя кривыми $f_1 = 2^x + 1$, $f_2 = x^5$, $f_3 = \frac{1-x}{3}$. В начале необходимо найти точки пересечения кривых комбинированным методом хорд и касательных, предварительно определив отрезки, на которых будут пересечения. Далее требуется реализовать функцию вычисления интеграла при помощи формулы прямоугольников. В результате программа должна печатать площадь найденной области между тремя функциями. Функции нахождения точек пересечения и интеграла должны быть предварительно протестированы.

Математическое обоснование

Выбор отрезков.

Для корректного применения комбинированного метода хорд и касательных для нахождения точек пересечения функций f_1, f_2, f_3 на отрезке [a, b] необходимы и достаточны следующие условия:

1) На концах отрезка функция F(x) = f(x) - g(x) имеет разные знаки, и на всём отрезке производная функции не меняет знак.

2) На данном отрезке первая и вторая производные функции F(x) не меняют свой знак (не обращаются в ноль) [2].

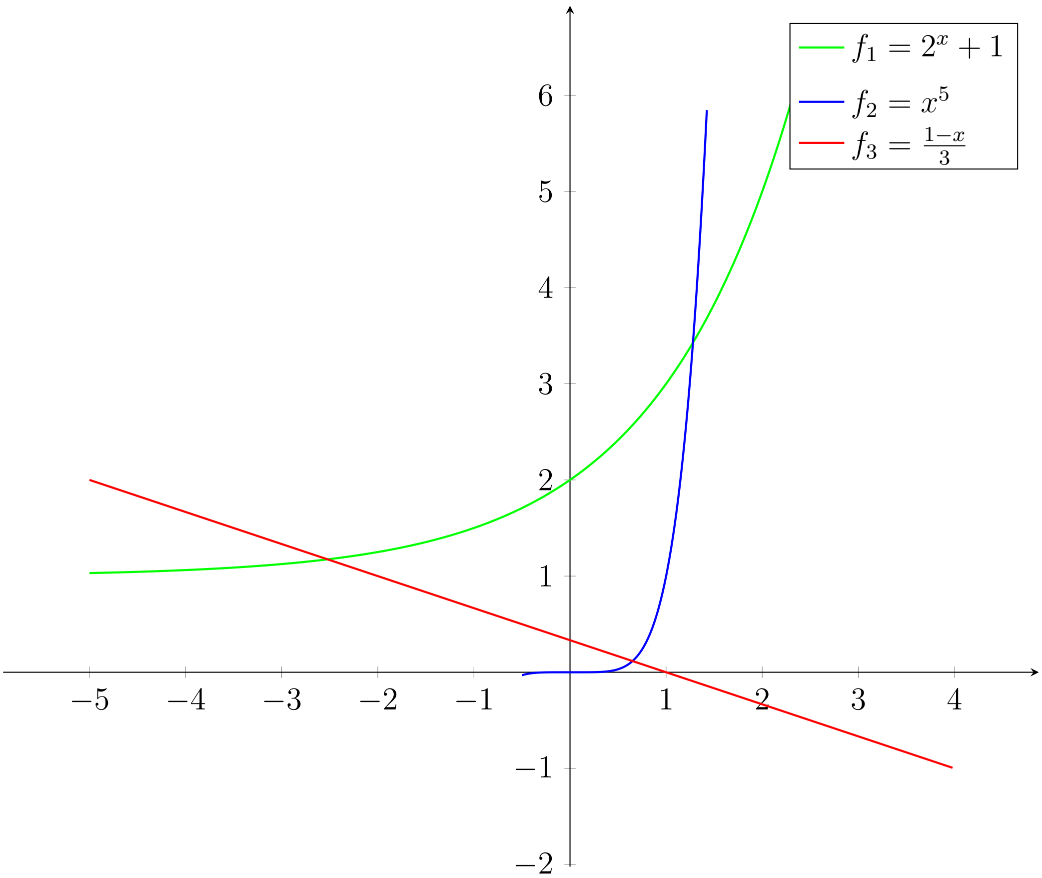

Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

$f_1' = 2^x ln(2); f_2' = 5x^4; f_3' = \frac{-1}{3}; f_1'' = ln(2)^2 2^x; f_2'' = 20x^3; f_3'' = 0$

Рассмотрим f(x) = $f_1(x)$, $g(x) = f_2(x)$ (рис. 1) на отрезке [0,3].
$F(0) = f(0) - g(0) = 1 + 1 - 0 = 2 \Rightarrow F(0) > 0$.
$F(3) = f(3) - g(3) = 8 + 1 - 32 = -23 \Rightarrow F(3) < 0$.
$F'(x) = f'(x) - g'(x) = 2^x * ln(2) - 5x^4 < 0 \ \forall x > 0$
$F''(x) = ln(2)^2 2^x - 20x^3 < 0 \ \forall x > 0$
$\Rightarrow$ отрезок [0, 3] удовлетворяет всем условиям для применения метода хорд и касательных.

Рассмотрим $f(x) = f_2(x), g(x) = f_3(x)$ (рис. 1) на отрезке [0,3].

$F(0) = f(0) - g(0) = 0 - 1/3 \Rightarrow F(0) < 0.$
$F(3) = f(3) - g(3) = 243 + 2/3 \Rightarrow F(3) > 0.$
$F'(x) = f'(x) - g'(x) = 5x^4 + 1/3 > 0\ \forall x > 0$
$F''(x) = 20x^3 > 0\ \forall x > 0$
$\Rightarrow$ отрезок [0, 3] удовлетворяет всем условиям для применения метода хорд и касательных.

Аналогично рассмотрим пару функций $f(x) = f_1(x), g(x) = f_3(x)$. Подходит отрезок [−3, 0].

Интегрирование и выбор точностей вычислений ε_1, ε_2.

Требовалось реализовать интегрирование функции на отрезке [a, b] при помощи формулы прямоугольников:

$$I_n = \int\limits_a^b F(x)dx = h(F_0 + F_1 + ... + F_{n-1});\ F_i = F(a + (i + 0.5)h);\ h = \frac{(b - a)}{n}$$

Для обеспечения требуемой точности ε_1 при приближенном вычислении интеграла F(x) по формуле прямоугольников нужно подобрать соответствующее число n разбиений отрезка интегрирования. Для достижения требуемой точности берется некоторое начальное число разбиений n_0 (в данной программе 20000) и последовательно вычисляются значения I_n при n, равном $2n_0$, $4n_0$, $8n_0$ и т.д. Известно правило Рунге $|I - I_n| \cong \frac{1}{3}|I_n - I_{2n}|$. Согласно этому правилу, когда на очередном шаге величина $\frac{1}{3}|I_n - I_{2n}|$ окажется меньше ε_1, в качестве приближенного значения для I можно взять I_{2n} [2].

Значит, достаточно посчитать интеграл два раза и сравнить $\frac{|I_{2n} - I_n|}{3}$. Итоговая точность должна быть равна 0.001. Рассмотрим, как ε_1 и ε_2 влияют на конечный результат: Итоговый результат вычисляется как $I_1 - I_2 - I_3$, где I_i - это интеграл функции f_i от точки пересечения с f_j до точки пересечения с f_s. Следовательно итоговый результат отличается не более чем на $3\varepsilon_2$. Пусть точки пересечений функци вычисляются с точностью ε_1, тогда интегралы, вычисленные с точностью ε_2, будут еще зависеть и от ε_1. Получаем: $3 * \varepsilon_1^2 + 3 * \varepsilon_2 < 0.001$ $\Rightarrow$ можем взять $\varepsilon_1 = \varepsilon_2 = 0.0001$.

Результаты экспериментов

В результате проведенных вычислений получены координаты точек пересечения кривых f_1, f_2, f_3 (таблица 1) и площадь фигуры S $= \int\limits_{-2.522}^{1.279} f_1(x)dx - \int\limits_{-2.522}^{0.651} f_2(x)dx - \int\limits_{0.651}^{1.279} f_3(x)dx =$ 7.05082 - 2.04711 - 0.716892 $=$ 4.2868 (рис. 2).

Кривые	x	y
1 и 2	1.2793	3.427
2 и 3	0.651	0.116
1 и 3	-2.522	1.174

Таблица 1: Координаты точек пересечения

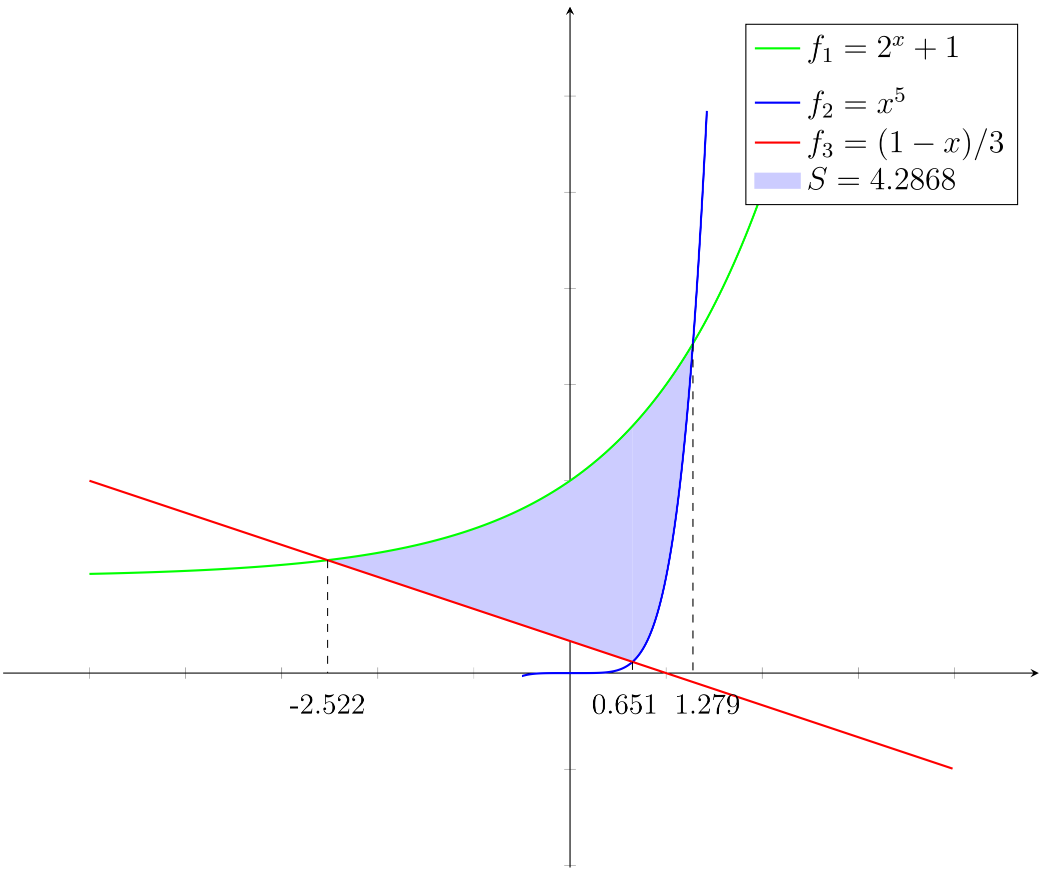

Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Программа состоит из 1 модуля на языке C и 1 модуля на Ассемблере NASM.

1. Модуль main.c.

1) `int main(int argc, char* argv[])` - принимает на вход аргументы `int argc, char* argv[]` из командной строки и обрабатывает их на совпадение с ключами (-help, -iterations, -testintegral, -testroot, -root, -functions).

2) `void help()` - функция, вызываемая ключом -help из командной строки, печатает на стандартный поток вывода список ключей для работы с программой.

3) `void iterations()` - функция, вызываемая ключом -iterations из командной строки, печатает на стандартный поток вывода координаты точек пересечения функций и количество итераций вычисления.

4) `void abscissas()` - функция, вызываемая ключом -root из командной строки, печатает на стандартный поток вывода абсциссы точек пересечения функций.

5) `void functions()` - функция, вызываемая ключом -functions из командной строки, печатает на стандартный поток вывода функции f_1, f_2, f_3.

6) `double integral(double (*f)(double), double a, double b, double eps)` - функция подсчета интеграла при помощи формулы прямоугольников. Принимает на вход функцию f, левую и правую границы отрезка, на котором вычисляется интеграл, и требуемую точность вычислений.

7) `double root(double (*f)(double), double (*g)(double), double (*df) (double), double (*dg)(double), double a, double b, double eps)` - функция вычисления абсцисс пересечения кривых при помощи комбинированного метода хорд и касательных; принимает на вход две функции, левую и правую границу отрезка, на котором существует точка пересечения, соответствующие производные для функций и требуемую точность вычислений.

8) `chord_method(double (*f)(double), double (*g)(double), double a, double b, double *A, double *B)` - функция реализующая метод хорд для поиска точки пересечения функций; принимает на вход две функции, левую и правую границу отрезка, приближаемые к точке пересечения границы, которые изменяются в ходе поисков корня.

9) `double tangent_method(double (*f)(double), double (*g)(double), double (*df) (double), double (*dg)(double), double a, double b, double *A, double *B)` - функция реализующая метод касательных для поиска точки пересечения функций; принимает на вход две функции, соотвествующие им производные, левую и правую границу отрезка, приближаемые к точке пересечения границы, которые изменяются в ходе поисков корня.

10) `int case_choice(double (*f)(double), double (*g)(double), double a, double b)` - функция, выбирающая случай сближения слева/справа для метода хорд и касательных; принимает на вход две функции, левую и правую границу отрезка.

11) `void test_integral(char* function, char* x, char* y, char* eps_c)`

- функция, вызываемая ключом -testintegral из командной строки, печатает на стандартный поток вывода значение интеграла от введенной функции на заданном пользователем отрезке. Принимает на вход функцию $f_1/f_2/f_3$, границы отрезка интегрирования и точность вычислений.

12) `void test_root(char* function_1, char* function_2, char* x, char* y, char* eps_c)` - функция, вызываемая ключом -testroot из командной строки, печатает на стандартный поток вывода абсциссу точки пересечения введенных функций на заданном пользователем отрезке. Принимает на вход две функции $f_1/f_2/f_3$, границы отрезка для нахождения корня и точность вычислений.

2. Модуль functions.asm.

1) `f_1` - функция, принимающая на вход координату double x, вычисляет в ней значение функции $f_1(x) = 2^x + 1$.

2) `f_2` - функция, принимающая на вход координату double x, вычисляет в ней значение функции $f_2(x) = x^5$.

3) `f_3` - функция, принимающая на вход координату double x, вычисляет в ней значение функции $f_3(x) = (1 - x)/3$.

4) `df_1` - функция, принимающая на вход координату double x, вычисляет в ней значение производной функции $f_1'(x) = ln(2)2^x$.

5) `df_2` - функция, принимающая на вход координату double x, вычисляет в ней значение производной функции $f_2'(x) = 5x^4$.

6) `df_3` - функция, принимающая на вход координату double x, вычисляет в ней значение производной функции $f_3'(x) = -1/3$.

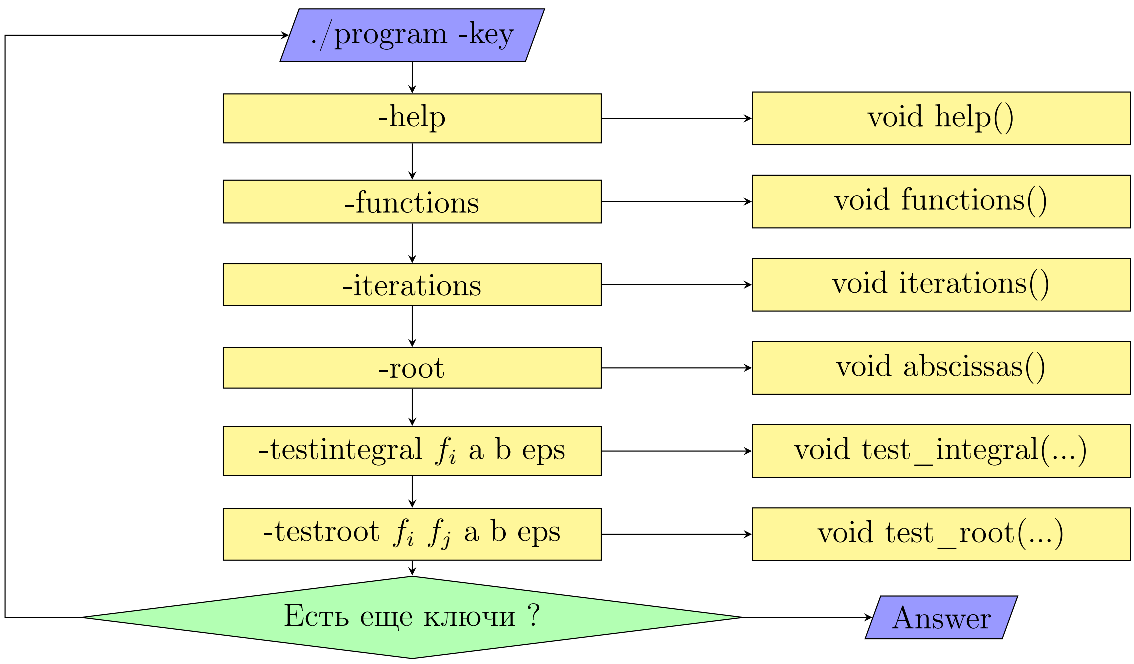

Рис. 3: Работа программы

Сборка программы (Make-файл)

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

Текст Makefile:

```
all: main.o functions.o
   @gcc -m32 -o program main.o functions.o -lm
main.o: main.c
   @gcc -m32 -std=c99 -c -o main.o main.c
functions.o: functions.asm
   @nasm -f elf32 -o functions.o functions.asm
clean:
   rm -rf *.o
```

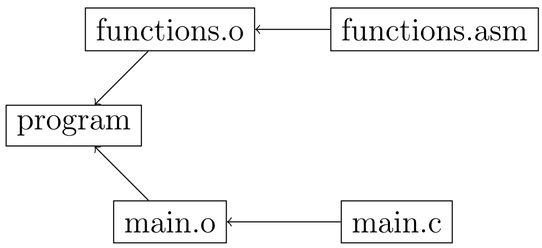

Рис. 4: Зависимость между модулями

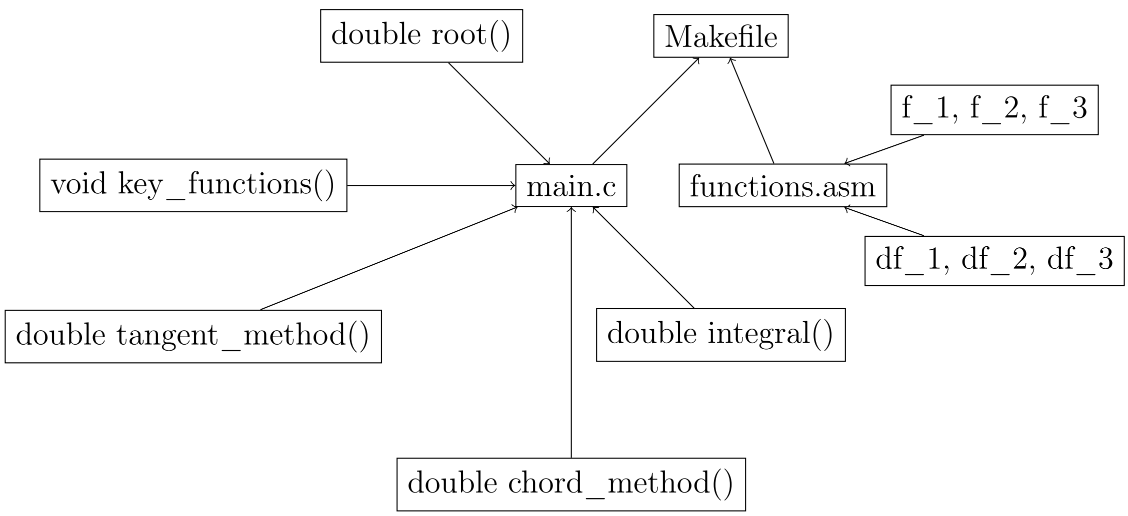

Рис. 5: Работа программы

Отладка программы, тестирование функций

Программа была протестирована, проведена проверка корректности работы функций f_1, f_2, f_3, df_1, df_2, df_3, а также функций нахождения корня и интеграла. Результаты тестирования представлены в таблицах 2, 3, 4.

Функции	Результат работы функции root	Точное значение
f_1, f_2	1.279347	1.27935
f_2, f_3	0.650503	0.65052
f_1, f_3	-2.522223	-2.52222

Таблица 2: Тестирование функции root

Функция	Отрезок	Результат функции integral	Точное значение
$f_1 = 2^x + 1$	$[-10.98, 15.67]$	75243.400407	75243.40045
$f_2 = x^5$	$[0.5, 8]$	43690.664033	43690.66406
$f_3 = \frac{(1-x)}{3}$	$[-8, 5]$	10.83333	10.83333

Таблица 3: Тестирование функции integral

Функция	координата x	Результат работы функций	Точное значение
f_1	5	33	33
f_1	-4.67	1.03928	1.03928
df_1	5	22.18071	22.18071
df_1	-4.67	0.027228	0.027228
f_2	2.89	201,59939	201,59939
f_2	-9.8	-90392.07968	-90392.07968
df_2	2.89	348,78787	348,78787
df_2	-9.8	46118,408	46118,408
f_3	0.05	0.316666	0.316
f_3	-98	33	33
df_3	0.05	-0.3333	-0.3333
df_3	-98	-0.3333	-0.3333

Таблица 4: Тестирование функций f_1, f_2, f_3, df_1, df_2, df_3

Для тестирования программы предусмотрены ключи командной строки:

1) Для тестирования функции нахождения точек пересечения -testroot <первая функция> <вторая функция> <левая граница отрезка> <правая граница отрезка> <точность вычислений>.

2) Для тестирования функции нахождения интеграла -testintegral <функция> <левая граница интегрирования> <правая граница интегрирования> <точность вычислений>.

Программа на Си и на Ассемблере

Тексты программ находятся в архиве titushin_report.zip, приложенном к отчёту.

Анализ допущенных ошибок

В процессе выполнения работы ошибок допущено не было.

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. — Москва: Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В.Ломоносова, 2001.