

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 / 2

Выполнил:
студент 106 группы
Федоренко И. А.

Преподаватели:
Корухова Л. С.
Манушин Д. В.

Москва
2020

Содержание

Постановка задачи	2
Математическое обоснование	3
Результаты экспериментов	4
Структура программы и спецификация функций	5
Сборка программы (Make-файл)	6
Отладка программы, тестирование функций	7
Программа на Си и на Ассемблере	9
Анализ допущенных ошибок	10
Библиографический список	11

Постановка задачи

Требуется с заданной точностью $\varepsilon = 0.001$ вычислить площадь плоской фигуры, ограниченной тремя кривыми, используя комбинированный метод приближенного решения уравнений для поиска пересечений кривых, и квадратурную формулу трапеций для приближенного вычисления определенных интегралов. Уравнения кривых $y = f_1(x)$, $y = f_2(x)$ и $y = f_3(x)$, где:

- $f_1 = 2^x + 1$.
- $f_2 = x^5$.
- $f_3 = (1 - x)/3$.

При решении задачи необходимо:

- Функции, вычисляющие значения f_1 , f_2 , f_3 и их производных реализовать на языке ассемблера с соглашением вызова cdecl. Остальные функции программы реализовать на языке Си.
- Реализовать поддержку функций командной строки при задании которых: печатаются абсциссы точек пересечения кривых; печатается число итераций, потребовавшихся на приближенное решение уравнений при поиске точек пересечения.
- Реализовать поддержку ключа командной строки -help, при использовании которого программа должна выводить все допустимые ключи командной строки.
- Реализовать функцию *root*($f, f_1, g, g_1, a, b, eps1$), вычисляющую с точностью ε_1 корня x уравнения $f(x) = g(x)$ на отрезке $[a, b]$ (функции f_1 и g_1 - производные функций f и g соответственно).
- Реализовать функцию *integral*($f, a, b, eps2$), вычисляющую с точностью ε_2 величину определенного интеграла от функции $f(x)$ на отрезке $[a, b]$
- Протестировать функции *root* и *integral*. Предоставить возможности тестирования, активируемые опциями командной строки.
- Осуществить сборку программы с помощью утилиты *make*. В соответствующем файле определить цели *all* и *clean* первая из которых полностью собирает программу, а вторая — удаляет все промежуточные файлы (в частности, объектные модули).

Математическое обоснование

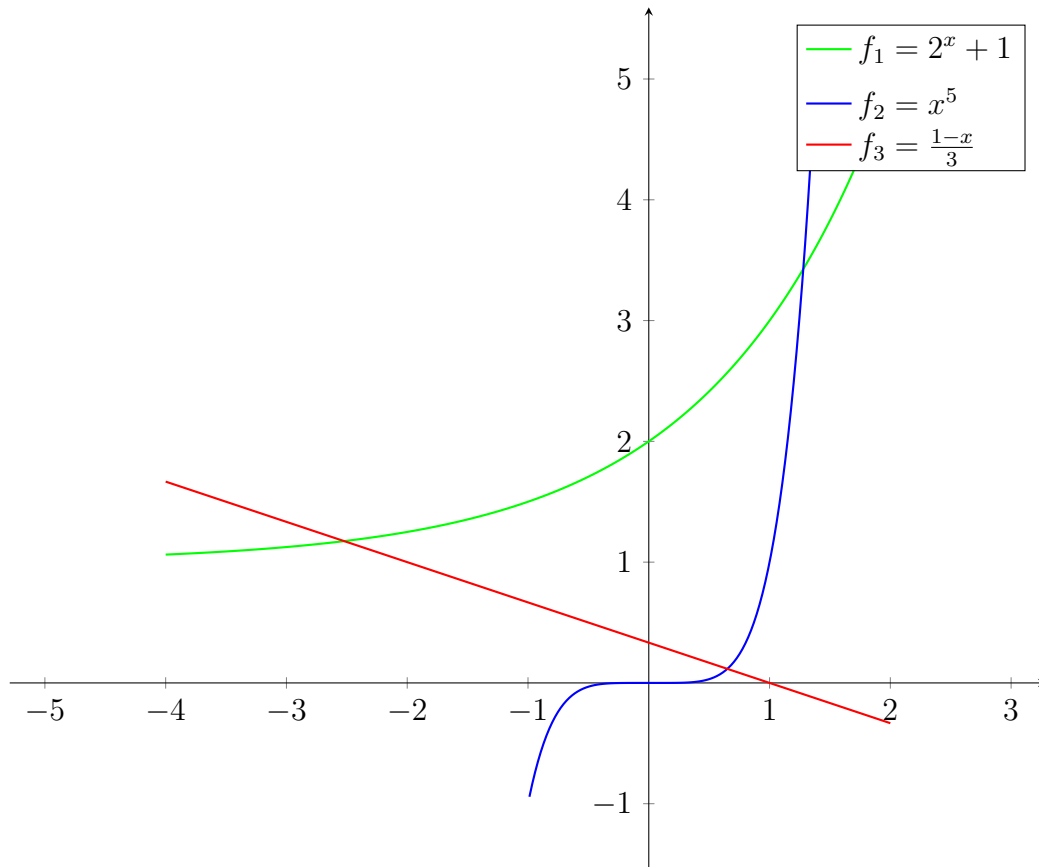


Рис. 1: График функций

При вычислении точек пересечений кривых используется комбинированный метод приближенного решения уравнений, включающий в себя метод хорд и метод касательных [1]. Уменьшение отрезка поиска происходило до тех пор, пока его длина была больше, чем ε_1 , что обеспечило требуемую точность вычислений.

Введем обозначения: x_1 – абсцисса точки пересечения f_1 и f_3 , x_2 – абсцисса точки пересечения f_2 и f_3 , x_3 – абсцисса точки пересечения f_1 и f_2 . Точка x_1 вычислялась на отрезке $[-3, -2]$, x_2 – на отрезке $[0.4, 0.8]$, x_3 – на отрезке $[1.1, 1.3]$. Отрезки выбирались таким образом, чтобы соответствующая функция $f_i - f_j$ была непрерывна дифференцируема на выбранном отрезке, а так же, чтобы точки на концах отрезка находились по разные стороны от оси абсцисс (т. е. чтобы выполнялось неравенство $(f_i(a) - f_j(a))(f_i(b) - f_j(b)) < 0$).

Вычисление x_1 , x_2 и x_3 производились с точностью $\varepsilon_1 = \frac{\varepsilon}{10}$.

Далее происходило вычисление площадей фигур при помощи квадратурной формулы трапеций. Считался определенный интеграл от f_1 на отрезке $[x_1, x_3]$ (площадь S_1), от f_3 на отрезке $[x_1, x_2]$ (площадь S_2) и от f_2 на отрезке $[x_2, x_3]$ (площадь S_3). Тогда площадь искомой фигуры $S = S_1 - S_2 - S_3$.

Сумма считалась до тех пор, пока $\frac{1}{3}|I_{2n} - I_n| > \varepsilon_2$ (правило Рунге [2] гарантирует требуемую точность). Интегралы вычислялись с точностью $\varepsilon_2 = \frac{\varepsilon}{10}$.

Результаты экспериментов

Координаты точек пересечения кривых, вычисленные с точностью $\varepsilon_1 = 0.0001$, представлены в таблице 1.

Кривые	x	y
1 и 3	-2.5222	1.1741
2 и 3	0.6505	0.1165
1 и 2	1.2794	3.4273

Таблица 1: Координаты точек пересечения

Так, искомая площадь равна $S = 4.2868$ (рис. 2). Так как интегралы вычислялись с точностью ε_2 , то погрешность вычисления площади не превосходит $3(\varepsilon_2 + 2\varepsilon_1) < \varepsilon$.

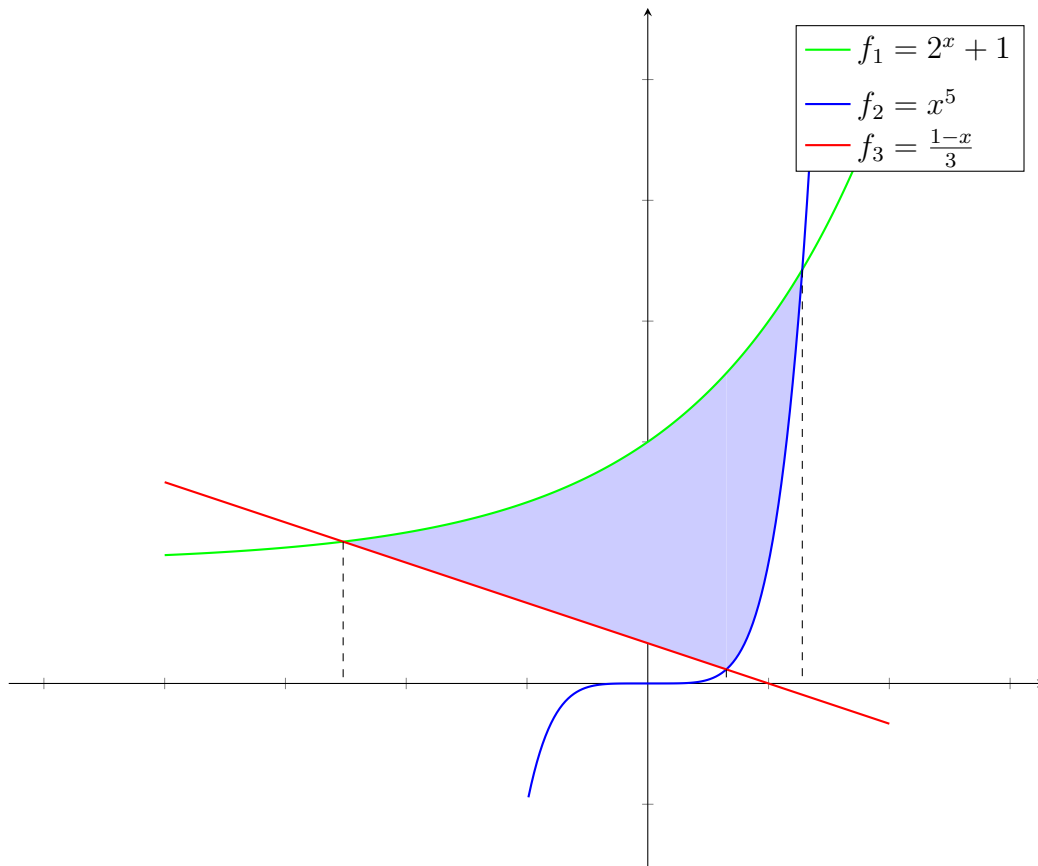


Рис. 2: Фигура, ограниченная заданными кривыми

Структура программы и спецификация функций

Программа состоит из 10 модулей:

- *main.c* с функцией *int main(int argc, char *argv[])* обрабатывающей опции командной строки, заданные пользователем.
- *header.h* – заголовочный файл, подключающий требуемые библиотеки и объявляющий все функции.
- *root.c* – файл с функцией *double root(double(*f)(double x), double(*f1)(double x), double(*g)(double x), double(*g1)(double x), double a, double b, double eps1)*, считающей абсциссу пересечения функций *f* и *g*, и принимающей в качестве аргументов функции *f*, *g*, функции, задающие их производные – *f₁*, *g₁*, границы отрезка поиска *a*, *b*, требуемую точность *eps1*.
- *integral.c* – файл с функцией *double integral(double(*F)(double x), double a, double b, double eps2)*, считающей определенный интеграл от функции *F* на отрезке $[a, b]$ с точностью *eps2*.
- *functions.c* – файл с реализованными тестовыми функциями и их производными.
- *asmfunction.asm* – файл с реализованными функциями *f₁*, *f₂* и *f₃* и их производными.
- *testroot.c* – файл, в котором тестируются функция *root*.
- *testintegral.c* – файл, в котором тестируются функция *integral*.
- *solve.c* – файл с функцией *void solve(int *params)*, решающей задачу. Функция принимает аргумент *params* отвечающий за вывод дополнительной информации о решении.
- *Makefile* – файл для сборки программы, в котором прописаны все зависимости.

Сборка программы (Make-файл)

Сборка программы осуществляется с помощью утилиты *make*. В *Makefile* прописаны все зависимости и определены цели *all* и *clean*. При сборке программы будут компилироваться только те файлы, которые были изменены.

```
1 all: main
2
3 clean:
4     rm -rf main *.o
5
6 main: main.o functions.o root.o integral.o solve.o testroot.o testintegral.o asmfunctions.o
7     gcc -o main main.o functions.o root.o integral.o solve.o testroot.o testintegral.o asmfunctions.o
8
9 main.o: main.c
10    gcc -m32 -c -o main.o main.c
11
12 functions.o: functions.c
13    gcc -m32 -c -o functions.o functions.c
14
15 asmfunctions.o: asmfunctions.asm
16    nasm -f elf32 -o asmfunctions.o asmfunctions.asm
17
18 root.o: root.c
19    gcc -m32 -c -o root.o root.c
20
21 integral.o: integral.c
22    gcc -m32 -c -o integral.o integral.c
23
24 solve.o: solve.c
25    gcc -m32 -c -o solve.o solve.c
26
27 testroot.o: testroot.c
28    gcc -m32 -c -o testroot.o testroot.c
29
30 testintegral.o: testintegral.c
31    gcc -m32 -c -o testintegral.o testintegral.c
32
33 main.o functions.o root.o integral.o solve.o testroot.o testintegral.o: header.h
34
```

Отладка программы, тестирование функций

При написании программы проводилось тестирование каждой написанной функции. При неправильной работе программы отладка происходила с помощью последовательного комментирования строк и вывода результата работы программы на экран. Для тестирования функций *root* и *integral* выделены специальные модули программы, отвечающие за это.

Тестовые функции для *root* (погрешность $\varepsilon_1 = 0.0001$):

- $\frac{1}{2}x^4 + x^3$
Отрезок: $[-2.5, -1.75]$
Правильный ответ: -2.0
Ответ программы: -2.0
- $2\sin x - 1$
Отрезок: $[0.2, 0.8]$
Правильный ответ: $\frac{\pi}{6}$
Ответ программы: 0.5236
- $-e^x + 1$
Отрезок: $[-0.5, 0.5]$
Правильный ответ: 0.0
Ответ программы: 0.0

Тестовые функции для *integral* (погрешность $\varepsilon_2 = 0.0001$):

- $\frac{1}{2}x^4 + x^3$
Отрезок: $[0, 1]$
Правильный ответ: 0.35
Ответ программы: 0.35
- $2\sin x - 1$
Отрезок: $[1, 2]$
Правильный ответ: $2(\cos 1 - \cos 2) - 1$
Ответ программы: 0.9129
- $-e^x + 1$
Отрезок: $[1, 2]$
Правильный ответ: $-e^2 + e + 1$
Ответ программы: -3.6708

- x^5

Отрезок: $[0.5, 1]$

Правильный ответ: $\frac{63}{384}$

Ответ программы: 1.6406

В программе, приложенной в архиве, представлены так же результаты тестирования дополнительных функций, искомые значения которых не вычислимы аналитически. Для получения этих значений были использованы такие ресурсы как *wolframalpha* [3] и *desmos* [4].

В программе так же представлены результаты тестирования функций с различными значениями ε_1 и ε_1 .

Программа на Си и на Ассемблере

Исходные тексты программы содержатся в приложенном архиве

Анализ допущенных ошибок

Во время работы возникли ошибки, были предприняты решения по их исправлению:

- Удвоение ключа приводило к удвоению вывода, перемещение позиций ключей меняло порядок вывода. Было переделано в соответствующий вид.
- Неправильная работа программы при некоторых значениях опциональных аргументов ключей. Исправлено.
- Неправильное форматирование ключа -help. Было исправлено так, чтобы соответствовало требуемому оформлению.
- Неправильная работа при вводе опциональных флагов ключей. Исправлено.
- Обработка опциональных параметров при неправильных числах происходила неправильно. Было исправлено с помощью функции atoi.
- Отсутствие некоторых сообщений об ошибках. Были добавлены требуемые сообщения.

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). М.: МГУ, 2001.
- [3] <https://www.wolframalpha.com/>
- [4] <https://www.desmos.com/calculator>
- [5] Падарян В. А., Соловьев М. А. Лекции по курсу "Архитектура ЭВМ и язык ассемблера" <http://asmcourse.cs.msu.ru/>