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 9 / метод хорд / формула прямоугольников

Выполнил:
студент 106 группы
Токарь Дмитрий Сергеевич

Преподаватель:
Корухова Людмила Сергеевна

Москва
2022

Содержание

Постановка задачи

Требуется реализовать численный метод, позволяющий вычислить, с определенной точностью, площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых:

- $f_1 = \frac{3}{(x-1)^2+1}$
- $f_2 = \sqrt{x + 0.5}$
- $f_3 = e^{-x}$

Точность нахождения площади: $\varepsilon = 0.001$

Площадь фигуры представляется в виде алгебраической суммы определенных интегралов. Вычисление определенного интеграла заданных функций, на отрезках между точками их пересечения, производится с помощью метода прямоугольников с некоторой точностью ε_2. Точки пересечения вычисляются с помощью метода хорд или метода касательных (метод определяется на этапе препроцессирования) для нахождения приближенных корней уравнения $F(x) = 0$, с некоторой точностью ε_1. Отрезок, на котором применяется метод вычисления корней, вычисляется аналитически, с учетом значений первой и второй производной функции. Значения ε_1 и ε_2 вычисляются, также аналитически, таким образом, чтобы гарантировалось вычисление площади фигуры с точностью ε. Интерфейс программы должен представлять возможность тестирования данных методов.

Математическое обоснование

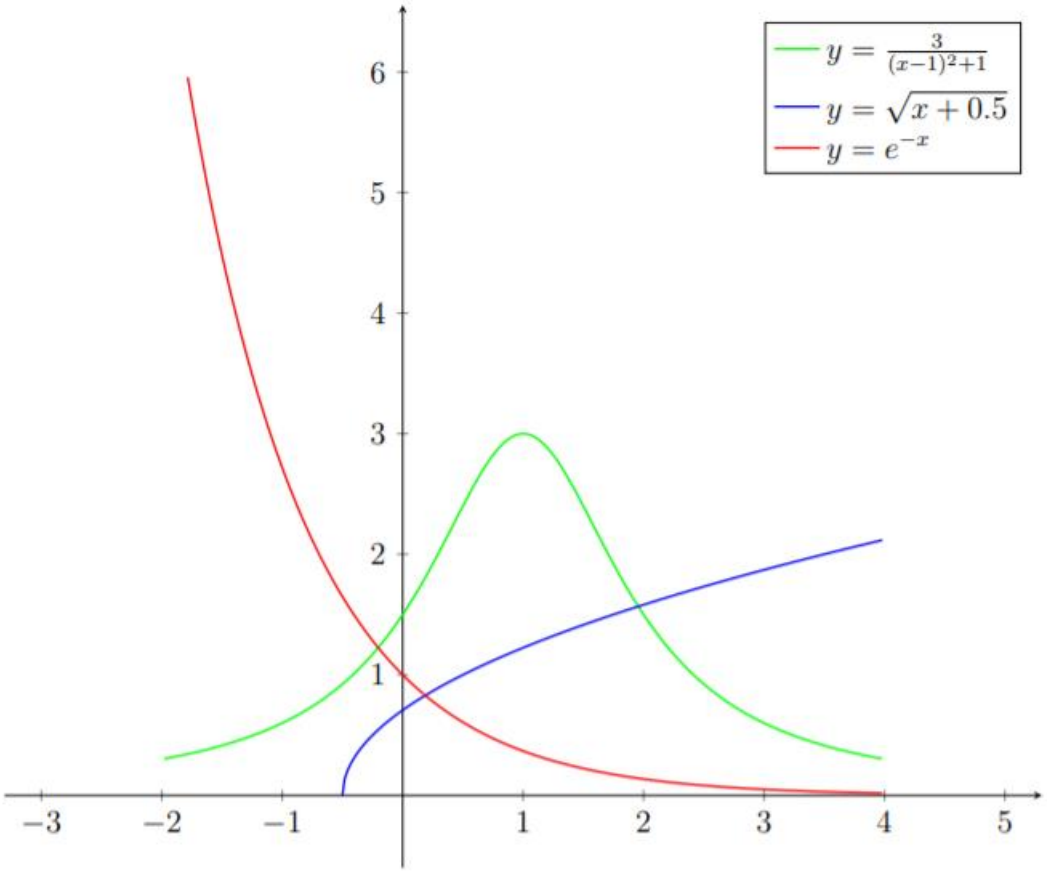


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Для выполнения задания необходимо провести подготовительную работу, и провести аналитически часть вычислений.

1. **Выбор отрезков для вычисления абсцисс точек пересечения.**

   Метод хорд состоит в разбиении отрезка [a, b] на два отрезка с помощью хорды и выборе такого нового отрезка от точки пересечения хорды с осью абсцисс до неподвижной точки, что на нем функция меняет знак и содержит решение, причём подвижная точка приближается к ε-окрестности решения.

   Метод касательных состоит в проведении касательной в одном из концов отрезка [a, b], и получения нового отрезка от точки пересечения касательной с осью абсцисс до неподвижной точки, при этом подвижная точка приближается к ε-окрестности решения.

   Для обоих методов берём a и b такие, что $F(a) \cdot F(b) < 0$, а на данном отрезке функция имеет монотонную и непрерывную производную, сохраняющую определенный знак [1]. Метод хорд применим для решения уравнения вида $F(x) = 0$ на отрезке [a, b], если ни одна точка отрезка [a, b] не является ни стационарной, ни критической, то есть $f'(x) \neq 0$ и $f''(x) \neq 0$.

Все заданные условием функции непрерывны на области определения, следовательно и $F(x)$ непрерывна. Тогда, найдем подходящие отрезки для функции поиска абсцисс точек пересечения:

- Для функций f_1 и f_2 это отрезок [1.7; 2.2]
- Для функций f_2 и f_3 это отрезок [0; 0.5]
- Для функций f_1 и f_3 это отрезок [-0.26; 0.32]

2. Интегрирование и выбор ε_1 и ε_2.

Интеграл вычислялся с помощью применения формулы прямоугольников:

$$\int_a^b f(x)dx = \frac{b-a}{n}\sum_{k=0}^{n-1} f\left(\frac{b-a}{2n} + k\frac{b-a}{n}\right) + R$$

$$R = (\mathrm{b} - \mathrm{a})^3 \frac{\mathrm{f}''(\xi)}{24} \quad [1]$$

$$\varepsilon_2 = R = (\mathrm{b} - \mathrm{a})^3 \frac{\mathrm{f}''(\xi)}{24}$$

$$\frac{\mathrm{f}''(\xi)}{24} = \frac{\varepsilon_2}{(\mathrm{b} - \mathrm{a})^3} < \frac{\varepsilon_2}{(0.39)^3} = 16.858\varepsilon_2$$

Выразим погрешность с учетом ε_1:

$$\left|(\mathrm{b} - \mathrm{a} + 2\varepsilon_1)^3 \frac{\mathrm{f}''(\xi)}{24} - (\mathrm{b} - \mathrm{a})^3 \frac{\mathrm{f}''(\xi)}{24}\right| =$$

$$= \left| 6(\mathrm{b} - \mathrm{a})^2\varepsilon_1 \frac{\mathrm{f}''(\xi)}{24} + 12(\mathrm{b}-\mathrm{a})\varepsilon_1{}^2 \frac{\mathrm{f}''(\xi)}{24} + 8\varepsilon_1{}^3 \frac{\mathrm{f}''(\xi)}{24}\right| =$$

$$= |\, 6(0.39)^2\varepsilon_1 16.858\varepsilon_2 + 12(0.39)\varepsilon_1{}^2 16.858\varepsilon_2 + 8\varepsilon_1{}^3 16.858\varepsilon_2| =$$

$$= \left| 15.3846\varepsilon_1\varepsilon_2 + 78.8954\varepsilon_1{}^2\varepsilon_2 + 134.864\varepsilon_1{}^3\varepsilon_2\right| < 229.1436\varepsilon_1\varepsilon_2$$

Так как интеграл вычисляется три раза для вычисления общей площади, общую погрешность считаем равной $687.4308\varepsilon_1\varepsilon_2 < 0.001$. Подставив значения $\varepsilon_1 = \varepsilon_2 = 0.0001$, увидим, что они удовлетворяют неравенству.

Результаты экспериментов

Результаты вычислений:

Кривые	x	y
1 и 2	1.9561	1.5672
2 и 3	0.1874	0.8291
1 и 3	-0.2033	1.2254

Таблица 1: Координаты точек пересечения

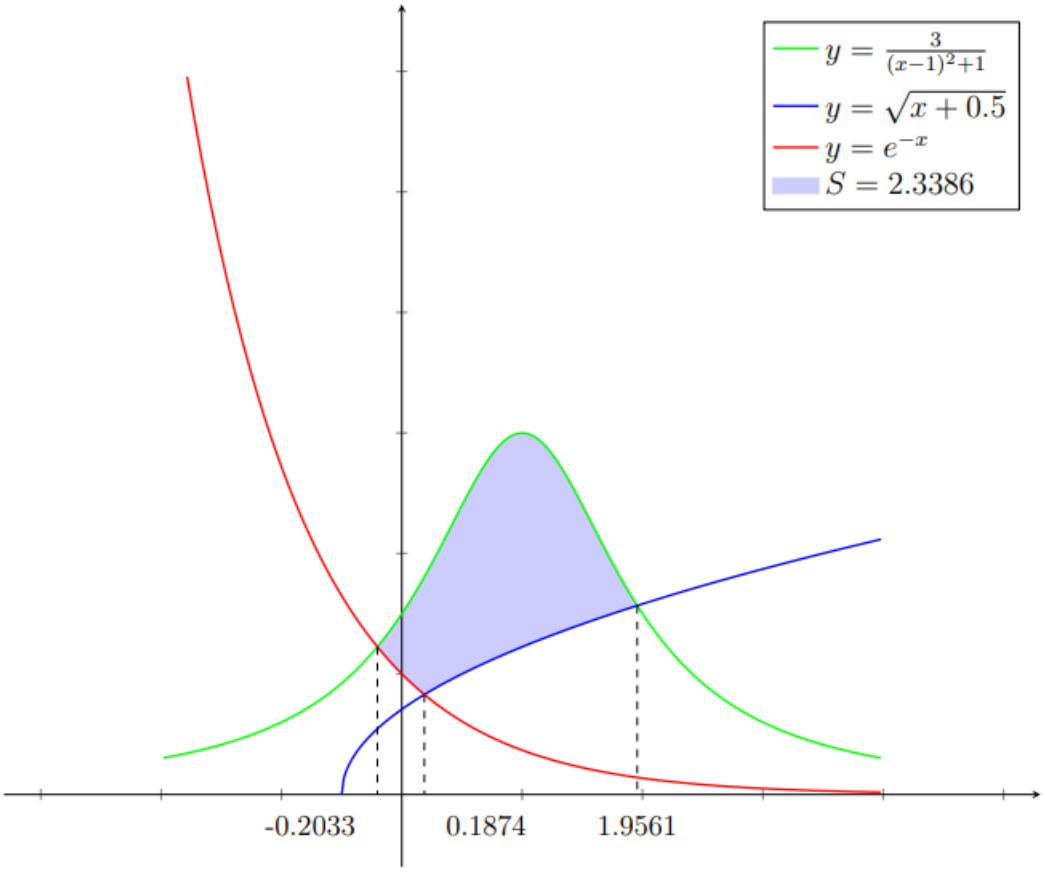


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Программа состоит из 5 модулей на языке Си, модуля на Ассемблере и заголовочного файла.

1. Модуль **main.c**:

   В модуле подключен заголовочный файл **"trsqr.h"**

   - `int main(int argc, char *argv[])`

     Основная функция, с которой начинается выполнение программы и которая обеспечивает взаимодействие всех модулей. Вычисляет абсциссы точек пересечения и число итераций для их нахождения, находит значения интегралов и площадь фигуры. Обеспечивает взаимодействие с пользователем и обрабатывает стандартные опции командной строки, принимает на вход из командной строки аргументы `int argc, char *argv[]`. Также предоставляет возможности тестирования функций вычисления корней и интегралов.

     Список ключей командной строки, обрабатываемых функцией:

     -help - Вывод всех доступных команд

     -points - Координаты точек пересечения

     -sqr - Площадь плоской фигуры

     -count - Число итераций для вычисления точек пересечения

     -testr - Тестирование функции root

     -testi - Тестирование функции integral

   - `double f1(double x)`

     Одна из функций для тестирования. Вычисляет значение функции $f(x) = x^2$.

   - `double f2(double x)`

     Одна из функций для тестирования. Вычисляет значение функции $f(x) = \sqrt{2x}$.

   - `double f3(double x)`

     Одна из функций для тестирования. Вычисляет значение функции $f(x) = \frac{8}{x}$.

- `double der_f1 (double x)`

Одна из функций для тестирования. Вычисляет значение производной функции $f(x) = x^2$.

- `double der_f2 (double x)`

Одна из функций для тестирования. Вычисляет значение производной функции $f(x) = \sqrt{2x}$.

- `double der_f3 (double x)`

Одна из функций для тестирования. Вычисляет значение производной функции $f(x) = \frac{8}{x}$.

2. Модуль **root.c**:

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

Функция, реализующая один из методов вычисления корня уравнения $f_i(x) = f_j(x)$. Принимает указатели на функции, вычисляющие значения $f_i(x)$ и $f_j(x)$, указатели на функции, вычисляющие значения их производных, указатель на счетчик итераций, границы отрезка a и b, на которых будет производиться поиск корня уравнения, точность вычисления корня *ε1*. При сборке, на этапе препроцессирования, выбирается метод решения (метод хорд или метод касательных) передачей символа R=0 (метод касательных) или R=1 (метод хорд) через ключ -D.

3. Модуль **`root.c`**:

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

Функция вычисляет интеграл функции $f(x)$ методом прямоугольников на отрезке $[a, b]$ с точностью ε. Принимает указатель на функцию, вычисляющую значение $f(x)$, границы отрезка интегрирования a и b, точность вычисления *ε2*.

4. Модуль **`testr.c`**:

В модуле подключен заголовочный файл **`"trsqr.h"`**

- `void testr (void)`

Функция не принимает никаких входных данных и ничего не

возвращает. С помощью нее можно произвести тестирование функции root на предложенных функциях. Пользователю предлагается выбрать две функции из трех имеющихся, задать участок интегрирования и точность вычисления. Также функция поддерживает команды **-continue** (продолжить тестирование) и **-end** (закончить тестирование). Вызвать данную функцию можно использовав опции командной строки **-testr**.

5. Модуль **testi.c**:

В модуле подключен заголовочный файл **"trsqr.h"**

- `void` **testi** `(void)`

Функция не принимает никаких входных данных и ничего не возвращает. С помощью нее можно произвести тестирование функции **integral** на предложенных функциях. Пользователю предлагается выбрать одну из трех имеющихся функций, и задать точность вычисления. Также функция поддерживает команды **-continue** (продолжить тестирование) и **-end** (закончить тестирование). Вызвать данную функцию можно использовав опции командной строки **-testi**.

6. Модуль **asm_func.asm**:

Модуль содержит в себе функции для вычисления значений трех заданных функций, а также их производных:

- `double` **func1**`(double x)`

$$f_1(x) = \frac{3}{(x - 1)^2 + 1}$$

- `double` **func2**`(double x)`

$$f_2(x) = \sqrt{x + 0.5}$$

- `double` **func3**`(double x)`

$$f_3(x) = e^{-x}$$

- `double` **der_func1**`(double x)`

$$f_1'(x) = \frac{3}{((x - 1)^2 + 1)}$$

- `double der_func2(double x)`

$$f_2'(x) = \frac{1}{2\sqrt{x + 0.5}}$$

- `double der_func3(double x)`

$$f_3'(x) = -e^{-x}$$

7. Модуль **`trsqr.h`**:

   Заголовочный файл содержит в себе прототипы всех функций из модуля **`asm_func.asm`**, функций **`root`** и **`integral`**, функций **`testr`** и **`testi`** а также тестовых функций, подключает библиотеки **`<stdio.h>`, `<string.h>`, `<math.h>`.**

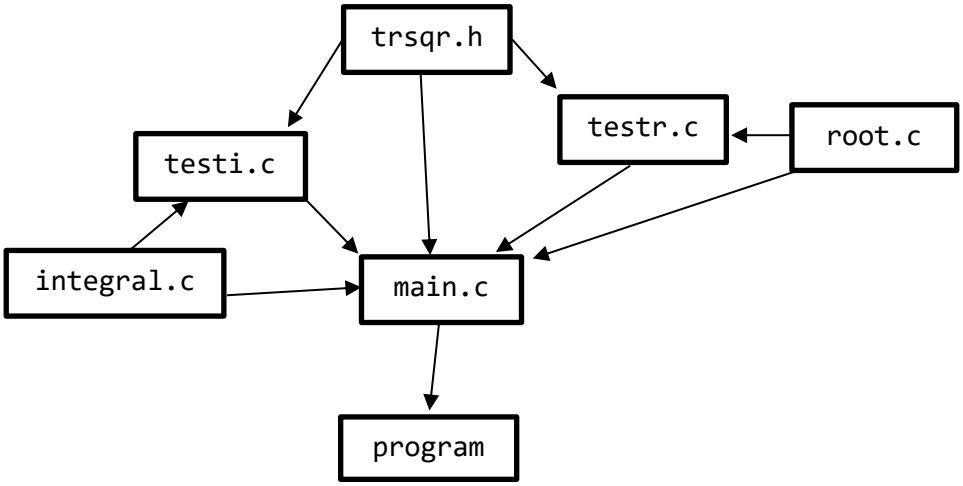
Сборка программы (Make-файл)

Makefile собирает модули в исполняемый файл **program**. Сборка происходит по команде **make all**, с возможностью выбора метода нахождения корня уравнения через добавления команды **R=0** (метод касательных) или **R=1** (метод хорд (по умолчанию)). Удаление промежуточных файлов происходит по команде **make clean**.

```
ASM = asm_func
#---------Program consists of 6 files:-----------------------------------------
#---------"asm_functions" - math functions written on nasm----------------------
#---------"root" - calculating point of intersection of two functions------------
#---------"integral" - calculating Riemann integral-----------------------------
#---------"main" - calculates sqr of curved triangle and gives UI----------------
#---------"testr" - allows testing "root" function------------------------------
#---------"testi" - allows testing "integral" function--------------------------


SRC = asm_func.o testr.o testi.o integral.o root.o main.o


#---------Flag for c-functions -m32 to build them in x86-32----------------------


CFLAGS = -m32 -c -o


#---------Format for assembly elf32(UNIX x86-32)---------------------------------


ASMFLAGS = -f elf32 -o


#---------Variable "R" determine the method  for "root" function:----------------
#---------Type "make R=0" to use tangent method----------------------------------
#---------Type "make R=1" to use the chord method(default)----------------------


R = 1


.PHONY: all clean
all: program
clean:
#---------Type "make clean" to remove all object files---------------------------
```

```
	rm -rf *.o
program: $(SRC)
	gcc -m32 -o $@ $(SRC) -lm
integral.o: integral.c
	gcc $(CFLAGS) integral.o integral.c
main.o: main.c trsqr.h
	gcc $(CFLAGS) main.o main.c
testr.o: testr.c trsqr.h
	gcc $(CFLAGS) testr.o testr.c
testi.o: testi.c trsqr.h
	gcc $(CFLAGS) testi.o testi.c
root.o: root.c
	gcc $(CFLAGS) root.o root.c -D 'RT=$(R)'
asm_func.o: asm_func.asm trsqr.h
	nasm $(ASMFLAGS) $(ASM).o $(ASM).asm
```

Схема зависимостей модулей:



Отладка программы, тестирование функций

Тестирование функций **root** и **integral** проводилось с помощью встроенных функций для тестирования. Результаты вычислений проверялись с помощью сервиса *www.wolframalpha.com*.

Функция **root** тестировалась на следующих уравнениях:

- $\sqrt{2x} = x^2$

Выбранный отрезок: [1.0, 2.0]

Результат работы функции: 1.259942

- $\frac{8}{x} = x^2$

Выбранный отрезок: [2.0, 3.0]

Результат работы функции: 2.000000

- $\sqrt{2x} = \frac{8}{x}$

Выбранный отрезок: [2.0, 4.0]

Результат работы функции: 3.174854

Функция **integral** тестировалась на следующих функциях:

- $f(x) = x^2$

Пределы интегрирования: [1.0, 5.0]

Результат работы функции: 41.3333

- $f(x) = \sqrt{2x}$

Пределы интегрирования: [2.0, 4.0]

Результат работы функции: 4.881819

- $f(x) = \frac{8}{x}$

Пределы интегрирования: [4.0, 4.5]

Результат работы функции: 0.942247

Программа на Си и на Ассемблере

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

Анализ допущенных ошибок

1. Ошибки в модуле **`main.c`**:

    - Передача аргументов в функцию осуществлялась через функцию **`fgets`**

      Исправлено: Для чтения ключей из командной строки добавлены аргументы функции **`main`**: `int argc, char *argv[].`

    - Количество итераций для вычисления считалось для каждой точки пересечения отдельно.

      Исправлено: Указатели переписаны под подсчет общего числа итераций.

    - Программа обрабатывала только один передаваемый ключ.

      Исправлено: Добавлен цикл **`for`** для обработки всех ключей.

    - При отсутствии ключей, программа не выводила значение площади по умолчанию

      Исправлено: Добавлено условие на отсутствие ключей.

2. Ошибки в модуле **`root.c`**:

    - Функция была выполнена рекурсивно, из-за чего вычисление знаков производных происходило перед каждой итерацией.

      Исправлено: Функция была переписана на цикл **`while`**.

    - Некорректный подсчет итераций из-за неправильного расположения счетчика.

      Исправлено: Счетчик перенесен в нужное место в функции.

Список литературы

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