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 5 / 3 / 2

Выполнил:
студент 106 группы
Шурыгин В. Е.

Преподаватель:
Манушин Д. В.

Москва
2021

Содержание

Постановка задачи **2**

Математическое обоснование **3**

Результаты экспериментов **5**

Структура программы и спецификация функций **6**

Сборка программы (Make-файл) **8**

Отладка программы, тестирование функций **9**

Тестирование функции *root* **9**

Тестирование функции *integral* **10**

Программа на Си и на Ассемблере **11**

Анализ допущенных ошибок **12**

Список цитируемой литературы **13**

Постановка задачи

В данном проекте требовалось реализовать численный метод, позволяющий с заданной точностью ε вычислять площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых y = f1(x) = $0.35 \cdot x^2 - 0.95 \cdot x + 2.7$, y = f2(x) = $3 \cdot x + 1$ и y = f3(x) = $\frac{1}{x+2}$ заранее определены вариантом задания. При этом вычисление значений функции необходимо было реализовать с помощью ассемблера NASM. Для нахождения площади использовался метод трапеций — метод численного интегрирования функции одной переменной, заключающийся в замене на каждом элементарном отрезке подынтегральной функции на многочлен первой степени, то есть линейную функцию. Площадь под графиком функции при таком подходе аппроксимируется прямоугольными трапециями.

Кроме того, для нахождения границ отрезка, точек пересечения функций, приближенного вычисления определенного интеграла использовался метод касательных. Основная идея метода заключается в следующем: задаётся начальное приближение вблизи предположительного корня, после чего строится касательная к графику исследуемой функции в точке приближения, для которой находится пересечение с осью абсцисс. Эта точка берётся в качестве следующего приближения. И так далее, пока не будет достигнута необходимая точность. С помощью данного метода были найдены абсциссы точек пересечения данных условием задачи функций.

Наконец, отрезки для применения метода касательных, а также выбор знака в алгебраической сумме определенных интегралов вычислялись аналитически. Был изучен "вид"и "расположение"искомой фигуры, оставалось написать программу для поиска точных значений с помощью изложенных выше методов.

Программу требовалось реализовать так, чтобы она могла поддерживать опции командной строки, а сборку программы требовалось осуществить с помощью утилиты make.

Математическое обоснование

Для сходимости метода касательных необходимо выполнение следующих условий:

- на концах отрезка функция имеет разные знаки
- на всём отрезке первая производная не меняет знак (и не обращается в ноль)
- на всём отрезке вторая производная не меняет знак (и не обращается в ноль)

Для сходимости метода трапеций требуется непрерывность второй производной у функции. Погрешность метода $R = \frac{(b-a)^3}{12n^2} f''(\xi), a < \xi < b$, то есть представляет собой $O(\frac{1}{n^2})$.(доказательство вы можете найти в [1])

Пусть $f_1 = 0.35 \cdot x^2 - 0.95 \cdot x + 2.7$, $f_2 = 3x + 1$, $f_3 = \frac{1}{x+2}$. Рассмотрим f_1 и f_2. Отрезок, выбранный для нахождения точки пересечения функций - $[-1.99; 2]$. Обозначим за $F(x) = f_1(x)$ - $f_2(x)$. $F(-1.99) \cdot F(-1.3) = (0.35 \cdot (-1.99)^2 - 0.95 \cdot (-1.99) + 2.7 - 3 \cdot (-1.99) - 1) \cdot (0.35 \cdot 2^2 - 0.95 \cdot 2 + 2.7 - 3 \cdot 2 - 1) = 10.946... \cdot -4.8 < 0 =>$ значения на концах отрезков имеют разные знаки. Рассмотрим первую производную. F'(x) $= 0.7x - 0.95 - 3$. При всех $x \in [-1.99; 2]$ она отрицательна, так как $0.7x <= 1.4 < 0.95 + 3$. Значит, первая производная не меняет знак. Рассмотрим вторую. $F''(x) = 0.7$. При всех x на нашем отрезке она положительна. Отрезок $[-1.99; 2]$ удовлетворяет достаточному признаку для нахождения корня.
Пусть теперь $F(x) = f_1(x)$ - $f_3(x)$. Выбранный отрезок - $[-1.99; -1.3]$. $F(-1.99) \cdot F(-1.3) = (0.35 \cdot (-1.99)^2 - 0.95 \cdot (-1.99) + 2.7 - \frac{1}{0.01}) \cdot (0.35 \cdot (-1.3)^2 - 0.95 \cdot (-1.3) + 2.7 - \frac{1}{0.49}) = -4.8 \cdot 3.584... < 0 =>$ значения на концах отрезков имеют разные знаки. Рассмотрим первую производную. F'(x) $= 0.7x - 0.95 - \frac{1}{(x+2)^2}$. При всех $x \in [-1.99; -1.3]$ выражение $0.7x - 0.95 < 0.7 \cdot (-1.3) - 0.95 = -1.86$, а $\frac{1}{(x+2)^2} > 2.04$, значит, $F'(x) < -3.9 < 0$ на всем рассматриваемом отрезке. Рассмотрим вторую: $F''(x) = 0.7 - \frac{1}{x+2^3}$. При всех $x \in [-1.99; -1.3]$ выражение $\frac{1}{(x+2)^3} > 2.9$, значит $F''(x) < 0$ на всем рассматриваемом отрезке. Отрезок $[-1.99; -1.3]$ удовлетворяет достаточному признаку для нахождения корня.
Пусть теперь $F(x) = f_2(x)$ - $f_3(x)$. Выбранный отрезок - $[-1.99; 2]$. $F(-1.99) \cdot F(2) = (3 \cdot (-1.99) + 1 - \frac{1}{0.01}) \cdot (3 \cdot 2 + 1 - \frac{1}{4}) = (-5 + 0.03) \cdot 6.75 < 0 =>$ значения на концах отрезков имеют разные знаки. Рассмотрим первую производную. F'(x) $= 3 + \frac{1}{(x+2)^2} > 0 \ \forall x$.Рассмотрим вторую. $F''(x) = \frac{-10}{x^3} < 0 \ \forall x > 0$. Отрезок $[4; 5]$ удовлетворяет достаточному признаку для нахождения корня.
При подсчете интеграла и нахождения корня использовались $\varepsilon_1 = \varepsilon_2 = 10^{-4}$. Для того, чтобы итоговая погрешность ε равнялась 10^{-3} было взято $\varepsilon_1 = 10^{-4}$, после чего решено неравенство относительно ε_2. Рассуждения при этом были следующие. (использовалось [2]) При нахождении точек пересечения функций мы теряем не более $\varepsilon_1 \cdot Fabsmax$, где $Fabsmax = max(|F(a)|, |F(b)|)$, где a, b -

границы отрезка, на котором ищется корень, $F(x)$ - функция, у который ищется корень. Такое выражение для потери площади характеризуется тем, что мы оцениваем её сверху площадью прямоугольника со сторонами ε_1 и $Fabsmax$. При подсчёте каждого интеграла, потеря точности по оценке сверху составляет не более чем $\varepsilon_1 \cdot Fabsmax_{ij} + \varepsilon_1 \cdot Fabsmax_{kl} + \varepsilon_2$ (здесь считается потеря с двух концов отрезка интегрирования и, собственно, сама погрешность интегрирования). Значит, итоговая оценка сверху: $2\varepsilon_1(Fabsmax_{12} + Fabsmax_{13} + Fabsmax_{23}) + 3\varepsilon_2$. $Fabsmax_{ij} = max(|f_i(a) - f_j(a)|, |f_i(b) - f_j(b)|)$.
На основе того, что в погрешности фигурирует разность значений функций используем такой приём: при реализации функции integral поставим в условие выхода из цикла, чтобы разность функций была меньше 1. Тогда итоговая погрешность будет составлять $2\varepsilon_1(1 + 1 + 1) + 3\varepsilon_2$ Теперь подставим $\varepsilon_1 = 10^{-4}$, $\varepsilon_2 = 10^{-4}$ и заметим, что $6 \cdot \varepsilon_1 + 3 \cdot \varepsilon_2 < \varepsilon = 10^{-3}$.

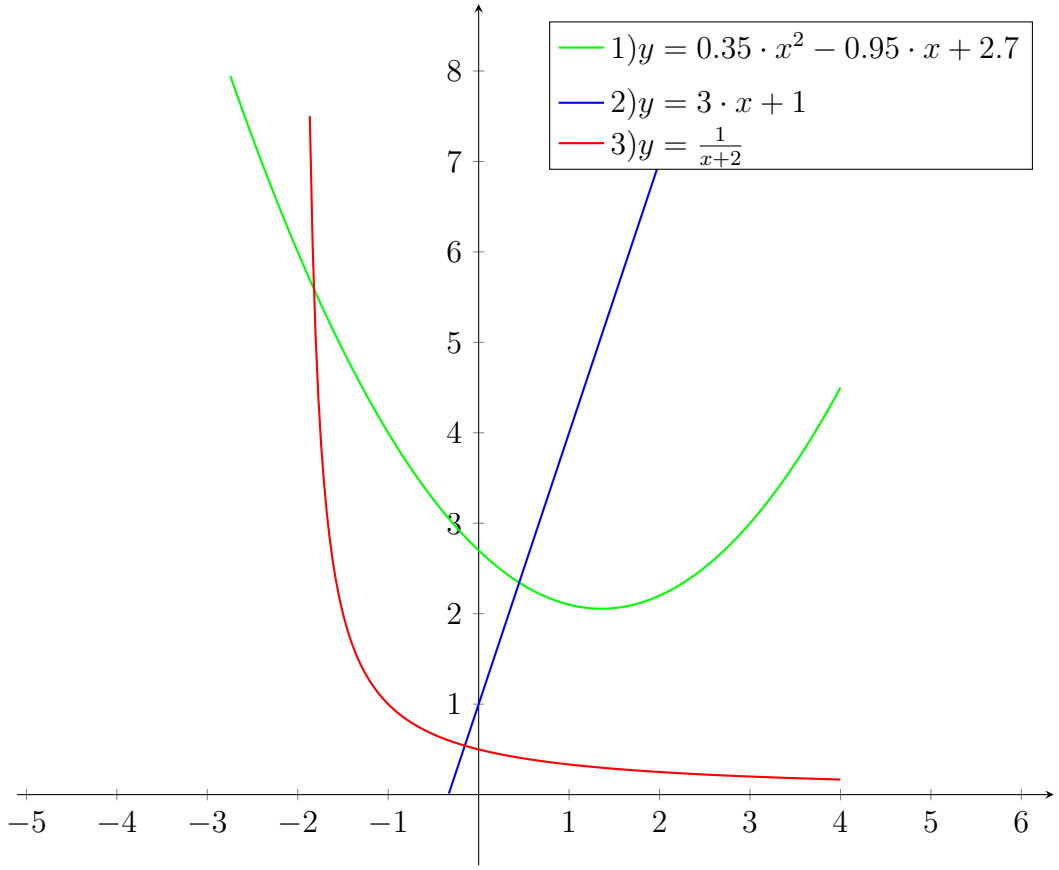


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Результаты экспериментов

Кривые	x	y
1 и 2	0.448178	2.344545
2 и 3	-0.152872	0.541383
1 и 3	-1.821140	5.590869

Таблица 1: Координаты точек пересечения

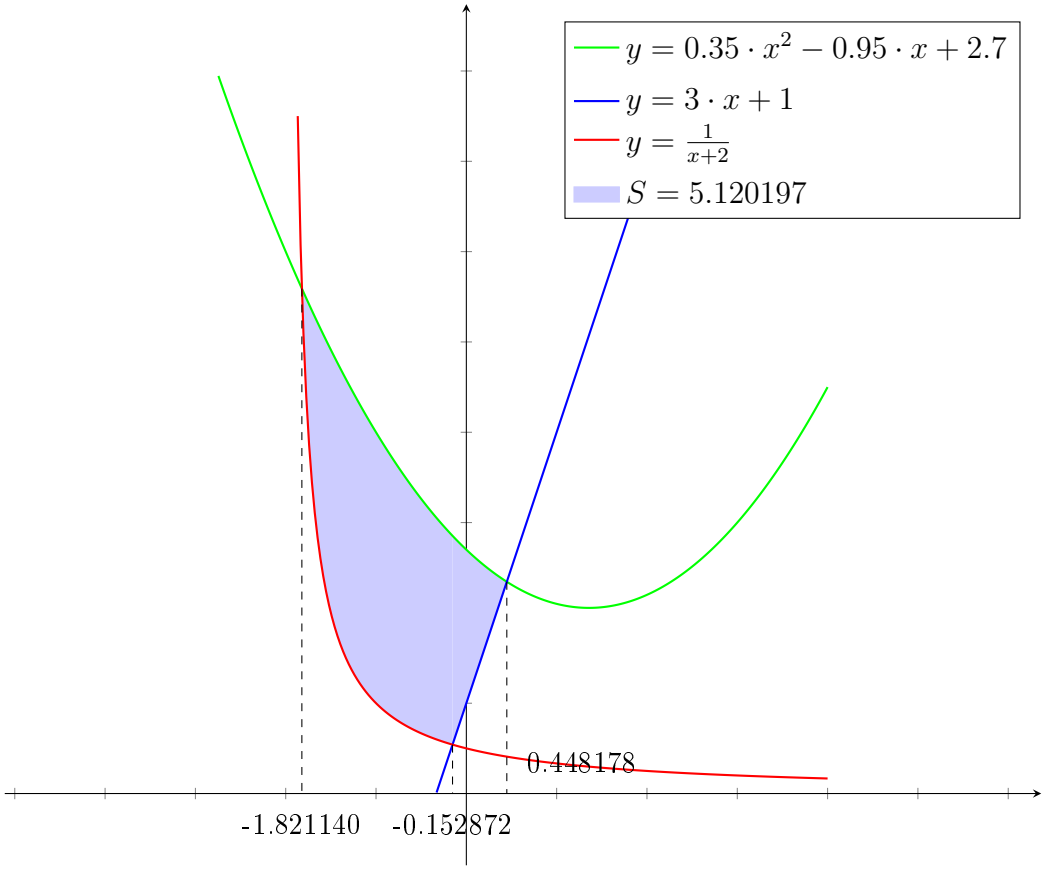


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Проект состоял из 2 модулей: одного на языке Си и одного на языке ассемблера.
Язык Си:

- main.c

Язык ассемблера:

- functions.asm

Библиотеки:

- math.h
- string.h
- stdlib.h

function.asm

Данный модуль обеспечивает вычисление заданных по условию проекта функций, а также их производных. Все функции написаны с использованием соглашения cdecl.

main.c

Главный модуль проекта. С его помощью происходит взаимодействие с командной строкой и вызов различных функций в зависимости от ключей командной строки. Также, с его помощью происходит печать в консоль требуемого пользователем результата.

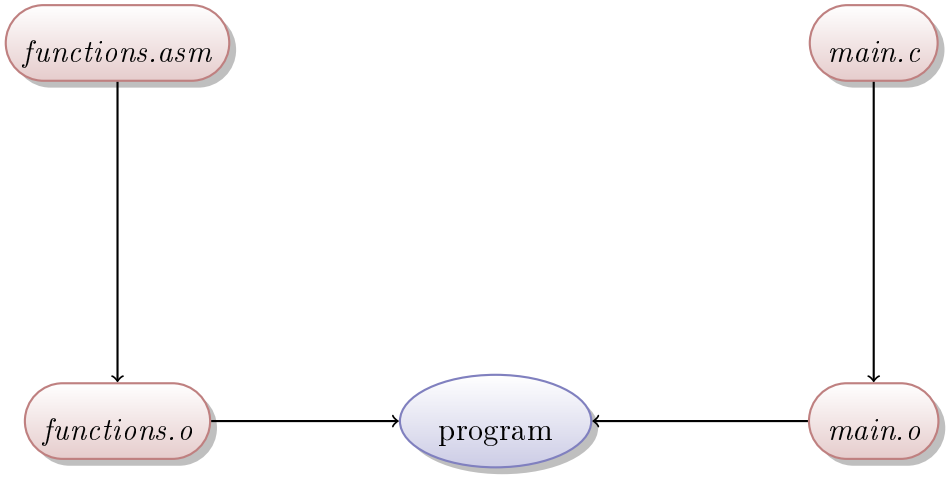
- $double\ integral(double(*f)(double),\ double\ a,\ double\ b,\ double\ eps)$
  Эта функция осуществляет метод трапеций для вычисления определенного интеграла заданной функции на заданном промежутке с заданной точностью.

- $double\ root(double(*f)(double),\ double(*g)(double),\ double\ a,\ double\ b,$
  $double\ eps,\ double(*fdif)(double),\ double(*gdif)(double))$
  Эта функция осуществляет метод касательных для вычисления абциссы точки пересечения двух заданных функций на заданном промежутке с заданной точностью. Также требует передачи в нее производных заданных функций. (для реализации использовалось [3])

Ключи командной строки:

- -help. Выводит список доступных команд с описанием их использования.
- -result. Выводит ответ на поставленную задачу перед проектом - площадь сектора, образованного пересечением кривых.
- -points. Выводит абсциссы точек пересечения заданных по условию кривых.
- -iters. Выводит количество итераций, произведенных для нахождения корней кривых, заданных по условию.
- -testroot. Позволяет протестировать функцию *root*. Пользователь сам может выбрать функцию из предложенных, а также может выбрать отрезок и точность.
- -testintegral. Позволяет протестировать функцию *integral*. Пользователь сам может выбрать функцию из предложенных, а также может выбрать отрезок и точность.

Схема сборки программы:



Сборка программы (Make-файл)

```
.PHONY: all clean

FLAGS = -Wall -Wextra -Werror -m32 -lm

all: program

program: main.o functions.o
gcc $(FLAGS) functions.o main.o -o program

main.o: main.c
gcc $(FLAGS) -c main.c -o main.o

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

clean:
rm -rf program *.o
```

Makefile собирает все модули в один файл - full. Делается это с помощью ключа "all". Удаление всех объектных файлов, а также итогового файла "full"происходит с помощью ключа "clean". В Make-файле присутствует метка ".PHONY которая позволяет избежать конфликтов, в случае если в проекте будет существовать файлы *all* и/или *clean*. Итоговый файл зависит от всех объектных, которые в свою очередь зависят от соответствующих си-/асм-файлов. (для реализации использовалось [4])

Отладка программы, тестирование функций

Тестирование численных методов проводилось с помощью нескольких вспомогательных функций.

Тестирование функции *root*

В программе присутствует возможность выбора отрезка для нахождения корня и точности пользователем с помощью командной строки. Здесь же будут приведены результат работы функции на отрезках и точностях, указанных в таблице.

№	Фукнция	Производная	Отрезок	Корень	Точность
1	$x^2 + x - 6$	$2x + 1$	$[0; 5]$	2.000	0.001
2	$ln(x)$	$\frac{1}{x}$	$[0,5; 2]$	1.000	0.001
3	$sin(x)$	$cos(x)$	$[2.5; 4.5]$	3.141	0.001

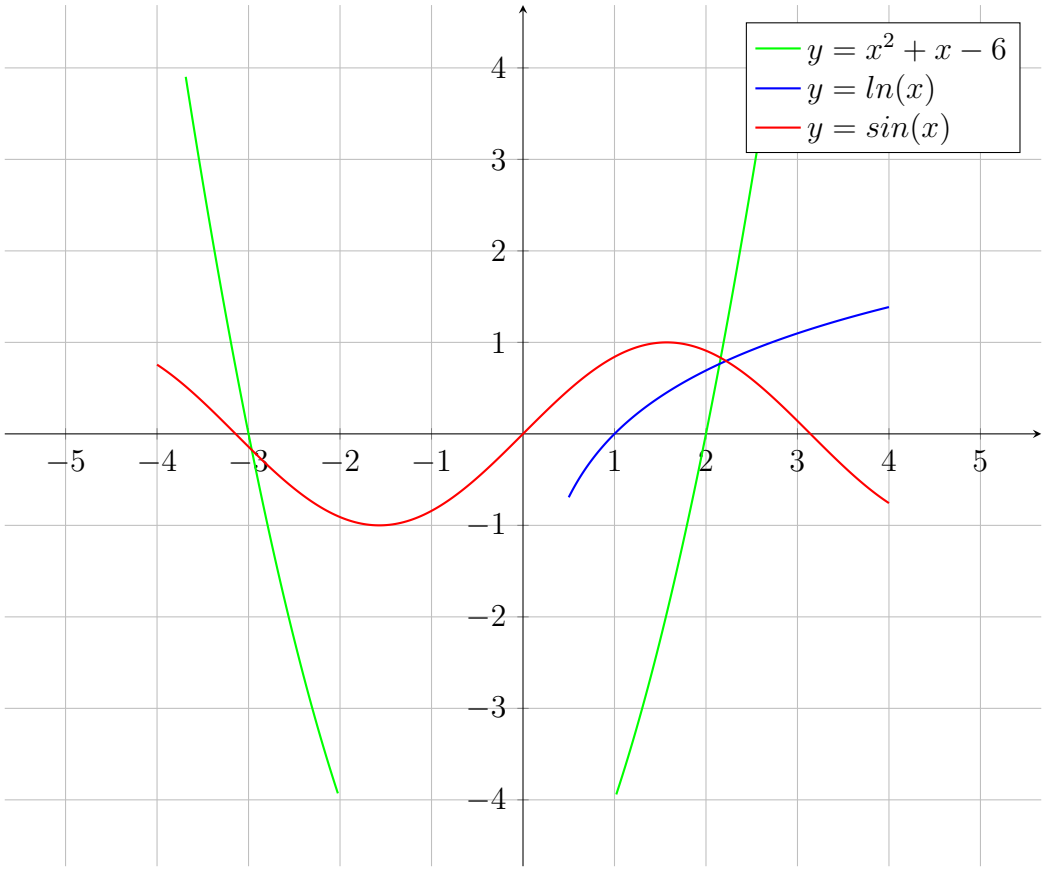


Рис. 3: Графики тестовых функций

Тестирование функции *integral*

В программе присутствует возможность выбора отрезка для нахождения определенного интеграла и точности пользователем с помощью командной строки. Здесь же будут приведены результат работы функции на отрезках и точностях, указанных в таблице.

№	Функция	Первообразная	Отрезок	Значение интеграла	Точность
1	x	$\frac{x^2}{2}$	$[0; 4]$	8.000	0.001
2	$-x^2 + 1$	$\frac{-x^3}{3} + x$	$[0; 1]$	0.667	0.001
3	$\sqrt{x}$	$\frac{2x^{\frac{3}{2}}}{3}$	$[0; 1]$	0.667	0.001

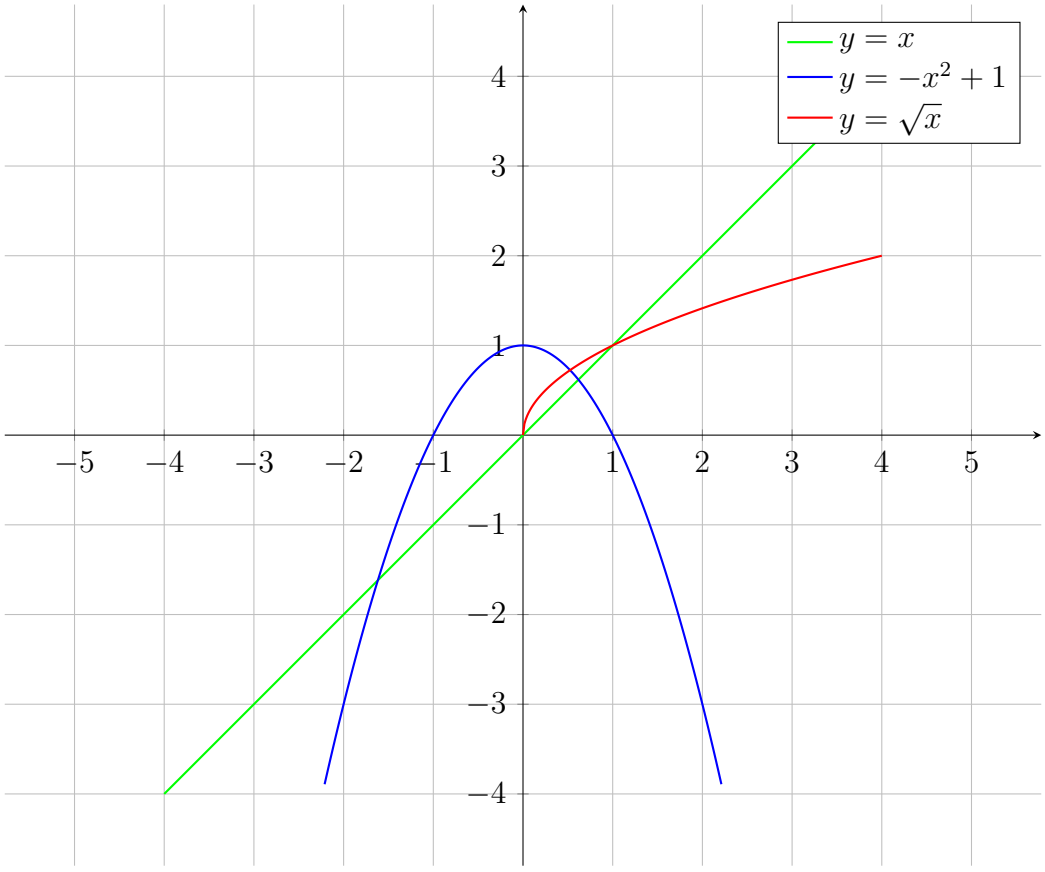


Рис. 4: Графики тестовых функций

Программа на Си и на Ассемблере

Исходные тексты программ имеются в архиве, который приложен к этому отчету. После использования цели all в Make-файле для запуска программы необходимо использовать команду ./program ..., где ... - ключи. Чтобы увидеть список всех ключей, необходимо ввести ./program -help. Чтобы узнать подробную информацию о том, какую задачу решает программа, можно ввести ./program -verbose или ./program -v.

Анализ допущенных ошибок

- На первом этапе разработки (весь код писался на языке Си, без взаимодействия с командной строчкой) не было обработки корректности получаемых значений из функций *root* и *integral*, а только сравнение итогового ответа с вычисленным аналитически. Позже это было доработано функциями *testroot* и *test_integral*;

- В первой версии второго этапа не был инициализирован сопроцессор (однако программа при этом всё равно работала), было доработано;

- Запуск ./program *test_root* приводил к зависанию программы, так как для поиска точки пересечения функций f1 и f2 правую границу отрезка следовало изменить;

- Опции -a, -b и -eps могли принимать некорректное значение параметра, не сигнализируя об этом. В случае -eps это приводило к бесконечному циклу (например, строка запуска ./program test_integral -eps 0); Также программа зависала в случае, если передать в test_integral значения a $>$ b.

Список литературы

[1] Ильин В. А., Садовничий В. А., Сендов Бл. Х. Математический анализ. Ч. 1 — Москва: Издательство Проспект, 2004.

[2] Пантаев М.Ю. Матанализ с человеческим лицом, или Как выжить после предельного перехода. Полный курс математического анализа. -Ленанд, 2016

[3] Трифонов Н.П., Пильщиков В.Н. «Задания практикума на ЭВМ»

[4] Статья «Просто о make» https://habr.com/ru/post/211751/