Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по №6

«Сборка многомодульных программ. Вычисление корней уравнений и определенных интегралов.»

Вариант 9 / 2 / 3

Выполнил:
студент 106 группы
Поцелуев А. А.

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

Москва
2021

Содержание

Постановка задачи

В данном задании требовалось реализовать многомодульную программу, вычисляющую площадь фигуры, ограниченной тремя кривыми, заданными заранее определенными функциями. Для вычисления площади следовало найти точки попарного пересечения кривых - это было реализовано с помощью метода хорд. Вычисление площади проводилось с помощью метода Симпсона. Отрезки для применения метода хорд, точность вычисления корней и интегральных сумм были подобраны аналитически. Вычисление значений исходных функций было реализовано на языке ассемблера, остальной код написан на языке C.

Математическое обоснование

Для сходимости метода хорд 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 Симпсона требуется непрерывность четвертой производной у функции. Погрешность метода $R = \frac{f^{(4)}(\xi)}{2880n^4}(b-a)^5, a < \xi < b.$, является $O(\frac{1}{n^4})$.

Пусть $f_1 = \frac{3}{(x-1)^2+1}$, $f_2 = \sqrt{x+0.5}$, $f_3 = e^{-x}$.
Рассмотрим $F_{12} = f_1 - f_2$ на отрезке $[1.6; 4]$. $F(1.6) \cdot F(4) = (\frac{3}{(1.6-1)^2+1} - \sqrt{1.6+0.5}) \cdot (\frac{3}{(4-1)^2+1} - \sqrt{4+0.5}) = -1.378 < 0 =>$ значения на концах отрезков имеют разные знаки. Рассмотрим первую производную. $F'(\text{x}) = -\frac{6(x-1)}{((x-1)^2+1)^2} - \frac{1}{2\sqrt{x+0.5}} < 0$, $\forall x \in [1.6; 4]$. Значит, первая производная не меняет знак. Рассмотрим вторую. $F''(x) = \frac{6 \cdot (3x^2-6x+2)}{((x-1)^2+1)^3} + \frac{1}{4(\sqrt{(x+0.5)^3})} > 0$, $\forall x \in [1.6; 4] =>$ отрезок $[1.6; 4]$ удовлетворяет достаточному признаку для нахождения корня.

Рассмотрим $F_{23} = f_2 - f_3$ на отрезке $[0; 1]$. $F(0) \cdot F(1) = (\sqrt{0+0.5} - e^0) \cdot (\sqrt{1+0.5} - e^{-1}) = -0.251 < 0 =>$ значения на концах отрезков имеют разные знаки. Рассмотрим первую производную. $F'(\text{x}) = \frac{1}{2\sqrt{x+0.5}} + e^{-x} > 0$, $\forall x \in [0; 1]$. Значит, первая производная не меняет знак. Рассмотрим вторую. $F''(x) = -\frac{1}{4(\sqrt{(x+0.5)^3})} - e^{-x} < 0$, $\forall x \in [0; 1] =>$ отрезок $[0; 1]$ удовлетворяет достаточному признаку для нахождения корня.

Рассмотрим $F_{13} = f_1 - f_3$ на отрезке $[-1; 1]$. $F(-1) \cdot F(1) = (\frac{3}{(-1-1)^2+1} - e^1) \cdot (\frac{3}{(1-1)^2+1} - e^{-1}) = -5.576 < 0 =>$ значения на концах отрезков имеют разные знаки. Рассмотрим первую производную. $F'(\text{x}) = -\frac{6(x-1)}{((x-1)^2+1)^2} + e^{-x} > 0$, $\forall x \in [-1; 1]$. Значит, первая производная не меняет знак. Рассмотрим вторую. $F''(x) = \frac{6 \cdot (3x^2-6x+2)}{((x-1)^2+1)^3} - e^{-x} > 0$, $\forall x \in [-1; 1] =>$ отрезок $[-1; 1]$ удовлетворяет достаточному признаку для нахождения корня.

При вычислении интеграла и нахождении корня использовались $\varepsilon_1 = \varepsilon_2 = 10^{-5}$. При вычислении корней итоговая погрешность оказывается не более $\varepsilon_1 \cdot F_{absmax}$, где $Fabsmax = max(|F(a)|, |F(b)|)$, где a, b - границы отрезка, на котором ищется корень, $F(x)$ - функция, у который ищется корень. Учитывая погрешность ε_1 вычисления корней, выразим погрешность вычисления интеграла ε_2:
$\varepsilon_2 = \frac{f^{(4)}(\xi)}{2880n^4}(b - a + 2\varepsilon_1)^5$

Откуда общая погрешность вычислений:
$|(\frac{f^{(4)}(\xi)}{2880n^4}(b-a+2\varepsilon_1)^5)$ -$(\frac{f^{(4)}(\xi)}{2880n^4}(b-a)^5)| < \varepsilon$, где $\varepsilon = 0.001$ по условию.

Подставив ε_2 и упростив подставим значения $\varepsilon_1 = \varepsilon_2 = 0.00001$, которые удовлетворяют этому неравенству. Необходимая точность достигнута.

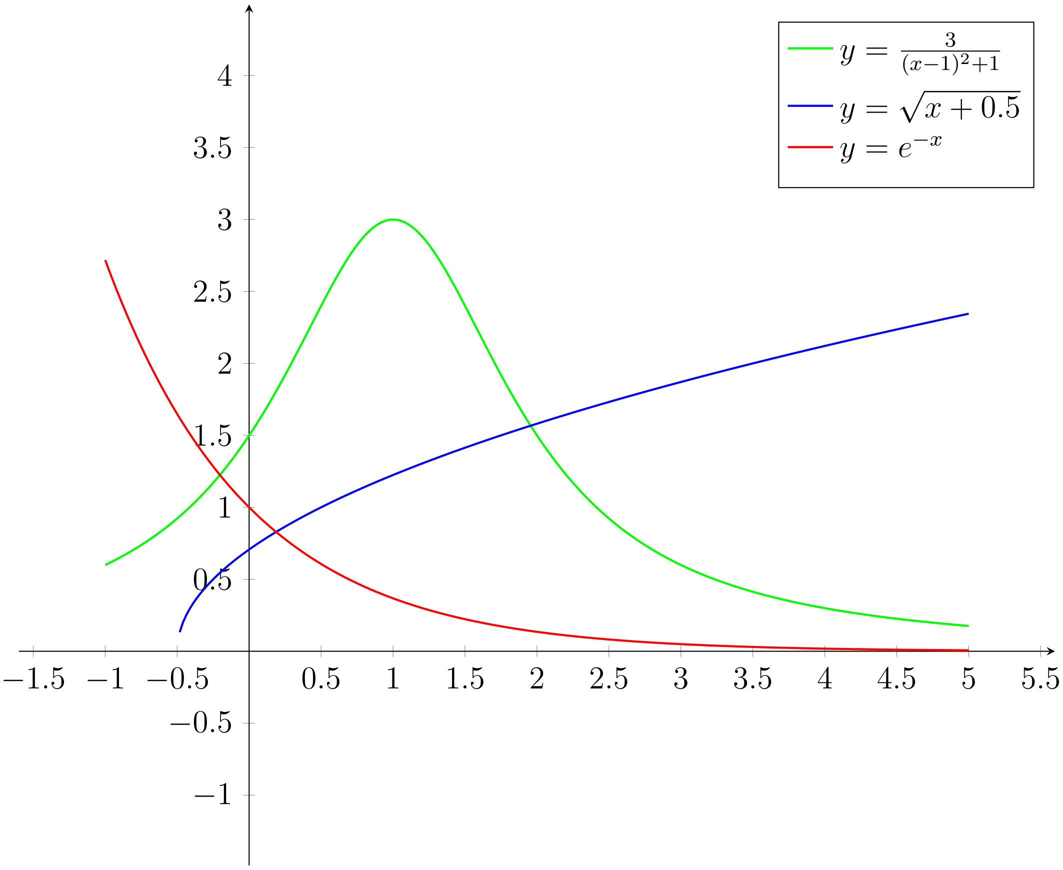


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Результаты экспериментов

Кривые	x	y
1 и 2	1.956153	1.567210
2 и 3	0.187412	0.829102
1 и 3	-0.203335	1.225480

Таблица 1: Координаты точек пересечения

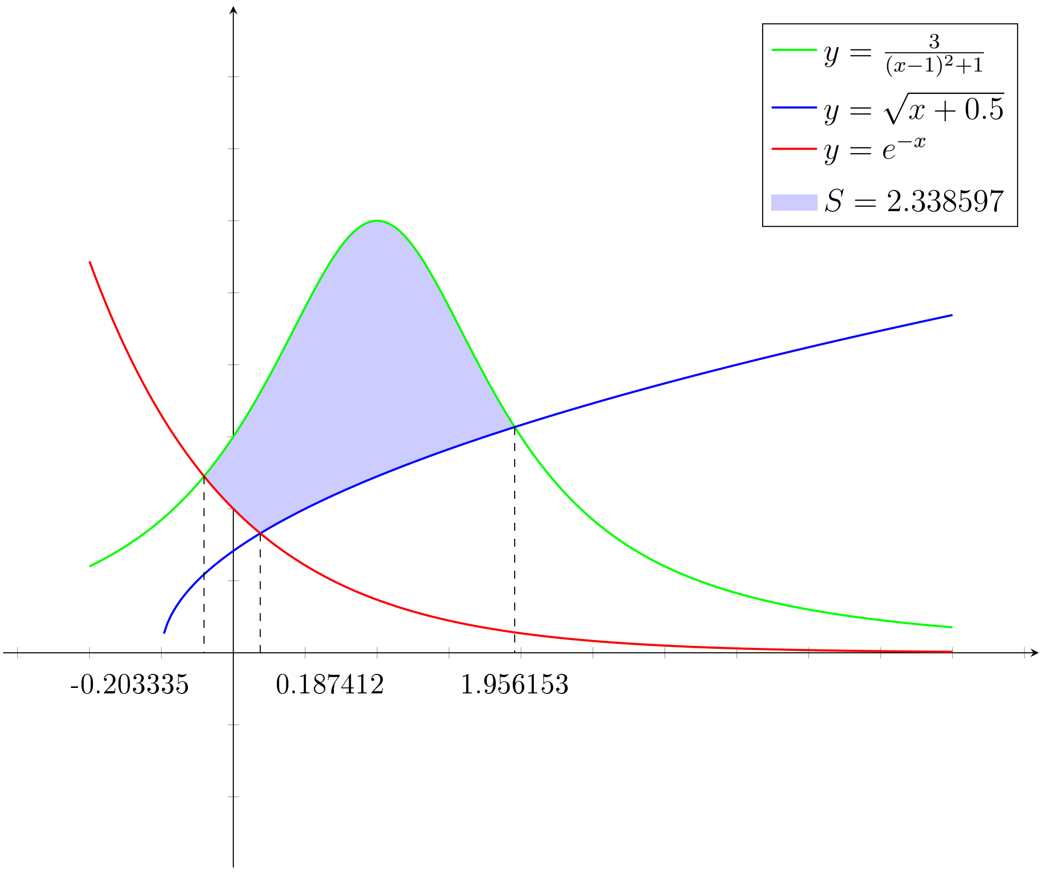


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Программа состоит из 8 модулей, написанных на 2-ух языках: язык ассембера и язык Си.
Язык Си:

- root.c
- integral.c
- main.c
- test.c

Язык ассемблера:

- f1.asm
- f2.asm
- f3.asm

Библиотека:

- flist.h

root.c

В данном модуле реализуется следующая функция:

- $double\ root(double(*f)(double),\ double(*g)(double),\ double\ a,\ double\ b,\ double\ eps)$
  Эта функция находит абциссу точки пересечения двух заданных функций на заданном промежутке с заданной точностью с помощью метода хорд.

integral.c

Данный модуль выполняет вычисление определенного интеграла заданной функции. Содержит в себе следующую функцию:

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

test.c

В данном модуле заданы функции для тестирования программы.

f1.asm, f2.asm, f3.asm

Данные модули обеспечивает вычисление заданных по условию проекта функций Все функции написаны с использованием соглашения cdecl.

flist.h

Данный модуль представляет собой библиотеку, содержащую в себе описание всех внешних функций.

main.c

Главный модуль проекта. Содержит следующие функции:

- $void\ help(void)$
  Эта функция печатает все возможные опции командной строки.

- $main\ (int\ argc, char^{**}\ argv)$
  Дальнейшее поведение программы зависит от полученных аргументов командной строки.

  – -help Выводит список доступных команд.

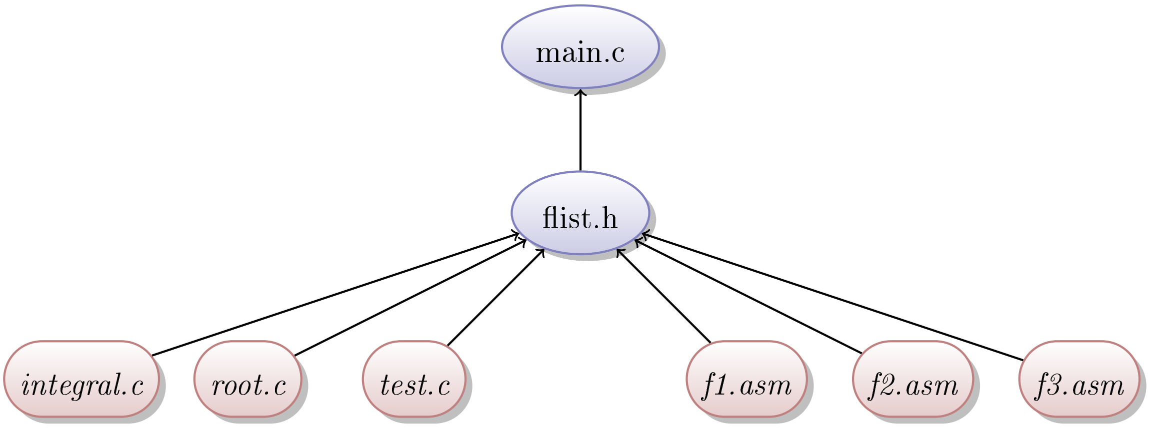
  – -roots Выводит абциссы точек пересечения исходных кривых.

  – -iters Выводит количество итераций, произведенных для нахождения абсцисс точек пересечения исходных кривых.

  – -test Позволяет протестировать функции $root$ и $integral$. Пользователь может выбрать тип проверки, а затем функцию(или пару функций для тестировния $root$) и ввести отрезок, на котором ищется площадь(или корень для функции $root$ соответстенно).

  – -v Выводит условие задания(исходные функции и названия методов, использованных в программе).

  – -at Запускает автоматическую проверку функций $root$ и $integral$. Для этого были выбраны некоторые функции и для них были вычислены верные значения корней и площади. При запуске тестирования полученные в результе работы программы значения сравниваются с предпосчитанными верными, и в случае их совпадения провека является пройденной.


main.c
flist.h
integral.c
root.c
test.c
f1.asm
f2.asm
f3.asm

Сборка программы (Make-файл)

```
GCC = gcc -m32
NASM = nasm -f elf32

.PHONY: all clean

all: prjct

prjct: main.o f1.o f2.o f3.o root.o integral.o test.o
$(GCC) $^ -o prjct -lm

main.o: main.c
$(GCC) $^ -c

root.o: root.c
$(GCC) $^ -c

integral.o: integral.c
$(GCC) $^ -c

test.o: test.c
$(GCC) $^ -c

f1.o: f1.asm
$(NASM) $^

f2.o: f2.asm
$(NASM) $^

f3.o: f3.asm
$(NASM) $^

clean:
rm -rf *.o
```

makefile собирает все модули в один файл - prjct. Делается это с помощью ключа "all". Удаление всех объектных файлов, образовавшихся в процессе сборки, происходит с помощью ключа "clean". Метка ".PHONY"позволяет избежать конфликтов, в случае если в проекте будет существовать файлы *all* и/или *clean*. Из .asm и .c файлов формируются объектные файлы, от которых зависит итоговый файл *prjct*. Сами объектные файлы зависят от использованных в них внешних библиотек.

Отладка программы, тестирование функций

Тестирование численных методов проводилось с помощью нескольких вспомогательных функций.

Тестирование функции *root*

В программе присутсвует возможность выбора функции для тестирования, точности поиска и отрезка, в котором ищется корень. Ниже приведен результат работы функции на модельных примерах.

№	Фукнция	Отрезок	Корень	Точность
1	$x^2 + 3x + 1 - \frac{1}{x}$	$[-1.2; -0.5]$	-1.000	0.001
2	$cos(x) - \frac{1}{x}$	$[4.0; 6.0]$	4.917	0.001
3	$x^2 + 3x + 1 - cos(x)$	$[-3.0; -1.5]$	-2.276	0.001

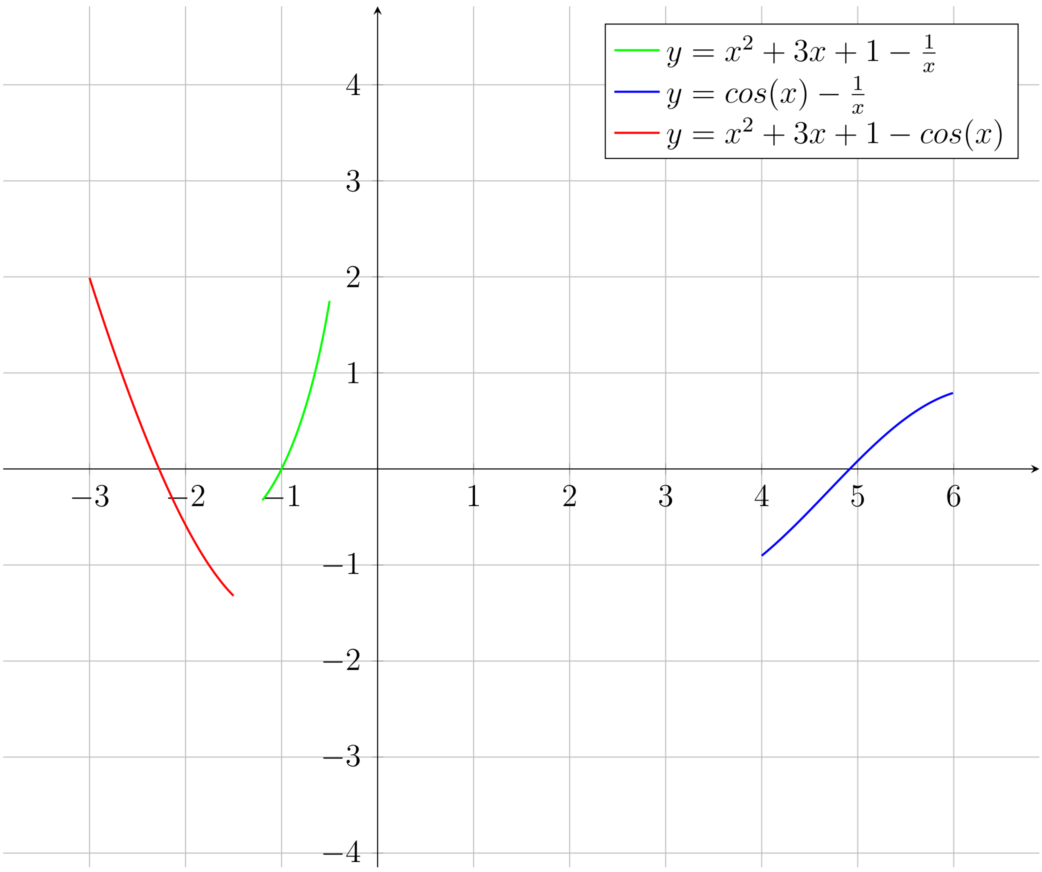


Рис. 3: Графики тестовых функций

Тестирование функции *integral*

В программе присутсвует возможность выбора функции для тестирования, точности поиска и отрезка для нахождения определенного интеграла. Ниже приведен результат работы функции на модельных примерах.

№	Фукнция	Отрезок	Значение интеграла	Точность
1	$x^2 + 3x + 1$	$[0; 1]$	2.833	0.001
2	$\frac{1}{x}$	$[1; 4]$	1.386	0.001
3	$cos(x)$	$[1; 3]$	-0.701	0.001

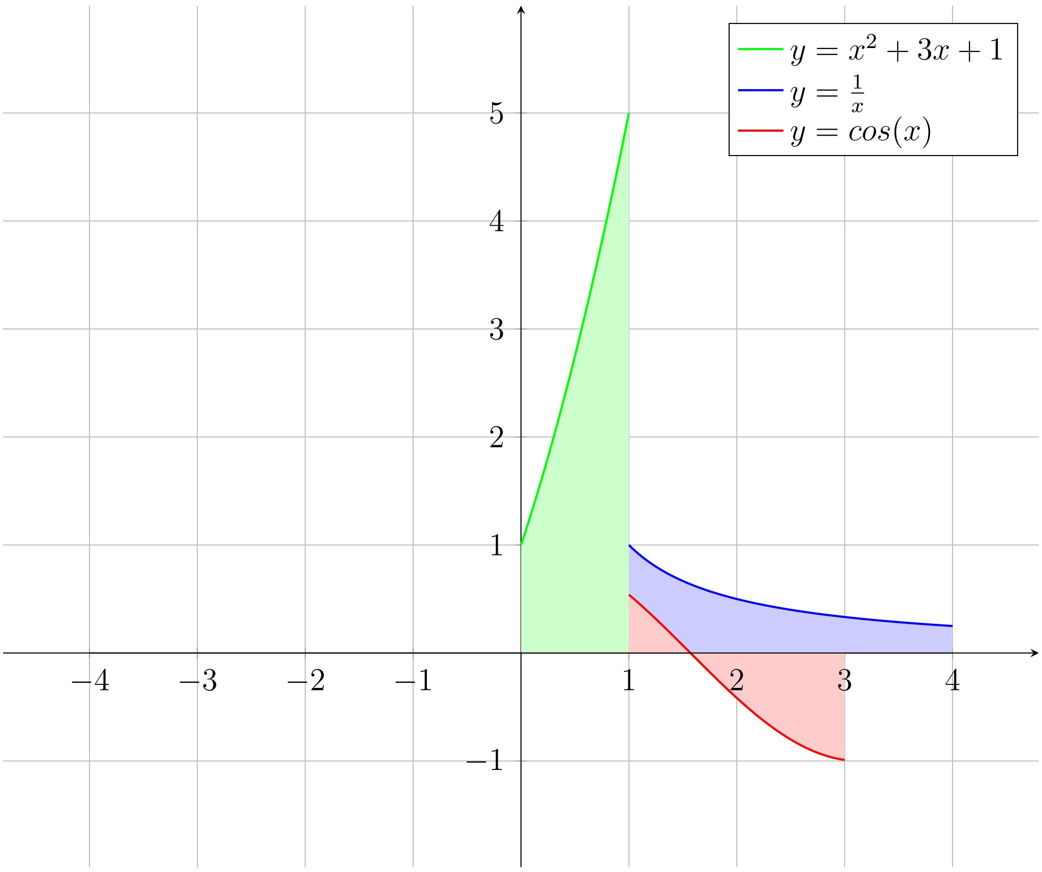


Рис. 4: Графики тестовых функций

Программа на Си и на Ассемблере

Исходный текст программы содержится в архиве и приложен к данному отчету.

Анализ допущенных ошибок

В процессе написания программы были допущены следующие ошибки:

- В функции *integral* в модуле integral.c при подсчете интегральных сумм I_{h} и $I_{2\mathrm{h}}$ оба значения вычислялись дважды. Позже это было исправлено - на каждом шаге используется результат предыдущего вычисления.

- При реализации вычислений на языке ассемблера периодически возникали некоторые ошибки.

Список литературы

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

[2] Бахвалов, Жидков, Кобельков. Численные методы. — Наука.