

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 / 2 / 1

Выполнила:
студентка 106 группы
Королёва Д. Л.

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

Москва
2024

Содержание

Постановка задачи	2
Математическое обоснование	3
Результаты экспериментов	5
Структура программы и спецификация функций	7
Сборка программы (Make-файл)	14
Отладка программы, тестирование функций	15
Программа на Си и на Ассемблере	17
Анализ допущенных ошибок	18
Список цитируемой литературы	19

Постановка задачи

В задании требуется с заданной точностью $\varepsilon = 10^{-3}$ вычислить площадь плоской фигуры, ограниченной тремя кривыми, заданными следующими функциями:

$$f_1 = 2^x + 1 \quad f_2 = x^5 \quad f_3 = \frac{1-x}{3}$$

Площадь вычисляется как разность площадей под графиками функций на соответствующих отрезках, ограниченных точками пересечения графиков функций, которые вычисляются в ходе выполнения программы.

В данном варианте задания площадь по графикам функции предлагалось считать используя метод прямоугольников.

Для поиска точек пересечения графиков функций $f(x)$ и $g(x)$ рассматривается промежуточная функция $F(x) = f(x) - g(x)$ и находится её корень на выбранном отрезке с помощью метода хорд.

Отрезки, на которых происходит поиск корней предварительно были вычислены аналитически.

Математическое обоснование

В методе хорд существенным является направление выпуклости функции и тип её монотонности на заданном отрезке. Так, если функция выпукла вниз и монотонно возрастает, то приближение идёт слева направо, если функция выпукла вниз и монотонно убывает - справа налево (т.е. фиксируется конец отрезка - точка b и вычисляется новое значение для точки $x = a$ на первой итерации метода). Данное замечание изображено на рисунке 1.

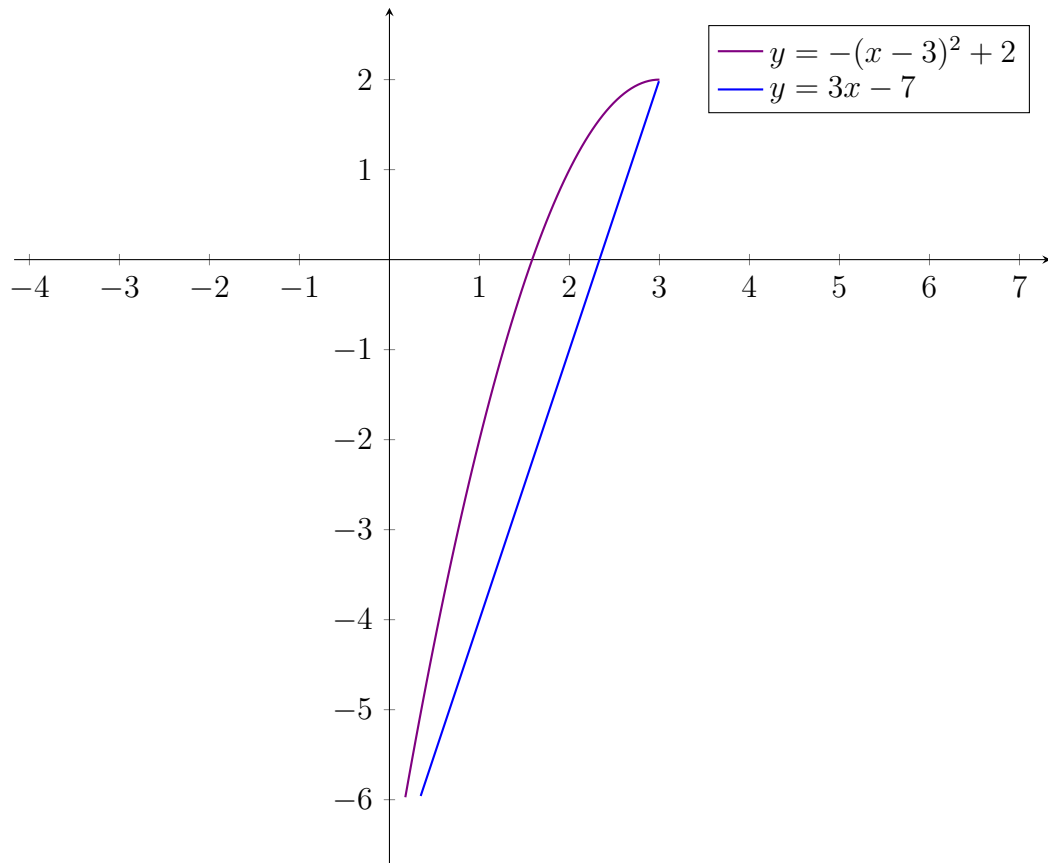


Рис. 1: Выпуклая вверх, возрастающая функция \Rightarrow приближаемся справа налево

Для того, чтобы понять тип монотонности функции на рассматриваемом отрезке $F(x)$ рассматривается знак её приращения $\text{delta}(F(x)) = F(x+x_0) - F(x_0)$, где $x = b$, $x_0 = a$.

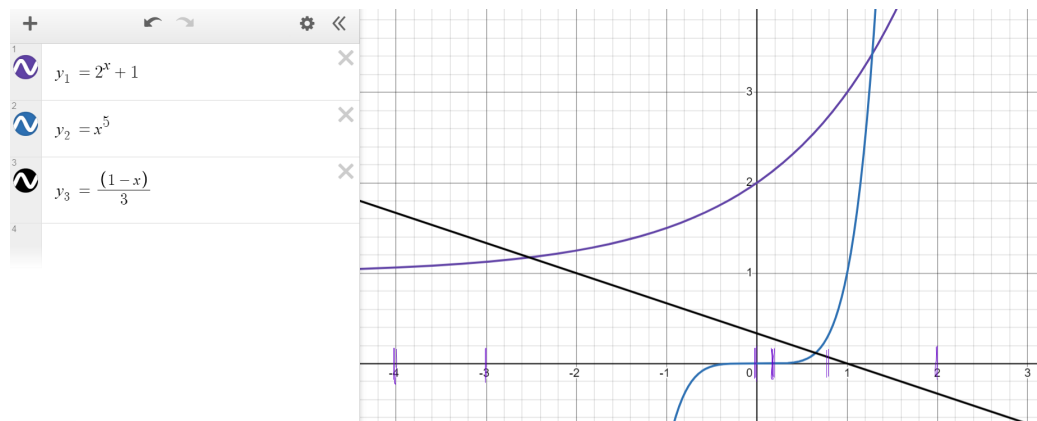
Для того, чтобы понять тип выпуклости функции на отрезке используется неравенство Йенсена: $F(\frac{a+b}{2})$ сравнивается с $\frac{F(a)+F(b)}{2}$.

Отрезки вычисляются таким образом, чтобы на них имелся только один корень, иначе метод хорд не будет сходиться к какой-то одной из них. Для вычисления отрезков использовался графический калькулятор Desmos :

$$f_1 = 2^x + 1$$

$$f_2 = x^5$$

$$f_3 = \frac{1-x}{3}$$



•

Из данных графиков были выбраны следующие отрезки:

для $y_1, y_2 - [0, 2]$

для $y_1, y_3 - [-4, -2]$

для $y_2, y_3 - [0.3, 0.9]$

Для того, чтобы при суммировании площадей значение было вычисленно с точностью $\varepsilon = 10^{-3}$ требуется, чтобы сами площади были вычислены с точностью, на один знак большей, чтобы избежать погрешности вычислений с плавающей точкой, поэтому $\varepsilon_2 = 10^{-4}$.

Из-за погрешности вычислений с плавающей точкой при расчёте значений функции в точке точность вычисления корня должна быть ещё на один порядок выше, поэтому $\varepsilon_1 = 10^{-5}$. При сравнении ответа, полученного своей программой с ответом от онлайн программы "Калькулятор Интегралов" и графическим калькулятором Desmos было подтверждено, что указанная точность достигается.

При численном интегрировании условием окончания вычисления интегральных сумм использовалась оценка погрешности численного метода: $\frac{(b-a)^3}{24n^2} f''(\xi)$ [1]. Где константа 24 - фиксированная, а $f''(\xi)$ и $(b-a)$ были посчитаны для данных функций аналитически. Таким образом в программе сначала высчитывается шаг для необходимой точности, как $step = \frac{\sqrt[3]{10^{-5}}}{24*5.0}$, после чего считается необходимое количество итераций для подсчёта площади.

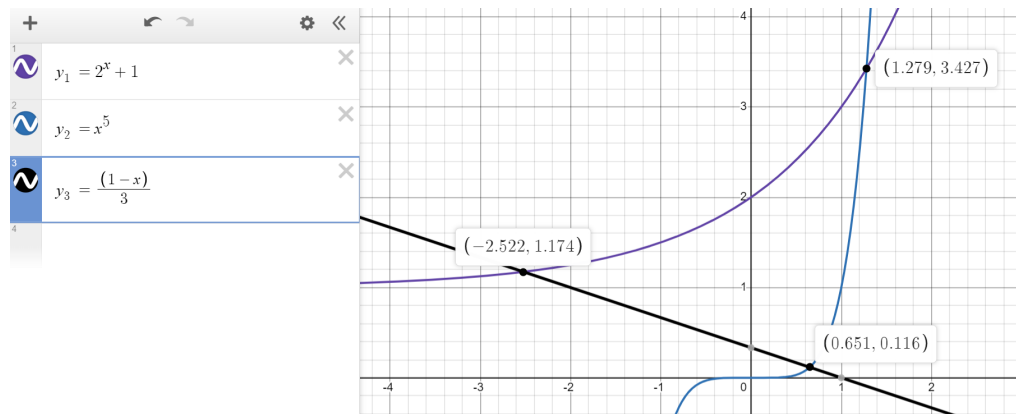
Результаты экспериментов

Координаты по x точек пересечения, посчитанные методом хорд:

Кривые	x
1 и 2	1.279293
2 и 3	0.649762
1 и 3	-2.522391

Таблица 1: Координаты точек пересечения

Координаты точек пересечения, посчитанных в Desmos:



Формула для подсчёта площади выглядит следующим образом:
 $integral(y1[root3, root1]) - integral(y2[root1, root2]) - integral(y3[root2, root1])$,
 где $root1 = root(y1, y2)$ $root2 = root(y2, y3)$ $root3 = root(y1, y3)$
 Отрезки интегрирования и значения площадей под графиками на данных отрезках, посчитанные с помощью программы "Калькулятор Интегралов":

кривая	начало отрезка	конец отрезка	площадь
1	-2.522	1.279	7.050
2	0.651	1.279	0.716
3	-2.522	0.651	2.047

Таблица 2: Значения площадей под графиками

Результат работы программы "Калькулятор Интегралов 4.287.

Результат работы моей программы - 4.287873.

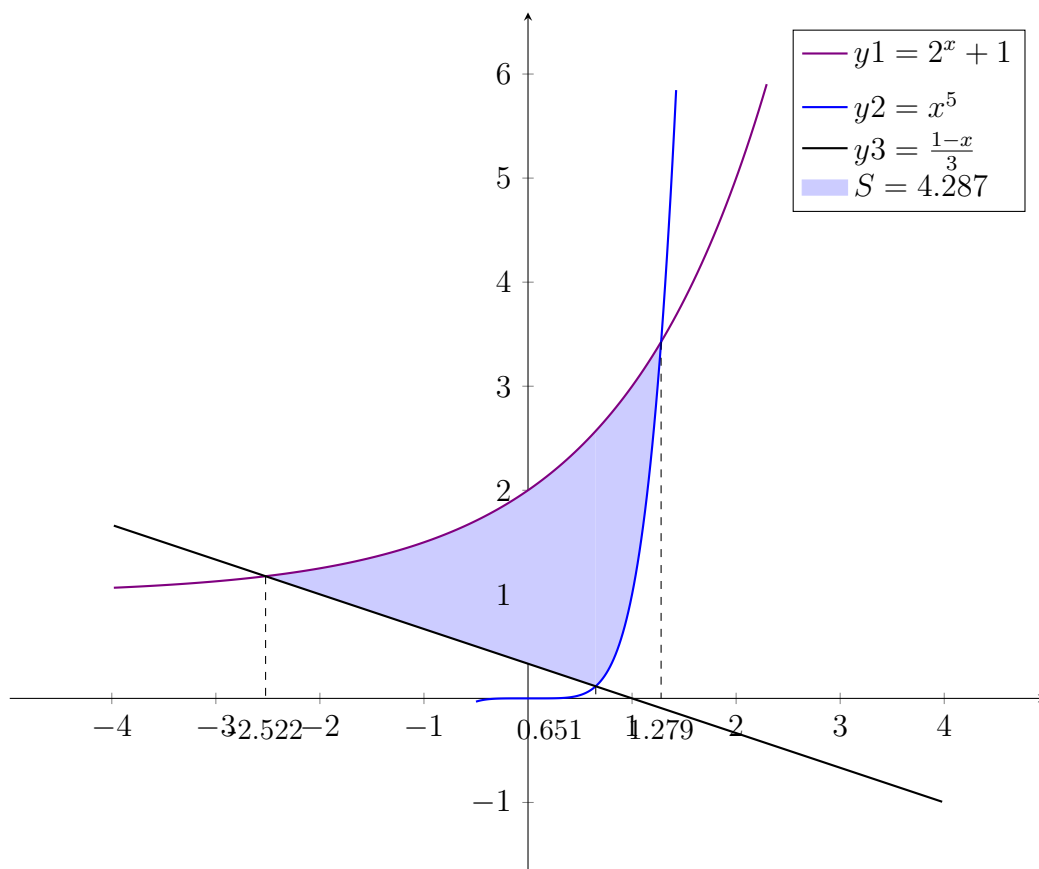


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Программа состоит из файлов:

`calculation_functions.c`, `print_functions.c` `ascii_art_func.c` и соответствующих заголовочных файлов (`.h`), `ascii_art.txt`, `main.c`, `input_functions.s` и соответствующего файла (`.h`).

Каждый из них содержит следующие модули:

`calculation_functions.c`: содержит функции *root* и *integral*

`double root(double(*f)(double), double(*g)(double), double left_border, double right_border, double eps1, int* iterations)` — функция нахождения точки пересечения двух графиков функций методом хорд. Принимает указатели на две функции, а также границы отрезка, на котором производится поиск и точность, с которой будет найден корень, также принимает указатель на переменную, в которой будет посчитано количество итераций. Возвращает найденное значение корня с точностью `eps1`.

`double integral(double(*f)(double), double left_border, double right_border, double eps2)` — функция вычисления определенного интеграла заданной функции на отрезке $[a, b]$ с точностью `eps2` методом прямоугольников. Принимает указатель на функцию, границы отрезка интегрирования и точность, с которой происходит интегрирование. Функция возвращает значение интеграла на отрезке.

`input_functions.s` — содержит реализации функций f_1, f_2, f_3 из задания и их производных.

`ascii_art_func.c`: содержит функцию `void print_ascii_art(FILE* file, int length)` для вывода в начале исполнения программы содержимого файла `ascii_art.txt` - изображения русалки и названия программы.

`print_functions.c`: содержит функции вывода значений корней, интегралов, площади, ограниченной графиками функции, числа итераций, тестовых значений интегралов и корней:

```
void print_iters_roots(void);
void print_x_coordinates(void);
void print_integral_value(void);
void print_coordinates_and_integral(void);
void print_face_border(void);
void print_usual_border(void);
void print_instructions_info(void);
void print_test_roots(int index_func1, int index_func2, double a, double b);
void print_test_integral(int index_func1, double a, double b);
```

Программа запускается из консоли и поддерживает следующие флаги:

- без флагов — выводит значение площади.

- `-test-root` <номер первой функции> <номер второй функции> <левая граница отрезка> <правая граница отрезка> — находит точку пересечения двух выбранных функций на выбранном интервале.

```
(kali@spider)-[~/Prac_6_task]
$ ./Mermaid -test-root 1 3 -4 -2
```



```
Root's x coordinate of functions #1 and #3 is: -2.522391
(* *) (* *) (* *)
```

- `-test-integral` <номер функции> <левая граница отрезка> <правая граница отрезка> — находит интеграл выбранной функции на выбранном отрезке.

```
(kali@spider)-[~/Prac_6_task]
$ ./Mermaid -test-integral 1 -2.522 1.279
```



```
The integral of 1-st function is 7.050537.
(* *) (* *) (* *)
```

Если вводится неправильный ключ, то программа сообщит вам об этом и выведет опцию **-help**:

[illegible]

Поддерживается воод нескольких ключей подряд:

[illegible]

Сборка программы (Make-файл)

текст Make-файла

```
FINAL = Mermaid

COMPILER = gcc
ASSEMBLER = nasm

CFLAGS = -std=c99 -c -m32 -g
SFLAGS = -f elf32
OFLAGS = -no-pie -z noexecstack -m32 -lm

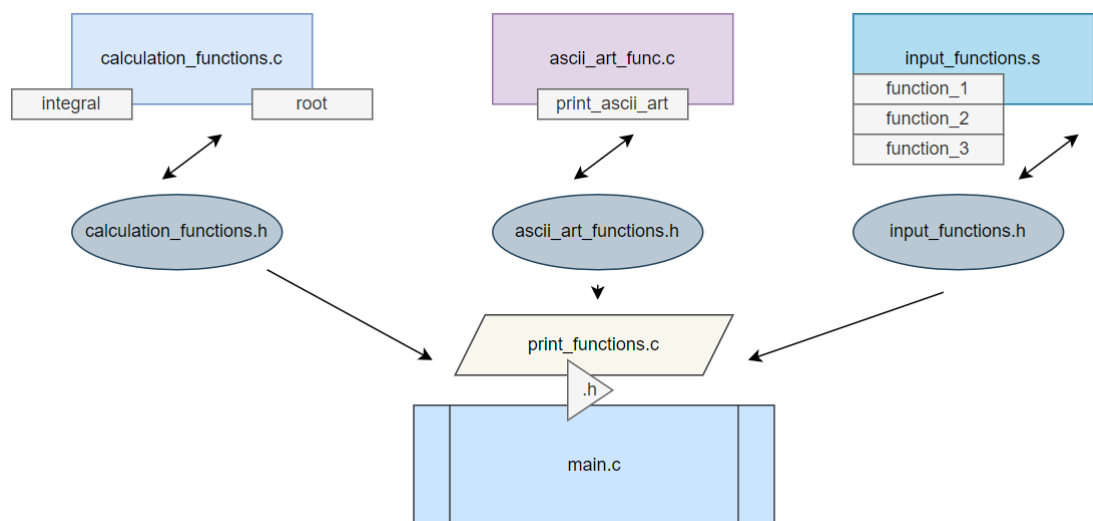
SRCC = $(wildcard *.c)
SRCS = $(wildcard *.s)

OBJC = $(patsubst %.c, %.o, $(SRCC))
OBJS = $(patsubst %.s, %.o, $(SRCS))

$(FINAL) : $(OBJC) $(OBJS)
    $(COMPILER) $(OBJC) $(OBJS) $(OFLAGS) -o $(FINAL)

%.o : %.c
    $(COMPILER) $(CFLAGS) $< -o $@
%.o : %.s
    $(ASSEMBLER) $(SFLAGS) $< -o $@
clean :
    rm $(FINAL) *.o
```

Схема работы компоновщика (также на последнем шаге подключается math.h).



Отладка программы, тестирование функций

Отладка *root*:

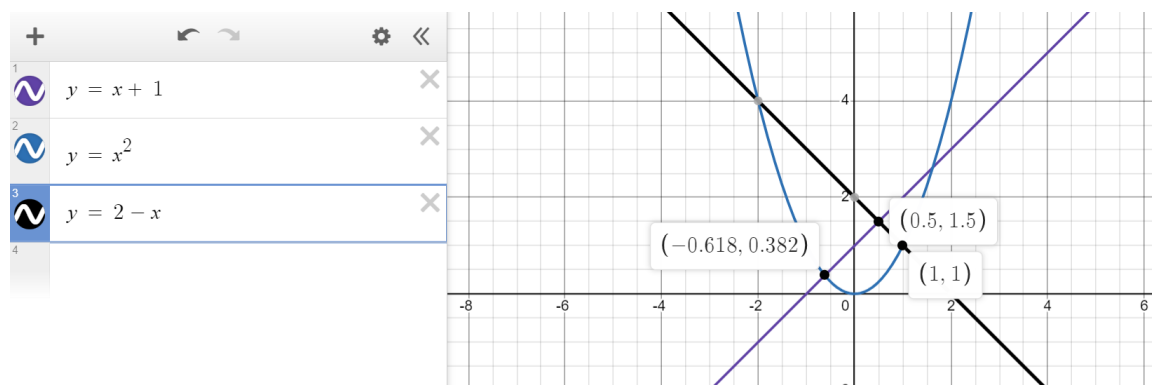
Будем тестировать на следующих функциях, чьи точки пересечения можно посчитать в графическом калькуляторе Desmos, либо посчитать вручную, корни ищем на отрезке $[-2, 0]$ для y_1, y_2 , на отрезке $[0, 0.7]$ для y_1, y_3 , на отрезке $[0, 2]$ для y_2, y_3 :

функция №1	функция №2	ответ программы
$y = x + 1$	$y = x^2$	-0.617612
$y = x + 1$	$y = 2 - x$	0.500000
$y = x^2$	$y = 2 - x$	1.000000

```
(kali@spider)-[~/Prac_6_task]
$ ./Mermaid -roots
```

```
Root's x coordinate of functions #1 and #2 is: -0.617612
Root's x coordinate of functions #2 and #3 is: 1.000000
Root's x coordinate of functions #1 and #3 is: 0.500000
(* *)
(* *)
(* *)
```


Корни, полученные в Desmos:



Тестирование *integral*: Будем вычислять определенный интеграл функции f на интервале $[a, b]$ и сравнивать с полученным программой "Калькулятор Интегралов".

функция f	интеграл	a	b	ответ программы	"Калькулятор Интегралов"
$y = x + 1$	$y = \frac{x^2}{2} + x$	-2	2	4.0000	4
$y = x^2$	$y = \frac{x^3}{3}$	-1	3	9.32323	9.3333
$y = 2 - x$	$y = 2x - \frac{1}{2}x^2$	2	3	-0.5000	-0.5

Программа на Си и на Ассемблере

Исходные тексты программы приложены к этому отчету.

Анализ допущенных ошибок

1. На каждой следующей итерации пересчитывала значения предыдущей итерации в функции `integral`, что приводило к чрезмерному расходу памяти.
2. При работе написании Makefile связывала не те файлы были забыты некоторые флаги компиляции (`-no-pie -m32`)
3. При написании функции `verb|integral|` при отладке устанавливала ограничение на количество итераций, из-за чего не достигалась нужная точность.
4. При написании функций на языке `asm` забывала команду `fini` при работе с `x87`.
5. При описании работы с входными параметрами `int agrc, char* argv[]` были ошибки с индексацией.
Все перечисленные ошибки исправлены. Программа работает корректно.

Список литературы

- [1] Садовничая И. В. Методы вычисления определенных интегралов. — Москва: 2024.
- [2] «Задания практикума на ЭВМ» Трифонов Н.П., Пильщиков В.Н

Список использованных сайтов:

<https://www.geeksforgeeks.org/>

<https://app.diagrams.net/>

<https://www.integral-calculator.ru/>

<https://www.desmos.com/calculator?lang=ru>