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 2 / 3 / 2

Выполнил: студент 104 группы

Митрофанов А.А.

Преподаватель:

Гуляев Д.А.

Москва 2022

**Содержание**

[Постановка задачи](#_TOC_250004) 3

[Математическое обоснование](#_TOC_250003) 4

Результаты экспериментов 6

[Структура программы и спецификация функций](#_TOC_250002) 7

Сборка программы (Make-файл) 7

[Отладка программы, тестирование функций](#_TOC_250001) 8

Программа на Си и на Ассемблере 9

[Анализ допущенных ошибок](#_TOC_250000) 10

Список цитируемой литературы 11

# Постановка задачи

В данной работе решалась задача поиска площади фигуры, ограниченной тремя кривыми с заданной точностью *ε*. В частности:

* требовалось реализовать численный метод подсчёта интегралов с помощью формулы трапеций;
* написать алгоритм нахождения точек пересечения графиков функций методом Ньютона (касательных), далее они и будут являться вершинами фигуры, площадь которой должна быть вычислена,
* отрезки для применения метода нахождения корней пересечения графиков были выбраны аналитическим меотдом.

**Математическое обоснование**

При наличии хорошего приближения изначальной границы поиска к корню функции f(x)=0 можно использовать метод Ньютона, или метод касательных. Таким образом, для корректной работы функции root(...), описанной далее, необходимо, чтобы функция на промежутке поиска всегда существовала.

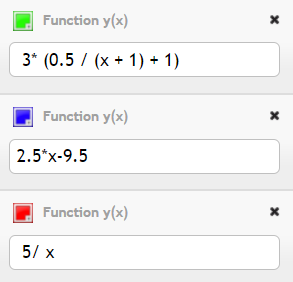
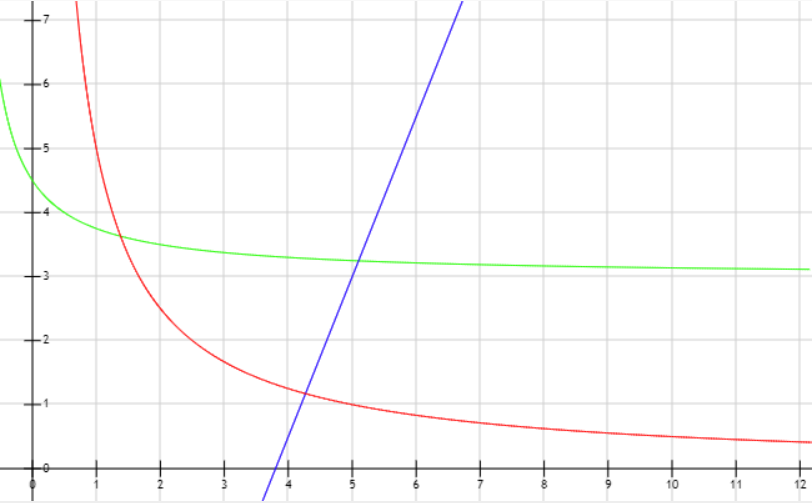
Алгоритм основан на следующем следствии из теоремы:

Если f(a) \* f(b) < 0, причем f’(x), f’’(x) отличны от нуля (и, следовательно, сохраняют определенные знаки при всех значениях x из промежутка [a, b], то, исходя из начального приближения, удовлетворяющего условию f(x)\*f''(x)>0, можно методом Ньютона по формуле xn+1=xn-(f(xn)/f’(xn)) вычислить единственный корень ξ уравнения, удовлетворяющий условию f(ξ)=0.

|  |  |
| --- | --- |
| *F*(*x*) | *F*(*x*) |
| *f*1(*x*) − *f*2(*x*)  *f*1(*x*) − *f*3(*x*)  *f*2(*x*) − *f*3(*x*) |  |

В качестве отрезков разбиения возьмём отрезки [1,5.5]. Изображение выглядит как текст

Автоматически созданное описание



**Результаты экспериментов**

|  |  |  |
| --- | --- | --- |
| Кривые | x | y |
| 1 и 2  2 и 3  1 и 3 | 5.05  4.2  1.3 | 3.1  1.1  3.6 |

|  |  |
| --- | --- |
| S фигуры | 4.1203 |

# 

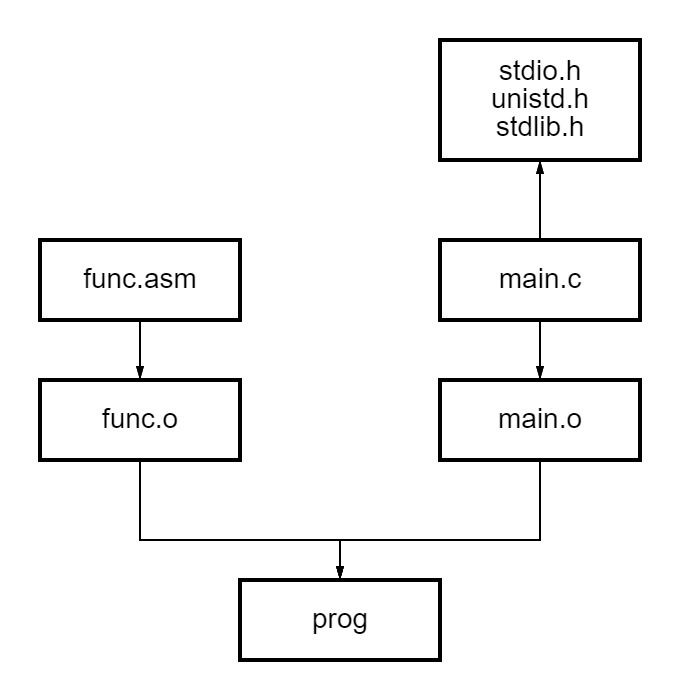
# Структура программы и спецификация функций

Для решения проблемы, поставленной выше, было реализовано две функции:

1) float rootNewton (func, func, func, func, func, func, float, float, float) - функция, для нахождения корня уравнения f(x)-g(x)=0 методом касательных. На вход функции подаются две функции f и g, отрезок, на котором будет происходить поиск и точность.

2)float trapezoidalintegral(func, float, float, float) – вычисление определённого интеграла методом трапеций. На вход функции подаются: функция, интеграл которой необходимо вычислить, пределы интегрирования и точность.

Разбиение программы на компоненты (модули, функции) и связь между этими компонентами:



**Сборка программы (Make-файл)**

Make-файл, использующийся для сборки программы, содержится в архиве, приложенном к отчёту. Основная программа содержится в файле "main.c", описание функций f1, f2, f3 в "func.asm". При сборке файлы компилируются до объектного кода, а затем линкуются.

Содержание Make-файла:

all: prog

prog: main.o func.o

gcc -o prog main.o func.o -m32 -lm

main.o: main.c

gcc -c -o main.o main.c -m32 -lm

func.o: func.asm

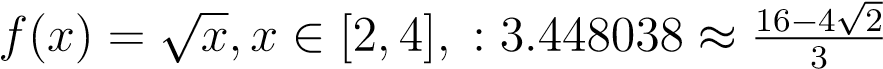
nasm -f elf32 -o func.o func.asm

clean:

rm \*.o

# Отладка программы, тестирование функций

Тестирование функции integral(...):

1. ,
2. ,
3. *f*(*x*) = 1*/x,x* ∈ [1*,e*]*,* : 1*.*000411 ≈ 1,
4. *f*(*x*) = *cos*(*x*)*,x* ∈ [0*,π/*2]*,* : 1*.*000008 ≈ 1,

Тестирование функции root(...):

1. *f*(*x*) = 3*x*2 − 4*,x* ∈ [1*.*5*,*3]*,* : 2*.*00000 ≈ 2,
2. *f*(*x*) = *ln*(*x*) − 1*,x* ∈ [2*,*3]*,* : 2*.*718282 ≈ *e*,
3. *f*(*x*) = – 4*,x* ∈ [10*,*20]*,* : 16*.*000124 ≈ 16,

# Программа на Си и на Ассемблере

# В архиве, приложенному к данному отчёту, находятся все необходимы текста программ как на Си, так и на языке Ассемблера.

# Анализ допущенных ошибок

Изначально программа на си работала с типом double, подключив функции из ассемблера, от типа double пришлось отказаться в пользу float. Также по невнимательности была допущена ещё одна ошибка: константы в data изначально имели тип word, в связи с чем возникли сложности с линковкой. К сожалению, очень много времени ушло на решение данной проблемы.

# Список литературы

1. Ильин В.А., Садовничий В.А., Сендов Бл.Х. Математический анализ. Т.1 — Москва: Наука, 1985.
2. http://dit.isuct.ru/IVT/sitanov/Literatura/M501/Pages/Glava25*.htm.https* :

*//ru.wikipedia.org/wiki, http://www.apmath.spbu.ru/*