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 № 9/2/2

Выполнил:
Студент 101 группы
Казаринов А. В.

Преподаватели:
Дудина И. А.
Кузьменкова Е. А.

Москва

2020

Содержание

Постановка задачи

Была поставлена задача реализовать метод вычисления площади фигуры, ограниченной тремя заданными кривыми, с точностью $\varepsilon = 0.001$. Уравнения кривых:

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

Задача состоит из двух шагов.

Первый шаг: найти вершины фигуры, являющиеся точками пересечения кривых, *методом хорд (секущих)*. Точки находятся решением приближённо уравнений $f_i(x) - f_j(x) = 0$, где $i = 1, 2, j = 2, 3, i \neq j$, с точностью ε_1.

Второй шаг: представить площадь фигуры в виде алгебраической суммы определённых интегралов. Вычислить данные интегралы нужно *по формуле трапеций* с точностью ε_2. В качестве пределов интегрирования берутся абсциссы точек пересечения кривых, вычисленные на первом шаге.

Для решения задачи требуется создать функции root и integral для нахождения корней уравнения и вычисления определенных интегралов соответственно. Данные функции необходимо протестировать. Также отрезок, на котором функция root будет искать корень, должен быть вычислен аналитически.

Сборка программы должна осуществляться при помощи утилиты make.

Математическое обоснование

I. Обоснование выбора отрезков для поиска точек пересечения кривых[1]

1) Пересечение кривых f_1 и f_2 ищется на отрезке [1, 2], $F_1 = f_1 - f_2$
$F_1(x) = \frac{3}{(x-1)^2+1} - \sqrt{x+0.5}$, $F_1(1) * F_1(2) = \left(3 - \sqrt{1.5}\right) * (1.5 - \sqrt{2.5}) < 0$
$F_1(x)$ непрерывна на отрезке [1, 2] как разность непрерывных функций.

2) Пересечение кривых f_2 и f_3 ищется на отрезке [0, 1], $F_2 = f_2 - f_3$
$F_2(x) = \sqrt{x+0.5} - e^{-x}$, $F_2(0) * F_2(1) = \left(\sqrt{0.5} - 1\right) * (\sqrt{1.5} - e^{-1}) < 0$
$F_2(x)$ непрерывна на отрезке [0, 1] как разность непрерывных функций.

3) Пересечение кривых f_1 и f_3 ищется на отрезке [-1, 0], $F_3 = f_1 - f_3$
$F_3(x) = \frac{3}{(x-1)^2+1} - e^{-x}$, $F_3(-1) * F_3(0) = (0.6 - e) * (1.5 - 1) < 0$
$F_3(x)$ непрерывна на отрезке [-1, 0] как разность непрерывных функций.

Для выбранных отрезков выполнена непрерывность функций F_1, F_2, F_3 и различие знаков этих функции на концах отрезков.

II. Обоснование выбора значений ε_1 и ε_2[2]

Возьмём ε_1 = 0.0001 и ε_2 = 0.00005
$\int \frac{3dx}{(x-1)^2+1} = 3\tan^{-1}(x-1) + C$, $\int e^{-x}dx = -e^{-x} + C$,

$\int \sqrt{x+0,5}\,dx = \frac{2(x+0,5)^{1.5}}{3} + C$.

Вычислим данные интегралы на отрезках $[x_i - \varepsilon_1, x_i + \varepsilon_1]$, то есть вычислим погрешность, возникающую из-за точности ε_1.

$3\tan^{-1}(x-1)\,\Big|_{1.9561}^{1.9563} \approx 0.00032$; $3\tan^{-1}(x-1)\,\Big|_{-0.2034}^{-0.2032} \approx 0.00025$;

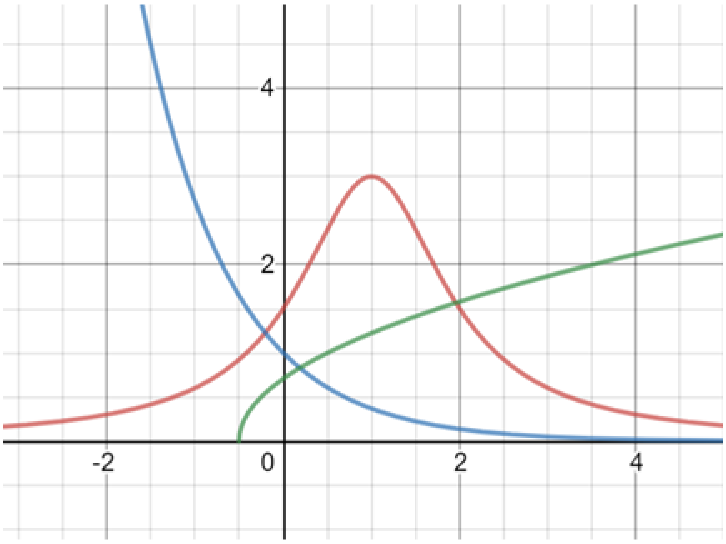
$\frac{2(x+0,5)^{1.5}}{3}\Big|_{1.9561}^{1.9563} \approx 0.00032$; $\frac{2(x+0,5)^{1.5}}{3}\Big|_{0.1875}^{0.1873} \approx 0.00017$;

$-e^{-x}\Big|_{0.1875}^{0.1873} \approx 0.00017$; $-e^{-x}\Big|_{-0.2034}^{-0.2032} \approx 0.00025$;

(0.00032 + 0.00025 + 0.00032 + 0.00017 + 0.00017 + 0.00025) / 2 + 3 * ε_2 = 0.00089 < ε

Суммарная погрешность ниже ε = 0.001, то есть условие выполнено.

III. Графики заданных кривых



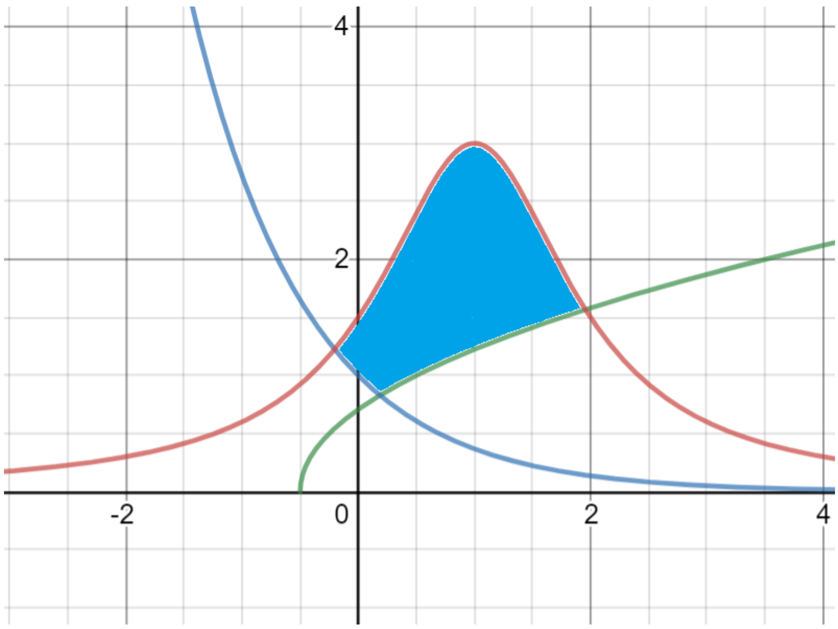
	$f_1 = \frac{3}{(x-1)^2+1}$
	$f_2 = \sqrt{x+0.5}$
	$f_3 = e^{-x}$

Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Результаты экспериментов

Кривые	x	y
f_1 и f_2	1.9562	1.5671
f_2 и f_3	0.1874	0.8291
f_1 и f_3	-0.2033	1.2254

Таблица 1: Координаты точек пересечения



	$f_1 = \frac{3}{(x-1)^2+1}$
	$f_2 = \sqrt{x+0.5}$
	$f_3 = e^{-x}$
S = 2.339±0.001	

Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

1) Файл **function.asm** содержит три функции f1, f2, f3, описанные на языке ассемблера. Данные функции получают на вход точку и выдают значение математической функции $f_1,\ f_2,\ f_3$ соответственно. Для работы со значениями с плавающей точкой используется сопроцессор x87.

2) **function.h** – заголовочный файл для файла function.c

3) Файл **integral.c** содержит функцию *integral (f, a, b, eps2)*, написанную на языке C. Функция *integral* должна на выдать значение определённого интеграла от функции f на отрезке [a, b] с точность eps2. В начале программы реализован небольшой цикл for, после которого в переменной *summ* лежит значение равное

$$0.5f_0 + f_1 + \cdots + f_{n_0-1} + 0.5f_{n_0}, \text{ где } f_i = (a + ih),\ \ h = \frac{a-b}{n_0},\ \ n_0 = 20.$$

Основой функции является двойной цикл for. Внутренний цикл добавляет в переменную *summ* значения f_i, но только те, которые ранее не вычислялись. Внешний цикл последовательно вычисляет значения I_{2n} при n равном $n_0, 2n_0, 4n_0$. Для этого достаточно переменную *summ* умножить на переменную h, которая делится пополам каждую итерацию. В конце каждой итерации цикла в переменных *int1* и *int2* лежат значения I_n и I_{2n} соответственно. На первой итерации значение I_{n_0} уже готово, так как посчитано до двойного цикла. В конце двойного цикла проверяется условие выхода из него (по правилу Рунге проверяется достигло значение интеграла требуемой точности).

4) Файл **main.c** содержит код, позволяющий поддерживать следующие опции командной строки: ключ командной строки -help, печать абсцисс точек пересечения кривых, печать числа итераций, потребовавшихся root, печать исходных функций, печать ординат точек пересечения кривых, вывод площади заданной фигуры (области), тестирование функции root, тестирование функции integral.

5) **Makefile** представлен в следующем разделе.

6) Файл **root.c** содержит функцию *root (f, g, a, b, eps1)*. В начале функции созданы несколько переменных, затем действия функции разбиваются на два случая: если функция F ($F = f - g$) возрастает и ее график расположен ниже хорды или если функция F убывает и ее график расположен выше хорды, то имеет место случай 1, иначе — случай 2. В первом случае проверяется не попал ли корень в правую eps1-окрестность точки c. Если попал, то значение c + eps1 / 2 будет корнем с нужной точностью. Иначе вызывается root для тех же функций и отрезка [c, b]. Во втором случае рассматривается левая eps1-окрестность точки c и выводится значение c + eps1 / 2, если корень лежит в данной окрестности. Иначе вызывается root для отрезка [a, c].

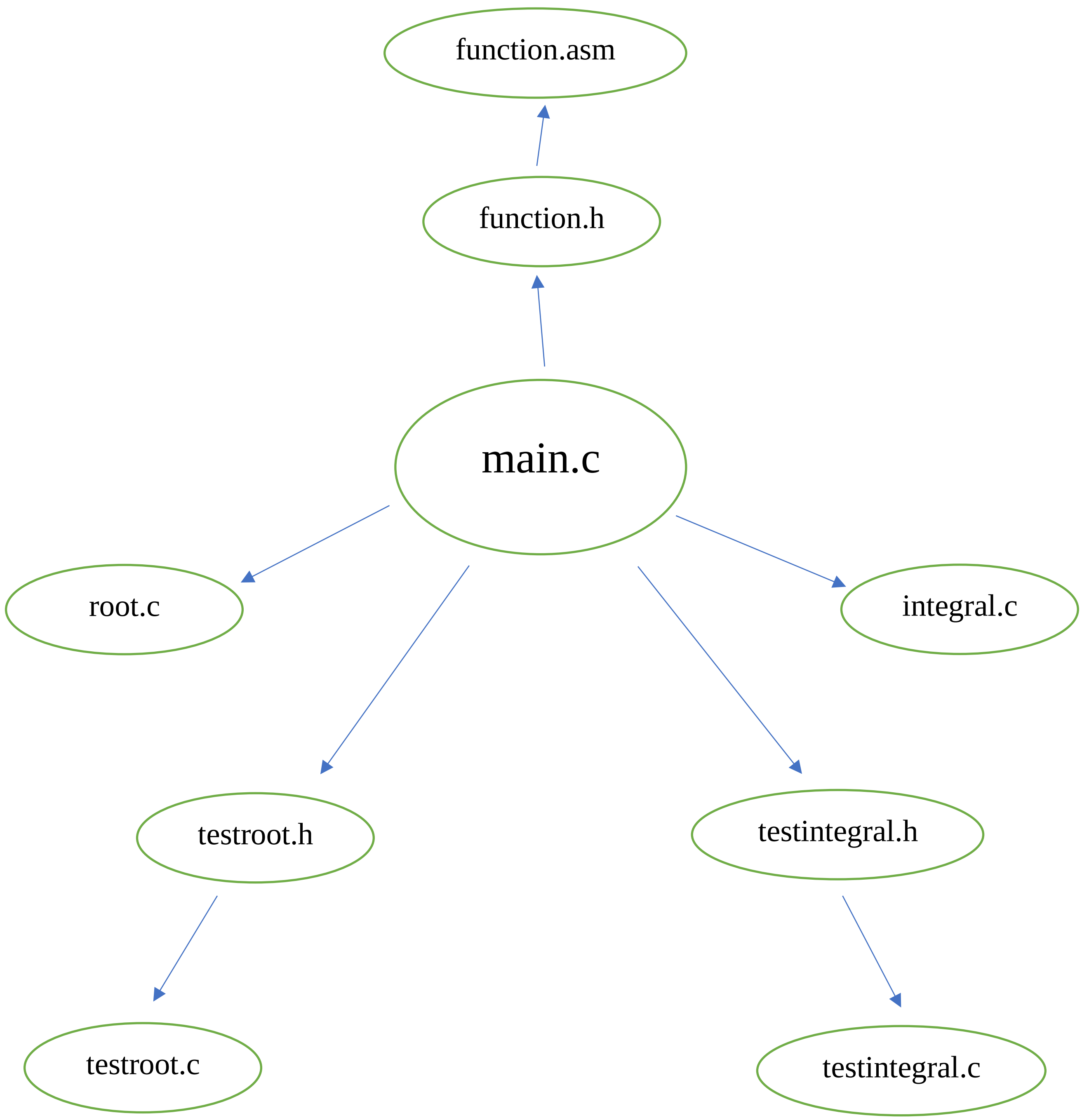
7) **testintegral.c** содержит три функции, вычисляющие значения математических функций. Подробнее об этих функциях в разделе отладка программы, тестирование функций.

8) **testintegral.h -** заголовочный файл для файла testintegral.c.

9) **testroot.c** содержит три пары функций, вычисляющих значения математических функций. Подробнее об этих функциях в разделе отладка программы, тестирование функций.

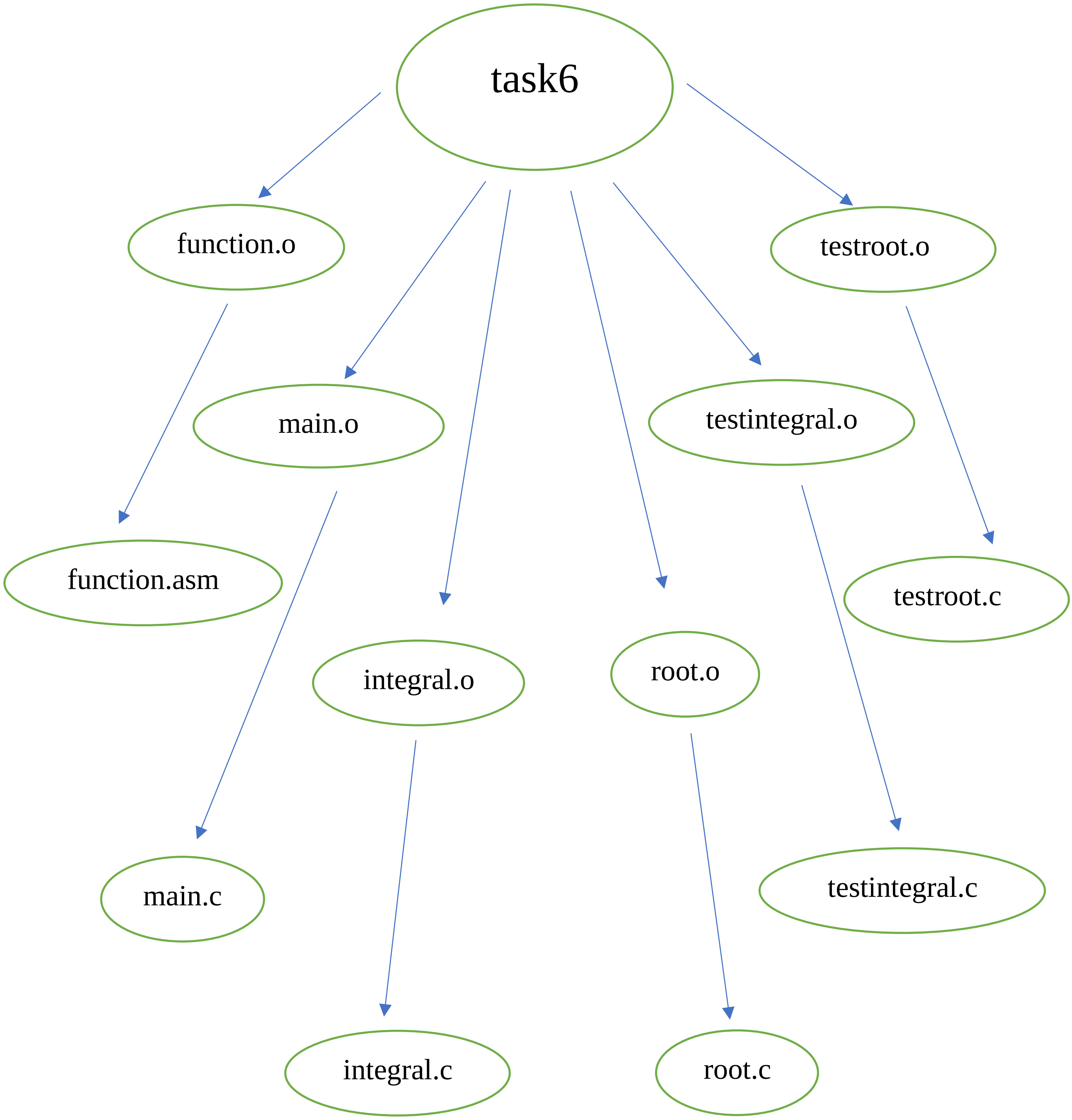
10) **testroot.h -** заголовочный файл для файла testroot.c.

Графическое разбиение программы на компоненты



Сборка программы (Make-файл)

I.Диаграмма зависимостей



II.Текст make-файла

```
SYSTYPE=UNIX
OBJFORMAT=elf32
CC=gcc
CFLAGS+=-m32 -Wall -g -O2 -W

all: task6
function.o: function.asm
	nasm -g -f $(OBJFORMAT) $< -o $@ -D$(SYSTYPE)
main.o: main.c
	$(CC) $(CFLAGS) -c $< -o $@
root.o: root.c
	$(CC) $(CFLAGS) -c $< -o $@
integral.o: integral.c
	$(CC) $(CFLAGS) -c $< -o $@
testroot.o: testroot.c
	$(CC) $(CFLAGS) -c $< -o $@
testintegral.o: testintegral.c
	$(CC) $(CFLAGS) -c $< -o $@
task6: function.o main.o root.o integral.o testroot.o testintegral.o
	$(CC) $(CFLAGS) $^ -lm -o $@
.PHONY: clean
clean:
	-rm -rf *.o
```

Отладка программы, тестирование функций

Для тестирования функции root используются 3 пары функций: x^2 и $x+2$, $sinx$ и x^3, $\sqrt{x}$ и $2^x - 14$.

x^2 и $x+2$ пересекаются в точках $(-1,\ 1)$, (2, 4). Для поиска точки $(-1,\ 1)$ можно взять отрезок $[-2, 0]$. Для поиска точки $(2,\ 4)$ можно взять отрезок $[1,\ 3]$. $F_{t1}(x) = x^2 - x - 2$ непрерывна на всей числовой прямой. $F_{t1}(-2) * \ F_{t1}(0) = 4 * (-2) < 0$. $F_{t1}(1) * \ F_{t1}(3) = (-2) * 4 < 0$.

$-x$ и x^3 пересекаются в точке (0, 0). Для поиска этой точки возьмём отрезок $[-1, 1]$. $F_{t2}(x) = \ -x - x^3$. $F_{t2}(-1) * \ F_{t2}(1)$ = 2 ∗ (-2) < 0

$\sqrt{x}$ и $2^x - 14$ пересекаются в точке (4, 2). Для поиска этой точки можно взять отрезок [3, 5]. $F_{t3}(x)$ = $\sqrt{x} - \ 2^x + 14$ непрерывна на [0, +∞). $F_{t3}(3) * \ F_{t3}(5)$ = ($\sqrt{3}$ + 6) ∗ ($\sqrt{5} - 18) < 0$.

Для тестирования функции integral используются три функции x^2, $sinx$, $\sqrt{x}$. x^2и $sinx$ интегрируемы на всей числовой оси, $\sqrt{x}$ интегрируема на отрезке, принадлежащем интервалу [0, +∞).

Например, x^2 можно проинтегрировать от 0 до 10, $sinx$ – от 0 до π, $\sqrt{x}$ – от 0 до 5. $\int_0^{10} x^2 = \ \frac{1000}{3}$, $\int_0^{\pi} sinx = \ 2$, $\int_0^{5} \sqrt{x} \approx 7.4534$.

Программа на Си и на Ассемблере

В архиве содержатся следующие файлы:

1) function.asm
2) function.h
3) integral.c
4) main.c
5) Makefile
6) root.c
7) testintegral.c
8) testintegral.h
9) testroot.c
10) testroot.h
11) Отчёт по заданию №6

Перечисленных файлов достаточно для сборки программы.

Анализ допущенных ошибок

При отладке функции integral была выявлена ошибка. Счётчик увеличивался на один при каждой итерации, требовалось увеличение в два раза. В связи с этим число новых точек росло следующим образом: 20, 40, 60, 80, 100… . Но на самом деле при размельчении промежутков разбиения (делении их пополам) возникало новых точек 20, 40, 80, 160… . Таким образом, программа учитывала часть значений из-за чего итоговая сумма имела недобор. Ошибка была выявлена при просмотре значений переменных (было заметно, что цикл не доходит до конца заданного отрезка на итерациях больших двух).

В процессе сборки программы возникала ошибка undefined reference по отношению к функциям из библиотеки math.h (sin, pow, sqrt), которые использовались для создания тестирующих функций. Оказалось, что компоновщику не передавалась опция -lm.

Список литературы

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

[2] Митин И. В., Русаков В. С. Анализ и обработка экспериментальных данных 1998. – 48с.