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 7 / 3 / 3

Выполнил:
студент 102 группы
Савиных Ю. С.

Преподаватель:
Кулагин А. В.

Москва
2020

Содержание

Постановка задачи

Необходимо реализовать метод Симпсона для подсчёта площади фигуры с точностью $\varepsilon = 0.001$, ограниченной тремя функциями:

- $f_1(x) = \ln x$
- $f_2(x) = -2x + 14$
- $f_3(x) = \frac{1}{2-x} + 6$

Для вычисления точек персечения функций использовать метод Ньютона, причём отрезок, к которому будет применяться метод касательных, вычислить аналитически.

Математическое обоснование

Для вычисления площади фигуры, ограниченной тремя заданными кривыми(рис. 1), необходимо вычислить точки пересечения, а значит корни уравнений $f_1(x) = f_2(x)$, $f_1(x) = f_3(x)$, $f_2(x) = f_3(x)$ т.е. корни уравнения $f_1(x) - f_2(x) = 0$, $f_1(x) - f_3(x)$, $f_2(x) - f_3(x) = 0$, которые в дальнейшем для простоты обозначим $F_1(x) = 0$, $F_2(x) = 0$, $F_3(x) = 0$ соответственно. Из нижеприведённых условий сходимости метода Ньютона [1]:

- Искомый корень изолирован на $[a, b]$
- $F(x)$ имеет непрерывную и монотонную первую производную.
- Первая производная $F(x)$ сохраняет знак на $[a, b]$

следует, что метод касательных необходимо применять на отрезке, на котором гарантированно существует и единственен корень. Достаточным условием существования корня на отрезке $[a, b]$ таково: на концах отрезка функция F(x) (т.е. наши функции $F_1(x)$, $F_2(x)$, $F_3(x)$) имеет разные знаки и на всём отрезке производная не меняет знак. Проанализируем функции $F_1(x)$, $F_2(x)$, $F_3(x)$, с целью вычислить отрезки, на которых существует единственный корень.
Поиск a, b:
Из построения графиков $f_1(x)$ и $f_2(x)$ (рис. 1) видно, что корень существует на отрезке $[5, 7]$, причём единственным образом. Проверим это утверждение с помощью достаточного условия существования корня:

- $F_1(5) = \ln 5 + 2 * 5 - 14 < 0$ (т.к. $\ln 5 < 3$)
- $F_1(7) = \ln 7 + 2 * 7 - 14 > 0$
- $F_1'(x) = 1/x + 2$, очевидно, что $F_1'(x) > 0$ на отрезке $[5, 7]$.

Значит корень действительно существует. Аналогичные рассуждения для $F_2(x)$ и $F_3(x)$ на отрезках $[2.1, 4]$ и $[4, 5]$ соответственно:

- $F_2(2.1) > 0$ ($F_3(4) > 0$)
- $F_2(4) < 0$ ($F_3(5) < 0$)
- $F_2'(x) = -\frac{1}{x} - \frac{1}{(2-x)^2}$. Значит $F_2'(x) \leq 0$ на отрезке $[2.1, 4]$ ($F_3'(x) = -2 - \frac{1}{(2-x)^2}$. Значит $F_3'(x) < 0$ на отрезке $[4, 5]$).

Точность метода Ньютона
Погрешность метода Ньютона [1]:

$$|x_i - c| \leq \frac{|F(x_i)|}{m}$$

Где x_i - i-ый член иттерационной последовательности, c - корень, m - минимальное значение $|F_x'|$ на сегменте $[a, b]$.
Величины $\varepsilon_1 = 0.0001$ и $\varepsilon_2 = 0.0001$, при которых вычисление площади происходит с погрешностью $\varepsilon = 0.001$, были подобраны вручную.

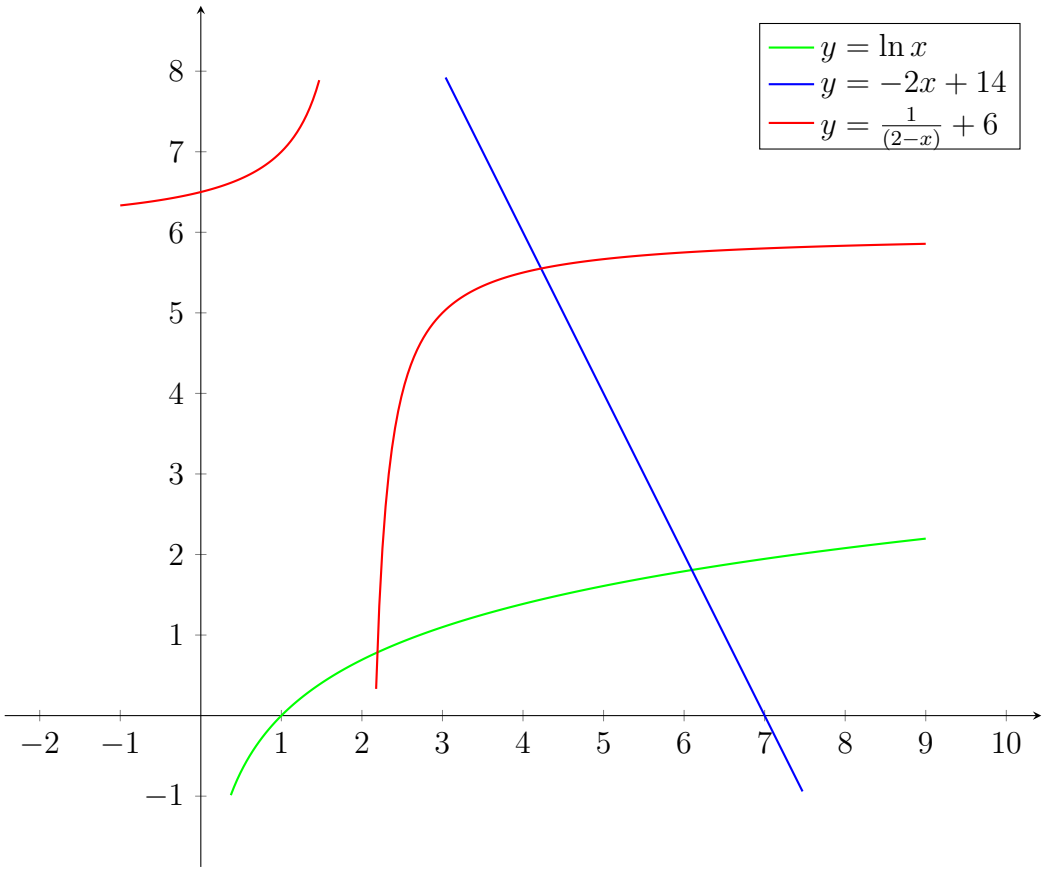


Рис. 1: Плоская фигура, ограниченная графиками заданных уравнений

Результаты экспериментов

Координаты точек пересечения предаствлены в таблице 1 и на рис. 2

Кривые	x	y
1 и 3	2.191742	0.784653
2 и 3	4.224745	5.550510
1 и 2	6.096169	1.807662

Таблица 1: Координаты точек пересечения

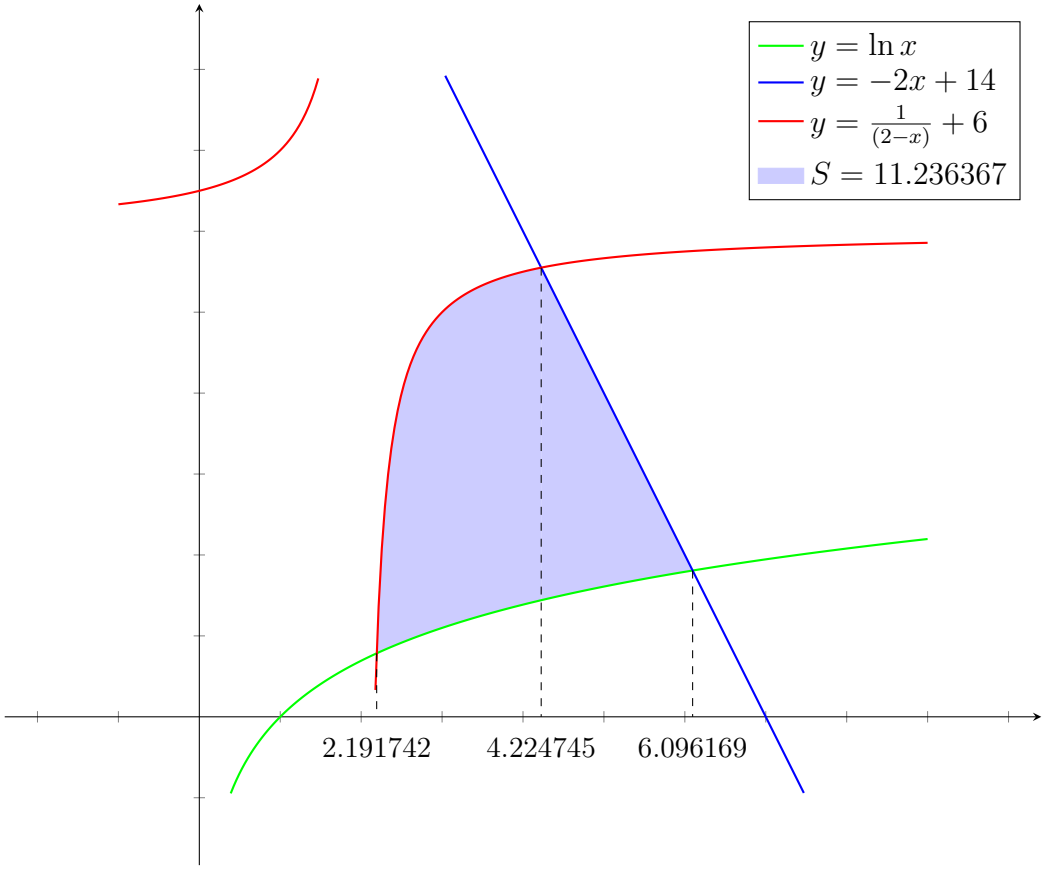


Рис. 2: Плоская фигура, ограниченная графиками заданных уравнений

Структура программы и спецификация функций

Программа состоит из трёх модулей:

1. main.c
2. fun.h
3. fun.asm

Функции реализованные в main.c:

`double root(double (*f)(double), double (*g)(double), double a, double b, double eps, double (*df)(double), double (*dg)(double))`

Функция методом Ньтона вычисляет с точностью ε корень уравнения $f(x) = g(x)$, который гарантированно существует на отрезке $[a, b]$. При вычисление корня методом касательных необходимо вычислить первые производные функций f и g, которые передаются в `root` 6 и 7 аргументами. Возвращает посчитанный корень.

`double I_n (double (*f) (double), double a, double b, int n)`

Вычисляет и возвращает интегральную сумму в методе Симпсона для функции f на отрезке $[a, b]$ с разбиением на n отрезков:

$$I_n = \frac{h}{3} * (F_0 + 2F_1 + 4F_2 + ... + 2F_{n-2} + 4F_{n-1} + F_n),$$

где $F_i = F(a + ih)$, $h = \frac{b-a}{n}$, причём n - чётное.

`double integral(double (*f) (double), double a, double b, double eps)`

Функция с точностью ε вычисляет и возвращает $\int\limits_a^b f(x)dx$

`void test_root()`

Функция запускающая тестирование фукнции `root`

`void test_integral()`

Функция запускающая тестирование фукнции `integral`

Фукнции инициализированные в fun.h и реализованные в fun.asm:

`double f1 (double x)`

Функция вычисляет и возвращает значение $f(x) = \ln x$

`double f2 (double x)`

Функция вычисляет и возвращает значение $f(x) = -2x + 14$

`double f3 (double x)`

Функция вычисляет и возвращает значение $f(x) = \frac{1}{2-x} + 6$

`double df1 (double x)`

Функция вычисляет и возвращает значение производной функции $f(x) = \ln x$ в точке x

`double df2 (double x)`

Функция вычисляет и возвращает значение производной функции $f(x) = -2x + 14$ в точке x

`double df3 (double x)`

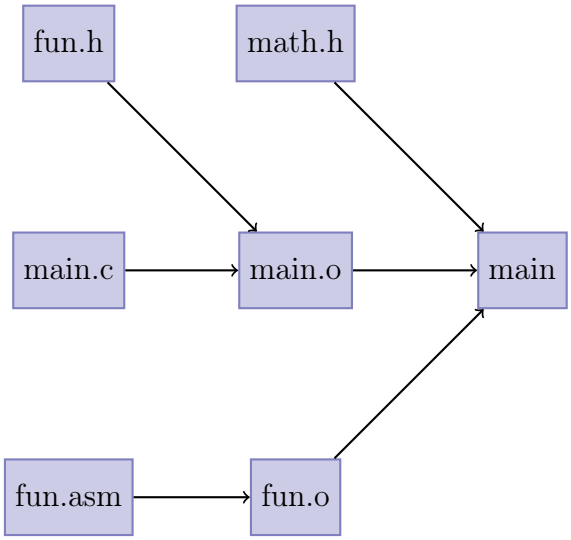
Функция вычисляет и возвращает значение производной функции $f(x) = \frac{1}{2-x} + 6$ в точке x

`double zero (double x)`

Функция всегда возвращает ноль. Это функция была необходима для проведения 3 тестов фукнции root, где корень можно было посчитать аналитически с высокой точностью.

Сборка программы (Make-файл)

Зависимости между модулями программы:



Текст Makefile:

```
default:
	@ echo This is the help
	@ echo all - build project
	@ echo clean - remove all objects
all: main.c fun.asm
	@ gcc -Wall -m32 -std=c99 -c -o main.o main.c
	@ nasm -Wall -f elf32 -o fun.o fun.asm
	@ gcc -Wall -m32 -o main main.o fun.o -lm
clean:
	@ rm main.o
	@ rm fun.o
```

Отладка программы, тестирование функций

Тестирование проводилось с помощью функий `test_root` и `test_integral`. Тесты метода Ньютона:

1. Входные данные: $f(x) = f_1(x) = \ln x$, $g(x) = 0$ (zero(x)), $a = 0.1$, $b = 3$, $\varepsilon = 0.001$.
   Выходные данные:
   Root approximately equal to 0.999999
   Iterations = 5
   Результат аналитических вычислений root = 1

2. Входные данные: $f(x) = f_2(x) = -2x+14$, $g(x) = 0$ (zero(x)), $a = 5$, $b = 10$, $\varepsilon = 0.001$.
   Выходные данные:
   Root approximately equal to 7.000000
   Iterations = 1 Результат аналитических вычислений root = 7

3. Входные данные: $f(x) = f_3(x) = \frac{1}{2-x}+6$, $g(x) = 0$ (zero(x)), $a = 2.01$, $b = 3$, $\varepsilon = 0.00001$.
   Выходные данные:
   Root approximately equal to 2.166667
   Iterations = 8
   Результат аналитических вычислений root = $\frac{13}{6} = 2.1(6) \approx 2.166667$

Тесты метода Симпсона:

1. Входные данные: $f(x) = f_1(x)$, $a = 1$, $b = 11$, $\varepsilon = 0.1$.
   Выходные данные: Integral is approximately equal to 16.376808.
   Результат аналитических вычислений:
   $\int\limits_1^{11} \ln x dx = 11 \ln 11 - 10 \approx 16.376848001$

2. Входные данные: $f(x) = f_2(x)$, $a = 0$, $b = 10$, $\varepsilon = 1$.
   Выходные данные: Integral is approximately equal to 40.000000.
   Результат аналитических вычислений:
   $\int\limits_0^{10} -2x + 14 dx = 40$

3. Входные данные: $f(x) = f_3(x)$, $a = 3$, $b = 5$, $\varepsilon = 0.001$.
   Выходные данные: Integral is approximately equal to 10.901388.
   Результат аналитических вычислений:
   $\int\limits_3^5 \frac{1}{2-x} + 6 dx = 12 - \ln 3 \approx 10.9013877113319$

Программа на Си и на Ассемблере

Программа была написана на Си и на Ассемблере. Исхдные тексты программы имеются в архиве, который приложен к отчёту.

Анализ допущенных ошибок

При написание Makefile возникла ошибка при выполнении цели all. Проблема была в том, что в зависимостях был указан только файл main.c. Исправление: all: main.c fun.asm.

Список литературы

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