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 11 / метод хорд / формула прямоугольников

Выполнил:
студент 106 группы
Синюков М. В.

Преподаватель:
Корухова Л. С.
Манушин Д. В
Соловьев М. А.

Москва
2020

Содержание

Постановка задачи

С точностью $\varepsilon = 0.001$ вычислить площадь плоской фигуры, ограниченой 3 кривыми, уравнения которых y=F1(x) , y=F2(x) , y=F3(x) задаются в текстовом виде на этапе сборки программы. Необходимо разработать 2 программы: основную — для вычисления и вспомогательную — для построения исполняемого кода, вычисляющего значения функций. С точностью $\varepsilon_1 = 0.0001$ вычислить абсциссы точек пересечения кривых, используя 2 метода приближенного решения уравнения $\mathrm{F(x)} = 0$: метод хорд (секущих) и метод касательных (Ньютона). Отрезки для поиска пересечения задаются вместе с уравнениями. Требуется представить площадь заданной фигуры как алгебраическую сумму определенных интегралов и вычислить эти интегралы с некоторой точностью $\varepsilon_2 = 0.0001$ по квадратурной формуле прямоугольников.

Математическое обоснование

Метод касательных [2] На каждой итерации $x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$. [1.1]

Метод хорд. [3] Итерационная формула для метода хорд получается при подстановке в [1.1] метода касательных выражения для приближенного вычисления производной функции в точке:

$x^{k+1} = \frac{a*f(b)-b*f(a)}{F(b)-F(a)}$, где в зависимости от знака произведения первой и второй производных на отрезке, выбирается a=x^k (случай положительной производной) или b=x^k (случай отрицательной производной)

Оценка точности Пусть функции ограничены значением: 10.

A_i-вычислителенный интеграл для функции F_i.
S_i–Истинное значение интеграла для функции F_i. Пределы истинных интервалов обозначи a_i
V_i–Истинное значение для интеграла функции F_i на вычесленных интервалах.Пределы вычесленных интервалов обозначим b_i
Погрешность вычесления площади интеграла: $\varepsilon = |A_1+A_2+A_3\text{-}S_1\text{-}S_2\text{-}S_3| \leq |A_1|+A_2+A_3\text{-}S_1\text{-}S_2\text{-}S_3| \leq \sum\limits_{i=1}^{3} |A_i-S_i| \leq \sum\limits_{i=1}^{3} |A_i-V_i+V_i-S_i| \leq \sum\limits_{i=1}^{3} |A_i-V_i| + \sum\limits_{i=1}^{3} |V_i-S_i| \leq \sum\limits_{i=1}^{2} \int_{a_i}^{b_i} F_1$
$+ \sum\limits_{i=1,3} \int_{a_i}^{b_i} F_2 + \sum\limits_{i=2}^{3} \int_{a_i}^{b_i} F_3 + 3 * \varepsilon_2 \leq 6 * \varepsilon_1 * max_{i \in 1,2,3}(max_{x \in [a_1, a_3]}(|F_i(x))|) + 3 * \varepsilon_2$
$\leq 60* \varepsilon_2 + 3 * \varepsilon_1$
Следовательно ε_1=0.00001 и ε_2= 0.00001 будет достаточно , чтобы гарантировать точность вычислений 0.001

Для вычисления площади фигруры на пересечении кривых варианта 8 функция F1(x)= e^x +2 была разложена в ряд Тейлора, взяты эллемент суммы до 7-го порядка:
F(x)= 3 + x+ $x^2/2$+ $x^3/6$ + $x^4/24$ + $x^5/120$ +$x^6/720$ + $x^7/5040$
Остальные функции были взяты в первозданном виде, без приближения многочленами.
F2(x)=2x-8
F3(x)= -5/x

Второй набор данных для тестирования содержал функции:
F1(x)= $cos(x)$ + 2
F2(x)= $x^2/2$
F3(x)=2.5

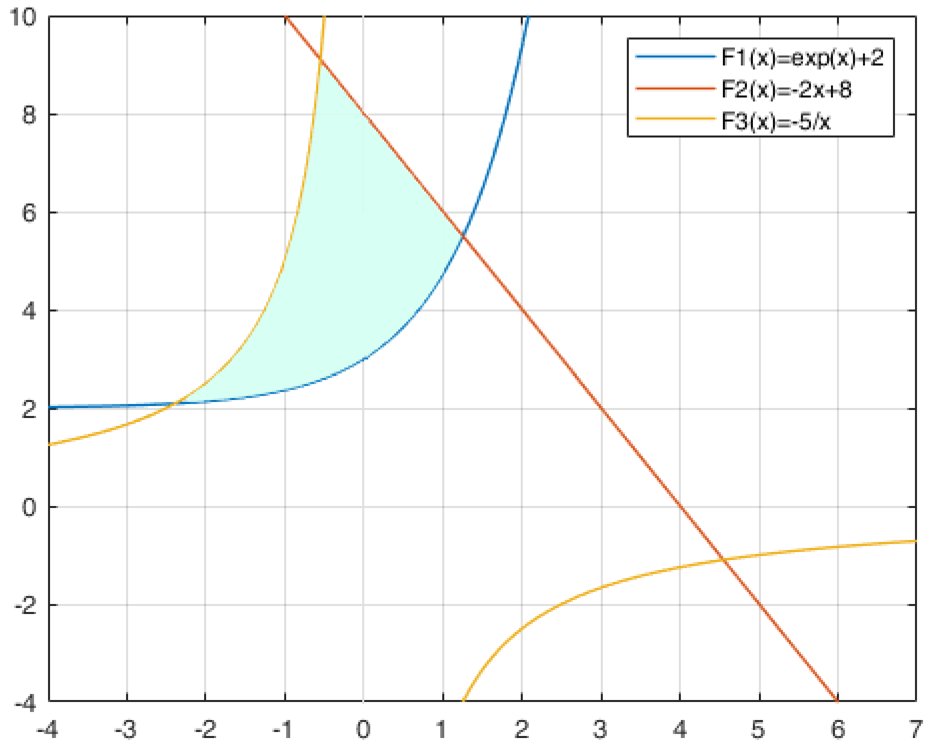


Рис. 1: Плоская фигура, ограниченная графиками уравнений 1-го набора

Результаты экспериментов

В данном разделе необходимо провести результаты проведенных вычислений: координаты точек пересечения (таблица 4) и площадь полученной фигуры.

Кривые	x
y= e^x + 2 и y = -5/x	-0.549516
y=e^x +2 и y = -2*x +8	1.2517579314
y $= -2*x+8/2$ и y$=-5/x$	4.54950975

Таблица 1: Координаты точек пересечения кривых 1-го набора

Площадь фигуры 1: 9.806974

Площадь фигуры 2: 0.460531

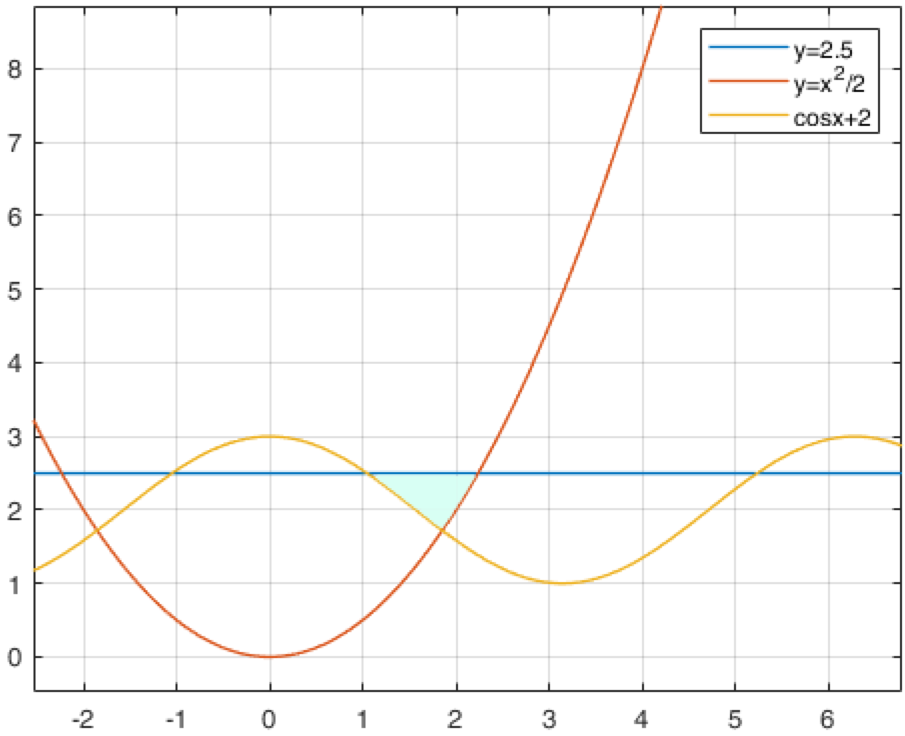


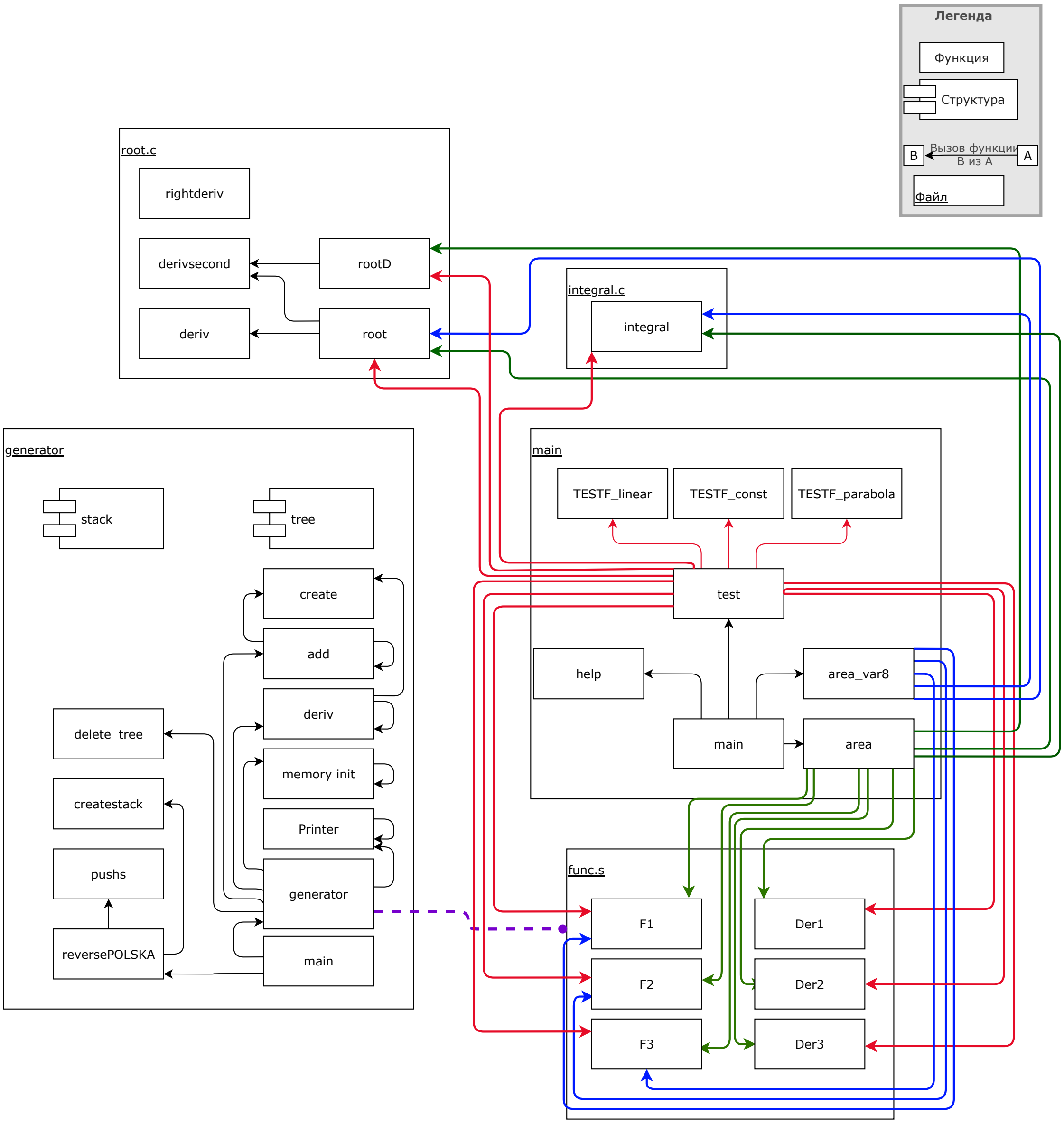
Рис. 2: Плоская фигура, ограниченная графиками уравнений 2-го набора

Кривые	x
y= $cos(x) + 2$ и y $= 2.5$	1.047193
y=$x^2/2$ и y $= 2.5$	2.236065
y $= x^2/2$ и y=$cos(x)$	1.854678

Таблица 2: Координаты точек пересечения кривых 2-го набора

Структура программы и спецификация функций

Рис. 3: Диаграмма связи функций и модулей



Generator.c

- struct stack *createstack(char sign,double key)
  Выделить память под стек, поместив на вершину эллемент с ключём key, знаком sign

- struct stack *pushs(char sign,double key,struct stack *old)
  Положить эллемент на врешину стека, заданного последним параметром. Поля эллемента заданы первыми 2-мя аргументами

- FILE *reversePOLSKA(FILE * input)
  Преобразование обратной польской записи в прямую. Возвращаемое значение - поток с преобразованной формулой. Аргументом является поток с обратной польской записью

- struct tree *create(int sign, double key)
  Выделить память под дерево из одного эллемента с ключём key

- void delete_tree(struct tree *tree)
  Отчистить память от дерева, рекурсивно удаляя потомков.

- struct tree *add()
  Построить дерево операций из польской записи в потоке FILE *inFile;

- struct tree *deriv(struct tree *tree)
  Построить дерево операций производной функции, дерево которой передано в качестве аргумента

- void memory_intit(struct tree *tree)
  Вывести в FILE *pFile ассемблерный листинг резервирования памяти под дерево tree

- void Printer(struct tree *tree)
  Вывод в FILE *pFile ассемблерный листинг выражения , записанного в дереве tree

- void generator()
  Вывод в в FILE *pFile ассемблерного листинга функций 3 функций из файла *inFile и их производных

- int main(int argc, char **argv)
  Подготовка к вызову generator, вызов generator. Включающие в себя переопределение входного потока *inFile в соответствии с переданным параметром – названием файла и запись названия файла в filename для дальнейшего использования в main.c

main.c

- double TESTF_const(double x)
  Функция для тестирования работы функций корня и интеграла: y=1

- double TESTF_linear(double x)
  Функция для тестирования работы функций корня и интеграла: y=x

- double TESTF_parabola(double x)
  Функция для тестирования работы функций корня и интеграла: y=x**2

- void help()
  Вывод help

- void test(_Bool OPTION_TANGENT)
  Произведение тестирования для функций корня и интеграла с возможностью выбора метода вычисления корнея

- double area_var8()
  Вычисления площади фигуры,ограниченной функциями из варианта 8.При условии нахождения в func.s листинга функций варианта.

- double area(double interval1,double interval2, _Bool OPTION_TANGENT)
  Вычисление площади фигуры, ограниченной функциями из func.s на интервале (a,b), метод вычисления корня задаётся последним параметром

- int main(int argc, char **argv)
  Вызов функций на основе переданных аргументов

root.c

- double rightderiv(double x, double (*F)(double)
  Правая производная фунции в точке

- double deriv(double x, double (*F)(double))
  Приближенно вычесленная производная F в точке x

- double derivsecond(double x, double (*F)(double))
  Приближенно вычесленная вторая производная F в точке x

- double root(double (*f)(double), double (*g)(double), double a, double b, double eps)
  Вычисление корня f(x)=g(x) на отрезке (a,b) с точностью eps методом хорд

- double rootD(double (*f)(double), double (*g)(double), double a, double b, double eps,double (*d1)(double),double (*d2)(double))
  Вычисление корня f(x)=g(x) на отрезке (a,b) с точностью eps методом Ньютона , используя заданные в явном виде производные f'(x) , g'(x). Производные указаны последними 2-я аргументами

integral.c

- integral(double (*f)(double), double a, double b, double eps)
  Вычисление интеграла функции f на отрезке (a,b) с точностью eps

Сборка программы (Make-файл)

Текст Make-файла:

```
COMPILER=gcc
OPT=-O2 -std=gnu99 -m32 -lm
SPEC_FILE=in.txt
TANGENT= -D
CHORD= method1

.PHONY: all clean help test

all: main

clean:
rm -rf main.o integral.o root.o main
rm -rf func.o generator.o

generator: generator.c
$(COMPILER) generator.c -o generator $(OPT)
./generator $(SPEC_FILE)

func.o: func.s
nasm -DUNIX -Werror -f elf -o func.o func.s

func.s: generator
./generator $(SPEC_FILE)

integral.o: integral.c
$(COMPILER) integral.c -c -o integral.o $(OPT)

root.o: root.c
$(COMPILER) root.c -c -o root.o $(OPT)

main.o: main.c
$(COMPILER) main.c -c -o main.o $(OPT)

main: main.o integral.o root.o func.o
$(COMPILER) -o main main.o integral.o root.o func.o  $(OPT)

help: main
./main -help

test: main
./generator tests.txt
./main -test -axis -iter
```

```
area: main
./main -area

generate: generator
./generator $(SPEC_FILE)

{TANGENT}: main
./main -D -area

{CHORD}:
./main -area
```

Отладка программы, тестирование функций

Кривые	Метод хорд	Метод касательных	Истинное значение	f'
y=x^2 и y = x	0.999992	1	1	2x и 1
y =x^3 и y= 0	0.00008	0.00002	0	3x и 0
y =$-5/x$ и y=-2 * x + 8	4.549509	4.545455	4.54950975	$5/x^2$ и -2

Таблица 3: Тестирование функций для вычисления точек пересечения кривых

Аналитическое обоснование:
1)

$$x = x^2$$

$$x * (x - 1) = 0$$

$$x = 1$$

2)

$$x^3 = 0$$

$$x = 0$$

3)

$$-5/x = -2x + 8$$

$$-2x^2 + 8x + 5 = 0$$

$$-2x^2 + 8x + 5 = 0$$

$$x \approx 4.54950975$$

Кривая	Отрезок	Интеграл	Истинное значение
sin x / x	(-1 1)	1.892166	1.892166
$(sinx)^2$	(- π, π)	3.141592	π
sin(x)	(-4,4)	0.000000	0
f(x) = x^2	(-5,1)	42.000220	42
f(x) = 1	(0,25)	25.0000	25
(x) = x	(0,5)	12.499981	12.5

Таблица 4: Тестирование функции интеграла

Аналитическое обоснование[3]:
1)

$$\int_{-\pi}^{\pi} (sinx)^2 dx = \int_{-\pi}^{\pi} (1/2 - 1/2 * cos2x) dx = \left(\frac{x - sin(2x)/2}{2} \right) \Big|_{-\pi}^{\pi} = \pi.$$

2)
$\int_{-4}^{4} (sinx) dx$ =[осевая симметрия sin относительно 0]=0 3)

$$\int_{-5}^{1} x^2 dx = \left(\frac{x^3}{3} \right) \Big|_{-5}^{1} = 5^3/3 + 1/3 = 42.$$

4)

$$\int_0^{25} 1dx = (x)\Big|_0^{25} = 25.$$

5)

$$\int_0^5 xdx = \left(x^2/2\right)\Big|_0^5 = 12.5.$$

Программа на Си и на Ассемблере

Исходные тексты программы имеются в архиве, приложеном к отчету.

Анализ допущенных ошибок

1. Была допущена ошибка с переопределением функций F1-F3. Исправленно изменением
   extern F1 на global F1
   extern F2 на global F2
   extern F3 на global F3
   в модуле func.s

2. Ошибка в построении дерева производной. При вызвове функции нахождения производной сложной функции от поддерева, поддерево основной функции модифицировалось. Исправлено выделением памяти под эллемент дерева производной проведением изменений в нём, без модификации основного дерева.

3. Ошибка в построении дерева производной. При вызвове функции нахождения производной сложной функции от поддерева, поддерево основной функции модифицировалось. Исправлено выделением памяти под эллемент дерева производной проведением изменений в нём, без модификации основного дерева.

Список литературы

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

[2] Самаский А. А., Гулин А. В. Численные Методы.Учеб. пособие для вузов. — М.:Наука, 1989.

[3] Костомаров Д.П., Фаворский А.П. Вводные лекции по численным методам: Учеб. пособие. – М.: Логос, 2004