МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В. ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ №1

**«ВЫЧИСЛЕНИЕ КОРНЕЙ УРАВНЕНИЙ**

**И ОПРЕДЕЛЕННЫХ ИНТЕГРАЛОВ»**

**Вариант 5-2-1**

Выполнил: студент 102 группы

Петухов Иван Андреевич

Проверил: преподаватель

Сенюкова Ольга Викторовна

МОСКВА

2014

СОДЕРЖАНИЕ

1. Постановка задачи 2
2. Математическое обоснование 3
3. Структура программы на Си, спецификации функций 4
4. Отладка программы, тестирование функций 5
5. Программа на Си 6
6. Полученные результаты 7
7. Анализ ошибок 8

**ПОСТАНОВКА ЗАДАЧИ**

С заданной точностью ɛ вычислить площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых

, , ;

При решении задачи необходимо:

- с некоторой точностью eps1 вычислить абсциссы точек пересечения кривых, используя метод хорд(секущих) решения уравнения F(x)=0

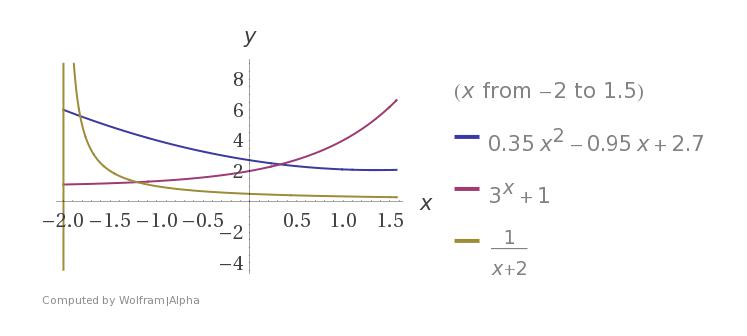
- представить площадь заданной фигуры как алгебраическую сумму определенных интегралов и вычислить эти интегралы с некоторой точностью eps2 методом прямоугольников

**МАТЕМАТИЧЕСКОЕ ОБОСНОВАНИЕ**

Исследуемые функции:

, , ;

Графики функций:



Для нахождения координаты точки пересечения функций f(x) и g(x) используется функция F(x) = f(x)-g(x).

Были рассмотрены 3 функции F:

;

Выбираем точки, которые будут являться границами а и b для расчёта корней. Глядя на график, возьмем а = -1.99, b = 1.

Для данной задачи возьмем для наибольшей точности eps1 = 0.0005.

**СТРУКТУРА ПРОГРАММЫ НА СИ**

Программа на Си состоит из главной функции *int main(void)*, а также

- *double deltaf(double (\*f)(double), double (\*g)(double), double x)*, рассчитывающая F(x)

- *double root\_hord(double(\*f)(double), double(\*g)(double), double a, double b, double eps1)*, вычисляющая корни методом хорд.

- *double integral (double (\*f)( double), double a, double b, double eps2)*, корректировка границ интеграла

- *double integral\_n (double (\*f)( double), double a, double b, int n),* вычисляющая интеграл методом прямоугольников

**ОТЛАДКА ПРОГРАММЫ, ТЕСТИРОВАНИЕ ФУКЦИЙ**

Для тестрирования функции *double root\_hord(double(\*f)(double), double(\*g)(double), double a, double b, double eps1)* использовались следующие функции:

;

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Границы | Ожидаемый р-т | Полученный рез-т |
|  | [1, 3] | 2 | 2.000 |
|  | [0; 3] | 2 | 2.000 |
|  | [0.1; 1] | 0.5 | 0.500 |

Для тестрирования функции *double integral\_n (double (\*f)( double), double a, double b, int n)* использовались следующие функции:

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Границы | Ожидаемый р-т | Полученный рез-т |
|  | [-1,1] | 0 | 0.000 |
|  | [1,e] | 1 | 1.000 |

**ПРОГРАММА НА СИ**

1. Сами выражения

double f1(double x)

{

return (0.35 \* x \* x - 0.95 \* x + 2.7);

}

double f2(double x)

{

return (pow(3.0, x) + 1);

}

double f3(double x)

{

return (1 / (x + 2));

}

1. Функция *root*

double deltaf(double (\*f)(double), double (\*g)(double), double x)

{

return ((\*f)(x) - (\*g)(x));

}

double root\_hord(double(\*f)(double), double(\*g)(double), double a, double b, double eps1)

{

double deltaf\_a;

deltaf\_a = deltaf(f, g, a);

double deltaf\_b;

deltaf\_b = deltaf(f, g, b);

double c;

c = (a \* deltaf\_b - b \* deltaf\_a) / (deltaf\_b - deltaf\_a);

double deltaf\_c;

deltaf\_c = deltaf(f, g, c);

if (((deltaf\_c \* deltaf(f, g, (c + eps1/2))) < 0) || ((deltaf\_c \* deltaf(f, g, (c - eps1/2))) < 0))

return c;

if (deltaf\_a < 0 && deltaf(f, g, ((a + b) / 2)) < ((deltaf\_a + deltaf\_b) / 2))

a = c;

else

if (deltaf\_a > 0 && deltaf(f, g, (a + b) / 2) > ((deltaf\_a + deltaf\_b) / 2))

a = c;

else

b = c;

return root\_hord(f, g, a, b, eps1);

}

1. Функции *integral*

double integral\_n (double (\*f)( double), double a, double b, int n)

{

double result;

result = 0.0;

double h;

h = (b - a) / (double) n;

int i;

for(i =0; i< n; i++)

{

if((a + (i + 0.5)\*h) >= b)

break;

result += f (a + (i + 0.5) \* h);

}

result \*= h;

return result;

}

double integral (double (\*f)( double), double a, double b, double eps2)

{

if (a > b)

{

double t;

t = a;

a = b;

b = t;

}

int n = 20;

double I0, In, delta;

do

{

I0 = integral\_n (f, a, b, n);

n = 2 \* n;

In = integral\_n (f, a, b, n);

delta = (I0 - In) / 3.0;

if (delta < 0)

delta = -delta;

} while (delta > eps2);

return In;

}

1. Функция *main*

int main(void)

{

double eps1 = 0.0005;

double eps2 = 0.0005;

double x1, x2, x3, s1, s2, s3, S;

double a, b;

scanf("%lf %lf", &a, &b);

x1 = root\_hord(f1, f2, a, b, eps1);

x2 = root\_hord(f1, f3, a, b, eps1);

x3 = root\_hord(f2, f3, a, b, eps1);

printf("x1 = %.3lf x2 = %.3lf x3 = %.3lf\n", x1, x2, x3);

s1 = integral(f1, x2, x1, eps2);

s2 = integral(f2, x3, x1, eps2);

s3 = integral(f3, x2, x3, eps2);

S = s1-s2-s3;

printf("ploshad = %.3lf", S);

return 0;

}

**ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ**

Полученные точки пересечения:

Вычисленное программой значение интеграла равно:

**Ploshad** = 3.948

**АНАЛИЗ ОШИБОК**

В процессе написания данной программы я столкнулся с некоторыми ошибками, в последствии мною исправленные:

* Некорректная точность вычислений
* Отсутствие корректировки границ при а > b
* Отсутствие определения подходящего случая для вычисления функции *root.* (определение положения функции выше/ниже оси абсцисс

**СПИСОК ЛИТЕРАТУРЫ**

1. Ильин В.А., Садовничий В.А., Сендов Бл.Х. Математический анализ. Т.1 – М.: Наука, 1985