**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра программной инженерии**

Лабораторная работа 9

По дисциплине «Основы алгоритмизации и программирования»

На тему «Решение инженерных задач на основе циклических программ»

Выполнил:

Студент 1 курса 10 группы

Мамонько Денис Александрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант 7**

Задание 5**.** В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом ***трапеций*** и методом ***парабол***. Для всех вариантов принять **n** = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

**Метод трапеции:**

Код программы:

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

int main()//объявление функции

{

setlocale(LC\_ALL, "RU");//подключаем русский язык

float n = 200, a = 5, b = 11;//объявляем переменные

double h = (b - a) / n;

double x = a;

double s = 0;

while (x < (b - h)) {//пишем условие, при котором будет осуществляться метод трапеции

s += h \* (exp(x) + 2 + exp(x + h) + 2)/2;//формула для метода трапеции

x = x + h;

}

cout << "Метод трапеции: " << s;//вывод s

}

Результат программы:



**Метод параболы:**

Код программы:

#include <iostream>//подключаем библиотеку

using namespace std;//чтобы не писать каждый раз std

int main()//объявляем функцию

{

setlocale(LC\_ALL, "RUSSIAN");//подключаем кириллицу

float n = 200, a = 5, b = 11, s1=0, s2=0, i=1, x, h, z;//объявляем переменные

h = (b - a) / (2\*n);//формула h

x = a + 2 \* h;//формула x

while (i<n)//пишем условие, при котором будет осуществляться метод парабол

{

s1 = s1 + (exp(x) + 2);

x = x + h;

s2 = s2 + (exp(x) + 2);

x = x + h;

i = i + 1;

}

z = h / 3 \* (exp(a) + 2 + 4 \* (exp(a + h) + 2) + 4 \* s1 + 2 \* s2 + exp(b) + 2);//вычисление значения площади криволинейной трапеции

cout << "Метод параболы: " << z;//вывод метода интегралов/парабол

}

Результат программы:



Задание 6. В соответствии со своим вариантом найти отрезок (значения **a** и **b**), который содержит один корень, ***отделив корни*** уравнения ***графическим*** методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков. Написать программу вычисления корня уравнения методом ***дихотомии***. Точность вычислений принять равной **e** = 0,0001 для всех вариантов. Найти корень уравнения с помощью приложения Excel. Сравнить результаты.

**Метод дихотомии:**

**Отрезок (0;1)**

Код программы:

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

//объявление функции func(x), которая представляет уравнение для поиска корня

double func(double x) {

return pow(x, 3) + 3 \* x - 1;//уравнение, для которого ищется корень

}

int main()

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

double a, b, e;//объявляем переменные

cout << "Введите a: ";//ввод a

cin >> a;

cout<< "Введите b: ";//ввод b

cin >> b;

cout << "Введите e: ";//ввод e

cin >> e;

double x;//объявление текущего приближения корня

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;//вычисление середины интервала выбранного отрезка

if (func(x) \* func(a) <= 0)

b = x;//если функция имеет разные знаки на концах интервала, то объявляем правую границу

else

a = x;//иначе левую

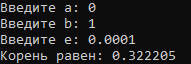
}

x = (a + b) / 2;//финальное приближение корня

cout << "Корень равен: " << x;//вывод искомого корня

}

Результат:



**Корень в Excel:**

****

**Дополнительные задания:**

**Вариант 12**

**Метод трапеции:**

Код программы:

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

int main()//объявление функции

{

setlocale(LC\_ALL, "RU");//подключаем русский язык

float n = 200, a = 5, b = 11;//объявляем переменные

double h = (b - a) / n;

double x = a;

double s = 0;

while (x < (b - h)) {//пишем условие, при котором будет осуществляться метод трапеции

s += h \* (exp(x) + 6 + exp(x + h) + 6)/2;//формула для метода трапеции

x = x + h;

}

cout << "Метод трапеции: " << s;//вывод s

}

Результат:



**Метод параболы:**

Код программы:

#include <iostream>//подключаем библиотеку

using namespace std;//чтобы не писать каждый раз std

int main()//объявляем функцию

{

setlocale(LC\_ALL, "RUSSIAN");//подключаем кириллицу

float n = 200, a = 5, b = 11, s1=0, s2=0, i=1, x, h, z;//объявляем переменные

h = (b - a) / (2\*n);//формула h

x = a + 2 \* h;//формула x

while (i<n)//пишем условие, при котором будет осуществляться метод парабол

{

s1 = s1 + (exp(x) + 6);

x = x + h;

s2 = s2 + (exp(x) + 6);

x = x + h;

i = i + 1;

}

z = h / 3 \* (exp(a) + 6 + 4 \* (exp(a + h) + 6) + 4 \* s1 + 2 \* s2 + exp(b) + 6);//вычисление значения площади криволинейной трапеции

cout << "Метод параболы: " << z;//вывод метода интегралов/парабол

}

Результат программы:



**Метод дихотомии:**

**Отрезок (0;4)**

Код программы:

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

//объявление функции func(x), которая представляет уравнение для поиска корня

double func(double x) {

return pow(x, 3) + 2 \* x - 4;

}//уравнение, для которого ищется корень

int main()

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

double a, b, e;//объявляем переменные

cout << "Введите a: ";//ввод a

cin >> a;

cout<< "Введите b: ";//ввод b

cin >> b;

cout << "Введите e: ";//ввод e

cin >> e;

double x;//объявление текущего приближения корня

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;//вычисление середины интервала выбранного отрезка

if (func(x) \* func(a) <= 0)

b = x;//если функция имеет разные знаки на концах интервала, то объявляем правую границу

else

a = x;//иначе левую

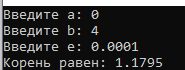
}

x = (a + b) / 2;//финальное приближение корня

cout << "Корень равен: " << x;//вывод искомого корня

}

Результат:



**Корень в Excel:**

****

**Вариант 1**

**Метод трапеции:**

Код программы:

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

int main()//объявление функции

{

setlocale(LC\_ALL, "RU");//подключаем русский язык

float n = 200, a = 1, b = 3;//объявляем переменные

double h = (b - a) / n;

double x = a;

double s = 0;

while (x < (b - h)) {//пишем условие, при котором будет осуществляться метод трапеции

s += h \* (pow(x,3)-3+pow(x+h,3)-3)/2;//формула для метода трапеции

x = x + h;

}

cout << "Метод трапеции: " << s;//вывод s

}

Результат:



**Метод парабол:**

Код программы:

#include <iostream>//подключаем библиотеку

using namespace std;//чтобы не писать каждый раз std

int main()//объявляем функцию

{

setlocale(LC\_ALL, "RUSSIAN");//подключаем кириллицу

float n = 200, a = 1, b = 3, s1 = 0, s2 = 0, i = 1, x, h, z;//объявляем переменные

h = (b - a) / (2 \* n);//формула h

x = a + 2 \* h;//формула x

while (i < n)//пишем условие, при котором будет осуществляться метод парабол

{

s1 = s1 + (pow(x,3)-3);

x = x + h;

s2 = s2 + (pow(x,3)-3);

x = x + h;

i = i + 1;

}

z = h / 3 \* (pow(x,3)-3+4\*(pow(x+h,3)-3) + 4 \* s1 + 2 \* s2+pow(b,3)-3);//вычисление значения площади криволинейной трапеции

cout << "Метод параболы: " << z;//вывод метода интегралов/парабол

}

Результат программы:



**Метод дихотомии:**

**Отрезок (0;3)**

Код программы:

double func(double x) {

return pow(x, 3) + x - 3;

}//уравнение, для которого ищется корень

int main()

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

double a, b, e;//объявляем переменные

cout << "Введите a: ";//ввод a

cin >> a;

cout<< "Введите b: ";//ввод b

cin >> b;

cout << "Введите e: ";//ввод e

cin >> e;

double x;//объявление текущего приближения корня

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;//вычисление середины интервала выбранного отрезка

if (func(x) \* func(a) <= 0)

b = x;//если функция имеет разные знаки на концах интервала, то объявляем правую границу

else

a = x;//иначе левую

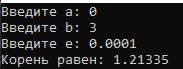
}

x = (a + b) / 2;//финальное приближение корня

cout << "Корень равен: " << x;//вывод искомого корня

}

Результат:



**Корень в Excel:**

****

**Вариант 3**

**Метод трапеции:**

Код программы:

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

int main()//объявление функции

{

setlocale(LC\_ALL, "RU");//подключаем русский язык

float n = 200, a = 1, b = 6;//объявляем переменные

double h = (b - a) / n;

double x = a;

double s = 0;

while (x < (b - h)) {//пишем условие, при котором будет осуществляться метод трапеции

s += h \* (1+pow(x,3)+1+pow(x+h,3))/2;//формула для метода трапеции

x = x + h;

}

cout << "Метод трапеции: " << s;//вывод s

}

Результат программы:



**Метод парабол:**

Код программы:

#include <iostream>//подключаем библиотеку

using namespace std;//чтобы не писать каждый раз std

int main()//объявляем функцию

{

setlocale(LC\_ALL, "RUSSIAN");//подключаем кириллицу

float n = 200, a = 1, b = 6, s1 = 0, s2 = 0, i = 1, x, h, z;//объявляем переменные

h = (b - a) / (2 \* n);//формула h

x = a + 2 \* h;//формула x

while (i < n)//пишем условие, при котором будет осуществляться метод парабол

{

s1 = s1 + (1+pow(x,3));

x = x + h;

s2 = s2 + (1 + pow(x, 3));

x = x + h;

i = i + 1;

}

z = h / 3 \* (1 + pow(a, 3) + 4 \* (1+pow(a+h,3)) + 4 \* s1 + 2 \* s2 + 1 + pow(b,3));//вычисление значения площади криволинейной трапеции

cout << "Метод параболы: " << z;//вывод метода интегралов/парабол

}

Результат:



**Метод дихотомии:**

**Отрезок (0;1)**

Код программы:

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

//объявление функции func(x), которая представляет уравнение для поиска корня

double func(double x) {

return pow(x, 3) + 2 \* x - 1;

}//уравнение, для которого ищется корень

int main()

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

double a, b, e;//объявляем переменные

cout << "Введите a: ";//ввод a

cin >> a;

cout<< "Введите b: ";//ввод b

cin >> b;

cout << "Введите e: ";//ввод e

cin >> e;

double x;//объявление текущего приближения корня

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;//вычисление середины интервала выбранного отрезка

if (func(x) \* func(a) <= 0)

b = x;//если функция имеет разные знаки на концах интервала, то объявляем правую границу

else

a = x;//иначе левую

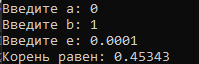
}

x = (a + b) / 2;//финальное приближение корня

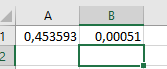
cout << "Корень равен: " << x;//вывод искомого корня

}

Результат программы:



**Код в Excel:**

****