**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра ПИ**

Лабораторная работа № 9

По дисциплине «Основы алгоритмизации и программирования»

На тему «Решение инженерных задач на основе циклических программ»

Выполнил:  
Студент 1 курса, 10 группа  
Короткевич Артём Сергеевич  
Преподаватель: асс. Харланович А.В

2024, Минск

**Вариант 8**

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**#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float n = 200, a = 0, b = 3;

double h = (b - a) / n;

double x = a;

double s = 0;

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

{

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

x = x + h;

}

cout << "Метод трапеции: " << s;

}

**Результат**



**Метод параболы:**

**Код**  
#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float n = 200, a = 0, b = 3, s1 = 0, s2 = 0, i = 1, x, h, p;

h = (b - a) / (2 \* n);

x = a + 2 \* h;

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

{

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

x = x + h;

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

x = x + h;

i = i + 1;

}

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

cout << "Метод параболы: " << p;//вывод метода параболы

}

**Результат**



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;2)**

**Код**#include <iostream>

using namespace std;

double func(double x) //объявляем функцию func(x), которая представляет уравнение для поиска корня

{

return pow(x, 3) + x - 4; //уравнение, по которому ищется корень

}

int main()

{

setlocale(LC\_ALL, "Russian");

double a, b, e;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

double x; //объявляем текущее приближения корня

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

x = (a + b) / 2; //вычисляем середину интервала выбранного отрезка

if (func(x) \* func(a) <= 0) //если функция имеет разные знаки на концах интервала, то объявляем правую границу

b = x;

else

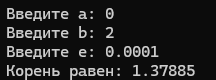
a = x; //иначе левую

}

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

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

}

**Результат:**

**Корень в Excel:**

****

**Дополнительные задания:**

**Вариант 13**

**Метод трапеции:**

**Код**#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float n = 200, a = 8, b = 14;

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

x = x + h;

}

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

}

**Результат:**



**Метод параболы:**

**Код**#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian);

float n = 200, a = 8, b = 14, s1 = 0, s2 = 0, i = 1, x, h, p;

h = (b - a) / (2 \* n);

x = a + 2 \* h;

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

{

s1 = s1 + (2 + pow(x, 3));

x = x + h;

s2 = s2 + (2 + pow(x, 3));

x = x + h;

i = i + 1;

}

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

cout << "Метод параболы: " << p; //вывод метода параболы

}

**Результат**



**Метод дихотомии:**

**Отрезок (0;3)**

**Код**#include <iostream>

using namespace std;

double func(double x) //объявляем функцию func(x), которая представляет уравнение для поиска корня

{

return exp(x) + x - 4; //уравнение, для которого ищется корень

}

int main()

{

setlocale(LC\_ALL, "Russian");

double a, b, e;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

double x; //объявляем текущее приближения корня

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

x = (a + b) / 2; //вычисляем 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

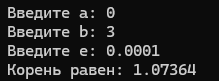
}

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

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

}

**Результат**



**Корень в Excel:**

****

**Вариант 2**

**Метод трапеции:**

**Код**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float n = 200, a = 4, b = 7;

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(cos(x), 3) + pow(cos(x + h), 3)) / 2; //формула для метода трапеции

x = x + h;

}

cout << "Метод трапеции: " << s;

}

**Результат:**



**Метод парабол:**

**Код**#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 = 4, b = 7, s1 = 0, s2 = 0, i = 1, x, h, p;//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;

}

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

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

}

**Результат**



**Метод дихотомии:**

**Отрезок (0;2)**

Код программы:

#include <iostream>

using namespace std;

double func(double x) //объявляем функцию func(x), которая представляет уравнение для поиска корня

{

return cos(x) + x - 7; //уравнение, для которого ищется корень

}

int main()

{

setlocale(LC\_ALL, "Russian");

double a, b, e;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

double x; //объявляем текущее приближения корня

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

x = (a + b) / 2; //вычисляем 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

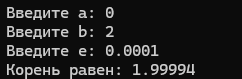
a = x; //иначе левую

}

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

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

}

**Результат:**

**Корень в Excel:**

****

**Вариант 10**

**Метод трапеции:**

**Код**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float n = 200, a = 4, b = 8;

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, 2) - 4 + pow(x + h, 2) - 4) / 2; //формула для метода трапеции

x = x + h;

}

cout << "Метод трапеции: " << s;

}

**Результат**



**Метод парабол:**

**Код**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float n = 200, a = 1, b = 6, s1 = 0, s2 = 0, i = 1, x, h, p;

h = (b - a) / (2 \* n);

x = a + 2 \* h;

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;

}

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

cout << "Метод параболы: " << p; //вывод метода параболы

}

**Результат**



**Метод дихотомии:**

**Отрезок (0;2)**

**Код**#include <iostream>

using namespace std;

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

{

return exp(x) + 2 \* pow(x, 2) - 3; //уравнение, по которому ищется корень

}

int main()

{

setlocale(LC\_ALL, "Russian");

double a, b, e;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

double x; //объявляем текущеее приближения корня

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

x = (a + b) / 2; //вычисляем 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

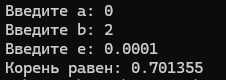
}

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

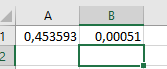
cout << "Корень равен: " << x; //вывод искомого корня

}

**Результат**



**Код в Excel:**

****