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»

Выполнила:

Студентка 1 курса 10 группы

Рублевская Маргарита Владимировна

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант№ 13**

Функция:

2 + x3

Границы отрезков:

a = 8, b = 14

1. 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.

1)Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return 2 + pow(x, 3);

}

double trapezoidalMethod(double a, double b, int n) // Реализуем метод трапеций для вычисления площади криволинейной трапеции.

{

double h = (b - a) / n; // Шаг.

double x = a; // Начальное значение x.

double sum = 0; // Инициализация суммы.

while (x <= (b - h)) {

sum += h \* (f(x) + f(x + h)) / 2; // Площадь трапеции.

x += h; // Увеличение x на шаг

}

return sum; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 8;

double b = 14;

int n = 200;

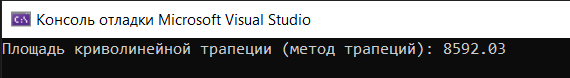
double result = trapezoidalMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода трапеций.

std::cout << "Площадь криволинейной трапеции (метод трапеций): " << result << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



2) Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return 2 + pow(x, 3);

}

double parabolicMethod(double a, double b, int n) // Реализуем метод парабол для вычисления площади криволинейной трапеции.

{

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

double x = a + 2 \* h; // **И**нициализации переменных.

double s1 = 0; // **И**нициализации переменных.

double s2 = 0; // **И**нициализации переменных.

int i = 1; // **И**нициализации переменных.

// Вычисляем площадь, путем суммирования значений функции f(x) на параболических отрезках.

while (i < n) {

s2 += f(x);

x += h;

s1 += f(x);

x += h;

i++;

}

double result = (h / 3) \* (f(a) + f(b) + 4 \* s1 + 2 \* s2); // Вычисление площади криволинейной трапеции по формуле.

return result; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 8;

double b = 14;

int n = 200;

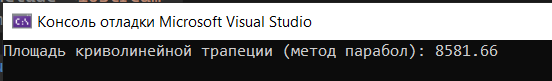
double result = parabolicMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода парабол.

std::cout << "Площадь криволинейной трапеции (метод парабол): " << result << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



2. 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.

Функция:

e x + x − 4

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return exp(x) + x - 4;

}

// Реализуем метод дихотомии.

double findRoot(double a, double b, double e) {

double x;

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

x = (a + b) / 2;

if (f(x) \* f(a) <= 0) {

b = x;

}

else {

a = x;

}

}

return x; // Возврат значения х.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE,"RU"); // Используется для вывода русского текста.

double a, b, e; //Ввод переменных.

// Выводим фразы и значения на экран.

std::cout << "Введите значение a: ";

std::cin >> a;

std::cout << "Введите значение b: ";

std::cin >> b;

std::cout << "Введите точность e: ";

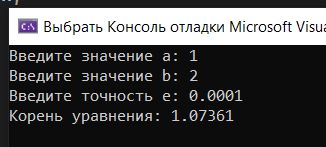
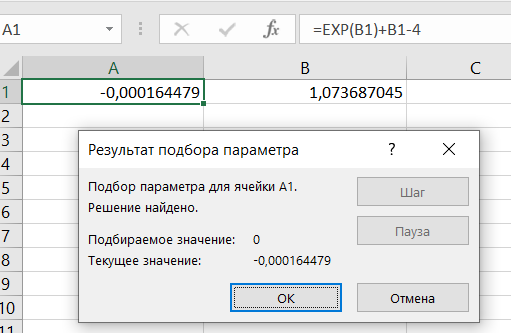
std::cin >> e;

double root = findRoot(a, b, e); // Возвращаем значение корня функции на указанном интервале.

std::cout << "Корень уравнения: " << root << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:

**Дополнительные задания**

**Задание 1) Вариант№ 1**

1.Функция:

x3 – 3

Границы отрезков:

a = 1, b = 3

1)Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 3) - 3;

}

double trapezoidalMethod(double a, double b, int n) // Реализуем метод трапеций для вычисления площади криволинейной трапеции.

{

double h = (b - a) / n; // Шаг.

double x = a; // Начальное значение x.

double sum = 0; // Инициализация суммы.

while (x <= (b - h)) {

sum += h \* (f(x) + f(x + h)) / 2; // Площадь трапеции.

x += h; // Увеличение x на шаг

}

return sum; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 1;

double b = 3;

int n = 200;

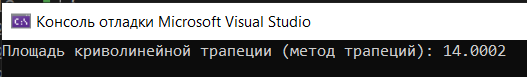
double result = trapezoidalMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода трапеций.

std::cout << "Площадь криволинейной трапеции (метод трапеций): " << result << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



2)Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 3) - 3;

}

double parabolicMethod(double a, double b, int n) // Реализуем метод парабол для вычисления площади криволинейной трапеции.

{

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

double x = a + 2 \* h; // **И**нициализации переменных.

double s1 = 0; // **И**нициализации переменных.

double s2 = 0; // **И**нициализации переменных.

int i = 1; // **И**нициализации переменных.

// Вычисляем площадь, путем суммирования значений функции f(x) на параболических отрезках.

while (i < n) {

s2 += f(x);

x += h;

s1 += f(x);

x += h;

i++;

}

double result = (h / 3) \* (f(a) + f(b) + 4 \* s1 + 2 \* s2); // Вычисление площади криволинейной трапеции по формуле.

return result; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 1;

double b = 3;

int n = 200;

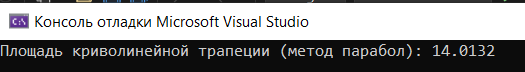
double result = parabolicMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода парабол.

std::cout << "Площадь криволинейной трапеции (метод парабол): " << result << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



2.Функция:

x3 + x – 3

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

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

}

//Реализуем метод дихотомии.

double findRoot(double a, double b, double e) {

double x;

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

x = (a + b) / 2;

if (f(x) \* f(a) <= 0) {

b = x;

}

else {

a = x;

}

}

return x; // Возврат значения х.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE,"RU"); // Используется для вывода русского текста.

double a, b, e; //Ввод переменных.

// Выводим фразы и значения на экран.

std::cout << "Введите значение a: ";

std::cin >> a;

std::cout << "Введите значение b: ";

std::cin >> b;

std::cout << "Введите точность e: ";

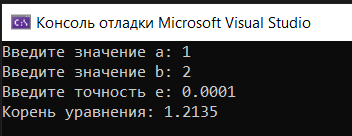
std::cin >> e;

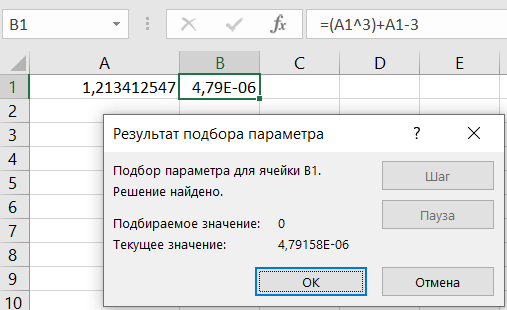
double root = findRoot(a, b, e); // Возвращаем значение корня функции на указанном интервале.

std::cout << "Корень уравнения: " << root << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



**Задание 2) Вариант№ 8**

1. Функция:

x3 – 1

Границы отрезков:

a = 0, b = 3

1) Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 3) - 1;

}

double trapezoidalMethod(double a, double b, int n) // Реализуем метод трапеций для вычисления площади криволинейной трапеции.

{

double h = (b - a) / n; // Шаг.

double x = a; // Начальное значение x.

double sum = 0; // Инициализация суммы.

while (x <= (b - h)) {

sum += h \* (f(x) + f(x + h)) / 2; // Площадь трапеции.

x += h; // Увеличение x на шаг

}

return sum; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 0;

double b = 3;

int n = 200;

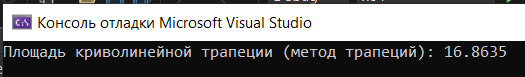
double result = trapezoidalMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода трапеций.

std::cout << "Площадь криволинейной трапеции (метод трапеций): " << result << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



2) Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 3) - 1;

}

double parabolicMethod(double a, double b, int n) // Реализуем метод парабол для вычисления площади криволинейной трапеции.

{

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

double x = a + 2 \* h; // **И**нициализации переменных.

double s1 = 0; // **И**нициализации переменных.

double s2 = 0; // **И**нициализации переменных.

int i = 1; // **И**нициализации переменных.

// Вычисляем площадь, путем суммирования значений функции f(x) на параболических отрезках.

while (i < n) {

s2 += f(x);

x += h;

s1 += f(x);

x += h;

i++;

}

double result = (h / 3) \* (f(a) + f(b) + 4 \* s1 + 2 \* s2); // Вычисление площади криволинейной трапеции по формуле.

return result; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 0;

double b = 3;

int n = 200;

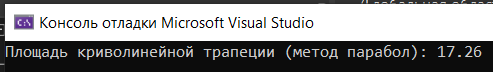
double result = parabolicMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода парабол.

std::cout << "Площадь криволинейной трапеции (метод парабол): " << result << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



2. Функция:

x3 + x – 4

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 3) + x - 4;

}

//Реализуем метод дихотомии.

double findRoot(double a, double b, double e) {

double x;

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

x = (a + b) / 2;

if (f(x) \* f(a) <= 0) {

b = x;

}

else {

a = x;

}

}

return x; // Возврат значения х.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE,"RU"); // Используется для вывода русского текста.

double a, b, e; //Ввод переменных.

// Выводим фразы и значения на экран.

std::cout << "Введите значение a: ";

std::cin >> a;

std::cout << "Введите значение b: ";

std::cin >> b;

std::cout << "Введите точность e: ";

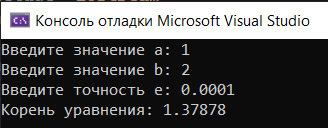
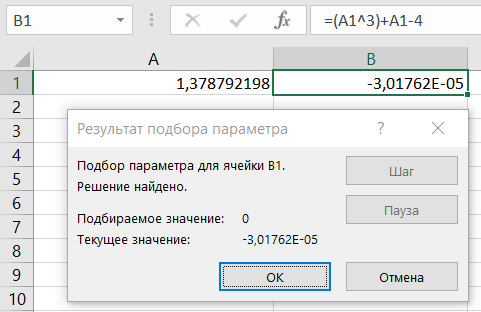
std::cin >> e;

double root = findRoot(a, b, e); // Возвращаем значение корня функции на указанном интервале.

std::cout << "Корень уравнения: " << root << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:

**Задание 3) Вариант№ 16**

1. Функция:

x2 + 1 / x

Границы отрезков:

a = 0, b = 4

1) Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 2) + (1 / x);

}

double trapezoidalMethod(double a, double b, int n) // Реализуем метод трапеций для вычисления площади криволинейной трапеции.

{

double h = (b - a) / n; // Шаг.

double x = a; // Начальное значение x.

double sum = 0; // Инициализация суммы.

while (x <= (b - h)) {

sum += h \* (f(x) + f(x + h)) / 2; // Площадь трапеции.

x += h; // Увеличение x на шаг

}

return sum; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 0;

double b = 4;

int n = 200;

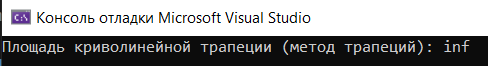
double result = trapezoidalMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода трапеций.

std::cout << "Площадь криволинейной трапеции (метод трапеций): " << result << std::endl; // Выводим фразу и значение на экран.

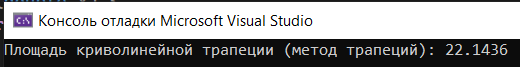
return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



С исправленными данными (a = 1, b = 4):



2) Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 2) + 1 / x;

}

double parabolicMethod(double a, double b, int n) // Реализуем метод парабол для вычисления площади криволинейной трапеции.

{

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

double x = a + 2 \* h; // **И**нициализации переменных.

double s1 = 0; // **И**нициализации переменных.

double s2 = 0; // **И**нициализации переменных.

int i = 1; // **И**нициализации переменных.

// Вычисляем площадь, путем суммирования значений функции f(x) на параболических отрезках.

while (i < n) {

s2 += f(x);

x += h;

s1 += f(x);

x += h;

i++;

}

double result = (h / 3) \* (f(a) + f(b) + 4 \* s1 + 2 \* s2); // Вычисление площади криволинейной трапеции по формуле.

return result; // Возврат вычисленной площади.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

//Ввод переменных.

double a = 0;

double b = 4;

int n = 200;

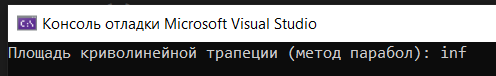
double result = parabolicMethod(a, b, n); // Возвращаем вычисленную площадь криволинейной трапеции с использованием метода парабол.

std::cout << "Площадь криволинейной трапеции (метод парабол): " << result << std::endl; // Выводим фразу и значение на экран.

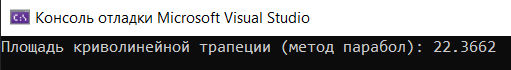
return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS:



С исправленными данными (a = 1, b = 4):



2. Функция:

x2 + 4x – 2

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом.

#include <cmath> // Подключаем библиотеку для работы с математическими функциями.

// Возвращаем значение функции для заданного x.

double f(double x) {

return pow(x, 3) + x - 2;

}

//Реализуем метод дихотомии.

double findRoot(double a, double b, double e) {

double x;

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

x = (a + b) / 2;

if (f(x) \* f(a) <= 0) {

b = x;

}

else {

a = x;

}

}

return x; // Возврат значения х.

}

int main() // Главная функция программы.

{

setlocale(LC\_CTYPE, "RU"); // Используется для вывода русского текста.

double a, b, e; //Ввод переменных.

// Выводим фразы и значения на экран.

std::cout << "Введите значение a: ";

std::cin >> a;

std::cout << "Введите значение b: ";

std::cin >> b;

std::cout << "Введите точность e: ";

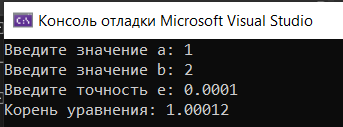
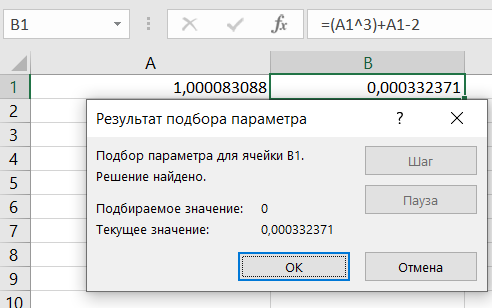
std::cin >> e;

double root = findRoot(a, b, e); // Возвращаем значение корня функции на указанном интервале.

std::cout << "Корень уравнения: " << root << std::endl; // Выводим фразу и значение на экран.

return 0; // Возвращение нулевого значения, чтобы сообщить ОС об успешном завершении программы.

}

Консоль VS: