ГУО “БГУИР”

Факультет компьютерного проектирования

Кафедра Инженерной психологии и эргономики

Отчёт по

Лабораторной работе №6

СРЕДА ПРОГРАММИРОВАНИЯ VISUAL C++. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

Подготовил:

Студент гр.110101

Ладутько Я.Д.

Проверила:

Семижон Е.А.

Минск 2022

***Цель работы****:* изучить алгоритмы поиска корней нелинейных алгебраических уравнений с заданной точностью.

Вариант 16.



Код:

**///MySpace.h**

#pragma once

#include <iostream>

template <typename T>

void input(T& a)

{

while (true)

{

std::cin >> a;

if (std::cin.fail() || (std::cin.peek() != '\n' && std::cin.peek() != ' ' && std::cin.peek() != '\t'))

{

std::cin.clear();

std::cin.ignore(10000, '\n');

std::cout << "Неверный ввод" << std::endl;

}

else

{

return;

}

}

}

void input(int& x, int a, int b);

void input(double& x, double a, double b);

void input(bool& b);

**///MySpace.cpp**

#include"MySpace.h"

using namespace std;

void input(int& x, int a, int b)

{

while (true)

{

input(x);

if (x < a || x > b)

{

cout <<

"Неверный ввод\n"

"Введите число на промежутке от " << a << " до " << b << endl;

}

else

return;

}

}

void input(double& x, double a, double b)

{

while (true)

{

input(x);

if (x < a || x > b)

{

cout <<

"Неверный ввод\n"

"Введите число на промежутке от " << a << " до " << b << endl;

}

else

return;

}

}

void input(bool& b)

{

char c;

while (true)

{

input(c);

if (c != 'y' && c != 'n')

{

cout <<

"Неверный ввод\n"

"Введите y или n\n";

}

else

{

b = (c == 'y' ? true : false);

return;

}

}

}

**///Lab6.cpp**

#include <iomanip>

#include <iostream>

#include "MySpace.h"

using namespace std;

double calcFunc(double x)

{

return 0.1 \* x \* x \* x + x \* x - 10 \* sin(x) - 8;

}

bool findRoot(double e, double prew\_x, double& result, int& iterations)

{

double derivative, func, x;

for(int k = 1; k <= 100; k++)

{

derivative = 3 / 10 \* prew\_x \* prew\_x + 2 \* prew\_x - 10 \* cos(prew\_x);

func = calcFunc(prew\_x);

x = prew\_x - func / derivative;

if(x - prew\_x < e)

{

result = x;

iterations = k;

return true;

}

prew\_x = x;

}

return false;

}

int main()

{

double h, e = 1e-6, result;

int a = -4, b = 4, iters;

setlocale(LC\_ALL, "RU");

cout << "Уравнение: 0.1x^3 + x^2 - 10sin(x) - 8\n";

cout << "Введите шаг h в интервале [0.000001, 0.4]: ";

input(h, 0.000001, 0.4);

cout << "|" << fixed << setw(8) << "Epsilon";

cout << "|" << fixed << setw(10) << "Root";

cout << "|" << fixed << setw(10) << "Iterations|\n";

cout.fill('-');

cout << setw(32) << '-';

cout.fill(' ');

cout << endl;

bool isFound;

for(double x = a; x <= b - h; x += h)

{

if(calcFunc(x) <= 0 && calcFunc(x+h) > 0 || calcFunc(x) >= 0 && calcFunc(x + h) < 0)

{

isFound = findRoot(e, x, result, iters);

if(isFound)

{

cout << "|" << fixed << setw(8) << e;

cout << "|" << fixed << setw(10) << result;

cout << "|" << fixed << setw(10) << iters;

cout << "|\n";

}

}

}

}

***Вывод:*** изучил алгоритмы поиска корней нелинейных алгебраических уравнений с заданной точностью.

