МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №2

По дисциплине «Вычислительная математика»

За семестр

Тема: «Решение систем линейных алгебраических уравнений»

Выполнил:

студент курса

группы АС-574

Парфеевец Д.И.

Проверил:

Пролиско Е.Е.

Брест 2021

Вариант 107

Задание. Решить систему линейных алгебраических уравнений, заданных расширенной матрицей  
Решение получить методом Гаусса, а также методом простых итераций и методом Зейделя с точностью до 10^(-6)

*Код программы:*

#include <iostream>

#include<math.h>

#include<cmath>

#include<stdio.h>

using namespace std;

void GayssMethod(double matrix[3][4], double E) //метод Гаусса

{

for (int i = 0; i <= 2; i++) //прямой ход метода Гаусса

{

double c = matrix[i][0];

matrix[i][3] = matrix[i][3] / c;

for (int j = 0; j <= 2; j++)

{

matrix[i][j] = matrix[i][j] / c;

}

}

for (int i = 1; i <= 2; i++)

{

matrix[i][3] = matrix[i][3] - matrix[0][3];

for (int j = 0; j <= 2; j++)

{

double c = matrix[0][j];

matrix[i][j] = matrix[i][j] - c;

}

}

for (int i = 1; i <= 2; i++)

{

double c = matrix[i][1];

matrix[i][3] = matrix[i][3] / c;

for (int j = 1; j <= 2; j++)

{

matrix[i][j] = matrix[i][j] / c;

}

}

matrix[2][3] = matrix[2][3] - matrix[1][3]; //обратный ход

for (int j = 1; j <= 2; j++)

{

matrix[2][j] = matrix[2][j] - matrix[1][j];

}

double x3 = matrix[2][3] / matrix[2][2];

double x2 = matrix[1][3] - matrix[1][2] \* x3;

double x1 = matrix[0][3] - matrix[0][1] \* x2 - matrix[0][2] \* x3;

cout << endl;

cout << "x1 = " << x1 << endl;

cout << "x2 = " << x2 << endl;

cout << "x3 = " << x3 << endl;

}

void SimpleIterMethod(double matrix[3][4], double E) // метод простых итераций

{

int k = 0;

double x1 = 0, x2 = 0, x3 = 0;//выбираем начальное приближение

while (true)

{

double kr1 = (matrix[0][3] - (matrix[0][2] \* x3) - (matrix[0][1] \* x2)) / matrix[0][0];

double kr2 = (matrix[1][3] - (matrix[1][2] \* x3) - (matrix[1][0] \* x1)) / matrix[1][1];

double kr3 = (matrix[2][3] - (matrix[2][1] \* x2) - (matrix[2][0] \* x1)) / matrix[2][2];

if (3 \* E > abs(kr1 - x1) + abs(kr2 - x2) + abs(kr3 - x3)) break;

x1 = kr1;

x2 = kr2;

x3 = kr3;

k++;

}

cout << "x1= " << x1 << endl;

cout << "x2= " << x2 << endl;

cout << "x3= " << x3 << endl;

cout << "Numder of iterations:= " << k << endl;

cout << "Nevyska x1=" << abs(matrix[0][3] - (matrix[0][0] \* x1 + matrix[0][1] \* x2 + matrix[0][2] \* x3)) << endl;

cout << "Nevyska x2=" << abs(matrix[1][3] - (matrix[1][0] \* x1 + matrix[1][1] \* x2 + matrix[1][2] \* x3)) << endl;

cout << "Nevyska x3=" << abs(matrix[2][3] - (matrix[2][0] \* x1 + matrix[2][1] \* x2 + matrix[2][2] \* x3)) << endl;

}

void ZeidelMethod(double matrix[3][4], double E) //метод Зейделя

{

int k = 0;

double x1 = 0, x2 = 0, x3 = 0;

while (true)

{

double kr1 = (matrix[0][3] / matrix[0][0]) - (matrix[0][2] / matrix[0][0]) \* x3 - (matrix[0][1] / matrix[0][0]) \* x2;

double kr2 = (matrix[1][3] / matrix[1][1]) - (matrix[1][2] / matrix[1][1]) \* x3 - (matrix[1][0] / matrix[1][1]) \* kr1;

double kr3 = (matrix[2][3] / matrix[2][2]) - (matrix[2][1] / matrix[2][2]) \* kr2 - (matrix[2][0] / matrix[2][2]) \* kr1;

if (3 \* E > abs(kr1 - x1) + abs(kr2 - x2) + abs(kr3 - x3)) break;

x3 = kr3;

x2 = kr2;

x1 = kr1;

k++;

}

cout << "x1= " << x1 << endl;

cout << "x2= " << x2 << endl;

cout << "x3= " << x3 << endl;

cout << "Numder of iterations:= " << k << endl;

cout << "Nevyska x1=" << abs(matrix[0][3] - (matrix[0][0] \* x1 + matrix[0][1] \* x2 + matrix[0][2] \* x3)) << endl;

cout << "Nevyska x2=" << abs(matrix[1][3] - (matrix[1][0] \* x1 + matrix[1][1] \* x2 + matrix[1][2] \* x3)) << endl;

cout << "Nevyska x3=" << abs(matrix[2][3] - (matrix[2][0] \* x1 + matrix[2][1] \* x2 + matrix[2][2] \* x3)) << endl;

}

int main()

{

double matrix[3][4] = { //данный массив

{ -16.4, 8.2, -6.4, 198.6},

{ -8.2, -10.9, 0.6, -31.9},

{ -8.6, 8.0, -19.1, 121.7},

};

double E = 0.000001; //точность

cout << "Method Gausa:";

GayssMethod(matrix, E);

cout << endl << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << endl << "Method simple iteratuons:" << endl;

SimpleIterMethod(matrix, E);

cout << endl << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << endl << "Method Zeiselya:" << endl;

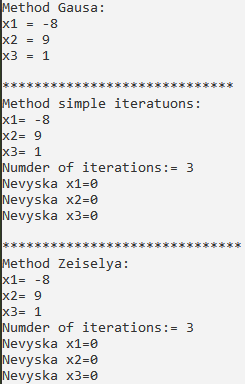
ZeidelMethod(matrix, E);

system("pause");

return 0;

}

*Результат работы программы:*



Вывод: Получил решение методом Гаусса, методом простых итераций и методом Зейделя. Значения корней сошлись во всех трех решениях.