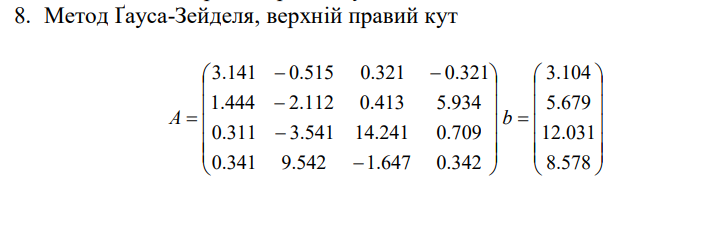
Варіант 8



#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    const int N = 4; // кількість рівнянь

    double eps; // точність

    double A[N][N] = {{3.141, -0.515, 0.321,-0.321},

                      {0.341,9.542,-1.647,0.342},

                      {0.311, -3.541, 14.241,0.709},

                      {1.444, -2.112, 0.413,5.934}}; // матриця коефіцієнтів

    double b[N] = {3.104,8.578,12.031,5.679}; // вектор вільних членів

    double x[N] = {0, 0, 0, 0}; // вектор невідомих

    double x\_prev[N]; // вектор невідомих на попередньому кроці

    double norm, sum;

    eps=0.01;

    int k = 0; // лічильник ітерацій

    do {

        norm = 0;

        // зберігаємо попереднє наближення

        for (int i = 0; i < N; i++) {

            x\_prev[i] = x[i];

        }

        // розраховуємо нове наближення

        for (int i = 0; i < N; i++) {

            sum = 0;

            for (int j = 0; j < N; j++) {

                if (j != i) {

                    sum += A[i][j] \* x[j];

                }

            }

            x[i] = (b[i] - sum) / A[i][i];

            // обчислюємо норму різниці між поточним і попереднім наближеннями

            norm += fabs(x[i] - x\_prev[i]);

        }

        k++;

    } while (norm > eps); // повторюємо, поки не досягнута потрібна точність

    // виводимо результати

    cout << "Розв'язок СЛАР методом Ґауса-Зейделя(ε="<<eps<<"): " << endl;

    for (int i = 0; i < N; i++) {

        cout << "x[" << i + 1 << "] = " << x[i] << endl;

    }

    cout << "Кількість ітерацій: " << k << endl;

    eps=0.0001;

    do {

        norm = 0;

        // зберігаємо попереднє наближення

        for (int i = 0; i < N; i++) {

            x\_prev[i] = x[i];

        }

        // розраховуємо нове наближення

        for (int i = 0; i < N; i++) {

            sum = 0;

            for (int j = 0; j < N; j++) {

                if (j != i) {

                    sum += A[i][j] \* x[j];

                }

            }

            x[i] = (b[i] - sum) / A[i][i];

            // обчислюємо норму різниці між поточним і попереднім наближеннями

            norm += pow(x[i] - x\_prev[i], 2);

        }

        k++;

    } while (sqrt(norm) > eps); // повторюємо, поки не досягнута потрібна точність

    // виводимо результати

    cout << "Розв'язок СЛАР методом Ґауса-Зейделя(ε="<<eps<<"): " << endl;

    for (int i = 0; i < N; i++) {

        cout << "x[" << i + 1 << "] = " << x[i] << endl;

    }

    cout << "Кількість ітерацій: " << k << endl;

    cout<<A[0][0]\*x[0]+A[0][1]\*x[1]+A[0][2]\*x[2]+A[0][3]\*x[3]<<"≈"<<b[0]<<endl;

    cout<<A[1][0]\*x[0]+A[1][1]\*x[1]+A[1][2]\*x[2]+A[1][3]\*x[3]<<"≈"<<b[1]<<endl;

    cout<<A[2][0]\*x[0]+A[2][1]\*x[1]+A[2][2]\*x[2]+A[2][3]\*x[3]<<"≈"<<b[2]<<endl;

    cout<<A[3][0]\*x[0]+A[3][1]\*x[1]+A[3][2]\*x[2]+A[3][3]\*x[3]<<"≈"<<b[3]<<endl;

    return 0;

}

