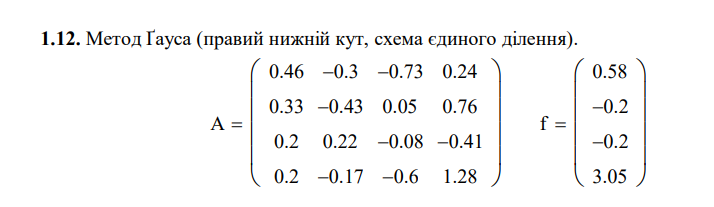
Варіант 12



#include <iostream>

#include <vector>

using namespace std;

int main() {

    // Вхідні дані: матриця коефіцієнтів та вектор вільних членів

    double A[4][4] = {{0.46, -0.3, -0.73, 0.24}, {0.33, -0.43, 0.05, 0.76}, {0.2, 0.22, -0.08, -0.41}, {0.2, -0.17, -0.6, 1.28}};

    double b[4] = {0.58, -0.2, -0.2, 3.05};

    // Розмірність матриці

    int n = 4;

    // Крок 1: зведення матриці до трикутної форми

    for (int k = 0; k < n-1; k++) {

        // Вибір головного елемента

        int max\_row = k;

        for (int i = k+1; i < n; i++) {

            if (abs(A[i][k]) > abs(A[max\_row][k])) {

                max\_row = i;

            }

        }

        if (max\_row != k) {

            swap(A[k], A[max\_row]);

            swap(b[k], b[max\_row]);

        }

        // Операції з рядками

        for (int row = k+1; row < n; row++) {

            double multiplier = A[row][k] / A[k][k];

            for (int col = k; col < n; col++) {

                A[row][col] -= multiplier \* A[k][col];

            }

            b[row] -= multiplier \* b[k];

        }

    }

    // Крок 2: знаходження розв'язку зворотнім ходом

    double x[4];

    for (int k = n-1; k >= 0; k--) {

        x[k] = b[k];

        for (int i = k+1; i < n; i++) {

            x[k] -= A[k][i] \* x[i];

        }

        x[k] /= A[k][k];

    }

    // Вивід результату

    cout << "Розв'язок: ";

    for (int i=0;i<n;i++) {

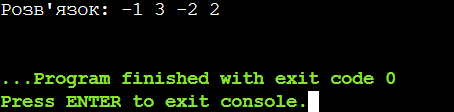
        cout << x[i] << " ";

    }

    cout << endl;

    return 0;

}



Піідставивши отримані корені у рівння можемо легко перевірити що вони задовольняють рівняння