

**Міністерство освіти і науки України  
Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"  
Фізико-технічний інститут**

«Методи обчислень»

**Лабораторна робота №3**  
Варіант 2

«Розв'язання СЛАР ітераційними методами»

**Виконала:**  
студентка групи ФБ-95  
Гурджия Валерія Вахтангівна

Київ 2022

### Завдання

Якщо матриця не є матрицею із діагональною перевагою, привести систему до еквівалентної, у якій є діагональна перевага (письмово). Реалізувати програму, що реалізує розв'язання за ітераційним методом, який відповідає заданому варіантові. Обчислення проводити з  $\varepsilon = 10^{-4}$ . Для кожної ітерації розраховувати вектор нев'язки  $r = |b - Ax|$ , де  $x$  - отриманий розв'язок.

№ вар.	Матриця системи $A$	Вектор правої частини $b$
2	1,00 0,42 0,54 0,66 0,42 1,00 0,32 0,44 0,54 0,32 1,00 0,22 0,66 0,44 0,22 1,00	0,3 0,5 0,7 0,9

Метод розв'язання: **метод простої ітерації**

### Письмова частина

$$A = \begin{pmatrix} 1,00 & 0,42 & 0,54 & 0,66 \\ 0,42 & 1,00 & 0,32 & 0,44 \\ 0,54 & 0,32 & 1,00 & 0,22 \\ 0,66 & 0,44 & 0,22 & 1,00 \end{pmatrix}$$

1. Перевіримо, чи є матриця матрицею з діагональною перевагою.

Означення матриці з діагональною перевагою:

$$\forall i: |a_{ii}| > \sum_{j \neq i} |a_{ij}|, i = 1, \dots, n$$

$$1 < 0,42 + 0,54 + 0,66$$

Як бачимо, матриця не є виду з діагональною перевагою.

Перетворення на матрицю з діагональною перевагою:

1	0,42	0,54	0,66	0,3
0,42	1	0,32	0,44	0,5
0,54	0,32	1	0,22	0,7
0,66	0,44	0,22	1	0,9

Помножимо перший рядок на -0,42 та додамо до 2 рядка

1	0,42	0,54	0,66	0,3
0	0,8236	0,0932	0,1628	0,374
0,54	0,32	1	0,22	0,7
0,66	0,44	0,22	1	0,9

Помножимо перший рядок на -0,54 та додамо до 3 рядка

1	0,42	0,54	0,66	0,3
0	0,8236	0,0932	0,1628	0,374
0	0,0932	0,7084	-0,1364	0,538
0,66	0,44	0,22	1	0,9

Помножимо перший рядок на -0,66 та додамо до 4 рядка

1	0,42	0,54	0,66	0,3
0	0,8236	0,0932	0,1628	0,374
0	0,0932	0,7084	-0,1364	0,538
0	0,1628	-0,1364	0,5644	0,702

Помножимо другий рядок на -0,11316 та додамо до 3 рядка

1	0,42	0,54	0,66	0,3
0	0,8236	0,0932	0,1628	0,374
0	0	0,697853	-0,15482	0,495678
0	0,1628	-0,1364	0,5644	0,702

Помножимо другий рядок на -0,19767 та додамо до 4 рядка

1	0,42	0,54	0,66	0,3
0	0,8236	0,0932	0,1628	0,374
0	0	0,697853	-0,15482	0,495678
0	0	-0,15482	0,53222	0,628072

Помножимо третій рядок на 0,221856 та додамо до 4 рядка

1	0,42	0,54	0,66	0,3
0	0,8236	0,0932	0,1628	0,374
0	0	0,697853	-0,15482	0,495678
0	0	0	0,497871	0,738041

Помножимо другий рядок на -0,5 та додамо до 1 рядка

1	0,0082	0,4934	0,5786	0,113
0	0,8236	0,0932	0,1628	0,374
0	0	0,697853	-0,15482	0,495678
0	0	0	0,497871	0,738041

Помножимо третій рядок на -0,5 та додамо до 1 рядка

1	0,0082	0,144473	0,656011	-0,13484
0	0,8236	0,0932	0,1628	0,374
0	0	0,697853	-0,15482	0,495678
0	0	0	0,497871	0,738041

Перевіряємо:

$$1 > 0,0082 + 0,144473 + 0,656011 = 0,80868$$

$$0,8236 > 0 + 0,932 + 0,1628 = 0,256$$

$$0,697853 > 0 + 0 + 0,15482 = 0,15482$$

$$0,738041 > 0 + 0 + 0 + 0 = 0$$

Тепер це матриця з діагональною перевагою

Можемо застосовувати метод простої ітерації.

## Результат роботи програми

```
Итерация № 1
Корни x:          -0.13484    0.454104    0.71029    1.48239
Вектор невязки r:  1.078808    0.307533    0.229504    0.000000

Итерация № 2
Корни x:          -1.213648    0.080703    1.039162    1.482394
Вектор невязки r:  0.044451    0.030651    0.000000    0.000000

Итерация № 3
Корни x:          -1.258099    0.043488    1.039162    1.482394
Вектор невязки r:  0.000305    0.000000    0.000000    0.000000

Итерация № 4
Корни x:          -1.257794    0.043488    1.039162    1.482394
Вектор невязки r:  0.000000    0.000000    0.000000    0.000000

Итерация № 5
Корни x:          -1.257794    0.043488    1.039162    1.482394
Вектор невязки r:  0.000000    0.000000    0.000000    0.000000

C:\Users\user\Desktop\labs\методы\lab 3\методи лаб 3\Debug\методи лаб 3.
```

## Код програми

```
#include <iostream>
#include <cmath>
#include <vector>
using namespace std;

bool check(vector<vector<float>> matrix_A) {
    for (int i = 0; i < matrix_A.size(); i++) {
        float sum = 0;
        for (int j = 0; j < matrix_A[i].size(); j++) {
            if (i != j) {
                sum += abs(matrix_A[i][j]);
            }
        }
        if (abs(matrix_A[i][i]) <= sum) {
            return false;
        }
    }
    return true;
}

void SimpleIterationMethod(vector<vector<float>> matrix_A, vector<float> vector_B, float
epsilon) {
    vector<vector<float>> X = { {0, 0, 0, 0} };
    vector<float> Delta;
    vector<float> R;
    float maxDelta = 1;
    int count = 0;
    vector<float> answers;
    while (maxDelta > epsilon) {
        cout << "Итерация № " << count+1;
        for (int i = 0; i < matrix_A.size(); i++) {
            float sum = 0;
            for (int j = 0; j < matrix_A[i].size(); j++) {
                if (i != j) {
                    sum += matrix_A[i][j] * X[count][j];
                }
            }
            float x = (vector_B[i] - sum) / matrix_A[i][i];
            answers.push_back(x);
        }
        X.push_back(answers);
        if (count > 0) {
            for (int j = 0; j < X[X.size() - 1].size(); j++) {
                float delta = abs(X[X.size() - 1][j] - X[X.size() - 2][j]);
                Delta.push_back(delta);
            }
            maxDelta = 0;
            for (int i = 0; i < Delta.size(); i++) {
                if (Delta[i] > maxDelta) {
                    maxDelta = Delta[i];
                }
            }
        }
        for (int i = 0; i < matrix_A.size(); i++) {
            float AX = 0;
            for (int j = 0; j < matrix_A[i].size(); j++) {
                AX += matrix_A[i][j] * answers[j];
            }
            float r = abs(vector_B[i] - AX);
            R.push_back(r);
        }
    }
}
```

```

        cout << "\nКорни x: \t\t";
        for (auto a : answers) {
            cout << a << " ";
        }
        cout << "\nВектор невязки r: \t";
        for (auto r : R) {
            cout << fixed << r << " ";
        }
        cout << endl << endl;
        R.clear();
        answers.clear();
        Delta.clear();
        count++;
    }
}

int main()
{
    setlocale(LC_ALL, "ru");
    vector<vector<float>> matrix_A = {
        {1, 0.0082, 0.144473, 0.656011},
        {0, 0.8236, 0.0932, 0.1628},
        {0, 0, 0.697853, -0.15482},
        {0, 0, 0, 0.497871} };
    vector<float> vector_B = { -0.13484, 0.374, 0.495678, 0.738041 };
    float epsilon = 0.0001;

    if (!check(matrix_A)) {
        cout << "Матрица не диагонально преобладающая!\n";
    }
    else {
        SimpleIterationMethod(matrix_A, vector_B, epsilon);
    }
}

```