

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
“ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Лабораторна робота №2
**ПРЯМІ МЕТОДИ РОЗ’ЯЗАННЯ СИСТЕМ ЛІНІЙНИХ
АЛГЕБРАЇЧНИХ РІВНЯНЬ.**
Варіант-27

Виконав:
Терещенко Денис, КА-96
Прийняла:
Шубенкова І. А.

КИЇВ - 2020

1. Постановка завдання.

1. Проаналізувати конкретну СЛАР та обґрунтувати обрання методу розв'язання. Реалізувати обраний метод у вигляді окремої процедури або методу відповідного об'єкту "СЛАР".
2. Розв'язати СЛАР з точністю $\varepsilon = 10^{-5}$.
3. За необхідності зробити ітераційне уточнення.
4. Обчислити A^{-1} та $\det A$. Визначник матриці повинен бути бічним результатом процедури з першого пункту. Обернену матрицю треба знайти, скориставшись тією ж процедурою, передаючи їй як вектори правої частини стовпці одиничної матриці. Тому краще за все реалізувати будь-який прямий метод у формі, що дозволяє працювати з низкою стовпців правої частини.

2. Аналітична частина.

Завдання варіанту:

$$A = \begin{bmatrix} 7.03 & 0.94 & 1.13 & 1.135 & -0.81 \\ 1.26 & 3.39 & 1.3 & -1.63 & -1.53 \\ 0.81 & -2.46 & 6.21 & 2.1 & -1.067 \\ 1.345 & 0.16 & 2.1 & 5.33 & 16 \\ 1.29 & 0.87 & 1.333 & -8 & 15 \end{bmatrix} \quad b = \begin{bmatrix} 2.1 \\ 0.84 \\ -3.44 \\ -0.92 \\ -1.47 \end{bmatrix}$$

Матриця є несиметричною, тому будемо застосовувати метод Гаусса.

3. Лістинг програми.

```

1  #include "stdafx.h"
   #include <iostream>
3  #include <vector>

5  using namespace std;

7  void Decomposition(vector <vector <double>> A, vector <vector <double>> &L,
   vector <vector <double>> &U, int n)
9  {
   for (int j = 0; j < n; j++) {
11     L[j][0] = A[j][0];
       U[0][j] = A[0][j] / L[0][0];

```

```

13     }
14     for (int i = 1; i < n; i++) {
15         for (int j = i; j < n; j++) {
16             for (int k = 0; k < i; k++) {
17                 U[i][j] += (L[i][k] * U[k][j]);
18                 L[j][i] += (L[j][k] * U[k][i]);
19             }
20             L[j][i] = A[j][i] - L[j][i];
21             U[i][j] = (1 / L[i][i])*(A[i][j] - U[i][j]);
22         }
23     }
24 }
25
26 void print_matrix(vector <vector <double>> A, int n)
27 {
28     for (int i = 0; i < n; i++)
29     {
30         for (int j = 0; j < n; j++)
31         {
32             cout << "\t" << A[i][j] << "\t";
33         }
34         cout << endl;
35     }
36 }
37
38 void print_matrixv(vector <double> A, int n)
39 {
40     for (int i = 0; i < n; i++)
41     {
42         cout << "\t" << A[i] << "\t";
43     }
44     cout << endl;
45 }
46
47 void inverse(vector <vector <double>> A, vector <vector <double>> &B, int n)
48 {
49     for (int i = 0; i < n; i++)
50     for (int j = 0; j < n; j++) {
51         if (i == j) B[i][j] = 1 / A[i][j];
52         else {
53             if (i < j) B[i][j] = 0;
54             else
55                 for (int k = 0; k < i; k++) {
56                     B[i][j] += (((-1 / A[i][i])*B[k][j]) * A[i][k]);
57                 }
58         }
59     }
60 }
61
62 void inverse2(vector <vector <double>> A, vector <vector <double>> &B, int n)
63 {
64     for (int i = 0; i < n; i++)
65     for (int j = 0; j < n; j++) {
66         if (i == j) B[i][j] = 1 / A[i][j];

```

```

63         else {
64             if (i > j) B[i][j] = 0;
65             else
66                 for (int k = 0; k < j; k++) {
67                     B[i][j] += (((-1 / A[i][i])*B[i][k]) * A[k][j]);
68                 }
69         }
70     }
71 }
72 void monm(vector <vector <double>> A, vector <vector <double>> B,
73     vector <vector <double>> &R, int n)
74 {
75     for (int i = 0; i < n; i++)
76         for (int j = 0; j < n; j++)
77             for (int k = 0; k < n; k++)
78                 R[i][j] += A[i][k] * B[k][j];
79 }
80 void calc_monv(vector <vector <double>> A, vector <double> B,
81     vector <double> &R, int n)
82 {
83     for (int i = 0; i < n; i++)
84         for (int j = 0; j < n; j++)
85             R[i] += A[i][j] * B[j];
86 }
87 void vonm(vector <double> A, vector <vector <double>> B,
88     vector <double> &R, int n)
89 {
90     for (int i = 0; i < n; i++)
91         for (int j = 0; j < n; j++)
92             R[i] += A[j] * B[i][j];
93 }
94 void vmv(vector<double> b, vector <double> x, vector<double>&r,int n) {
95     for (int i = 0; i < n; i++) {
96         r[i] = b[i] - x[i];
97     }
98 }
99 int main()
100 {
101     int n;
102     cin >> n;
103     double n1;
104     vector <vector <double>> A(n), L(n), U(n), R(n), Li(n), Ui(n), E(n);
105     vector <double> B(n,0), Y(n,0), X(n,0), r(n,0),r1(n,0);
106     for (size_t row = 0; row < n; ++row) {
107         A[row].resize(n);
108         L[row].resize(n);
109         U[row].resize(n);
110         R[row].resize(n);
111         Li[row].resize(n);
112         Ui[row].resize(n);

```

```

113     E[row].resize(n);
114 }
115 for (int i = 0; i < n; i++)
116 {
117     for (int j = 0; j < n; j++)
118     {
119         cin >> n1;
120         A[i][j] = n1;
121         L[i][j] = 0;
122         U[i][j] = 0;
123         R[i][j] = 0;
124         Ui[i][j] = 0;
125         Li[i][j] = 0;
126         E[i][j] = 0;
127     }
128 }
129 for (int i = 0; i < n; i++)
130 {
131     cin >> n1;
132     B[i] = n1;
133 }
134 Decomposition(A, L, U, n);
135 cout << "|----- input matrix -----|" << endl;
136 print_matrix(A, n);
137 cout << "|----- U matrix -----|" << endl;
138 print_matrix(U, n);
139 cout << "|----- L matrix -----|" << endl;
140 print_matrix(L, n);
141 monm(L, U, R, n);
142 cout << "|----- L*U matrix -----|" << endl;
143 print_matrix(R, n);
144 inverse(L, Li, n);
145 cout << "|----- inverse L matrix -----|" << endl;
146 print_matrix(Li, n);
147 inverse2(U, Ui, n);
148 cout << "|----- inverse U matrix -----|" << endl;
149 print_matrix(Ui, n);
150 calc_monv(Li, B, Y, n);
151 cout << "|----- L(-1)*b -----|" << endl;
152 print_matrixv(Y, n);
153 calc_monv(Ui, Y, X, n);
154 cout << "|----- solution -----|" << endl;
155 print_matrixv(X, n);
156 calc_monv(A, X, r, n);
157 vmv(B, r, r1, n);
158 cout << "|----- residual vector -----|" << endl;
159 print_matrixv(r1, n);
160 cout << "|----- inverse A matrix -----|" << endl;
161 print_matrix(Ai, n);
162 calc_monv(Ai, A, Y, n);

```

```

163     cout << "|----- det A -----|" << endl; calc_det(L, U);
      return 0;
165 }

```

4. Результати роботи.

```

|----- input matrix -----|
7.03      0.94      1.13      1.135      -0.81
1.26      3.39      1.3      -1.63      -1.53
0.81      -2.46      6.21      2.1      -1.067
1.345     0.16      2.1      5.33      16
1.29      0.87      1.333     -8      15
|----- U matrix -----|
1      0.133713      0.16074      0.161451      -0.11522
0      1      0.340668      -0.569119      -0.429866
0      0      1      0.0729795      -0.298746
0      0      0      1      3.36677
0      0      0      0      1
|----- L matrix -----|
7.03      0      0      0      0
1.26      3.22152      0      0      0
0.81      -2.56831      6.95474      0      0
1.345     -0.0198435      1.89057      4.96358      0
1.29      0.697511      0.888027      -7.87611      42.2308
|----- L*U matrix -----|
7.03      0.94      1.13      1.135      -0.81
1.26      3.39      1.3      -1.63      -1.53
0.81      -2.46      6.21      2.1      -1.067
1.345     0.16      2.1      5.33      16
1.29      0.87      1.333     -8      15
|----- inverse L matrix -----|
0.142248      0      0      0      0
-0.0556358     0.310412      0      0      0
-0.0371129     0.114632      0.143787      0      0
-0.0246319     -0.0424208      -0.0547666      0.201467      0
-0.00723972     -0.015449      -0.0132376      0.037574      0.0236794
|----- inverse U matrix -----|
1      -0.133713      -0.115188      -0.229143      0.794801
0      1      -0.340668      0.59398      -1.6717
0      0      1      -0.0729795      0.544451
0      0      0      1      -3.36677
0      0      0      0      1

```

```

|----- L^(-1)*b -----|
0.29872      0.143911      -0.476273      -0.0843135      -0.05202
|----- solution -----|
0.312312      0.343043      -0.498442      0.0908259      -0.05202
|----- residual vector -----|
4.44089e-16      5.55112e-16      8.88178e-16      5.55112e-16      6.66134e-16
|----- inverse A matrix -----|
0.144984      -0.0133376      -0.0200064      -0.0100561      0.00850358
-0.044635      0.298825      -0.0517625      0.0427649      -0.0291451
-0.0167602      0.100349      0.150524      -0.00370488      0.0206768
0.00187008      -0.00698634      -0.0104795      0.0740976      -0.0802018
-0.0075509      -0.00952683      -0.0142903      0.0404361      0.021947
|----- det A -----|
28341

```

Результат. З точністю $\varepsilon = 10^{-5}$ отримали наступний роз'язок системи лінійних рівнянь:

$$x_1 = 0.312312 \quad x_2 = 0.343047 \quad x_3 = -0.498441 \quad x_4 = 0.0908238 \quad x_5 = -0.0520194$$

$$\det A = 28341.234$$

$$A^{-1} = \begin{bmatrix} 0.144984 & -0.013337 & -0.020006 & -0.010056 & 0.008508 \\ -0.044635 & 0.298825 & -0.051762 & 0.042764 & -0.029145 \\ -0.016760 & 0.100349 & 0.150524 & -0.003704 & 0.020676 \\ 0.001870 & -0.006986 & -0.010479 & 0.074097 & -0.080201 \\ -0.007550 & -0.009526 & -0.014290 & 0.040436 & 0.021947 \end{bmatrix}$$