# A elaborat: Sandiuc Vitalie, IS21Z

**A verificat: Țîcău Vitalie**

# PROIECT DE PROGRAM Nr. 2. REZOLVAREA SISTEMELOR DE ECUAŢII ALGEBRICE LINIARE (SEAL). INVERSAREA MATRICELOR. CALCULUL DETERMINANŢILOR NELINIARE

## I. Formularea problemei

Fie sistemul de ecuații algebrice liniare

76x**1+**1x2+2x**3**+3x**4**+4x**5**=2  
1x**1**+75x**2**+3x**3**+4x**4**+4x**5**=3  
2x**1**+3x**2**+74x**3**+4x**4**+5x**5**=4  
3x**1**+4x**2**+3x**3**+73x**4**+5x**5**=5

4x**1**+4x**2**+3x**3**+2x**4**+72x**5**=6

Se cere de aflat soluțiile sistemului dat, utilizând metoda Radacina Patrata și Jacobi.

## II. Programul

**a). Metoda rădăcinii pătrate**

#include <iostream>

#include <math.h>

#include <cstdlib>

#include <stdio.h>

using namespace std;

int i, j, k, n = 5, l;

double u[5][5], ut[5][5], y[3], x[3];

double a[5][5] = {

{76,1,2,3,4},

{

1,75,3,4,4

},

{

2,3,74,4,5

},

{

3,4,3,73,5

},

{

4,4,3,2,72

}

};

double b[5] = {2,3,4,5,6 };

double sum;

int fun();

int main()

{

cout << "Sistemul" << endl;

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

u[i][j] = 0;

cout << " " << a[i][j] << " ";

}

cout << " =" << b[i] << endl;

}

l = fun();

if (l == 1)

cout << "Nu e corect" << endl;

system("pause");

}

int fun()

{

for (j = 0; j < n; j++) {

sum = 0;

for (k = 0; k < j; k++) {

sum += u[j][k] \* u[j][k];

}

if ((a[j][j] - sum) < 0) {

return 1;

}

u[j][j] = sqrt(a[j][j] - sum);

if (u[j][j] == 0) {

return 1;

}

for (i = j + 1; i < n; i++) {

sum = 0;

for (k = 0; k < j; k++)

sum += u[i][k] \* u[j][k];

u[i][j] = (a[i][j] - sum) / u[j][j];

}

}

cout << "Matricea U" << endl;

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

printf(" %8.2lf", u[i][j]);

}

cout << endl;

}

for (i = 0; i < n; i++)

for (j = 0; j < n; j++)

ut[j][i] = u[i][j];

y[0] = b[0] / u[0][0];

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

sum = 0;

for (j = 0; j < i; j++)

sum += u[i][j] \* y[j];

y[i] = (b[i] - sum) / u[i][i];

}

cout << "Rezultat Y" << endl;

for (i = 0; i < n; i++) {

printf(" %8.3lf", y[i]);

}

cout << endl;

x[n-1] = y[n-1] / u[n-1][n-1];

for (i = n - 2; i >= 0; i--) {

sum = 0;

for (j = i + 1; j < n; j++)

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

x[i] = (y[i] - sum) / u[i][i];

}

cout << "Rezultat X" << endl;

for (i = 0; i < n; i++) {

printf(" %8.3lf", x[i]);

}

cout << endl;

cout << "Verificare" << endl;

for (i = 0; i < 4; i++) {

sum = 0;

for (j = 0; j < 4; j++)

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

cout << sum << endl;

}

return 0;

}

**b). Metoda iterativă: metoda Jacobi.**

## #include <iostream>

## using namespace std;

## int main() {

## double A[10][10], M[10][10], B[10], C[10], x[10][100], sum[10];

## int i, j, n, k, kmax;

## cout << "Introduceti rangul ecuatiei: \n";

## cin >> n;

## cout << "Introduceti coloanele matricei \n";

## for (i = 1; i <= n; i++) {

## for (j = 1; j <= n; j++) {

## cin >> A[i][j];

## }

## }

## cout << "Matricea initiala=\n";

## for (i = 1; i <= n; i++) {

## for (j = 1; j <= n; j++) {

## cout << A[i][j] << "\t\t";

## }

## cout << "\n\n";

## }

## cout << "M=\n";

## for (i = 1; i <= n; i++) {

## for (j = 1; j <= n; j++) {

## if (i == j) {

## M[i][j] = 0;

## }

## else {

## M[i][j] = -A[i][j] / A[i][i];

## }

## }

## }

## for (i = 1; i <= n; i++) {

## for (j = 1; j <= n; j++) {

## cout << M[i][j] << "\t\t";

## }

## cout << "\n\n";

## }

## cout << "Introduceti valorile de dupa egal: \n";

## for (i = 1; i <= n; i++) {

## cin >> B[i];

## }

## cout << "\nC=\n";

## for (i = 1; i <= n; i++) {

## C[i] = B[i] / A[i][i];

## cout << C[i] << endl;

## }

## cout << "Introduceti numarul de iteratii:" << endl;

## cin >> kmax;

## k = 0;

## for (i = 1; i <= n; i++) {

## sum[i] = 0;

## x[i][k] = C[i]; //valorile initiale

## }

## for (k = 0; k <= kmax; k++) {

## for (i = 1; i <= n; i++) {

## for (j = 1; j <= n; j++) {

## sum[i] += M[i][j] \* x[j][k];

## }

## x[i][k] = C[i] + sum[i];

## sum[i] = 0;

## }

## }

## cout << "Raspuns:\n\n";

## for (i = 1; i <= n; i++) {

## cout << x[i][kmax] << endl;

## }

## return 0;

## }

## III. Rezultatele sugestive

**a). Metoda rădăcina pătrate**

Sistemul

76 1 2 3 4 =2

1 75 3 4 4 =3

2 3 74 4 5 =4

3 4 3 73 5 =5

4 4 3 2 72 =6

Matricea U

8.72 0.00 0.00 0.00 0.00

0.11 8.66 0.00 0.00 0.00

0.23 0.34 8.59 0.00 0.00

0.34 0.46 0.32 8.52 0.00

0.46 0.46 0.32 0.18 8.45

Rezultat Y

0.229 0.343 0.446 0.542 0.651

Rezultat X

0.020 0.033 0.049 0.055 0.077

Verificare

6.58858

13.1245

18.8264

25.0111

**b). Metoda iterativă: metoda Jacobi.**

Dati rangul sistemului: 5

Dati elementele matricei:

76 1 2 3 4

1 75 3 4 4

2 3 74 4 5

3 4 3 73 5

4 4 3 2 72

Dati coloana termenilor liberi:

2 3 4 5 6

Introduceti precizia de calcul:

1e-9

Sistemul initial

76.00 1.00 2.00 3.00 4.00 | 2.00

1.00 75.00 3.00 4.00 4.00 | 3.00

2.00 3.00 74.00 4.00 5.00 | 4.00

3.00 4.00 3.00 73.00 5.00 | 5.00

4.00 4.00 3.00 2.00 72.00 | 6.00

Matricea M si Vetorul C

0.00 -0.01 -0.03 -0.04 -0.05 | 0.03

-0.01 0.00 -0.04 -0.05 -0.05 | 0.04

-0.03 -0.04 0.00 -0.05 -0.07 | 0.05

-0.04 -0.05 -0.04 0.00 -0.07 | 0.07

-0.06 -0.06 -0.04 -0.03 0.00 | 0.08

Aproximatia a 1 a solutiei este

0.026 0.040 0.054 0.068 0.083

Aproximatia a 2 a solutiei este

0.017 0.029 0.042 0.057 0.075

Aproximatia a 3 a solutiei este

0.019 0.031 0.044 0.059 0.077

Aproximatia a 4 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 5 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 6 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 7 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 8 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 9 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 10 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 11 a solutiei este

0.018 0.031 0.044 0.059 0.077

Aproximatia a 12 a solutiei este

0.018 0.031 0.044 0.059 0.077

Verificare

-2.7488311626910900e-009 -3.2846825259724710e-009 -3.6560545424446644e-009 -3.7775029493314349e-009 -3.2747686784517782e-009

**IV. Concluzii**

1. Valoarea teoretica calculata coincide cu cea din program.
2. Evaluarea erorii obţinute

La aplicarea metodei de rezolvare a sistemelor de ecuații liniare există eroare de rotungire, fapt ce nu permite primirea rezultatelor exacte.

1. Rezultatele testării

În urma testării rezultatelor obținute, am primit mici erori, deoarece sunt prezente erorile de rotungire, iar aceste fapt nu permite obținerea rezultatelor precise.