****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Raport**

**pentru lucrarea de laborator Nr.2**

***la cursul de “Metode Numerice”***

Elaborat : Calancea Catalin  
Grupa: MI-222

Verificat:

Conferentiar

Buzurniuc Stefan

**Chișinău – 20****23**

**CUPRINS**

[INTRODUCERE 3](#_Toc151369371)

[SARCINA 4](#_Toc151369372)

[COD 4](#_Toc151369373)

[CONCLUZIA 11](#_Toc151369374)

# INTRODUCERE

Rezolvarea sistemelor de ecuații liniare algebrice (SELA) reprezintă un aspect fundamental al algebrei lineare, având numeroase aplicații practice în diverse domenii științifice și inginerești. În cadrul acestui laborator, vom aborda metode de rezolvare a sistemelor de ecuații liniare, concentrându-ne pe trei tehnici distincte: metoda eliminării succesive a necunoscutelor (metoda lui Gauss), metoda lui Jacobi și metoda iterativă Gauss-Saidel. Scopul nostru principal este să investigăm eficiența și acuratețea fiecărei metode în rezolvarea sistemelor complexe de ecuații.

În prima parte a lucrării, vom explora metoda lui Gauss, o tehnică de eliminare a necunoscutelor prin operații elementare de linie. Această metodă oferă o soluție directă a sistemului de ecuații. În continuare, vom investiga metoda lui Jacobi, o abordare iterativă care converge către soluția sistemului cu o anumită precizie. Totodată, vom explora și metoda iterativă Gauss-Saidel, o variantă eficientă a metodei lui Jacobi.

În partea a doua a lucrării, ne vom concentra asupra calculului determinantului matricei asociate sistemului de ecuații liniare. Determinantul reprezintă o măsură a consistenței și invertibilității sistemului, furnizând informații importante despre proprietățile matricei.

Prin acest laborator, vom obține o înțelegere mai profundă a diferitelor tehnici utilizate în rezolvarea sistemelor de ecuații liniare, precum și a importanței determinantului în analiza acestor sisteme. Această cunoaștere va servi drept fundament pentru abordarea problemelor mai complexe și pentru utilizarea acestor concepte în contexte practice din diverse domenii.

# SARCINA

1. Să se resolve sistemul dat de ecuații utilizînd: а) metoda eliminării succesive a necunoscutelor (metoda lui Gauss; b) metoda lui Iacoby cu exactitatea 0.00001; c) metoda iteranivă Gauss – Saidel cu exactitatea 0.00001.

2. Să se calculeze determinantul matricei sistemului.



(N = 4)

# COD

#include <iostream>

#include <math.h>

#define N 4

using namespace std;

float A[4][4] =

{

(7.1 + 0.1\*N), 1.1, 0.9, -1.3,

1.1, (8.2 +0.2\*N), 2.1, 0.5,

0.9, 2.1, (9.1+0.3\*N), 1.2,

-1.3, 0.5, 1.2, (10.5+0.4\*N),

};

float B[4] =

{(10.8 + 0.5\*N),

(10.9 +1.5\*N),

(11.5+2.5\*N),

(11.2+N)};

bool verificaSolutie(float x[N]) {

float epsilon = 0.000001;

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

float suma = 0.0;

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

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

}

if (fabs(suma - B[i]) > epsilon) {

return true;

}

}

return false;

}

*// Determinantul*

float determinant() {

int n = 4;

float det = 1.0;

for (int k = 0; k < n; k++) {

int maxRow = k;

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

if (fabs(A[i][k]) > fabs(A[maxRow][k])) {

maxRow = i;

}

}

if (maxRow != k) {

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

float temp = A[k][j];

A[k][j] = A[maxRow][j];

A[maxRow][j] = temp;

}

det = -det;

}

if (A[k][k] == 0.0) {

return 0.0;

}

det \*= A[k][k];

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

float factor = A[i][k] / A[k][k];

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

A[i][j] -= factor \* A[k][j];

}

}

}

return det;

}

*// Metoda Gauss*

void Gauss() {

int n = 4;

int k1 = 0;

for (int k = 0; k < n; k++) {

if (A[k][k] != 0.0) {

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

A[i][k] /= A[k][k];

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

A[i][j] -= A[i][k] \* A[k][j];

}

B[i] -= A[i][k] \* B[k];

}

} else {

int lin = k;

do {

lin++;

} while (lin < n && A[lin][k] == 0.0);

if (lin == n) {

cout << "Sistemul nu are solutie unica" << endl;

return;

}

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

double aux = A[k][j];

A[k][j] = A[lin][j];

A[lin][j] = aux;

}

double aux\_b = B[k];

B[k] = B[lin];

B[lin] = aux\_b;

k--;

}

k1++;

}

if (A[n - 1][n - 1] == 0.0) {

cout << "Sistemul nu are solutie unica" << endl;

return;

}

B[n - 1] /= A[n - 1][n - 1];

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

double S = 0.0;

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

S += A[i][j] \* B[j];

}

B[i] = (B[i] - S) / A[i][i];

}

cout << "Metoda Gauss cu eroarea 0.000001:" << endl;

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

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

}

cout << "\nNr. de iteratii: " << k1 << endl;

}

*// Metoda Jacobi*

void jacobi() {

int i, j, m, k1 = 0;

float v, x[N], q[N][N], d[N], s[N][N], t[N][N], x1[N], er;

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

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

if (i == j) {

s[i][j] = 1 / A[i][j];

t[i][j] = 0;

} else {

s[i][j] = 0;

t[i][j] = A[i][j];

}

}

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

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

v = 0;

for (m = 0; m < N; m++)

v += s[i][m] \* t[m][j];

q[i][j] = -v;

}

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

v = 0;

for (m = 0; m < N; m++)

v += s[i][m] \* B[m];

d[i] = v;

}

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

x[i] = d[i];

do {

k1++;

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

x1[i] = x[i];

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

v = 0;

for (m = 0; m < N; m++)

v += x1[m] \* q[i][m];

x[i] = v + d[i];

}

er = fabs(x1[0] - x[0]);

for (m = 0; m < N; m++)

if (er < fabs(x1[m] - x[m]))

er = fabs(x1[m] - x[m]);

} while (er > 0.000001);

cout << "Metoda Jacobi cu eroarea 0.000001:" << endl;

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

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

}

cout << "\nNr. de iteratii: " << k1 << endl;

}

*// Metoda Gauss-Seidel*

void gaussseidel() {

int i, j, m, k1 = 0, k;

float v, x[N], q[N][N], d[N], s[N][N], t[N][N], x1[N], er;

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

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

if (i == j) {

s[i][j] = 1 / A[i][j];

t[i][j] = 0;

} else {

s[i][j] = 0;

t[i][j] = A[i][j];

}

}

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

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

v = 0;

for (m = 0; m < N; m++)

v += s[i][m] \* t[m][j];

q[i][j] = -v;

}

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

v = 0;

for (m = 0; m < N; m++)

v += s[i][m] \* B[m];

d[i] = v;

}

k1 = 0;

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

x[i] = d[i];

do {

k1++;

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

x1[i] = x[i];

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

v = 0;

for (m = 0; m < N; m++)

v += x[m] \* q[i][m];

x[i] = v + d[i];

}

er = fabs(x1[0] - x[0]);

for (m = 1; m < N; m++)

if (er < fabs(x1[m] - x[m]))

er = fabs(x1[m] - x[m]);

} while (er > 0.000001);

cout << "Metoda Gauss-Seidel cu eroarea 0.000001:" << endl;

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

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

}

cout << "\nNr. de iteratii: " << k1 << endl;

}

*// Meniu pentru alegerea metodei*

int main() {

int i;

while (true) {

cout << "" << endl;

cout << "[ 1 ] Gauss" << endl;

cout << "[ 2 ] Jacobi" << endl;

cout << "[ 3 ] Gauss-Seidel" << endl;

cout << "[ 4 ] Determinant" << endl;

cout << "[ 0 ] Iesi din consola" << endl;

cout << "Comanda >> ";

cin >> i;

float x[N]; *// Vectorul pentru a stoca solutiile*

switch (i) {

case 1:

Gauss();

cout << "Verificare solutie Gauss: " << (verificaSolutie(B) ? "Solutie valida" : "Solutie invalida") << endl;

break;

case 2:

jacobi();

cout << "Verificare solutie Jacobi: " << (verificaSolutie(B) ? "Solutie valida" : "Solutie invalida") << endl;

break;

case 3:

gaussseidel();

cout << "Verificare solutie Gauss-Seidel: " << (verificaSolutie(B) ? "Solutie valida" : "Solutie invalida") << endl;

break;

case 4:

cout << "Determinantul matricei sistemului: " << determinant() << endl;

break;

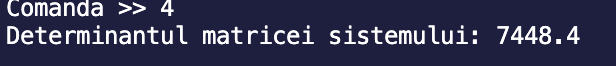
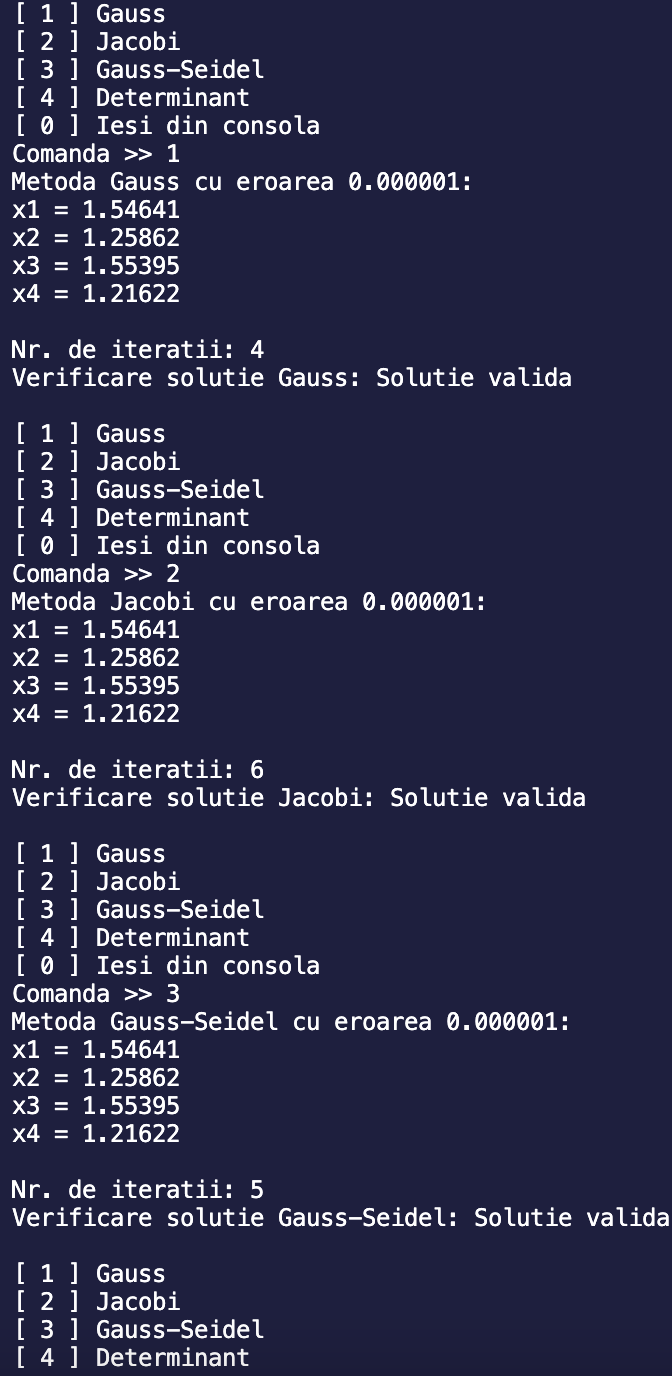
case 0:

return 0;

}

}

}



# CONCLUZIA

În cadrul acestui laborator, am explorat metode variate pentru rezolvarea sistemelor de ecuații liniare algebrice (SELA), evidențiind aspecte distincte ale fiecărei tehnici. Metoda lui Gauss s-a dovedit a fi o unealtă eficientă pentru rezolvarea directă a sistemelor de ecuații, folosind operații elementare de linie. Metoda lui Jacobi și metoda iterativă Gauss-Saidel au oferit perspective interesante asupra abordărilor iterative, cu accent pe convergența către soluții precise.

În paralel, am investigat determinantul matricei asociate sistemului, identificându-l ca o măsură crucială a proprietăților sistemului de ecuații. Valoarea determinantului furnizează informații despre consistența și invertibilitatea matricei, contribuind astfel la înțelegerea esențială a sistemului în cauză.

Prin acest laborator, am dobândit competențe în utilizarea unor metode variate de rezolvare a SELA și în interpretarea semnificației determinantului asociat. Aceste abilități reprezintă nu doar instrumente esențiale în algebră liniară, ci și resurse fundamentale pentru abordarea și rezolvarea problemelor practice dintr-o varietate de domenii, cum ar fi fizica, ingineria sau informatica. Astfel, cunoștințele acumulate în acest laborator ne vor servi ca bază solidă pentru investigarea ulterioară a problemelor complexe și pentru aplicarea acestor concepte în contexte practice specifice.