

**Eliminarea Gaussiană**

Clasă: K12

**Cuprins**

[Definiţie 2](#_Toc125493500)

[Metoda de rezolvare a metodei 3](#_Toc125493501)

[Implementare 5](#_Toc125493502)

[Algoritmul 7](#_Toc125493503)

[Programul c++ 9](#_Toc125493504)

[Exercitiu 1 11](#_Toc125493505)

[Exercitiu 2 12](#_Toc125493506)

[Exercitiu 3 15](#_Toc125493507)

[Exerciţiu 4 (de la examenele BAC) 18](#_Toc125493508)

[Exercitiu 5 19](#_Toc125493509)

[Exercitiu 6 20](#_Toc125493510)

[Exercitiu 7 23](#_Toc125493511)

[Surse 25](#_Toc125493512)

# Definiţie

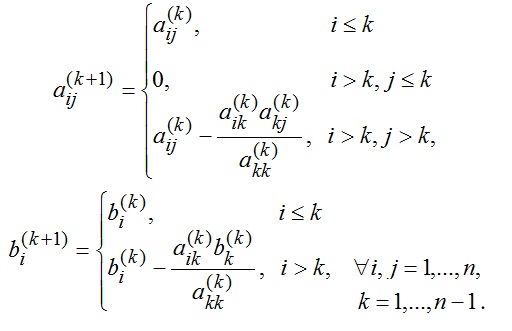
In [matematică](https://ro.alegsaonline.com/art/62814), **eliminarea gaussiană (**denumită și **reducere de rânduri**) este o metodă utilizată pentru a rezolva sisteme de ecuații liniare. Este denumită după Carl **Friedrich Gauss**, un celebru matematician german care a scris despre această metodă, dar nu a inventat-o.

Eliminarea gaussiană este o tehnica pentru transformarea matricei A la forma superior triunghiulara. Matricea de transformare T este o matrice inferior triunghiulara unitara obţinută ca o secvenţă (produs) de transformari inferior triunghiulare elementare de forma T = Tn-1Tn-2 . . . T1, unde matricile Tp sunt inferior triunghiulare, de ordin n de forma:

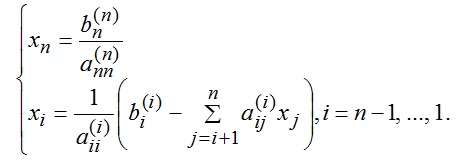
# ****Metoda de rezolvare a metodei****

Notăm, inițial, A(1)=A, b(1)=b, indicele superior reprezentând etapa.

Relațiile de recurență în metoda de eliminare a lui Gauss sunt:



Sistemul se rezolvă prin metoda substituției inverse conform relațiilor:



Eliminarea Gaussiană este o metodă de rezolvare a ecuaţiilor matriciale de forma Ax=b. Algoritmul nu este unul complicat, insă apare destul de des la concursurile de programare şi are aplicaţii interesante.

Să presupunem că avem următorul sistem:

\begin{pmatrix}
a_{11} &  a_{12}  & \ldots & a_{1n}\\
a_{21} &  a_{22}  & \ldots & a_{2n}\\
\vdots & \vdots   & \ddots & \vdots\\
a_{n1} &  a_{n2} & \ldots  & a_{nn}
\end{pmatrix} * 
\begin{pmatrix}
x_{1} \\
x_{2} \\
\vdots \\
x_{n}
\end{pmatrix} = 
\begin{pmatrix}
b_{1} \\
b_{2} \\
\vdots \\
b_{n}
\end{pmatrix}

Pentru a rezolva sistemul vom transforma toate elementele de sub diagonala principală a matricei extinse în 0 pentru a putea scrie fiecare necunoscută doar în funcţie de necunoscutele cu indici mai mari.


\begin{pmatrix}
a_{11} &  a_{12} & \ldots & a_{1n} & b_{1}\\
a_{21} &  a_{22} & \ldots & a_{2n} & b_{2}\\
a_{31} &  a_{32} & \ldots & a_{3n} & b_{3}\\
\vdots & \vdots  & \ddots & \vdots \\
a_{n1} &  a_{n2} & \ldots  & a_{nn} & b_{n}
\end{pmatrix}

\rightarrow

\begin{pmatrix}
a_{11} &  a_{12} & \ldots & a_{1n} & b_{1}\\
0      &  a'_{22} & \ldots & a'_{2n} & b'_{2}\\
0      &  0      & \ldots & a'_{3n} & b'_{3}\\
\vdots & \vdots  & \ddots & \vdots \\
0 &  0 & \ldots  & a'_{nn} & b'_{n}
\end{pmatrix}


Având matricea sub această formă putem să aflăm uşor fiecare necunoscută din ecuaţia în care necunoscutele cu indici mai mici au coeficientul 0:  
![
\:
x_{n}=\frac{b'_{n}}{a'_{nn}}](data:image/gif;base64,R0lGODlhRgAeAPMPAAAAABERESIiIjMzM0RERFVVVWZmZnd3d4iIiJmZmaqqqru7u8zMzN3d3e7u7v///yH5BAAAAAAALAAAAABGAB4AAATx8MlJq7046827/5cAjuSoGGWqSouQTMeyzmTQTAWte+3EvLtg5jBYJIDCpIUgewSUUIrowXhGo7mHAXGNKhQJ7kMRA3eVDAaBeugoFgcE4rA+Z8iPg8KxSfAXAQ4/HweFhodtJAcMeQuMGjcPCFlnAJaXmJmaAHUWBWJ2SgGPMmQJCwqUFYisI1MfCk8LACw3ZG0OtF0nJkUMcQuRDwWMqWcxQlMGqHxYQgxZegpBR3Avg6EzCCh509k6Ao+v3zNWDQQOTeQqR2BbEmSoQF8LcOsgaWu/EqcyBs333LTR04zbAIABNyxq9Igbk4QQI2aIAAA7)  
x_{n-1}=\frac{b'_{n-1}-a'_{n-1n}*x_{n}}{a'_{n-1n-1}}  
 \vdots  
 x_{i}=\frac{b'_{i}-\sum\limits_{j=i+1}^n a'_{ij}*x_{j}}{a'_{ii}}

Acum că ştim să aflăm necunoscutele din forma triunghiulară a matricei ne mai rămâne doar să transformăm matricea.  
Pentru a transforma matricea în formă triunghiulară vom aplica două operaţii:  
L_{i} \longleftrightarrow L_{j} : interschimbarea a două linii  
L_{j} \longleftarrow L_{j}+a*L_{i} unde L_{i} este o linie a matricei extinse.

De exemplu:


\begin{pmatrix}
2  &  1 & -1 & 8 \
-3 & -1 & 2 & -11 \
-2 & 1 & 2 & -3
\end{pmatrix} \longrightarrow

\begin{pmatrix}
2 & 1 & -1 & 8 \
0 & \frac{1}{2} & \frac{1}{2} & 1 \
0 & 2 & 1 & 5
\end{pmatrix} \longrightarrow
\begin{pmatrix}
2 & 1 & -1 & 8 \
0 & \frac{1}{2} & \frac{1}{2} & 1 \
0 & 0 & -1 & 1
\end{pmatrix}
  
 Pentru a obţine a doua matrice am înmulţit prima linie cu \frac{3}{2} şi am adunat-o la a doua linie, iar apoi am adunat-o la ultima linie(\frac{2}{2}=1). Pentru a obţine ultima matrice am înmulţit a doua linie cu -\frac{2}{\frac{1}{2}}=-4.

Cum se poate observa şi din exemplu la fiecare pas construim o coloană şi o linie din matricea finală, coloana fiind umplută cu 0 sub linia fixată. Să presupunem că vrem să transformăm toate elementele de sub linia i aflate pe coloana j în 0. Pentru fiecare linie k ( k>i ) vom înmulţi linia i cu -\frac{a_{kj}}{a_ij} şi o vom aduna la linia k astfel elementul de pe coloana j transformându-se în 0. În cazul în care a_{ij}=0 trebuie să căutăm o linie k(k > i) astfel încât a_{kj}\neq 0. În cazul în care această linie nu există, sistemul nu are soluţie. Aplicând aceşti pasi vom ajunge în final la o matrice triunghiulară din care vom afla necunoscutele. Complexitatea algoritmului este O(N^3)

# Implementare

Codul de mai jos rezolva si cazul când avem mai multe ecuaţii decât necunoscute.

void elim(int n,int m,double s[][]) {*//sistem cu n ecuatii m necunoscute*

for(int i=1,j=1,k;i<=n && j<=m;) {

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

if(s[k][j]!=0) break;*//cautam o linie pe care sa o folosim pentru a forma zerouri pe coloana j*

if(k>n) {*//nu am gasit nicio linie pentru care s[i][j] e nenul deci trecem la urmatoarea coloana linia i nefiind finala*

++j;

continue;

}

if(k!=i)for(int l=1; l<=m+1; ++l) swap(s[i][l],s[k][l]);*//interschimbam liniile pentru a avea pe linia i si coloana j un element nenul*

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

for(int l=m+1; l>=j; --l)

s[k][l]-=((s[k][j]\*s[i][l])/s[i][j]);*//aplicam transformarea pentru fiecare linie mai mare ca i pentru a avea 0 pe coloana j sub linia i*

++i; ++j;

}

*//aflam necunoscutele*

for(int i=n; i;--i)

for(int j=1; j<=m+1; ++j) if(fabs(s[i][j])>EPS) {

*//pentru ca e posibil sa avem mai multe ecuatii decat necunoscute*

*//cautam pe fiecare linie primul coeficient nenul, acestia aparand de la dreapta la stanga*

if(j==m+1) {*//linia nu are coeficienti nenuli deci nu avem solutie*

g<<"Imposibil";

exit(0);

}

x[j]=s[i][m+1];

for(int k=j+1; k<=m; ++k) x[j]-=s[i][k]\*x[k];

x[j]/=s[i][j];

break;*//trecem la linia precedenta*

}

}

# Algoritmul

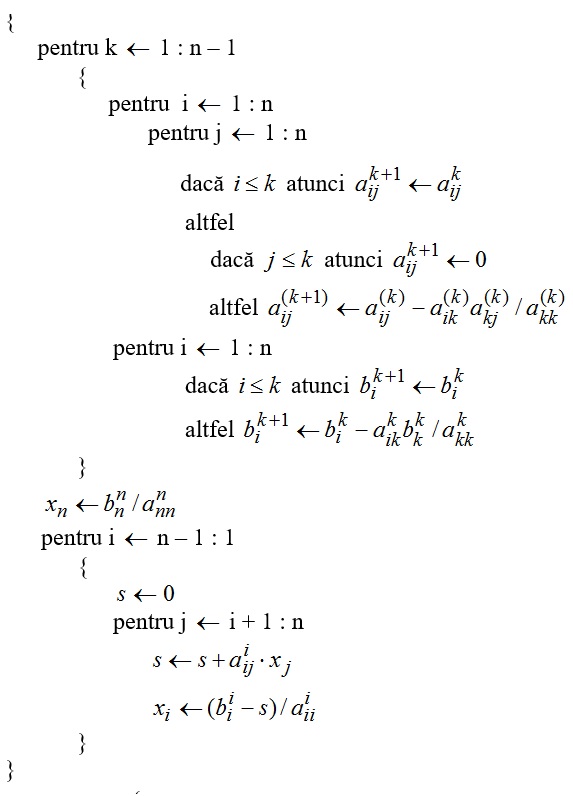
Algoritmul asociat metodei de eliminare Gauss este:

Intrări:

* n = numărul de ecuații și necunoscute ale sistemului
* A = matricea sistemului
* b = vectorul termenilor liberi

Ieșiri:

* x = vectorul soluție



# Programul c++

#include "stdafx.h"

#include

using namespace std;

#define max 15

void main(void)

{

float s;

float a[max + 1][max + 1][max + 1], b[max + 1][max + 1], x[max + 1];

int n, i, j, k;

cout << "dati dimensiunea matricii" << endl; cin >> n;

cout << "dati matricea sistemului " << endl;

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

{

cout << "a[" << i << j << "]=";

cin >> a[i][j][1];

}

cout << endl;

cout << "dati vectorul termenilor liberi " << endl;

for (i = 1; i <= n; i++)

{

cout << "b[" << i << "]=";

cin >> b[i][1];

}

cout << endl;

for (k = 1; k <= n - 1; k++)

{

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

{

if (i <= k)a[i][j][k + 1] = a[i][j][k];

else if (j <= k)a[i][j][k + 1] = 0;

else a[i][j][k + 1] = a[i][j][k] - a[i][k][k] \* a[k][j][k] / a[k][k][k];

}

for (i = 1; i <= n; i++)

if (i <= k)b[i][k + 1] = b[i][k];

else b[i][k + 1] = b[i][k] - a[i][k][k] \* b[k][k] / a[k][k][k];

}

x[n] = b[n][n] / a[n][n][n];

for (i = n - 1; i >= 1; i--)

{

s = 0;

for (j = i + 1; j <= n; j++)

s = s + a[i][j][i] \* x[j];

x[i] = (b[i][i] - s) / a[i][i][i];

}

cout << "Solutia aproximativa este:" << endl;

for (i = 1; i <= n; i++)

cout << x[i] << ' ';

cout << endl;

system("pause");

}

# Exercitiu 1

Se considera urmatorul system:



Coeficienții sunt scrieți sub formă de tabel, iar în dreapta într-o coloană separată - membri liberi. Coloana cu membri liberi este separată pentru comoditate.Matricea care include această coloană se numește extinsă.

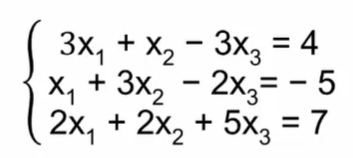


În plus, matricea principală cu coeficienți trebuie redusă la forma triunghiulară superioară. Acesta este punctul principal de rezolvare a sistemului prin metoda Gauss. Pur și simplu, după anumite manipulări, matricea ar trebui să arate astfel, astfel încât să existe doar zerouri în partea sa din stânga jos:

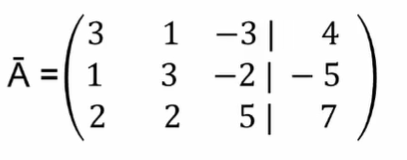


# Exercitiu 2

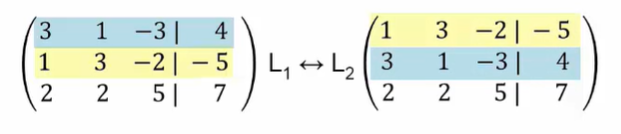
Se considera urmatorul sistem:

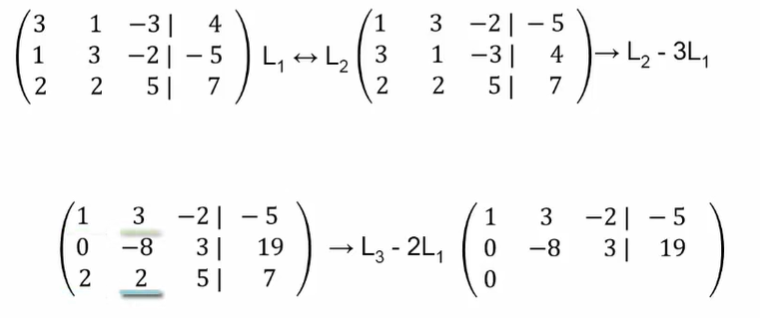


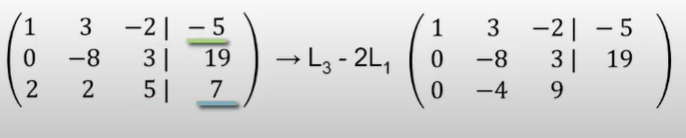
Matricea extinsa asociata sistemului este:



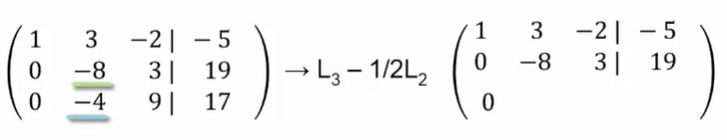
Intersichimbam linia L1  cu linia L2 pentru a avea peprima linie pe prima pozitie cea mai mica valoare

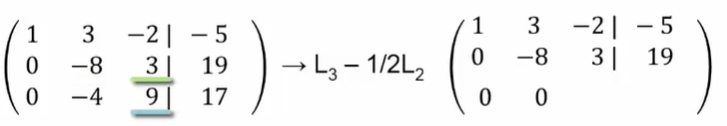


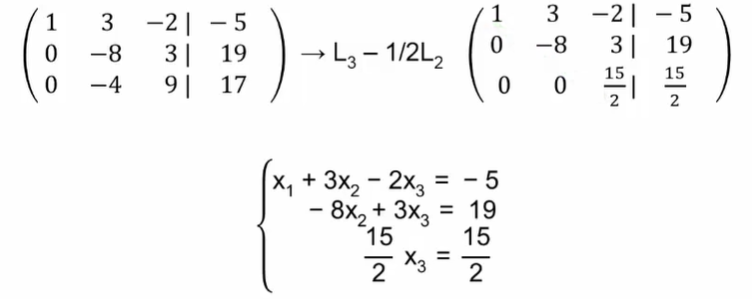


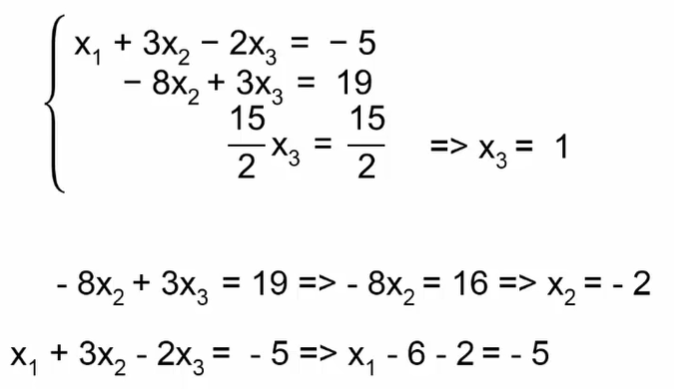












# Exercitiu 3

Să se aplice metoda de eliminare a lui Gauss pentru rezolvarea sistemului:



**Rezolvare**:

Matricea A asociată sistemului (în etapa 1) și vectorul termenilor liberi b sunt:





Soluțiile sunt:

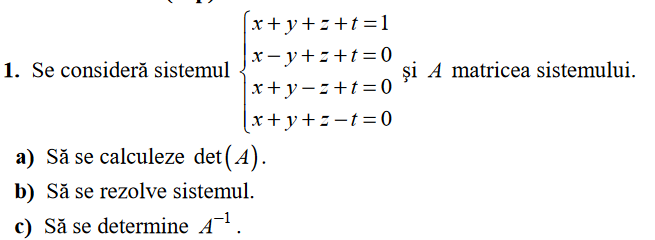


adică soluția sistemului este:

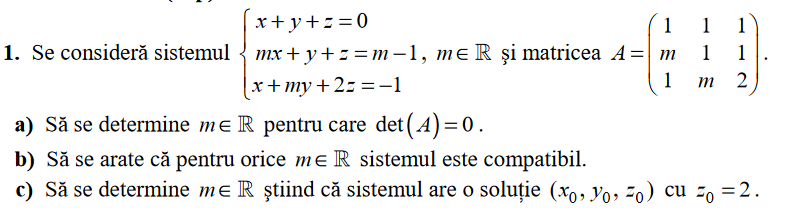


# Exerciţiu 4 (de la examenele BAC)

Bacalaureat 2018



Bacalaureat 2017



# Exercitiu 5

Se considera urmatorul system:



Coeficienții sunt scrieți sub formă de tabel, iar în dreapta într-o coloană separată - membri liberi. Coloana cu membri liberi este separată pentru comoditate.Matricea care include această coloană se numește extinsă.



În plus, matricea principală cu coeficienți trebuie redusă la forma triunghiulară superioară. Acesta este punctul principal de rezolvare a sistemului prin metoda Gauss. Pur și simplu, după anumite manipulări, matricea ar trebui să arate astfel, astfel încât să existe doar zerouri în partea sa din stânga jos:



# 

# Exercitiu 6

Se considera urmatorul sistem:

Diagram

Description automatically generated with low confidence

Matricea extinsa asociata sistemului este:

A picture containing text, clock

Description automatically generated

Intersichimbam linia L1  cu linia L2 pentru a avea peprima linie pe prima pozitie cea mai mica valoare

Diagram

Description automatically generated

Calendar

Description automatically generated with medium confidence

A picture containing text, clock, gauge

Description automatically generated



A picture containing logo

Description automatically generated

Logo

Description automatically generated

Diagram

Description automatically generated

Text, letter

Description automatically generated

# Exercitiu 7

Să se aplice metoda de eliminare a lui Gauss pentru rezolvarea sistemului:



**Rezolvare**:

Matricea A asociată sistemului (în etapa 1) și vectorul termenilor liberi b sunt:





Soluțiile sunt:



adică soluția sistemului este:



# Surse

<https://en.wikipedia.org/wiki/Regular_polygon>

<https://www.youtube.com/watch?v=qetSusATv2w>