

**Gaussova eliminácia**

Trieda školy: K12

**Obsah**

[Definícia 2](#_Toc125623445)

[Spôsob riešenia metódy 3](#_Toc125623446)

[Implementácia 5](#_Toc125623447)

[Algoritmus 7](#_Toc125623448)

[Program c++ 9](#_Toc125623449)

[Cvičenie 1 11](#_Toc125623450)

[Cvičenie 2 12](#_Toc125623451)

[Cvičenie 3 15](#_Toc125623452)

[Cvičenie 4 (zo skúšok BAC) 18](#_Toc125623453)

[Cvičenie 5 19](#_Toc125623454)

[Cvičenie 6 20](#_Toc125623455)

[Cvičenie 7 23](#_Toc125623456)

[Zdroje 25](#_Toc125623457)

# Definícia

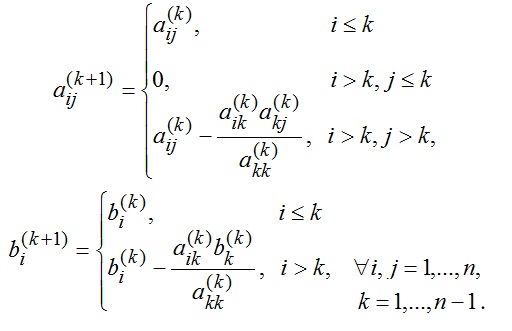
V matematike je Gaussova eliminácia (nazývaná aj redukcia riadkov) metóda používaná na riešenie sústav lineárnych rovníc. Je pomenovaná po Carlovi Friedrichovi Gaussovi, slávnom nemeckom matematikovi, ktorý o tejto metóde písal, ale nevynašiel ju.

Gaussova eliminácia je technika transformácie matice A na horný trojuholníkový tvar. Transformačná matica T je unitárna dolná trojuholníková matica získaná ako postupnosť (súčin) elementárnych dolných trojuholníkových transformácií tvaru T = Tn-1 Tn-2 . . . T1 , kde matice Tp sú dolné trojuholníkové, v tvare rádu n:

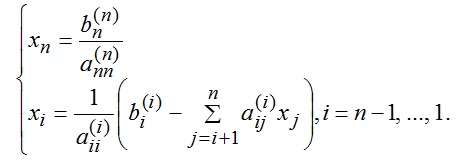
# ****Spôsob riešenia metódy****

Na začiatku si všimneme, že A(1)=A, b(1)=b, pričom horný index predstavuje stupeň.

Vzťahy rekurencie v Gaussovej eliminačnej metóde sú:



Systém sa rieši metódou inverznej substitúcie podľa vzťahov:



Gaussova eliminácia je metóda na riešenie maticových rovníc v tvareAx=b . Algoritmus nie je zložitý, ale pomerne často sa objavuje v programátorských súťažiach a má zaujímavé aplikácie.

Predpokladajme, že máme nasledujúci systém:

\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}

Pri riešení sústavy transformujeme všetky prvky pod hlavnou diagonálou rozšírenej matice na 0, aby sme každú neznámu mohli zapísať len v termínoch neznámych s vyššími indexmi.


\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}


Ak máme maticu v tomto tvare, môžeme ľahko nájsť každú neznámu v rovnici, kde neznáme s nižšími indexmi majú koeficient 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}}

Teraz, keď už vieme, ako nájsť neznáme z trojuholníkového tvaru matice, zostáva už len transformovať maticu.

Na transformáciu matice do trojuholníkového tvaru použijeme dve operácie:  
L_{i} \longleftrightarrow L_{j} : zámena dvoch riadkov  
L_{j} \longleftarrow L_{j}+a*L_{i} kdeL_{i} je riadok rozšírenej matice.

Napríklad:


\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}


Aby som získal druhú maticu, vynásobil som prvý riadok číslom\frac{3}{2} a pridal som ho k druhému riadku a potom som ho pridal k poslednému riadku (\frac{2}{2}=1 ). Na získanie poslednej matice som vynásobil druhý riadok číslom-\frac{2}{\frac{1}{2}}=-4 .

Ako vidno z príkladu, v každom kroku vytvoríme stĺpec a riadok z výslednej matice, pričom stĺpec je vyplnený 0 pod pevnou čiarou. Predpokladajme, že chceme previesť všetky prvky pod riadkom i v stĺpci j na 0. Pre každý riadok k ( k>i ) vynásobíme riadok i číslom-\frac{a_{kj}}{a_ij} a pripočítame ho k riadku k, čím sa prvok v stĺpci j zmení na 0. V prípadea_{ij}=0 musíme hľadať taký riadok k(k > i), abya_{kj}\neq 0 . Ak tento riadok neexistuje, systém nemá riešenie. Aplikáciou týchto krokov nakoniec dostaneme trojuholníkovú maticu, z ktorej nájdeme neznáme. Zložitosť algoritmu je O(N^3)

# Implementácia

Nižšie uvedený kód rieši aj prípad, keď máme viac rovníc ako neznámych.

void elim(int n,int m,double s[][]) *{//systém s n neznámymi m rovnicami*

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

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

if(s[k][j]!=0) break;*// hľadáme riadok, ktorý použijeme na vytvorenie núl v stĺpci j*

if(k>n) *{// Nenašiel som žiadny riadok, pre ktorý by s[i][j] bola nula, takže sa presunieme do ďalšieho stĺpca, pričom riadok i nie je posledný*

++j;

pokračovať;

}

if(k!=i)for(int l=1; l<=m+1; ++l) swap(s[i][l],s[k][l]);*//vymeníme riadky, aby sme mali nulový prvok na riadku i a v stĺpci j*

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]);*//aplikujeme transformáciu pre každý riadok väčší ako i, aby sme mali 0 v stĺpci j pod riadkom i*

++i; ++j;

}

*//poznávame neznáme*

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

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

*//pretože je možné mať viac rovníc ako neznámych*

*//hľadáme prvý nulový koeficient na každom riadku, ktorý sa objavuje sprava doľava*

if(j==m+1) *{//priamka nemá nenulové koeficienty, takže nemáme riešenie*

g<<"Nemožné";

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;*//prechádzame na predchádzajúci riadok*

}

}

# Algoritmus

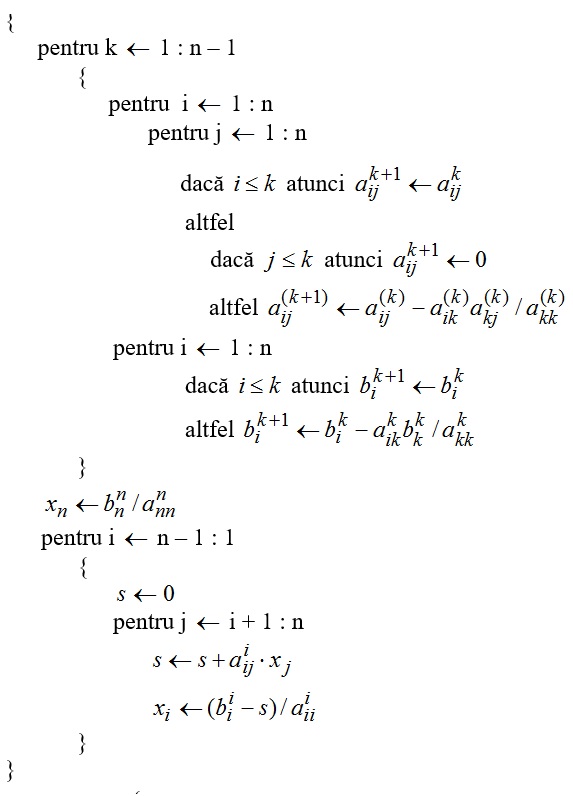
Algoritmus spojený s Gaussovou eliminačnou metódou je:

Vstupy:

* n = počet rovníc a neznámych systému
* A = systémová matica
* b = vektor voľných členov

Výstupy:

* x = vektor riešenia



# Program 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++)

{

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

inak ak (j <= k)a[i][j][k + 1] = 0;

inak 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++)

ak (i <= k)b[i][k + 1] = b[i][k];

inak 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 << "Približné riešenie je:" << endl;

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

cout << x[i] << ' ';

cout << endl;

system("pause");

}

# Cvičenie 1

Uvažuje sa o tomto systéme:



Koeficienty sú zapísané v tabuľkovej forme a vpravo v samostatnom stĺpci - voľné členy. Stĺpec s voľnými členmi je kvôli pohodliu oddelený. Pole, ktoré obsahuje tento stĺpec, sa nazýva rozšírené.

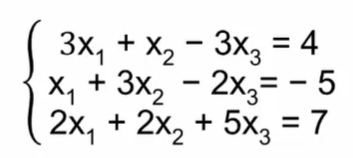


Okrem toho sa musí hlavná koeficientová matica redukovať na horný trojuholníkový tvar. To je hlavný bod riešenia systému Gaussovou metódou. Jednoducho, po niekoľkých manipuláciách by mala matica vyzerať takto, aby v jej ľavej dolnej časti boli len nuly:

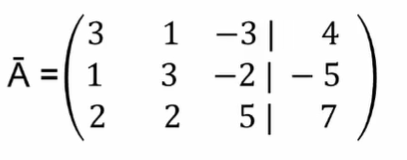


# Cvičenie 2

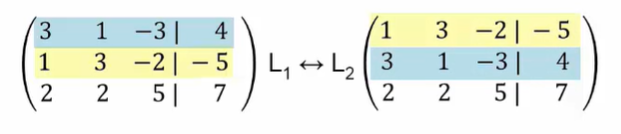
Uvažuje sa o tomto systéme:

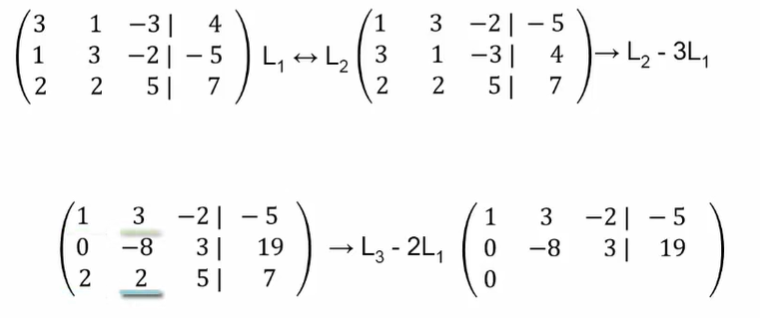


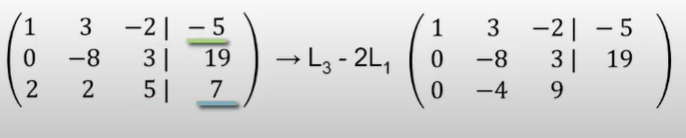
Rozšírená matica spojená so systémom je:



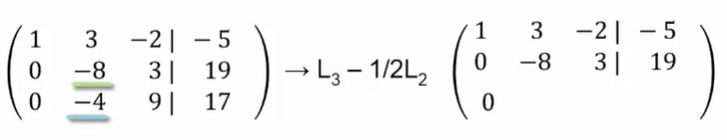
Zmenili sme riadok L1  s riadkom L2 , aby sme mali najnižšiu hodnotu na prvom riadku na prvej pozícii

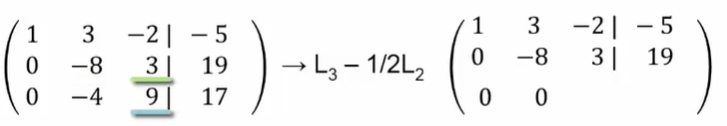


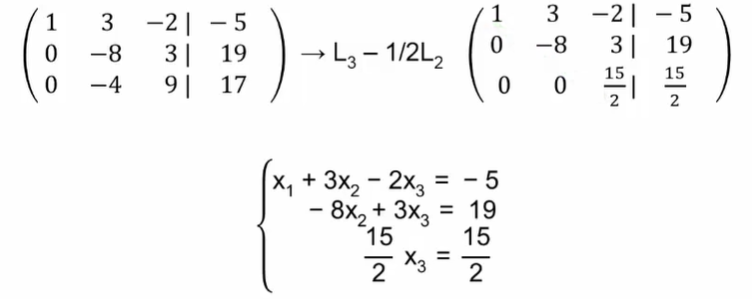


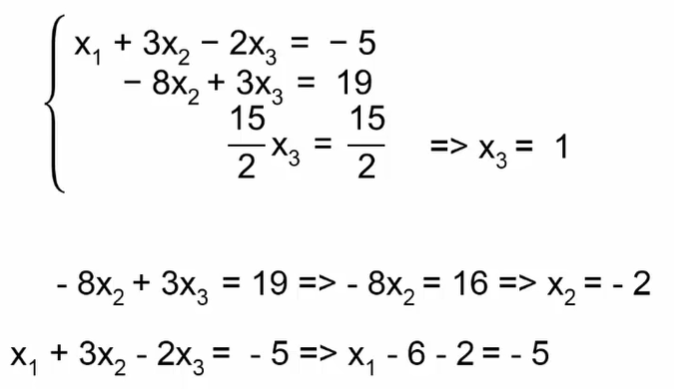












# Cvičenie 3

Na riešenie systému použite Gaussovu eliminačnú metódu:



**Riešenie**:

Matica A priradená k systému (v kroku 1) a vektor voľných členov b sú:





Riešenia sú:

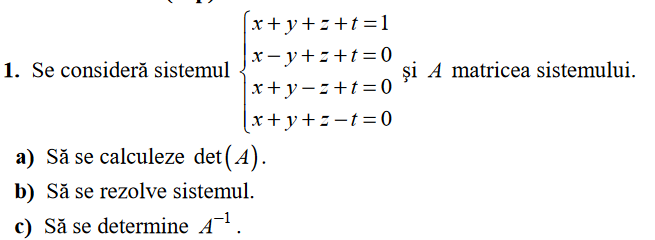


to znamená, že riešenie systému je:

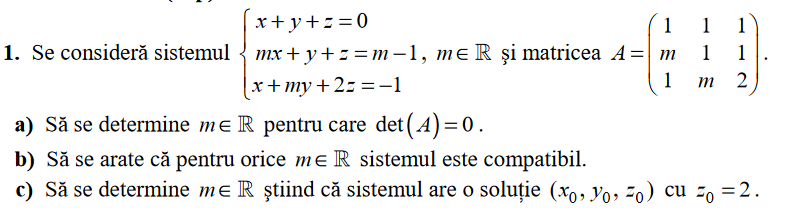


# Cvičenie 4 (zo skúšok BAC)

Skúška BAC 2018



Skúška BAC 2017



# Cvičenie 5

Uvažuje sa o tomto systéme:



Koeficienty sú zapísané v tabuľkovej forme a vpravo v samostatnom stĺpci - voľné členy. Stĺpec s voľnými členmi je kvôli pohodliu oddelený. Pole, ktoré obsahuje tento stĺpec, sa nazýva rozšírené.



Okrem toho sa musí hlavná koeficientová matica redukovať na horný trojuholníkový tvar. To je hlavný bod riešenia systému Gaussovou metódou. Jednoducho, po určitej manipulácii by mala matica vyzerať takto, aby v jej ľavej dolnej časti boli len nuly:



# 

# Cvičenie 6

Uvažuje sa o tomto systéme:

Diagram

Description automatically generated with low confidence

Rozšírená matica spojená so systémom je:

A picture containing text, clock

Description automatically generated

Zmenili sme riadok L1  s riadkom L2 , aby sme mali najnižšiu hodnotu na prvom riadku na prvej pozícii

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

# Cvičenie 7

Na riešenie systému použite Gaussovu eliminačnú metódu:



**Riešenie**:

Matica A spojená so systémom (v kroku 1) a vektor voľných členov b sú:





Riešenia sú:



to znamená, že riešenie systému je:



# Zdroje

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

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