

**Gaussova eliminace**

Třída školy: K12

**Obsah**

[Definice 2](#_Toc125623445)

[Způsob řešení metody 3](#_Toc125623446)

[Provádění 5](#_Toc125623447)

[Algoritmus 7](#_Toc125623448)

[Program c++ 9](#_Toc125623449)

[Cvičení 1 11](#_Toc125623450)

[Cvičení 2 12](#_Toc125623451)

[Cvičení 3 15](#_Toc125623452)

[Cvičení 4 (ze zkoušek BAC) 18](#_Toc125623453)

[Cvičení 5 19](#_Toc125623454)

[Cvičení 6 20](#_Toc125623455)

[Cvičení 7 23](#_Toc125623456)

[Zdroje 25](#_Toc125623457)

# Definice

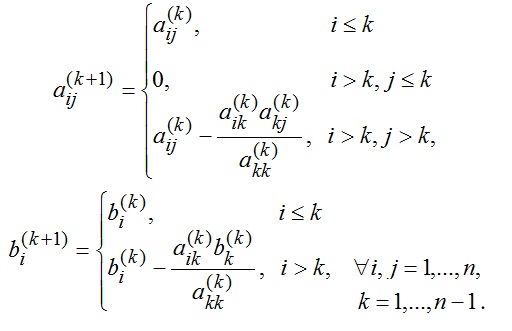
V matematice je Gaussova eliminace (nazývaná také redukce řádků) metoda používaná k řešení soustav lineárních rovnic. Je pojmenována po Carlu Friedrichu Gaussovi, slavném německém matematikovi, který o této metodě psal, ale nevynalezl ji.

Gaussova eliminace je technika transformace matice A do horního trojúhelníkového tvaru. Transformační matice T je unitární dolní trojúhelníková matice získaná jako posloupnost (součin) elementárních dolních trojúhelníkových transformací tvaru T = T Tn-1n-2 . . . T1 , kde matice Tp jsou dolní trojúhelníkové, řád n ve tvaru:

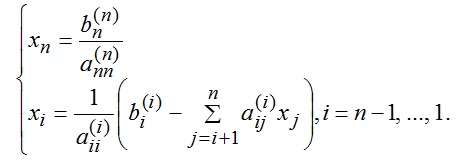
# ****Způsob řešení metody****

Zpočátku si všimneme, že A(1)=A, b(1)=b, přičemž horní index představuje stupeň.

Vztahy rekurence v Gaussově eliminační metodě jsou:



Systém se řeší metodou inverzní substituce podle vztahů:



Gaussova eliminace je metoda řešení maticových rovnic tvaruAx=b . Nejedná se o složitý algoritmus, ale v programátorských soutěžích se objevuje poměrně často a má zajímavé aplikace.

Předpokládejme, že máme následující 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}

Při řešení soustavy převedeme všechny prvky pod hlavní diagonálou rozšířené matice na 0, abychom mohli každou neznámou zapsat pouze v termínech neznámých s vyššími indexy.


\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}


Když máme matici v tomto tvaru, můžeme snadno najít každou neznámou v rovnici, kde neznámé s nižšími indexy mají 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}}

Nyní, když víme, jak zjistit neznámé z trojúhelníkového tvaru matice, zbývá už jen matici transformovat.

Pro transformaci matice do trojúhelníkového tvaru použijeme dvě operace:  
L_{i} \longleftrightarrow L_{j} : záměna dvou čar  
L_{j} \longleftarrow L_{j}+a*L_{i} kdeL_{i} je řádek rozšířené matice.

Například:


\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}


Abych získal druhou matici, vynásobil jsem první řádek číslem\frac{3}{2} a přičetl jsem ho k druhému řádku a poté jsem ho přičetl k poslednímu řádku (\frac{2}{2}=1 ). Pro získání poslední matice jsem druhý řádek vynásobil-\frac{2}{\frac{1}{2}}=-4 .

Jak je patrné z příkladu, v každém kroku sestavíme sloupec a řádek z výsledné matice, přičemž sloupec je vyplněn 0 pod pevnou čarou. Předpokládejme, že chceme všechny prvky pod řádkem i ve sloupci j převést na 0. Pro každý řádek k ( k>i ) vynásobíme řádek i číslem-\frac{a_{kj}}{a_ij} a přičteme jej k řádku k, čímž se prvek ve sloupci j změní na 0. V případěa_{ij}=0 musíme hledat řádek k(k > i) takový, žea_{kj}\neq 0 . Pokud tento řádek neexistuje, soustava nemá řešení. Aplikací těchto kroků nakonec získáme trojúhelníkovou matici, ze které zjistíme neznámé. Složitost algoritmu je O(N^3)

# Provádění

Níže uvedený kód řeší také případ, kdy máme více rovnic než neznámých.

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

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

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

if(s[k][j]!=0) break;*// hledáme řádek, který použijeme k vytvoření nul ve sloupci j*

if(k>n) *{// Nenašel jsem žádný řádek, pro který by s[i][j] bylo nulové, takže přejdeme na další sloupec, přičemž řádek i není poslední.*

++j;

pokračovat;

}

if(k!=i)for(int l=1; l<=m+1; ++l) swap(s[i][l],s[k][l]);*//vyměníme řádky tak, aby na řádku i a ve sloupci j byl nulový prvek.*

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]);*//použijeme transformaci pro každý řádek větší než i, aby ve sloupci j pod řádkem i byla 0*

++i; ++j;

}

*//poznáváme neznámé*

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

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

*//protože je možné mít více rovnic než neznámých*

*//hledáme první nulový koeficient na každém řádku, který se objevuje zprava doleva.*

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

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;*//přejdeme na předchozí řádek*

}

}

# Algoritmus

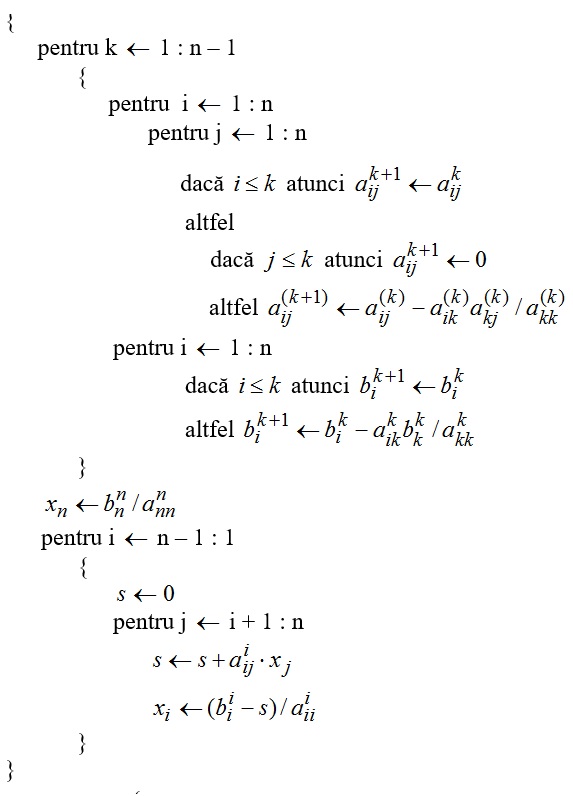
Algoritmus spojený s Gaussovou eliminační metodou je:

Vstupy:

* n = počet rovnic a neznámých systému
* A = matice systému
* b = vektor volných členů

Výstupy:

* x = vektor řešení



# 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++)

{

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

else if (j <= k)a[i][j][k + 1] = 0;

jinak 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];

jinak 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 << "Přibližné řešení je:" << endl;

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

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

cout << endl;

system("pause");

}

# Cvičení 1

Uvažuje se následující systém:



Koeficienty jsou zapsány v tabulce a vpravo v samostatném sloupci - volné členy. Sloupec s volnými členy je pro větší pohodlí oddělen. Pole, které obsahuje tento sloupec, se nazývá rozšířené.

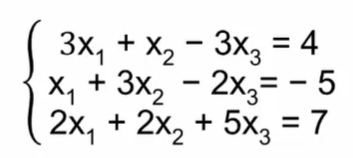


Kromě toho je třeba redukovat hlavní matici koeficientů na horní trojúhelníkový tvar. To je hlavní bod řešení systému Gaussovou metodou. Zjednodušeně řečeno, po několika manipulacích by matice měla vypadat tak, aby v její levé dolní části byly pouze nuly:

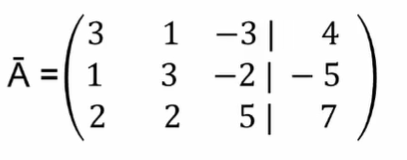


# Cvičení 2

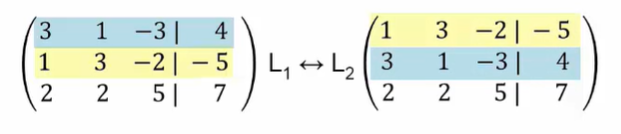
Uvažuje se následující systém:

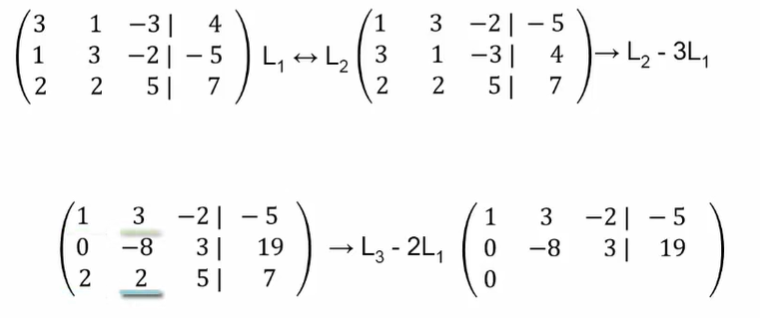


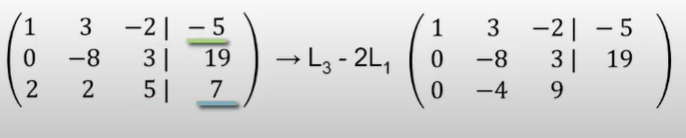
Rozšířená matice spojená se systémem je:



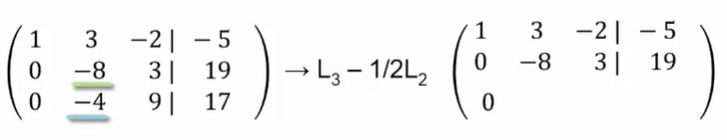
Měnili jsme řádek L1  s řádkem L2 tak, aby na prvním řádku byla nejnižší hodnota na první pozici.

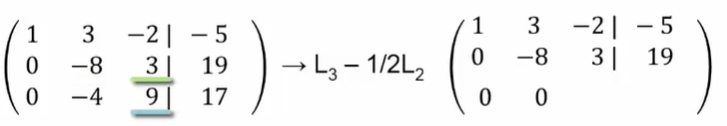


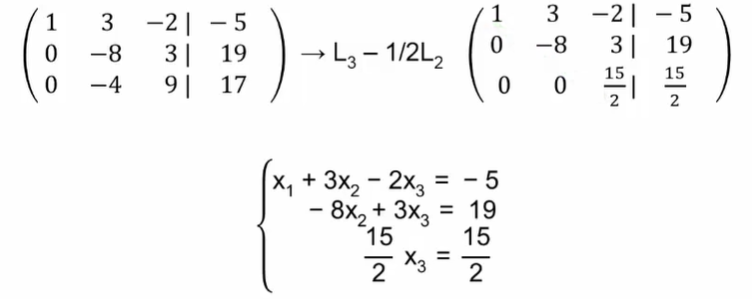


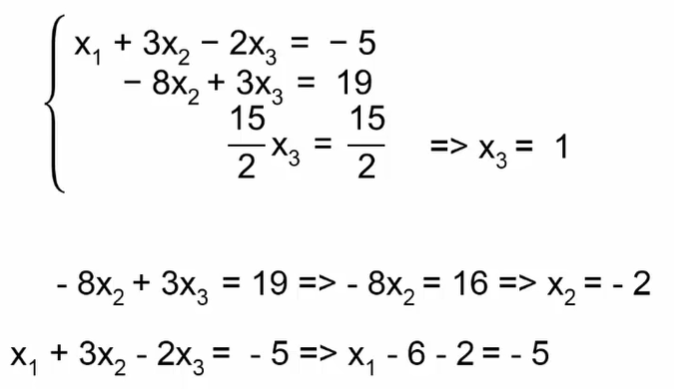












# Cvičení 3

K řešení systému použijte Gaussovu eliminační metodu:



**Řešení**:

Matice A přiřazená systému (v kroku 1) a vektor volných členů b jsou:





Řešení jsou následující:

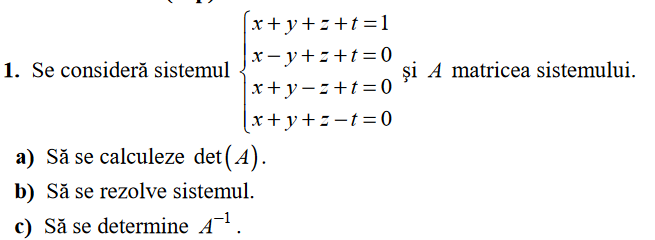


to znamená, že řešení systému je:

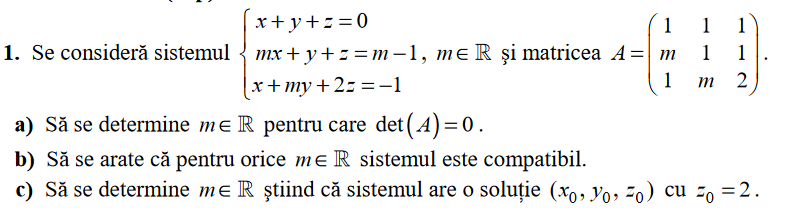


# Cvičení 4 (ze zkoušek BAC)

Zkouška BAC 2018



Zkouška BAC 2017



# Cvičení 5

Uvažuje se následující systém:



Koeficienty jsou zapsány v tabulce a vpravo v samostatném sloupci - volné členy. Sloupec s volnými členy je pro větší pohodlí oddělen. Pole, které obsahuje tento sloupec, se nazývá rozšířené.



Kromě toho je třeba redukovat hlavní matici koeficientů na horní trojúhelníkový tvar. To je hlavní bod řešení soustavy Gaussovou metodou. Zjednodušeně řečeno, po určité manipulaci by matice měla vypadat tak, aby v její levé dolní části byly pouze nuly:



# 

# Cvičení 6

Uvažuje se následující systém:

Diagram

Description automatically generated with low confidence

Rozšířená matice spojená se systémem je:

A picture containing text, clock

Description automatically generated

Měnili jsme řádek L1  s řádkem L2 tak, aby na prvním řádku byla nejnižší hodnota na první pozici.

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

# Cvičení 7

K řešení systému použijte Gaussovu eliminační metodu:



**Řešení**:

Matice A spojená se systémem (v kroku 1) a vektor volných členů b jsou:





Řešení jsou následující:



to znamená, že řešení systému je:



# Zdroje

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

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