

Digitální učební materiál



Číslo projektu:	CZ.1.07/1.5.00/34.0548
Název školy:	Gymnázium, Trutnov, Jiráskovo náměstí 325
Název materiálu:	VY_32_INOVACE_154_IVT
Autor:	Ing. Pavel Bezděk
Tematický okruh:	Algoritmy
Datum tvorby:	Srpen 2013
Ročník:	4. ročník a oktáva
Anotace:	Algoritmus XIV. – Algoritmus řešení soustav lineárních rovnic I. - Gaussova eliminace 1
Metodický pokyn:	Při výuce nutno postupovat individuálně. Části DUM – „ Pro hloubavé“ jsou určeny pro zájemce o studium na technických a matematicko-fyzikálních oborech vysokých škol.
Pokud není uvedeno jinak, je použitý materiál z vlastních zdrojů autora DUM.	



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Autor	Ing. Pavel Bezděk		
Vytvořeno dne	31. 8. 2013		
Odpilotováno dne	10. 3. 2014	ve třídě	8.Y
Vzdělávací oblast	Informatika a informační a komunikační technologie		
Vzdělávací obor	Informatika a výpočetní technika		
Tematický okruh	Algoritmus		
Téma	Algoritmus XIV. – Algoritmus řešení soustav lineárních rovnic I. - Gaussova eliminace 1		
Klíčová slova	Algoritmus, Gaussova eliminace, matice		

Řešení soustav lineárních rovnic

Gaussova eliminace

Gaussova eliminace

Řešíme např. soustavu rovnic:

$$x + y + z + t = 10$$

$$x - y + z - t = -2$$

$$2x + 3y + z - 2t = 3$$

$$x - 2y - 3z + 2t = -4$$

Nejprve předeme zápis soustavy rovnic na maticový tvar a pak upravujeme řádky matice až budou pod hlavní diagonálou samé nuly, vznikne trojúhelníková matice.

Převádíme do maticového tvaru

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 10 \\ 1 & -1 & 1 & -1 & -2 \\ 2 & 3 & 1 & -2 & 3 \\ 1 & -2 & -3 & 2 & -4 \end{bmatrix}$$

S řádky matice můžeme provádět tyto úpravy:

- **Násobit (popř. dělit) libovolný řádek matice nenulovým číslem.**
- **Prohazovat mezi sebou libovolné řádky matice.** (Např. zaměnit 1. řádek s 3. řádkem, 2. řádek s 5. řádkem atd.)
- **Přičítat (odečítat) libovolný řádek od jiného řádku.** (Tedy i libovolný (nenulový) násobek jednoho řádku, od libovolného (nenulového) násobku jiného řádku.)
- **Je-li některý řádek vytvořen součtem násobků jiných řádků, je možné ho vymazat.** (Po úpravách by byl stejný, jako některý z řádků a po dalších úpravách by obsahoval samé nuly.)

Nejprve vytvoříme nuly v 1. sloupci

Přičítáme nebo odečítáme násobky 1. řádku postupně k ostatním řádkům, tak aby místo 1. prvku v každém řádku byla nula(kromě 1. řádku).

- 1. řádek odečteme od 2. řádku**
- Dvojnásobek 1.řádku odečteme od 3. řádku**
- 1. řádek odečteme od 4. řádku**

(Kdyby se hodil se pro postupné přičítání nebo odečítání od jiných řádků lépe jiný řádek, zaměníme ho nejprve s prvním řádkem.)

Vytvoření nul v 1. sloupci.

2. řádek vydělíme (-2).

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 10 \\ 0 & -2 & 0 & -2 & -12 \\ 0 & 1 & -1 & -4 & -17 \\ 0 & -3 & -4 & 1 & -14 \end{bmatrix} \quad /(-2)$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 10 \\ 0 & 1 & 0 & 1 & 6 \\ 0 & 1 & -1 & -4 & -17 \\ 0 & -3 & -4 & 1 & -14 \end{bmatrix}$$

Nyní musíme dostat nuly se 2. sloupci, kromě 2. řádku

- **S 1. prvním řádkem již nepracujeme.**
- **2. řádek odečteme od 3. řádku, abychom dostali ve 2. sloupci v 3. řádku nulu.**
- **3-násobek 2. řádku přičteme k 4. řádku a tím dostaneme ve 2. sloupci 4. řádku nulu.**

Dostali jsme nuly ve 2. sloupci.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 10 \\ 0 & 1 & 0 & 1 & 6 \\ 0 & 0 & -1 & -5 & -23 \\ 0 & 0 & -4 & 4 & 4 \end{bmatrix} \begin{array}{l} \\ \\ *(-1) \\ / 4 \end{array}$$

Vytváření nul v 3.sloupci

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 10 \\ 0 & 1 & 0 & 1 & 6 \\ 0 & 0 & 1 & 5 & 23 \\ 0 & 0 & -1 & 1 & 1 \end{bmatrix}$$

- 3. řádek přičítáme k 4. řádku. Abychom dostali ve 3. sloupci 4. řádku nulu.

Vytvořili jsme nuly i v 3. sloupci

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 10 \\ 0 & 1 & 0 & 1 & 6 \\ 0 & 0 & 1 & 5 & 23 \\ 0 & 0 & 0 & 6 & 24 \end{bmatrix} / 6$$

4. řádek vydělíme 6.

Výsledná trojúhelníková matice.
Pod hlavní diagonálou jsou **samé nuly**.

$$\begin{bmatrix} \mathbf{1} & 1 & 1 & 1 & 10 \\ \mathbf{0} & \mathbf{1} & 0 & 1 & 6 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & 5 & 23 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & 4 \end{bmatrix}$$

Zpětnou substitucí dopočítáme kořeny
soustavy rovnic.

Naše soustava rovnic se změnila na tvar:

$$x + y + z + t = 10$$

$$y + t = 6$$

$$z + 5t = 23$$

$$t = 4$$

zpětná substituce:

$$t=4$$

$$z+5t=23$$

$$z+20=23$$

$$z=3$$

$$y + t = 6$$

$$y + 4 = 6$$

$$y=2$$

$$x+y+z+t=10$$

$$x+2+3+4=10$$

$$x=1$$

Program na Gaussovu eliminaci - PASCAL

```
program GaussovaEliminace;
uses crt;
var a : array[1..6,1..6] of real;    b : array[1..6] of real;    x : array[1..6] of real;
    i, j, k, m, n, o, p, r : integer;    vymena : boolean;
procedure Vstup;
begin
    write('Vloz pocet radku soustavy linearnich rovnic : ');
    readln(n);
    for i := 1 to n do    for j := 1 to n+1 do
        begin
            gotoxy(10*j,i+9);
            if (j <= n) then begin write('a[' ,i,j,']: '); read(a[i,j]) end
                           else begin write('b[' ,i,']: '); readln(b[i]) end
        end;
    end;
end;
```

```

procedure Vymena_radku(c,d: integer);
var pom : real;      e, f : integer;
begin
  if (a[c,d] = 0) then
    begin
      e := c;
      repeat
        vymena := false;
        Inc(e);
        if (a[e,d] <> 0) then
          begin
            vymena := true;      for f := 1 to n do
                                  begin
                                    pom := a[c,f];
                                    a[c,f] := a[e,f];
                                    a[e,f] := pom;
                                  end;
            pom := b[c];      b[c] := b[e];      b[e] := pom;
          end;
        until (k = n);
      end;
    end;

```

```

procedure Gauss;
var pom : real;
begin
  for i := 1 to n-1 do    for j := i to n-1 do    Vymena_radku(j,i);
  for i := 1 to n-1 do
    begin
      if (a[i,i] <> 0) then
        begin
          k := i;
          repeat
            Inc(k); if (a[k,i] <> 0) then
              begin
                pom := a[i,i]/a[k,i];
                for o := i to n do          a[k,o] := a[k,o]*pom;
                b[k] := b[k]*pom;
              end;
            for o := i to n do          a[k,o] := a[k,o]-a[i,o];
                b[k] := b[k]-b[i];
          until (k = n);
        end;
    end;
end;

```

```
if a[n,n]=0 then writeln('Chyba: Matice je singularni.')
else begin
  x[n] := b[n]/a[n,n];  i := n - 1;
  repeat
    pom := 0;
    for j := i + 1 to n do
      pom := pom + a[i,j]*x[j];
    x[i] := (b[i]-pom)/a[i,i];
    i := i - 1;
  until (i = 0);
end;
end;
```

```
procedure Vystup;
begin
  for i := 1 to n do
    for j := 1 to n+1 do
      begin
        if (j <= n) then write(' ',a[i,j]:8:2)
        else writeln(' ',b[i]:8:2);
      end;
    writeln;
  end;
```

```
procedure Vystup2;  
begin  
  for i := 1 to n do    write('  x',i,' = ',x[i]:8:5);  writeln; writeln;  
end;  
  
begin  
  clrscr;  
  writeln('Program pro vypocet soustavy linearnich rovnic pomoci ');  
  writeln(' Gaussovy eliminacni metody. ');  writeln;  
  Vstup;  
  Gauss;  
  writeln;  writeln;  
  write(' Trojuhelnikova matice:');  
  writeln;  writeln;  
  Vystup;  
  write(' Reseni soustavy:');  
  Vystup2;  
  writeln;  write('Stiskni ENTER ...');  
  readln;  
end.
```

Gaussova eliminace

C++

```
// GEM
//#include "stdafx.h"
#include <stdio.h>
#include <iostream>
#include<math.h>
#include<iomanip>
using namespace std;

float GEM(float *a[],float *b[], int N, int M)
{
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < M; j++)
        {
            b[i][j] = a[i][j];
        }
    }
}
```

```

int max = 0;
    float pom;
    for (int i = 0; i<N; i++)          //nalezení pivota
    {
        for (int j = i + 1; j<N; j++)
        {
            if (fabs(b[j][i])>fabs(b[i][i]))
                max = j;
        }

        for (int k = i; k<M; k++)
        { pom = b[i][k]; b[i][k] = b[max][k]; b[max][k] = pom; }

        for (int j = i + 1; j<N; j++)
        {
            for (int k = N; k >= i; k--)
            {
                
$$b[j][k] = b[j][k] - b[i][k] * b[j][i] / b[i][i];$$

            }
        }
    }

```

```

float r[N];    int res = M-1;
    for(int i=0; i<N; i++)    r[i]=b[i][res];
    int pr = 0;
    if(pr==0)    {
        for(int j=N-1; j>=0; j--)
            {    pom=0;    for(int k=j+1; k<=N; k++)    pom=pom+b[j][k]*r[k];
                r[j]=(b[j][N]-pom)/b[j][j];
            }
    }

    cout<<"-----"<<endl;
    cout<<endl<<"  Reseni soustavy: "<<endl;
    for (int j = 0; j < M-1; j++)
        { cout<<endl<<"  x"<<j+1<<"= " <<setprecision(2)<< r[j];
          //printf("x%d = %5.3f \n",j+1, r[j]);
        }
        cout<<endl;

    cout<<"-----"<<endl;
        return **b;
}

```

```

void vypismat(float *a[],int N,int M)
{   cout<<endl;
    for (int i = 0; i<N; i++)
        {   //vypisuje prvky, kdyz dojde na konec radky, odradkuje
            for (int j = 0; j<M; j++)
                {   cout<<setw(7)<<setprecision(1) <<a[i][j];   //printf(" %5.3f ", a[i][j]);
                    if (j == M - 1) cout<<endl;   if (j == N-1)       cout<<" |";
                }
        }
}

void naplmatici(float *a[],int N,int M)
{   int res; cout<<endl;
    cout<<" Vyplnte matici: "<<endl; for (int x = 0; x < N; x++)
        {   for (int j = 0; j < M; j++)
            {   cout<<"   matice["<<x<<"]["<<j<<"]=" ";   cin>>a[x][j];
                                                    //scanf("%f", &a[x][j]);
            }
        }
}

```

```

int main(int argc)
{
    int N, M;
    cout<<" GAUSSOVA ELIMINACE "<<endl; cout<<" Zadejte pocet radku matice: ";
    cin>>N;          //scanf("%d", &N);
    cout<<" Zadejte pocet sloupcu matice: ";
    cin>>M;          //scanf("%d", &M);
    cout<<endl;
    //-----
    float**pole;
    pole = new float*[N];
    for (int i = 0; i < N; i++) { pole[i] = new float[M];
                                }

    naplmatici(pole,N,M);
    //-----
    float**pole1;
    pole1 = new float*[N];
    for (int i = 0; i < N; i++)
        { pole1[i] = new float[M];
          }
}

```

```

GEM(pole, pole1, N, M);
//-----
cout<<"  Matice soustavy rovnic: "<<endl;
vypismat(pole,N,M);
cout<<"-----"<<endl;
cout<<"  Trojuhelnikova matice: "<<endl;

vypismat(pole1,N,M);

//-----

for (int i = 0; i<N; i++) {
    delete[] pole[i];
    delete[] pole1[i];
}
delete[] pole;
delete[] pole1;

return 0;
}

```

Stabilita algoritmu Gaussova eliminace

Pokud při úpravách řádků násobíme velmi velkými čísly, pak rozdíl, přesného a vypočteného řešení na počítači, je nepříjemně velký.

V tom případě není algoritmus numericky stabilní.

Test

Vypočtete tyto soustavy rovnic:

$$x_1 + x_2 + x_3 - x_4 = 8$$

$$3x_1 - 2x_2 + 2x_3 - x_4 = 7$$

$$2x_1 - 5x_2 - 3x_3 + 2x_4 = -21$$

$$4x_1 + 3x_2 - 7x_3 - 2x_4 = -13$$

$$x_1 + x_2 + 2x_3 + 3x_4 = 7$$

$$2x_1 - 3x_2 + x_3 + x_4 = -1$$

$$3x_1 - 2x_2 + 3x_3 + 4x_4 = 6$$

$$5x_1 \qquad 7x_3 + 10x_4 = 20$$

$$x_1 + x_2 + x_3 - x_4 = 8$$

$$3x_1 - 2x_2 + 2x_3 - x_4 = 7$$

$$2x_1 - 5x_2 - 3x_3 + 2x_4 = -21$$

$$4x_1 + 3x_2 - 7x_3 - 2x_4 = -13$$

Nejprve převedeme na maticový tvar

$$\begin{bmatrix} 1 & 1 & 1 & -1 & 8 \\ 3 & -2 & 2 & -1 & 7 \\ 2 & -5 & -3 & 2 & -21 \\ 4 & 3 & -7 & -2 & -13 \end{bmatrix}$$

Pod hlavní diagonálou(červená čísla) musím dostat nuly (na místech zelených čísel).

Přičtu (-3)-násobek 1. řádku k 2. řádku;

Přičtu (-2)-násobek 1. řádku k 3. řádku;

Přičtu (-4)-násobek 1. řádku k 4. řádku;

$$\begin{bmatrix} 1 & 1 & 1 & -1 & 8 \\ 0 & -5 & -1 & 2 & -17 \\ 0 & -7 & -5 & 4 & -37 \\ 0 & -1 & -11 & 2 & -45 \end{bmatrix}$$

**Vyměním 2. a 4. řádek a
nový 2. řádek vynásobím (-1)**

$$\begin{bmatrix} 1 & 1 & 1 & -1 & 8 \\ 0 & 1 & 11 & -2 & 45 \\ 0 & -7 & -5 & 4 & -37 \\ 0 & -5 & -1 & 2 & -17 \end{bmatrix}$$

Přičtu 7-násobek 2. řádku k 3. řádku;
Přičtu 5-násobek 2. řádku k 4. řádku;

$$\begin{bmatrix}
 1 & 1 & 1 & -1 & 8 \\
 0 & 1 & 11 & -2 & 45 \\
 0 & 0 & 72 & -10 & 278 \\
 0 & 0 & 54 & -8 & 208
 \end{bmatrix}
 \begin{matrix}
 \\
 \\
 /2 \\
 /2
 \end{matrix}$$

Vydělím 3. a 4. řádek 2

$$\begin{bmatrix}
 1 & 1 & 1 & -1 & 8 \\
 0 & 1 & 11 & -2 & 45 \\
 0 & 0 & 36 & -5 & 139 \\
 0 & 0 & 27 & -4 & 104
 \end{bmatrix}
 \begin{array}{l}
 \\
 \\
 *(-27) \\
 * 36
 \end{array}$$

Přičtu (-27)-násobek 3.řádku ke 36-násobku 4. řádku

$$\begin{bmatrix} 1 & 1 & 1 & -1 & 8 \\ 0 & 1 & 11 & -2 & 45 \\ 0 & 0 & -972 & 135 & -3753 \\ 0 & 0 & 972 & -144 & 3744 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & -1 & 8 \\ 0 & 1 & 11 & -2 & 45 \\ 0 & 0 & 36 & -5 & 139 \\ 0 & 0 & 0 & -9 & 9 \end{bmatrix}$$

Ještě bych mohl vydělit 4. řádek (-9), ale to již není třeba.

$$-9x_4 = -9 \quad x_4 = -9/(-9)$$

$$\underline{\underline{x_4 = 1}}$$

$$36x_3 - 5x_4 = 139 \quad 36x_3 - 5 = 139 \quad 36x_3 = 144 \quad \underline{\underline{x_3 = 4}}$$

$$x_2 + 11x_3 - 2x_4 = 45$$

$$x_2 + 44 - 2 = 45 \quad x_2 = 45 - 44 + 2 = 3$$

$$\underline{\underline{x_2 = 3}}$$

$$x_1 + x_2 + x_3 - x_4 = 8 \quad x_1 + 3 + 4 - 1 = 8$$

$$x_1 = 8 - 3 - 4 + 1$$

$$\underline{\underline{x_1 = 2}}$$

$$x_1 + x_2 + 2x_3 + 3x_4 = 7$$

$$2x_1 - 3x_2 + x_3 + x_4 = -1$$

$$3x_1 - 2x_2 + 3x_3 + 4x_4 = 6$$

$$5x_1 \qquad 7x_3 + 10x_4 = 20$$

Převeďte na maticový tvar

$$\begin{bmatrix} 1 & 1 & 2 & 3 & 7 \\ 2 & -3 & 1 & 1 & -1 \\ 3 & -2 & 3 & 4 & 6 \\ 5 & 0 & 7 & 10 & 20 \end{bmatrix}$$

**Přičtu (-2)-násobek 1. řádku k 2. řádku;
Přičtu (-3)-násobek 1. řádku k 3. řádku;
Přičtu (-5)-násobek 1. řádku k 4. řádku;**

$$\begin{bmatrix} 1 & 1 & 2 & 3 & 7 \\ 0 & -5 & -3 & -5 & -15 \\ 0 & -5 & -3 & -5 & -15 \\ 0 & -5 & -3 & -5 & -15 \end{bmatrix}$$

3. a 4. řádek jsou lineárně závislé, proto je mohu odstranit (po úpravách by byly nulové).

$$\begin{bmatrix} 1 & 1 & 2 & 3 & 7 \\ 0 & 5 & 3 & 5 & 15 \end{bmatrix}$$

Zvolím parametry:

$$\underline{\underline{x_3 = s}} \quad \underline{\underline{x_4 = t}}$$

$$5x_2 + 3s + 5t = 15$$

$$\underline{\underline{x_2 = 3 - 3/5s - t}}$$

$$x_1 + 3 - 3/5s - t + 2s + 3t = 7$$

$$x_1 = 7 - 3 + (3-10)/5s + t - 3t$$

$$\underline{\underline{x_1 = 4 - 7/5s - 2t}}$$

Řešení: $\left[\underset{x_1}{4 - 7/5s - 2t}, \underset{x_2}{3 - 3/5s - t}, \underset{x_3}{s}, \underset{x_4}{t} \right]$

Použité zdroje

Mgr. PADRTA, David. Společnost Mgr. David Padrta: ALGORITMY A PROGRAMOVÁNÍ. [online]. 06. září 2001 [cit. 2013-08-31]. Dostupné z: <http://www.isd.cz/programy/gauss.html>

MÍKA, Stanislav. *Numerické metody algebry*. Vyd. 2. Praha: SNTL, 1985, 169 s. Matematika pro vysoké školy technické, sešit IV.

BLAŽEK, Jaroslav. *Algebra a teoretická aritmetika I.*, 1. vyd. Praha: Státní pedagogické nakladatelství, 1983. 278 s.