Počítačová podpora v elektrotechnice

Ing. Lenka Šroubová, Ph.D.
lsroubov@kte.zcu.cz

Ing. Petr Kropík, Ph.D.
pkropik@kte.zcu.cz

Katedra teoretické elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

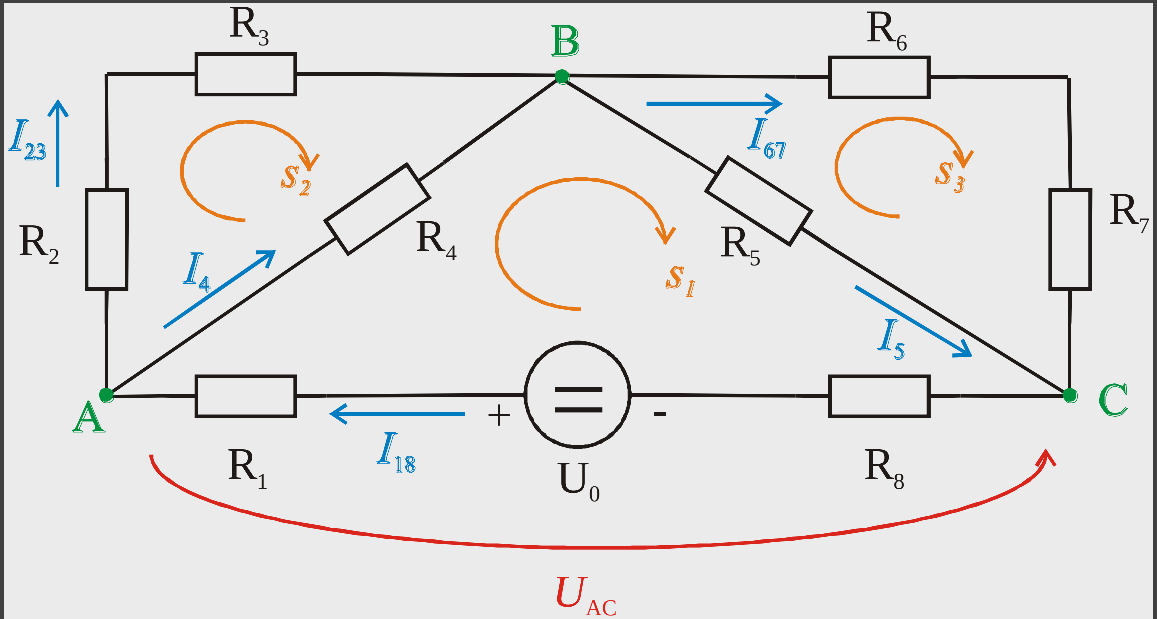
Místnost: EK602

FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY OF WEST BOHEMIA

Výpočet soustavy lineárních rovnic

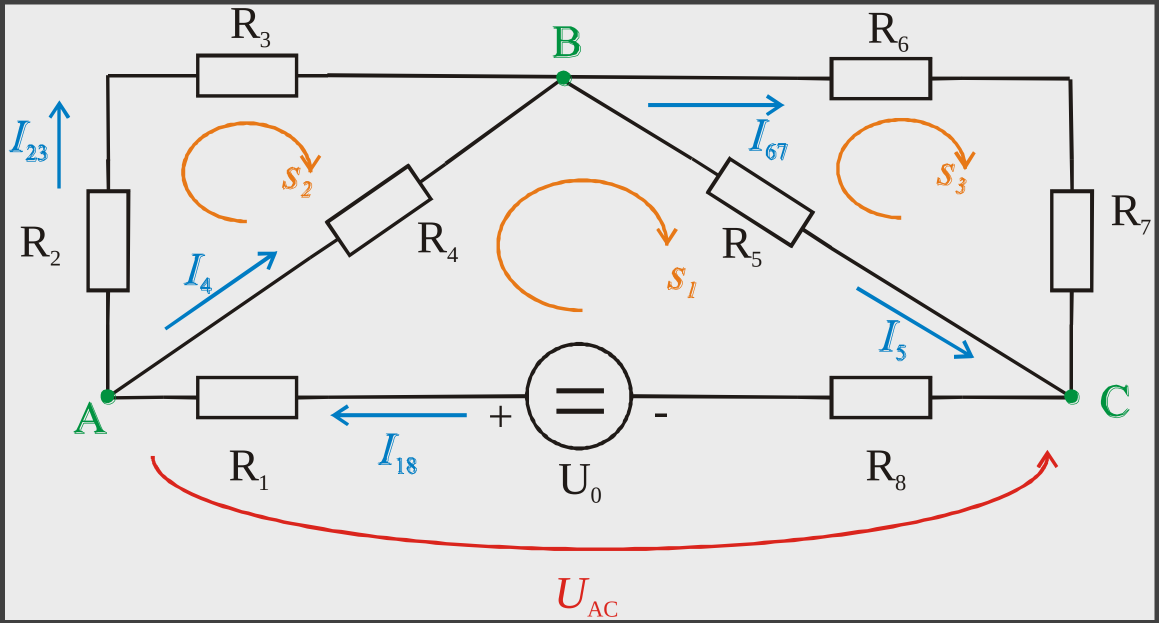
Příklad – elektrický obvod se stejnosměrným zdrojem napětí

Určete proudy I_{18}, I_{23} , I_4 , I_5 , I_{67} v obvodu na obr., je-li dáno:
R_1 = 1 Ω, R_2 = 2 Ω, R_3 = 3 Ω, R_4 = 5 Ω, R_5 = 3 Ω, R_6 = 2 Ω,
R_7 = 4 Ω, R_8 = 4,5 Ω, U_0 = 60 V.
Řešte pomocí přímé aplikace Kirchhoffových zákonů.



Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí



1. K. z. pro **uzel A**: $I_{18} - I_{23} - I_4 = 0$

1. K. z. pro **uzel C**: $I_5 + I_{67} - I_{18} = 0$

2. K. z. pro **smyčku s_1**: $R_1 I_{18} + R_4 I_4 + R_5 I_5 + R_8 I_{18} - U_0 = 0$

2. K. z. pro **smyčku s_2**: $R_2 I_{23} + R_3 I_{23} - R_4 I_4 = 0$

2. K. z. pro **smyčku s_3**: $R_6 I_{67} + R_7 I_{67} - R_5 I_5 = 0$

Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí

1. K. z. pro **uzel A**: $I_{18} - I_{23} - I_4 = 0$
1. K. z. pro **uzel C**: $I_5 + I_{67} - I_{18} = 0$
2. K. z. pro **smyčku s_1**: $R_1 I_{18} + R_4 I_4 + R_5 I_5 + R_8 I_{18} - U_0 = 0$
2. K. z. pro **smyčku s_2**: $R_2 I_{23} + R_3 I_{23} - R_4 I_4 = 0$
2. K. z. pro **smyčku s_3**: $R_6 I_{67} + R_7 I_{67} - R_5 I_5 = 0$

Rovnice upravíme, seřadíme proudy, na levé straně ponecháme členy s neznámými, ostatní členy převedeme na pravou stranu.

1. K. z. pro **uzel A**: $1 * I_{18} - 1 * I_{23} - 1 * I_4 + 0 * I_5 + 0 * I_{67} = 0$
1. K. z. pro **uzel C**: $-1 * I_{18} + 0 * I_{23} - 0 * I_4 + 1 * I_5 + 1 * I_{67} = 0$
2. K. z. pro **smyčku s_1**: $(R_1 + R_8) * I_{18} + 0 * I_{23} + R_4 * I_4 + R_5 * I_5 + 0 * I_{67} = U_0$
2. K. z. pro **smyčku s_2**: $0 * I_{18} + (R_2 + R_3) * I_{23} - R_4 I_4 + 0 * I_5 + 0 * I_{67} = 0$
2. K. z. pro **smyčku s_3**: $0 * I_{18} + 0 * I_{23} - 0 * I_4 - R_5 I_5 + (R_6 + R_7) * I_{67} = 0$

Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí

Soustava rovnic:

$$
\begin{aligned}
&1*I_{18} - 1*I_{23} - 1*I_4 + 0*I_5 + 0*I_{67} = 0\\
&-1*I_{18} + 0*I_{23} - 0*I_4 + 1*I_5 + 1*I_{67} = 0\\
&(R_1 + R_8)*I_{18} + 0*I_{23} + R_4*I_4 + R_5*I_5 + 0*I_{67} = U_0\\
&0*I_{18} + (R_2 + R_3)*I_{23} - R_4 I_4 + 0*I_5 + 0*I_{67} = 0\\
&0*I_{18} + 0*I_{23} - 0*I_4 - R_5 I_5 + (R_6 + R_7)*I_{67} = 0
\end{aligned}
$$

Řešíme soustavu 5 rovnic o 5 neznámých:

$$
\begin{bmatrix}
1 & -1 & -1 & 0 & 0\\
-1 & 0 & 0 & 1 & 1\\
R_1 + R_8 & 0 & R_4 & R_5 & 0\\
0 & R_2 + R_3 & -R_4 & 0 & 0\\
0 & 0 & 0 & -R_5 & R_6 + R_7
\end{bmatrix}
\cdot
\begin{bmatrix}
I_{18}\\ I_{23}\\ I_4\\ I_5\\ I_{67}
\end{bmatrix}
=
\begin{bmatrix}
0\\ 0\\ U_0\\ 0\\ 0
\end{bmatrix}
$$

Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí

$$\begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \\ R_1 + R_8 & 0 & R_4 & R_5 & 0 \\ 0 & R_2 + R_3 & -R_4 & 0 & 0 \\ 0 & 0 & 0 & -R_5 & R_6 + R_7 \end{bmatrix} \cdot \begin{bmatrix} I_{18} \\ I_{23} \\ I_4 \\ I_5 \\ I_{67} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ U_0 \\ 0 \\ 0 \end{bmatrix}$$

```
A=[        1,      -1,  -1,   0,      0;...
          -1,       0,   0,   1,      1;...
     R1+R8,         0,  R4,  R5,      0;...
           0,R2+R3,-R4,   0,      0;...
           0,       0,   0,-R5,R6+R7];

b=[0;0;U0;0;0];
```

Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí

```
R1=1; R2=2; R3=3; R4=5; R5=3; R6=2; R7=4; R8=4.5;
U0=60; % jednotlivé hodnoty odporů a napětí

A = [1,-1,-1,0,0;-1,0,0,1,1,R1+R8,0,R4,R5,0;...
     0,R2+R3,-R4,0,0;0,0,0,-R5,R6+R7]; % matice A
b = [0;0;U0;0;0];                       % sloupcový vektor b
x = A\b           % maticová operace - dělení zleva
x =
    6.0000
    3.0000
    3.0000
    4.0000
    2.0000
```

Řešení soustavy rovnic je:

$$I_{18} = \underline{\underline{6\ \mathrm{A}}}$$
$$I_{23} = \underline{\underline{3\ \mathrm{A}}}$$
$$I_{4} = \underline{\underline{3\ \mathrm{A}}}$$
$$I_{5} = \underline{\underline{4\ \mathrm{A}}}$$
$$I_{67} = \underline{\underline{2\ \mathrm{A}}}$$

Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí

Jiná možnost řešení soustavy rovnic:

```
R=[1,2,3,5,3,2,4,4.5]; % hodnoty odporů - vektor
U0=60;

A=[1,-1,-1,0,0;-1,0,0,1,1;R(1)+R(8),0,R(4),R(5),0;…
    0,R(2)+R(3),-R(4),0,0;0,0,0,-R(5),R(6)+R(7)];
b = [0,0,U0,0,0];    % b zadán jako řádkový vektor
x = A\b.'            % transpozice vektoru b
x =
     6.0000
     3.0000
     3.0000
     4.0000
     2.0000
```

Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí

Jiná možnost řešení soustavy rovnic:

```
R=[1,2,3,5,3,2,4,4.5]; % hodnoty odporů - vektor
U0=60;
```

V matici A prvky vektoru R

```
A=[1,-1,-1,0,0;-1,0,0,1,1;R(1)+R(8),0,R(4),R(5),0;...
    0,R(2)+R(3),-R(4),0,0;0,0,0,-R(5),R(6)+R(7)];
b = [0,0,U0,0,0];     % b zadán jako řádkový vektor
x = A\b.'             % transpozice vektoru b
x =
     6.0000
     3.0000
     3.0000
     4.0000
     2.0000
```

Výpočet soustavy lineárních rovnic

Pokračování příkladu – elektrický obvod se stejnosměrným zdrojem napětí

Jiná možnost řešení soustavy rovnic:

```
R=[1,2,3,5,3,2,4,4.5]; % hodnoty odporů - vektor
U0=60;
```

V matici A prvky vektoru R

```
A=[1,-1,-1,0,0;-1,0,0,1,1;R(1)+R(8),0,R(4),R(5),0;...
    0,R(2)+R(3),-R(4),0,0;0,0,0,-R(5),R(6)+R(7)];
b = [0,0,U0,0,0];     % b zadán jako řádkový vektor
x = A\b.'             % transpozice vektoru b
x =
     6.0000
     3.0000
     3.0000
     4.0000
     2.0000
```

Vektor b musí být sloupcový, aby mohlo proběhnout dělení zleva

Operátory

Relační operátory

`==` porovnání na rovnost (je rovno)

`~=` porovnání na nerovnost (není rovno)

`<`, `>`, je menší, je větší

`<=`, `>=` je menší nebo rovno, je větší nebo rovno

`~` negace (**`not`**)

Logické operátory

`&` a zároveň (**`and`**)

`|` nebo (**`or`**)

`~` negace (**`not`**)

Pozor!

`=` přiřazení

`==` porovnání na rovnost

Operátory

Příklady:

`&` – a zároveň (logický operátor)

(`and` – funkce provádějící stejnou operaci)

`(3<5)&(4<6)` – pravda a zároveň pravda

```
ans = 1
```

`(3<5)&(4>=6)` – pravda a zároveň nepravda

```
ans = 0
```

`(3>5)&(4>6)` – nepravda a zároveň nepravda

```
ans = 0
```

Lze psát i takto:

`and((3>5),(4<6))` – nepravda a zároveň pravda

`ans = 0` – funkce `and` s dvěma argumenty

Operátory

Příklady:

`|` – nebo (logický operátor)

(`or` – funkce provádějící stejnou operaci)

```
(3<5)|(4~=6)
ans = 1
```
– pravda nebo pravda

```
(3<5)|(4==6)
ans = 1
```
– pravda nebo nepravda

```
(3>5)|(4==6)
ans = 0
```
– nepravda nebo nepravda

Lze psát i takto:

```
or((3>5),(4~=6))
ans = 1
```
– nepravda nebo pravda

– funkce `or` s dvěma argumenty

Operátory

Příklady:

xor – funkce „exkluzivní“ nebo

```
(xor((3<5),(4<6))
ans = 0
```

– pravda nebo pravda exkluzivně

– pozor 0 – výsledkem je nepravda

```
xor((3>5),(4<6))
ans = 1
```

– nepravda nebo pravda exkluzivně

```
xor((3>5),(4>6))
ans = 0
```

– nepravda nebo nepravda exkluzivně

```
whos
 Name      Size            Bytes  Class     Attributes
 ans       1x1                 1  logical
```

Operátory

Příklady:

`~` – negace

(`not` – funkce provádějící stejnou operaci)

```
~0
ans =1
```
– negace 0 (nepravdy) je 1 (pravda)

```
~1
ans =0
```
– negace 1 (pravdy) je 0 (nepravda)

```
~5
ans =0
```
– negace 5 (pravdy) je 0 (nepravda)

```
~(3<5)
ans =0
```
– negace pravdy je nepravda

```
not(3<5)
ans =0
```
– lze psát i takto, nepravda

Tedy pak:

```
(~(3<5))&(4<6)
ans = 0
```
– nepravda a zároveň pravda

Speciální typy matic

eye – **jednotková** matice je čtvercová matice **s jednotkami na hlavní diagonále** (jinde jsou nuly) nebo obdélníková matice s **jednotkovou submaticí**, např:

```
eye(2)
ans =
     1     0
     0     1
```

```
eye(2,3)
ans =
     1     0     0
     0     1     0
```

Násobení jednotkovou maticí

```
A=[1,2;3,4]
A =
     1     2
     3     4
```

```
A*eye(2)
ans =
     1     2
     3     4
```

```
eye(2)*A
ans =
     1     2
     3     4
```

```
B = [1:4;2:2:8]
B =
     1     2     3     4
     2     4     6     8
```

```
B*eye(4)
ans =
     1     2     3     4
     2     4     6     8
```

```
eye(2)*B
ans =
     1     2     3     4
     2     4     6     8
```

Speciální typy matic

`ones(n)` – matice naplněná jedničkami o rozměru n x n

`ones(m,n)` – matice naplněná jedničkami o rozměru m x n

```
K = ones(3)
K =
     1     1     1
     1     1     1
     1     1     1
```

```
L = ones(2,3)
L =
     1     1     1
     1     1     1
```

```
ones(4,2)
ans =
     1     1
     1     1
     1     1
     1     1
```

Speciální typy matic

`ones(n)` – matice naplněná jedničkami o rozměru n x n

`ones(m,n)` – matice naplněná jedničkami o rozměru m x n

```
K = ones(3)
K =
      1      1      1
      1      1      1
      1      1      1
```

```
L = ones(2,3)
L =
      1      1      1
      1      1      1
```

```
ones(4,2)
ans =
      1      1
      1      1
      1      1
      1      1
```

Pozor: **matice plná jedniček**

```
ones(2)
ans =
      1    1
      1    1
```

matice jednotková

```
eye(2)
ans =
      1    0
      0    1
```

Speciální typy matic

matice naplněná čísly 5 o rozměru 3 x 3

```
F = 5.*ones(3)
F =
     5     5     5
     5     5     5
     5     5     5
```

matice naplněná čísly 7 o rozměru 4 x 2

```
7.*ones(4,2)
ans =
     7     7
     7     7
     7     7
     7     7
```

matice naplněná čísly -4 o rozměru 2 x 3

```
G = -4.*ones(2,3)
G =
    -4    -4    -4
    -4    -4    -4
```

Speciální typy matic

`zeros(`*n*`)` – matice naplněná nulami o rozměru n x n

`zeros(`*m*`,`*n*`)` – matice naplněná nulami o rozměru m x n

```
N = zeros(3)
N =
     0     0     0
     0     0     0
     0     0     0
```

```
Q = zeros(2,3)
Q =
     0     0     0
     0     0     0
```

```
zeros(4,2)
ans =
     0     0
     0     0
     0     0
     0     0
```

Speciální typy matic

`magic(n)` – "magická" matice - magický čtverec o rozměru $n \times n$, součet prvků na diagonále je stejný jako součet prvků v jednotlivých řádcích a sloupcích matice

```
M = magic(3)
M =
     8     1     6
     3     5     7
     4     9     2
```

```
diag(M)
ans =
     8
     5
     2
```

```
sum(diag(M))
ans =
    15
```

```
M.'
ans =
     8     3     4
     1     5     9
     6     7     2
```

```
sum(M)
ans =
    15    15    15
```

```
sum(M.')
ans =
    15    15    15
```

Speciální typy matic

`pascal(n)` – Pascalův trojúhelník o rozměru n x n

```
pascal(7)
ans =
      1      1      1      1      1      1      1
      1      2      3      4      5      6      7
      1      3      6     10     15     21     28
      1      4     10     20     35     56     84
      1      5     15     35     70    126    210
      1      6     21     56    126    252    462
      1      7     28     84    210    462    924
```

První řádek a první sloupec jsou tvořeny pouze číslem jedna.
Druhý řádek a druhý sloupec jsou tvořeny seřazenými přirozenými čísly.
Další řádky a sloupce jsou tvořeny součtem čísel vlevo a nahoře.

Speciální typy matic

`hilb(n)` – čtvercová matice o rozměru $n \times n$, pro jejíž prvky platí:

$$H_{kl} = 1 / (k + l - 1)$$

kde k a l jsou indexy příslušného řádku a sloupce

```
H = hilb(5)
H =
        1       1/2     1/3     1/4     1/5
        1/2     1/3     1/4     1/5     1/6
        1/3     1/4     1/5     1/6     1/7
        1/4     1/5     1/6     1/7     1/8
        1/5     1/6     1/7     1/8     1/9
```

Např. $H_{34} = 1 / (3 + 4 - 1) = 1/6$ – prvek v **3. řádku** a **4. sloupci**

Speciální typy matic

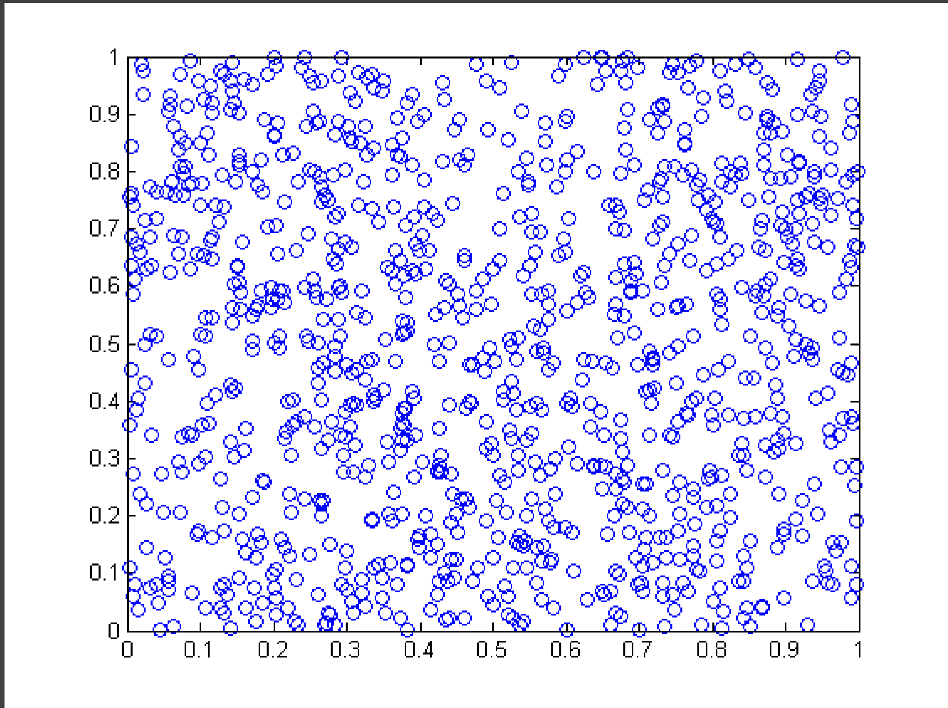
rand – pseudo-náhodné číslo v rozmezí 0 až 1 (desetinné)

rand(*n*) – matice o rozměru *n* x *n* obsahující pseudo-náhodná čísla v rozmezí 0 až 1

rand(*m*,*n*) – matice s *m* řádky a *n* sloupci obsahující pseudo-náhodná čísla v rozmezí 0 až 1

Např.

```
x = rand(1,1000);
y = rand(1,1000);
plot(x,y,'o')
```



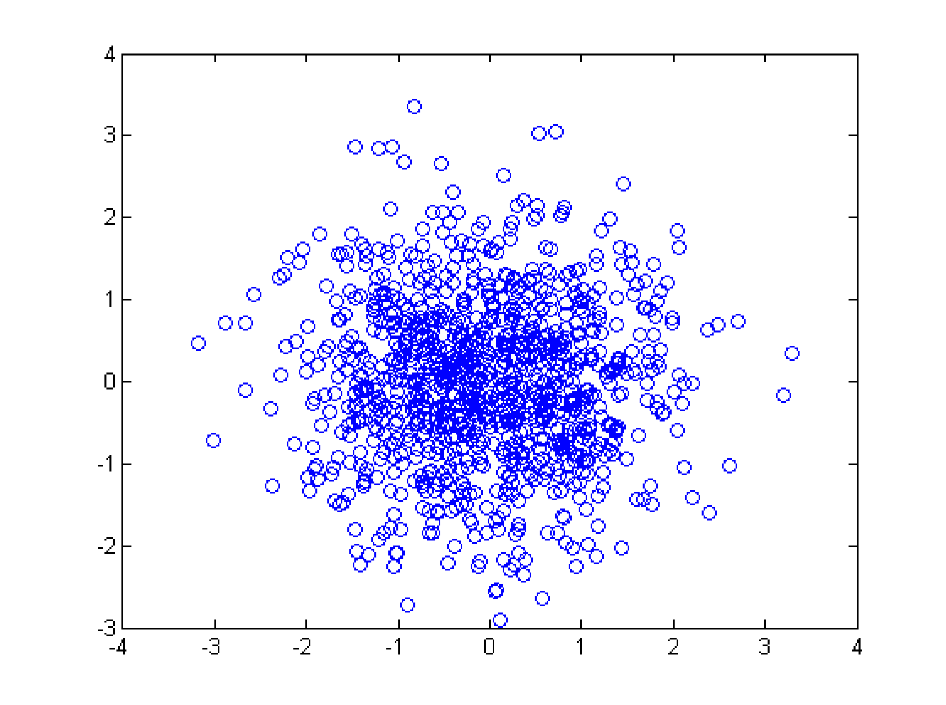
Speciální typy matic

randn(*n*) – matice o rozměru *n* x *n* obsahující pseudo-náhodné hodnoty, které jsou získány z normálního rozdělení s průměrem nula a směrodatnou odchylkou jedna.

randn(*m*,*n*) – matice s *m* řádky a *n* sloupci obsahující pseudo-náhodné hodnoty, které jsou získány z normálního rozdělení s průměrem nula a směrodatnou odchylkou jedna.

Např.

```
xn = randn(1,1000);
yn = randn(1,1000);
plot(xn,yn,'o')
```



Speciální typy matic

Jednoduchá funkce na generování pseudonáhodných čísel:

```
function vysledek = nahoda(r, s, odKol, doKol)
% r - počet řádků
% s - počet sloupců
% odKol - dolní mez (od jakého čísla generujeme)
% doKol - horní mez (do jakého čísla generujeme)
% round - zaokrouhlení na nejbližší celé číslo
vysledek = round(rand(r,s) .* (doKol-odKol)) + odKol;
end
```

Volání funkce:

```
M = nahoda(3, 5, -4, 4)
M =
    -3     1    -1    -4     2
     4    -4     3    -3     1
     4    -2    -4     1     0
```

Práce s maticemi a vektory

Složení matice ze dvou řádkových vektorů:

```
x = [1:5]
x =
     1   2   3   4   5
y = [9,3,4,3,2]
y =
     9   3   4   3   2
```

```
A = [x;y]
```

; středník
oddělovač řádků matice při zadávání

```
B = [x,y]
```

, – čárka
oddělovač položek v řádku matice

Práce s maticemi a vektory

Složení matice ze dvou řádkových vektorů:

```
x = [1:5]
x =
      1   2   3   4   5
y = [9,3,4,3,2]
y =
      9   3   4   3   2
```

```
A = [x;y]
A =
      1   2   3   4   5
      9   3   4   3   2
```

; – středník
oddělovač řádků matice při zadávání

```
B = [x,y]
B =
      1   2   3   4   5   9   3   4   3   2
```

, – čárka
oddělovač položek v řádku matice

Práce s maticemi a vektory

Složení matice ze dvou sloupcových vektorů:

```
x = [1:5];
y = [9,3,4,3,2];
```

```
x.'          y.'
ans =        ans =
     1            9
     2            3
     3            4
     4            3
     5            2
```

```
C = [x.',y.']
C =
     1     9
     2     3
     3     4
     4     3
     5     2
```

,
čárka

```
D = [x.';y.']
D =
     1
     2
     3
     4
     5
     9
     3
     4
     3
     2
```

;
středník

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

- pomocí tzv. indexů (indexuje se od 1)

```
a=[9:-1:4]
a =
     9     8     7     6     5     4
```

`a(3)` – přístup k 3. prvku

```
ans =
       7
```

```
b=[10;8]
b =
    10
     8
```

`b(2)` – přístup k 2. prvku

```
ans =
       8
```

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

- pomocí tzv. indexů (indexuje se od 1)

```
a=[9:-1:4]
a =
     9     8     7     6     5     4
a(3)
```

– přístup k 3. prvku

```
ans =
     7
```

```
a(1,3)
ans =
     7
```

– přístup k prvku v 1. řádku, 3. sloupci

```
b=[10;8]
b =
    10
     8
b(2)
```

– přístup k 2. prvku

```
ans =
     8
```

```
b(2,1)
ans =
     8
```

– přístup k prvku ve 2. řádku, 1. sloupci

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

- pomocí tzv. indexů (indexuje se od 1)

```
a=[9:-1:4]
a =
    9     8     7     6     5     4
```

`a(3)` – přístup k 3. prvku

```
ans =
     7
```

```
a(1,3)
ans =
     7
```

– přístup k prvku v 1. řádku, 3. sloupci

– pokud se jedná o řádkový, resp. sloupcový **vektor**, nikoli o matici, lze **vynechat index 1** na pozici řádku, resp. sloupce

```
b=[10;8]
b =
    10
     8
```

`b(2)` – přístup k 2. prvku

```
ans =
     8
```

```
b(2,1)
ans =
     8
```

– přístup k prvku ve 2. řádku, 1. sloupci

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
a=[9:-1:4]
a =
    9    8    7    6    5    4
```

```
a(end)
ans =
     4
```

```
b=[10;8]
b =
    10
     8
```

– přístup k **posledními** prvku

```
b(end)
ans =
     8
```

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
a=[9:-1:4]
a =
     9     8     7     6     5     4
```

```
b=[10;8]
b =
    10
     8
```

– přístup k **posledními** prvku

```
a(end)
ans =
     4
```

```
b(end)
ans =
     8
```

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
     1     2     3     4     5
     9     3     4     3     2
     1     3     5     7     9
     9     8     7     6     5

D(3,4)
ans =
     7
```

- **matice**(**index řádku**, **index sloupce**)

- přístup k prvku ve 3. řádku, 4. sloupci

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
     1     2     3     4     5
     9     3     4     3     2
     1     3     5     7     9
     9     8     7     6     5

D(3:4,1:3)
ans =
          1     3     5
          9     8     7
```

– přístup k prvkům ve **3**. až **4**. řádku a **1**. až **3**. sloupci

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =

     1     2     3     4     5
     9     3     4     3     2
     1     3     5     7     9
     9     8     7     6     5

D(3:4,1:3)

ans =

     1     3     5
     9     8     7
```

: - výčet, rozsah

```
3:4     % vektor
ans =
     3   4
1:3     % vektor
ans =
     1   2   3
```

– přístup k prvkům ve **3**. až **4**. řádku a **1**. až **3**. sloupci

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
    1     2     3     4     5
    9     3     4     3     2
    1     3     5     7     9
    9     8     7     6     5

D(3:4,:)
ans =
         1     3     5     7     9
         9     8     7     6     5
```

: - výčet, rozsah

– přístup k prvkům ve **3. až 4. řádku** a **všech sloupcích**

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
     1     2     3     4     5
     9     3     4     3     2
     1     3     5     7     9
     9     8     7     6     5

D([1,3,4],3:end)
ans =
     3     4     5
     5     7     9
     7     6     5
```

– přístup k prvkům ve **1.,3. a 4. řádku** a **3. až posledním sloupci**

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
     1     2     3     4     5
     9     3     4     3     2
     1     3     5     7     9
     9     8     7     6     5

D([1,3,4],3:end)
ans =
     3     4     5
     5     7     9
     7     6     5
```

– přístup k prvkům ve 1.,3. a 4. řádku a 3. až posledním sloupci

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
    1     2     3     4     5
    9     3     4     3     2
    1     3     5     7     9
    9     8     7     6     5

D([1,3,4],3:end)
ans =
    3     4     5
    5     7     9
    7     6     5
```

```
[1,3,4]
ans = % vektor
     1   3   4
```

prvky odděleny čárkami , vektor ohraničen [ ]

– přístup k prvkům ve **1.,3. a 4. řádku** a **3. až posledním sloupci**

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
     1     2     3     4     5
     9     3     4     3     2
     1     3     5     7     9
     9     8     7     6     5

D(:,:)
```

Práce s maticemi a vektory

Přístup k jednotlivým prvkům matic a vektorů:

```
D=[1:5;9,3,4,3,2;1:2:9;9:-1:5]

D =
     1     2     3     4     5
     9     3     4     3     2
     1     3     5     7     9
     9     8     7     6     5

D(:,:)
```

Samotná : (rozsah, výčet) má význam (u indexů matic), že chci použít všechny možné (dostupné) hodnoty.

– vytiskne celou matici **D**, **všechny** řádky i sloupce