



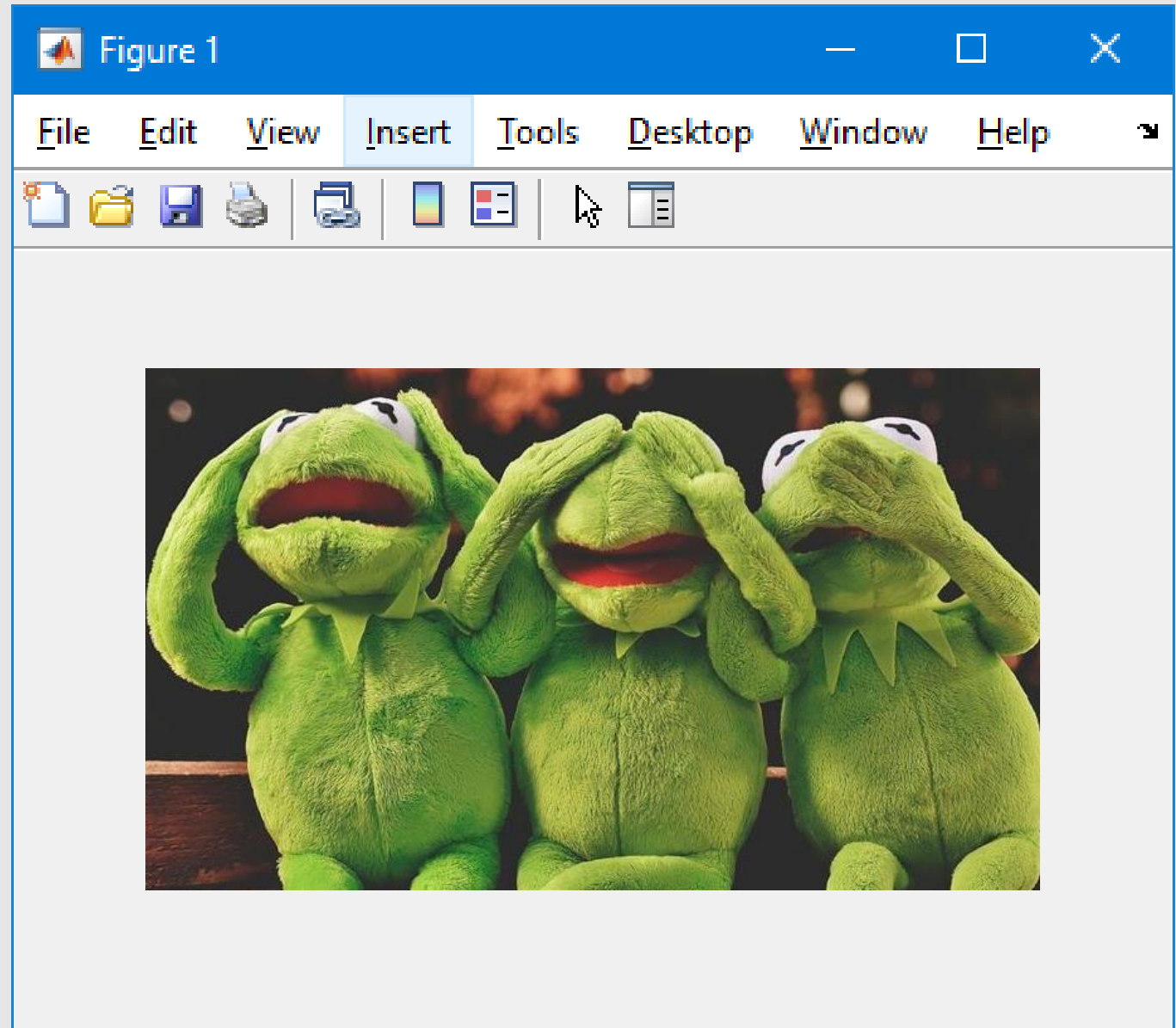
Úvod do Matlabu

Prednáška č. 4

- **Vizualizácia dát – okno Figure**
- Grafy spojitých priebehov – plot, area, semilogx
- Grafy diskretných priebehov – stem, stairs, bar, histogram
- Trojrozmerné grafy – plot3, mesh, surf
- Niektoré užitočné príkazy

Vizualizácia dát – okno Figure

- V Matlabe sa všetky obrázky, grafy, 3D grafy a podobne zobrazujú do špeciálneho okna **Figure**.
- Vyvolať ho môžeme príkazom **figure**. V tomto prípade sa vytvorí nové okno Figure 1. Pokiaľ zavoláme funkciu **figure** znova vytvorí sa ďalšie okno, Figure 2. Nevýhodou tohto spôsobu je neprehľadnosť pri väčšom množstve zobrazení.
- Ďalším spôsobom je volanie funkcie **figure** s definovaním jeho čísla. Takto vieme zvoliť okno, do ktorého sa dáta zobrazia. Výhodou je to, že dáta môžeme zobraziť zakaždým do toho istého okna. V takom prípade funkciu voláme nasledovne **figure(1)**.



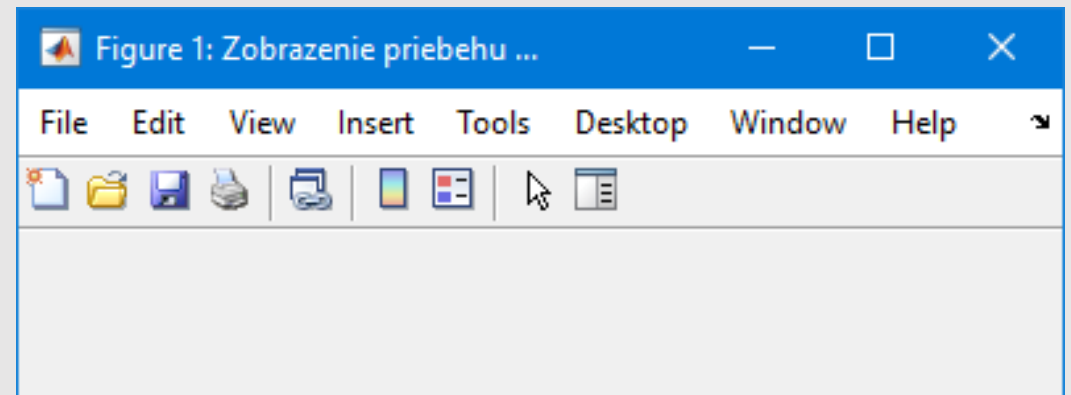
Vizualizácia dát – okno Figure

- V prípade, že chceme nastaviť aj iné parametre okna figure, môžeme to urobiť **priamo pri zadávaní príkazu** alebo tiež neskôr pomocou príkazu **set(h,'Name','Novy nazov')**.

```
h = figure('Name','Zobrazenie priebehu ... ');  
  
>> whos h  
Name      Size      Bytes   Class      Attributes  
h         1x1       112    matlab.ui.Figure  
  
>> disp(h)  
Figure (1: Zobrazenie priebehu ...) with properties:  
  
    Number: 1  
    Name: 'Zobrazenie priebehu ...'  
    Color: [0.9400 0.9400 0.9400]  
    Position: [-847 651 560 420]  
    Units: 'pixels'
```

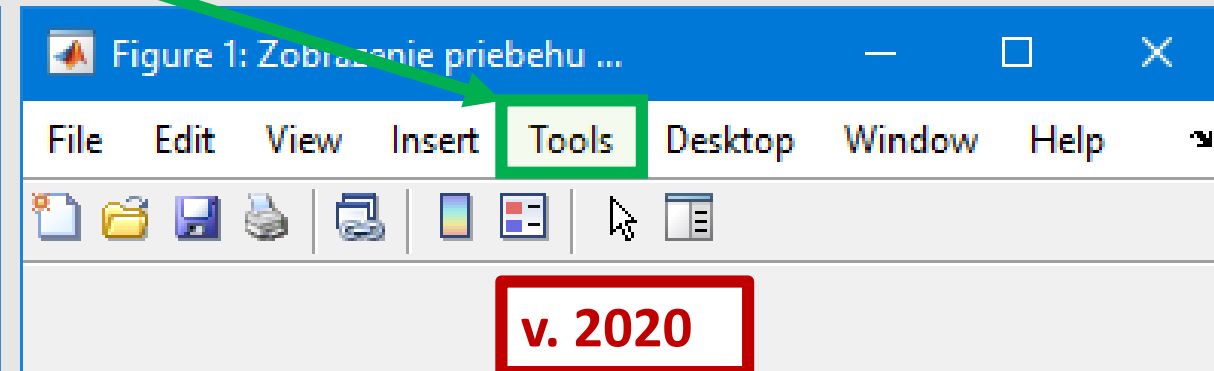
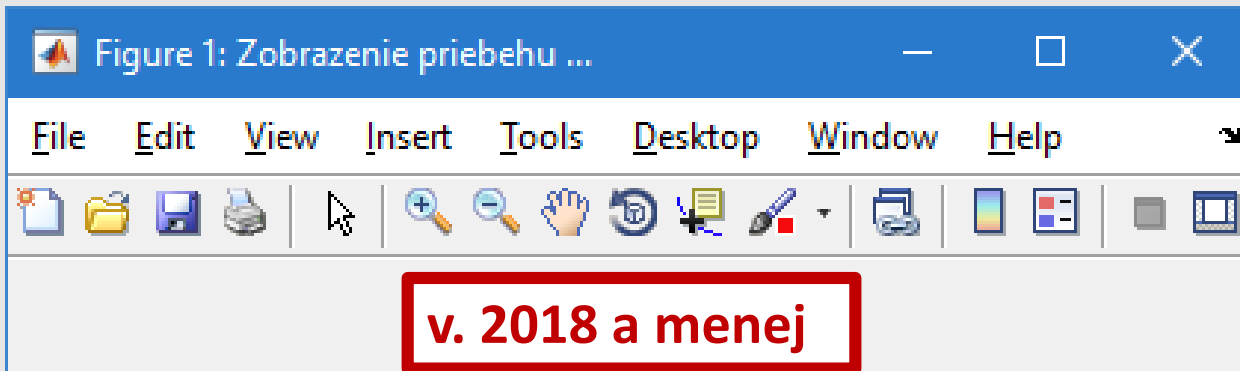
Môžeme si všimnúť, že sme zámerne výstup funkcie priradili premennej **h**. V podstate táto nie je premennou, ale objektom, ktorého vlastnosti vieme meniť cez príkaz **set()**

- Okno **Figure** je možné zavrieť, pomocou krížika v pravom hornom rohu okna alebo pokiaľ ho chceme zavrieť počas behu programu, môžeme použiť funkciu **delete**, ktorú zavoláme so vstupným parametrom objektu **delete(h)**.



Vizualizácia dát – okno Figure - Nástroje

- Po príchode nových verzií Matlabu sa zmenila horná lišta nástrojov! Nástroje sú stále prístupné ale požadovaný nástroj je potrebné zvoliť cez možnosť **Tools**!



- Nový (New)** – Vytvorí nové okno figure. Ekvivalent k príkazu **figure**
- Otvoriť (Open)** – Otvoriť obrázok Matlabu uložený na pevnom disku (iba .fig súbory)
- Uložiť (Save)** – Uložiť obsah okna v ľubovoľnom formáte. Ekvivalent príkazu **savefig()**
- Tlačiť (Print)** – Vytlačiť obsah okna na tlačiarni
- Editovať (Edit)** – Možnosť upravovať aktuálne zobrazenie
- Priblížiť oddialiť (Zoom)** – Úprava priblíženia
- Uchopiť (Pan)** – Pohybovať zobrazením
- Rotovať (Rotate)** – Rotovať zobrazenie (zmysel má iba pre 3D)
- Dátový kurzor (Data cursor)** – Možno myšou zisťovať hodnoty bodov zobrazenia
- Výber dát/zvýraznenie (Brush/Select Data)** – Editácia zobrazenia
- Vložiť legendu (Insert legend)** – Vložiť legendu. Ekvivalent príkazu **legend()**



Úvod do Matlabu

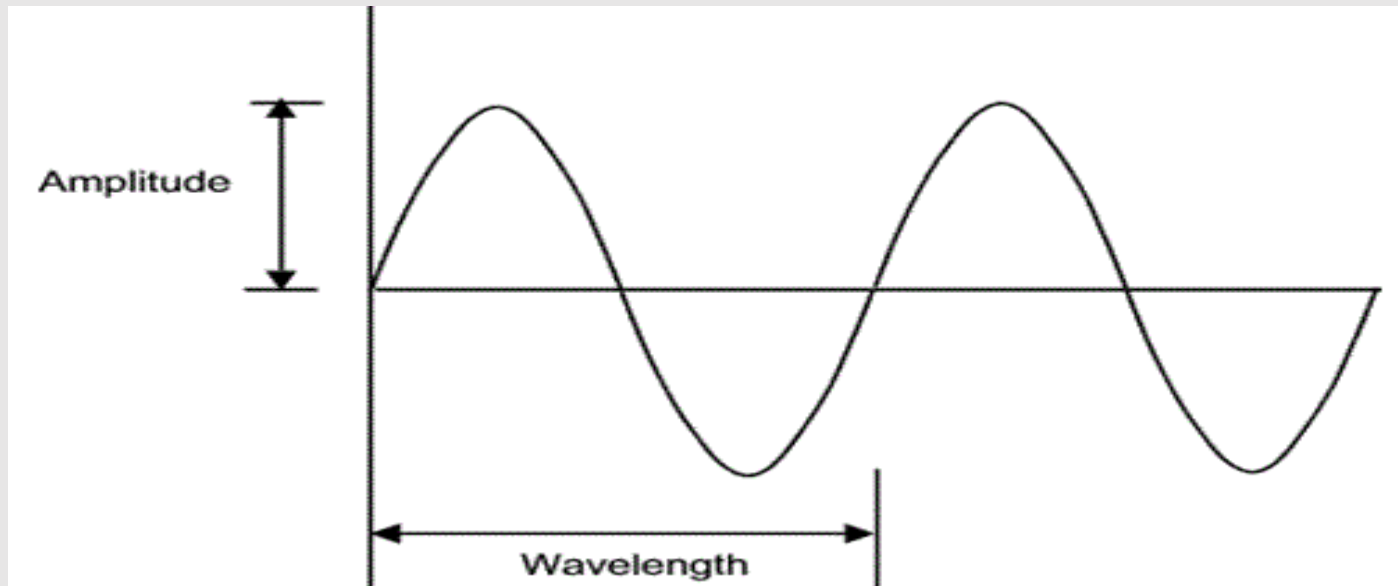
Prednáška č. 4

- Vizualizácia dát – okno Figure
- **Grafy spojitých priebehov – plot, area, semilogx**
- Grafy diskretných priebehov – stem, stairs, bar, histogram
- Trojrozmerné grafy – plot3, mesh, surf
- Niektoré užitočné príkazy

Grafy spojitéch priebehov

- Tieto grafy patria medzi najzakladanejšie a v oblasti elektroniky a spracovania signálov medzi najpoužívanejšie.
- Ide o takzvané „líniové grafy“ (line plots) alebo grafy spojitéch priebehov.
- Povedzme, že chceme zobrazíť graf funkcie $y=\sin(x)$ v intervale $\langle 0; 2\pi \rangle$. Údaje pre taký priebeh vygenerujem napríklad takto:

```
x = 0:2*pi
x =
    0    1    2    3    4    5    6
>> y = sin(x)
y =
    0    0.8415    0.9093    0.1411   -0.7568   -0.9589   -0.2794
```

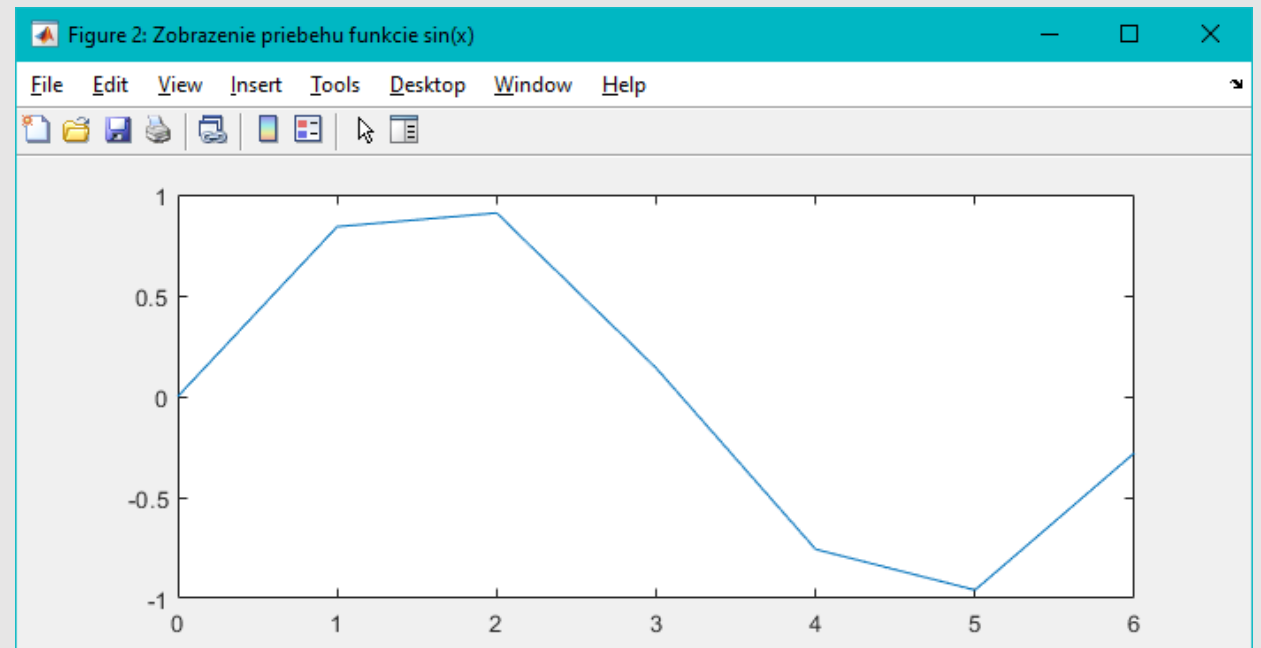


Grafy spojitéch priebehov – PLOT

- Tieto grafy patria medzi najzakladanejšie a v oblasti elektroniky a spracovania signálov medzi najpoužívanejšie.
- Ide o takzvané „líniové grafy“ (line plots) alebo grafy spojitéch priebehov.
- **Povedzme, že chceme zobrazíť graf funkcie $y=\sin(x)$ v intervale $\langle 0; 2\pi \rangle$. Údaje pre taký priebeh vygenerujem napríklad takto:**
- **Samotné vykreslenie pomocou funkcie plot môže potom vyzeráť takto:**
- **Vidíme, že vo funkcii plot zadávame premenné v poradí x a potom y!**
- **Môžeme si všimnúť, že priebeh funkcie $y=\sin(x)$ sa dá rozoznať, ale je akosi „kostrbatý“.**

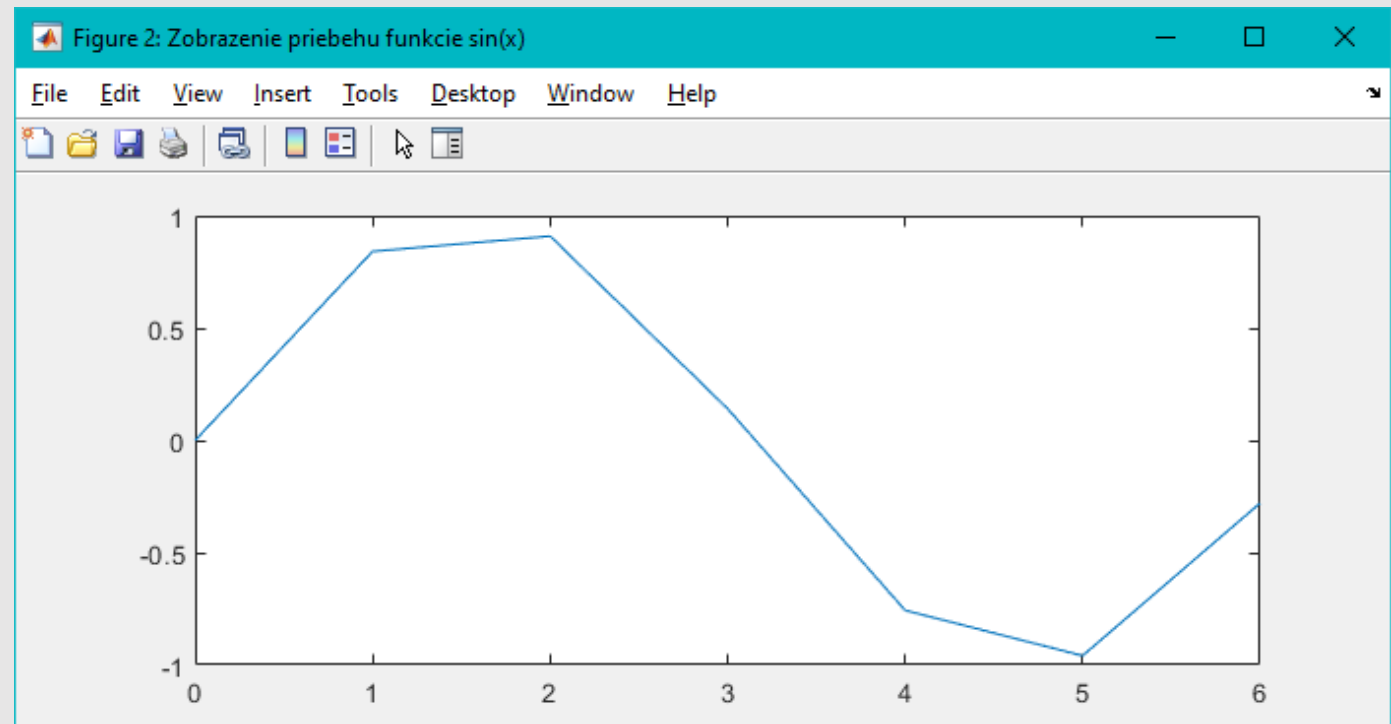
```
x = 0:2*pi
x =
    0    1    2    3    4    5    6
>> y = sin(x)
y =
    0    0.8415    0.9093    0.1411   -0.7568   -0.9589   -0.2794
```

```
>> h = figure('Name','Zobrazenie priebehu funkcie sin(x)'); plot(x,y);
```



Grafy spojitéch priebehov – PLOT

- Tieto grafy patria medzi najzakladanejšie a v oblasti elektroniky a spracovania signálov medzi najpoužívanejšie.
- Ide o takzvané „líniové grafy“ (line plots) alebo grafy spojitéch priebehov.
- **Môžeme si všimnúť, že priebeh funkcie $y=\sin(x)$ sa dá rozoznať, ale je akosi „kotrbatý“.**
- Tento problém je spôsobený tým, že funkcia plot používa linerárnu interpoláciu (spojenie rovnou čiarou) medzi diskretnými hodnotami funkcie.
- **Riešenie spočíva v jemnejšom krokovanií hodnoty premennej x . Napr. krok môže byť 0,2 alebo aj menej.**

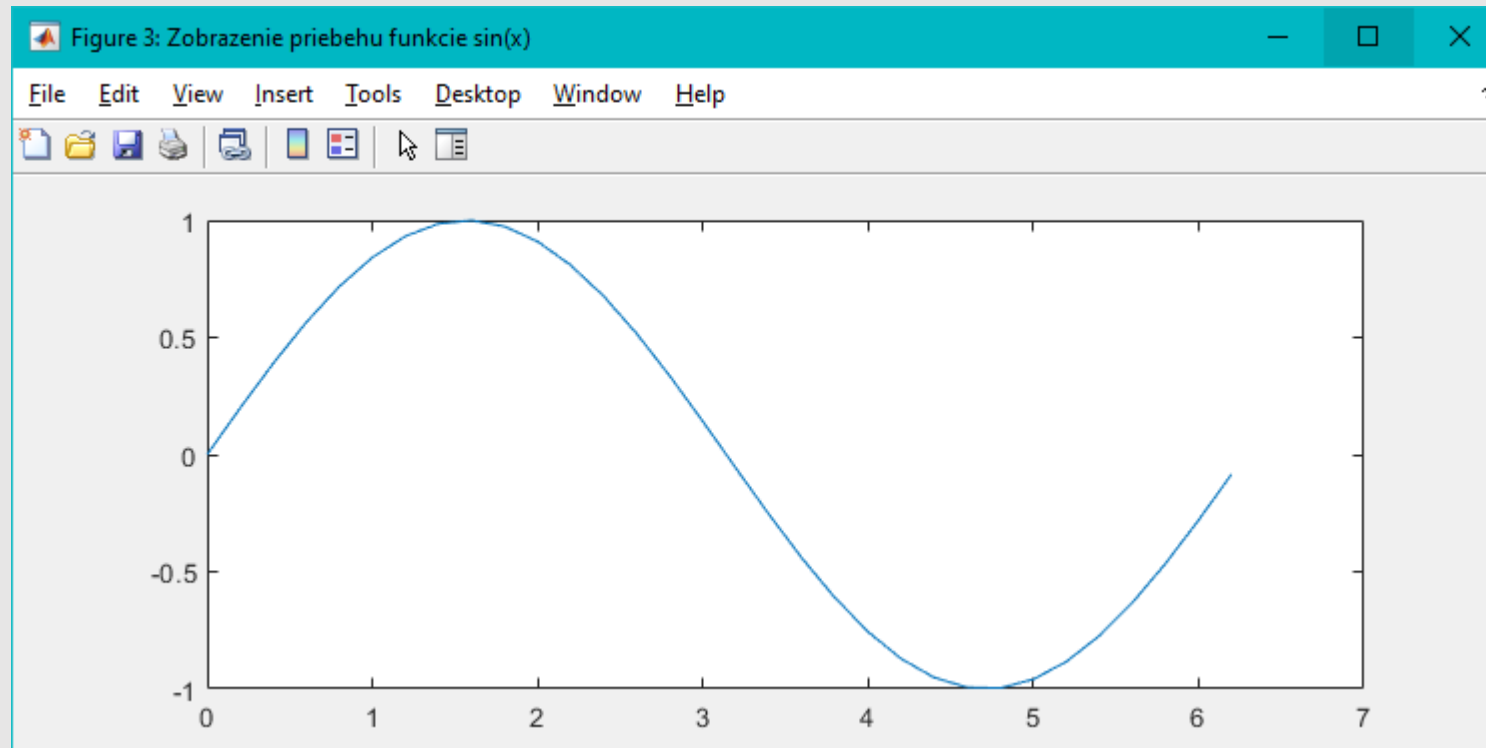


```
>> x = 0:0.2:2*pi;  
>> y = sin(x);  
>> h = figure('Name','Zobrazenie priebehu funkcie sin(x)'); plot(x,y);
```


Grafy spojitych priebehov – PLOT

- Tieto grafy patria medzi najzakladanejsie a v oblasti elektroniky a spracovania signálov medzi najpouzivanejsie.
- Ide o takzvané „líniové grafy“ (line plots) alebo grafy spojitych priebehov.
- **Riešenie spočíva v jemnejšom krokovani hodnoty premennej x. Napr. krok môže byť 0,2 alebo aj menej.**

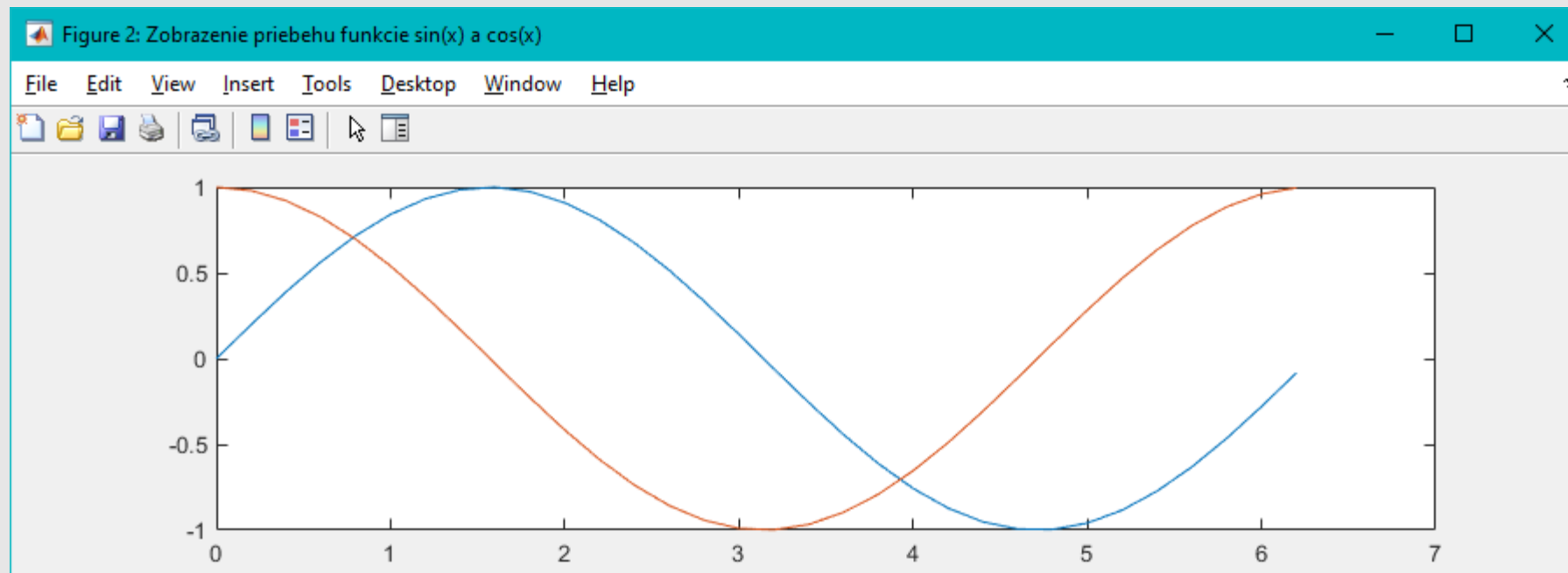
```
>> x = 0:0.2:2*pi;  
>> y = sin(x);  
>> h = figure('Name','Zobrazenie priebehu funkcie sin(x)'); plot(x,y);
```



Grafy spojitych priebehov – PLOT

- Funkciou plot je možné zobrazit aj viacero priebehov súčasne.

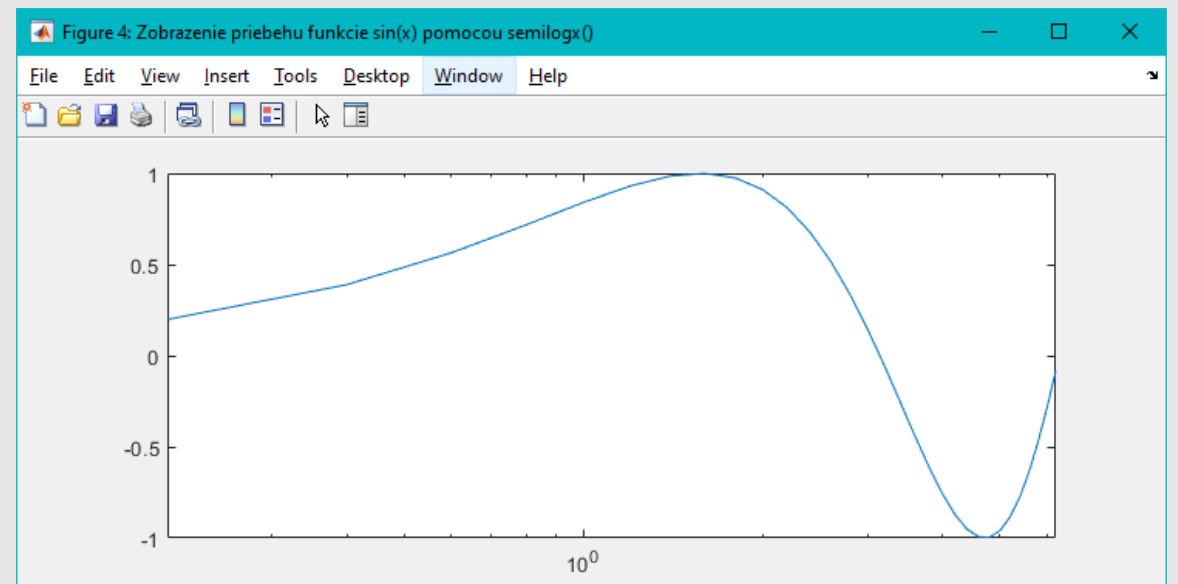
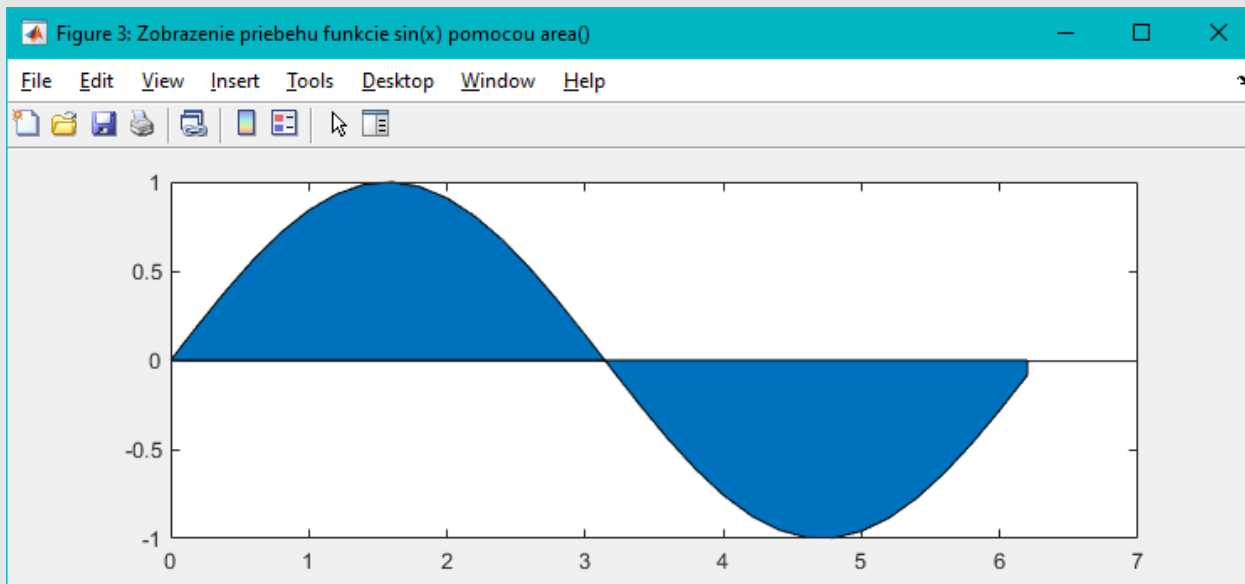
```
>> x = 0:0.2:2*pi;  
>> y = sin(x);  
>> z = cos(x);  
>> h=figure('Name','Zobrazenie priebehu funkcie sin(x) a cos(x)');  
plot(x,y,x,z)
```



Grafy spojitych priebehov – AREA a SEMILOGX

- Pomocou funkcie **area()** vieme zobrazit podobny graf ako pomocou funkcie **plot()**, rozdiel je však v tom, ze plocha pod krivkou je vyplnená.
- Funkcia **semilogx()** zasa vykresli priebeh funkcie, pricom os x bude v logaritmickej mierke.

```
>> x = 0:0.2:2*pi;  
>> y = sin(x);  
>>figure('Name','Zobrazenie priebehu funkcie sin(x) pomocou area()');  
area(x,y);  
>>figure('Name','Zobrazenie priebehu funkcie sin(x) pomocou  
semilogx()'); semilogx(x,y);
```





Úvod do Matlabu

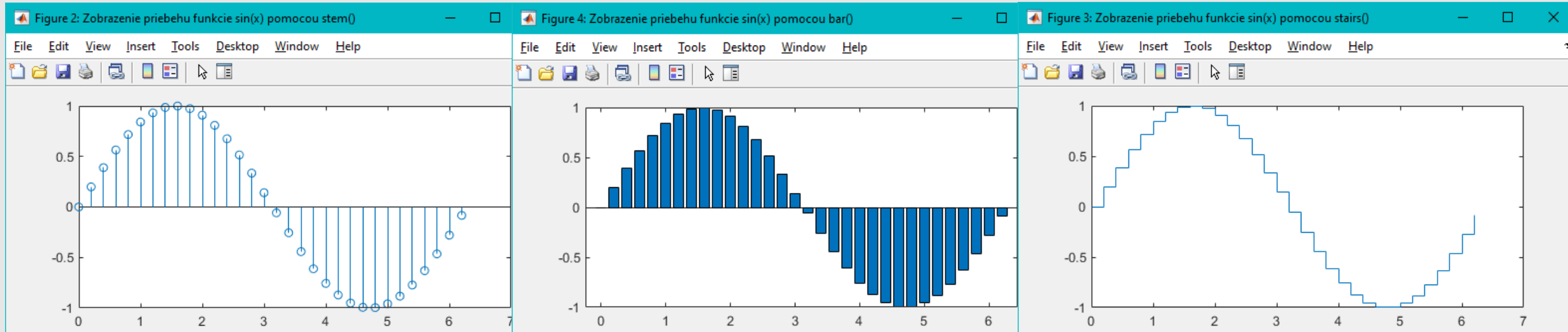
Prednáška č. 4

- Vizualizácia dát – okno Figure
- Grafy spojitých priebehov– plot, area, semilogx
- **Grafy diskrétnych priebehov – stem, stairs, bar, histogram**
- Trojrozmerné grafy – plot3, mesh, surf
- Niektoré užitočné príkazy

Grafy diskrétnych priebehov – STEM, STAIRS a BAR

- V Matlabe spravidla pracujeme s diskrétnymi hodnotami funkcie.
- Niekedy je vhodnejšie resp. presnejšie priebeh funkcie zobrazovať v diskkrétnej podobe. Na toto nám môžu poslúžiť funkcie **stem()**, **stairs()** a **bar()**.
- Volanie týchto funkcií je v podstate rovnaké, ako tomu bolo v prípade funkcie **plot()**.

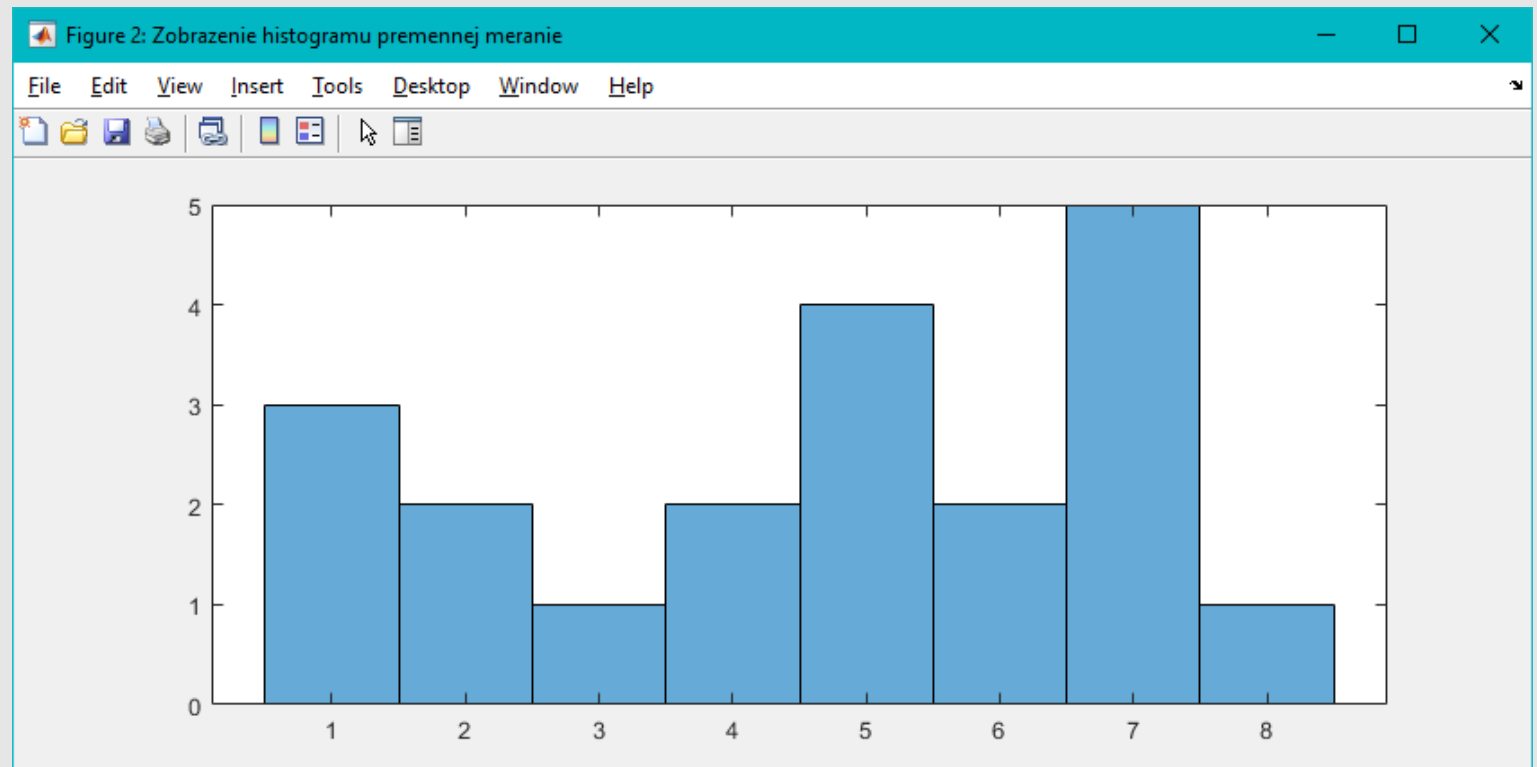
```
>> x = 0:0.2:2*pi;  
>> y = sin(x);  
>> z = cos(x);  
>> figure('Name','Zobrazenie priebehu funkcie sin(x) pomocou stem()');  
stem(x,y);  
>> figure('Name','Zobrazenie priebehu funkcie sin(x) pomocou stairs()');  
stairs(x,y);  
>> figure('Name','Zobrazenie priebehu funkcie sin(x) pomocou bar()');  
bar(x,y);
```



Grafy diskrétnych priebehov – HISTOGRAM

- Histogram z matematiky poznáme ako zobrazenie počtu výskytov jednotlivých stavov danej premennej. Na horizontálnej osi sú hodnoty premennej a na vertikálnej početnosť ich výskytov.
- **Histogram môžeme zobraziť pomocou funkcie histogram(), ktorá najprv zo vstupnej premennej vypočíta histogram a následne ho zobrazí.**

```
>> meranie = [1 1 2 2 1 3 4 4 5 5 5 5 6 7 7 7 7 8 7 6];  
>> figure('Name', 'Zobrazenie histogramu premennej meranie');  
histogram(meranie)
```





Úvod do Matlabu

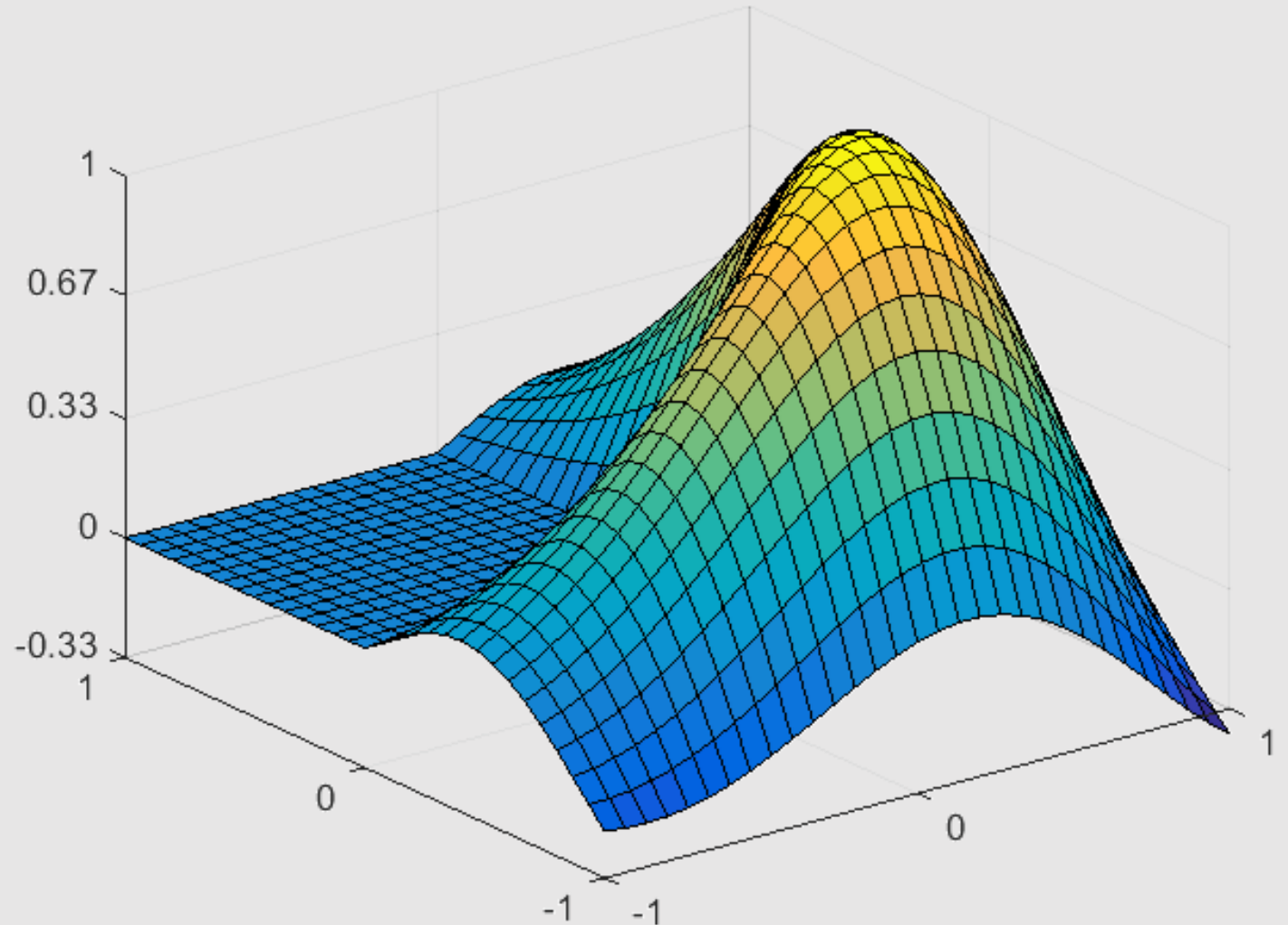
Prednáška č. 4

- Vizualizácia dát – okno Figure
- Grafy spojitých priebehov– plot, area, semilogx
- Grafy diskretných priebehov – stem, stairs, bar, histogram
- **Trojrozmerné grafy – plot3, mesh, surf**
- Niektoré užitočné príkazy

Trojrozmerné grafy

Častokrát je v Matlabe potrebné zobrazíť priestorové funkcie. Na to slúžia hlavne funkcie **plot3()**, **mesh()** a **surf()**.

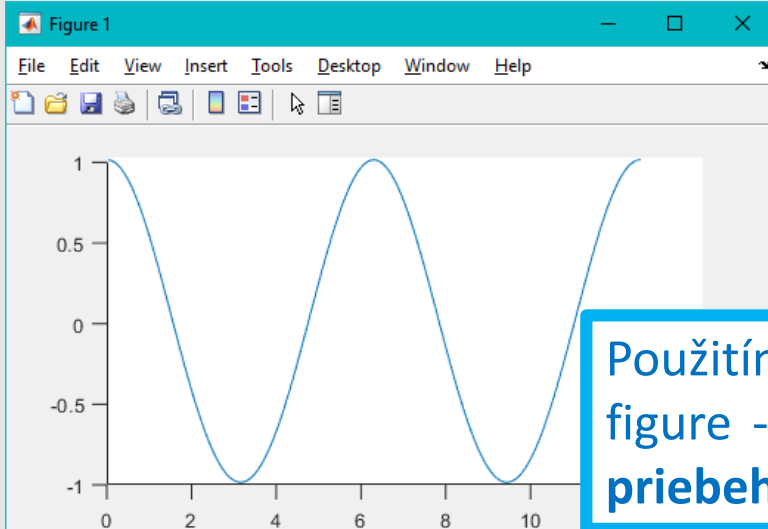
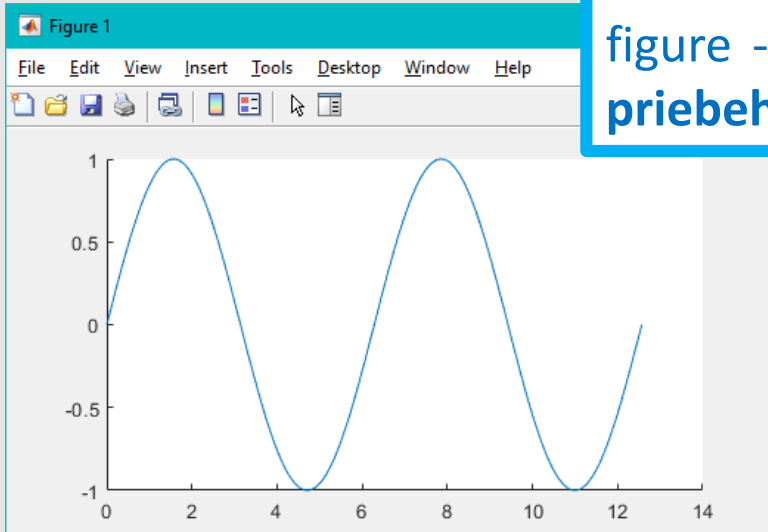
- Funkcia **plot3()** zobrazí čiary (podobne ako **plot()**), ale zobrazí ich v 3D priestore.
- Funkcia **mesh()** zobrazí 3D funkciu ako mriežku, ktorej farba a výška sú dané veľkosťou prvkov zobrazenej matice Z.
- Naopak, funkciu **surf()** použijeme vtedy, ak máme textúru, ktorou budeme 3D graf textúrovať.



Trojrozmerné grafy – PLOT3

- Funkcia **plot3()** zobrazí čiary (podobne ako **plot()**), ale zobrazí ich v 3D priestore.

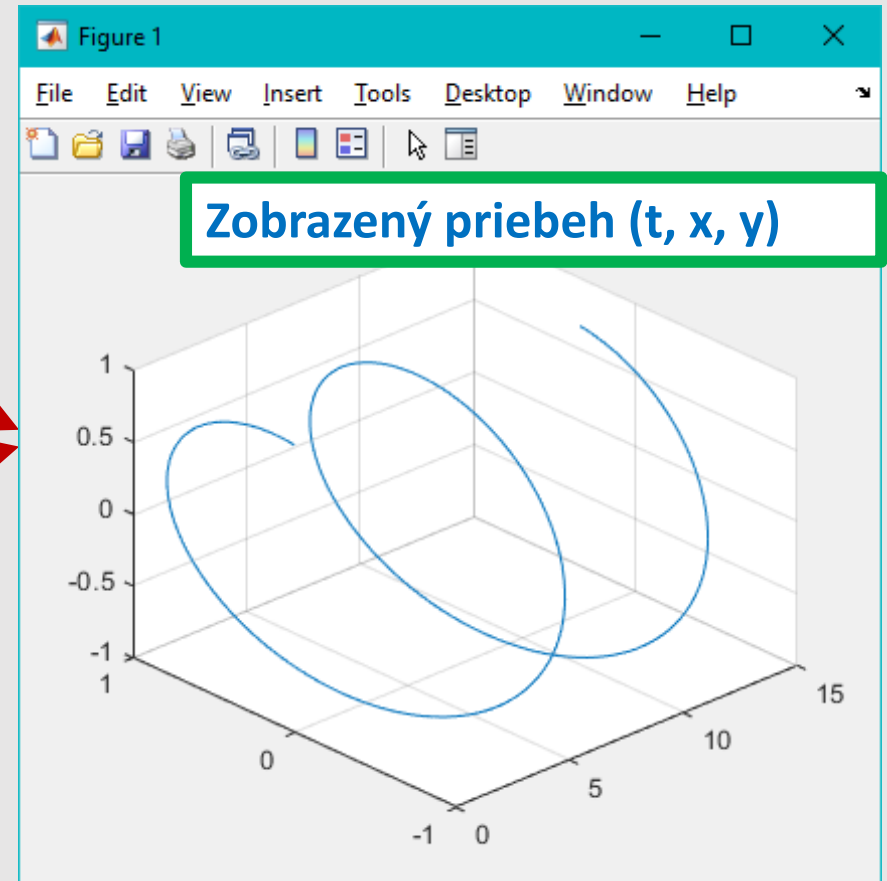
Použitím funkcie okna figure - „rotate“ **zobrazený priebeh (t, x)**



Použitím funkcie okna figure - „rotate“ **zobrazený priebeh (t, y)**

```
>> t = 0:pi/50:4*pi;  
>> x = sin(t);  
>> y = cos(t);  
>> plot3(t,x,y);
```

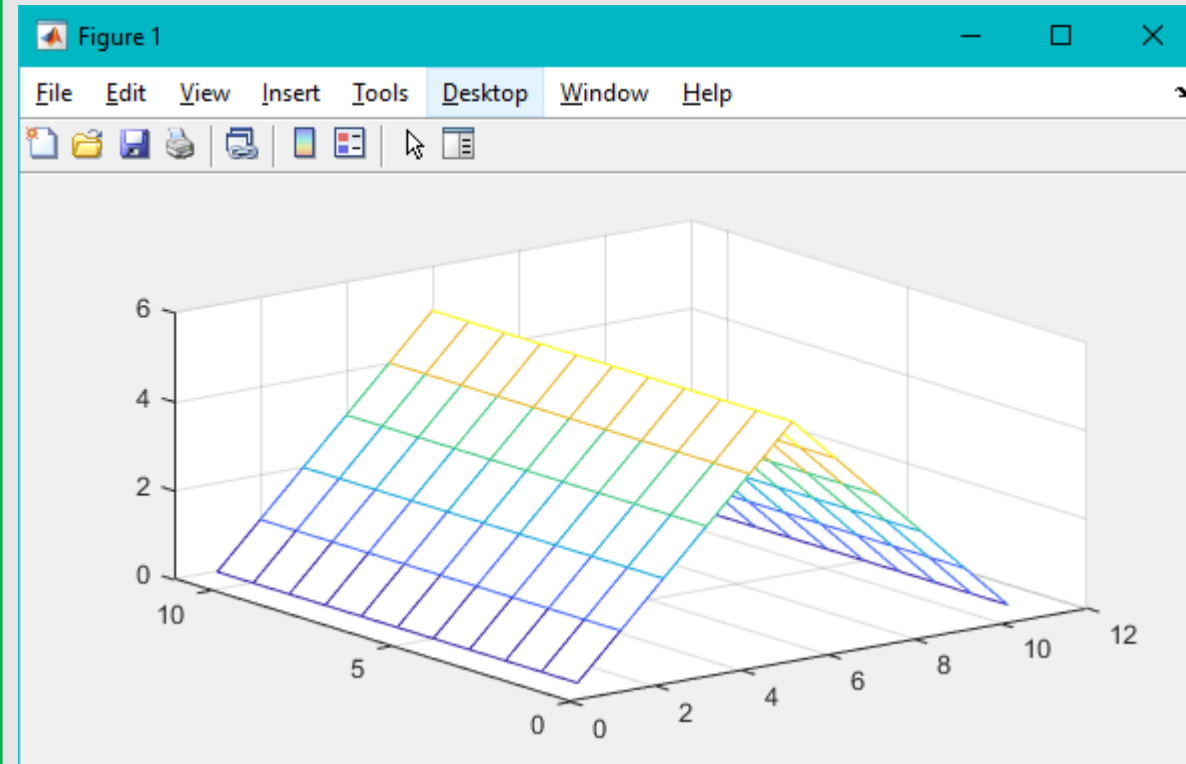
Zobrazený priebeh (t, x, y)



Trojrozmerné grafy – MESH

- Pomocou príkazu **mesh()** môžeme zobrazovať matice a tiež funkcie dvoch premenných.
- Majme napríklad maticu, ktorá má všetky riadky rovnaké a v riadku hodnota premennej do polovice stúpa a potom klesá (trojuholníkový priebeh).

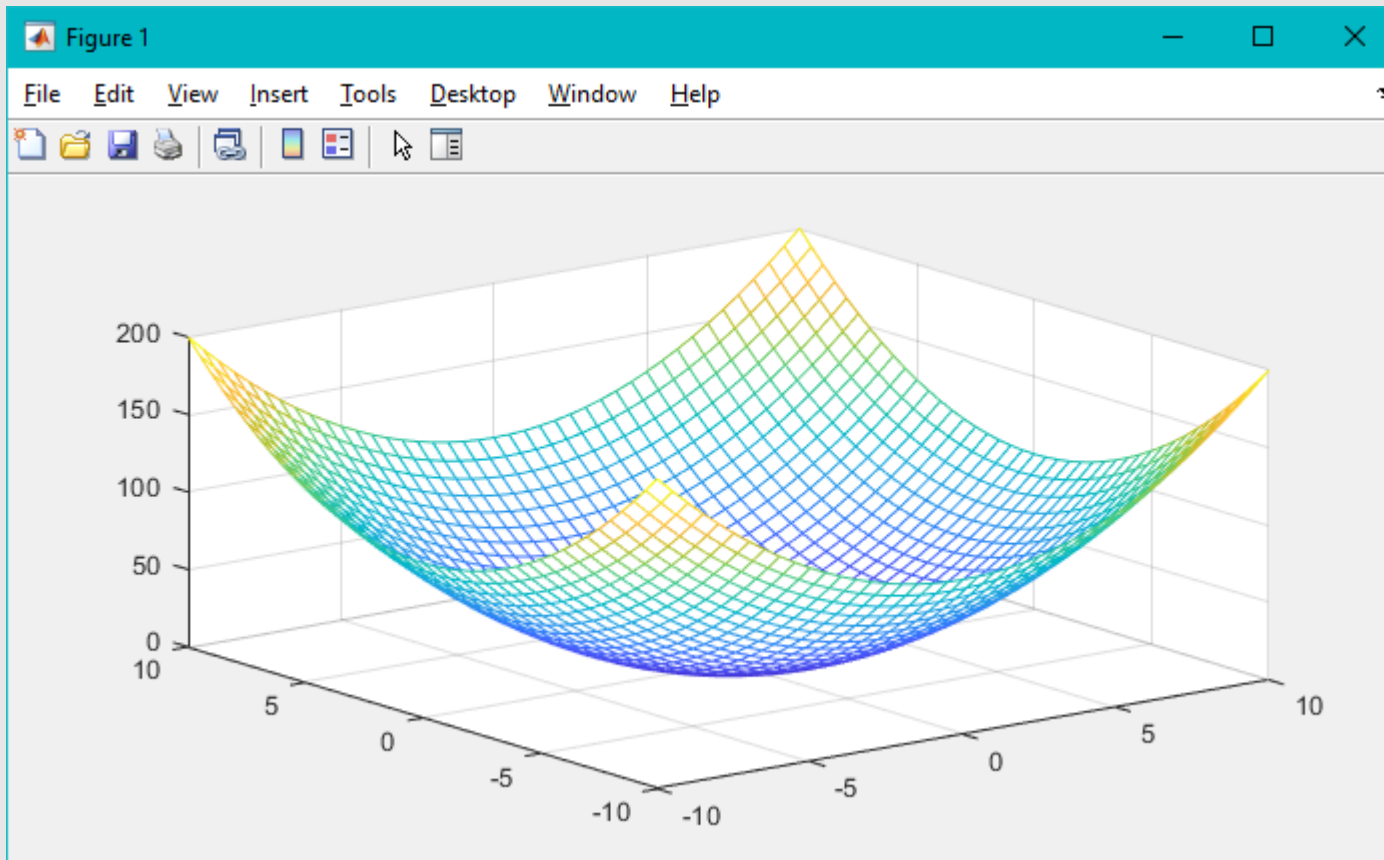
```
>> c = [0 1 2 3 4 5 4 3 2 1 0];  
>> c1= repmat(c,11,1)  
|  
c1 =  
  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
    0     1     2     3     4     5     4     3     2     1     0  
  
>> figure(1), mesh(c1)
```



Trojrozmerné grafy – MESH

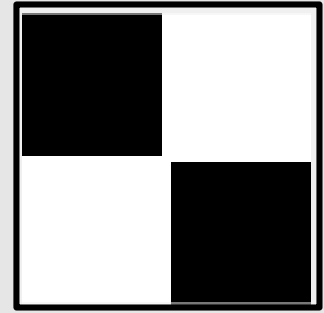
- Pomocou príkazu **mesh()** môžeme zobrazovať matice a tiež funkcie dvoch premenných.
- V prípade, že chceme zobrazíť funkciu dvoch premenných napríklad: $f(x, y) = x^2 + y^2$ je potrebné tiež pripraviť 2D priestor pomocou funkcie **meshgrid**.

```
[x, y] = meshgrid(-10:0.5:10);  
f_xy = x.^2 + y.^2;  
mesh(x, y, f_xy)
```



Trojrozmerné grafy – SURF

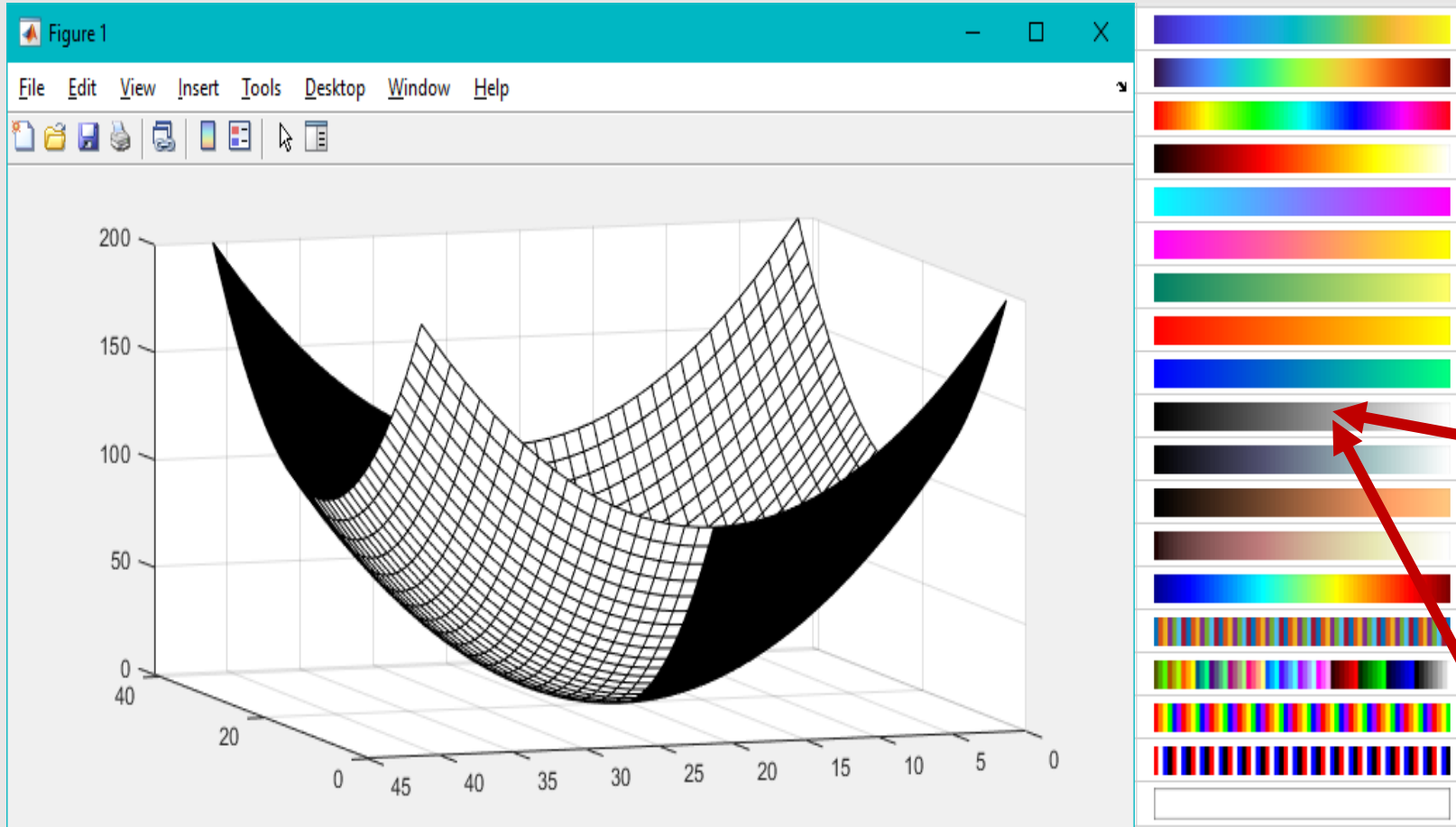
- Príkaz `surf()` použijeme vtedy, ak chceme 3D graf textúrovať. Pre jednoduchosť uvažujme funkciu $f(x, y) = x^2 + y^2$.



Textúra šachovnice

```
[x, y] = meshgrid(-10:0.5:10);  
f_xy = x.^2 + y.^2;  
  
% vytvorenie jednoduchovej textury  
textura = ones(length(x));  
textura(1:end/2,1:end/2) = 0;  
textura(end/2+1:end,end/2+1:end) = 0;  
  
figure(1); surf(f_xy, textura)  
colormap gray;
```

Graf môže mať rôzne palety farieb, momentálne je nastavená paleta „gray“





Úvod do Matlabu

Prednáška č. 4

- Vizualizácia dát – okno Figure
- Grafy spojitých priebehov – plot, area, semilogx
- Grafy diskretných priebehov – stem, stairs, bar, histogram
- Trojrozmerné grafy – plot3, mesh, surf
- **Niektoré užitočné príkazy**

Niektoré užitočné príkazy – Označenie názvu grafu a názvov jeho osí

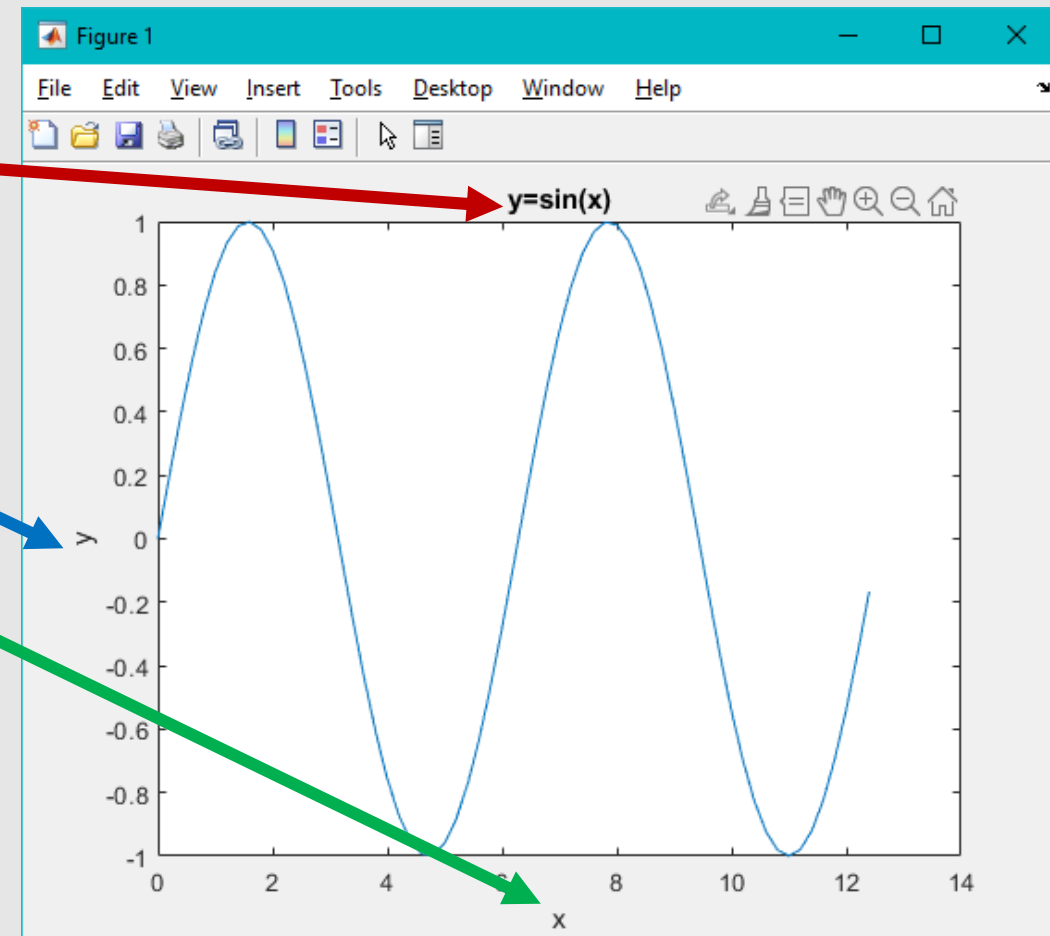
- Pomenovanie grafu a osí môžeme zabezpečiť pomocou funkcií **title()**, **xlabel()**, **ylabel()** a **zlabel()**. Tieto funkcie by však mali nasledovať bezprostredne za funkciou **plot()**.

```
>> x = 0:0.2:4*pi;  
>> y = sin(x);  
* Zobrazenie grafu, jeho názvu a označenie osí  
>> figure(1); plot(x,y); title('y=sin(x)'); xlabel('x'); ylabel('y');
```

Title(...)

ylabel(...)

xlabel(...)



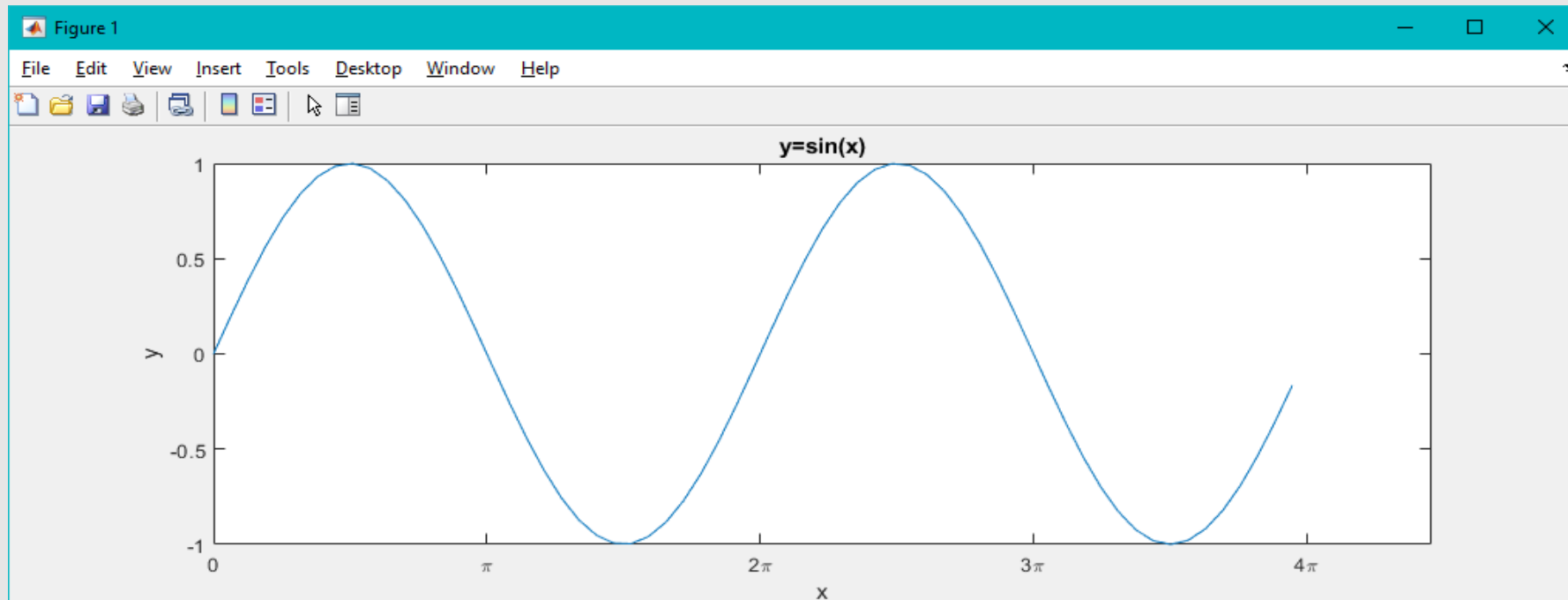
Niektoré užitočné príkazy – Označenie štítkov na osiach

- Pre označenie štítkov musíme použiť príkaz **set()** a nastaviť parametre **xTick** a **xTickLabel**

```
>> set(gca,'xTick',0:pi:4*pi)  
>> set(gca,'xTickLabel',{'0','\pi','2\pi','3\pi','4\pi'})
```

- Príkaz **gca** vráti objekt pre aktuálny graf a preto by vyššie uvedené príkazy mali nasledovať hneď za funkciou **plot()**. Inak je potrebné objekt grafu priradzovať priamo pri volaní funkcie **plot()** a príkaz **set()** aplikovať na tento objekt.

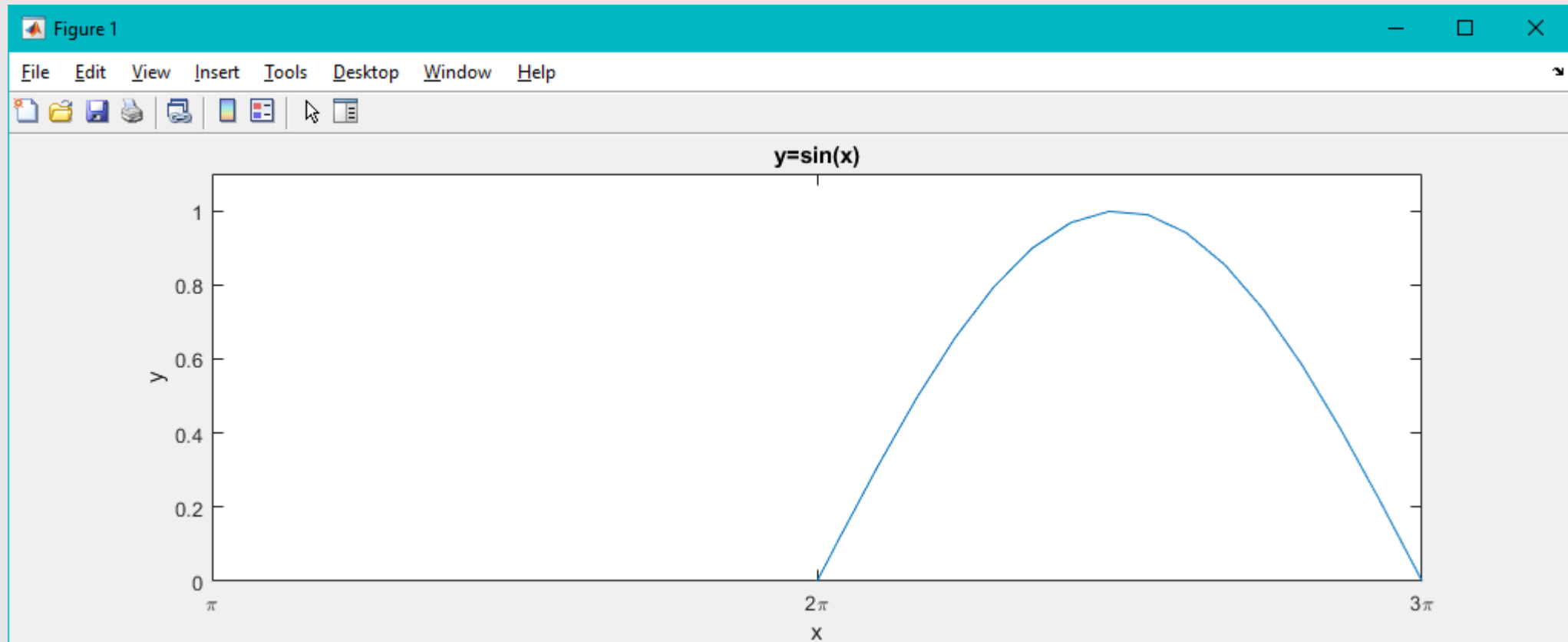
```
figure(1);plot(x,y);p=gca; title('y=sin(x)');xlabel('x');ylabel('y')  
set(p,'xTick',0:pi:4*pi)  
set(p,'xTickLabel',{'0','\pi','2\pi','3\pi','4\pi'})
```



Niektoré užitočné príkazy – Zmena oblasti zobrazenia

- Niekedy je potrebné zobrazovať iba určitú časť grafu bez toho, aby sme robili výsek patričných dát a tie zobrazovali zvlášť, stačí nastaviť limity pre jednotlivé osi. Pre tento účel poznáme príkazy `xlim()`, `ylim()` a `zlim()`.

```
xlim([pi 3*pi]);  
ylim([0 1.1]);
```

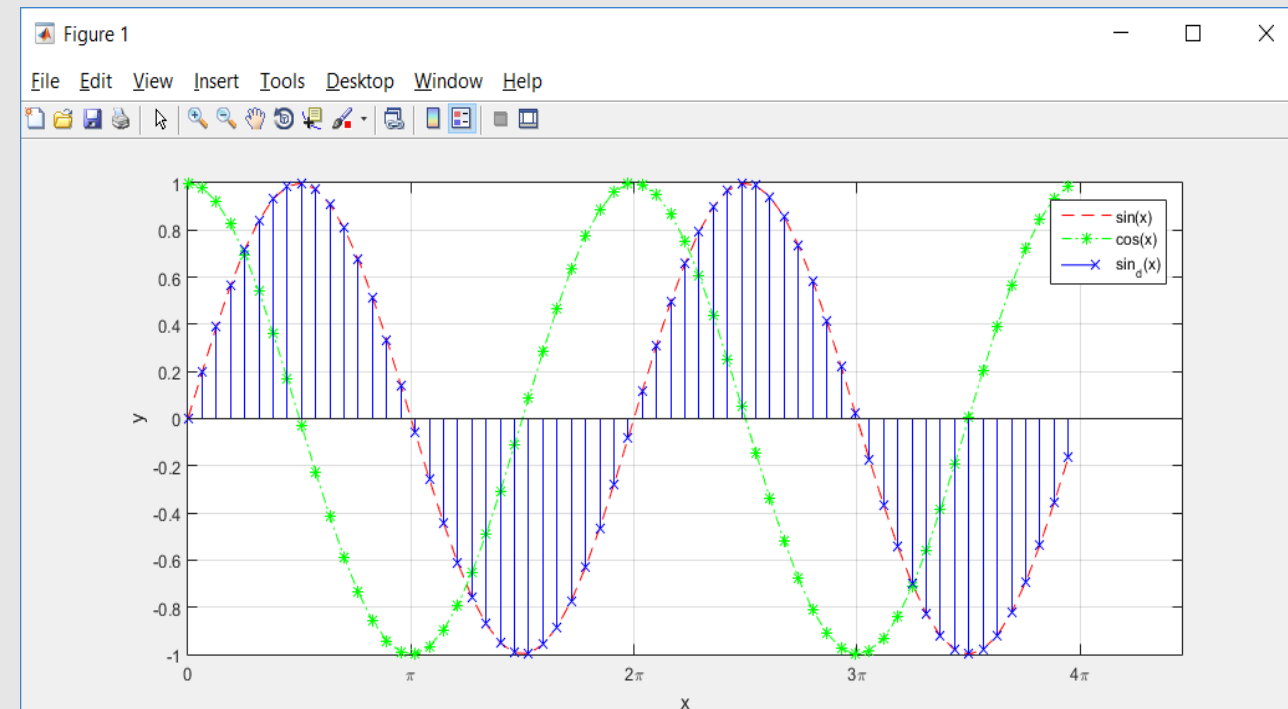


Niektoré užitočné príkazy – Zobrazenie mriežky, zmena farieb ...

- Medzi ďalšie nastavenia patrí zobrazenie mriežky zmena farieb a typov čiar resp. vyznačenie markerov.
- Zaujímavou funkciou je aj funkcia **hold on/off**. Funkcia hold „podrží“ to, čo je zobrazené v grafe a umožní do neho vykresliť ďalšie grafy, ktoré môžu byť aj iného typu.
- Ak do jedného grafu vykresľujeme viacero funkcií súčasne, je vhodné použiť príkaz **legend()**, pomocou ktorého je možné jednotlivé grafy odlíšiť.
- Parameter **grid on/off** zapína alebo vypína mriežku v grafe
- Príkaz **plot()** má aj iné parametre ako tie, ktoré sme používali doteraz.
 - ‘--r’ sa vzťahuje na zobrazenie premennej **y**
 - ‘-.g*’ sa vzťahuje na zobrazenie premennej **z**.

Tieto nové parametre nazývame špecifikátory štýlu čiar. Používajú sa nasledovne: prvý znak špecifikátora nastavuje typ čiar, druhý znak predstavuje jej farbu a tretí v prípade potreby nastavuje typ markeru.

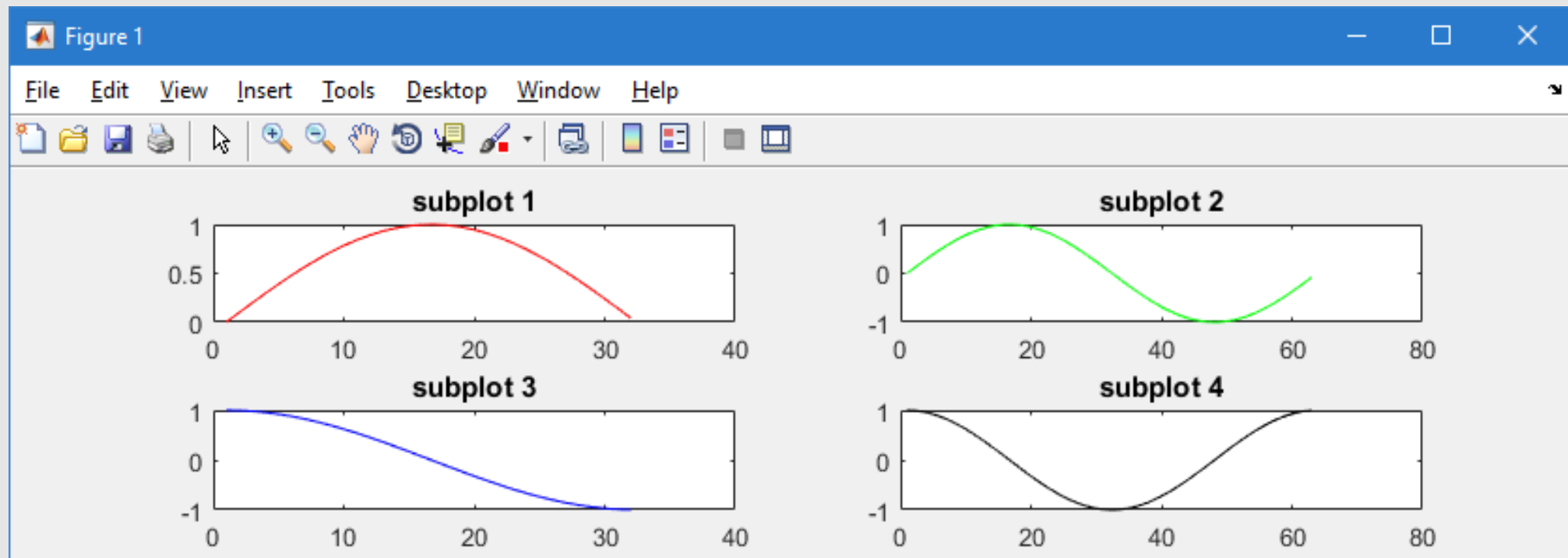
```
x = 0:0.2:4*pi;  
y = sin(x);  
z = cos(x);  
figure(1); plot(x,y, '--r',x,z, '-.g*');  
p = gca;  
set(p, 'xTick', 0:pi:4*pi)  
set(p, 'xTickLabel', {'0', '\pi', '2\pi', '3\pi', '4\pi'})  
set(p, 'xgrid', 'on')  
set(p, 'ygrid', 'on')  
xlabel('x'); ylabel('y');  
hold on  
stem(x,y, 'bx');  
hold off  
legend('sin(x)', 'cos(x)', 'sin d(x)')
```



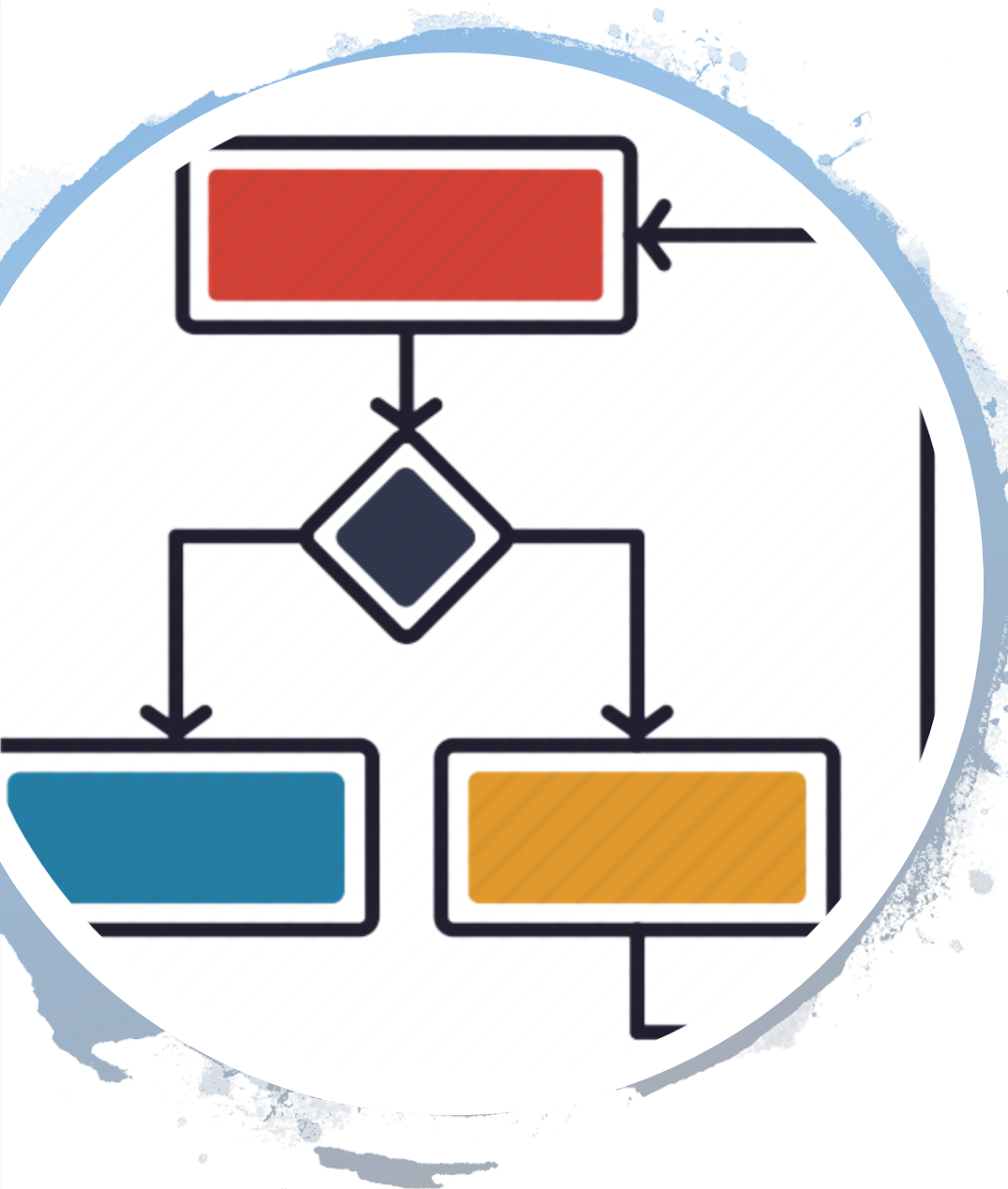
Niektoré užitočné príkazy – **SUBPLOT**

- Niekedy je potrebné vykresliť viacero grafov do jedného okna **figure**. K tomuto účelu slúži funkcia **subplot(m,n,p)**, ktorá sa volá s tromi vstupnými parametrami.
 - m** predstavuje počet riadkov,
 - n** počet stĺpcov,
 - p** poradové číslo grafu.

```
figure(1);  
subplot(2,2,1),plot(sin(0:0.1:pi),'r'); title('subplot 1');  
subplot(2,2,2),plot(sin(0:0.1:2*pi),'g');title('subplot 2');  
subplot(2,2,3),plot(cos(0:0.1:pi),'b');title('subplot 3');  
subplot(2,2,4),plot(cos(0:0.1:2*pi),'k');title('subplot 4');
```



Nabudúce



- **Riadiace príkazy**

- if
- else
- elseif

- **Slučky/cykly**

- for
- while
- parfor

- **Vlastné funkcie**

- Anonymné funkcie
- Funkcie definované v skripte
- Regulárna funkcia
- Viaceré funkcie definované v jednom m-súbore