


# 8 Grafika v Matlabe

## 8.1 Grafické prostredie v MATLABe

Grafické funkcie automaticky otvoria nové grafické okno, pokiaľ už predtým žiadne neexistuje. V prípade, že je už nejaké grafické okno otvorené, je použité na vykresľovanie to, ktoré "je aktívne" (jedná sa o posledné okno, ktoré bolo otvorené alebo na ktoré bolo kliknuté).

Funkcia	Opis
<code>clf</code>	zmazanie aktuálneho obrazca
<code>close</code>	uzatvorenie okna grafického obrázku
<code>figure</code>	otvorenie okna grafického obrázku
<code>gcf</code>	číslo aktuálneho okna grafického obrázku

Príkaz `figure` poskytuje nasledovné možnosti zápisu:

- pre vytvorenie nového okna: `figure`
- pre vytvorenie okna s číslom `h`: `figure(h)` 
- pre vytvorenie okna s vrátením jeho čísla do premennej `k`: `k=figure`

### Príklad 8.1:

```
% otvorenie okna grafického obrázku č.3
>> figure(3)

% otvorenie okna grafického obrázku č.2
>> figure(2)

>> gcf           % číslo aktuálneho okna grafického obrázku
ans =
     2

>> close        % uzatvorenie aktívneho okna grafického obrázku
>> close(3)     % uzatvorenie okna grafického obrázku č.3
>> close all    % uzatvorenie všetkých okien grafických obrázkov
```

## 8.2 Vytváranie dvojrozmerných grafov

### Základné grafické funkcie zobrazenia

Funkcia	Opis
<b>fill</b>	vyplnenie dvojrozmerného mnohouholníka
<b>loglog</b>	zobrazenie, ak osi majú logaritmické mierky
<b>plot</b>	lineárne zobrazenie
<b>plotyy</b>	lineárne zobrazenie s mierkami na ľavej a pravej strane
<b>semilogx</b>	zobrazenie, ak os x je v logaritmickej mierke
<b>semilogy</b>	zobrazenie, ak os y je v logaritmickej mierke

### Lineárne zobrazenie

Lineárne zobrazenie bodov, priebehov veličín je možné uskutočniť funkciou `plot(x1, y1, x2, y2, ..., xn, yn)`, kde  $x_1, y_1, x_2, y_2, \dots, x_n, y_n$  sú vektory s hodnotami súradníc bodov.

#### Príklad 8.2: Lineárne zobrazenie paraboly

```
>> x = [-5:0.1:5];
>> y = x.^2;
>> plot 🍌 % vykreslí hodnoty 'y' v závislosti na ich poradí
>> plot(x,y) % vykreslí hodnoty 'y' v závislosti na hodnotách 'x'
```

### Farba, typ a znaky čiar

Pri zobrazovaní pomocou funkcie `plot` je možné zadať farbu, typ a znaky vykreslovanej čiary nasledovným spôsobom: `plot(x, y, 'farba_typ_znak')`, kde `farba_typ_znak` je reťazec zložený zo symbolov vyjadrujúcich farbu, typ a znak čiary.

Symbol	Farba (RGB)	Symbol	Typ čiar
<b>c</b>	cyan (0 1 1)	-	plná čiara (auto)
<b>m</b>	magenta (1 0 1)	--	čiarkovaná
<b>y</b>	žltá (1 1 0)	-.	bodkočiarkovaná
<b>r</b>	červená (1 0 0)	:	bodková
<b>g</b>	zelená (0 1 0)	<b>none</b>	žiadna
<b>b</b>	modrá (0 0 1)		
<b>w</b>	biela (1 1 1)		
<b>k</b>	čierna (0 0 0)		
Znak	Opis znaku		
<b>+</b>	plus		
<b>o</b>	kruh		
<b>*</b>	hviezdica		
<b>.</b>	bodka		
<b>x</b>	křížik		
<b>square</b>	štvrtok		
<b>diamond</b>	kosoštvorec		
<b>V</b>	dole orientovaný trojuholník		
<b>&lt;</b>	vľavo orientovaný trojuholník		

Znak	Opis znaku
>	vpravo orientovaný trojuholník
pentagram	päťcípá hviezda
hexagram	šesťcípá hviezda
none	žiadny (auto)

**Príklad 8.3: Farba, typ a znaky čiary**

```
>> x = [-5:5];
>> y = x.^2;
>> plot(x,y,'r-*',x,y/2,'m-.')
```

## Zobrazenie v logaritmických súradniciach

Pri funkciách `semilogx`, `semilogy` a `loglog` ide o podobné zobrazenie ako v prípade lineárneho zobrazenia, len niektorá z osí alebo obe sú v logaritmickej mierke.

**Príklad 8.4: Definovanie premenných pre rôzne zobrazenia**

```
>> x = 0:0.01:4;
>> y = sin(0:0.01:4);
>> z = x.^2;
```

**Príklad 8.5: Zobrazenie, ak osi majú logaritmické mierky**

```
>> figure
>> loglog(x,y)
```

## Vyplnenie dvojrozmerného mnohoúhelníka

Funkciu `fill(x,y,c)` je možné zobrazit farebný mnohoúhelník, kde `x,y` sú vektory súradníc bodov mnohoúhelníka a `c` je premenná, ktorá definuje farbu.

**Príklad 8.6: Vyplnenie trojuholníka červenou farbou**

```
>> figure
>> fill([0,1,0.5],[0,0,1],'r')
```

## Ďalšie zobrazenia

**Príklad 8.7: Lineárne zobrazenie s mierkami na ľavej a pravej strane**

```
>> figure
>> plotyy(x,y,x,z)
```

**Príklad 8.8: Bodeho diagram - definovanie premenných**

```
>> [amp,faza,w] = bode([1],[1 2 1])
amp =
    0.9999
    0.9996
    ...
    0.0004
    0.0001
```

```

faza =
  -1.1459
  -2.2915
  ...
 -177.7085
 -178.8541

w =
  0.0100
  0.0200
  ...
 50.0000
100.0000

>> amp_db = 20*log10(amp)
amp_db =
  -0.0009
  -0.0035
  ...
 -67.9623
 -80.0009

```

**Príklad 8.9: Lineárne zobrazenie Bodeho diagramu**

```
>> plot(w, amp_db)
```

**Príklad 8.10: Zobrazenie, ak os x je v logaritmickej mierke**

```
>> semilogx(w, amp_db)
```

## 8.3 Základné riadiace funkcie pre zobrazenie

Funkcia	Opis
<b>axes</b>	vytvorenie osí na ľubovoľnú pozíciu
<b>axis</b>	nastavenie parametrov osí a vzhľadu
<b>caxis</b>	nastavenie rozsahu farieb osí
<b>cla</b>	zmazanie osí
<b>box</b>	zobrazenie rámika osí
<b>gca</b>	vracia číslo aktuálnych osí
<b>get</b>	zistenie nastavenia grafického objektu
<b>hold</b>	zachovanie aktuálneho grafu
<b>set</b>	nastavenie vlastností grafického objektu
<b>subplot</b>	rozdelenie grafického okna na subokná

## Nastavenie osí

Funkciou `axes` je možné vytvoriť osi a umiestniť ich na ľubovoľné miesto v grafickom okne

### Príklad 8.11:

```
% nastavenie parametrov osí a vzhľadu
% axes('position',[left, bottom, width, height]),
% kde left, bottom sú body poč. súradnicového systému
% a width, height sú veľkosti x-ovej a y-ovej osi (rozmery 1x1)
>> axes('position',[0.2, 0.2, 0.7, 0.7])
```

### Príklad 8.12:

```
% nastavenie rozmerov osí x, y
>> axis([0,100,0,100])

% nastavenie rozmerov osí x, y a z
>> axis([0,10,0,10,1,5])

% funkcia vracia vektor nastavenia osí
>> axis
ans =
     0     10     0     10     1     5

% rozmery osí sa nastavujú automaticky
>> axis auto

% vypnutie zobrazenia osí
>> plot([0:0.01:pi],sin([0:0.01:pi]))
>> axis off

% zapnutie zobrazenia osí
>> axis on
```

## Zobrazenie viacerých grafov

Ak je požadované pridať do jedného grafu viac priebehov alebo v ňom vykonať ďalšie grafické operácie, potom je potrebné použiť príkaz `hold on`.

### Príklad 8.13: Zachovanie aktuálneho grafu

```
>> plot(x,y,'r')
>> hold on
>> plot(x,z,'b')
```

**Príklad 8.14: Zrušenie zachovania aktuálneho grafu**

```
>> hold off
```

Príkaz `subplot(m,n,p)` rozdelí graf na subokná, kde  $m, n, p$  vyjadrujú počet riadkov, stĺpcov a poradie okna

**Príklad 8.15: Ukážka rozdelenia grafického okna na subokná - 3 riadky a 4 stĺpce**

```
>> subplot(3,4,7)
```

```
1  2  3  4
5  6  7  8
9 10 11 12
```

**Príklad 8.16: Rozdelenie grafického okna na subokná**

```
>> subplot(1,2,1)
```


```
>> plot(x,y,'r')
```

```
>> subplot(1,2,2)
```

```
>> plot(x,z,'b')
```

**Nastavenie parametrov grafických objektov**


Pri vytvorení grafického objektu mu Matlab prideli identifikačné číslo. Pomocou tohto čísla je možné potom pristupovať k jednotlivým objektom. Číslo objektu vracia vždy funkcia, ktorá ho vytvorila.

Nastavenie objektu je možné získať príkazom `get` , kde  $h$  je číslo objektu. Nastaviť určitú položku vlastnosti je možné príkazom `set(h, 'položka', hodnota)`.

**Príklad 8.17: Získanie čísla objektu**

```
>> h = plot(w, amp_db);
```

**Príklad 8.18: Výpis položiek vlastností**

```
>> get 
    Color = [0 0 1]
    EraseMode = normal
    LineStyle = -
    LineWidth = [0.5]
    Marker = none
    MarkerSize = [6]
    MarkerEdgeColor = auto
    MarkerFaceColor = none
    XData = [ (1 by 45) double array]
    YData = [ (1 by 45) double array]
    ZData = []

    BeingDeleted = off
```

```

ButtonDownFcn =
Children = []
Clipping = on
CreateFcn =
DeleteFcn =
BusyAction = queue
HandleVisibility = on
HitTest = on
Interruptible = on
Parent = [102.017]
Selected = off
SelectionHighlight = on
Tag =
Type = line
UIContextMenu = []
UserData = []
Visible = on

```

#### **Príklad 8.19: Zmena typu čiary**

```

% pôvodné: LineStyle = -
% zmeniť na: LineStyle = x
>> set(h, 'LineStyle', 'x')

```

#### **Príklad 8.20: Zmena farby čiary**

```

% pôvodné: Color = [0 0 1] (R,G,B)
% zmeniť na: Color = [0 1 0] (Color = green alebo Color = g)
>> set(h, 'Color', 'green')

```

## 8.4 Označenie a popis grafov

Funkcia	Opis
<b>grid</b>	čiarová sieť
<b>gtext</b>	text umiestnený myšou
<b>legend</b>	vytvorenie legendy priebehov
<b>text</b>	umiestnenie textu
<b>title</b>	názov grafu
<b>xlabel</b>	popis osi x
<b>ylabel</b>	popis osi y
<b>zlabel</b>	popis osi z

**Príklad 8.21: Vykreslenie a popis grafov**

```
>> t = 0:0.01:2*pi;           % definovanie premennych
>> y = sin(t);

>> plot(t,y)                  % vykreslenie
>> xlabel('t = 0 \rightarrow 2pi') % popis osi x
>> ylabel('sin(t)')          % popis osi y
>> title('Priebeh funkcie sínus') % názov grafu
```

**Príklad 8.22: Zápís textu pomocou súradníc dát**

```
>> text(3*pi/4, sin(3*pi/4), '\leftarrow sin(t) = 0.707')
>> text(7*pi/6, sin(7*pi/6), 'sin(t) = -0.5 \rightarrow', ...
'HorizontalAlignment', 'right')
```

**Príklad 8.23: Legenda**

```
>> y1 = sin(t); y2 = cos(t); y3 = y1+y2;
>> plot(t,y1,t,y2,t,y3);
>> legend('y_1=sin(t)', 'y_2=cos(t)', 'y_3=sin(t)+cos(t)')
```

**Príklad 8.24: Mriežka (hlavná, vedľajšia)**

```
>> plot(t,y1,t,y2,t,y3);
>> grid on
>> grid minor
```

**Poznámka:** ak chceme zmeniť orientáciu popisu vertikálnej osi 🖱️ z vertikálnej na vodorovnú môžeme použiť príkaz `set(gca, 'YLabel', 'Rotation', 0.0)`.

**Špeciálne znaky (T<sub>E</sub>X)**

Znak	Zápis	Znak	Zápis	Znak	Zápis
$\alpha$	<code>\alpha</code>	$\beta$	<code>\beta</code>	$\gamma$	<code>\gamma</code>
$\delta$	<code>\delta</code>	$\epsilon$	<code>\epsilon</code>	$\omega$	<code>\omega</code>
$\lambda$	<code>\lambda</code>	$\xi$	<code>\xi</code>	$\pi$	<code>\pi</code>
$\rho$	<code>\rho</code>	$\sigma$	<code>\sigma</code>	$\tau$	<code>\tau</code>
$\sigma$	<code>\sigma</code>	$\Delta$	<code>\Delta</code>	$\Sigma$	<code>\Sigma</code>
$\nabla$	<code>\nabla</code>	$\partial$	<code>\partial</code>	$\infty$	<code>\infty</code>
$\sqrt{\quad}$	<code>\surd</code>	$\int$	<code>\int</code>	$\neq$	<code>\neq</code>
$\in$	<code>\in</code>	$\subset$	<code>\subset</code>	$\subseteq$	<code>\subseteq</code>
$\leq$	<code>\leq</code>	$\geq$	<code>\geq</code>	$\uparrow$	<code>\uparrow</code>
$\wedge$	<code>\wedge</code>	$\vee$	<code>\vee</code>	$\downarrow$	<code>\downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\leftarrow$	<code>\leftarrow</code>	$\rightarrow$	<code>\rightarrow</code>
$\neg$	<code>\neg</code>	$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>
horný index	<code>^{\{horný index\}}</code>	dolný index	<code>_{\{dolný index\}}</code>		



## 8.5 Špeciálne grafy

Špeciálne grafy sú využívané najmä v oblastiach:

- štatistické grafy
- zobrazenie diskretných dát
- priame a rýchlostné vektorové grafy
- hladinové zobrazenie

### Štatistické grafy

Funkcia	Opis
<b>bar</b>	dvojrozmerný zvislý stĺpcový graf
<b>bar3</b>	trojrozmerný zvislý stĺpcový graf
<b>barh</b>	dvojrozmerný vodorovný stĺpcový graf
<b>bar3h</b>	trojrozmerný vodorovný stĺpcový graf
<b>errorbar</b>	zobrazenie s odchýlkou
<b>hist</b>	zobrazenie histogramu
<b>rose</b>	zobrazenie uhlového histogramu
<b>polar</b>	zobrazenie v polárnych súradniciach
<b>area</b>	plošný graf
<b>pie</b>	kruhový graf

#### Príklad 8.25:

```
>> Y = [6 3 2;9 6 4;7 5 4;5 6 5;4 3 2];  
>> bar3(Y)  
>> xlabel('Os x'), ylabel('Os y'),  
>> zlabel('Os z')
```

#### Príklad 8.26: Zobrazenie v polárnych súradniciach

```
>> t = 0:0.01:2*pi;  
>> polar(t, abs(sin(2*t)).*cos(2*t));
```

### Zobrazenie diskretných dát

Funkcia	Opis
<b>stem</b>	dvojrozmerné kmeňové zobrazenie
<b>stem3</b>	trojrozmerné kmeňové zobrazenie
<b>stairs</b>	schodové zobrazenie

#### Príklad 8.27: schodové zobrazenie

```
>> x = 0:0.25:10;  
>> stairs(x, sin(x))
```

#### Príklad 8.28: Stonkové zobrazenie

```
>> x = 0:0.1:4;  
>> y = sin(x.^2).*exp(-x);  
>> stem(x,y)
```

### Priame a rýchlostné vektorové grafy

Funkcia	Opis
<b>compass</b>	zobrazenie vektorov v polárnych súradniciach
<b>feather</b>	zobrazenie vektorov pozdĺž vodorovnej čiary
<b>quiver</b>	zobrazenie 2-D vektorov definovaných (u,v) parametrami
<b>quiver3</b>	zobrazenie 3-D vektorov definovaných (u,v,w) parametrami

**Príklad 8.29: Zobrazenie v polárnych súradniciach, ktoré sú zadané uhlom a veľkosťou**

```
>> uhol = [45 0 90 150 230 300];
>> velkost = [5 7 10 6 12 8];
>> ruhol = uhol*pi/180; % prepočet na radiány
>> [x,y] = pol2cart(ruhol,velkost); % prepočet do kart. súradníc
>> compass(x,y)
```

**Príklad 8.30: Zobrazenie vektorov pozdĺž x-ovej osi, ktoré sú zadané uhlom od 90° do 0° a veľkosťou 5**

```
>> uhol = 90:-10:0;
>> velkost = 5*ones(size(uhol));
>> [u,v] = pol2cart(uhol*pi/180,velkost);
>> feather(u,v)
```

**Príklad 8.31: Zobrazenie vektorov intenzity el. magnetického poľa dvoch nábojov pomocou zobrazenia zmeny súradníc**

```
>> x = -2:.2:2;
>> y = -1.5:.2:1.5;
>> [xx,yy] = meshgrid(x,y);
>> zz = xx.*exp(-xx.^2-yy.^2);
>> [px,py] = gradient(zz,.2,.2);
>> quiver(x,y,px,py,2)
```

## Hladinové zobrazenie - rezy a pohľady

Funkcia	Opis
<b>clabel</b>	popis hladinového zobrazenia
<b>contour</b>	plošné hladinové obrazenie
<b>contour3</b>	trojrozmené hladinové obrazenie
<b>contourf</b>	plošné hladinové obrazenie s farebnou výplňou
<b>pcolor</b>	pseudofarebné zobrazenie

**Príklad 8.32: Príkladová funkcia 'peaks'**

```
>> z = peaks(25);
>> pcolor(z)
>> colormap(hsv)
```

**Príklad 8.33: Hladinové zobrazenie obrazca s popisom jednotlivých hladín**

```
>> [C,h] = contour(z,10);
>> clabel(C,h)
```

**Príklad 8.34: Hladinové zobrazenie obrazca s farebným rozlíšením**

```
>> [C,h] = contourf(z,10);
>> caxis([-10 10])
```

## 8.6 Vytváranie trojrozmerných grafov

### Príprava údajov

Pri zobrazení trojrozmerného obrazca potrebujeme mať dve matice (napr.  $X$ ,  $Y$ ), ktorých prvky  $X(i, j)$ ,  $Y(i, j)$  budú obsahovať súradnice bodov v rovine. Na generovanie týchto matíc slúži funkcia `meshgrid`:

`[X, Y] = meshgrid(v1, v2)` - vektor  $v1$  určuje interval na osi  $x$  s daným krokom a vektor  $v2$  určuje interval na osi  $y$  s daným krokom

`[X, Y] = meshgrid(v)` - vektor  $v$  určuje interval na osi  $x$  aj na osi  $y$

### Základné funkcie 3-D zobrazenia

V 3-D zobrazeniach sa používajú podobné funkcie `fill3` a `plot3` ako v 2-D grafike

#### Funkcia Opis

**fill3** Vyplnenie trojrozmerného mnohoúhelníka

**plot3** zobrazenie čiar a bodov 3-D priestore (lineárne zobrazenie)

#### Príklad 8.35: Zobrazenie špirály (lineárne zobrazenie)

```
>> t = 0:pi/50:10*pi;
```

```
>> plot3(sin(t), cos(t), t)
```

#### Príklad 8.36: Zobrazenie trojrozmerného obrazca

```
>> [x, y] = meshgrid([-2:0.1:2]);
```

```
>> z = x.*exp(-x.^2-y.^2);
```

```
>> plot3(x, y, z)
```

#### Príklad 8.37: Zobrazenie kocky s rezom

```
>> x = [0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1 0 0];
```

```
>> y = [0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1];
```

```
>> z = [0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0];
```

```
>> axis([0 1 0 1 0 1]);
```

```
>> hold on
```

```
>> for i=1:2:23
```

```
plot3([x(i) x(i+1)], [y(i) y(i+1)], [z(i) z(i+1)], 'r-');
```

```
end
```

```
>> fill3([0 0.5 0], [0 0 0.5], [0.5 1 1], 'g')
```

### Zobrazenie povrchu obrazca maticovou reprezentáciou

#### Funkcia Opis

**mesh** trojrozmerné sieťové zobrazenie povrchu

**meshc** kombinácia zobrazení mesh/contour

**meshz** zobrazenie mesh s nulovou rovinou

**slice** objemové predstavenie zobrazenia

**surf** zobrazenie povrchu zatienením

Funkcia	Opis
<b>surf</b>	kombinácia zobrazení surf/contour
<b>surf</b>	zobrazenie surf s vysvietením
<b>surface</b>	vytvorenie povrchu grafického obrazca
<b>waterfall</b>	vlnové zobrazenie (priestorové)

## Možnosti nastavenia zobrazenia

Funkcia	Opis
<b>axis</b>	nastavenie typu osí
<b>caxis</b>	nastavenie rozsahu farieb z aktuálnej farebnej palety
<b>colormap</b>	nastavenie farebnej palety
<b>hidden</b>	skrytie siete 3-D zobrazenia
<b>shading</b>	nastavenie módu tieňovania
<b>brighten</b>	vyjasňovanie alebo stmavovanie farebnej palety
<b>lighting</b>	nastavenie módu osvetlenia
<b>rotate3d</b>	zapnutie/vypnutie možnosti natáčania obrazca
<b>view</b>	špecifikácia bodov zobrazenia v 3-D grafe
<b>viewmtx</b>	transformácia matic pre zobrazenie

### Príklad 8.38: Zobrazenie povrchu obrazca tieňovaním

```
>> z = peaks(25);           % vytvorí maticu 25x25
>> surf(z)                 % zobrazenie povrchu zatienením
>> shading flat           % tieňovanie 'flat'
>> shading interp         % tieňovanie 'interpolated'
>> shading faceted       % tieňovanie 'faceted' (pôvodné)
>> colormap(copper)      % zmena farby (pink)
>> surfc(z)              % kombinácia zobrazení surf/contour
```

### Príklad 8.39: Zobrazenie povrchu obrazca sieťou

```
>> z = peaks(25);           % vytvorí maticu 25x25
>> mesh(z)                % zobrazenie povrchu sieťou
>> colormap(cool)        % zmena farby (cool)
>> colormap(hot)         % zmena farby (hot)
>> colormap(gray)        % zmena farby (gray)
>> meshc(z)              % kombinácia zobrazení mesh/contour
>> meshz(z)              % zobrazenie mesh s nulovou rovinou
```

### Príklad 8.40: Zobrazenie povrchu obrazca vlnami

```
>> z = peaks(25);           % vytvorí maticu 25x25
>> waterfall(z)          % zobrazenie povrchu vlnami
>> colormap(pink)        % zmena farby (pink)
```

## 8.7 Práca s farebnými obrazmi

Funkcia	Opis
<b>image</b>	zobrazenie obrazu (vytvorenie objektu obrazu)
<b>imagesc</b>	mierkovanie dát a zobrazenie obrazu
<b>imread</b>	načítanie obrazu z grafického súboru
<b>imwrite</b>	uloženie obrazu do grafického súboru
<b>imfinfo</b>	získanie informácie o obraze z grafického súboru
<b>axis</b>	nastavenie mierky a vzhľadu osí pri zobrazení

### Typy obrazov

Farebný obraz je v MATLABe tvorený maticou dát a maticou palety farieb.

Zápis zobrazenia	Typ obrazu
<b>image(X); colormap(map)</b>	indexový
<b>image(I,[0,1]); colormap(gray)</b>	intenzívny
<b>image(RGB)</b>	trojfarebný

#### Príklad 8.41: Zobrazenie obrazu z dát uložených v earth.mat

```
>> load earth; % Načítanie matice dát X, matice farebnej palety map
>> image(X) % Zobrazenie obrazu
>> colormap(map);
>> axis image % Nastavenie osí v pomere 1:1
```

#### Príklad 8.42: Vygenerovanie náhodného farebného obrazca

```
>> X = magic(6); % Vytvorenie matice X, matice farebnej palety map
>> image(X) % Zobrazenie farebného obrazu
```

### Načítanie a uloženie grafického súboru

MATLAB umožňuje načítanie a uloženie dát farebného obrazu v niekoľkých grafických súboroch a nasledovných formátoch: BMP, HDF, JPEG(JPG), PCX, TIFF, XWD.

#### Príklad 8.43: Načítanie, zobrazenie a uloženie obrazu

```
[X,MAP] = imread('penguin','bmp');
image(X);
axis image;
imwrite(X,'penguin.jpg') % Uloženie farebného obrazu
```