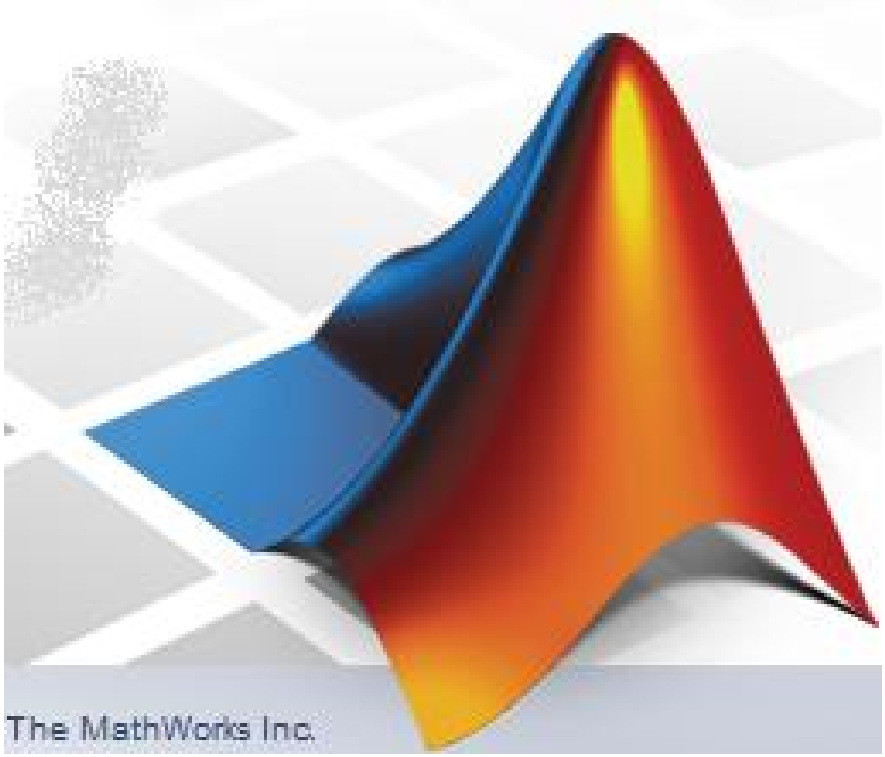
MATLAB (1) - úvod do programovania vedeckých problémov



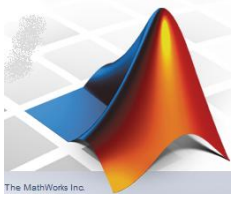
LS 2017, 4.predn.

MATLAB (1) - úvod do programovania vedeckých problémov, LS 2017

Program predmetu:

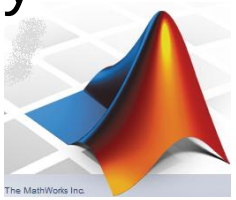
1. týždeň: úvod, základné info o Matlabe, pracovné prostredie Matlabu, interaktívny režim, prvé info o písaní skriptov
2. týždeň: základné operácie s maticami, import a export dát, základné grafické zobrazovanie (grafy a mapy)
3. týždeň: práca s reťazcami, práca so súbormi
4. týždeň: pokročilejšia grafika - popis grafov a máp, 2D grafy
5. týždeň: príkazy, stavba programov, M-súborov
6. týždeň: funkcie – zabudované v Matlabe, tvorba vlastných funkcií
7. týždeň: príklady programovania úloh z oblasti prírodných vied
8. týždeň: príklady programovania úloh z oblasti prírodných vied
9. týždeň: tvorba vlastných aplikácií, práca s GUI (Graphical User Interface)
10. týždeň: tvorba vlastných aplikácií, nástroj GUIDE

pozn.: zmeny vyhradené



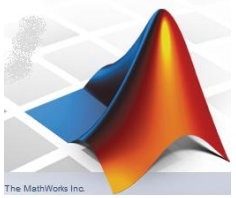
doteraz:

- štruktúra matíc: A(riadok,stlpec)
- pracovná plocha (najdôl. časť: *Command Window*)
- základné pracovné príkazy: `clear;clc;` ...
- význam znakov `% ; : ...`
- grafické príkazy: `contour(), plot(), subplot(),` ...
- generovanie matíc, aritm. operácie s maticami
- práca s reťazcami: príkazy `strtok(), str2double()`
- načítanie ASCII súborov: príkaz `dlmread(), importdata(), fscanf(),fprintf(), fopen(), fclose(), xlsread()`
- komunikácia cez grafické prosredie Windows: príkazy `uigetfile(), msgbox()`



Obsah (4. prednáška):

- základné poznatky o práci s grafikou
- pokročilejšie grafické zobrazovanie (1D a 2D grafy)
- načítavanie obrázkov (rastrov) zo súborov typu *.jpg, *.png, *.tif - funkcia `imread()`



Základné poznatky o práci s grafikou v prostredí Matlab:

aktívne okno pri kreslení

(okno, do ktorého Matlab realizuje aktuálny grafický príkaz):

- okno, v ktorom bol realizovaný posledný grafický príkaz
- okno, určené príkazom `figure` alebo `figure(number)` (je možné ich takto otvoriť niekoľko za sebou)
- okno, na ktoré užívateľ klikol myšou

Pozn.: Pred začiatkom nového výpočtu je dobré si staré otvorené okná zatvoriť príkazom `close all;`
Cielene vieme pozatvárať okná s daným poradovým číslom – napr. `close 3;`

práca s grafikou – 1D grafy

1D grafika:

Základný príkaz `plot()`:
Okrem zadania zobrazovaných matíc je možné nastaviť v uvedenom reťazci druh čiary grafu, jeho farbu a znak pre zobrazenú hodnotu:

```
plot(x,y,'color_linestyle_marker')
```

Presné „hodnoty“ týchto troch parametrov sú v tabuľkách
na nasledujúcich 2 snímkoch a niektoré z nich v týchto príkladoch.

príklad:

```
close all; clear all; clc;
x = -pi:.1:pi;
y = cos(x);
plot(x,y); %standardny graf
figure, plot(x,y,'y:square');
  %y je pre yellow, : je bodkovana ciara, square je storcek
figure, plot(x,y,'--rs');
  %-- je prerusovana ciara, r je cervena, s je storcek
figure, plot(x,y,'xg-');
  %x je krizik, g je zelena, - je plna ciara
%poradie tychto parametrov moze byt poprehadzovane!
```

1D grafika – tzv. LineSpec:

Konkrétne hodnoty pre parametre `'color_linestyle_marker'`

Symbol	Color (RGB)	Symbol	Line Style
c	cyan (0 1 1)	–	solid line (default)
m	magenta (1 0 1)	- -	dashed line
y	yellow (1 1 0)	:	dotted line
r	red (1 0 0)	–.	dash-dot line
g	green (0 1 0)	none	no line
b	blue (0 0 1)		
w	white (1 1 1)	–	–
k	black (0 0 0)	–	–

Konkrétne hodnoty pre parametre `'color_linestyle_marker'`

Marker Specifier	Description
+	plus sign
o	circle
*	asterisk
.	point
x	cross
square	square
diamond	diamond
^	upward pointing triangle
v	downward pointing triangle
>	right pointing triangle
<	left pointing triangle
pentagram	five-pointed star
hexagram	six-pointed star
none	no marker (default)

medzera alebo nič

Úloha: skúste zobraziť funkciu cos() z predchádzajúceho príkladu – – modrou farbou, plnou čiarou a s krúžkami.

Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line

Specifier	Marker Type
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
'square' or s	Square
'diamond' or d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
'pentagram' or p	Five-pointed star (pentagram)

1D grafika:

Základný príkaz `plot()`:
Ďalej je možné nastaviť pomocou ďalších reťazcov hrúbku čiary, veľkosť znaku a podobne – `LineWidth`, `MarkerEdgeColor`, `MarkerFaceColor`, `MarkerSize`.

príklad:

```
close all; clear all; clc;
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
%nastavenie roznych parametrov ciary a znaku
 plot(x,y,'--rs','LineWidth',2, ...
                 'MarkerEdgeColor','k',...
                 'MarkerFaceColor','g',...
                 'MarkerSize',5);
 figure, plot(x,y,'-mo','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor',[.49 1 .63],...
                'MarkerSize',9);
```

1D grafika:

Pozn.: Zmeny všetkých týchto vlastností grafických objektov je možné realizovať pomocou príkazu `set()` na zmenu nastavenia vlastností objektov. V takom prípade ale musí byť napr. príkaz `plot()` priradený určitej matici.

príklad:

```
close all; clear all; clc;
x = 0:.2:12;
y1 = bessel(1,x); %specialne funkcie
y2 = bessel(2,x);
y3 = bessel(3,x);

plot(x,y1,x,y2,x,y3); %kreslenie standardne

figure(2)
h = plot(x,y1,x,y2,x,y3); %individualne nastavenie parametrov

set(h,'LineWidth',2,{'LineStyle'},{'--';':';'-.'});
set(h,{'Color'},{'r';'g';'b'});
```

1D grafika:

Popis osí a grafov:

```
axis([minx maxx miny maxy]);   %definovanie rozsahov osi
xlabel('popis','FontSize',size);   %popis osi x
ylabel('popis','FontSize',size);   %popis osi y
legend('popis');   %popis ciary grafu
title('popis');    %popis celeho obrazku (nazov)
grid on;    %polozenie siete
text(suradnice umiestnenia,'popis'); %umiestni text
gtext('popis'); %umiestni text na miesto po kliku mysou
```

Tieto príkazy je možné realizovať aj dodatočne (už po nakreslení samotného grafu) – sú doplňované v aktívnom okne.

Pozn.: Pri úprave rozsahov osí existuje viacero možností (viď. Help), napr.:

```
axis equal; %nastavi rovnake rozsahy
```

alebo

```
axis square; %nastavi rovnake dlzky
```

Pozn.: Pri úprave rozsahov osí existuje viacero možností (viď. Help), napr.:

```
axis equal; %nastavi rovnake rozsahy
```

alebo

```
axis square; %nastavi rovnake dlzky
```

MATLAB Function Reference

axis

Axis scaling and appearance

Syntax

```
axis([xmin xmax ymin ymax])
axis([xmin xmax ymin ymax zmin zmax cmin cmax])
v = axis
axis auto
axis manual
axis tight
axis fill
axis ij
axis xy
axis equal
axis image
axis square
axis vis3d
axis normal
axis off
axis on
```

ukážka z Helpu

Príklad:

```
close all; clear all; clc;
t = 0:pi/20:2*pi;
plot(sin(t),2*cos(t));
axis square;
```

práca s grafikou

1D grafika:

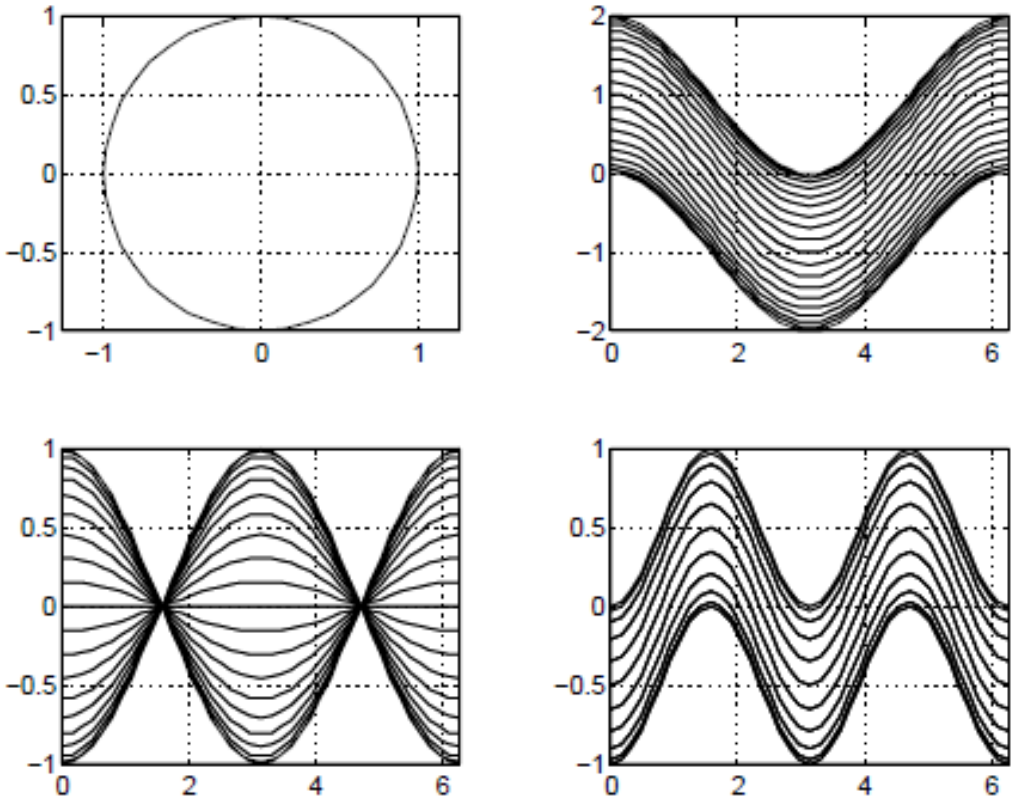
Popis osí a grafov:

príklad.:

```
close all; clear all; clc;
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
%nastavenie roznych parametrov ciary a znaku
plot(x,y,'--rs','LineWidth',2, ...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',5);
axis([min(x) max(x) min(y) max(y)]);
xlabel('x [m]','FontSize',10); ylabel('z[m]','FontSize',10);
legend('graf');
grid on;
set(gca,'XTick',-pi:pi/2:pi); %umiestni ‚ticks‘ s krokom pi/2 od –pi po +pi
set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'}); %popise ‚ticks‘
gtext('manual. inflex.');
```

príklad (popis obrázkov):

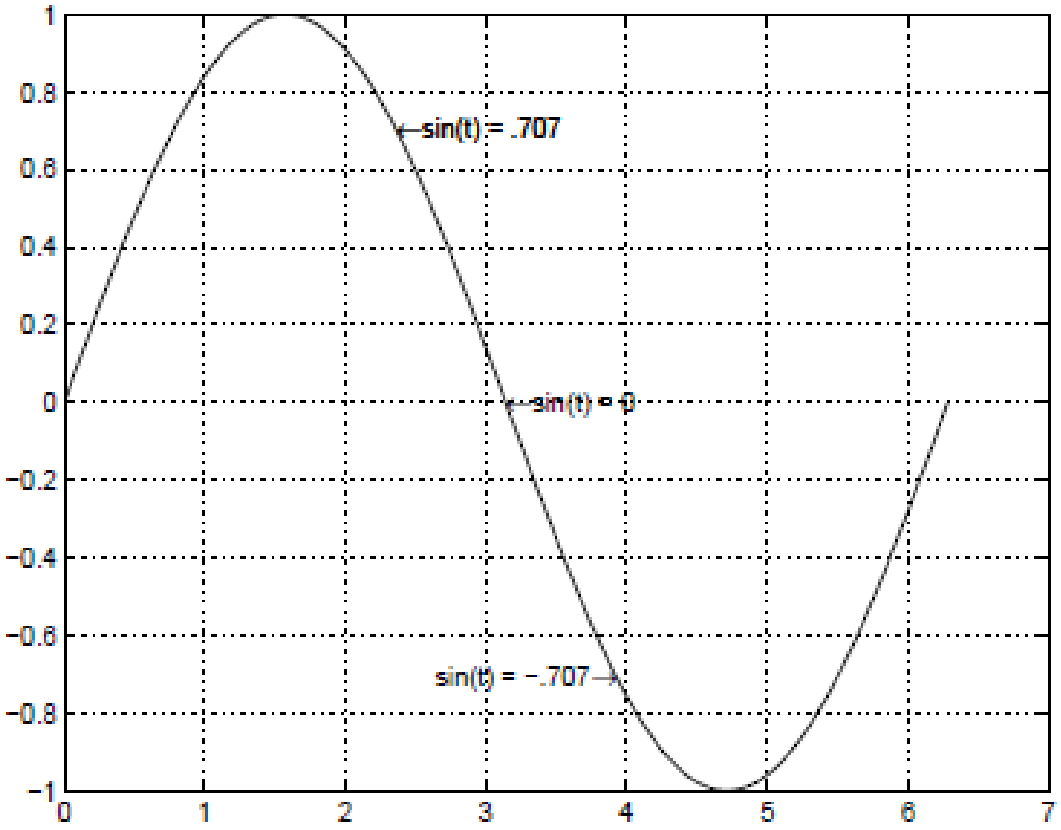
```
close all; clear all; clc;
t = 0:pi/20:2*pi;
[x,y] = meshgrid(t);

subplot(2,2,1)
plot(sin(t),cos(t))
axis equal

subplot(2,2,2)
z = sin(x)+cos(y);
plot(t,z)
axis([0 2*pi -2 2])

subplot(2,2,3)
z = sin(x).*cos(y);
plot(t,z)
axis([0 2*pi -1 1])

subplot(2,2,4)
z = (sin(x).^2)-(cos(y).^2);
plot(t,z)
axis([0 2*pi -1 1])
```



Pozn.: Funkcia `meshgrid()` generuje matice vhodné pre 2D grafy.

príklad (umiestnenie textu do obrázku):

```
close all; clear all; clc;
x = 0:0.1:2*pi;
y=sin(x);
plot(x,y);
text(3*pi/4,sin(3*pi/4),'\leftarrowsin(t) = .707');
text(pi,sin(pi),'\leftarrowsin(t) = 0');
text(5*pi/4,sin(5*pi/4),'sin(t) = -.707\rightarrow','HorizontalAlignment','right');
```



1D grafika – sumarizácia grafov:

Function	Used to Create
`plot`	Graph with linear scales for both axes
`loglog`	Graph with logarithmic scales for both axes
`semilogx`	Graph with a logarithmic scale for the x-axis and a linear scale for the y-axis
`semilogy`	Graph with a logarithmic scale for the y-axis and a linear scale for the x-axis
`plotyy`	Graph with y-tick labels on the left and right side

základné typy funkcií (príkazov) pre grafy v prostredí Matlab

1D grafika – sumarizácia grafov:

```
plot(y) %klasický čiarový graf matice y voči porad. číslu
plot(x,y) %klasický čiarový graf matice y voči x
```

pozn.: x a y musia mať rovnaké rozmery!

```
loglog(x,y) %graf s osami v log10 mierkach
semilogy(x,y) %y-ová os v log10 mierke
semilogx(x,y) %x-ová os v log10 mierke
plotyy(x1,y1,x2,y2)
     %dva grafy v jednom s popisom osí naľavo a napravo
plot3(x,y,z) %čiarový graf v priestore
bar(x,y) %stĺpcový graf
stem(x,y) %graf izolovaných bodov s vertik. úsečkami
```

práca s grafikou – 1D grafy

`plot(y)` – klasicky graf (matica oproti jej porad. cislu elementu)

príklad:

```
close all; clear all; clc;
x  = 0:pi/100:2*pi;
y = sin(x);
plot(y);
```

`plot(x,y)` – klasicky graf (matica y voci matici x)

príklad:

```
close all; clear all; clc;
x  = 0:pi/100:2*pi;
y = sin(x);
plot(x,y);
```

`plot(x,y1,x,y2)` – klasicky graf (viacej matic voci x, autom. zmena farieb)

príklad:

```
close all; clear all; clc;
x  = 0:pi/100:2*pi;
y = sin(x); y2 = sin(x-.25); y3 = sin(x-.5);
plot(x,y,x,y2,x,y3);
%alebo
figure, plot(x,y,'-',x,y2,'--',x,y3,':') ;
```

práca s grafikou – 1D grafy

`loglog(x,y)` – graf s osami v $\log_{10}$ mierkach

príklad:

```
close all; clear all; clc;
x = logspace(-1,2); %vytvorí x v log10 mierke od -1 do 2
loglog(x,exp(x));
grid on;
```

`semilogy(x,y)` – y-ová os v $\log_{10}$ mierke

príklad:

```
close all; clear all; clc;
x = 0:.1:10;
semilogy(x,10.^x);
grid on;
```

`semilogx(x,y)` – x-ová os v $\log_{10}$ mierke

podobne...

práca s grafikou – 1D grafy

`plot3(x,y,z)` – čiarový graf v priestore

príklad - špirála:

```
close all; clear all; clc;
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t);
grid on;
axis square; %bez tohto príkazu by zostala vertikálna os sploštená
```

`bar(x,y)` a `barh(x,y)` – stĺpcový graf (tiež `bar3()` a `bar3h()`)

príklad:

```
close all; clear all; clc;
x = -2.9:0.2:2.9;
bar(x,exp(-x.*x),'r');
figure, barh(x,exp(-x.*x),'r');
```

`stem(x,y)` – graf izol. bodov s vertik. useckami (napr. diskretne spektra)

príklad:

```
close all; clear all; clc;
t = 0:pi/30:pi/2;  y = sin(2*t) + cos(5*t);
stem(t,y);
```

práca s grafikou – 2D grafy

2D grafika:

príkaz `contour()`:

```
contour(Z)
contour(Z,n)
contour(Z,v)
contour(X,Y,Z)
contour(X,Y,Z,n)
contour(X,Y,Z,v)
contour(...,LineSpec)
contour(ax,...)
```

a jemu podobné príkazy:

```
contourf(Z); %vyplnene izociary
contour3(Z); %izociary v 3D pohlade
contourslice(Z) %izociary v rezoch v priestore
mesh(Z); %priestorova “dratena plocha”
surf(Z); %priestorova plocha (plosky su vyplnene farbou)
image(Z); %farebna mapa, POZOR!!! Rozdielne definovany zaciatok
          %zaciatok (vlavo hore); treba pouzit jeden
          %z preklapacich prikazov – flipud()
```

2D grafika:

Príkazy `contourf()` vs `image()`

Príklad:

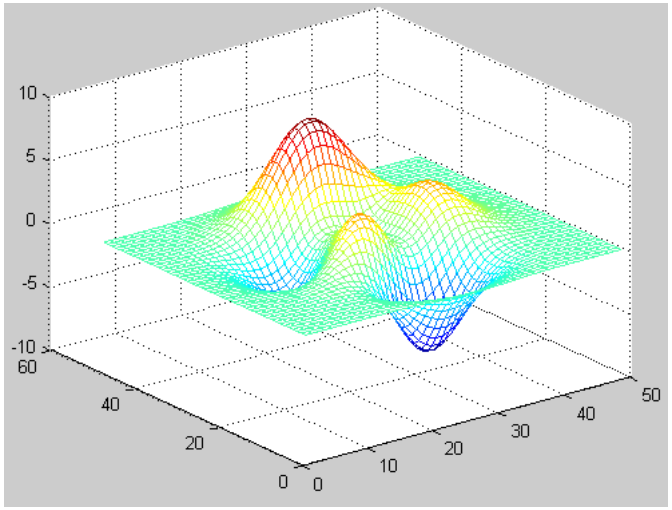
```
close all; clear all; clc;
Z = peaks; %pomocna funkcia Matlab
%plotting as contourf()
contourf(Z);
%plotting as image()
figure, image(Z);
%correcting of the image plot
Zud=flipud(Z);
figure, image(Zud);
```

Pozn.: Funkcia `image( )` je vhodná najmä na zobrazovanie obrázkov a snímok (súbory typu JPG, PNG, TIF, atď.).
Dostaneme sa k tomu ku koncu tejto prednášky.

práca s grafikou – 2D grafy

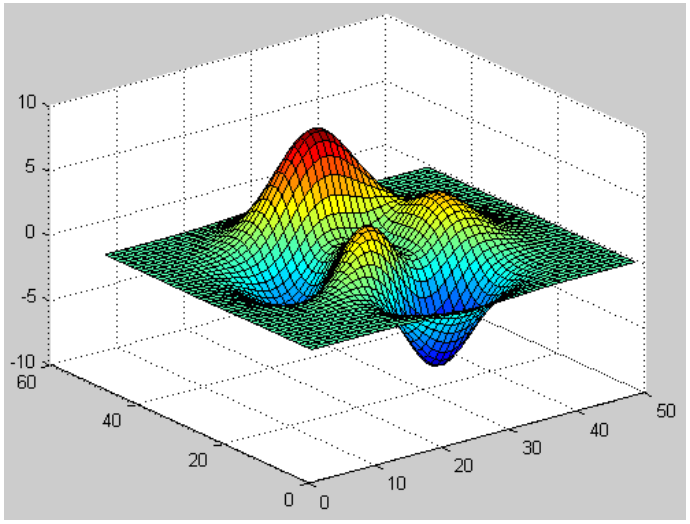
2D grafika:

Príkazy `mesh()` vs `surf()`

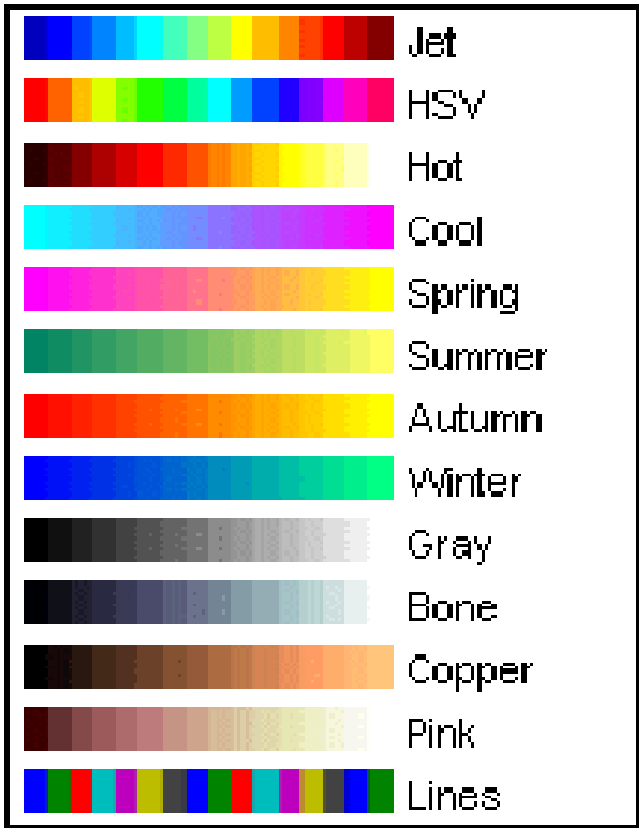


Príklad:

```
close all; clear all; clc;
Z = peaks; %pomocna funkcia Matlab
%plotting as mesh()
mesh(Z);

%plotting as surf()
figure, surf(Z);
```



2D grafika:



Ďalšie príkazy a úpravy obrázkov:

```
colormap()
colormap('default')
colorbar
colorbar() %vid Help
```

Príklad:

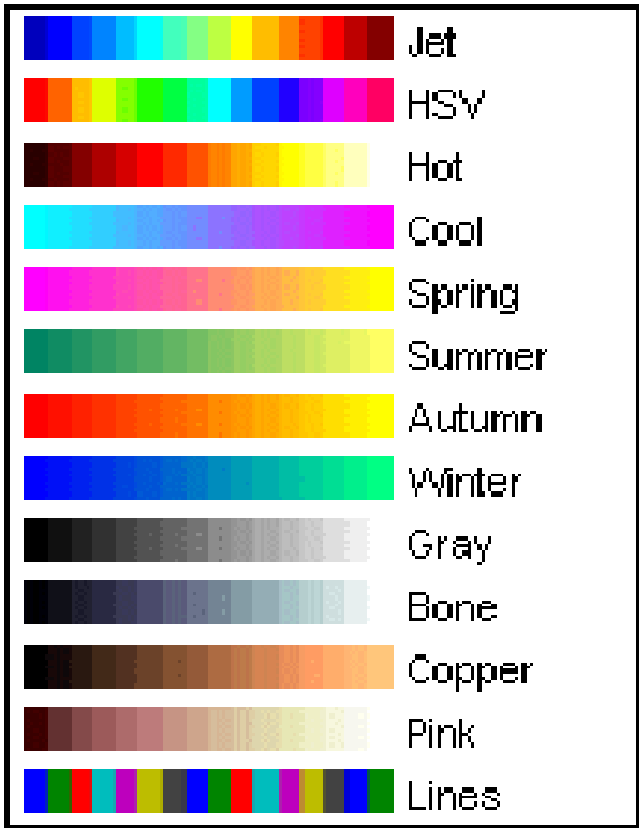
```
close all; clear all; clc;
Z = peaks;
contourf(Z);
legend('peaks');
colormap(hot);  %alebo colormap hot;
colorbar;
figure, contourf(Z);
legend('peaks');
colormap('default');
colorbar;
```

2D grafika:

Príklady (Matlab Help):

```
close all; clear all; clc;
load flujet
image(X)
colormap(jet)
figure, image(X)
colormap(hot)
```

```
close all; clear all; clc;
load spine
image(X)
colormap bone
figure, image(X)
colormap(HSV)
```



2D grafika:

Ďalšie príkazy a úpravy obrázkov:

```
light()
lighting
shading
```

Príklad:

```
close all; clear all; clc;
Z = peaks(60);
surf(Z);
colormap hot;
light('Position',[-2,2,20])
shading interp; lighting phong;
figure, surf(Z);
colormap hot;
light('Position',[20,-20,20])
shading interp; lighting flat;
```

MATLAB Function Reference

lighting

Specify lighting algorithm

Syntax

```
lighting flat
lighting gouraud
lighting phong
lighting none
```

MATLAB Function Reference

shading

Set color shading properties

Syntax

```
shading flat
shading faceted
shading interp
shading(axes_handle,...)
```

2D grafika:

príkaz *contour()*:

špecialitka - popis izočiar:
príkaz `set()` – nastavenie grafických parametrov
príklad:

```
clear all; close all; clc;
[X,Y] = meshgrid(-2:.2:2,-2:.2:3);
Z = X.*exp(-X.^2-Y.^2);
[C,h] = contour(X,Y,Z); %returns contour matrix C and a handle h
set(h,'ShowText','on','TextStep',get(h,'LevelStep')*2);
```

práca s grafikou – doplnky

Pripomienka príkazu `hold` (prednáška č.2, snímka č.26).
Používame, keď chceme niečo dokresliť do už existujúceho obrázku so zachovaním pôvodných grafických nastavení.

MATLAB Function Reference

hold

Retain current graph in figure

Syntax

```
hold on
hold off
hold all
hold
hold(axes_handle,...)
```

Description

The `hold` function determines whether new graphics objects are added to the graph or replace

```
close all; clear all; clc;
t = 0:pi/30:2*pi;  y1 = sin(t); y2 = cos(t);
plot(t,y1); hold; bar(t,y2); % ciarovy a stlpcovy graf
```

práca s grafikou – 2D grafy

2D grafika:

Zobrazovanie grafických súborov -
príkaz `image()`
v kombinácii s funkciou `imread()`

MATLAB Function Reference

imread

Read image from graphics file

Synopsis

```
A = imread(filename, fmt)
```

Vyskúšajte si skript *nacitanie_jpg.m*.

```
close all; clear all; clc;
[fname,pname] = uigetfile('*.jpg','input of data (image import');
shot = imread(fname);
image(shot);
```

Zadanie č.3:

Vyskúšajte si funkcie `image()`v kombinácii s `imread()` na vašich ľubovoľných súboroch – ktoré si sami vyberiete a zobrazte niekoľko z nich vo forme obrázku `subplot()` (povedzme v štruktúre 2x2).
Môžete vyskúšať aj takú verziu funkcie `imread()`, ktorá načíta grafický súbor priamo z udanej http adresy (viď help).