

Seminář 3

Vizualizace dat

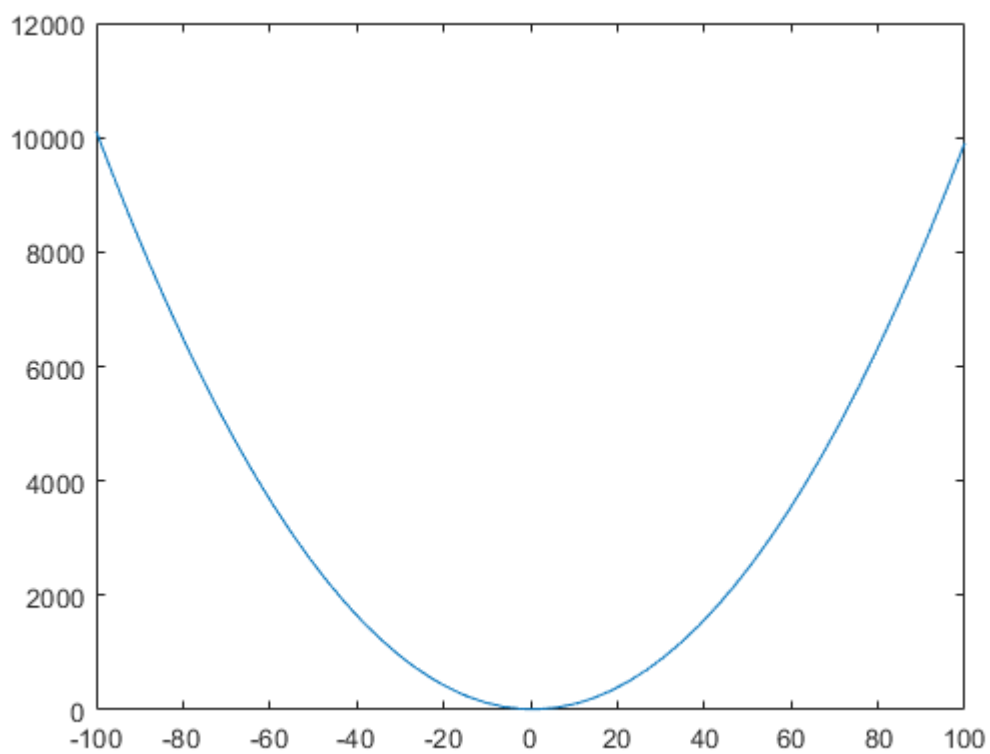
Dvourozměrná data

Vytvoříme ukázková data

```
x = linspace(-100, 100, 50);  
y = x.^2 - x + 10;
```

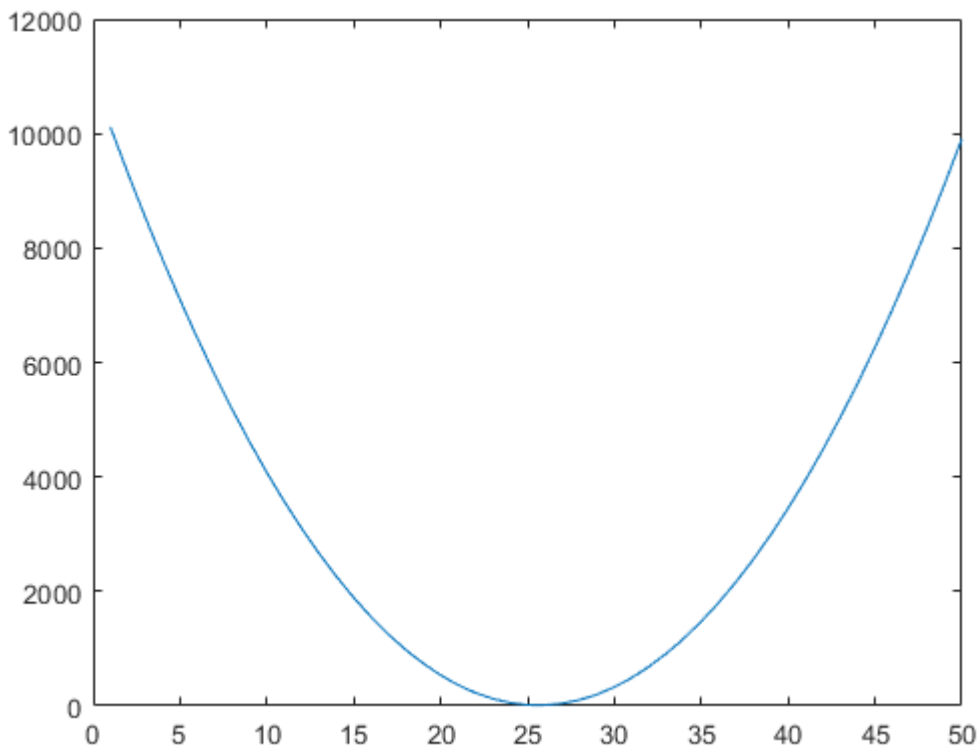
Nejjednodušší vykreslení pomocí plot()

```
plot(x, y);
```



Pokud zadáme jen jeden vektor hodnot, x je bráno od 1 do počtu hodnot.

```
figure, plot(y);
```



Vzhled vykreslení je možné ovlivnit dalšími parametry.

help `plot`

plot Linear plot.

plot(X,Y) plots vector Y versus vector X. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up. If X is a scalar and Y is a vector, disconnected line objects are created and plotted as discrete points vertically at X.

plot(Y) plots the columns of Y versus their index. If Y is complex, **plot(Y)** is equivalent to **plot(real(Y),imag(Y))**. In all other uses of **plot**, the imaginary part is ignored.

Various line types, plot symbols and colors may be obtained with **plot(X,Y,S)** where S is a character string made from one element from any or all the following 3 columns:

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		

h hexagram

For example, `plot(X,Y,'c+:')` plots a cyan dotted line with a plus at each data point; `plot(X,Y,'bd')` plots blue diamond at each data point but does not draw any line.

`plot(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3,...)` combines the plots defined by the (X,Y,S) triples, where the X's and Y's are vectors or matrices and the S's are strings.

For example, `plot(X,Y,'y-',X,Y,'go')` plots the data twice, with a solid yellow line interpolating green circles at the data points.

The `plot` command, if no color is specified, makes automatic use of the colors specified by the axes `ColorOrder` property. By default, `plot` cycles through the colors in the `ColorOrder` property. For monochrome systems, `plot` cycles over the axes `LineStyleOrder` property.

Note that RGB colors in the `ColorOrder` property may differ from similarly-named colors in the (X,Y,S) triples. For example, the second axes `ColorOrder` property is medium green with RGB `[0 .5 0]`, while `plot(X,Y,'g')` plots a green line with RGB `[0 1 0]`.

If you do not specify a marker type, `plot` uses no marker. If you do not specify a line style, `plot` uses a solid line.

`plot(AX,...)` plots into the axes with handle AX.

`plot` returns a column vector of handles to lineseries objects, one handle per plotted line.

The X,Y pairs, or X,Y,S triples, can be followed by parameter/value pairs to specify additional properties of the lines. For example, `plot(X,Y,'LineWidth',2,'Color',[.6 0 0])` will create a plot with a dark red line width of 2 points.

Example

```
x = -pi:pi/10:pi;
y = tan(sin(x)) - sin(tan(x));
plot(x,y,'--rs','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
```

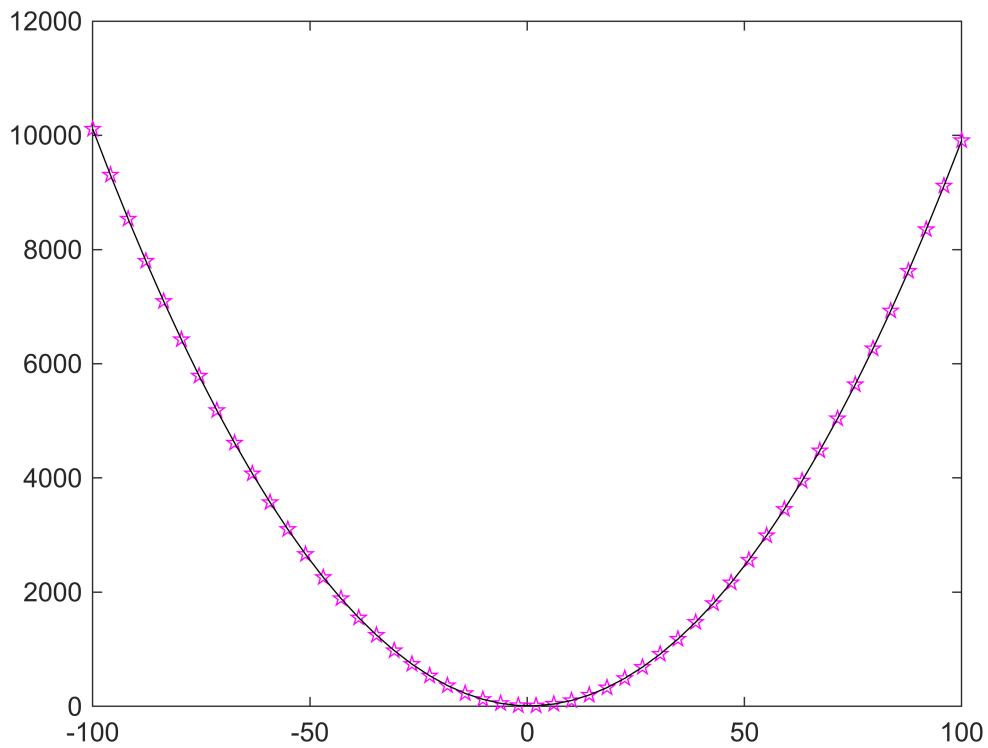
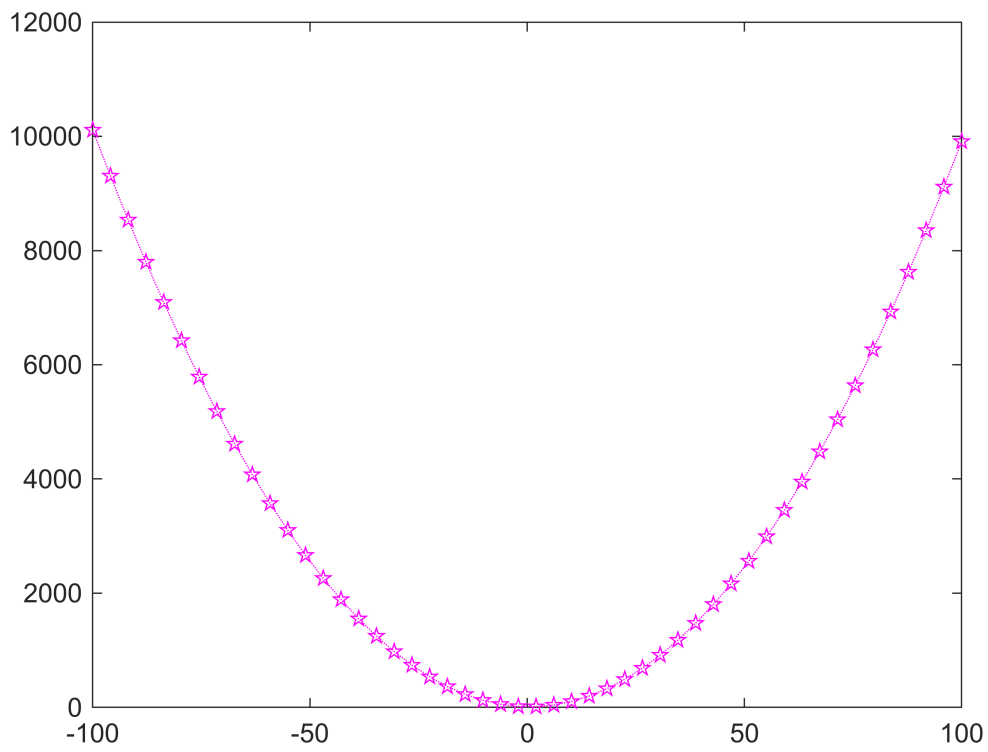
See also `plottools`, `semilogx`, `semilogy`, `loglog`, `plotyy`, `plot3`, `grid`, `title`, `xlabel`, `ylabel`, `axis`, `axes`, `hold`, `legend`, `subplot`, `scatter`.

Documentation for `plot`

Other functions named `plot`

Například

```
plot(x, y, 'mp:');
```

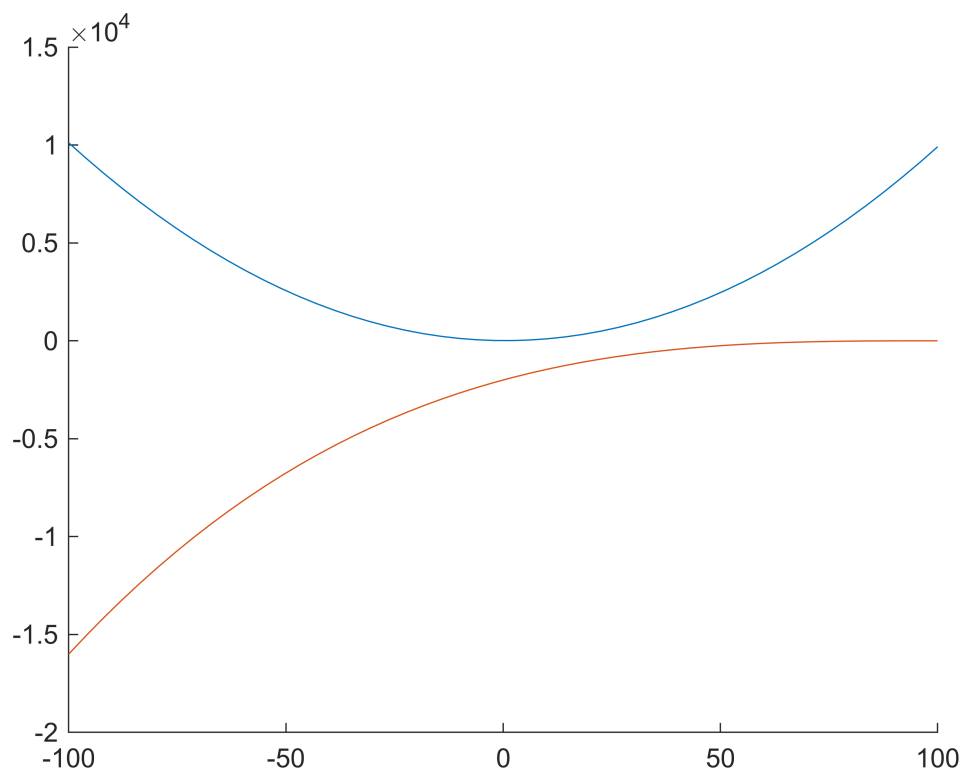


Vykreslete data červenou barvou, bez čáry se značkami ve tvaru trojúhelníku 2x zvětšenými.

Jak bychom vykreslili jen každou 5. hodnotu?

Grafy jsou automaticky kresleny do prostředí figure. Volání funkce plot automaticky způsobí refresh tohoto prostředí. Pro nakreslení více grafů do jednoho je zapotřebí použít `hold on` a `hold off`

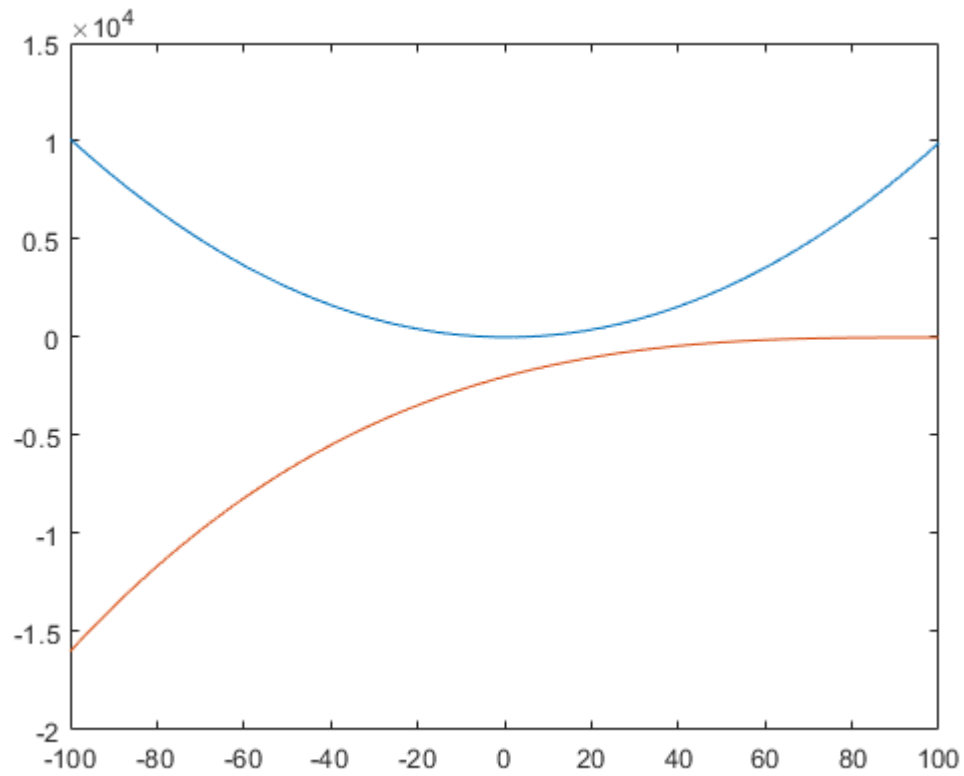
```
x = linspace(-100, 100, 50);  
y1 = x.^2 - x + 10;  
y2 = (x-100).^3./500;  
  
figure,  
hold on  
plot(x, y1);  
plot(x, y2);  
hold off
```



Případně je možné jako druhý parametr použít matici a v tom případě každý řádek představuje hodnoty jednoho grafu.

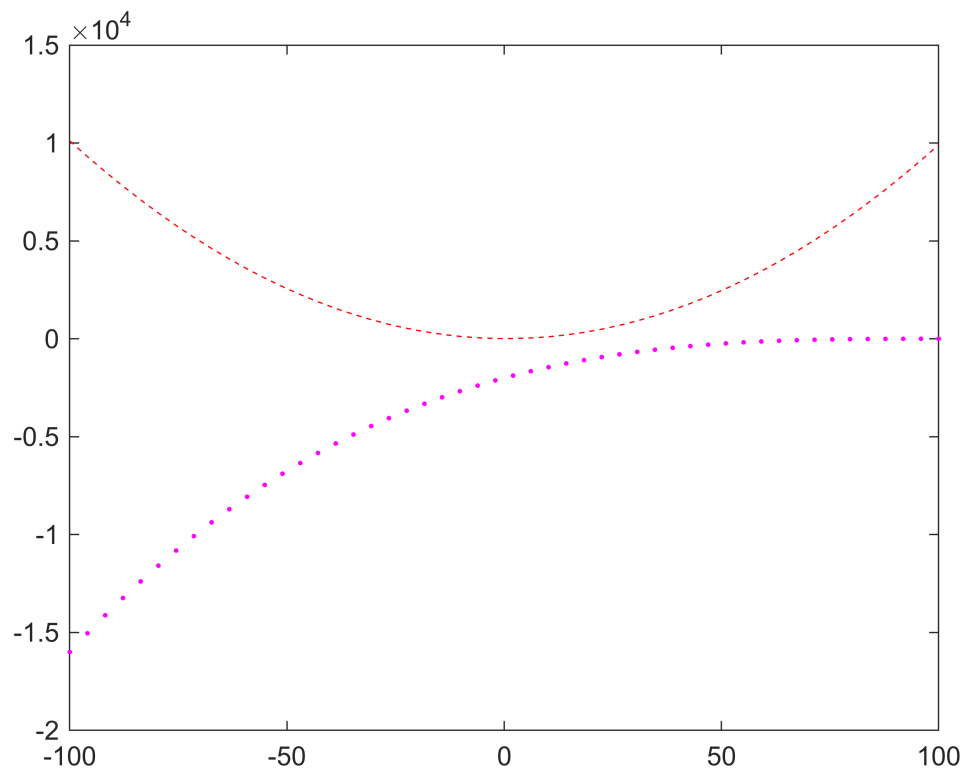
```
x = linspace(-100, 100, 50);  
y1 = x.^2 - x + 10;  
y2 = (x-100).^3./500;  
  
Y = [y1; y2];
```

```
figure,  
plot(x, Y);
```



Nebo všechna data vložit do jednoho `plot()`.

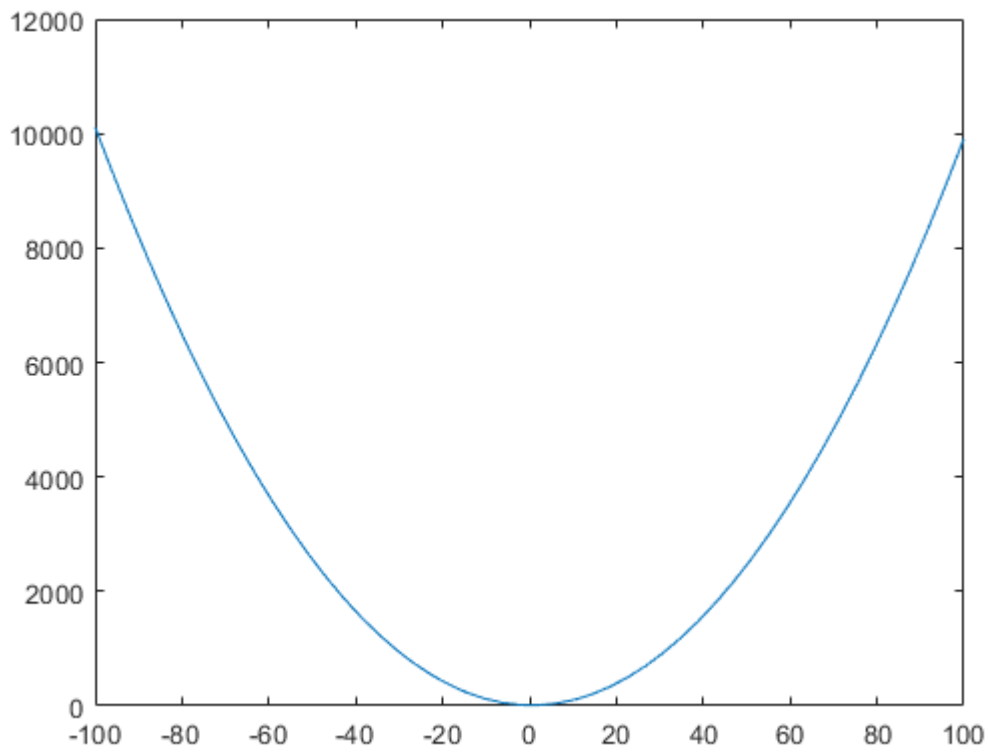
```
x = linspace(-100, 100, 50);  
y1 = x.^2 - x + 10;  
y2 = (x-100).^3./500;  
  
figure,  
plot(x, y1, 'r--', x, y2, 'm.');
```



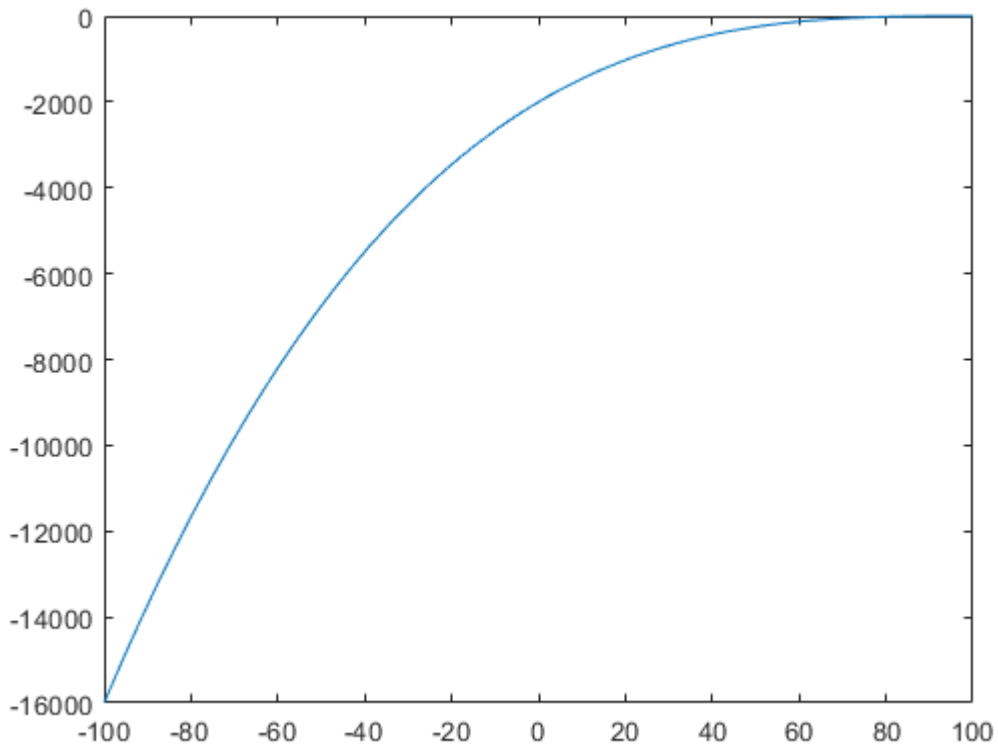
Oproti předchozímu přístupu můžeme každému grafu nastavit styl, jak se má vykreslit a dokonce data nemusí být stejně velká.

Nové okno pro vykreslování můžeme vytvořit pomocí figure.

```
plot(x, y1);
```



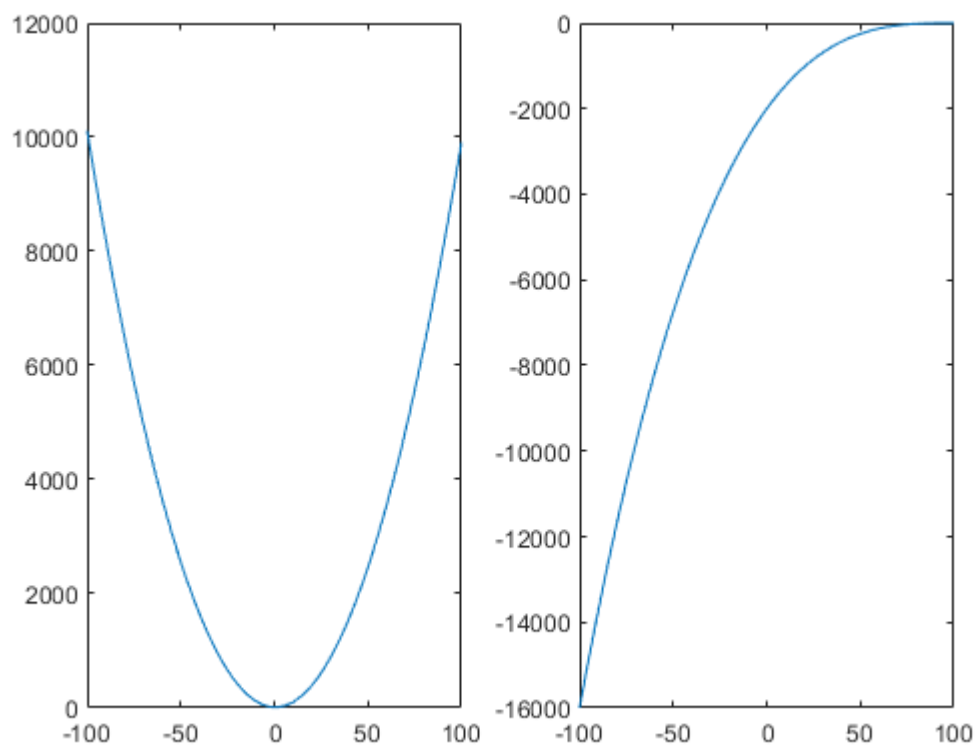
```
figure, plot(x, y2);
```



Čárka odděluje jednotlivé příkazy na jednom řádku.

Pro vykreslení více grafů (oken) v rámci jediného okna lze využít funkci `subplot(pocet_radku,pocet_sloupcu,index)`.

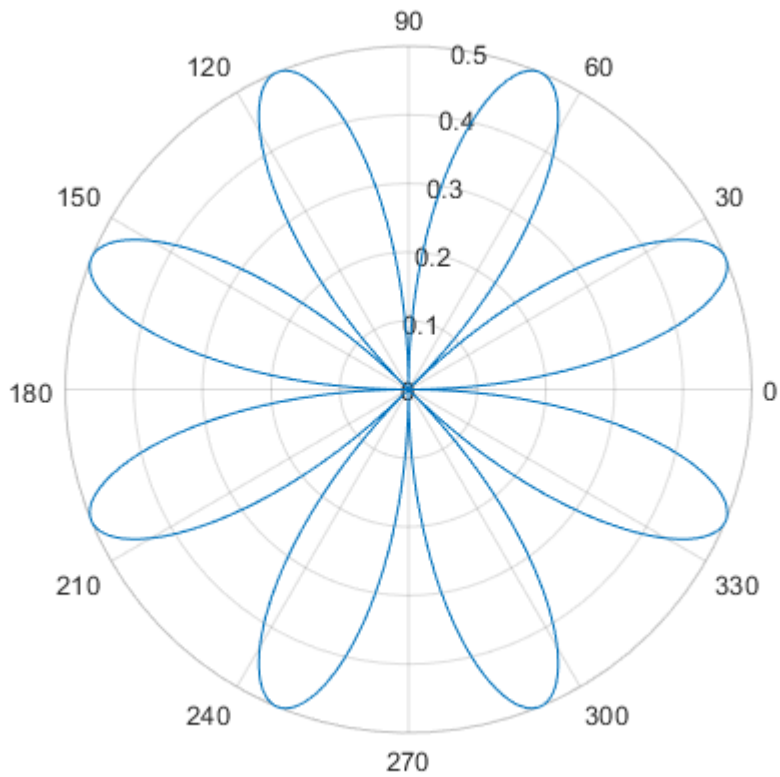
```
figure,  
subplot(1,2,1)  
plot(x, y1);  
subplot(1,2,2)  
plot(x, y2);
```



Subplots jsou indexovány postupně po řádcích.

Chceme-li funkci zobrazit v polárních souřadnicích, použijeme místo `plot` funkci `polarplot`.

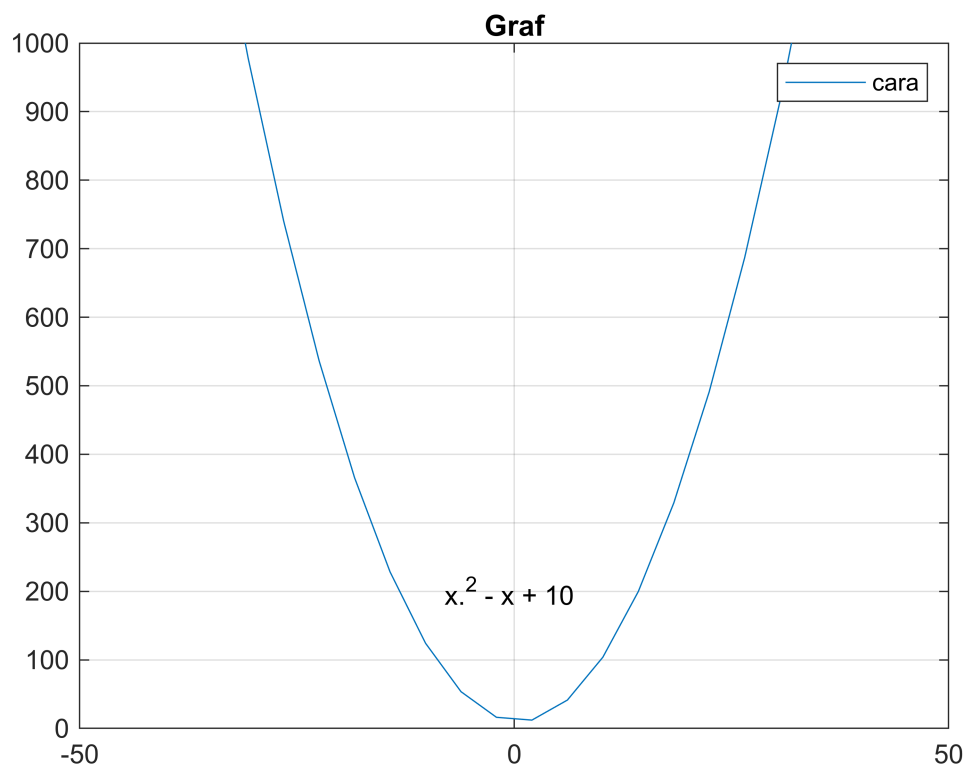
```
theta = 0:0.01:2*pi;  
rho = sin(2*theta).*cos(2*theta);  
figure, polarplot(theta,rho)
```



Nastavení parametrů

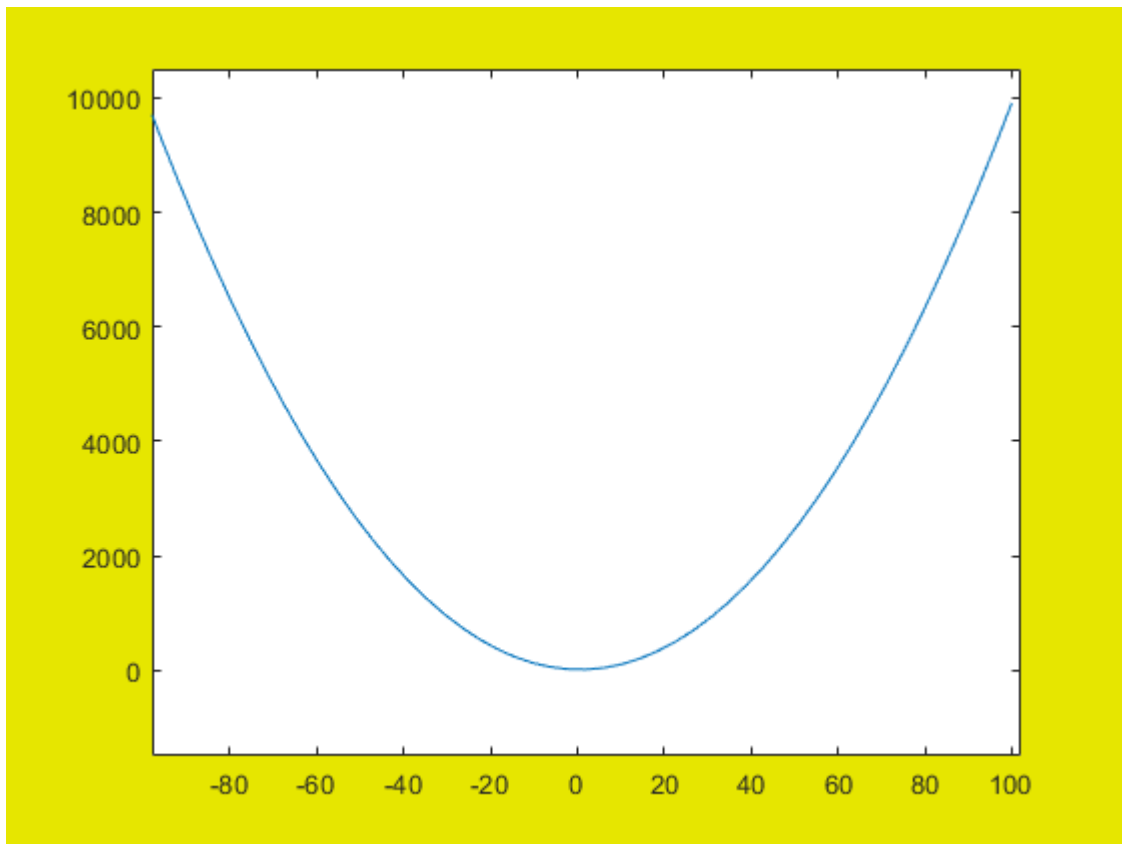
title, xlabel, ylabel, zlabel, text, legend, axis, grid, box (pro více informací použijte nápovědu)

```
figure
plot(x, y1);
text(-8,200, 'x.^2 - x + 10');
axis([-50, 50, 0, 1000]);
title('Graf');
legend('cara')
grid on;
```



K již vykresleným objektům je možné přistoupit (gcf, gca, gco, findobj)

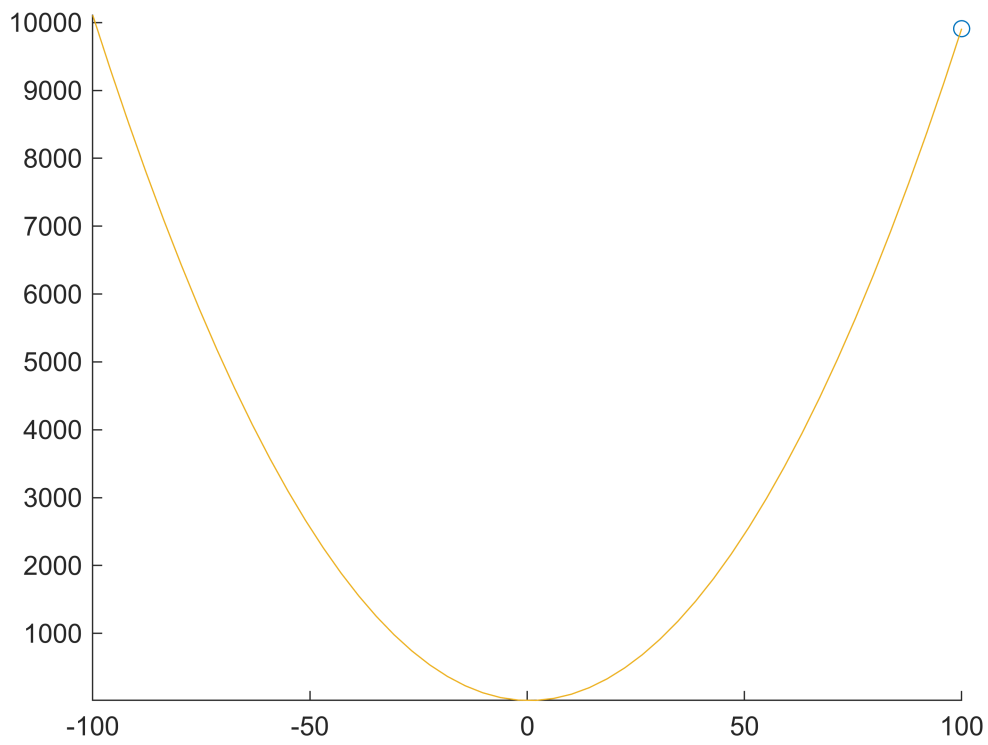
```
figure
plot(x, y1);
h = gcf;
h.Color = [0.9 0.9 0];
```



Animace

Vykreslení grafu můžeme animovat pomocí funkce `comet()`

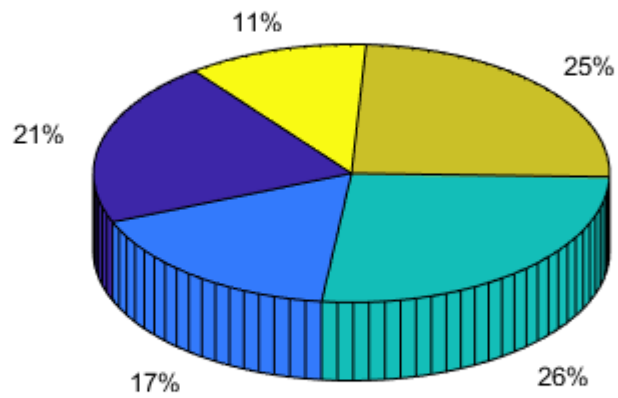
```
figure, comet(x, y1);
```



Grafy

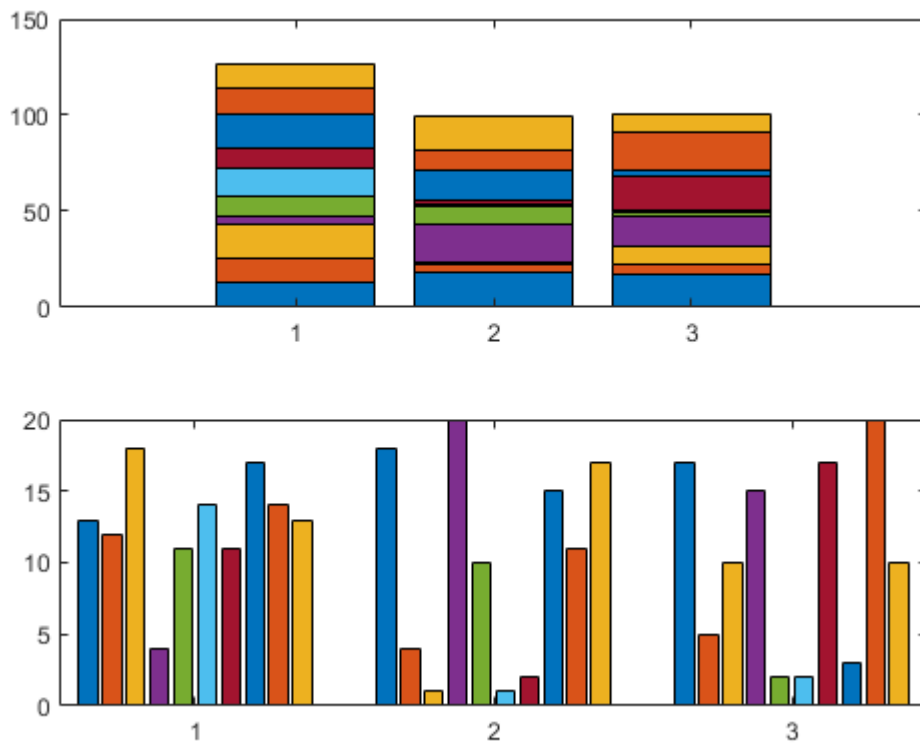
bar, barh, bar3, bar3h, area, pie, pie3, scatter

```
data = randi(20,[1,5]);  
figure, pie3(data)
```



Pokud máme pro všechny vzorky více hodnot, můžeme nastavit u sloupcových grafů `grouped` nebo `stacked`

```
data = randi(20,[3,10]);  
figure  
subplot(2,1,1)  
bar(data, 'stacked');  
subplot(2,1,2)  
bar(data, 'grouped');
```



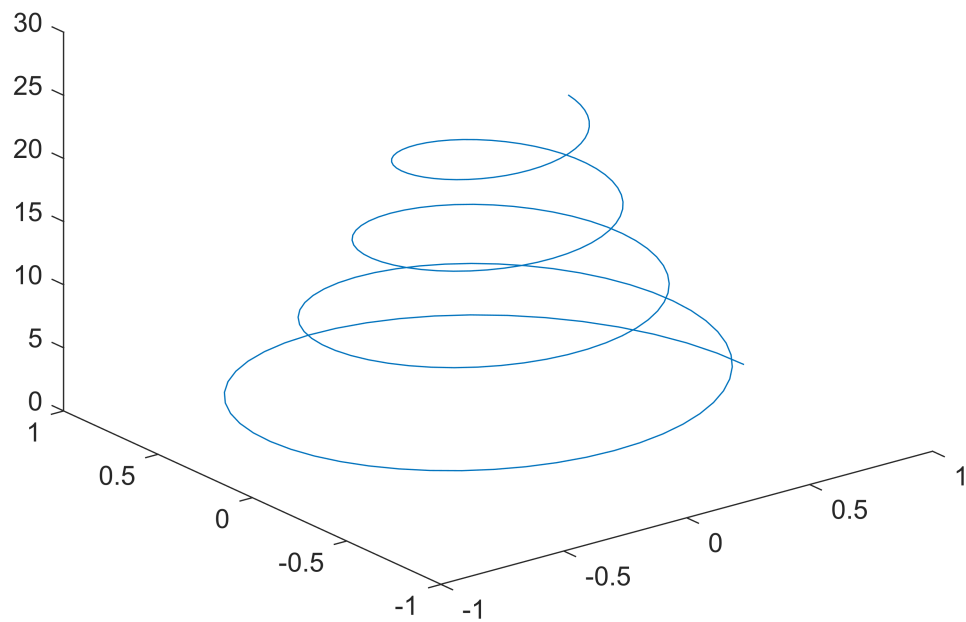
Trojrozměrná data

plot3, mesh, meshgrid, surf, contour3

Generování dat

```
z = linspace(0,8*pi, 200);
x = exp(-z/20).*cos(z);
y = exp(-z/20).*sin(z);
```

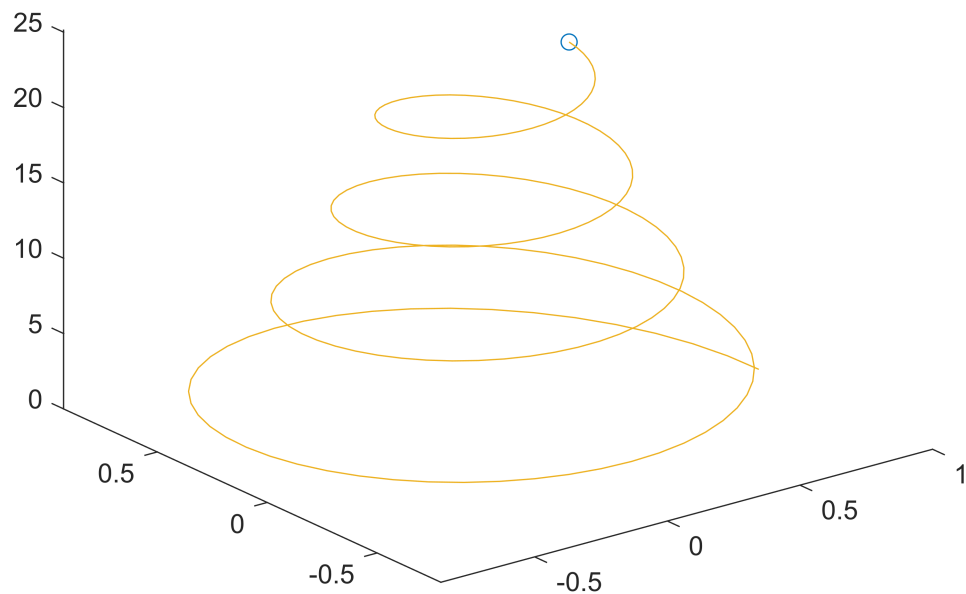
```
figure, plot3(x,y,z);
```



 Změna pohledu na graf. Případně pomocí funkce `view()`

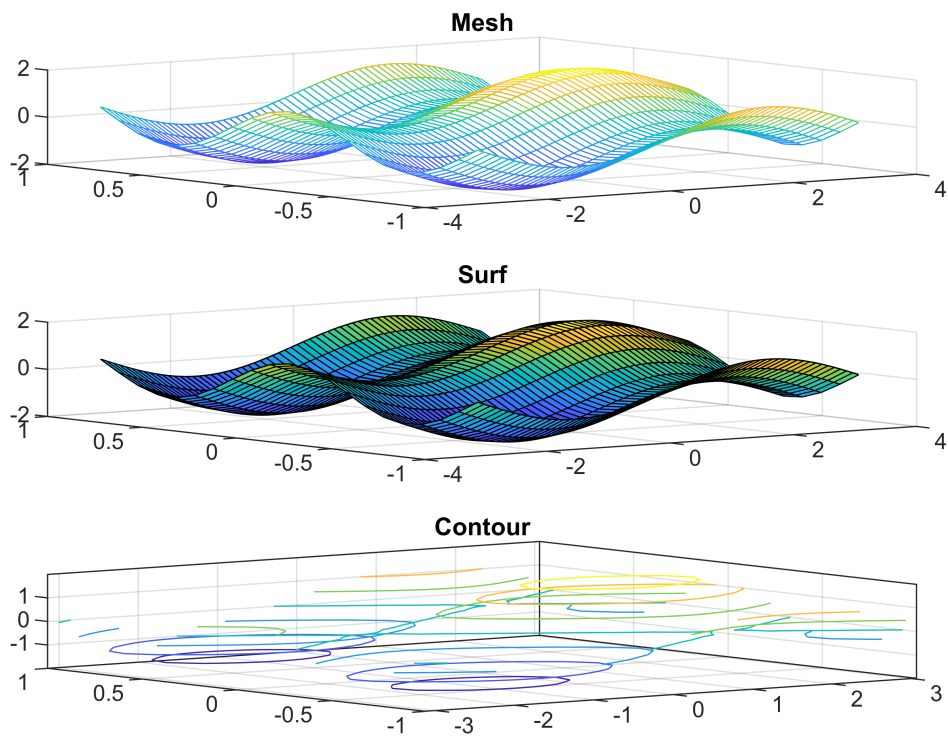
Animace

```
figure, comet3(x,y,z);
```

```
x = [-pi:0.1:pi];  
y = [-1:0.1:1];  
  
[X,Y] = meshgrid(x,y);  
Z = sin(X)+cos(5*Y);
```

```
figure  
subplot(3,1,1)  
mesh(X, Y, Z);  
title('Mesh')  
subplot(3,1,2)  
surf(X, Y, Z);  
title('Surf')  
subplot(3,1,3)  
contour3(X, Y, Z);  
title('Contour')
```



Další

diskrétní data: stem, stem3, stairs

vektory a vektorová pole: feather, quiver, quiver3

kovexní obal: convhull

triangulace dat: triplot

mnohoúhelníky: fill, fill3

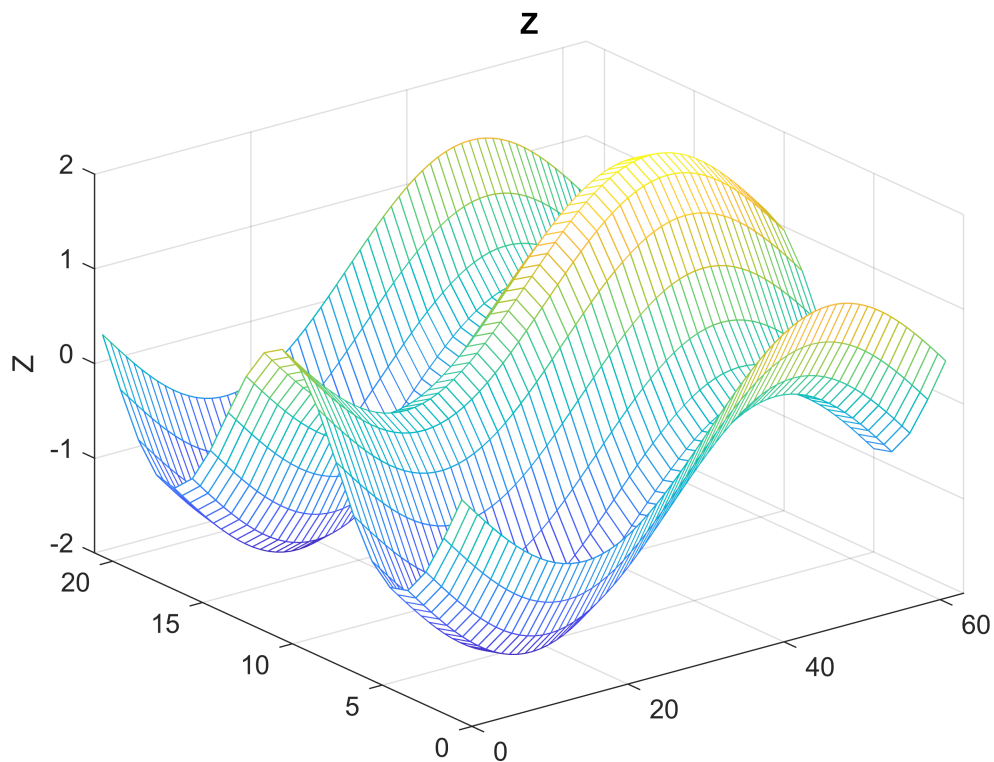
<https://uk.mathworks.com/products/matlab/plot-gallery.html>

Live tasks

Vyzkoušejte si práci s INSERT - Task - Create Plot

```
figure,
% Create mesh of Z
s = mesh(Z);

% Add xlabel, and title
xlabel("Z")
title("Z")
```



Statistické zpracování dat

Už dříve jsme si ukázali funkce `min`, `max` a `mean`.

Mezi další základní statistické funkce patří:

`mink` a `maxk`, které vrací k nejmenším, respektive největším hodnot vektoru.

```
v = randi(20,[1,10])
```

```
v = 1x10
    4    20    20    10    17    3    9    19    16    20
```

```
mink(v,3)
```

```
ans = 1x3
     3     4     9
```

`bounds`, která vrací nejmenší a největší hodnotu vektoru.

```
[v_min, v_max] = bounds(v)
```

```
v_min = 3
v_max = 20
```

Medián spočítáme pomocí `median`

```
median(v)
```

```
ans = 16.5000
```

mode slouží k výpočtu nejčastěji zastoupené hodnoty

```
mode(v)
```

```
ans = 20
```

Standardní odchylku pomocí std, varianci pomocí var a root-mean-square value rms

```
std(v)
```

```
ans = 6.2370
```

```
var(v)
```

```
ans = 38.9000
```

```
rms(v)
```

```
ans = 13.1111
```

topkrows, která vrací prvních k řádků matice seřazených podle některého (defaultně prvního) sloupce.

```
M = randi(20,[4,4])
```

```
M = 4x4
```

```
14 14 14 6
 1 16  4  1
17 15 15 2
19  8  1 17
```

```
topkrows(M,2)
```

```
ans = 2x4
```

```
19  8  1 17
17 15 15  2
```