

16. Matlab - 2D grafy

Foltin Martin · MATLAB/Comsol

25.09.2009



Silnou stránkou prostredia Matlab je grafika. Grafika je nástroj na prehľadné zobrazovanie výsledkov, ktoré nie sú na prvý pohľad len zo spleti čísel zjavné. Matlab disponuje pokročilo grafikou v oblasti 2D, tak aj 3D. Spomenúť treba aj pokročilejšie nástroje pre animáciu a virtuálnu realitu. Dnes sa však budeme venovať jednoduchším veciam a na tie pokročilejšie sa zameriame v budúcich číslach nášho seriálu. Svoju pozornosť teda upriamime na tvorbu 2D grafov, ktoré sa aj v technickej praxi najčastejšie využívajú.

Z matematického pohľadu pomocou 2D grafov obvykle zobrazujeme závislosť jednej veličiny od druhej. Túto závislosť môžeme zapísať jednoduchou rovnicou (1)

$$y = f(x) \quad (1)$$

kde x je premennou nezávislou a y je premenná závislá (nakolko jej hodnota závisí od vstupujúcej hodnoty x do funkcie f).

Podme si demonštrovať možnosti Matlabu na funkcii (2). Jedná sa o kvadratickú funkciu a vyšetrujme ju na intervale od -2 do +2.

$$y = x^2 - 1 \quad (2)$$

Najskôr potrebujeme vytvoriť vektor nezávislej premennej x . Dohodli sme sa, že bude z intervalu od -2 do +2 a pre naše potreby bude postačovať ak budú prvky vektora vzdialené od seba o 0,1.

```
>> x=-2:0.1:2;
```

Teraz môžeme vypočítať funkčné hodnoty v zmysle rovnice (2)

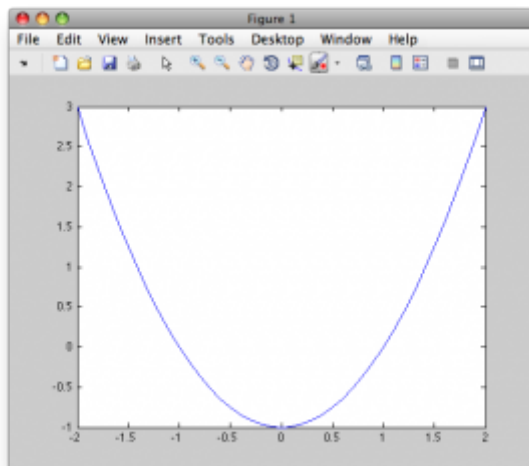
```
>> y=x.^2-1;
```

Máme teda vektor nezávislej premennej x a závislej y s príslušnými funkčnými hodnotami. Na zobrazenie funkčnej závislosti medzi oboma veličinami využijeme príkaz **plot**. Táto funkcia má 2 základné parametre. Prvým je vektor nezávislej

premennej a druhým je vektor závislej premennej. Zadávanie vektora nezávislej premennej nie je povinné. Ak teda zadáme iba jeden vektor, tak Matlab automaticky doplní veličinu nezávislú a to tak, že vygeneruje vektor s hodnotami od 1 až do K s krokom 1, kde K je počet prvkov vektora závislej premennej. V našom príklade ale môžeme využiť oba vektory.

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

Výsledkom je graf funkcie (2) na intervale od -2 do +2 (obr. 1).

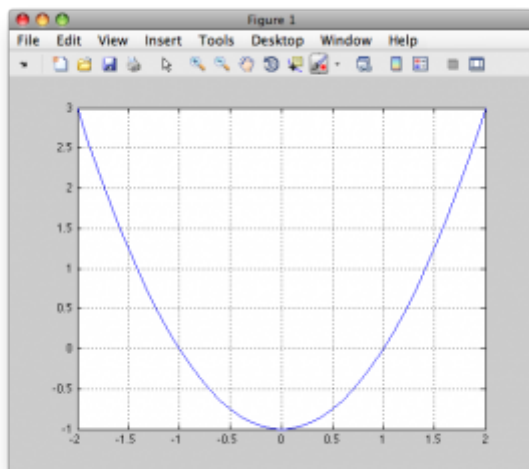


Obr. 1. Graf vykreslený pomocou príkazu **plot**

Graf môžeme teraz upravovať pomocou príkazov pre prácu s grafmi. Pre pridanie mriežky slúži príkaz **grid**.

```
>> grid
```

Výsledok vidíme na obr. 2.



Obr. 2. Do grafu je pridaná mriežka - **grid**

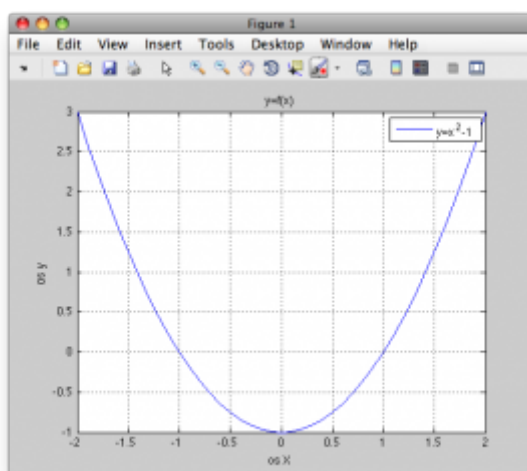
Opätovným zadaním príkazu **grid**, mriežku vypíname. Obdobne môžeme použiť príkaz **grid** s parametrom **on** alebo **off**, ak chceme mriežku zapnúť alebo vypnúť.

```
>> grid on
>> grid off
```

Dôležitá skupina príkazov je určená pre popis grafu. Príkazmi tak pridáme do grafu popisy osí (**xlabel** a **ylabel**), nadpis grafu (**title**), alebo legendu (**legend**). Parametrami týchto príkazov je reťazec, ktorý chceme na danej pozícii umiestniť.

```
>> xlabel('os X')
>> ylabel('y=x^2')
>> ylabel('os y')
>> title('y=f(x)')
>> legend('y=x^2-1')
```

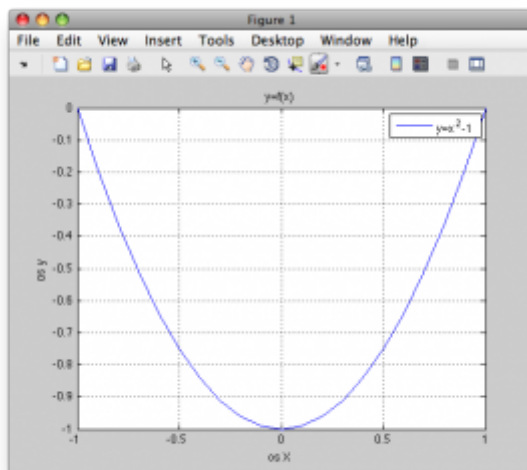
Výsledok tejto sekvencie príkazov vidíme na obr. 3. Za povšimnutie stojí, ako Matlab interpretuje zápis $y=x^2-1$. Do grafu automaticky zapisuje 2 ako mocninu (horný index).



Obr. 3 Graf s popisom osí, nadpisom a legendou

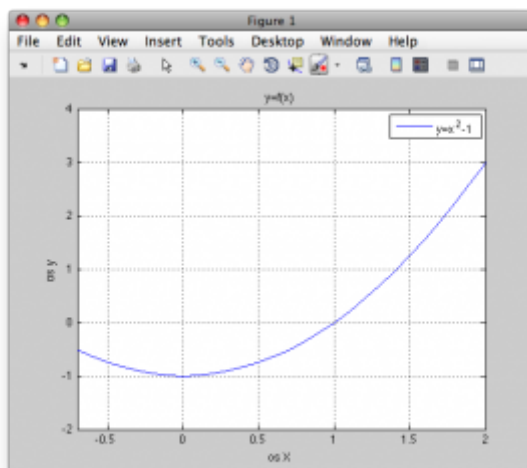
Aj po vykreslení grafu, môžeme meniť škálu v ktorej sa vykresluje. Dajú sa meniť rozsahy osí samostatne (príkazy **xlim** a **ylim**), alebo oboch osí naraz (príkaz **axis**). V prvých dvoch prípadoch sú parametre 2 a to minimum a maximum pre danú os. V prípade **axis** sú parametre 4, čiže minimum a maximum pre x-ovú os a minimum a maximum pre os y.

```
>> xlim([-1 1])
```



Obr. 4 Zmena x-ovej osi pomocou príkazu **xlim**

```
>> axis([-0.7 2 -2 4])
```



Obr. 5 Zmena oboch osí súčasne príkazom **axis**

Priblížili sme si ako vytvoríme graf a spravíme jeho popis. V grafe však máme len jeden priebeh. Matlab však dokáže zobrazíť v jednom grafe priebehov niekoľko. Najskôr si vytvoríme druhú závislú premennú y_2 (3).

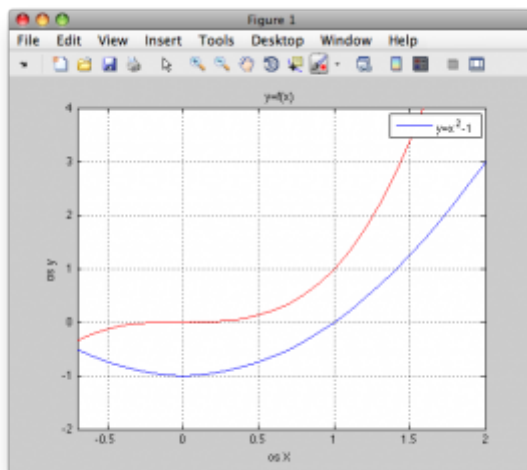
$$y_2 = x^3 \quad (3)$$

Predpokladajme, že chceme aj túto premennú zobrazovať na rovnakom intervale ako premennú y .

```
>> y2=x.^3;
```

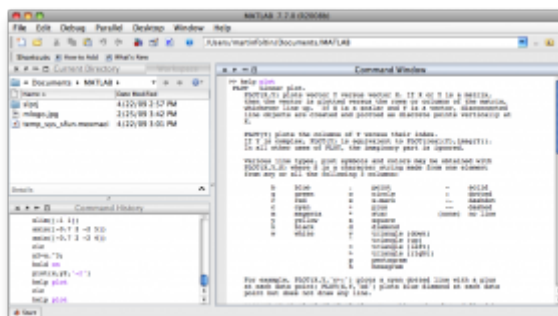
Ak by sme teraz použili príkaz **plot**, tak by sme si predošlý priebeh premazali. My by sme ich ale chceli obzrieť súčasne. Preto musíme zafixovať priebehy v grafe príkazom **hold on**. Až po zafixovaní pristúpime k vykresleniu druhého priebehu.

```
>> hold on
>> plot(x,y2, '-r')
```



Obr. 6 dva priebehy v grafe

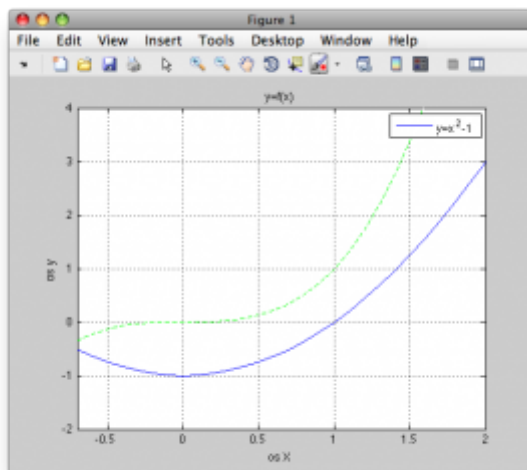
V tomto príklade sme doplnili príkaz **plot** o nepovinný parameter **-r**. Nepovinné parametre pre príkaz **plot** sa zadávajú ako string. V rámci tohto parametra nastavujeme farbu a typ čiary zobrazovaného priebehu. Prvý znamená typ čiary a druhý farbu. Znak mínus teda predstavuje spojitú čiaru a r červenú farbu (red). Význam nepovinných parametrov je zrejmý po zadaní príkazu `help plot`



Obr. 7 Význam nepovinných parametrov v príkaze **plot**

Znamená to teda, že ak by sme chceli vykresliť priebeh y_2 zelenou farbou a prerušovanou čiarou, použijeme parametre **-g** a nahradíme ním parameter **-r**.

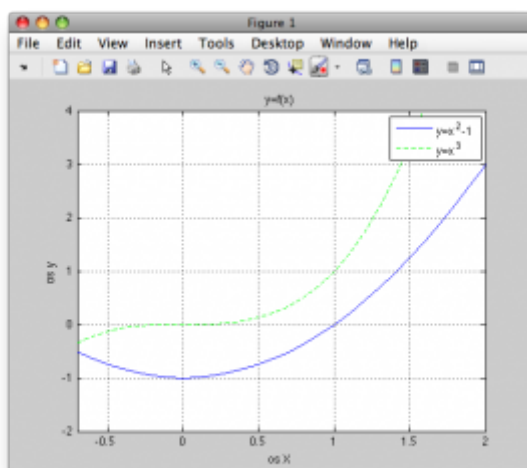
```
>> plot(x,y2,'-- g')
```



Obr. 8 Priebeh y_2 s parametrom $-g$

Vidíme, že legenda sa pridaním nového priebehu nezmenila. Preto ju musíme doplniť.

```
>> legend('y=x^2-1', 'y=x^3')
```



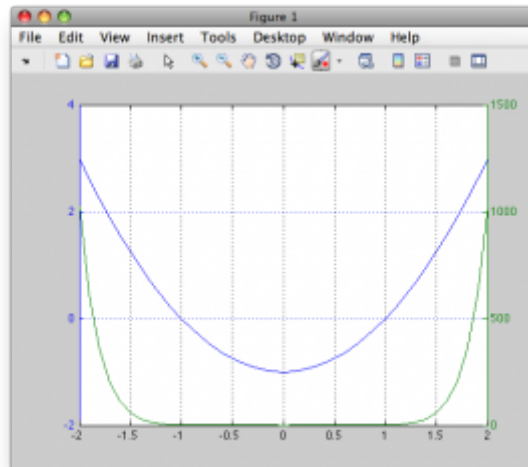
Obr. 9 doplnenie legendy

V uvedenom príklade sme síce vykreslili dva priebehy do jedného obrázku, využili sme ale jednotné škálovanie osi y . V tomto prípade ešte nedošlo ku komplikáciám a oba priebehy sa pohybovali v rádovo rovnakých hodnotách. Môže sa ale stať, že ďalší priebeh ktorý chceme zobrazit dosahuje rádovo iné hodnoty. V takom prípade by sa stal graf značne neprehľadným. Preto Matlab obsahuje príkaz aj na vykreslenie dvoch priebehov, pričom každý ma vlastné škálovanie. Jedná sa o príkaz **plotyy**. Syntax príkazu je **plotyy(x1,y1,x2,y2)**. Pre tento príkaz neexistujú nepovinné parametre, ktorými by sme mohli meniť farbu priebehov. Zmeny sa samozrejme robiť dajú. Jedná sa však už o pokročilejšie narábanie s grafickými objektmi a tejto téme sa budeme venovať v niektorom z pokračovaní seriálu.

Na príklade si ukážeme použitie príkazu **plotyy**. Do obrázku chceme zakresliť priebeh premennej $y(2)$ a $y^3(4)$

$$y^3 = x^{10} \quad (4)$$

```
>> y3=x.^10;
>> plotyy(x,y,x,y3)
>> grid
```

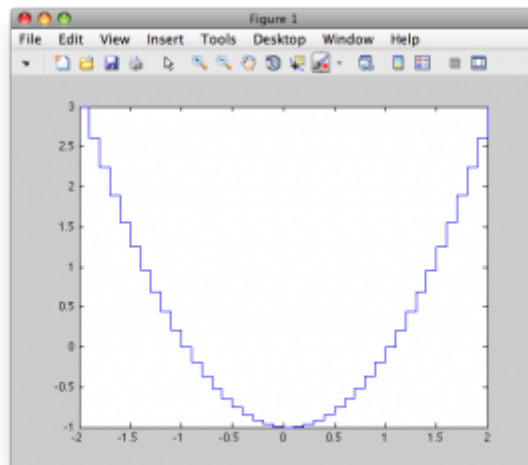


Obr. 10. zobrazenie dvoch rádo vo odlišných priebehov

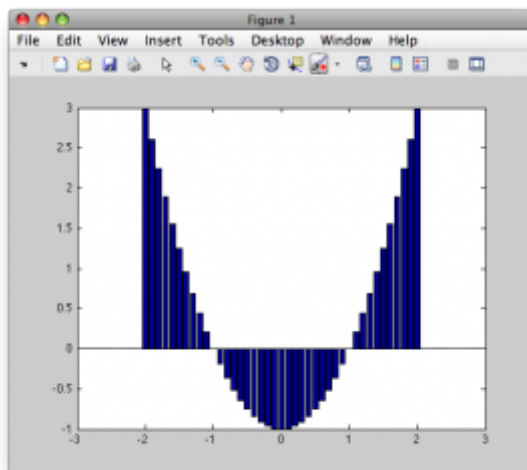
Na vykreslovanie 2D grafov existuje ešte niekoľko príkazov. Princíp ich použitia je analogický ako pre príkaz **plot**. Jedná sa o príkazy **stairs**, **bar**, **stem**.

```
>> stairs(x,y)
>> bar(x,y)
>> stem(x,y)
```

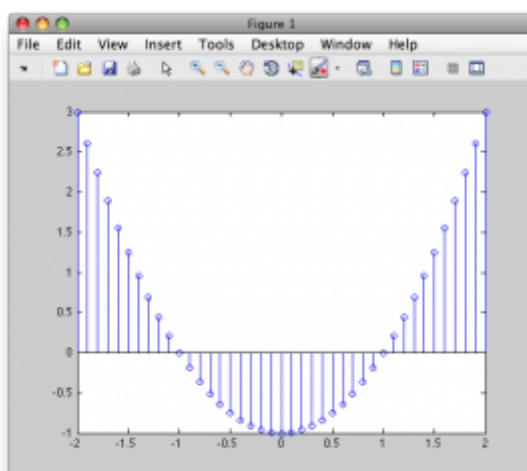
Zobrazenia priebehov po zadaní príkazov dostávame grafy.



Obr. 11 **stairs(x,y)**



Obr. 12 *bar(x,y)*

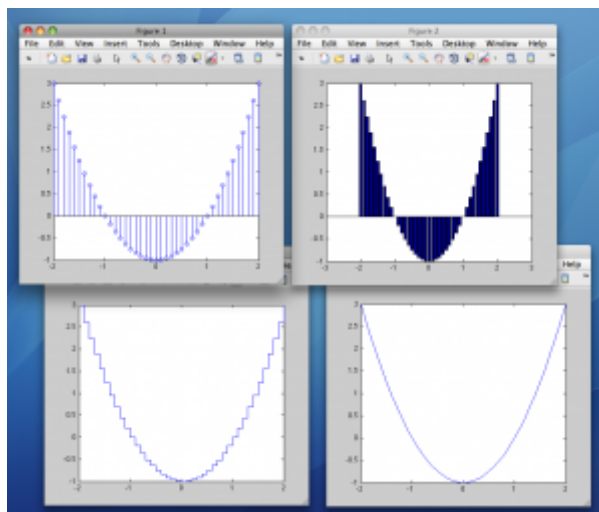


Obr. 13 *stem(x,y)*

Zobrazenie viac grafov

Ak potrebujeme naraz pracovať s viac obrázkami súčasne, máme dve možnosti. Buď vytvoríme viac samostatných grafov, alebo do jedného obrázku vložíme niekoľko podobrázkov. V prvom prípade použijeme príkaz **figure**. Tento príkaz vytvorí nový, prázdny obrázok a my doň môžeme kresliť priebehy. Ak potrebujeme aktuálny obrázok zatvoriť, použijeme príkaz **close**. Ak chceme zavrieť všetky obrázky naraz, použijeme **close all**.

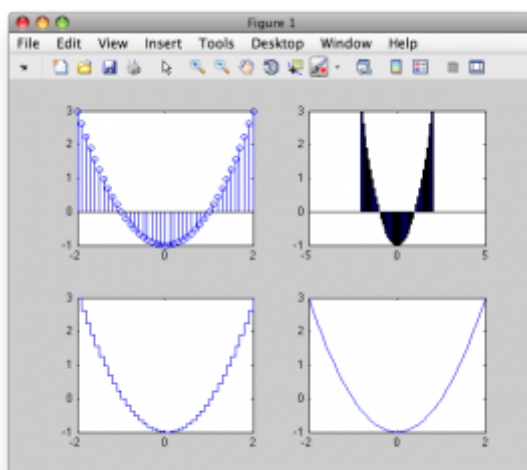
```
>> figure
>> stem(x,y)
>> figure
>> bar(x,y)
>> figure
>> stairs(x,y)
>> figure
>> plot(x,y)
>> stem(x,y)
```

Obr 14 Viac samostatných obrázkov

Druhá možnosť je využiť príkaz **subplot**. Tento príkaz má 3 parametre. Počet pozícií v riadku, počet pozícií v stĺpci a aktívna pozícia. Aktívna pozícia je skalárna hodnota a vyjadruje poradové číslo pozície, pričom prvá pozícia je vľavo hore a posledná vpravo dole. Predošlé 4 grafy tak dostaneme do jedného obrázku takto:

```
>> subplot(2,2,1)
>> stem(x,y)
>> subplot(2,2,2)
>> bar(x,y)
>> subplot(2,2,3)
>> stairs(x,y)
>> subplot(2,2,4)
>> plot(x,y)
```



Obr. 15 Využitie príkazu **subplot**

V tejto časti seriálu sme sa venovali úvodu do práce s grafmi. Pred nami ostávajú veľké kapitoly ako 3D grafy, interaktívny nástroj na spracovanie grafov, objektová práca s grafmi a animácie. Tieto témy budeme preberať v niektorom z pokračovaní. Podrobné informácie k práci s grafmi sa nachádzajú v [používateľskej príručke](#).

Literatúra

1. Matlab 7 - Graphics, The MathWorks,
http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/graphg.pdf, 24.9.2009
-