

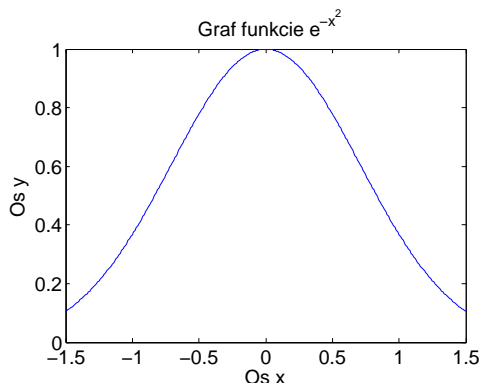
1 Grafika v Matlabe

matlab umožňuje užívateľovi vytvárať rôzne grafické objekty. Postupne sa oboznámime s niektorými dvoj- a trojrozmernými typmi grafov a animáciami. V prípade záujmu sa oplatí pozrieť si demá v Matlabe alebo internetovú stránku

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

1.1 2-D grafika: funkcia plot

Funkcia `plot` slúži na nakreslenie dvojrozmerných grafov (tzn. krivky v 2D), ako napríklad tento:



Základná syntax príkazu je nasledovná: `plot(x,y)`, kde x je vektor x -ových súradníc, y je vektor y -ových súradníc. Graf potom vykreslí dvojice bodov (x, y) , ktoré automaticky pospája modrou súvislou čiarou.

1.1.1 Zdokonaľovanie grafov

Poznamenajme, že „čistý“ výstup z príkazu `plot(x,y)` nie je všetko to, čo vidíme na vyššie uvedenom obrázku. Popisy osí a názov obrázku sme museli dodať, a tiež sme nastavili väčšiu veľkosť písma. Obrázky sa dajú rôznymi spôsobmi upravovať, tu sú niektoré z možností:

- `title` - pridá do grafu jeho názov, napr. `title('Graf funkcie e^{-x^2}')`
- `xlabel` - pridá do grafu popis horizontálnej osi, napr. `xlabel('Os x')`
- `ylabel` - pridá do grafu popis vertikálnej osi, napr. `ylabel('Os y')`
- `legend` - pridá do grafu legendu, napr. `legend('Funkcia e^{-x^2}')`
- `text` - pridá do grafu text na zadané miesto, napr. `text(-0.5,0.3,'Tu je moj text')`
- `axis` - nastaví rozsah zobrazenia osí, syntax `axis([xmin xmax ymin ymax])`
- `axis off` - vypne zobrazovanie osí
- `axis square` - osi budú opticky (nie rozsahovo) rovnako veľké, graf bude v štvorcovom okne

Všetky tieto nastavenia sa do programu píšú až *za* samotný kľúčový príkaz `plot`. Mnohé z týchto vlastností, ale aj mnohé ďalšie sa dajú do grafu dodať aj manuálne po zbehnutí programu priamo v grafickom okne, kliknutím na položku **Insert**. Taktiež sa dá meniť druh alebo farba čiary, markery na bodoch grafu, atď. V grafickom okne sa tak dá urobiť dvojkliknutím na vybraný objekt, avšak šikovnejšie je požadovaný vzhľad grafu naprogramovať, aby sme ich nemuseli ručne dorábať po každom zbehnutí programu.

Pre zmenu druhu čiary môžeme použiť rozšírenú syntax `plot(x,y,'--')`, kde do apostrofov vložíme symbol pre zvolený druh čiary. Možné symboly sú:

-	plná čiara (default)	:	bodkovaná čiara
--	prerušovaná čiara	-.	bodkočiarkovaná čiara

Pre zmenu farby čiary môžeme k použiť nasledujúce nastavenia:

y	žltá (yellow)	m	purpurová (magenta)
c	modrozelená (cyan)	r	červená (red)
g	zelená (green)	b	modrá (blue)
w	biela (white)	k	čierna

Na zvýraznenie bodov na grafe môžeme použiť tzv. markery, ktoré môže mať rôzny tvar:

+	znak plus	s	štvorček
*	znak hviezdica	p	päťcípá hviezda
o	krúžok	d	kosoštvorec
x	krížik	h	šesťcípá hviezda

Môžeme vytvárať ľubovoľné kombinácie druhov bodov a čiar, ktoré ich spájajú. Napríklad zelenú bodkočiarkovanú čiaru by sme dosiahli pomocou 'gp-.' na tretej pozícii v príkaze plot.

Príklad 1.1 Otvorte si súbor **graf1.m** a pozrite sa, ako je vytvorený graf z obrázka. Ďalej si vyskúšajte rôzne nastavenia druhov čiar, farieb, markerov, atď.

Príklad 1.2 Zistite, čo znamenajú jednotlivé voľby v súbore **graf2.m**.

1.2 Viac grafov v jednom obrázku (hold on, subplot, figure)

Neraz potrebujeme vykresliť v jednom okne viac grafov naraz. Toto sa dá v Matlabe dosiahnuť rôznymi spôsobmi. Prvým z nich je do príkazu plot vymenovať všetky funkcie, ktoré chceme vykresliť, pričom vždy usíme nanov zadávať aj vektor x -ových súradníc, hoci sú totožné pre rôzne funkcie: `>> plot(x,y,x,z,x,u)`. V prípade, že chceme ovplyvňovať tvar čiary a podobné nastavenia, tieto musia prísť vždy za príslušnú dvojicu vektorov, napr.

```
>> plot(x,y,'gp-.',x,z,'ms--',x,u,'r')
```

Príklad 1.3 Naprogramujte takýmto spôsobom do jedného obrázka grafy dvoch funkcií: $f(x) = e^{-x^2}$ a $g(x) = e^{-x}$ na intervale $x \in [-1.5, 1.5]$, pričom krok delenia intervalu zvolte $\Delta x = 0.01$.

Druhou možnosťou, ako dosiahnuť tento istý výsledok, je použiť príkaz hold on nasledujúcim spôsobom:

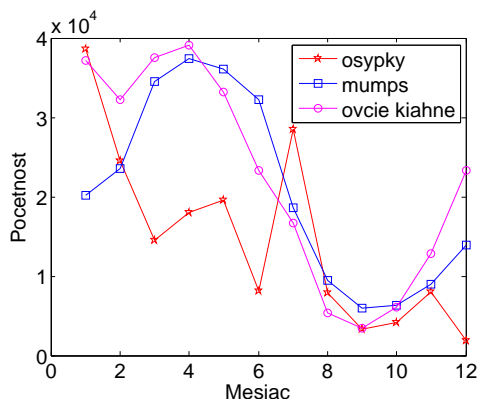
```
>> plot(x,y,'gp-.')
>> hold on
>> plot(x,z,'ms--')
>> hold on
>> plot(x,u,'r')
>> hold off
```

Príkaz hold on akoby „pridrží“ grafické okno a dáva Matlabu povel, že ďalší graf sa bude kresliť do toho istého okna a na to isté „plátno“.

V prípade, že chceme vykresliť naraz viac grafov, ale v samostatných oknách, použijeme príkaz `figure`:

```
>> figure
>> plot(x,y,'gp-.')
>> figure
>> plot(x,z,'ms--')
>> figure
>> plot(x,u,'r')
```

Príklad 1.4 Pozrite si program `priklad_holdon.m` a preštudujte si zobrazovanie viacerých grafov na to isté plátno:

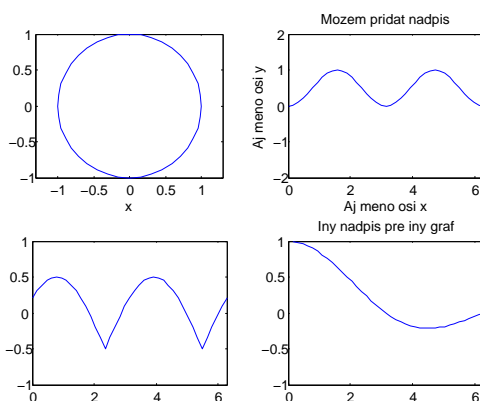


A na záver, ak chceme naraz vykresliť viac grafov v jednom Figure Window, ale v samostatných „plátnach“ (teda aby sa grafy neprekrývali), použijeme príkaz `subplot`. Tento príkaz má syntax

```
subplot(i,j,k)
```

pričom i je počet riadkov, na koľko sa má rozdeliť grafické okno, j je počet stĺpcov, a k je poradové číslo podokna, do ktorého chceme aktuálny graf vykresliť. Podokná sa číslujú od 1 po riadkoch, z ľavého horného rohu do pravého dolného.

Príklad 1.5 Pozrite si program `subploty2.m` a preštudujte si použitie príkazu `subplot`:



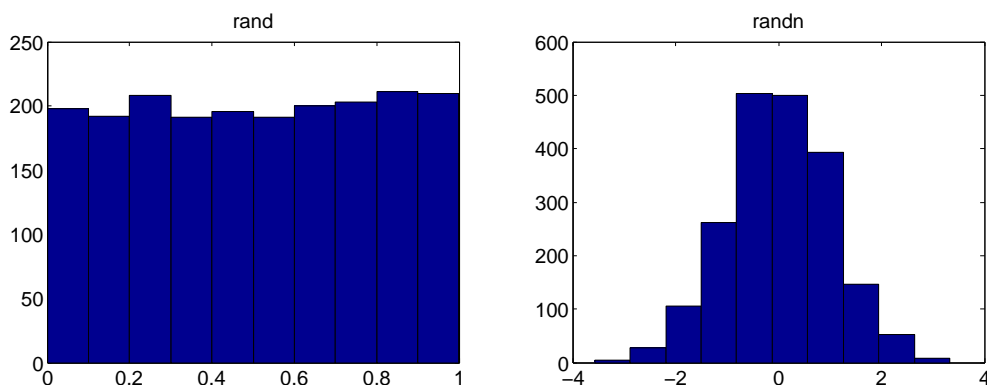
Uvedené spôsoby znázorňovania viacerých grafov fungujú pre ľubovoľné typy kresliacich príkazov, nielen `plot`.

1.3 2-D grafika: histogramy a stĺpčekové grafy

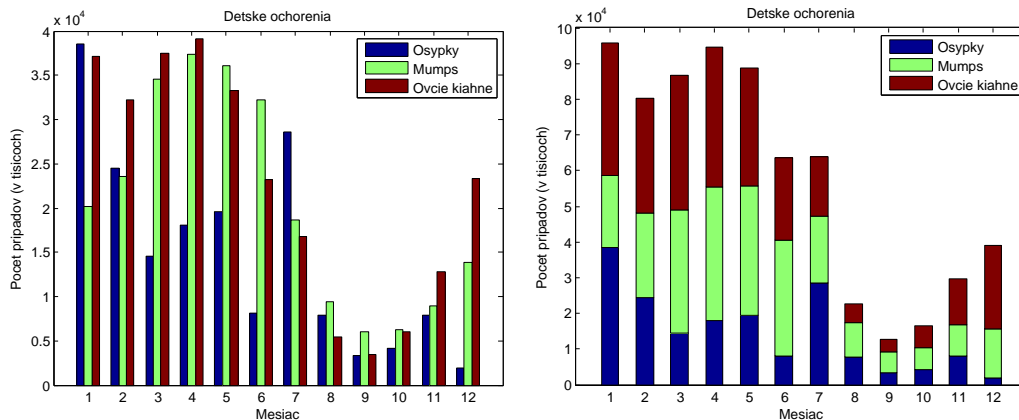
Histogram v jeho najjednoduchšej forme zobrazuje početnosť výskytu realizácií nejakej náhodnej premennej. Jeho základná syntax je `hist(x)`, kde x je vektor dát.

Príklad 1.6 Pozrite si program `histogram_chripka.m` a preštudujte si použitie príkazu `hist`. V súbore sa načítajú dáta o veku pacientov, u ktorých bolo v dedinke Okrajová zaznamenaná v uplynulej sezóne chrípkové ochorenie. Pozn.: dáta sú vymyslené.

Príklad 1.7 Vygenerujte vektor \mathbf{a} náhodných čísel dĺžky 1 000 pomocou príkazu `rand` a vektor \mathbf{b} náhodných čísel tej istej dĺžky pomocou príkazu `randn`. Dajte si vykresliť histogramy z oboch vektorov spolu do jedného grafického okna, ale na samostatné plátna. Čo pozorujete ohľadom tvaru histogramov? Čo je zobrazené na horizontálnej osi a čo na vertikálnej? Zapamätajte si: príkaz `rand` generuje náhodné čísla z rovnomerného rozdelenia, príkaz `randn` z normálneho rozdelenia.



Príklad 1.8 Pozrite si program `priklad_bar.m` a preštudujte si použitie príkazu `bar`. V súbore sa umelo vytvorila dáta, ktoré akože predstavujú počet prípadov troch ochorení (osýpky, mumps, ovčie kiahne) za každý mesiac v priebehu jedného roka a tieto výsledky sa potom zobrazia do stĺpčekového grafu, a to dvoma rôznymi spôsobmi v dvoch samostatných grafických oknách. Grafy sú napokon vizuálne upravené pomocou dodatočných príkazov.

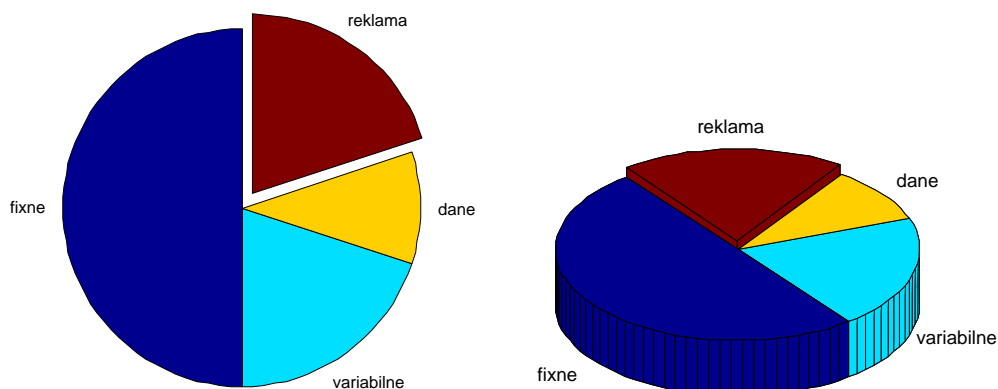


Ďalšie typy stĺpčekových grafov, ktoré sú v Matlabe k dispozícii, sú napríklad horizontálny stĺpčekový graf, dvojrozmerný histogram, scatter a ďalšie. Ich príklady a použitie si môžete naštudovať prostredníctvom internetovej stránky

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

1.4 2-D grafika: koláčový graf

Príklad 1.9 *S pomocou Helpu nakreslite koláčový graf, v ktorom znázorníte pomer nákladov firmy: fixné náklady 50, variabilné náklady 20, dane 10, reklama 20. Príkazy: pie, pie3.*



2 3-D grafika

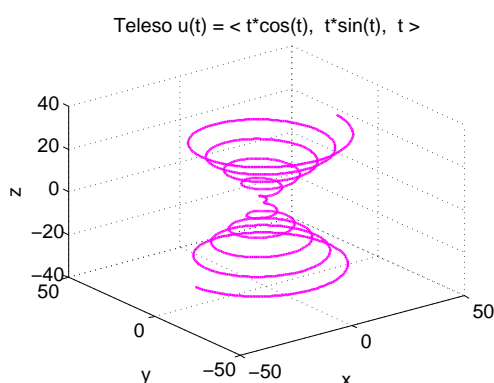
V trojrozmere môžeme zobrazovať buď 3D krivky, alebo plochy. Vykreľovanie 3D kriviek funguje veľmi podobne ako v 2D. Pri plochách je to však už trochu zložitejšie - treba si najprv definovať spodnú mriežku bodov a nad každým bodom vypočítať funkčnú hodnotu, ktorá sa má vykresliť.

2.1 3-D grafika: kreslenie kriviek

Na kreslenie kriviek v 3D slúži príkaz `plot3`, pričom ako vstupné argumenty dostáva tri vektory: vektor x -ových súradníc, vektor y -ových súradníc, vektor z -ových súradníc. V grafe sa potom vykreslia príslušné usporiadané trojice. Základná syntax tohto príkazu je:

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

Príklad 2.1 *Pozrite si súbor `graf4.m` a naštudujte si použitie príkazu `plot3`.*



2.2 3-D grafika: kreslenie plochy (meshgrid, surf, mesh)

Na kreslenie trojrozmerných grafov funkcií dvoch premenných je najskôr potrebné zdefinovať spodnú mriežku, nad ktorou sa má graf vykresliť. Táto mriežka sa generuje pomocou príkazu `meshgrid(x,y)`, kde x je vektor deliacich bodov na x -ovej osi, y je vektor deliacich bodov na y -ovej osi. Príkaz `meshrid` z týchto dvoch vektorov vytvorí matice definujúce celú mriežku: maticu x -ových súradníc všetkých bodov mriežky a maticu y -ových súradníc všetkých bodov

mriežky. Tieto dve matice sú dvoma výstupmi funkcie `meshgrid`, preto volanie tejto funkcie má obvykle tvar `[X,Y] = meshgrid(x,y)`.

Ďalším krokom po vytvorení mriežky je výpočet funkčných hodnôt (matica Z) nad každým bodom mriežky a následné vykreslenie grafu pomocou funkcie `surf` alebo `mesh`. Mesh vykreslí iba sieť, zatiaľ čo `surf` navyše aj vyfarbí príslušné plošky. Syntax:

```
mesh(X,Y,Z), alebo surf(X,Y,Z)
```

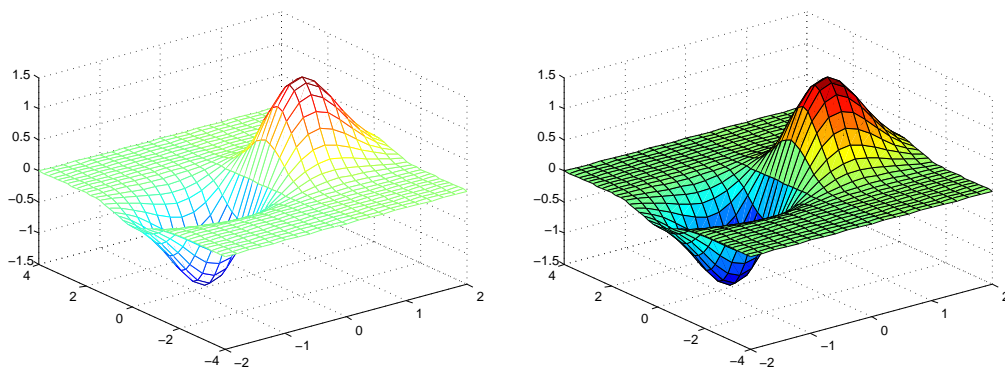
Podobne ako v 2D, aj tu sa dajú grafy skrásľovať, napríklad si môžete nastaviť farebnú sadu, ktorá má byť na grafe použitá, a to z nasledujúcich možností:

príkaz `colormap` a za ním jedna z volieb `jet`, `bone`, `autumn`, `hot`, `copper`, ...

Taktiež si môžete zvoliť, či chcete zobraziť aj sieť alebo iba hladkú farebnú plochu, pomocou volieb

príkaz `shading` a za ním jedna z volieb `flat`, `faceted`, `interp`

Príklad 2.2 Pozrite si súbor `priklad_grafplochy.m` a naštudujte si použitie príkazov `meshgrid`, `mesh` a `surf`, a tiež úprav obrázkov pomocou nastavení `colormap` a `shading`. Doprogramujte popisy osí a názov grafu.



2.3 3-D grafika: vrstevnice (`contour`, `contourf`)

Na kreslenie úrovňových kriviek Matlab používa funkciu `contour`, resp. `contourf`. Táto funkcia očakáva ako vstup maticu, ktorá reprezentuje výšku v rovine $x-y$. Počet úrovňových množín je zvolený automaticky an základe hodnôt vstupnej matice. Modifikáciou funkcie `contour` je funkcie `contourf`, ktorá navyše vyplní plochy ohraničené jednotlivými úrovňami rovnakou farbou. Funkcia `clabel` zobrazí číselné hodnoty úrovňových kriviek do vrstevnicového grafu.

Príklad 2.3 Pozrite si súbor `graf6.m` a naštudujte si, ako sa v Matlabe kreslia vrstvenice funkcie.

2.4 Animácia

Matlab okrem vykresľovania statických grafov umožňuje vytvoriť sled obrázkov - animáciu. Animáciu v Matlabe môžete vytvoriť jedným z dvoch nasledujúcich spôsobov:

1. Uložiť si niekoľko obrázkov a potom ich prehrať ako film.

2. Spojite zmazať a prekresľovať objekty na obrazovke, robiac pri tom inkrementálne zmeny na každom ďalšom obrázku.

Príklad 2.4 Pozrite si súbor `prvaAnimacia.m` a naštudujte si, ako sa v Matlabe robí animácia so sledu statických grafov.

Príkaz `moviein` s celočíselným vstupným parametrom `n` udáva, že animácia pozostáva z `n` snímok. Tieto snímky sú postupne generované v rámci `for`-cyklu.

Funkcia `getframe` vracia kostru filmu. Kostra filmu v sebe obsahuje informáciu o súčasných osiach, okrem pomenovania osí.

Každá snímka je uložená v stĺpci matice `m`. Posledný príkaz v súbore je `movie(m)`. Tento príkaz hovorí Matlabu, aby premietol animáciu, ktorú predtým vytvoril a uložil do matice `m`.

2.5 Uloženie obrázkov z Matlabu

Vytvorený obrázok môžeme uložiť pomocou `File -> Save As`. Do rôznych prezentácií a dokumentov je vhodné zvoliť formát `pdf`, `eps`, v horšej kvalite aj `jpg` alebo `png`. Pre praktické účely odporúčam si vždy obrázok uložiť aj vo formáte `fig` - to je formát, ktorý je znovuotvorablený v Matlabe, ak neskôr potrebujete niečo na obrázku upraviť a nechcete ho celý počítať a vytvárať odznova.

2.6 Ďalšie príklady

Príklad 2.5 Otvorte si súbor `pridavanie_ciar.m` a naštudujte si spôsoby pridávania čiar do grafu.

Príklad 2.6 Nakreslite graf funkcie $y = \sin(x) + x - x\cos(x)$ na intervale $0 \leq x \leq 30$ a na intervale $-100 \leq x \leq 100$ v dvoch osobitných obrázkoch. Pridajte názov a popis osí. Zvoľte správne nakrokovanie tak, aby ste dostali hladké krivky.

Príklad 2.7 Do jedného obrázku nakreslite funkcie $f(x) = x$, $g(x) = x^3$, $h(x) = e^x$ a $z(x) = e^{x^2}$ na intervale $[0, 1]$. Popíšte krivky použitím príkazu `legend`. Zvoľte vhodné nakrokovanie.

Príklad 2.8 Nakreslite graf, kde demonštrujete správanie funkcie $f(x, y) = \frac{xy^2}{x^2+y^2}$ v okolí bodu $(0, 0)$. Použite dostatočne husté nakrokovanie v okolí bodu $(0, 0)$.

Príklad 2.9 Nakreslite obrázok k úlohe z mikroekonómie, v ktorej krivka AC je daná rovnicou: $AC(q) = q/10 + 10000/q$ a krivka MC rovnicou: $MC(q) = q/5$. Nakreslite krivky AC a MC spoločného obrázku, krivky farebne odlište. Vyznačte priesečník oboch kriviek, tj. bod $(100\sqrt{10}, 20\sqrt{10})$.

Príklad 2.10 Zostrojte animáciu pozostávajúcu z piatich obrázkov. Každý z obrázkov zobrazí plochu definovanú ako $f(x, y) = \sin(nx)\sin(ny)$ na intervale $[0, 2\pi] \times [0, 2\pi]$ pre $n = 1, \dots, 5$.

Príklad 2.11 Otvorte si súbor `priklad_colorbars.m` a naštudujte si spôsoby zobrazenia farebnej škály vedľa obrázka. Pre viac možností si pozrite `Help` k príkazu `colorbar`.

Príklad 2.12 Otvorte si súbor `latex_do_grafu.m` a naštudujte si spôsob, ako do grafu pridať matematický text.

Príklad 2.13 Naštudujte si pomocou `Helpu` použitie príkazu `fill`.

Príklad 2.14 Naštudujte si pomocou `Helpu` použitie príkazu `area`. Tento typ grafu môžete využiť napr. v predmete *Finančná matematika*, kde si takýmto spôsobom môžete zobraziť zloženie portfólia v priebehu času.

Príklad 2.15 Naštudujte si pomocou Helpu použitie príkazu `stem`. Tento typ grafu sa využíva napríklad v odbore spracovania digitálnych signálov.

Príklad 2.16 Naštudujte si pomocou Helpu použitie príkazu `scatter`. Na jeho praktické vyskúšanie si stiahnite nejaké vhodné dáta.

Príklad 2.17 Z internetovej stránky <https://www.mathworks.com/products/matlab/plot-gallery.html> si pozrite použitie ďalších typov grafov.