Octave: tvorba grafů a histogramů, fitování

- grafy se tvoří pomocí příkazu plot(x, y, ...), kde x jsou x-ové souřadnice, y jsou y-ové souřadnice a ... jsou další volitelné vlastnosti grafu, viz níže a internet

Jednoduché příklady grafů

x=[0 5 1 3];

y=[4 -2 -3 8];

plot(x,y)

 - první bod [0,4], druhý [5,-2], třetí [1,-3] atd., spojené čárou

x10=0:pi/5:2\*pi;

 - vytvoření proměnné x10, která je od 0 do 2π s krokem po π/5

plot(x10,sin(x10))

 - graf sinu(x10)

x200=0:pi/100:2\*pi;

plot(x200,sin(x200))

 - to samé jeko vpředchozím případě, ale s více body

Popis os, titulek grafu

- pomocí příkazů title(), xlabel() a ylabel() pod příkazem plot()

- př.

 plot(x, y)

title('Zavislost vysky na case')

xlabel('Casove kroky')

ylabel('Vyska [m]')

- text pro titulek i popis os musí být v uvozovkách “ “ nebo ‘ ‘, české znaky nepoužívejte

Změna barvy, stylu, popisky

plot(0:5,5:-1:0,'^',0:6,6:-1:0,'-.',0:8,8:-1:0,'g\*;popisek;' , 0:.1:9,9:-.1:0,'.r',1:10,9:-1:0,'--g\*',2 :10,9:-1:1,'c;usecka;')

- 0:5 - x-ová souřadnice

- 5:-1:0 - y-ová souřadnice (začíná v 5 a s krokem -1 jde do 0)

- barva: r, g, b, m, c nebo w (červená, zelená, modrá, purpurová, azurová a bílá)

- styl: čárový (výchozí) -, čárkovaný --, tečkovaný ., čárkatečka -., hvězdičky \*, trojúhelníčky ^

- bodový graf: uvedením symbolu, který se má v daném bodě vykreslit \*, +, o, x (pokud se před znak umístí --, budou body spojeny čárkovaně pokud se před znak umístí -, budou body spojeny čarou

- mezi ; a ; lze umístit popisek

Více grafů do jednoho grafu

a) první možnost - zadání všech grafů do jediného příkazu plot, který může mít variabilní počet parametrů:

xVice=0:pi/100:2\*pi;

plot(xVice,sin(xVice),xVice,cos(xVice))

b) druhá možnost - pomocí funkce hold, která podrží aktuální obsah výstupního zařízení

plot(xVice,sin(xVice)) #udělá graf

ishold #jaká je hodnota funkce hold? Pokud je zaplá, vytiskne se ans = 0

hold on #zapnutí funkce hold (podržet aktuální obsah výstupního zařízení)

plot(xVice,cos(xVice))

hold off #vypnutí funkce hold

Více grafů vedle sebe

- pomocí příkazo subplot() před příkazem plot()

- subplot vypadá následovně: subplot(počet řádků, počet sloupců,index pozice grafu)

- př.

 subplot(2,1,1)

plot(x1, y1)

title('Prvni graf')

subplot(2,1,2)

plot(x2, y2)

title('Druhy graf')

- celkový obraz má tedy dva řádky a jeden sloupce

Histogram

 data = load(‘rychlosti.txt‘);

 načtení dat

 x = data(:,1);

z dat vyber jen první sloupec (dvojtečka znamená vyber vše, tzn. od začátku do konce (např. 1:10 by bylo od prvního do desátého místa))

 hist(x,15)

 vytvoř histogram z načteného x, histogram bude mít 15 sloupců

 prumer = mean(x)

 výpočet průměru

m = median(x)

 výpočet mediánu

Fitování

- pomocí metody nejmenších čtverců



- výsledné řešení má minimalizovat součet čtverců odchylek vůči každé rovnici (na obrázku se zeleně vyznačené odchylky umocní na druhou a sečtou a tento součet se sanží metoda minimalizovat)

- nutné načíst balíček optim

 pkg load optim

- dále definovat x-ové a y-ové souřadnice dat, která mají být fitována

 x = 0:0.005:5\*T0;

y = (n0.\*( exp((-lam).\*x))).+ 10000000.\*randn (size (x));

- pak nadefinovat funkci, která bude data fitovat s proměnnými, které budou optimalizovány metodou (v našem případě proměnná p(1))

 function y = ffun(x, p)

y = (p(1).\*exp(-p(2)\*x));

endfunction

- dále zvolit výchozí bod pro proměnnou p(1)

 p = [3\*(10^8) (log(2))/T0];

- a zbývá použít funkci leasqr, která počítá několik věcí (yfit, pfit, cvg, iter, viz. nápověda, co znamená co)

 [yfit pfit cvg iter] = leasqr(x, y, p, "ffun");