

2.5 Úlohy kapitoly 2 - riešenia

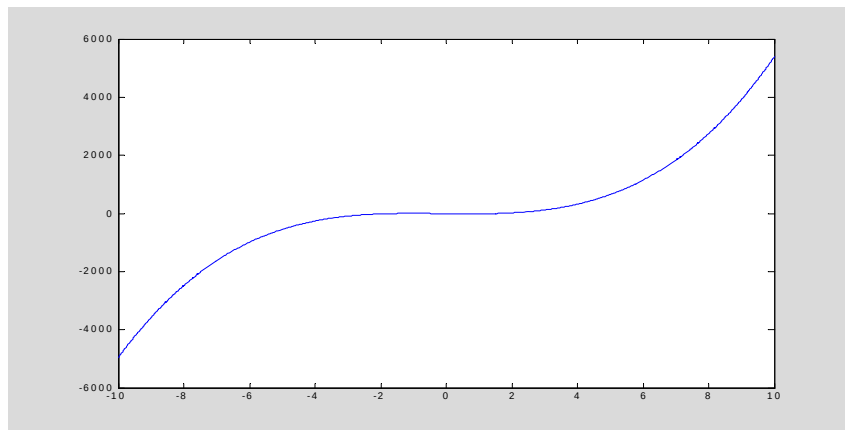
2.5.1 Lokalizujte a separujte korene rovnice $5.3x^3 + 2.2x^2 - 12.5x - 6 = 0$. Použite Newtonovu metódu na aproximovanie koreňov s presnosťou $5e-7$. Modifikujte M-funkciu Newton, v ktorej namiesto "eval" použijete "polyval" (resp. "polyder"). Všimnite si, že štart z bodu $x = -1$ generuje postupnosť iterácií, ktorá konverguje ku koreňu ležiacemu v intervale $[1, 2]$.

Riešenie: Znázorníme graf daného polynómu (ľavej strany rovnice). Najprv definujme vektor koeficientov polynómu:

```
p = [5.3 2.2 -12.5 -6]
p =
    5.3000    2.2000   -12.5000    -6.0000
```

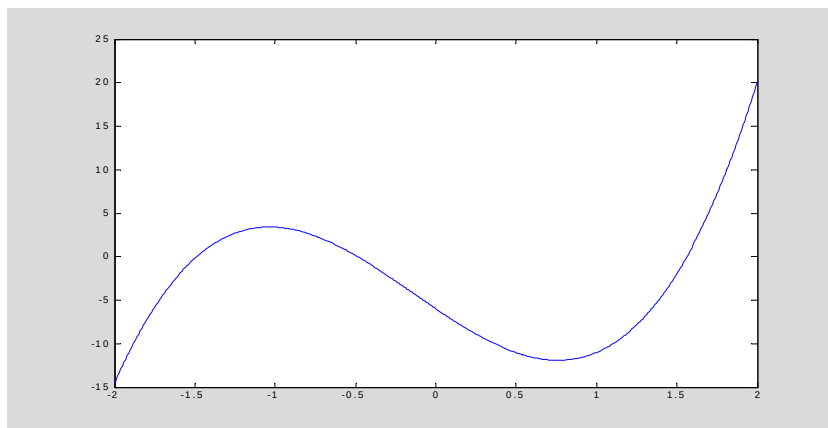
Zrejme pre veľké kladné hodnoty argumentu x budú funkčné hodnoty kladné (koeficient pri x^3 je kladný) a pre záporné hodnoty argumentu, ktoré sú ďalej od nuly, budú funkčné hodnoty záporné (najvyššia mocnina je nepárna). Preto začnime plotovanie polynómu takto:

```
x = linspace(-10, 10, 1001); y = polyval(p,x); plot(x, y)
```



Vidíme, že sme boli zbytočne opatrní, zmeňme interval $[-10, 10]$ na $[-2, 2]$ v nádeji, že znamienkové zmeny nestratíme, a pritom graf polynómu bude lepšie slúžiť na lokalizáciu koreňov.

```
x = linspace(-2, 2, 401); y = polyval(p,x); plot(x, y)
```



Použijeme príkaz "ginput" (grafický input) a určíme aproximácie koreňov. Zľava prvým je koreň, ktorý leží v okolí bodu -1.5 , druhý leží v okolí bodu -0.5 , tretí v okolí bodu 1.5 . Funkciou polyval nájdeme hodnoty polynómu v týchto bodoch (očakávame hodnoty blízke 0):

```
u = [ -1.5 -0.5 1.5 ]; v = polyval(p, u)
v =
   -0.1875    0.1375   -1.9125
```

Hodnota v bode $u = 1.5$ je síce trochu ďaleko od nuly, avšak teraz nejde o kvalitné aproximácie, ale o stanovenie intervalov, v ktorých ležia korene. Návrh môže vyzeráť takto:

```
pre koreň v okolí -1.5 zvolíme interval [-1.6 -1.4]
pre koreň v okolí -0.5 zvolíme interval [-0.6 -0.4]
pre koreň v okolí 1.5 zvolíme interval [1.4 1.6].
```

Overme, že na každom z nich registrujeme znamienkovú zmenu. To sa dá urobiť mnohými spôsobmi, napr. aj takto

```
prod( polyval(p, [1.4 1.6]) )
ans =
    -6.2277
```

pre tretí interval to urobme jednoduchšie (preverenie druhého prenechávame na čitateľa):

```
polyval(p, [1.4 1.6])
ans =
    -4.6448    1.3408
```

Ako vidíme, aj tu registrujeme znamienkovú zmenu. Lokalizácia a separácia koreňov je vykonaná.

Teraz použijeme Newtonovu metódu na aproximovanie koreňa – povedzme toho kladného koreňa, ktorý sa nachádza v intervale [1.4 1.6]. Modifikujeme funkciu "newton" na funkciu "newtonp" takto:

```
function xn = newtonp(p,xs,tol)
% NEWTONP      Newtonova metoda na riesenie
%              algebraickej rovnice p(x)= 0
% -----
% xn = newton(p,xs,tol)
%
% p  = vektor koeficientov polynomu
% xs = bod startu
% tol = zastav. podmienka: |xn - xs| < tol
%
% xn = aproximacia korena
% -----
fs = polyval(p,xs);
fds= polyval(polyder(p),xs);
xn = xs - fs/fds;
%
while abs(xn - xs) >= tol
    xs = xn;
    fs = polyval(p,xs);
    fds= polyval(polyder(p),xs);
    xn = xs - fs/fds;
end
```

Aproximáciu koreňa označme ako "akor", avšak pred volaním funkcie "newtonp" zmeňme format:

```
format long
akor = newtonp(p, 1.5, 5e-7)
akor =
    1.56075804641339
```

Najprv nás zaujíma hodnota $p(\text{akor})$, funkčná hodnota polynómu v bode "akor". Ak nie je dostatočne malá (v absolútnej hodnote), tak to signalizuje nejaký omyl.

```
polyval(p, akor)
ans =
    8.881784197001252e-016
```

V poriadku! Teraz chceme zistiť, či sme dodržali žiadanú presnosť. Totiž použitie tolerancie automaticky nezaručuje, že presnosť sa rovná hodnote "tol". Definujme "akorL" (ľavá) "akorP" (pravá) takto:

```
akorL = akor - 5e-7; akorP = akor + 5e-7; [akorL, akorP]
ans =
    1.56075754641339    1.56075854641339
```

Teraz nás zaujíma, či na intervale [akorL, akorP] nastáva znamienková zmena:

```
polyval(p, [akorL, akorP])
ans =
    1.0e-004 *
    -0.16549588106152    0.16549601605576
```

Ako vidíme, interval [akorL, akorP] obsahuje koreň. Hodnotu 1.560758046 môžeme považovať za aproximáciu kladného koreňa, a pritom chyba tejto aproximácie nepresahuje hodnotu 5e-7.

2.5.2 Reálny koreň rovnice $163x^3 - 767x^2 + 1200x - 625 = 0$ leží v intervale $[1, 2]$. Ak použijeme Newtonovu metódu na aproximovanie koreňa s toleranciou $5e-7$, tak štart z bodu $x = 1$ generuje postupnosť až 18-tich iterácií, kým štart z bodu $x = 2$ postupnosť len 5-tich iterácií. Viete si to vysvetliť?

Riešenie:

Znázorníte (plotujte) graf polynómu na intervale $[1, 2]$ a potom na intervale $[1.4, 1.8]$. Predstavte si, ako pracuje Newtonova metóda. Sformulujte svoju hypotézu.

2.5.3 Nech $p = [1.00, -11.50, 45.60 + a, -73.57, 40.13]$. Nájdite interval pre hodnoty parametra a , aby algebraická rovnica $p(x) = 0$ mala štyri reálne korene.

Riešenie:

Postupujme skusmo, napr. takto:

a= -0.1; p = [1.00, -11.50, 45.60+a, -73.57, 40.13]

p =
1.0000 -11.5000 45.5000 -73.5700 40.1300

roots(p)
ans =
5.1241
2.6414
2.5895
1.1450

Teraz voľme kladnú hodnotu:

a= 0.3; p = [1.00, -11.50, 45.60+a, -73.57, 40.13]

p =
1.0000 -11.5000 45.9000 -73.5700 40.1300

roots(p)
ans =
4.2830
4.1477
1.8448
1.2245

Môžeme na základe zisteného povedať, že hľadaným intervalom je napr. interval $[-0.1 \ 0.3]$?

2.5.4 Nech $p = [1, -9, 25, -14, -9, -81, -135]$. Zvoľte "format long" a použitím M-funkcie "roots" nájdite aproximácie koreňov rovnice $p(x) = 0$. Podrobte získané aproximácie ďalšiemu vyšetrovaniu a nakoniec určte **reálne korene** rovnice čo najpresnejšie.

Riešenie.

p = [1, -9, 25, -14, -9, -81, -135]

p =
1 -9 25 -14 -9 -81 -135

roots(p)
ans =
4.51330197056140
3.14620732850251 + 1.80527057519086i
3.14620732850251 - 1.80527057519086i
-0.37707187720302 + 1.42114460818402i
-0.37707187720302 - 1.42114460818402i
-1.05157287316038

Z výpisu vidíme, že daná algebraická rovnica má dva reálne korene. Ich aproximácie sú:

u = -1.05157287316038

v = 4.51330197056140

Máme stanoviť chybu týchto aproximácií, resp. spresniť tieto aproximácie. Podrobme analýze záporný koreň **u = -1.05157287316038**. Na základe grafu polynómu p (urobte!), definujme bod "uL" takto:

uL = u - 5e-9

uL =
-1.05157287816038

Zistíme hodnoty polynómu v krajných bodoch intervalu $[uL, u]$

```
polyval(p, [uL, u])
ans =
    1.0e-005 *
    0.14377023092038 -0.00000002557954
```

Ako vidíme, na základe zisteného môžeme povedať, že chyba aproximácie $u = -1.05157287316038$ je menšia ako $5e-9$. Tento výsledok sme dostali za pomoci grafického znázornenia priebehu polynómu p v okolí uvažovaného bodu. Teraz sa pokúsme odhadnúť chybu uplatnením vzťahu

$$|\alpha - \alpha^*| \leq \frac{|f(\alpha^*)|}{m}$$

K tomu potrebujeme ohraničiť zdola hodnoty derivácie $|f'(x)|$ pre x z intervalu $[uL, u]$. Presvedčme sa, že za m môžeme vziať hodnotu 287, napr. takto

```
q = polyder(p)
q =
    6    -45    100    -42    -18    -81
polyval(q, [uL, u])
ans =
    1.0e+002 *
    -2.87540508821834 -2.87540505581493
```

Odhad chyby dáva podiel:

```
abs(polyval(p, u))/287
ans =
    8.912731180266065e-016
```

Ako vidíme, pohybujeme sa blízko hranice strojového epsilon ($\text{eps} = 2.2e-16$). Preto môžeme byť s aproximáciou $u = -1.05157287316038$ spokojní a nemá zmysel pokúšať sa spresňovať túto aproximáciu. Chyba tejto aproximácie nepresahuje hodnotu $9e-16$.

Poznámka (pre veľmi zvedavých). Vyskúšajme ako aproximáciu hodnotu $w = u - 3e-16$.

```
w = u - 3e-16; polyval(p, [u, w])
ans =
    -2.842170943040401e-014    2.842170943040401e-014
```

Vieme na základe zisteného (predsa len) spresniť aproximáciu u ?

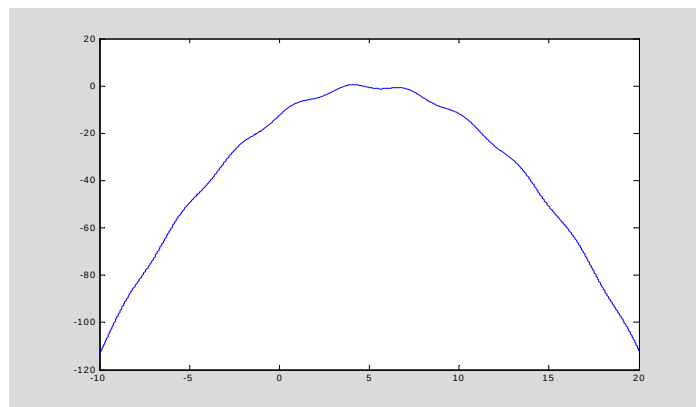
2.5.5 Lokalizujte a aproximujte korene rovnice: $\sin(ax) - b(x - c)^2 = 0$, v ktorej zvolíte parametre takto: nech a, c sú z intervalu $[2, 5]$ a parameter b zvolíte z intervalu $[0.2, 1]$. Odhadnite chyby získaných aproximácií.

Riešenie. Zvoľme parametre a definujme funkciu ľavej strane rovnice ako "inline" objekt napríklad takto:

```
a = 2; c = 5; b = .5; f = inline('sin(a*x) - b*(x-c).^2', 'x', 'a', 'b', 'c')
f =
    Inline function:
    f(x,a,b,c) = sin(a*x) - b*(x-c).^2
```

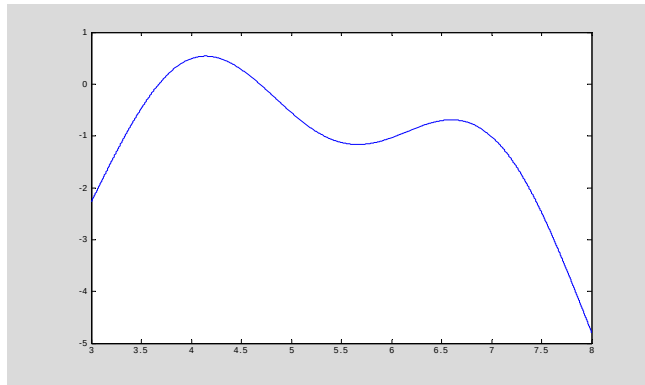
Plotujme graf našej funkcie:

```
u = linspace(-10, 20, 30001); v = f(u,a,b,c); plot(u,v)
```



Funkcia sinus ovplyvňuje hodnoty (a teda tvar) funkcie f len v okolí bodu 5, keď člen $b \cdot (x - 5)^2$ nadobúda malé hodnoty. Korene rovnice treba hľadať v okolí bodu 5. Plotujme preto ešte raz:

```
u = linspace(3, 8, 5001); v = f(u,a,b,c); plot(u,v)
```



Teraz je už jednoduché lokalizovať a separovať korene: prvý koreň leží v okolí bodu 3.7 a druhý v okolí bodu 4.7 (zistili sme použitím funkcie "ginput").

Aj keď zvyšok práce môžeme prenechať čitateľovi, poznamenávame, že použitie M-funkcie "fzero" nie je možné pre funkciu definovanú ako "inline" objekt (ak obsahuje parametre). Môžeme sa síce pýtať na hodnoty funkcie napr. takto:

```
f(3.7, a, b, c)
```

```
ans =  
0.0537
```

ale nevieme preniesť funkciu "f" ako argument funkcie "fzero". Môžeme to urobiť ale tak, že definujeme funkciu bez parametrov, resp. s konkrétnymi hodnotami pre parametre:

```
g = inline('sin(2*x) - 0.5*(x-5).^2')
```

```
g =  
Inline function:  
g(x) = sin(2*x) - 0.5*(x-5).^2
```

Overme:

```
g(3.7)  
ans =  
0.0537
```

Čo naozaj súhlasí s predchádzajúcim. Teraz volajme "fzero"

```
akor = fzero(g, 3.7)  
akor =  
3.67593754843751
```

pre základnú informáciu:

```
g(akor)  
ans =  
-4.440892098500626e-016
```

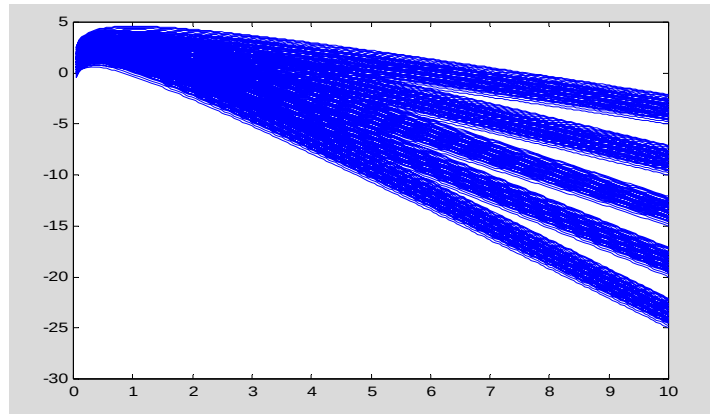
Zdá sa, že zvyšok úlohy môžeme prenechať na samostatnú prácu čitateľa.

2.5.6 Lokalizujte a aproximujte korene rovnice: $\ln(a \cdot x) - b \cdot x + c = 0$, na intervale $[0, 10]$. Zvoľte parametre tak, aby $a \in [2, 5]$, $b \in [1, 3]$, $c \in [2, 4]$. Odhadnite chyby získaných aproximácií.

Riešenie.

Preverme, že navrhnuté intervaly naozaj vedú k rovniciam, ktoré majú korene v intervale $[0, 10]$.

```
x = 3; y = 0; plot(x,y), x = 0:0.05:10; hold on,  
for a = 2:0.5:5,  
    for b = 1:0.5:3,  
        for c = 2:0.5:4,  
            y = log(a*x) - b*x + c; plot(x,y),  
        end,  
    end,  
end,  
shg
```



Zvoľme napr. $a = 3$, $b = 2$, $c = 4$. definujeme ľavú stranu ako "inline" objekt:

```
f = inline('log(3*x) - 2*x + 4')
```

```
f =
    Inline function:
    f(x) = log(3*x) - 2*x + 4
```

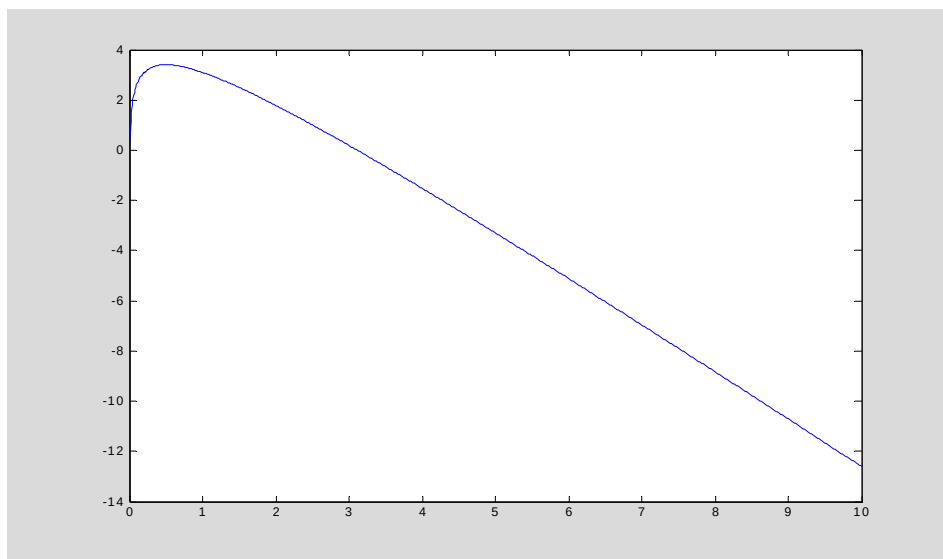
Teraz plotujeme:

```
x = linspace(0, 10, 1001); y = f(x); plot(x,y)
```

Warning: Log of zero.

> In C:\MATLAB6p5p1\toolbox\matlab\funfun\inlineeval.m at line 13

In C:\MATLAB6p5p1\toolbox\matlab\funfun\@inline\subsref.m at line 25
MATLAB upozornil, že argumentom logaritmickej funkcie bola NULA !!!!!



Vlasovým krížom aproximujeme koreň (použité "ginput"). Za aproximáciu koreňa vezmeme napr. hodnotu $x_0 = 3.1$

A teraz volajme M-funkciu "fzero" a vyčíslime hodnotu funkcie v získanej aproximácii koreňa:

```
akor = fzero(f, 3.1), f(akor)
```

```
akor =
    3.11788331832040
```

```
ans =
    0
```

Pre zaujímavosť, vyčíslime f-hodnotu v bezprostredných susedoch bodu akor:

```
lavy = akor - eps; pravy = akor + eps; [f(lavy), f(pravy)]
```

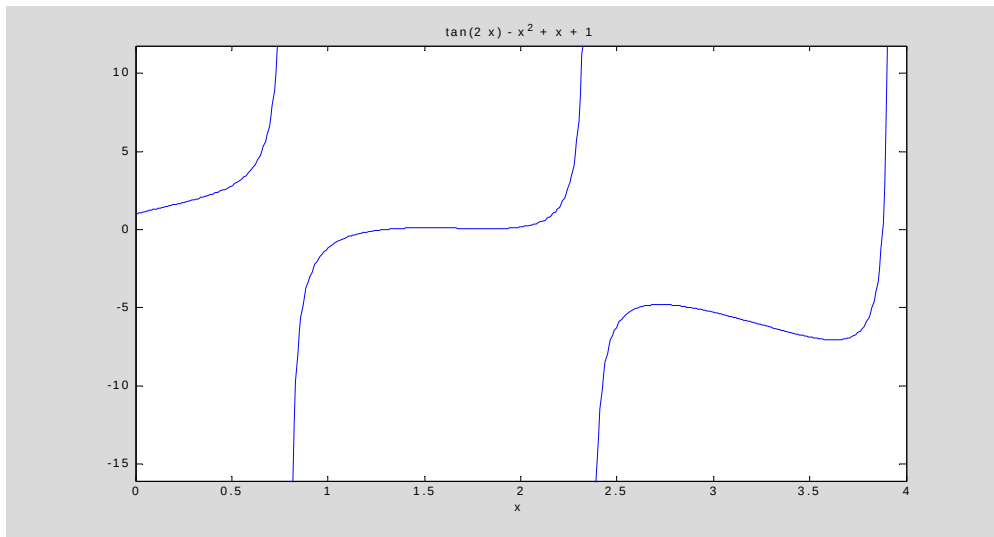
```
ans =
    1.0e-015 *
    0.44408920985006    -0.88817841970013
```

Takto sme zistili, že chyba aproximácie $akor = 3.11788331832040$ je menšia ako $2.2e-16$, teda chyba nepresahuje hodnotu strojového epsilon.

2.5.7 Na intervale $[0, 4]$ lokalizujte a aproximujte korene rovnice: $\tan(ax) + bx^2 + cx + 1 = 0$, v ktorej zvolíte parametre tak, aby $a \in [1, 2]$, $b \in [-1, -0.5]$, $c \in [0, 5]$. Odhadnite chyby získaných aproximácií.

Riešenie. Namiesto riešenia len poznámku. Problémom môže byť hneď plotovanie funkcie, pretože obsahuje funkciu tangens, ktorá nie je ohraničenou. Preto namiesto "plot" použijeme "fplot", alebo "ezplot". Ilustrujme to pri voľbe: $a = 2$, $b = -1$, $c = 1$

```
f = inline('tan(2*x) - x.^2 + x + 1')
f =
    Inline function:
    f(x) = tan(2*x) - x.^2 + x + 1
ezplot(f, [0, 4])
```



príkazom "ginput" (grafický input), pomocou vlasového križa, získame prvé aproximácie koreňov

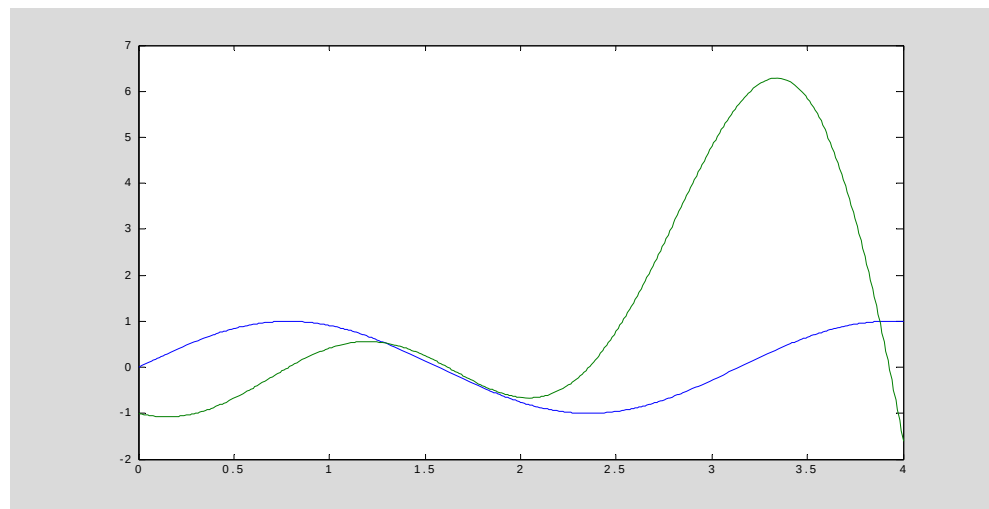
```
ginput(3)
ans =
    1.2633333333333333    -0.00256938427531
    1.8233333333333333    -0.00256938427531
    3.876666666666667    -0.06320640696315
```

Lokalizáciu koreňov rovnice môžeme urobiť aj tak, že prepíšeme rovnicu $f(x) = 0$ na tvar $g(x) = h(x)$. V našej situácii

$$\sin(ax) = -(bx^2 + cx + 1) \cdot \cos(ax)$$

a plotujeme zvlášť ľavú a zvlášť pravú stranu. Takto nepracujeme s neohraničenou funkciou tangens. Urobme len obrázok, opäť pre $a = 2$, $b = -1$, $c = 1$

```
x = 0:0.01:4; y = sin(2*x); z = (x.^2 - x - 1) .* cos(2*x); plot(x,y,x,z)
```



x -ové súradnice priesečníkov sú argumenty, v ktorých platí $g(x) = h(x)$, teda v ktorých platí $f(x) = 0$. Sú to hľadané korene rovnice $f(x) = 0$.

2.5.8 Lokalizujte a aproximujte korene rovnice: $\cotg(ax) = \frac{bx}{x^2 + c}$, na intervale $[0, 5]$, keď jej parametre zvolíte tak, aby $a \in [1, 3]$, $b \in [-6, 6]$, $c \in [1, 4]$. Odhadnite chyby získaných aproximácií. Riešenie tejto úlohy sa ničím zásadným nelíši od predchádzajúcej úlohy.

2.5.9 Ako sme už uviedli, rýchlosť v [m/s] parašutistu v čase t , padajúceho voľným pádom, sa rovná

$$v(t) = \frac{gm}{c} [1 - \exp(\frac{-ct}{m})]$$

Nech $g = 9.81$ [m/s²], $m = 65$ [kg], $c = 11$ [kg/s]. V akom čase dosiahne parašutista rýchlosť 120km/hod.?

Riešenie. Namiesto riešenia len pomôcku. Máme určiť argument funkcie $v = v(t)$, v ktorom má funkcia v predpísanú hodnotu, t.j. hodnotu 120000/3600 (vzťah pre rýchlosť je v m/s).

Výsledok: Parašutista dosiahne uvedenú rýchlosť v čase $t = 5.05659951195132$ [s]

Pozn. Z rovnice je možné (logaritmovaním) vyjadriť čas ako funkciu rýchlosti. To znamená, že na riešenie úlohy nemusíme použiť metódy tejto kapitoly.

2.5.10 Ak pod uhlom α (k horizontále) vystrelíme projektil hmotnosti m rýchlosťou v a pôsobí naň sila F v horizontálnom smere, tak pre vzdialenosť d bodu dopadu platí vzťah:

$$d = \frac{v^2}{g} \sin(2\alpha) + \frac{2F}{mg^2} v^2 \sin^2(\alpha)$$

Nech $v = 600$ [m/s], $F = -90$ [kg.m/s²], $m = 12$ [kg]. Pod akým uhlom je treba vystreliť projektil, aby dopadol vo vzdialenosti 15 km?

Výsledok: Naš model je veľmi zjednodušený, v ňom pre dva uhly dosiahneme predpísanú vzdialenosť. Prvý uhol má veľkosť 15.68142503790328 stupňov a druhý 36.91972256264036 stupňov.

Pomôcka: Napíšme M-funkciu "vzdialenost", ktorej argumentom bude uhol (v stupňoch) a hodnotou vzdialenosť bodu dopadu (v metroch). Uvedomme si, že MATLAB pracuje s trigonometrickými funkciami tak, že ich argumenty musia byť v radiánoch.

```
function d = vzdialenost(alfa)
% pomocna fun pre ulohu kap.2: 2.5.10
% argin:  alfa = uhol vystrelu k horizontale (v stupnoch)
% argout: d    = vzdialenost dopadu(v metroch)
% -----
x = alfa*pi/180;
g = 9.81;
v = 600;
F = -90;
m = 12;
d = (v^2)*sin(2*x)/g + (2*F)*v^2*(sin(x)).^2/(m*g^2);
```

Zrejme treba nájsť argument, resp. argumenty, v ktorých má uvažovaná funkcia predpísanú hodnotu. To môžeme urobiť tak, že modifikujeme našu funkciu, aby išlo o hľadanie koreňa rovnice.