Dobrý den,
dnes máme na řadě řešení nelineárních rovnic.

Úkol na tento týden na konci textu. Deadline na odevzdání 19.4. 2020

Projděte si prosím z přednášek pdf:
http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf

Pro cvičení použijeme zejména stránky: 1-5, 7 a 9-10

A) Metoda půlení intervalů
- teo: str 3 z http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf
- skript: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/pulintervalu.m

zde chceme řešit rovnici sin(*x*)=0.5 pro *x*
vyjádříme si f(*x*)=0 → naše funkce na řádku 7: sin(*x*)-0.5

7) příkaz *inline* je další možností jak zadefinovat funkci v Matlabu se kterou se můžete setkat. Zatím je funkční i do naší verze 2019, ale chystá se její zrušení (viz příkaz *doc inlline*) a plné nahrazení pomocí anonymních funkcí, které jsme si ukázali v minulém cvičení (viz http://kfe.fjfi.cvut.cz/~matysma4/nme/cv6_napln.pdf)

9) řešení nelineárních rovnic je vždy iterační → zadefinujeme si počet iterací a budeme řešení postupně zpřesňovat

11-12) uděláme počáteční odhad, že řešení je mezi 0 a 1
14-15) připravíme si kontejnery na odhad chyby ε – což je délka intervalu mezi body kde hledáme řešení; a kontejner na jednotlivá zpřesňující se řešení

17-26) Provedeme metodu půlení intervalu z http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf
18) nový bod *c* vybereme v polovině mezi *a* a *b* → c=(a+b)/2;
19) naše chyba v dané iteraci odpovídá délce intervalu <*a*,*b*>
20) a řešení odpovídá zvolenému bodu *c*, tj. *(a+b)/2*;
21-25) vybereme podinterval <a,c> nebo <c,b> ve kterém je obsažen kořen
otesujeme jestli f(a)*f(c)<0 → tj. Jestli se kořen nachází mezi *a* a *c*
pokud ano tak dosadíme *c* do *b*
jinak je kořen mezi *b* a *c* → dosadíme *c* do *a*
tím máme nový interval <*a*,*b*> na který aplikujeme další iteraci

28,29) konečné řešení a chyba je na konci příslušných kontejnerů → klíčové slovo *end*
v přechozím iterování se kontejnery vždy nafoukli o 1 v každé iteraci a hodnoty byly zapsány do posledního prvku

Pak už jen vykreslíme průběh řešení
připomínám funkcí *subplot* se kterou vytvoříte několik grafů fo jednoho obrázku
subplot(131); znamená že chceme 1 řádek 3 sloupce a třetí číslo je index jednotlivého podobrázku se kterým pracujeme (tady 1.2 nebo 3)
log10 je desítkový logaritmus (pozor *log* je přirozený logaritmus)

vidíme že pro tento jednoduchý případ jsme dostali relativně přesný výsledek už po 10ti iteracích.
Metoda půlení intervalů je spolehlivá (vždy konverguje), ale v blízkosti kořene je pomalá.
Chtěli bychom u kořene přepnout na jinou metodu → metody využívající sečnu

B) Metoda sečen – sekantová metoda
teorie: http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf strany 3-4
skript: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/secny.m
zkusíme na rovnici : exp(*x*)=6
což je přibližně 1.1978 a zkusíme odhadovat na intervalu <1;5>

skript je podobný předchozímu skriptu ale na řádku
18) dosadíme místo bodu v polovině předchozího intervalu <a;b> bod který leží na průsečíku spojnice těchto dvou bodů a osou *x*. (sečnu)

Vzoreček pro tento bod → viz první část úkolu na tento týden
počáteční krok:
a→ xi
b→ xi_p1
c → nový bod

19) chyba je opět velikost intervalu který uvažujeme

20-21) posuneme body, zahodíme první bod, druhý bod se stane prvním a nový bod druhým.
tj. namísto intervalu <a;b> (tj. <xi;xi_p1>) bereme do další iterace interval <b;c>
→ nemusíme zachovat kořen v intervalu → metoda nemusí konvergovat
→ pro zachování konvergence je potřeba upgrade na metodu regula falsi viz druhá část úkolu

zbytek stejné jako v předchozím skriptu.

Můžete si zkusit rovnici 'exp(x)-6' zadat do skriptu na metodu půlení intervalu a porovnat (nezapomeňte zvednout b, třeba taky na 5). Všimněte si rozdílu ve vývoji chyby.

C) Metoda tečen – Newton–Raphsonova
teo: http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf strana 5
skript: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/tecny.m

Tato metoda je vhodná pokud dokážeme určit / rychle spočítat první derivace zadané funkce.
Proto ve skriptu použito opět exp(x) pro rovnici
exp(x)=3 → f(x)=0
f(x) = exp(x)-3
df(x)/dx = exp(x)

19) pro bod *c* použijeme vzoreček z teorie na straně 5

c=xi_p1-f(xi_p1)/df(xi_p1);

tj. x_novy = x_pred – f(x_pred) / df(x_pred)

k určení nového bodu tedy potřebujeme jen jeden předchozí bod namísto dvou
20) k určení chyby ale opět použijeme oba body

D) Müllerova metoda
teo: http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf stana 7
podívejte se také na text: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/mullerovamet.pdf

skript: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/muller.m

Chceme tedy najít kořeny nějakého polynomu vyššího řádu
v našem případě f=4*x^3 - 2*x^2 - 4*x - 3;
11-17) odhadneme 3 body (x1,x2,x3 → a,b,c) tak aby byl třetí bod nejblíže předpokládaného kořene
19) budeme iterativně zpřesňovat dokud nedosáhneme dostatečné přesnosti (chceme kořen → absolutně přesně by bylo pro y(kořen) = 0)

připravíme si vzorečky z http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/mullerovamet.pdf
xn , xn0 jsou +- větve → dva kořeny
chceme vybrat kořen z větve blíže x3 → řádky 29-34)

36-41) zahodíme starý x1 a vše ostatní posuneme

E) Důkaz ohraničení absolutní hodnoty kořenů polynomu shora
viz kfe.fjfi.cvut.cz/~matysma4/nme/cv07/omezenikorenu.pdf

F) Soustavy nelineárních rovnic:
Metoda prosté iterace
teo: http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf strana 9

skript: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/prostaiterace.m

máme nějakou 2D soustavu
f1(x,y)=0
f2(x,y)=0

přepíšeme do tovaru
x = phi1(x,y)
y = phi2(x,y)

Chceme jenom *x* a *y* → nezajímají nás *x*^2,*y*^2 atd…
pro phi1 čapneme z f1 člen s *x* a vše ostatní hodíme na druhou stranu
pro phi2 čapneme z f2 člen s *y* a vše ostatní hodíme na druhou stranu
takže:

máme f1 = x^2 + 4*x - y^2 - 2*y -1; =0
→ - 4*x = x^2 - y^2 - 2*y -1
x = - (x^2 - y^2 - 2*y -1)/4
x = (y^2 + 2*y +1 - x^2)/4
a to provoláme phi1

máme f2=x^2 + 5*y -4; =0
→ y = - (x^2-4) / 5
y = (4 - x^2) / 5
a to provoláme phi2

23,24) zadáme počáteční nepřesný odhadneme
33-36) zpřesňujeme pomocí vzorce [x,y]=[phi1(x,y),phi2(x,y)] z přednášky

Newton–Raphsonova metoda pro systém nelineárních rovnic
teo: http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf strana 10
skript http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/newtonraphson.m

Máme zadány stejné f1 a f2 jako v předchozím případě.

Na rozdíl od Newton–Raphsonovy metody pro jednu rovnici nepotřebujeme určit jen jednu derivaci ale celý Jakobián →
6-12) Připravíme si numerický výpočet derivací podle *x* a *y* pomocí dopředné diference (viz cvičení 2 http://kfe.fjfi.cvut.cz/~matysma4/nme/cv02/priklad21.pdf)
+-*h* ve schématu se přiřadí k *x*,*y* podle druhu derivace

35-38) připravíme si matici *a* ve formě Jakobiánu
prvním parametrem předáváme celou funkci → musíme použít function_handle @
(anebo možnost použití anonymních funkcí jako v cvičení 6 v http://kfe.fjfi.cvut.cz/~matysma4/nme/cv06/chebyshev.m)

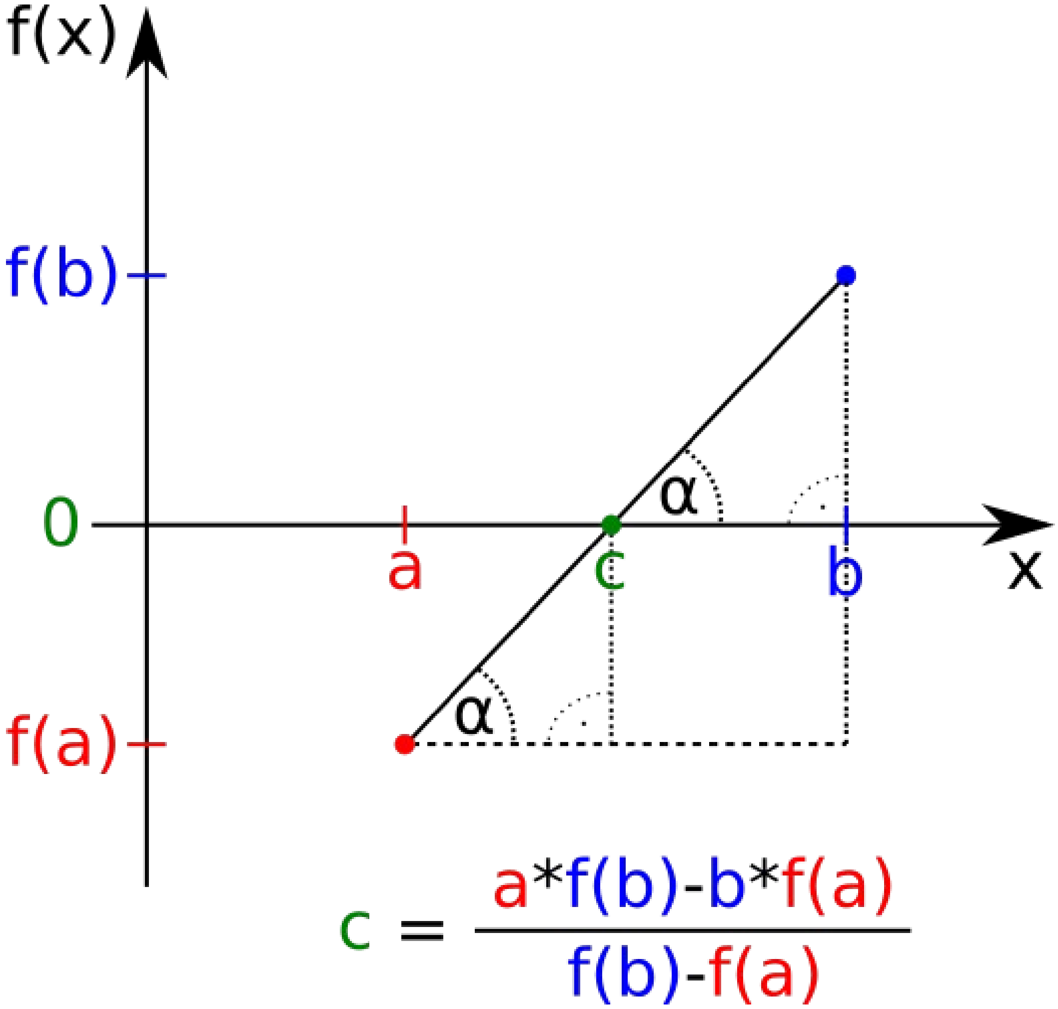
40-41) připravíme si pravou stranu *b*

Ze soustavy nelineárních rovnic jsme po použití Taylorova rozvoje atd. Dostali soustavu lineárních rovnic kterou umíme řešit.
43) Tady pro jednoduchost opět použijeme Matlabovský řešič soustavy lineárních rovnic mldivide (\) jinak řešení pomocí LA metod viz cvičení 3

45-46) o krok se posuneme a opakujeme

Úkoly: 2 části a,b)

Úkol a)
Odvoďte vzorec pro použití sekantové metody z přiloženého obrázku:



Hint: použijte třeba rovnosti pro kotangens(α) či jinou goniometrickou funkci
Stačí opět na papír a vyfotit / oskenovat

Úkol b)
upravte skript: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv07/secny.m

tak aby se použila metoda regula falsi
viz strana 4 v http://kfe.fjfi.cvut.cz/~limpouch/numet/nr.pdf
(výběr nového bodu z předchozích dvou bodů)

Deadline na odevzdání 19.4. 2020