Dobrý den, dnes si projdeme hledání extrémů funkce.

Úkol na tento týden na konci textu. Deadline na odevzdání 3.5. 2020

Teorie ke cvičení z přednášek: http://kfe.fjfi.cvut.cz/~limpouch/numet/ext1.pdf strany 1-4

Pro začátek budeme hledat extrém funkce jedné proměnné.
Budeme hledat minimum, to ohraničíme a budeme se k němu přibližovat podobně jako v minulém cvičení kde jsme hledali kořen.

K vysvětlení zlatého řezu / Fibonacciho posloupnosti a jejich uplatnění v přírodě se mi líbí interaktivní stránky: https://www.mathsisfun.com/numbers/nature-golden-ratio-fibonacci.html
Projděte si a zkuste si v appce jak funguje vyplňování květu slunečnice buňkami. Pozor appka bere čísla s desetinnou tečkou (i když čárka je v příkladech v nadpisu appky). Zkuste si různé možnosti např. 0.25 , 0.33 , 0.6, 0. 61, a přibližujte se postupně k 0.6182... .
Slunečnice bude fungovat i pro dopněk čísla do 1 tj. 0.3818. Na celém číslu zde nezáleží, funguje i 1.6182 , 1.3818, 2.6182, 2.3818 ... Vždy uděláme novou buňku otočíme se a uděláme další (takže celé otočky se neprojeví).

Vlastní zlatý řez je definován jako poměr dvou úseček / stran obdelníku:
a/b == a+b/a
delší ku kratší == součet / delší
viz
https://www.mathsisfun.com/numbers/golden-ratio.html

v apce si můžete zkusit pro jaký poměr tohle platí.

Pro interval <0,1> tj. Úsečku o délce 1 dostaneme zlatý řez pokud ho rozdělíme na dvě úsečky ~0.618 a ~0.382, které odpovídají i našim otočkám ve vytváření květu slunečnice.

Matematické odvození je také na straně 2 v přednášce
http://kfe.fjfi.cvut.cz/~limpouch/numet/ext1.pdf

Tyto poměry (~0.382 a 1 - ~0.382 = ~0.618) použijeme v následujícím programu:
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv08/zlatyrez.m

Zkusíme to pro funkci -sin(x) které má minimum v pi/2 = ~ 1.5708

Pro ohraničení extrému musíme znát hodnoty ve 3 bodech (pro kořen stačili 2)
chceme a<b<c aby f(a)>f(b)<f(c) a pro upřesnění budeme potřebovat 4tý bod

počáteční odhad dáme na interval <0,pi>

13) Zvolíme si přesnost
16) Pro přesnější odhad poměru použijeme vzoreček z přednášky na straně 2
19) Dosadíme bod b tak aby dělil úsečku <a,c> (c-a) v poměru zlatého řezu.

22) while 1 → nenulová číselná hodnota supluje True (funguje i v mnoha ostatních jazycích) mohli bychom dát třeba i while 5. Kdybychom dali while 0 tak neproběhne nikdy.
- takhle vždycky True a cyklus přerušíme až pomocí *break* na řádku 24)

23-25) Když se přiblížíme na přesnost tak skončíme cyklus (jinak by běžel až do násilného vypnutí programu/počítače)

27-36) chtěli bychom přidat jeden bod do 0.382 a druhý do (1 - ~0.382 = ~0.618)
27) – pokud je b v první půlce intervalu <a,c> tak čtvrtý bod (*x*) dáme do druhé poloviny (~0.618)
31) → pokud je b ve druhé polovině, tak:
34) b nazveme x
35) b napočteme znovu do první půlky intervalu (~0.382)

- chceme zachovat a<b<x<c → proto tyto podmínky

38) otestujeme jestli je minimum na <a,x> (tj. Pro a<b<x platí f(a) >= f(b) <= f(x))
39) pokud ano tak zkrátíme interval z prava: <a, c> → <a,x>
→ c=x; takže máme do další iterace a<b<c a můžeme pokračovat dál

40) jinak platí druhá možnost a minimum (pokud existuje) je na intervalu <b, c>
zkrátíme zleva <a, c> → <b,c>
→ potřebujeme do další iterace přepis na *a<b<c*
41) uložíme *b* do pomocné proměnné *temp*
42) změníme *b*
43) do *a* uložíme původní *b*

45) v každé iteraci vypíšeme nový interval <a,c> na kterém hledáme minimum a který postupně zmenšujeme

---

Minimum té samé funkce zkusíme najít pomocí Parabolické interpolace
teo: http://kfe.fjfi.cvut.cz/~limpouch/numet/ext1.pdf strana 3
Chceme body proložit parabolou, tu zderivovat a využít toho že v minimum je derivace rovna nule.

Odhad minima pomocí interpolace Lagrangeovým kvadratickým polynomem – viz teo.
Dosadíme rovnici za *x* na řádek 24)

10) odhadneme minimum – můžeme zvolit náhodně mezi *b* a *c*

27-43) chceme opět do další iterace předat *a<b<c*
obdobně jako v předchizím kódu řešíme posun levého / pravého okraje a následné zarovnání a předání struktury *a<b<c*
- viz komentáře v kódu

Hledání minima funkce více proměnných
A) projděte si text k metodě konjugovaných gradientů:
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv08/konjugsmergrad.pdf

B) Simplexová metoda (amoeba)
chceme sledovat úbytek/nárůst funkce f(x,y,z...)
Použijeme N+1 simplex v N rozměrném prostoru → takže ve 2D chceme 3 body → trojúhelník

teo: http://kfe.fjfi.cvut.cz/~limpouch/numet/ext1.pdf strana 4
- viz obrázek: budeme provádět 4 základní operace: reflexi (překlopení trojúhelníku), zvětšení, zmenšení v jednom směru a popřípadě zmenšení ve všech směrech směrem k minimu
článek na wiki s animacemi: https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method

skript: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv08/simplexm.m

profil hledané funkce si můžeme vykreslit třeba ve wolframu:
https://www.wolframalpha.com/input/?i=100*%28x1-x2%5E2%29%5E2%2B%281-x1%29%5E2

9-11) potřebujeme tedy na začátku nastavit koeficienty pro reflexi (jen překlopíme do stejné vzdálenosti → 1), zvětšení a zmenšení

16) odhadneme počáteční body simplexu. Tady simplex ve 2d prostoru máme *x1*, *x2*
19-21)vypočteme funkční hodnoty v těchto bodech
25-39) chceme jen zarovnat body zasebou abychom měli y(1) <y(2) <y(3)
funkce *deal* je zde jen pro redistribuci ve strukturách tj.
[x(2,1), x(3,1)] = deal(x(3,1), x(2,1));
dá do x(2,1) =x(3,1)
a do x(3,1) původní x(2,1)
tj. nemusíme použít pomocné *temp* jako v metodě zlatého řezu

v bodu 3 je nejvyšší hodnota funkce z daných třech bodů. Předpokládáme růst funkce ve směru tohodle rohu trojúhelníku → uděláme reflexi do opačného směru (kam předpokládáme pokles funkce)

46-51) spočteme těžiště všech ostatních bodů kromě bodu s nejvyšší hodnotou
-tj. Ve 2d polovinu mezi body 1 a 2 (vynecháme bod 3)
- bod *c* jsme si definovali jako dvouprvkový kontejner na řádku 13)

54-56) provedeme reflexi oproti těžišti → bod *xt*
na druhou stranu od bodu 3 → proto je zde na řádku 55) -x(3,i)
57) spočteme funkční hodnotu v tomto bodě

60-77) zkontrolujeme jestli jsme si reflexí pomohli, tj. Jestli v tomto směru je opravdu pokles funkce, tj. jestli yt<y(3)

a) pokud ano tak nahradíme bod 3 novým bodem → máme simplex v obráceném směru
 -61) tady jde *i* přes 1:2 protože máme 2d prostor
 - chceme najít směr nejnižšího poklesu
 -65) pokud je hodnota nižší než v bodě 1 (nejnižší v původním simplexu) tak bychom chtěli v tomto směru pokračovat → protáhneme trojúhelníku
 - název celé operace i s reflexí → reflexe se zvětšením
 -67) v kódu už jsme předtím reflexi provedli a jenom protahujeme → je zde opačné znaménko než v řádku 55) (tj. xt(i)-c(i) namísto c(i)-x(3,i))
 -70) zkontrolujeme jestli trend poklesu v tomto směru pokračuje. Tj. Jestli je hodnota po reflexi se zvětšením menší než hodnota jen při samotné reflexi. Pokud ano tak dáme za bod 3 nový bod.

b) pokud ne tak necháme původní a zkusíme zmenšovat

79) Zkusíme jestli je ozrcadlený bod (z původního bodu 3 oproti těžišti na druhou stranu) větší než bod 2 → nevýhodný směr → chceme zmenšit v celé této „rovině“
tak provedeme naopak zmenšení oproti těžišti
-81) oproti řádku 55) (reflexi) je zde opět obrácené znaménko (tj. x(3,i)-c(i) namísto c(i)-x(3,i))
a oproti řádku 67) (zvětšení) x(3,i) namísto xt(i)
→ zmenšujeme tedy „původní“ trojúhelník a né ten reflektovaný

84) zkoumáme jestli je nově získaný bod nižší než nejvyšší (bod 3)
a) ano → tak ho za něj dosadíme (tj zmenšili jsme trojúhelník v nevýhodném směru → předpokládáme že jsme se zbavili části vysokých bodů)
b) ne → chceme zmenšit trojúhelník směrem k minimu (bodu 1)

92) obdobné řádku 81 ale neprovede zmenšení vůči bodu těžiště *c*, ale vůči bodu 1
→ x(i,j) = x(1,j)...
for cyklus přes i nám dosadí body 2 a 3
for cyklus přes j pro dvě složky bodů v našem 2d prostoru
- tím provedeme zmenšení v obou zbývajících rozích trojúhelníku (body 2,3) směrem k minimu (bod 1)

Metoda je jednoduchá na popsání, ale bez nakreslení se to tak nemusí zdát.
To nás přivádí k úkolu na tento týden:

Úkol:
ve skriptu na simplexovou metodu: http://kfe.fjfi.cvut.cz/~matysma4/nme/cv08/simplexm.m
Vytvořte animaci popisující jednotlivé úkony: zvolení těžiště, reflexii, zvětšení, zmenšení v jednom směru, zmenšení ve všech směrech směrem k minimu, záměny bodů (řádky 24-39 shrňte jedním obrázkem s titulkem přeuspořádání bodů)
V obrázku popište jaká operace se provedla. Krom bodů 1,2 a 3 vyznačte (kde to bude vhodné) i body *xt*, *xe* a *xc* . Popište i vyhodnocení podmínek, kde je to vhodné přidejte i else větev k if podmínce pro případy kdy if neproběhne, tak ať se vypíše hláška do obrázku.

Hint: možné řešení:
využijte a obměňte následující kus kódu:

```
[x(4,1), x(4,2)] = deal(x(1,1), x(1,2));
plot(x(1,1),x(1,2),'ro',x(2,1),x(2,2),'bo',x(3,1),x(3,2),'go', ...
    c(1),c(2),'mo',x(:,1),x(:,2),'k','MarkerSize',12)
title('popis akce')
legend('bod 1','bod 2','bod 3','teziste','Location','northeastoutside')
waitforbuttonpress
```

A změňte popis akce v *title* na co potřebujete: výběr těžiště / reflexe / zmenšení / zvětšení/ podmínka ano/ne záměna bodů atd.

Například vhodné nakopírovat na řádek 52 a získáte animaci postupného hledání extrému, pozice bodů a těžiště.

Na další obrázek se skočí zmáčknutím libovolného tlačítka
vit funkce *waitforbuttonpress* ze cvičení 5

řádek: [x(4,1), x(4,2)] = deal(x(1,1), x(1,2)); je přidán jen pro uzavření trojúhelníku v obrázku

Dejte obdobné kusy kódu na další vhodná místa viz zadání.
Vyplňte si to hlavně tak aby vám to pomohlo pochopit průběh metody.

Deadline na odevzdání 3.5. 2020