Dobrý den
dneska uděláme první část aproximací, jmenovitě interpolace.
Zadání úkolu na tento týden je na konci textu, deadline 5. 4. 2020.

Langrangeův interpolační polynom.
Teorie z přednášek:
http://kfe.fjfi.cvut.cz/~limpouch/numet/aprox.pdf

Dovysvětlující text:
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/lintpol.pdf

Máme body $x_0,x_1,x_2 \ldots\ldots x_n$ (tj. n+1 bodů)
a chceme je proložit polynomem stupně nejvýše n abychom prošly všemi těmito body.

Skript:
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/lagrange.m
Skript má 4 části:
4-10) připravíme si matlabovskou funkci která zavolá funkci kterou bychom chtěli interpolovat (zkuste si pak odkomentovat ostatní funkce a spustit).

12-22) vlastní funkce počítající Lagrangeův polynom.
K té se vrátíme při použití

25-29) Nastavení parametrů
- jsme v oblasti <-10 ; 10> a známe tam 1001 bodů. Nebudeme interpolovat polynomem stupně 1000 → výpočetně náročné, ale zkusíme polynomy do stupně n=26. Polynom tedy bude procházet postupně jedním bodem, dvěma body, třemi body atd. S postupnou zvětšující se přesností.

31-72) Hlavní for cyklus kde postupně zkusíme funkci interpolovat s interpolačním polynomem vzrůstajícího stupně.
-Tento skript využívá *waitforbuttonpress* na řádku 70, takže po spuštění klikejte do obrázku / mačkejte libovolnou klávesu a budete mít animaci vývoje interpolace když budeme zvyšovat stupeň interpolačního polynomu.

32) Vybereme vhodné reprezentativní body v ekvidistantní vzdálenosti. V prvním kroku chceme jeden bodu v půlce oblasti tj. v (b-a)/2, pak bychom chtěli 2 body v 1/3 a 2/3, pak 3 body v 1/4,2/4, 3/4 atd.

34-39) Připravíme si ekvidistantní body v oblasti, které budeme používat a korespondující hodnoty funkce v těchto bodech.

41-44) Budeme chtít interpolační polynom který projde všemi vybranými body, chceme ale i jeho hodnoty ve všech 1001 zadaných bodech, tj. I tam kde neprochází.

Na řádku 43 voláme eL(k) = Lagr(cx(k),xx,val,i); Proto je až zde dobré dovysvětlit řádky 12-22)
Máme 1001 bodů v oblasti <10,10> uložené v cx = a : (b-a)/1000 : b
Projdeme všechny tyto body přes k=1:1001
Do funkce Lagr dáme parametry:
p1 -> cx(k) → tj. Daný bod ve kterém chceme interpolaci udělat
p2 → xx → vektor vybraných *n* bodů kterými prochází L. polynom; řádek
p3 → val → vektor hodnot funkce v těchto bodech
p4 → i → stupeň interpolačního polynomu (velký for cyklus 31-72)

Zpátky do řádků 12-22)
Chceme součet jednotlivých polynomů v jednom bodu x1. → chceme sumu → standardní postup z předchozích cvik, připravíme a vynulujeme proměnnou kterou dále navyšujeme.
tj. 13) y=0;
14-21) for cyklus kde přičteme všechny polynomy
15) v jednom bodě jednotlivý polynom prochází a v ostatním je 0 → vybereme ty ostatní body
logická operace ~= nerovná se
v matlabu lze provézt takhle vektorově
zkuste si třeba porovnat příkazem
1:5~=2 → vybere jen prvky kde se nerovná 2
20) vlastní výpočet polynomu → přepsání rovnice Obecný tvar ze strany 2 dokumentu
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/lintpol.pdf

zde jsme použili funkci produkt *prod* pro přenásobení jednotlivých prvků daného vektoru
zkuste si třeba
a = 1
b=2:4
prod(b-a) → (2-1)*(3-1)*(4-1)= 1*2*3=6

57-70) Už jen připravíme vykreslení.
Připravíme si 2 grafy – subplot(2,1,1) a subplot(2,1,1) → říkáme že chceme 2 řádky, 1 sloupec a pak je index jednotlivého obrázku

V prvním grafu subplot(2,1,1) jen vykreslíme původní funkci a naši interpolaci (červeně) zavoláme hold on / hold off aby se nám nepřepsal předchozí plot a uděláme kolečka pro body které jsme použili pro interpolační polynom – řádek 61

V druhém grafu vykreslíme chybu naší interpolace v jednotlivých bodech: *funkce(cx)-eL*
vypíšeme do obrázku také normu rozdílu pro celou oblast *norm(funkce(cx2)-eL2)*

Jak již bylo zmíněno funkcí *waitforbuttonpress*; ve for cyklu si můžete udělat jednoduchou animaci kde program čeká na vaše kliknutí

Kdyby náhodou nešlo vypnout tak v Matlabu funguje CTRL+ C při běžícím programu jako příkaz pro ukončení programu

přikazem *close* (*close all*) můžete zavřít (všechna) okna grafů.

Vidíme že pro takhle pěknou funkci nám to celkem hezky aproximuje už polynom stupně 5.
Zkuste si i ostatní funkce → postupně zakomentujte a odkomentujte před připravené funkce na řádcích 5-9 (nebo si můžete vymyslet vlastní…).

Dané funkce reprezentují některé problematické případy, zkuste si rozmyslet proč jsou jednotlivé případy problematické a použití tohoto typu aproximace není pro ně moc vhodné.

Doporučuju opět i video od Dominiky Mašlárové na tvorbu jednoduchého Lagrangeova polynomu:
https://www.youtube.com/watch?v=Sg9d-pg1BWQ

Můžete se i podívat na naprogramování Nevillova algoritmu, který standardně na cvičeních nedělám, ale znáte ho z přednášky.
https://www.youtube.com/watch?v=nFNhnpuQQvo

Ukázali jsme si, že použití Lagrangeova polynomu je vhodné jen pro některé funkce a malý počet bodů. Uděláme si tedy ještě interpolace funkce po částech. Tj. chceme odhadnout hodnotu pomocí dvou (či více) nejbližších okolních známých bodů.

Přečtěte si prosím připravený text:
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/kubspline.pdf

Dále si projdeme skript:
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/kubspline.m

Uložte si do stejné složky jako skript i vstupní data:
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/spline.dat

Máme zadaných 20 bodů a chtěli bychom udělat interpolaci na dalších 100 bodů
→ řádky 3 – 9)

14-22) → připravíme si prázdné kontejnery
- z toho *rho* a *mu* použijeme pro výpočet soustavy s tridiagonální matici viz cvičení č. 3

24-30) → přepis úplně poslední rovnice na straně 3 v souboru :
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/kubspline.pdf

f odpovídá pravé straně, pořadí *c*,*a*,*b* je naschvál, tento zápis odpovídá tridiagonální matici.
V zápisu pak odpovídá vůči bodu *x(i)* kde se požaduje spojitost první derivace:

x1 → x(i)
x0 je o bod míň tj. x(i-1)
x2 je o bod dále, tj x(i+1)
obdobně pro y

V Matlabu … slouží pro pokračování v příkazu na další řádek 3 tečky ... to je použito u f na řádkách 28-29)

32-39) je použití zpětného běhu pro řešení této soustavy s tridiagonální maticí
→ to jsme si ukázali ve 3tím cvičení, tak to zkopírujeme ze souboru
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv03/tridiagmat.m

Teď máme pro zadané body: *x*,*y* a *y*‘‘, takže všechno co potřebujeme pro spline a chceme ho udělat pro bodu = 100;
Chtěli bychom na 100 bodech mezi *x1* a *xn* (interpolujeme → musíme zůstat v známém rozsahu), takže si připravíme krok pro ekvidistantní body → řádek 45

46-47) připravíme si ty body a kontejner pro funkční hodnoty které hledáme pomocí interpolace.

Bacha:
x → známé zadané body 20 -> indexujeme přes j
xn → neznámé body které získáváme interpolací 100 → indexujeme přes i

51-57) Chceme najít nejbližší *x(j)* body okolo bodu *xn(i)* který hledáme
takže jdeme od prvního *x* bodu doprava dokud je menší než *xn(i)* který hledáme
53) ukončíme while cyklus kdybychom došli nakonec

55) skočí o jeden doprava, v dalším kole cyklu ale zjistíme že jsme už za bodem, skončí while cyklus tak se na řádku 57) vrátíme o jeden doleva →
index j odpovídá nejbližšímu bodu nalevo od *xn(i)* ,které hledáme a index j+1 bodu napravo
tj. x(j+1) je okrajový bod x2 a *x(j)* okrajový bod x1 v rovnicích 3a) 3b)
a naše *xn(i)* odpovídá hledanému x v týž rovnicích

60-63) pak pouze přepíšeme tyto rovnice (3a,3b) podle stejného klíče

dále vypočteme hodnotu interpolačního splinu 66) podle rovnice 2) kde *y2* jsou naše druhé derivace které jsme hledali z tridiagonální matice.

70 – 73) vykreslíme celou interpolaci a červenými křížky zadané body.
Poslední co si teď probereme je aproximace derivace - odvození aproximace metodou vyššího řádu přesnosti

Pokud jste si stáhli soubory dopředu stáhněte si prosím znovu soubor
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/aproxder_b.pdf
verzi b, opravil jsem tam jednu chybějící 2ku.

Text si prosím projděte je to celkem přímočaré ale nebojte se zeptat kdyby v tom byli nějaké nejasnosti, kdyžtak si to můžeme projít i nějak hromadně přes videokonferenci.

Jen možná připomenu co se děje v přechodu z rovnice 1 na rovnici 2. Funguje tu takový to porovnávání koeficientů. Máme koeficienty (f_0,f_0‘,f_0‘‘,f_0‘‘‘,f_0‘‘‘‘) takže v rovnici 1) na levé straně

0 * f_0 + 1 * f_0‘ + 0 * f_0‘‘ + 0 * f_0‘‘‘ + 0 *f_0‘‘‘‘

a na pravé straně jsme to uspořádali abychom měli taky

= [...]*f_0 + [...]*f_0‘ + [...]*f_0‘‘ + [...]*f_0‘‘‘ + [...]*f_0‘‘‘‘

kde v každém […] máme kombinaci ABCDE kterou pak v rovnici 2 zapíšeme do řádků do matice a chceme řešit soustavu pro neznámé ABCDE.
Pro koeficient f_0‘ máme v rovnici 1) ještě před […] přenásobeno *h proto to převedeme na druhou stranu a v pravé straně rovnice 2) pak máme na druhé pozici 1/h.

Pro druhou derivaci to pak funguje stejně ale řešíme soustavu pro f_0‘‘ která má před sebou v rovnici 1) h^2 /2

Tímto postupem jsme získali diferenciální schéma pro aproximaci derivace. Výslednou rovnici (pod rovnicí 2)) pak můžeme implementovat například do skriptu ze druhého cvičení kde jsme si ukazovali metody prvního a druhého řádu
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv02/chyba_metody.m

řešení je pak na
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/derivace.m

jmenovitě řádky 20-22) kde f(x) odpovídá f0 v textu (které tam není) a poté
f(x-2h) → f-2
f(x-h) → f-1
f(x+h) → f1
f(x+2h) → f2

Úkol na tento týden – zopakujte si Taylorův rozvoj, který se používá pro odvozování těchto schémat a určení řádu metody. Většina z vás ho nějak bude mít na zkoušce a je potřeba abyste ho uměli používat.

Opište na papír obecný vzorec Taylorova rozvoje f(x+h) ze souboru
http://kfe.fjfi.cvut.cz/~matysma4/nme/cv05/aproxder_b.pdf

a poté napište rozvoje pro:
f(x + 1/2 * h)
f(x - 1/2 * h)
f(x - 3 * h)
Se zanedbáním členů s O(h^5) stejně jako v textu, s použitím faktoriálů atd.
Můžete si to zkontrolovat třeba na https://www.wolframalpha.com/
(kde vám to ale rozepíše faktoriály), ale zkuste si to sami, ať to umíte používat, bude se vám to ještě hodit.

To mi prosím **vyfoťte/naskenujte a pošlete** opět na NMEcvMM@seznam.cz
s cv5 a vaším příjmením v předmětu.
Deadline je příští neděle **5. 4. 2020**. Připomínám, že za včasné odevzdání je vám uznána docházka za dané cvičení a za včasné odevzdání všech úkolů vám bude odpuštěna velká závěrečná úloha. Využijte toho prosím.