Milí studenti,

děkuji vám za vložené elaboráty. Projděte si moje odpovědi.

S pozdravem,

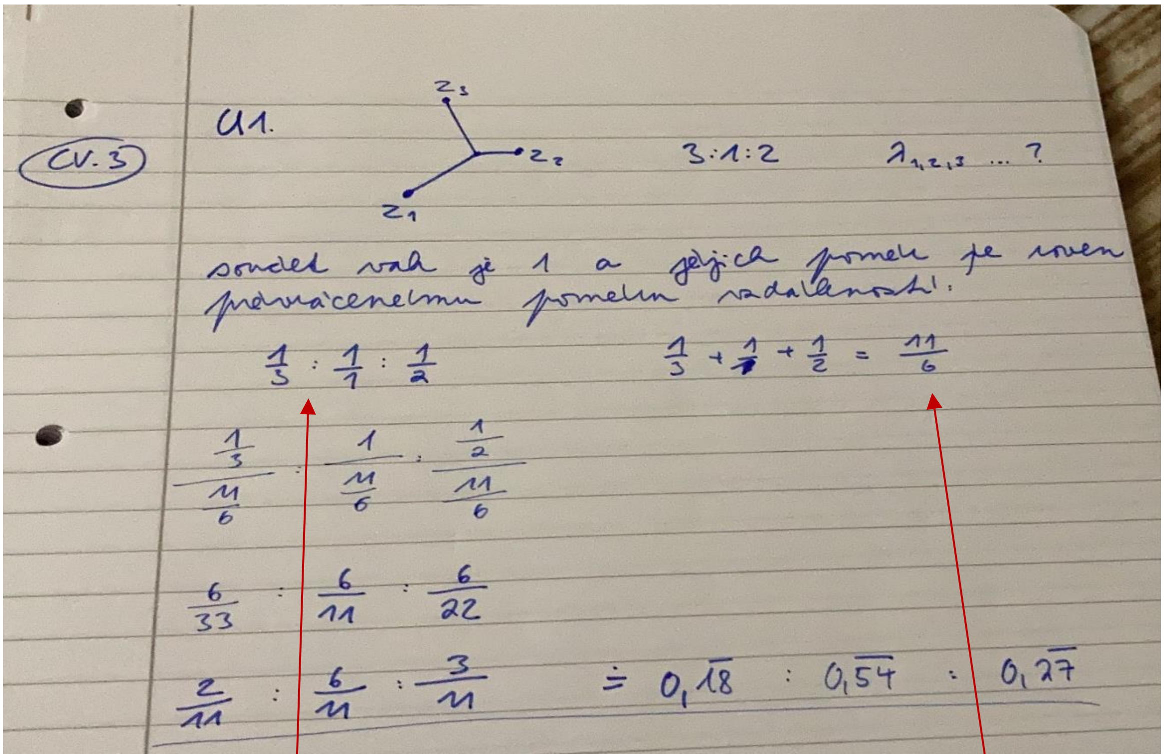
JJ

Úkol U1: Vypočtěte (tužka -papír) tyto váhy, pro případ, že by poměr vzdáleností byl přesně 3:1:2.

Úlohu lze řešit úvahou nebo podle vzorce. Kolegyně založila svůj postup na té úvaze. Napsala

... součet vah je 1 a jejich poměr je roven převrácenému poměru vzdáleností

a podle toho provedla odpovídající výpočet. Váhy nejprve navrhla jako 1/*r* a zjistila jejich součet. Touto hodnotou je pak normalizovala.



To je správně, ovšem v zadání geost_3.pdf bylo uvedeno „..., kde interpolujeme pomocí metody IDW (s mocninou 2)“, takže jsem předpokládal, že použijete mocninu 2. V uvedeném postupu by se tedy mělo objevit

$$\frac{1}{3^2} : \frac{1}{1^2} : \frac{1}{2^2} \qquad \frac{1}{9} + \frac{1}{1} + \frac{1}{4} = \frac{4+36+9}{36} = \frac{49}{36}$$

a dále už je to analogické. Váhy vyjdou 4/49 , 36/49, 9/49.

Jiní provedli výpočet podle vzorce (2.4) GPI

U1

$$\lambda_1 = \frac{\frac{1}{3^2}}{\frac{1}{3^2}+\frac{1}{1^2}+\frac{1}{2^2}} = \frac{1/9}{1/9+1/1+1/4} = \frac{4}{49}$$

$$\lambda_2 = \frac{\frac{1}{1^2}}{\frac{1}{3^2}+\frac{1}{1^2}+\frac{1}{2^2}} = \frac{\frac{1}{1}}{\frac{49}{36}} = \frac{36}{49}$$

$$\lambda_3 = \frac{1/2^2}{\frac{49}{36}} = \frac{9}{49}$$

$$\underline{\underline{1}}$$

U2: Ačkoliv se jedná o plošná data (2D), lze příslušné váhy vypočíst pomocí matlabovských funkcí pro data na ose (1D, `intpol_IDW_1b` popř. `intpol_IDW`), které jsem Vám zaslal minule. Zkuste přijít na to, jak, a napište to do elaborátu.

Šlo o to, jak chytře umístit data na osu x, aby byly vzdálenosti od počítaného bodu 3,1,2. Přišli jste většinou na to, že nejjednodušší je umístit počítaný bod do počátku (tj. x=0) a zadat datové body do 3,1 a 2 (tj. xd=[3 1 2]). Volání funkce intpol_IDW pak mohlo být

>> intpol_IDW([3 1 2],[10 20 30],0,2)

lam = 0.0816 0.7347 0.1837

Hodnoty dat jsem zvolil z=[10 20 30], ale jsou libovolné, neboť nám jde jen o váhy.

Lze vymyslet i jiné konfigurace datových bodů, např. může být první bod vlevo od počátku, xd(1)=-3

>> intpol_IDW([-3 1 2],[10 20 30],0,2)

Příkladem jsme ilustrovali, že metoda IDW uvažuje pouze vzdálenosti k počítanému bodu, nikoliv vzájemnou konfiguraci datových bodů, jejich konkrétní rozložení („geometrii vzorkování“).

U3: Doplňte trojici datových bodů o čtvrtý bod, který je poblíž třetího a jehož vzdálenost od počítaného bodu je 2, podobně jako obr. 2.10 v GPI, a zopakujte výpočet vah. Váhy porovnejte s předchozím případem U1. Vyslovte svůj názor.

Někdo to počítal na papíře ale většinou jste využili volání funkce obdobně jako v předchozí úloze. Například můžeme umístit 4. bod prakticky na stejné místo jako 3. bod a dát mu mírně odlišnou hodnotu (na čemž vlastně při výpočtu vah nezáleží). Volání funkce pak může být třeba

>> intpol_IDW([3 1 2 2],[10 20 30 31],0,2)
lam = 0.0690 0.6207 0.1552 0.1552
neboli
lam= 2/29 18/29 9/58 9/58

Z vašich názorů vyjímám

Přidání bodu způsobí obecně pokles všech vah, ale pokud přidaný bod bude blízko jinému, tak součet těchto vah bude vyšší. Díky tomu budou mít větší vliv na zjišťovanou výslednou hodnotu bodu.

Z toho vyvozuji, že když je někde shluk o podobných hodnotách, jejich váha se zvýší tolikrát, kolik měření tam máme, a ovlivňuje vypočtenou hodnotu třeba i o dost více než „osamělá“ měření v blízkosti interpolovaného bodu.

Ještě moje poznámka: Metoda IDW má tedy problém se shluky bodů. Přitom ten shluk může vzniknout v dobré víře, že nějaká lokalita je zajímavá a tudíž je třeba ji více proměřit :)

U4: Výsledný kód mi pošlete.

Po pravdě úplně nevím, jak se to má udělat. Mám v tom trochu zmatek. Moc prosím, jestli byste to mohl v odpovědích rozebrat trošku podrobněji, případně i s příklady. Děkuji

Je mi jasné, že při našem způsobu výuky zákonitě vzniknou situace, kdy ne úplně všichni budou něco hned umět udělat nebo to mít dobře. To se pak ale společně opraví v mojí odpovědi. Zrovna na tomto místě jsem nějaké problémy předpokládal.

Takže se pojďme teď na ten úkol podívat. Začalo to kousek před U4, kde bylo uvedeno:

Při řešení příkladu vás patrně napadlo, že by se hodila funkce, která umí IDW ve 2D. O to se teď pokusíme. Resp. nejprve se pokusíte sami.

Otevřete funkci `intpol_IDW_1b`, uložte ji pod názvem `intpol_IDW_2D_1b` (save as) a zkuste ji doplnit o další souřadnici (yd, y) a změnit příslušným způsobem její příkazy tam, kde je třeba.
Pokud byste potřebovali někam dát odmocninu, tak je to sqrt(...). Název této funkce je odvozen od „square root“.

Moje přání bylo, abyste si uvědomili, že ve 2D musíme použít pro euklidovskou vzdálenost namísto absolutní hodnoty Pythagorovu větu a odmocninu (což jsem naznačil). Takže ta funkce pro 1 bod měla vypadat takto (toto poslala naše kolegyně, jen jsem vyznačil, kde se něco měnilo oproti 1D)

```
function z=intpol_IDW_2D_1b(xd,yd,zd,x,y,a)

nd=length(xd);

for i=1:nd
   r=sqrt((xd(i)-x)^2+(yd(i)-y)^2); % pythagorka
   if r==0  %jestli x = 0, z se vlastne rovná zd
       z=zd(i);
       return
   else
       lam(i)=1/r^a; % vaha pocitana podle vzorecku
   end
end

lam=lam/sum(lam)

z=lam*zd' % finalni vypocet z, zname jeho vahu, nasobime vektory, takze prevod na
sloupcovy vektor'
```

A analogicky se měla změnit i funkce `intpol_IDW_2D`, kterou jste nejprve zkoušeli vyrobit sami a kdo neuspěl si možná vzali tu, co jsem poslal.

Pokud se ovšem někomu funkce pro jeden bod `intpol_IDW_2D_1b` nepovedla, nemohl to celé zkompletovat.

Každopádně, nyní máme kompletní a fungující obě funkce a jdeme dále (využijeme je i v další lekci).

U5: Výsledný kód, volání funkce z příkazové řádky i výsledek mi vložte do elaborátu.

S tím jste si jakžtakž poradili, s drobnými problémy, které nebudu komentovat, protože tuto funkci pro 2D spustíme na začátku další lekce. Takže uvidíte. Za zmínku ale stojí následující příspěvek.

Kolega poslal tento kód, kde vidím záludnou chybu (vyznačeno)

```
function intpol_IDW_2D_moje(xd,yd,zd,x,y,a)

% interpolace metodou IDW ve vice bodech

figure
plot3(xd,yd,zd,'o')
hold on

n=length(x)

for i=1:n
    z(i)=intpol_IDW_2D_moje(xd,yd,zd,x(i),y(i),a);
    plot(x(i),z(i),'r.')
    pause(0.1)
end

plot3(xd,yd,zd,'o')
```

Namísto vyvolání funkce pro jeden bod `intpol_IDW_2D_1b` je uvedeno jméno volající funkce. Funkce `intpol_IDW_2D_moje` tedy volá (pokouší se spustit) sama sebe. Na to pozor.

(Další chyba, drobnost - příkaz plot by se měl nahradit plot3.)

U6: Zkuste přijít na to, jak zahustit síť (zjemnit vykreslenou plochu) a tento obrázek mi pošlete.

Kolegyně píše

Hustší mřížku jsem zajistila pomocí úpravy příkazu, přidáním kroku, po jakém se bude interpolovat

```
intpol_IDW_2D([-2 1 0],[0 0 2],[10 20 30],[-5:0.5:3],[-2:0.5:4],2)
```

Tím jsme si, doufám, ujasnili, že příkaz typu 1:10 vytvoří sekvenci s krokem 1 (1,2,...,10), kdežto příkaz 1:0.1:10 vytvoří sekvenci s krokem 0.1 (tj. 1, 1.1, 1.2, ..., 9.9, 10). Vyzkoušejte si to na příkladech přímo z příkazové řádky Matlabu.

Dotaz:

Jak funguje následující příkaz (ve funkci *intpol_IDW_2D(xd,yd,zd,x,y,a)*)?

x=Xg(:);

y=Yg(:);

Zkuste sami zadat na příkazovou řádku příkaz X=[1 2;3 4] a pak X(:). Uvidíte, že matice se převedla na vektor (sloupcový).

Když se dostanete do situace, že nevíte, co daný příkaz znamená, je dobré ho vyzkoušet na jednoduchém příkladu z příkazové řádky.

Kolegyně napsala

Poznámka: Mohl byste prosím Vás ještě vysvětlit, co přesně vyjadřuje *xd, yd, zd, x, y*, a jaké jsou mezi nimi vztahy? Nedokážu si to stále dost dobře představit.

Asi myslíte v souvislosti s danou funkcí. Tyto proměnné vidíme například v závorce za názvem funkce

`intpol_IDW_2D_moje(xd,yd,zd,x,y,a)`

Víme už , že to jsou takzvané vstupní parametry. Jejich hodnoty musíme funkci při jejím vyvolání (spuštění) dodat, aby mohla počítat.

V kódu funkce jsou `xd,yd,zd` data (datové vektory) a `x,y` mohou být buď (a) přímo souřadnice bodů, kde počítáme, nebo (b) to mohou být pomocné údaje z nichž ty souřadnice vypočítáme.

(a) Například zde jsou x a y souřadnicemi bodů, kde počítáme:

```
function z=intpol_IDW_2D_1b(xd,yd,zd,x,y,a)

nd=length(xd);

for i=1:nd
   r=sqrt((xd(i)-x)^2+(yd(i)-y)^2); % pythagorka
   if r==0  %jestli x = 0, z se vlastne rovná zd
       z=zd(i);
       return
   else
       lam(i)=1/r^a; % vaha pocitana podle vzorecku
   end
end
```

(b) Ve funkci `intpol_IDW_2D(xd,yd,zd,x,y,a)` se ze zadaných vektorů x a y vytvoří nejprve síť (grid), jejíž jednotlivé body se pak vloží zpět do proměnných x a y a teprve pak se s nimi počítá:

```
intpol_IDW_2D(xd,yd,zd,x,y,a)
...
[Xg,Yg]=meshgrid(x,y);

x=Xg(:);
y=Yg(:);
 ...
```

Význam proměnných je v obou případech zřejmý z toho, jak se s nimi zachází v průběhu výpočtu (v kódu funkce).

Jak už jsem psal minule, názvy proměnných uvnitř funkce jsou formální, můžeme je nazvat jinak, ale musíme to systematicky dodržovat.

Závěrečná poznámka:

Předpokládám, že někteří se v prvních lekcích trápili s Matlabem. Ale nebojte, už jsme se naučili prakticky vše, co budeme v dalším průběhu potřebovat, dalších příkazů už moc nepřibyde a vše půjde stále lépe :)

A v budoucnu se vám znalost Matlabu vyplatí, pokud vím, máte či budete ho mít i jinde.