

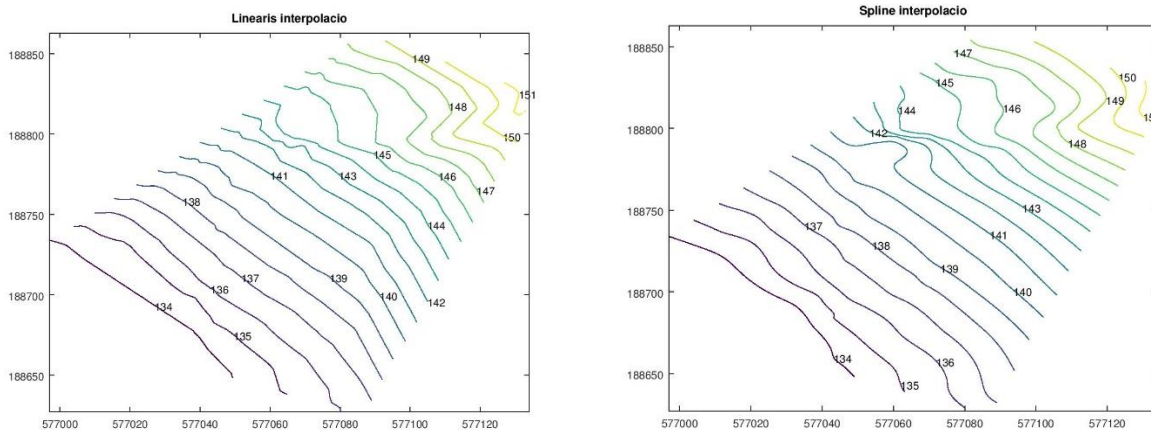
TEREPMETSZET KÉSZÍTÉS, DOMBORZATMODELL 2.

Az előző gyakorlatban a szórt pontokban lévő magassági értékekből állítottunk elő domborzatmodellt egy 1x1 méteres rácshálóra lineáris és spline interpolációval. hasonlítsuk most össze a két megoldást! A domborzat.mat fájlban az előző órán kapott adatok vannak, amiket a save paranccsal lehetett elmenteni:

```
save domborzat.mat XI YI ZI ZI2 x y z
```

x,y,z változók a terepfelmérés szórt pontjai, XI, YI a rácsháló x,y pontjai, ZI a lineáris interpoláció során kapott z értékek, ZI2 pedig a spline interpoláció során kapott z értékek.

A lineáris és spline interpolációval kapott szintvonalakat elmentettük az interp_lin.jpg és az interp_splien.jpg képekbe. A spline interpoláció jóval simább szintvonalakat eredményez, mint a lineáris interpoláció, azok kissé szögletesek. Viszont van egy terület, ahol a két ábra nagyon eltér egymástól. Vizsgáljuk ezt meg alaposabban!



KÉTFÉLE INTERPOLÁCIÓ KÖZÖTTI ELTÉRÉS PONTOKBAN

Töltsük le a domborzat.mat fájlt, majd olvassuk be a tartalmát (XI, YI, ZI, ZI2, x,y,z)! Számoljuk ki a kétféle interpoláció között az eltéréseket. Jelenítsük meg színátmenetes térképen az eredményt! Néhány pontban kérdezzük le grafikusán az eltérés nagyságát, majd jelenítsük meg a maximális eltérés helyét kiemelve!

```
> clear all; close all; clc;
> page_screen_output(0);% laponkénti megjelenítés leállítás Octave-ban
> load domborzat.mat
>
> % Elteres a ketfele interpolacio kozott
> DZ = ZI - ZI2;
>
> % Megjelenítés színátmenetes térképen
> figure(1);
> imagesc(XI(1,:), YI(:,1), DZ); % színátmenetes térkép
> axis xy;
> colorbar; % jelkulcs
> title('Eltérések');
> colormap(jet)
```

>

Egy felülnézeti ábrán a **ginput** paranccsal választhatunk ki egy pontot. Ezért használtuk megjelenítésre most a 3D **surf** parancs helyett az **imagesc** színátmenetes megjelenítést.

```
> msgbox('Adjon meg egy pontot a terkepre kattintva!')
> [xA yA] = ginput(1);
> % A pontban az eltérés értékét az interp2 paranccsal számíthatjuk
> zA = interp2(XI, YI, DZ, xA, yA)
```

Foglaljuk most ciklusba az egészet, és kérdezzük meg a felhasználótól szeretne-e még több pontot lekérdezni? A kérdés grafikus ablakban történő feltevéséhez többféle parancs közül választhatunk, használhatjuk az **inputdlg**, **menu** vagy a **questdlg** parancsot is. Most ez utóbbit használjuk. Gyakorlásképp próbálja ki a másik kettőt is, ami már korábban is szerepelt!

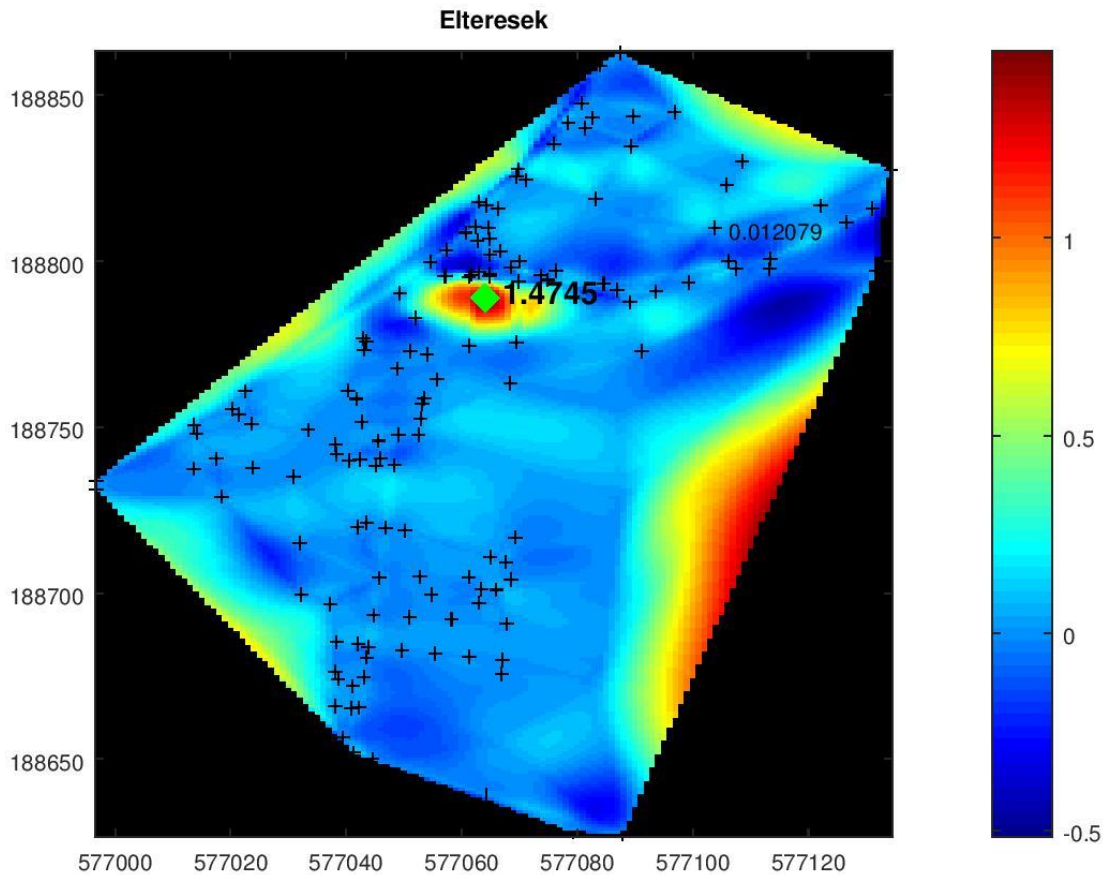
```
> kerdes = 'Szeretne meg pontokat lekérdezni?';
> valasz = 'igen';
> while strcmp(valasz, 'igen')
>     valasz = questdlg(kerdes, 'Kerem valaszoljon!', 'nem', 'igen', 'nem')
>     if strcmp(valasz, 'igen')
>         [xA yA] = ginput(1)
>         zA = interp2(XI, YI, DZ, xA, yA)
>         text(xA, yA, num2str(zA))
>     end
> end
```

Az **inputdlg** paranccsnál meg lehet adni egy címet az ablaknak (pl. most: 'Kerem valaszoljon!'), két gombot és hozzá a végén, hogy melyik legyen az alapértelmezett, ha csak simán 'enter'-t nyomunk. A ciklus addig menjen, amíg a válasz igen. Mivel a ciklus elején történik a feltételvizsgálat, ezért, a kérdésfeltevés után is meg kell vizsgálni a feltételt, és csak akkor folytatni a lekérdezést, ha a válasz 'igen'. Másik megoldás lehet, ha azt vizsgáljuk, hogy 'nem' választ kaptunk-e, és ebben az esetben egy **break** utasítással is ki lehet lépni a ciklusból.

MAXIMÁLIS ELTÉRÉS HELYE, ÉRTÉKE

Határozzuk meg a maximális eltérés értékét, helyét és jelenítsük meg!

```
> DZmax = max(max(abs(DZ))) % maximális eltérés a rácshálóban
> % a max. elteres helye a mátrixban (sor, oszlop)
> [imax jmax] = find(abs(DZ) == DZmax)
> % a max. elteres X, Y koordinátái a méterszer méteres rácshálóban
> xmax = XI(imax, jmax)
> ymax = YI(imax, jmax)
> % Kiplottolása nagyobb méretű zöld rombuszal, zöld kitöltéssel
> hold on;
> plot(xmax, ymax, 'gd', 'MarkerSize', 12, 'MarkerFaceColor', 'g');
> % Írjuk fel az eltérés értékét az ábrára!
> x0 = 3; y0 = 3; % eltolás értékek a szöveghez
> t = text(xmax+x0, ymax+y0, num2str(DZmax));
> set(t, 'fontSize', 14, 'Fontweight', 'bold');
> % Eredmény megjelenítése külön grafikus ablakban:
> msgbox(sprintf('A maximalis elteres: %.2f', DZmax))
> % Mérési pontok feltüntetése
> plot(x, y, 'k+'); % mérési pontok
```



Az eltérés ott a legnagyobb, ahol nincsenek mért pontjaink. Ezeken a helyeken a lineáris interpoláció egyszerűen levágja az értékeket egy egyenessel, a spline pedig a korábbi görbületekkel folytatva számítja a felületet. Ezt a legkönnyebben metszetek felvételével nézhetjük meg.

TEREPMETSZETEK É-D ÉS K-NY IRÁNYOKBAN

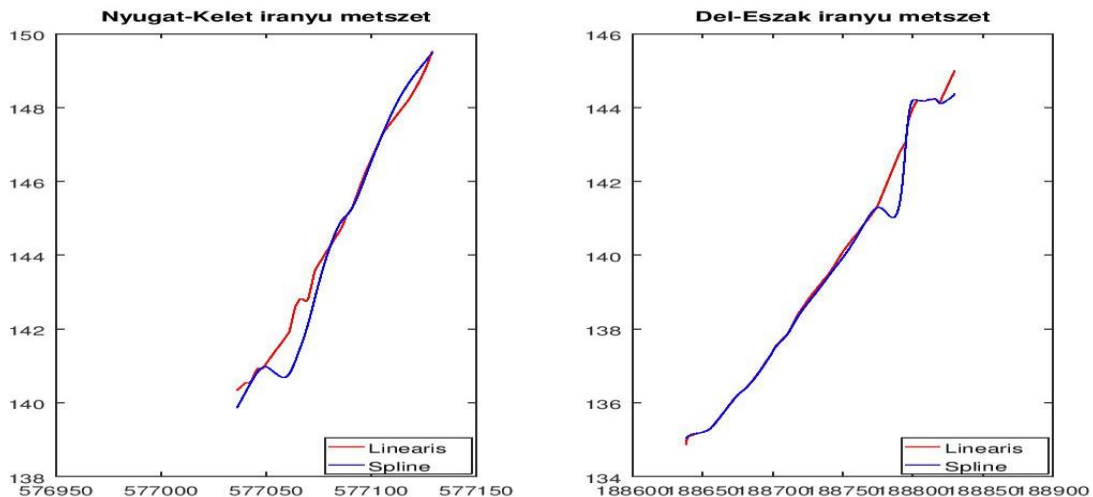
Készítsünk Észak-Dél és Kelet-Nyugat irányú metszeteket a maximális eltérés helyén! A metszetekhez kiválasztjuk a méterszer méteres rácshálónkból (XI, YI, ZI) azokat a sorokat (K-Ny) vagy oszlopokat (É-D), amelyek megfelelnek a maximális eltérésnek.

```
> % Nyugat-Kelet irányú metszet a maximális eltérés helyén
> figure(2);
> subplot(1,2,1)
> plot(XI(imax,:), ZI(imax,:), 'r'); hold on;
> plot(XI(imax,:), ZI2(imax,:), 'b')
> legend('Linearis', 'Spline', 'Location', 'southeast');
> title('Nyugat-Kelet irányu metszet');
>
> % Észak-Dél irányu metszet a maximális eltérés helyén
> subplot(1,2,2)
> plot(YI(:,jmax), ZI(:,jmax), 'r');hold on;
> plot(YI(:,jmax), ZI2(:,jmax), 'b');
> legend('Linearis', 'Spline', 'Location', 'southeast');
> title('Dél-Eszak irányu metszet');
>
```

```

> % A szebb megjelenítés miatt nyújtjuk meg az ábrát x irányban!
> % kérdezzük le az ábra alap pozícióját, méreteit
> % [pos_x, pos_y, size_x, size_y]
> pos = get(gcf, 'Position')
> % Nyújtjuk meg a x irányú méretet a kétszeresére!
> pos(3) = pos(3)*2
> set(gcf, 'Position', pos)

```



TEREPMETSZETEK TETSZŐLEGES IRÁNYBAN

A tetszőleges metszet felvételéhez két pontot kell kiválasztani a **ginput** paranccsal!

```

> % Metszet tetszőleges irányban
> pause(1); close(gcf); % bezárjuk az aktuális ábrát 1 mp után
> msgbox('Adjon meg két pontot metszet készítéséhez!');
> figure(1);
> [xA yA] = ginput(2)
> hold on;
> plot(xA, yA, 'r*-', 'Linewidth', 4);

```

Ezután a két végpont között felveszünk 100 pontot, és ezekben kiszámoljuk interpolációval a z értékeket, akár a lineáris, akár a spline interpoláció eredményét felhasználva az **interp2** paranccsal.

```

> xm = linspace(xA(1), xA(2), 100);
> ym = linspace(yA(1), yA(2), 100);
> zm = interp2(XI, YI, ZI, xm, ym);
> zm2 = interp2(XI, YI, ZI2, xm, ym);

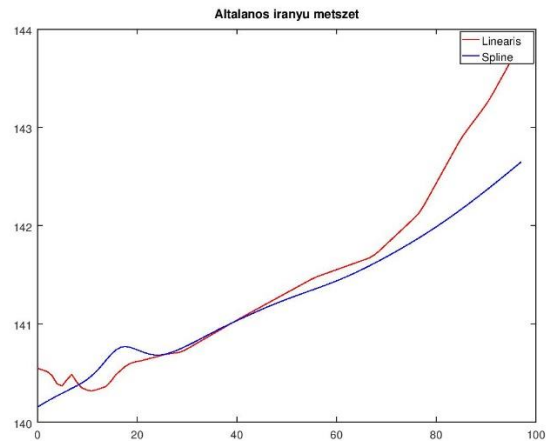
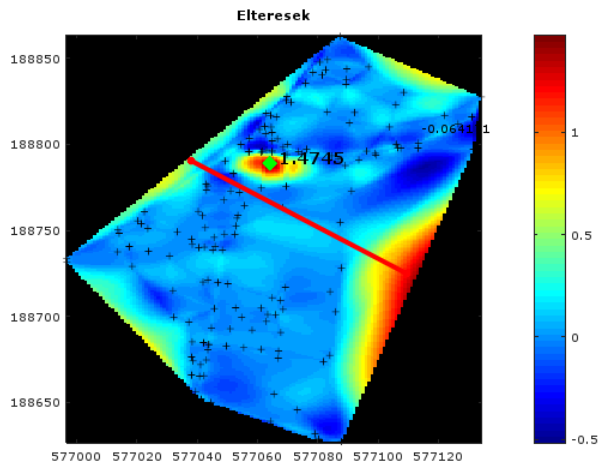
```

A z értékeket a kezdőponttól mért távolság függvényében fogjuk megjeleníteni. Ehhez ki kell számolnunk a távolságokat!

```

> s = sqrt((xm-xA(1)).^2 + (ym-yA(1)).^2);
> figure(3);
> plot(s, zm, 'r'); hold on;
> plot(s, zm2, 'b');
> legend('Linearis', 'Spline');
> title('Általános irányu metszet');

```



TEREPRENDEZÉS UTÁNI FÖLDTÉRFOGATOK SZÁMÍTÁSA

A területen elegyengették a kisebb nagyobb egyenetlenségeket, ezután készült egy újabb felmérés a területről. számoljuk ki ezek alapján a szükséges földmunka mennyiségét!

Először töltsük be a simított állományt!

```
> simitas = load('simitas_coo.txt');
> xs = simitas(:,2);
> ys = simitas(:,3);
> zs = simitas(:,4);
```

Interpoláljunk lineárisan rácshálóra, majd nézzük meg a különbséget a korábbi felületeinkkel!

```
> ZI3 = griddata(xs, ys, zs, XI, YI);
```

Számoljuk ki a szükséges földmunka mennyiségét, majd ábrázoljuk ezt! Ehhez számoljuk ki a két felület magasság különbségét a rácspontokban, rendezés előtt és után!

```
> foldmunka = ZI3 - ZI;
> figure(4);
> imagesc(XI(1,:), YI(:,1), foldmunka);
> axis xy;
> colorbar; colormap(jet);
> title('Foldmunka');
```

Számoljuk ki a szükséges töltés/bevágás mennyiségét! Mivel a rácsháló méterszer méteres így könnyű dolgunk van, csak összegezni kell a 'pixelek' számát.

```
> toltas = sum(foldmunka(foldmunka > 0)) % 4054.9
> bevagas = sum(foldmunka(foldmunka < 0)) % -2011.9
> egyenleg = toltas+bevagas % 2043.0
```

