

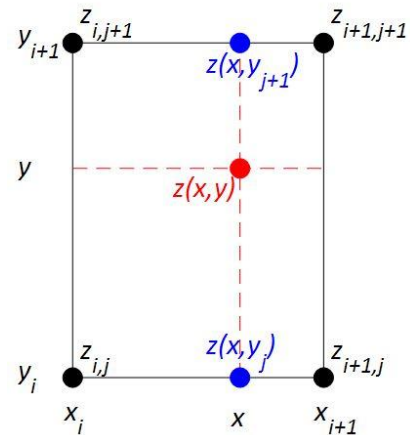
11. KÉTVÁLTOZÓS INTERPOLÁCIÓ, REGRESSZIÓ

Az előző gyakorlatokban regressziós modelleket, valamint adatok globális és lokális egydimenziós interpolációját vizsgáltuk, például polinomok, lineáris szakaszok vagy spline-ok segítségével. A feladataink azonban sok esetben kétdimenziós jellegűek, így egydimenziós regressziót vagy interpolációt nem tudunk alkalmazni. Szerencsére számos korábban vizsgált módszer kiterjeszhető magasabb dimenziókra is.

INTERPOLÁCIÓ SZABÁLYOS RÁCS ALAPJÁN¹

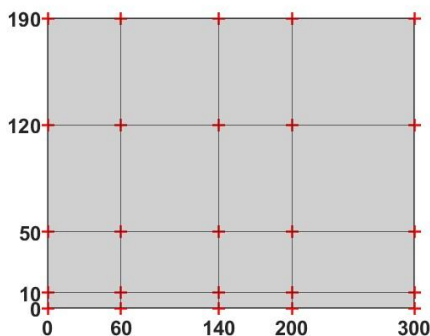
Amennyiben a mért pontjaink egy szabályos rács mentén helyezkednek el, akkor az interpoláció legegyszerűbb módja az úgynevezett bilineáris interpoláció. Az interpoláció három egyváltozós interpolációs lépésből áll:

- 1) Interpoláció az $y = y_j$ mentén vízszintesen: $z(x, y_j)$,
- 2) interpoláció az $y = y_{j+1}$ mentén vízszintesen: $z(x, y_{j+1})$,
- 3) végül a harmadik interpoláció az $x = \text{állandó}$ mentén függőlegesen, $z(x, y_j)$ és $z(x, y_{j+1})$ között, hogy megkapjuk a pont magasságát: $z(x, y)$



Megjegyzés: A fentiek csak a bilineáris interpoláció lépései, azonban több pont (pl. $j-1, j+2$ stb.) bevonásával magasabb fokú polinomiális interpoláció is lehetséges (mint például a bikubikus interpoláció).

Nézzünk egy példát, ahol a mért tereppontok egy rács mentén állnak rendelkezésre! A mérés elrendezése (x,y koordináták) az alábbi ábrán látható, a mért magasságok pedig a Z mátrixban (ami a **terep.txt** fájlból beolvasható). A két ábra alapján például a (0,0) pontban a magasság 133.2, a (300,190) pontban pedig 119.8.



$$Z = \begin{bmatrix} 135.0 & 131.7 & 136.4 & 139.8 & 119.8 \\ 134.8 & 133.0 & 139.7 & 135.5 & 120.0 \\ 124.4 & 130.0 & 140.0 & 139.2 & 122.3 \\ 131.1 & 133.8 & 138.1 & 137.7 & 121.1 \\ 133.2 & 137.1 & 143.0 & 135.0 & 123.3 \end{bmatrix}$$

Válaszoljunk a következő kérdésekre!

- 1) Mekkora a terep magassága a (100,100) pontban?
- 2) Ábrázoljuk ebben a pontban a Ny-K irányú metszetet!

¹ Felhasználva: Paláncz Béla (2012): Numerikus módszerek példatár + előadás fóliák

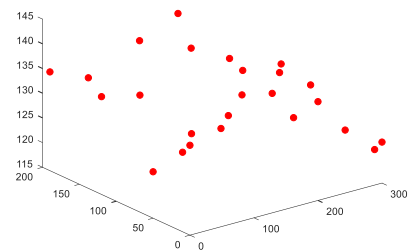
- 3) Mennyi földet kell hozni vagy elvinni, ha a fenti mérési eredményekkel rendelkező domborzatot 135 m magasságú sík tereppé szeretnénk átalakítani?

A megoldáshoz elő kell állítanunk a rácspontok x,y koordinátáit is és beolvasni a hozzájuk tartozó z értékeket. Figyeljünk arra, hogy a matematikai koordináta rendszer kezdőpontja a bal alsó sarokba esik, a mátrixok kezdőpontja (első sor, első oszlop) pedig a bal felső sarokba! A koordináta rendszerek eltérése miatt itt az y koordinátákat fentről lefelé kell megadnunk!

```
> % Tereprendezés
> clc; clear all; close all;
> % A magasság értékek beolvasása
> Z = load('terep.txt')
> % A rácspontok koordinátái (figyeljünk az adatok sorrendjére!)
> x=[0 60 140 200 300];
> y=[190 120 50 10 0];
> % Rácsháló előállítás
> [X,Y]=meshgrid(x,y)
```

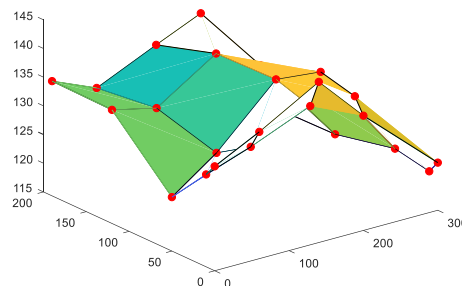
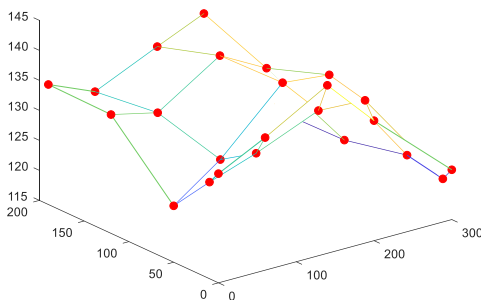
Ezzel előállt egy rácsháló x,y,z koordinátákkal. Jelenítsük meg a mért pontokat 3 dimenzióban (**plot3**)!

```
> figure(1)
> plot3(X,Y,Z, 'ro', 'MarkerFaceColor', 'r')
```



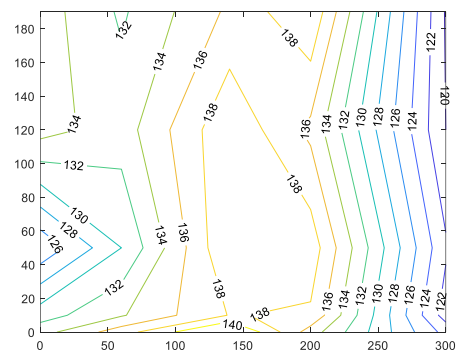
Szemléletesebben is meg lehet jeleníteni a domborzatot rácshálóként (**mesh**), vagy felületként (**surf**). Nézzük meg ezeket is!

```
> hold on; mesh(X,Y,Z)
> surf(X,Y,Z)
```



Másik gyakran használt lehetőség a szintvonalas ábrán történő megjelenítés (**contour**), ahol opcióként megadhatjuk azt is, hogy melyik szintvonalak jelenjenek meg (pl. most állítsuk be, hogy 120 és 140 méter között 2 méterenként legyenek szintvonalak), illetve bekapcsolhatjuk a szintvonalak feliratozását is a **set** paranccsal, ha két kimenettel hívjuk meg a **contour**-t, és a második kimenet tulajdonságait módosítjuk.

```
> figure(2)
> [C,h] = contour(X,Y,Z,118:2:144)
```



```
> set(h, 'ShowText', 'on')
```

A **set** paranccsal nem csak a feliratokat lehet bekapcsolni, hanem a szintvonalak tulajdonságai is módosíthatók. Pl. a *'LevelList'* tulajdonsághoz megadhatunk egy vektort, hogy pontosan milyen magasságú szintvonalakat akarunk megjeleníteni!

A fenti parancsok csak megjelenítésre szolgálnak, egy tetszőleges pontban nem adhatjuk meg velük a magasságot, ahhoz interpolációra vagy regresszióra van szükség.

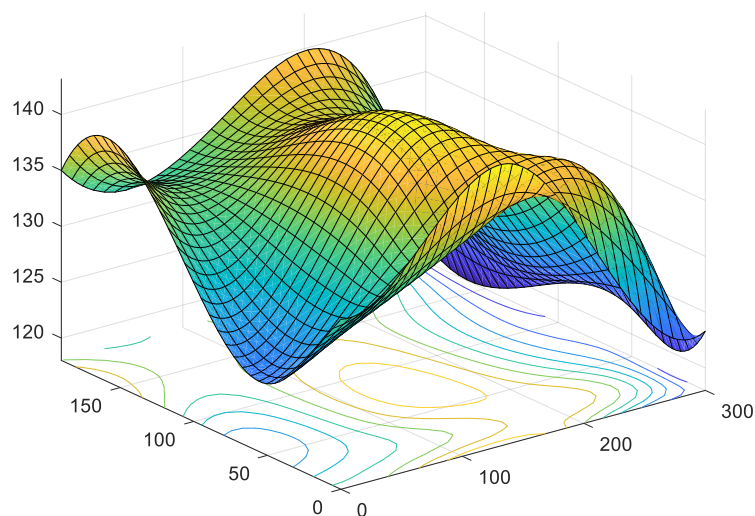
1) Mekkora a terep magassága a (100,100) pontban?

A kérdés megválaszolásához illesszünk a pontokra interpolációs felületet! Az interpolációhoz egy független változó esetén eddig az **interp1** parancsot használtuk, most 2 független változó esetén az **interp2** parancs használható. Az **interp2** csak rácshálóban megadott pontok/adatok esetén alkalmazható (a rácspontok egyenlő távolsága nem követelmény, csak a 'cubic' módszer alkalmazásakor). Több különböző módszerrel is meghívható:

- nearest - legközelebbi szomszéd,
- linear - bilineáris interpoláció,
- spline - spline interpoláció,
- cubic - 2D köbös spline interpoláció (bicubic), csak egyenletes ráctávolság esetén alkalmazható, ellenkező esetben spline interpoláció kerül helyette alkalmazásra.

Definiáljunk egy interpolációs függvényt és rajzoljuk fel az illesztett interpolációs felületet (**fsurf** vagy **ezsurf**)!

```
> % spline interpolációs függvény definiálása
> F=@(u,v) interp2(X,Y,Z,u,v,'spline');
> % A terep megjelenítése
> figure(3);
> fsurf(F,[0,300,0,190] , 'ShowContours', 'on')
> % Ellenőrzés
> F(0,0) % 133.2000
> F(300,190) % 119.8000
> % Terep magassága a 100,100 pontban?
> F(100,100) % 137.8079
```

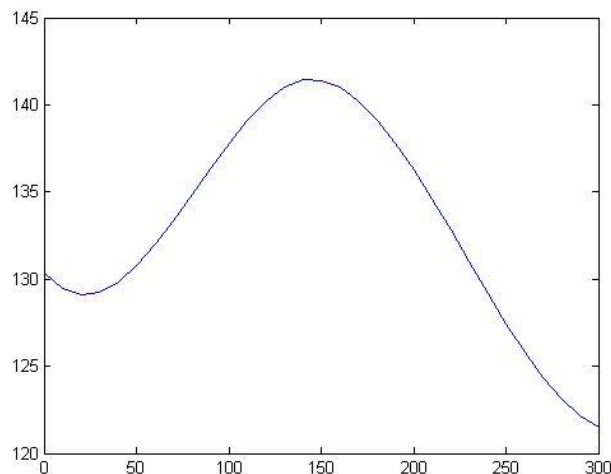


Ellenőrzésként megnéztük a magasságot két mért pontban, a (0,0) pontban a magasság 133.2 volt, a (300,190)-ben pedig 119.8, ugyanez lett az interpoláció eredménye is ezekben a pontokban. Az első kérdésre a válasz, a (100,100) pont magassága spline interpolációval 137.8 m-re adódott. Próbaképp cseréljük le a 'spline' módszert a 'nearest'-re, 'linear'-ra, hogy lássuk, mi történik ezekben az esetekben!

2) Ábrázoljuk ebben a pontban a Ny-K irányú metszetet!

A második feladat, hogy rajzoljuk fel a Ny-K irányú metszetet ebben a pontban. Ehhez vegyünk fel 50 pontot az $y=100$ keresztmetszetben a $P_1(0,100)$ és a $P_2(300,100)$ pontok között! Használjuk az előbb meghatározott függvényt a magasságok kiszámítására!

```
> % Ny-K irányú terepmetszet
> x0 = linspace(0,300,50)
> y0 = ones(size(x0))*100
> z0 = F(x0,y0)
> figure(4); plot(x0,z0);
```



3) Mennyi földet kell hozni vagy elvinni, ha a fenti mérési eredményekkel rendelkező domborzatot 135 m magasságú sík tereppé szeretnék átalakítani?

Számítsuk ki a feladat utolsó részét is, a földmunka térfogatának nagyságát, ha a domborzatot 135 m magasságú sík tereppé szeretnék átalakítani. Ehhez ki kell számítani a terep és a tengerszint által határolt földtér fogatot, illetve a 135 méteres vízszintes sík és a tengerszint által határolt földtér fogatot a megadott téglalap alakú területre! A két térfogat különbsége fogja megadni a földmunka térfogatának nagyságát. A terep által határolt földtér fogatot az alábbi kettős integrállal (**integral2**) számíthatjuk:

$$V_t = \int_0^{190} \int_0^{300} F(x,y) dx dy$$

A vízszintes $z=\text{áll.}$ sík által határolt földtömeg mennyisége egy téglalast térfogata: $V_{sz} = z \cdot 300 \cdot 190$.

```
> format longG
> vt=integral2(F,0,300,0,190) % 7614016.55348964
```

```
> % vízszintes sík által határol földtér fogat
> Vs=135*300*190 % 7695000
> % A tereprendezéshez szükséges földtér fogat
> Vr=Vs-Vt % 80983.4465103578
```

Mivel a tervezett vízszintes sík által határolt térfogat nagyobb volt, mint a terep alatti, ezért feltöltésre van szükség, méghozzá 80983 m³ földre.

POLINOMIÁLIS REGRESSZIÓ

A térbeli mérési pontokra természetesen nem csak interpolációval, hanem regresszióval is illeszthetünk felületet. Az egyik leggyakrabban alkalmazott kétváltozós regresszió a polinomiális regresszió, amikor különböző fokszámú algebrai polinomokkal írjuk le a felületet. Ide tartozik a sík illesztés (elsőrendű algebrai polinom), vagy paraboloid felület (másodrendű polinom) illesztése is. Nagyon magas fokszámú polinomot nem szabad alkalmazni a túltanulás miatt. Az illesztő pontokban a hibaösszeg csökkenése ugyanis nem jelent feltétlenül megbízhatóbb modellt, sokszor a pontok között a felületben nagyfokú hullámmás, oszcilláció lép fel.

Kétváltozós esetben az elsőrendű polinomot, vagyis regressziós síkot a következő módon írhatunk fel:

$$z = f(x, y) = c_1 + c_2x + c_3y$$

A másodfokú regressziós polinom (paraboloid felület) felírása a következő:

$$z = f(x, y) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2$$

Harmadfokú regressziós polinom általánosan:

$$z = f(x, y) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$$

Hasonlóképp negyed-, ötöd- stb.fokú polinomokat is felírhatnánk, de a túltanulás jelensége miatt 5. fokúnál magasabbat nem szoktak alkalmazni.

Illesszünk most egy másodfokú regressziós polinomot a mért térbeli pontokra, és ennek felhasználásával adjunk becslést a terep magasságára a 100,100 koordinátájú pontban! n pont esetén n darab lineáris egyenletet írhatunk fel, és másodfokú polinomiális felület esetében 6 meghatározandó paraméterünk van. A fölös mérések száma: $n-6$. Az egyenletrendszer mátrixos alakban ($A \cdot c = b$):

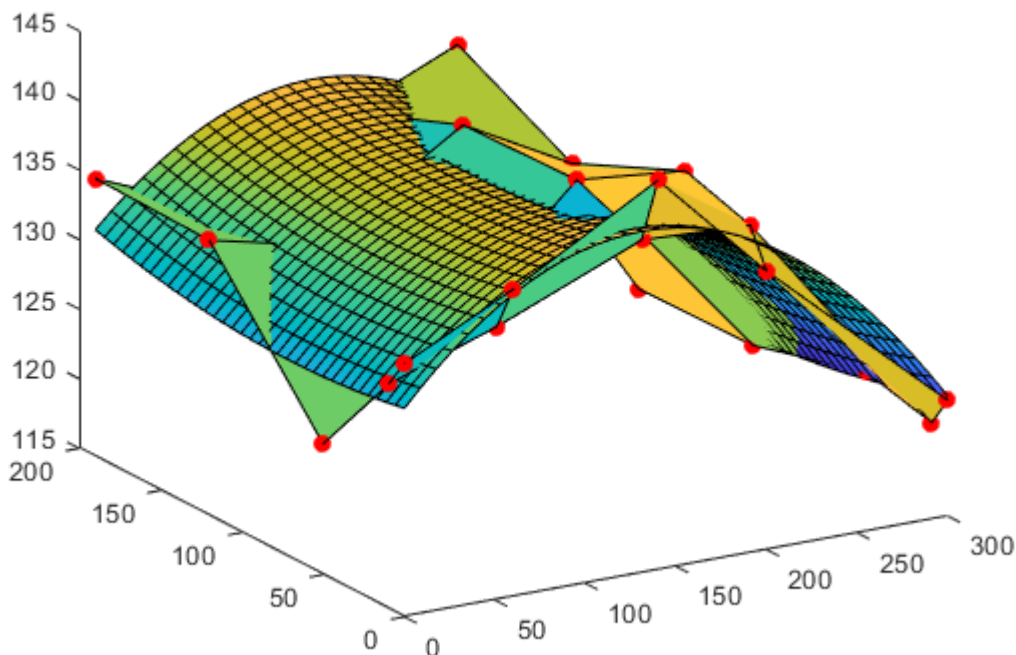
$$A = \begin{pmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & x_ny_n & y_n^2 \end{pmatrix}; c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_6 \end{pmatrix}; b = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$$

Az A alakmátrixban x,y koordináták és ezek megfelelő hatványai szerepelnek az oszlopvektorokban. Most a koordináták nem oszlopvektorban, hanem rácshálóban állnak rendelkezésre, mátrixokban megadva. A regresszió meghatározásához ezért ebben az esetben első lépésként ezeket a mátrixokat oszlopvektorokká kell alakítanunk. Ezt megtehetjük a következő alakú paranccsal: $M(:)$. Ez a parancs az M mátrix összes elemét egy oszlopvektorba rakja be, egymás után listázva az egyes oszlopok tartalmát.

Mivel a feladatban jóval több, mint 6 adatunk van, ezért egy túlhatározott lineáris egyenletrendszerrel kell megoldanunk, a legkisebb négyzetek módszerével minimalizálva a maradék ellentmondásokat. A feladat megoldása lényegében ugyanúgy megy, mint egyváltozós esetben.

```

> % Másodfokú polinomiális felület (paraboloid) illesztése
> format shortG
> % Matrikban tárolt rácspontok koordinátáinak oszlopvektorra alakítása
> x = X(:); y = Y(:); z = Z(:);
> % A együttható mátrix definiálása
> % z = c1 + c2*x + c3*y + c4*x^2 + c5*x*y + c6*y^2
> A = [ones(size(x)), x, y, x.^2, x.*y, y.^2]
> % paraméterek meghatározása
> c = A\z
> % felület megadása, egyes ábrába berajzolása
> f2 = @(x,y) c(1)+c(2)*x+c(3)*y+c(4)*x.^2+c(5)*x.*y+c(6)*y.^2
> figure(1); hold on; fsurf(f2,[0 300 0 190])
> view([-30 25])
    
```



Határozzuk meg az illesztett regressziós felület esetében is az $x=100, y=100$ pont magasságát!

```

> % x1=100, y1=100, z1=?
> z1 = f2(100,100) % 137.6
    
```

A megoldást megkaphatjuk a Matlab egy másik beépített függvényét, a **regress**-t használva is, ekkor az együtthatók mellett megkaphatjuk a 95 százalékos konfidencia intervallumukat és a maradék ellentmondásokat is. Ezt felhasználhatjuk az illesztés korigált tapasztalati szórásának meghatározásához is.

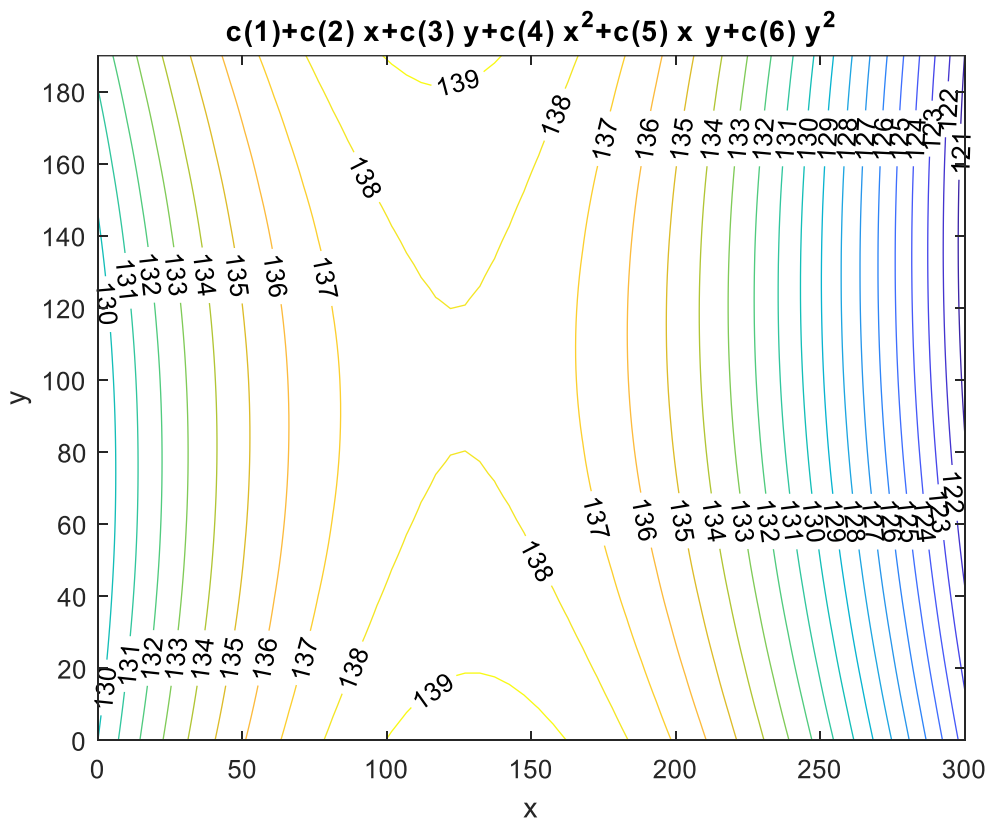
```

> % megoldás regress paranccsal
    
```


- > [c int95 err]=regress(z,A)
- > % korr. tapasztalati szórás számítása
- > S = sum(err.^2) % 227.03
- > n = length(x) % pontok száma, n=25
- > np = length(c) % paraméterek száma, np=6
- > szoras = sqrt(S/(n-np)) % 3.4567

Ábrázoljuk szintvonalakkal is a felületet! Használjunk 1 méteres szintvonalközt az alap 2 méteres helyett! Mivel itt egy függvénnyel definiált felület van, nem rácshálóban megadott pontok, ezért a **contour** parancs helyett az **fcontour** vagy **ezcontour** szükséges. A Matlab 2017-es verziójától az **ezcontour** használata nem javasolt, hanem helyette az **fcontour**. Viszont ez utóbbi esetében egyelőre nem működik a szintvonalak feliratozása, ezért használjuk most az **ezcontour** parancsot. Ha a feliratokat be szeretnénk kapcsolni, akkor a szintvonalas ábrát el kell menteni egy változóba, és utána a **set** parancs segítségével állíthatjuk be a feliratozást ('**ShowText**',**on**), illetve itt adhatjuk meg, hogy melyik szintvonalak jelenjenek meg vagy a ('**LevelList**',120:1:140) tulajdonságot használva, felsorolva egy vektorban a megjelenítendő szintvonalakat, vagy a ('**LevelStep**',1) tulajdonsággal megadva a szintvonalköz értékét.

- > figure(5)
- > h = ezcontour(f2,[0 300 0 190])
- > set(h,'**ShowText**','**on**','**LevelStep**',1)



SZABÁLYTALAN ELRENDEZÉSŰ PONTOK

Regresszió esetben nincs különbség szabályos és szabálytalan elrendezésű mérési pontok között. Az illesztett felület paramétereit ugyanúgy kell meghatározni mindkét esetben, túlhatározott lineáris egyenletrendszert megoldva polinomiális felületillesztés

esetében. A szabályos rácshoz képest könnyebbséget jelent, hogy itt többnyire vektorokban vannak megadva az adatok, ezért az előző példában használt mátrixok oszlopvektorra alakítása lépés kihagyható. Az interpoláció azonban jóval bonyolultabb szabálytalan elrendezésű pontok esetében. Nézzünk erre egy példát! ²

A magyar Duna vízgyűjtőjének területei Ausztriában és Bajorországban található. Ha itt jelentős mennyiségű csapadék esik, akkor a Dunán árhullám vonul le. Feladatunk a budapesti tetőző vízállás előrejelzése a következő történelmi adatok alapján:

- x: az árhullámot kiváltó csapadék, amely 15 bajor illetve osztrák csapadékjelző állomás adatainak középértéke [mm];
- y: a Duna vízállása Budapestnél az árhullámot kiváltó esőzések kezdetekor [cm];
- z: a becsülendő érték Budapestnél, az árhullám tetőzéséhez tartozó vízszint [cm].

Az eddig mért adatokat az alábbi táblázat tartalmazza (az adatok az arhullam.txt file-ban elérhetőek), ebben egy sor egy adott időponthoz tartozó x, y, z értékeket jelenti:

Csapadék [mm]	Vízállás [cm]	Tetőzés [cm]
58	405	590
52	450	660
133	350	780
179	285	770
98	330	710
72	400	640
72	550	670
43	480	520
62	450	660
67	610	690
64	380	500
33	460	460
57	425	610
62	560	710
54	420	620
48	620	660
68	390	620
74	350	590
95	570	740

Kérdés:

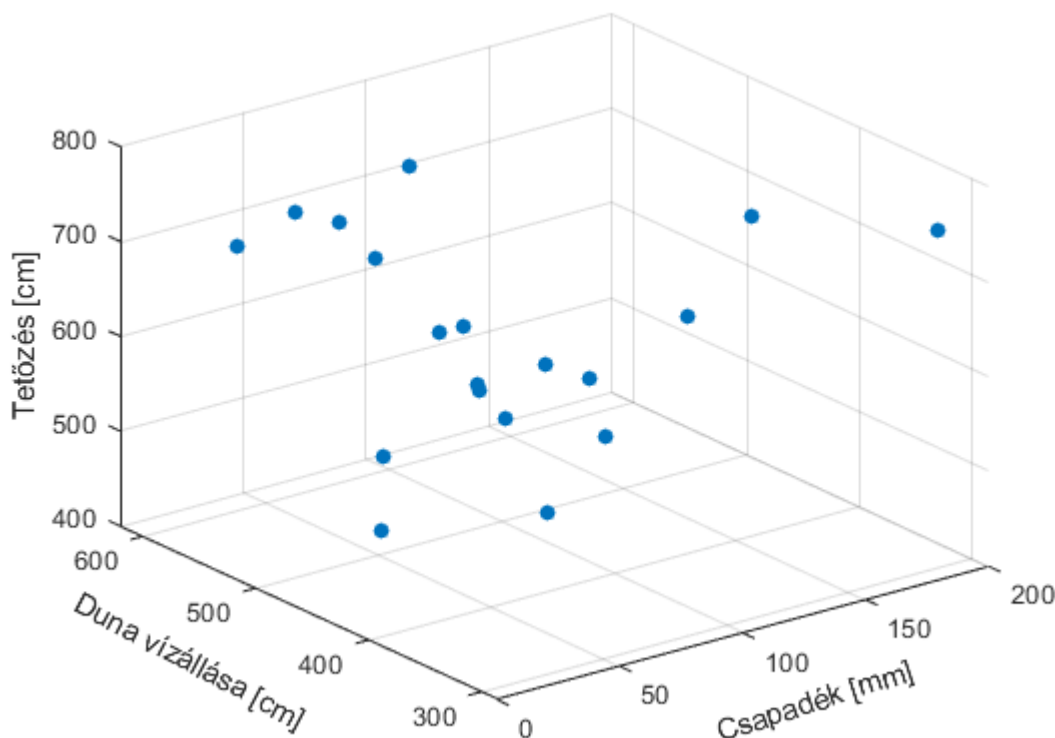
- Ha az osztrák és bajor vízgyűjtő területen átlagosan 100 mm csapadék esett, ugyanekkor a Duna vízszintje Budapesten 400 cm-nél volt, akkor mekkorára várható az árhullám tetőzése Budapesten?

Illesszünk lineáris és köbös spline interpolációs felületet is a pontokra, és ezek felhasználásával adjunk becslést az árhullám várható tetőzési magasságra!

² Felhasználva: Paláncz Béla (2012): Numerikus módszerek példatár + előadás fóliák

Először olvassuk be és ábrázoljuk az adatokat! 3D szórt pontokat a **plot3** vagy a **scatter3** paranccsal tudunk megjeleníteni.

```
> clc; clear all; close all;
> xyz=load('arhullam.txt')
> x=xyz(:,1); % árhullámot kiváltó csapadék [mm]
> y=xyz(:,2); % Duna vízállása Budapestnél az esőzés kezdetekor [cm]
> z=xyz(:,3); % a becslendő érték, árhullám tetőzése Budapestnél [cm]
> % Ábrázolás
> figure(1);
> scatter3(x,y,z,'filled')
> xlabel('Csapadék [mm]', 'Rotation',18)
> ylabel('Duna vízállása [cm]', 'Rotation',-28)
> zlabel('Tetőzés [cm]')
```



INTERPOLÁCIÓ SZABÁLYTALAN ELRENDEZÉSŰ PONTOK ESETÉN

Szabálytalan elrendezésű pontok esetében többféle interpolációs módszert szoktak alkalmazni, pl.:

- a legközelebbi szomszéd módszere, amikor a pont a hozzá legközelebb eső pont értékét kapja
- krigelés, amikor távolságot is figyelembe vesszük valamilyen súlyozással
- radiál-bázis függvények, amikor több bázisfüggvényt illesztünk a pontokhoz
- szabályos geometriai elemekre bontás.

A Matlab beépített függvényei között a szabályos geometriai elemekre bontást találjuk meg, a legközelebbi szomszéd mellett. A **griddata** függvény esetében is több interpolációs módszert lehet alkalmazni:

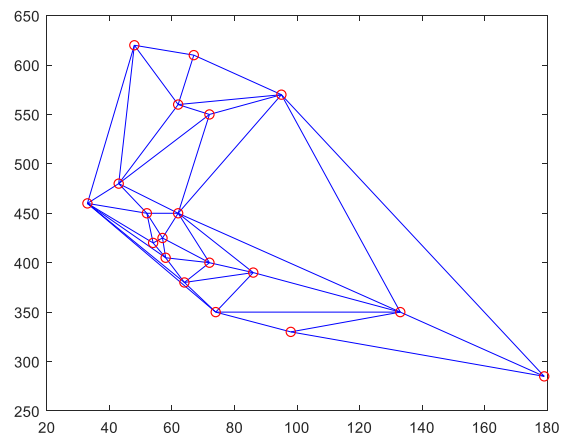
- nearest: legközelebbi szomszéd interpoláció
- linear: háromszög alapú lineáris interpoláció, ekkor minden háromszögre egy síkot illesztünk (TIN model, triangular irregular network)
- natural: háromszög alapú 'natural neighbor interpolation', a szomszédos pontok súlyozott átlagán alapul
- cubic: háromszög alapú köbös interpoláció ('bicubic spline' interpoláció).
- v4 - biharmonikus spline interpoláció (ez nem háromszögelésen alapul)

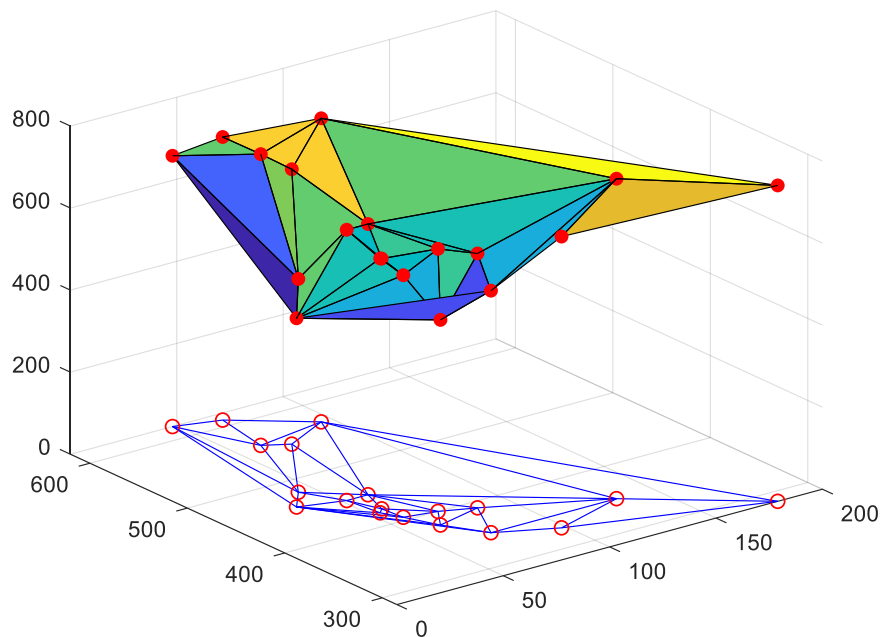
Szabálytalan térbeli pontok esetén a **griddata** parancs mellett használható a **scatteredInterpolant** parancs is, ami azonban csak 3 módszerrel használható ('linear', 'nearest' és 'natural'). Köbös és spline interpoláció ennél a parancsnál nincs.

A háromszög alapú interpoláció az adatok Delaunay-háromszögelését használja. Ez a módszer a pontok területét olyan háromszögekre osztja fel, amelyek körülírt körei nem tartalmaznak más pontot. Ez maximalizálja a legkisebb szög méretét bármelyik háromszögben, hogy elkerüljük a rendkívül hegyesszögű háromszögeket. Néhány alkalmazási területe: domborzatmodellezés (segítségével könnyen számíthatók az esés-, kitettség viszonyok, meghatározhatóak a szintvonalak), illetve a végelem-módszerben is gyakran használják, mivel könnyen előállítható.

A következő ábrák a Delaunay háromszögelését mutatják az előző példa pontjait használva.

```
> % delaunay háromszögelés
> tri = delaunay(x,y) % Delaunay
háromszögek létrehozása a
pontok x,y koord. alapján
> figure(2);
> triplot(tri,x,y) % Delaunay
háromszögek kirajzolása
> hold on; plot(x,y,'ro');
> figure(3);
> trisurf(tri,x,y,z); % 3D
ábrázolása a háromszögeknek
> hold on;
> scatter3(x,y,z,'ro','filled');
> plot(x,y,'ro');
> triplot(tri,x,y);
```





Válaszoljunk a feltett kérdésre! Mekkora lesz az árhullám várható tetőzése Budapesten háromszög alapú lineáris és köbös interpolációval, ha a lehullott csapadék 100 mm és az aktuális vízállás Budapestnél 400 cm? Ezt megtehetjük a megfelelő pontra elvégzett interpolációval, a **griddata** parancs használatával.

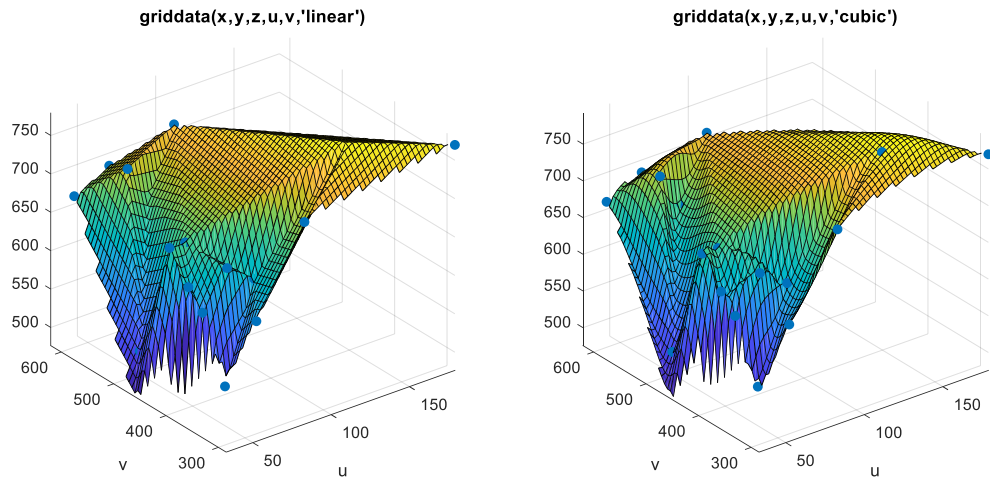
```
> zlin = griddata(x,y,z,100,400,'linear') % 724.7377
> zkob = griddata(x,y,z,100,400,'cubic') % 735.3939
```

A lineáris interpoláció alapján 725 cm-en várható a tetőzés, a köbös alapján pedig 735 cm-en. Függvényként is definiálhatjuk a fentieket, és akkor tetszőleges pontra könnyen meghívhatjuk, illetve ábrázolni is tudjuk az interpolációs felületet.

```
> % Interpolációs függvény definiálása
> flin = @(u,v) griddata(x,y,z,u,v,'linear')
> fkob = @(u,v) griddata(x,y,z,u,v,'cubic')
> flin(100,400) % 724.7377
> fkob(100,400) % 735.3939
```

Ábrázoljuk a függvényeket térben. Sajnos most az **fsurf** nem tudja megjeleníteni a függvényeket (Matlab R2024b), csak a korábbi verziókból még megmaradt **ezsurf** függvény működik szabálytalan határok esetén.

```
> figure(4); subplot(1,2,1)
> scatter3(x,y,z,'filled'); hold on;
> ezsurf(flin,[min(x),max(x),min(y),max(y)])
> subplot(1,2,2)
> scatter3(x,y,z,'filled'); hold on;
> ezsurf(fkob,[min(x),max(x),min(y),max(y)])
```



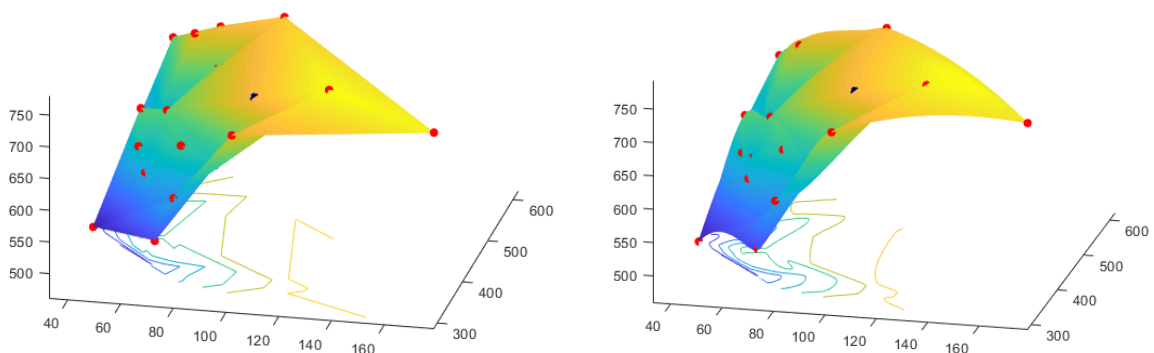
Egy másik térbeli ábrázolási mód sűrű rácshálával illetve szintvonalakkal történhet. Ehhez először egy sűrű rácst kell definiálnunk és kiszámolni a rácspontokban az interpolált függvény értékeit.

```
> % 1000x1000-as rács előállítás
> xx = linspace(min(x),max(x),1000);
> yy = linspace(min(y),max(y),1000);
> [X, Y] = meshgrid(xx,yy);
> ZL = flin(X,Y);
> ZC = fkob(X,Y);
```

Az ábrázoláshoz használhatjuk a **mesh** parancsot, ami rácshálóban megadott felületet jelenít meg, vagy a **meshc** parancsot, ami a szintvonalakat is alárajzolja. használjuk most ez utóbbit!

```
> figure(); hold on;
> scatter3(x,y,z,'ro','filled');
> meshc(X,Y,ZL), view([-10 25])
> plot3(100,400,zlin,'bo','MarkerFaceColor','k')
> figure(); hold on;
> scatter3(x,y,z,'ro','filled');
> meshc(X,Y,ZC), view([-10 25])
> plot3(100,400,zkob,'bo','MarkerFaceColor','k')
```

A következő két ábrán látszik, hogy a második, a köbös interpoláció, simább felületet eredményezett, a szintvonalak sem olyan szögletesek.



A FEJEZETBEN HASZNÁLT ÚJ FÜGGVÉNYEK

diff	- vektor elemeinek különbsége, közelítő numerikus derivált, szimbolikus derivált számítása
cumsum	- vektor elemeinek folyamatos összegzése
meshgrid	- 2-3 dimenziós rács előállítása vektorban tárolt x,y(z) koordinátákból
plot3	- Pontok 3D megjelenítése
mesh, meshc	- Rácshálóban adott 3D pontok megjelenítése térbeli rácsként, a parancsot c-vel kiegészítve a szintvonalakat is alárajzolja
surf	- Rácshálóban adott 3D pontok megjelenítése színezett felületként (kitöltött térbeli rács)
contour	- Rácshálóban adott 3D pontok alapján szintvonalak rajzolása
set	- Grafikus objektum tulajdonságainak beállítása (pl. 'ShowText', 'LevelStep', 'LevelList' ezcontour parancs esetén)
interp2	- 2D interpoláció rácshálóban adott pontokból tetszőleges pontra (módszerek: 'linear', 'nearest', 'spline', 'cubic')
integral2	- Kettős integrál számítása numerikusan, szabályos téglalap tartományon
scatter3	- Szórt pontok 3D megjelenítése
regress	- Többváltozós lineáris regresszió legkisebb négyzetek módszerével
fsurf, ezsurf	- 3D felületek kirajzolása megadott függvény alapján
fcontour, ezcontour	- Szintvonalak kirajzolása függvény alapján
griddata	- Interpoláció szórt pontok alapján tetszőleges pontra vagy rácusra ('linear', 'nearest', 'natural', 'cubic', 'v4' módszerek)
scatteredInterpolant	- Interpoláció szórt pontok alapján, mint griddata, csak kevesebb módszer: 'linear', 'nearest', 'natural'