

TÉRBELI MEGJELENÍTÉSEK

Az eddigiekben a kétdimenziós ábrázolási lehetőségeket használtuk csak, de sokszor szükség van térbeli, 3D megjelenítésekre is, lehet ez térbeli görbe vagy felület ábrázolása. Nézzünk ezekre is néhány példát.

TÉRBELI GÖRBÉK MEGJELENÍTÉSE PARAMÉTERESEN

Ábrázoljuk a következő paraméteres alakban megadott két, térbeli spirált!

$$x1 = \cos(2 \cdot \pi \cdot t)$$

$$y1 = \sin(2 \cdot \pi \cdot t)$$

$$z1 = t$$

$$x2 = t \cdot \cos(2 \cdot \pi \cdot t)$$

$$y2 = t \cdot \sin(2 \cdot \pi \cdot t)$$

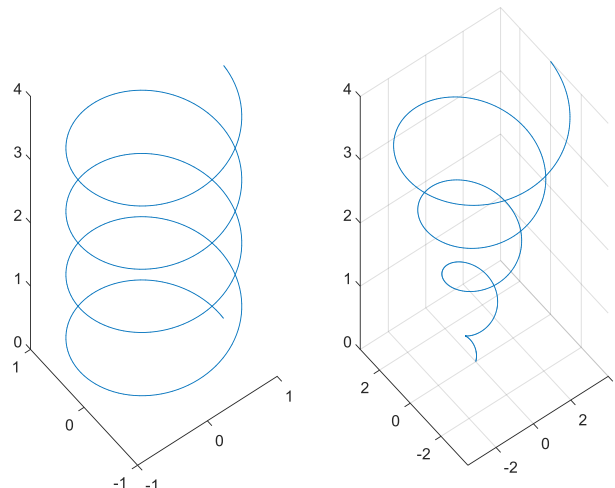
$$z2 = t$$

Az első esetben térbeli koordinátákkal megadott görbét fogunk ábrázolni a **plot3** paranccsal. Ehhez először ki kell számolnunk a spirál adott sűrűségben felvett pontjainak koordinátáit. A két spirált egymás mellett szeretnénk egy ábrába felrajzolni. A t paraméter értékét vegyük fel 0 és 4 között, 0.01 közökkel.

```
> % megadás 3D koordinátákkal
> clc; clear; close all;
> t = 0:0.01:4;
> x1 = cos(2*pi*t);
> y1 = sin(2*pi*t);
> z1 = t;
> figure(); subplot(1,2,1); plot3(x1,y1,z1)
```

Természetesen térbeli görbét is megjeleníthetünk paraméteres függvényekkel, ehhez az **fplot3** parancsot használhatjuk.

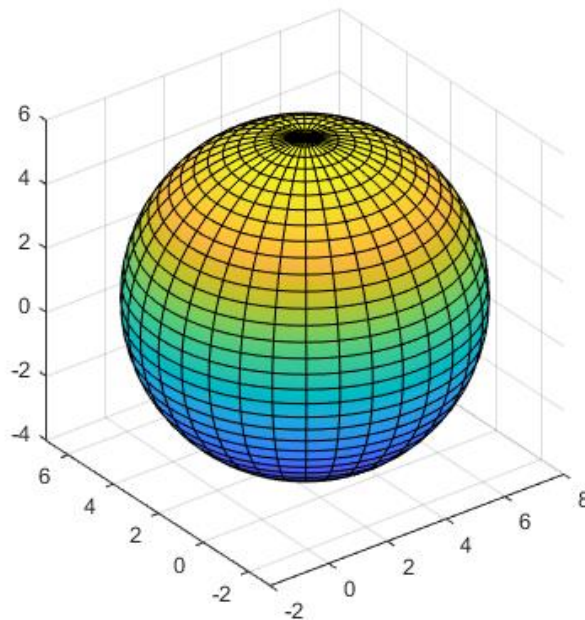
```
> % megadás paraméteres függvényekkel
> x2 = @(t) t.*cos(2*pi*t);
> y2 = @(t) t.*sin(2*pi*t);
> z2 = @(t) t;
> subplot(1,2,2)
> fplot3(x2,y2,z2,[0,4])
```



KÉTVÁLTOZÓS FÜGGVÉNY PARAMÉTERES MEGJELENÍTÉSE

Szükség lehet paraméteresen megadott felületek megjelenítésére is, pl. egy gömb megjelenítése esetén. Jelenítsünk meg egy gömböt, (3,2,1) középponttal és R=5 sugárral! Ebben az esetben 2 paraméterre lesz szükség, ami két szöveget jelent, mint a földrajzi szélesség, hosszúság megadása.

```
> %% Paraméteres 3D felület
> x0 = 3; y0 = 2; z0 = 1; R = 5;
> x = @(t,s) x0 + R*sin(t).*cos(s)
> y = @(t,s) y0 + R*sin(t).*sin(s)
> z = @(t,s) z0 + R*cos(t)
> figure(1);
> fsurf(x,y,z,[0,pi,-pi,pi])
> axis equal
```



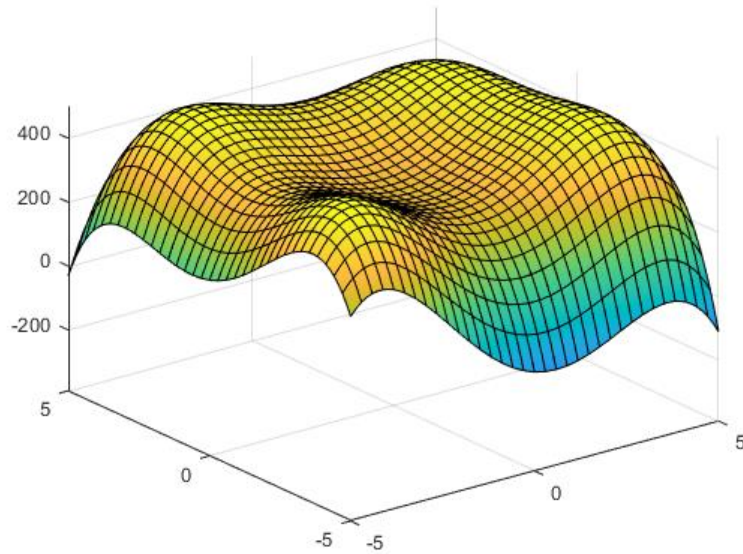
EXPLICIT KÉTVÁLTOZÓS FÜGGVÉNYEK MEGJELENÍTÉSE

Jelenítsük meg a következő függvényt:

$$f(x,y) = 500 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$$

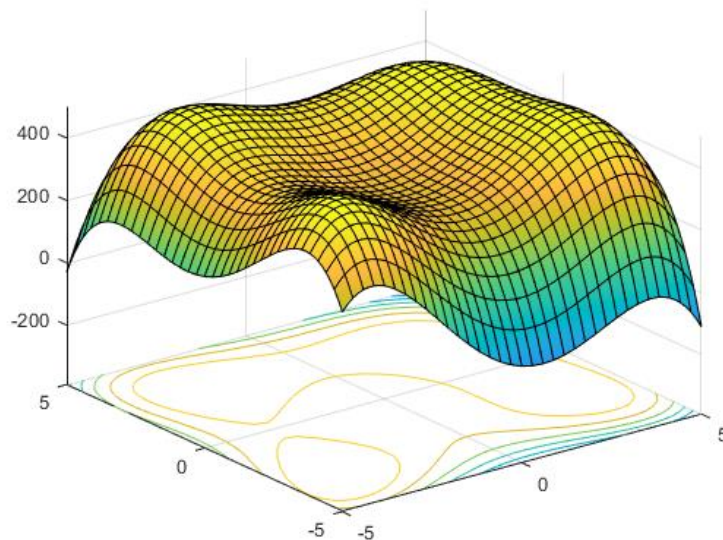
A legegyszerűbben egy kétváltozós függvényt az **fsurf** vagy **ezsurf** segítségével jeleníthetünk meg.

```
> %% 3D ábrázolások - felület
> clc; close all; clear all;
> f = @(x,y) 500 - (x.^2+y-11).^2 - (x+y.^2-7).^2
> figure(1);
> fsurf(f, [-5,5,-5,5]) % vagy ezsurf(f, [-5,5,-5,5])
```



Berajzolhatjuk a 3D ábrába a szintvonalakat is:

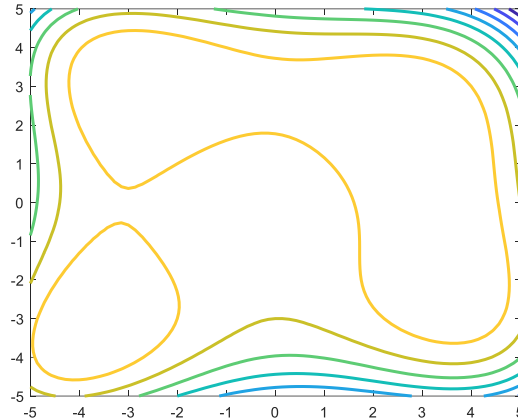
```
> fsurf(f, [-5, 5, -5, 5], 'showContours', 'on')
```



Sok esetben a kétváltozós függvény tisztán szintvonalas megjelenítése sokkal szemléletesebb. Ehhez az **fcontour** vagy az **ezcontour** parancs használható.

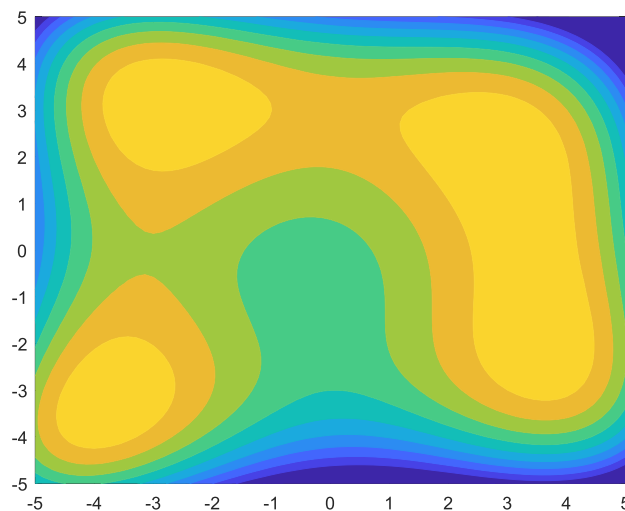
```
> % kétváltozós függvény szintvonalas megjelenítése  
> figure(2);  
> fcontour(f, 'Linewidth', 2) % vagy ezcontour(f)
```

Az **ezcontour** parancs használata esetén a tulajdonságokat a már korábban látott **set** paranccsal lehet beállítani.



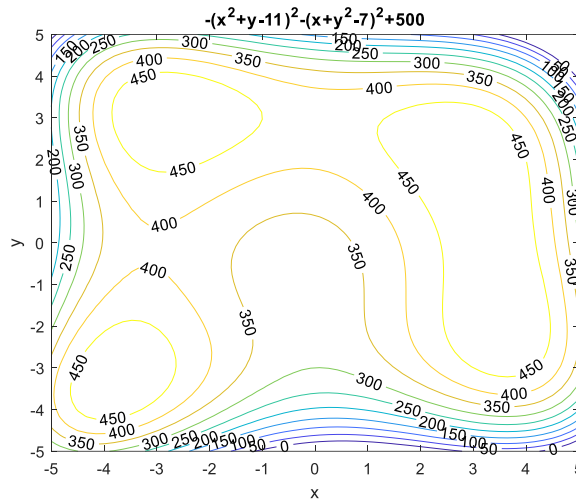
A szintvonalakat lehet sűríteni, egy vektorban megadhatóak, hogy pontosan melyik szintvonalakat akarjuk megjeleníteni, és akár színtöltést is adhatunk neki.

```
> % szintvonalak beállítása, kitöltés
> figure(3);
> fcontour(f, [-5,5,-5,5], 'LevelList',0:50:600, 'Fill','on')
```



Még szemléletesebb egy szintvonalas térkép, ha a szintvonalak feliratozva is vannak. Ehhez sajnos az **fcontour** parancs nem, csak az **ezcontour** használható.

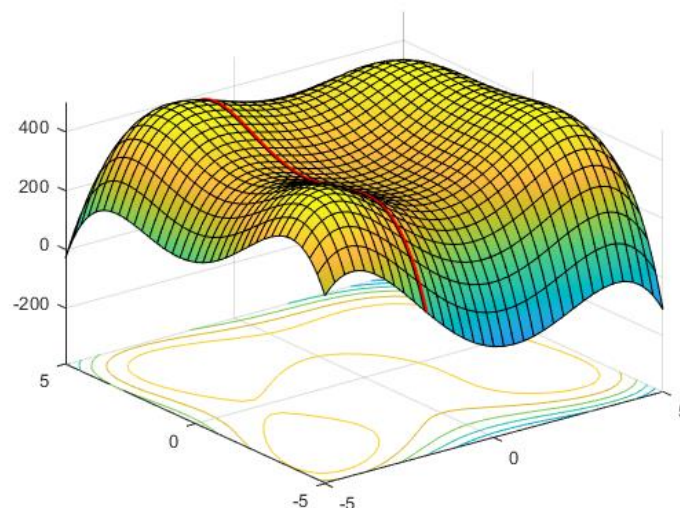
```
> % szintvonalak feliratait csak az ezcontour-ra lehet megjeleníteni
> figure(4);
> h = ezcontour(f, [-5,5,-5,5])
> set(h, 'ShowText', 'on', 'LevelList',0:50:600)
```



METSZETEK, FELÜLETI VONALAK MEGJELENÍTÉSE

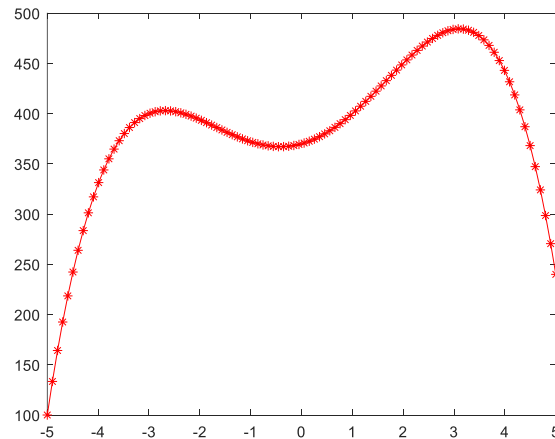
Gyakran van szükség arra, hogy egy felületen metszeteket vegyünk fel és ezeket megjelenítsük. Ezt általában úgy tehetjük meg a legegyszerűbben, hogy a metszet két végpontja között felvesszünk sok pontot és ezekben a pontokban kiszámoljuk a magasság értékeket. Nézzünk meg az előző felületen egy D-É irányú metszetet $x=-2$ értéknél!

- ```
> % Metszetek, D-É irányú metszet x=-2-nél
> ym1 = linspace(-5,5,100);
> xm1 = -2*ones(size(ym1));
> zm1 = f(xm1,ym1); % magasságok kiszámolása
> figure(1); hold on;
> plot3(xm1,ym1,zm1,'r','Linewidth',2) % berajzolás
```



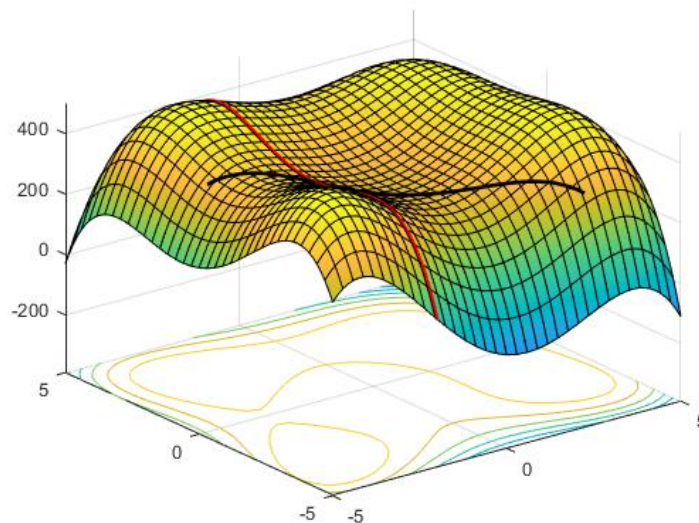
Jelenítsük meg egy külön ábrán a metszetet az y-z síkban!

- ```
> figure(5); plot(ym1,zm1,'r*-' )
```

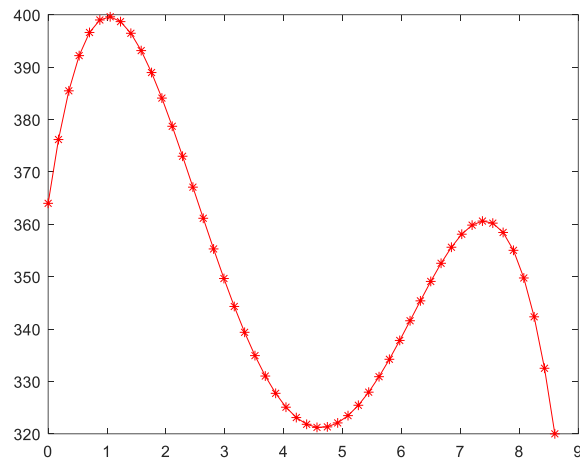


Hasonlóképp felvehetünk metszetet két tetszőleges pont között is. Legyen most a két pont: A(-4,3); B(1,-4). Rajzoljuk fel a két pont közti metszetet a térbeli ábrába és egy külön 2D ábrába a vízszintes távolságok függvényében a magasságokat.

```
> % Tetszőleges irányú metszet: A(-4,1); B(3,-4)
> xm2 = linspace(-4,3,50);
> ym2 = linspace(1,-4,50);
> zm2 = f(xm2,ym2); % magasságok kiszámolása
> figure(1); plot3(xm2,ym2,zm2,'k','LineWidth',2) % berajzolás
```

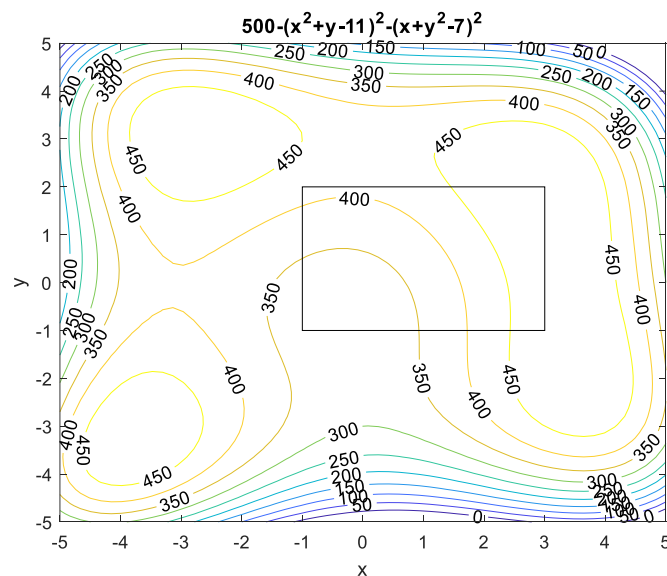


```
> % vízszintes távolságok függvényében magasságok
> dx = diff(xm2); dy = diff(ym2); dt = sqrt(dx.^2+dy.^2);
> tav = [0, cumsum(dt)]; % távolságok a kezdőponttól
> figure(6); plot(tav,zm2,'r*-')
```



Az ábrába berajzolhatjuk egy tervezett épület határvonalát is. Legyen most ez az épület egy 4x3 méteres téglalap alakú épület a (-1,-1) bal alsó sarokponttal. Rajzoljuk be az épület határvonalát a szintvonalas ábrába és a 3D ábrába is! A szintvonalas ábrába könnyen berajzolhatjuk a **rectangle** nevű paranccsal, ahol pozíciónak meg kell adni a bal alsó sarokpont koordinátáit és a méreteket.

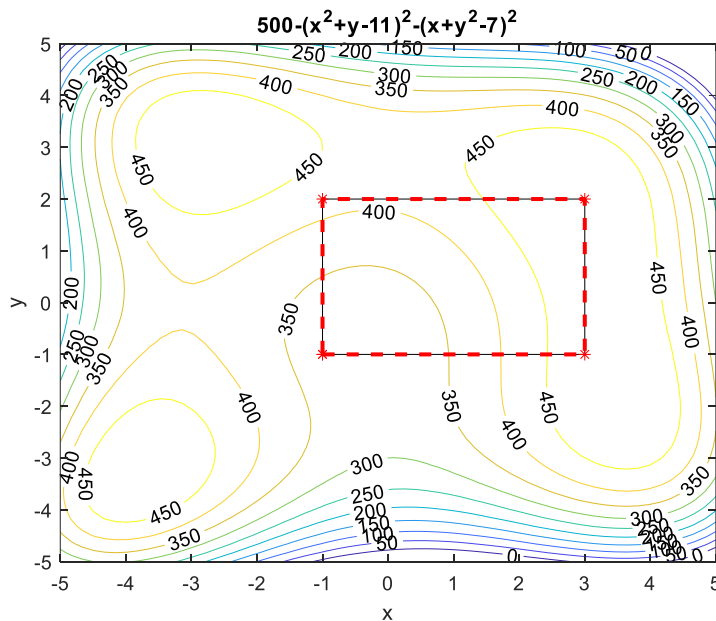
- > % Téglalap alakú épület elhelyezése
- > figure(4); hold on; rectangle('Position',[-1,-1,4,3])



A 3D ábrába történő berajzolás már egy kicsit nehezebb, ugyanis ki kellene számolni magasságokat az épület határvonala mentén. Ehhez be kellene sűríteni a pontokat a zárt alakzat mentén. Ezt úgy tehetjük meg, hogy egy lineáris spline-t, vagyis törtvonalas görbét illesztünk a pontokra, mint többértékű görbék esetén.

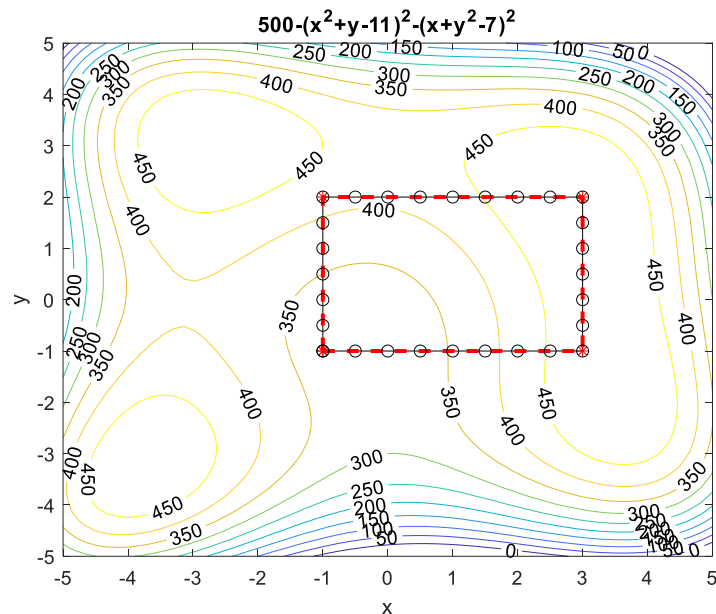
- > % lineáris spline illesztése a sarokpontokra
- > % sarokpontok megadása kezdőponttól a kezdőpontig - zárt görbe!
- > x = [-1,3,3,-1,-1]; y = [-1,-1,2,2,-1]; plot(x,y,'r*')
- > % paraméter a kezdőponttól mért távolság legyen
- > dx = diff(x); dy = diff(y); dt = sqrt(dx.^2+dy.^2);
- > tav = [0, cumsum(dt)] % távolságok a kezdőponttól: 0 4 7 11 14
- > max(tav) % 14
- > xsp = @(t) interp1(tav,x,t)

- > `ysp = @(t) interp1(tav,y,t)`
- > `fplot(xsp,ysp,[0,max(tav)],'r--','LineWidth',2)`



Az illesztett lineáris spline függvények segítségével felvehetünk pontokat az épület oldala mentén és ezekben a pontokban utána kiszámolhatjuk a magasságokat is. Vegyünk fel most fél méterenként pontokat az épület kerületén!

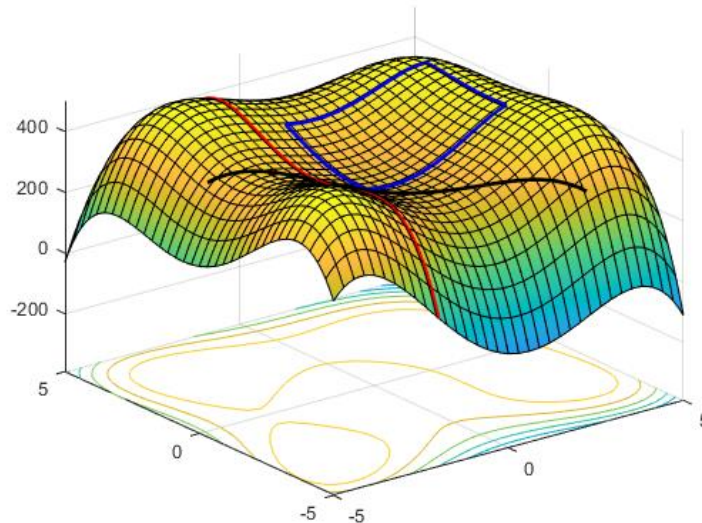
- > `% Pontok felvétele fél méterenként az épület kerülete mentén`
- > `ti = 0:0.5:max(tav);`
- > `xi = xsp(ti);`
- > `yi = ysp(ti);`
- > `plot(xi,yi,'ko')`



Számoljuk ki a pontok magasságait is és rajzoljuk be a térbeli ábrába is az épület körvonalát!

- > `% Térbeli koordináták`
- > `zi = f(xi,yi);`


```
> figure(1); plot3(xi,yi,zi,'b','Linewidth',2)
```

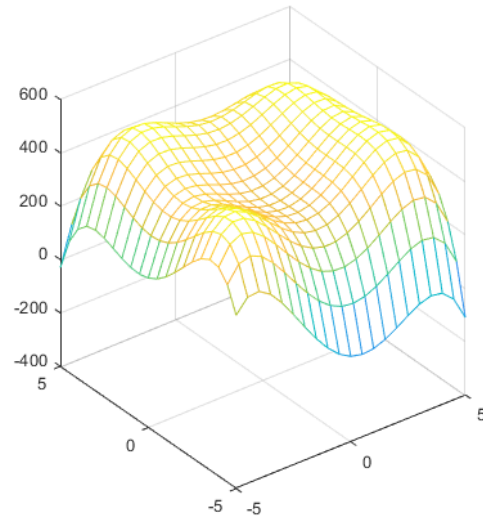
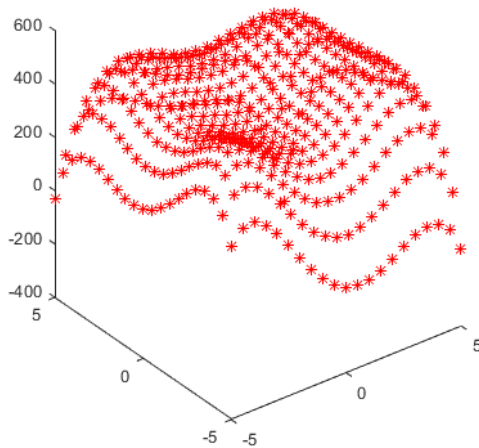


SZABÁLYOS RÁCSHÁLÓBAN ADOTT PONTOK MEGJELENÍTÉSE

Sokszor van szükség egy felmért domborzat ábrázolására. A felmért pontok elhelyezkedhetnek egy szabályos rács pontjaiban, vagy szabálytalanul, elszórtan. Általában ez utóbbi a jellemző eset. Szórt pontokra többnyire valamilyen interpolációs vagy regressziós módszerrel szoktak felületet illeszteni, és azokat függvényként megjeleníteni, vagy egy rácsnáló sarokpontjaira interpolálnak és ezt a rácsot jelenítik meg. Többváltozós interpolációról, regresszióról csak később lesz szó, ezért most csak a szabályos rácsban elhelyezkedő pontok megjelenítéséről fogunk foglalkozni

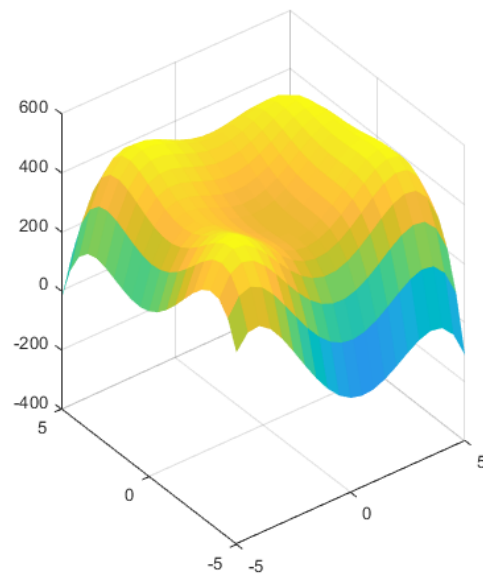
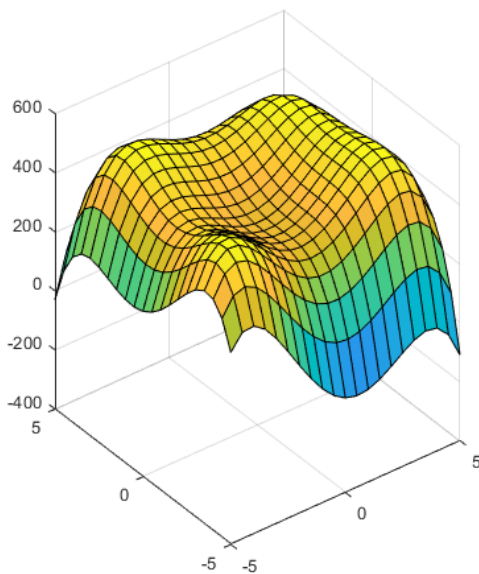
Térjünk vissza az előző $f(x,y)$ függvényünkhöz, és számoljuk ki a függvény értékeit egy szabályos rács pontjaiban! Így 3 mátrixot fogunk kapni az X,Y és Z koordinátákra. Először hozzunk létre x és y irányban egy-egy vektort, olyan sűrűn, amilyen sűrűn a rácsnálót szeretnénk felvenni, majd ebből a **meshgrid** paranccsal lehet elkészíteni a rácsnáló X,Y mátrixait. A Z értékeket pedig ki fogjuk számolni felhasználva az $f(x,y)$ függvényt. A rácsnáló pontjait megjeleníthetjük a **plot3** paranccsal, a rácsot magát pedig a **mesh** paranccsal.

```
> %% Szabályos rácsnálóban adott pontok megjelenítése
> x = -5:0.5:5; y = -5:0.5:5;
> [X,Y] = meshgrid(x,y)
> Z = f(X,Y)
> figure(7); subplot(1,2,1);
> plot3(X,Y,Z, 'r*')
> subplot(1,2,2);
> mesh(X,Y,Z)
```



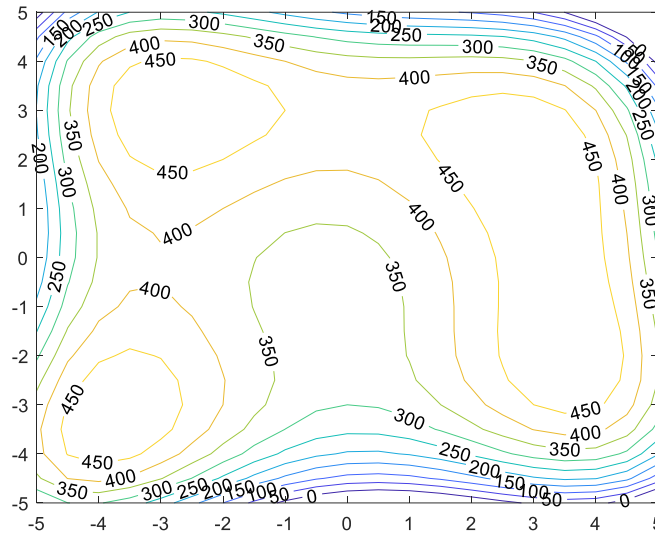
Felületként is megjeleníthetjük a rácsháló pontjait a **surf** paranccsal.

- > `figure(8); subplot(1,2,1)`
- > `surf(X,Y,Z)`
- > `subplot(1,2,2); surf(X,Y,Z, 'EdgeColor', 'none')`



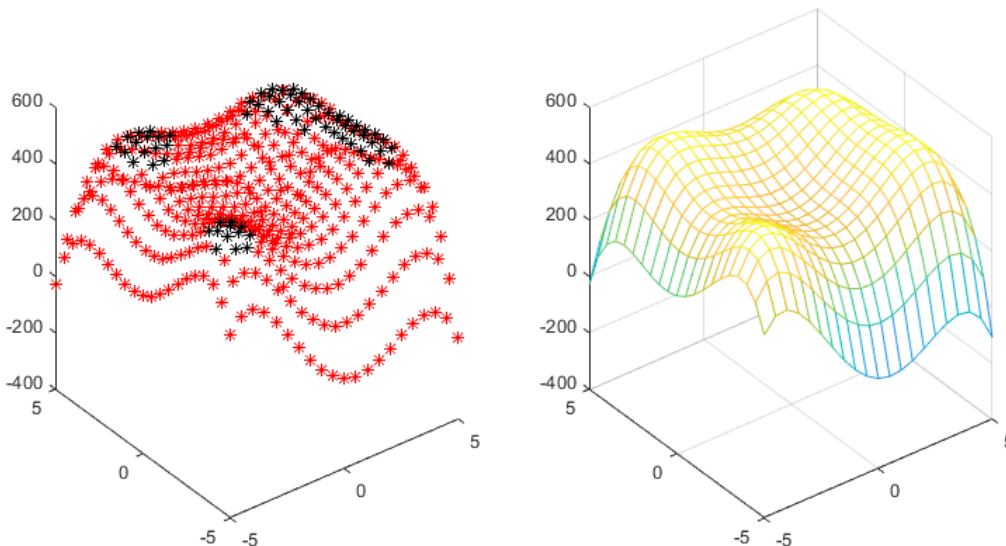
Természetesen szintvonalas ábrán is megjeleníthetjük a pontjainkat, ehhez a **contour** parancs használható. Itt is beállítható a feliratok megjelenítése és a berajzolandó szintvonalak egy vektorban.

- > `figure(9)`
- > `contour(X,Y,Z, 'showText', 'on', 'LevelList', 0:50:600)`



Szükség lehet arra is, hogy bizonyos feltételnek megfelelő térbeli pontokkal dolgozzunk, pl. adott magasság felett/alatt. Ezt az egyváltozós regressziónál is használt logikai indexeléssel itt is megtehetjük. kérdezzük le a 460 m feletti pontokat!

- > % Adott magasság feletti pontok lekérdezése
- > felt = Z>460
- > xh = X(felt); yh = Y(felt); zh = Z(felt);
- > figure(7); subplot(1,2,1); hold on; plot3(xh,yh,zh,'k*')



VEKTORMEZŐK MEGJELENÍTÉSE

Az eddigi esetekben skalár vagy skalár-vektor függvényekkel dolgoztunk, azonban az építőmérnöki gyakorlatban gyakran van szükség vektormezők vagyis vektor-vektor függvények ábrázolására. Ez történhet 2D-ben vagy 3D-ben is, ilyen lehet például geodéziában koordináta transzformációk estében a transzformációhoz felhasznált pontok maradék eltérései, mechanikában egy test adott pontjaiban fellépő

feszültségek nagysága és iránya, vagy folyóban az áramlások iránya, sebessége egy műtárgy körül. A következőkben megnézzük, hogyan lehet ezeket szemléletesen ábrázolni.

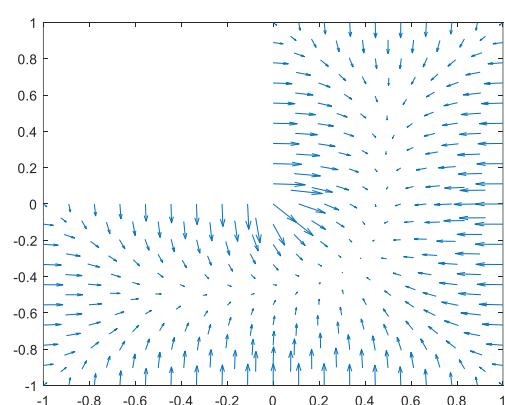
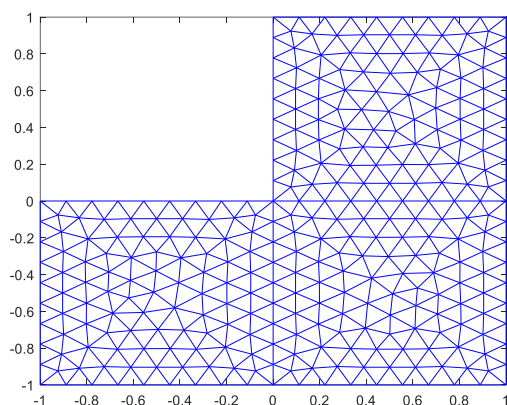
SÍKBELI VEKTORMEZŐ, HÁROMSZÖGHÁLÓ

Mechanikában sokszor használnak végeselemes módszert bonyolult alakú statikusan terhelt elem feszültségi állapotának és alakváltozásának meghatározására. Ilyenkor a modellezendő testet egyszerűbb geometriai elemekre bontják, síkbeli esetben például háromszögekre vagy négyszögekre, térbeli esetben hasábokra vagy tetraéderekre. A modellben az elemek csak csomópontjaikon csatlakoznak egymáshoz. A csomópontok alapján felírható az elemek majd az egész feladat merevségi mátrixa és egy lineáris egyenletrendszer megoldásával kiszámítható a csomópontok elmozdulása, majd a közelítő feszültségek. Itt a feladat megoldásával nem foglalkozunk, csak egy készen kapott síkbeli háló és az eredmények megjelenítésével.

Töltsük be Matlab-ba a `fem_triangles.mat` fájlt! Ebben a fájlban 6 mátrix típusú változó van elmentve. Az `nx` és `ny` oszlopvektorokban a csomópontok koordinátái találhatóak. Itt a síkbeli elemet háromszögekre bontották, a `T` mátrix 3 oszlopból áll, amelyekben háromszögekhez tartozó csomópontok sorszáma van benne. Az `u` vektorban az eredményül kapott csomóponti feszültségek nagysága, az `ux`, `uy` vektorokban pedig a megoldás gradiense található.

Ábrázoljuk síkban a háromszöghálót (`triplot` parancs) és a gradiens vektorokat (`quiver` parancs)!

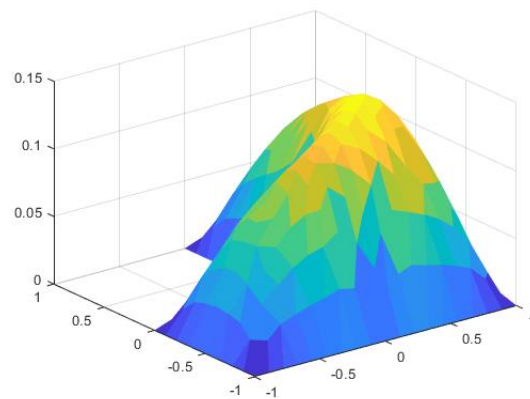
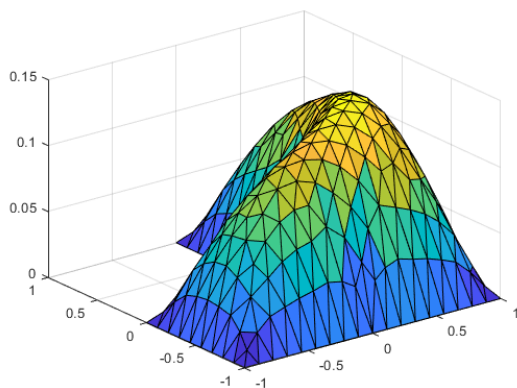
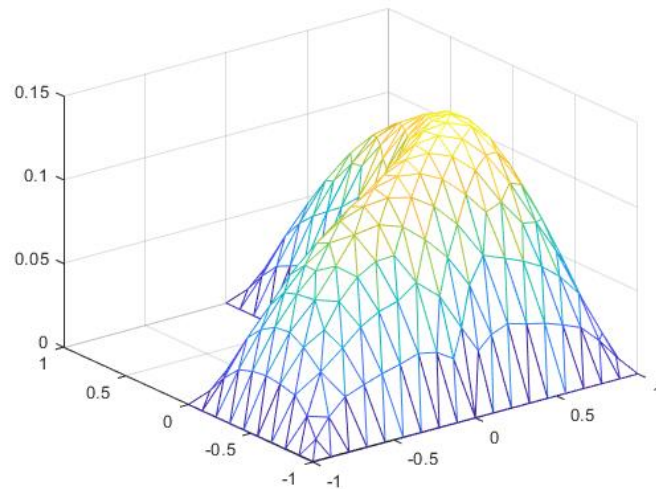
```
> % Plotting Finite Element Mesh
> clc; clear all; close all;
> load fem_triangles.mat
> % T - triangle node numbers
> % nx,ny - x,y coordinates of the nodes
> % u - solution at nodes
> % ux,uy - solution gradients
> figure(1); triplot(T,nx,ny) % háromszögháló ábrázolása
> figure(2); quiver(nx,ny,ux,uy) % gradiensek ábrázolása vektorokkal
```



Térben is ábrázolhatjuk a megoldásokat, akár térbeli háromszöghálóval (`trimesh` parancs), akár felületként (`trisurf` parancs).

```
> figure(3); trimesh(T,nx,ny,u) % térbeli háromszög háló
```

- > `figure(4); trisurf(T,nx,ny,u) % térbeli felület háromszögekből`
- > `figure(5); trisurf(T,nx,ny,u,'EdgeColor','none')`



TÉRBELI ÁRAMLÁSOK MEGJELENÍTÉSE¹

Az építőmérnöki gyakorlatban nem csak 2D-s, hanem 3D-s vektormezők megjelenítése is szükséges lehet, például a folyók műtárgyak körüli áramlásának a modellezésekor. Nézzünk most egy ilyen példát, hogy ebben az esetben milyen lehetőségeink vannak. A példa a Mosoni-Duna fölötti Duna egyik gázlójának az áramlási képe két sarkantyúval.

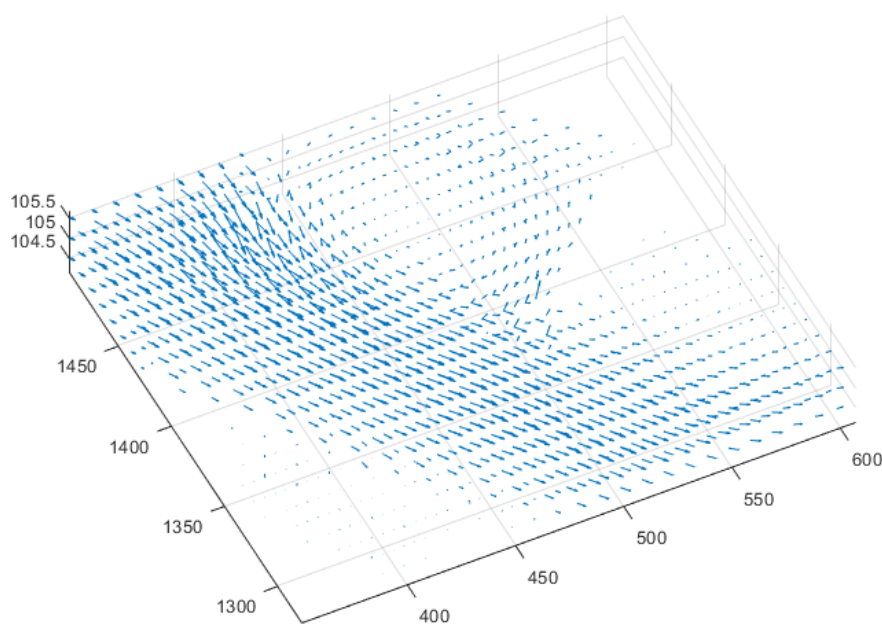
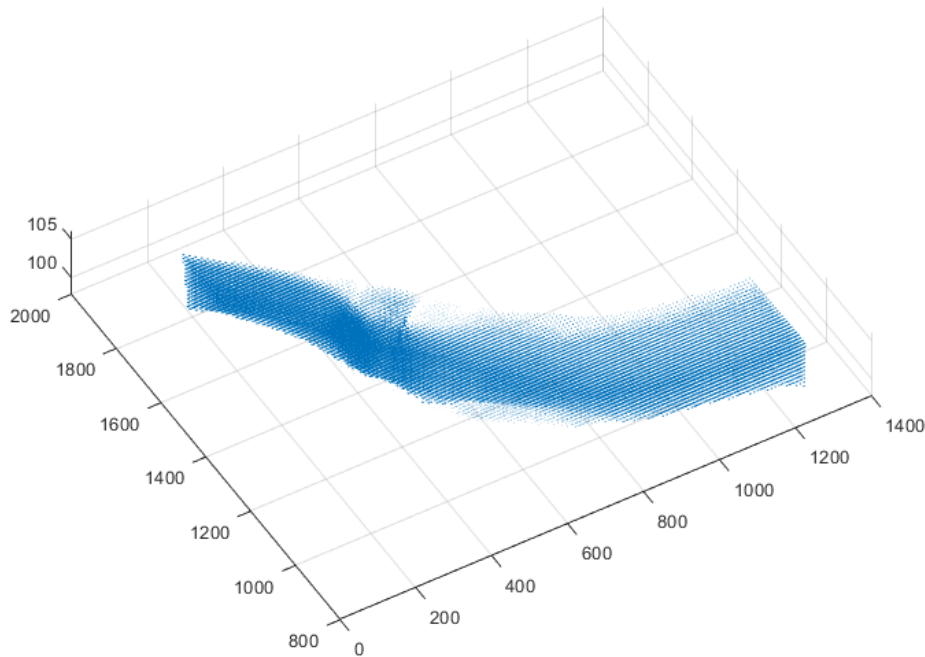
Két állományban találhatóak az adatok. Az `'aramlat.mat'` Matlab adatfájl egy 217x195x38 méretű, szabályos téglatest mátrixban tárolja az áramlási sebességvektorokat (5 méter x 5 méter x 0.25 m felbontással). A meder szabálytalan, ezért vannak cellák, amelyek a tartományon kívülre esnek. Ezekben a sebesség 0,0,0. Az `'aramlat.txt'` egy egyszerűsített szöveges változata az előző állománynak, amelyben csak a nem nulla sebességű elemek vannak benne. A hat oszlopban a koordináták (x, y, z) és a sebességvektorok összetevői (ux, uy, uz).

¹ Kiegészítő anyag

Olvassuk be először a szöveges állományt, amelyben lényegesen kevesebb adat van, így gyorsabban meg lehet jeleníteni a 3D sebességvektorokat.

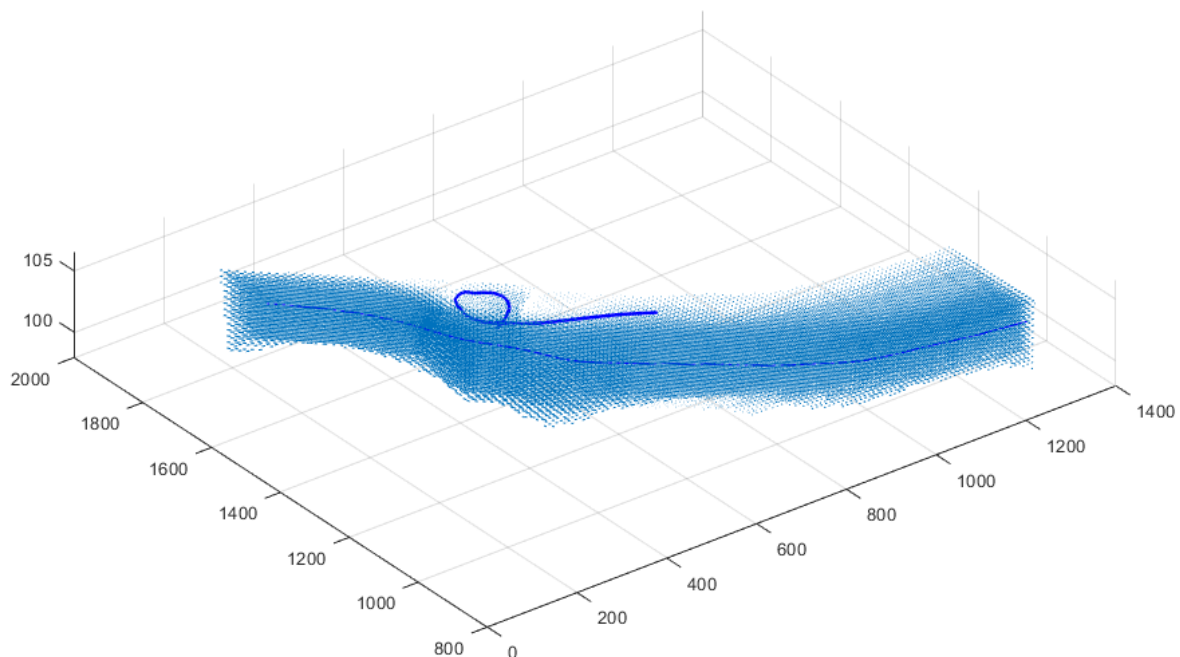
```
> % 3D vektorok  
> clc; clear all; close all;  
> adat = load('aramlat.txt');  
> x1 = adat(:,1);y1 = adat(:,2);z1 = adat(:,3);  
> ux1 = adat(:,4);uy1 = adat(:,5);uz1 = adat(:,6);  
> figure(1)  
> quiver3(x1,y1,z1,ux1,uy1,uz1) % nem áttekinthető, túl sűrű  
> daspect([25 25 1]) % célszerű módosítani a tengelyek arányait
```

Sajnos ennyi adatnál ez nem túl áttekinthető, túlságosan sűrű, nem is látszódnak a vektorok, csak, ha belenagyítunk.



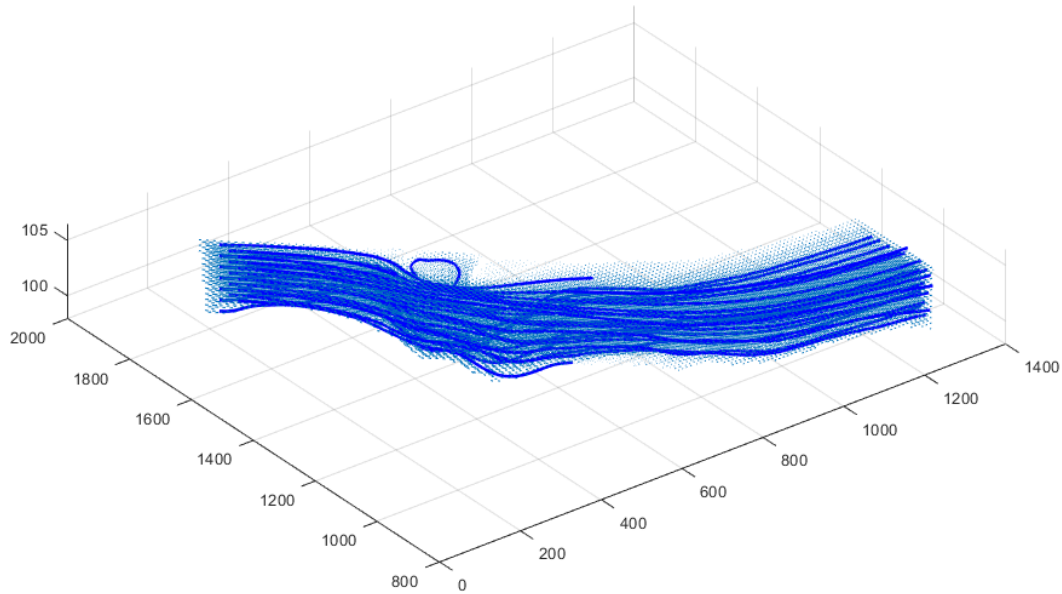
Jobban megjeleníthetjük az áramlásokat, ha áramvonalakat ('streamline') rajzolunk az ábrába. Az áramvonalak megjelenítéséhez azonban szabályos téglatest alakú rácsban megadott adatokat lehet csak használni, így olvassuk most be az 'aramlat.mat' adat állományt, amiben 6 változó található, mindegyik 3D rácsháló formájában megadva. Áramvonalakat a **streamline** paranccsal jeleníthetünk meg. Meg kell adni a koordinátákat, sebességvektorokat és azt a pontot, ahonnan szeretnénk indítani az áramlási vonalat. A tulajdonságait (pl. szín, vastagság) a **set** paranccsal módosíthatjuk.

```
> % áramvonalak berajzolása
> load aramlat.mat
> streamline(x,y,z,ux,uy,uz,200,1700,105)
> h = streamline(x,y,z,ux,uy,uz,480,1400,104.5)
> set(h,'Linewidth',2)
```

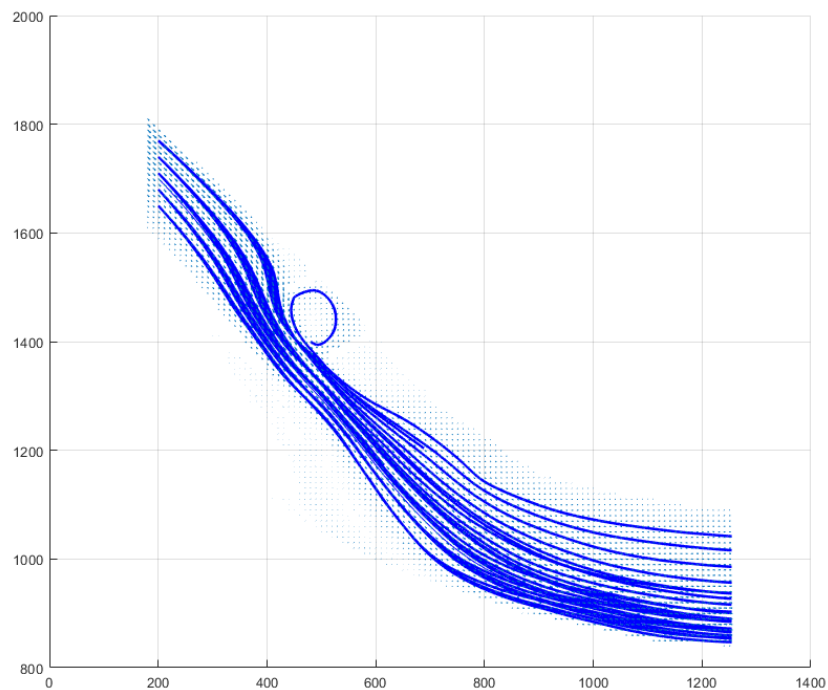


Jobban láthatjuk az áramlás dinamikáját, ha felveszünk egy kezdő metszetet és ott egy rácshálóban több kiinduló pontot is megadunk. Legyen most ez a metszet az $x = 200$ m-nél, 5×7 -es mátrixban (y irányban 30 és z irányban 1 méterenként).

```
> % Több áramvonal együttes berajzolása
> x0 = 200
> y0 = 1650:30:1800 % 5 érték
> z0 = 100:1:106 % 7 érték
> [XV, YV, ZV] = meshgrid(x0,y0,z0);
> H = streamline(x,y,z,ux,uy,uz,XV,YV,ZV);
> set(H,'Linewidth',2)
```

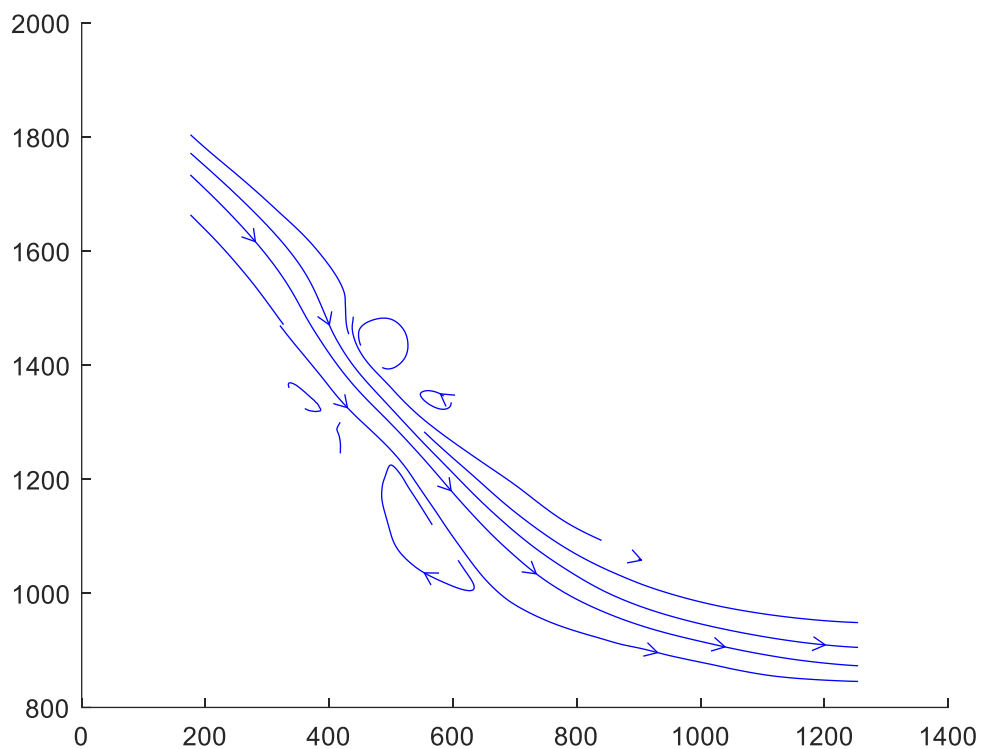


Felülnézetből elforgatva ugyanez:



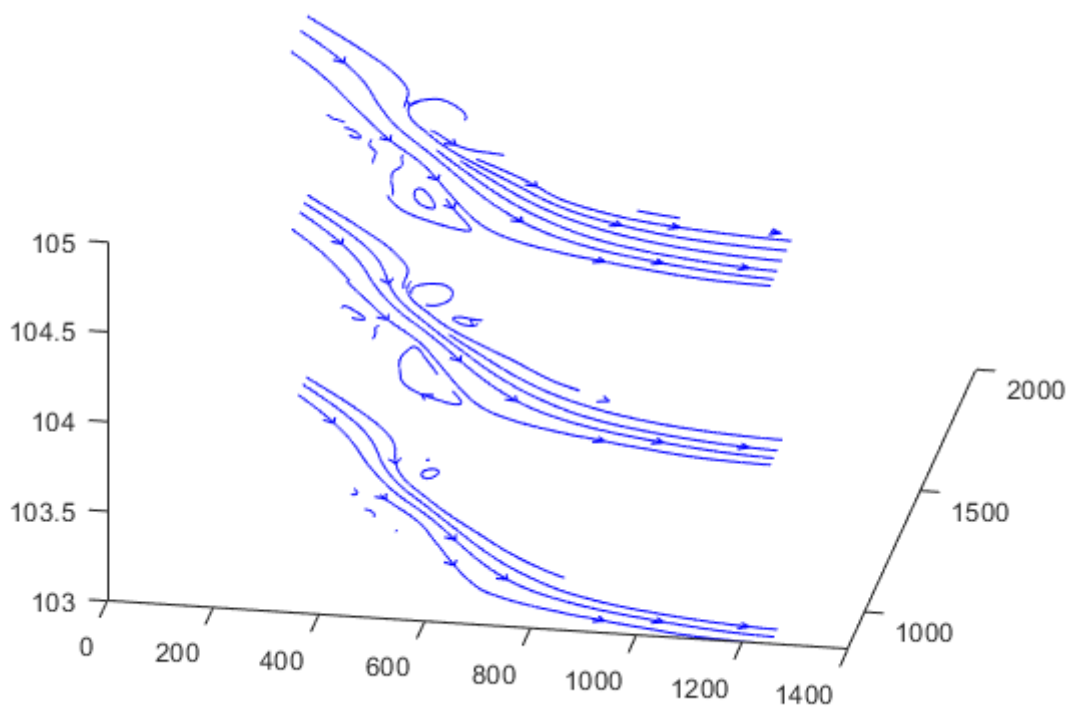
Egy másik szemléletes megjelenítési módszer, ha metszeteket készítünk, ehhez a **streamslice** parancs használható. Készítsünk egy metszetet 104 méteres magasságban! Itt meg kell adni a metszet helyét. természetesen nem csak z irányú metszet készíthető.

```
> % Metszetek készítése
> figure(2)
> streamslice(x,y,z,ux,uy,uz, [], [], 104);
```

Egy vektorban akár több értéket is megadhatunk:

- > figure(3)
- > streamslice(x,y,z,ux,uy,uz, [], [], 103:105);



A FEJEZETBEN HASZNÁLT FÜGGVÉNYEK

plot3	- 3D adatsorok megjelenítése
fplot3	- térbeli görbék megjelenítése paraméteresen
subplot	- Egymás alatti/melletti ábrák megjelenítése mátrix szerűen
fsurf, ezsurf	- Függvénnyel megadott felület megjelenítése
fcontour, ezcontour	- Függvénnyel megadott felület szintvonalainak a megjelenítése
rectangle	- Téglalap rajzolás, megívás bal alsó sarok és oldalhosszak megadásával: <code>rectangle('Position',[xmin,ymin,a,b])</code> ,
meshgrid	- Rácsháló készítése
mesh	- Felület megjelenítése rácshálószerűen 3D adatsor alapján
surf	- Felület megjelenítése 3D rácshálóban adott adatsor alapján
contour	- Szintvonalak megjelenítése 3D rácshálóban adott adatsor alapján
triplot	- Háromszögek megjelenítése
quiver	- 2D vektormező megjelenítése
trimesh	- Háromszöghálóban adott felület megjelenítése hálóként
trisurf	- Háromszöghálóban adott felület megjelenítése
quiver3	- 3D vektormező megjelenítése
streamline	- áramlási vonalak megjelenítése 3D-ben
streamslice	- áramlási vonalak megjelenítése metszetben