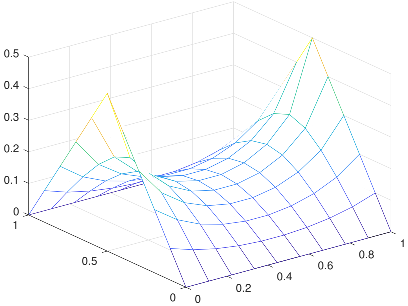
Minimális felület

Adott egy $\Omega \subset \mathbb{R}^2$ tartomány, Γ peremmel. Legyen $r : \Gamma \to \mathbb{R}$ adott függvény. Olyan $q : \overline{\Omega} \to \mathbb{R}$ függvényt keresünk, mely a peremen egybeesik r-rel, és amelynek a gráfja minimális felszínű.
Írjunk egy Matlab-kódot, mely előállítja a felület közelítését akkor, ha $\Omega = [0,1]^2$ és adott az r függvény.

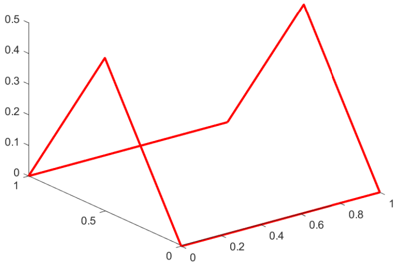


$\Omega = [0,1]^2$ és $r(x,y) = \frac{1}{2} - \left|y - \frac{1}{2}\right|$

1. lépés

Szükségünk lesz egy függvényre, mely előállítja a ,,keretet'' (az r függvényt). Egészítsük ki a lenti kódot úgy, hogy pl. az ábrán látható keret pontjait állítsa elő! (Nem kell rajzolnia.)

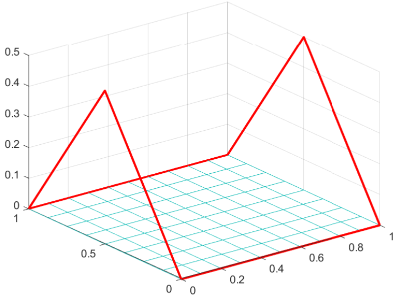
```
function z=frame(x,y)
    z=
end
```



2. lépés

Rácsozzuk be a $[0,1]^2$ tartományt az x-tengely mentén m (pl. 11), az y-tengely mentén n (pl. 9) osztópontot használva!

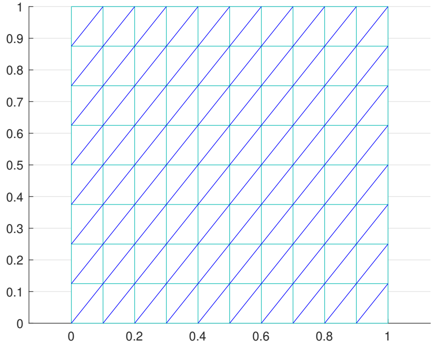
```
m=11; n=9;
x=linspace(0,1,m);
y=linspace(0,1,n);
[X,Y]=meshgrid(x,y);
```



3. lépés

Az így kapott kis téglalapok mindegyikét két háromszögre osztjuk, a felületet minden ilyen kicsi háromszög fölött egy síkkal fogjuk közelíteni.

A közelítő síkot egyértelműen meghatározza, hogy a kis háromszög csúcspontjaiban mennyi a függvény értéke.



A felületnek így egy háromszögenként lineáris interpolációját kapjuk, ehhez minden egyes rácspontban meg kell mondanunk a függvényértéket. Legyen Z az ezen értékek mátrixa.

4. lépés

A függvényértékek a perempontokban (tehát Z első és utolsó sorában és oszlopában) adottak (a hártya felfeszül a keretre). A feladat: találjuk meg a belső rácspontokban az értékeket úgy, hogy a felület minimális legyen. Összesen $(m-2)(n-2)$ értéket keresünk.

A Matlab optimalizáló függvényei a célfüggvény változóját vektoralakban várják, a minimumhelyet is ilyen alakban szolgáltatják.

Egészítsük ki a lenti függvényt úgy, hogy egy alkalmas méretű z vektor, az X és Y tömbök, valamint a keretet előállító függvény ismeretében feltöltse a Z tömböt.

```
function Z=buildZ(X,Y,z,fun)
    Z=zeros(size(X));
    Z(1,:)=fun(X(1,:),Y(1,:));
    .....
    Z(2:end-1,2:end-1)=reshape(z,size(X)-2);
end
```

5. lépés

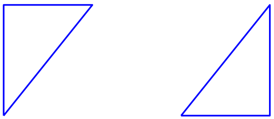
Össze kell adnunk a kis háromszögek fölötti síkdarabok területét.

Ha a tartomány kis háromszögének csúcspontjai (x_i, y_i), $i = 1, 2, 3$, és ezekben a pontokban a függvény értéke rendre z_1, z_2, z_3, akkor a megfelelő síkdarab területe:

$$A = \frac{1}{2} \left\| \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \times \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{bmatrix} \right\|$$

A vektoriális szorzatot a Matlab `cross` függvényével számíthatjuk.
Ezt az összes kis háromszögre egyszerre is elvégezhetjük.

A lenti függvény az összes olyan háromszög fölött összegzi a területet, melyek olyan állásúak, mint a bal oldali. Egészítse ki a kódot úgy, hogy a másik típusú háromszögek fölött is elvégezze az összegzést. A bemenő értékek között szerepel az a függvény, amely a z vektorból előállítja a Z mátrixot, illetve a keretet előállító függvény.



```
function A=sum_area_tr(X,Y,z,fun1,fun2)
    Z=fun1(X,Y,z,fun2);
    R=cat(3,X,Y,Z);
    c1=cross(R(1:end-1,1:end-1,:)-R(2:end,1:end-1,:),...
       R(2:end,2:end,:)-R(2:end,1:end-1,:));

    A= sum(vecnorm(c1,2,3),'all');
    A=A/2;
end
```

6. lépés

Állítsuk elő a minimalizálandó függvényt, hívjuk meg a minimalizálót, majd ábázoljuk a felületet!

```
objf=@(z) sum_area_tr(X,Y,z,@buildZ,@frame);
z0=zeros((n-2)*(m-2),1);
zopt=fminunc(objf,z0);
Z=buildZ(X,Y,zopt,@frame);
figure;mesh(X,Y,Z)
```

Próbáljuk ki a kódot különböző keretekkel, különböző finomságú rácsokon!