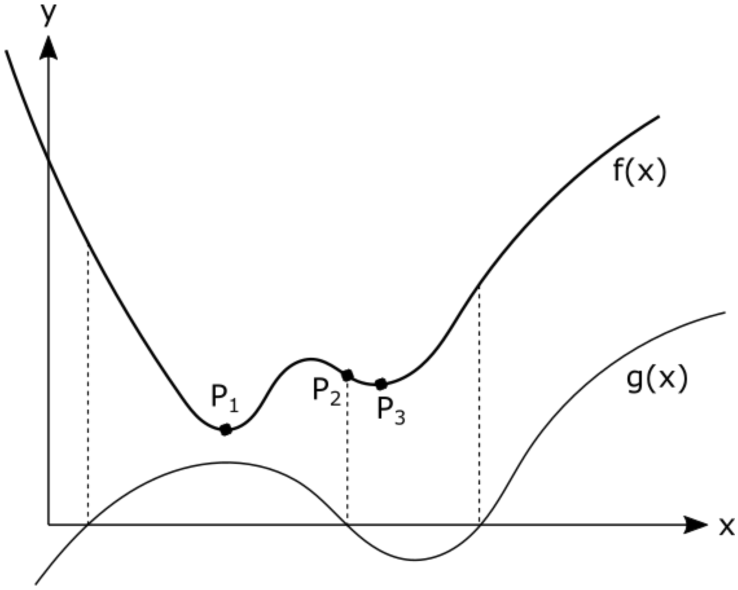
16. OPTIMALIZÁCIÓ MEGKÖTÉSEKKEL

A feltételes szélsőérték feladatok esetében úgy keressük a függvény minimumát, hogy közben a pontoknak ki kell elégíteniük valamilyen megkötést, feltételt is. Itt lehet egy vagy több feltétel, ezek lehetnek egyenletekkel vagy egyenlőtlenségekkel megadva (lásd a következő ábrán), lehetnek lineárisak vagy nemlineárisak is. A különböző esetekben más-más módszert lehet alkalmazni (pl. Lagrange-módszer, büntetőfüggvény módszere, Karush-Kuhn-Tucker-feltételek, lineáris programozás).



1 Az f(x) függvény minimuma: P_1 - megkötés nélkül, P_2 - g(x)=0 megkötéssel, P_3 - g(x)<0 megkötéssel

EGYENLŐSÉGGEL ADOTT MEGKÖTÉS

Válasszuk külön a megkötéses szélsőérték keresés feladatai közül azokat, amelyeknél csak egyenlőséggel adott feltételek vannak, illetve azokat, amelyeknél egyenlőtlenséggel adott feltételeket is találunk. Az előbbiek kezelése jóval egyszerűbb, történhet például Lagrange-módszerrel vagy büntetőfüggvény módszerrel is. Nézzünk egy példát ilyen típusú feladatra!

Egy acél tárgyakat gyártó üzem nyereségét szeretnénk maximalizálni. A költségeink legnagyobb részét a munkabér teszi ki és mellette az acél nyersanyag költsége. Egy munkaóra költsége 20$ és egy tonna acél 170$-ba kerül. A profitot a következő függvénnyel tudjuk megadni:

$$f(h,s) = 200 \cdot h^{2/3} \cdot s^{1/3}$$

ahol *h* a munkaórák száma, *s* pedig az acél mennyisége tonnában.

Ezen kívül van egy rendelkezésre álló költségkeret: 25000 $. Ebből a költségkeretből szeretnénk kihozni a maximális nyereséget. A költségekre a következő függvényt tudjuk felírni:

$$20 \cdot h + 170 \cdot s$$

Ahhoz, hogy a profitot maximalizáljuk felhasználjuk a teljes rendelkezésre álló költségkeretet. Ebben az esetben a következő feltételnek kell teljesülnie:

$$20 \cdot h + 170 \cdot s = 25000$$

A megoldáshoz adjuk meg ezt a megkötést nullára rendezett alakban:

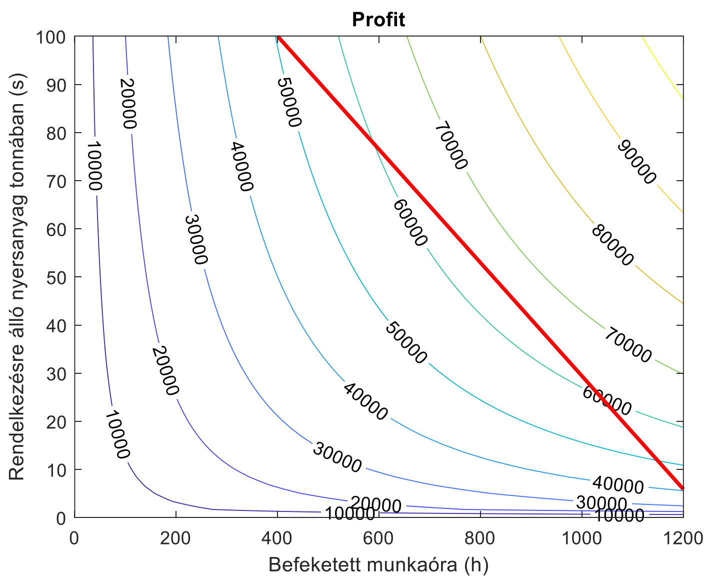
$$g(h, s) = 20 \cdot h + 170 \cdot s - 25000 = 0$$

<u>Megjegyzés:</u> Ebben a feladatban a megkötés egy egyenes egyenlete, de a feladat ugyanígy megoldható más nemlineáris egyenlettel adott megkötés esetében is, a megkötés 0-ra rendezése után.

Most az f kétváltozós felület értékét szeretnénk maximalizálni (célfüggvény), úgy, hogy teljesül a g függvényben megadott feltétel. A megoldáshoz először ábrázoljuk az f célfüggvényt szintvonalakkal és a g feltételt egy síkbeli görbével.

A célfüggvény megjeleníthető egy kétváltozós felületként akár térben az **fsurf** paranccsal, vagy az **fcontour** vagy **ezcontour** paranccsal szintvonalas formában 2D-ben. Az **ezcontour** használata esetén tudjuk feliratozni is a szintvonalakat a **set** paranccsal, a **ShowText** tulajdonság bekapcsolásával (**on**), a szintvonalközt pedig a **LevelStep** vagy **LevelList** paraméterek megadásával változtathatjuk. A megkötésünk implicit alakban van megadva, ezt az **fimplicit** paranccsal rajzolhatjuk be, amennyiben előtte 0-ra rendeztük.

```
> clc; clear all; close all; format shortG;
> % Profit - célfüggvény
> f = @(h,s) 200*h.^(2/3).*s.^(1/3)
> % Költségkeret: 20$/óra, 170$/tonna: 20*h+170*s = 25000$
> g = @(h,s) 20*h+170*s-25000
> % szintvonalas megjelenítés
> figure(1); h1 = ezcontour(f,[0 1200 0 100])
> set(h1,'ShowText', 'on','LevelStep', 10000)
> % implicit alakban adott megkötés megjelenítése
> hold on; fimplicit(g,[0 1200 0 100],'r','LineWidth',2)
> xlabel('Befeketett munkaóra (h)')
> ylabel('Rendelkezésre álló nyersanyag tonnában (s)')
> title('Profit')
```



LAGRANGE-MÓDSZER

Hogyan tudunk egy ilyen problémát megoldani? Az ábra alapján a maximális nyereség a megadott feltétel mentén valahol 60-70000$ környékén lesz. Méghozzá pontosan ott, ahol a feltételként megadott egyenes érintené a maximum értékéhez berajzolt szintvonalat. Ebben a pontban a célfüggvény szintvonalainak normálisa vagyis a felület gradiense párhuzamos a feltétel normálisával. A nagyságuk eltérhet egy λ arányossági tényezővel (lásd a lenti ábrát). Jelen esetben ez az összefüggés így írható fel[1]:

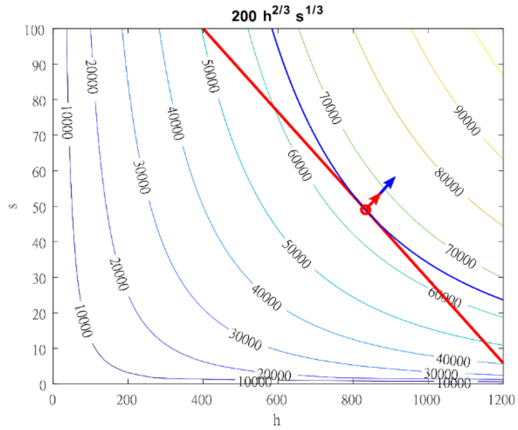
$$\nabla f(h,s) = \lambda \cdot \nabla g(h.s)$$

Ezt rendezzük nullára:

$$\nabla f(h,s) - \lambda \cdot \nabla g(h.s) = 0$$

A gradiensek egyezésén kívül természetesen a megkötéseknek is teljesülnie kell, vagyis jelen esetben a megoldásnak rajta kell lennie a megadott egyenesen:

$$g(h,s) = 0$$



Kétváltozós esetben, amennyiben egy megkötésünk van, akkor két egyenlet írható fel a gradiensek egyezésére és egy a megkötésre. Ilyenkor egy 3 egyenletből álló egyenletrendszert fogunk kapni 3 ismeretlennel, ahol 2 ismeretlen a 2 keresett változó, a harmadik pedig az arányossági tényező, λ.

Ezen alapul az ún. Lagrange-módszer, a λ arányossági tényezőt pedig Lagrange-multiplikátornak nevezzük. A gradiensek párhuzamosságán kívül teljesülnie kell még az eredeti feltételnek is. A Lagrange-módszer szépsége abban rejlik, hogy egy formulával megadható mind a gradiensek párhuzamossága, mind a megkötések teljesülése is.

Lagrange-módszer használható megkötéses minimum és maximum hely keresésére is, azonban csak egyenlőséggel megadott megkötés(ek) esetén, egyenlőtlenség esetében nem. A Lagrange-módszernél a megkötéses optimalizációt egy egyenletrendszer megoldására vezetjük vissza (ami lehet lineáris vagy nemlineáris).

Nézzük meg általánosan a Lagrange-módszert. Az eredeti feladat helyett írjuk fel az alábbi függvény megkötés nélküli minimalizálását:

$$L(x,\lambda) = f(x) - \lambda^T \cdot g(x) = f(x) - \sum_{i=1}^{m} \lambda_i \cdot g_i(x)$$

[1] A párhuzamosság nem szükséges követelmény, csak ha $\nabla f \neq 0$ az adott helyen. Amikor a feltétel a felület lokális szélsőértékét metszi, akkor ott $\nabla f = 0$ lesz, így nem lesz szükségszerűen merőleges a feltétel a szintvonalakra. A $\nabla f = \lambda \cdot \nabla g$ egyenlet azonban ilyenkor is igaz lesz. Erre egy szemléletes példa: https://math.stackexchange.com/questions/2578903/lagrange-multipliers-tangency

ahol λ_i-k a Lagrange-féle multiplikátorok (szorzók), x vektor pedig az ismeretleneket tartalmazza[2]. A minimum szükséges feltétele a parciális deriváltak eltűnése, azaz

$$\frac{d}{dx}L(x,\lambda) = \frac{d}{dx}f(x) - \lambda^T \cdot \frac{d}{dx}g(x) = \nabla f(x) - \sum_{i=1}^{m} \lambda_i \cdot \nabla g_i(x) = 0$$

$$\frac{d}{d\lambda_i}L(x,\lambda) = g_i(x) = 0$$

$i = 1,2,\cdots,m$, ahol m az egyenlőséggel adott megkötések száma. Az x változóban szereplő ismeretlenek szerinti deriváltak adják meg a gradiensek párhuzamosságát, a λ szerinti derivált pedig az eredeti megkötést magát.

A szélsőérték szükséges feltétele, hogy a fenti egyenletek teljesüljenek. A minimum elégséges feltétele, hogy az $L(x,\lambda)$ függvény *Hesse mátrixa* pozitív definit legyen, azaz a mátrix sajátértékei pozitívak legyenek a szélsőérték helyén. Amennyiben a Hesse mátrix a megoldás helyén negatív definit, akkor a függvénynek lokális maximuma van a megadott feltétel mellett.

Oldjuk meg a példában adott megkötéses feladatot Lagrange módszer használatával! A nemlineáris egyenlettel megadott megkötés esetén a következő Lagrange függvényt tudjuk felírni:

$$L(h,s,\lambda) = 200 \cdot h^{\frac{2}{3}} \cdot s^{\frac{1}{3}} - \lambda \cdot (20 \cdot h + 170 \cdot s - 25000)$$

A minimum szükséges feltétele a parciális deriváltak eltűnése, azaz a következő egyenletrendszer megoldása:

$$\frac{dL(h,s,\lambda)}{dh} = 0 \qquad \frac{dL(h,s,\lambda)}{ds} = 0 \qquad \frac{dL(h,s,\lambda)}{d\lambda} = 0$$

A deriváltakat kiszámítva a következő egyenletrendszert kapjuk:

$$\frac{400}{3} \cdot \frac{s^{1/3}}{h^{1/3}} - 20 \cdot \lambda = 0$$

$$\frac{200}{3} \cdot \frac{h^{2/3}}{s^{2/3}} - 170 \cdot \lambda = 0$$

$$20 \cdot h + 170 \cdot s - 25000 = 0$$

Látjuk, hogy ezt a fajta felírást használva egyszerre megkaptuk a gradiensekre vonatkozó két feltételt (első 2 egyenlet) és magát a megkötést is visszakaptuk egyben (3. egyenlet). Most már csak ezt az egyenletrendszert kell megoldanunk! Természetesen a Matlabot is segítségül hívhatjuk a deriváltak meghatározásához szimbolikus számításokat alkalmazva.

```
> L = @(h,s,lambda) f(h,s)-lambda*g(h,s);
> syms h s lambda
> LS = L(h,s,lambda)
> % 200*h^(2/3)*s^(1/3) - lambda*(20*h + 170*s - 25000)
```

[2] A λ-t tartalmazó tagot lehet hozzáadni is az eredeti függvényhez, a feladat ugyanúgy megoldható, csak λ előjele fog megváltozni.

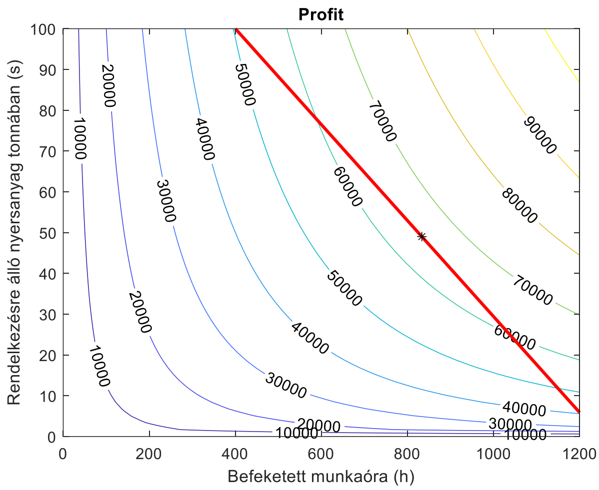
```
> dh=diff(LS,h) % (400*s^(1/3))/(3*h^(1/3)) - 20*lambda = 0
> ds=diff(LS,s) % (200*h^(2/3))/(3*s^(2/3)) - 170*lambda = 0
> dl=diff(LS,lambda) % 25000 - 170*s - 20*h = 0
```

Ez egy nemlineáris egyenletrendszer h, s, λ változókra. Írjuk fel az egyenletrendszert Matlabban és oldjuk meg a korábban tanultak szerint az **fsolve** használatával! Először egy egyenletrendszerbe kell betenni a deriváltakat, utána pedig a szimbolikus kifejezéseket vissza kell alakítani függvénnyé. Figyeljünk oda, hogy az **fsolve** a többváltozós függvényeket csak vektorváltozós alakban tudja kezelni, ezért a többváltozós függvényt itt is a vektorizálni kell!

```
> FLsym = [dh;ds;dl]
> %  (400*s^(1/3))/(3*h^(1/3)) - 20*lambda
> % (200*h^(2/3))/(3*s^(2/3)) - 170*lambda
> %                    25000 - 170*s - 20*h
> FL = matlabFunction(FLsym) % FL = @(h,lambda,s) [lambda.*2.0e+1+...
> % A nemlineáris egyenletrendszer vektorizálása
> FL = @(v) FL(v(1),v(2),v(3)) % @(h,lambda,s) -> v=[h,lambda,s]
```

A megoldáshoz kezdőértékekre is szükségünk van. Amire oda kell figyelni, hogy a **matlabFunction** használata során milyen sorrendben írta be a Matlab a változókat az FL függvényhez, a kezdőértékeket is ilyen sorrendben kell megadni és a végeredményt is így kapjuk. Most a változók h, λ, s sorrendben szerepelnek az FL függvényben. A szintvonalas ábra segítségével tudunk kezdőértéket választani a h, s változóknak (ahol a legnagyobb értékű szintvonalat megközelíti a görbe). A lambda változóhoz nem tudunk kezdőértéket rendelni, válasszuk ennek értékét most 1-nek.

```
> x0 = [800; 1; 50] % kezdőérték h-ra, lambda-ra, s-re
> sol = fsolve(FL,x0,optimset('Display','iter'))
> h1 = sol(1) % 833.33
> l1 = sol(2) % 2.5927
> s1 = sol(3) % 49.02
> plot(h1,s1,'k*');
> zopt1 = f(h1,s1) % 64819
```



A kapott eredmények szerint akkor lesz maximális a profitunk (64819$), amikor a meglévő 25000$ költségvetésből 833 munkaórát és 49 tonna acélt fizetünk ki. Ami még érdekes, hogy a λ sem pusztán egy arányossági tényező (2.59), hanem levezethető, hogy ez a nyereség megváltozásának nagyságát adja meg a befektetett összeg függvényében. Amennyiben a jelenleginél 1 dollárral többet fektetnénk be, akkor 2,59 dollárral nőne a nyereségünk!

BÜNTETŐFÜGGVÉNY MÓDSZERE

A Lagrange-módszerrel történő megoldás esetén szükség volt a gradiensek számítására, ami nem minden esetben egyértelmű és könnyen számítható. Vannak olyan megoldási módszerek, mint pl a büntetőfüggvény módszere, ahol nincs szükség gradiensek számítására, a nélkül is meg tudjuk oldani az egyenlőséggel adott megkötéses optimalizálás feladatát. Ehhez nézzünk most egy másik példát! Adott a következő felület a $-0.5 \leq x \leq 0.5;\ -0.5 \leq y \leq 1$ tartományon:

$$f(x,y) = \frac{\sin(2\cdot\pi\cdot x)\cdot\cos(5\cdot y)}{(2+x^3)\cdot(1+2\cdot y^5)}$$

Határozza meg a minimumot a megadott egyenes mentén!

$$y = 0.6\cdot x + 0.3$$

A megkötés most egy explicit formában megadott lineáris egyenlőség. Ha megnézzük az f célfüggvényt, akkor látjuk, hogy most egy jóval bonyolultabb függvényről van szó, mint az előző példában. Természetesen itt is lehetne a Lagrange-módszert használni, de jóval bonyolultabb egyenleteket kapnánk a deriválás után (a feladat Lagrange-módszerrel történő megoldása a gyakorló feladatok között megtalálható). Most azonban nézzük meg a büntetőfüggvény módszerét, ami nem igényli a gradiensek kiszámítását. Ez a megoldás mind lineáris, mind nemlineáris megkötés esetén alkalmazható, de csak egyenlőséggel adott megkötésnél, egyenlőtlenség esetén nem.

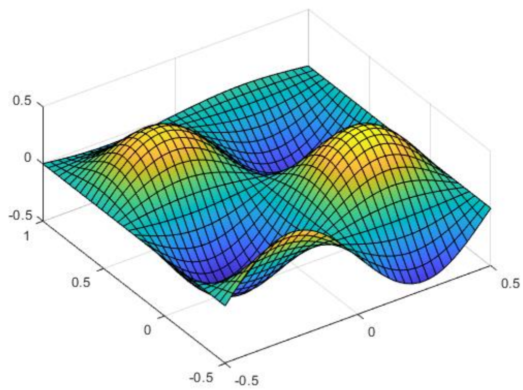
Definiáljuk a célfüggvényt és a megkötést is Matlab-ban az ábrázoláshoz!

```
> clc; clear all; close all; format shortG;
> % Célfüggvény, kétváltozós felület definiálása
> f = @(x,y) sin(2*pi*x).*cos(5*y)./((2+x.^3).*(1+2*y.^5))
> % Megkötés definiálása egyvált. függvényként explicit módon
> e = @(x) 0.6*x+0.3
```

CÉLFÜGGVÉNY ÉS MEGKÖTÉS TÉRBELI ÁBRÁZOLÁSA[3]

Az első lépés természetesen most is, hogy magát a célfüggvényt ábrázoljuk, a megkötésekkel együtt. Ezt megtehetjük térbeli ábrán vagy szintvonalakkal. Többnyire elegendő a szintvonalas ábrázolás, de a jobb szemléltetés kedvéért most nézzük meg először a térbeli ábrázolást! Először ábrázoljuk a célfüggvény felületét térben az **fsurf** paranccsal (bizonyos esetekben a korábbi változat, az **ezsurf** lehet szükséges).

```
> figure(1);
> fsurf(f,[-0.5 0.5 -0.5 1])
```



Ezután készítsük el a megkötés térbeli ábráját is! Ez kétféleképpen történhet. Az egyik, hogy függőleges metsző felületként rajzoljuk be a megkötést (**fsurf** paranccsal). Másik megoldás, hogy magát a metszésvonalat rajzoljuk be térben. Ehhez felveszünk megfelelő sűrűségben pontokat a feltétel mentén, kiszámoljuk a hozzájuk tartozó magasságokat a felületen és berajzoljuk a térbeli görbét (**plot3**). Nézzük meg mindkét megoldást. Kezdjük a metszésvonallal!

```
> % egyenessel adott feltétel berajzolása térbeli görbével
> xi = linspace(-0.5,0.5,50)'; % 50 pont felvétele a tartományban
> yi = e(xi); % útvonal mentén az y koord. kiszámítása
> zi = f(xi,yi); % terepi magasságok az útvonal mentén
```

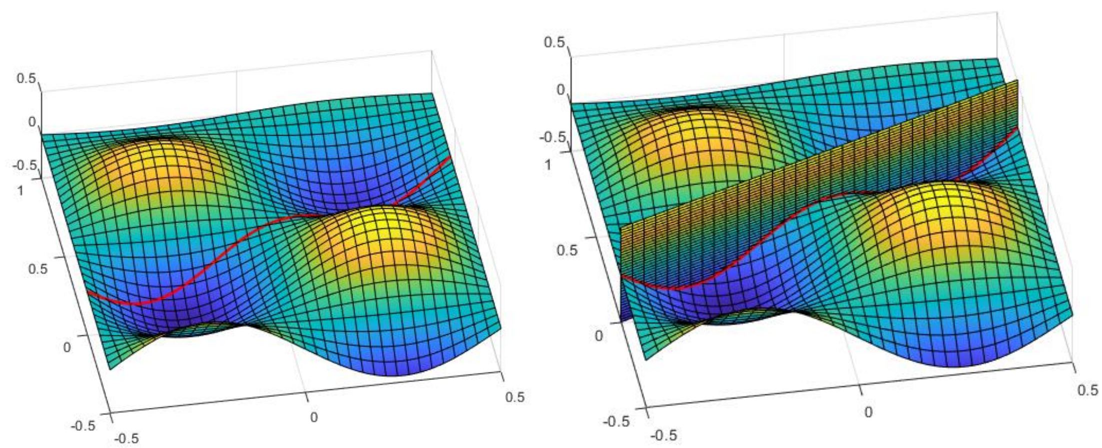
[3] Otthoni átnézésre

```
> hold on; plot3(xi,yi,zi,'r','LineWidth',2)
> view([-10 70]) % nézőpont beállítása (azimut, magassági szög)
```

Az eredményt lásd a lenti bal oldali ábrán!

A megkötést megadhatjuk függőleges felületként is paraméteresen és ábrázolhatjuk az **fsurf** paranccsal (`fsurf(funx,funy,funz,uvinterval)`). Térbeli eseten két paraméterünk lesz, az egyik legyen most a z koordináta maga, ezt adjuk meg a funz függvénynek, hiszen függőleges felületként minden z értéket fel kell, hogy vegyen. A másik paraméterünk legyen az x koordináta. Az y koordináta ennek lesz a függvénye: $y = 0.6 \cdot x + 0.3$. Meg kell adni a két paraméter, jelen esetben x és z értelmezési tartományát is. Azt tudjuk, hogy $-0.5 \leq x \leq 0.5$, de a szintvonalas ábra alapján a z értékeke is a [-0.5,0.5] tartományon belül marad.

```
> % megkötés függőleges felülettel paraméteresen
> xp = @(x,z) x
> yp = @(x,z) 0.6*x+0.3
> zp = @(x,z) z
> fsurf(xp,yp,zp,[-0.5,0.5,-0.5,0.5])
```



Az ábrák alapján egyértelmű, hogy két lokális minimumhelyünk van a görbe mentén. Ha mindkettő értékét kiszámoljuk, akkor el tudjuk dönteni, hogy a kettő közül melyik a kisebb, melyik a globális minimum a tartományon az adott feltétel mentén.

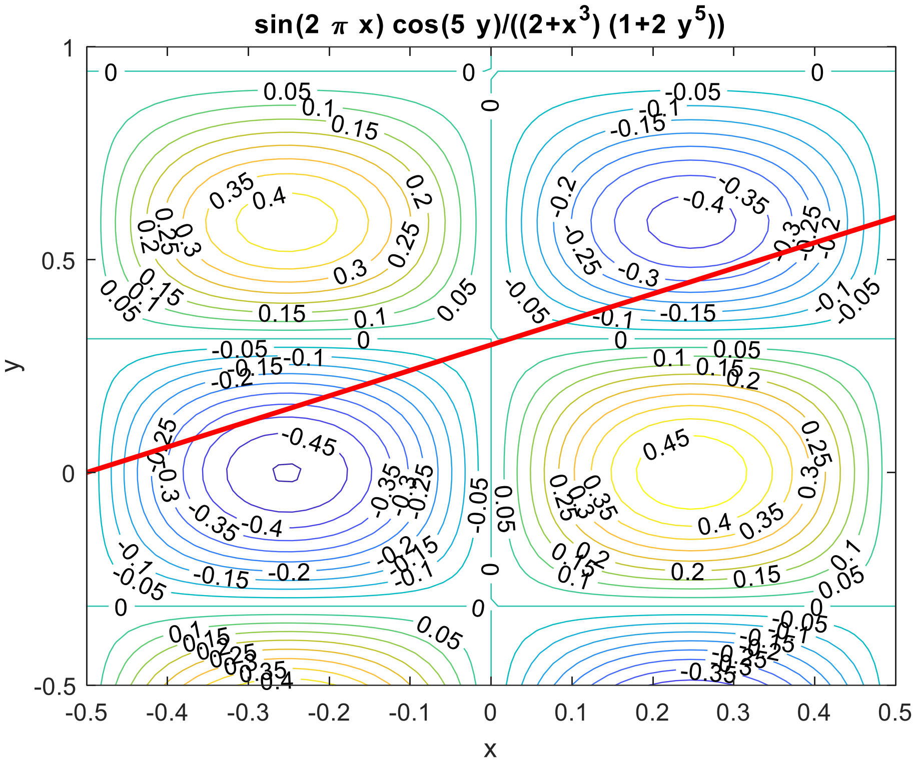
SZINTVONALAS ÁBRÁZOLÁS

A megoldás megtalálásához többnyire elegendő a szintvonalas ábrázolás is, nincs szükség térbeli ábrára. A szintvonalas ábrából könnyebb kezdőértéket választani, illetve a szintvonalak értékének feliratozása esetén azt is könnyebb eldönteni, hogy több lokális minimumhely esetén melyik lesz a legkisebb, a globális minimum helye.

Először rajzoljuk fel a célfüggvényt szintvonalas ábraként, használjunk 0.05-ös szintvonalközt és feliratozzuk a szintvonalakat! Ezt követően rajzoljuk be a szintvonalas ábrába a megkötést is, eltérő vonaltípussal. Az explicit formában adott egyenest a megszokott **fplot** paranccsal rajzolhatjuk fel. Ha implicit formában lenne adott a megkötés, akkor az **fimplicit** parancsot használhatnánk.

```
> figure(2); h1 = ezcontour(f,[-0.5 0.5 -0.5 1])
> set(h1,'ShowText','on','LevelStep', 0.05)
> % megkötés berajzolása szintvonalas ábrába
> hold on; fplot(e,[-0.5,0.5],'r','LineWidth',2)
```

A szintvonalas ábra alapján is látszódik, hogy a kisebbik x koordinátájú lokális minimumhely lesz most a globális minimum is, hiszen itt a feltétel a -0.4-es szintvonal mellett halad el, a másik esetben pedig -0.35-ös szintvonal mellett.



MEGOLDÁS BÜNTETŐFÜGGVÉNY MÓDSZERREL

Ez a módszer is csak egyenlőséggel adott feltételekkel történő optimalizációra alkalmas, mint a Lagrange-módszer. A feladat a következő formában adott:

Célfüggvény: $z = f(x, y)$

Egyenlőséggel adott megkötések: $g_i(x, y) = 0$

A módszer lényege, hogy a feltételeket beépíti a célfüggvénybe, így kapunk egy feltételnélküli optimalizálási feladatot. A megkötéssel definiált minimalizáció helyett, most tekintsük a következő megkötés nélküli feladatot, amit minimalizálunk:

$$F(x, K) = f(x) + K \cdot g(x)^T \cdot g(x)$$

ahol $K > 0$, egy új paraméter, aminek a segítségével 'megbüntetjük' *g(x)* feltétel nullától való eltérését, méghozzá úgy, hogy ezt az eltérést négyzetesen vesszük figyelembe. A *K* paraméter értékének növelésével az új, megkötés nélküli probléma megoldása tart az eredeti egyenlőségi megkötéssel rendelkező probléma megoldásához**.**

Minél nagyobb a K, annál nagyobb a "büntetés" a g(x) értékének a nullától történő eltérése miatt. Ezt a kvadratikus büntetés-függvényt Courant büntetés függvénynek nevezik. A mostani feladatot felírhatjuk ebben az alakban:

$$F(x, y, K) = f(x, y) + K \cdot g(x, y)^2$$

Megj.: amennyiben nem egy, hanem két megkötésünk lenne, a feladatot a következő alakban írhatnánk fel: $F(x,y,K) = f(x,y) + K \cdot (g_1(x,y)^2 + g_2(x,y)^2)$

Határozzuk meg a feltételes minimum helyét a büntetőfüggvény módszerével! Próbáljunk ki több K paramétert is és nézzük meg hogyan változik a megoldás! A büntetés-függvény skalár paramétere legyen rendre 10, 100, 1000 és 10000. Miután visszavezettük a feladatot egy feltétel nélküli optimalizációs feladatra, ezt a feladatot már megoldhatjuk akár a kvázi Newton-módszert használó **fminunc**, vagy a szimplex módszert használó **fminsearch** függvényével is.

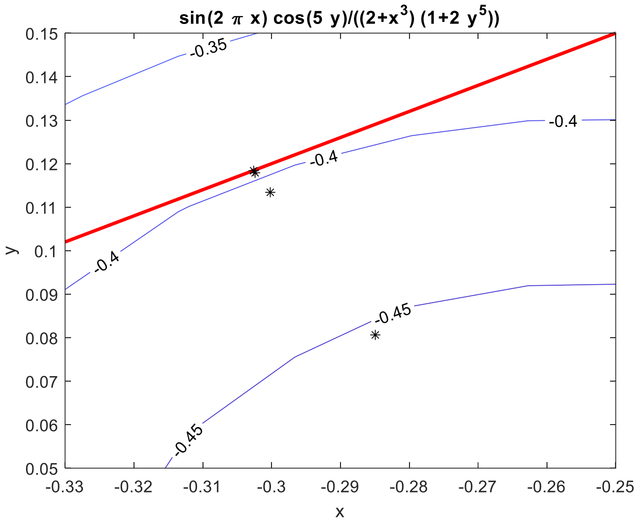
Mivel jelen esetben a feltétel nem nullára rendezett alakban áll rendelkezésre, hanem egy egyenes egyenleteként, először rendezzük ezt nullára:

$$g(x,y) = 0.6 \cdot x + 0.3 - y = 0$$

A másik fontos szempont, amire oda kell figyelnünk a Matlab-ban, hogy vektorváltozóssá kell alakítanunk mind az *f* célfüggvényt, mind a *g* megkötést!

```
> % Büntetőfüggvény módszere, megkötés 0-ra rendezése
> g = @(x,y) 0.6*x+0.3-y
> % Célfüggvényt, és megkötést is vektorváltozós alakban kell megadni!
> F = @(v) f(v(1),v(2)); % vektorváltozóssá alakítás
> G = @(v) g(v(1),v(2)); % vektorváltozóssá alakítás
> % a minimalizálandó büntetés-függvény, különböző K paraméterekkel
> P10 = @(v)  F(v) + 10* G(v).^2;
> P100 = @(v)  F(v) + 100* G(v).^2;
> P1000 = @(v)  F(v) + 1000* G(v).^2;
> P10000 = @(v)  F(v) + 10000* G(v).^2;
> % az első lokális minimum különböző K értékek esetén
> x01 = [-0.25;0.1] % kezdőérték x-re, y-ra
> [sol1 min1] = fminsearch(P10, x01)
> [sol2 min2] = fminsearch(P100, x01)
> [sol3 min3] = fminsearch(P1000, x01)
> [sol4 min4] = fminsearch(P10000, x01)
> sol = [sol1,sol2,sol3,sol4]
> %      -0.28493     -0.30019     -0.30239     -0.30239
> %      0.080633      0.11341       0.1179       0.1179
> fsol = [min1,min2,min3,min4]
> %      -0.43069     -0.40224     -0.39835     -0.39835
> % ellenőrizzük le a feltétel értékét is az egyes pontokban
> felt = g(sol(1,:),sol(2,:))
> %  0.048408     0.0064735    0.00066395   0.00066395
> % Ábrázoljuk az egyes megoldásokat
> figure(2); plot(sol(1,:),sol(2,:),'k*')
> axis([-0.33 -0.25 0.05 0.15])
```

Látszik, hogy a K paraméter növelésével a megoldás egyre közelebb kerül a $g(x,y) = 0$ feltételhez, ahhoz a ponthoz, ahol a feltétel érinti a célfüggvény szintvonalait, amit a Lagrange-módszerrel is kerestünk.



BEÉPÍTETT MATLAB FÜGGVÉNY (FMINCON) - LINEÁRIS EGYENLŐSÉGGEL ADOTT MEGKÖTÉS

Természetesen a Matlab-nak is van saját beépített függvénye a feltételes szélsőérték kereséshez, ez az **fmincon** függvény (find minimum of constrained function). Most a következő formában fogjuk meghívni:

```
[x,fval] =  fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
```

Itt *'fun'* a vektorváltozós célfüggvény, *'x0'* a kezdőértékek vektora. Az A,b változók megadásával lineáris egyenlőtlenséggel, *Aeq, beq* változók megadásával pedig lineáris egyenlőséggel adott megkötést tehetünk, a következő formában:

$$Aeq \cdot x = beq$$

Az értelmezési tartományt az alsó és felső határok (lb – lower boundary, ub – upper boundary) megadásával lehet beállítani. Oldjuk meg az előző feladatot a beépített függvénnyel is!

A célfüggvény:

$$f(x,y) = \frac{\sin(2\cdot\pi\cdot x)\cdot\cos(5\cdot y)}{(2+x^3)\cdot(1+2\cdot y^5)}$$

Az egyenes egyenletével adott megkötés: $y = 0.6\cdot x + 0.3$

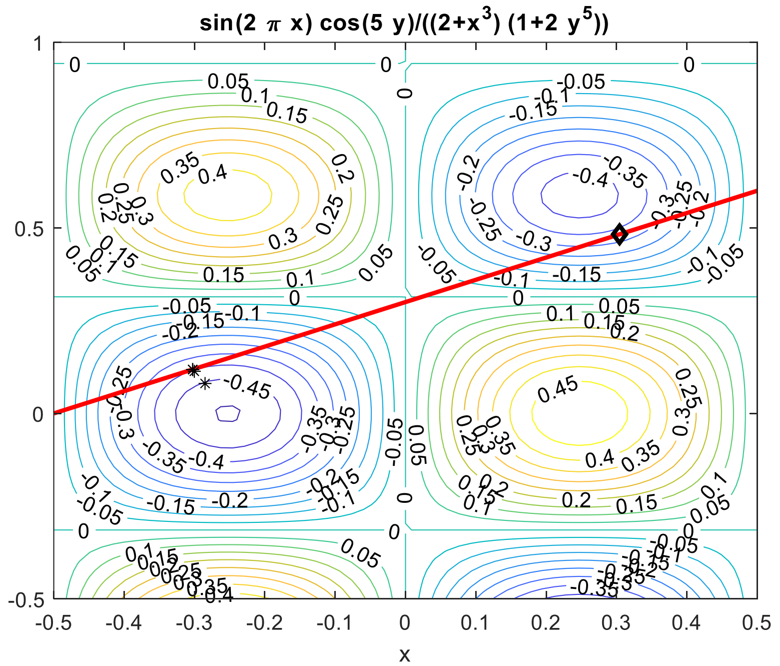
A megadott alsó/felső határok: $x \in [-0.5, 0.5]; y \in [-0.5, 1]$

A lineáris egyenlőséggel adott megkötésünk az $Aeq \cdot x = beq$ alakban kell megadnunk, vagyis az ismeretlenek (most épp x,y) együtthatóit az Aeq mátrixban, a konstanst pedig a beq változóban (vagy vektorban, ha több feltételünk is van). Alakítsuk át a megkötést erre a formára:

$$0.6\cdot x - y = -0.3 \qquad Aeq = [0.6 \quad -1];\ beq = -0.3$$

Mivel nincs egyenlőtlenséggel adott lineáris megkötésünk, ezért $A; b$ üres mátrix lesz. Alsó/felső határ van megadva (lb,ub). Kezdeti értékre természetesen itt is szükség van. Keressük most meg a feltétel mentén a másik lokális minimumot a beépített függvénnyel!

```
> % Megoldás fmincon függvénnyel
> axis([-0.5 0.5 -0.5 1]) % zoom az egész tartományra
> x02 = [0.3;0.4] % kezdőérték x-re, y-ra
> A=[]; b=[]; % Nincs lineáris egyenlőtlenséggel adott megkötés
> Aeq=[0.6 -1]; beq=[-0.3]; % Van: Megkötés lineáris egyenlettel
> % Alsó korlát x,y-ra és felső korlát x,y-ra
> lb=[-0.5,-0.5]; ub=[0.5,1];
> % Megoldás – F vektorváltozós függvény!
> [sol2 fsol2] = fmincon(F,x02,A,b,Aeq,beq,lb,ub)
> % sol2 =   [0.30431; 0.48259];   fsol2 =  -0.3294
> plot(sol2(1),sol2(2),'kd','LineWidth',2)
```



EGYENLŐTLENSÉGET IS TARTALMAZÓ MEGKÖTÉSES OPTIMALIZÁLÁS (FMINCON ÁLTALÁNOSAN)

Az **fmincon** egy általánosan használható szélsőérték kereső algoritmus, nem csak egyenletekkel, hanem egyenlőtlenségekkel is lehet feltételeket megadni, akár lineáris, akár nemlineáris formában, felső/alsó korláttal és egyéb opciókkal együtt. A következő formában lehet meghívni (lehet több/kevesebb be és kimenete is):

```
[x,fval] =  fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

Itt a paraméterek:

- *fun*: minimalizálandó függvény
- *x0*: kezdőérték
- *A,b*: $A \cdot x < b$ lineáris egyenlőtlenségekkel adott megkötések
- *Aeq,beq*: $Aeq \cdot x = beq$ lineáris egyenletrendszerrel adott megkötések
- *lb,ub*: $lb < x < ub$, alsó és felső határ (lower/upper bound)
- *nonlcon*: nemlineáris megkötések: $c(x) \leq 0$ és $ceq(x) = 0$ (nonlinear constraint)
- *options*: opciók

A feltételeket a megadott sorend szerint kell megadni, amilyen feltételünk nincs, annak a helyére üres mátrixot kell tenni. A gyakorló feladatok között található egy komplex építőmérnöki probléma, de most egy egyszerűbb feladaton nézzük meg a megoldást!

Oldjuk meg az előző feladatot más feltételekkel! Adott a következő felület a $-0.5 \le x \le 0.5$; $-0.5 \le y \le 1$ tartományon.

$$f(x,y) = \frac{\sin(2\cdot\pi\cdot x)\cdot\cos(5\cdot y)}{(2+x^3)\cdot(1+2\cdot y^5)}$$

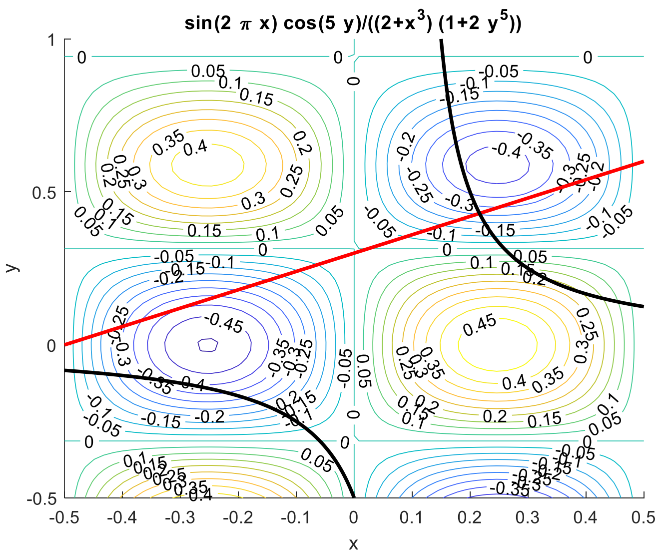
Határozza meg a minimumot a megadott feltételek mellett!

$$20\cdot x\cdot y - 2\cdot y = 1$$

$$y > 0.6\cdot x + 0.3$$

Az első megkötés egy implicit formában megadott nemlineáris egyenlőség, a második megkötés pedig egy lineáris egyenlőtlenség, ami azt mondja meg, hogy az előbb megadott egyenes felett elhelyezkedő területen keressük a minimumot, a nemlineáris egyenlettel megadott görbén. Rajzoljuk fel egy új ábrába a szintvonalas felületet és a két megkötést is! Az eredeti szintvonalas ábrát és az egyenest átvehetjük az előző példából. A nemlineáris feltétel nem explicit formában van megadva, ezért a megjelenítéshez használjuk most az **fplot** parancs helyett az **fimplicit** parancsot, miután 0-ra rendeztük a megkötést!

```
> clc; clear all; close all; format shortG;
> f = @(x,y) sin(2*pi*x).*cos(5*y)./((2+x.^3).*(1+2*y.^5))
> figure(3); hold on;
> h1 = ezcontour(f,[-0.5 0.5 -0.5 1])
> set(h1,'ShowText','on','LevelStep', 0.05)
> e = @(x) 0.6*x+0.3;
> fplot(e,[-0.5,0.5],'r','LineWidth',2)
> g1 = @(x,y) 20*x.*y-2*y-1
> fimplicit(g1,'k','LineWidth',2)
```



Nézzük meg először a lineáris egyenlőtlenséggel adott megkötést!

Ez azt jelenti, hogy a berajzolt egyenes ($y = 0.6\cdot x + 0.3$) feletti területen keressük a minimumhelyet. A feltételt a következő alakba kell hozni:

$$A \cdot x < b$$

Ahol a relációs jel kisebb oldalán vannak az ismeretlenek, a nagyobb oldalán pedig a konstans (vagy több ilyen feltétel esetén vektorban a konstansok). Alakítsuk át az $y > 0.6 \cdot x + 0.3$ egyenlőtlenséget ebbe a formába és írjuk fel az A,b együtthatókat!

$$0.6 \cdot x - y < -0.3 \qquad A = [0.6 \quad -1];\ b = -0.3$$

<u>Megjegyzés</u>: amennyiben az egyenes alatt keresnénk a minimumot, akkor az $y < 0.6 \cdot x + 0.3$ egyenlőtlenséget kellene ilyen alakba hozni:

$$-0.6 \cdot x + y < 0.3 \qquad A = [-0.6 \quad 1];\ b = 0.3$$

Lineáris egyenlettel adott megkötésünk nincs, így *Aeq* és *beq* helyére üres mátrixot kell írni. Alsó/felső korlát maradt ugyanaz, mint az előbb.

```
> A=[0.6 -1]; b=-0.3; % lineáris egyenlőtlenség
> Aeq=[]; beq=[]; % lineáris egyenlet nincs
> lb=[-0.5,-0.5]; ub=[0.5,1]; % Alsó/felős korlát x,y-ra
```

Nézzük most a nemlineáris megkötéseket! Ezekből is lehetne egyenlettel vagy egyenlőtlenséggel adott megkötés, ezt a *nonlcon* paraméterben adhatjuk meg.

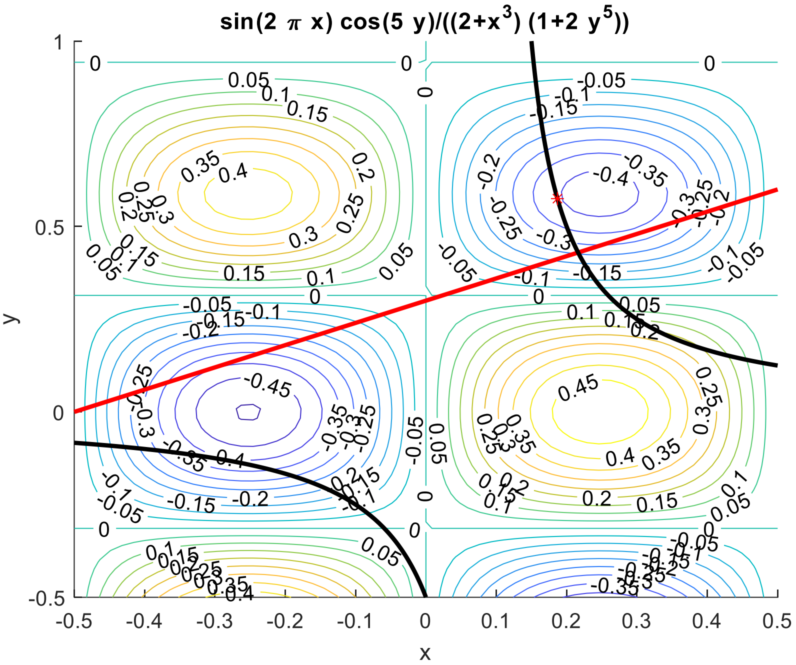
Egyenlőtlenség: $c(x) \leq 0$ Egyenlet: $ceq(x) = 0$

Itt kétféle feltételt kell egy paraméterben megadni, egy olyan függvényt kell előállítani, aminek két kimenete van, ezt a **deal** paranccsal állíthatjuk elő. A feltételeket $c(x) \leq 0$ és $ceq(x) = 0$ alakban kell megadni, 0-ra rendezve. Amennyiben csak az egyik féle feltételünk van, akkor a másik üres mátrix lesz. Itt is vektorváltozós függvényt kell definiálni.

```
> % nemlineáris megkötések vannak, de csak egyenlettel megadva
> c = []; % egyenlőtlenség nincs
> ceq = @(x,y) 20*x.*y-2*y-1 % 0-ra rendezett egyenlet
> nonlcon = @(v) deal(c, ceq(v(1),v(2))) % feltétel vektorváltozósan
```

A megoldáshoz a célfüggvénynek is vektorváltozósnak kell lennie. Meg kell adnunk egy kezdőértéket is. A szintvonalas ábráról vehetjük azt a pontot, ahol a nemlineáris feltétel egyenes feletti része megközelíti a legkisebb szintvonalat. Opcionális paramétereket is megadhatunk, pl. hogy írja ki az iterációs lépéseket, vagy hogy legyen a megadott tolerancia a függvényértékre 10^{-9}.

```
> F = @(v) f(v(1),v(2)) % vektorváltozós célfüggvény kell
> x0 = [0.2;0.5] % kezdőérték az ábrából
> opc = optimset('Display','iter','TolFun',1e-9);
> sol  = fmincon(F,x0,A,b,Aeq,beq,lb,ub,nonlcon,opc)
> % 0.18693,  0.57515
> plot(sol(1),sol(2),'r*')
```



Mi történne abban az esetben, ha egy vízszintes vagy függőleges egyenes mentén keresnénk a minimumot? Pl. x=-0.2 egyenesek mentén, vagy y = 0.5?

$x = -0.2$	$A \cdot x = b$ alakban:	$1 \cdot x + 0 \cdot y = -0.2$	$Aeq = [1 \quad 0]; beq = 0.2$
$y = 0.5$	$A \cdot x = b$ alakban:	$0 \cdot x + 1 \cdot y = 0.5$	$Aeq = [0 \quad 1]; beq = 0.5$

Abban az esetben, ha pl. x<-0.2 feltételünk lenne egyenlőtlenséggel megadva, akkor is lehet a fentihez hasonlóan megadni *A,b* értékét, vagy a legegyszerűbb az x-re vonatkozó felső határt -0.2-re állítani.

LINEÁRIS PROGRAMOZÁSI FELADAT[4]

A lineáris programozás a mérnöki alkalmazásoknál egy nagyon hasznos optimalizálási technika. A 'lineáris' szó arra utal, hogy mind célfüggvény, mind a megkötések csak lineáris függvényei a nemnegatív változóknak. Ahhoz, hogy alkalmazni lehessen három feltételnek kell teljesülnie:

1) A célfüggvény lineáris függvénye a változóknak
2) A megkötések is lineáris függvényei a változóknak
3) A változók csak nemnegatív értéket vehetnek fel

Többféle technika is van ennek a megoldására, pl. szimplex módszer, belső pont módszer stb. Most nem megyünk bele a részletekbe, a Matlab beépített **linprog** algoritmusát fogjuk használni:

```
[x fval] = linprog(f,A,b,Aeq,beq,lb,ub,options)
```

Látjuk, hogy nagyon hasonlóak a paraméterek, mint az **fmincon** függvényél. Itt a paraméterek:

[4] Otthoni átnézésre

– *fun*: minimalizálandó függvény vektorként megadva
– *A,b*: $A \cdot x < b$ lineáris egyenlőtlenségekkel adott megkötések
– *Aeq,beq*: $Aeq \cdot x = beq$ lineáris egyenletrendszerrel adott megkötések
– *lb,ub*: $lb < x < ub$, alsó és felső határ (lower/upper bound)
– *options*: opciók

Nincs benne kezdőérték megadás, mivel ebben az esetben nincs rá szükség, ami nagy előny sok esetben. Illetve nem lehet nemlineáris feltételt sem megadni. A többi feltétel megadása ugyanúgy működik, mint az **fmincon** esetében, kivéve a minimalizálandó függvényt. Nézzünk egy egyszerű példát rá!

Minimalizálandó célfüggvény: $z = -2 \cdot x + 8 \cdot y$

Feltételek: $3 \cdot x + 4 \cdot y \leq 80$

$$-3 \cdot x + 4 \cdot y \geq 8$$

$$x + 4 \cdot y \geq 40$$

$$x, y \geq 0$$

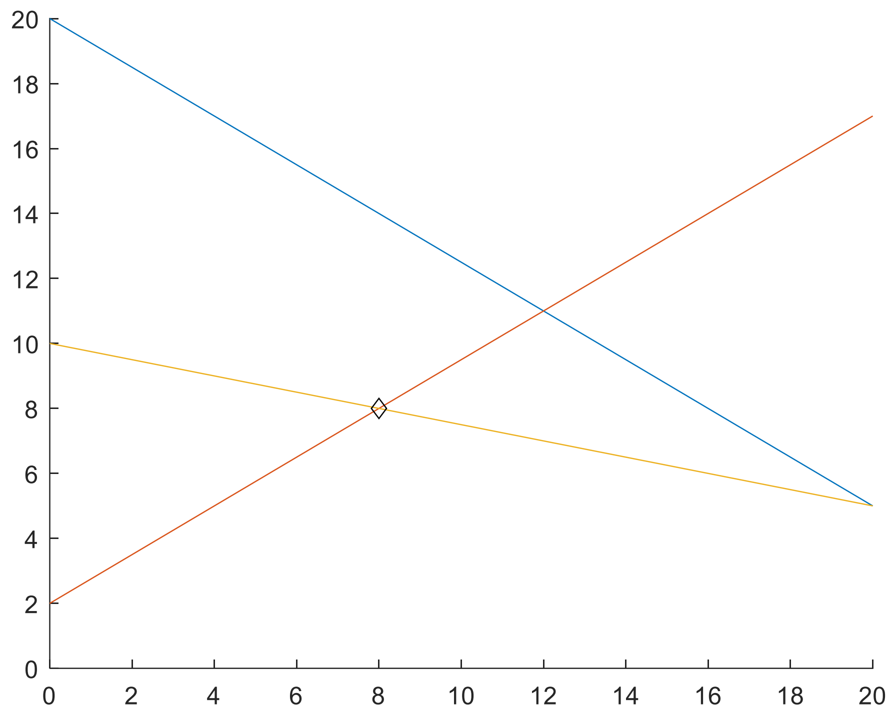
A célfüggvény megadása vektorként az együtthatók megadását jelentik: [-2 8]

Az egyenlőtlenségeket $A \cdot x < b$ alakba kell hozni, ahol a relációs jel kisebbik oldalán vannak a változók, nagyobbikon pedig a konstansok.

$$\begin{array}{lcl} 3 \cdot x + 4 \cdot y \leq 80 & & 3 \cdot x + 4 \cdot y \leq 80 \\ -3 \cdot x + 4 \cdot y \geq 8 & \rightarrow & 3 \cdot x - 4 \cdot y \leq -8 \\ x + 4 \cdot y \geq 40 & & -x - 4 \cdot y \leq 40 \end{array}$$

Oldjuk meg a feladatot Matlab-ban!

```
> clear all; clc; close all;
> % a célfüggvény együttható vektora
> c=[-2 8];
> % Megkötések ábrázolása
> g1 = @(x,y) 3*x+4*y-80
> g2 = @(x,y) -3*x+4*y-8
> g3 = @(x,y) x+4*y-40
> figure(1); hold on; fimplicit(g1,[0,20]); fimplicit(g2,[0,20]);
  fimplicit(g3,[0,20]);
> % Lineáris megkötés egyenlőtlenséggel, A*x <= b alakban
> A = [3 4; 3 -4; -1 -4];
> b = [80; -8; -40];
> % Lineáris megkötés egyenlőséggel : nincs
> Aeq = []; beq = [];
> % A változókra alsó korlát : van
> lb = [0 0];
> % A változókra felső korlát : nincs
> ub = [];
> sol = linprog(c,A,b,Aeq,beq,lb,ub,optimset('Display','iter'))
> % sol = 8 8
> plot(sol(1),sol(2),'kd')
```



A gyakorló feladatok között található egy ennél komplexebb építőmérnöki példa a lineáris programozásra.

GYAKORLÓ FELADATOK

GYAKORLÓ FELADAT 1. – LAGRANGE-MÓDSZER

Oldjuk meg a büntetőfüggvény módszer alfejezetben lévő példát most Lagrange módszerrel is! Adott a következő felület a $-0.5 \leq x \leq 0.5$; $-0.5 \leq y \leq 1$ tartományon.

$$f(x,y) = \frac{\sin(2\cdot\pi\cdot x)\cdot\cos(5\cdot y)}{(2+x^3)\cdot(1+2\cdot y^5)}$$

Határozza meg a minimumot a Lagrange-módszerrel a következő megkötés mellett!

$$y = 0.6\cdot x + 0.3$$

A lineáris egyenlettel megadott megkötés esetén a következő Lagrange függvényt tudjuk felírni (a megkötést nullára rendezve):

$$L(x,\mathrm{y},\lambda) = \frac{\sin(2\cdot\pi\cdot x)\cdot\cos(5\cdot y)}{(2+x^3)\cdot(1+2\cdot y^5)} - \lambda\cdot(0.6\cdot x + 0.3 - y)$$

A minimum szükséges feltétele a parciális deriváltak eltűnése, azaz a következő egyenletrendszer megoldása:

$$\frac{dL(x,\mathrm{y},\lambda)}{dx} = 0 \qquad \frac{dL(x,\mathrm{y},\lambda)}{dy} = 0 \qquad \frac{dL(x,\mathrm{y},\lambda)}{d\lambda} = 0$$

A deriváltakat kiszámítva a következő egyenletrendszert kapjuk:

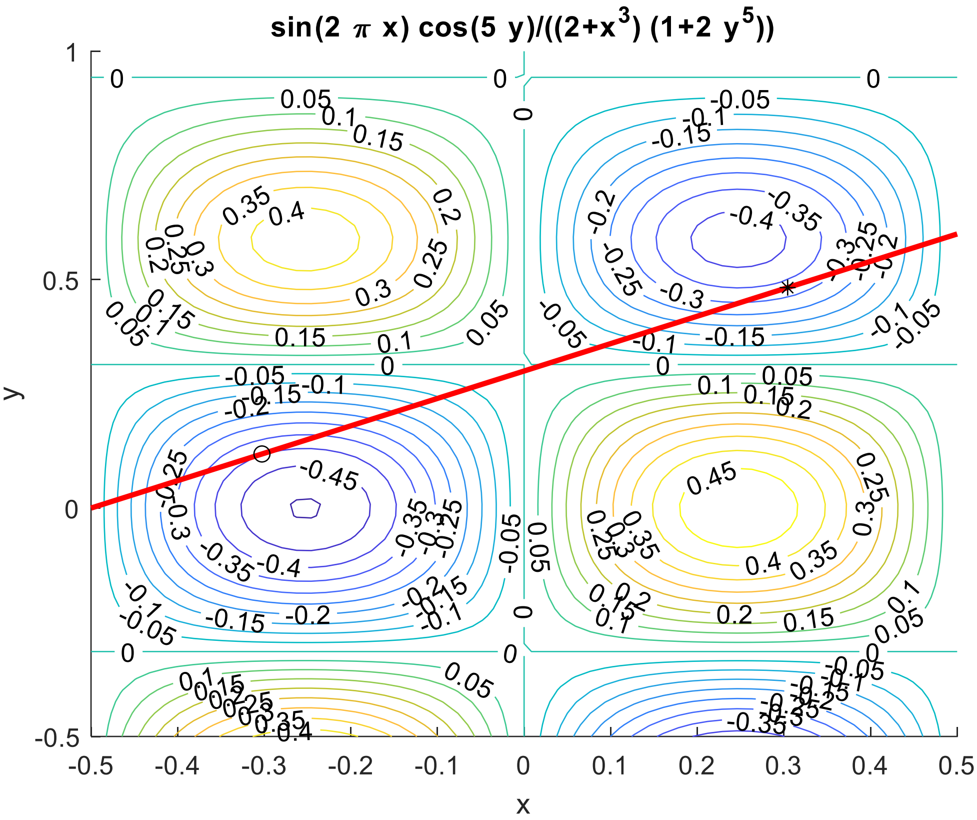
$$\frac{\cos(5\cdot y)}{1+2\cdot y^5}\cdot\left(\frac{\cos(2\cdot\pi\cdot x)\cdot 2\cdot\pi}{(2+x^3)}-\frac{\sin(2\cdot\pi\cdot x)\cdot 3\cdot x^2}{(2+x^3)^2}\right)-0.6\cdot\lambda\ =0$$

$$\frac{\sin(2\cdot\pi\cdot x)}{(2+x^3)}\cdot\left(\frac{-\sin(5\cdot y)\cdot 5}{(1+2\cdot y^5)}-\frac{\cos(5\cdot y)\cdot 10\cdot y^4}{(1+2\cdot y^5)^2}\right)+\lambda=0$$

$$0.6\cdot x+0.3-y=0$$

Ebben az esetben kicsit bonyolultabb volt a parciális deriváltak meghatározása papíron, de természetesen a Matlabot is segítségül hívhatjuk ehhez, szimbolikus számításokat alkalmazva.

```
> %% Megoldás Lagrange-módszerrel - lineáris megkötés
> clc; clear all; close all; format shortG;
> f = @(x,y) sin(2*pi*x).*cos(5*y)./((2+x.^3).*(1+2*y.^5))
> figure(3); hold on;
> h1 = ezcontour(f,[-0.5 0.5 -0.5 1])
> set(h1,'ShowText','on','LevelStep', 0.05)
> e = @(x) 0.6*x+0.3;
> fplot(e,[-0.5,0.5],'r','LineWidth',2)
>
> g = @(x,y) 0.6*x+0.3-y % 0-ra rendezve!
> L = @(x,y,lambda) f(x,y)-lambda*g(x,y);
>
> syms x y lambda
> dx=diff(L(x,y,lambda),x)
> % (2*pi*cos(5*y)*cos(2*pi*x))/((x^3 + 2)*(2*y^5 + 1)) - (3*lambda)/5
  - (3*x^2*cos(5*y)*sin(2*pi*x))/((x^3 + 2)^2*(2*y^5 + 1))
> dy=diff(L(x,y,lambda),y)
> % lambda - (5*sin(5*y)*sin(2*pi*x))/((x^3 + 2)*(2*y^5 + 1)) -
  (10*y^4*cos(5*y)*sin(2*pi*x))/((x^3 + 2)*(2*y^5 + 1)^2)
> dl=diff(L(x,y,lambda),lambda)
> % y - (3*x)/5 - 3/10
>
> % A nemlineáris egyenletrendszermegoldása fsolve-val
> FLsym = [dx;dy;dl]
> FL = matlabFunction(FLsym) % @(lambda,x,y)...
> % A nemlineáris egyenletrendszer vektorizálása
> FL = @(v) FL(v(1),v(2),v(3)) % v = (lambda,x,y)
>
> % a megoldás
> x01 = [1; -0.3;0.1];  % az 1. kezdőérték lambda,x,y sorrendben
> x02 = [1; 0.3;0.5];  % a 2. kezdőérték lambda,x,y sorrendben
>
> xyl1 = fsolve(FL,x01,optimset('Display','iter'))
> % xyl1 = [-1.3387,-0.30265,0.11841]
> xyl2 = fsolve(FL,x02,optimset('Display','iter'))
> % xyl2 = [1.3002, 0.30431,0.48259]
> plot(xyl1(2),xyl1(3),'ko'); plot(xyl2(2),xyl2(3),'k*')
> zopt1 = f(xyl1(2),xyl1(3)) %  -0.3979
> zopt2 = f(xyl2(2),xyl2(3)) %  -0.3294
```



GYAKORLÓ FELADAT 2. – MINIMÁLIS FELÜLETŰ KÚP

Szeretnénk meghatározni egy minimális felületű, egységnyi térfogatú kúp adatait (sugár, magasság).

A kúp felszíne: $A = r^2 \cdot \pi + \pi \cdot r \cdot a$ ahol a az alkotó hossza: $a = \sqrt{r^2 + h^2}$

A kúp térfogata: $V = \dfrac{r^2 \cdot \pi \cdot h}{3} = 1$

a) Írja fel a kúp felületének sugártól és magasságtól függő Matlab függvényét! Írja fel az egységnyi térfogatnak megfelelő megkötés Matlab függvényét is!
b) Oldja meg a feltételes szélsőérték feladatot a sugárra és magasságra többféle módszerrel is. Mindegyik esetben ellenőrizze a megkötés teljesülését és határozza meg, hogy mekkora lesz a magasság és a sugár aránya, illetve mekkora lesz a kapott felszín?
    i. Matlab beépített függvénnyel
    ii. Lagrange módszerével
    iii. büntető függvény módszerrel (K=1000)

Megoldás:

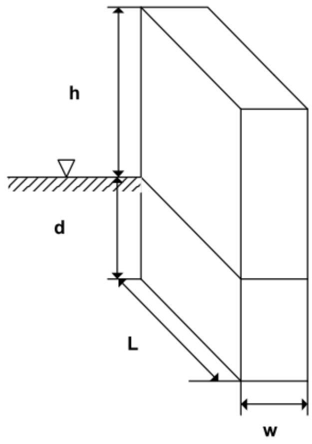
```
> %% 1A - cone surface
> clc; clear all; close all;
>
> % A=R^2*pi+pi*R*a, a = sqrt(r^2+h^2)
```

```
> % V=pi*r^2*h/3=1;
> % a)
> A = @(r,h) r.^2*pi+pi*r.*sqrt(r.^2+h.^2)
> V = @(r,h) pi*r.^2.*h/3-1
> A = @(v) A(v(1),v(2)); V = @(u) V(u(1),u(2));
>
> % b-i) Matlab built-in function
> nonlcon = @(u) deal([],V(u))
> v0 = [0.5, 0.5]
> x = fmincon(A,v0,[],[],[],[],[0,0],[],nonlcon)
> r = x(1) % 0.69632
> h = x(2) % 1.9695
> ratio = h/r % 2.8284
> S = A([r,h]) % 6.0929
> V([r,h]) %  -4.0023e-10
>
> % Lagrange method
> syms r h lam
> Lfv = A([r h])+lam*V([r h])
> % lam*((h*pi*r^2)/3 - 1) + pi*r^2 + pi*r*(h^2 + r^2)^(1/2)
> F = gradient(Lfv,[r,h,lam])
> %   pi*(h^2 + r^2)^(1/2) + 2*pi*r + (pi*r^2)/(h^2 + r^2)^(1/2) +
  (2*pi*h*lam*r)/3
> %  (pi*lam*r^2)/3 + (pi*h*r)/(h^2 + r^2)^(1/2)
> %  (h*pi*r^2)/3 - 1
> sol=solve(F)
> %       h: [3x1 sym]
> %     lam: [3x1 sym]
> %       r: [3x1 sym]
> double([sol.r sol.h sol.lam])
> % 0.69632 + 0i 1.9695 + 0i -4.062 + 0i
> % 0.34816 - 0.60303i   -0.98475 + 1.7056i   2.031 + 3.5178i
> % 0.34816 + 0.60303i   -0.98475 - 1.7056i   2.031 - 3.5178i
> r = double(sol.r(1)) % 0.69632
> h = double(sol.h(1)) % 1.9695
> ratio = h/r % 2.8284
> S = A([r,h]) % 6.0929
> V([r,h]) % 0
>
> %b-iii) penalty function method
> Bfv=@(u) A(u)+1000*V(u).^2
> x =fminsearch(Bfv,v0)
> % x =    0.69585       1.9681
> ratio = x(2)/x(1) % 2.8284
> S = A(x) % 6.0847
> V(x) % -0.0020296
```

GYAKORLÓ FELADAT 3. – KOMPLEX ÉPÍTŐMÉRNÖKI PROBLÉMA[5]

Oldjunk meg a Matlab beépített függvényével egy bonyolultabb kétváltozós szélsőérték keresési feladatot, ahol van megadott alsó/felső korlát, lineáris és nemlineáris egyenlőtlenséggel adott megkötés is.

Az energiaköltségek megtakarítása érdekében egy részben földbe süllyesztett épületet kell tervezni. A 25 szintes épület teljes padlóterülete legalább 20 000 m^2 kell, hogy legyen. Az épület w szélességének és L hosszának az előírt aránya w/L = 1/1.618, és L legfeljebb 50 m lehet. Az egyes emeletek magassága 3.5 m. Az épület energiaköltsége a föld feletti részének felületére vonatkoztatva 100 \$/év/m^2. Az évi teljes energiaköltség nem haladhatja meg a 225 000 \$-t. Határozzuk meg az épület méreteit úgy, hogy a földmunkák költsége (amely arányos az épület föld alatti részének térfogatával) minimális legyen.



A minimalizálandó célfüggvény értéke

$$f(d,w) = \text{állandó} \cdot 1.618 \cdot d \cdot w^2 . \qquad \text{(állandó = 1/10000)}$$

Az egyenlőtlenségi megkötések:

$$g_1(d,w) = 20000 - 25 \cdot 1.618 w^2 \le 0 \qquad \text{(teljes padlóterület 25 szintre)}$$

$$g_2(d,w) = 1.618 w - 50 \le 0 \qquad \text{(épület hossza)}$$

$$g_3(d,w) = 45815 w - 523.6 wd + 161.8 w^2 - 225000 \le 0 \quad \text{(évi teljes energiaköltség)}$$

A változókra vonatkozó megkötések:

$$d > 0$$

$$w > 0$$

A megoldás lépései:

1. A MatLab többváltozós megkötéses optimalizálási eljárásának alkalmazásához adjuk meg a szükséges lineáris egyenlőtlenségi megkötéseket és a változókra vonatkozó korlátokat.
2. Írjunk függvényt a nemlineáris egyenlőtlenségi megkötésekre.
3. A d = 50, w = 10 kezdőértékből kiindulva határozzuk meg a feladat megoldását a MatLab beépített eljárásával.
4. Melyek lesznek az aktív egyenlőtlenségi megkötések?

```
> %% optimalizáció egyenlőtlenségi megkötésekkel
> clc; clear all; close all
>
> % A beépített fmincon függvény szükséges paraméterezése:
> %  X = fmincon(FUN,X0,A,B,Aeq,Beq,LB,UB,NONCON,OPTIONS)
```

[5] Paláncz Béla példatárából (2012)

```
> % min F(X)  subject to: A*X  <= B, Aeq*X  = Beq (linear constraints)
> %  X                    C(X) <= 0, Ceq(X) = 0 (nonlinear constraints)
> %                       LB <= X <= UB       (bounds)
>
> % a célfüggvény (vektor változós): 1.618 d w^2 / 10000
> f = @(x) 1.618*x(1)*x(2)^2/1e4;
>
> % Lineáris megkötés egyenlőtlenséggel van: -50 + 1.618 w <= 0
> A = [0 1.618]; b = [50];
>
> % Lineáris megkötés egyenlőséggel nincs
> Aeq = [ ];  beq = [ ];
>
> % A változókra alsó korlát van: d > 0 és w > 0
> lb = [0; 0];
> % A változókra felső korlát: nincs
> ub = [ ];
>
> % Nemlineáris egyenlőségi megkötés: nincs
> ceq = [ ];
> % Nemlineáris egyenlőtlenségek: C(X) <= 0
> g1 = @(d,w) 20000 - 25*1.618*w.^2
> g2 = @(d,w) 45815*w - 523.6*w.*d + 161.8*w.^2 - 225000
> % Vektorban a két megkötés, vektorváltozókkal
> c = @(v) [g1(v(1),v(2)); g2(v(1),v(2))]
> % Megjegyzés:
> %   az fmincon-nak két kimeneti értéket [c ceq] előállító függvény
  kell,
> %   ezért ezeket a beépített deal függvénnyel állítjuk elő
> nonlincon = @(v) deal(c(v), ceq)
> % nonlincon = @(v) deal([20000-25*1.618*v(2)^2;
> %              45815*v(2)-523.6*v(2)*v(1)+161.8*v(2)^2-225000], []);
>
> % Alternatív megoldás: külön függvény m-fájlban:
> % function [c ceq] = nonlincon(x)
> % c = [20000 - 25*1.618*x(2)^2;
> %      45815*x(2) - 523.6*x(2)*x(1) + 161.8*x(2)^2 - 225000];
> % ceq = [];
> % end
>
> % A kezdőérték
> x0 = [50; 10];
>
> % A megoldás a Matlab beépített függvényével:
> x = fmincon(f, x0, A, b, Aeq, beq, lb, ub, nonlincon, optimset
  ('Display', 'iter''TolFun',1e-9))
> % x = 75.0459; 22.2360
>
> % Megjegyzés: ha a nonlincon-t külső függvényként hívjuk, akkor
  @nonlincon kell paraméterként:
> % fmincon(f, x0, A, b, Aeq, beq, lb, ub, @nonlincon)
>
> % ellenőrizzük le a nemlineáris megkötések értékeit:
> [f1,f2] = nonlincon(x)
> % f1 =  1.0e-05 * [-0.0113;  -0.5822]
> % f2 = []
```

GYAKORLÓ FELADAT 4. – LINEÁRIS PROGRAMOZÁS A VÍZTISZTÍTÁSBAN[6]

Lineáris programozási feladatnak azt az esetet nevezzük, amikor a célfüggvény és a megkötések is lineárisak. Nézzünk most egy ilyen példát! Az ábrán látható folyón (1-4) és annak mellékfolyóján (2-3) négy víztisztító üzem működik, amelyek a közeli nagyvárosok *P* (mg/nap) szennyvízét dolgozzák fel *x* mértékben, azaz a folyóba jutó szennyeződés:

$$W = (1 - x) \cdot P$$

Ha x_i az *i.* város szennyvíztisztítójának tisztítási foka, akkor az eltávolított szennyezés $x_i \cdot P_i$.

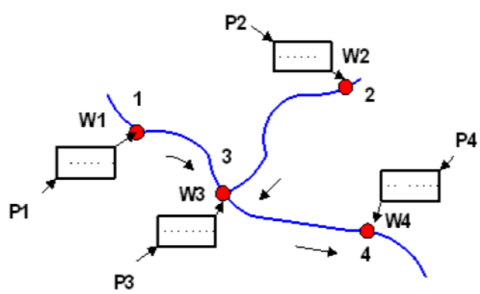
Amikor W_i szennyeződés belép a folyóba, ott feltételezés szerint tökéletesen keveredik a folyóval érkező $Q_{i,j}$ térfogatáramú (i→j) és c_i koncentrációjú szennyeződéssel. Azaz a szennyvíz-koncentrációk (mg/L) értéke a bekeveredés után az egyes szakaszokon:

$$c_1 = \frac{1 - x_1}{Q_{13}} \cdot P_1$$

$$c_2 = \frac{1 - x_2}{Q_{23}} \cdot P_2$$

$$c_3 = \frac{R_{13} \cdot Q_{13} \cdot c_1 + R_{23} \cdot Q_{23} \cdot c_2 + (1 - x_3) \cdot P_3}{Q_{34}}$$

$$c_4 = \frac{R_{34} \cdot Q_{34} \cdot c_3 + (1 - x_4) \cdot P_4}{Q_{45}}$$



A megfelelő (i→j) folyószakaszon természetes lebomlás is történik. Ezt fejezik ki az R_{ij} tényezők. A rendelkezésre álló adatokat az alábbi táblázat tartalmazza:

Város *i*	Szennyvíz terhelés P_i (mg/nap)	Fajlagos tisztítási költség, d_i (Ft/mg)	Folyó-szakasz i→j	Szakasz víz-hozama, $Q_{i,j}$ (L/nap)	Term. lebomlási arány R_{ij}	Megengedett szennyeződési koncentráció, c_{si} (mg/L)
1	$1.0 \cdot 10^9$	$2 \cdot 10^{-6}$	1→3	$1.0 \cdot 10^7$	0.5	20
2	$2.0 \cdot 10^9$	$2 \cdot 10^{-6}$	2→3	$5.0 \cdot 10^7$	0.35	20
3	$4.0 \cdot 10^9$	$4 \cdot 10^{-6}$	3→4	$1.1 \cdot 10^8$	0.6	20
4	$2.5 \cdot 10^9$	$4 \cdot 10^{-6}$	4→5	$2.5 \cdot 10^8$	-	20

A napi tisztítás üzemeltetési költsége:

$$Z(x) = \sum_{i=1}^{4} d_i \cdot P_i \cdot x_i$$

Az optimalizálási feladat, hogy úgy határozzuk meg az egyes üzemek tisztítási fokát (x_i), hogy ezáltal minimalizájuk az üzemeltetési költséget (*Z*), de ugyanakkor az egyes csomópontokban a szennyeződés koncentrációja ne haladjon meg egy előírt mértéket, azaz $c_i < c_{si}$. A célfüggvény lineáris és a megkötések is azok.

A megfelelő értékeket behelyettesítve a célfüggvény:

[6] Paláncz Béla példatárából (2012)

$$Z = 2000 \cdot x_1 + 4000 \cdot x_2 + 16000 \cdot x_3 + 10000 \cdot x_4$$

A megkötések a koncentrációkra pedig:

$$100 \cdot (1 - x_1) \leq 20$$

$$40 \cdot (1 - x_2) \leq 20$$

$$47.2727 - 4.54545 \cdot x_1 - 6.36364 \cdot x_2 - 36.3636 \cdot x_3 \leq 20$$

$$22.48 - 1.2 \cdot x_1 - 1.68 \cdot x_2 - 9.6 \cdot x_3 - 10 \cdot x_4 \leq 20$$

További megkötések a tisztítási fokokra:

$$0 \leq x_i \leq 1$$

A feladat tehát egy szabványos lineáris programozási probléma!

```
>  %% Lineáris programozás
>  clear all; clc
>  % a célfüggvény együttható vektora
>  c=[2000;4000;16000;10000];
>  % Lineáris megkötés egyenlőtlenséggel : vannak
>  % A*x <= b alakban
>  A = [-100, 0, 0, 0; 0, -40, 0, 0; - 4.54545,-6.36364,-36.3636,0; -
   1.2, -1.68,-9.6, -10];
>  b = [-80;-20; -27.2727;-2.48];
>  % Lineáris megkötés egyenlőséggel : nincs
>  Aeq = []; beq = [];
>  % A változókra alsó korlát : van
>  lb = [0 0 0 0];
>  % A változókra felső korlát : van
>  ub = [1 1 1 1];
>  sol = linprog(c,A,b,Aeq,beq,lb,ub,optimset('Display','iter'))
>  % sol = 0.8; 0.5; 0.5625; 0
```

A FEJEZETBEN HASZNÁLT ÚJ FÜGGVÉNYEK

fmincon	-	feltételes szélsőérték keresés
deal	-	bemenetek szétosztása kimenetekre
linprog	-	lineáris programozási feladat megoldása