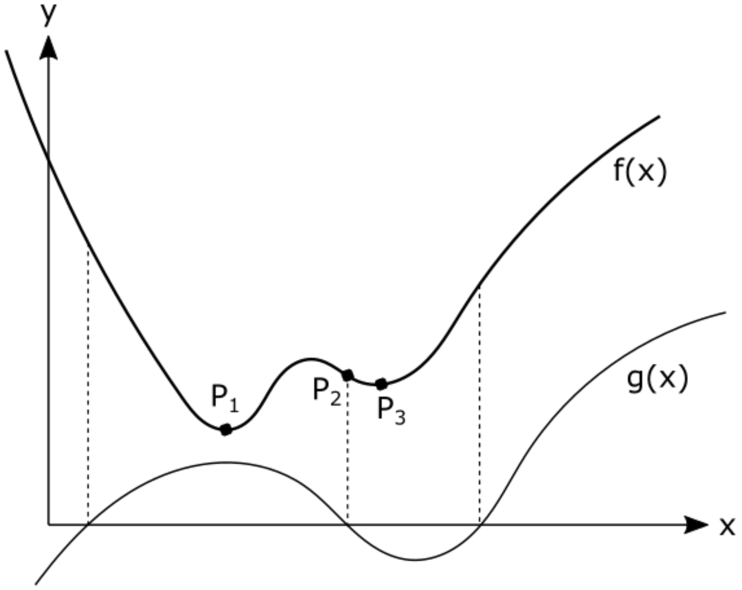
OPTIMALIZÁCIÓ MEGKÖTÉSEKKEL

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 (LAGRANGE ÉS BÜNTETŐ FÜGGVÉNY MÓDSZERE)

Válasszuk külön a megkötéses szélsőérték keresés feladati 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!

Keressük meg az alábbi függvény két lokális minimumát az $x \in [-4,4]; y \in [-4,4]$ tartományon:

$$f(x,y) = x^2 + y^2$$

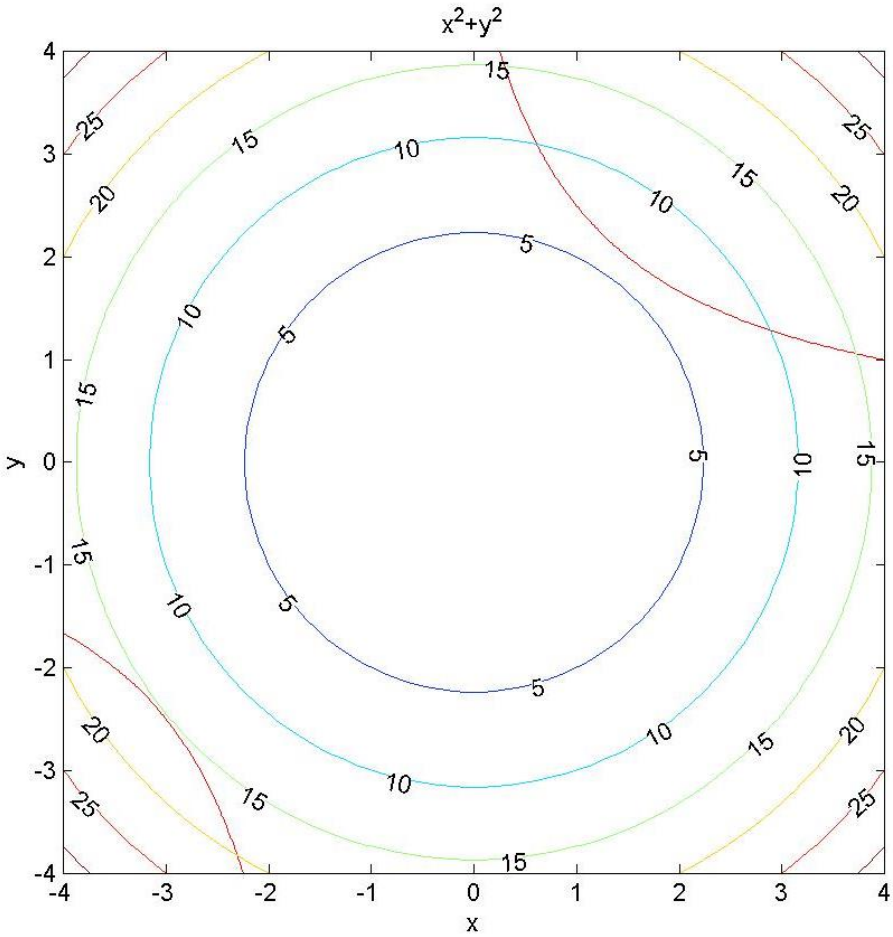
az alábbi megkötés mellett

$$x \cdot y + y - 5 = 0$$

Először ábrázoljuk a célfüggvényt és a megkötést is!

```
> %% f(x,y)=x^2+y^2 minimuma [-4,4]x[-4,4] intervallumon
> % g(x,y)=x+y-2=0 megkötés mellett
> clear all;format short;clc;
>
```

```
> % Függvény definiálása
> f = @(x,y) x.^2+y.^2
> g = @(x,y) x.*y+y-5
>
> % megkötés ábrázolása
> figure(1)
> h=fimplicit(g,[-4,4,-4,4]);
> set(h, 'Color','r')
>
> % szintvonalas ábra
> hold on
> h=ezcontour(f,[-4 4 -4 4]);
> set(h,'ShowText','on');
> axis square
```



Az ábra alapján két lokális minimumhelyünk lesz, az x,y koordinátákra tudunk adni kezdőértéket, az egyikre kezdőértéknek válasszuk az [1,2] pontot, a másiknak pedig a [-3,-2.5] pontot.

Oldjuk meg a feladatot Lagrange módszerrel is és a büntető függvény módszerével is!

LAGRANGE-MÓDSZER

Keressük az $f(x)$ függvény minimumát a $g(x) = 0$ feltétel mellett. Az eredeti feladat helyett tekintsük az alábbi függvény megkötés nélküli minimalizálását:

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

ahol λ_i-k a Lagrange-féle multiplikátorok (szorzók). A minimum szükséges feltétele a parciális deriváltak eltűnése, azaz

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

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

$i = 1,2, \cdots, m$, ahol *m* az egyenlőséggel adott megkötések száma.

A minimum elégséges feltétele, hogy ezen kívül az $L(x, \lambda)$ függvény *Hesse mátrixa* pozitív definit legyen, azaz a mátrix sajátértékei pozitívak legyenek a szélsőérték helyén.

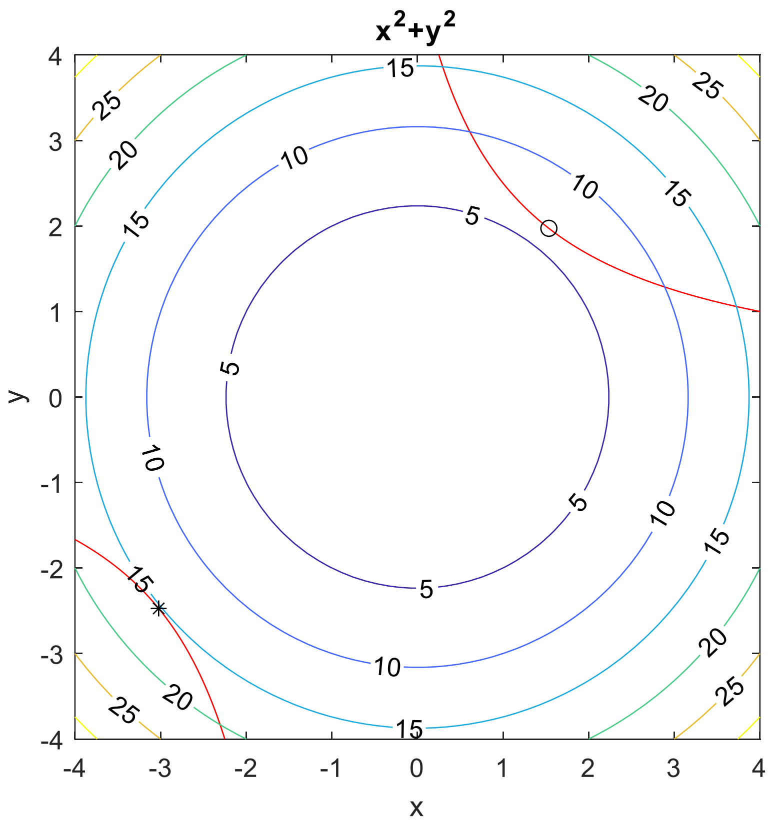
Oldjuk meg a feladatunkat Lagrange módszer használatával! A parciális deriváltak kiszámítását szimbolikusan végezzük el. Számítsuk ki a lokális minimumokban a függvény értékeit is és rajzoljuk be a megoldásokat a szintvonalas ábrába.

```
> %% megoldás Lagrange-módszerrel
> % a Lagrange-függvény megadása:
> L = @(x,y,lambda) f(x,y)+lambda*g(x,y);
> syms x y lambda
> dx=diff(L(x,y,lambda),x)
> % ans = 2*x+lambda*y
> dy=diff(L(x,y,lambda),y)
> % ans = 2*y+lambda*(x+1)
> dl=diff(L(x,y,lambda),lambda)
> % ans = x*y+y-5
>
> % A szükséges feltételek szerint a parciális deriváltak nullák
> % 2*x + lambda*y = 0
> % 2*y + lambda*(x+1) = 0
> % x*y + y - 5 = 0
> % Ez egy nemlineáris egyenlet rendszer!
```

A parciális deriváltak elvégzése után egy nemlineáris egyenletrendszert kapunk, aminek keressük a megoldását. Írjuk fel az egyenletrendszert és oldjuk meg a korábban tanultak szerint, az fsolve használatával! Az ábra alapján az x,y koordinátákra tudunk adni kezdőértékeket, ezek most legyenek az [1,2] és a [-3,-2.5] pontok. A lambda változóhoz az ábra alapján nem tudunk kezdőértéket rendelni, válasszuk ennek értékét most 1-nek.

```
>  % A nemlineáris egyenletrendszer
> G = @(x,y,lambda) [2*x+lambda*y; 2*y+lambda*(x+1); x*y+y-5]
> % A nemlineáris egyenletrendszer vektorizálása
> % x => x(1), y => x(2), lambda =>x(3)
> G = @(x) G(x(1),x(2),x(3))
```

```
>
> % a megoldás
> x01 = [1;2;1];  % az 1. kezdőérték
> xyl1 = fsolve(G,x01,optimset('Display','iter'))
> % xyl1 =
> %     1.5349
> %     1.9725
> %    -1.5563
>
> % berajzoljuk az ábrába
> figure(1)
> plot(xyl1(1),xyl1(2),'ko')
>
> % az egyik lokális minimum értéke
> zopt1 = f(xyl1(1),xyl1(2))
> % ans = 6.2465
>
> % a másik lokális minimum megkeresése
> x02 = [-3;-2.5;1];  % a 2. kezdőérték
> xyl2 = fsolve(G,x02,optimset('Display','iter'))
> % xyl2 =
> %    -3.0224
> %    -2.4723
> %    -2.4450
>
> % berajzoljuk az ábrába
> plot(xyl2(1),xyl2(2),'k*')
> % a minimum értéke
> zopt2 = f(xyl2(1),xyl2(2))
> % ans = 15.2472
```

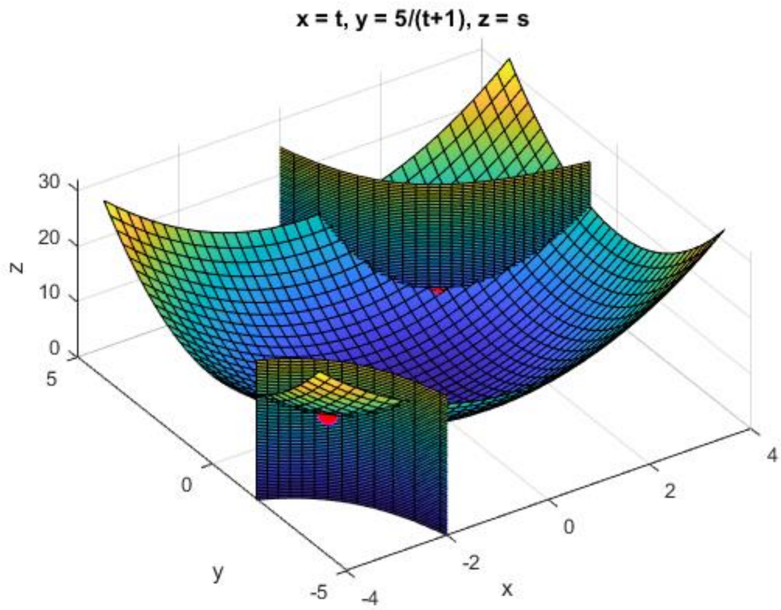


Mivel itt a nemlineáris egyenletrendszer algebrai polinomokból áll, így a feladatot megoldhattuk volna a szimbolikus számítást használó **solve** segítségével is, kezdőérték megadása nélkül, egy lépésben megkapva az összes gyököt, jelen esetben két valós és két komplex gyököt.

```
>  % Megoldás solve használatával
>   sol = solve([dx;dy;dl],[x,y,lambda])
>  %    struct with fields:
>  %          x: [4×1 sym]
>  %          y: [4×1 sym]
>  %     lambda: [4×1 sym]
>  x = double(sol.x); y = double(sol.y); lambda = double(sol.lambda);
>  [x y lambda]
>  %    1.5349 + 0.0000i   1.9725 + 0.0000i  -1.5563 + 0.0000i
>  %   -0.7562 - 2.1948i   0.2499 + 2.2503i   2.0006 - 0.4499i
>  %   -0.7562 + 2.1948i   0.2499 - 2.2503i   2.0006 + 0.4499i
>  %   -3.0224 + 0.0000i  -2.4723 + 0.0000i  -2.4450 + 0.0000i
```

A jobb szemléltetés kedvéért készítsük el a minimalizálandó függvény és a megkötés térbeli ábráját és rajzoljuk be a kapott megoldásokat! A megkötés függőleges felületét csak paraméteresen tudjuk ábrázolni (`fsurf(funx,funy,funz,uvinterval)`).

```
>  % a függvény ábrázolása 3D-ben
>  figure(2); fsurf(f,[-4,4,-4,4]); hold on;
>  % megkötés mint paraméteres felület ábrája
>  % Függőleges felületet csak paraméteresen tudunk ábrázolni:
>  % x.*y+y-5 = 0  -> y*(x+1) = 5  ->  y = 5/(x+1)
>  xp = @(u,v) u
>  yp = @(u,v) 5./(u+1)
>  zp = @(u,v) v
>  fsurf(xp,yp,zp,[-4,4,0,25])
>  % y tengely beállítása
>  ylim([-5,5])
>  % megoldások berajzolása
>  plot3(xyl1(1),xyl1(2),zopt1,'r*')
>  plot3(xyl2(1),xyl2(2),zopt2,'mo','MarkerFaceColor','r','MarkerSize',1
   0)
```



BÜNTETŐFÜGGVÉNY MÓDSZERE

Oldjuk meg az előző példát büntetőfüggvény módszerével is! 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:

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

ahol $K > 0$, egy új paraméter, aminek a segítségével 'megbüntetjük' *g(x)* 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 kisebb értékű feltételes minimum helyét a büntetőfüggvény módszerével is. A büntetés-függvény skalár paramétere legyen rendre 10, 100, 1000 és 10000. Vessük össze a megoldást a Lagrange-módszerrel kapott megoldással. Miután visszavezettük a feladatot egy feltétel nélküli optimalizációs feladatra, a korábban használt módszerrel megoldhatjuk a feladatot. Használhatjuk akár a Matlab beépített kvázi Newton-módszert használó **fminunc**, vagy a szimplex módszert használó **fminsearch** függvényét is.

```
> %% Büntetőfüggvény módszere
> % a minimalizálandó büntetés-függvény
> punish = @(x,y,K)  f(x,y) + K * g(x,y).^2;
> punish = @(x,K) punish(x(1), x(2), K); % x,y helyett vektorváltozó
> % ennek minimumát különböző K paraméterekre határozzuk meg
>
> % A punish függvénynek két paramétere van, viszont az fminunc ill. az
> % fminsearch számára egy paraméteres függvény kell, ezért anonymous
> % függvényeket definiálunk a K paraméter aktuális értékével.
> x0 = [1,2]
> Ksol1 = fminsearch(@(x) punish(x,10), x0)  % Ksol = 1.5179    1.9550
> Ksol2 = fminsearch(@(x) punish(x,100), x0) % Ksol = 1.5332    1.9707
> Ksol3 = fminsearch(@(x) punish(x,1000),x0) % Ksol = 1.5348    1.9723
> Ksol4 = fminsearch(@(x) punish(x,10000),x0)% Ksol = 1.5348    1.9725
> % A Lagrange-módszerrel kapott megoldás:          1.5349    1.9725
> % Ellenőrzés: a függvény és a megkötés értéke a kapott pontokban:
> f(Ksol1(1),Ksol1(2)),g(Ksol1(1),Ksol1(2)) % 6.1258,   -0.0776
> f(Ksol2(1),Ksol2(2)),g(Ksol2(1),Ksol2(2)) % 6.2344,   -0.0078
> f(Ksol3(1),Ksol3(2)),g(Ksol3(1),Ksol3(2)) % 6.2453, -7.7697e-04
> f(Ksol4(1),Ksol4(2)),g(Ksol4(1),Ksol4(2)) % 6.2464, -7.7373e-05
```

A Lagrange-módszerrel kapott megoldás: 1.5349, 1.9725. Látszik, hogy a K paraméter növelésével ehhez tart a megoldásunk.

MEGOLDÁS BEÉPÍTETT MATLAB FÜGGVÉNNYEL (FMINCON)

Természetesen a Matlab-nak is van saját beépített függvénye a feltételes szélsőérték kereséshez, az **fmincon** függvény. (**fminunc** - find minimum of unconstrained function (megkötés nélküli szélsőérték), **fmincon** - find minimum of constrained function (megkötéses szélsőérték)).

Az **fmincon** egy általánosan használható szélsőérték keresés, nem csak egyenletekkel, hanem egyenlőtlenségekkel is lehet feltételeket megadni, akár lineáris, akár nemlineáris formában, illetve felső/alsó korlátot is lehet megadni. A legegyszerűbb meghívása:

```
x = fmincon(fun,x0,A,b)
```

Itt *'fun'* a célfüggvény, *'x0'* a kezdőérték, *'A'* és *'b'* pedig a lineáris egyenlőtlenséggel ($A \cdot x < b$) megadott megkötések együttható mátrixa, ill. az egyenletrendszer jobb oldala. Amennyiben nem csak lineáris egyenlőtlenségek vannak a megkötésekre, akkor több paraméterrel kell meghívni a függvényt:

```
x  = 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*: $A \cdot x = b$ 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 contraint)
– *options*: opciók

A feltételeket megadott sorend szerint kell megadni, amilyen feltételünk nincs, annak a helyére üres mátrixot kell tenni. A függvényt meghívhatjuk több kimenetettel is, ha több információra van szükségünk:

```
[x,fval,exitflag,output,lambda,grad,hessian] =  fmincon(...)
```

Oldjuk meg az előző feladatot a beépített függvénnyel is!

A célfüggvény:	$f(x,y) = x^2 + y^2$
A nemlineáris egyenlettel (*ceq*) adott megkötés:	$x \cdot y + y - 5 = 0$
A megadott alsó/felső határok:	$x \in [-4,4]; y \in [-4,4]$

```
> F=@(x) f(x(1),x(2)); % vektorizált célfüggvény
> x0=[1;2]; % kezdeti érték
> % Megkötésből nincs se lineáris egyenlőtlenség, se egyenlet
> A=[]; b=[]; Aeq=[]; beq=[];
> % Alsó/felső korlát van
> lb=[-4,-4]; ub=[4,4];
> % nemlineáris megkötések vannak, de csak egyenlettel megadva
> c = []; % egyenlőtlenség nincs
> ceq = @(x) [g(x(1),x(2))] % vektorizált megkötés
```

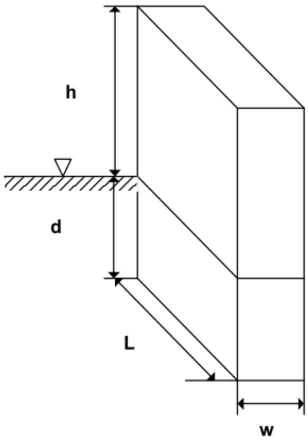
Az **fmincon**-nak egyben két kimeneti értéket [c ceq] előállító függvény kell, ezeket a beépített **deal** függvénnyel állíthatjuk elő.

```
> nonlincon = @(x) deal(c, ceq(x))
> sol  = fmincon(F,x0,A,b,Aeq,beq,lb,ub,nonlincon)
> % sol =
> %     1.5349
> %     1.9725
> % A Lagrange-módszerrel kapott megoldás ugyanez: 1.5349, 1.9725
> figure(2)
> plot3(sol(1),sol(2),f(sol(1),sol(2)),'kd','MarkerFaceColor','g')
> % ellenőrizzük a megkötéseket:
> [glin,gnonlin] = nonlincon(sol)
> % glin = []; gnonlin = -1.3407e-08
> % A függvény érték:
> f(sol(1),sol(2)) % 6.2465
```

KÉTVÁLTOZÓS FÜGGVÉNY FELTÉTELES SZÉLSŐÉRTÉKE EGYENLŐTLENSÉGI MEGKÖTÉSEKKEL

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

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

Az egyenlőtlenségi megkötések:

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

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

$$g_3(d,w) = 45815w - 523.6wd + 161.8w^2 - 225000 \le 0$$ (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,NONLCON,OPTIONS)
> % 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 Matlab egysoros függvénnyel - 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 = @(x) [g1(x(1),x(2)); g2(x(1),x(2))]
> % Megjegyzés:
```

```
>  %   az fmincon-nak két kimeneti értéket [c ceq] előállító függvény
   kell,
>  %   ezért ezeket a beépített deal függvénnyel állítjuk elő
>  nonlincon = @(x) deal(c(x), ceq)
>  % nonlincon = @(x) deal([20000-25*1.618*x(2)^2;
>  %                  45815*x(2)-523.6*x(2)*x(1)+161.8*x(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))
>  % a megoldás:
>  % 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, optimset
   ('Display','iter'))
>
>  % ellenőrizzük le a nemlineáris megkötések értékeit:
>  [f1,f2] = nonlincon(x)
>  % f1 =  1.0e-05 * [-0.0113;  -0.5822]
>  % f2 = []
```

GYAKORLÓ FELADAT 1.

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: $V = \dfrac{r^2 \cdot \pi \cdot h}{3}$

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)

<u>Megoldás:</u>

```
> %% 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 2.

Adott a következő felület a $-0.5 < x < 0.5$; $-0.5 < y < 1$ tartományon.

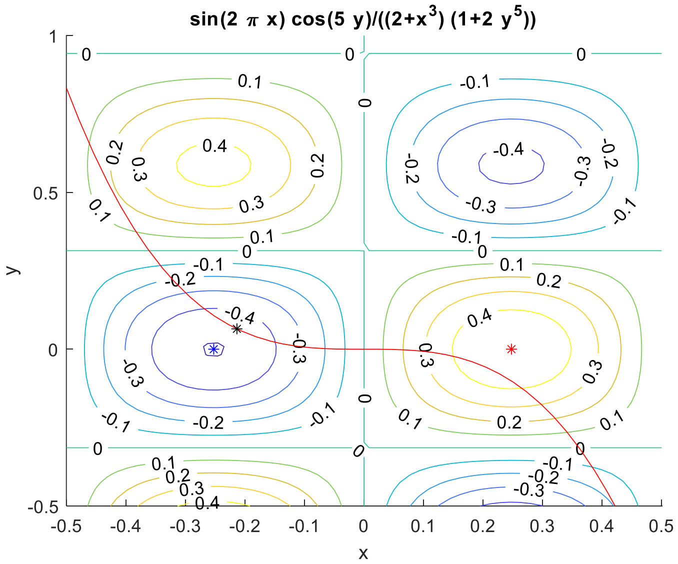
$$f(x,y) = \frac{\sin(2\cdot\pi\cdot x)\cdot\cos(5\cdot y)}{(2+x^3)\cdot(1+2\cdot y^5)}$$

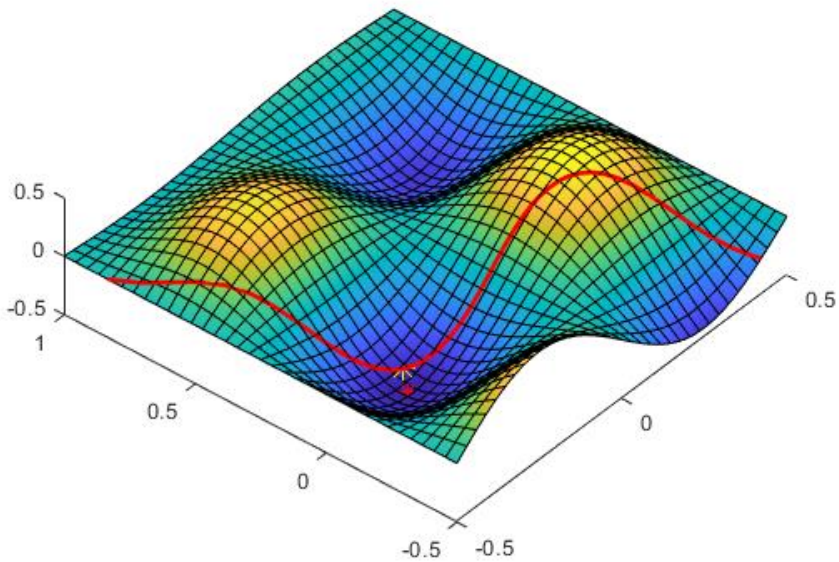
Határozza meg egy-egy lokális minimumát, maximumát a felületnek, majd a globális minimumot genetikus algoritmussal! Hatrározza meg a minimumot a következő megkötés mellett is:

$$y = \frac{-10\cdot x^3}{1.5}$$

```
> %% Felület, genetikus alg, útvonal menti optimalizálás
> clear all; clc; close all
> % Szintvonalas ábra
> figure(1);hold on
> f=@(x,y) sin(2*pi*x).*cos(5*y)./((2+x.^3).*(1+2*y.^5))
> h=ezcontour(f,[-0.5,0.5,-0.5,1])
> set(h,'ShowText','on')
> % Felület
> figure(2); hold on;
> fsurf(f,[-0.5,0.5,-0.5,1])
> view([-50 70]) % azimuth, elevation angle
>
> % Lokális szélsőértékek meghatározása
> F1=@(u)f(u(1),u(2))
> [x1 f1]=fminsearch(F1,[-0.35,0]) %  x1 = -0.25247   3.6791e-05, f1 =
  -0.504
> figure(1); plot(x1(1),x1(2),'b*')
>
> F2=@(u)-f(u(1),u(2))
> [x2 f2]=fminsearch(F2,[0.35,0]) % x2 = 0.24765   0.00018951, f2 = -
  0.49618
> plot(x2(1),x2(2),'r*')
>
> %Globális minimum
> options =
  gaoptimset('Generations',20,'PopulationSize',200,'Display','iter');
> xga = ga(F1,2,[],[],[],[],[-0.5,-0.5],[0.5,1],[],options)
> [xy_min zmin]=fminsearch(F1,xga)
> figure(2); hold on; plot3(xy_min(1),xy_min(2),F1(xy_min),'r*')
>
> % Szélsőérték keresése a megkötés mellett - útvonal mentén
> % megkötés berajzolása
> figure(1); hold on
> g = @(x) -10*x.^3/1.5
> h3 = fplot(g,[-0.5,0.5])
> set(h3,'Color','r');
> % térbeli pontok
> xi = linspace(-0.5,0.5,50)'; % 50 pont felvétele a tartományban
> yi = g(xi); % útvonal mentén az y koord. kiszámítása
> zi = f(xi,yi); % terepi magasságok az útvonal mentén
> figure(2); hold on;plot3(xi,yi,zi,'r','LineWidth',2)
> axis([-0.5,0.5,-0.5,1])
>
> % Szélsőérték keresés útvonal mentén
```

```
>  A=[];b=[];ub=[];lb=[];Aeq=[];beq=[];
>  % megkötést nullára rendezzük, vektorizáljuk!
>  nonlcon=@(x) deal([],[10*x(1)^3+1.5*x(2)])
>  x0=[-0.2;0];
>  sol_min=fmincon(F1,x0,A,b,Aeq,beq,lb,ub,nonlcon)
>  % Függvény minimum értéke a megkötés mellett:
>  sol_z = F1(sol_min)
>  figure(1); plot(sol_min(1),sol_min(2),'k*')
>  figure(2); plot3(sol_min(1),sol_min(2),sol_z,'y*','MarkerSize',10)
```





A FEJEZETBEN HASZNÁLT ÚJ FÜGGVÉNYEK

fmincon - feltételes szélsőérték keresés
deal - bemenetek szétosztása kimenetekre