OPTİMİZASYON TEKNİKLERİ

LABORATUVAR DERSİ

Temel Kavramlar – Tek Değişkenli Optimizasyon/ Kısıtsız Doğrusal Olmayan Optimizasyon

- **x0:** başlangıç değerleri (tahmini)
- **options:** "optimset" komutu ile seçilen optimizasyon seçeneklerinin atandığı değişken. Optimset yardımıyla standart veya büyük ölçekli algoritma seçimine bağlı olarak optimizasyon metodunun seçenekleri belirlenir. Bunlardan bazıları Jakobien veya Hessian matrisinin hesaplanıp hesaplanmayacağı, ekrana ne kadar bir bilgi yazılacağı gibi seçeneklerdir.
- **Display:** Çözüm adımlarından hangisinin komut satırında gösterileceği belirtilir **'off':** herhangi bir çıktı göstermez **'iter':** her bir iterasyonda çıktıları gösterir **'final':** sadece son çıktıyı gösterir **'notify':** optimizasyon yakınsamadığı zamanki çıktıyı gösterir.
- **exitflag:** Seçilen algoritmanın sonuca yakınsayıp yakınsamadığını belirtir. Eğer sıfırdan büyük ise lokal minimum değerinin bulunduğunu belirtir.
- **fun:** hedef fonkisyonu içeren m-dosyasının ismi
- **fval:** Optimum noktada hedef fonksiyonun değeri

Temel Kavramlar – Tek Değişkenli Optimizasyon/ Kısıtsız Doğrusal Olmayan Optimizasyon

- **Output:** Optimizasyon çözümü hakkında daha detaylı bilgi verir. Verilecek örneklerde komut satırına output yazılıp Enter'lanırsa aşağıdaki bilgiler elde edilir:

```
output =

  struct with fields:

    iterations: 35
     funcCount: 69
     algorithm: 'Nelder-Mead simplex direct search'
       message: 'Optimization terminated:↵ the current x sat
```

Iterations: toplam iterasyon sayısını
funcCount: Fonksiyonun değerlendirme sayısı
stepsize: Son iterasyonda seçilen adım uzunluğu
firstorderopt: birinci derece optimumluk şartı
algortihm: Kullanılan algoritmayı ve seçilen metodu gösterir.

Tek Değişkenli Optimizasyon - fminbnd

Sabit aralıkta minimum tek değişkenli işlevi bulun..

- **Sözdizimi**

x = fminbnd(fun,x1,x2)

x = fminbnd(fun,x1,x2,options)

x = fminbnd(problem)

[x,fval] = fminbnd(___)

[x,fval,exitflag] = fminbnd(___)

[x,fval,exitflag,output] = fminbnd(___)

Tek Değişkenli Optimizasyon - fminbnd

- **fminbnd**, x tarafından belirtilen bir problem için minimumu bulan tek boyutlu bir küçültücüdür, x_1 ve x_2 sonlu skalerdir ve f(x) bir skaler döndüren bir fonksiyondur.

$$\min_x f(x) \text{ ve } x_1 < x < x_2$$

- $\boldsymbol{x = fminbnd(fun, x_1, x_2)}$ $x_1 < x < x_2$ aralığında fun içinde açıklanan skaler değerli işlevin yerel küçültücüsü olan bir x değeri döndürür.

Tek Değişkenli Optimizasyon - fminbnd

Örnek: $0 < x < 2\pi$ aralığında $\sin(x)$ fonksiyonunun minimumunu aldığı noktayı bulun.

```
fun = @sin;
x1 = 0;
x2 = 2*pi;
x = fminbnd(fun,x1,x2)
```

Çıktı: `x = 4.7124`

Kesinliği göstermek için bu işlemler, x=3π/2 doğru değeriyle aynıdır.

```
3*pi/2
```

Çıktı: `x = 4.7124`

Tek Değişkenli Optimizasyon - fminbnd

- **x = fminbnd(fun,x1,x2,options)** «options» da belirtilen optimizasyon seçenekleri ile minimize eder. Bu seçenekleri ayarlamak için optimset'i kullanırız.
- **Örnek:** $0 < x < 2\pi$ için $\sin(x)$ işlevini en aza indirmek için **fminbnd**'nin attığı adımları izleyin.

```
fun = @sin;
x1 = 0;
x2 = 2*pi;
options = optimset('Display','iter');
x = fminbnd(fun,x1,x2,options)
```

```
 Func-count     x          f(x)         Procedure
    1        2.39996      0.67549        initial
    2        3.88322     -0.67549        golden
    3        4.79993     -0.996171       golden
    4        5.08984     -0.929607       parabolic
    5        4.70582     -0.999978       parabolic
    6         4.7118            -1       parabolic
    7        4.71239            -1       parabolic
    8        4.71236            -1       parabolic
    9        4.71242            -1       parabolic

Optimization terminated:
 the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04

x = 4.7124
```

**optimset: Optimizasyon seçenekleri yapısı oluşturur veya değiştirir.

Tek Değişkenli Optimizasyon - fminbnd

- **x = fminbnd(problem),** problemin bir structure olduğu problem için minimumu bulur.
- **[x,fval] = fminbnd(___),** herhangi bir girdi argümanı için, x çözümünde «fun» da hesaplanan amaç fonksiyonunun değerini döndürür.
- **Örnek:** $0 < x < 2\pi$ için minimum $\sin(x)$'in yerini ve minimumun değerini bulun.

```
fun = @sin;
[x,fval] = fminbnd(fun,0,2*pi)
```

Çıktı:

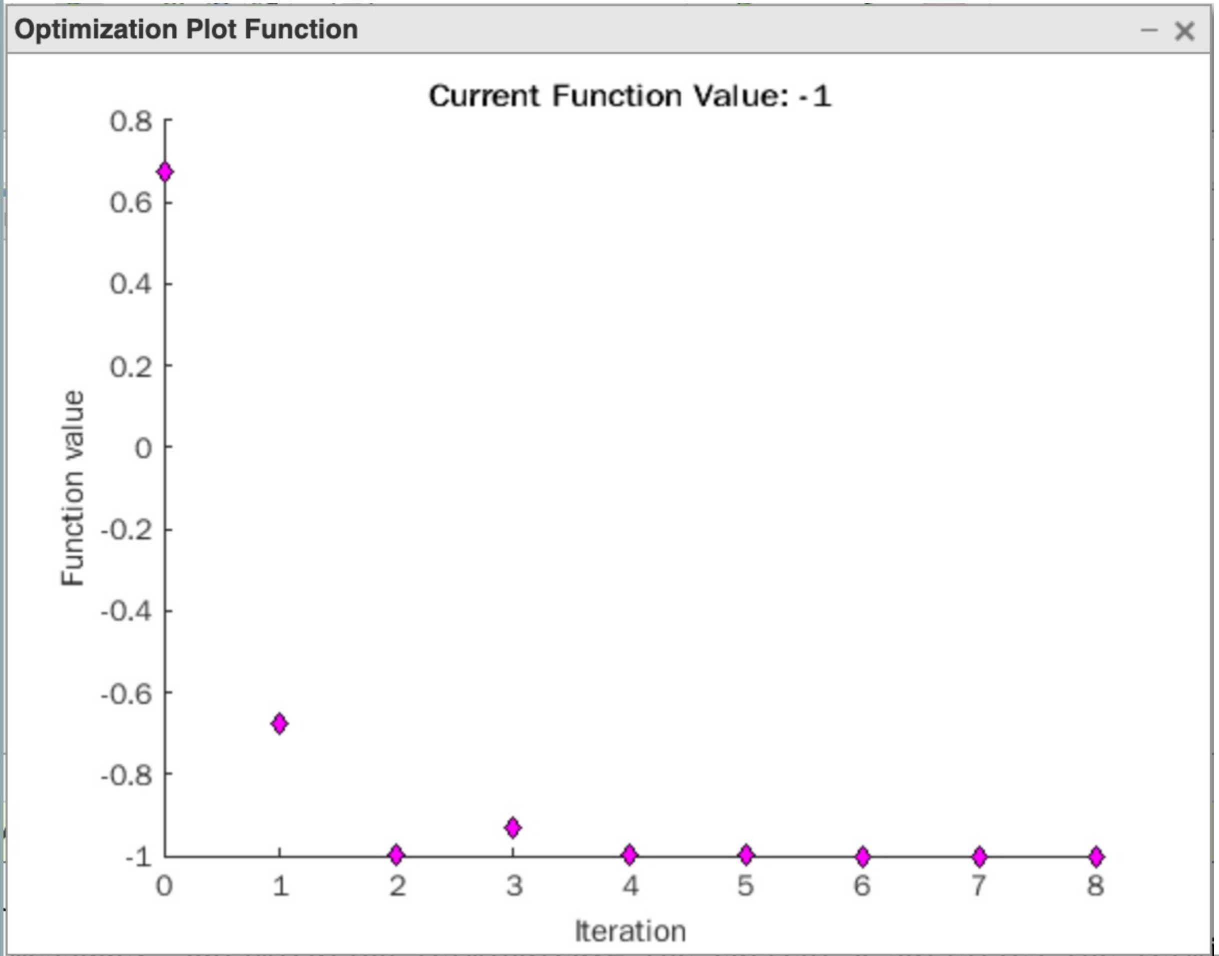
```
x = 4.7124

fval = -1.0000
```

Tek Değişkenli Optimizasyon - fminbnd

- **[x,fval,exitflag] = fminbnd(___)** ayrıca çıkış koşulunu açıklayan bir çıkış bayrağı değeri döndürür.
- **Örnek:** Tüm çıktıları talep ederek fminbnd çözüm süreciyle ilgili tüm bilgileri döndürün. Ayrıca, bir çizim işlevi kullanarak çözüm sürecini izleyin.

```
fun = @sin;
x1 = 0;
x2 = 2*pi;
options = optimset('PlotFcns',@optimplotfval);
[x,fval,exitflag,output] = fminbnd(fun,x1,x2,options)
```

Tek Değişkenli Optimizasyon - fminbnd




```
x = 4.7124

fval = -1.0000

exitflag = 1

output = struct with fields:
    iterations: 8
     funcCount: 9
     algorithm: 'golden section search, parabolic interpolation'
       message: 'Optimization terminated:...'
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch

Türevsiz yöntemi kullanarak minimum kısıtsız çok değişkenli işlevi bulun.

- **Sözdizimi**

```
x = fminsearch(fun,x0)
x = fminsearch(fun,x0,options)
x = fminsearch(problem)
[x,fval] = fminsearch(___)
[x,fval,exitflag] = fminsearch(___)
[x,fval,exitflag,output] = fminsearch(___)
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch

Doğrusal olmayan programlama çözücü tarafından belirtilen bir problemin minimumunu arar.

$$\min_{x} f(x)$$

f (*x*) bir skaler döndüren bir fonksiyondur ve *x* bir vektör veya bir matristir.

- $\boldsymbol{x = fminsearch(fun, x_0)}$, $\boldsymbol{x_0}$ noktasında başlar ve "fun" içinde tanımlanan fonksiyonun yerel minimum x'ini bulmaya çalışır.

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Rosenbrock'un İşlevini En Aza İndirin

Örnek: Birçok algoritma için herkesin bildiği zor bir optimizasyon problemi olan Rosenbrock'un işlevini en aza indirelim:

$$f(x) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2$$

Fonksiyon, minimum 0 değeri ile x = [1,1] noktasında minimize edilir.

Başlangıç noktasını x0 = [-1.2,1] olarak ayarlayın ve `fminsearch`'ü kullanarak Rosenbrock'un işlevini en aza indirin.

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Rosenbrock'un İşlevini En Aza İndirin

```
fun = @(x)100*(x(2) - x(1)^2)^2 + (1 - x(1))^2;
x0 = [-1.2,1];
x = fminsearch(fun,x0)
```

Çıktı:

```
x =

    1.0000    1.0000
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Optimizasyon Sürecini İzleyin -

- **x = fminsearch(fun,x0,options)** yapı seçeneklerinde belirtilen optimizasyon seçenekleri ile minimize eder. Bu seçenekleri ayarlamak için optimset'i kullanın.

Örnek: fminsearch bir minimumu bulmaya çalışırken süreci izlemek için seçenekleri ayarlayın. Her yinelemede amaç fonksiyonunu çizmek için seçenekleri ayarlayın.

```
options = optimset('PlotFcns',@optimplotfval);
```

- Amaç işlevini Rosenbrock'un işlevine ayarlayın,

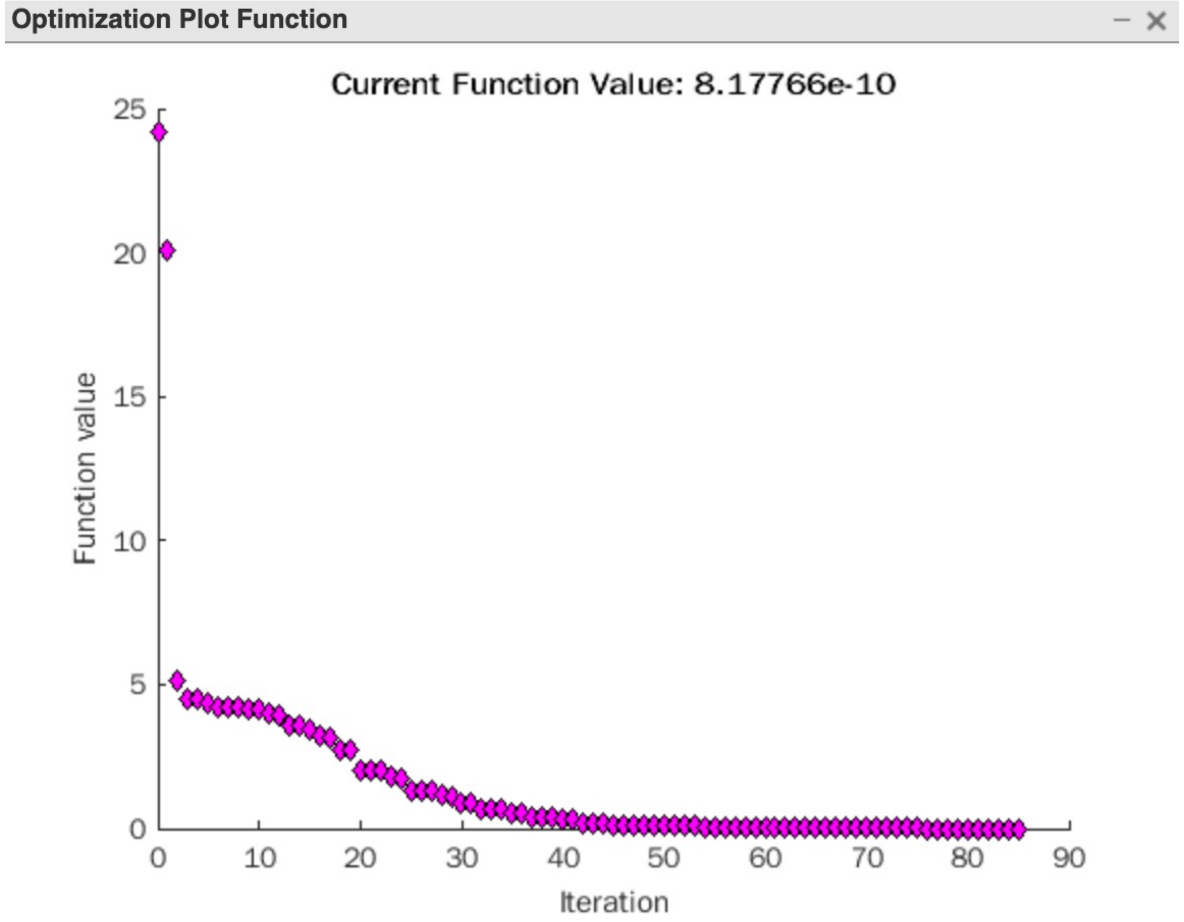
$$f(x) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2$$

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Optimizasyon Sürecini İzleyin -

- Fonksiyon, minimum 0 değeri ile x = [1,1] noktasında minimize edilir.
- Başlangıç noktasını x0 = [-1,2,1] olarak ayarlayın ve fminsearch'ü kullanarak Rosenbrock'un işlevini en aza indirin.

```
fun = @(x)100*(x(2) - x(1)^2)^2 + (1 - x(1))^2;
x0 = [-1.2,1];
x = fminsearch(fun,x0,options)
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Optimizasyon Sürecini İzleyin -



Çıktı: x =

1.0000 1.0000

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Dosya Tarafından Belirtilen Bir Fonksiyonu Küçült

- Bir dosya çalıştırılarak değerleri verilen bir nesnel işlevi simge durumuna küçültün. Bir işlev dosyası gerçek bir x vektörünü kabul etmeli ve nesnel işlevin değeri olan gerçek bir skaler döndürmelidir. Aşağıdaki kodu kopyalayın ve **«objectivefcn1» adlı bir dosya olarak ekleyin.**

```
function f = objectivefcn1(x)
f = 0;
for k = -10:10
        f = f + exp(-(x(1)-x(2))^2 - 2*x(1)^2)*cos(x(2))*sin(2*x(2));
end

x0 = [0.25,-0.25];
x = fminsearch(@objectivefcn1,x0)
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Dosya Tarafından Belirtilen Bir Fonksiyonu Küçült

- Matlab'da aşağıdaki kodu yazdığımızda, fonksiyon küçültme kodu otomatik olarak çalışıyor.

Çıktı :

```
x =

   -0.1696   -0.5086
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Ekstra Parametrelerle Küçült

- Bazen amaç fonksiyonunuzun ekstra parametreleri vardır. Bu parametreler optimize edilecek değişkenler değil, optimizasyon sırasında sabit değerlerdir. Örneğin, Rosenbrock tipi işlevde bir a parametreniz olduğunu varsayalım.

$$f(x, a) = 100(x_2 - x_1^2)^2 + (a - x_1)^2.$$

- Bu fonksiyon, $x_1 = a,\ x_2 = a^2$'de minimum 0 değerine sahiptir. Örneğin, $a = 3$ ise, anonim bir işlev oluşturarak parametreyi amaç işlevinize dahil edebilirsiniz.
- Ekstra argümanlar olarak ekstra parametreleriyle amaç fonksiyonunu oluşturun.

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Ekstra Parametrelerle Küçült

```
f = @(x,a)100*(x(2) - x(1)^2)^2 + (a-x(1))^2;
```

Parametreyi workspace’e yazın.

```
a = 3;
```

Parametrenin çalışma alanı değerini içeren tek başına x'in anonim bir fonksiyonunu oluşturun.

```
fun = @(x)f(x,a);
```

Problemi x0 = [-1,1.9]'dan başlayarak çözün.

```
x0 = [-1,1.9];
x = fminsearch(fun,x0)
```

Çıktısı:

```
x = 1×2

    3.0000    9.0000
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Minimum Konum ve Değeri Bulun

- `fminsearch` kullanarak bir minimum amaç fonksiyonunun hem konumunu hem de değerini bulun.
- Üç değişkenli bir problem için isimsiz bir amaç fonksiyonu yazın.

```
x0 = [1,2,3];
fun = @(x)-norm(x+x0)^2*exp(-norm(x-x0)^2 + sum(x));
```

- x 0'dan başlayarak minimum «fun»'ı bulun. Minimumun da değerini bulun.

```
[x,fval] = fminsearch(fun,x0)
```

Çıktısı:

```
x = 1×3

    1.5359    2.5645    3.5932

fval = -5.9565e+04
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Optimizasyon Sürecini İnceleyin

- Bir optimizasyonun sonuçlarını hem çalışırken hem de tamamlandıktan sonra inceleyin.
- Çözücü (solver) çalışırken optimizasyon hakkında bilgi veren yinelemeli görüntüleme sağlamak için seçenekleri ayarlayın. Ayrıca, çözücü çalışırken amaç fonksiyonu değerini göstermek için bir çizim fonksiyonu ayarlayın.

options = optimset('Display','iter','PlotFcns',@optimplotfval);

- Bir amaç fonksiyonu ve başlangıç noktası belirleyin. MATLAB yolunuza bir dosya olarak `objectivefcn1` kodunu ekleyin.

```
function f = objectivefcn1(x)
f = 0;
for k = -10:10
        f = f + exp(-(x(1)-x(2))^2 - 2*x(1)^2)*cos(x(2))*sin(2*x(2));
end
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Optimizasyon Sürecini İnceleyin

```
options=optimset('Display','iter');
x0 = [0.25,-0.25];
fun = @objectivefcn1;
```

- Tüm çözücü çıktılarını alın. Çözücü bittikten sonra sonuçları incelemek için bu çıktıları kullanın.

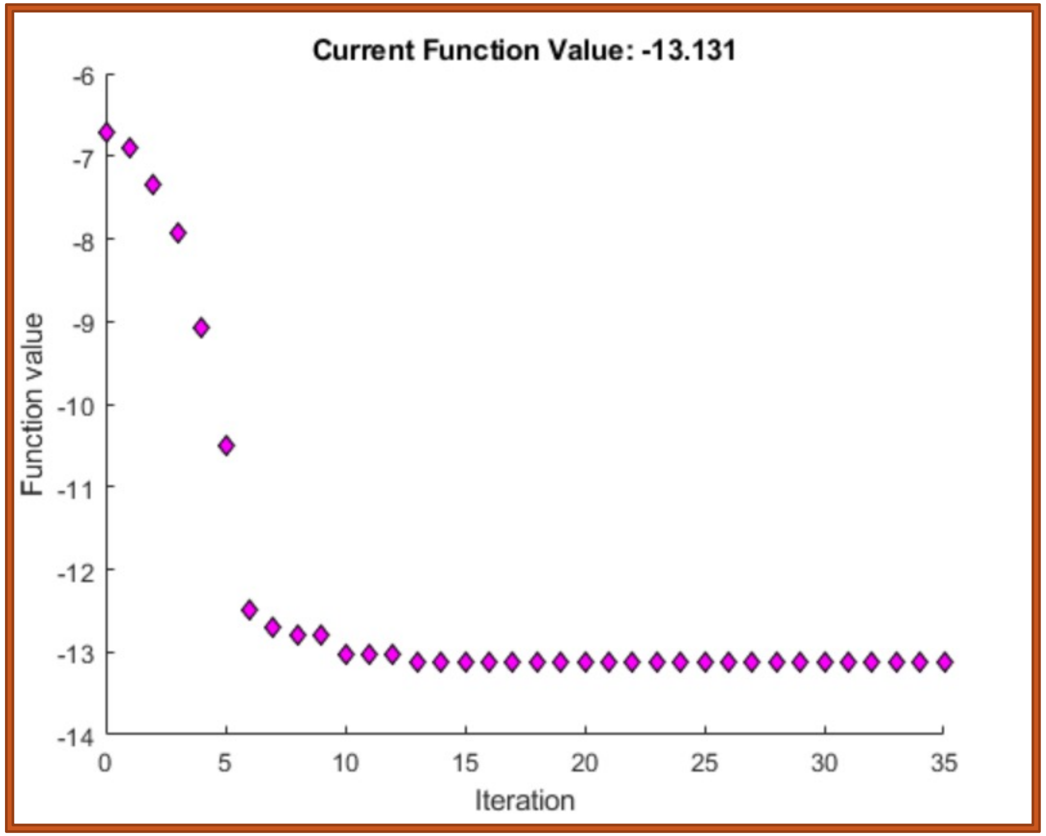
```
[x,fval,exitflag,output] = fminsearch(fun,x0,options)
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Optimizasyon Sürecini İnceleyin

Iteration	Func-count	min f(x)	Procedure
0	1	-6.70447	
1	3	-6.89837	initial simplex
2	5	-7.34101	expand
3	7	-7.91894	expand
4	9	-9.07939	expand
5	11	-10.5047	expand
6	13	-12.4957	expand
7	15	-12.6957	reflect
8	17	-12.8052	contract outside
9	19	-12.8052	contract inside
10	21	-13.0189	expand
11	23	-13.0189	contract inside
12	25	-13.0374	reflect
13	27	-13.122	reflect
14	28	-13.122	reflect
15	29	-13.122	reflect
16	31	-13.122	contract outside
17	33	-13.1279	contract inside
18	35	-13.1279	contract inside
19	37	-13.1296	contract inside
20	39	-13.1301	contract inside
21	41	-13.1305	reflect
22	43	-13.1306	contract inside
23	45	-13.1309	contract inside
24	47	-13.1309	contract inside
25	49	-13.131	reflect
26	51	-13.131	contract inside
27	53	-13.131	contract inside
28	55	-13.131	contract inside
29	57	-13.131	contract outside
30	59	-13.131	contract inside
31	61	-13.131	contract inside
32	63	-13.131	contract inside
33	65	-13.131	contract outside
34	67	-13.131	contract inside
35	69	-13.131	contract inside

```
Optimization terminated:
 the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04
 and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04


x =

   -0.1696   -0.5086


fval =

  -13.1310


exitflag =

     1


output =

  struct with fields:

    iterations: 35
     funcCount: 69
     algorithm: 'Nelder-Mead simplex direct search'
       message: 'Optimization terminated:...'
```

Kısıtsız Doğrusal Olmayan Optimizasyon - fminsearch - Optimizasyon Sürecini İnceleyin



Kısıtlı Doğrusal Olmayan Optimizasyon - fmincon

Minimum kısıtlı doğrusal olmayan çok değişkenli fonksiyonu bulun

- **Sözdizimi**

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
x = fmincon(problem)
[x,fval] = fmincon(___)
[x,fval,exitflag,output] = fmincon(___)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(___)
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Giriş Bağımsız Değişkenleri

fun — Küçültme işlevi
fonksiyon tanıtıcısı| fonksiyon adı

x0 — Başlangıç noktası
gerçek vektör| gerçek dizi

A — Doğrusal eşitsizlik kısıtlamaları
gerçek matris

b — Doğrusal eşitsizlik kısıtlamaları
gerçek vektör

Aeq — Doğrusal eşitlik kısıtlamaları
gerçek matris

beq — Doğrusal eşitlik kısıtlamaları
gerçek vektör

lb — Alt Sınıf
gerçek vektör| gerçek dizi

ub — Üst Sınır
gerçek vektör| gerçek dizi

nonlcon — Doğrusal olmayan kısıtlamalar
fonksiyon tanıtıcısı| fonksiyon adı

options — Optimizasyon seçenekleri
optimoptions çıktısı | optimset getirileri gibi bir yapı

problem — Problem Yapısı
yapı

Kısıtlı Doğrusal Olmayan Optimizasyon - fmincon

Tanım: Doğrusal olmayan programlama çözücü.

$$\min_{x} f(x) \text{ öyle } ki \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A.x \leq b \\ Aeq.x = beq \\ lb \leq x \leq ub, \end{cases}$$

b ile belirtilen bir problemin minimumunu bulur ve beq vektörlerdir, A ve Aeq matrislerdir, $c(x)$ ve $ceq(x)$ vektörleri döndüren fonksiyonlardır ve $f(x)$ bir skaler döndüren bir fonksiyondur. $f(x), c(x)$ ve $ceq(x)$ doğrusal olmayan fonksiyonlar olabilir.

$x, lb\ ve\ ub$ vektörler veya matrisler olarak geçirilebilir.

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Eşitsizlik Kısıtlaması

- `x = fmincon(fun,x0,A,b)` x_0 'dan başlar ve fun'da tanımlanan fonksiyonun $A * x \leq b$ doğrusal eşitsizliklerine tabi olan bir x'i minimize edeni bulmaya çalışır. x0 bir skaler, vektör veya matris olabilir.

Örnek: Doğrusal bir eşitsizlik kısıtlaması olduğunda Rosenbrock fonksiyonunun minimum değerini bulun.

Amaç «fun» fonksiyonunu Rosenbrock'un işlevi olacak şekilde ayarlayın. (1,1) noktasında minimum objektif değeri 0'dır.

```
fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Eşitsizlik Kısıtlaması

$x(1) + 2x(2) \leq 1$ ile sınırlandırılmış $[-1{,}2]$ noktasından başlayarak minimum değeri bulun. Bu kısıtı $A = [1{,}2]$ ve $b = 1$ alarak $Ax <= b$ biçiminde ifade edin. Bu kısıtlamanın, çözümün kısıtsız çözümde (1,1) olmayacağı anlamına geldiğine dikkat edin, çünkü bu noktada $x\,(1) + 2x(2) = 3 > 1$.

```
x0 = [-1,2];
A = [1,2];
b = 1;
x = fmincon(fun,x0,A,b)
```

Çıktısı:

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    0.5022    0.2489
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Eşitsizlik ve Eşitlik Kısıtlaması

- `x = fmincon(fun,x0,A,b,Aeq,beq)` fun’ı `Aeq*x = beq` ve `A*x ≤ b` doğrusal eşitliklerine bağlı olarak en aza indirir. Eşitsizlik yoksa, `A = []` ve `b = []` olarak ayarlayın.
- **Örnek :** Hem doğrusal eşitsizlik kısıtlaması hem de doğrusal eşitlik kısıtlaması olduğunda Rosenbrock fonksiyonunun minimum değerini bulun.
- Amaç fonksiyonu fun’ı Rosenbrock'un işlevi olacak şekilde ayarlayın.

```
fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Eşitsizlik ve Eşitlik Kısıtlaması

`[0.5,0]` noktasından başlayarak `x(1)+2x(2)≤1` ve `2x(1)+x(2)=1` ile sınırlandırılmış minimum değeri bulun.

- Doğrusal eşitsizlik kısıtını `A*x <= b` biçiminde `A = [1,2]` ve `b = 1` alarak ifade edin.
- Doğrusal eşitlik kısıtını `Aeq*x = beq` biçiminde, `Aeq = [2,1]` ve `beq = 1` alarak ifade edin.

```
x0 = [0.5,0];
A = [1,2];
b = 1;
Aeq = [2,1];
beq = 1;
x = fmincon(fun,x0,A,b,Aeq,beq)
```

Çıktısı:

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    0.4149    0.1701
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Sınırlı Kısıtlamalarla Küçültün

`x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)` x'deki tasarım değişkenleri üzerinde bir dizi alt ve üst sınır tanımlar, böylece çözüm her zaman `lb ≤ x ≤ ub` aralığında olur. Eşitlik yoksa, `Aeq = []` ve `beq = []` olarak ayarlayın. Eğer `x(i)` aşağıda sınırsız ise, `lb(i) = -Inf` olarak ayarlayın ve `x(i)` yukarıda sınırsız ise, `ub(i) = Inf` olarak ayarlayın.

Örnek: Sınırlı kısıtlamaların varlığında bir amaç fonksiyonunun minimumunu bulun. Amaç fonksiyonu, iki değişkenli basit bir cebirsel fonksiyondur.***

```
fun = @(x)1+x(1)/(1+x(2)) - 3*x(1)*x(2) + x(2)*(1+x(1));
```

Kısıtlı Doğrusal Olmayan Optimizasyon - fmincon - Sınırlı Kısıtlamalarla Küçültün

- x'in pozitif değerlere sahip olduğu, $x(1) \leq 1$ ve $x(2) \leq 2$ olduğu bölgeye bakın.

```
lb = [0,0];
ub = [1,2];
```

- Problemin doğrusal kısıtlaması yoktur, bu nedenle bu argümanları [ ] olarak ayarlayın.

```
A = [];
b = [];
Aeq = [];
beq = [];
```

Kısıtlı Doğrusal Olmayan Optimizasyon - fmincon - Sınırlı Kısıtlamalarla Küçültün

- Bölgenin ortasında bir başlangıç noktası deneyin.

```
x0 = (lb + ub)/2;
```

- Problemi çöz.

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

Çıktısı :

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    1.0000    2.0000
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Sınırlı Kısıtlamalarla Küçültün

- Farklı bir başlangıç noktası farklı bir çözüme yol açabilir.

```
x0 = x0/5;
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
```

Çıktısı :

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

   1.0e-06 *

    0.4000    0.4000
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Olmayan Kısıtlamalar

- `x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)` minimizasyonu doğrusal olmayan c(x) eşitsizliklerine veya nonlcon'da tanımlanan ceq(x) eşitliklerine tabi tutar. `fmincon, c(x) ≤ 0` ve `ceq(x) = 0` olacak şekilde optimize eder. Sınır yoksa, `lb = []` ve/veya `ub = []` olarak ayarlayın.
- **Örnek :** Doğrusal olmayan kısıtlamalara tabi bir fonksiyonun minimumunu bulun. Rosenbrock'un fonksiyonunun bir daire içinde en aza indirildiği noktayı bulun ve aynı zamanda sınır kısıtlamalarına da tabi.

```
fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
```

$0 \le x(1) \le 0.5, 0.2 \le x(2) \le 0.8$ bölgesine bakın.

```
lb = [0,0.2];
ub = [0.5,0.8];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Olmayan Kısıtlamalar

- Ayrıca [1/3,1/3] merkezli ve yarıçapı 1/3 olan dairenin içine bakın. Matlab yolunuzda «circlecon.m» adlı bir dosya oluşturun ve aşağıdaki kodu oraya yazın.

```
function [c,ceq] = circlecon(x)
c = (x(1)-1/3)^2 + (x(2)-1/3)^2 - (1/3)^2;
ceq = [];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Olmayan Kısıtlamalar

◦ Doğrusal kısıtlama yoktur, bu nedenle bu argümanları [ ] olarak ayarlayın.

```
A = [];
b = [];
Aeq = [];
beq = [];
```

Tüm kısıtlamaları karşılayan bir başlangıç noktası seçin.

```
x0 = [1/4,1/4];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Doğrusal Olmayan Kısıtlamalar

- Problemi çöz.

```
nonlcon = @circlecon;
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
```

Çıktısı:

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    0.5000    0.2500
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Varsayılan Olmayan Seçenekler

- `x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)` options'da belirtilen optimizasyon seçenekleri ile minimize eder. Bu seçenekleri ayarlamak için `optimoptions`'ı kullanın. Doğrusal olmayan eşitsizlik veya eşitlik kısıtlamaları yoksa, `nonlcon = []` olarak ayarlayın.
- **Örnek:** Yinelemeleri gerçekleştikçe görüntülemek ve farklı bir algoritma kullanmak için seçenekleri ayarlayın. `fmincon` çözüm sürecini gözlemlemek için Display seçeneğini 'iter' olarak ayarlayın. Ayrıca, bazen varsayılan 'interior-point' algoritmasından daha hızlı veya daha doğru olan 'sqp' algoritmasını deneyin.

```
options = optimoptions('fmincon','Display','iter','Algorithm','sqp');
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Varsayılan Olmayan Seçenekler

- Birim diskdeki $||x||^2 \leq 1$ minimum Rosenbrock fonksiyonunu bulun. İlk önce doğrusal olmayan kısıtlamayı temsil eden bir işlev oluşturun. Bunun için Matlab yolunuz üzerinde «unitdisk.m» adlı bir dosya olarak oluşturup, aşağıdaki kodu yazın.

```
function [c,ceq] = unitdisk(x)
c = x(1)^2 + x(2)^2 - 1;
ceq = [];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Varsayılan Olmayan Seçenekler

- Kalan problem özelliklerini oluşturun. Ardından `fmincon`'u çalıştırın.

```
fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
A = [];
b = [];
Aeq = [];
beq = [];
lb = [];
ub = [];
nonlcon = @unitdisk;
x0 = [0,0];
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Varsayılan Olmayan Seçenekler

Çıktısı:

Iter	Func-count	Fval	Feasibility	Step Length	Norm of step	First-order optimality
0	3	1.000000e+00	0.000e+00	1.000e+00	0.000e+00	2.000e+00
1	12	8.913011e-01	0.000e+00	1.176e-01	2.353e-01	1.107e+01
2	22	8.047847e-01	0.000e+00	8.235e-02	1.900e-01	1.330e+01
3	28	4.197517e-01	0.000e+00	3.430e-01	1.217e-01	6.172e+00
4	31	2.733703e-01	0.000e+00	1.000e+00	5.254e-02	5.705e-01
5	34	2.397111e-01	0.000e+00	1.000e+00	7.498e-02	3.164e+00
6	37	2.036002e-01	0.000e+00	1.000e+00	5.960e-02	3.106e+00
7	40	1.164353e-01	0.000e+00	1.000e+00	1.459e-01	1.059e+00
8	43	1.161753e-01	0.000e+00	1.000e+00	1.754e-01	7.383e+00
9	46	5.901601e-02	0.000e+00	1.000e+00	1.547e-02	7.278e-01
10	49	4.533081e-02	2.898e-03	1.000e+00	5.393e-02	1.252e-01
11	52	4.567454e-02	2.225e-06	1.000e+00	1.492e-03	1.679e-03
12	55	4.567481e-02	4.424e-12	1.000e+00	2.095e-06	1.501e-05
13	58	4.567481e-02	0.000e+00	1.000e+00	2.212e-12	1.406e-05

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    0.7864    0.6177
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Problem Yapısı Kullanın

- x = fmincon(problem), problemde tanımlanan bir yapı olan problem için minimumu bulur.

Örnek: Ayrı bağımsız değişkenler yerine bir sorun yapısı kullanarak «Varsayılan Olmayan Seçeneklerdeki» ile aynı sorunu çözün. Seçenekler (options) ve bir sorun yapısı (problem structure) oluşturun. Alan adları ve gerekli alanlar için soruna bakın.

```
options = optimoptions('fmincon','Display','iter','Algorithm','sqp’);
problem.options = options;
problem.solver = 'fmincon’;
problem.objective = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
problem.x0 = [0,0];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Problem Yapısı Kullanın

- Doğrusal olmayan kısıtlama işlevi unitdisk, bu örneğin sonunda görünür. Doğrusal olmayan kısıtlama fonksiyonunu probleme dahil edin.

```
problem.nonlcon = @unitdisk;
```

- Problemi çöz.

```
x = fmincon(problem)
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Problem Yapısı Kullanın

Çıktısı :

```
 Iter  Func-count
         Fval     Feasibility   Step Length       Norm of    First-order
                                                                  step     optimality
    0           3    1.000000e+00     0.000e+00     1.000e+00     0.000e+00     2.000e+00
    1          12    8.913011e-01     0.000e+00     1.176e-01     2.353e-01     1.107e+01
    2          22    8.047847e-01     0.000e+00     8.235e-02     1.900e-01     1.330e+01
    3          28    4.197517e-01     0.000e+00     3.430e-01     1.217e-01     6.172e+00
    4          31    2.733703e-01     0.000e+00     1.000e+00     5.254e-02     5.705e-01
    5          34    2.397111e-01     0.000e+00     1.000e+00     7.498e-02     3.164e+00
    6          37    2.036002e-01     0.000e+00     1.000e+00     5.960e-02     3.106e+00
    7          40    1.164353e-01     0.000e+00     1.000e+00     1.459e-01     1.059e+00
    8          43    1.161753e-01     0.000e+00     1.000e+00     1.754e-01     7.383e+00
    9          46    5.901601e-02     0.000e+00     1.000e+00     1.547e-02     7.278e-01
   10          49    4.533081e-02     2.898e-03     1.000e+00     5.393e-02     1.252e-01
   11          52    4.567454e-02     2.225e-06     1.000e+00     1.492e-03     1.679e-03
   12          55    4.567481e-02     4.424e-12     1.000e+00     2.095e-06     1.501e-05
   13          58    4.567481e-02     0.000e+00     1.000e+00     2.212e-12     1.406e-05

Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    0.7864    0.6177
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Amaç Fonksiyon Değerini Elde Edin

- `[x,fval] = fmincon(___)`, herhangi bir sözdizimi için, x çözümünde amaç fonksiyonu fun'ın değerini döndürür.

Örnek: Çözümdeki amaç fonksiyonunun değerini elde etmek için fval çıktısıyla fmincon'u çağırın. «Sınırlı Kısıtlamalarla Küçült» örneği iki çözüm gösteriyor. Hangisi daha iyi bir çözümdür? Çözümün yanı sıra fval çıktısını isteyen örneği çalıştırın.

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Amaç Fonksiyon Değerini Elde Edin

```
fun = @(x)1+x(1)./(1+x(2)) - 3*x(1).*x(2) + x(2).*(1+x(1));
lb = [0,0];
ub = [1,2];
A = [];
b = [];
Aeq = [];
beq = [];
x0 = (lb + ub)/2;
[x,fval] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Amaç Fonksiyon Değerini Elde Edin

Çıktısı:

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    1.0000    2.0000


fval =

   -0.6667
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Amaç Fonksiyon Değerini Elde Edin

- Farklı bir başlangıç noktası x0 kullanarak sorunu çalıştırın.

```
x0 = x0/5;

[x2,fval2] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
```

Çıktısı:

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x2 =

   1.0e-06 *

    0.4000    0.4000


fval2 =

    1.0000
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Ekstra Çıktılar Kullanarak Çözümü İnceleyin

- `[x,fval,exitflag,output] = fmincon(___)` ayrıca fmincon'un çıkış koşulunu açıklayan bir çıkış bayrağı değeri ve optimizasyon süreci hakkında bilgi içeren bir yapı çıktısı döndürür.

Örnek: Bir çözümün kalitesini kolayca incelemek için `exitflag` ve `output` çıktılarını isteyin.

Birim diskinde Rosenbrock'un işlevini en aza indirme problemi $||x||^2$'yi kurun. İlk önce doğrusal olmayan kısıtlamayı temsil eden bir işlev oluşturun. Bunu Matlab yolunuz üzerinde «unitdisk.m» adlı bir dosya olarak kaydedin.

```
function [c,ceq] = unitdisk(x)
c = x(1)^2 + x(2)^2 - 1;
ceq = [];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Ekstra Çıktılar Kullanarak Çözümü İnceleyin

- Kalan sorun belirtimlerini oluşturun.

```
fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
nonlcon = @unitdisk;
A = [];
b = [];
Aeq = [];
beq = [];
lb = [];
ub = [];
x0 = [0,0];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Ekstra Çıktılar Kullanarak Çözümü İnceleyin

- `fval`, `exitflag` ve `output` çıkışlarını kullanarak `fmincon`'u arayın.

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

Çıktısı :

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint toleranc

<stopping criteria details>

x =

    0.7864    0.6177


fval =

    0.0457


exitflag =

     1


output =

  struct with fields:

         iterations: 24
          funcCount: 84
    constrviolation: 0
           stepsize: 6.9162e-06
          algorithm: 'interior-point'
      firstorderopt: 2.4373e-08
       cgiterations: 4
            message: 'Local minimum found that satisfies the constraints.↵↵O
       bestfeasible: [1×1 struct]
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Tüm Çıktıları Al

`[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(___)` ayrıca şunu döndürür:

- lambda — x çözümünde Lagrange çarpanlarını içeren alanları olan yapı.
- grad — x çözümünde fun gradyanı.
- hessian — x çözümünde fun'ın Hessian'ı.

`fmincon,` isteğe bağlı olarak, bildirilen çözümü analiz etmek için kullanabileceğiniz birkaç çıktı döndürür.

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Tüm Çıktıları Al

Birim diskinde Rosenbrock'un işlevini en aza indirme problemini kurun. İlk önce doğrusal olmayan kısıtlamayı temsil eden bir işlev oluşturun. Bunu Matlab yolunuz üzerinde «unitdisk.m» adlı bir dosya olarak kaydedin.

```
function [c,ceq] = unitdisk(x)
c = x(1)^2 + x(2)^2 - 1;
ceq = [];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Tüm Çıktıları Al

Kalan sorun özelliklerini oluşturun.

```
fun = @(x)100*(x(2)-x(1)^2)^2 + (1-x(1))^2;
nonlcon = @unitdisk;
A = [];
b = [];
Aeq = [];
beq = [];
lb = [];
ub = [];
x0 = [0,0];
```

Kısıtlı Doğrusal Olmayan Optimizasyon – fmincon - Tüm Çıktıları Al

Tüm `fmincon` çıktılarını isteyin.

```
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
```

Çıktısı :

```
Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    0.7864    0.6177


fval =

    0.0457


exitflag =

     1


output =

  struct with fields:

         iterations: 24
          funcCount: 84
    constrviolation: 0
           stepsize: 6.9162e-06
          algorithm: 'interior-point'
      firstorderopt: 2.4373e-08
       cgiterations: 4
            message: 'Local minimum found that satisfies the constraints.↵↵Opt
       bestfeasible: [1×1 struct]


lambda =

  struct with fields:

         eqlin: [0×1 double]
      eqnonlin: [0×1 double]
       ineqlin: [0×1 double]
         lower: [2×1 double]
         upper: [2×1 double]
    ineqnonlin: 0.1215


grad =

   -0.1911
   -0.1501


hessian =

  497.2903 -314.5589
 -314.5589  200.2392
```