

6.12 CONJUGATE GRADIENT (FLETCHER-REEVES) METHOD

383

Öyleyse,

$$X_4 = X_3 + X_3 * S_3 = \frac{CO.S)}{C1.2J} \cdot 1.0 \cdot \frac{f-0,2')}{ben 0,2J} \cdot \frac{CIO)}{C1,4J}$$

X4'teki gradyan şu şekilde verilir:

$$\frac{f-0,20)}{[-0,20J} V / < "$$

Vf4 = $\nabla f(X_4)$ olduğundan, X4 optimum değildir ve dolayısıyla bir sonraki aşamaya geçmemiz gerekir.

yineleme. Bu işleme optimum nokta olan $X^* = f-1.0)$ kadar devam edilmelidir. - , J . - U bulundu.

(1.5)

Yakınsama Kriterleri. Yinelemeli süreci sonlandırmak için aşağıdaki kriterler kullanılabilir:

1. Ardışık iki yinelemede fonksiyon değerindeki değişiklik küçük olduğunda:

$$\left| \frac{f(X_i) - f(X_{i-1})}{f(X_i)} \right| \leq \epsilon_1 \quad (6.71)$$

2./'nin kısmi türevleri (gradyanın bileşenleri) küçük olduğunda:

$$\left| \frac{\partial f}{\partial x_i} \right| < C_2, \quad \text{ben} = 1, 2, \dots, A_i \quad (6.72)$$

3. Ardışık iki yinelemede tasarım vektöründeki değişiklik küçük olduğunda:

$$\|X_{i+1} - X_i\| < \epsilon_3 \quad (6.73)$$

6.12 BİRLEŞİK GRADYAN (FLETCHER-REEVES) YÖNTEMİ

En dik iniş yönteminin yakınsama özellikleri, eşlenik gradyan yöntemine dönüştürülerek büyük ölçüde geliştirilebilir (bu, fonksiyonun gradyanının kullanımını içeren eşlenik yönler yöntemi olarak düşünülebilir). Eşlenik yönleri kullanan herhangi bir minimizasyon yönteminin ikinci dereceden yakınsak olduğunu (Bölüm 6.7'de) gördük. İkinci dereceden yakınsamanın bu özelliği çok kullanışlıdır çünkü yöntemin ikinci dereceden bir fonksiyonu n veya daha az adımda en aza indirmesini sağlar. Herhangi bir genel fonksiyona optimum noktaya yakın ikinci dereceden bir değerle oldukça iyi bir şekilde yaklaşılabildiğinden, ikinci dereceden yakınsak herhangi bir yöntemin sonlu sayıda yinelemede optimum noktayı bulması beklenir.

6.12d Fletcher-Reeves Yönteminin Geliştirilmesi

Powell'ın eşlenik yön yönteminin yineleme başına n adet tek değişkenli minimizasyon gerektirdiğini ve her yinelemenin sonunda yeni bir eşlenik yön oluşturduğunu gördük. Dolayısıyla ikinci dereceden bir fonksiyonun minimumunu bulmak için genel olarak n^2 tek değişkenli minimizasyon gerekir. Öte yandan amaç fonksiyonunun gradyanlarını değerlendirebilirsek, her tek boyutlu minimizasyondan sonra yeni bir eşlenik yön oluşturabilir ve dolayısıyla daha hızlı yakınsama sağlayabiliriz. Eşlenik yönlerin oluşturulması ve Fletcher-Reeves yönteminin geliştirilmesi bu bölümde tartışılmaktadır.

6.12d Fletcher-Reeves Yönteminin Geliştirilmesi İkinci dereceden bir fonksiyona $f(X) = \frac{1}{2}X^TAX + B^T X + C$ uygulanan en dik iniş yöntemini, ardışık yönlerin karşılıklı olarak eşlenik olması koşulunu uygulayarak değiştirerek bir algoritma geliştirmeyi düşünün. Minimizasyon için başlangıç noktası X_1 olsun ve ilk arama yönü en dik iniş yönü olsun:

$$S_1 = -\nabla f_1 = -AX_1 - B \quad (6.74)$$

$$X_2 = X_1 + \alpha_1 S_1 \quad (6.75)$$

veya

$$S_1 = -\nabla f_1 \quad (6.76)$$

burada α_1 , S_1 yönündeki minimum adım uzunluğudur, böylece

$$\nabla f(X_2) = 0 \quad (6.77)$$

Denklem (6.77) şu şekilde genişletilebilir:

$$\nabla f(X_1 + \alpha_1 S_1) = 0 \quad (6.78)$$

buradan α_1 'nin değeri şu şekilde bulunabilir:

$$\alpha_1 = \frac{-\nabla f(X_1)^T S_1}{\nabla f(X_1)^T A \nabla f(X_1)} \quad (6.79)$$

Şimdi ikinci arama yönünü S_1 ve $-\nabla f_2$ 'nin doğrusal birleşimi olarak ifade edin:

$$S_2 = -\nabla f_2 + \beta_2 S_1 \quad (6.80)$$

burada (32, S_1 ve S_2 'nin konjuge olmasını sağlayacak şekilde seçilecektir. Bu şunu gerektirir:

We have seen that Powell's conjugate direction method requires n^2 single-variable minimizations per iteration and sets up a new conjugate direction at the end of each iteration. Thus it requires, in general, n^2 single-variable minimizations to find the minimum of a quadratic function. On the other hand, if we can evaluate the gradients of the objective function, we can set up a new conjugate direction after every one-dimensional minimization, and hence we can achieve faster convergence. The construction of conjugate directions and development of the Fletcher–Reeves method are discussed in this section.

6.12.1 Development of the Fletcher–Reeves Method

Consider the development of an algorithm by modifying the steepest descent method applied to a quadratic function $f(\mathbf{X}) = \frac{1}{2}\mathbf{X}^T\mathbf{A}\mathbf{X} + \mathbf{B}^T\mathbf{X} + C$ by imposing the condition that the successive directions be mutually conjugate. Let \mathbf{X}_1 be the starting point for the minimization and let the first search direction be the steepest descent direction:

$$\mathbf{S}_1 = -\nabla f_1 = -\mathbf{A}\mathbf{X}_1 - \mathbf{B} \quad (6.74)$$

$$\mathbf{X}_2 = \mathbf{X}_1 + \lambda_1^* \mathbf{S}_1 \quad (6.75)$$

or

$$\mathbf{S}_1 = \frac{\mathbf{X}_2 - \mathbf{X}_1}{\lambda_1^*} \quad (6.76)$$

where λ_1^* is the minimizing step length in the direction \mathbf{S}_1 , so that

$$\mathbf{S}_1^T \nabla f|_{\mathbf{X}_2} = 0 \quad (6.77)$$

Equation (6.77) can be expanded as

$$\mathbf{S}_1^T [\mathbf{A}(\mathbf{X}_1 + \lambda_1^* \mathbf{S}_1) + \mathbf{B}] = 0 \quad (6.78)$$

from which the value of λ_1^* can be found as

$$\lambda_1^* = \frac{-\mathbf{S}_1^T (\mathbf{A}\mathbf{X}_1 + \mathbf{B})}{\mathbf{S}_1^T \mathbf{A} \mathbf{S}_1} = -\frac{\mathbf{S}_1^T \nabla f_1}{\mathbf{S}_1^T \mathbf{A} \mathbf{S}_1} \quad (6.79)$$

Now express the second search direction as a linear combination of \mathbf{S}_1 and $-\nabla f_2$:

$$\mathbf{S}_2 = -\nabla f_2 + \beta_2 \mathbf{S}_1 \quad (6.80)$$

where β_2 is to be chosen so as to make \mathbf{S}_1 and \mathbf{S}_2 conjugate. This requires that

$$\mathbf{S}_1^T \mathbf{A} \mathbf{S}_2 = 0 \quad (6.81)$$

Denklemin Değiştirilmesi (6.80) Denk. (6.81) şuna yol açar

$$S[A(-Vf_2 + j82S_1)] = 0 \quad (6.82)$$

Denklem (6.76) ve (6.82) verimi

$$-(X' \sim X_1) A (Vf_2 - P_2 S_1) = 0 \quad (6.83)$$

Gradyanların farkı ($Vf_2 - Vf$) şu şekilde ifade edilebilir:

$$(V/2 - Vf) = (AX_2 + B) - (AX_1 + B) = A(X_2 - X_1) \quad (6.84)$$

Denklem yardımıyla. (6.84), Denk. (6.83) şu şekilde yazılabilir:

$$(V/2 - Vf)Y(Vf_2 - P_1 S_x) = 0 \quad (6.85)$$

burada A matrisinin simetrisi kullanılmıştır. Denklem (6.85) şu şekilde genişletilebilir:

$$V/2r Vf_2 - Vf] Vf_2 - p_2 Vf_2 S_1 + P_2 Vfx S_1 = 0 \quad (6.86)$$

$Vf Vf_2 = -S]Vf_2$ olduğundan $= 0$ Denklemden. (6.77), Denk. (6.86) verir

$$^2 = _ YfIVt = YI^A \quad (6.87)$$

Daha sonra üçüncü arama yönünü S_1 , S_2 ve $-Vf_3$ 'ün doğrusal birleşimi olarak ele alıyoruz:

$$S_3 = -V/3 + \&S_2 + S_3S_i \quad (6.88)$$

burada P_3 ve S_3 'ün değerleri S_3 'ün S_1 ve S_2 'ye konjuge edilmesiyle bulunabilir. $S[AS_3 = 0$ koşulunu kullanarak S_3 'ün değerinin sıfır olduğu bulunabilir (bkz. Problem 6.40). $S_2AS_3 = 0$ koşulu kullanıldığında P_3 değeri şu şekilde elde edilebilir (bkz. Problem 6.41)

böylece Denklem. (6.88) olur

THE CONJUGATE GRADIENT (FLETCHER-REEVES) METHOD

385

Substituting Eq. (6.80) into $\mathbf{L}_1^T \mathbf{S}_3 = -\nabla f_3 + (\mathbf{S}_3^T \mathbf{S}_2)$ (6.90)

$$\mathbf{S}_1^T \mathbf{A} (-\nabla f_2 + \beta_2 \mathbf{S}_1) = 0 \quad (6.82)$$

Equations (6.76) and (6.82) yield

$$-\frac{(\mathbf{X}_2 - \mathbf{X}_1)^T}{\lambda_1^*} \mathbf{A} (\nabla f_2 - \beta_2 \mathbf{S}_1) = 0 \quad (6.83)$$

The difference of the gradients $(\nabla f_2 - \nabla f_1)$ can be expressed as

$$(\nabla f_2 - \nabla f_1) = (\mathbf{A}\mathbf{X}_2 + \mathbf{B}) - (\mathbf{A}\mathbf{X}_1 + \mathbf{B}) = \mathbf{A}(\mathbf{X}_2 - \mathbf{X}_1) \quad (6.84)$$

With the help of Eq. (6.84), Eq. (6.83) can be written as

$$(\nabla f_2 - \nabla f_1)^T (\nabla f_2 - \beta_2 \mathbf{S}_1) = 0 \quad (6.85)$$

where the symmetricity of the matrix \mathbf{A} has been used. Equation (6.85) can be expanded as

$$\nabla f_2^T \nabla f_2 - \nabla f_1^T \nabla f_2 - \beta_2 \nabla f_2^T \mathbf{S}_1 + \beta_2 \nabla f_1^T \mathbf{S}_1 = 0 \quad (6.86)$$

Since $\nabla f_1^T \nabla f_2 = -\mathbf{S}_1^T \nabla f_2 = 0$ from Eq. (6.77), Eq. (6.86) gives

$$\beta_2 = -\frac{\nabla f_2^T \nabla f_2}{\nabla f_1^T \mathbf{S}_1} = \frac{\nabla f_2^T \nabla f_2}{\nabla f_1^T \nabla f_1} \quad (6.87)$$

Next we consider the third search direction as a linear combination of \mathbf{S}_1 , \mathbf{S}_2 , and $-\nabla f_3$ as

$$\mathbf{S}_3 = -\nabla f_3 + \beta_3 \mathbf{S}_2 + \delta_3 \mathbf{S}_1 \quad (6.88)$$

where the values of β_3 and δ_3 can be found by making \mathbf{S}_3 conjugate to \mathbf{S}_1 and \mathbf{S}_2 . By using the condition $\mathbf{S}_1^T \mathbf{A} \mathbf{S}_3 = 0$, the value of δ_3 can be found to be zero (see Problem 6.40). When the condition $\mathbf{S}_2^T \mathbf{A} \mathbf{S}_3 = 0$ is used, the value of β_3 can be obtained as (see Problem 6.41)

$$\beta_3 = \frac{\nabla f_3^T \nabla f_3}{\nabla f_2^T \nabla f_2} \quad (6.89)$$

so that Eq. (6.88) becomes

$$\mathbf{S}_3 = -\nabla f_3 + \beta_3 \mathbf{S}_2 \quad (6.90)$$

burada j33 Denklem tarafından verilmektedir. (6.89). Aslında Denk. (6.90) şu şekilde genelleştirilebilir

$$\mathbf{S}_1 = -\nabla f_1 + \beta_1 \mathbf{S}_0, \quad (6.91)$$

Neresi

$$\mathbf{S}_i = -\nabla f_i + \beta_i \mathbf{S}_{i-1}, \quad (6.92)$$

Denklemler (6.91) ve (6.92), Fletcher-Reeves yönteminde [6.13] kullanılan arama yönlerini tanımlar.

6.12.2 Fletcher-Reeves Yöntemi Fletcher-Reeves yönteminin yinelemeli prosedürü şu şekilde ifade edilebilir:

1. Rastgele bir \mathbf{X}_1 başlangıç noktasıyla başlayın. 2. İlk arama yönünü $\mathbf{S}_1 = -\nabla f(\mathbf{X}_1) = -\nabla f_1$ olarak ayarlayın. 3. İlişkiye göre \mathbf{X}_2 noktasını bulun

$$\mathbf{X}_2 = \mathbf{X}_1 + \alpha_1 \mathbf{S}_1$$

burada α_1 , \mathbf{S}_1 yönündeki optimal adım uzunluğudur. $i = 2$ olarak ayarlayın ve bir sonraki adıma geçin.

4. $\nabla f = \nabla f(\mathbf{X}_i)$ 'yi bulun ve ayarlayın

$$\mathbf{S}_i = -\nabla f_i + \beta_i \mathbf{S}_{i-1} \quad (6.93)$$

5. \mathbf{S}_i yönünde optimum adım uzunluğu α_i 'i hesaplayın ve yeni nokta

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \alpha_i \mathbf{S}_i \quad (6.94)$$

6. \mathbf{X}_{i+1} noktasının optimalliğini test edin. \mathbf{X}_{i+1} optimum ise, durdurun işlem. Aksi halde $i = i + 1$ değerini ayarlayın ve 4. adıma geçin.

Notlar:

1. Fletcher-Reeves yöntemi ilk olarak Hestenes ve Stiefel [6.14] tarafından ikinci dereceden bir denklemin durağan koşullarından türetilen doğrusal denklem sistemlerini çözmek için bir yöntem olarak önerilmiştir. Bu yöntemde kullanılan S_1 yönleri A-eşleniği olduğundan, ikinci dereceden bir fonksiyon için sürecin n veya daha az döngüde yakınsaması gerekir. Bununla birlikte, kötü koşullandırılmış wh (konturları oldukça eksantrik ve çarpık olan) ikinci dereceden ifadeler için yöntem,

$$S_i = -\nabla f_i + \beta_i S_{i-1} \quad (6.91)$$

where

$$\beta_i = \frac{\nabla f_i^T \nabla f_i}{\nabla f_{i-1}^T \nabla f_{i-1}} \quad (6.92)$$

Equations (6.91) and (6.92) define the search directions used in the Fletcher-Reeves method [6.13].

6.12.2 Fletcher-Reeves Method

The iterative procedure of Fletcher-Reeves method can be stated as follows:

1. Start with an arbitrary initial point X_1 .
2. Set the first search direction $S_1 = -\nabla f(X_1) = -\nabla f_1$.
3. Find the point X_2 according to the relation

$$X_2 = X_1 + \lambda_1^* S_1$$

where λ_1^* is the optimal step length in the direction S_1 . Set $i = 2$ and go to the next step.

4. Find $\nabla f_i = \nabla f(X_i)$, and set

$$S_i = -\nabla f_i + \frac{|\nabla f_i|^2}{|\nabla f_{i-1}|^2} S_{i-1} \quad (6.93)$$

5. Compute the optimum step length λ_i^* in the direction S_i , and find the new point

$$X_{i+1} = X_i + \lambda_i^* S_i \quad (6.94)$$

6. Test for the optimality of the point X_{i+1} . If X_{i+1} is optimum, stop the process. Otherwise, set the value of $i = i + 1$ and go to step 4.

Remarks:

1. The Fletcher-Reeves method was originally proposed by Hestenes and Stiefel [6.14] as a method for solving systems of linear equations derived from the stationary conditions of a quadratic. Since the directions S_i used in this method are A-conjugate, the process should converge in n cycles

or less for a quadratic function. However, for ill-conditioned quadratics (whose contours are highly eccentric and distorted), the method may

yakınsama için n'den fazla döngüye ihtiyaç vardır. Bunun nedeninin yuvarlama hatalarının kümülatif etkisi olduğu tespit edilmiştir. S1 Denklem tarafından verildiğinden. (6.93), X_f 'nin belirlenmesinde yer alan yanlışlıklardan ve ardışık $|V|^{1/2} S / |V/J| - 1/2$ terimlerinin toplanmasında yer alan yuvarlama hatasından kaynaklanan herhangi bir hata ileriye taşınır. S1- vektörü aracılığıyla. Böylece S arama yönleri bu hatalar nedeniyle giderek kirlenecektir. Bu nedenle, pratikte, yeni arama yönünü en dik iniş yönü olarak alarak, örneğin her m adımdan sonra yöntemi periyodik olarak yeniden başlatmak gerekir. Yani, her m adımdan sonra $S_m + X$, alışlagelmiş form yerine $-V/m + x$ 'e eşitlenir. Fletcher ve Reeves $m = n + 1$ değerini önermişlerdir; burada n, tasarım değişkenlerinin sayısıdır.

2. Yukarıda belirtilen sınırlamalara rağmen, Fletcher-Reeves yöntemi en dik iniş yönteminden ve desen arama yöntemlerinden çok daha üstündür, ancak tartışılan Newton ve yarı-Newton (değişken metrik) yöntemlerden oldukça daha az verimli olduğu ortaya çıkmıştır. son bölümlerde.

Örnek 6.10 Minimize edin $f(X) = X_1^2 - X_2^2 + 2X_1 + X_1X_2 + X_2^2$ başlangıç

$X_1 = 0$ noktasından f.

ÇÖZÜM
Yineleme 1

$$\left(\frac{df}{dx^i} \right) = F_i = 2X_1 + 4X_2 + 2X_1^2 + 2X_1X_2 + 2X_2^2$$

$$VZ_1 = VZ(X_1) = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Arama yönü $S_1 = -VZ_1 = \begin{bmatrix} -2 \\ -4 \end{bmatrix}$ (En uygun adımı bulmak için)

S_1 boyunca X_f uzunluğunu küçültüyoruz $(X_1 + X_1S_1)$ X_1 'e göre. Burada

$$f(X_1 + X_1S_1) = f(-X_1, -4X_1) = -2X_1^2$$

$$\frac{df}{dx} = 0 \text{ da } X_f = 1$$

require much more than n cycles for convergence. The reason for this has been found to be the cumulative effect of rounding errors. Since \mathbf{S}_i is given by Eq. (6.93), any error resulting from the inaccuracies involved in the determination of λ_i^* , and from the round-off error involved in accumulating the successive $|\nabla f_i|^2 \mathbf{S}_{i-1} / |\nabla f_{i-1}|^2$ terms, is carried forward through the vector \mathbf{S}_i . Thus the search directions \mathbf{S}_i will be progressively contaminated by these errors. Hence it is necessary, in practice, to restart the method periodically after every, say, m steps by taking the new search direction as the steepest descent direction. That is, after every m steps, \mathbf{S}_{m+1} is set equal to $-\nabla f_{m+1}$ instead of the usual form. Fletcher and Reeves have recommended a value of $m = n + 1$, where n is the number of design variables.

2. Despite the limitations indicated above, the Fletcher-Reeves method is vastly superior to the steepest descent method and the pattern search methods, but it turns out to be rather less efficient than the Newton and the quasi-Newton (variable metric) methods discussed in the latter sections.

Example 6.10 Minimize $f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$ starting from the point $\mathbf{X}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$.

SOLUTION

Iteration 1

$$\nabla f = \begin{Bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \end{Bmatrix} = \begin{Bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{Bmatrix}$$

$$\nabla f_1 = \nabla f(\mathbf{X}_1) = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}$$

The search direction is taken as $\mathbf{S}_1 = -\nabla f_1 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$. To find the optimal step

length λ_1^* along \mathbf{S}_1 , we minimize $f(\mathbf{X}_1 + \lambda_1 \mathbf{S}_1)$ with respect to λ_1 . Here

$$f(\mathbf{X}_1 + \lambda_1 \mathbf{S}_1) = f(-\lambda_1, +\lambda_1) = \lambda_1^2 - 2\lambda_1$$

$$\frac{df}{d\lambda_1} = 0 \quad \text{at} \quad \lambda_1^* = 1$$

Therefore,

$$\mathbf{X}_2 = \mathbf{X}_1 + \lambda_1^* \mathbf{S}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} + 1 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

Yineleme 2

$V_f(X_2) = \frac{1}{2} (X_2^2 - 1)^2$, Denklem (6.93) bir sonraki arama yönünü şu şekilde verir:

Neresi

$$|Y_A|_2 = 2 \text{ ve } |V/2|_2 = 2$$

Öyleyse,

$$-\frac{1}{2} (X_2^2 - 1)^2$$

X^* bulmak için simge durumuna küçültüyoruz

$$\begin{aligned} f(X_2) &= \frac{1}{2} (X_2^2 - 1)^2 \\ &= \frac{1}{2} (X_2^4 - 2X_2^2 + 1) \\ &= \frac{1}{2} X_2^4 - X_2^2 + \frac{1}{2} \end{aligned}$$

X_2 'ye göre. $f'(X_2)$ olarak $f'(X_2) = 2X_2^3 - 2X_2 = 0$, $X^* = \pm 1$ de, şunu elde ederiz:

Böylece iki iterasyonda optimum noktaya ulaşılır. Bu noktanın optimum olduğunu bilmesek bile bir sonraki iterasyonda bu noktadan hareket edemeyiz. Bu aşağıdaki şekilde doğrulanabilir.

Yineleme 3 Şimdi

$$V_f = V_f(X_3) = \frac{1}{2} (X_3^2 - 1)^2, \quad |V/2|_2 = 2, \quad \text{ve} \quad |V/3|_2 = 0.$$

Böylece

388 NONLINEAR PROGRAMMING II: UNCONSTRAINED OPTIMIZATION TECHNIQUES

Bu, azaltılacak / daha da artırılacak bir arama yönünün olmadığını ve dolayısıyla X_3 'ün optimum olduğunu gösterir.

Since $\nabla f_2 = \nabla f(X_2) = \begin{Bmatrix} 0 \\ -1 \end{Bmatrix}$, Eq. (6.93) gives the next search direction as

$$S_2 = -\nabla f_2 + \frac{|\nabla f_2|^2}{|\nabla f_1|^2} S_1$$

where

$$|\nabla f_1|^2 = 2 \quad \text{and} \quad |\nabla f_2|^2 = 2$$

Therefore,

$$S_2 = -\begin{Bmatrix} -1 \\ -1 \end{Bmatrix} + \left(\frac{2}{2}\right) \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ +2 \end{Bmatrix}$$

To find λ_2^* , we minimize

$$\begin{aligned} f(X_2 + \lambda_2 S_2) &= f(-1, 1 + 2\lambda_2) \\ &= -1 - (1 + 2\lambda_2) + 2 - 2(1 + 2\lambda_2) + (1 + 2\lambda_2)^2 \\ &= 4\lambda_2^2 - 2\lambda_2 - 1 \end{aligned}$$

with respect to λ_2 . As $df/d\lambda_2 = 8\lambda_2 - 2 = 0$ at $\lambda_2^* = \frac{1}{4}$, we obtain

$$X_3 = X_2 + \lambda_2^* S_2 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} + \frac{1}{4} \begin{Bmatrix} 0 \\ 2 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1.5 \end{Bmatrix}$$

Thus the optimum point is reached in two iterations. Even if we do not know this point to be optimum, we will not be able to move from this point in the next iteration. This can be verified as follows.

Iteration 3

Now

$$\nabla f_3 = \nabla f(X_3) = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}, \quad |\nabla f_2|^2 = 2, \quad \text{and} \quad |\nabla f_3|^2 = 0.$$

Thus

$$S_3 = -\nabla f_3 + (|\nabla f_3|^2/|\nabla f_2|^2)S_2 = -\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} + \left(\frac{0}{2}\right) \begin{Bmatrix} 0 \\ 2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

This shows that there is no search direction to reduce f further, and hence \mathbf{X}_3 is optimum.

6.13 NEWTON'UN YÖNTEMİ

Newton'un Bölüm 5.12.1'de sunulan yöntemi çok değişkenli fonksiyonların minimizasyonu için genişletilebilir. Bunun için Taylor serisi açılımını kullanarak $\mathbf{X} = \mathbf{X}^1 + \Delta \mathbf{X}$ fonksiyonunun ikinci dereceden yaklaşımını düşünün.

$$f(\mathbf{X}) \approx f(\mathbf{X}^1) + \nabla f(\mathbf{X}^1)^T (\mathbf{X} - \mathbf{X}^1) + \frac{1}{2} (\mathbf{X} - \mathbf{X}^1)^T \mathbf{J}(\mathbf{X}^1) (\mathbf{X} - \mathbf{X}^1) \quad (6.95)$$

burada $\mathbf{J}(\mathbf{X}^1) = [\mathbf{J}]_{1 \times n}$, \mathbf{X}^1 - noktasında değerlendirilen ikinci kısmi türevlerin (Hessian matrisi) matrisidir. Denklem kısmi türevlerini ayarlayarak (6.95) minimumu için sıfıra eşit $\nabla f(\mathbf{X})$, elde ederiz

$$\frac{\partial f}{\partial x_j}(\mathbf{X}^1) + \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{X}^1) (x_i - x_i^1) = 0, \quad j = 1, 2, \dots, n \quad (6.96)$$

DJ

Denklemler (6.96) ve (6.95) şunu verir:

$$\nabla f(\mathbf{X}^1) + \mathbf{J}(\mathbf{X}^1) (\mathbf{X} - \mathbf{X}^1) = 0 \quad (6.97)$$

Eğer $\mathbf{J}(\mathbf{X}^1)$ tekil değilse Denklem. (6.97) geliştirilmiş bir yaklaşım elde etmek için $\mathbf{X} = \mathbf{X}^1 + \Delta \mathbf{X}$ şu şekilde çözülebilir:

$$\mathbf{X}^1 + \Delta \mathbf{X} = \mathbf{X}^1 - [\mathbf{J}(\mathbf{X}^1)]^{-1} \nabla f(\mathbf{X}^1) \quad (6.98)$$

Denklemden yüksek mertebeden terimler ihmal edildiği için. (6.95), Denk. (6.98) optimum \mathbf{X}^* çözümünü bulmak için yinelemeli olarak kullanılacaktır.

$\mathbf{X}^1, \mathbf{X}^2, \dots$ noktalarının sırası. $\dots, \mathbf{X}^1 + \Delta \mathbf{X}$ 'in, $[\mathbf{J}(\mathbf{X}^1)]$ 'in tekil olmaması koşuluyla, \mathbf{X}^* çözümüne yeterince yakın herhangi bir \mathbf{X}^1 başlangıç noktasından gerçek \mathbf{X}^* çözümüne yaklaştığı gösterilebilir. Newton'un yönteminin amaç fonksiyonunun ikinci kısmi türevlerini ($[\mathbf{J}(\mathbf{X}^1)]$ matrisi biçiminde) kullandığı ve dolayısıyla ikinci dereceden bir yöntem olduğu görülebilir.

Örnek 6.11 Newton yönteminin ikinci dereceden bir fonksiyonun minimumunu bir yinelemede bulduğunu gösterin.

ÇÖZÜM İkinci dereceden fonksiyon şu şekilde verilsin:

$$f(\mathbf{X}) = \mathbf{X}^T \mathbf{A} \mathbf{X} + \mathbf{B}^T \mathbf{X} + C$$

Minimum $f(\mathbf{X})$ şu şekilde verilir:

6.13 NEWTON'S METHOD

$$\nabla f = [A]X + B = 0$$

389

veya

6.13 NEWTON'S METHOD

$$X^* = -[A]^{-1}B$$

Newton's method presented in Section 5.12.1 can be extended for the minimization of multivariable functions. For this, consider the quadratic approximation of the function $f(X)$ at $X = X_i$ using the Taylor's series expansion

$$f(X) = f(X_i) + \nabla f_i^T (X - X_i) + \frac{1}{2} (X - X_i)^T [J_i] (X - X_i) \quad (6.95)$$

where $[J_i] = [J]|_{X_i}$ is the matrix of second partial derivatives (Hessian matrix) of f evaluated at the point X_i . By setting the partial derivatives of Eq. (6.95) equal to zero for the minimum of $f(X)$, we obtain

$$\frac{\partial f(X)}{\partial x_j} = 0, \quad j = 1, 2, \dots, n \quad (6.96)$$

Equations (6.96) and (6.95) give

$$\nabla f = \nabla f_i + [J_i](X - X_i) = 0 \quad (6.97)$$

If $[J_i]$ is nonsingular, Eqs. (6.97) can be solved to obtain an improved approximation ($X = X_{i+1}$) as

$$X_{i+1} = X_i - [J_i]^{-1} \nabla f_i \quad (6.98)$$

Since higher-order terms have been neglected in Eq. (6.95), Eq. (6.98) is to be used iteratively to find the optimum solution X^* .

The sequence of points X_1, X_2, \dots, X_{i+1} can be shown to converge to the actual solution X^* from any initial point X_1 sufficiently close to the solution X^* , provided that $[J_1]$ is nonsingular. It can be seen that Newton's method uses the second partial derivatives of the objective function (in the form of the matrix $[J_i]$) and hence is a second-order method.

Example 6.11 Show that the Newton's method finds the minimum of a quadratic function in one iteration.

SOLUTION Let the quadratic function be given by

$$f(X) = \frac{1}{2} X^T [A] X + B^T X + C$$

The minimum of $f(X)$ is given by

$$\nabla f = [A]X + B = 0$$

$$\mathbf{X}^* = -[\mathbf{A}]^{-1}\mathbf{B}$$

Denklemin yinelemeli adımı. (6.98) verir

$$\mathbf{X}_{k+1} = \mathbf{X}_k - [\mathbf{H}(\mathbf{X}_k)]^{-1} \mathbf{F}(\mathbf{X}_k) \quad (\text{E1})$$

burada \mathbf{X}_k , k 'inci yinelemenin başlangıç noktasıdır. Böylece Denk. (E1) kesin çözümü verir

$$\mathbf{X}_{k+1} = \mathbf{X}^* = -\mathbf{M}^{-1} \mathbf{B}$$

Şekil 6.17 bu süreci göstermektedir.

Örnek 6.12 Minimize $f(\mathbf{x}) = x_1 - x_2 + 2x_1^2 + x_1x_2 + x_2^2$ olarak

başlangıç noktası $\mathbf{x}_0 = \mathbf{J}_n$ 'dir.

ÇÖZÜM

Eşitliğe göre \mathbf{x}_2 'yi bulmak için. (6.98), $[\mathbf{H}(\mathbf{x}_1)]^{-1}$ 'e ihtiyacımız var, burada

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 4x_1 + x_2 & x_1 + 2x_2 \\ x_1 + 2x_2 & 2x_2 + 1 \end{bmatrix}$$

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 4x_1 + x_2 & x_1 + 2x_2 \\ x_1 + 2x_2 & 2x_2 + 1 \end{bmatrix}$$

Öyleyse,

390 NONLINEAR PROGRAMMING II: UNCONSTRAINED OPTIMIZATION TECHNIQUES

The iterative step in Eq. (6.98) is the Newton-Raphson method. Şekil 6.17 İkinci dereceden bir fonksiyonun tek adımda minimizasyonu.

$$\mathbf{X}_{i+1} = \mathbf{X}_i - [\mathbf{A}]^{-1}([\mathbf{A}]\mathbf{X}_i + \mathbf{B}) \quad (\text{E}_1)$$

where \mathbf{X}_i is the starting point for the i th iteration. Thus Eq. (E₁) gives the exact solution

$$\mathbf{X}_{i+1} = \mathbf{X}^* = -[\mathbf{A}]^{-1}\mathbf{B}$$

Figure 6.17 illustrates this process.

Example 6.12 Minimize $f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$ by taking the starting point as $\mathbf{X}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$.

SOLUTION To find \mathbf{X}_2 according to Eq. (6.98), we require $[\mathbf{J}_1]^{-1}$, where

$$[\mathbf{J}_1] = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}_{\mathbf{X}_1} = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

Therefore,

$$[\mathbf{J}_1]^{-1} = \frac{1}{4} \begin{bmatrix} +2 & -2 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix}$$

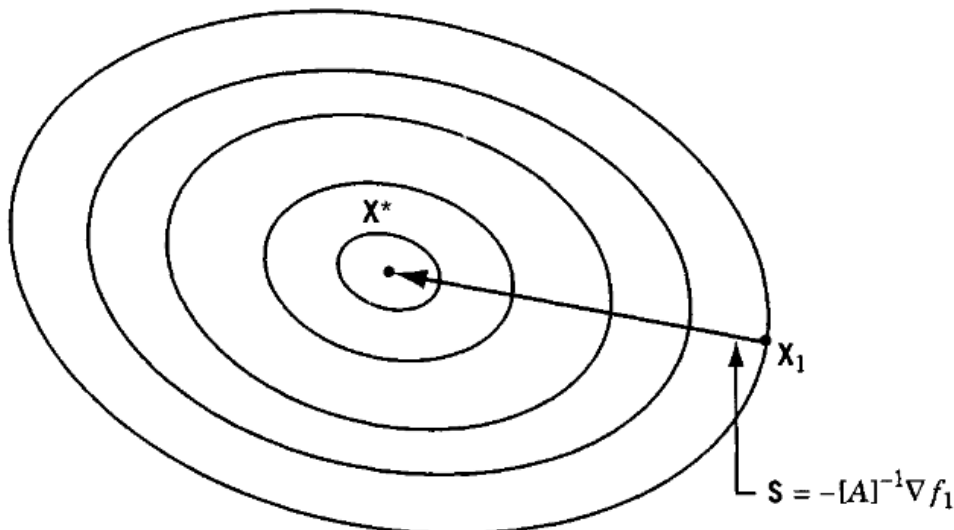


Figure 6.17 Minimization of a quadratic function in one step.

Gibi

$$g_1 \sim [df/dx_2]X_i \sim [-1 + 2x, + 3x_2] \quad (OO) \sim [-1]$$

Denklem (6.98) şunu verir

$$— \quad ' \quad - \quad [\quad : \quad) \quad - \quad | \quad \text{ÇİND} \quad \text{Ben} \quad] \quad P \quad \wedge \quad r \quad 1 \quad)$$

X_2 'nin optimum nokta olup olmadığını görmek için şunu değerlendiririz:

$$= Cdf/dxt') = r \quad \text{ben} + 4JC, + 2X_2) \quad _CQT)$$

$$g_2 \quad Uf/axJ^* \sim 1 - 1 + 2xt + 2x_2 \setminus U^{3/2}) \sim (ah$$

$g_2 = 0$ olduğundan X_2 optimum noktadır. Böylece yöntem bu ikinci dereceden fonksiyon için tek bir yinelemede birleşti.

Eğer/(X) ikinci dereceden olmayan bir fonksiyon ise, Newton'un yöntemi bazen ıraksayabilir ve eyer noktalarına ve bağlı maksimumlara yakınsabilir. Denklem değiştirilerek bu sorun önlenabilir. (6.98) olarak

$$X/ + 1 = X_1 + XfS_1 = X_1 - Xf[J_i V_1 \quad \text{Kablosuz} \quad (6.99)$$

burada $\setminus f, S_1 = -[Z,]^{1/2} V/j$ yönündeki minimum adım uzunluğudur. Denklem (6.99) ile gösterilen değişikliğin bir takım avantajları vardır. İlk olarak, minimumu daha küçük bir değerde bulacaktır. Orijinal yöntemle göre adım sayısı daha fazladır. İkincisi, her durumda minimum noktayı bulurken, bazı durumlarda orijinal yöntem yakınsamayabilir. Üçüncüsü, tüm bu avantajlarla genellikle bir eyer noktasına veya maksimuma yaklaşmayı önler. Bu yöntem Bu avantajlara rağmen, yöntemin aşağıdaki özelliklerinden dolayı pratikte pek kullanışlı olmadığı görülmektedir:

1. $n \times n$ matrisinin $[J]$.
2. Matrisin $[J_1]$ elemanlarını hesaplamak çok zor ve bazen imkansız hale gelir.
3. Her adımda $[J_1]$ matrisinin ters çevrilmesini gerektirir.

4. Her adımda [JJ"1 V/- miktarının değeriendirilmesini gerektirir.

6.13 NEWTON'S METHOD

391

Bu özellikler, yöntemi çok sayıda değişken içeren karmaşık amaç fonksiyonu içeren problemler için kullanışsız hale getirir.

$$\mathbf{g}_1 = \left\{ \frac{\partial f}{\partial x_1} \right\}_{\mathbf{X}_1} = \left\{ \begin{matrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 3x_2 \end{matrix} \right\}_{(0,0)} = \left\{ \begin{matrix} 1 \\ -1 \end{matrix} \right\}$$

Equation (6.98) gives

$$\mathbf{X}_2 = \mathbf{X}_1 - [\mathbf{J}_1]^{-1} \mathbf{g}_1 = \left\{ \begin{matrix} 0 \\ 0 \end{matrix} \right\} - \left[\begin{matrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{matrix} \right] \left\{ \begin{matrix} 1 \\ -1 \end{matrix} \right\} = \left\{ \begin{matrix} -1 \\ \frac{3}{2} \end{matrix} \right\}$$

To see whether or not \mathbf{X}_2 is the optimum point, we evaluate

$$\mathbf{g}_2 = \left\{ \frac{\partial f}{\partial x_1} \right\}_{\mathbf{X}_2} = \left\{ \begin{matrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{matrix} \right\}_{(-1, 3/2)} = \left\{ \begin{matrix} 0 \\ 0 \end{matrix} \right\}$$

As $\mathbf{g}_2 = \mathbf{0}$, \mathbf{X}_2 is the optimum point. Thus the method has converged in one iteration for this quadratic function.

If $f(\mathbf{X})$ is a nonquadratic function, Newton's method may sometimes diverge, and it may converge to saddle points and relative maxima. This problem can be avoided by modifying Eq. (6.98) as

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^* \mathbf{S}_i = \mathbf{X}_i - \lambda_i^* [\mathbf{J}_i]^{-1} \nabla f_i \quad (6.99)$$

where λ_i^* is the minimizing step length in the direction $\mathbf{S}_i = -[\mathbf{J}_i]^{-1} \nabla f_i$. The modification indicated by Eq. (6.99) has a number of advantages. First, it will find the minimum in lesser number of steps compared to the original method. Second, it finds the minimum point in all cases, whereas the original method may not converge in some cases. Third, it usually avoids convergence to a saddle point or a maximum. With all these advantages, this method appears to be the most powerful minimization method. Despite these advantages, the method is not very useful in practice, due to the following features of the method:

1. It requires the storing of the $n \times n$ matrix $[\mathbf{J}_i]$.
2. It becomes very difficult and sometimes, impossible to compute the elements of the matrix $[\mathbf{J}_i]$.
3. It requires the inversion of the matrix $[\mathbf{J}_i]$ at each step.
4. It requires the evaluation of the quantity $[\mathbf{J}_i]^{-1} \nabla f_i$ at each step.

These features make the method impractical for problems involving a compli-

cated objective function with a large number of variables.

6.14 MARQUARDT YÖNTEMİ

En dik iniş yöntemi, tasarım vektörü X_1 -, optimum X^* noktasından uzakta olduğunda fonksiyon değerini azaltır. Newton yöntemi ise X tasarım vektörü optimum X^* noktasına yakın olduğunda hızlı yakınsar. Marquardt yöntemi [6.15] hem en dik iniş hem de Newton yöntemlerinden yararlanmaya çalışır. Bu yöntem, Hessian matrisinin $[J_1]$ köşegen elemanlarını şu şekilde değiştirir:

$$U_i = [J_i] + C_i I_i \quad (6.100)$$

burada $[J_i]$ bir birim matristir ve O_{11} , $[J_1]$ pozitif tanımlı olmadığında $[J_i]$ 'nin pozitif tanımlılığını sağlayan pozitif bir sabittir. α yeterince büyük olduğunda (104 civarında), $\alpha [I_i]$ teriminin $[J_i]$ baskın olduğu ve matrisin tersinin $[J_1]$ olduğu not edilebilir.

$$U_i^{-1} = [J_i]^{-1} + \alpha^{-1} [I_i]^{-1} \quad (6.101)$$

Böylece arama yönü S_1 - şu şekilde hesaplanırsa

$$S_1 = - [U_1]^{-1} \nabla f_1 \quad (6.102)$$

S_1 - büyük α , değerleri için en dik iniş yönü haline gelir. Marquardt yönteminde α değeri başlangıçta büyük alınır ve iteratif süreç ilerledikçe kademeli olarak sıfıra düşürülür. Böylece, α değeri büyük bir değerden sıfıra düştükçe, arama yönteminin özellikleri en dik iniş yöntemininkinden Newton yönteminin özelliklerine doğru değişir. Marquardt yönteminin değiştirilmiş bir versiyonunun yinelemeli süreci aşağıdaki gibi açıklanabilir.

1. Rasgele bir başlangıç noktası X_1 ve sabitler (X_1 (104 düzeyinde), c_1 ($0 < C_1 < 1$), C_2 ($c_2 > 1$) ve ϵ (10^{-2} düzeyinde) ile başlayın. yineleme numarası $k = 1$ 'dir.
2. Fonksiyonun gradyanını hesaplayın, $\nabla f_1 = \nabla f(X_1)$.
3. X_1 - noktasının optimalliğini test edin. Eğer $\|\nabla f\| = \|\nabla f(X_1)\| < \epsilon$, X_1 optimumdur ve dolayısıyla süreci durdurun. Aksi halde 4. adıma gidin.
4. Yeni X_{k+1} vektörünü şu şekilde bulun:

$$X/ + 1 = X1- + S1- = X1- - [[J1.] + CK1-[Z]]^{-1} \nabla f \quad (6.103)$$

3925./-+1 ve / değerlerini karşılaştırm. İff+i 'AIN< f, go to step 6. I f ^ + i 'UE^
Z, 7. adıma gidin.

6.14 MARQUARDT METHOD

The steepest descent method reduces the function value when the design vector \mathbf{X}_i is away from the optimum point \mathbf{X}^* . The Newton method, on the other hand, converges fast when the design vector \mathbf{X}_i is close to the optimum point \mathbf{X}^* . The Marquardt method [6.15] attempts to take advantage of both the steepest descent and Newton methods. This method modifies the diagonal elements of the Hessian matrix, $[J_i]$, as

$$[\tilde{J}_i] = [J_i] + \alpha_i [I] \quad (6.100)$$

where $[I]$ is an identity matrix and α_i is a positive constant that ensures the positive definiteness of $[\tilde{J}_i]$ when $[J_i]$ is not positive definite. It can be noted that when α_i is sufficiently large (on the order of 10^4), the term $\alpha_i [I]$ dominates $[J_i]$ and the inverse of the matrix $[\tilde{J}_i]$ becomes

$$[\tilde{J}_i]^{-1} = [[J_i] + \alpha_i [I]]^{-1} \approx [\alpha_i [I]]^{-1} = \frac{1}{\alpha_i} [I] \quad (6.101)$$

Thus if the search direction \mathbf{S}_i is computed as

$$\mathbf{S}_i = -[\tilde{J}_i]^{-1} \nabla f_i \quad (6.102)$$

\mathbf{S}_i becomes a steepest descent direction for large values of α_i . In the Marquardt method, the value of α_i is taken to be large at the beginning and then reduced to zero gradually as the iterative process progresses. Thus, as the value of α_i decreases from a large value to zero, the characteristics of the search method change from those of a steepest descent method to those of the Newton method. The iterative process of a modified version of Marquardt method can be described as follows.

1. Start with an arbitrary initial point \mathbf{X}_1 and constants α_1 (on the order of 10^4), c_1 ($0 < c_1 < 1$), c_2 ($c_2 > 1$), and ϵ (on the order of 10^{-2}). Set the iteration number as $i = 1$.
2. Compute the gradient of the function, $\nabla f_i = \nabla f(\mathbf{X}_i)$.
3. Test for optimality of the point \mathbf{X}_i . If $\|\nabla f_i\| = \|\nabla f(\mathbf{X}_i)\| \leq \epsilon$, \mathbf{X}_i is optimum and hence stop the process. Otherwise, go to step 4.
4. Find the new vector \mathbf{X}_{i+1} as

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{S}_i = \mathbf{X}_i - [[J_i] + \alpha_i [I]]^{-1} \nabla f_i \quad (6.103)$$

5. Compare the values of f_{i+1} and f_i . If $f_{i+1} < f_i$, go to step 6. If $f_{i+1} \geq f_i$, go to step 7.

