T.C.

BALIKESİR ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

NEWTON-RAPHSON YÖNTEMİ

MERTCAN AYDEMİR

Doç.Dr. Beyza Billur İskender Eroğlu

BALIKESİR, 04 – 2025

Newton Yöntemi (1)

Bölüm 10.1'deki Örnek 2'deki problem, üç değişkenli (x1, x2, ve x3) üç denklemin cebirsel olarak çözülmesiyle yakınsak bir sabit nokta problemini çözmeye dönüştürülmüştür. Ancak, tüm değişkenler için açık bir temsil bulmak genellikle zordur. Bu bölümde, daha genel bir durumda dönüşümü gerçekleştirmek için algoritmik bir prosedür ele alıyoruz.

Tek boyutlu durumda uygun bir sabit nokta yöntemi elde eden algoritmayı oluşturmak için, şu özelliğe sahip bir φ\varphiφ fonksiyonu bulduk:

$$g(x)=x-\varphi(x)f(x)$$

Bu yöntem, g fonksiyonunun sabit noktası p'ye kuadratik yakınsama sağlar.
Bu koşuldan hareketle, Newton yöntemi φ(x)=1\ f′(x) seçilerek geliştirilmiştir; burada f′(x)≠ 0 olduğu varsayılmıştır.
Benzer bir yaklaşım, n-boyutlu durumda bir matris kullanılarak gerçekleştirilir."

$$A(\mathbf{x}) = \begin{bmatrix} a_{11}(\mathbf{x}) & a_{12}(\mathbf{x}) & \cdots & a_{1n}(\mathbf{x}) \\ a_{21}(\mathbf{x}) & a_{22}(\mathbf{x}) & \cdots & a_{2n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ a_{n1}(\mathbf{x}) & a_{n2}(\mathbf{x}) & \cdots & a_{nn}(\mathbf{x}) \end{bmatrix},$$

Burada her bir $a_{ij}(x)$ elemanı, R^n 'den R'ye bir fonksiyondur. Bu durum, A(x) matrisinin uygun şekilde bulunmasını gerektirir.

$$G(x) = x - A(x)\ {-1}\ F(x)$$

Bu yöntem, G fonksiyonunun sabit noktası p üzerinde F(x)=0 probleminin çözümüne kuadratik yakınsama sağlar; burada A(x) matrisinin tekil olmaması (yani terslenebilir olması) varsayılmaktadır.
Aşağıdaki teorem, 80. sayfadaki Teorem 2.8 ile paralellik göstermektedir.
Bu teoremin ispatı, G'yi p noktası etrafında n değişkenli Taylor serisi cinsinden ifade edebilme yeteneğini gerektirir.
p, G(x)= x denkleminin bir çözümü olsun. δ>0 olacak şekilde bir sayı mevcut olsun.

⇨ $\partial x_j/\partial g_i$ türevi, her i=1,2,…,n ve j=1,2,…,n için $N\delta=\{x \mid \|x-p\|<\delta\}$ kümesinde süreklidir.

⇨ $\partial^2 g_i(x) / \partial x_j \, \partial x_k$ ikinci mertebeden türevleri süreklidir ve
$x \in N\delta_x$ kümesi için, her i=1,2,…,n j=1,2,…,n ve k=1,2,…,n için
$| \partial^2 g_i(x) / \partial x_j \, \partial x_k | \leq M$ olacak şekilde bir sabit M vardır.

⇨ $\partial g_i(p) / \partial x_k=0$, her i=1,2,…,n ve k=1,2,…,n için geçerlidir.

$$\|\mathbf{x}^{(k)} - \mathbf{p}\|_\infty \leq \frac{n^2 M}{2} \|\mathbf{x}^{(k-1)} - \mathbf{p}\|_\infty^2, \quad \text{for each } k \geq 1.$$

Teorem 10.7'yi uygulamak için, A(x)'in, Eq. (10.5)'te verilen formda, R^n 'den R'ye giden fonksiyonlardan oluşan bir n × n matris olduğunu varsayalım;
burada matrisin belirli elemanları daha sonra seçilecektir.
Ayrıca, A(x) ' in F(x) = 0 denkleminin bir çözümü olan p noktasının yakınında tekil olmadığını (yani terslenebilir olduğunu) varsayalım ve $b_{ij}(x)$ ifadesiyle $A(x)^{-1}$ matrisinin i. satır ve j. sütunundaki elemanı gösterelim.

$$\frac{\partial g_i}{\partial x_k}(\mathbf{x}) = \begin{cases} 1 - \sum_{j=1}^{n} \left(b_{ij}(\mathbf{x}) \frac{\partial f_j}{\partial x_k}(\mathbf{x}) + \frac{\partial b_{ij}}{\partial x_k}(\mathbf{x}) f_j(\mathbf{x}) \right), & \text{if } i = k, \\ -\sum_{j=1}^{n} \left(b_{ij}(\mathbf{x}) \frac{\partial f_j}{\partial x_k}(\mathbf{x}) + \frac{\partial b_{ij}}{\partial x_k}(\mathbf{x}) f_j(\mathbf{x}) \right), & \text{if } i \neq k. \end{cases}$$

Teorem 10.7, her i=1,2,…,n ve k=1,2,…,n için $\partial g_i(p) / \partial x_k = 0$ olması gerektiğini ifade eder. Bu da, i=k olduğu durumda şunu ifade eder.

$$0 = 1 - \sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_i}(\mathbf{p}),$$

that is,

$$\sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_i}(\mathbf{p}) = 1.$$

When $k \neq i$,

$$0 = -\sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_k}(\mathbf{p}),$$

so

$$\sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_k}(\mathbf{p}) = 0.$$

NEWTON-RAPHSON (2)

Belki de tüm kök bulma formülleri arasında en yaygın kullanılan yöntem Newton-Raphson yöntemidir . Eğer kök için başlangıç tahmini x_i ise, [x_i , f(x_i)] noktasından bir teğet çizilebilir. Bu teğetin x-ekseniyle kesiştiği nokta, genellikle kökün daha iyi bir tahminini verir. Newton-Raphson yöntemi, bu geometrik yorum temel alınarak türetilir.Şekil 6.4'te gösterildiği gibi, x noktasındaki birinci türev eğime eşittir

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

bu ifade yeniden düzenlenerek elde edilebilir

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Bu, Newton-Raphson formülü olarak adlandırılır.

KAYNAKÇALAR

(1) Numerical Analysis (Richard L. Burden ve J. Douglas Faires)

(2) Applied Numerical Methods (Steven C. Chapra)

Benim yaptığım araştırmalarımdan ve bulduğum kaynakçalardan çıkardığım bilgilere göre Newton- Raphson yönteminin bir x_0 başlangıç noktasının olduğu ve bu noktadan verilen f fonksiyonuna bir teğet doğrusu çizililir. Bu çizilen teğetin x eksenini kestiği noktayla yaklaşık kökün bulunması amaçlanır. Newton-Raphson yönteminin iterasyon dizisini veren denklem aşadaki gibi olur .

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

n sayısı 0 dan başlayarak x_0 başlangıç noktasına kadar gider.Newton-Raphson yönteminin hata analizini yaparken kullandığımız formül aşağıda verilmiştir.

$$|e_{n+1}| \leq \frac{|f''(c)|}{2|f'(x_n)|} |e_n|^2$$

Newton-Raphson yönteminin hata analizinin de fonksiyonun türevi sıfır olma durumu karşımıza çıkarsa oluşan durumda yöntem çalışmaz. Bu olumsuz durumdan kurtulmak için kullanıllan başka bir yöntem vardır. Bu yönteme Sekant Yöntemi adı verilir. Bu yöntem ile oluşan formül aşağıda verilmiştir.

$$X_{n+1} = x_n – f(x_n) * (x_n - x_{n-1}) / <[f(x_n) – f(x_{n-1})]$$

Bu förmülde n değeri için n = 1,2,… olur.

ÖRNEK: **$f(x)=x^3+x^2-8x-12=0$ denkleminin katlı köklerini başlangıç noktasını $x_0 = 0$ alarak 5 adımda hesaplayınız.**

ÇÖZÜM:

$f(x_0) = -12$ $\qquad$ $x_{n+1}= x_n- f(x_n)/ f'(x_n)$ $\qquad$ $n=0,1,2,3,4$

$f'(x) =3x^2 + 2x -8$ $\quad$ => $\quad$ $f'(x_0) = -8$

1. İterasyon:

$X_1=x_0- f(x_0)/ f'(x_0)$ => -1.5

$E_1=|x_1-x_0|= |-1.5-0|=1.5$

$f(x_1) = -1.125$ $\qquad$ $f'(x_1) = - 4.25$

2. İterasyon:

$X_2= x_1 - f(x_1)/f'(x_1) = - 1.7647$

$E_2=|x_2-x_1|= |- 1.7647- (-1.5) | = 0.2647$

$f(x_2) = -0.2638$ $\qquad$ $f'(x_2) = -2.1922$

3. İterasyon:

$X_3= x_2 - f(x_2)/f'(x_2) = -1.8850$

$E_3=|x_3-x_2|= |-1.8850 -(-1.7647) |= 0.1203$

$f(x_3) = -0.0646$ $\qquad$ $f'(x_3) =-1.1103$

4. İterasyon:

$X_4= x_3 - f(x_3)/f'(x_3) = -1.9432$

$E_4=|x4-x3|= 0.0582$

$f(x_4) = -0.0159$ $\qquad$ $f'(x_4) = -0.5583$

5. İterasyon:

$X_5 = x_4 - f(x_4)/f'(x_4) = -1.9717$

$E_5 = |x_5 - x_4| = 0.0285$

MATLAB KODU

% Newton-Raphson yöntemi ile katlı kök bulma + Hata hesabı + Grafik hesaplayan program Matlab da yazdığım kod:

```
clc;        clear;          close all;

% Fonksiyon ve türevi tanımlanıyor

f = @(x) x.^3 + x.^2 - 8.*x - 12;

df = @(x) 3.*x.^2 + 2.*x - 8;

% Başlangıç noktası

x = 0;

% Adım sayısı

N = 5;

% Hata vektörü

hatalar = NaN(1, N);

% Başlık yazdır

fprintf('Adım\t x\t\t\t Hata\n');

fprintf('%d\t %.6f\t  -\n', 0, x);

% Her adımda sonucu ve hatayı yazdır
```

```
for k = 1:N

x_old = x;

x = x - f(x)/df(x);

hata = abs(x - x_old);

hatalar(k) = hata;

fprintf('%d\t %.6f\t %.6f\n', k, x, hata);

end

% Grafik çizimi

adimlar = 1:N;

figure;

plot(adimlar, hatalar, '-o', 'LineWidth', 2, 'MarkerSize', 8);

grid on;

xlabel('Adım Sayısı');

ylabel('Hata (|x_{n+1} - x_n|)')

title('Newton-Raphson Yöntemi Hata Grafiği');
```