

Bölüm 6

Nonlineer cebirsel denklemler

Uygulamalı bilimlerde çok sayıda problem, cebirsel bir fonksiyonun veya sistemin sıfır yeri veya sıfıryerlerinin bulunmasını gerektirir. Bu bölümde

- tek değişkenli ve reel değerli fonksiyonların sıfır yerlerini belirlemek amacıyla kullanılan sabit nokta iterasyon yöntemi,
- özel bir sabit nokta iterasyon yöntemi olan *Newton* yöntemi ve
- *Newton* yöntemindeki türeve sonlu fark yaklaşımı ile elde edilen giriş yöntemini inceliyoruz. Ayrıca
- nonlineer cebirsel sistemler için *Newton* yöntemi ve *Newton* benzeri bir yöntem,
- aralık üzerindeki tüm sıfıryerlerini belirlemek amacıyla ikiye bölme yöntemi ve
- r yarıçaplı disk içerisindeki tüm reel ve kompleks kökleri bulacak biçimde *Newton* yönteminin vektörel versiyonunu geliştirerek inceliyoruz.

Bu döküman "MATLAB/Octave Uygulamalarıyla Sayısal Analize Giriş" isimli çalışmamızın altıncı bölümünü oluşturmaktadır. Konuyla ilgili olarak zaman zaman faydalandığımız ve bölüm sonunda sunduğumuz temel kaynakları öneririz.

Öncelikle sıfır yeri belirlenmesini gerektiren bazı problemlere göz atalım.

6.1 Pratik sıfır yeri problemleri

Fonksiyon sıfır yerlerinin ve özellikle de polinom sıfır yerlerinin(köklerinin) belirlenmesi problemi matematik başta olmak üzere bir çok alanda güncelliğini koruyan problemidir.

Özel olarak polinom sıfır yerlerinin belirlenebilmesi için çok sayıda çalışma gerçekleştirilmiş, derecesi üç ve dört olan polinomlar için *Cardano*¹ formülleri adı verilen ve pratik olarak kullanılamayacak kadar karmaşık formüller geliştirilmiştir. Ancak daha yüksek dereceli polinomların sıfır yerleri için herhangi bir formül geliştirilemeyeceği *Abel* ve Galois(Galoa okunur)'nın çalışmaları ile gösterilmiştir.

Sıfır yeri belirleme problemi değişik alanlarda sıkça karşılaşılan bir problemidir:

- Bir f fonksiyonunun grafiğinin x eksenini kesim noktaları, ekstremum noktaları veya büküm noktalarının belirlenmesi, matematiksel açıdan olduğu kadar ilgili fonksiyonun temsil ettiği pratik değerlerin analizi açısından da oldukça önemlidir. Bu bilgiler yardımıyla verilen fonksiyonun veya temsil ettiği pratik değerlerin belirli bir aralıktaki davranışını tahmin edebiliriz.
- Ekonomide bir ürünün x adetinin üretimi sonucunda oluşan ve $C(x)$ ile gösterilen maliyet ile ürünün satışından elde edilen $R(x)$ gelir fonksiyonları yardımıyla tanımlanan $f(x) = R(x) - C(x)$ fonksiyonunun sıfır yeri, gelirin maliyeti karşıladığı üretim miktarını verir ki bu değer belirlenmesi oldukça önemlidir.
- $dx/dt = f(x)$ diferensiyel denkleminin denge noktaları(eğer mevcutsa) f fonksiyonunun reel sıfır yerleridir.

•

$$\begin{aligned} dx/dt &= f(x, y) \\ dy/dt &= g(x, y) \end{aligned}$$

sisteminin denge noktaları

$$\begin{aligned} f(x, y) &= 0 \\ g(x, y) &= 0 \end{aligned}$$

¹Geralamo Cardano(1501-1576) İtalyan matematikçi, fizikçi, astrolog ve hekim.

sisteminin reel çözüm kümesidir. En genel halde

$$dX/dt = F(X), X = (X_1, X_2, \dots, X_n)^T, F = (F_1, F_2, \dots, F_n)^T$$

sisteminin denge noktaları

$$F(X) = 0$$

nonlineer cebirsel sisteminin reel çözüm kümesidir.

- Diferensiyel denklemler için özdeğer problemleri de sıfır yeri belirleme problemleri arasında yer alırlar. Örneğin

$$\begin{aligned} y'' + \lambda y &= 0 \\ y(0) &= 0, y(1) + y'(1) = 0 \end{aligned}$$

denkleminin sıfırdan farklı çözüme sahip olduğu ve problemin özdeğerleri olarak bilinen λ değerleri, $f(\lambda) = \tan(\lambda) + \lambda$ fonksiyonunun sıfır yerleridir ve bu sıfır yerleri ancak sayısal yöntemler yardımıyla belirlenebilirler[3].

Bu örnekler daha da artırılabilir. Öncelikle f fonksiyonunun sıfır yeri için sabit nokta iterasyon yöntemini inceleyelim.

6.2 Sabit nokta iterasyon yöntemi

TANIM 6.1. *Bir problemin çözümünde $n + 1$ inci adımda elde edilen yaklaşım, n ve/veya daha önceki adımlarda elde edilen yaklaşımları kullanıyorsa, bu tür yöntemlere iterasyon yöntemleri veya iteratif yöntemler adı verilir.*

$$g : [a, b] \rightarrow [a, b]$$

sürekli bir fonksiyon olmak üzere g nin $[a, b]$ aralığında en az bir sıfır yeri vardır.

İspat. [Alıştırma 1].

TANIM 6.2. $x_0 \in [a, b]$ keyfi bir başlangıç noktası olsun.

$$x_{n+1} = g(x_n), n = 0, 1, 2, \dots \quad (6.1)$$

ile tanımlanan $\{x_n\}_{n=0}^{\infty}$ dizisine g fonksiyonu ve x_0 başlangıç noktası ile üretilen iterasyon dizisi adı verilir.

Önerme 6.1. *Eğer $\{x_n\}_{n=0}^{\infty}$ dizisi bir r noktasına yakınsıyorsa ve g fonksiyonu r noktasında sürekli ise bu taktirde r noktası g nin sabit noktasıdır.*

İspat.

$$r = \lim_{n \rightarrow \infty} x_{n+1} = \lim_{n \rightarrow \infty} g(x_n) = g(\lim_{n \rightarrow \infty} x_n) = g(r)$$

dir. (Yukarıdaki işlemlerde g fonksiyonunun sürekliliğini hangi adımda kullandık?)

Verilen bir f fonksiyonunun sıfır yerini bulmak için sıkça kullanılan yöntem, *sıfır yeri belirleme problemini sabit nokta belirleme problemine dönüştürmektir.* Böylece f nin sıfır yerini belirleme problemi uygun bir g fonksiyonu için g nin sabit noktasını belirleme problemine dönüştürülmüş olur, yani

$$f(x) = 0 \iff x = g(x)$$

dir. f fonksiyonunun sıfır yerini sabit nokta kabul eden çok sayıda g fonksiyonu bulunabilir. Ancak bu fonksiyonlardan bazıları f nin sıfır yeri için yakınsak iterasyon üretirken, diğer bir kısmı ise başlangıç noktası sıfır yerine ne kadar yakın seçilirse seçilsin ıraksak bir iterasyon üretebilir.

ÖRNEK 6.1. $f(x) = x^2 - x - 1$ fonksiyonun sıfır yerlerini belirleme probleminin aşağıda verilen g fonksiyonlarının sabit noktalarını belirleme problemine denk olduğunu gözlemleyerek, $x_0 = \sqrt{2}$ ile oluşturulan iterasyonların yakınsaklığını araştırınız.

- $g(x) = x^2 - 1$
- $g(x) = x + c(x^2 - x - 1), c \in R$ ve

$$c = -2/\sqrt{5}, -3/4, -1/\sqrt{5}$$

- $g(x) = \frac{x^2+1}{2x-1}$

Çözüm.

- Yukarıda verilen g fonksiyonlarının sabit noktalarının f nin sıfıyerleri olduğu kolayca görülebilir. Örneğin

$$g(x) = x^2 - 1 = x \Leftrightarrow f(x) = x^2 - 1 - x = 0$$

dir. f nin sıfıyerlerinden birisi altın oran olarak bilinen

$$r_1 = (1 + \sqrt{5}) / 2 \cong 1.6180$$

ve diğeri ise ,

$$r_2 = (1 - \sqrt{5}) / 2 \cong -0.6180$$

dir.

- $g(x) = x^2 - 1$ fonksiyonunu göz önüne alalım. Hiçbir x_0 ile (6.1) yardımıyla oluşturulan iterasyon $r_1 = (1 + \sqrt{5}) / 2$ sabit noktasına yakınsamaz: Örneğin r_1 e yakın seçilen $x_0 = \sqrt{2}$ için

$$\begin{aligned} x_1 &= g(x_0) = g(\sqrt{2}) = 1 \\ x_2 &= g(x_1) = g(1) = 0 \\ x_3 &= g(x_2) = g(0) = -1 \\ x_4 &= g(x_3) = g(-1) = 0 \end{aligned}$$

biçimde yakınsak olmayan salımlı bir dizi elde edilir. Daha da yakın komşulukta seçilen $x_0 = 1.6$ için de aynı salımlı dizi elde edilir. O halde $g(x) = x^2 - 1$ iterasyon fonksiyonu f nin r_1 sabit noktasını belirlemek için uygun değildir.

- $g(x) = x + c(x^2 - x - 1)$ iterasyon fonksiyonu $x_0 = \sqrt{2}$ ve

– $c = -\frac{2}{\sqrt{5}}$ değeri için belirli bir adımdan sonra 1.6880 ve 1.5437 değerlerini alternatif olarak almak suretiyle hiç bir noktaya yakınsamaz:

$$\begin{aligned}
x_1 &= g(\sqrt{2}) = 1.7847, \\
x_2 &= g(x_1) = 1.4265, \\
x_3 &= g(x_2) = 1.7767, \\
&\vdots \\
x_n &= g(x_{n-1}) = 1.6880, \\
x_{n+1} &= g(x_n) = 1.5437, \\
x_{n+2} &= g(x_{n+1}) = 1.6880, \dots
\end{aligned}$$

- $c = -3/4$ değeri için 57 adımda $|x_{n+1} - x_n| < \epsilon = 1E - 10$ kriterini sağlayan 1.61803398878648 noktasına yakınsar.
- $c = -1/\sqrt{5}$ değeri için 5 adımda 1.61803398874989 değerine yakınsar.

O halde c parametresinin değerine göre aynı başlangıç değeriyle oluşturulan iterasyon bazen ıraksak, bazen yavaş yakınsak veya veya bazen de hızlı yakınsak bir dizi oluşturabilmektedir.

- Öte yandan

$$g(x) = \frac{x^2 + 1}{2x - 1}$$

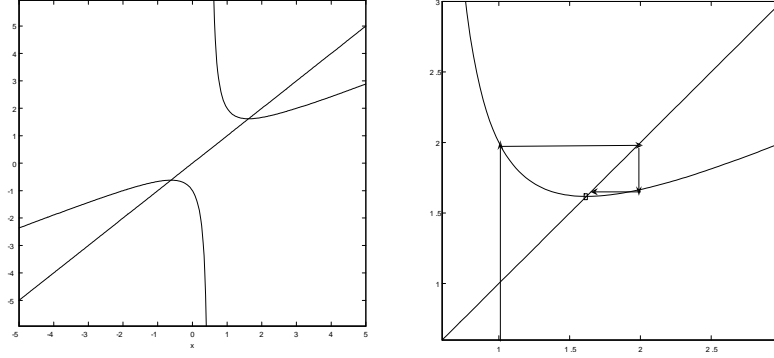
iterasyon fonksiyonu da $x_0 = \sqrt{2}$ için 5 adımda 1.61803398874989 değerine yakınsar.

Şekil 6.1(a) da bu g fonksiyonunu ve $h(x) = x$ ile tanımlanan h fonksiyonunun grafikleri sunulmaktadır.

- g fonksiyonunun biri pozitif ve diğeri negatif olan iki sabit noktası (g ve h fonksiyonlarının grafiklerinin kesim noktaların apsisleri) olduğu görülmektedir. Şekil 6.1(b) de $x_{n+1} = g(x_n)$ iterasyonu ile elde edilen x_n noktaları için $(x_n, g(x_n))$ çiftlerinin $(r_1, g(r_1))$ noktaya yakınsadığı görülmektedir.

Yukarıdaki iterasyonları aşağıda sunduğumuz Program 6.2 yardımıyla elde ettik.

Yukarıdaki örnekten, bazı g fonksiyonları ile oluşturulan iterasyonların ıraksadıkları, diğerlerinin ise farklı hızlarda yakınsadıklarını gözlemledik. Acaba



Şekil 6.1: (a) g ve h fonksiyonları, (b) $(x_i, g(x_i))$ nokta çiftleri

yakınsak iterasyon üreten iterasyon fonksiyonları belirli kriterler yardımıyla ayırt edebilir miyiz? Hatta yakınsak olanların yakınsama hızları için ne söyleyebiliriz? Bunun için aşağıdaki Teoremi inceleyelim:

TEOREM 6.1. $g : [a, b] \rightarrow [a, b]$ sürekli ve (a, b) aralığında türevlenebilir bir fonksiyon ve $r \in [a, b]$, g nin bu aralıktaki bir sabit noktası olsun. Ayrıca $\forall x \in (a, b)$ için

$$|g'(x)| \leq K < 1$$

özellikliğini sağlayan K sabitinin var olduğunu kabul edelim. Bu durumda $x_0 \in [a, b]$ olmak üzere

$$x_{n+1} = g(x_n), n = 0, 1, 2, \dots$$

ile tanımlanan iterasyon her $n \geq 0$ için

$$|x_n - r| \leq K^n |x_0 - r|$$

eşitsizliğini sağlar ve sonuç olarak bu iterasyonla üretilen $\{x_n\}_{n=0}^{\infty}$ dizisi r sabit noktasına yakınsar.

İspat.

Türevler için ortalama değer teoreminden

$$\begin{aligned} |x_1 - r| &= |g(x_0) - r| = |g(x_0) - g(r)| = |g'(c_0)| |x_0 - r| \\ &\leq K |x_0 - r|, c_0 \in (a, b) \end{aligned}$$

```

function x1=sabiter(g,x0)
-----
%Verilen g ve x0 ile sabit nokta iterasyon yöntem uygulaması, ec.
%Örneğin
%   >> g=@(x) (x^2+1)/(2*x-1);
%   >> sabiter(g,1);
%   ans = 1.6180
-----

min_deger=1e-10; max_deger=1e5;max_sayac=100;
test=1;sayac=0;
while test
    x1=g(x0);
    fark=abs(x1-x0);
    test=(fark>min_deger)&(abs(x1)<max_deger)&(sayac<max_sayac);
    x0=x1;
    sayac=sayac+1;
end
if abs(x1)>max_deger
    disp('iterasyon iraksaktır'); x1=[];
elseif sayac==max_sayac
    disp(['iterasyon ',num2str(sayac), 'adimda yakinsamamistir']);
    x1=[];
end

%-----

```

Program 6.1: Sabit nokta iterasyon uygulaması

elde ederiz. Yani, x_1 noktası r ye x_0 dan daha yakındır. Tümevarım gereği

$$|x_{n-1} - r| \leq K^{n-1}|x_0 - r|$$

olduğunu kabul edelim. Tekrar ortalama değer teoremi ve tümevarım hipotezi gereği

$$\begin{aligned}
 |x_n - r| &= |g(x_{n-1}) - r| = |g(x_{n-1}) - g(r)| = |g'(c_{n-1})||x_{n-1} - r| \\
 &\leq K|x_{n-1} - r| \\
 &\leq K^n|x_0 - r|, c_{n-1} \in (a, b)
 \end{aligned}$$

elde ederiz. Buradan $n \rightarrow \infty$ için limit almak suretiyle $0 < K < 1$ olduğundan $\lim_{n \rightarrow \infty} |x_n - r| = 0$ elde ederiz ve sıkıştırma teoremi yardımıyla $\lim_{n \rightarrow \infty} x_n = r$ olduğu görülür.

İterasyonun yakınsaması için bir diğer yeter şart aşağıdaki teoremle ifade edilmektedir:

Sonuç 6.1. $g : [a, b] \rightarrow [a, b]$ sürekli ve (a, b) aralığında türevlenebilir ve türevi sürekli bir fonksiyon ve $r \in (a, b)$, g nin bu aralıktaki bir sabit noktası ve ayrıca $|g'(r)| < 1$ olsun. Bu durumda r ye yeterince yakın $x_0 \in [a, b]$ için

$$x_{n+1} = g(x_n), n = 0, 1, 2, \dots$$

ile üretilen $\{x_n\}_{n=0}^{\infty}$ dizisi r sabit noktasına yakınsar.

İspat. $|g'(r)| < 1$ ve g' fonksiyonu $[a, b]$ de sürekli olduğundan

$$|g'(x)| \leq K < 1, \forall x \in (r - \delta, r + \delta)$$

sağlanacak biçimde $\delta > 0$ sabiti mevcuttur. $x_0 \in (r - \delta, r + \delta)$ ile Teorem 4.1' i uygulanarak sonuç elde edilir.

Uyarı. $|g'(r)| = 1$ olması durumunda üretilen iterasyon başlangıç nokta seçimine göre yakınsak veya iraksak olabilir (Alıştırma 7).

Sonuç 6.2. $g : [a, b] \rightarrow [a, b]$ sürekli ve (a, b) aralığında türevlenebilir ve türevi sürekli bir fonksiyon ve $r \in [a, b]$, g nin bu aralıktaki bir sabit noktası ve $|g'(r)| > 1$ olsun. Bu durumda $x_0 \in [a, b]$, $x_0 \neq r$ için

$$x_{n+1} = g(x_n), n = 0, 1, 2, \dots$$

ile üretilen $\{x_n\}_{n=0}^{\infty}$ dizisi r sabit noktasına yakınsamaz.

İspat. Alıştırma 8.

ÖRNEK 6.2. Teorem 6.1 yardımıyla Örnek 6.1 e ait sonuçları analiz ediniz.

Çözüm.

- $g(x) = x^2 - 1$ fonksiyonu ile üretilen dizi iraksadı çünkü

$$g'((1 + \sqrt{5})/2) = 1 + \sqrt{5} > 1$$

dir ve yukarıdaki sonuca göre hiç bir x_0 ile üretilen dizi yakınsamaz.

Aynı sonuç diğer sabit nokta için de geçerlidir, çünkü

$$|g'((1 - \sqrt{5})/2)| = |1 - \sqrt{5}| > 1$$

dir.

- $c = -\frac{2}{\sqrt{5}}$ için

$$g(x) = x - \frac{2}{\sqrt{5}}(x^2 - x - 1)$$

olup,

$$g'(x) = \frac{-4}{\sqrt{5}}x + \frac{2}{\sqrt{5}} + 1$$

ve

$$g'((1 + \sqrt{5})/2) = -1$$

elde edilir. İterasyon bu durumda ıraksamıştır. Aynı sonuç diğer sabit nokta için de geçerlidir, çünkü

$$g'((1 - \sqrt{5})/2) = 3 > 1$$

dir.

- $c = -\frac{3}{4}$ için

$$g(x) = x - \frac{3}{4}(x^2 - x - 1)$$

olup,

$$g'(x) = -\frac{3}{2}x + \frac{7}{4}$$

ve

$$|g'((1 + \sqrt{5})/2)| = |-0.677| < 1$$

elde edilir ve bu durumda iterasyonun pozitif sabit noktası için yakınsamış olması beklenen bir sonuçtur. Ancak

$$g'((1 - \sqrt{5})/2) \doteq 2.677 > 1$$

dir ve iterasyon negatif sabit nokta için yakınsamaz.

- $c = -\frac{1}{\sqrt{5}}$ için

$$g(x) = x - \frac{1}{\sqrt{5}}(x^2 - x - 1)$$

olup,

$$g'(x) = \frac{-2}{\sqrt{5}}x + \frac{1}{\sqrt{5}} + 1$$

ve

$$g'((1 + \sqrt{5})/2) = 0$$

elde edilir ve bu durumda da iterasyonun yakınsamış olması sürpriz değildir. Ancak $-0.618 < x_0 \leq 3.8$ için iterasyon $(1 + \sqrt{5})/2$ ye yakınsarken, $x_0 < -0.619$ ve $x_0 > 3.9$ için iterasyon ıraksamaktadır. Öte yandan bu g fonksiyonu sadece pozitif sabit noktasını belirlemek için kullanılabilir, çünkü

$$g'((1 - \sqrt{5})/2) = 2 > 1$$

dir.

•

$$g(x) = \frac{x^2 + 1}{2x - 1}$$

için,

$$g'(x) = \frac{(x - 1)^2 + x^2 - 1}{(2x - 1)^2}$$

olup,

$$g'((1 + \sqrt{5})/2) = 0$$

dir. Ayrıca $(1, \infty)$ aralığındaki her x için $0 < g'(x) < 1$ dir. Dolayısıyla herhangi $x_0 \in (1, \infty)$ için iterasyon yakınsaktır. $x_0 = 0.6$ için $|g'(0.6)| = 12 > 1$ olmasına rağmen iterasyon bu başlangıç noktası ile de yakınsaktır. Ayrıca aynı iterasyon fonksiyonu negatif sabit nokta için de yakınsak iterasyon üretir, çünkü

$$g'((1 - \sqrt{5})/2) = 0 < 1$$

dir.

Sonuç olarak

- İterasyon fonksiyonu ile üretilen dizinin, fonksiyonun sahip olduğu her bir sabit nokta için yakınsak iterasyon üretmesi beklenmemelidir.
- Yakınsak iterasyon genelde sadece sabit noktanın yeterince yakın komşuluğunda seçilen x_0 başlangıç noktaları için elde edilebilir.
- r noktası g nin sabit noktası olmak üzere $|g'(r)| < 1$ şartını sağlayan g iterasyon fonksiyonu ile yakınsak dizi üretebilmek için r nin yeterince yakın komşuluğunda seçilen x_0 için $|g'(x_0)| < 1$ şartı, genelde yeterlidir ancak gerekli değildir.

6.3 İterasyon fonksiyonu seçimi ve *Newton-Raphson* yöntemi

Yukarıdaki örneklerden, bazı iterasyon fonksiyonlarının başlangıç noktası sabit noktaya çok yakın seçilmesine rağmen, sabit noktaya yakınsayan iterasyon üretmeyebileceğini gözlemledik. Öte yandan, bir iterasyon fonksiyonunun bir sabit nokta için yakınsak iterasyon üretirken, bir diğer sabit nokta için ıraksak bir iterasyon oluşturabileceğine dikkat ettik. Ayrıca yakınsak iterasyon üretmesine rağmen, farklı iterasyon fonksiyonlarının farklı hızlarda yakınsayan diziler oluşturduğunu da gözlemledik. Ancak bunları incelerken iterasyon fonksiyonlarının nasıl seçildiği üzerinde durmadık.

Bu bağlamda *Newton*², hala güncelliğini koruyan bir öneri sunmaktadır: Sıfır noktası komşuluğunda türevlenebilen f fonksiyonunun sıfır yerini bulma problemi

$$g(x) = x - f(x)/f'(x)$$

ile tanımlanan g fonksiyonunun sabit noktasını bulma problemine denktir, diğer bir deyimle her iki problem de aynı çözüm kümesine sahiptirler. Bu özel g iterasyon fonksiyonu ve sıfır yerine yeterince yakın seçilen x_0 başlangıç noktası ile oluşturulan

$$x_{n+1} = x_n - f(x_n)/f'(x_n), n = 0, 1, 2, \dots$$

iterasyonuna *Newton* veya *Newton-Raphson*³ iterasyonu ve g fonksiyonuna da *Newton-Raphson* iterasyon fonksiyonu adı verilmektedir.

ÖRNEK 6.3. $f(x) = x^2 - 7$ fonksiyonunun $x_0 = 3$ noktasına yakın sıfır yeri için $x_i, i = 1, 2, 3, 4$ *Newton-Raphson* yaklaşımlarını hesaplayınız. Her adımda virgülden sonra doğru hesaplanan basamak sayısının nasıl arttığını gözlemleyiniz.

Öncelikle *Newton-Raphson* iterasyon fonksiyonunu hesaplayalım:

$$\begin{aligned} g(x) &= x - \frac{f(x)}{f'(x)} \\ &= x - \frac{x^2 - 7}{2x} \\ &= \frac{x}{2} + \frac{7}{2x} \end{aligned}$$

²Isaac Newton 1642 – 1726, İngiliz fizikçi ve matematikçi

³Joseph Raphson, 1648-1715, İngiliz matematikçi.

olarak elde ederiz. O halde iterasyonumuz

$$x_{n+1} = g(x_n) = \frac{x_n}{2} + \frac{7}{2x_n}, n = 0, 1, \dots$$

olarak tanımlanır. İlk üç yaklaşımı kolayca hesaplayabiliriz:

$$\begin{aligned} x_1 &= \frac{x_0}{2} + \frac{7}{2x_0} = \frac{3}{2} + \frac{7}{6} = \frac{8}{3} \doteq 2.\underbrace{6}_{6666666666666667}, \\ x_2 &= \frac{x_1}{2} + \frac{7}{2x_1} \doteq 2.\underbrace{645}_{833333333333333}, \\ x_3 &= \frac{x_2}{2} + \frac{7}{2x_2} \doteq 2.\underbrace{645751}_{312335958} \\ x_4 &= \frac{x_3}{2} + \frac{7}{2x_3} = 2.\underbrace{645751311064}_{591} \end{aligned}$$

Öte yandan elde ettiğimiz bu değer virgülden sonra onbeş basamağa kadar yuvarlanmış

$$\sqrt{7} \doteq 2.645751311064591\dots$$

değeridir. Her adımda virgülden sonraki doğru basamak sayısı, bir önceki adımın **en az iki** katıdır.

Newton-Raphson yöntemine ait Program 6.2 aşağıda verilmektedir.

Gözlem 6.1. *Örnek 6.1 son şıkkında kullandığımız*

$$g(x) = \frac{x^2 + 1}{2x - 1} = x - \frac{x^2 - x - 1}{2x - 1}$$

fonksiyonu esasen Newton-Raphson iterasyon fonksiyonudur, dolayısıyla bu fonksiyon seçimiyle yukarıda Newton-Raphson yöntemini uygulamış olduk ve iterasyonun her iki sabit noktaya da hızlı bir biçimde yakınsadığını gözlemledik.

6.3.1 Newton-Rapson yönteminin geometrik yorumu

Newton-Rapson yöntemine göre sıfır yerine yeterince yakın seçilen x_0 için elde edilen $x_1 = x_0 - f(x_0)/f'(x_0)$ yaklaşımı Şekil 4.2 de de görüldüğü üzere $(x_0, f(x_0))$ noktasında $y = f(x)$ fonksiyonun grafiğine çizilen teğet doğrusunun (yani, $y = f(x_0) + f'(x_0)(x - x_0)$ ile tanımlanan doğrunun) x eksenini kesim noktasıdır.

```

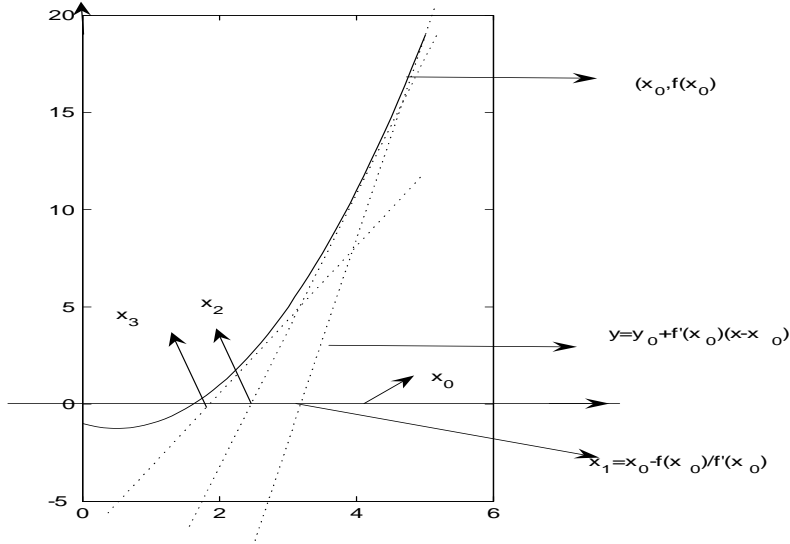
function x1=newton(f,fp,x0)
% Newton-Raphson örneđi, ec.
%-----
min_deger=1e-10;max_deger=1e5;max_sayac=100;
sayac=0;
test=1;
while test
    delx=-fp(x0)\ f(x0);
    x1=x0+delx;
    fark=norm(delx,inf);
    sayac=sayac+1;
    test=(fark>min_deger)&(norm(x1,inf)<max_deger)&(sayac<max_sayac);
    x0=x1;
end
%Yakınsamama durumunda yorumlar
if norm(x1,inf)>max_deger
    disp('iterasyon iraksaktır'); x1=[];
elseif sayac==max_sayac
    disp(['iterasyon ',num2str(sayac), ' adimda yakınsamamıstır']);
    x1=[];
end
%-----

```

Program 6.2: Newton-Raphson iterasyon uygulaması

Diđer bir bakıř ađısıyla f fonksiyonunun sıfır yerini dođrudan bulmak yerine, x_0 noktasındaki Taylor ađılımmın lineer kısmının sıfır yeri x_1 yaklaşımı olarak kabul edilir. Daha sonra x_1 noktasındaki Taylor ađılımmın lineer kısmının sıfır yeri x_2 olarak kabul edilir ve dizinin diđer elemanları benzer biçimde elde edilir, Şekil 6.2.

Benzer biçimde yöntem x_2 yaklaşımını, $(x_1, f(x_1))$ noktasından çizilen teđetin x eksenini kesim noktası olarak kabul eder. Bu yaklaşımlar eđer bir noktaya yakınsıyorlarsa, yakınsadıkları nokta g fonksiyonunun sabit noktası ve dolayısıyla da f nin sıfır yeridir.

Şekil 6.2: *Newton-Raphson* yöntemi ile f nin sıfır yeri için yaklaşımlar

6.3.2 *Newton-Raphson* yönteminin yakınsaklığı

- r noktası f nin sıfır yeri ve $f'(r) \neq 0$, $f''(r)$ tanımlı ise $g'(r) = 0$ dir.

$$g(x) = x - f(x)/f'(x)$$

olup, $f(r) = 0$ olduğundan

$$g'(r) = 1 - \left((f'(r))^2 - f(r)f''(r) \right) / (f'(r))^2 = 0$$

olarak elde edilir.

- *Newton-Raphson* yöntemi sıfır noktasının küçük komşuluğunda kuadratik (ikinci basamaktan) yakınsak bir yöntemdir: r noktası g fonksiyonunun sabit noktası, $f'(r) \neq 0$ ve x_n ise n - inci adımda elde edilen yaklaşım olmak üzere

$$x_{n+1} = g(x_n) = g(r) + g'(r)(x_n - r) + \frac{1}{2}g''(c_x)(x_n - r)^2, c_x : x_n \text{ ve } r \text{ arasında}$$

veya $g(r) = r$ ve $g'(r) = 0$ olduğunu kullanarak

$$x_{n+1} - r = \frac{1}{2}g''(c_x)(x_n - r)^2$$

elde ederiz. Dolayısıyla $n + 1$ -inci adımdaki hata n -inci adımdaki hatanın karesiyle orantılıdır. Bu sonuç ise yöntemin, eğer yakınsak ise, kuadratik olarak yakınsadığını gösterir. Örnek 6.3 de elde ettiğimiz yaklaşımlar, yakınsamanın kuadratik olduğunu gösterir.

- *Newton-Raphson* iterasyonunun r noktasına yakınsaması için yeter şart x_0 noktasının r noktasını içeren ve

$$|g'(x)| = |f(x)f''(x)|/(f'(x))^2 < 1$$

şartını sağlayan komşulukta seçilmesidir. Bu şart yukarıda da belirtildiği gibi gerekli değildir: Örnek 4.2 de $x_0 = 0.6$ için $|g'(x_0)| > 1$ dir, ancak iterasyon yine de yakınsaktır.

- Eğer bir g iterasyon fonksiyonu r sabit noktasının komşuluğunda $n - inci$ mertebeden türe ve sahip ve

$$g'(r) = g''(r) = \dots = g^{(n-1)}(r) = 0, g^{(n)}(r) \neq 0$$

ise g ile elde edilen iterasyon $n - inci$ basamaktan yakınsaktır (Alıştırma 9). $n = 1$ olması durumunda yöntem *lineer yakınsaktır* denir. Bu sonuç yukarıda *Newton* yöntemi için elde edilen sonucun genel halidir ve benzer biçimde ispatlanabilir.

- *Newton-Raphson* fonksiyonundan farklı olarak

$$g(x) = x - \frac{1}{\sqrt{5}}(x^2 - x - 1)$$

ile pozitif sabit nokta için türetilen iterasyon da kuadratik yakınsaktır, çünkü $g'((1 + \sqrt{5})/2) = 0$ ve $g''((1 + \sqrt{5})/2) = -2/\sqrt{5} \neq 0$ dir.

6.3.3 Newton-Raphson yönteminde başlangıç noktası seçimi

Newton-Raphson yönteminin en zayıf yönü, başlangıç noktası seçiminin sıfır yerine "yakın" seçilmesini gerektirmesidir. Tahmini noktanın ne kadar "yakın"

seçilmesi gerektiği problemde probleme göre değişmektedir. Örneğin altın oranı sıfır yeri kabul eden örneğimizde (Örnek 6.1), $x_0 \in (1, \infty)$ olması durumunda altın orana yakınsayabildiğimizi görmüştük. Dolayısıyla bu problem için başlangıç noktası seçimi herhangi bir sorun teşkil etmemiştir.

Ancak bazı fonksiyonlar başlangıç noktası tahmininin iyi yapılmış olmasını gerektirirler.

ÖRNEK 6.4. $f(x) = ((x-2)^2 + 1/100)(x-3)$ fonksiyonu için farklı başlangıç noktası seçimi ile *Newton-Raphson* yönteminin yakınsaklığını araştırınız.

Çözüm.

Fonksiyonunun $x = 3$ noktasındaki sıfır yerini $|f(x_n)| < 10^{-6}$ kriterini sağlayacak yaklaşımlar elde etmek için gerekli iterasyon sayıları (iter) aşağıdaki tabloda verilmektedir.

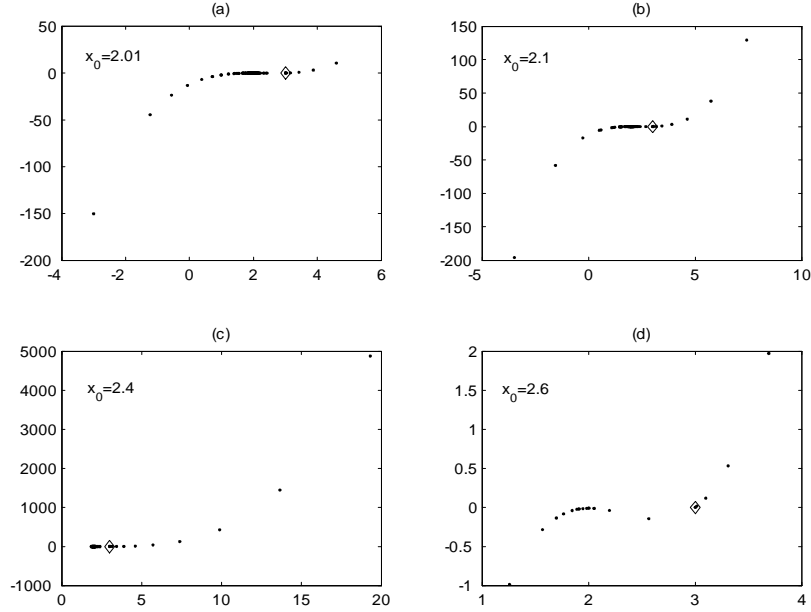
x_0	2	2.01	2.02	2.04	2.06	2.08	2.1	2.2	2.3	2.4	2.5	2.9
iter	1	144	15	27	9	52	129	12	28	44	10	4

Yukarıdaki tablodan sıfır yeri komşuluğunda seçilen farklı noktalar ile *Newton-Raphson* iterasyonunun $|f(x_n)| < 10^{-6}$ kriteri ile yakınsaması için farklı sayıda iterasyon sayıları gerekmektedir. Ancak gerekli iterasyon sayılarının düzensiz bir dağılıma sahip olduğu görülmektedir. Örneğin $x_0 = 2$ için sadece 1 adet iterasyon gerekirken, $x_0 = 2.01$ için 144 adet ve $x_0 = 2.02$ için ise 15 adet iterasyon gerekmektedir.

Herbir başlangıç noktası için elde edilen iterasyonlar yakınsak, ancak yakınsama hızları çok farklıdır. Yine yakınsamanın monoton değil, salınımlı olduğu görülmektedir. Sonuç olarak örneğimizde başlangıç noktası seçiminin iterasyonunu yakınsama hızı üzerinde etkisinin çok fazla olduğu görülmektedir. Bunun nedeni sıfır noktasının geniş bir komşuluğunda f fonksiyonunun sıfıra çok yakın değerler alması ve f' türevinin $x_1 = 2.005, x_2 = 2.6616$ sıfıryerlerine sahip olması ve dolayısıyla $f(x)/f'(x)$ fonksiyonunun süreksiz olmasıdır.

6.3.4 Newton-Rapson ve karmaşık kökler

Hatırlatma 6.1. Genel olarak iterasyon yöntemleri karmaşık sıfıryerlerini belirlemek için de kullanılabilirler. Bu sonuç özel olarak *Newton-Raphson* yöntemi için de geçerlidir, ancak bunun için başlangıç noktasının da karmaşık bir sayı olarak seçilmesi gerekir.



Şekil 6.3: Farklı başlangıç değerleri ile x_i Newton-Raphson yaklaşımları için $(x_i, f(x_i))$ değerleri:(a) $x_0 = 2.01$,(b) $x_0 = 2.1$ (c) $x_0 = 2.4$ (d) $x_0 = 2.6$

ÖRNEK 6.5. $f(x) = x^2 + 1$ fonksiyonunun $x_0 = 1 + i$ noktası komşuluğundaki sıfır yerini Newton-Rapson yöntemi ile belirleyiniz.

Çözüm.

Newton-Rapson yöntemi yardımıyla aşağıdaki yaklaşımları elde ederiz:

i	x_i
0	$1 + i$
1	$0.2500 + 0.7500i$
2	$-0.0750 + 0.9750i$
3	$0.0017 + 0.9973i$
4	$-0.0000 + 1.0000i$

6.3.5 Katlı kökler için Newton-Raphson yöntemi

Mevcut haliyle Newton-Raphson yöntemi katlı kökler için lineer olarak yakın-sayan bir yöntemdir. Eğer r noktası f fonksiyonunun $m > 1$ katlı bir sıfır yeri ise

$$f(x) = (x - r)^m h(x), h(r) \neq 0$$

olacak biçimde h fonksiyonu vardır. Bu durumda g iterasyon fonksiyonu

$$\begin{aligned} g(x) &= x - f(x)/f'(x) \\ &= x - \frac{(x - r)^m h(x)}{m(x - r)^{m-1} h(x) + h'(x)(x - r)^m} \\ &= x - \frac{(x - r)h(x)}{mh(x) + h'(x)(x - r)} \end{aligned}$$

olarak yazılabilir ve

$$g'(x) = 1 - \frac{\begin{bmatrix} (h(x) + (x - r)h'(x))(mh(x) + h'(x)(x - r)) \\ -(mh'(x) + h''(x)(x - r) + h'(x)(x - r)h(x)) \end{bmatrix}}{(mh(x) + h'(x)(x - r))^2}$$

elde edilir. Buradan

$$\begin{aligned} g'(r) &= 1 - h(r)mh(r)/(mh(r))^2 \\ &= 1 - 1/m \neq 0 \end{aligned}$$

olur. $g'(r) < 1$ olduğu için *Newton-Raphson* iterasyonu katlı kökler uygun başlangıç noktası ile sadece lineer olarak yakınsaktır. Bu durumda yukarıda tanımlanan g iterasyon fonksiyonu yeniden düzenlenerek

$$g(x) = x - mf(x)/f'(x) \quad (6.2)$$

olarak yazılırsa, yine $g(r) = r$ olduğu açıktır. Ayrıca yukarıdaki işlemler tekrar edilerek $g'(r) = 0$ elde ederiz. Bu sonuç ise yöntemin uygun başlangıç noktası ile en az kuadratik olarak yakınsayacağını ifade eder. Bir örnek üzerinde her iki yöntemin performansını karşılaştırmak mümkündür.

ÖRNEK 6.6. $f(x) = x^2 - 2x + 1$ fonksiyonu için $x_0 = 2$ alarak basit kökler ve katlı kökler için *Newton-Raphson* yöntemiyle sıfır yeri yaklaşımlarını hesaplayınız.

Çözüm.

<i>İterasyon</i>	<i>Newton(BasitKökler)</i>	$(x_{n+1} - r)/(x_n - r), r = 1$
0	2.0000	
1	1.5000	$0.5/1 = 0.5$
2	1.2500	$0.25/0.5 = 0.5$
3	1.1250	$0.125/0.250 = 0.5$
4	1.0625	$0.625/0.1250 = 0.5$
5	1.0313	$0.313/0.625 = 0.5008$
6	1.0156	$0.0156/0.0313 = 0.4984$
7	1.0078	$0.0078/0.0156 = 0.5$
8	1.0039	$0.0039/0.0078 = 0.5$
9	1.0020	$0.0020/0.0039 = 0.5128$
10	1.0010	$0.0010/0.0020 = 0.5$

Tablo 6.1: Basit kökler için Newton yaklaşımları

Basit kökler için kuadratik olarak yakınsak olan *Newton-Raphson* yöntemini $x = 1$ noktasında iki katlı köke sahip f fonksiyonu için uygulayarak aşağıdaki tabloda verilen ve son sütundan görüldüğü üzere lineer olarak yakınsayan yaklaşımları elde ederiz, yakınsama lineerdir çünkü $(x_{n+1} - r)/(x_n - r)$ oranı yaklaşık olarak sabittir:

Öte yandan aynı başlangıç değeri ile $m = 2$ değeri için yukarıda verilen $x_0 = 2$ başlangıç noktası ile uygulayarak ilk adımda $x_1 = 1$ gerçek sıfır yerini elde ederiz.

Katlı sıfır yerleri belirlemek için basit sıfır yeri için geliştirilen versiyonun kullanılması uygun olmadığı gibi basit sıfır yerini belirlemek için de, katlı sıfır yeri için (6.2) ile düzenlenen versiyon kullanılırsa istenilen sonuçlar elde edilemeyebilir.

Örneğin $f(x) = x^2 - x - 1$ fonksiyonu ve $x_0 = \sqrt{2}$ başlangıç noktası ve basit sıfır yerleri için *Newton-Raphson* yöntemi Örnek 4.1 de belirtildiği gibi 5 adımda yakınsarken, $m = 2$ ile katlı sıfır yerleri için düzenlenen versiyon 2.0000 ve 1.3333 değerleri arasında alternatif olarak salınım yapmaktadır (Alıştırma 14).

6.3.6 Sayısal türev ile yaklaşım(Kiriş yöntemi)

Newton-Raphson yönteminde $f'(x_i)$ türev değeri şüphesiz fonksiyonun türevinin de bilinmesini gerektirmektedir. Türev bilgisi gerektirmeyen alternatif

bir yöntem ise

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

yaklaşımının kullanılmasıdır. Bu durumda tek birbaşlangıç değeri yerine x_0 ve x_1 başlangıç değerleri ile başlatılabilen

$$x_{i+1} = x_i - \frac{f(x_i)}{\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i), i = 2, 3, \dots \quad (6.3)$$

yaklaşımları elde edilir. Bu yöntem giriş yöntemi olarak bilinir. *Newton-Raphson* yönteminde olduğu gibi $(x_i, f(x_i))$ noktasındaki teğetin x eksenini kesim noktası yerine, $(x_{i-1}, f(x_{i-1})), (x_i, f(x_i))$ noktalarından geçen ve

$$y - f(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}(x - x_i)$$

ile verilen giriş doğrusunun x eksenini kesim noktası sıfır yeri için x_{i+1} yaklaşımı olarak kabul edilir.

ÖRNEK 6.7. $f(x) = x^2 - x - 1$ fonksiyonu için $x_0 = 3, x_1 = 2$ başlangıç tahminleri ve giriş yöntemi ile x_2, x_3, x_4 ve x_5 sıfır yeri yaklaşımlarını hesaplayınız.

Çözüm.

İlk yaklaşım

$$\begin{aligned} x_2 &= x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1) \\ &= 2 - \frac{2 - 3}{f(2) - f(3)} f(2) \\ &= 2 - \frac{-1}{1 - 5} = 2 - \frac{1}{4} = \frac{7}{4} \cong 1.75 \end{aligned}$$

olarak elde edilir. Benzer biçimde diğer yaklaşımlar

$$x_3 = 5/3 \cong 1.6667, \quad x_4 = 47/29 \cong 1.6207, \quad x_5 = 322/199 \cong 1.6181.$$

olarak elde edilir.

6.4 Nonlinear sistemler için *Newton* yöntemi ve benzeri bir yöntem

İki değişkenli f ve g fonksiyonları için

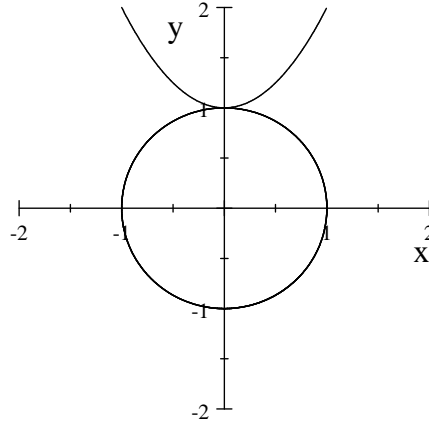
$$\begin{aligned} f(x, y) &= 0 \\ g(x, y) &= 0 \end{aligned} \quad (6.4)$$

nonlinear sisteminin (p, q) ile gösterilen sıfır yerine sahip olduğunu kabul edelim.

Örneğin

$$\begin{aligned} x^2 + y^2 - 1 &= 0 \\ -x^2 + y - 1 &= 0 \end{aligned}$$

nonlinear sisteminin tek çözümü Şekil 6.4 ile belirtilen çember ve parabolün tek bir kesişim noktası olan $(0, 1)$ noktasıdır.



Şekil 6.4: Birim çember ve parabol

Ayrıca bu noktanın bir D komşuluğunda

$$J = \begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix}$$

Jacobien matrisinin tersinir olduğunu kabul edelim. Tek değişkenli fonksiyonlarda olduğu gibi, bu defa da nonlinear sistemin (x_0, y_0) noktasındaki Taylor açılımınının lineer kısmınının sıfır yerini (x_1, y_1) yaklaşımı olarak alalım:

$$\begin{aligned} f(x, y) &= f(x_0, y_0) + (x - x_0)f_x(x_0, y_0) + (y - y_0)f_y(x_0, y_0) + \dots \\ g(x, y) &= g(x_0, y_0) + (x - x_0)g_x(x_0, y_0) + (y - y_0)g_y(x_0, y_0) + \dots \end{aligned} \quad (6.5)$$

Taylor açılımınının lineer kısımları ile oluşturulan

$$\begin{aligned} f(x_0, y_0) + (x - x_0)f_x(x_0, y_0) + (y - y_0)f_y(x_0, y_0) &= 0 \\ g(x_0, y_0) + (x - x_0)g_x(x_0, y_0) + (y - y_0)g_y(x_0, y_0) &= 0 \end{aligned} \quad (6.6)$$

lineer sistemini göz önüne alalım ve bu sistemin (x, y) çözümünü (x_1, y_1) ile gösterelim.

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, \Delta X = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}, F(x, y) = \begin{bmatrix} f(x, y) \\ g(x, y) \end{bmatrix}$$

notasyonu ile (6.6) sistemi,

$$J(X^{(0)})\Delta X = -F(X^{(0)}) \quad (6.7)$$

olarak yazılabilir. Bu sistem çözülerek elde edilen ΔX ile

$$X^{(1)} = X^{(0)} + \Delta X$$

yaklaşımı veya eleman bazında yazılarak

$$X^{(1)} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

elde edilir. Benzer biçimde diğer yaklaşımlar, $n = 0, 1, \dots$ için

$$J(X^{(n)})\Delta X = -F(X^{(n)}) \quad (6.8)$$

$$X^{(n+1)} = X^{(n)} + \Delta X$$

olarak elde edilir.

ÖRNEK 6.8.

$$\begin{aligned}x^2 + y^2 &= 1 \\x - y &= 0\end{aligned}$$

denkleminin çözümü için $(x_1, y_1), (x_2, y_2)$ Newton yaklaşımlarını $(x_0, y_0) = (1, 2)$ başlangıç noktası için hesaplayınız.

Çözüm.

Verilen sistemi

$$\begin{aligned}f(x, y) &= 0 \\g(x, y) &= 0\end{aligned}$$

olarak yazarsak,

$$F(x, y) = \begin{bmatrix} f(x, y) \\ g(x, y) \end{bmatrix} = \begin{bmatrix} x^2 + y^2 - 1 \\ x - y \end{bmatrix}$$

ile tanımladığımız vektör değerli fonksiyon için

$$F(x_0, y_0) = F(1, 2) = \begin{bmatrix} f(1, 2) \\ g(1, 2) \end{bmatrix} = \begin{bmatrix} 1^2 + 2^2 - 1 \\ 1 - 2 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

elde ederiz. Ayrıca

$$J = \begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix} = \begin{bmatrix} 2x & 2y \\ 1 & -1 \end{bmatrix}$$

olup,

$$J(x_0, y_0) = J(1, 2) = \begin{bmatrix} 2 & 4 \\ 1 & -1 \end{bmatrix}$$

olur. Bu durumda

$$J(X^{(0)})\Delta X = -F(X^{(0)})$$

denkleminin sistemi

$$\begin{bmatrix} 2 & 4 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = - \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

sistemine döndürür. Bu lineer sistemi uygun bir yöntemle çözerek $\Delta x = 0$, $\Delta y = -1$ elde ederiz. O halde

$$X^{(1)} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

elde ederiz. Benzer biçimde

$$F(X^{(1)}) = F(x_1, y_1) = F(1, 1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

ve

$$J(x_1, y_1) = J(1, 1) = \begin{bmatrix} 2 & 2 \\ 1 & -1 \end{bmatrix}$$

olup,

$$J(X^{(1)})\Delta X = -F(X^{(1)})$$

denklem sistemi

$$\begin{bmatrix} 2 & 2 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = - \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

sistemine dönüşür. Bu sistem de çözülerek, $\Delta x = -1/4$, $\Delta y = -1/4$ olarak elde edilir. O halde

$$\begin{aligned} X^{(2)} &= \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1/4 \\ -1/4 \end{bmatrix} = \begin{bmatrix} 3/4 \\ 3/4 \end{bmatrix} \end{aligned}$$

elde ederiz.

Yukardaki nonlinear sistemin çözümü için gerekli yaklaşımları *Newton* yöntemiyle ve yukarıda incelenen fonksiyon sıfır yerlerini belirlemek amacıyla geliştirdiğimiz *newton.m* isimli Program 6.2 yardımıyla elde edilebilir. Bu durumda f : çözülmesi gereken nonlinear sistem ve fp ise söz konusu sistemin Jakobien matrisidir.

Yöntemi Örnek 6.8 için çalıştıralım:

Test

```
>> f=@(x) [x(1)^2+x(2)^2-1;x(1)-x(2)];
```

```
>> fp=@(x) [2*x(1) 2*x(2);1 -1];
```

```
>> x0=[1 2]';
```

```
>> newton(f,fp,x0) komutu ile
```

```
 x0 = 1 0.75000 0.70833 0.70711 0.70711
```

```
     1 0.75000 0.70833 0.70711 0.70711
```

yaklaşımlarını elde ederiz.

6.4.1 Nonlinear sistemler için Newton benzeri uygulama

Yukarıdaki tanımdan görüldüğü üzere, skaler denklemlerde f fonksiyonu ve f nin türevini gerektiren Newton yöntemi, Nonlinear cebirsel sistemler için de, söz konusu sistem ve Jacobiyen matrisininin de kullanıcı tarafından sağlanmasını gerektirmektedir . Ancak, giriş yöntemini hatırlayacak olursak f nin türevi yerine, türev için geri fark yaklaşımını kullanarak türev bilgisinin sağlanma zorunluluğunu ortadan kaldırmak mümkündür. Benzer bir işlemi cebirsel sistemler için kısmi türevleri içeren Jakobiyen matrisi için de gerçekleştirerek, Jacobiyenin hesaplanması zorunluluğundan kurtulabiliriz.

Bu amaçla 6.4 sistemi ve $X^{(0)} = (x_0, y_0)$ başlangıç noktası verilmiş olsun. $X^{(1)} = (x_1, y_1)$ noktasını $X^{(0)}$ noktasının yakın komşuluğunda bir nokta olarak seçelim. Newton yönteminden hareketle Giriş yöntemini elde ederken kullandığımız

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

yaklaşımına benzer olarak

$$\begin{aligned} J(X^{(i)}) &= \begin{bmatrix} f_x(x_i, y_i) & f_y(x_i, y_i) \\ g_x(x_i, y_i) & g_y(x_i, y_i) \end{bmatrix} \\ &\cong \begin{bmatrix} \frac{f(x_i, y_i) - f(x_{i-1}, y_i)}{x_i - x_{i-1}} & \frac{f(x_i, y_i) - f(x_i, y_{i-1})}{y_i - y_{i-1}} \\ \frac{g(x_i, y_i) - g(x_{i-1}, y_i)}{x_i - x_{i-1}} & \frac{g(x_i, y_i) - g(x_i, y_{i-1})}{y_i - y_{i-1}} \end{bmatrix} \\ &: = \hat{J}(X^{(i)}) \end{aligned}$$

yaklaşımını kullanabiliriz. Bu durumda

$$Y = F(X^{(1)})$$

olmak üzere,

$$\hat{J}(X^{(1)}) = \begin{bmatrix} \frac{f(x_1, y_1) - f(x_0, y_1)}{x_1 - x_0} & \frac{f(x_1, y_1) - f(x_1, y_0)}{y_1 - y_0} \\ \frac{g(x_1, y_1) - g(x_0, y_1)}{x_1 - x_0} & \frac{g(x_1, y_1) - g(x_1, y_0)}{y_1 - y_0} \end{bmatrix}$$

ile

$$\begin{aligned} \hat{J}(X^{(1)})\Delta X &= -Y \\ X^{(2)} &= X^{(1)} + \Delta X \end{aligned}$$

yaklaşımını elde ederiz. Bir sonraki iterasyona $X^{(0)} = X^{(1)}$, $X^{(1)} = X^{(2)}$ olarak devam ederiz, ve bu işleme uygun bir sonlandırma kriteri sağlanana kadar devam ederiz. *Newton* benzeri bu yönteme ait Program 6.3 aşağıda verilmektedir.

Test

```
>> F=@(x) [x(1)^2/4+x(2)^2/9-1;x(1)^2/9+x(2)^2/4-1];
>> X0=[1 1]';
>> newtonsqv(F,x0)
ans =
1.6641
1.6641
>> X0=[1 -1]';
>> newtonsqv(F,X0)
ans =
1.6641
-1.6641
```

Alıştırmalar 6.1.

1. $g : [a, b] \rightarrow [a, b]$ sürekli bir fonksiyon ise g nin $[a, b]$ aralığında en az bir sabit noktası olduğunu gösteriniz. (İpucu, $g(a) = a$ veya $g(b) = b$ ise bu durumda en az bir sabit nokta zaten mevcuttur. $g(a) > a$ ve $g(b) < b$ olduğunu kabul ederek $f(x) = x - g(x)$ fonksiyonunun $[a, b]$ de en az bir sıfır yeri olması gerektiğini aradeğer teoremi yardımıyla gösteriniz.)
2. Aşağıda verilen g fonksiyonlarının aynı sabit noktalara sahip olduklarını gösteriniz.

$$(a) \quad g(x) = -\frac{1}{5}x(x-3)(x+3)$$

$$(b) \quad g(x) = \frac{1}{8}x(x^2 - 12)$$

$$(c) \quad g(x) = \frac{1}{5}x(2x^2 - 13)$$

$$(d) \quad g(x) = \frac{2x^3}{3x^2-4}$$

3. Soru 2'de verilen g fonksiyonlarının bütün sabit noktalarını sıfır yeri kabul eden en düşük dereceli bir f polinom fonksiyonunu belirleyiniz.
4. Soru 3'de belirlediğiniz f fonksiyonunun hangi sıfırlarını bulabilmek için hangi g leri kullanabilirsiniz? Hangisini kullanmayı tercih edersiniz?

```

function X1=newtonsqv(F,X0)
%-----
% Quasi Newton- Newton benzeri yöntem
% Nonlinear sistem sıfır yeri belirler.
% Jacobiyen sayısal yaklaşımla belirlenir.
% F nin tek bir fonksiyon olması durumunda, kiriş yöntemine indirgenir.
% ec, 13 Nisan 2019.
%-----
min_deger=1e-5; max_deger=1e5;max_sayac=40;
test=1;sayac=0;
X1=X0+0.1;
while test
    Yp=jacobi(F,X0,X1);
    Y=F(X1);
    Delx=-Yp \ Y;
    X2=X1+Delx;
    sayac=sayac+1;
    fark=norm(X2-X1,inf);
    test=(fark>min_deger)&(norm(X2,inf)<max_deger)&(sayac<max_sayac);
    X0=X1;X1=X2;
end
%Yakınsamama durumunda mesajlar ....

function J=jacobi(F,X0,X1)
n=length(X0);
for i=1:n
    for j=1:n;
        dxv=zeros(n,1);
        dx=X1(j)-X0(j);
        dxv(j)=dx;
        J(i,j)=(F(X1)(i)-F(X1-dxv)(i))/(X1(j)-X0(j));
    end
end
end

```

Program 6.3: Nonlinear sistemler için Newton benzeri yöntemi uygulaması

5. $f(x) = x^2 - 5$ fonksiyonunun pozitif sıfır yerini bulmak için $g(x) = x + cf(x)$ iterasyon fonksiyonu ve seçilen c için $|g'(x_0)| < 1$ şartını sağlayan başlangıç noktası ile oluşturulan $x_{n+1} = g(x_n)$ iterasyonu göz önüne alalım.

- İterasyonun yakınsaması için c hangi aralıkta değerler almalıdır?
- Kuadratik olarak yakınsak olan iterasyon üretebilmek için c ne olmalıdır?

6. Soru 5'de verilen f fonksiyonunun sıfırlarlarının aşağıda verilen g fonksiyonlarının sabit noktaları olduğunu gösteriniz. Hangi g ler ile oluşturulan iterasyonlar yakınsar? Yakınsak iterasyonlar için yakınsama basamaklarını belirleyiniz.

- $g(x) = x(6 - x^2)$
- $g(x) = \frac{1}{8}x(13 - x^2)$
- $g(x) = \frac{1}{2}x(3 - \frac{1}{5}x^2)$

7. $g(x) = x - (x - 1)^2$ iterasyon fonksiyonu verilsin.

- Verilen iterasyon fonksiyonunun $r = 1$ sabit noktasına sahip olduğunu ve $g'(r) = 1$ olduğunu gözlemleyiniz.
- $r = 1$ noktasına yeterince yakın ve $x_0 > 1$ başlangıç noktası için oluşturulan iterasyonun $r = 1$ sabit noktasına yakınsadığını uygun bir $x_0 > 1$ ile x_1, x_2, x_3 değerlerini hesaplayarak gözlemleyiniz.
- $x_0 < 1$ şartını sağlayan hiçbir başlangıç noktası için g ile oluşturulan iterasyonun $r = 1$ sabit noktasına yakınsamadığını uygun bir $x_0 > 1$ ile x_1, x_2, x_3 değerlerini hesaplayarak gözlemleyiniz.

8. $g : [a, b] \rightarrow [a, b]$ sürekli ve (a, b) aralığında türevlenebilir ve türevi sürekli bir fonksiyon ve $r \in [a, b]$, g nin bu aralıktaki bir sabit noktası olsun. Ayrıca $|g'(r)| > 1$ olduğunu kabul edelim. Bu durumda $x_0 \in [a, b]$, $x_0 \neq r$ için $x_{n+1} = g(x_n)$, $n = 0, 1, 2, \dots$ ile üretilen $\{x_n\}$ dizisi r sabit noktasına yakınsamayacağını gösteriniz.

9. Eğer bir g iterasyon fonksiyonu r sabit noktasının komşuluğunda $n - \text{inci}$ mertebeden türeve sahip ve

$$g'(r) = g''(r) = \dots = g^{(n-1)}(r) = 0, g^{(n)}(r) \neq 0$$

ise g ile elde edilen iterasyon $n - inci$ basamaktan yakınsak olduğunu gösteriniz.

10. $g(x) = x - 1/2 \sin(2x)$ iterasyon fonksiyonu ile oluşturulan iterasyonların $r = 0$ ve $r = \pi$ sabit noktalarına yakınsak, ancak $r = \pi/2$ sabit noktasına yakınsak olmadığını ilgili teori yardımıyla gösteriniz.

11. $g(x) = x - \tan(2x)$ iterasyon fonksiyonu verilsin.

- $r = 0$ in g nin bir sabit noktası olduğunu ve $g'(0) = -1$ olduğunu gözlemleyiniz.
- g ile oluşturulan iterasyonların hiçbir $x_0 \neq 0$ için $r = 0$ sabit noktasına yakınsamayacağını uygun x_0 başlangıç değerleri seçerek gözlemleyiniz.

12. $f(x) = (x + 2) \exp(-x)$ fonksiyonunun sıfır yerini Newton yöntemiyle belirleyebilmek için en büyük $x_0 > -2$ başlangıç tahmini ne olabilir?

13. $f(x) = x - \sinh(x)$ fonksiyonu verilsin. Uygun bir x_0 başlangıç noktası seçerek, sırasıyla $m = 1; 2$ ve 3 için

$$x_{n+1} = x_n - m f(x_n) / f'(x_n), n = 0, 1, 2, \dots$$

Newton-Raphson iterasyonlarını $|x_{n+1} - x_n| < 10^{-5}$ sonuçlandırma kriteri sağlanıncaya kadar hesaplayınız. Her bir m için gerekli iterasyon sayısını not ediniz? Hangi m değeri için en hızlı yakınsak diziyi elde ettiniz? Neden?

14. $f(x) = x^2 - x - 1$ fonksiyonu ve $x_0 = \sqrt{2}$ başlangıç noktası ve basit sıfır yerleri için Newton-Raphson yönteminin Örnek 6.1 de belirttiği gibi 5 adımda yakınsarken, $m = 2$ ile katlı sıfır yerleri için düzenlenen versiyonun 2.0000 ve 1.3333 değerleri arasında alternatif olarak salınım yaptığını gözlemleyiniz.

15. Eğer r noktası f fonksiyonunun basit sıfır yeri ve büküm noktası değilse, bu taktirde f ile oluşturulan g Newton-Raphson iterasyon fonksiyonunun r noktasında bir ekstremuma sahip olduğunu ispatlayınız.

16. Bu bölümde verilen Newtons isimli programı (Program 6.2) Örnek 6.8 de verilen denklem sistemi ve $(x_0, y_0) = (2, 1)$ ile çalıştırarak elde edilen sonuçların doğruluğunu kontrol ediniz. İterasyonların sistemin gerçek çözümüne yakınsadığını gözlemleyiniz.

17. Aşağıdaki nonlinear sistemlerin grafiklerini çizerek, kaç adet reel çözüme sahip olduklarını tahmin ediniz.

- (a)

$$y^2 - 4x = 1$$

$$y^2 + 4x = 1$$

- (b)

$$(x - 3/2)^2 + (y - 5/2)^2 = 1/2$$

$$(x - 3/2)^2 + (y - 1)^2 = 5/4$$

- (c)

$$x^2/9 + y^2/4 = 1$$

$$x^2/4 + y^2/9 = 1$$

- (d)

$$x^2/9 + y^2/4 = 1$$

$$x^2/4 - y^2/9 = 1$$

18. Soru 17(a)'ya ait nonlinear sistem için $(x_0, y_0) = (3, 3)$ başlangıç noktası ile (x_1, y_1) ve (x_2, y_2) yaklaşımlarını *Newton* yöntemi yardımıyla hesaplayınız.

19. Soru 17(b)'ye ait nonlinear sistem ve aşağıda verilen başlangıç değerlerin her birisi için (x_1, y_1) ve (x_2, y_2) yaklaşımlarını *Newton* yöntemi yardımıyla hesaplayınız.

- $(x_0, y_0) = (3, 3)$
- $(x_0, y_0) = (-3, 3)$
- $(x_0, y_0) = (-3, -3)$
- $(x_0, y_0) = (3, -3)$

Eldettiğiniz yaklaşımlar, şekildeki grafiklerin arakesit noktalarına yaklaşıyor mu?

20. Soru 17(b)'ye ait nonlineer sistemin yaklaşık çözümünü Soru 19 da verilen (x_0, y_0) başlangıç noktaları ve Program 6.3 yardımıyla elde ediniz.
21. Soru 17(c)'nin her bir reel çözümünü uygun başlangıç değerleri ve Program 6.3 ile belirleyiniz.

6.5 Aralık üzerinde sıfıryerleri

6.5.1 Vektörel ikiye bölme yöntemi

Birinci bölümde incelenen klasik ikiye bölme yöntemi $[a, b]$ aralığında sürekli ve $f(a)f(b) < 0$ şartını sağlayan fonksiyonun söz konusu aralık içerisindeki tek bir sıfır yerini belirler. Ancak fonksiyon verilen aralığın uç noktalarında işaret değiştirmeyip, alt aralık veya aralıklarda işaret değiştirebilir. O halde verilen bir aralığın kullanıcı tarafından belirlenen uzunluktaki alt aralıkları taranarak işaret değişiminin gerçekleştiği alt aralıklar belirlenebilir. Bu bölümde amacımız

- fonksiyonun işaret değiştirdiği alt aralıkları bulmak,
- klasik ikiye bölme yöntemini eş zamanlı olarak her bir alt aralığa uygulayarak verilen keyfi bir $[a, b]$ aralığında yer alan tüm reel sıfıryerlerini belirlemektir.

Öncelikle birinci amaca yönelik aşağıdaki algoritmayı dikkate alalım: Algoritma 6.1 için geliştirdiğimiz Program 6.4 aşağıda verilmektedir.

ÖRNEK 6.9. Program 6.4 ile $f(x) = \cos(6 \cos^{-1}(x))$ Chebyshev⁴ fonksiyonunun $[-1, 1]$ aralığında işaret değiştirdiği 0.1 uzunluklu alt aralıkları belirleyiniz.

>> $f(x) = \sin(5x)$ fonksiyonunun $[0, 4]$ aralığı içerisinde sıfıryerlerini içeren $dx = 0.1$ uzunluklu alt aralıkları belirleyiniz.

```
>> [A,B,sifirlar]=ikibolaralik(f,0,4,0.1)
A =
0.6000 1.2000 1.8000 2.5000 3.1000 3.7000
B =
```

⁴Pafnuty Lvovich Chebyshev, 1821-1894, Rus matematikçi

Algoritma 6.1 İşaret değişim aralıklar algoritması

1. input: f, a, b, dx
2. A: aralık sol uç noktalar vektörünü boş küme olarak tanımla.
3. B: aralık sağ uç noktalar vektörünü boş küme olarak tanımla.
4. Tesadüfen bulunabilen sıfıryerleri saklamak için sıfırlar isimli boş bir küme tanımla.
5. sonuçlandırma kriteri olarak kullanılan test değişkenine 1 değerini ata.
6. Süreksizlik kontrolü için *Maxzipla* değişkenine 5 değerini ata.
7. test=1 olduğu sürece
 - (a) $f(a)=0$ ise a yı sıfırlar dizisine aktar.
 - (b) $f(a+dx)=0$ ise $a+dx$ i sıfırlar dizisine aktar.
 - (c) $f(a)f(a+dx)<0$ ve $|f(a+dx)-f(a)|<Maxzipla$ ise a yı A dizisine ve $a + dx$ i de B dizisine aktar.
 - (d) $a= a+dx$ olarak al.
 - (e) $a \geq b$ ise test=0 olarak al.
8. A ve B dizileri ile sıfırlar dizisini çağıran programa geri gönder.

0.7000 1.3000 1.9000 2.6000 3.2000 3.8000

sifirlar = 0

O halde verilen fonksiyonun

$$[0.6, 0.7], [1.2, 1.3], [1.8, 1.9], [2.5, 2.6], [3.1, 3.2], [3.7, 3.8]$$

aralıklarının herbirinde enaz bir sıfır yeri vardır. Ayrıca $x = 0$ fonksiyonun bir sıfır yeridir.

İkiye bölme yöntemini eş zamanlı olarak yukarıda verilen alt aralıklara uygulayarak, herbir alt aralıktaki sıfır yerini belirlemek istiyoruz. Bu amaçla aşağıdaki algoritmayı uygulayalım

ÖRNEK 6.10. $f(x) = \sin(5x)$ fonksiyonunun $[0, 4]$ aralığında yeralan sı-

Algoritma 6.2 Vektörel ikiye bölme yöntemi algoritması

1. input f, a, b, dx
 2. Program 6.4 yi çağırarak suretiyle AB ile gösterilen aralık uç nokta vektörleri ile tesadüfen bulunabilen *sifirlar* vektörünü oluştur.
 3. *sifirlar* vektörünü X ile gösterilen vektöre aktaralım. $A = AB(:, 1); B = AB(:, 2)$ olarak tanımla.
 4. A ve X boş küme ise işaret değişim aralığı belirlenemediği için programı sonlandır.
 5. A boş küme ise X de varsa tesadüfi değerleri sıfıryeri olarak geri gönder.
 6. *fark* değişken değerini 1, *eps* sonuçlandırma kriteri değerini $1e - 5$ olarak kabul et.
 7. *fark* değişkeninin sonsuz normu *eps* den büyük olduğu sürece aşağıdakileri tekrarla:
 - (a) $C = (A + B)/2$ orta noktalar vektörünü hesapla.
 - (b) $f(A) \cdot f(C) < 0$ eşitsizliğini sağlayan A ve C vektörlerinin *ii* ile gösterilen indislerini belirle.
 - (c) Eğer *ii* indis kümesi boştan farklı ise $B(ii) = C(ii)$ olarak değişir.
 - (d) $f(A) \cdot f(C) \geq 0$ eşitsizliğini sağlayan A ve C vektörlerinin *jj* ile gösterilen indislerini belirle.
 - (e) Eğer *jj* indis kümesi boştan farklı ise $A(jj) = C(jj)$ olarak değiştir.
 - (f) $fark = |B - A|$ vektörünü tanımla.
 - (g) $fark \leq eps$ veya $|f(C)| < eps$ özelliğini sağlayan j_0 yakınsak aralık indislerini belirle.
 - (h) $X = C(j_0)$ ile elde edilen sıfıryerlerini X vektörüne ata.
 - (i) $fark > eps$ özelliğini sağlayan henüz yakınsamayan alt aralıkların j_1 indislerini belirle.
 - (j) Eğer j_1 boştan farklı ise $A = A(j_1)$ ve $B = B(j_1)$ olarak değiştir.
-

```

% Sıfıryerlerini içeren aralık uc noktalar vektörü
% ve tesadüfen bulunan sıfıryerlerini
% çağıran programa gönderir.
% A: sol uç noktalar
% B: sağ uç noktalar
%-----
function [A,B,sifirlar]=ikibolaralik(f,a,b,dx)
test=1;
A=[];B=[];sifirlar=[];
while test
    if f(a)==0
        sifirlar=[sifirlar;a];
    end
    if f(a+dx)==0
        sifirlar=[sifirlar;a+dx];
    end
    if f(a)*f(a+dx)<0
        A=[A;a];B=[B;a+dx];
    end
    a=a+dx;
    if a>=b test=0;
end
end

```

Program 6.4: İşaret değişim aralıklar uygulaması

fıryerlerini vektörel ikiye bölme yöntemi yardımıyla belirleyiniz.

```
>> ikibolv(f,0,4)
```

0	0.628318530717962	1.256637061435919
1.884955592153875	2.513274122871833	3.141592653589795
...
8.168140899333460	8.796459430051419	9.424777960769381

Yukarıdaki tabloda verilen sonuçlarla gerçek değerler karşılaştırıldığında, elde edilen yaklaşımların virgülden sonra ondört basamağa kadar doğru olduğu anlaşılmaktadır.

ÖRNEK 6.11. $f(x) = \tan(x)$ fonksiyonunun $[0, 10]$ aralığında yeralan sıfır yerlerini yöntemi yardımıyla belirleyiniz.

```
>> ikibolv(f,0,10)
```

```
ans = 0 3.141592653589784 6.283185307179581 9.424777960769376
```

elde ederiz.

İkiye bölme yönteminin bilinen skaler versiyonu verilen aralığın uç noktalarında fonksiyonun işaret değiştirmiş olmasını gerektirirken, yukarıda incelenen vektörel versiyon da fonksiyonun işaret değiştirdiği alt aralıkları belirleyerek, her bir alt aralıktaki sıfır yerini bulur. Ancak, fonksiyonun katlı sıfır yerlerini içeren aralıklarda işaret değiştirmeyebileceğini biliyoruz. Örneğin $f(x) = x^2$ fonksiyonu $[-1, 1]$ aralığında işaret değiştirmedeği halde $x = 0$ noktasında iki katlı sıfır yerine sahiptir. Öte yandan $f(x) = x^3$ fonksiyonu da $x = 0$ da üç katlı sıfır yerine sahiptir ve aynı aralıkta işaret değiştirir.

Bu amaçla aralık uç noktalarında işaret değiştirmeyi gerektirmeyen iteratif yöntemlerin vektörel versiyonları çalışılabilir.

6.5.2 Vektörel Newton Yöntemi

Sabit nokta iterasyon yöntemleri bir disk içerisinde yer alan tüm sıfır yerlerini (reel ve kompleks) bulacak biçimde geliştirilebilirler. Özel olarak *Newton* yöntemi *MATLAB/Octave* vektör cebiri kullanılarak bu amaçla geliştirilebilir. Reel sıfır yerleri için uygun başlangıç vektör tahmini ile [6] yüksek lisans tez çalışmasında sunulmuştur.

Bu bölümde [6] de geliştirilen reel sıfır yerleri algoritmasını, sıfır merkezli diskteki tüm sıfır yerlerini belirleyecek biçimde geliştirerek Algoritma 6.3 ile aşağıda veriyoruz.

Algoritma 6.3 ya ait Program 6.5 bölüm alıştırmaları sonunda verilmektedir.

ÖRNEK 6.12. $f(x) = x^3 - 1, x^5 - 1, x^7 - 1, x^9 - 1$ fonksiyonlarının tüm sıfır yerlerini yukarıdaki algoritma yardımıyla belirleyerek, sıfır yerlerini birim çember üzerinde gösteriniz.

Çözüm.

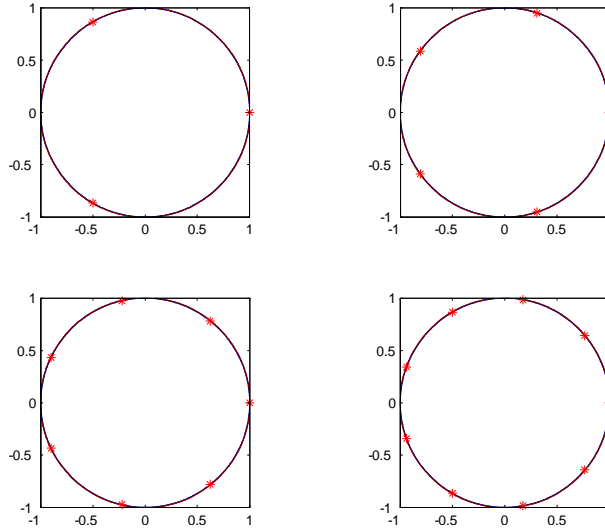
Sırasıyla herbir fonksiyon için aşağıdaki program parçasını uygun grafik penceresinde çalıştıralım.

```

x=-1:0.01:1;
y=x;
[X,Y]=meshgrid(x,y);
Z=X.^2+Y.^2-1;
subplot(2,2,1);
contour(X,Y,Z,[0 0.01]);
hold on;
f=inline('x.^3-1');
df=inline('3*x.^2');
Xff=cvnewton(f,df,2,1);
plot(Xff,'*r');axis('square');

```

Elde ettiğimiz ekran çıktısı Şekil 6.5 te sunulmaktadır .



Şekil 6.5: $x^n - 1$ in $n = 3, 5, 7$ ve 9 için sıfıryerleri.

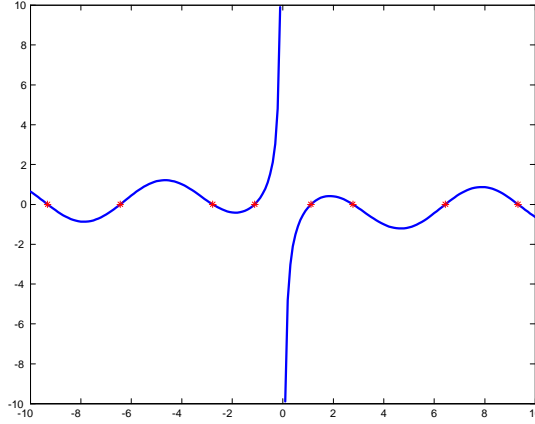
ÖRNEK 6.13. $f(x) = \sin(x) - 1/x$ fonksiyonunun $[-10, 10]$ aralığındaki sıfıryerlerini Program 6.5 ile belirleyiniz.

Çözüm. `>> f=inline('sin(x)-1./x')`

`>> df=inline('cos(x)+1./x.^2')`

`>> Xff=cvNewton(f,df,10,1)`

Xff = -9.3172 -6.4391 -2.7726 -1.1142 1.1142 2.7726 6.4391 9.3172
 Elde edilen sıfıryerleri ve fonksiyon grafiği Şekil 6.6 de gösterilmektedir.



Şekil 6.6: $f(x) = \sin(x) - 1/x$ fonksiyonu grafiği ve sıfıryerleri(*)

sıfır yerini içeren aralıkta fonksiyonun işaret değiştirmemesi durumunda vektörel ikiye bölme yöntemi yerine vektörel *Newton* yönteminin kullanılması uygundur. Aşağıdaki örnekte fonksiyonun sıfıryerlerinde işaret değiştirmedine dikkat ediniz.

ÖRNEK 6.14. $f(x) = \sec(x) - 1$ fonksiyonunun $[-10, 10]$ aralığındaki sıfıryerlerini Program 6.5 ile belirleyiniz.

Çözüm. `>>f=inline('sec(x)-1')`

`>>df=inline('sec(x).*tan(x)')`

`>>Xff=cvNewton(f,df,10)`

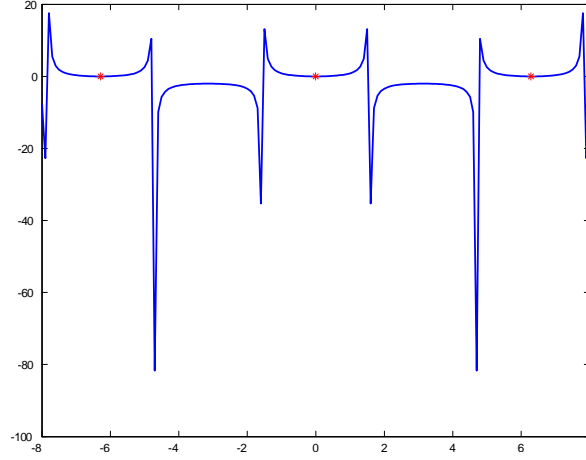
`>>Xff=-6.2832 -0.0000 6.2832`

Alıştırmalar 6.2.

1. Vektörel ikiye bölme yöntemine ait algoritma 6.2 için uygun program hazırlayarak aşağıda verilen fonksiyonların karşılarında belirtilen aralıklardaki sıfıryerlerini belirleyiniz.

(a) $f(x) = x^3 + 3x^2 - 2x - 1, [-5, 5]$

(b) $f(x) = \cos(x) - 1/x^2, [-8, 8]$

Şekil 6.7: $\sec(x)-1$ fonksiyonu ve sıfıryerleri

(c) $f(x) = e^{-x^2} - \sin(x), [-7, 7]$

(d) $f(x) = \tan(x) - \cos(x), [-10, 10]$

2. Yukarıda verilen 6.3 algoritmasına ait *cvNewton* isimli Program 6.5 aşağıda verilmiştir. *cvNewton* programını yukarıda bölüm sonunda verilen çözümlü örnekler üzerinde test yapınız. Aynı sonuçları elde ediyormusunuz?
3. Soru 2 de incelediğiniz programı, Soru 1 de verilen örnekler üzerinde de test yapınız. Vektörel ikiyebölme yöntemi ile aynı sonuçları elde ediyormusunuz?
4. Proje(Polinomlar için Vektörel Newton Yöntemi): Fonksiyon ve türevini kullanıcıdan alarak belirtilen yarıçaplı disk içerisindeki tüm sıfıryerlerini belirleyen 6.3 algoritmasını, verilen polinomun tüm sıfıryerlerini bulacak biçimde ve *CVNewtonp* ismiyle geliştiriniz. Kullanıcının sadece polinomun katsayılar vektörünün girmesi yeterli olmalıdır. (İpucu: Öncelikle verilen bilgidен hareketle sıfıryerlerini içeren disk yarıçapını ilgili teoriyi araştırarak tahmin ediniz. Ayrıca türev polinomunun katsayılar vektörünü, verilen polinom katsayılar vektörü yardımıyla belirleyiniz.) Test sonuçlarınızı MATLAB/Octave roots fonksiyonu ile elde edebileceğiniz sonuçlarla karşılaştırınız.

Algoritma 6.3 Disk içersindeki tüm kökler için Vektörel Newton Algoritması

1. input: $f, df, r, secenek, dx, secenek = 0$: reel sıfıryerleri, $secenek = 1$: tüm sıfıryerleri
2. varsayılan değerler: $secenek = 0, dx = 0.1$;
3. Reel sıfıryerler için $X0 = -r : dx : r$;
4. Tüm sıfıryerleri için $X = [-r : dx : r]x[-r : dx : r]; Y = X; Z = X + iY$;
5. Z matrisinin satırlarını uç uca ekleyerek $X0$ başlangıç vektörü oluştur;
6. $Xf \leftarrow f(X0) = 0$;
7. $sayac = 0; Maxsayac = 15; carp = 1e10; eps = 1E - 14$;
8. $Maxsayac$ değerine ulaşılan kadar aşağıdakileri tekrarla;
 - (a) $df(X0)$ vektörünü hesapla ve $df(X0) \neq 0$ olan $i0$ indislerini belirle;
 - (b) $X0 \leftarrow X0(i0)$;
 - (c) $X1 \leftarrow X0 - f(X0)./df(X0)$;
 - (d) $|X1| < r$ eşitsizliğini sağlayan ii indislerini belirle;
 - (e) $X0 \leftarrow X0(ii), X1 \leftarrow X1(ii)$;
 - (f) $fark \leftarrow |X1 - X0|$;
 - (g) $fark \leq eps$ eşitsizliğini sağlayan $j0$ indislerini belirle;
 - (h) $fark > eps$ eşitsizliğini sağlayan $j1$ indislerini belirle;
 - (i) $sayac \leftarrow sayac + 1$;
 - (j) $Xf \leftarrow X0(j0)$, yakınsayan bileşenleri Xf vektörüne yığ;
 - (k) $X0 \leftarrow X1(j1)$, yakınsamayan bileşenleri güncel başlangıç vektör al;
 - (l) $X0$ boş ise mevcut Xff i kullanıcıya ilet;
 - (m) $secenek = 1$ ise
 - i. reel ve sanal kısımları virgülden sonra on basamağa kadar yuvarla;
 - ii. farklı bileşenleri belirle ve Xff vektöründe biriktir.
 - (n) $secenek = 0$ ise
 - i. yakınsak bileşenleri Xff vektöründe biriktir;
 - ii. Xff vektörünü sırala;
 - iii. ardışık elemanlar arasındaki farkı dx den büyük olanları farklı kabul et.


```

function Xff=cvnewton(f,df,r,secenek,dx)
%-----
% Vektörel Newton B(0,r) deki kompleks/reel kökleri bulur
% secenek girilmezse reel sıfır yerlerini bulur.
% secenek=1 ise tüm sıfır yerlerini bulur.
% ec, Mayıs, 2015
% Bu program [7] de verilen programın geliştirilmiş versiyonudur.
%-----

carp=1e6; %kompleks kökler için yuvarlama parametresi
sayac=0; %sayac değişkeni
eps=1E-14; %yakınsama kontrol parameteresi
maxsayac=25; % maximum iterasyon sayısı

if nargin==3 secenek=0;dx=0.1;
end;
if nargin==4 dx=0.1;
end
if secenek==0
    X0=-r:dx:r;X0=X0'; %reel sıfır yerleri için başlangıç
elseif secenek==1 % tahmin vektörü
    x=-r:dx:r;y=j*x;n=length(x);n2=n*n;
    [X,Y]=meshgrid(x,y);
    Z=X+Y;
    X0=reshape(Z,n2,1); % tüm sıfıryerleri için başlangıç
    % tahmin vektörü(reel ve kompleks)
else error('kullanım: Xf=cvnewton(f,df,r,secenek,dx)');
end
test=sayac<maxsayac; % döngü değişkeninin ilk değeri 1 e eşit
ii=find(f(X0)==0); % tesadüfen bulunabilen sıfır yerleri
Xf=X0(ii); % sıfır yerlerini saklayan vektörel değişken
while test
    df=df(X0); %türevin X0 daki değerler vektörü
    i0=find(df0~=0); % türevin sıfırdan farklı olduğu indisler
    X0=X0(i0); %türevi sıfırdan farklı başlangıç değerleri
    X1=X0-f(X0)./df(X0); %Vektörel Newton adımı

    ii=find(abs(X1)<r); % r yarıçaplı bölgedeki sıfır yerlerinin indisleri
    X0=X0(ii);
    X1=X1(ii);
    fark=abs(X1-X0); %yaklaşımlar arası fark
    j0=find(fark<=eps); %yakınsayan bileşenlerin indisleri
    j1=find(fark>eps); %yakınsamayanlarKaradeniz Teknik Matematik, erhan@ktu.edu.tr
    sayac=sayac+1; %döngü değişken değer artırımı
if length(j0)>0
    Xf=[Xf;X1(j0)]; % yakınsayan bileşenleri X vektörüne diz
end
end

```

```

if length(j1)>0
    X0=X1(j1);          % yakınsamayan bileşenlerle devam et
end

if isempty(X0)
    break; %bölgede yakınsamayan bileşen yoksa döngüden çık
end
test=sayac<maxsayac; % maxsayac ulaşılanaya kadar işlemleri tekrarla
end
if secenek==1
    Xfr=round(carp*real(Xf))/carp; %reel kısımları yuvarla
    Xfs=round(carp*imag(Xf))/carp; % sanal kısımları yuvarla
    Xfy=round(carp*Xfr)/carp+i*round(carp*Xfs)/carp; %yuvarlanmış degerler
    Xff=unique(Xfy); %farklı olan yuvarlanmış degerleri belirle
else
    ii=find(abs(f(Xf))<1e-5); % yakınsak bileşen indisleri
    if length(ii)>0
        Xf=Xf(ii);
        Xf=sort(Xf); %sırala
        Xff(1)=Xf(1);n=length(Xf)-1;
        for ii=2:n
            if abs(Xf(ii+1)-Xf(ii))>dx
                Xff=[Xff;Xf(ii+1)]; %farklı olanları seç
            end
        end
    else display('reel kok yok');
    end
end
end

```

Program 6.5: Geliştirilmiş Vektörel Newton yöntemi uygulaması

Kaynaklar

- [1] Atkinson, K. An Introduction to Numerical Analysis, John Wiley & Sons, 1988.
- [2] Coşkun, E. MATLAB/Octave ile Sayısal Hesaplama ve Kodlama(URL:aves.ktu.edu.tr/erhan/dokumanlar).
- [3] Edwards & Penney, çeviri ed. Akın, Ö., Diferensiyel Denklemler ve Sınır Değer Problemleri, Palme Yayıncılık, 2006.
- [4] Hildebrand, F. B., Introduction to Numerical Analysis, Dover Publications, Inc., 1987.
- [5] Kincaid, D., Cheney, W., Numerical Analysis, Brooks/Cole, 1991.
- [6] Memoğlu, M., Vektörel sıfır yeri ve ekstremum nokta belirleme algoritmaları, Yüksek Lisans Tez Çalışması, KTÜ, 2012.
- [7] Octave, GNU özgür yazılım(*URL:Octave.sourceforge.net*).
- [8] Press, H. W. ve ark., Numerical Recipes in C, Cambridge University Press, 1988.
- [9] Stoer, J., Bulirsh, R., Introduction to Numerical Analysis, Springer-Verlag, 1976.
- [10] S. W., Warren, Zill, D. G., Calculus: Early Transcendentals, Çeviri: Matematik Cilt I, II(Çeviri editörü İsmail Naci Cangül), Nobel Akademik Yayıncılı