Bölüm 1

Sayısal Analiz Süreci

Bu bölümde matematiksel analiz yöntemleri olarak bilinen *analitik*, *sayısal*, *kalitatif* ve *sembolik analiz* yöntemlerini kısaca tanıtarak, *sayısal analiz* yönteminin bütün aşamalarını üç elemanter problem üzerinde inceliyoruz.

1.1 Matematiksel analiz yöntemleri

Çözümü matematiksel yöntemlerin kullanılmasını gerektiren bir problemin, ilgili matematiksel yöntemler yardımıyla incelenmesi veya diğer bir ifade ile "analiz" edilmesi *matematiksel analiz* olarak adlandırılmaktadır. Herhangi bir matematiksel problem *analitik, sayısal, kalitatif* ve *sembolik analiz* adı verilen analiz türlerinden birisi veya birden fazlası yardımıyla analiz edilebilir.

Analitik analiz, verilen problemi

- teorik olarak geliştirilmiş matematiksel formülasyon veya argümanlar yardımıyla ve
- kağıt-kalem gibi temel araçları kullanarak irdeleme sürecidir.

Analitik analiz yöntemi, verilen problem için akla gelen yöntemdir, ancak her problem için ve özellikle son yüzyılda uygulamalı matematik kapsamında incelenmesi gereken problemlerin çözümü veya analizi için yeterli değildir.

Konuya ışık tutması açısından aşağıda sunulan ve bir kısmı ortaöğretim matematik ders müfredatlarında ve diğer bir kısmı ise elemanter lisans derslerinde incelenen temel konulardan birkaçına göz atalım:

1. $a, b, c \in R$ ve $a \neq 0$ olmak üzere $ax^2+bx+c=0$ denkleminin köklerinin belirlenmesi,

2. $(x_0, y_0), (x_1, y_1)$ noktalarından geçen doğru denkleminin belirlenmesi,

3. $a_{ij} \in R, b_i \in R$ olmak üzere

$$\begin{aligned} a_{11}x + a_{12}y &= b_1 \\ a_{21}x + a_{22}y &= b_2 \end{aligned}$$

   denklem sisteminin çözümü,

4. $\int_0^1 \sin(x)dx$ integralinin hesaplanması ve

5. $y' = t - y, t \in (a, b), y(a) = y_0$ başlangıç değer probleminin çözümünün belirlenmesi problemini göz önüne alalım.

Birinci problem, üzerinde uzun yıllar öncesinden beri çalışmış bir problemdir.

$$3x^2 + 5x - 1 = 0$$

gibi bazı özel denklem türlerinin çözümü için geometrik yaklaşımlar 800 lü yılların başında Harezmi[1] tarafından geliştirilmiştir. Katsayıları verilen en genel ikinci dereceden polinom denklemin çözümünü veren

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

formülü(analitik yöntemi) mevcuttur ve bu formül kullanılarak problemin kağıt-kalem çözümü(kağıt üzerinde ve kalem yardımıyla) elde edilebilir. O halde bu problemin analitik yöntemle analizi mümkündür. Ancak şimdi de

$$ax^3 + bx^2 + cx + d = 0$$

veya

$$ax^4 + bx^3 + cx^2 + dx + e = 0$$

[1] 780(Horasan)-850(Bağdat), Cebir ismi, yazmış olduğu 'El-Kitab 'ül Muhtasar(özet) fi 'l Hesab 'ül cebri ve 'l Mukabele(karşılaştırma ve denkleştirme)' isimli kitaptaki ül-cebri(Al Cebri ->algebra) sözcüğünden türetilmiştir[8].

denklemlerinin köklerini bulmaya çalışalım. Bu durumda 3. veya 4. dereceden polinomların köklerini bulmamız gerekmektedir. Söz konusu denklem köklerinin belirlenmesi için de Hayyam [2] ve/veya Cardano[3]ya atfedilen ve köklü ifadeler içeren karmaşık formüller mevcuttur ve bu formüller kağıt üzerinde ve kalem yardımıyla ilgili köklerin belirlenmesi için genelde kullanılamazlar. Bununla beraber derecesi beşe eşit veya beşten daha büyük olan en genel bir polinomun köklerinin belirlenmesi için daha düşük dereceli polinom denklemlerine benzer olarak köklü ifadeler içeren hiç bir formülasyonun verilemeyeceğini ise Abel[4] ve Galois[5] isimli matematikçilerin çalışmalarından biliyoruz.

Öte yandan *transandant* fonksiyon olarak bilinen üstel, logaritmik veya trigonometrik fonksiyonlar içeren denklemlerin çözümü için herhangi bir genel formül söz konusu değildir. Örneğin

$$e^x - (x+4) = 0$$

denklemini sağlayan ve birisi negatif($x \doteq -3.9813$) diğeri ise pozitif($x \doteq 1.7490$) olan x değerlerini belirlemek için bir formül mevcut değildir. En genel halde $f(x) = 0$ denkleminin sıfır yerlerini belirleme problemini, bu bölümün ilerleyen kısımlarında sayısal analiz sürecine örnek olarak ve ayrıca 6. Bölümde ise daha kapsamlı olarak inceleyeceğiz.

*İkinci problem*e baktığımızda $x_0 \neq x_1$ için (x_0, y_0),(x_1, y_1) noktalarından geçen doğru denkleminin

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$$

bağıntısıyla verildiğini hatırlayalım. Bu bağıntı yardımıyla

$$r \in (\min(x_i), \max(x_i)), i = 0, 1$$

noktasındaki bilinmeyen bir değer tahmin edilebilir. Ancak en genel halde, verilen keyfi sayıdaki(örneğin $n \geq 3$)

$$(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$$

[2] *Ömer Hayyam(1048-1131, İranlı bilim adamı)*

[3] *Girolamo Cardano (1501-1578, İtalyan bilim adamı)*

[4] Abel, Niels Henrik(1802-1829, Norveçli bilim adamı).

[5] Galois, Evariste(1811-1832). Galois gruplar teorisini geliştiren ve genç yaşta yaşamını yitiren Fransız matematikçi.

nokta kümesini esas alarak, istenilen bir

$$r \in (\min(x_i), \max(x_i)), r \neq x_i, i = 0, 1, ..., n$$

noktasındaki değeri tahmin etme işlemi olarak bilinen *interpolasyon* işlemi için analitik yöntem mevcut olsa bile yöntemin elektronik ortamda uygulanması gerekir. Söz konusu problemi 4. Bölümde inceleyeceğiz.

Benzer biçimde verilen keyfi sayıda nokta çiftine yakın noktalardan geçen en genel biçimdeki uygun bir eğrinin belirlenmesi problemi(*eğri uydurma*) analitik olarak incelenebilecek problem değildir, ve bu problemi 5. Bölümde inceleyeceğiz.

Üçüncü problem lineer bir denklem sistemidir ve çözümü yok etme yöntemi adını verilen bir analitik yöntemle kolayca belirlenebilir. Ancak denklem ve bilinmeyen sayısının artması durumunda ilgili sistemin "*kağıt-kalem*" çözümü aşırı dikkat ve zaman gerektirir ve hatta belirli bir noktadan sonra ise mümkün olmaz. Ancak, özellikle mühendislikte bir çok araştırma problemi, binler, yüzbinler ve hatta milyonlarca bilinmeyenli lineer sistemlerin çözümünü gerektirmektedir. 7. Bölümde lineer denklem sistemleri için sıkça kullanılan sayısal yöntemleri inceleyeceğiz.

*Dördüncü problem*e gelince

$$\int_0^1 \sin(x)dx = 1 - \cos(1)$$

olduğunu kolayca görebiliriz, çünkü **'integrand'** olarak adlandırılan

$$f(x) = sin(x)$$

fonksiyonunun belirsiz integrali mevcuttur. Kalkülüs'ün Temel Teoremi[9] yardımıyla söz konusu belirli integrali hesaplayabiliriz. Ancak şimdi de

$$\int_0^1 \sin(x^2)dx$$

intregralini göz önüne alalım. Türevi $\sin(x^2)$ fonksiyonuna eşit olan ve sonlu sayıda elemanter fonksiyonunun lineer bileşimi biçiminde ifade edilebilen hiç bir fonksiyon bulamayız. Bu ve benzeri bir çok fonksiyonun sonlu sayıda elemanter fonksiyonun lineer bileşimi biçiminde ifade edilebilen belirsiz integrali mevcut değildir. O halde bu problem için de analitik yöntemi uygun bir yöntem değildir. 8. Bölümde sıkça kullanılan sayısal integrasyon yöntemlerini inceleyeceğiz.

Son olarak *beşinci problem* birinci basamaktan sabit katsayılı bir başlangıç değer problemidir ve çözümü

$$y = t - 1 + (1 - a + y_0)e^{(a-t)}$$

olarak elde edilir. Ancak şimdi de

$$\begin{aligned} y\prime &= t - y^2, t \in (a, b) \\ y(a) &= y_0 \end{aligned}$$

başlangıç değer problemini göz önüne alalım. Bilinen analitik yöntemlerle problemi çözemeyiz! Analitik yöntemler çok özel denklem türleri(lineer, değişkenlerine ayrılabilen, homojen, vb) için geçerlidirler ve en genel bir Bayağı Diferensiyel Denklem için uygun değildir. Benzer durum Kısmi Diferensiyel Denklemler için de geçerlidir. Bu amaçla geliştirilen ve *sonlu fark yöntemleri* adı verilen sayısal çözüm yöntemlerini, "*Diferensiyel Denklemler için Sonlu Fark Yöntemleri[4]*" isimli çalışmada inceliyoruz.

Yukarıda göz önüne aldığımız problemlerin bir kısmında veri sayısının artması ve diğer bir kısmında ise problem formülasyonununda yapılan ufak değişiklikten sonra analitik yöntemlerin artık mümkün olmadığını gözlemliyoruz. Bu durumda sıkça başvurulan yöntemler *sayısal analiz* yöntemleridir.

Sayısal analiz, genellikle analitik yöntemle analizin mümkün olmadığı veya bu yöntemle elde edilen çözümün kullanışlı olmadığı durumlarda bilgisayar yardımıyla problemin gerçek ya da genellikle yaklaşık çözümünü belirleme ve irdeleme(analiz etme) yöntemidir.

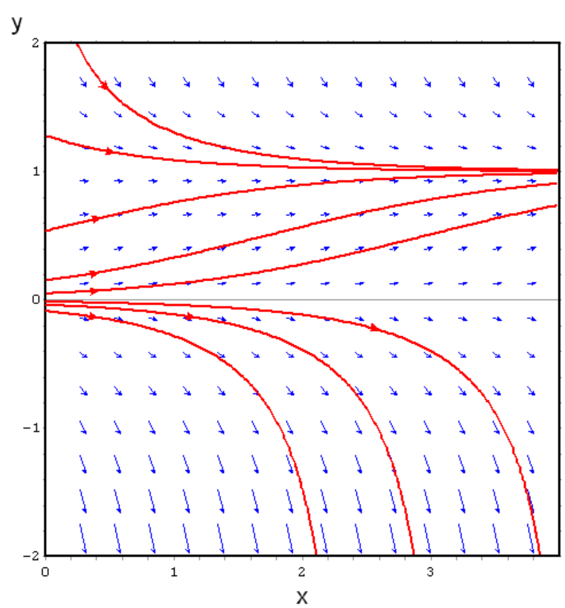
Genellikle bilgisayar ortamında gerçekleştirilebilen sayısal analiz yöntemleri, matematiğin doğuşu itibariyle kullanılmakta olan analitik yöntemlere kıyasla oldukça yeni bir yöntem türüdür ve özellikle XX. yüzyılın ikinci yarısında gelişen ve yaygınlaşan bilgisayar teknolojisine paralel olarak önem kazanmıştır. Kısaca sayısal yöntemler olarak adlandırılan sayısal analiz yöntemleri ise çok çeşitli araştırma alanlarında kullanılmakta olup, bu kaynağın esas konusunu teşkil etmektedir.

Öte yandan matematiksel literatürde *kalitatif analiz* ise çözümü belirlemeksizin(ya da belirleyemeksizin) çözüm hakkında bilgi edinmeyi sağlayan analiz yöntemidir. Bu yöntem de modern ve kısmen yeni bir analiz yöntemidir. Örneğin bir diferensiyel denklemi çözmeden yön alanları yardımıyla çözüm eğrilerinin davranışı belirlenebilir. Denge çözümler ve bu çözümlerin kararlılık analizleri ilgili problemlerin çözümünü belirlemeden gerçekleştirilebilir. Örneğin, $y = y(x)$ olmak üzere

$$y' = y(1 - y), y(0) = y_0$$

başlangıç değer problemini çözmeden çözüm eğrilerinin davranışını tahmin edebiliriz:

- $y_0 > 1$ için $y' < 0$ olup, çözüm eğrileri azalarak $y = 1$ asimtotuna teğet olur,
- $0 < y_0 < 1$ için $y' > 0$ olup, çözüm eğrileri artarak $y = 1$ asimtotuna teğet olur ve
- $y_0 < 1$ için $y' < 0$ olup, çözüm eğrileri artan x değerleri için azalır. Söz konusu yön alanları Şekil 1.1 de sunulmaktadır.



Şekil 1.1: Kalitatif analiz örneği: $y\prime = y(1 - y)$ denkleminin yön alanları ve bazı çözüm eğrileri.

Son olarak *sembolik analiz* yönteminden bahsedelim. Sembolik analiz, analitik analizi mümkün olan ancak daha çok işlem karmaşıklığına neden olan problemlerin analitik çözümlerinin bilgisayar cebir programı adı verilen programlar yardımıyla elde etme yöntemidir. Sembolik analiz, *Maxima* [[6],[3]], veya *MATLAB* [[5],[2]] *Sembolik Araç Kutusu* veya *Maple*, *Mathematica*, *Macysma* ve benzeri yazılımlar yardımıyla gerçekleştirilebilen ve kısmen yeni bir analiz yöntemidir.

Şekil 1.1 de verilen yön alanları *Maxima* yazılımının *plotdf* fonksiyonu yardımıyla elde edilmiştir:

$$plotdf(y * (1-y), [x, 0, 4], [y, -2, 2]);$$

Bu örnekte gösterilen yön alanları herhangi bir *yazılıma ihtiyaç duyulmadan da* elde edilebilir. Elde edilen yön alanlarına teğet olan çözüm eğrilerinin davranışı kolayca tahmin edilebilir.

Aşağıda verilen başlangıç değer problemini göz önüne alalım.

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

Verilen problemin analitik çözümünün

$$y = (x^2 - 2x + 2)/2 - e^{-x}$$

ile verildiği yukarıda bahsedilen yazılımlardan herhangi birisi yardımıyla kolayca görülebilir. Örneğin *Maxima* ortamında söz konusu problemin çözümünü veren komutlar aşağıda sunulmaktadır:

```
(%i1)      denklem:'diff(y,x,2)+'diff(y,x)=x;
```

(denklem) $\frac{d^2}{d x^2} y + \frac{d}{d x} y = x$

```
(%i2)      cozum:ode2(denklem,y,x);
```

(cozum) $y = \%k2 \, \%e^{-x} + \frac{x^2 - 2x + 2}{2} + \%k1$

```
(%i3)      ic2(cozum,x=0,y=0,'diff(y,x)=0);
```

(%o3) $y = \frac{x^2 - 2x + 2}{2} - \%e^{-x}$

MATLAB/OCTAVE ve *Maxima*'nın matematiksel amaçla kullanımını, özetle ve sırasıyla [2] ve [3] nolu kaynaklarda inceliyoruz.

1.2 Sayısal analiz süreci

Sayısal analiz süreci

- uygun bir matematiksel dille ifade edilmiş bir **problem** ile başlayıp,
- problemin çözümü için gerekli **sayısal yöntem**(ve mümkünse yakınsaklık analizi),
- söz konusu sayısal yöntemin uygulanabilmesi için, takip edilmesi gereken adımların kümesi olarak adlandırılan **algoritma**[6],
- algoritmanın uygun bir programlama diline dönüştürülmüş **kod**u veya diğer deyimle **program**ı,
- geliştirilen programın örnek problemler üzerinde **test** edilmesi(uygulama) ve
- sonuç, yorum, yöntemin kritiği ile mümkünse zayıf yönler için alternatif arayış aşamalarından oluşmaktadır.

Sayısal analiz aşamalarını öncelikle verilen bir nokta komşuluğunda sıfır yerini içeren aralığı belirleme problemi üzerinde inceleyelim:

1.2.1 Problem I: Verilen sürekli bir fonksiyonun, verilen bir nokta komşuluğunda reel sıfır yerini(eğer mevcutsa) içeren aralığı belirleme problemi

Verilen bir x_0 noktası komşuluğunda sürekli olan bir f fonksiyonunun, söz konusu komşulukta sıfır yerini içeren $[a, b]$ aralığını belirleme problemini göz önüne alalım.

[6] Algoritma sözcüğü El-Harizmi'nin isminin latince okunuşundan türetilmiştir[8].

1. **Problem ifadesi(sıfır yerini içeren aralık belirleme)** Reel sayıların uygun bir alt kümesi üzerinde tanımlı ve reel sıfır yeri olan sürekli bir f fonksiyonun verilen bir x_0 noktasının $(x_0 - R, x_0 + R)$,örneğin $R = 10$,komşuluğundaki sıfır yerini içeren $X = [a, b]$ aralığını belirleyiniz.

2. **Sayısal yöntem(***sağ veya sol yönde tarama***)** x_0 noktasını içeren uygun bir $[x_{\min} = x_0 - R, x_{\max} = x_0 + R]$ kümesine sıfır yeri tarama aralığı adı verelim. $x = x_0$ noktasından başlayarak önce sağa doğru, uygun bir $h > 0$ adım uzunluğu ile ilerleyelim: $x < x_{\max}$ olduğu sürece

$$f(x)f(x+h) <= 0$$

eşitsizliğini sağlayan ilk $(x, x+h)$ nokta çiftini belirleyelim. Bu durumda istenilen aralık $X = [x, x+h]$ dır.

Eğer belirtilen kriterleri sağlayan nokta çifti bulunamaz ise, bu durumda $x = x_0$ noktasından başlayarak, $x > x_{\min}$ olduğu sürece

$$f(x-h)f(x) \leq 0$$

eşitsizliğini sağlayan ilk $(x-h, x)$ nokta çiftini belirleyelim. Bu durumda $X = [x-h, x]$ dir.

Eğer sol yönde tarama işleminde de belirtilen kriteri sağlayan nokta çifti bulunamaz ise bu durumda $[x_{\min}, x_{\max}]$ tarama aralığı içerisinde sıfır yerini içeren alt aralık belirlenememiş olur.

3. **Algoritma** Yönteme ait Algoritma 1.1 aşağıda verilmektedir:

4. **Program(Kod)** Algoritma 1.1'e ait Program 1.1 aşağıda verilmektedir.

5. **Uygulama**

ÖRNEK 1.1. *$f(x) = exp(x) - x - 4$ fonksiyonunun $x_0 = 0$ noktası komşuluğundaki sıfır yerini içeren ve $h = 0.1$ uzunluklu $[a, b]$ aralığını belirleyiniz.*

Çözüm.

Algoritma 1.1 Sıfır yerini içeren aralık belirleme algoritması.

(a) Girdi : f, x_0

(b) Varsayılan parametreler:

i. $R = 10$ varsayılan tarama yarıçapı

ii. $x_{\min} = x_0 - R, x_{\max} = x_0 + R$ sıfır yeri tarama aralığı.

iii. $h = 0.1$ ardışık noktalar arası mesafe, $x = x_0$ ilk tahmini değer

(c) Sağ yönde tarama:

i. $x < x_{\max}$ olduğu sürece

A. eğer $f(x)f(x+h) \leq 0$ ise $X = [x, x+h]$ tanımla ve çık

B. değilse $x = x + h$ olarak tanımla ve c(i) ye git

(d) Sol yönde tarama

i. $x = x_0$

ii. $x > x_{\min}$ olduğu sürece

A. eğer $f(x-h)f(x) \leq 0$ ise $X = [x-h, x]$ tanımla ve çık

B. değilse $x = x - h$ olarak tanımla ve d(ii) ye git

(e) Sıfır yeri için tahmini aralık bulunamadı yaz, X=[] tanımla ve çık.

(f) Çıktı: X

```
>> f=@(x) exp(x)-x-4


>> X=bul(f,0)

X= 1.7000   1.8000
```

ÖRNEK 1.2. *$f(x) = \ln(x) - x + 4$ fonksiyonunun $x_0 = 10$ noktası komşuluğundaki sıfır yerini içeren ve $h = 0.1$ uzunluklu $[a, b]$ aralığını belirleyiniz.*

Çözüm.

```
>> f=@(x) log(x)-x+4
```

```
%----------------------------------------------------------------
function X=bul(f,x0)

         R=10;    %  varsayılan tarama bölgesi yarıcapı
 x_min=x0-R; x_max=x0+R;   % aralık uç noktaları
 h=0.1;   %tarama adım uzunluğu
                                x=x0; %  başlangıç noktası
 while x<x_max  %sağ yönde arama
   if f(x)*f(x+h)<=0 X=[ x,x+h]; return;
   else
     x=x+h;
   end;
 end

   x=x0; %  başlangıç noktası
  while x>x_min    %sol yönde arama
   if f(x-h)*f(x)<=0 X=[x-h,x]; return;
   else
   x=x-h;
   end;
  end
 disp('sifiryerini içeren aralik bulunamadi');X=[];
%--------------------------------------------------------------
```

Program 1.1: Verilen nokta komşuluğunda sıfır yerini içeren aralık belirleme uygulaması.

```
>> X=bul(f,10)

X=5.7000  5.8000
```

6. **Yöntemin analizi ve geliştirme önerileri**

   - Süreksiz fonksiyonlar için süreksizlik noktalarını içeren aralık yukarıda tanımlanan yöntem ile yanlışlıkla sıfır yeri olarak yorumlanabilir. Örneğin $f(x) = 1/x$ fonsiyonuna $x = 0$ noktasını içeren bir aralıkta yöntem uygulandığı taktirde bu tür bir yanlış sonuç oluşabilir. Yöntem *Sürekli Fonksiyonlar için Ara Değer Teoremi*ni (Teorem 1.1) esas aldığı için sadece sürekli fonksiyonlara uygulanabilir.

- Sürekli fonksiyonlar için ara değer teoremi gereği, yöntem sadece sıfır noktası komşuluğunda işaret değiştiren sürekli fonksiyonların sıfır yerlerini(yani tek katlı sıfır yerlerini) belirlemek amacıyla kullanılabilir, fakat $f(x) = x^2$ gibi çift katlı sıfır yerlerine sahip olan, yani sıfır yeri komşuluğunda işaret değiştirmeyen fonksiyonların sıfır yerlerinin belirlenmesinde kullanılamaz.

Problem II de ise verilen bir aralığın uç noktalarında işaret değiştiren fonksiyonun sıfır yerini belirleme problemi incelenmektedir.

1.2.2 Problem II: Verilen bir fonksiyonun, verilen bir aralıktaki sıfır yeri için yaklaşım belirleme problemi

1. **Problem ifadesi**

   f fonksiyonu $[a, b]$ aralığında tanımlı ve aralığının uç noktalarında işaret değiştiren (yani, $f(a)f(b) < 0$) sürekli bir fonksiyon olsun. Fonksiyonunun $[a, b]$ aralığındaki r sıfır yeri için yeterince küçük $\epsilon > 0$ ile $|f(c)| < \epsilon$ kriterini sağlayan $c \cong r$ yaklaşımını belirleyiniz.

   Söz konusu problemin çözümü mevcuttur. Aşağıda ifade edilen ve Sürekli Fonksiyonlar için ara değer Teoremi(Teorem 1.1) olarak bilinen teorem gereğince f fonksiyonunun $[a, b]$ aralığında en az bir reel sıfır yeri olduğunu biliyoruz. O halde f sürekli olduğu için $|f(c)| < \epsilon$ *kriterini sağlayan en az bir $c \cong r$* sıfır yeri yaklaşımı mevcuttur. Öncelikle sürekli fonksiyonlar için ara değer teoremini hatırlayalım:

   TEOREM 1.1. *(Sürekli fonksiyonlar için ara değer teoremi) f fonksiyonu $[a, b]$ kapalı aralığında sürekli ve*

   $$m := \min_{a \le x \le b} f(x); M := \max_{a \le x \le b} f(x)$$

   olarak tanımlansın. Bu taktirde $\forall\ s \in [m, M]$ için $f(r) = s$ denklemini sağlayan ve $[a, b]$ aralığı içerisinde bulunan en az bir r noktası mevcuttur.

Özetle ifade etmek gerekirse, sürekli fonksiyonlar için ara değer teoremi, sürekli bir fonksiyonun kapalı aralıktaki en küçük ve en büyük değer dahil, bu değerler arasında kalan her değeri en az bir noktada alacağını ifade etmektedir.

Sonuç 1.1. *Ara değer teoreminin bir sonucu olarak, eğer f fonksiyonu $[a, b]$ aralığında sürekli ve $f(a)f(b) < 0$ ise $f(r) = 0$ denklemini sağlayan ve (a, b) aralığı içerisinde yer alan en az bir r noktası mevcuttur.*

Çünkü $f(a)f(b) < 0$ olması durumunda sıfır noktası $f(a)$ ile $f(b)$ arasındadır, o halde f fonksiyonu altında sıfır noktasına resmedilen (a, b) aralığında en az bir nokta mevcut olmalıdır. Bu sonuca göre kapalı aralıkta sürekli olan ve bu aralığın uç noktalarında işaret değiştiren herhangi bir fonksiyonun söz konusu aralık içerisinde en az bir sıfır yeri mevcuttur.

Bir sonraki aşama ise söz konusu sıfır yerini elektronik ortamda belirleyecek olan ve *sayısal yöntem* olarak bilinen uygun bir matematiksel yöntemin belirlenmesidir.

2. **Sayısal Yöntem**(*ikiye bölme yöntemi*)

   Sayısal yöntem problemin elektronik ortamda çözümü için ne yapılması gerektiğini ifade eder.

   Bu amaçla ikiye bölme yöntemi adı verilen sayısal yöntemi kullanalım.

   Bu yöntem ile $[a, b]$ aralığı ile başlanarak, bu aralık her adımda iki eşit parçaya bölünür ve fonksiyonun işaret değiştirdiği alt aralık belirlenir. Aralık bölme işlemine fonksiyonun işaret değiştirdiği yeni aralık ile devam edilir. İşlemin ne zaman sonlandırılacağını *sonuçlandırma kriteri* adı verilen kriter belirler.

   Sonuçlandırma kriteri olarak, belirlenen yeni aralığın orta noktasındaki fonksiyon değerinin mutlak değerce verilen(veya tanımlanan) yeterince küçük bir $\epsilon > 0$ dan küçük olma kriteri olarak kabul edelim. Bu durumda elde edilen en son alt aralığın orta noktasını sıfır yeri için bir yaklaşım olarak kabul edebiliriz.

   Öte yandan elde edilen en son aralığın uzunluğunun yeterince küçük pozitif bir sayıdan küçük olması kriteri gibi daha farklı sonuçlandırma kriteri de kullanılabilir.

ÖRNEK 1.3. *$f(x) = x^2 - 1$ fonksiyonunun $[0, 3]$ aralığındaki sıfır yeri için ikiye bölme yöntemi ile ilk üç yaklaşımı elde ediniz.*

Çözüm.

- f fonksiyonu sürekli ve
$$f(0)f(3) = -8 < 0$$
olduğu için fonksiyonun verilen aralıkta bir sıfır yeri mevcuttur ve ikiye bölme yöntemi uygulanabilir. Sıfır yerini içeren ve $n-$inci adımda($n \geq 1$) elde edilen aralığı $[a_n, b_n]$ ve bu aralığın orta noktasını da c_n ile gösterelim. Buna göre ilk adımda
$$a_1 = 0, b_1 = 3, c_1 = 3/2$$
elde ederiz.
- $f(c_1) = 5/4 \neq 0$ olduğundan sıfır yeri henüz elde edilmemiştir, sıfır yerini içeren ikinci alt aralığı belirleyelim: Bu alt aralık ya
$$[a_1, c_1] = [0, 3/2] \text{ veya } [c_1, b_1] = [3/2, 3]$$
aralığı olmalıdır.
$$f(0)f(3/2) = -5/4 < 0$$
olduğu için yeni alt aralık
$$[a_2, b_2] = [0, 3/2] \text{ olup, bu aralığın orta noktası ise } c_2 = 3/4$$
tür.
- $f(c_2) = f(3/4) = -7/16 \neq 0$ olduğundan sıfır yeri henüz elde edilmemiştir. Sıfır yerini içeren üçüncü alt aralığı da benzer biçimde belirleyebiliriz: Bu alt aralık ya
$$[a_2, c_2] = [0, 3/4] \quad \text{veya } [c_2, b_2] = [3/4, 3/2]$$
aralığı olmalıdır.
$$f(3/4)f(3/2) = -35/64 < 0$$
olduğu için yeni alt aralık
$$[a_3, b_3] = [3/4, 3/2] \quad \text{ve bu aralığın orta noktası } c_3 = 9/8$$
dir.

- Dolayısıyla verilen fonksiyonun sıfır yeri için problemde istenilen ilk üç yaklaşımı

$$c_1 = 3/2, c_2 = 3/4, c_3 = 9/8$$

olarak elde etmiş olduk.

Aşağıda göreceğimiz üzere ikiye bölme yöntemi ile her bir adımda elde edilen aralığın orta noktası olarak tanımlanan $\{c_n\}$ dizisi fonksiyonun sıfır yerine yakınsamaktadır.

Ancak verilen keyfi bir fonksiyonun sıfır yerini belirlemek için gerekli işlemleri yukarıda olduğu gibi kağıt ve kalem ile gerçekleştiremeyiz. Bilgisayar ortamında gerçekleştirilmesi gereken bu işlemler için her bir adımda gerçekleştirilmesi gerekli olan işlemler açık ve net olarak ifade edilmelidir.

3. **Algoritma**

Sayısal yöntemin *hangi adımlar* takip edilerek ve *nasıl* uygulanacağını ifade eder. Daha açık bir ifade ile *Algoritma*

- Girdi(veya *input)* adı verilen verilerin alınması,
- yöntemin icrası için gerekli her bir adım ve
- Çıktı(veya Output) adı verilen ve kullanıcıya iletilecek sonuçların açık ve net bir biçimde ifade edildiği **komutlar kümesidir**.

Model problemimiz için *Algoritma* 1.2 aşağıda sunulmaktadır. Bu algoritmada her bir adımda elde edilen alt aralık yine $[a, b]$ aralığı olarak tanımlanmaktadır, *Problem I*' deki uygulamanın aksine burada indis kullanmıyoruz.

(a) adımında kullanıcının sıfır yerini belirlemek istediği fonksiyonu ve bu fonksiyonun işaret değiştirdiği aralığının a ve b ile gösterilen uç noktalarını tanımlaması istenmektedir. Ayrıca sıfır yeri için uygun bir yaklaşımın belirlendiğini test yapmak amacıyla yeterince küçük $\epsilon > 0$ sabitinin tanımlanması istenmektedir.

(b) de $f(a)f(b) > 0$ olması durumda yöntemin çalışmayacağı ifade edilmektedir.

Algoritma 1.2 Ikiye bölme yöntemi algoritması

(a) Girdi: f, a, b, ϵ

(b) Eğer $f(a)f(b) > 0$ ise çık,

(c) $c = (a+b)/2$ alt aralığın orta noktası olmak üzere $a, c, b, f(c)$ degerlerini yaz,

(d) $|f(c)| > \epsilon$ olduğu sürece i,ii,iii adımlarını tekrarla,

  i. Yeni alt aralığı belirle: eğer $f(a)f(c) < 0$ ise $b = c$, değilse $a = c$ (sıfır yerini içeren yeni $[a, b]$ aralığı)

  ii. Yeni alt aralığın orta noktası: $c = (a + b)/2$

  iii. Çıktı: $a, c, b, f(c)$ degerlerini yaz.

(c) de $[a, b]$ aralığının orta noktası belirlenerek c değişkenine atanmakta ve $a, c, b, f(c)$ değerleri yazdırılmaktadır.

(d) de $|f(c)| > \epsilon$ olduğu sürece

$(d) - (i)$ de sıfır yerini içeren alt aralık belirlenerek, bu alt aralık tekrar $[a, b]$ aralığı olarak adlandırılmaktadır. Eğer $f(a)f(c) < 0$ ise yeni $[a, b] = [a, c]$, yani $b = c$, değilse yeni $[a, b] = [c, b]$, yani $a = c$ dir.

$(d) - (ii)$ de yeni alt aralığın orta noktası belirlenmekte

$(d) - (iii)$ de ise $a, c, b, f(c)$ değerleri yazdırılmaktadır.

4. **Program(Kod)**

Bir sonraki aşama ise algoritması belirlenen probleme ait program geliştirme aşamasıdır. Bu aşama algoritmada belirtilen komutlar kümesinin Bilgisayar Dili'ne dönüştürülmesi aşamasıdır. Bu bağlamda *Programlama Dili* olarak adlandırılan uygun bir dil(*Basic, Pascal, Fortran, C*, vs) veya *Programlanabilme özelliğine sahip yazılım* kullanılır.

Uygulamalarımız için çoğunlukla *MATLAB* veya aynı yazım kurallarını kullanan *OCTAVE*'ı [2] kullanıyoruz. Yukarıdaki problem için *OCTAVE* programı aşağıda verilmektedir.

OCTAVE $(http://www.gnu.org/software/OCTAVE)$ adresinden kolayca erişilebilen ücretsiz bir yazılımdır ve matematiksel işlemler için kullanımını[2] nolu kaynakta özet olarak inceliyoruz.

```
%-----------------------------------------------------------

function c=ikibol(f,a,b)
% f fonksiyonunun  [a,b] aralığındaki
% sıfır yerini ikiye bölme yöntemi ile bulur.
%Yazılımı : c=ikibol(f,a,b)

        eps=1e-5;  %parametre
if f(a)*f(b)>0
   error("ikiye bolme yöntemi uygulanamaz");
end;

c=(a+b)/2;
fprintf('     a             c              b                f(c)  \n');
fprintf('%9.6f  %9.6f  %9.6f  %9.6f\n',a,c,b,f(c));
while  abs(f(c))>eps
if f(a)*f(c)< 0
   b=c;
else
  a=c;
end
  c=(a+b)/2;
 fprintf('%9.6f  %9.6f  %9.6f  %9.6f  \n',a,c,b,f(c));
 end

%---------------------------------------------------------
```

Program 1.2: Matlab veya Octave ile ikiye bölme yöntemi uygulaması.

5. **Uygulama**

   ÖRNEK 1.4. *$f(x) = cos(x) - x$ fonksiyonunun $[0, 2]$ aralığındaki sıfır yerini ikiye bölme yöntemi yardımıyla belirleyiniz.*

   Çözüm.

   f fonksiyonunu

   ```
   f=@(x)  cos(x)-x ;
   ```

komutu ile tanımlayalım. Daha sonra aşağıdaki komut yardımıyla Program 1.2 i çalıştırarak aşağıda gösterilen sonuçları elde edebiliriz.

```
>>ikibol(f,0,2)

 0.000000  1.000000  2.000000 -0.459698
 0.000000  0.500000  1.000000  0.377583
 0.500000  0.750000  1.000000 -0.018311
 0.500000  0.625000  0.750000  0.185963
 0.625000  0.687500  0.750000  0.085335
 0.687500  0.718750  0.750000  0.033879
 0.718750  0.734375  0.750000  0.007875
 0.734375  0.742188  0.750000 -0.005196
 0.734375  0.738281  0.742188  0.001345
 0.738281  0.740234  0.742188 -0.001924
 0.738281  0.739258  0.740234 -0.000289
 0.738281  0.738770  0.739258  0.000528
 0.738770  0.739014  0.739258  0.000120
 0.739014  0.739136  0.739258 -0.000085
 0.739014  0.739075  0.739136  0.000017
 0.739075  0.739105  0.739136 -0.000034
 0.739075  0.739090  0.739105 -0.000008

ans =

    0.7391
```

6. **Yöntemin Analizi**

Hazırlanan programın doğru çalıştığının kontrol edilmesini takip eden son aşama ise elde edilen sonuçların irdelenmesi ve yorumlanmasıdır. Bu bağlamda cevaplandırılması gereken sorular:

- Yöntemin söz konusu aralıktaki sıfır yerini her zaman belirleyip ya da belirleyemeyeceği(Teorem 1.2) ve
- sıfır yerini belirleyebilme hızı(orta noktalardan oluşan dizinin yakınsama hızıdır.

Bu sorulara vereceğimiz cevaplar yardımıyla *yöntemin yakınsaklığı, yakınsama hızı* ve *bilgisayar sistem kaynaklarını hangi düzeyde kullandığı* konusunda bilgi sahibi oluruz. Bu bilgiler *yöntemin "**kimlik**" bilgileridir* ve bir diğer yöntemle karşılaştırılırken kullanılır.

TEOREM 1.2. *f fonksiyonu $[a, b] = [a_1, b_1]$ aralığının uç noktalarında işaret değiştiren sürekli bir fonksiyon ve r de fonksiyonun her n için $f(a_n)f(b_n) < 0$ şartını sağlayan $[a_n, b_n]$ aralığındaki sıfır yeri ve $\{c_n\}_{n=1}^{\infty}$ ise $c_n = (a_n + b_n)/2$ ile tanımlanan orta noktalar dizisi olsun. Bu taktirde*

$$\lim_{n\to\infty} c_n = r$$

dir.

İspat[1] r sıfır yeri için

$$|r - c_n| \leq \frac{b_n - a_n}{2}$$

olduğu açıktır. Öte yandan her bir alt aralığın uzunluğu önceki alt aralığın uzunluğunun yarısına eşit olduğundan

$$|r - c_n| \leq \frac{b_n - a_n}{2} = \frac{b_{n-1} - a_{n-1}}{2^2} = \cdots = \frac{b_1 - a_1}{2^n}$$

elde ederiz. Yukarıdaki eşitsizlikten

$$\lim_{n\to\infty} |r - c_n| = 0$$

elde ederiz. Öte yandan

$$-|r - c_n| \leq r - c_n \leq |r - c_n|$$

eşitsizliği ve genel matematik dersinden bilinen *sıkıştırma teoreminden [9]* sonuç açıkça görülür.

Şimdi de yakınsama hızını belirlemek amacıyla "*yakınsama basamağı*" kavramını tanımlayalım.

TANIM 1.1. *Bir r noktasına yakınsayan $\{c_n\}_{n=0}^{\infty}$ dizisi verilmiş olsun. Eğer her $n \geq N$ için*

$$|r - c_{n+1}| \leq \alpha |r - c_n|^{\beta}$$

eşitsizliği sağlanacak biçimde $\alpha > 0, \beta \geq 1$ reel sayıları ve $N > 0$ tamsayısı mevcutsa $\{c_n\}_{n=1}^{\infty}$ dizisi β-ıncı basamaktan yakınsak bir dizidir denir. $\beta = 1$ olması durumunda yakınsama için $\alpha \in (0,1)$ olmalıdır ve bu durumda diziye lineer yakınsak *dizi adı verilir. $\beta = 2$ olması durumda ise diziye* kuadratik yakınsak *dizi adı verilir.*

- İkiye bölme yöntemi için

  $$|r - c_{n+1}| \leq \frac{b_{n+1} - a_{n+1}}{2} = \frac{1}{2}\frac{b_n - a_n}{2}$$

  ve

  $$|r - c_n| \leq \frac{b_n - a_n}{2}$$

  eşitsizliklerini karşılaştırarak

  $$|r - c_{n+1}| \cong \frac{1}{2}|r - c_n|$$

  elde ederiz. O halde yöntem lineer olarak yakınsaktır ve

  $$|r - c_{n+1}|/|r - c_n| \cong \frac{1}{2}$$

  sabiti ise *ortalama yakınsaklık oranı* olarak tanımlanır.

7. **Geliştirme önerileri ve alternatif yöntemler**

En iyi sayısal yöntem, gerçek sonucu en büyük hassasiyetle ve minimal bilgisayar bellek ve zaman kaynağı kullanımı ile elde eden yöntemdir. Alternatif olarak daha değişik yöntemler uygulanabilir:

- **Kirişle Bölme(Regula Falsi, false position) yöntemi***:* Genellikle *Regula Falsi* olarak bilinen yöntem, her adımda aralığı orta noktasından ikiye bölmek yerine, $f(a)f(b) < 0$ olmak üzere $(a, f(a))$, $(b, f(b))$ noktalarını birleştiren kirişin x eksenini kesim noktası yardımıyla aralığı iki alt parçaya böler. Kirişin ekseni kesim noktasını belirlemek için öncelikle kiriş denklemini göz önüne alalım:

  $$y = f(a) + \frac{f(b) - f(a)}{b - a}(x - a)$$

Bu doğrunun $x = c$ olarak adlandıracağımız x eksenini kesim noktasını, $y = 0$ almak suretiyle

$$\begin{aligned} x &= c = a - \frac{(b-a)}{f(b)-f(a)} f(a) \\ &= b - \frac{(b-a)}{f(b)-f(a)} f(b) \end{aligned} \tag{1.1}$$

olarak elde ederiz.

Bu yönteme ait algoritmayı, Algoritma 1.2'de küçük bir değişiklik yapmak suretiyle elde edebiliriz: Algoritma 1.2'de

$$c = (a+b)/2$$

yerine

$$c = b - \frac{(b-a)}{f(b)-f(a)} f(b)$$

almak yeterlidir.

ÖRNEK 1.5. *$f(x) = x^2 - 1$ fonksiyonunun $[0, 3]$ aralığındaki sıfır yeri için kirişle bölme yöntemi ile ilk üç yaklaşımı elde ediniz.*

Çözüm.

f fonksiyonu sürekli ve

$$f(0)f(3) = -8 < 0$$

olduğu için fonksiyonun verilen aralıkta bir sıfır yeri mevcuttur ve kirişle bölme yöntemi uygulanabilir. Sıfır yerini içeren ve $n-$inci adımda($n \geq 1$) elde edilen aralığı $[a_n, b_n]$ ve $(a_n, f(a_n))$,$(b_n, f(b_n))$ noktalarını birleştiren kirişin x eksenini kestiği noktayı da c_n ile gösterelim. Buna göre

$$c_n = b_n - \frac{(b_n - a_n)}{f(b_n) - f(a_n)} f(b_n)$$

dir.

- İlk adımda
$$a_1 = 0, b_1 = 3, c_1 = \frac{1}{3}$$
dir. $f(1/3) = -8/9 \neq 0$ olup, sıfır yeri henüz belirlenmemiştir.

- Şimdi sıfır yerini içeren ikinci alt aralığı belirleyelim: Bu alt aralık ya
$$[a_1, c_1] = [0, 1/3] \text{ veya } [c_1, b_1] = [1/3, 3]$$
aralığı olmalıdır.
$$f(1/3)f(3) = -64/9 < 0$$
olduğu için yeni alt aralık
$$[a_2, b_2] = [1/3, 3]$$
ve
$$c_2 = 3/5$$
dir. $f(3/5) = 9/25 - 1 = -16/25 \neq 0$ olup, sıfır yeri henüz belirlenmemiştir.
- Sıfır yerini içeren üçüncü alt aralığı da benzer biçimde belirleyebiliriz: Bu alt aralık ya
$$[a_2, c_2] = [1/3, 3/5] \text{ veya } [c_2, b_2] = [3/5, 3]$$
aralığı olmalıdır.
$$f(3/5)f(3) = -128/25 < 0$$
olduğu için yeni alt aralık
$$[a_3, b_3] = [3/5, 3]$$
ve
$$c_3 = 7/9$$
dur.

O halde sıfır yeri için kirişle bölme yöntemi ile elde ettiğimiz ilk üç yaklaşımı
$$c_1 = 1/3, c_2 = 3/5, c_3 = 7/9$$
olarak elde ederiz.

Kirişle bölme yöntemi de her bir adımda elde edilen $[a_n, b_n]$ aralığında yer alan ve $(a_n, f(a_n)), (b_n, f(b_n))$ noktalarını birleştiren kirişin x eksenini kesim noktası olarak elde edilen $\{c_n\}$ noktalar dizisinin fonksiyonun sıfır yerine yakınsadığını öngörür(Alıştırma 14).

Uyarı. *ikibol.m isimli dosyada da*

$$c = b - \frac{(b-a)}{f(b) - f(a)} f(b)$$

değişikliğini yaparak elde ettiğiniz programı kirislebol.m *isimli dosyada kaydediniz.* function ikibol(f,a,b) *satırını ise* function kirislebol(f,a,b) *olarak değiştirmeyi unutmayınız(Alıştırma 8).*

Bir diğer alternatif ise *kirişlerle yaklaşım yöntemidir:*

- **Kirişlerle yaklaşım Yöntemi:** Kirişle bölme yöntemindeki sıfır yerini içeren aralıkla başlayıp(yani $f(a)f(b) < 0$) ve yine sıfır yerini içeren alt aralık ile devam etme kriterinden vazgeçilebilir.
  - Bu durumda verilen f fonksiyonu ve belirlenmesi istenilen sıfır yerine yakın komşuluktaki keyfi iki x_1, x_2 noktası ile başlayarak $(x_1, f(x_1)), (x_2, f(x_2))$ noktalarından geçen kirişin x eksenini kestiği x_3 noktası (1.1)

  $$x_3 = x_2 - \frac{(x_2 - x_1)}{f(x_2) - f(x_1)} f(x_2)$$

  olarak ta ifade edilebilir.
  - Daha sonra en güncel iki yaklaşım noktası olarak $x_1 := x_2$, $x_2 := x_3$ alarak, işleme $|f(x_3)| > \epsilon$ kriteri doğru olduğu sürece devam edilir. Elde edilen en son x_3 değeri sıfır yeri için yaklaşım kabul edilir. Bu yöntem kirişle bölme yöntemi ile çoğu kez karıştırılır ve kısaca *Kiriş(secant) veya Kirişlerle yaklaşım yöntemi* olarak adlandırılır(Bölüm 6).
- **Parabollerle yaklaşım yöntemi:** Kirişlerle yaklaşım yöntemi de geliştirilebilir.
  - Sıfır yeri komşuluğunda iki nokta yerine, x_1, x_2 ve x_3 gibi üç nokta alarak,

  $$(x_1,f(x_1)),(x_2,f(x_2)),(x_3,f(x_3))$$

  noktalarından geçen ikinci dereceden polinomun x_4 ile göstereceğimiz sıfır yerini fonksiyonun sıfır yeri için yaklaşım kabul edelim. Bir sonraki adımda $x_1 = x_2, x_2 = x_3, x_3 = x_4$ alarak işleme devam edelim. $|f(x_3)| > \epsilon$ olduğu sürece işleme devam edelim. Kriteri sağlamayan ilk x_3 değerini sıfır yeri olarak kabul edelim. Yukarıda ana hatlarıyla bahsedilen ve parabollerin sıfır yerleri ile verilen fonksiyonun sıfır yerine yaklaşımı esas alan bu yöntem, 1956 yıllında David E. Muller[7] tarafından geliştirilmiş olup Muller yöntemi olarak bilinir([1]) .

Problem III ile verilen bir sonraki örneğimizde, sayısal analiz aşamalarını sürekli bir fonksiyonun verilen bir nokta komşuluğundaki **sıfır yerini belirleme** problemi üzerinde inceliyoruz.

1.2.3 Problem III: Sürekli bir fonksiyonun, verilen bir nokta komşuluğundaki sıfır yeri için yaklaşım belirleme problemi

Problem I için geliştirilen yöntem ile *Problem II*'deki yöntemi birleştirerek karma(**hibrid)** olarak adlandırılan bir yöntem geliştirebiliriz.

1. **Problem ifadesi**

   Sürekli bir f fonksiyonunun, verilen bir x_0 noktası komşuluğundaki sıfır yerini belirleyiniz.

2. **Yöntem(karma yöntem)**

   Öncelikle f fonksiyonunun verilen bir x_0 noktası komşuluğundaki sıfır yerini içeren $[a, b]$ aralığını *Problem I* için geliştirdiğimiz yöntem ile belirledikten sonra, elde edilen aralığı ikiye bölme yöntemine göndererek fonksiyonunu sıfır yerini belirleyebiliriz.

3. **Algoritma** Yönteme ait Algoritma 1.3 aşağıda verilmektedir.

4. **Program**

   Algoritmaya ait Program 1.3 aşağıda verilmektedir.

5. **Uygulama**

[7] David E. Muller(1924-2008) Amerikalı matematik ve bilgisayar bilimci.

Algoritma 1.3 Verilen nokta komşuluğunda sıfır yeri belirleme algoritması.

(a) Girdi f,x0

(b) f nin sıfır yerini içeren [a,b] aralığını Problem I deki yöntem ile belirler

(c) i. eğer f(a)=0 ise c=a,

  ii. değil ve eğer f(b)=0 ise c=b dir,

  iii. değilse Problem II deki yöntem ile c=ikibol(f,a,b) ile c yi bul

(d) Çıktı: c

```
%------------------------------------------------------------
    function c=fsifir(f,x0);
    X=bul(f,x0);
    if isempty(X)
        c=[]; return;
      else
        a=X(1);b=X(2);
     if f(a)==0 c=a;
        elseif f(b)==0 c=b;
        else
         c=ikibol(f,a,b);
     end
   end
%-------------------------------------------------------------
```

Program 1.3: Girilen nokta komşulundaki sıfır yerini belirleme uygulaması.

ÖRNEK 1.6. *$f(x) = \ln(x) - x + 4$ fonksiyonunun $x_0 = 4$ noktası komşuluğundaki sıfır yerini belirleyiniz.*

Çözüm.

```
>> f=@(x)  log(x)-x+4;


>> fsifir(f,4)

ans =
```

```
5.7490
```

Aynı işlem MATLAB/OCTAVE fzero fonksiyonu yardımıyla da gerçekleştirilebilir:

```
>> fzero(f,4)

ans =

5.7490
```

ÖRNEK 1.7. *$f(x) = xsin(1/x)$ fonksiyonunun $x_0 = 4$ noktası komşuluğundaki sıfır yerini belirleyiniz.*

Çözüm.

```
>> f=@(x)  x*sin(1/x);


>> fsifir(f,4)

ans =

0.3183
```

elde ederiz.

Benzer yazılımlarla karşılaştırma:

MATLAB/OCTAVE *fzero* fonksiyonunun yukarıda tanımlanan f fonksiyonunun sıfır yerini $x_0 = 4$ başlangıç noktasıyla belirleyememektedir:

```
>> fzero(f,4)

Exiting fzero: aborting search for an interval containing
a sign change because NaN or Inf function value encountered
during search.
```

```
(Function value at -Inf is NaN.)

Check function or try again with a different starting value.
ans =NaN
```

Ancak başlangıç noktası fonksiyonun sıfır yerine daha yakın seçilerek, *fzero* yardımıyla da sıfır yeri belirlenebilmektedir:

>>fzero(f,1)

ans =

0.3183

6. **Yöntemin analizi ve Alternatif arayışlar**

   Verilen bir reel x_0 noktası komşuluğunda tek katlı ve reel sıfır yerini bulan bu yöntem

   (a) çift katlı sıfır yerlerini ve

   (b) karmaşık sıfır yerini belirleyemez.

   Verilen bir reel x_0 noktası komşuluğunda reel sıfır yerini bulan yöntem, karmaşık sıfır yerini bulmak için geliştirilebilir.

Problem I-III ile sayısal analiz sürecine örnek olarak incelediğimiz ikiye bölme ve kirişle bölme yöntemleri **bölümleme yöntemleri** olarak adlandırılırlar, çünkü sıfır yerini içeren aralık, mevcut aralığın uygun bir biçimde bölünmesi suretiyle elde edilmektedir.

Yukarıda kısaca özetlenen kirişler ve parabolerle yaklaşım yöntemleri **yinelemeli(veya iteratif) yöntemler** olarak adlandırılırlar ve yinelemeli yöntemleri Bölüm 6 da inceliyoruz.

Alıştırmalar 1.1.

1.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix}$$

olmak üzere $AX = b$ denklem sisteminin çözümünü belirleme probleminin sayısal analizini aşağıdaki adımları uygulayarak gerçekleştiriniz:

(a) $det(A) \neq 0$ *ise Cramer yöntemini sayısal yöntem olarak deneyiniz.* $det(A) = 0$ *ise yöntemin uygulanamayacağı mesajını kullanıcıya iletiniz.*

(b) *Yönteme ait algoritmanızı adım adım ve açık bir biçimde yazınız. Algoritma kullanıcıdan* $A_{2\times2}$ *matrisi ve* $b_{2\times1}$ *vektörünü alarak* $X_{2\times1}$ *çözümünü sunmalıdır.*

(c) *Algoritmaya ait programınızı MATLAB/OCTAVE yazılım diline uygun olarak geliştiriniz.*

(d) *Programınız farklı* A *matrisleri ve* b *sağ yan vektörleri için test yapınız.*

2.

$$ax^2 + bx + c = 0$$

denkleminin köklerini belirleme probleminin sayısal analizini aşağıdaki adımları uygulayarak gerçekleştiriniz.

(a) $a \neq 0$ *için bilinen kuadratik polinom kök formüllerini sayısal yöntem olarak deneyiniz.*

(b) *Yönteme ait algoritmanızı adım adım ve açık bir biçimde yazınız. Algoritma kullanıcıdan* a, b, c *katsayılarını alarak, reel veya kompleks kökleri belirlemelidir.*

(c) *Algoritmaya ait programınızı MATLAB/OCTAVE yazılım dilinde geliştiriniz.*

(d) *Programınızı farklı* a, b, c *katsayıları için test yapınız.*

(e) $a = 0$ *olması durumundaki kökü ayrıca hesaplatmayı unutmayınız.*

3. *Soru 2 deki analiziniz yardımıyla* $A_{2\times2}$ *matrisinin özdeğerlerini belirleme probleminin sayısal analizini gerçekleştiriniz.*

4.

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

çember ve parabolünün arakesit noktalarını belirleme probleminin sayısal analizini gerçekleştiriniz. Verilen r *yarıçapı ve* $a \neq 0$ *katsayısı için her iki arakesit noktası belirlenmelidir.*

5. $f(x) = x^2 - 5x$, $a = -2, b = 3$ verilsin. f nin $[a, b]$ aralığındaki sıfır yeri için ilk üç yaklaşımı

   (a) ikiye bölme ve

   (b) kirişle bölme yöntemleri ile belirleyiniz.

6. $f(x) = x^3 - x - 1$, $a = -2, b = 3$ verilsin. f nin $[a, b]$ aralığındaki sıfır yeri için ilk üç yaklaşımı

   (a) ikiye bölme ve

   (b) kirişle bölme yöntemleri ile belirleyiniz.

7. İkiye bölme yöntemine ait Program 1.2'yi aşağıda verilen fonksiyonlar ve $[a, b]$ aralıkları için çalıştırarak sıfır yerlerini belirleyiniz. Epsilon sabitini $eps = 1e - 10$ olarak alınız.

   (a) $f(x) = x^2 - 5x$, $a = -2, b = 3$

   (b) $f(x) = x^3 - x - 1$, $a = -2, b = 3$

   (c) $f(x) = ln(x + 1) - \frac{1}{4}x^2$, $a = 1, b = 4$

   (d) $f(x) = 5e^{-x} - coshx$, $a = 0, b = 4$

8. Soru 7'yi kirişle bölme(regula falsi) yöntemi için tekrarlayınız.

9. Kullanıcıdan f fonksiyonu , a,b değerleri ve eps toleransı ile birlikte yontem isimli bir değişken değerini de alarak $yontem = 1$ olması durumunda ikiye bölme yöntemini, $yontem = 2$ olması durumunda kirişle bölme yöntemini çağırarak sıfır yerini belirleyen bir uygulama geliştiriniz.

10. Soru 7'de verilen fonksiyonlar ve yine verilen a değerleri için $x_0 = a$ alarak Program 1.1 yardımıyla sıfır yerlerini içeren aralıkları hesaplayınız.

11. İkiye bölme yöntemini, her adımda elde edilen aralık içerisinde rasgele üretilen bir noktayı(örneğin MATLAB/OCTAVE ortamında $rand$ fonksiyonu ile) c noktası olarak kabul edecek biçimde modifiye ediniz. Elde ettiğiniz yöntemi ikiye bölme yöntemiyle karşılaştırınız. Ne gözlemliyorsunuz?

12. Program 1.3 ile tanımlanan $fsifir$ programı ve MATLAB/OCTAVE'a ait $fzero$ programları yardımıyla Soru 7'de verilen fonksiyonların sıfır yerlerini yine aynı soru da verilen a değerlerini başlangıç noktası alarak hesaplayınız.

13. MATLAB/OCTAVE ortamında $roots$ komutu, katsayıları verilen polinomun sıfır yerlerini belirler. Buna göre Soru 7(a),(b) deki polinomların katsayılarını girerek köklerini elde ediniz. Elde ettiğiniz sonuçlar yukarıdaki formül ile elde edilen sonuçlarla aynı olmalıdır. Örneğin

    $>> roots([a\ b\ c])$ komutu ile

    $$p(x){=}ax^2 + bx + c$$

    polinomunun kökleri hesaplanır.

14. Kirişle bölme yönteminde her bir adımda elde edilen $[a_n, b_n]$ aralığında yer alan ve $(a_n, f(a_n)), (b_n, f(b_n))$ noktalarını birleştiren kirişin x eksenini kesim noktası olarak elde edilen $\{c_n\}$ noktalar dizisinin fonksiyonun sıfır yerine yakınsadığını gösteriniz.

15. Bilgisayarınızın bellek kullanım kapasitesini test yapınız:

    MATLAB/OCTAVE ortamında $A = rand(n)$ komutu ile $n = 1000$ için rasgele bir $A_{n\times n}$ matrisini üreterek, matrisin tersi, determinantı ve rankını bulmaya çalışınız.

    $$n = 2000, 4000, 10000$$

    için aynı işlemleri yapmaya çalışarak tersini hesaplayabileceğiniz en büyük boyutlu matrisi tahmin etmeye çalışınız.

16. Rand komutu yardımıyla üreteceğiniz ve elemanları rasgele elamanlardan oluşan A matrisi ve b vektörü için MATLAB/OCTAVE ortamında $\mathbf{x} = A\backslash\mathbf{b}$ komutuyla çözebileceğiniz en büyük boyutlu $A\mathbf{x} = \mathbf{b}$ lineer cebirsel denklem sisteminin boyutunu belirlemeye çalışınız.

17. Bilgisayarınızın CPU hızını test yapınız:

    $$S_N = \sum_{k=1}^{N} 1/k$$

    toplamını

    $$N = 1000, 10000, 100000, 1000000$$

değerleri için hesaplayarak her bir işlem için kullanılan CPU zamanını belirlemeye çalışınız. Artan N değerleri için CPU zamanı nasıl değişmektedir. N değerlerine karşı, toplama işlemi için gerekli CPU zamanlarının grafiğini çizdiriniz. Bu amaçla MATLAB/OCTAVE ortamında $cputime$ komutunu skaler cebirsel işlemlerle ve döngü kullanmak suretiyle uygulayan aşağıdaki programı kullanabilirsiniz:

```
%-------------------------------------------------------
        function sonuc=topla(N)
        topla=0;
        zaman=cputime;
        for i=1:N
           topla=topla+1/i;
        end
        zaman=cputime-zaman;
        disp(['zaman=',num2str(zaman)]);
%-------------------------------------------------------
```

Program 1.4: For döngüsü ile seri toplamı için cpu testi.

18. *Soru 17 deki toplam for döngüsü kullanılmadan ve vektör cebirsel bir algoritma ile aşağıda belirtilen adımları takip etmek suretiyle gerçekleştiren toplav isimli vektör cebirsel işlem gerçekleştiren bir fonksiyon programı hazırlayınız. Hazırlayacağınız programı Soru 17 deki N değerleri için çalıştırarak CPU zamanlarını karşılaştırınız. Hangi kod daha az CPU zamanı gerektirmektedir?*

    (a) *$birler$ isimli N elemanlı ve 1 sayılarından oluşan birler=ones(1,N) vektörünü tanımlayınız.*

    (b) *$indis$ isimli indis=1:N vektörünü tanımladıktan sonra MATLAB/OCTAVE './' operatörü ile dizi=1./indis vektörünü tanımlayınız.*

    (c) *(a) da elde ettiğiniz birler dizisi ile (b) de elde ettiğiniz dizi isimli vektörün iç çarpımını (birler*dizi') hesaplayınız.*

19. *Soru 17'yi*

$$S_N = \sum_{k=1}^{N} 1/k^2$$

toplamı için tekrarlayınız. Belirtilen N değerleri için elde ettiğiniz kısmi toplamlar dizisi Euler sayısı olarak bilinen $\pi^2/6$ *değerine yakınsamalıdır.*

20. *Soru 19 daki* S_N *için Soru 18 de özetlenen adımları uygulayan vektör cebirsel işlem gerçekleştiren bir fonksiyon programı hazırlayınız. Soru 17 de belirtilen* N *değerleri için Soru 19 da elde ettiğiniz sonuçları karşılaştırınız. Hangi kod daha az CPU zamanı gerektirmektedir?*

Kaynaklar

[1] Atkinson, K. An Introduction to Numerical Analysis, John Wiley ve Sons, 1988.

[2] Coşkun, E. OCTAVE ile Sayısal Hesaplama ve Kodlama(URL:erhancoskun.com.tr).

[3] Coşkun, E. Maxima ile Sembolik Hesaplama ve Kodlama(URL:erhancoskun.com.tr).

[4] Coşkun, E. Diferensiyel Denklemler için Sonlu Fark Yöntemleri(URL:erhancoskun.com.tr).

[5] MATLAB, Mathworks(*URL:mathworks.com).*

[6] Maxima, GNU özgür yazılım(*URL:maxima.sourceforge.net*).

[7] OCTAVE, GNU özgür yazılım(*URL:OCTAVE.sourceforge.net*).

[8] Pottmeyer, L., News on quadratic polynomials, Snapshots of modern mathematics from Oberfolfach, 2/2017(URL:imaginary.org).

[9] S. W., Warren, Zill, D. G., Calculus: Early Transcendentals, Çeviri: Matematik Cilt I, II(Çeviri editörü İsmail Naci Cangül), Nobel Akademik Yayıncılık, 2010.