

SAYISAL ANALİZ

Yrd.Doç.Dr. Abdullah SEVİN



SAYISAL ANALİZ

3. Hafta

SAYISAL ANALİZDE HATA KAVRAMI VE HATA TÜRLERİ

İÇİNDEKİLER

1. Sayısal Analizde Problem Çözümünde İzlenilecek Adımlar
2. Matematiksel Modelleme
3. Sayısal Analizde Hata Kavramı
4. Sayısal Analizde Hataların Sebepleri
5. Sayısal Hata ve Hata Türleri
 - ☐ Mutlak Hata
 - ☐ Bağıl Hata
 - ☐ Yaklaşım Hataları
 - ☐ Yüzde Hata
 - ☐ Hata Sınırlaması
6. Anlamlı Basamak
7. Sayıların Bilgisayarda Gösterilmesi
8. Kesme ve Yuvarlama Hataları

Giriş

- Sayısal analiz, matematik problemlerinin bilgisayar yardımı ile çözümlenme tekniğidir. Genellikle analitik olarak çözümleri çok zor veya imkansız olan matematiksel problemleri belirli bir hata aralığında çözmek için kullanılır.

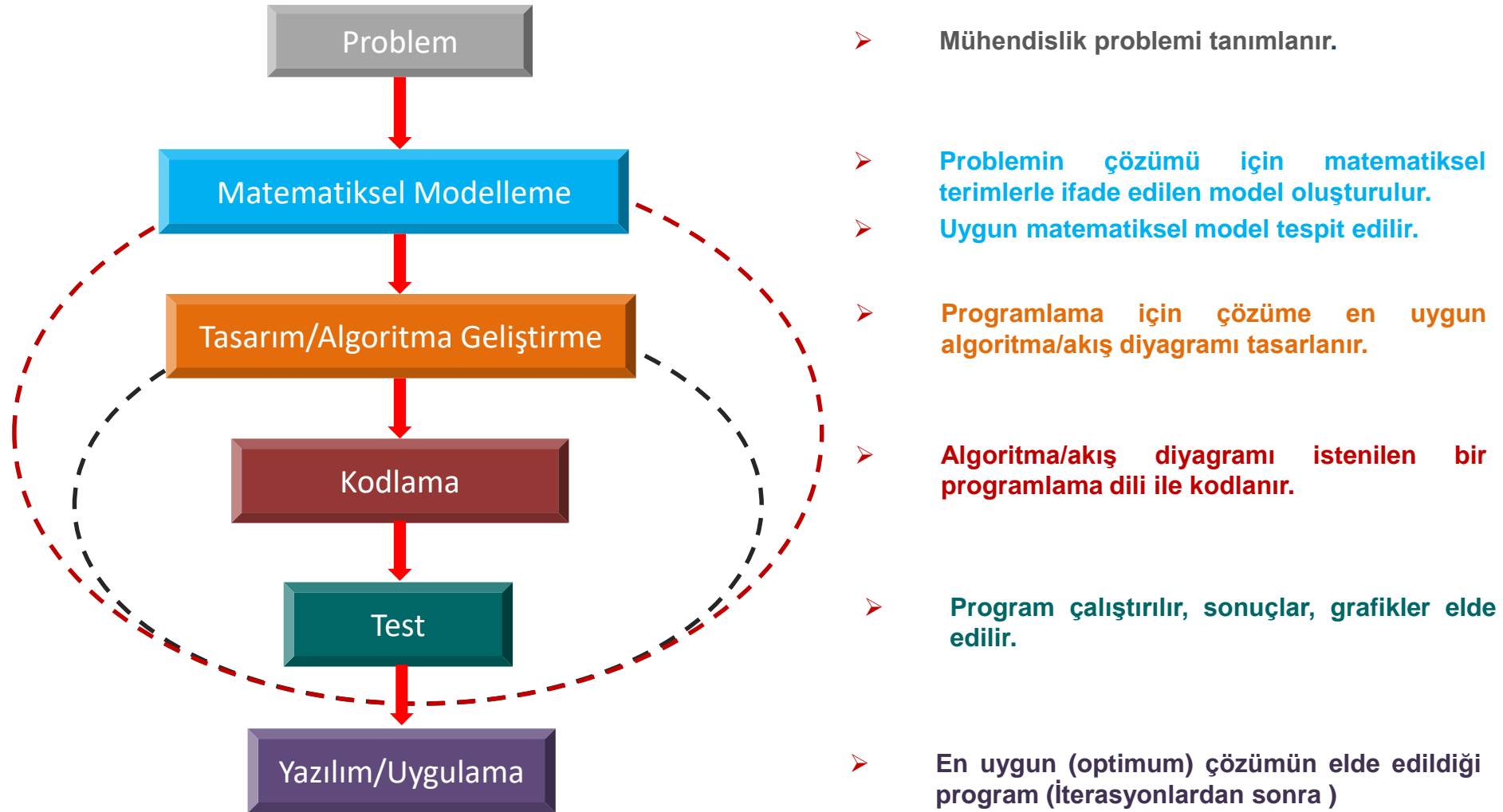


Sayısal analiz işleyiş şeması



- Sayısal analiz yöntemleri ile bulunan/hesaplanan sonuçlar kesin değerler değil yaklaşık değerlerdir.

Sayısal Analizde Problem Çözümünde İzlenilecek Adımlar (Detaylı)



Not: Hatalara göre süreç, tekrar gözden geçirilir, işlemler tekrar edilir.

PROGRAMLAMA NEDİR?

- ❑ Programlamanın (Program Geliştirmenin) genel yapısı sırasıyla şu adımları kapsar:

1. Problemin tanımlanması,
2. Problemin çözümlenmesi,
 - 2.1. Çözüm yolunun belirlenmesi,
 - 2.2. Çözüm yoluna uygun algoritmanın belirlenmesi,
 - 2.3. Algoritmaya uygun akış diyagramının çıkarılması,
 - 2.4. Algoritmayı gerçekleştirecek uygun programlama dilinin seçilmesi,
3. Problemin programlama dili komut seti yardımıyla kodlanması,
4. Hazırlanan programın denenmesi ve belgelendirilmesi

PROGRAMLAMA NEDİR?

1. Problemin tanımlanması,

- ❑ Bir problemin herhangi bir programlama dilinde kodlanmasına başlanmadan önce problemin tam olarak anlaşılması gerekmektedir. Aksi halde yanlış çözüm kaçınılmazdır.

2. Problemin Çözümlemesi

2.1 Çözüm Yolunun Belirlenmesi

- ❑ Giriş verilerinden sonuçta elde edilecek verilere nasıl, hangi yolla ulaşılabileceğinin tespiti gerekir.
- ❑ Bu durumun iyi analiz edilmesi gerekir.
- ❑ Problemin matematiksel modeli bu aşamada belirlenir.
- ❑ Hangi tekniğin en uygun olduğuna programcının bilgisi ve tekniği etki eder.

Problemin Çözülmesi

Örnek: Aranılan bir büyüklüğün herhangi bir $\{a\}$ kümesi içerisinde olup olmadığının araştırılması.

$\{a\} = \{3, 7, -10, 8, 1, -4, -94, 6, 2, -1, 34, 14, 78, -19, 99\}$ olsun.

$x = -12$ elemanının bu küme içinde yer alıp almadığını arayalım.

❑ Çözüm Yolları:

1. Verilen x değeri (-12) sırayla $\{a\}$ kümesinin bütün elemanları ile tek tek karşılaştırılarak arama yapılabilir.

2. Önce $\{a\}$ kümesi kendi içerisinde artan sırada (büyükten küçüğe doğru) sıralanır.

$\{a\} = \{-94, -19, -10, -4, -1, 1, 2, 3, 6, 7, 8, 14, 34, 78, 99\}$

Daha sonra verilen x değeri (-12) sıralanmış $\{a\}$ kümesi içerisinde baştan bütün elemanlar ile karşılaştırılarak arama yapılabilir. En son karşılaştırılan değer x değerinden büyük ise işlem kesilir.

Problemin Çözülmesi

☐ Çözüm Yolları:

3. $\{a\}$ kümesindeki eleman sayısı ikiye tam bölünür ve orta eleman bulunur. x değeri (-12) orta elemanla karşılaştırılır. Eğer orta eleman x değerinden büyük ise x değeri $\{a\}$ kümesinin ilk yarısında olacaktır. İlk yarıdaki eleman sayısı ikiye tam bölünerek 2. orta eleman bulunur. x değeri 2. orta elemanla karşılaştırılır.

Eğer 2. orta eleman x değerinden büyükse x $\{a\}$ kümesinin 2. yarısının ilk bölümünde olacaktır. Bu işlemler tek eleman kalıncaya kadar sürdürülür. Arama sonlandırılır.

☐ $\{a\}' = \{-94, -19, -10, -4, -1, 1, 2, 3\}$

☐ $\{a\}'' = \{-94, -19, -10, -4\}$

☐ $\{a\}''' = \{-10, -4\}$

Problemın Çözümlemeı

- ❑ Yukarıda belirtilen 3 farklı çözüm yolu problemin çözümünü sağlamaktadır.
- ❑ Bu çözüm yollarından hangisinin seçileceđi;
 1. Verinin büyüklüğüne,
 2. Amaçlanan işlem hızına,
 3. Yapılacak işlemin tekrarlanma sayısına bađlıdır.

Matematiksel Modelleme

Bağımlı değişken = **f** (bağımsız değişkenler, parametreler, zorlayıcı fonksiyonlar)

- Bağımlı değişken, sistemin davranışını ya da konumunu gösteren karakteristik
- Bağımsız değişkenler, sistemin davranışının incelendiği boyutlar (zaman, konum vb.)
- Parametreler, sistemin özelliklerini, yapısını ya da bileşenlerini yansıtan parametreler
- Zorlayıcı fonksiyonlar, sisteme etkiyen dış etkiler

❑ Örnek: Bir paraşütçünün düşme hızının hesabı (Newton 2. yasası)

Analitik Çözüm Modeli

$$v = \frac{g m}{c} \left(1 - e^{-(c/m)t} \right)$$

v: hız (diğer zorlayıcı kuvvetlere, parametrelere ve bağımsız değişkenlere bağlı olarak değişen bir bağımlı değişkendir.

t: zaman (bağımsız değişken)

g: yer çekimi sabiti (zorlayıcı kuvvet)

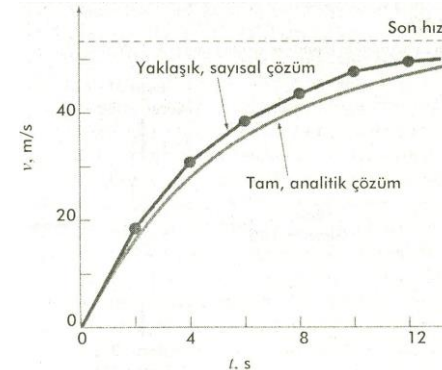
c: havanın direnç katsayısı (sistemin fiziksel özelliği)

m: kütle (sistemin fiziksel özelliği)

Sayısal Çözüm Modeli

$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c}{m} v(t_i) \right] (t_{i+1} - t_i)$$

Yeni değer = Eski değer + eğim * adım uzunluğu (Euler Yöntemi)



Algoritma

- ❑ Algoritma ; verilerin bilgisayara hangi çevre biriminden girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağını, sonucun nasıl ve nereye yazılacağını sözel olarak ifade edilmesi biçiminde tanımlanabilir.

Örnek :

Verilen iki sayının toplamının bulunmasının algoritması aşağıdaki gibi yazılır:

Algoritma :

Adım 1-Başla

Adım 2-Birinci sayıyı oku

Adım 3-İkinci sayıyı oku

Adım 4-İki sayıyı topla

Adım 5-Dur

Akış Şemaları (Diyagramları)

- Herhangi bir sorunun çözümü için izlenmesi gerekli olan aritmetik ve mantıksal adımların söz veya yazı ile anlatıldığı algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekline "akış şemaları" veya FLOWCHART adı verilir.

Örnek :

Adım 1-Başla

Adım 2-A'yı oku

Adım 3-B'yi oku

Adım 4- $D=A+B$

Adım 5- $E=A*B$

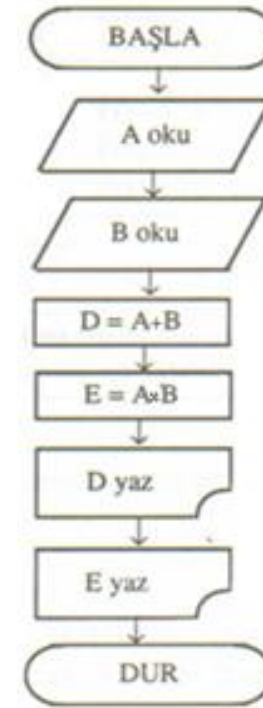
Adım 6-D'yi yaz

Adım 7-E'yi yaz








Adım 8-Dur



Akış Şeması (FLOWCHART)



Akış Şemaları (Diyagramları)

	Algoritmanın başladığını ya da sona erdiğini belirtmek için kullanılır.
	Klavye aracılığı ile giriş ya da okuma yapılacağını gösterir.
	Yazıcı(printer) aracılığı ile çıkış yapılacağını gösterir.
	Kart okuyucu aracılığıyla giriş yapılacağını gösterir.
	Araç belirtmeden giriş ya da çıkış yapılacağını gösterir.
	Hesaplama ya da değerlerin değişkenlere aktarımını gösterir.
	Aritmetik ve mantıksal ifadeler için karar verme ya da karşılaştırma durumunu gösterir.

Akış şemaları **içerik** ve **biçimlerine** göre genel olarak üç grupta sınıflandırılabilirler.

1. Doğrusal Akış Şemaları,
2. Mantıksal Akış Şemaları,
3. Döngüsel Akış Şemaları,

Doğrusal Akış Şemaları

İş akışları, giriş, hesaplama, çıkış biçiminde olan akış şemaları bu grup kapsamına girer.

Değişkenler

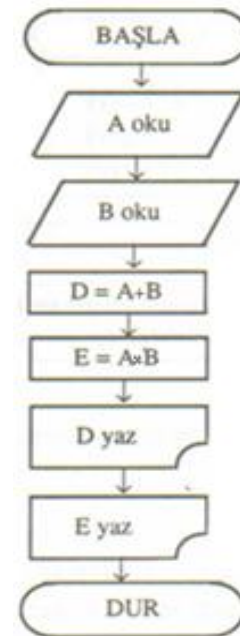
A: Birinci sayı B: İkinci sayı
D: İki sayının toplamını ($A+B$)
E: İki sayının bölümünü ($A*B$)

Algoritma

Adım 1-Başla
Adım 2-A'yı oku
Adım 3-B'yi oku
Adım 4- $D=A+B$
Adım 5- $E=A*B$
Adım 6-D'yi yaz
Adım 7-E'yi yaz
Adım 8-Dur



Akış Şeması



Mantıksal Akış Diyagramları

Mantıksal kararları içeren akış diyagramlarıdır

Değişkenler

A: Birinci sayı B: İkinci sayı

Algoritma

Adım 1-Başla

Adım 2-A,B'yi oku

Adım 3- $A=B$ ise Adım 7'ye git

Adım 4- $A>B$ ise Adım 6'ya git

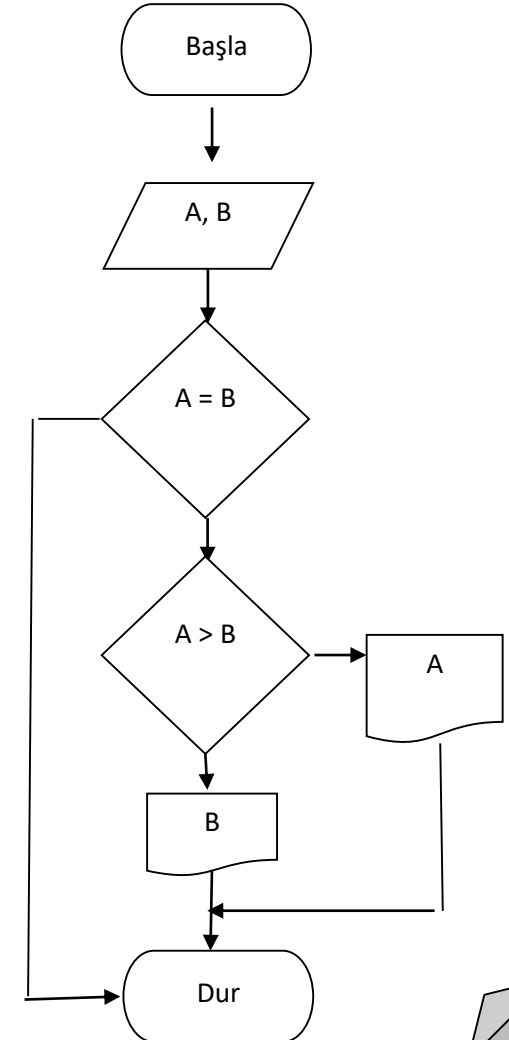
Adım 5-B'yi yaz Adım 7'e git

Adım 6-A'yı yaz Adım 7'e git

Adım 7-Dur



Akış Şeması



Döngülü Akış Diyagramları

Akış sürecinde yer alan herhangi bir adım ya da aşamanın birden fazla kullanıldığı akış diyagramlarına denir.

Örnek :

N sayısını ekrandan okutarak faktöriyelini hesaplayıp yazan programın algoritma ve akış diyagramını oluşturalım.

Değişkenler

NFAK=N faktöriyel ($N!$) değerini,
OGRSAY=1'den N'e kadar >sayıları gösterecek,
 $NFAK=1*2*.....*N$

Algoritma

Adım 1-Başla

Adım 2-N'i ekrandan oku

Adım 3-NFAK=1

Adım 4-OGRSAY=1

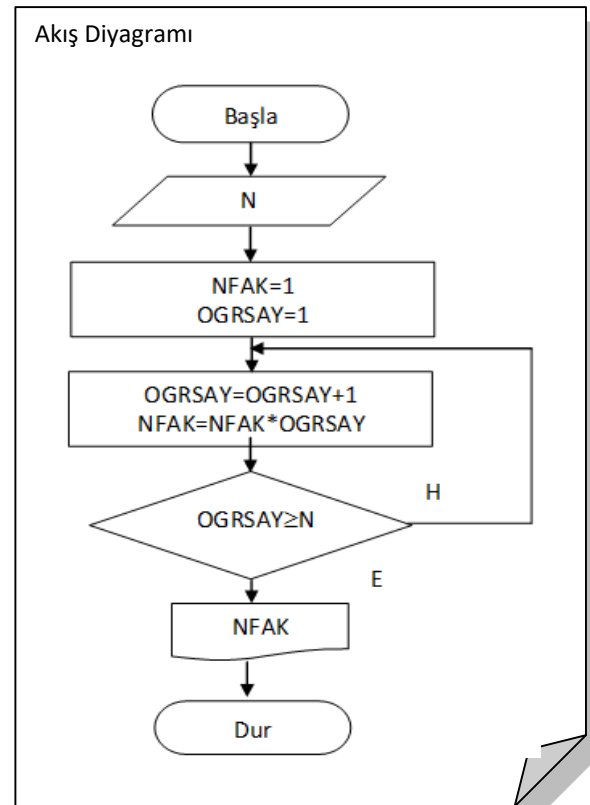
Adım 5-OGRSAY=OGRSAY+1

Adım 6-NFAK=NFAK*OGRSAY

Adım 7-Eğer OGRSAY \geq N

Adım 8-NFAK yaz

Adım 9-Dur



PROGRAMLAMA NEDİR?

- ❑ Programlama, herhangi bir problemin bir programlama dili kullanılarak çözülmesi için yazılan mantıksal kod bloklarına verilen addır.
- ❑ Amaç problemin çözümüne uygun şekilde hazırlanan program kodu ile problemi çözmeye çalışmaktır. Bu amaç için araç olarak herhangi bir programlama dilini kullanırız.
- ❑ Programlama diline ait hazır komutları kullanarak problemi çözmeye çalışırız. Bu komutlar programlama dilleri arasında farklılık göstermesine rağmen programlama mantığı bütün dillerde aynıdır.
- ❑ Unutulmamalıdır ki hazırlanan bir program, gerektiğinde başkaları tarafından da kullanılacaktır. Bu nedenle hazırlanan programın mümkün olduğunca hatalardan arındırılmış olması gerekmektedir. Beklenen sonuçları verecek şekilde hazırlanmış olması gerekmektedir.

Sayısal Analizde Hata Kavramı

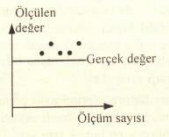
- ❑ Sayısal Analiz, matematiksel hesaplamaları ve problemleri tekrarlı işlemler ile bilgisayarlar (programlar) aracılığı ile çözmektir.
- ❑ Özetle, Sayısal Analiz, bir problemi hesap makinesi, bilgisayar gibi araçlarla yaklaşık olarak çözmek için kullanılan yöntemlerdir.
- ❑ Bilgisayarlı sayısal analizde çeşitli sebeplerle hatalı sonuçlar elde edilebilir.
 - ❑ Kullanıcıdan kaynaklanan hatalar
 - Modelleme hatası
 - Kodlama hatası
 - ❑ Bilgisayar ya da kullanılan yazılımdan kaynaklanan hatalar
 - Belirli bir uzunlukta sayı depolayabilme,
 - Yuvarlatmadan kaynaklanan hata,
 - Kesmeden kaynaklanan hata

Sayısal Analizde Hataların Sebepleri

- ❑ Fiziksel veya sosyal olayların matematiksel olarak çözümlerinde yapılan hatalar genellikle üç ana başlıkta toplanır.
 - ❶ **Modelleme Hatası**, bir olayın formüle edilmesi esnasında varsayımlardan kaynaklanan hatalardır.
 - ❑ **Örnek:** Serbest düşme problemlerinin modellenmesinde, hava ile cisim arasındaki sürtünme kuvvetinin ihmal edilmesinden dolayı meydana gelen hatalar bu tür hatalar grubuna girer.
 - ❷ **Ölçme Hatası**, deney ve gözlemede ölçmelerden dolayı meydana gelen hatalardır.
 - ❑ Örnekte, eğer serbest düşme yapan cismin, düştüğü mesafe veya havada düşerken geçen süre eğer yanlış ölçülürse bu tür hatalar ölçme hatası olarak tanımlanabilir.
 - ❸ **Sayısal hatalar** veya diğer bir deyimle modelin çözümlemesinde yapılan hatalardır.
 - ❑ Dersimizde sayısal hatalar ile ilgileneceğiz.

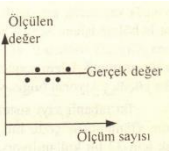
Sayısal Hata

- ❑ Sayısal hatalar, matematiksel işlemler ve değerlerin yaklaşık kullanımlarından ortaya çıkan farklardır.



- ❑ **Doğruluk:** Ölçme veya hesaplama sonuçlarının gerçek değere yakınlığıdır.

- ❑ Çoğu durumda kesinliğin değeri bilinmediğinden en iyi doğru cevap genellikle en iyi tahmin edilen değere göre ölçülür.



- ❑ **Kesinlik (hassasiyet):** Bir büyüklüğün defalarca ölçülmesi sonucu ölçümlerin birbiriyle ne kadar yakınlıkta olduğudur.

- ❑ Bilgisayarlı hesaplama doğruluğu, bilgisayarın kelime uzunluğu (bir kerede işlenen bit sayısı), ile doğrudan ilişkilidir.

- ❑ Bilgisayarlar sonlu sayıda rakamı saklayabilirler.

- ❑ Bu sebeple, hesaplamalar, tam değil, yaklaşımlar ile yapılabilir.



- ❑ Hesaplamalarda ne kadar hata vardır.
- ❑ Ne kadarlık bir hata kabul edilebilir

Sayısal Hata Türleri

❶ Mutlak Hata

- ❑ Bir büyüklüğün, analitik olarak (**doğru/gerçek**) bilinen değeri ile sayısal hesaplamalarda elde edilen **yaklaşık** (**hesaplanan**) değeri arasındaki mutlak farktır.

- ❑ $\varepsilon_{Mutlak} = |(\text{gerçek değer}) - (\text{yaklaşık değer})|$

- ❑ $\epsilon = |a - \check{a}|$



- Hesaplamalarda, genellikle hatanın işaretiyle değil, büyüklüğüyle ilgilenilir.
- Bu sebeple hata kavramında mutlak değer kullanılır.

Sayısal Hata Türleri

② Bağıl Hata (Relative Error)

□ Bir büyüklüğün, analitik olarak (doğru/**gerçek**) bilinen değeri ile sayısal hesaplamalarda elde edilen **yaklaşık** (*hesaplanan*) değeri arasındaki mutlak farkın, gerçek değere oranıdır.

$$\square \quad \epsilon_{Bağıl} = \frac{|(\text{gerçek değer}) - (\text{yaklaşık değer})|}{|\text{gerçek değer}|} = \frac{\text{Mutlak Hata}}{\text{Gerçek Değer}}$$

$$\square \quad \epsilon_r = \frac{\epsilon}{a} \quad (a \neq 0)$$

Sayısal Hata Türleri

③ Yaklaşım Hataları

- ❑ Uygulamalarda genellikle, gerçek sonuç baştan bilinemeyeceğinden, bu durumda hata , gerçek değerin bilinen en iyi tahminine göre normalize edilir.
- ❑ Sayısal yöntemler bir sonucu bulmak için genellikle **iteratif** yöntemler kullanır.
 - ❑ **İterasyon**, bir işlev ya da yöntemin tekrar tekrar uygulanmasıdır.
 - ❑ Yöntem doğru ise her adımda gerçek değere biraz daha yaklaşılr.
 - ❑ Dizi ya da tekrarlı işlemlerde kullanılır.

❖ Yaklaşık Mutlak Hata

$$\epsilon_a = |(en\ iyi\ tahmin) - (yaklaşık\ değer)|$$

❖ Yaklaşık Bağıl Hata

$$\epsilon_a = \frac{|(en\ iyi\ tahmin) - (yaklaşık\ değer)|}{|en\ iyi\ tahmin|}$$

Sayısal Hata Türleri

③ Yaklaşım Hataları

- ❑ Bir niceliğin tam değeri verilmediği zaman gerçek bir hatanın hesaplanması mümkün değildir.
- ❑ Yaklaşık hata, hatanın sınırlarını belirlemede kullanılır.
- ❑ Bu durumda **Scarborough Kriteri** kullanılabilir.
- Aşağıdaki Scarborough kriteri gerçekleştiğinde, sonucun en az **n** anlamlı basamak için kesinlikle doğrudur denilebilir.

$$\epsilon_s = \%(0.5 * 10^{2-n})$$

Sayısal Hata Türleri

4 Yüzde Hata

- ☐ Bağıl, mutlak, yaklaşım hatası gibi tüm hata ifadelerinin %100 ile çarpılması ile % değerleri elde edilir.
- ☐ Buna göre hatalar, bağıl yüzde hata, yüzde yaklaşım hatası şeklinde adlandırılır.

Sayısal Hata Türleri

❖ **Örnek 1:** Gerçek değeri 200 cm olan bir elektrik kablosunun uzunluğu ölçüldüğünde 199 cm bulunuyor. Mutlak hatayı, bağıl hatayı ve yüzde bağıl hatayı hesaplayınız?

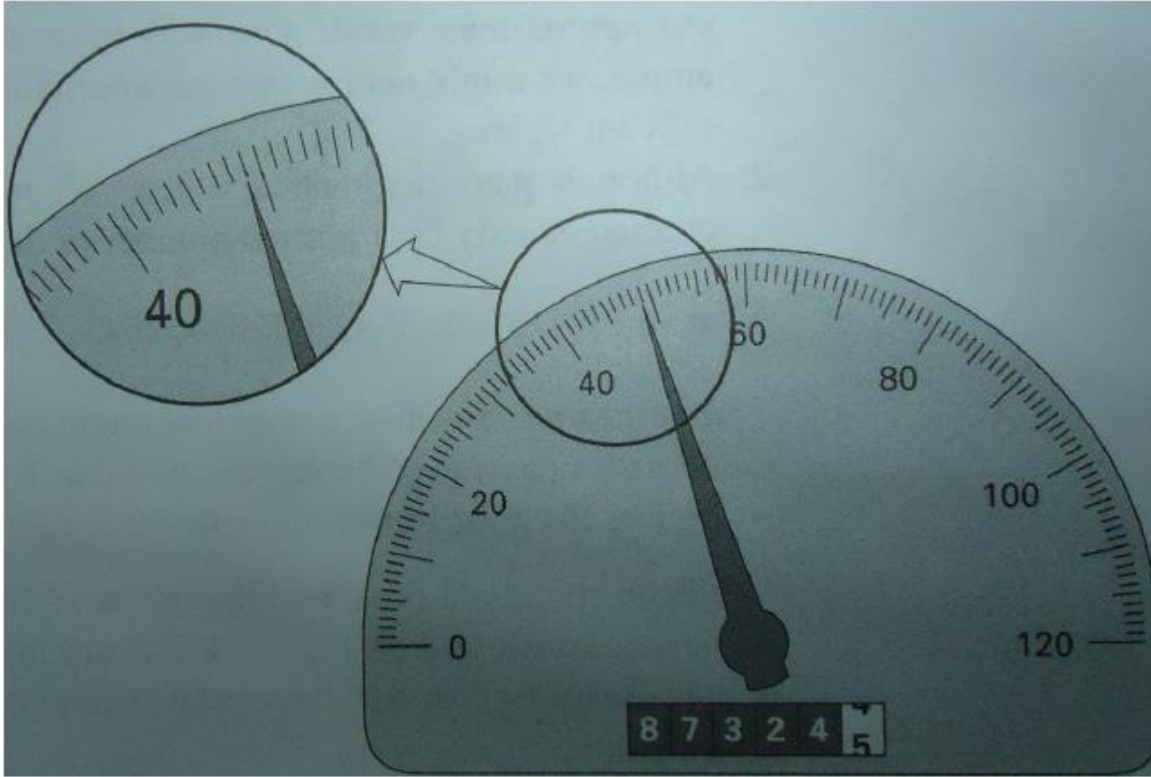
❑ **Mutlak Hata:** $\varepsilon_{mutlak} = |(gerçek\ değer) - (yaklaşık\ değer)|$
 $\varepsilon = |200 - 199| = 1\ cm$

❑ **Bağıl Hata:** $\varepsilon_{Bağıl} = \frac{|(gerçek\ değer) - (yaklaşık\ değer)|}{|gerçek\ değer|} = \frac{Mutlak\ Hata}{Gerçek\ Değer}$
 $\varepsilon_{Bağıl} = \frac{1}{200} = 0,005$

❑ **Yüzde Bağıl Hata:**

$$\% \varepsilon_{Bağıl} == \varepsilon_{Bağıl} * 100 = 0,005 * 100 = \% 0,5$$

Anlamalı Basamak



- ❑ Aracın hızı tam olarak kaç km/s
 - 48.9
 - 48.89
 - 48.85
- ❑ Aracın kilometresi kaçtır
 - 87324.45
 - 87324.48



- Bir sayının anlamlı basamakları, güvenle kullanılabilecek basamaklardır.

Anlamlı Basamak

- ❑ Mutlak ve Bağıl hata hesaplamasında gerçek değere ihtiyaç vardır.
- ❑ Ancak gerçek değer bilinmediğinde, özellikle dizi (seri) şeklindeki hesaplamalarda yaklaşık hata hesaplaması kullanılır.
- ❑ Bu işlemde son iki yineleme arasındaki rakamların kaç tanesinin tekrar ettiğine bakılarak fonksiyonun gerçek değerine ne kadar yaklaştığı (yakınsadığı) kontrol edilebilir.
- ❑ Her yeni tekrardaki değer, bir önceki değer ile yakınsa sonuç ya da ölçüm hassastır denilebilir. Ama doğruluk tartışılır.
- ❑ Örnek:

Tekrar Sayısı	Sonuç	Anlamlı Basamak Sayısı
1	$f(x)=abcdef$	0
2	$f(x)=abghik$	2

Anlamalı Basamak

- ❑ **Örnek 2:** Sin(x) in seri açılımı aşağıda görülmektedir.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

- ❑ sin($\pi/6$) değerinin gerçek değerini bulup, yukarıdaki seri açılıma göre sin($\pi/6$) için terim sayısına göre, mutlak ve bağıl hataları ile anlamalı basamak sayılarını gösteriniz?

TS	Sonuç	Mutlak Hata	Bağıl Hata	An. Bas.
1	0,44879895050	1,49E-02	3,44E-02	
2	0,43373273250	1,51E-04	3,48E-04	1
3	0,43388446480	7,26E-07	1,67E-06	3
4	0,43388373710	2,02E-09	4,65E-09	5
5	0,43388373910	1,76E-11	4,05E-11	8
6	0,43388373910	1,76E-11	4,05E-11	10
n	0,43388373912	0	0	

Sayısal Hata Türleri

- ❖ **Örnek 3:** Scarborough kriteri ile hataları anlamlı sayılar ile ilişkilendirmek?

$$\pi = 3.14159265358979$$

$$\tilde{\pi} = 3.14639184292698$$

$$\pi = 3.14159265358979$$

n

$$\epsilon_{\pi} = \pi - \tilde{\pi}$$

$$\epsilon_s = \% \left(0.5 * 10^{2-n} \right)$$

Sayısal Hata Türleri

- ❖ Matematikte fonksiyonlar/işlevler çoğu zaman sonsuz seriler ile gösterilir.
- ❖ Matematiksel bir ifadenin seri açılımında terim sayısı arttıkça, hesaplanan değer gerçeğe daha yaklaşır (*Mac Laurin Serisi Açılımı*).
- ❖ **Örnek 4:** e^x ifadesinin seri açılımı aşağıda görülmektedir.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

- ❑ $x=0.5$ değeri için $e^{0.5}$ in gerçek değeri 1.648721...
- ❑ Serinin birinci teriminden başlayarak, tek tek terim ekleyerek, $e^{0.5}$ sayısını tahmin edin.
- ❑ Her adımda (terim eklemeye) gerçek ve yüzde bağıl hataları bulun.
- ❑ Yaklaşık mutlak hata değeri, 3 anlamlı basamak veren belirli bir ε_s hata kriterinden daha küçük oluncaya kadar terim eklemeye devam edin.

Sayısal Hata Türleri

❖ Çözüm 4:

- ❑ İlk olarak sonucun en az 3 üç anlamlı basamak için doğru olmasını garanti eden hata kriteri belirlenebilir.

$$\epsilon_s = \%(0.5 * 10^{2-n})$$

$$\epsilon_s = \%(0.5 * 10^{2-3}) = \%0.05$$

- ❑ Bu seviyenin altına inene kadar terim eklenmeye devam edilir.

Sayısal Hata Türleri

❖ **Çözüm 4:** İlk terimden başlayarak terim ekleme,

□ İlk terim için; $e^x = e^{0.5} = 1$,

➤ Bağıl yüzde hata,
$$\% \epsilon_r = \frac{1,648721 - 1}{1,648721} * \%100 = \%39.34$$

□ İlk iki terim için; $e^x = 1 + x \Rightarrow e^{0.5} = 1 + 0.5 = 1.5$

➤ Bağıl yüzde hata,
$$\% \epsilon_r = \frac{1,648721 - 1.5}{1,648721} * \%100 = \%9.02$$

□ Yaklaşık bağıl yüzde hatanın hesaplanması:

➤ 1. adımda elimizdeki tek tahmin değeri bulunduğundan yaklaşık hata hesaplanamaz.

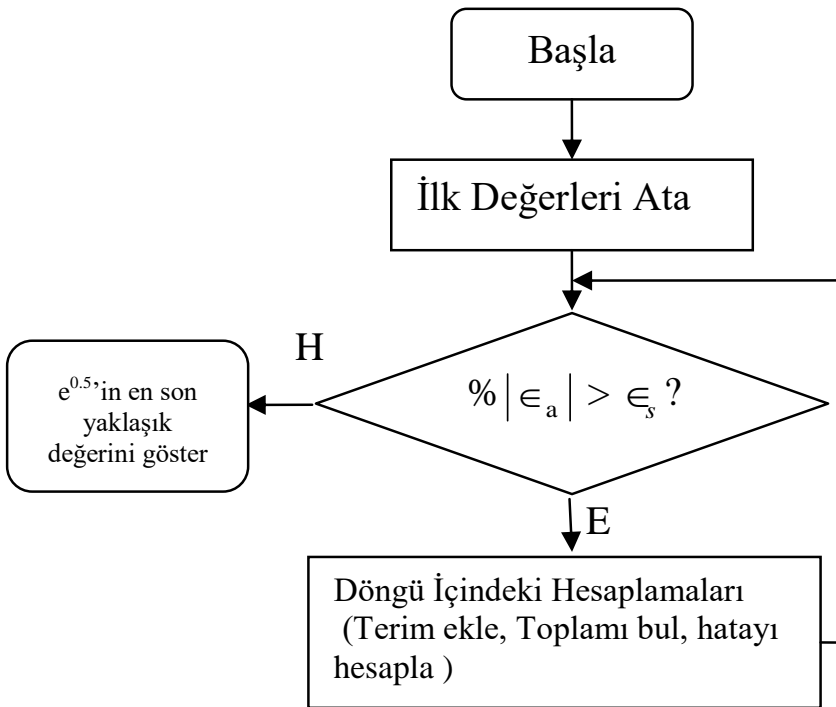
➤ 2. adıma geçtiğimizde, ilk adımda bir önceki tahmin değerine sahip olduğumuzdan yaklaşık hata hesaplanabilir.
$$\epsilon_a = \frac{1.5 - 1}{1.5} * \%100 = \%33.3$$

➤ Bu değer, % 0.05 hata sınırından küçük olmadığından yeni terim eklenmeye devam edilir

$$\epsilon_a > \epsilon_s \quad e^{0.5} = 1 + 0.5 + \frac{0.5^2}{2} = 1.625$$

Sayısal Hata Türleri

❖ Örnek 4'ün Çözüm Algoritması ve MATLAB Programı



```
Editor - D:\MATLAB7\work\programlar\bolum3\McLaurin.m*
File Edit Text Cell Tools Debug Desktop Window Help
[Icons]
1 % Mc Laurin Serisi Programi
2 % İlk Değerlerin Atanması
3 - a1=0; a0=0; Terim=0; x=0.5; n=0; es=5e-4; ea=5e-3;
4 %burada ea mutlak yüzde yaklasim hatasidir!
5
6 %Döngü baslangici
7 - while ea > es
8 - Terim=(x^n)/factorial(n)
9 - a1=a0+Terim
10 - ea=100*abs((a1-a0)/a1)
11 - a0=a1;
12 - n=n+1;
13 - end
14 - n
script Ln 13 Col 4 OVR
```

Sayıların Bilgisayarda Gösterilmesi

❑ Matematiksel hesaplamalarda farklı sayı sistemleri kullanılabilir.

- Hexadecimal(16), Decimal(10), Octal(8) ve binary(2)

❑ Decimal olarak 123 sayısının gösterimi;

- $123 = 1*100 + 2*10 + 3*1$
 $= 1*10^2 + 2*10^1 + 3*10^0$

❑ Bilgisayarlar ise ikilik (**binary**) sayı sistemine göre çalışır.

➤ 123 sayısının bilgisayarda binary olarak gösterimi;

- $123 = 1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$
 $= (1111011)_2$

➤ *Elektronik hesaplayıcılarda sayılar iki tabanında ancak belirli uzunlukta ifade edilebilir. Örneğin, reel sayılar için normal hassasiyette 32 bitlik bir yer ayrılan hesaplayıcıda 7 ondalık basamağa, çift hassasiyette ise 64 bitlik yer ayrılır ve buda yaklaşık 15 ondalık basamağa karşılık gelir. Bu nedenle değerler için hesaplayıcılardaki ayrılan yerler veri tipine göre değişmektedir. Buda farklı bir türde hataya neden olabilmektedir.*

Sayıların Bilgisayarda Gösterilmesi

- ❑ Sayıların temsili;
- ❑ 0 sayısı toplam olarak normal notasyona sahiptir. 0.10^0
- ❑ Bir sayının normal ya da kayan noktalı formda temsili
 - $x = 0.d_1 d_2 d_3 \dots d_k \times 10^n$
 - $d_1 \neq 0$ ve $d_k \neq 0$ la bir sayının önemli basamakları olduğunu ve sayının gerçek değerine göre güvenle kullanılabilen basamaklarıdır.
 - Ondalık noktaları kaydırmak için kullanılan 0 noktaları önemli basamaklar olarak sayılmaz. Kalan sıfırlar önemli olabilirde olmayabilirde.
 - Örnek;
 - $x=0,0002815 \rightarrow 4$ Önemli figüre sahiptir.
 - $x=1,200 \rightarrow 4,3$ veya 2 önemli figüre sahip olabilir.

Sayıların Bilgisayarda Gösterilmesi

❑ Sayıların Kayan Noktalı Gösterilimi (floating Point)

➤ $x = (\text{işaret}) m \cdot b^{(\text{işaret})E}$

➤ Burada m mantis, b taban (binary sistem için $b=2$) ve E üs'dür.

➤ 8 bitlik işlemciler için kayan noktalı gösterim,

Mantis işareti	Mantis basamakları				Üs işareti	Üs basamakları	
d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8

❑ Örnek:

Mantis işareti	Mantis basamakları				Üs işareti	Üs basamakları	
0	0	0	0	1	1	1	1

▪ $m = +(0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)$

$m = +(0 + 0 + 0 + 1) = 1$

$E = -[(1 \times 2^1) + (1 \times 2^0)] = -(2+1) = -3$

sayı = 1×2^{-3} (desimal sistemde 0,1250 sayısına eşittir.)

Prof.Dr. Asaf Varol'un Sunusundan Alınmıştır.



Kesme ve Yuvarlama Hataları

- ❑ Bilgisayarlar, işleyebildikleri kelime uzunluğunun sınırlı sebebiyle (sınırlı sayıda bilgiyi saklayabilmek) sayı değerlerinin sonlu olması gerekir.
- ❑ Hesaplama sırasında bir sayının izin verilemeyen (saklanamayan) kısmının atılması, hesaplanan değerin hatalı olarak bulunmasına sebep olur.
- ❑ Bu durumda karşılaşılan hatalara kesme ve yuvarlama hataları denir.
- ❑ Kesme hatası,
 - $1.26 \Rightarrow 1.2$ olarak algılanır. Bu durumda hata= 0.06
 - $1.21 \Rightarrow 1.2$ olarak algılanır. Bu durumda hata= 0.01
- ❑ Yuvarlama hatası,
 - $1.26 \Rightarrow 1.3$ olarak algılanır. Bu durumda hata= 0.04
 - $1.21 \Rightarrow 1.2$ olarak algılanır. Bu durumda hata= 0.01

Kesme ve Yuvarlama Hataları

❑ Yuvarlama Hatası:

- Örnek olarak $1/3$ kesrini bilgisayar 0.33333... gibi belli adet hane kullanarak yazar. Sayıların tanımlanması için kaç hane kullanılacağı rakamların nasıl tanımlandığı ve bilgisayarın mimarisi ile ilgilidir.
- Bu tür hatalara yuvarlama hatası (round-off error) denir.

Kesme ve Yuvarlama Hataları

- ❑ Kesme hatası, verilerin sayısal işlemlere girmesinden kaynaklanan hatadır.
- ❑ Sonsuz terimli bir seriyi uygun şekilde keserek sayısal sonuçlar elde edilir.
- ❑ Belirli terimden sonra gelen terimlerin ihmal edilmesi kesme hatası olarak bilinir.
- ❑ Burada yapılan hata atılan terimlerin toplamı kadar olur.
- ❑ Hesaplama ihmal edilen terimlerin toplamı yapılan kesme hatasına eşit olur.
- ❑ **Örnek:** Kesme hatasının oluşması:

Terim. S.	Fonksiyon	Değeri
1	$\frac{\pi}{7}$	= 0.4487989505
2	$\frac{\pi}{7} + \frac{1}{3!} \left(\frac{\pi}{7}\right)^3$	= 0.4337327325
3	$\frac{\pi}{7} + \frac{1}{3!} \left(\frac{\pi}{7}\right)^3 - \frac{1}{5!} \left(\frac{\pi}{7}\right)^5$	= 0.4338844648
4	$\frac{\pi}{7} + \frac{1}{3!} \left(\frac{\pi}{7}\right)^3 - \frac{1}{5!} \left(\frac{\pi}{7}\right)^5 + \frac{1}{7!} \left(\frac{\pi}{7}\right)^7$	= 0.4338837371
5	$\frac{\pi}{7} + \frac{1}{3!} \left(\frac{\pi}{7}\right)^3 - \frac{1}{5!} \left(\frac{\pi}{7}\right)^5 + \frac{1}{7!} \left(\frac{\pi}{7}\right)^7 - \frac{1}{9!} \left(\frac{\pi}{7}\right)^9$	= 0.4338837391
6	$\frac{\pi}{7} + \frac{1}{3!} \left(\frac{\pi}{7}\right)^3 - \frac{1}{5!} \left(\frac{\pi}{7}\right)^5 + \frac{1}{7!} \left(\frac{\pi}{7}\right)^7 - \frac{1}{9!} \left(\frac{\pi}{7}\right)^9 + \frac{1}{11!} \left(\frac{\pi}{7}\right)^{11}$	= 0.4338837391

Kesme ve Yuvarlama Hataları

- ❑ **Örnek:** Körfez savaşı esnasında (25.02.1991), kesme hatası sebebiyle patriot füzesi, skud füzesini algılayamadı.



- ❑ Bilgisayar saatinden alınan değer $1/10$ ile çarpılarak geçen süre hesaplanmaktaydı.
- ❑ Bilgisayar 24-bit sabit kayıtcı kullanmaktaydı.

Kesme ve Yuvarlama Hataları

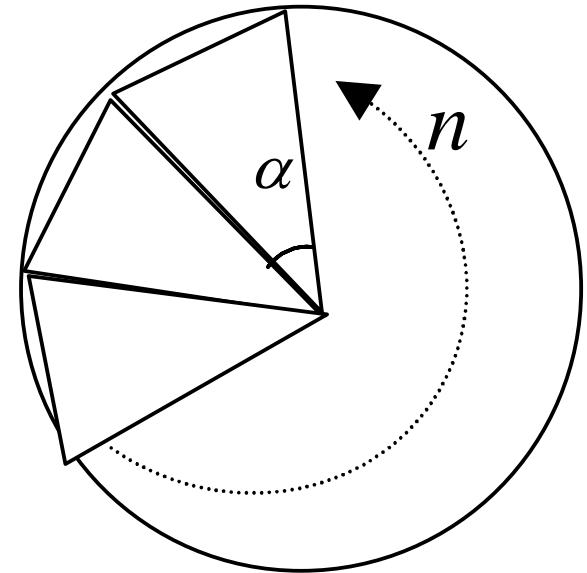
- ❑ 1/10 sayısı 2'li tabanda $1/2 + 1/2 + 1/2 + 1/2 + 1/2 + 1/2 + \dots$ şeklinde sonsuz terimlidir.
- 0.0001100110011001100110011001100....şeklinde
- ❑ Sayı: 0.0001100110011001100110011001100....
- ❑ 24-bit: 0.00011001100110011001100
- ❑ Kesme hatası: 0.00000000000000000000000011001100...
- ❑ 100 saatlik batarya süresince bu hata: $0.000000095 \times 100 \times 60 \times 60 \times 10 = 0.34$ sn
- ❑ Saniyede 1,676 metre yol alan skud füzesi 0.34 saniyede 500 metreden fazla gider.
- ❑ Bu kadarlık hata ise Skud'un menzil dışında görülmesine yol açar.

Örnek

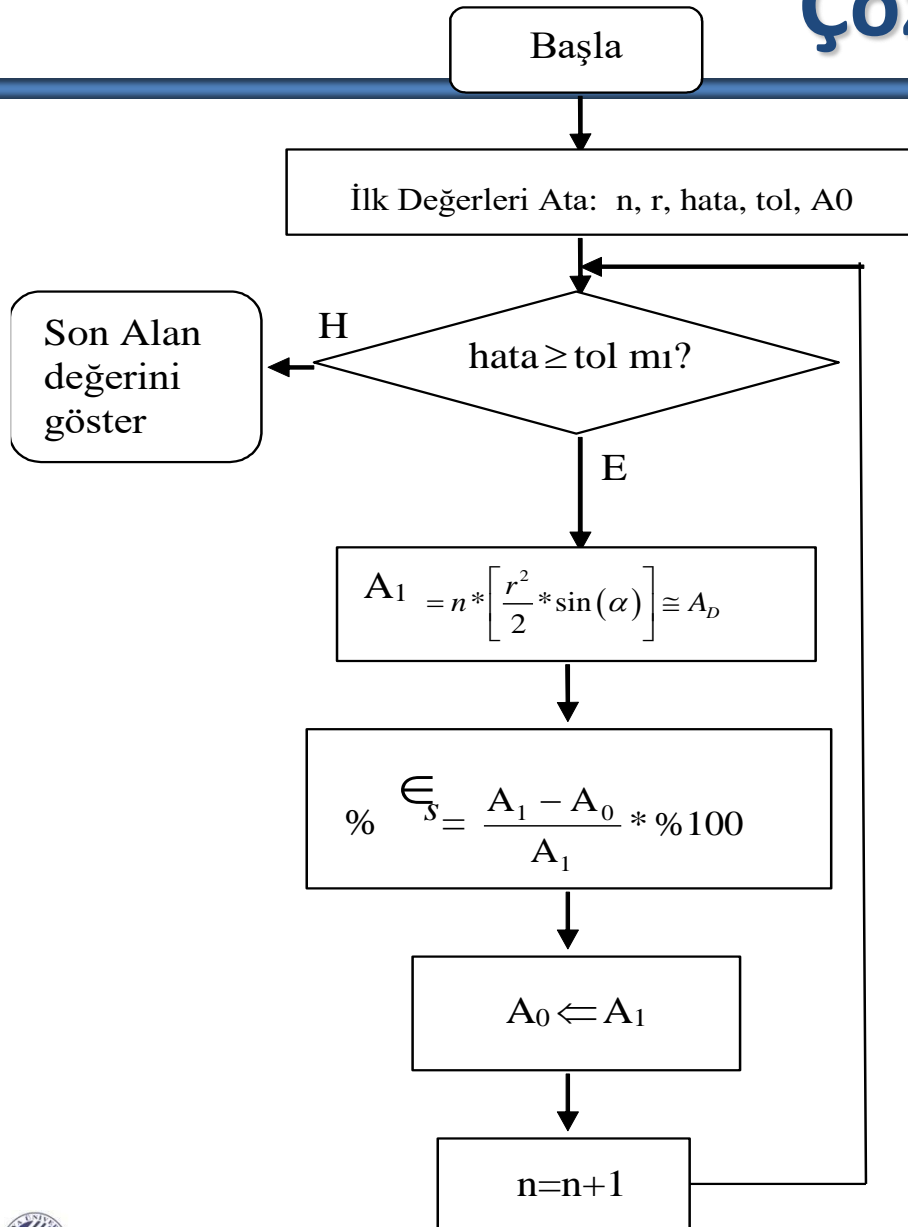
- ❑ Bir matematikçi dairenin alanını analitik olarak $A_D = \pi r^2$ bulmuştur.
- ❑ Sadece ikiz kenar üçgenlerin alanını hesaplayabilen bir bilgisayarla bu dairenin alanını $\% \epsilon_s \leq 0.01$ hata ile bulabilecek bir algoritma oluşturun ve programını MATLAB programlama dilinde yazınız.
- ❑ Program, her adımda daireyi üçgenlerden oluşan n eşit parçaya bölsün, istenen doğruluk sağlanmadıkça n 'i 1 arttırsın ($r=1$ cm olsun, n , 5'ten başlasın). Daireyi yaklaşık olarak oluşturacak n tane üçgenin alanı ;

$$n * A_{\text{üçgen}} = n * \left[\frac{r^2}{2} * \sin(\alpha) \right] \cong A_D$$

$$\alpha = \frac{2\pi}{n}$$



Çözüm



```
C:\Documents and Settings\asonda...
File Edit View Text Debug Breakpoints Web
Window Help

1 |
2 | clc; close all; clear all;
3 |
4 | n=6; hata=0.1; r=1; tol=0.01;
5 | Alan_gercek=(pi*r^2);
6 | A0=0;
7 |
8 | while hata>=tol
9 |     A1=n*(0.5*r^2*sin(2*pi/n));
10 |     hata=((A1-A0)/A1)*100;
11 |     A0=A1;
12 |     n=n+1;
13 | end
14 | Alan_gercek
15 | Hesaplanan_Alan=A1
16 | hata
17 | n

script Ln 1 Col 1
```

Çalışma Sorusu

- ❑ $f(x) = e^{-1,5}$ fonksiyonu için;
- ❑ İlk terimden başlayarak, 4 terim için adım adım $f(x) = e^{-1,5}$ fonksiyonunu hesaplayınız.
- ❑ Gerçek değer olarak 0.22313 kabul edip, her adım da mutlak hata, yüzde bağıl hatayı hesaplayınız.
- ❑ Gerçek değeri bilmediğinizi kabul edip her adım da yaklaşım hata ve yüzde yaklaşım bağıl hatayı hesaplayınız.
- ❑ Her iki çözümde de varsa anlamlı basamak sayısını gösteriniz.

$$e^{-x} = 1 - x + \frac{x^2}{2} - \frac{x^3}{3!} + \dots$$

$$e^{-x} = \frac{1}{e^x} = \frac{1}{1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \dots}$$

KAYNAKLAR

- Serhat YILMAZ, “*Bilgisayar İle Sayısal Çözümleme*”, Kocaeli Üniv. Yayınları, No:168, Kocaeli, 2005.
- Steven C. Chapra, Raymond P. Canale (Çev. H. Heperkan ve U. Kesgin), “*Yazılım ve Programlama Uygulamalarıyla Mühendisler İçin Sayısal Yöntemler*”, Literatür Yayıncılık.
- İlyas ÇANKAYA, Devrim AKGÜN, Sezgin KAÇAR “*Mühendislik Uygulamaları İçin MATLAB*”, Seçkin Yayıncılık
- Yüksel YURTAY, Sayısal Analiz Ders Notları, Sakarya Üniversitesi
- Prof.Dr. Asaf Varol, Sayısal Analiz Ders Notları, Fırat Üniversitesi