

3. MATLAB İLE PROGRAMLAMA

MATLAB aracılığı ile program yazarken ve hesaplama yaparken Amerikan alfabesinde bulunan karakterler, sayılar, aritmetik işaretler ve bazı özel karakterler kullanılmaktadır. Özel karakterlerden bazıları `:,!, %,;, [,], (,), &, {, }, '` şeklinde sayılabilir. Bunların yerine getirdiği işlemler yeri geldikçe anlatılacaktır. Daha önce hatırlatıldığı üzere dosya ve değişken isimleri ile tüm işaretlemelerde Türkçe karakterler **kullanılmamaktadır**. Temel aritmetik işlemlerin yürütülmesinde kullanılan işaretler ile MATLAB yazılışları Çizelge 3.1'de verilmiştir.

Çizelge 3.1 Aritmetik işlemler

İşaret	İşlem	MATLAB Yazılışı
^	Üs alma: x^y	<code>x^y</code>
.^	(elemanter)	<code>x.^y</code>
*	Çarpım : xy	<code>x*y</code>
.*	Elemanter çarpım	<code>x.*y</code>
/	Bölme : x/y	<code>x/y</code>
./	Elemanter bölme	<code>x./y</code>
+	Toplama: $x+y$	<code>x+y</code>
-	Çıkarma: $x-y$	<code>x-y</code>

Çizelgede `.` ile başlayan ifadeler tek tek düzey elemanları üzerine uygulandığından elemantar (element-wise) işlemlerdir. Toplama ve çıkarma işlemlerinin zaten karşılıklı elemanlara uygulanma zorunluluğu olduğundan özel bir işarete gerek bulunmamaktadır.

Yukarıdaki çizelgede verilen işlemlerin birbirlerine göre öncelikleri söz konusudur. MATLAB ifadeleri soldan sağa doğru işletilmektedir. Bunun dışında işlem sırası:

1. üs alma
2. çarpma&bölme
3. toplama&çıkarma

şeklinde dir. İşlemlerin sırası parantez kullanılarak değiştirilebilir. Parantez içine yazılan ifadeler birincil öncelik kazanır. Öncelik sırası değiştirildiğinde işlem sonucu değişebilmektedir. Bu nedenle işlem sırası ile ilgili emin olmadığınız durumlar için parantez kullanmakta yarar vardır. Örneğin $\Delta = b^2 - 4ac$ olmak koşulu ile $\frac{-b + \sqrt{\Delta}}{2a}$ ifadesi MATLAB ile hesaplanmak istenirse işlem sırası ve parantez kullanımına bağlı olarak aşağıdaki sonuçlar elde edilir:

```

a=4;
>>b=2;
>>c=-1;
>>delta=b^2-4*a*c
ans =
20
>> -b+sqrt(delta)/2*a %Önce bölme ve çarpma sonra toplama yapılır
ans =
6.9443
>> (-b+sqrt(delta))/2*a %Önce toplama sonra bölme sonra çarpma yapılır
ans =
4.9443
>> (-b+sqrt(delta))/(2*a) %Önce parantez içleri en son bölme yapılır
ans =
0.3090

```

Burada doğru yazılış ve sonuç son adımda yazıldığı şekildedir. Verilen örnekte ilk üç satırda bir değişkene değer atama işlemi görülebilir. Diğer taraftan ifadelerin sonunda yer alan noktalı virgül (;) işleci, işlem sonucunun komut satırında görünmesini engeller. Bu işaret aynı zamanda satır sonunu belirtmek için de kullanılır. Birden çok işlem aralarına ; işareti konularak aynı satırda yazılabilir:

```
>>a=4;b=2;c=-1;
```

MATLAB kütüphanesinde özel işlem ve hesaplamaların yapılabilmesi için yazılmış yüzlerce hazır fonksiyon bulunmaktadır. Bunlardan sık kullanılan matematik fonksiyonların bir listesi Çizelge 3.2'de verilmiştir.

Çizelge 3. 2 Sık kullanılan bazı matematik fonksiyonların MATLAB karşılıkları

Fonksiyon	MATLAB Yazılışı
e^x	exp(x)
\sqrt{x}	sqrt(x)
$ x $	abs(x)
$\log_{10}(x)$	log10(x)
$\ln(x)$	log(x)
$\sin x$	sin(x)
$\cos x$	cos(x)
$\tan x$	tan(x)
$\sin^{-1} x$	asin(x)
$\tan^{-1} x$	atan(x)

Çizelgede verilen trigonometrik fonksiyonların argümanının radyan cinsinden olduğu kabul edilmektedir. Açılan derece cinsinden girmek için aynı fonksiyon isimlerinin sonuna d harfi eklenerek çağrılabilir. Örneğin 60° nin sinüsünü hesaplamak için `sind(60)` ifadesi yazılabilir.

3.1 Değişkenler

Değişkenler MATLAB çalışma belleğinde sayısal veya alfa numerik bilgileri saklamak için kullanılan sembollerdir. Varsayılan olarak yaratılan sayısal bir değişkenin türü `double` olarak atanmaktadır. `double` çift duyarlıklı gerçel sayıları ifade etmektedir. Bu sebeple tam sayı (integer), gerçel sayı (real) gibi değişkenler için özel tanımlamalar yapmak gereksizdir. Değişken isimlerinde kullanılan harf ve karakterler de değişken türü açısından bir tanımlama yapmazlar. Değişken isimlendirme ile ilgili uyulması gereken kurallar:

1. Değişken isimleri küçük büyük harf duyarlıdır (A≠a)
2. Değişken isimleri sayı ile başlayamaz
3. Boşluk, `+`, `-`, `!`, `*`, `(`, `)` gibi özel karakterler değişken isminde kullanılamaz. `_` değişken isimlendirmede kullanılabilir bir işarettir.
4. `end`, `if`, `for` gibi MATLAB deyimleri ve kütüphane fonksiyonları ile aynı isimde değişken oluşturulamaz

şeklinde sıralanabilir. Aşağıdaki çizelgede doğru ve yanlış değişken isimlendirme örnekleri verilmiştir:

Çizelge 3. 3 Örnek değişken isimlendirme

Doğru	Yanlış
<code>g1</code>	<code>1g</code>
<code>F4</code>	<code>for</code>
<code>EndPoint</code>	<code>end</code>
<code>IfVal</code>	<code>if</code>
<code>N_plus</code>	<code>N+</code>
<code>k_fact</code>	<code>k!</code>
<code>d_2</code>	<code>d 2</code>

Ön tanımlı değişkenler:

MATLAB çalışma oturumu çeşitli amaçlar için kullanılmak üzere bazı ön tanımlı değişkenlerle birlikte başlatılmaktadır. `pi`, `inf`, `i`, `NaN` bunlara örnek olarak verilebilir.

Çizelge 3. 4 Ön tanımlı MATLAB değişkenleri

Değişken	Karşılığı
<code>pi</code>	3.1415926535
<code>inf</code>	∞
<code>-inf</code>	$-\infty$
<code>i</code>	$\sqrt{-1}$
<code>NaN</code>	Not a Number (Sayı değil)

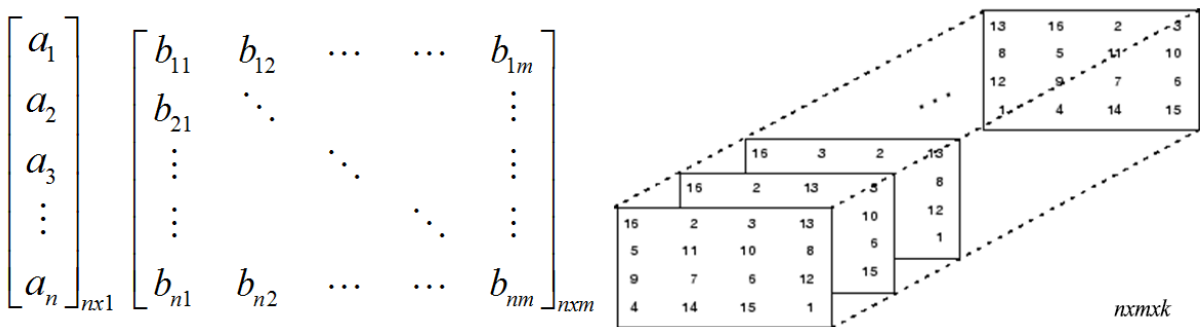
MATLAB programlama ortamında kullanılan belli başlı deęişken türleri double, character, cell, structure, double complex, logical şeklinde sıralanabilir. Sıralanan türde deęişkenler oluşturarak bellekte bilgi saklamak daha sonraki bölümlerde ayrıntılı olarak anlatılacaktır. Bellekte bulunan deęişkenlerin türleri ve içerikleri ile ilgili genel bilgiler "Workspace" penceresinden takip edilebilmektedir. Örnek bir görünüm Şekil 3.2'de verilmiştir. MATLAB veri yapısı içerisinde burada sayılmayan ve ileri düzey hesaplama ve programlama işlemlerinde kullanılan çok sayıda deęişken türü daha bulunmaktadır. Bununla ilgili MATLAB yardım belgelerine bakılabilir.

Name	Size	Class	Value	Min	Max
C	3x8	double (complex)	<3x8 double>	0.0357 + 1.0000i	0.9706...
K	1x6	char	'Başlık'		
SINIF	1x1	struct	<1x1 struct>		
a	3x8	double	<3x8 double>	0.0357	0.9706
d	1x4	cell	<1x4 cell>		
f1	3x8	logical	<3x8 logical>		

Şekil 3.1 Bellekte bulunan deęişkenlerin türleri ile birlikte "Workspace" penceresindeki görünümü

3.2 Yöney ve Dizely İşlemleri

MATLAB geçerli yollardan biri ile oluşturulmuş tüm deęişkenleri bir dizely olarak algılar ve tüm işlemlerde de bu şekilde değerlendirir. Yöneyle (vektör) yalnızca bir satır ya da bir sütundan oluşan dizelylerdir. Diğer bir söyleyişle $n \times 1$ ya da $1 \times n$ boyutlarında dizelylere yöney adı verilir. İki boyutlu dizelyler ise n satır, m ise sütun sayısını göstermek üzere $n \times m$ boyutlarındadır. Üç ve daha çok boyutlu dizelyler oluşturmak ve kullanmak da mümkündür.



Şekil 3.2 Bir-,iki- ve üç-boyutlu dizelyler

3.2.1 Dizi Oluşturma

MATLAB'da tüm deęişkenlerin dizi olarak saklandığı daha önce belirtilmişti. Dolayısı ile bir deęişkene deęer atama işlemleri ile diziler oluşturabilmektedir. Birincil deęer atama işlemleri = işaretidir. Bu işaretin

solunda yer alacak geerli bir deęiřken adına iřaretin saęında yer alan geerli sayı ya da iřlemin sonucu atanır. rneęin:

```
A1=5.6;
f=0.5*exp(-2*pi*fo);
```

Verilen rneklere iřlemlerin sonuları birer skaler olup 1x1 boyutlarında A_1 ve f dizilerinde bellekte saklanır. Gerel sayılardan oluřan yney ve dizeler oluřturmak iin ařaęıdaki rneklere incelenebilir:

```
ilk_yoney=[1 8 2.7 6.8 -0.2];
dizey1=[14 -8 6 0.5; 0.1 14 -0.8 1; 22 0.78 5 0];
```

Burada nceki rneklere farklı olarak kullanılan kşeli parantezler, ilerine yazılan bilgilerin bir dizinin elemanları olduęunu gstermektedir. Dięer bir deyiřle birden ok elemanı olan bir dizinin elemanları, kşeli parantezler ierisine yazılır. Dizinin elemanları birbirinden bořluk ya da virgl ile ayrılabilir. Kşeli parantez ierisinde kullanılan ; iřleci satırın bittięi ve alt satıra geildięi anlamına gelir. Verilen rneklere **ilk_yoney** 1x5 boyutlarında, **dizey1** ise 3x4 boyutlarındadır. Satır sonlarındaki ; ise iřlem sonucunun komut satırında grntlenmesini nlemek iindir. Iřlem sonucu zel olarak grntlenmek isteniyor ise silinebilir.

Karakter dizileri:

MATLAB karakter ve karakter dizilerini saklayan deęiřken oluřurmaya izin vermektedir. Karakter trnde deęiřkenler tek tırnak ' ' iřaretleri arasına yazılarak atanırlar. rneęin

```
isim='John Smith';
```

ifadesi ile 1x10 boyutunda bir karakter dizisi oluřturulabilir. Sayısal deęiřkenlerde saklanan deęerler farklı alanlarda (rneęin deęiřken ya da Őekilleri isimlendirmek gibi) kullanılmak zere karakter trnde deęiřkenlere dnřtrlebilir. Bunun iin `num2str` MATLAB fonksiyonu kullanılabilir.

```
mesaj=['Ocak ayı sıcaklık ortalaması', num2str(11), 'derecedir'];
```

rnekte de grldę zere birden ok karakter dizisi birleřtirilerek yeni bir karakter dizisi de elde edilebilmektedir.

Hcre tr veriler:

Geleneksel olarak bir dizinin elemanlarının tamamının aynı trden olması gereklidir. rneęin $n \times m$ boyutlarında bir dizinin tm elemanlarının sayısal olması gerekir ve her bir adreslenebilir birimde bir adet numerik veya alfanumerik bilgi saklanabilir. Ancak yeni nesil programlama dillerinde farklı trden

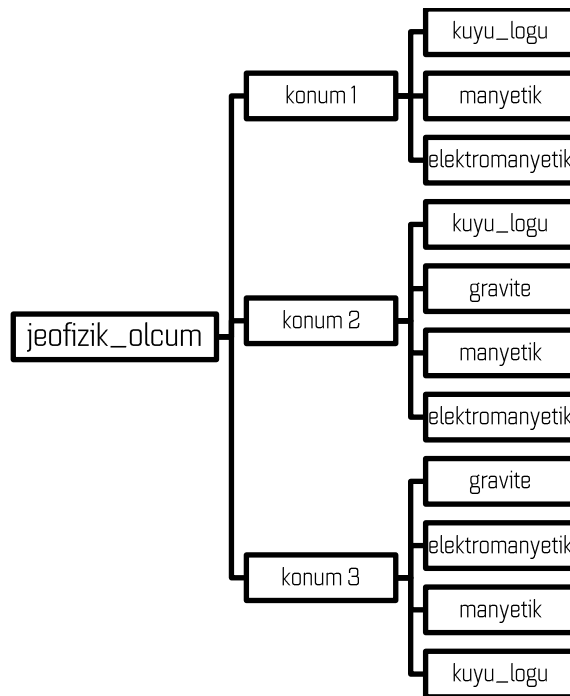
ve boyutta bilgiler tek bir değişkenin hücre (cell) olarak adlandırılan birimlerinde saklanabilir. MATLAB çalışma ortamında hücre türü değişkenler { } parantezleri içine yazılarak oluşturulabilir. Örneğin:

```
>> a={'isimler',pi, [1 2 3; 6 8 10]}
a =
    'isimler'    [3.141592653589793]    [2x3 double]
```

ifadesi ile 1x3 boyutlarında bir hücre dizisi oluşturulabilir. Bu dizinin her bir hücresinde farklı tür ve boyutta veri saklanılabildiği örnekte görülmektedir. Bu tür dizilerin hücrelerine ve hücre içeriğine erişim ile ilgili bilgiler daha sonra örneklerle açıklanacaktır.

Yapısal değişkenler (structure):

MATLAB altında tanımlı yapısal değişkenler çok kapsamlı veri kümelerinin tek bir isim altında saklanmasına olanak vermektedir. Bu tür değişkenler ile veri tabanı mantığında bilgi saklanabilmektedir. Aşağıda bir jeofizik araştırma sırasında toplanılan çeşitli veriler sistematik olarak sınıflandırılmıştır.



Üç farklı konumda, çeşitli yöntemlerden elde edilen sayısal bilgilerden oluşan bu veri kümesi yapısal tek bir değişken ile karşılanabilir. Yapısal değişkenlere alt alan isimleri eklemek mümkündür. Her bir alt alan içerisinde tür ve boyutu birbirinden bağımsız bilgiler saklanabilmektedir. Yukarıdaki veri kümesini MATLAB çalışma ortamında bir yapısal değişken ile tanımlamak amacı ile aşağıdaki örnek atama işlemleri takip edilebilir. Burada sadelik açısından her bir konum için birer örnek verilmiştir. Diğer taraftan her alt alan tüm konumlarda olmak zorunda değildir.

```
>>jeofizik(1).kuyu_logu.derinlik=[0 .5 1 2 5 10 20];
>>jeofizik(1).kuyu_logu.yogunluk=[2.2 2.36 2.33 2.41 2.40 2.41 2.5];
>>jeofizik(2).gravite.koordinat=[30.72 28.45 ];
>>jeofizik(2).gravite.okuma=[100.46];
>>jeofizik(2).gravite.notlar='yoğun yapılaşma bulunmaktadır';
>>jeofizik(3).elektromanyetik.frekanslar=[0.01 0.1 1 ];
>>jeofizik(3).elektromanyetik.empedans=[214+26i, -0.05+6i, 102-63i];
```

Örneklerden görüleceđi üzere yapısal deđiřkene \.' iřareti ile ayırarak alt alanlar (field) eklenebilmektedir. Yukarıda belirtildiđi řekilde oluřturulan yapısal deđiřkenin alan ve elemanlarına eriřim için ařađıdaki örnekler incelenebilir. Örnekte verilen jeofizik alıřmaların konum 1' de kaydedilen sonuçlarını görüntülemek üzere

```
>>jeofizik(1)
```

yazıldıđında deđiřkenin altında saklanan alanlar ařađıdaki řekilde görüntülenir:

```
ans =
    kuyu_logu: [1x1 struct]
    gravite: [1x1 struct]
    elektromanyetik: [1x1 struct]
```

alt alanlar da yapısal oldukları için onların altındaki alanlar için bir basamak daha ilerlemek gerekir. Örneđin konum 1'deki gravite alanında saklanılan bilgileri görmek için

```
>>jeofizik(1).gravite
```

yazılırsa

```
ans =
    koordinat: [30.7200 28.4500]
    okuma: 100.4600
    notlar: [1x48 char]
```

sonucu görüntülenir. Daha alt alanlara erişmek için aynı yol izlenebilir.

3.2.2 Özel dizeler oluşturma

Periyodik artan ya da azalan diziler:

Belirli bir adım ile artan ya da azalan diziler oluşturmak için `:` işleci kullanılır. Bu operatör MATLAB program yazımında ve komut satırı işlemlerinde çeşitli amaçlarla sıkça kullanılmaktadır. Bir başlangıç değerinden bir bitiş değerine bir artım miktarı ile değişen bir dizi oluşturmak için `x1:dx:x2` yazımı kullanılır. Burada `x1` başlangıç değeri, `dx` artım miktarı, `x2` ise bitiş değeridir. Konunun anlaşılması için aşağıda çeşitli örnekler verilmiştir:

```
>> teta=-pi:pi/4:pi
teta =
  Columns 1 through 4
   -3.1416   -2.3562   -1.5708   -0.7854
  Columns 5 through 9
    0    0.7854    1.5708    2.3562    3.1416
```

```
>> t=1:10
t =
  Columns 1 through 10
    1    2    3    4    5    6    7    8    9   10
```

```
>> R=[1:2:9 10:2:16]
R =
  Columns 1 through 9
    1    3    5    7    9   10   12   14   16
```

```
>> 3:-1:-3
ans =
    3    2    1    0   -1   -2   -3
```

```
>> 'a':2:'z'
ans =
acegikmoqsuwy
```


Bir aralığın belirli sayıda nokta ile örneklenmesi:

MATLAB ile alt ve üst sınırları bilinen bir aralığın belirli sayıda nokta ile tanımlanması için **linspace** ve **logspace** fonksiyonları kullanılmaktadır. Bunlardan ilki, aralığı doğrusal olarak bölerken diğeri logaritmik bir örnekleme yapmaktadır. Kullanımları aşağıdaki gibidir:

y = linspace(a,b,n) a ve b arasında doğrusal olarak örneklenmiş n adet nokta üretir (a, b dahil)
y = logspace(a,b,n) 10^a ve 10^b arasında logaritmik olarak örneklenmiş n adet veri üretir.

1 ile 5 arası 11 nokta ile doğrusal olarak örnekleme istenirse

```
>> linspace(1,5,11)
ans =
Columns 1 through 7
1.0000    1.4000    1.8000    2.2000    2.6000    3.0000    3.4000
Columns 8 through 11
3.8000    4.2000    4.6000    5.0000
```

aynı aralık logaritmik olarak aynı sayıda veri ile örneklenirse

```
>> logspace(log10(1),log10(5),11)
ans =
Columns 1 through 7
1.0000    1.1746    1.3797    1.6207    1.9037    2.2361    2.6265
Columns 8 through 11
3.0852    3.6239    4.2567    5.0000
```

elde edilir. Burada ardışık iki nokta arasındaki farkın logaritmik olarak arttığı izlenebilir.

Tüm elemanları 0 olan dizey:

```
>> s=zeros(n,m);
```

yazımı ile n satır m sütundan oluşan ve tüm elemanları sıfır olan bir dizey üretilebilir. Tüm elemanları sıfırdan oluşan dizeler bir altlık olarak kullanılmak üzere üretilmektedir. Toplamaya göre etkisiz eleman olan böyle bir dizeye istenilen bir skaler eklenerek tüm elemanları bu skalere eşit olan dizeler elde edilebilir. Diğer taraftan yalnızca belirli elemanları sıfırdan farklı olacak şekilde düzenlemek amacıyla da yine bir şablon olarak üretilip kullanılabilir.

```
>> s=zeros(3,4)
s =
     0     0     0     0
     0     0     0     0
     0     0     0     0

>> s=10+zeros(3,4)
S =
    10    10    10    10
    10    10    10    10
    10    10    10    10
```

Tüm elemanları 1 olan dizey:

Önceki fonksiyona benzer olarak tüm elemanları 1 olan bir dizey **ones (n,m)** yazılışı ile üretilebilir. Yine benzer olarak bir skaler ile çarpılarak tüm elemanları bu çarpana eşit olan dizeyler üretilebilir.

```
>> u=ones(5,4)
u =
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

```
>> u=17*ones(5,4)
u =
    17    17    17    17
    17    17    17    17
    17    17    17    17
    17    17    17    17
    17    17    17    17
```

Birim dizey oluşturma:

Birim dizey ana köşegeninde yer alan elemanları 1 diğer tüm elemanları 0 olan kare bir dizeydir. Bu tür dizeyler MATLAB kütüphanesinde yer alan bir fonksiyon ile üretilebilir. Kullanımı aşağıdaki gibidir:

```
>> I = eye(n) ;
```

ifadesi ile $n \times n$ boyutunda bir birim dizey oluşturulabilir. Birim dizey bir kare dizey olduğundan satır, sütun sayılarını ayrı ayrı vermeye gerek bulunmamaktadır. 5×5 boyutlarında birim dizey oluşturmak için aşağıdaki yazılış yeterlidir:

```
>> I=eye(5)
I =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
```

```
0 0 0 1 0
0 0 0 0 1
```

Rastgele sayılardan oluşan dizeler:

Sayısal hesaplama ve programlama sırasında çeşitli amaçları ile rastgele sayılardan oluşan dizelere gerek duyulmaktadır. MATLAB fonksiyon kütüphanesinde bu amaçla geliştirilmiş iki fonksiyon bulunmaktadır. [0 1] aralığında düzgün dağılan rastgele sayılardan oluşan n satır ve m sütun bir dize oluşturmak için `rand(n,m)` ifadesi yazılabilir. a ve b aralığında düzgün dağılan rastgele sayılardan oluşan nxm boyutlarında bir dize oluşturmak için ise aşağıdaki yazılış kullanılabilir:

```
>>r = a + (b-a).*rand(n,m)
```

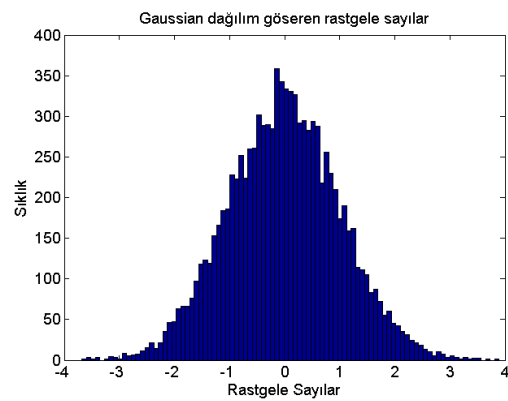
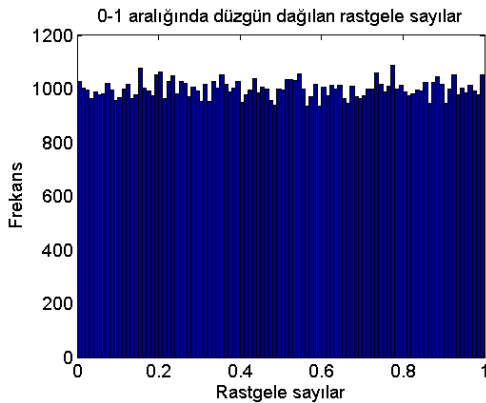
Benzer olarak sapma ve ortalaması belirli değerler olacak şekilde normal dağılım gösteren (Gaussian) rastgele sayılardan oluşan dizeler de üretilebilir. Bu amaçla `randn` fonksiyonu kullanılmaktadır. Örnek olarak standart sapması 2 ve ortalaması 1 olan ve normal dağılım gösteren 100x100 boyutlarında bir dize

```
>>r = 1 + 2.*randn(100,100);
```

ifadesi ile oluşturulabilir. Yukarıda anlatıldığı şekilde oluşturulan 100000x1 boyutlarında rastgele sayılar içeren dizelerin histogramları Şekil 3.3'de verilmiştir.

```
>> rN=rand(100000,1);hist(rN);
```

```
>>rG=randn(100000,1); hist(rG);
```



Şekil 3. 3 Düzgün (uniform) ve normal (Gaussian) dağılım gösteren rastgele sayılardan oluşan dizelerin histogramları

Dizelerin birleşiminden oluşan dizeler:

Boyutları uygun olmak kaydıyla bellekte bulunan ya da komut satırında üretilen dizeler alt alta ya da yan yana kolayca birleştirilebilir. Birleştirme işleminde `[]` ve `;` işaretleri kullanılmaktadır. Buna göre aşağıdaki örnekler durumu açıklaması için incelenebilir:

```
>>R1 = rand(3,5); % R1 3x5 boyutlarında
>>R2 = rand(8,5); % R2 8x5 boyutlarında
>>R = [R1;R2]; % R 11x5 boyutlarında
```

Burada aynı sütun sayısı olan iki dizay alt alta birleştirilerek yeni bir dizay elde edilmiştir. Bu örnekten yola çıkılarak yan yana birleştirme için satır sayılarının eşit olması gerektiği söylenebilir.

```
>> A=[1 3;5 7]
A =
     1     3
     5     7
>> B=[2 4;6 8]
B =
     2     4
     6     8
>> C=[A B]
C =
     1     3     2     4
     5     7     6     8
```

c dizeyi 2x2 boyutunda iki dizeyin yan yana birleştirilmesi ile elde edilmiştir (2x4).

3.2.3 Dizay İndeksleme

Önceki bölümlerde anlatılan tür ve yapıda oluşturulmuş MATLAB dizelerinde saklanan bilgilere erişmek ya da değiştirmek için eleman(lar)ın adresinin tarif edilmesi gereklidir. Bu işlemin en genel hali satır ve sütun numaralarının belirtilmesidir. **XLoc** iki boyutlu bir dizay olmak üzere

```
>>XLoc(5,6)
```

yazımı dizeyin 5nci satır 6ncı sütundaki elemanını belirtmektedir. Başka bir söyleyişle dizay adından sonra gelen parantezler indeksleri yazmak için kullanılmakta, ilk indeks satır numarasını, ikinci indeks sütun numarasını göstermektedir. Dizay bir boyutlu ise (yöne) adreslenebilir tek konum olduğundan **A(5)** şeklinde yazılır. Üç boyutlu bir dizay üç indeks ile tanımlanabilir. **J2(3,4,1)** örneğinde birinci indeks satır, ikinci indeks sütun ve üçüncü indeks sayfa numarasını göstermektedir (Bknz. [Şekil 3. 2](#)). Üçten daha çok indeksi olan dizeler de yaratılıp kullanılabilir. İndeksler virgüllerle ayrılarak yazılır.

Yukarıda belirtilen genel indeksleme işlemi bir defada yalnızca bir elemanı adreslemektedir. Çoklu adresleme için : işleci kullanılabilir. Oldukça büyük kolaylık sağlayan bu seçenek Çizelge 3.5'de verilen örnekler üzerinde anlatılmıştır.

Pozitif tam sayı değeri atanmış dizeler de indeks hanesine yazılabilir. Örneğin:

```
>>a = [1 3 11 15:2];
>>C(a,4) = 1;
```

Verilen örnekte a dizisinde tanımlanan numaralar C dizisinin satırlarını indekslemede kullanılmış 4ncü sütunun bu satır numaralarına gelen elemanlara 1 değeri atanmıştır. Dizay indekslemede kullanılan bir diğer deyim **end** ifadesidir. Bu ifade satır ya da sütunlardan sonuncusunu adreslemek için kullanılır.

```
>> number_of_cycles(4, 4:end) = 5;
```

örneğinde **number_of_cycles** değişkeninin 4 ten sonuncu sütuna kadar 4. satırda yer alan elemanları tarif edilmektedir.

Çizelge 3. 5 : işleci ile dizay indeksleme

İndeks	Açıklama	Yeri
Val2(1, :)	Val2 dizisinin 1. satırında yer alan tüm elemanlar	$\text{Val2} = \begin{bmatrix} 1 & 20 & 9 & 6 \\ 8 & -4 & 1 & 5 \\ 1.5 & 0 & 8.7 & 2 \\ 6 & 7 & 14 & 10 \end{bmatrix}$
D3(:, 4)	D3 dizisinin 4. Sütunundaki tüm elemanlar	$\text{D3} = \begin{bmatrix} 0.005 & 1 & 18 & -5.7 & 2.6 \\ -6.058 & 4.5 & 16 & 14.4 & -4.74 \\ 11.54 & 7.01 & 2.7 & 0.07 & 5.1 \\ 2 & 6.5 & 9 & 6.5 & 6 \\ 8.99 & 7 & 11 & 10.02 & 7.3 \end{bmatrix}$
Sta(1:3, 6)	Sta dizisinin 1den 3. Ye kadar satırları ve 6. Sütunundaki elemanlar	$\text{Sta} = \begin{bmatrix} 1.30 & 1.62 & 6.02 & 4.51 & 8.26 & 1.07 & 8.69 & 4.31 \\ 5.69 & 7.94 & 2.63 & 0.84 & 5.38 & 9.62 & 0.84 & 9.11 \\ 4.69 & 3.11 & 6.54 & 2.29 & 9.96 & 0.05 & 4.00 & 1.82 \\ 0.12 & 5.29 & 6.89 & 9.13 & 0.78 & 7.75 & 2.60 & 2.64 \\ 3.37 & 1.66 & 7.48 & 1.52 & 4.43 & 8.17 & 8.00 & 1.46 \end{bmatrix}$

YKon([1:2:5 7 8],2)

YKon dizeyinin 1,3,5 7 ve 8. satırları, ikinci sütunu

Ykon =

$$\begin{bmatrix} -1.58 & -1.75 & -0.03 & -1.12 \\ 0.51 & -0.29 & -0.80 & 2.53 \\ 0.28 & -0.83 & 1.02 & 1.66 \\ 0.03 & -0.98 & -0.13 & 0.31 \\ -1.33 & -1.16 & -0.71 & -1.26 \\ 1.13 & -0.53 & 1.35 & -0.87 \\ 0.35 & -2.00 & -0.22 & -0.18 \\ -0.30 & 0.96 & -0.59 & 0.79 \\ 0.02 & 0.52 & -0.29 & -1.33 \\ -0.26 & -0.02 & -0.85 & -2.33 \end{bmatrix}$$

Alıřtırmalar

- 0-10 aralıęında 2, 10-50 arasında 4 adım boyu ile artan bir dizi oluřturup D isimli deęiřkene aktarınız.
- $[-2\pi \ 2\pi]$ aralıęındaki aęıların sinüsünü $\pi/6$ artımlarla hesaplayan MATLAB ifadelerini yazınız.
- Önceki alıřtırmadaki aęılar için, sütunlarında sırası ile aęılar, aęıların sinüsü, kosinüsü ve tanjantı olan bir dizayn oluřturunuz.
- Ařaęıda verilen ifadelerin MATLAB karřılıklarını yazınız.

a. $y = 3x^2 + \log\left(\frac{x}{x+1}\right)$

b. $F = G \frac{m_1 m_2}{r^2}$

c. $t = \frac{x}{v_2} + \frac{2h\sqrt{v_2^2 - v_1^2}}{v_1 v_2}$

d. $\rho_a = k \frac{\Delta v}{I}$

e. $f(t) = 2 \sin(10\pi f_o t) + 0.5 \cos(4\pi f_o t)$

- $y = e^x$ fonksiyonunu x'in -3,3 aralıęındaki deęerleri için $dx=0.1$ artımlarla hesaplatınız.
- $y = e^{-i2\pi f_o t}$ fonksiyonunu $f_o = 5$ için 0-2 zaman aralıęında 0.25 adımlarla hesaplatınız.
- Bahar döneminde aldıęınız tüm derslerin arasınay, final ve bütönlöme notlarını saklayan yapısal bir deęiřken tanımlayıp, sınav notlarınızı giriniz.
- Ařaęıdaki dizeyleri oluřturayn MATLAB ifadelerini yazınız.

$$A = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \quad Z = \begin{bmatrix} 0 & 2 & 4 & 6 \\ 0 & 0 & 0 & 0 \\ 8 & 10 & 12 & 14 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 3 & 0 & 4 & 0 \\ 5 & 0 & 6 & 0 \\ 7 & 0 & 8 & 0 \end{bmatrix}$$

9. 0-100 aralığını 250 nokta ile doğrusal olarak örnekleyerek x isimli değişkene aktarınız.