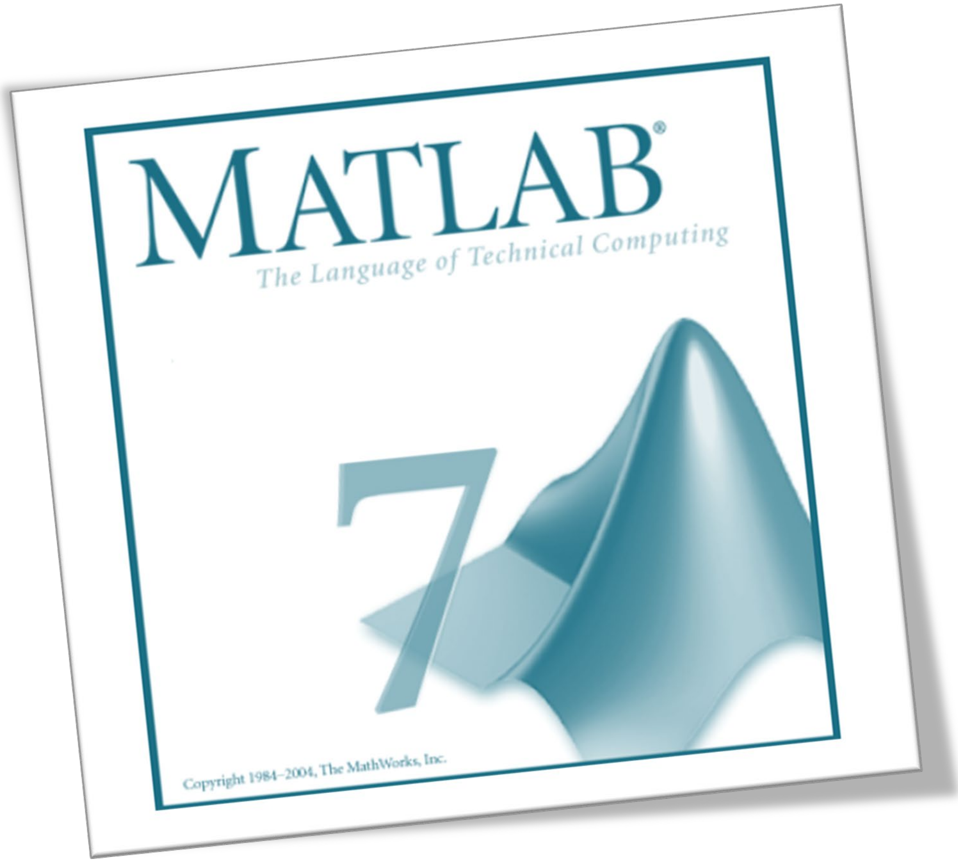
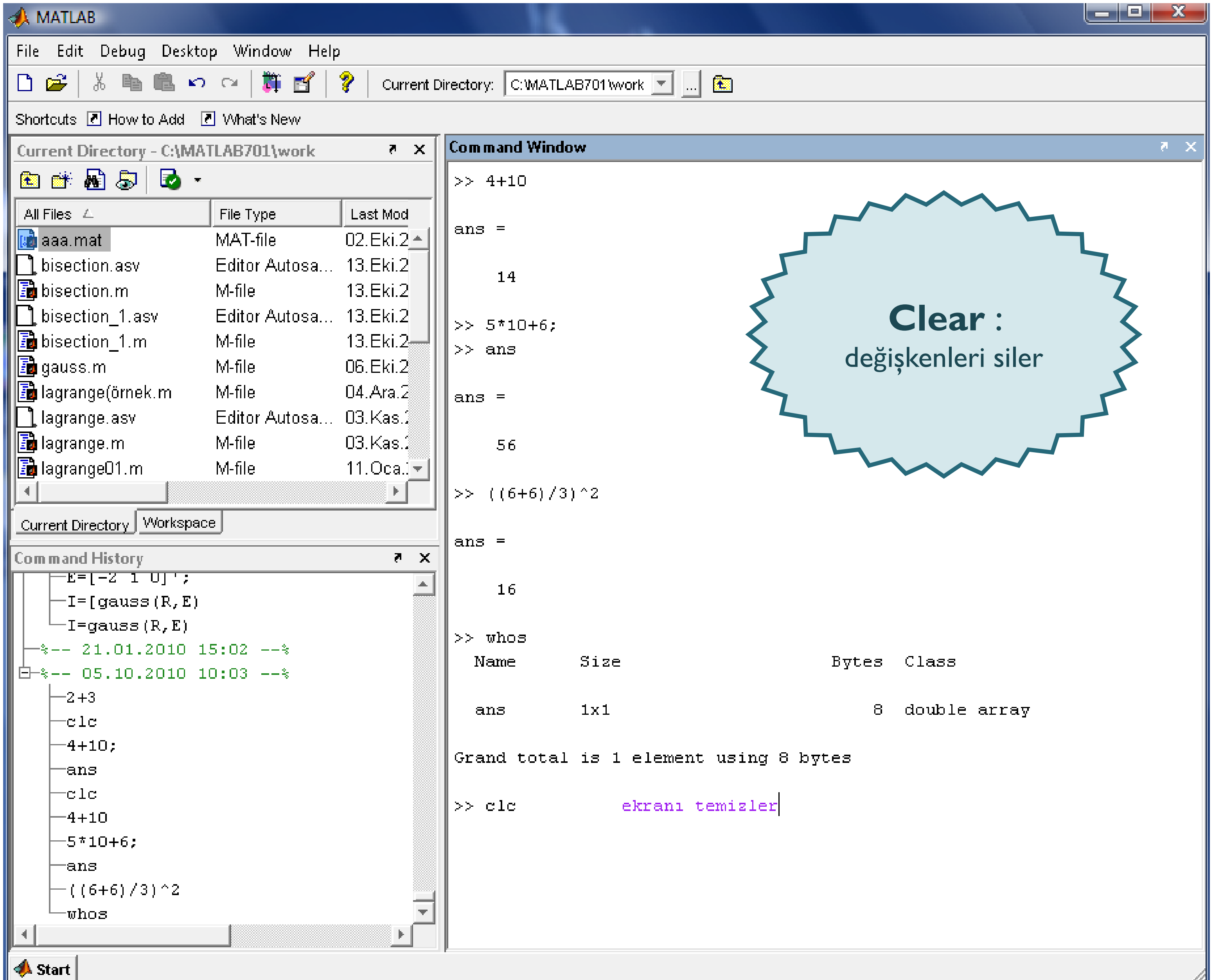
Dr. Yüksel YURTAY



Giriş

Değişkenler MATLAB'in temel kavramlarındandır,



Noktalı Virgül Kullanarak Sonuçları Gizlemek

Komuttan sonra noktalı virgül yazarsanız sonucun yazdırılmasını engellemiş olursunuz.

Örnekler:

```
>>ort = (a + b + c) / 3
ort = 20
>>a = 10;
>>b = 20;
>>c = 30;
>>d = 40;
>>ort = (a + b + c + d) / 4
ort =
25
>>the_average;
>>b
b =
20
>>e = 50
e =
50
```

Kendi değişkenlerinizi tanımlayabilmeniz ve kullanabilmeniz çok kullanışlıdır.

MATLAB'de Matris oluşturma

MATLAB'de matrisler köşeli parantezler içinde tanımlanır ([ ]).
Virgül (,), ve noktalı virgül (;) noktalama işaretleri sırasıyla satır ve sütun ayıracı olarak kullanılır.

Not: Satır ayıracı olarak virgül yerine boşluk, sütun ayıracı olarak ta alt satıra geçmeyi (enter) kullanabilirsiniz.

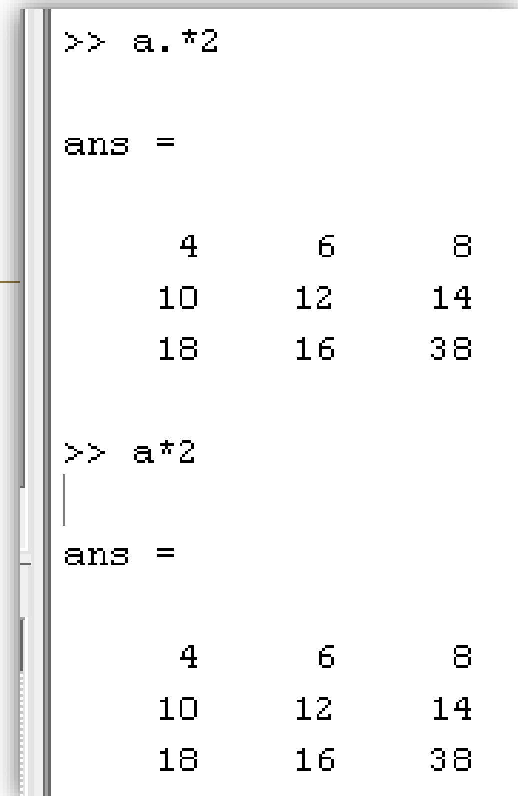
```
>>sayi = 3.1415
sayi = 3.1415

>>vektor1 = [1, 5, 7]
vektor1 =
1 5 7

>>vektor2 = [1; 5; 7]
vector2 = 1
5
7
```

```
>>matris = [8, 12, 19; 7, 3, 2; 12, 4, 23; 8, 1, 1]
matris =
8     12     19
7      3      2
12     4     23
8      1      1
```

```
>>matris = [8, 12, 19; 7, 3, 2; 12, 4, 23; 8, 1, 1]
matris =
8     12     19
7      3      2
12     4     23
8      1      1

>>com_matris = [matris, matris]
com_matris =
8 12 19 8 12 19
7  3  2  7  3  2
12 4 23 12 4 23
8   1   1   8   1   1

>> com_matris(3,2)
Ans=4

>> com_matris(1:3,2:4)
new_ com_matris =
            19 8
            2 7
```

```
>>a = [1 2 3 4 5 6]
a = 1 2 3 4 5 6

>>b = a .* 2
b = 2 4 6 8 10 12

>>c = a .^ 2
c = 1 4 9 16 25 36

>>d = a + 2
d = 3 4 5 6 7 8

>>e = a - 2
e = -1 0 1 2 3 4
```

Eleman elemana işlemleri skalarlar ile vektörler arasında da kullanabilirsiniz.




```
>>a = [1 2 3]
a = 1 2 3
>>b = [4 ; 5 ; 6]
b = 4
    5
    6
>>a * b
ans = 32
```

"32" sonucunu almak için, MATLAB ilk önce iki vektörün karşılıklı elemanları arasında şu işlemleri yapar:

"1*4 = 4", "2*5=10", ve "3*6=18". Sonra "4+10+18=32".

Matris elemanlarının işaretini inceleme :

```
>> d=[2 -2 3;-5 7 -9;-8 1 19]
     2    -2     3
    -5     7    -9
    -8     1    19
```

```
>> sign(d)
     1    -1     1
    -1     1    -1
    -1     1     1
```

round(2.449)=2 en yakın tam sayıya yuvarlar
ceil(2.449)=3 sayıyı yukarı yuvarlar
floor(2.449)=2 sayıyı aşağıya yuvarlar
fix(2.449)=2 sayıyı sıfıra en yakın tam sayıya yuvarlar

bölümden kalan bulma :

```
>> rem(13,5)
     3
```

Matrise ait sutun değerlerinin toplanması :

```
w =   8   2   4   7
      5   2   6   4
      4   7   2   9
>> sum(w)
  17   11   12   20
```

Matris sutun değerlerinin çarpımı :

```
>> prod(w)
  160   28   48   252
```

Matris sutun değerlerinin ortalama değerini alır

```
>> mean(w)
   5.6667   3.6667   4.0000   6.6667
```

Matlab şartlı deyimler ve döngüsel işlemler

if Şartlı Deyimi

```
if a< 5
        y=y+1;
         t=t+a;
End
```

Switch case Yapıları

```
switch deyim (  sayısal veya karakter  )
      case değer1
                komut ifadeleri        % eğer deyim değeri1 ise yürütülür
      case deger2
                komut ifadeleri      % eğer deyim değeri2 ise yürütülür
                . . .
       otherwise
                komut ifadeleri      % hiçbir case ile denk düşmezse yürütülür
  end
```

Matlab - Döngüler

while Döngüsü

```
while deyim
    Komut ifadeleri
End
```

for Döngüsü

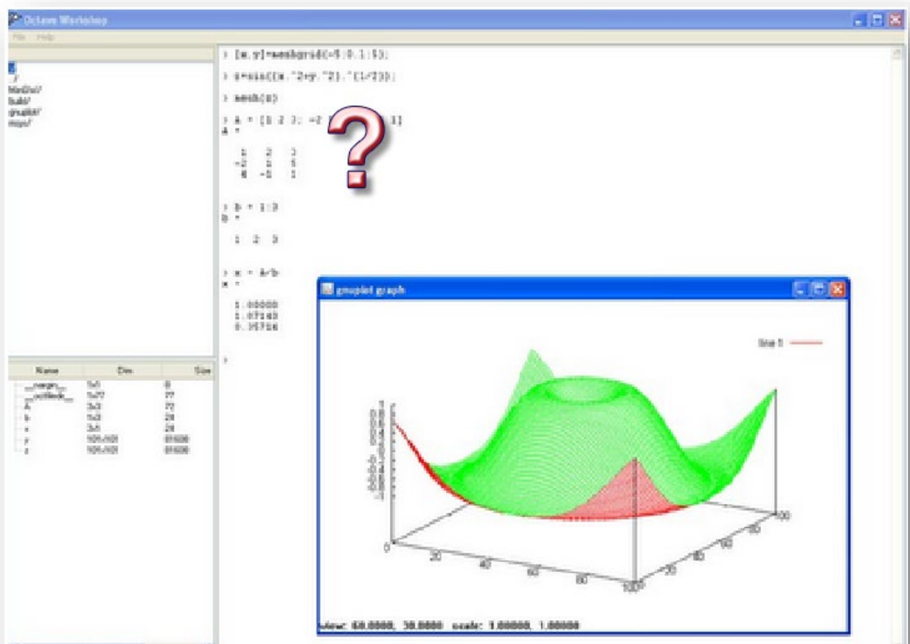
```
for   indeks=başlangıç:artış:son
      Komut ifadeleri
end

for    i= 2:6
      x(i )=2*x(i-1 );
End
```

break fonksiyon ifadesi kullanarak while döngüsünden herhangi bir anda çıkılabilir.
return komutların hali hazır sıralanmasını sona erdirir ve uyarılan fonksiyonu veya klavyeyi denetime geri döndürür.
Continue komutu, for veya while döngüsünde kontrolü bir sonraki yinelemeye geçirir.

Matlab

Ödev :

Örneğini verdiğimiz akış diyagramının matlab komutları ile programını yazınız.



Vektörler

Vektörler tek boyutlu sayı dizileridir. Elemanlarının sıralanma yönlerine göre sütun veya satır vektörü adlarını alırlar. Aşağıdaki A sıra vektörünü Matlab'e tanıtalım.

$A = \{2 \quad 4 \quad 5 \quad 7\}$ `A = [2 4 5 7];` veya `A = [2, 4, 5, 7];`

Şimdi de bir sütun vektörü Matlab'e tanıtalım. Matlab'de yeni bir satıra geçildiğini anlatmak için matris elemanları arasına (;) yerleştirilir.

$B = \begin{Bmatrix} 1 \\ 2 \\ 4 \\ 6 \end{Bmatrix}$ `B = [1; 2; 4; 6];`

A ve B vektörlerinin boyutları oldukça küçük olduğu için bu tanıtım işlemleri değişkenler editörü vasıtasıyla da yapılabilirdi. Ancak vektör boyutları büyüdükçe, manuel olarak tanıtım işlemi oldukça zorlaşmaktadır. Özellikle belli bir artıma sahip vektörlerin oluşturulmasında (:) operatörü kullanılmaktadır. Elemanları -12'den başlayıp 2'şer artarak 150'ye kadar devam C satır vektörünü oluşturalım.

$$C = \{-12 \quad -10 \quad -8 \quad \dots \quad 144 \quad 146 \quad 148 \quad 150\}$$

```
C = [-12:2:150];
```

Başlangıç değeri | Artış miktarı | Son değer

Benzer şekilde, elemanları 1200'den başlayan ve 10'ar inerek -1200'de biten bir D kolon vektörü oluşturalım.

$$D = \begin{Bmatrix} 1200 \\ 1190 \\ \vdots \\ -1190 \\ -1200 \end{Bmatrix}$$

```
D = [1200:-10:-1200]'
```

D kolon vektörünün oluşturulması için önce bir satır vektörü oluşturulmuş ve daha sonra (') operatörü vasıtasıyla transpozesi (devriği) alınmıştır.

Bir vektörün boyutu veya eleman sayısı length veya size komutu ile öğrenilebilir.

Örnek olarak C vektörünün eleman sayısı :

```
>> length(C)
ans = 82      → Sütun Sayısı
```

```
>> size(C)
ans = 1      → Satır Sayısı
```

Vektör indisleri

Bir vektörün elemanlarına atanılan değer değişkenler editörü veya eleman adresi vasıtasıyla değiştirilebilir. Vektör indisleri 1 den başlamaktadır. Satır vektörlerde ilk eleman soldaki eleman, sütun vektörlerde ise en üstteki elemandır. Örnek olarak, A vektörünün 3. elemanını 27 ile değiştirelim.

```
A(3) = 27
```

Benzer şekilde A vektörünün 2. elemanını silelim. Vektörün elemanına [ ] değeri atandığında eleman silinir.

```
 A =  2      4     27      7
>> A(2) = []
 A =  2    27      7
```

Atanacak eleman adresi eleman sayısından fazla ise aradaki elemanlara otomatik olarak 0 değeri atanır. Örnek olarak 3 elemanlı A vektöründe aşağıdaki atama operasyonunu gerçekleştirelim.

```
>> A(9) = 12
 A = 2 27 7 0 0 0 0 0 12
```

Bir vektörün son elemanına end komutu ile ulaşılabilir. `A(end)= 12`

Vektör İşlemleri

Skalerlerle ile 4 işlem vektörün her elemanına uygulanır.

```
A = A+3          A = 5 30 10 3 3 3 3 3 15
```

Benzer şekilde,

```
B = B*2
B =  2
     4
     8
    12
```

Vektörler arasında yapılacak işlemler lineer cebir kurallarını sağlamak durumundadır.

Örnek olarak, iki vektörün birbiriyle çarpılabilmesi için ilk vektörün sütun sayısı ile ikinci vektörün satır sayısı birbirine eşit olmalıdır.

Vektörlerde eleman elemana işlemler (.) operatörü ile gerçekleştirilir.

Eleman elemana operasyonlar

.* Eleman elemana çarpma

./ Eleman elemana bölme

.^ Eleman elemana üst alma

Yeni bir vektörün oluşturulmasında hafızadaki vektörlerden istifade edilebilir.

```
A = [2 4 5 7];

AAA = [A A A]

AAA = 2   4   5   7   2   4   5   7   2   4   5   7
```

Benzer şekilde hafızadaki bir vektörün parçalarından da yeni vektörler oluşturulabilir.

```
A = AAA(1:4)

A = 2   4   5   7
```

Command Window

```
>> zeros(5,3)

ans =

     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0

>> ones(5,3)

ans =

     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

Özel Vektör Yapıları

zeros(1,n) : Tüm elemanları sıfır olan n elemanlı satır vektör.

zeros(n,1) : Tüm elemanları sıfır olan n elemanlı sütun vektör.

ones(1,n) : Tüm elemanları bir olan n elemanlı satır vektör.

ones(n,1) : Tüm elemanları bir olan n elemanlı sütun vektör.

rand(1,n) : Elemanları 0 ile 1 arasından rastgele seçilmiş n elemanlı satır vektör.

rand(n,1) : Elemanları 0 ile 1 arasından rastgele seçilmiş n elemanlı sütun vektör.

randn(1,n) : Ortalaması 0 ve standart sapması 1 olan normal dağılımlı elemanlardan oluşan n elemanlı sütun vektör.

randn(n,1) : Ortalaması 0 ve standart sapması 1 olan normal dağılımlı elemanlardan oluşan n elemanlı sütun vektör.

Command Window

```
>> rand(5,2)*10

ans =

    5.0281    1.9343
    7.0947    6.8222
    4.2889    3.0276
    3.0462    5.4167
    1.8965    1.5087
```

Ortalama $\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$

Standart Sapma $s = \left(\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \right)^{\frac{1}{2}}$

Varyans $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$

Medyan : Büyüklüğe göre sıralanmış bir dizinin (n+1)/2. elemanı.

Veri Analizi Fonksiyonları

mean( )	Ortalama
median( )	Orta değer
std( )	Standart sapma
max( )	Vektörün maksimum değeri
min( )	Vektörün minimum değeri
var( )	Varyans
sum( )	Elemanların toplamı
prod( )	Elemanların çarpımı
sort( )	Elemanları büyüklüğe göre sıralama
flipud( )	Sıralamayı yukarıdan aşağıya değiştirme
fliplr( )	Sıralamayı soldan sağa değiştirme

Matrisler

Matrisler iki boyutlu sayı dizileridir. m satır ve n sütundan oluşan bir A matrisi ele alınırsa a_{ij}, matrisin i. satır ve j. sütununda yer alan elemandır.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \\ a_{m1} & \cdots & & a_{mn} \end{bmatrix}$$

Örnek olarak 4 hafta boyunca toplanan günlük maximum sıcaklıkları içeren HighTemp adlı matrisi ele alalım.

$$\mathbf{HighTemp} = \begin{bmatrix} 25 & 32 & 33 & 37 & 43 & 45 & 41 \\ 42 & 43 & 45 & 46 & 48 & 41 & 39 \\ 39 & 41 & 43 & 47 & 48 & 48 & 47 \\ 50 & 49 & 45 & 48 & 48 & 51 & 53 \end{bmatrix}$$

İkinci haftanın üçüncü günündeki en yüksek sıcaklığı bulmak istersek;

```
HighTemp(2,3) =

            45
```

Matris İşlemleri

Matris işlemlerinin anlatımında aşağıdaki matrisler kullanılacaktır.

$$\mathbf{A}=\begin{bmatrix}1 & 4\\ 2 & 3\end{bmatrix},\quad \mathbf{B}=\begin{bmatrix}1 & 4\\ 2 & 3\end{bmatrix},\quad \mathbf{C}=\begin{bmatrix}0 & 3\\ 1 & 2\end{bmatrix}$$

```
A = [1 4; 2 3];

B = [1 4; 2 3];

C=[0 3; 1 2]
```

Toplama ve Çıkarma: Aynı boyutlardaki matrislerde toplama ve çıkarma işlemi uygulanabilir. Örnek olarak, A ve C matrislerini toplayıp D değişkenine atayalım. ($d_{ij}=a_{ij}+c_{ij}$)

$$\mathbf{D}=\mathbf{A}+\mathbf{C}=\begin{bmatrix}1 & 7\\ 3 & 5\end{bmatrix}$$

```
D = A + C
```

Transpozisyon : Bir matrisin transpozesi veya devriği, satır ve sütunların yer değiştirilmesi ile elde edilir. Örnek olarak, A matrisinin transpozesini alıp D değişkenine atayalım. ($d_{ij}=a_{ji}$)

$$\mathbf{D}=\mathbf{A}'=\begin{bmatrix}1 & 2\\ 4 & 3\end{bmatrix}$$

```
D = A'
```

Skalerler ile 4 işlem : Skalerler ile 4 işlem yapılması durumunda matrisin her elemanı sırayla işleme girer ve aynı boyutda bir değişkene atılır. Yalnızca bölme işleminde dikkat edilmesi gereken bir uygulama vardır. Bir matris bir skalere bölünmesi durumunda elemanların hepsi o skalere bölünmektedir. Ancak bir skalerin bir matrise bölünmesi işlemi hata ile sonuçlanmaktadır. Örnek olarak D matrisini 2 ile çarpalım. ($d_{ij}=2 \cdot a_{ij}$.)

$$\mathbf{D} = 2 \cdot \mathbf{A} = \begin{bmatrix} 2 & 8 \\ 4 & 3 \end{bmatrix}$$

```
D = 2*A
```

İki matrisin çarpımı : İki matrisin çarpılabilmesi için ilk matrisin sütun sayısı ile ikinci matrisin satır sayısının birbirine eşit olması gerekmektedir. Elde edilen matris, ilk matrisin satır sayısı ile ikinci matrisin sütun sayısı boyutlarında olacaktır. Örnek olarak A ve C matrislerini toplayalım ve D değişkenine atayalım.

$$d_{ij} = \sum_{k=1}^{n} a_{ik} \cdot c_{kj}$$

$$\mathbf{D} = \mathbf{A} \cdot \mathbf{C} = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 0 & 3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 4 \cdot 1 & 1 \cdot 3 + 4 \cdot 2 \\ 2 \cdot 0 + 3 \cdot 1 & 2 \cdot 3 + 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} 4 & 11 \\ 3 & 12 \end{bmatrix}$$

```
D = A*C
```

Çarpma işleminde matrislerin çarpım sırası değişince sonuç da değişmektedir.

$$\mathbf{E} = \mathbf{C} \cdot \mathbf{A} = \begin{bmatrix} 6 & 9 \\ 5 & 10 \end{bmatrix}$$

```
E = C*A
```

<u>Determinant</u> : Satır ve sütun sayısı aynı olan bir A matrisinin determinantı aşağıda tanımlanmıştır.

$$|\mathbf{A}| = \sum_{k=1}^{n} a_{ik} C_{ik}, \quad i = 1, \cdots, n$$

Bu ifadede C_{ik}, a_{ik} nın kofaktörüdür.

$$C_{ik} = (-1)^{i+k} M_{ik}$$

M_{ik} ise a_{ik}'nın minörüdür. M_{ik}, A matrisinin i. satır ve k. sütunun silinmesinden sonra elde edilen matrisin determinantıdır.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Yukarıda verilen 3x3 boyutlu kare A matrisi için M_{33}'ü hesaplayalım.

$M_{33} = a_{11}a_{22} - a_{21}a_{12}$

Matris boyutları büyüdükçe hesaplanmaları oldukça yorucu bir hali almaktadır.

Matlab'de matris determinantı det fonksiyonuyla hesaplanmaktadır.

Örnek olarak;

```
det(A) =

        -5
```

<u>Matris İnversi:</u> Matrisin kendisiyle çarpılması sonucu birim matrisi veren matrise, matrisin inversi veya tersi adı verilir. Her matrisin inversi bulunmamaktadır.

Matris inversi aşağıdaki şekilde hesaplanmaktadır.

$$\mathbf{A}^{-1} = \frac{\mathrm{adj}(\mathbf{A})}{|\mathbf{A}|} \qquad \mathrm{adj}(\mathbf{A}) = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & \ddots & & \vdots \\ \vdots & & & \\ C_{n1} & \cdots & & C_{nn} \end{bmatrix}$$

2x2 ve 3x3 boyutlarındaki matrislerin inverslerinin alınması aşağıda gösterilmiştir.

$$\mathbf{A}^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{\mathrm{adj}(\mathbf{A})}{|\mathbf{A}|} = \frac{\begin{bmatrix} a_{11} & -a_{21} \\ -a_{12} & a_{22} \end{bmatrix}}{a_{11}a_{22} - a_{21}a_{12}} \qquad \mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Sayısal Analiz / Uygulama

$$A^{-1} = \frac{\begin{bmatrix} ei-fh & hc-ib & bf-ce \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-db \end{bmatrix}}{|A|} = \frac{\begin{bmatrix} ei-fh & hc-ib & bf-ce \\ gf-di & ai-gc & dc-af \\ dh-ge & gb-ah & ae-db \end{bmatrix}}{aei+bfg+cdh-gec-hfa-icb}$$

Matlab'de matris inversi inv( ) fonksiyonu ile alınmaktadır.

Örnek olarak A matrisinin inversini alalım;

```
Ainv = inv(A);

Ainv =
       -0.6          0.8

        0.4         -0.2
```

Şimdi de sonucun doğruluğunu kontrol edelim.

```
>> A * Ainv

ans =
      1      0

      0      1
```

Sonuç 2x2 boyutlu bir birim matristir. Birim matrisler oldukça sık kullanılan matrislerdir. Matlabde birim matris oluşturmak için eye( ) fonksiyonu kullanılır.

```
eye(2)

ans =
      1      0
```

Matris Rankı : Bir matrisin rankı, dolayısıyla bağımsız satır veya sütun sayısı rank( ) fonksiyonu ile hesaplanabilir.

```
>> rank(A)

ans =

     2
```

Bir A matrisi verilsin. A matrisinin basamak biçime dönüştürülmüşü olan matrisin, sıfırdan farklı satırları sayısına A matrisinin **rankı denir ve r(A) ile gösterilir.**

Özel olarak, herhangi bir sıfır matrisinin rankı 0 kabul edilir.

Tekillik kontrolü : Bir matrisin tekillik durumu cond( ) fonksiyonu ile ölçülebilir.

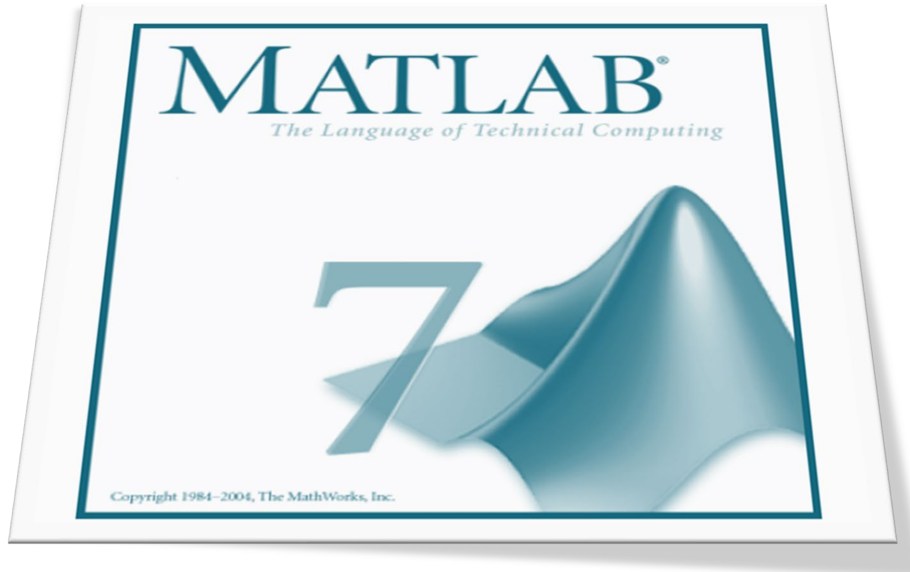
Cond fonksiyonu birim matrise uygulandığında 1 değerini alır. Tekil bir matriste ise sonsuz değerini alır.

```
cond(eye(5))

     = 1

S = [1 1; 1 1+10^-6]

cond(S)

     = Inf
```

Command Window

```
>> eye(3)
ans =
     1     0     0
     0     1     0
     0     0     1
>> cond(eye(3))
ans =
     1
>> rank(eye(3))
ans =
     3
```

Matris Ayrıştırması : Matrislerin ayrıştırması amacıyla lu( ), qr( ) ve svd( ) fonksiyonları kullanılmaktadır. Matris ayrıştırması özellikle büyük boyutlu matrislerle çalışılması durumunda hafıza kullanımında ve hesap zamanında olumlu sonuçlar verebilmektedir.


MATLAB®
The Language of Technical Computing
7
Copyright 1984–2004, The MathWorks, Inc.

Uygulama …