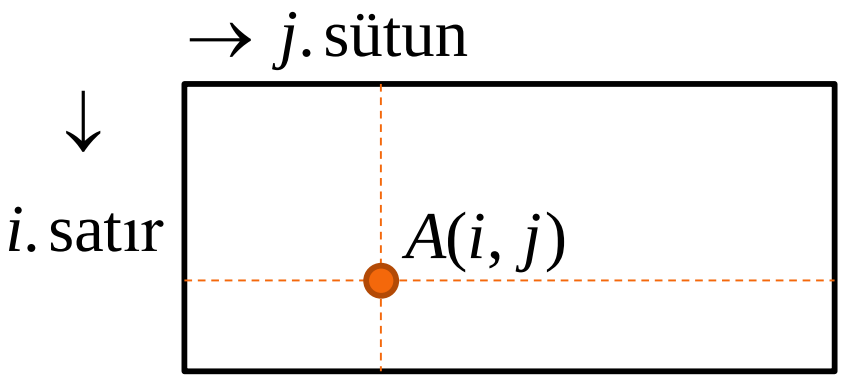
3. Hafta

Matris İşlemleri

Matris tanımlamak
Matris elemanlarına ulaşmak
Kolon operatörü, alt matrisler
Matrisin transpozesi
Girdi değiştirmek/eklemek/silmek
Matris sıralamak
Özel matrisler
Köşegenler, blok-köşegen matrisler
Matrisleri kopyalamak ve şeklini değiştirmek
Eleman bazlı operatörler
Matris fonksiyonları (eleman ve sütun işlemleri)
meshgrid, ndgrid
Örnekler

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



Command Window

```
>> A=[1 2 3; 2 0 4; 0 8 5]

A =

     1     2     3
     2     0     4
     0     8     5

>> size(A)        % [N, M] = size(A), N x M matris

ans =

     3     3

fx >>
```

Matris elemanlarına ulaşım

Command Window

```
>> A(1,1)

ans =

     1
```

% Matrisin 11 elemanı

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

```
>> A(2,3)

ans =

     4
```

% Matrisin 23 elemanı

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

```
>> A(:,2)

ans =

     2
     0
     8
```

% Matrisin 2. sütunu

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

```
>> A(3,:)

ans =

     0     8     5
```

% Matrisin 3. satırı

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

fx

Sütunları sıralamak

```
Command Window
>> A = [1 2 3; 2 0 4; 0 8 5]

A =

     1     2     3
     2     0     4
     0     8     5

>> A(:)        % Sütunları sıralar

ans =

     1
     2
     0
     2
     0
     8
     3
     4
     5
```

A(6)

A(9)

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

Sütun bazlı dizinleme

Satırları sıralamak
B = A'; B(:)

sub2ind, ind2sub
hazır fonksiyonlarına da bakın

Sütun bazlı matrisi oluşturmak

Command Window

```
>> A = zeros(3,3);
>> A(:) = [1 2 0  2 0 8  3 4 5]

A =

     1     2     3
     2     0     4
     0     8     5

fx >>
```

İstenen boyutta tanımla

Elemanları bir satırda (veya sütunda) gir

Elemanlar sütun bazlı yeniden oluşturuldu

Alt matrisler

```
Command Window
>> A = [ 2 4 1 3 5
         8 6 7 4 9
         3 2 5 2 1
         5 6 1 8 4];
>> A(3:4, 2:4)

ans =

     2     5     2
     6     1     8

>> A(1:3, [1,5])

ans =

     2     5
     8     9
     3     1

fx >>
```

$$A = \begin{bmatrix} 2 & 4 & 1 & 3 & 5 \\ 8 & 6 & 7 & 4 & 9 \\ 3 & 2 & 5 & 2 & 1 \\ 5 & 6 & 1 & 8 & 4 \end{bmatrix}$$

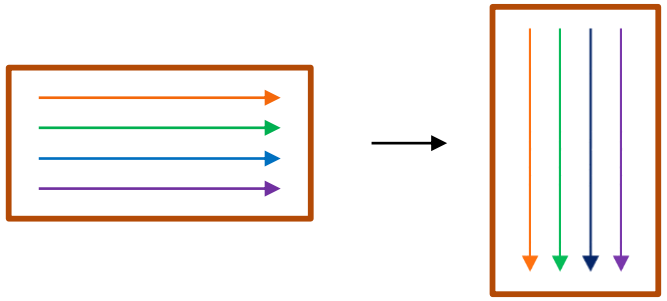
$$A = \begin{bmatrix} 2 & 4 & 1 & 3 & 5 \\ 8 & 6 & 7 & 4 & 9 \\ 3 & 2 & 5 & 2 & 1 \\ 5 & 6 & 1 & 8 & 4 \end{bmatrix}$$

Matrisin transpozesi:

```
Command Window
>> A=[1 2 3 4; 2 0 5 6; 0 8 7 9]

A =

     1     2     3     4
     2     0     5     6
     0     8     7     9

>> A'

ans =

     1     2     0
     2     0     8
     3     5     7
     4     6     9

fx >>
```

% 3 x 4

% 4 x 3



Transpoze operatörü

Satır veya sütun eklemek/silmek

```
Command Window
>> A = [1 2 3; 2 0 4; 0 8 5]

A =

     1     2     3
     2     0     4
     0     8     5

>> A(5,:)=[7 8 9]

A =

     1     2     3
     2     0     4
     0     8     5
     0     0     0
     7     8     9

>> A(:,2)=[]

A =

     1     3
     2     4
     0     5
     0     0
     7     9
```

% 5. satır ekler

4. satır otomatik atandı

% 2. sütunu siler

[ ] 0x0 matrisi belirtir

Alternatif olarak, A'yı 2. sütununu silip tekrar tanımlamak için
>> A = A(:, [1, 3]);

Satır ve sütunları değiştirmek

Command Window

```
>> A = [1 2 3; 2 0 4; 0 8 5]

A =

     1     2     3
     2     0     4
     0     8     5

>> A(:,2) = [20 30 40]'

A =

     1    20     3
     2    30     4
     0    40     5

>> A(3,:) = [50 60 70]

A =

     1    20     3
     2    30     4
    50    60    70
```

% 2. sütunu değiştirir

% 3. satırı değiştirir

Araya satır veya sütun eklemek

Command Window

```
>> A = [1 2 3; 2 0 4; 0 8 5]

A =

     1     2     3
     2     0     4
     0     8     5

>> A = [A(:,1:2), [10 20 30]', A(:,3)]

A =

     1     2    10     3
     2     0    20     4
     0     8    30     5

>> A = [A(1,:); [60 70 80 90]; A(2:3,:)]

A =

     1     2    10     3
    60    70    80    90
     2     0    20     4
     0     8    30     5

>>
```

% 2. ve 3. sütun arasına yeni bir sütun koyuldu

% 1. ve 2. satır arasına yeni bir satır koyuldu

Matrisleri sıralamak

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

C =

     1     2     5     6
     3     4     7     8

>> C = [A; B]

C =

     1     2
     3     4
     5     6
     7     8

fx >>
```

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \quad B=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$$

A ve B'nin aynı satır sayısına sahip olması gerekir

A ve B'nin aynı sütun sayısına sahip olması gerekir

Sütun ve satır eklemek

Command Window

```
>> A = [1 2; 3 4; 5 6];
>> b = [7; 7; 7];
>> c = [8 8 8]';
>> B = [A,b,c]

B =

     1     2     7     8
     3     4     7     8
     5     6     7     8

>> C = [b,A,c]

C =

     7     1     2     8
     7     3     4     8
     7     5     6     8
```

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}, b=\begin{bmatrix}7\\ 7\\ 7\end{bmatrix}, c=\begin{bmatrix}8\\ 8\\ 8\end{bmatrix}$$

```
>> D = [A; [7 7]]

D =

     1     2
     3     4
     5     6
     7     7

>> E = [[8 8]; A]

E =

     8     8
     1     2
     3     4
     5     6
```

Özel matrisler

```
Command Window
>> eye(3)          % 3x3 birim matris

ans =

     1     0     0
     0     1     0
     0     0     1

>> zeros(3)        % 3x3 sıfır matrisi

ans =

     0     0     0
     0     0     0
     0     0     0

>> ones(3)         % 3x3 bir matrisi

ans =

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

>> help eye
>> help zeros
>> help ones

Genel kullanım:
eye(N,M)
zeros(N,M)
ones(N,M)

---

rand(N,M)
randn(N,M)
randi(N,M)
fonksiyonlarına da bakın

Temel matrisler üzerine daha fazla bilgi için:

Command Window

```
>> help elmat
  Elementary matrices and matrix manipulation.

  Elementary matrices.
    zeros         - Zeros array.
    ones          - Ones array.
    eye           - Identity matrix.
    repmat        - Replicate and tile array.
    repelem       - Replicate elements of an array.
    linspace      - Linearly spaced vector.
    logspace      - Logarithmically spaced vector.
    freqspace     - Frequency spacing for frequency response.
    meshgrid      - X and Y arrays for 3-D plots.
    accumarray    - Construct an array with accumulation.
    :             - Regularly spaced vector and index into matrix.

  Basic array information.
    size          - Size of array.
    length        - Length of vector.
    ndims         - Number of dimensions.
    numel         - Number of elements.
    disp          - Display matrix or text.
    isempty       - True for empty array.
    isequal       - True if arrays are numerically equal.
    isequaln      - True if arrays are numerically equal, treating NaNs as equal.

  ...
```

>> help gallery ← **Çeşitli test matrisleri**

Köşegenler

```
Command Window
>> A = [1 2 3; 4 5 6; 7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9

>> d = diag(A)      % esas köşegen

d =

     1
     5
     9

>> d = diag(A,-1)     % ilk alt köşegen

d =

     4
     8
```

>> help diag

diag(diag(A)) = Ne yapar?

Köşegen matris oluşturmak

Command Window

```
>> d = [4 5 6];
>> A = diag(d)

A =

     4     0     0
     0     5     0
     0     0     6

>> d = [4 5];
>> A = diag(d,1)

A =

     0     4     0
     0     0     5
     0     0     0
```

% ya da sütun d = [4 5 6]'
% d esas köşegen

% d ilk üst köşegen

Blok köşegen matris oluşturmak

Command Window

```
>> A = [1 2;3 4]; B = [5 6;7 8];
>> C = [9 8 7; 6 5 4; 3 2 1];
>> blkdiag(A,B)

ans =

     1     2     0     0
     3     4     0     0
     0     0     5     6
     0     0     7     8

>> blkdiag(A,B,C)

ans =

     1     2     0     0     0     0     0
     3     4     0     0     0     0     0
     0     0     5     6     0     0     0
     0     0     7     8     0     0     0
     0     0     0     0     9     8     7
     0     0     0     0     6     5     4
     0     0     0     0     3     2     1
```

% matris boyutları gerektiği gibi genişler

Matrisleri kopyalamak – repmat kullanmak

```
Command Window
>> A = [1 2;3 4]

A =

     1     2
     3     4

>> repmat(A,3,4)

ans =

     1     2     1     2     1     2     1     2
     3     4     3     4     3     4     3     4
     1     2     1     2     1     2     1     2
     3     4     3     4     3     4     3     4
     1     2     1     2     1     2     1     2
     3     4     3     4     3     4     3     4

>> s = repmat('%7.4f     ',1,4)

s =

    '%7.4f     %7.4f     %7.4f     %7.4f     '
```

repmat, diziler (strings) ve hücre dizileri (cell arrays) gibi diğer data tipleri ile de çalışır.

Kopyalanan diziler (strings)

Bir matris veya vektörün şeklini değiştirmek

Command Window

```
>> a = [1 2 3 4 5 6];
>> reshape(a,2,3)

ans =

     1     3     5
     2     4     6

>> reshape(a,3,2)

ans =

     1     4
     2     5
     3     6

>> reshape(a,6,1)

ans =

     1
     2
     3
     4
     5
     6
```

B = reshape(A, P, Q)

bir MxM matrisi bir PxQ matrise çevirir.
(PxQ = NxM olmalıdır)

B A'nın elemanlarından sütun bazlı oluşturulur

Bir matris veya vektörün şeklini değiştirmek

Command Window

```
>> A = [1 5 9
        2 6 5
        3 7 0
        4 8 4];
>> reshape(A,3,4)

ans =

     1     4     7     5
     2     5     8     0
     3     6     9     4

>> reshape(A,2,6)

ans =

     1     3     5     7     9     0
     2     4     6     8     5     4

>> reshape(A,6,2)

ans =

     1     7
     2     8
     3     9
     4     5
     5     0
     6     4

>>
```

Eleman bazlı matris işlemleri

```
Command Window
>> A = [1 2;3 4]

A =

     1     2
     3     4

>> [A, A.^2;2.^A, A^2]

ans =

     1     2     1     4
     3     4     9    16
     2     4     7    10
     8    16    15    22

>> B = 10.^A

B =

          10         100
        1000       10000

>> [B, log10(B)]

ans =

          10         100           1           2
        1000       10000           3           4
```

eleman bazlı işlem

matris bazlı işlem

% alt blok formunda

% A.^2 ~= A^2

Eleman bazlı matris işlemleri

```
Command Window
>> A = [1 4;8 2], B = [1 2; 2 1]

A =

     1     4
     8     2


B =

     1     2
     2     1

>> A./B

ans =

     1     2
     4     2

>> A.\B

ans =

    1.0000    0.5000
    0.2500    0.5000
```

Matris işlemlerine dikkat edelim:

```
>> sym(A/B)      % A*inv(B)

ans =

[  7/3, -2/3]
[ -4/3, 14/3]

>> A\B           % inv(A)*B

ans =

    0.2000         0
    0.2000    0.5000

>>
```

Eleman bazlı matris işlemleri

Command Window

```
>> A = [1 4;8 2], B = [1 2;2 1]

A =

     1     4
     8     2


B =

     1     2
     2     1

>> A.*B

ans =

     1     8
    16     2

>> A.^B

ans =

     1    16
    64     2

>> B.^A

ans =

     1    16
   256     1

fx >>
```

Matris fonksiyonları

```
Command Window
>> X = [pi/2, pi/3; pi/4, pi/8]

X =

    1.5708    1.0472
    0.7854    0.3927

>> sin(X)

ans =

    1.0000    0.8660
    0.7071    0.3827

>> sin(sym(X))

ans =

[          1,              3^(1/2)/2]
[ 2^(1/2)/2, (2 - 2^(1/2))^(1/2)/2]

>>
```

Çoğu fonksiyon matrisler üzerinde eleman bazlı çalışmaktadır, örneğin trig., exp., log. fonksiyonları.

Diğerleri sütun bazlı çalışır, örneğin min, max, sort, diff, mean, std, median, sum, cumsum, prod, cumprod

Matris fonksiyonları

Command Window

```
>> A = [8 5 8
        9 1 3
        2 4 5
        6 2 2];
>> sum(A)

ans =

    25    12    18

>> cumsum(A)

ans =

     8     5     8
    17     6    11
    19    10    16
    25    12    18
```

Sütun bazlı çalışan fonksiyonlar ikinci bir argüman kullanılarak satır bazlı çalışabilir.

```
>> sum(A,2)

ans =

    21
    13
    11
    10

>> cumsum(A,2)

ans =

     8    13    21
     9    10    13
     2     6    11
     6     8    10
```

Matris fonksiyonları

```
Command Window
>> A = [8 5 8
        9 1 3
        2 4 5
        6 2 2];
>> mean(A), mean(A,2)

ans =

    6.2500    3.0000    4.5000


ans =

    7.0000
    4.3333
    3.6667
    3.3333
```

Her sütun için hesaplanır

Satırlar boyunca hesaplanır

```
>> [m,i] = min(A)

m =

     2     1     2


i =

     3     2     4

>> [m,i] = min(A,[],2);
>> [m,i]

ans =

     5     2
     1     2
     2     1
     2     2
```

min, max satır bazlı işlemler için çok farklı bir yazım gerektirir,
diff, std için de benzerdir.

Matris fonksiyonları

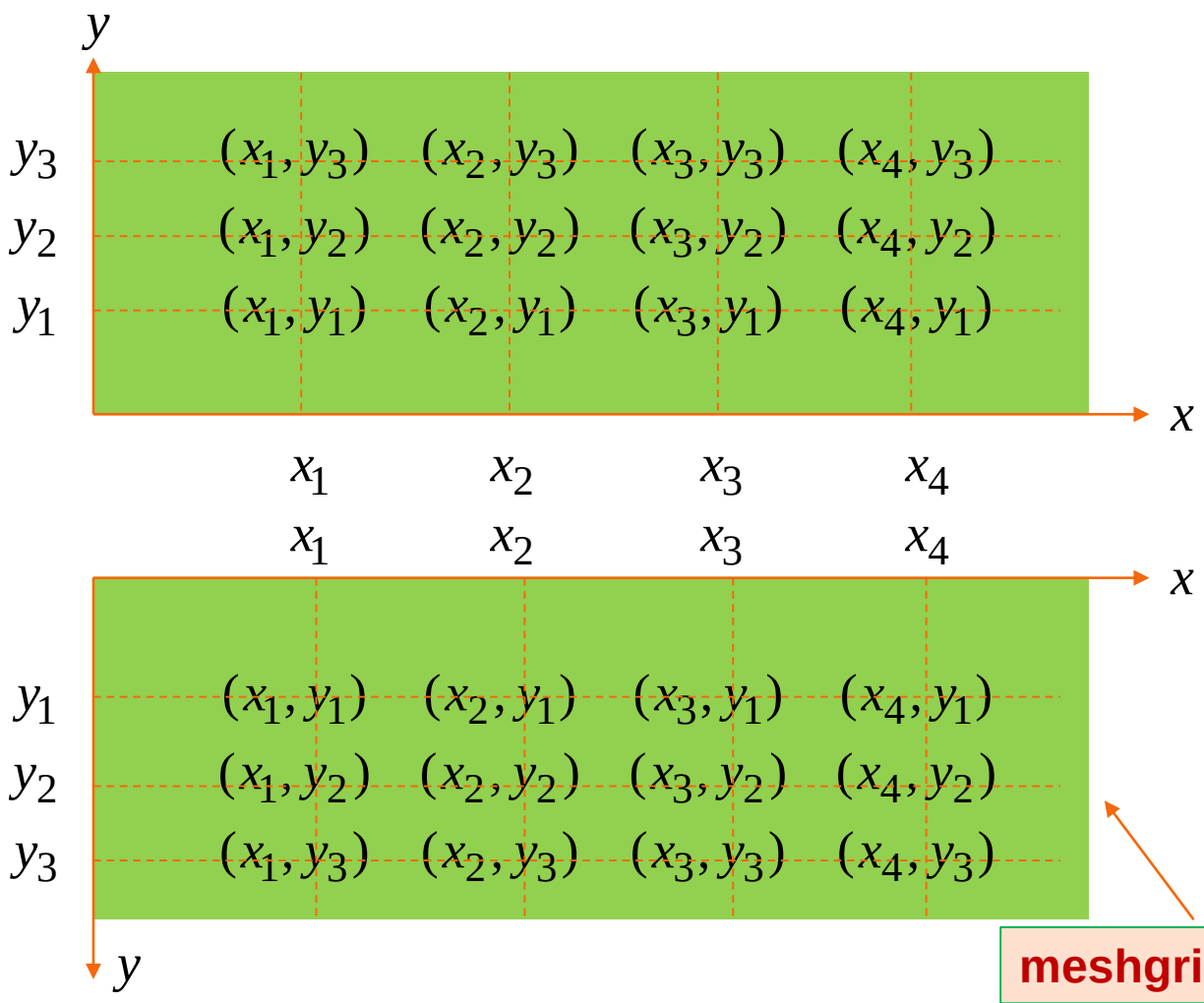
```
Command Window
>> A = [8 5 8
        9 1 3
        2 4 5
        6 2 2];
>> fliplr(A)

ans =

     8     5     8
     3     1     9
     5     4     2
     2     2     6

>> flipud(A)

ans =

     6     2     2
     2     4     5
     9     1     3
     8     5     8
```

```
>> rot90(A)

ans =

     8     3     5     2
     5     1     4     2
     8     9     2     6

fx >>
```

90 derece döndürür
Her satırı ters çevirir
Her sütunu ters çevirir

flipud(rot90(A)) ve **rot90(fliplr(A))**
A' ile aynıdır

meshgrid fonksiyonu



$$x = [x_1, x_2, x_3, x_4]$$
$$y = [y_1, y_2, y_3]$$
$$[X, Y] = \text{meshgrid}(x, y)$$

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_3 & x_4 \end{bmatrix},$$

$$Y = \begin{bmatrix} y_1 & y_1 & y_1 & y_1 \\ y_2 & y_2 & y_2 & y_2 \\ y_3 & y_3 & y_3 & y_3 \end{bmatrix}$$

meshgrid

İki değişkenli bir fonksiyon, z(x,y) bir yüzeyi tanımlar.

meshgrid, ndgrid

Command Window

```
>> x = [1 2 3 4];          % N boyut
>> y = [5 6 7];            % M boyut
>> [X,Y] = meshgrid(x,y)

X =

     1     2     3     4
     1     2     3     4
     1     2     3     4

Y =

     5     5     5     5
     6     6     6     6
     7     7     7     7
```

x satır bazlı kopyalandı

y sütun bazlı kopyalandı

X,Y MxN boyuttadır

Command Window

```
>> [Y,X] = meshgrid(y,x)
```

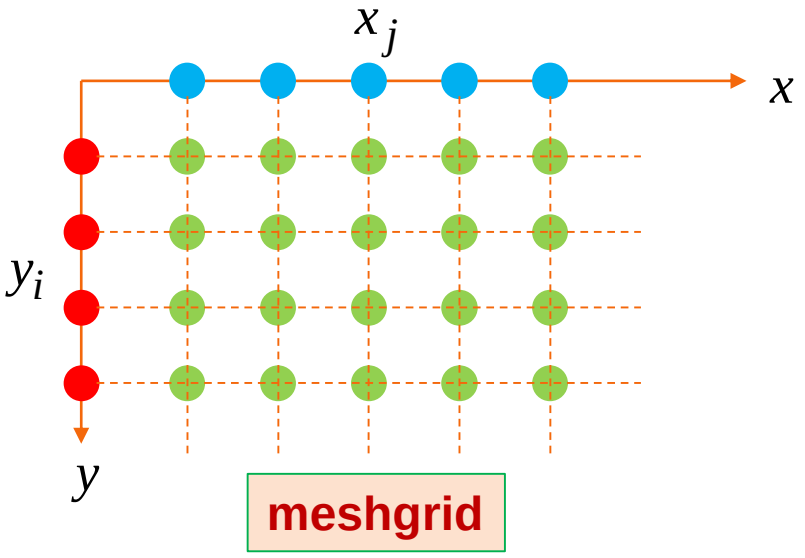
eşdeğer

```
>> [X,Y] = ndgrid(x,y)

X =

     1     1     1
     2     2     2
     3     3     3
     4     4     4

Y =

     5     6     7
     5     6     7
     5     6     7
     5     6     7
```

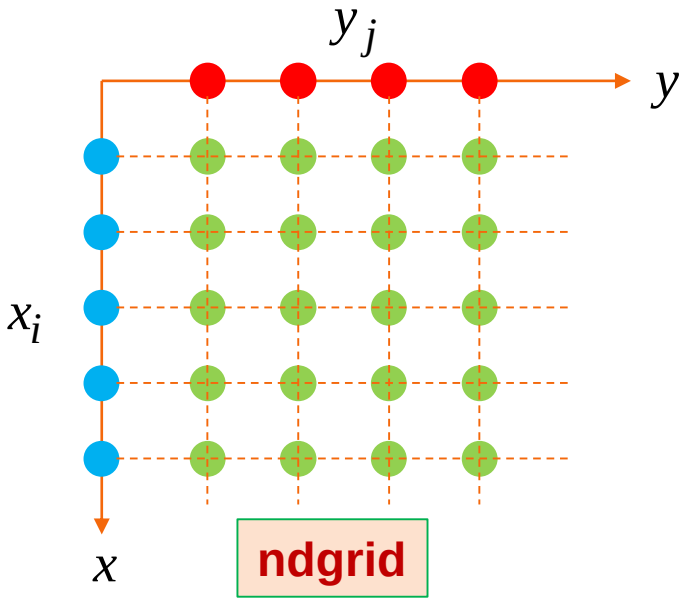
x sütun bazlı kopyalandı

x satır bazlı kopyalandı

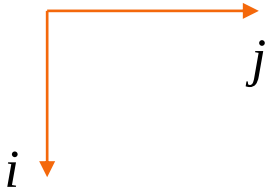
X,Y NxM boyuttadır



meshgrid



ndgrid



$[X, Y] = \text{meshgrid}(x, y)$

$x = \text{satırlar}$

$y = \text{sütunlar}$

$[X, Y] = \text{ndgrid}(x, y)$

$x = \text{sütunlar}$

$y = \text{satırlar}$

eşdeğer

$[Y, X] = \text{meshgrid}(y, x)$

$[Y, X] = \text{ndgrid}(y, x)$

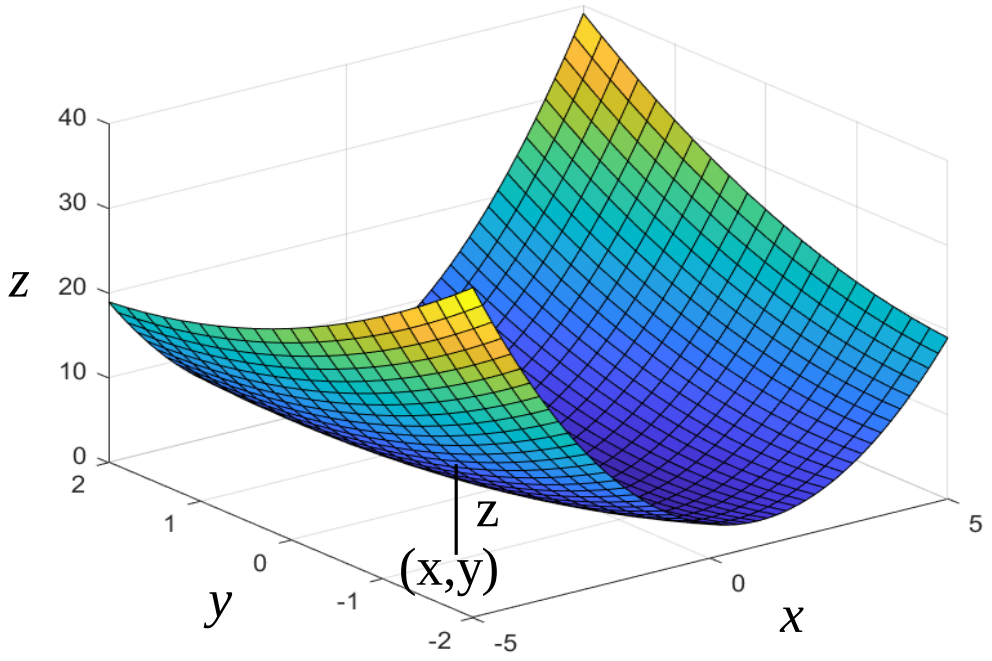
Örnek 1: Yüzey grafiği

```
x = linspace(-5,5,51);
y = linspace(-2,2,21);

[X,Y] = meshgrid(x,y);

Z = X.*Y + Y.^2 + X.^2;
figure; surf(X,Y,Z);
```

Eleman bazlı işlemler anlamlıdır, çünkü X,Y aynı boyuttadır.

meshgrid iki değişkenli fonksiyonların (z(x,y)) grafikleri olan yüzey grafikleri için kullanılan tipik bir fonksiyondur, örneğin

$$z = f(x,y) = x*y + x\wedge 2 + y\wedge 2$$

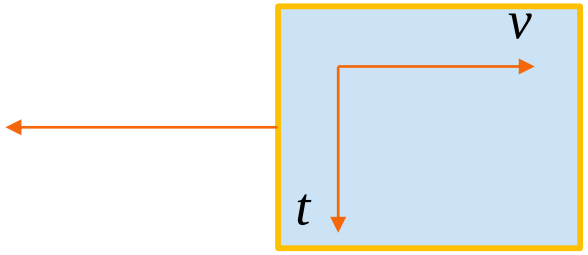


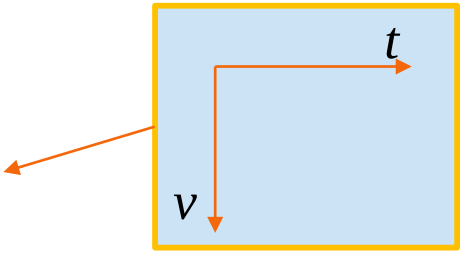
Örnek 2: t zamanına bağlı bir vektörde farklı v değerleri için x=vt 'nin hesabı

```
t = [0 1 2 3 4];
v = [1 -2 3];

[T,V] = meshgrid(t,v);
X1 = V.*T

[V,T] = meshgrid(v,t);
X2 = V.*T

[T,V] = ndgrid(t,v);
X3 = V.*T

[V,T] = ndgrid(v,t);
X4 = V.*T
```

Command Window

```
X1 =

     0     1     2     3     4
     0    -2    -4    -6    -8
     0     3     6     9    12


X2 =

     0     0     0
     1    -2     3
     2    -4     6
     3    -6     9
     4    -8    12


X3 =

     0     0     0
     1    -2     3
     2    -4     6
     3    -6     9
     4    -8    12

X4 =

     0     1     2     3     4
     0    -2    -4    -6    -8
     0     3     6     9    12
```





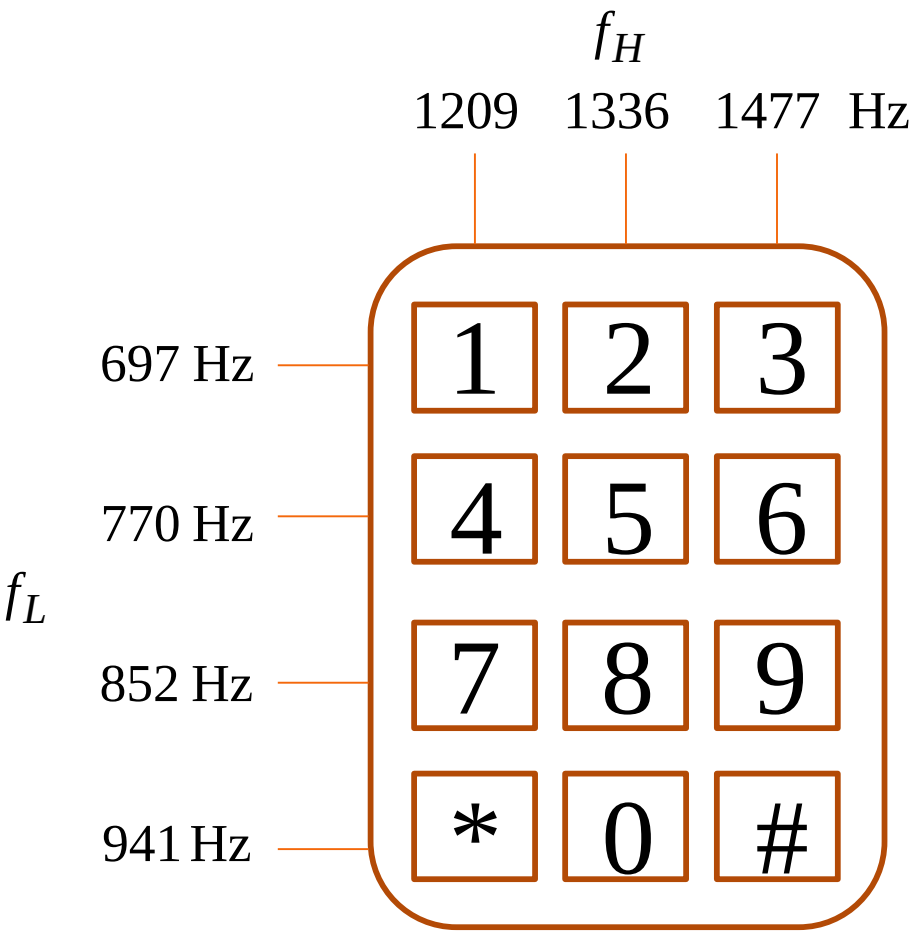
```
t = [0 1 2 3 4];
v = [1 -2 3];

[T,V] = meshgrid(t,v);
X1 = V.*T

fprintf('        t = %4.1f   %4.1f   %4.1f  %4.1f   %4.1f\n',t);
fprintf('-----------------------------------------------\n');
fprintf('v = %2.0f  |  %4.1f  %4.1f  %4.1f  %4.1f %4.1f\n', ...
          [v;X1']);
```

fprintf ile tablo yap

```
>> [v;X1']

ans =

     1    -2     3
     0     0     0
     1    -2     3
     2    -4     6
     3    -6     9
     4    -8    12

fx >>
```

```
         t =     0.0     1.0     2.0     3.0     4.0
-----------------------------------------------
v =  1    |     0.0     1.0     2.0     3.0     4.0
v = -2    |    -0.0    -2.0    -4.0    -6.0    -8.0
v =  3    |     0.0     3.0     6.0     9.0    12.0
fx >>
```

Örnek 3: Çift tonlu çoklu frekans (Dual-Tone-Multi-Frequency (DTMF)) tuş takımı diğer adıyla, tuşlu telefon (touch-tone phone)



$$y(t) = \sin(2\pi f_L t) + \sin(2\pi f_H t)$$

Her tuş bir düşük ve yüksek frekans çiftiyle (f_L, f_H) eşlenir.

f_L'yi sütun boyunca f_H'yi de satır boyunca kopyalamak için meshgrid kullanabiliriz.

```
K = ['1'    '2'    '3'
     '4'    '5'    '6'
     '7'    '8'    '9'
     '*'    '0'    '#'];

% eşdeğerdir,
% K = ['123'; '456'; '789'; '*0#'];

fL = [697, 770, 852, 941];
fH = [1209, 1336, 1477];

[FH, FL] = meshgrid(fH,fL);    % [FL, FH] = ndgrid(fL, fH)
```

Tuş takımı matrisi

$$F_L = \begin{bmatrix} 697 & 697 & 697 \\ 770 & 770 & 770 \\ 852 & 852 & 852 \\ 941 & 941 & 941 \end{bmatrix}, \quad F_H = \begin{bmatrix} 1209 & 1336 & 1477 \\ 1209 & 1336 & 1477 \\ 1209 & 1336 & 1477 \\ 1209 & 1336 & 1477 \end{bmatrix}$$

```
s = input('aranacak numarayı girin:   ', 's');

fs = 8192;              % ses kartı örnekleme oranı
T = 1/fs;               % örnekleme zaman aralığı
Td = 1/4;               % her bir tuşun süresi, sn
Tb = 1/10;              % boşluk zamanı, sn
Nb = round(Tb*fs);      % boşluk süresi örnekleri
yb = zeros(1,Nb);       % boşluk sinyali örnekleri

t = 0:T:Td;             % her tuş için zaman vektörü

y = [];                 % genel sinyali başlat
```

Aranacak bir telefon numarası girin, örneğin, s = 2125551212, ses kartı için örnekleme oranını ayarlayın, her tuş için çalma süresini ve her tuş arasındaki boşluk zamanını ayarlayın.

```
for k = 1:length(s)

    [i,j] = find(K==s(k));

    x = sin(2*pi*FL(i,j)*t)+sin(2*pi*FH(i,j)*t);

    y = [y,x,yb];
end

sound(y,fs);
```

k. tuşun yerini s(k)'yi tuş takımı matrisi ile bulur

Genel sinyali ekler

Genel sinyali çalar

Not: y sinyaline kodlanmış s telefon numarası y 'nin son işlemden geçirilmesiyle alıcıya alınabilir.
Bunu yapmanın bir metodu ayrık zamanlı Fourier dönüşümüdür ve keypad.m programında yer almaktadır. Bu derste bunu ele almayacağız.

```
% y = [];
% for k = 1:length(s)
%     q = find(K==s(k));
%     x = sin(2*pi*FL(q)*t)+sin(2*pi*FH(q)*t);
%     y = [y,x,yb];
% end
% sound(y,fs);

% y = [];
% for k = 1:length(s)
%     [i,j] = find(K==s(k));
%     x = sin(2*pi*fL(i)*t)+sin(2*pi*fH(j)*t);
%     y = [y,x,yb];
% end
% sound(y,fs);
```

meshgrid kullanılmayan eşdeğer metodlar

Karşılaştırma metodunu anlamak ve find fonksiyonunu kullanmak

Command Window

```
>> K = ['1'    '2'    '3'
        '4'    '5'    '6'
        '7'    '8'    '9'
        '*'    '0'    '#'];
>> K=='8'

ans =

  4×3 logical array

    0    0    0
    0    0    0
    0    1    0
    0    0    0

>> [i,j] = find(K=='8')

i =

     3


j =

     2

>> q = find(K=='8')

q =

     7
```

K'nin tüm elemanlarını '8' ile karşılaştırır

K'nin doğru elemanının yerine bulur.

'8' 'in yerinin i,j matris indisleri

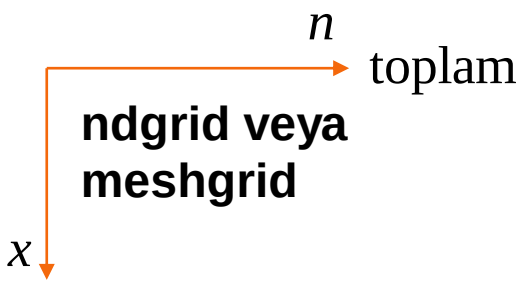
K'de '8' 'in sütun bazlı indeksi

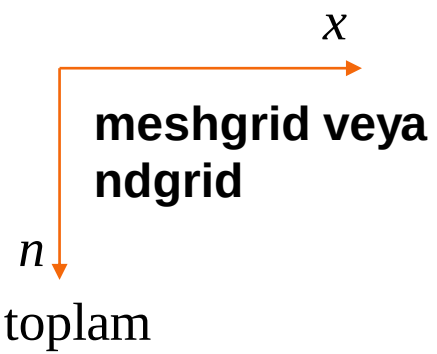
Örnek 4: Vektörleştirilmiş Taylor serisi hesaplamaları

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{N \to \infty} \sum_{n=0}^{N} \frac{x^n}{n!}$$

$$S(x, N) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

(x,n) değişkenlerinde bir matris olarak görülebilir ve n boyut boyunca toplanabilir





x sütun vektör, satır vektör veya bir matris olabilir, ama ndgrid veya meshgrid fonksiyonları içinde x(:) olarak kullanılır.

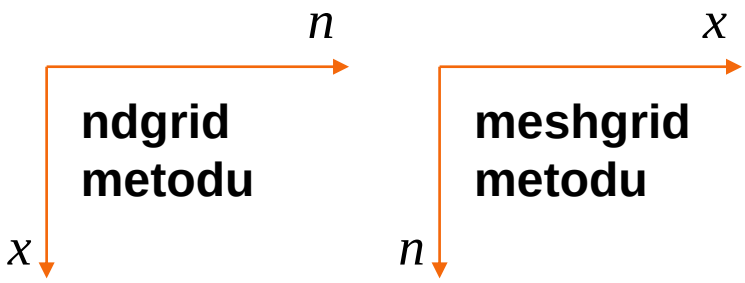
```
[X,n] = ndgrid(x,0:N);
S = sum(X.^n ./ factorial(n), 2);
```

ndgrid metodu:
x = sütunlar
n = satırlar
satır bazlı toplama

```
[X,n] = meshgrid(x,0:N);
S = sum(X.^n ./ factorial(n));
```

meshgrid metodu:
x = satırlar
n = sütunlar
sütun bazlı toplama

```
S = reshape(S, size(X));
```



x'in şekliyle uyan sütun, satır veya matris formuyla S'nin formunu değiştirir

kısmi vektörleştirmenin olduğu veya vektörleştirmenin olmadığı alternatif metodlar

```
S = zeros(size(X));
n = 0:N;
```

n vektörleştirildi, ama x değil

```
for i = 1:length(x)
    S(i) = sum(x(i).^n ./ factorial(n));
end
```

x vektörleştirildi, ama n değil

```
for n = 0:N
    S = S + x.^n / factorial(n);
end
```

vektörleştirme yok

```
for i = 1:length(x)
    for n = 0:N
      S(i) = S(i) + x(i)^n / factorial(n);
    end
end
```

```
x = [1 3 0 -4 10]';  N = 30;          % x sütun vektör

[X,n] = meshgrid(x, 0:N);
S = sum(X.^n ./ factorial(n));
S = reshape(S,size(x));

fprintf('\n       x          exp(x)          S(x,N)\n');
fprintf('    ------------------------------------\n');
fprintf('%7.2f       %12.6f       %12.6f\n',
[x,exp(x),S]');
```

```
     x              exp(x)              S(x,N)
  ------------------------------------------
    1.00            2.718282            2.718282
    3.00           20.085537           20.085537
    0.00            1.000000            1.000000
   -4.00            0.018316            0.018316
   10.00        22026.465795        22026.464036
fx >>
```

daha büyük N gerekir

Daha sonra istenen bir yakınsaklık derecesine ulaşmak için N'yi nasıl seçmemiz gerektiğini tartışacağız

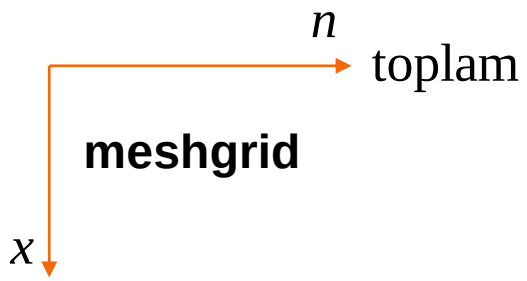
Örnek 5: Polinom değerlendirmesi

$$P(x) = c_0 x^M + c_1 x^{M-1} + \ldots + c_{M-1} x + c_M$$

$$= \sum_{n=0}^{M} c_n x^{M-n}$$

(x,n) değişkenlerinde bir matris olarak görülebilir ve n boyut boyunca toplanabilir

$$\mathbf{c} = [c_0, c_1, \ldots, c_M]$$




```
M = length(c)-1;
[n,X] = meshgrid(0:M, x);
C = meshgrid(c,x);

P = sum(C.*X.^(M-n), 2);

P = reshape(P, size(x));
```

Neden [X,n] yerine [n,X] kullandık?

$$P(x) = x^3 - 3x^2 + 4x + 2$$

```
x = [1, 2, 3]; c = [1, -3, 4, 2];

M = length(c)-1;
[n,X] = meshgrid(0:M, x);
C = meshgrid(c,x);

P = sum(C.*X.^(M-n), 2);

P = reshape(P, size(x))
```

```
P =

     4     6    14
```

```
polyval(c,x)
```

```
ans =

     4     6    14
```

polyval polinom değerlendirmesi için standart hazır bir fonksiyondur

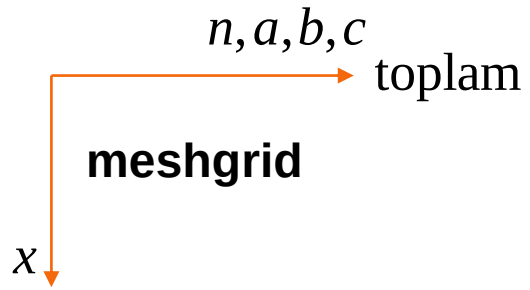
Örnek 6: Tepe değeri

polinom metodu diğer parametrik eğrilere kolayca genelleştirilebilir

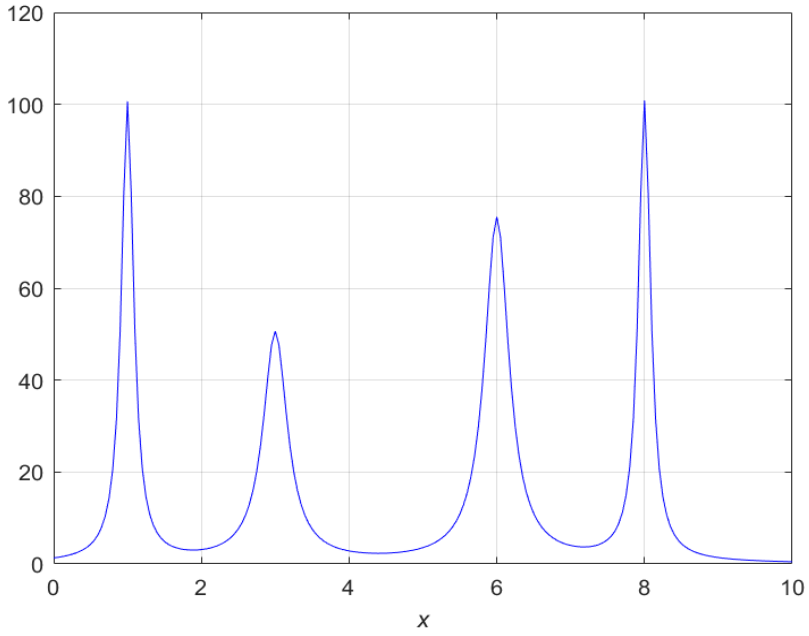
$$S(x)=\sum_{n=1}^{M}\frac{c_n}{(x-a_n)^2+b_n^2}=\frac{c_1}{(x-a_1)^2+b_1^2}+\ldots+\frac{c_M}{(x-a_M)^2+b_M^2}$$

Lorentz eğrileri olarak bilinir, kimyasal spektral tepe değerlerinin modellenmesinde kullanılır

```
x = linspace(0,10,201);
a = [1 3 6 8];
b = [0.1, 0.2, 0.2, 0.1];
c = [1 2 3 1];

[A,X] = meshgrid(a,x);
B = meshgrid(b,x);
C = meshgrid(c,x);

S = sum(C./((X-A).^2 + B.^2), 2);
S = reshape(S,size(x));
```



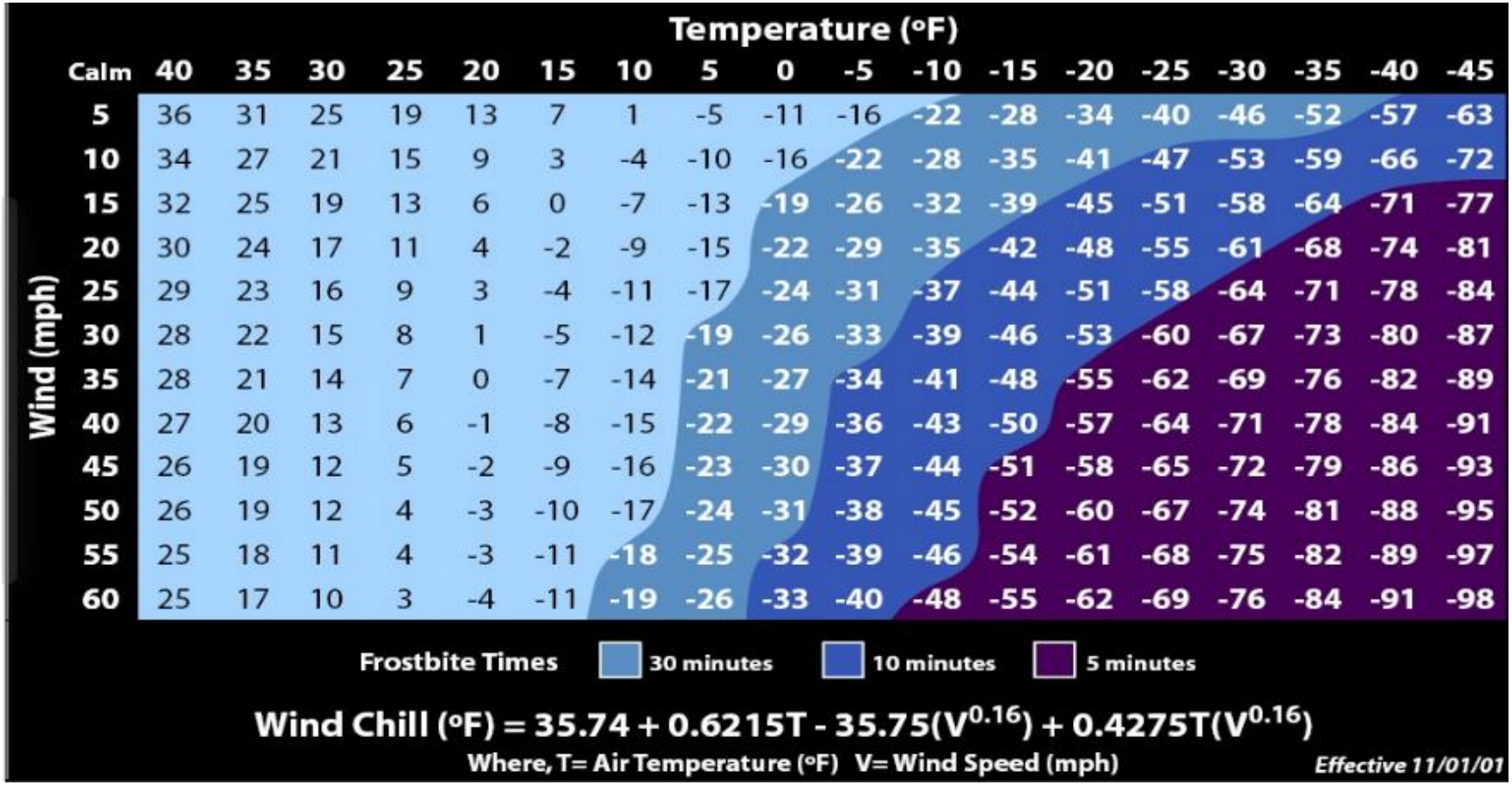
```
figure; plot(x,S,'b-');

xaxis(0,10,0:10);
yaxis(0,120,0:20:120);
xlabel('\it{x}'); grid;
```




```
S = zeros(size(x));

for n = 1:length(c)
    S = S + c(n)./((x-a).^2 + b(n)^2);
end
```

meshgrid kullanılmadığı genel yöntem

Örnek 6: Rüzgar - soğukluk sıcaklık indeksi



Temperature (°F)

Wind (mph)	Calm	40	35	30	25	20	15	10	5	0	-5	-10	-15	-20	-25	-30	-35	-40	-45
	5	36	31	25	19	13	7	1	-5	-11	-16	-22	-28	-34	-40	-46	-52	-57	-63
	10	34	27	21	15	9	3	-4	-10	-16	-22	-28	-35	-41	-47	-53	-59	-66	-72
	15	32	25	19	13	6	0	-7	-13	-19	-26	-32	-39	-45	-51	-58	-64	-71	-77
	20	30	24	17	11	4	-2	-9	-15	-22	-29	-35	-42	-48	-55	-61	-68	-74	-81
	25	29	23	16	9	3	-4	-11	-17	-24	-31	-37	-44	-51	-58	-64	-71	-78	-84
	30	28	22	15	8	1	-5	-12	-19	-26	-33	-39	-46	-53	-60	-67	-73	-80	-87
	35	28	21	14	7	0	-7	-14	-21	-27	-34	-41	-48	-55	-62	-69	-76	-82	-89
	40	27	20	13	6	-1	-8	-15	-22	-29	-36	-43	-50	-57	-64	-71	-78	-84	-91
	45	26	19	12	5	-2	-9	-16	-23	-30	-37	-44	-51	-58	-65	-72	-79	-86	-93
	50	26	19	12	4	-3	-10	-17	-24	-31	-38	-45	-52	-60	-67	-74	-81	-88	-95
	55	25	18	11	4	-3	-11	-18	-25	-32	-39	-46	-54	-61	-68	-75	-82	-89	-97
	60	25	17	10	3	-4	-11	-19	-26	-33	-40	-48	-55	-62	-69	-76	-84	-91	-98

Frostbite Times — 30 minutes — 10 minutes — 5 minutes

$$\text{Wind Chill (°F)} = 35.74 + 0.6215T - 35.75(V^{0.16}) + 0.4275T(V^{0.16})$$

Where, T= Air Temperature (°F) V= Wind Speed (mph)

Effective 11/01/01

Kaynak: National Weather Service

$$T_{rs} = 35.74 + 0.6215T - 35.75V^{0.16} + 0.4275TV^{0.16}$$

```
f = @(T,V) 35.74 + 0.6215*T - 35.75*V.^0.16 + 0.4275*T.*V.^0.16;
t = 40:-5:-45;    % 1x18
v = 5:5:60;       % 1x12

[T,V] = meshgrid(t,v);

Trs = f(T,V);     % 12x18

fprintf('          Fiziksel sıcaklık (F)\n');

fmt = ['      |   ', repmat('%3.0f    ',1,18), '\n'];
fprintf(fmt,t);

fprintf(['    V     ', repmat('-', 1, 90), '\n']);

fmt = ['%3.0f    |', repmat('%3.0f    ',1,18), '\n'];
fprintf(fmt, [v', Trs]');     % [v', Trs]' 19x12
```

Adsız fonksiyon

meshgrid

t

v

```
        Fiziksel sıcaklık (F)
     | 40    35    30    25    20    15    10     5     0    -5   -10   -15   -20   -25   -30   -35   -40   -45
 V   ----------------------------------------------------------------------------------------------------------
 5   | 36    31    25    19    13     7     1    -5   -11   -16   -22   -28   -34   -40   -46   -52   -57   -63
10   | 34    27    21    15     9     3    -4   -10   -16   -22   -28   -35   -41   -47   -53   -59   -66   -72
15   | 32    25    19    13     6    -0    -7   -13   -19   -26   -32   -39   -45   -51   -58   -64   -71   -77
20   | 30    24    17    11     4    -2    -9   -15   -22   -29   -35   -42   -48   -55   -61   -68   -74   -81
25   | 29    23    16     9     3    -4   -11   -17   -24   -31   -37   -44   -51   -58   -64   -71   -78   -84
30   | 28    22    15     8     1    -5   -12   -19   -26   -33   -39   -46   -53   -60   -67   -73   -80   -87
35   | 28    21    14     7     0    -7   -14   -21   -27   -34   -41   -48   -55   -62   -69   -76   -82   -89
40   | 27    20    13     6    -1    -8   -15   -22   -29   -36   -43   -50   -57   -64   -71   -78   -84   -91
45   | 26    19    12     5    -2    -9   -16   -23   -30   -37   -44   -51   -58   -65   -72   -79   -86   -93
50   | 26    19    12     4    -3   -10   -17   -24   -31   -38   -45   -52   -60   -67   -74   -81   -88   -95
55   | 25    18    11     4    -3   -11   -18   -25   -32   -39   -46   -54   -61   -68   -75   -82   -89   -97
60   | 25    17    10     3    -4   -11   -19   -26   -33   -40   -48   -55   -62   -69   -76   -84   -91   -98
```

Dizi yapısını anlamak adına:

```
fmt = ['%3.0f    |   ', repmat('%3.0f    ',1,18), '\n'];

s1 = '%3.0f    |   ';
s2 = repmat('%3.0f    ',1,18);
s3 = '\n';
fmt = [s1,s2,s3];      % üç diziyi sıralar
```

$\sin(x), 0 \le x \le \pi$ **eğrisinin altında kalan alanın Monte Carlo hesabı**
gerçek alan: A = 2

```
N = 10000; rng(101);
x = pi*rand(1,N);
y = rand(1,N);

i = find(y < sin(x));
j = find(y > sin(x));

scatter(x(i),y(i),1,'r');
hold on;
scatter(x(j),y(j),1,'b');

x = linspace(0,pi,100);
y = sin(x);
plot(x,y,'r-');

A = length(i)/N*pi
```

```
A =

    1.9915
```