



Bilgisayar Destekli Matematiksel Hesaplamalar

MTM3641

Arş. Gör. Dr. Fatih Aylıkçı

<https://avesis.yildiz.edu.tr/faylikci/>



Fatih Aylıkçı

İletişim:

Ofis: KMB-A237

Email: faylikci@yildiz.edu.tr

<https://avesis.yildiz.edu.tr/faylikci/>

Ders Tanımı

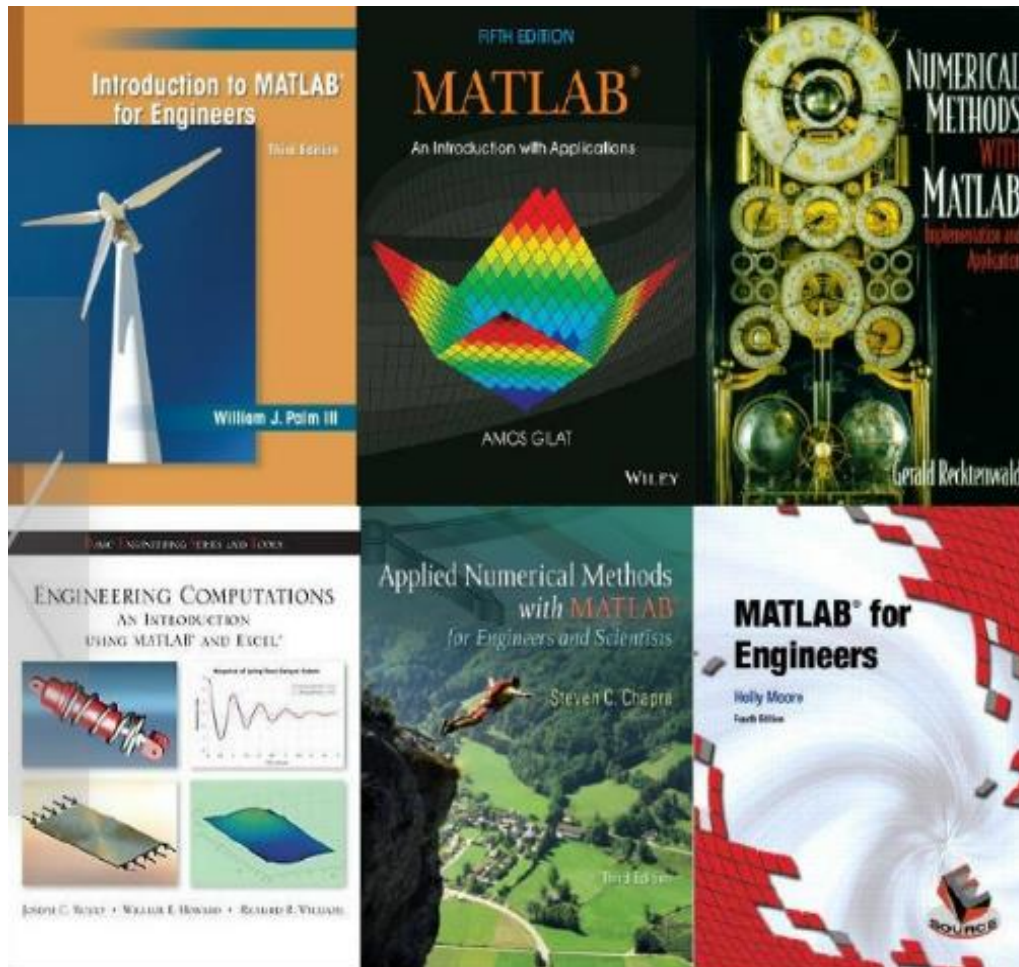
Hedefler¹

Sembolik hesap için kullanılabilecek paket programları tanıma

Paket programları kullanarak sembolik ve sayısal hesaplamaları etkin bir şekilde yapmayı öğrenme

¹<http://www.bologna.yildiz.edu.tr/index.php?r=course/view&id=1680&aid=24>

Ders Materyalleri



Course Material Credits

- Instructor Materials for the books listed,
- Sophocles J. Orfanidis, Rutgers University
- Andrew Pershing, Cornell University
- Kai Borre, Alborg University
- Alba M. Franco-Pereira, UC3M
- Selis Onel, Hacettepe University
- Network Research Lab., Korea University
- Zhiliang Xu, University of Notre Dame
- Computer-Aided Teaching of All Mathematics, University of Cambridge
- Lani Clough and Mark Reed, The University of North Carolina at Chapel Hill
- David Koslicki, Oregon State University
- Monica Tremont Hsu, Intuit

Bu ders, mühendisler ve bilim insanları için güçlü bir programlama dili ve geliştirme ortamı olan MATLAB'a bir giriş niteliğindedir.

MATLAB = Matrix Laboratory (Cleve Moler tarafından geliştirildi)

MATLAB The Mathworks Inc. şirketinin tescilli ticari markasıdır,
<http://www.mathworks.com>

MATLAB'ın temel özellikleri

- Hızlı geliştirme için interaktif bir arayüz ile yüksek düzeyli bir dilde kolay ve verimli programlama
- Etkili bir programlama için vektörleştirilmiş hesaplamalar ve otomatik bellek paylaşımı
- Modern sayısal hesaplama yöntemleri için hazır destek
- Karmaşık sayılar da dahil olmak üzere çeşitli modern veri yapıları ve veri türlerine sahiptir
- Yüksek kaliteli grafikler ve görselleştirme

- Cebirsel ve matematik işlemler sembolik matematik araç kutusu ve diferansiyel denklemlerin çözümleri
- SIMULINK ile simülasyon yeteneği
- Platformlar arası taşınabilir program dosyaları
- Uygulamalar ve simülasyonlar için çok sayıda eklenti araç kutusu
- Çok sayıda mevcut öğretici ve demo dahil, kullanıcı tarafından sağlanan büyük dosya ve araç kutusu veritabanı
- C/C++ gibi diğer dilleri temel alan uzantılara izin verir, Java ve nesne yönelimli programlamayı destekler.

MATLAB Toolbox uygulama alanları

Parallel Computing (2)

Math, Statistics and Optimization (8)

Control System Design and Analysis (6)

Signal Processing and Communications (7)

Image Processing and Computer Vision (4)

Test and Measurement, Data Acquisition (5)

Computational Finance, Datafeeds (7)

Computational Biology (2)

Code Generation and Application Deployment (11)

Database Connectivity

(54 toolbox + 35 simulink ürünü)

SIMULINK Uygulamaları

Fixed-point and Event-Based Modeling

Physical Modeling (mechanics, driveline, hydraulics, RF, electronics, power systems, biology)

Control Systems (design, optimization, aerospace)

Signal and Image Processing and Computer Vision

Communication Systems (digital, analog, wireless)

Code Generation (for embedded systems, DSP chips and FPGAs)

ve daha fazlası

MATLAB KURULUMU



- ✎ <https://www.mathworks.com/>
- ✎ Öncelikle yukarıdaki linkten siteye yıldız e-mail'iniz ve kendiniz belirleyeceğiniz bir şifreyle kayıt olmanız gerekmektedir.
- ✎ Sonra <http://distro.cc.yildiz.edu.tr/> linkinden Diğer Önemli Araçlar klasöründen MATLAB'ın kurulumunu başlatmanız gerekmektedir.
- ✎ Kurulum yaparken sizden bir aktivasyon seçeneği seçmeniz istenecektir. 'İnternet yardımıyla' seçeneği ile birlikte siteye üye olduğunuz yıldız e-mail'iniz ve şifrenizi girmeniz gerekmektedir. Kurulum yaklaşık yarım saat sürmektedir.

MATLAB Temelleri

MATLAB Desktop

MATLAB Editor

Yardım

Değişkenler, hazır sabitler, anahtar kelimeler

Sayılar, formatlar

Diziler (arrays) ve matrisler

İşlemler ve açıklamalar

Fonksiyonlar – hazır ve kullanıcı tanımlı

Temel grafik çizme

Fonksiyon maksimumu ve minimumu

Diziler (strings), hücre dizileri (cell arrays), fprintf

ilk hafta

ikinci hafta

1. Hafta

Giriş

Değişkenler, diziler, matrisler, grafikler



1. MATLAB Masaüstü

HELP

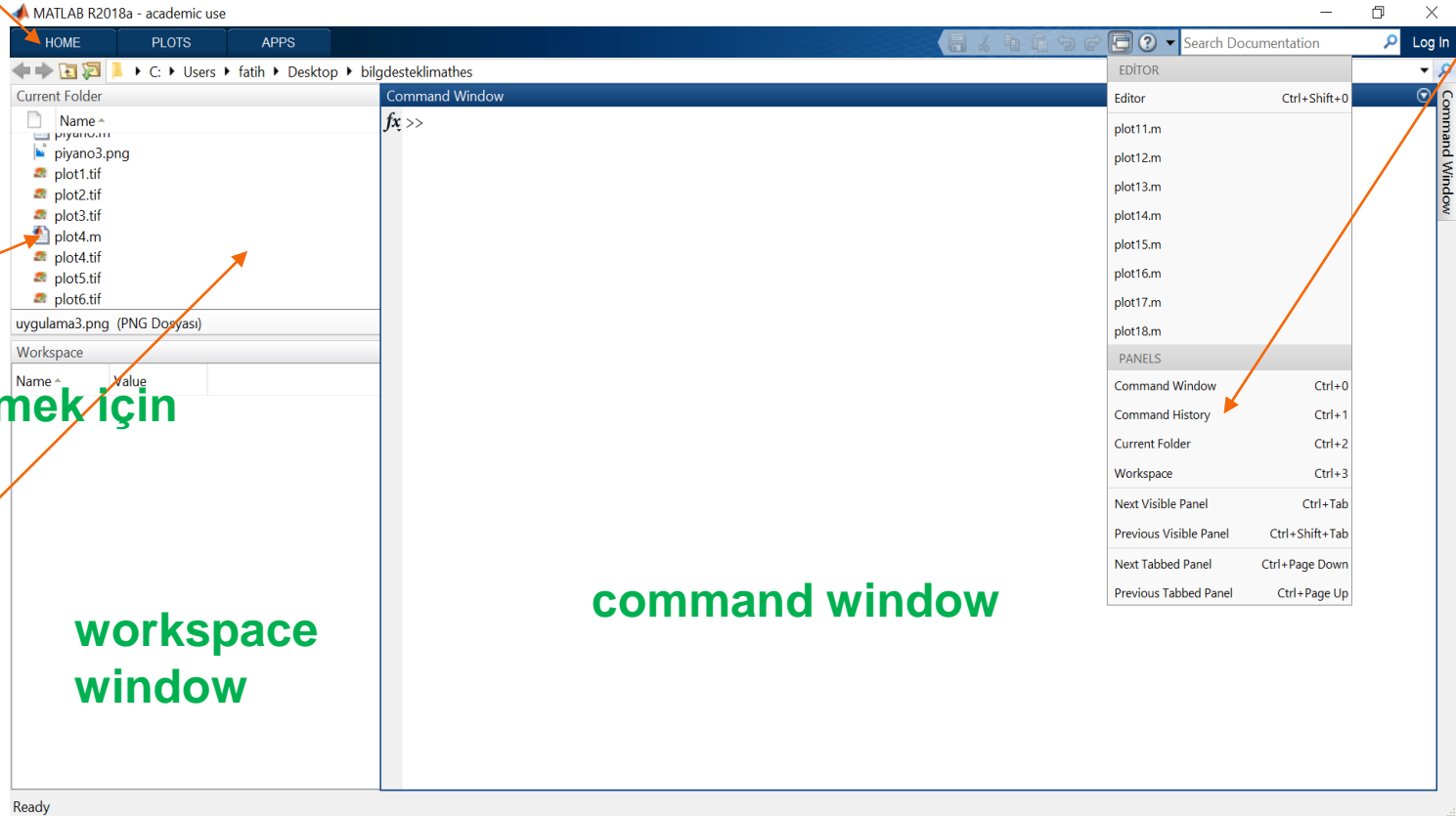
command history

M file
düzenlemek için
çift tıkla

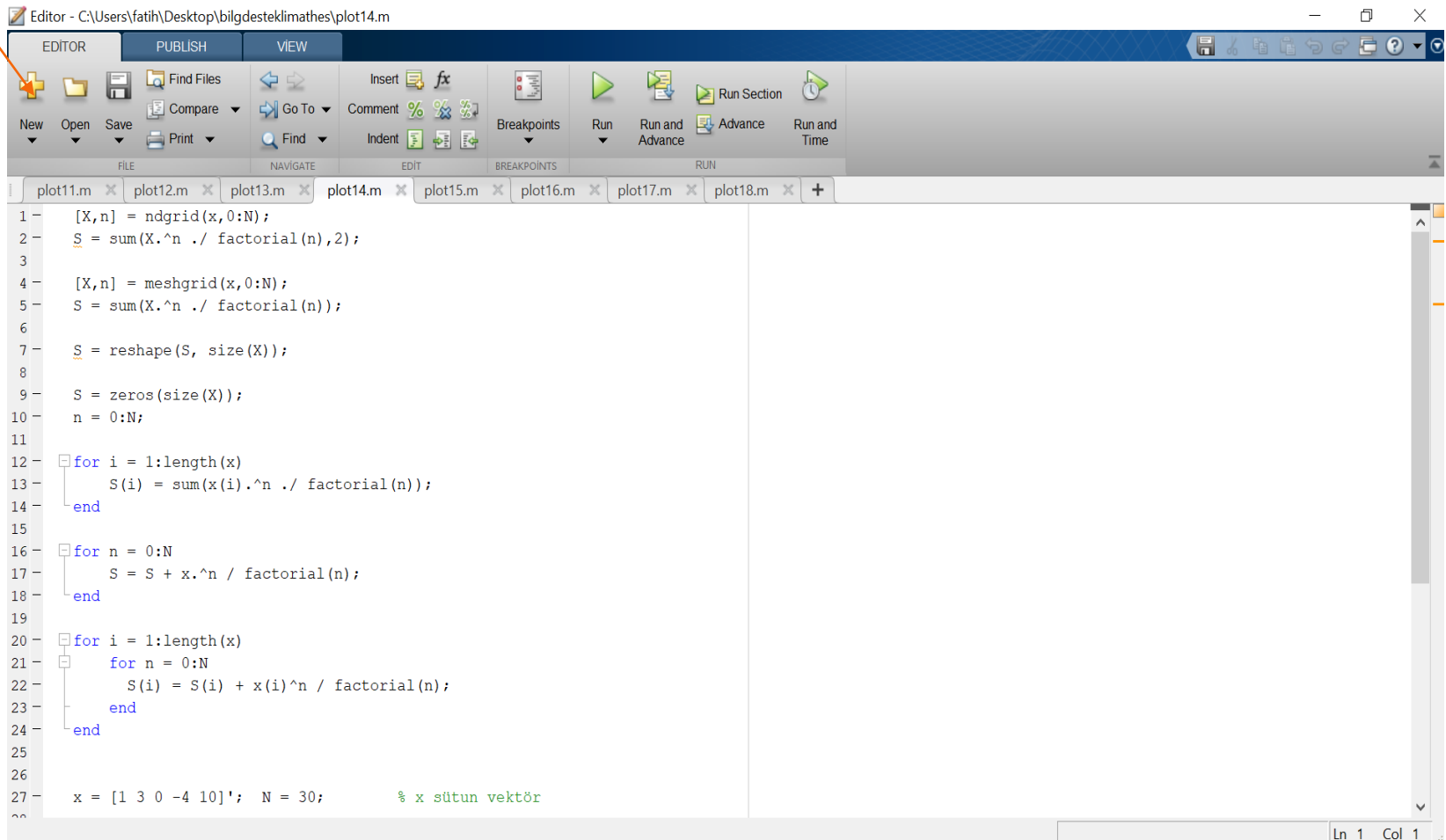
current
folder

workspace
window

command window



2. MATLAB Düzenleyici



The image shows the MATLAB Editor window with the file 'plot14.m' open. The window has a title bar 'Editor - C:\Users\fatih\Desktop\bilgdesteklimathes\plot14.m'. The menu bar includes EDITOR, PUBLISH, and VIEW. The toolbar contains icons for New, Open, Save, Find Files, Compare, Print, Go To, Find, Insert, Comment, Indent, Breakpoints, Run, Run and Advance, Run Section, and Run and Time. The code editor shows the following MATLAB code:

```
1 [X,n] = ndgrid(x,0:N);  
2 S = sum(X.^n ./ factorial(n),2);  
3  
4 [X,n] = meshgrid(x,0:N);  
5 S = sum(X.^n ./ factorial(n));  
6  
7 S = reshape(S, size(X));  
8  
9 S = zeros(size(X));  
10 n = 0:N;  
11  
12 for i = 1:length(x)  
13     S(i) = sum(x(i).^n ./ factorial(n));  
14 end  
15  
16 for n = 0:N  
17     S = S + x.^n / factorial(n);  
18 end  
19  
20 for i = 1:length(x)  
21     for n = 0:N  
22         S(i) = S(i) + x(i)^n / factorial(n);  
23     end  
24 end  
25  
26  
27 x = [1 3 0 -4 10]'; N = 30; % x sütun vektör
```

The status bar at the bottom right shows 'Ln 1 Col 1'.

3. Yardım

Yardım almanın birkaç yolu:

1) MATLAB masaüstünde bulunan **Help** menü öğesi arama yapılabilir yardım penceresini açar.

2) Aşağıdaki komutlardan da yardım desteği alınabilir:

Yorumlar % ile başlar.

>> helpdesk	% yardım tarayıcısını açar
>> help topic	% örneğin, help log10
>> doc topic	% örneğin, doc plot
>> help	% tüm yardım konularının dizisini açar
>> help dir	% tüm dizin üzerinde arama yapar
>> help syntax	% MATLAB syntax üzerinde arama yapar
>> help /	% Operatörler ve özel karakterler
>> docsearch text	% 'text' için HTML tarayıcı araması yapar
>> lookfor topic	% örneğin, lookfor acos

4. Değişkenler, Sabitler, Anahtar Kelimeler

Değişkenler özel bir tür veya depolama beyanı gerektirmez.

Örnekler:

>> x = 3;	% basit skaler
>> y = [4, 5, 6];	% uzunluğu 3 olan satır vektör
>> z = [4; 5; 6];	% uzunluğu 3 olan sütun vektör
>> A = [1, 2, 3; 4, 5, 6];	% 2x3 'lük matris
>> s = 'abcd efg';	% karakter dizisi (string)
>> C = {'abc', 'defg', '123-456'};	% 1x3 'lük hücre dizisi (cell array)

Matematik notasyon

$$y = [4 \quad 5 \quad 6], \quad z = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

class ve **size** fonksiyonları tanımlanan nesnenin sırasıyla türünü ve boyutunu gösterir. Örneğin;

```
>> class(C)  
>> size(C)
```

Command Window

```
>> x=3
```

```
x =
```

```
3
```

```
>> y=[4,5,6]
```

```
y =
```

```
4      5      6
```

```
>> z=[4;5;6]           % z = y'
```

```
z =
```

```
4
```

```
5
```

```
6
```

```
>> A=[1 2 3;4 5 6]
```

```
A =
```

```
1      2      3
```

```
4      5      6
```


Değişkenlerin neler? Onlar nasıl silinir?
Çalışma alanı penceresini ya da aşağıdaki komutları kullan:

who, whos, clear, clc, close

```
>> who
```

Your variables are:

```
A x y z
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
A	2x3	48	double	
x	1x1	8	double	
y	1x3	24	double	
z	3x1	24	double	

```
>> clear all
```

% tüm değişkenleri hafızadan siler

```
>> clc
```

% komut penceresini temizler

```
>> close all
```

% açık olan tüm figürleri kapatır

İşletim sistemi komutları

>> path	% arama yolunu gösterir
>> pathtool	% arama yolunu değiştirir
>> addpath dir	% yola dizin ekler
>> cd dir	% dizini değiştirir
>> pwd	% çalışma dizinini yazdırır
>> dir	% geçerli dizindeki tüm dosyaları diziler
>> what	% sadece MATLAB dosyalarını diziler
>> which file	% dosyanın yerini gösterir
>> edit file	% MATLAB editörünü çağırır
>> quit	% MATLAB'ı kapatır
>> exit	% MATLAB'ı kapatır

Baştan bir değişken olarak tekrar tanımlanamayan bilinen özel matematik sabitleri:

eps	% machine epsilon – kayan nokta doğruluğu (floating-point accuracy)
i, j	% sanal birimi, yani, $\sqrt{-1}$
Inf, inf	% sonsuz
intmax	% belirtilen tamsayı türünün en büyük değeri
intmin	% belirtilen tamsayı türünün en küçük değeri
NaN, nan	% not-a-number, örneğin, $0/0$, inf/inf
pi	% pi sayısı
realmax	% pozitif en büyük kayan noktalı sayı
realmin	% pozitif en küçük kayan noktalı sayı

Not: i, j genellikle dizi ve matrislerin indisleri için kullanılır. Eğer kompleks değerli verilerle uğraşıyorsanız i, j 'yi tekrar tanımlamaktan kaçının.

Özel sabitlerin değerleri

Command Window

```
>> eps
```

```
ans =
```

```
2.2204e-16
```

% 2^{-52} ye eşittir

% MATLAB kayan nokta doğruluğu, yani $2.2204 * 10^{-16}$

```
>> intmax
```

```
ans =
```

```
int32
```

```
2147483647
```

% 32-bit tamsayılar için $2^{31} - 1$

```
>> intmin
```

```
ans =
```

```
int32
```

```
-2147483648
```

% -2^{31} e eşittir

```
>> realmax
```

```
ans =
```

```
1.7977e+308
```

% $(2 - \text{eps}) * 2^{1023}$ e eşittir

% yani, $1.7977 * 10^{308}$

```
>> realmin
```

```
ans =
```

```
2.2251e-308
```

% $2^{-1022} = 2.2251 * 10^{-308}$

Değişken ismi olarak kullanılamayacak özel anahtar kelimeler

Command Window

```
>> iskeyword
```

```
ans =
```

```
20×1 cell array
```

```
{'break'      }  
{'case'       }  
{'catch'      }  
{'classdef'   }  
{'continue'   }  
{'else'       }  
{'elseif'     }  
{'end'        }  
{'for'        }  
{'function'   }  
{'global'     }  
{'if'         }  
{'otherwise'   }  
{'parfor'     }  
{'persistent' }  
{'return'     }  
{'spmd'       }  
{'switch'     }  
{'try'        }  
{'while'      }
```

fx

Ek olarak, 'true' , 'false'

double-precision

Varsayılan MATLAB, IEEE kayan nokta standardını izleyen çift duyarlıklı (64-bit) kayan noktalı sayıları kullanır.

single-precision

MATLAB eğer istenirse tek duyarlıklı (32-bit) kayan noktalı sayıları da kullanabilir.

Görüntü işleme (image processing) veya DSP çiplerini programlama gibi bazı uygulamalarda yararlı olan birkaç tamsayı veri türü de vardır. Tamsayı veri türleri 8, 16, 32 veya 64 bit ve işaretli veya işaretsiz olabilir:

`int8, int16, int32, int64`
`uint8, uint16, uint32, uint64`

Daha fazla bilgi için:

`>> help datatypes`

`>> help class`

`% veri tipini belirler`

Kompleks Sayılar

Varsayılan olarak, MATLAB tüm sayı ve ifadeleri karmaşık (reel olsalar bile) olarak kabul eder.

Karmaşık sayı işlemlerini gerçekleştirmek için özel bir bildirime gerek yoktur.
Örnekler:

```
>> z = 3+4i;           % ya da 3+4j, 3+4*i, 3+4*j
>> x = real(z);        % z'nin reel kısmı
>> y = imag(z);        % z'nin sanal kısmı
>> R = abs(z);         % z'nin mutlak değeri
>> theta = angle(z);   % z'nin radyan cinsinden argümanı
>> w = conj(z);        % kompleks eşlenik, w = 3 - 4i
>> isreal(z);          % z'nin reel veya kompleks olma testi
```

$$z = x + jy = Re^{j\theta}, R = |z| = \sqrt{x^2 + y^2}, \theta = \arctan \frac{y}{x}$$

Kartezyen ve kutupsal form

Matematiksel notasyon: $\theta = \text{Arg}(z)$

Command Window

```
>> z=3+4j  
  
z =  
  
    3.0000 + 4.0000i
```

```
>> x=real(z)
```

```
x =  
  
    3
```

```
>> y=imag(z)
```

```
y =  
  
    4
```

```
>> R=abs(z)
```

```
R =  
  
    5
```

```
>> theta=angle(z)
```

```
theta =  
  
    0.9273
```

```
>> abs(z-R*exp(j*theta))+abs(z-x-j*y)
```

```
ans =  
  
    6.2804e-16
```

% radyan cinsinden

% test

eşdeğer tanımlar:

$z = 3+4*j$

$z = 3+4i$

$z = 3+4*i$

$z = \text{complex}(3,4)$

Ekran formatları

>> format	% varsayılan – 4 ondalık sayı
>> format short	% varsayılan ile aynı
>> format long	% 15 ondalık sayı
>> format short e	% 4 ondalık sayı – üstel gösterim
>> format short g	% 4 ondalık sayı – üstel veya sabit
>> format long e	% 15 ondalık sayı – üstel gösterim
>> format long g	% üstel veya sabit
>> format shorteng	% 4 ondalık sayı – mühendislik
>> format longeng	% 15 ondalık sayı – mühendislik
>> format hex	% onaltılık taban
>> format rat	% rasyonel yaklaşım
>> format compact	% dikey boşluğu korur
>> format loose	% dikey boşluğu ihmal eder
>> vpa(x, digits)	% değişken – duyarlık – aritmetik (variable – precision - arithmetic)

Bunlar sadece ekran formatını etkiler – dahili olarak tüm hesaplamalar tam hassasiyetle (double precision) yapılır.

Örnek – $10 \cdot \pi$ sayısının farklı formatlardaki ekran değerleri

31.4159	% format ya da format short
31.415926535897931	% format long
3.1416e+001	% format short e
31.416	% format short g
3.141592653589793e+001	% format long e
31.4159265358979	% format long g
31.4159e+000	% format shorteng
31.4159265358979e+000	% format longeng

Command Window

```
>> vpa(10*pi)
```

% sembolik araç kutusu (symbolic toolbox)

```
ans =
```

```
31.415926535897932384626433832795
```

```
>> vpa(10*pi,20)
```

% basamak sayısını belirtir

```
ans =
```

```
31.415926535897932385
```

fx >>

```
>> help format  
>> help vpa  
>> help digits
```

Girdi/çıkış fonksiyonları: **disp**, **input**

Command Window

```
>> x=10; disp('the value of x is:'); disp(x);  
the value of x is:  
10
```

```
>> x = input('enter x: ')  
enter x: 100
```

% sayısal girdi
% 100, kullanıcı tarafından girildi

```
x =  
  
100
```

tek alıntıda hızlı dizi

```
>> y = input('enter string: ', 's');  
enter string: abcd efg  
>> y = input('enter string: ')  
enter string: 'abcd efg'
```

% dizi girdisi

```
y =  
  
'abcd efg'
```

Dizi alıntı yapılmadan girilir.
Dizi alıntı yapılarak girilir.

>> help fprintf
>> help sprintf

>> help disp
>> help input
>> help menu

5. diziler ve matrisler

diziler ve matrisler MATLAB'da en önemli veri nesneleridir.

Özetle aşağıdakileri inceleyeceğiz:

- a) Satır ve sütun vektörler**
- b) Transpoze operatörü, ' '**
- c) Kolon operatörü, :**
- d) Eşit aralıklı elemanlar, linspace**
- e) dizi elemanlarına ulaşım**
- f) Dinamik atama (allocation) ve atamayı kaldırma (deallocation)**
- g) Ön atama**

Etkili bir MATLAB programlamanın anahtarı:

VEKTÖRLEŞTİRMEK

ve döngülerden kaçınmak

Şu iki alternatif hesaplamayı karşılaştıralım:

```
x = [2, -3, 4, 1, 5, 8];  
y = zeros(size(x));  
for n = 1:length(x)  
    y(n) = x(n)^2  
end
```

```
x = [2, -3, 4, 1, 5, 8];  
y = x.^2;
```

eleman bazında üstelleştirme .^

sıradan üstelleştirme ^

cevap: $y=[4, 9, 16, 1, 25, 64]$

Command Window

```
>> x = [ 0 1 2 3 4 5] % satır vektör
```

```
x =
```

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

```
>> x = 0:5 % satır vektör
```

```
x =
```

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

```
>> x = [0 1 2 3 4 5]' % sütun vektör, (0:5)'
```

```
x =
```

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

fx >>

' operatörü veya transpoze operatörü satır vektörü sütun vektöre çevirir, ve tam tersi de doğrudur.

Uyarı: ' aslında eşlenik transpozedir,
eşleniksiz transpoze için .' kullanılmalıdır.

Command Window

```
>> z = [i; 1+2i; 1-i] % sütun vektör
```

```
z =
```

```
0.0000 + 1.0000i  
1.0000 + 2.0000i  
1.0000 - 1.0000i
```

```
>> z.' % eşleniksiz transpoze
```

```
ans =
```

```
0.0000 + 1.0000i    1.0000 + 2.0000i    1.0000 - 1.0000i
```

```
>> z' % eşlenikli transpoze
```

```
ans =
```

```
0.0000 - 1.0000i    1.0000 - 2.0000i    1.0000 + 1.0000i
```

```
>> (z.')' % (z')' ile aynıdır, veya conj(z) ile.
```

```
ans =
```

```
0.0000 - 1.0000i  
1.0000 - 2.0000i  
1.0000 + 1.0000i
```

linspace hakkında:

x = linspace(a, b, N+1);

x = a : (b-a)/N : b;

ile eşdeğerdir.

Yani; [a, b] aralığında eşit aralıklı N+1 nokta
ya da [a, b] aralığını N eşit alt aralığa bölmek

$$x(n) = a + \left(\frac{b-a}{N} \right) (n-1), \quad n = 1, 2, \dots, N+1$$

adım aralığı

Command Window

```
>> x = 0:0.2:1
```

% genel olarak, x = a : s : b

```
x =
```

```
0    0.2000    0.4000    0.6000    0.8000    1.0000
```

```
>> x = linspace(0,1,6)
```

% logspace komutuna da bakın

```
x =
```

```
0    0.2000    0.4000    0.6000    0.8000    1.0000
```

6 nokta, 5 alt aralık

fx >>

Command Window

```
>> x = 0 : 0.3 : 1  
  
x =  
  
0    0.3000    0.6000    0.9000  
  
>> x = 0 : 0.4 : 1  
  
x =  
  
0    0.4000    0.8000  
  
>> x = 0 : 0.7 : 1  
  
x =  
  
0    0.7000
```

fx >> |

adım aralığı

$x = a : s : b;$

$[a, b]$ içindeki alt aralık sayısı, $(b-a)/s$ yuvarlatılarak yani;

$N = \text{floor}((b-a)/s);$

tamsayısına yuvarlatılarak elde edilir.

$\text{length}(x)$ $N+1$ 'e eşittir.

**$x(n) = a + s \cdot (n-1),$
 $n = 1, 2, \dots, N+1$**

**% Örneklerde, yuvarlamadan önce, $(b-a)/s$ aşağıdaki üç durumdadır,
% $1 / 0.3 = 3.3333$, $1 / 0.4 = 2.5$, $1 / 0.7 = 1.4286$**

Not: MATLAB dizi indisleri her zaman 1 ile başlar, 0 ya da negatif olamaz.

>> x = [2, 5, -6, 10, 3, 4]
 ↑ ↑ ↑ ↑ ↑ ↑
 x(1) x(2) x(3) x(4) x(5) x(6)

**Mantıksal dizinleme dışında
(daha sonra anlatılacak.)**

Diğer dillerde, C/C++ ve Fortran gibi, indis 0'dan başlayabilir. Örneğin, aynı dizi C'de aşağıdaki gibi tanımlanmaktadır:

double x[6] = {2, 5, -6, 10, 3, 4}
 ↑ ↑ ↑ ↑ ↑ ↑
 x[0] x[1] x[2] x[3] x[4] x[5]

Genel Kural: $M = C + 1$

dizi elemanlarına ulaşım:

Command Window

```
>> x = [2, 5, -6, 10, 3, 4]
```

```
x =
```

```
     2     5    -6    10     3     4
```

```
>> length(x) % x'in uzunluğu, size(x)'e de bakın.
```

```
ans =
```

```
     6
```

```
>> x(1) % ilk eleman
```

```
ans =
```

```
     2
```

```
>> x(3) % üçüncü eleman
```

```
ans =
```

```
    -6
```

```
>> x(end) % son eleman (uzunluğu bilmeye gerek yok)
```

```
ans =
```

```
     4
```

dizi elemanlarına ulaşım:

```
>> x(end-3:end) % son 4
```

```
ans =
```

```
    -6    10     3     4
```

```
>> x(3:5) % 3'ten 5'e
```

```
ans =
```

```
    -6    10     3
```

```
>> x(1:3:end) % 3 girişte bir
```

```
ans =
```

```
     2    10
```

```
>> x(1:2:end) % 2 girişte bir
```

```
ans =
```

```
     2    -6     3
```

dizi elemanlarına ulaşım:

```
>> x(end:-1:1)    % sondan başa, fliplr(x) ile aynı
```

```
ans =
```

```
     4     3    10    -6     5     2
```

```
>> x([3,1,5])      % [x(3), x(1), x(5)]
```

```
ans =
```

```
    -6     2     3
```

```
>> x(end+3) = 8
```

```
x =
```

```
     2     5    -6    10     3     4     0     0     8
```

fx >>

**otomatik hafıza ataması
(automatic memory re-allocation)**

Otomatik hafıza ataması ve atamayı kaldırma (de-allocation):

Command Window

```
>> clear x  
>> x(3)=-6
```

```
x =  
  
     0     0    -6
```

```
>> x(6)=4
```

```
x =  
  
     0     0    -6     0     0     4
```

```
>> x(end)=[]
```

% son girdiyi sil

```
x =  
  
     0     0    -6     0     0
```

```
>> x=[2,5,-6,10,3,4];
```

```
>> x(3)=[]
```

% üçüncü girdiyi sil

```
x =  
  
     2     5    10     3     4
```

Ön atama (pre-allocation)

```
>> clear x  
>> x=zeros(1,6)    % 1x6 'lık sıfır dizisi
```

```
x =  
  
    0    0    0    0    0    0
```

```
>> x=zeros(6,1)    % 6x1 'lik sıfır dizisi
```

```
x =  
  
    0  
    0  
    0  
    0  
    0  
    0
```

Ön atama çok büyük diziler için kullanışlıdır, mesela, uzunluk $> 10^4$. Örnek olarak, ses veya görüntü dosyaları ile ilgilenirken, veya sonlu elemanlar metodunu kullanırken.

fx >> |

```
>> help zeros  
>> help ones
```

Dinamik atama (dynamic allocation) ve ön atama (preallocation) uygulaması

```
clear x;
for k=[3, 7, 10]
    x(k)=3+0.1*k;
    disp(x)
end
```

```
>> clear x
>> for k=[3,7,10]
    x(k)=3+0.1*k;
    disp(x);
end
```

% k sırayla 3, 7, 10 değerleri için çalışır ve anlık x vektörünü gösterir.

0	0	3.3000								
0	0	3.3000	0	0	0	3.7000				
0	0	3.3000	0	0	0	3.7000	0	0	4.0000	

fx >>

```
x=zeros(1,10);
for k=[3, 7, 10]
    x(k)=3+0.1*k;
    disp(x)
end
```

```
>> x=zeros(1,10);
>> for k=[3,7,10]
    x(k)=3+0.1*k;
    disp(x);
end
```

% 10 uzunluğunda x'in ön ataması

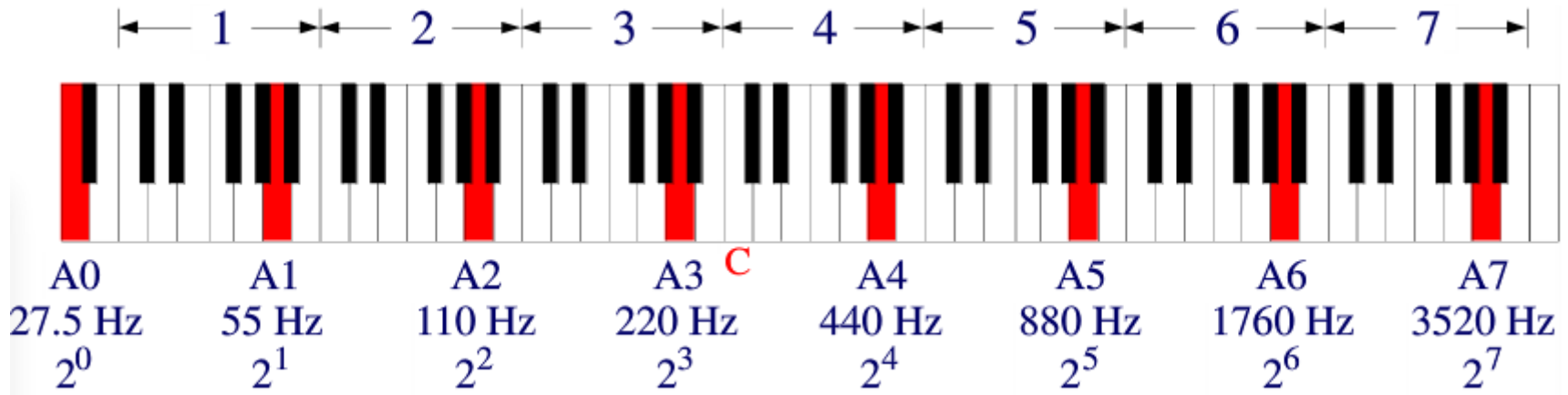
0	0	3.3000	0	0	0	0	0	0	0	0
0	0	3.3000	0	0	0	3.7000	0	0	0	0
0	0	3.3000	0	0	0	3.7000	0	0	4.0000	

fx >>

Örnek: Oktav frekans ölçekleri

Görevler:

- 1) 88 tuşlu standart bir piyano klavyesinin 88 frekansını hesaplamak ve çizmek
- 2) Oktav frekans ölçeklerin kavramını tanıtmak
- 3) Orta (dördüncü) oktavın ana notalarını (do, re, mi, fa, sol, la, si, do) oluşturup yazdırmak ve PC'nin ses kartında ileri ve geri oynatmak



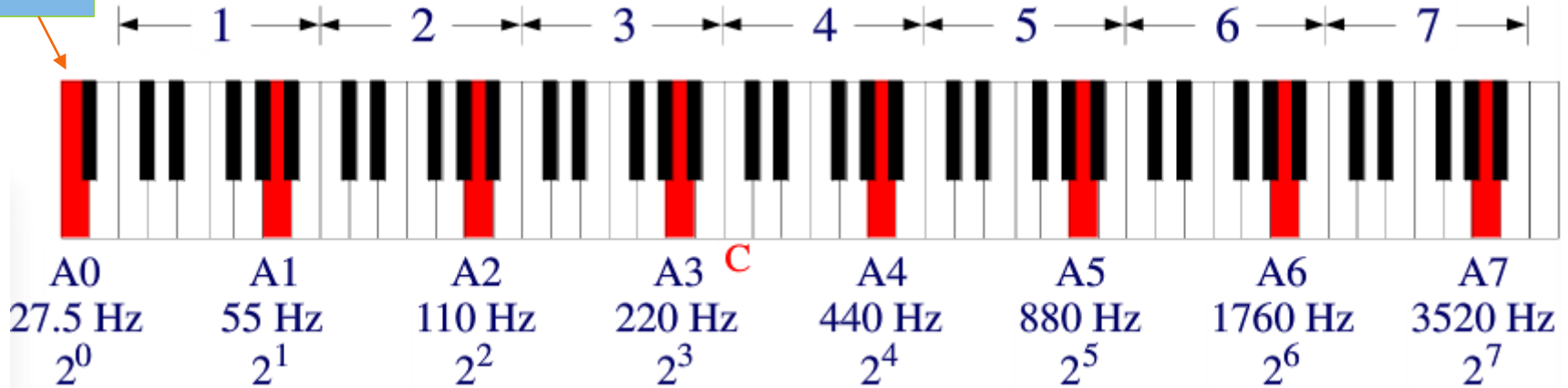
$$\text{oktav} = \log_2 \left(\frac{f}{f_0} \right) \Rightarrow f = f_0 \cdot 2^{\text{oktav}}$$

$$y(t) = \sin(2\pi f t)$$

Ton üretir ve bunu MATLAB'ın **sound()** fonksiyonuna gönderir

k=0

k=87



$$\text{oktav} = \log_2 \left(\frac{f}{f_0} \right) \Rightarrow f = f_0 \cdot 2^{\text{oktav}}$$

Tuşlar

k = [0, 1, 2, ..., 87]

ile indekslenir.

**Ardışık tuşlar bir oktavın
1/12'si ile ayrılır.**

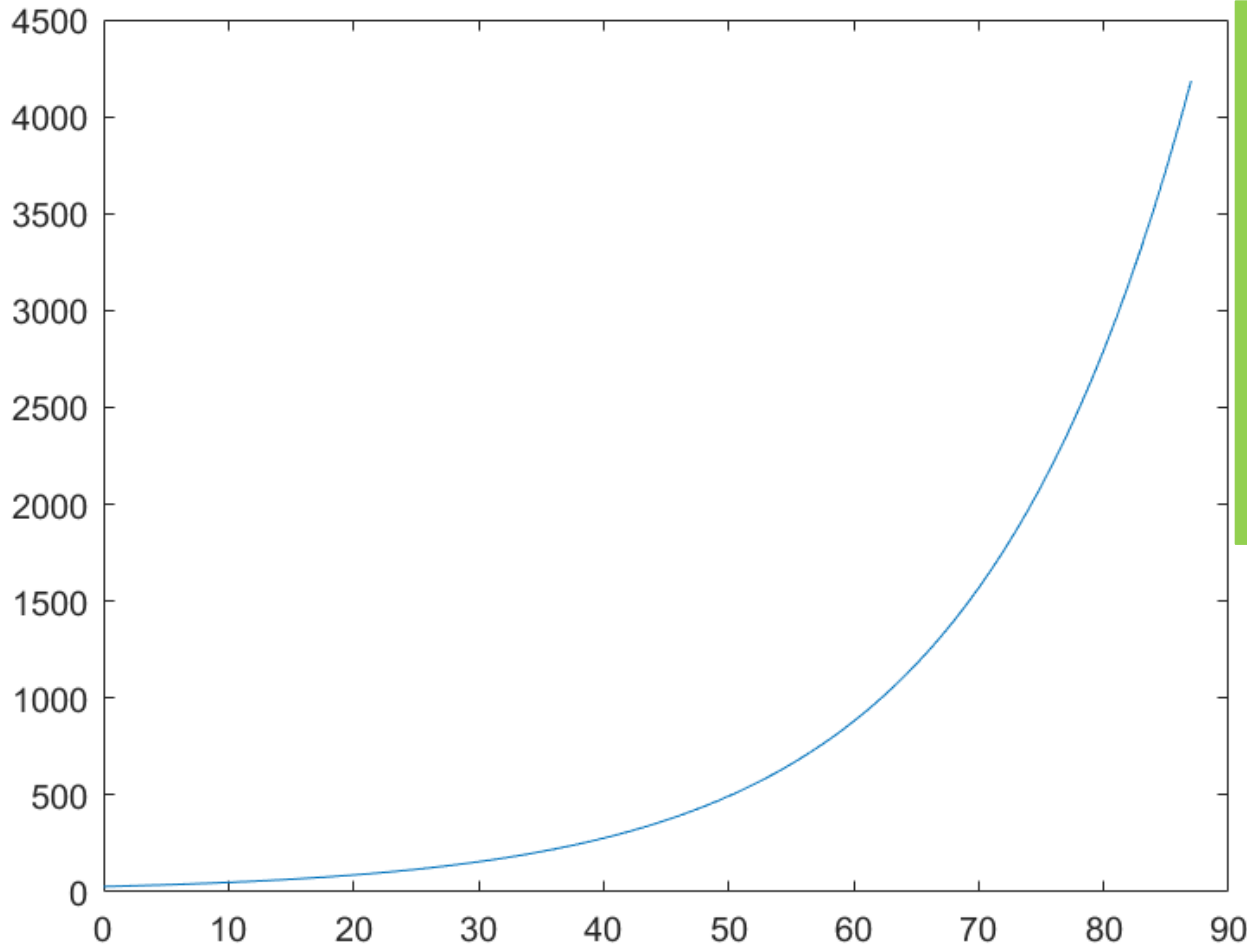
f0 = 27.5;

k = 0:87;

f = f0 * 2.^(k/12);

figure; plot(k,f);

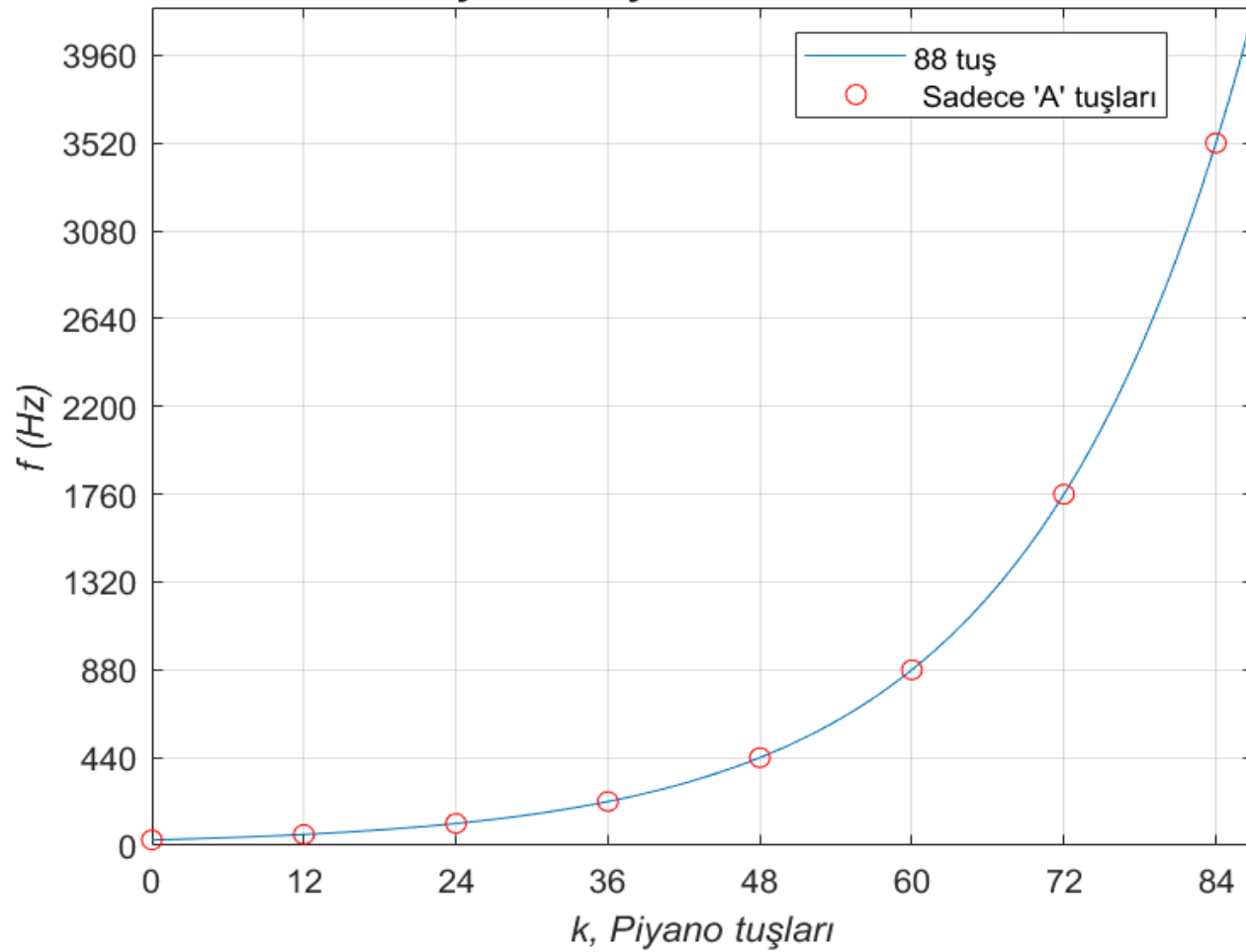
figure; plot(k, log2(f/f0))



**Varsayılan sistem
seçimleri kapsamında
eksen limitlerinin,
etiketlerin vs...
olmadığı sıradan basit
grafik**

**(daha iyi bir grafik
sonraki slaytta)**

Piyano klavyesindeki frekanslar



```
ka = 0:12:87;  
fa = f(ka+1);  
figure; plot(k, f, ka, fa, 'ro')
```

kırmızı, boş yuvarlak

ka-fa grafiği, yani A tuşları

```
% ka = [0, 12, 24, 36, 48, 60, 72, 84]
```

```
% fa = [27.5, 55, 110, 220, 440, 880, 1760, 3520]
```

Şimdi, grafiğe eksen etiketleri (**axis labels**), eksen limitleri (**axis limits**), ölçüm çentikleri (**tick marks**), grid, başlık (**title**) ve açıklamalar (**legends**) eklemek için birkaç komut verilecek.

```
>> help plot
```

Not: fa 'yı f 'in bir alt kümesi olarak tanımladık, fakat direkt olarak şöyle tanımlayabilirdik;

```
fa = f0 * 2.^(ka/12);
```

```
ka = 0:12:87;  
fa = f(ka+1);  
figure; plot(k, f, ka, fa, 'ro')
```

kırmızı, boş yuvarlak

ka-fa grafiği, yani A tuşları

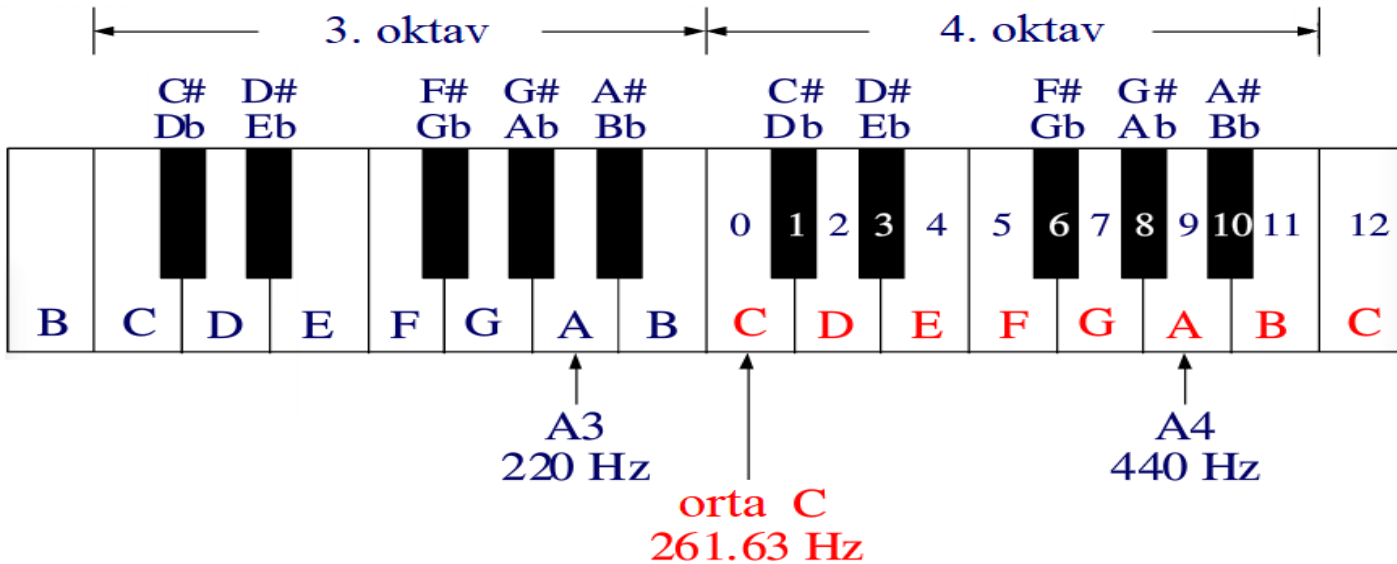
```
% ka = [0, 12, 24, 36, 48, 60, 72, 84]  
% fa = [27.5, 55, 110, 220, 440, 880, 1760, 3520]
```

```
xlim([0, 87]); set(gca, 'xtick', 0:12:87);  
ylim([0, 4200]); set(gca, 'ytick', 0:440:4000);  
title('Piyano klavyesindeki frekanslar');  
xlabel('\it{k}, Piyano tuşları');  
ylabel('\it{f} (Hz)');  
legend('88 tuş', 'Sadece "A" tuşları', 'location', 'nw');  
grid on;  
print -depsc octave2.eps  
print -dmeta octave2.wmf
```

eksen limitleri ve
ölçüm çentikleri

% grafiği color EPS dosyası olarak kaydeder
% grafiği windows metafile olarak kaydeder

Not: fa 'yı f 'in bir alt kümesi olarak tanımladık,
fakat direkt olarak şöyle tanımlayabilirdik;
 $fa = f_0 * 2^{(ka/12)}$;



Orta C A3 üzerindeki bir oktavın 3/12'si ya da A4 altındaki bir oktavın 9/12'sidir.

$$261.63 = 220.2^{3/12} = 440.2^{-9/12}$$

4. oktav tuşları, $k = 0:12$, MATLAB index = $k+1 = 1:13$

Majör tuşlar k 'nin bir alt kümesidir, $m = [0, 2, 4, 5, 7, 9, 11, 12]$

4. oktavda frekansların hesabı

$f_c = 220 * 2^{(3/12)};$
 $k = 0:12;$
 $f = f_c * 2^{(k/12)};$

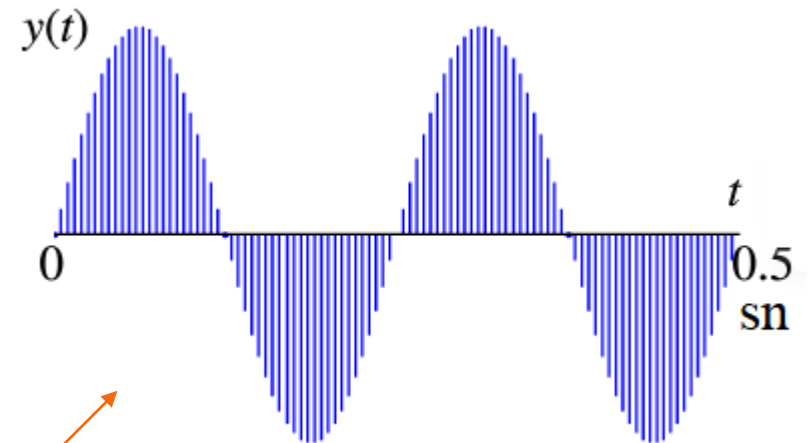
% Orta C'nin frekansı
% Sadece 4. oktavdaki tuşlar
% 4. oktavın frekansları

% $f_c = 440 * 2^{(-9/12)};$ % alternatif hesap

Sonra, her majör tuşun f 'leri için,
yarım saniyelik nota üretelim ve
PC'nin ses kartında çalalım (kartın
varsayılan örnek hızında,
 $f_s = 8192$ örnek / sn.):

$$y(t) = \sin(2\pi f t), 0 \leq t \leq 0.5 \text{ sn.}$$

Yani, $t = 0 : T : 0.5 \text{ sn.}$



Zaman örnekleri, $T = 1 / f_s = 0.122 \text{ msn.}$
varsayılan örnek aralığı ile aralıklandırılır.

4. oktavda majör notaların üretilmesi ve çalınması

```
fs = 8192; T = 1/fs;      % Varsayılan örnek oran
Tmax = 0.5;              % Notalar için yarım saniyelik süre
t = 0 : T : Tmax;        % length(t) = 4097 nokta
                          % T = 1 / fs = 0.1221 msn. adımları

m = [0 2 4 5 7 9 11 12]; % 4. oktavda majör tuşlar
                          % CDEFGABC = do re mi fa sol la si do

for i = m+1               % m+1 = [1 3 5 6 8 10 12 13]
    y = sin(2*pi*f(i)*t); % y'nin yarım saniyelik süresi var
    sound(y, fs);         % y fs oranında çalar
end
pause;                   % bir tuşa basılana kadar duraklatır

for i = fliplr(m+1)       % fliplr(m+1) = [13 12 10 8 6 5 3 1]
    y = sin(2*pi*f(i)*t); % ters sırada çalar
    sound(y, fs);
end
```

Frekansların ve tuş isimlerinin yazdırılması

k	okt=k/12	$f=f_c \cdot 2^{(k/12)}$	tuslar	
0	0.0000	261.63	C	do
1	0.0833	277.18	C#	
2	0.1667	293.66	D	re
3	0.2500	311.13	D#	
4	0.3333	329.63	E	mi
5	0.4167	349.23	F	fa
6	0.5000	369.99	F#	
7	0.5833	392.00	G	sol
8	0.6667	415.30	G#	
9	0.7500	440.00	A	la
10	0.8333	466.16	A#	
11	0.9167	493.88	B	si
12	1.0000	523.25	B#	do

fx >>

hücre dizileri
(cell arrays)

% Frekansların ve tuş isimlerinin yazdırılması

% Tuş isimlerini **hücre dizisi (cell array)** olarak tanımlarız

```
tuslar = {'C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#', ...  
         'A', 'A#', 'B', 'B#'};
```

boş dizi

```
notalar = {'do', '', 're', '', 'mi', 'fa', '', 'sol', '', ...  
          'la', '', 'si', 'do'};
```

cell array için {...} kullan

```
fprintf('\n');  
fprintf('k      okt=k/12      f=fc*2^(k/12)      tuslar\n');  
fprintf('-----\n');
```

```
for i=k+1  
    fprintf('%2d      %6.4f      %6.2f      %-2s      %-3s\n', ...  
          i-1, k(i)/12, f(i), tuslar{i}, notalar{i});
```

end

>> help fprintf % formatted printing
>> doc fprintf

Bir sonraki satıra
devam etmek için ...