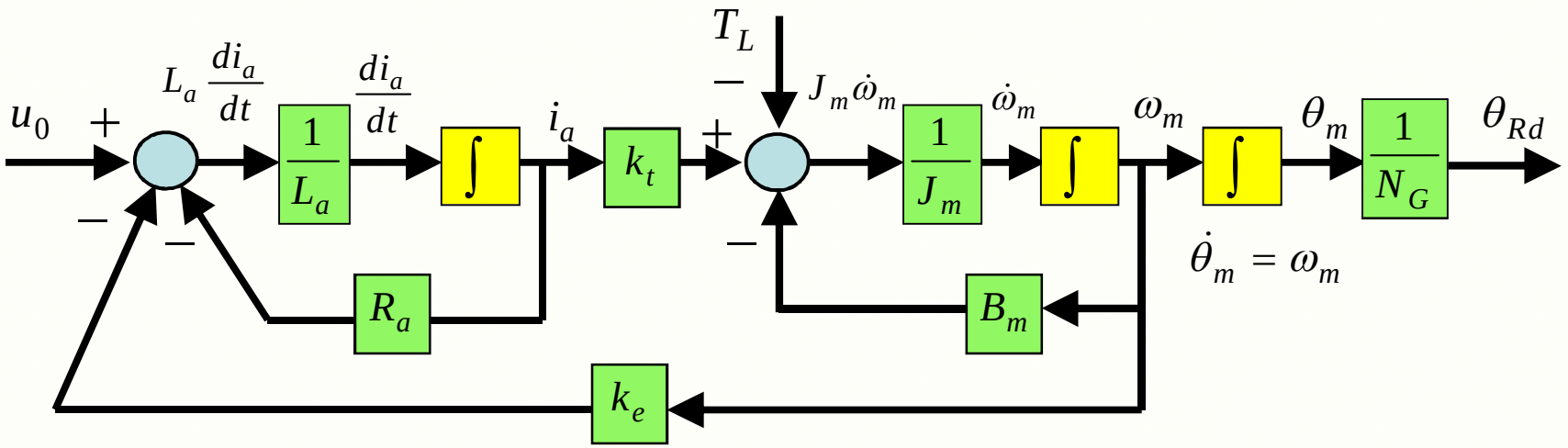
Engineering Softwares



Prof. Dr. İsmail H. Altaş

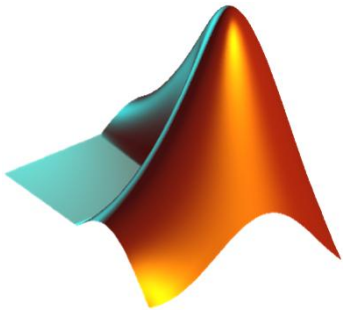
Karadeniz Teknik Üniversitesi
Elektrik-Elektronik Mühendisliği
61080 Trabzon

www.altas.org

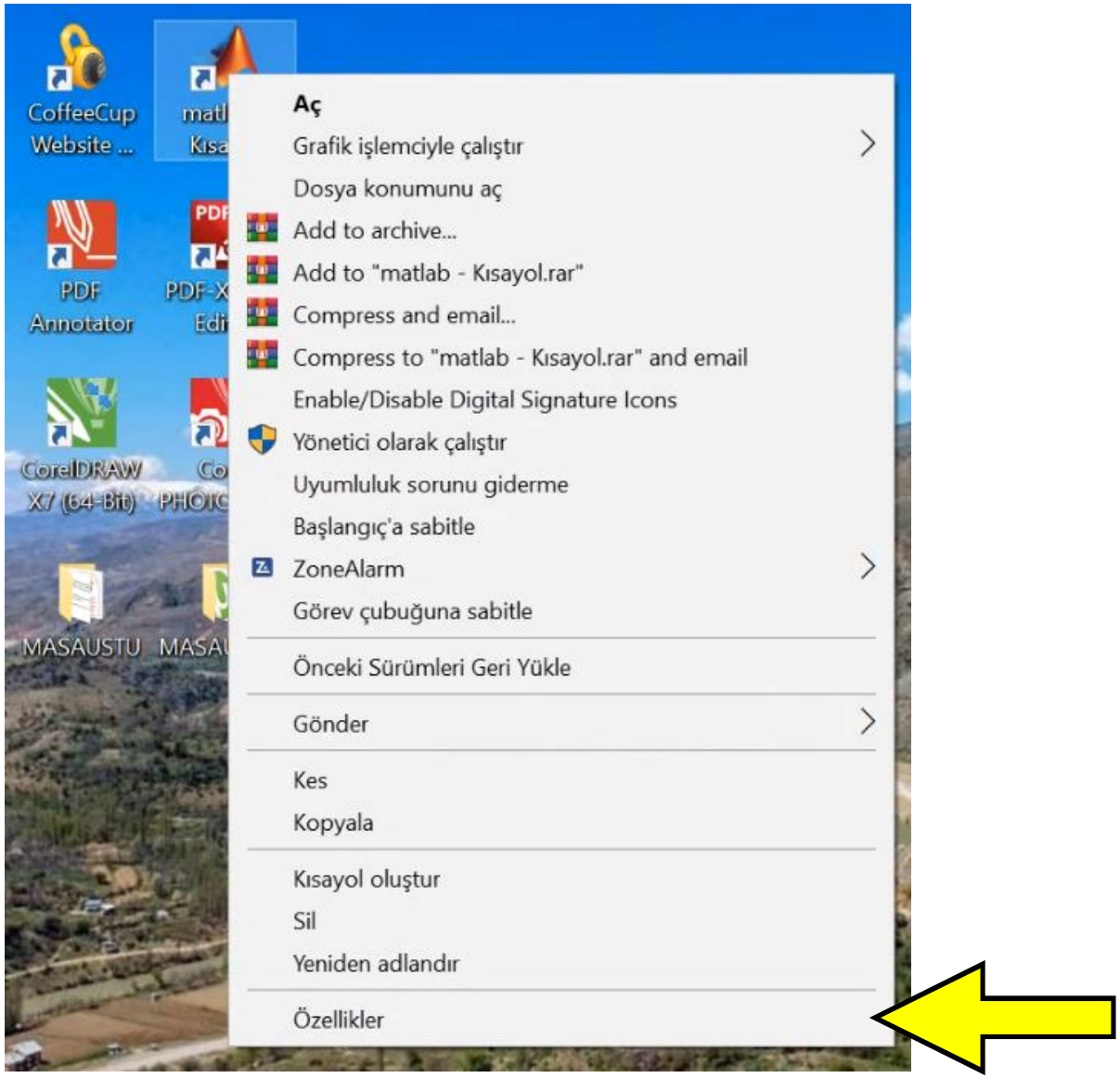
Engineering Softwares

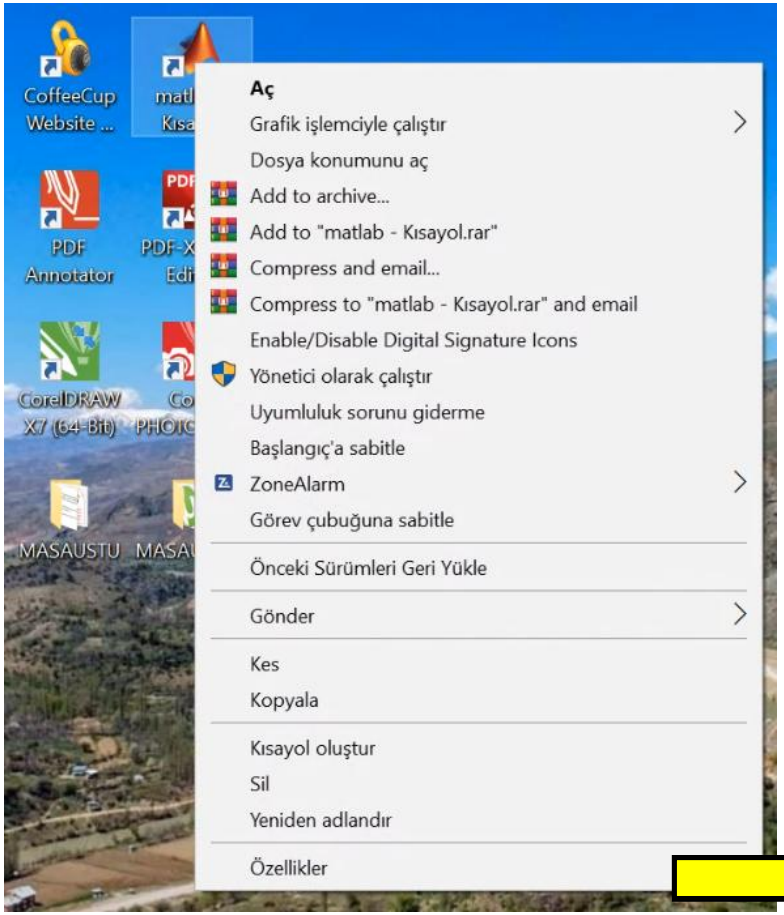
Chapter 1 – MATLAB Basics

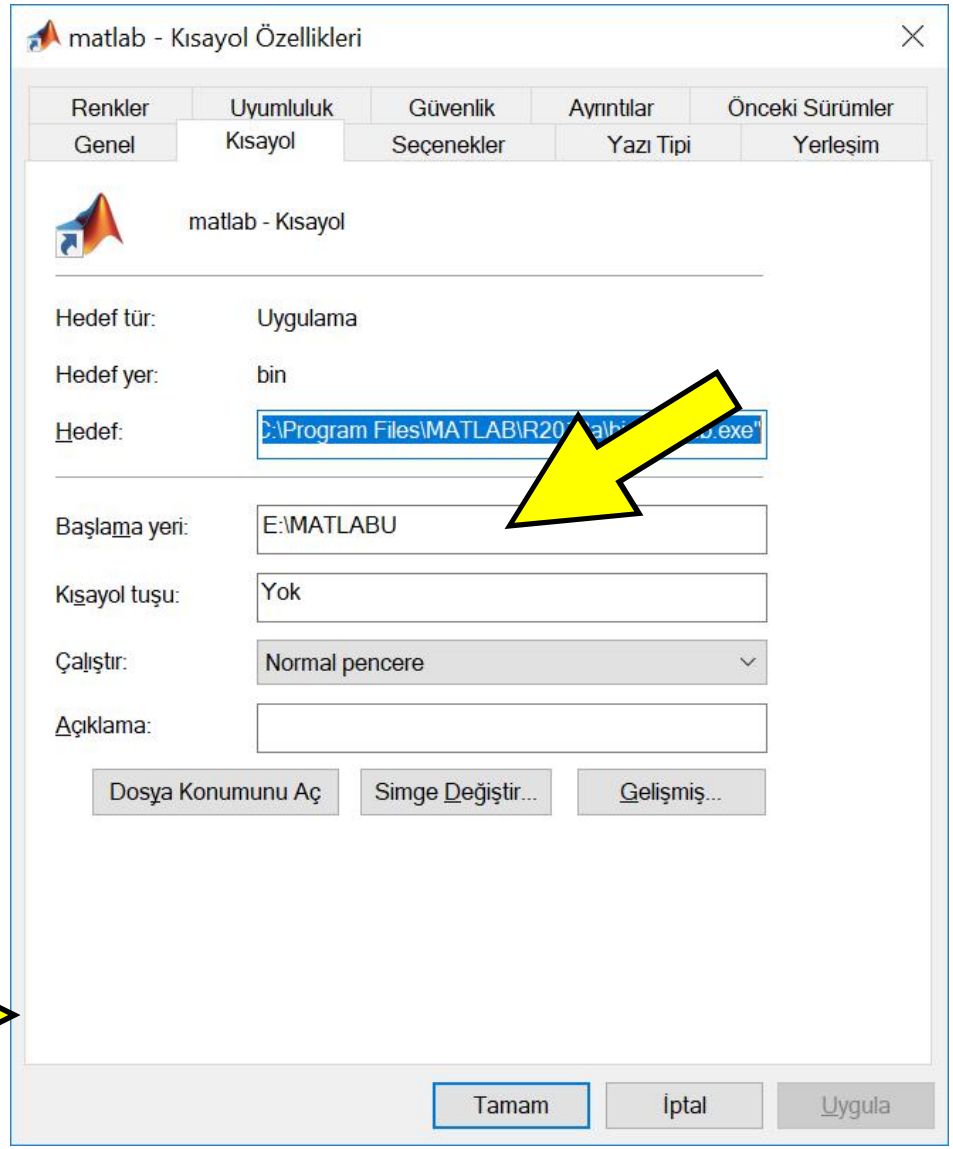
Publication of these course presentation notes by others on any platform is subject to permission. **Remember, Stealing is not sharing.**

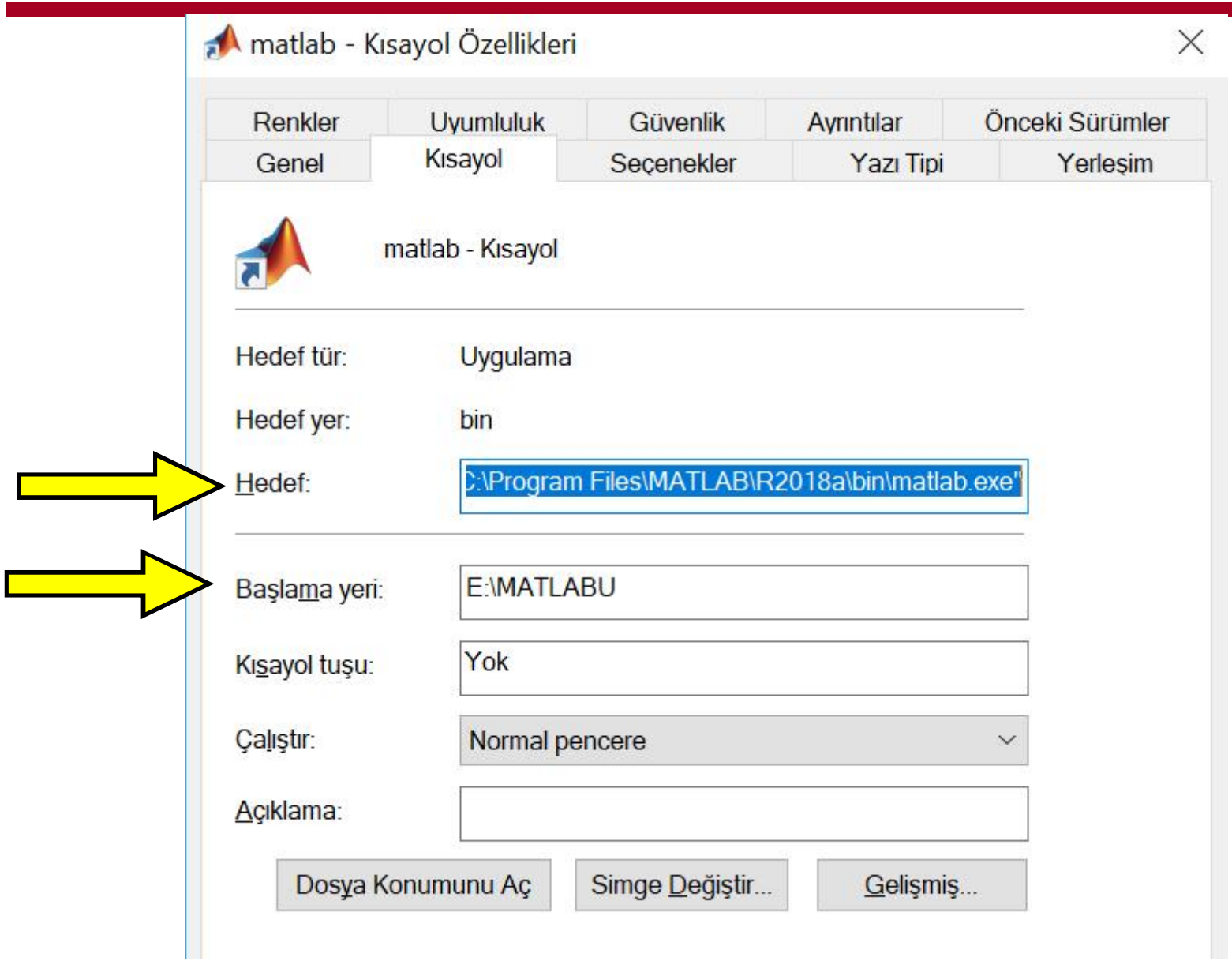


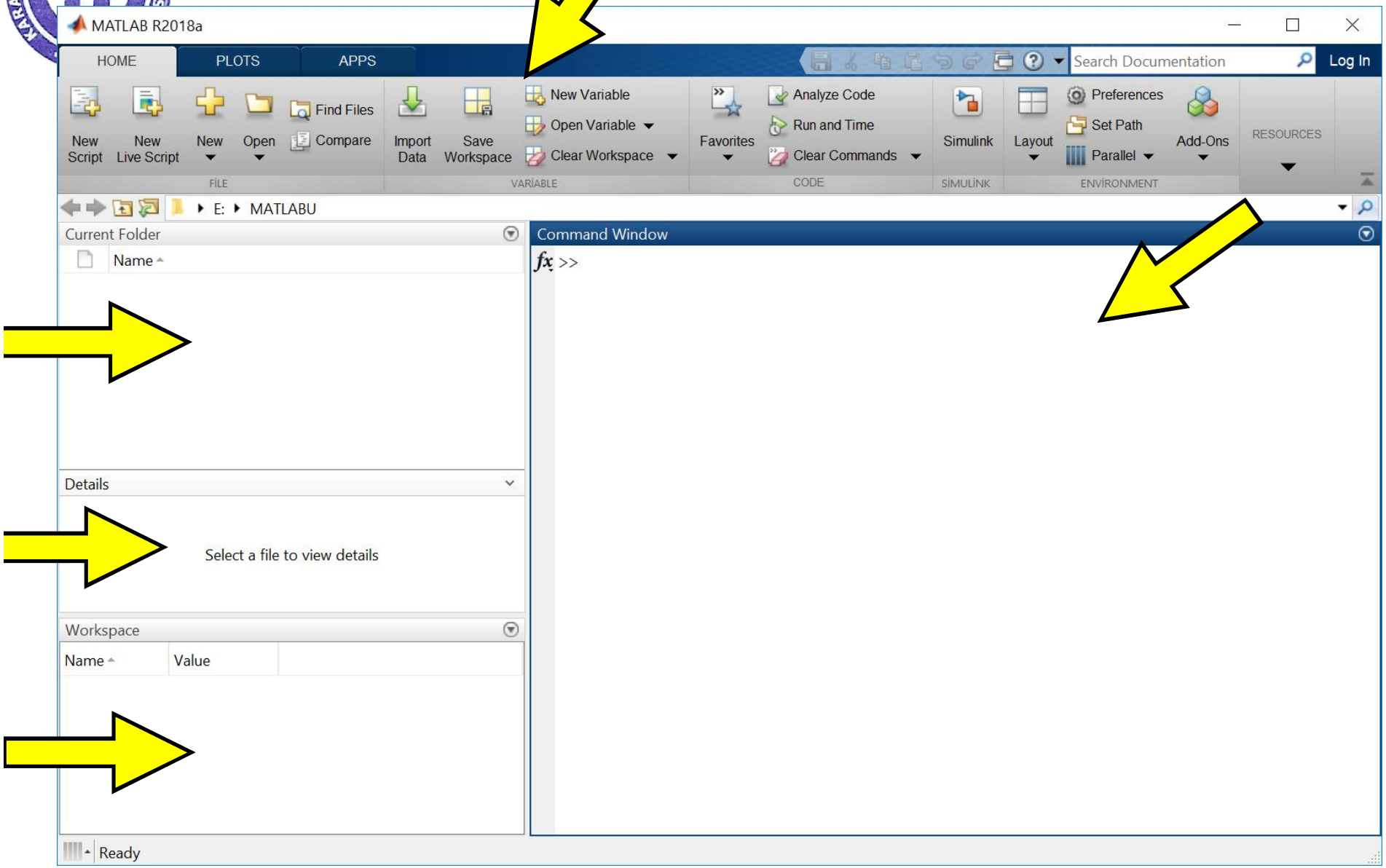
MATLAB

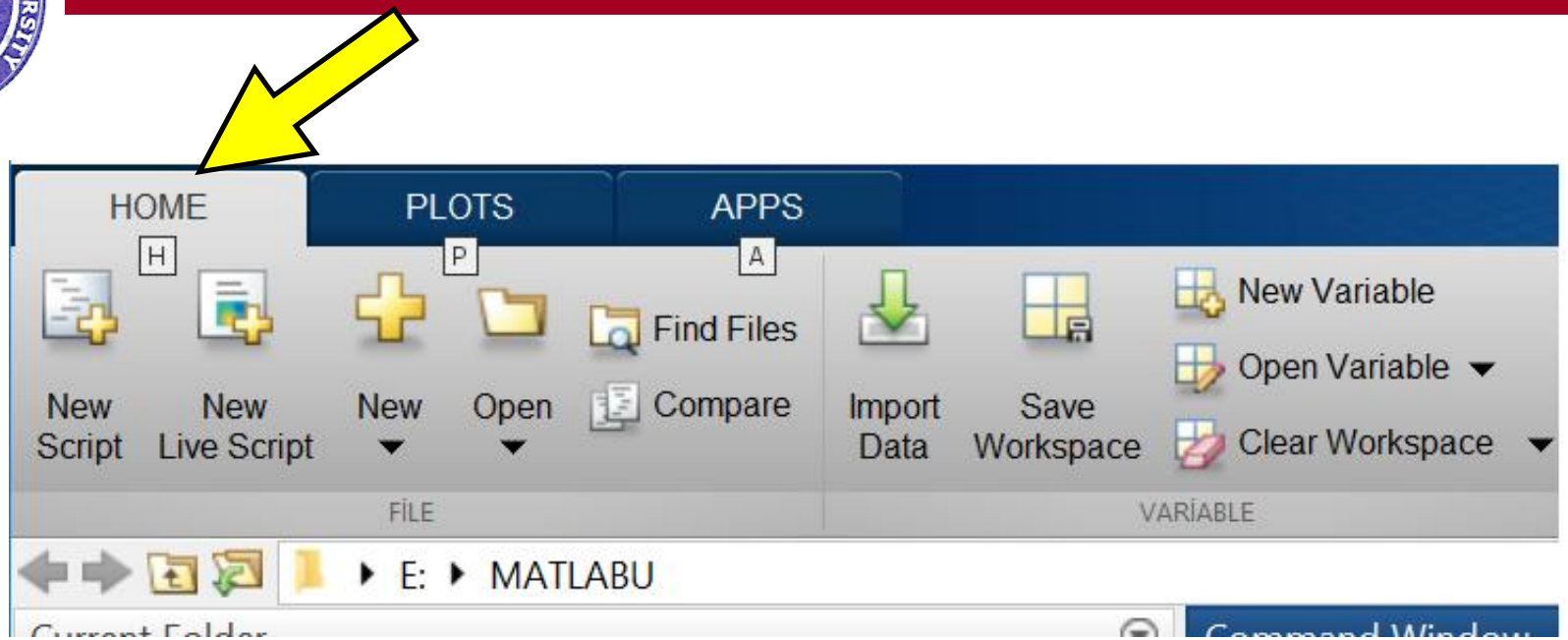

CoffeeCup Website ...
PDF Annotator
CorelDRAW X7 (64-Bit)
MASAUSTU
Aç
Grafik işlemciyle çalıştır
Dosya konumunu aç
Add to archive...
Add to "matlab - Kısayol.rar"
Compress and email...
Compress to "matlab - Kısayol.rar" and email
Enable/Disable Digital Signature Icons
Yönetici olarak çalıştır
Uyumluluk sorunu giderme
Başlangıç'a sabitle
ZoneAlarm
Görev çubuğuna sabitle
Önceki Sürümleri Geri Yükle
Gönder
Kes
Kopyala
Kısayol oluştur
Sil
Yeniden adlandır
Özellikler

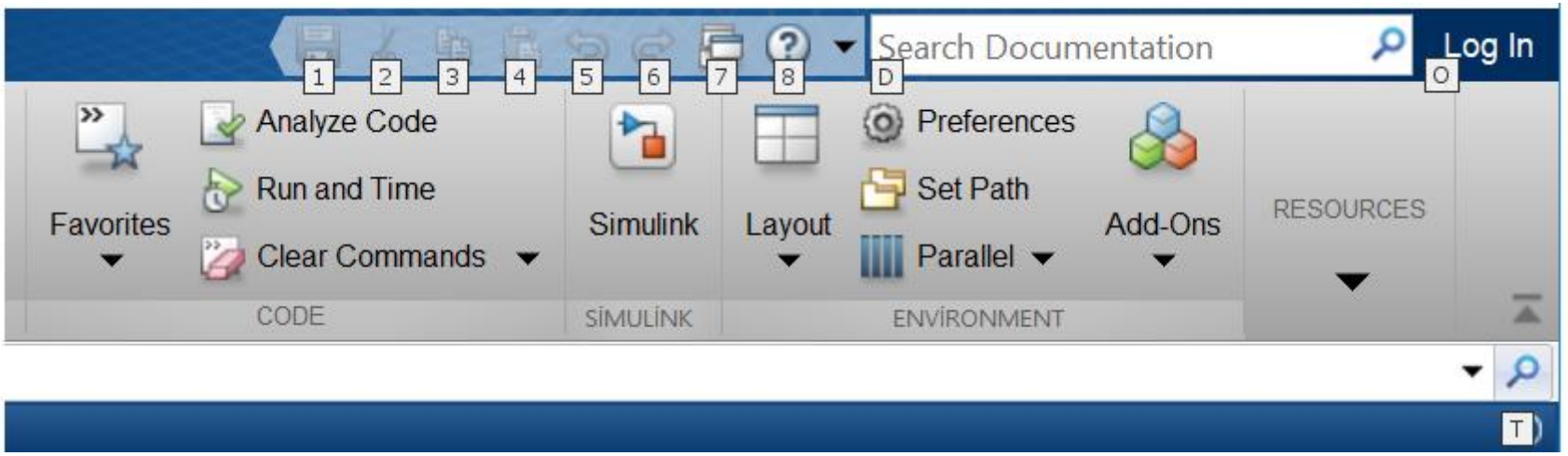

CoffeeCup Website ...
PDF Annotator
CorelDRAW X7 (64-Bit)
MASAUSTU
Aç
Grafik işlemciyle çalıştır
Dosya konumunu aç
Add to archive...
Add to "matlab - Kısayol.rar"
Compress and email...
Compress to "matlab - Kısayol.rar" and email
Enable/Disable Digital Signature Icons
Yönetici olarak çalıştır
Uyumluluk sorunu giderme
Başlangıç'a sabitle
ZoneAlarm
Görev çubuğuna sabitle
Önceki Sürümleri Geri Yükle
Gönder
Kes
Kopyala
Kısayol oluştur
Sil
Yeniden adlandır
Özellikler

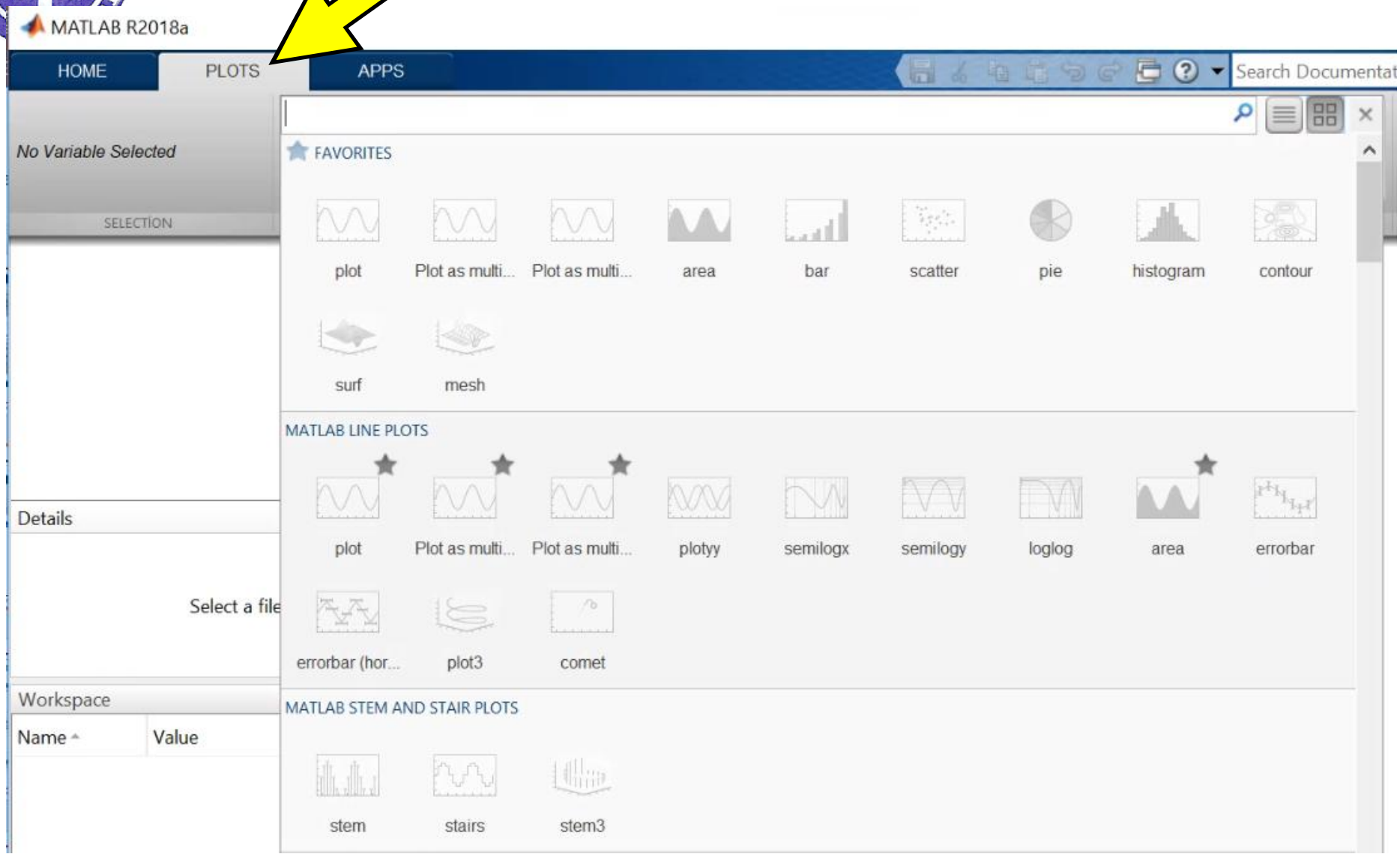

matlab - Kısayol Özellikleri
Renkler
Uyumluluk
Güvenlik
Ayrıntılar
Önceki Sürümler
Genel
Kısayol
Seçenekler
Yazı Tipi
Yerleşim
matlab - Kısayol
Hedef tür: Uygulama
Hedef yer: bin
Hedef:
Başlama yeri: E:\MATLABU
Kısayol tuşu: Yok
Çalıştır: Normal pencere
Açıklama:
Dosya Konumunu Aç
Simge Değiştir...
Gelişmiş...
Tamam
İptal
Uygula

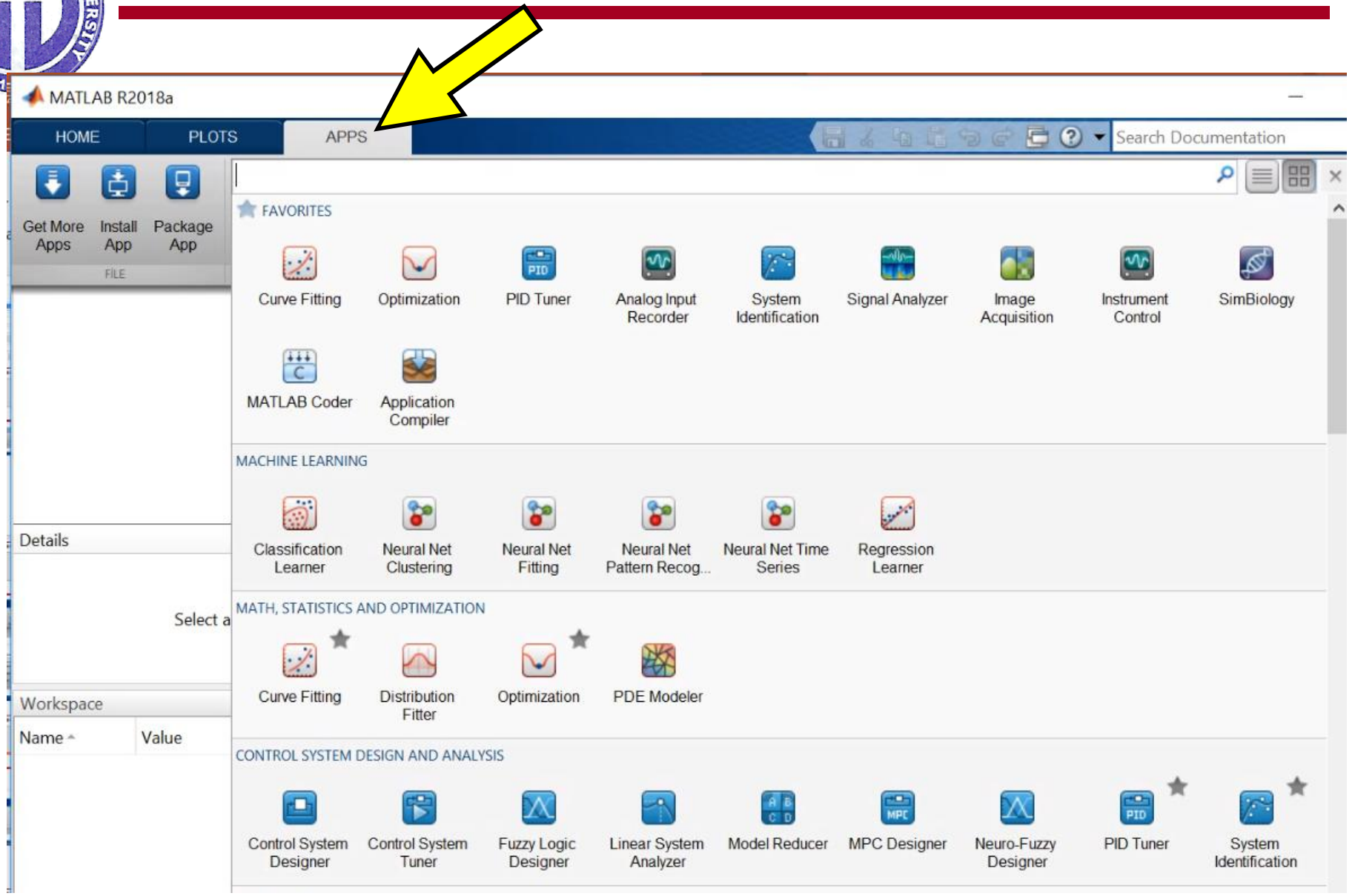

matlab - Kısayol Özellikleri
Renkler
Uyumluluk
Güvenlik
Ayrıntılar
Önceki Sürümler
Genel
Kısayol
Seçenekler
Yazı Tipi
Yerleşim
matlab - Kısayol
Hedef tür:
Uygulama
Hedef yer:
bin
Hedef:
C:\Program Files\MATLAB\R2018a\bin\matlab.exe"
Başlama yeri:
E:\MATLABU
Kısayol tuşu:
Yok
Çalıştır:
Normal pencere
Açıklama:
Dosya Konumunu Aç
Simge Değiştir...
Gelişmiş...

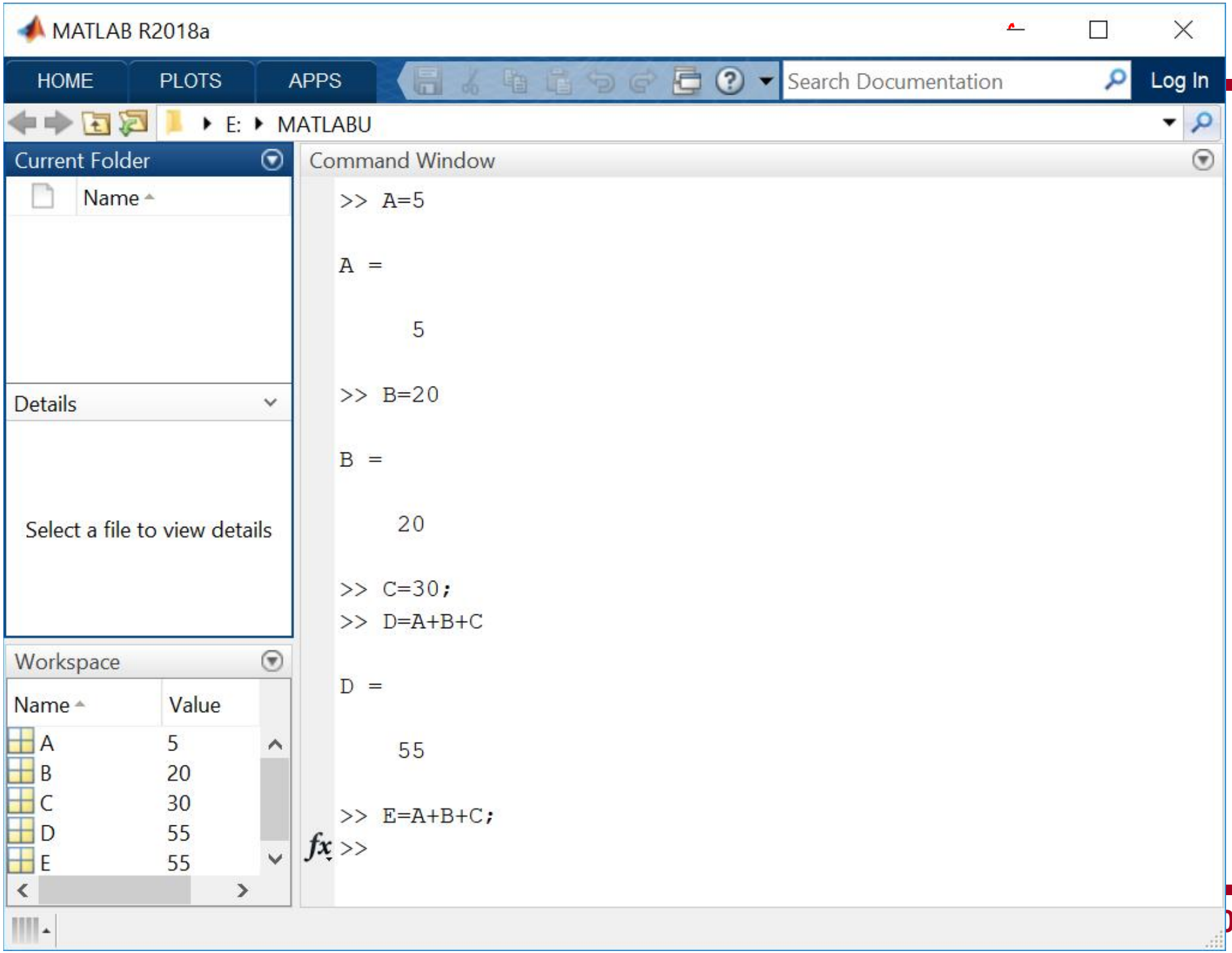

MATLAB R2018a
HOME
PLOTS
APPS
Search Documentation
Log In
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Parallel
Add-Ons
RESOURCES
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
E: ▸ MATLABU
Current Folder
Name
Command Window
fx >>
Details
Select a file to view details
Workspace
Name
Value
Ready

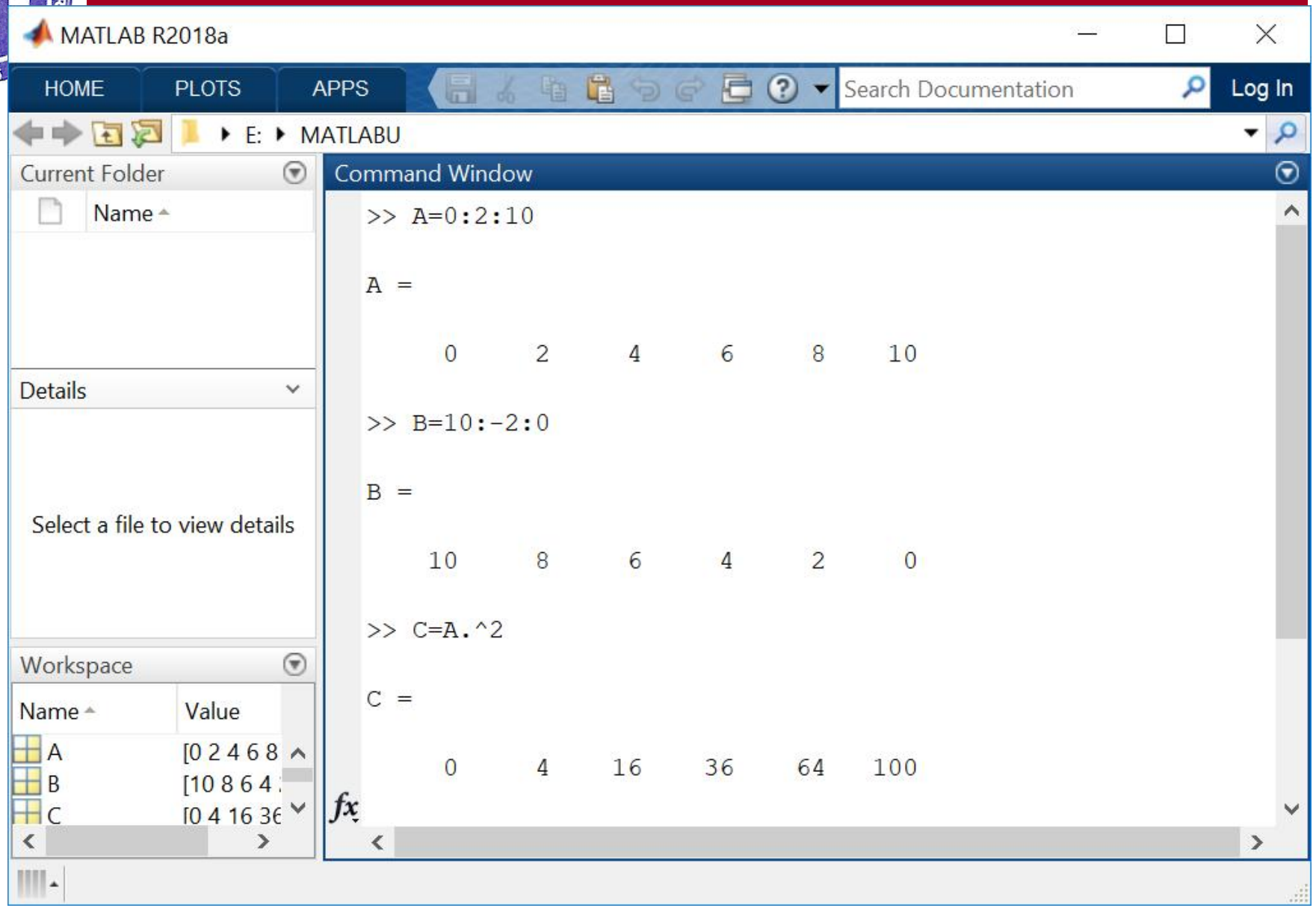

HOME
H
PLOTS
P
APPS
A
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
FİLE
VARİABLE
E:
MATLABU


1
2
3
4
5
6
7
8
Search Documentation
D
Log In
O
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Parallel
Add-Ons
RESOURCES
CODE
SİMULİNK
ENVİRONMENT
T

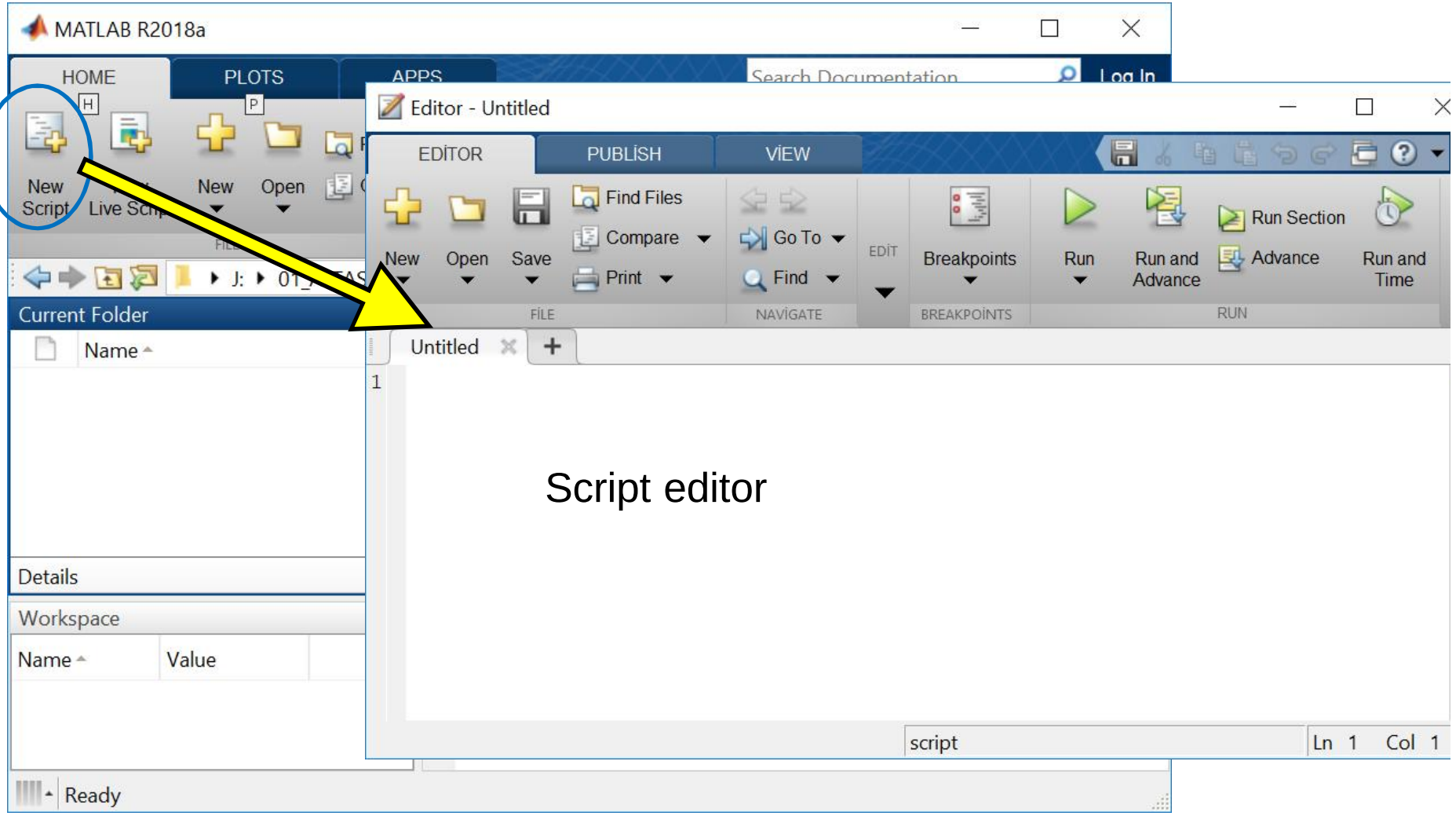

MATLAB R2018a
HOME
PLOTS
APPS
Search Documentat
No Variable Selected
SELECTION
FAVORITES
plot
Plot as multi...
Plot as multi...
area
bar
scatter
pie
histogram
contour
surf
mesh
MATLAB LINE PLOTS
plot
Plot as multi...
Plot as multi...
plotyy
semilogx
semilogy
loglog
area
errorbar
errorbar (hor...
plot3
comet
MATLAB STEM AND STAIR PLOTS
stem
stairs
stem3
Details
Select a file
Workspace
Name
Value


MATLAB R2018a
HOME
PLOTS
APPS
Search Documentation
Get More Apps
Install App
Package App
FILE
FAVORITES
Curve Fitting
Optimization
PID Tuner
Analog Input Recorder
System Identification
Signal Analyzer
Image Acquisition
Instrument Control
SimBiology
MATLAB Coder
Application Compiler
MACHINE LEARNING
Classification Learner
Neural Net Clustering
Neural Net Fitting
Neural Net Pattern Recog...
Neural Net Time Series
Regression Learner
MATH, STATISTICS AND OPTIMIZATION
Curve Fitting
Distribution Fitter
Optimization
PDE Modeler
CONTROL SYSTEM DESIGN AND ANALYSIS
Control System Designer
Control System Tuner
Fuzzy Logic Designer
Linear System Analyzer
Model Reducer
MPC Designer
Neuro-Fuzzy Designer
PID Tuner
System Identification
Details
Select a
Workspace
Name
Value

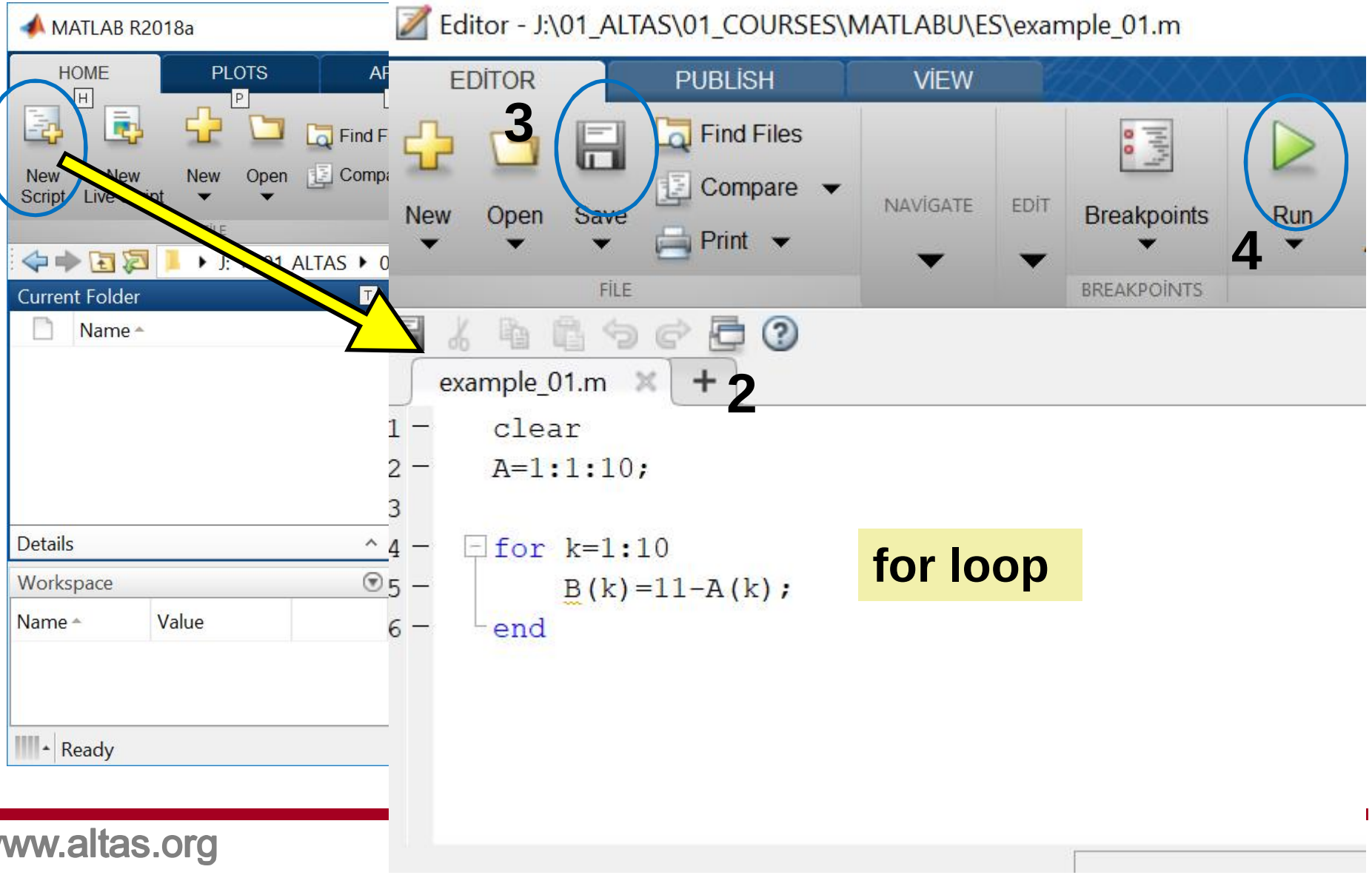

MATLAB R2018a
HOME
PLOTS
APPS
Search Documentation
Log In
E: ▸ MATLABU
Current Folder
Name
Details
Select a file to view details
Workspace
Name
Value
A 5
B 20
C 30
D 55
E 55
Command Window
>> A=5
A =
5
>> B=20
B =
20
>> C=30;
>> D=A+B+C
D =
55
>> E=A+B+C;
>>


MATLAB R2018a
HOME
PLOTS
APPS
Search Documentation
Log In
E: ▸ MATLABU
Current Folder
Name
Details
Select a file to view details
Workspace
Name
Value
A
[0 2 4 6 8
B
[10 8 6 4
C
[0 4 16 36
Command Window
>> A=0:2:10
A =
0 2 4 6 8 10
>> B=10:-2:0
B =
10 8 6 4 2 0
>> C=A.^2
C =
0 4 16 36 64 100

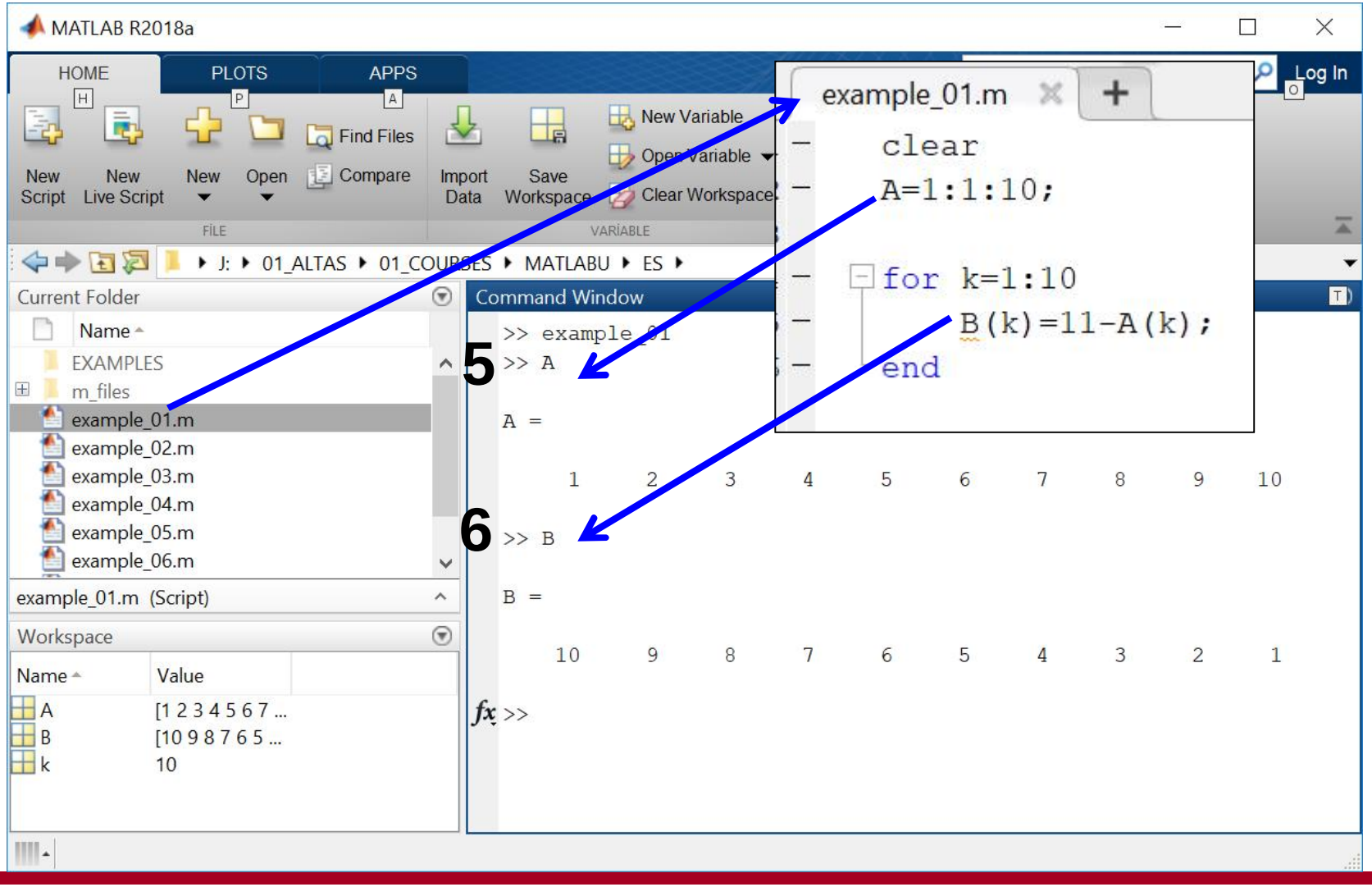
Writing codes with script editor



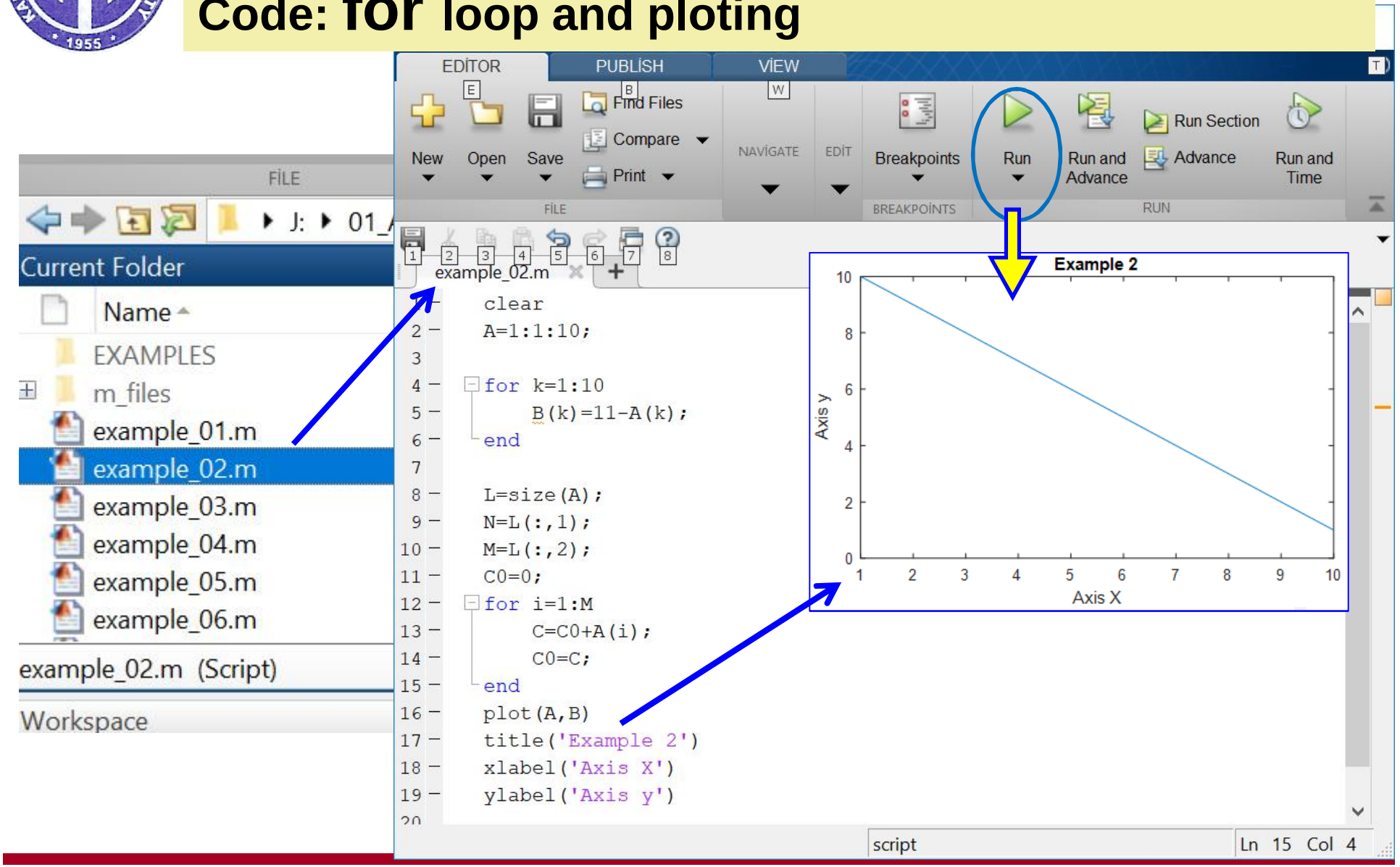
Code: for loop



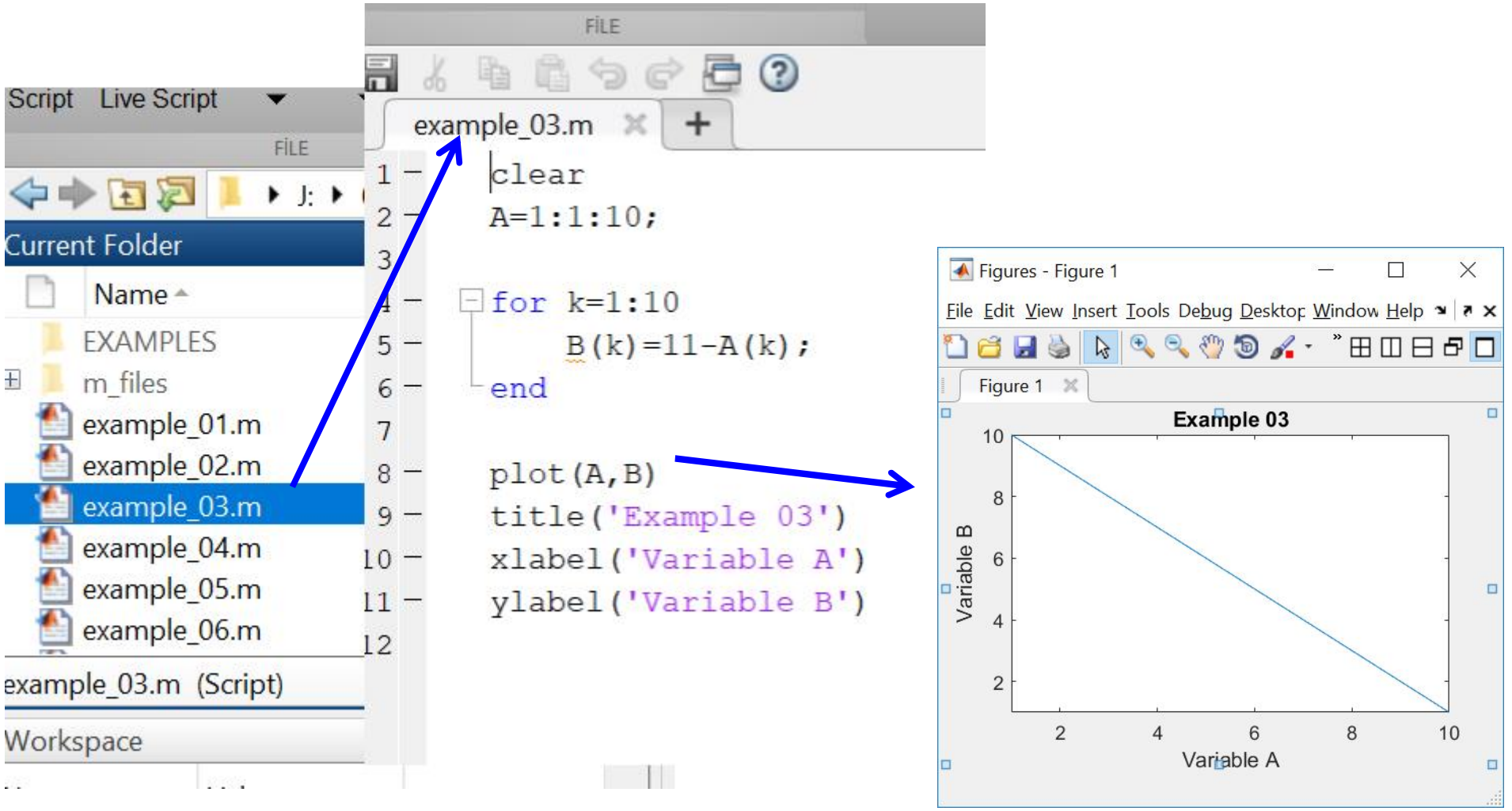
Code: for loop and data generation



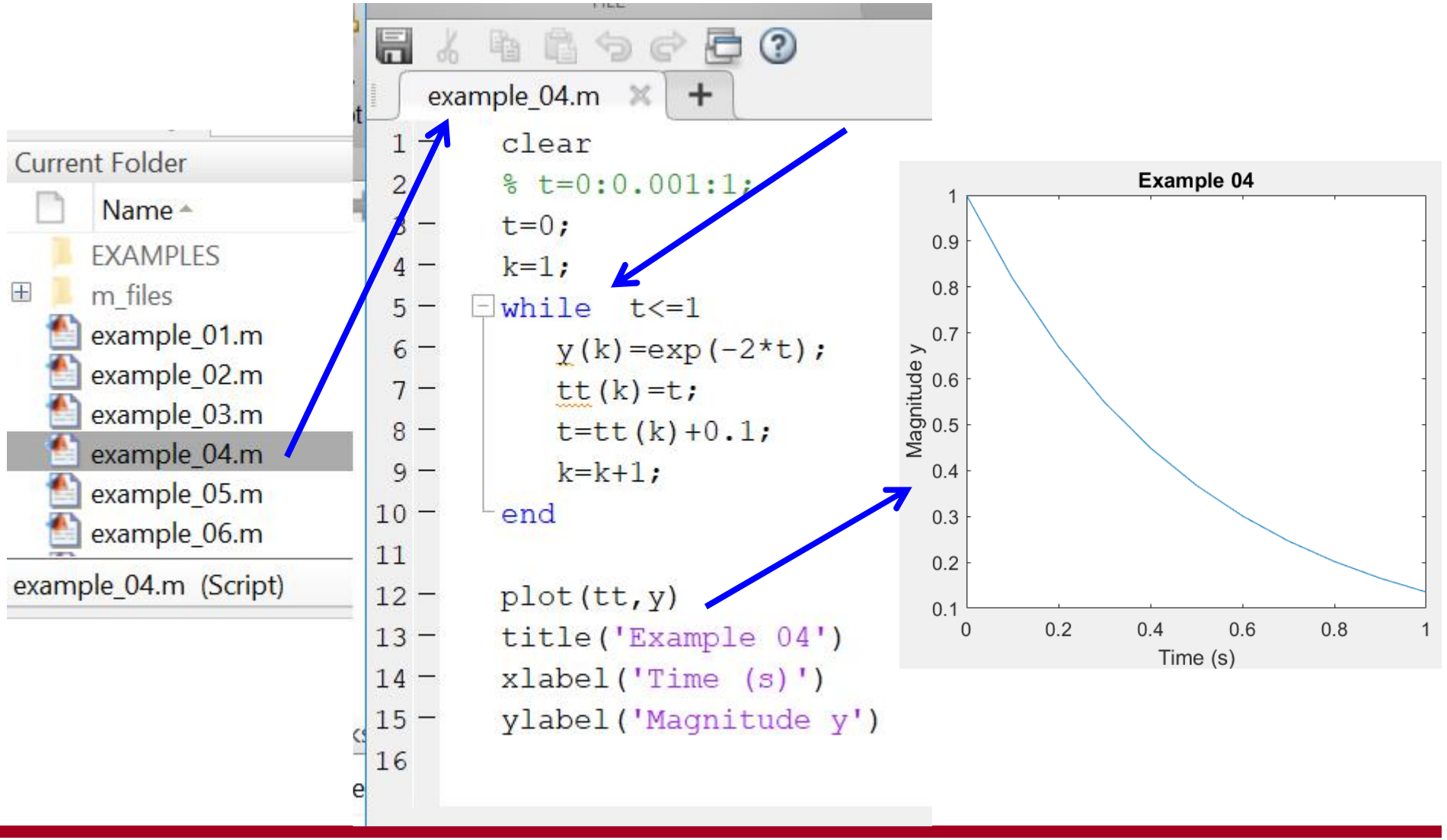
Code: for loop and ploting



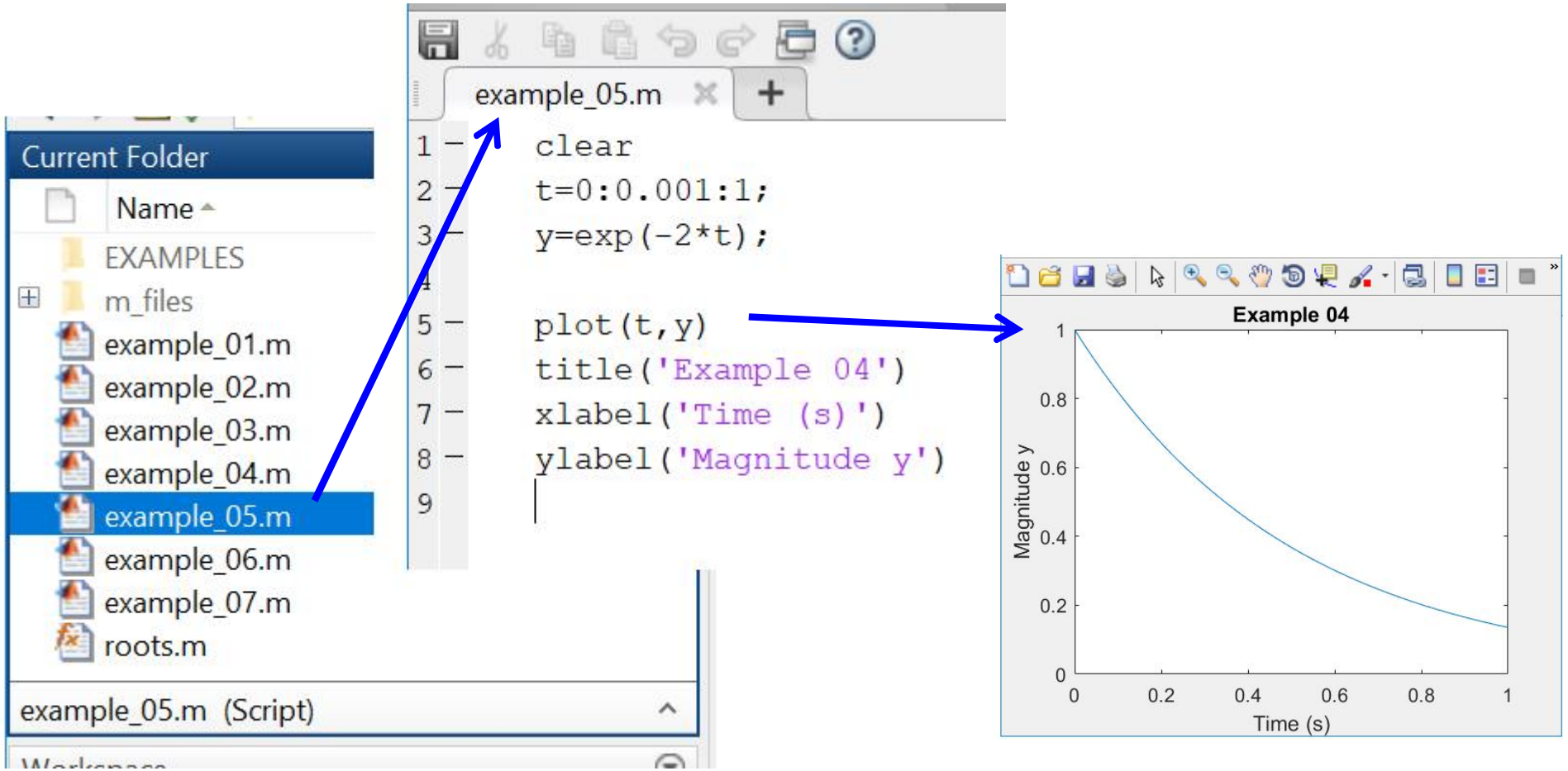
Code: for loop and ploting



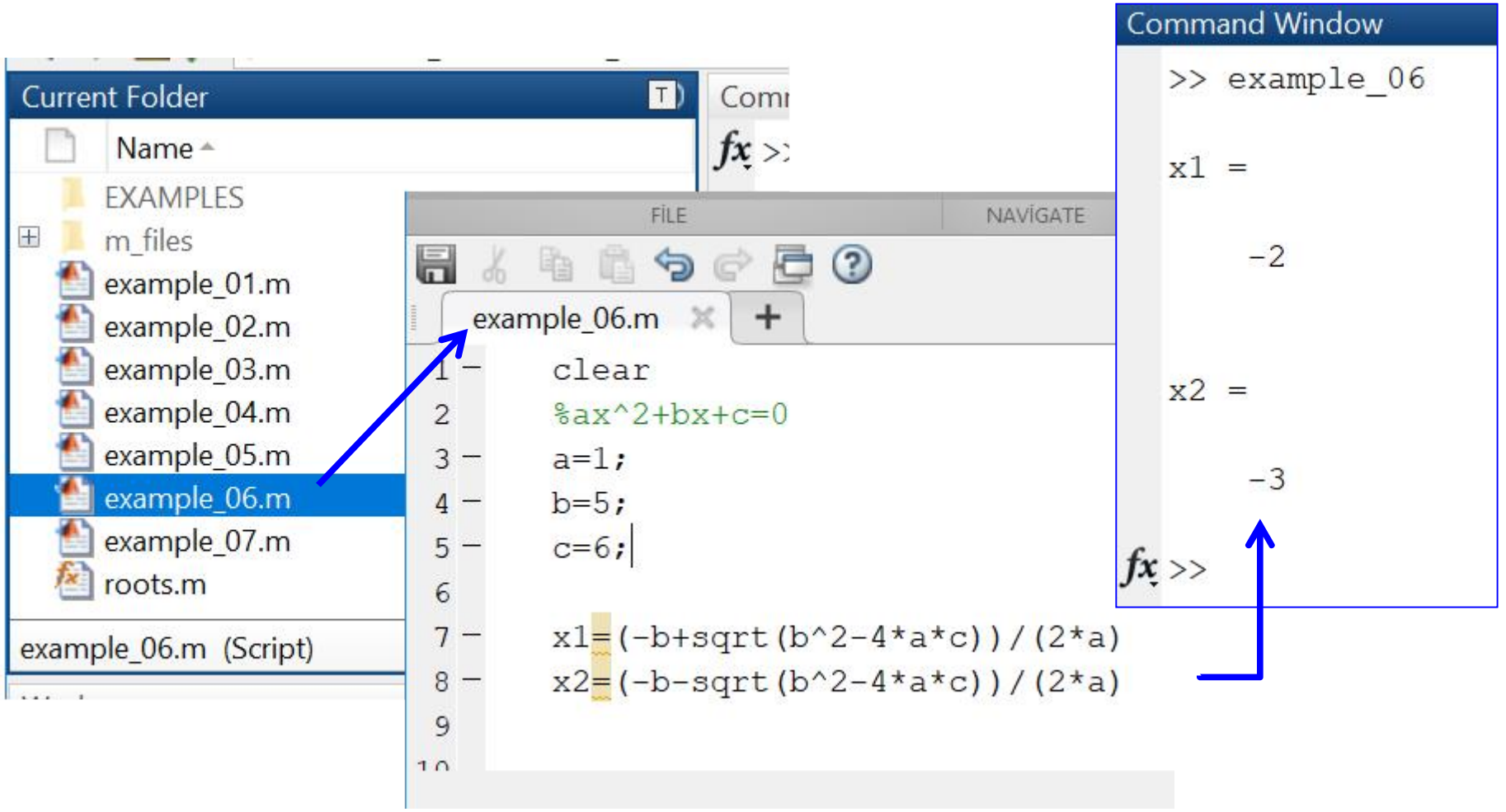
Code: while loop and ploting



Generating data without loops and ploting



Processing equations



Working with functions

roots.m

```
1   function X=roots(a,b,c)
2 -   x1=(-b+sqrt(b^2-4*a*c))/(2*a);
3 -   x2=(-b-sqrt(b^2-4*a*c))/(2*a);
4 -   X=[x1;x2]
5
```

Command Window

```
>> % 5X^2 +3X-7=0
>> X1X2=roots(5,3,-7)

X1X2 =

    0.9207
   -1.5207

>> % X^2 +10X+25=0
>> X1X2=roots(1,10,25)

X1X2 =

    -5
    -5

fx >>
```

Working with functions

Examples

Command Window

```
>> % X^2 +9X+20=0
>> X1X2=roots(1,9,20)

X1X2 =

    -4
    -5

fx >>
```

Command Window

```
>> % X^2 +6X+9=0
>> X1X2=roots(1,6,9)

X1X2 =

    -3
    -3
```

```
>> % X^2 +3X+9=0
>> X1X2=roots(1,3,9)

X1X2 =

  -1.5000 + 2.5981i
  -1.5000 - 2.5981i
```

EXAMPLE

Write a Matlab function to calculate equavalent value of **parallel** connected resistances.
PS: Resistance values will be entered as avector as below
R=[R1 R2 R3 …….. Rn]

```
function RT = paralelnr(R)
% R: paralel baðlý direnç deðerlerinden oluþan vektör
%    Bu vektörün sayýsal olarak önceden oluþturulmasý
%    gerekir
n=length(R);
RTOP1=0;
for i=1:n
   RTOP1=RTOP1+(1/R(i));
end
RT=1/RTOP1;
```

EXAMPLE

Write a Matlab function to calculate equavalent value of **series** connected resistances.
PS: Resistance values will be entered as avector as below
R=[R1 R2 R3 …….. Rn]

```
function RT = serinr(R)
% R: Seri baðlý direnç deðerlerinden oluþan vektör
%    Bu vektörün sayýsal olarak önceden oluþturulmasý
%    gerekir
n=length(R);
RT1=0;
for i=1:n
   RT1=RT1+R(i);
end
RT=RT1;
```

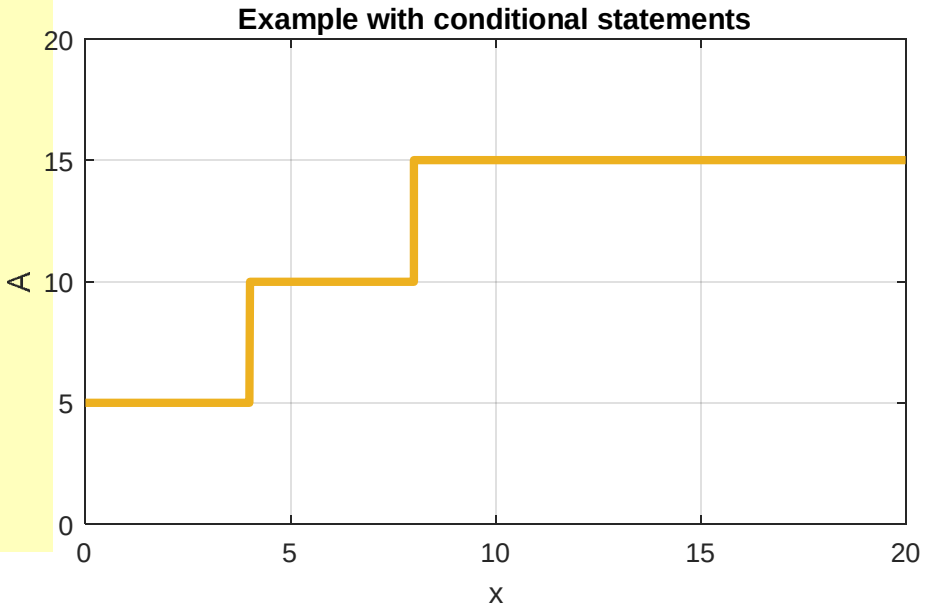
Conditional Statments

```
x=input('Enter x value ==>')

x1=4; x2=8; x3=12;

if x <= x1
       A=5
    elseif x<=x2
        A=10
    elseif x <= x3
        A=20
    else
        A =100
end
```

Conditional Statments

```
x1=4; x2=8; x3=12;
x=0; k=1;
while x <=20
    if x <= x1
       A(k)=5;
    elseif x <= x2
        A(k)=10;
     else
        A(k) =20;
    end
    xx(k)=x;
    x=x+0.01;
    k=k+1;
end
```

```
plot(0,25,xx,A)
title('Example with conditional
statements')
xlabel('x');  ylabel('A'); grid
```



Conditional Statments

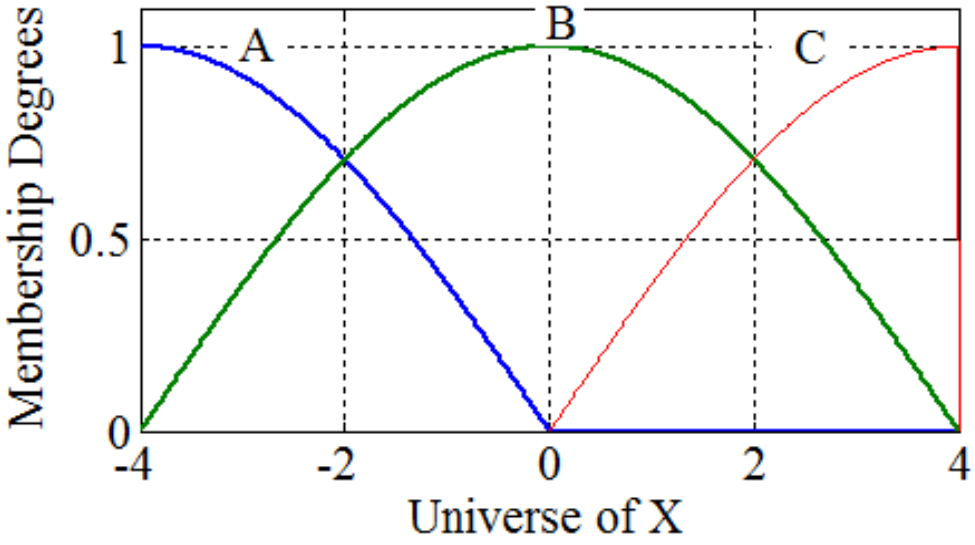
```
%   Inputs  X, n and  x1, x2 and x
clear;  clc;   clf
n=3; x1=-4; x2=-x1; X=x2-x1;
a=X/(n-1); b=(X/2)-a;
for m=1:n
   XP(m)=x1+a*(m-1);
end
XPA=XP(1); XPB=XP(2); XPC=XP(3);
A1=(b*pi)/(2*a);  A2=A1+(pi/2);
k=1;
```

```
function mu=sinus01m(a,A,t)
        T=4*a;
        w1=(2*pi/T);
        mu=abs(sin(w1*t+A));
```

```
for x=x1:0.01:x2;
   if x<XPA
      SA(k)=0; SB(k)=0; SC(k)=0;
    elseif x<XPB
       SA(k)=sinus01m(a,A1,x);
       SB(k)=sinus01m(a,A2,x);
       SC(k)=0;
    elseif x<XPC
       SA(k)=0;
       SB(k)=sinus01m(a,A2,x);
       SC(k)=sinus01m(a,A1,x);
    else
       SA(k)=0; SB(k)=0; SC(k)=0;
    end
    X(k)=x;  k=k+1;
end
```

Conditional Statments

```
plot(X,SA,X,SB,X,SC);
xlabel('Universe of X');
ylabel('Membership Degrees');
grid
```



Matrices and Arrays

Array Creation

To create an array with four elements in a single row, separate the elements with either a comma (,) or a space.

```
a = [1 2 3 4]
```

```
a =
1 2 3 4
```

This type of array is a *row vector*.

Matrices and Arrays

Array Creation

To create a matrix that has multiple rows, separate the rows with semicolons.

```
a = [1 2 3; 4 5 6; 7 8 10]
```

```
a =
1     2     3
4     5     6
7     8    10
```

```
b = 10+a
```

```
b=
11    12    13
14    15    16
17    18    20
```

Another way to create a matrix is to use a function, such as ones, zeros, or rand. For example, create a 5-by-1 column vector of zeros.

```
z = zeros(4,1)
```

```
z =
0
0
0
0
```

Matrices and Arrays

To **transpose** a matrix, use a single quote (’):

```
a =
     1     2     3
     4     5     6
     7     8    10
```

```
c = a’
```

```
>> c
ans =
     1     4     7
     2     5     8
     3     6    10
```

Inverse of a matrix

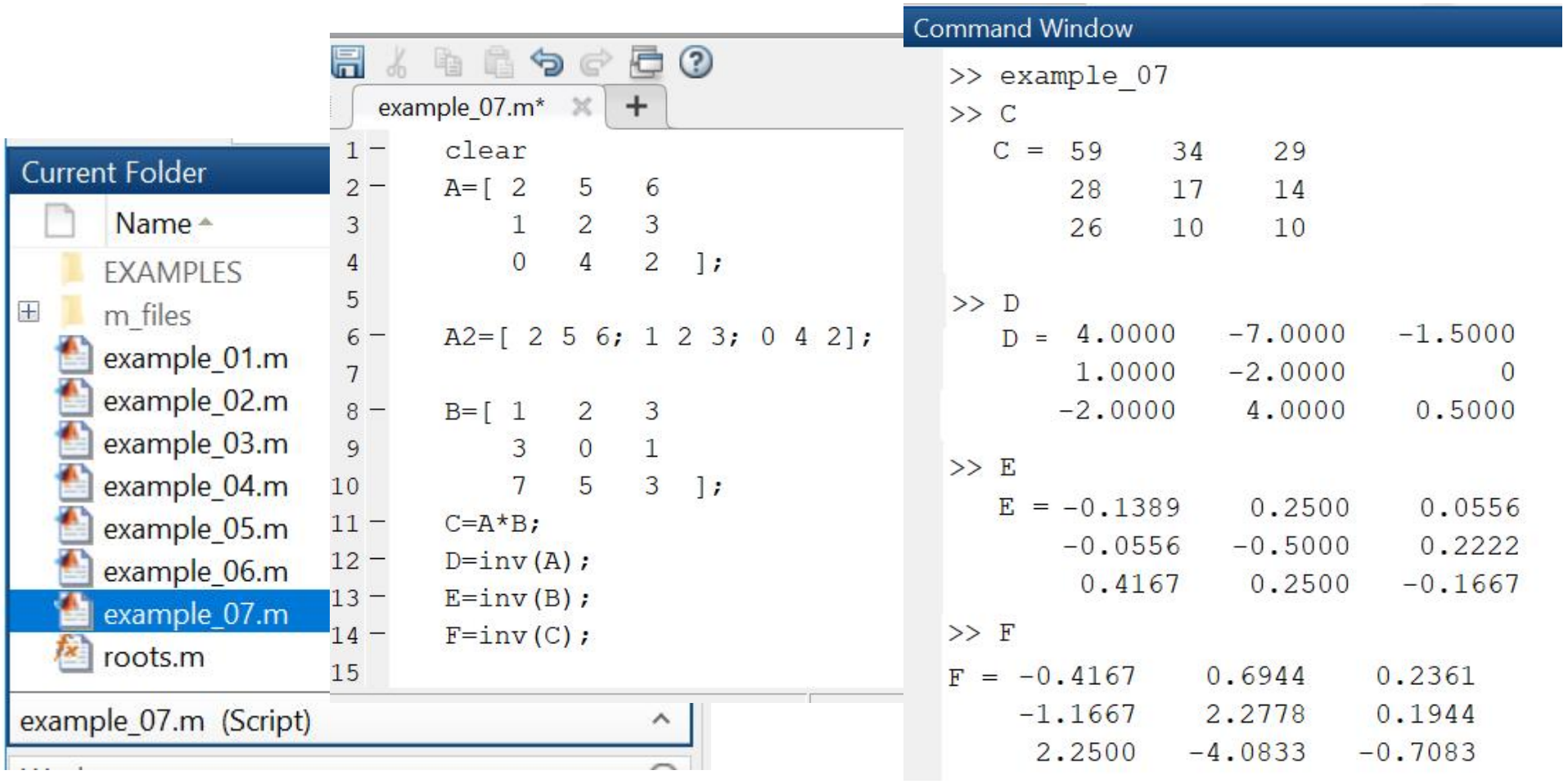
```
d= inv(a)
```

```
d =

   -0.6667   -1.3333    1.0000
   -0.6667    3.6667   -2.0000
    1.0000   -2.0000    1.0000
```

```
E= a*inv(a)
```

```
E =

    1.0000   -0.0000   -0.0000
    0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000
```

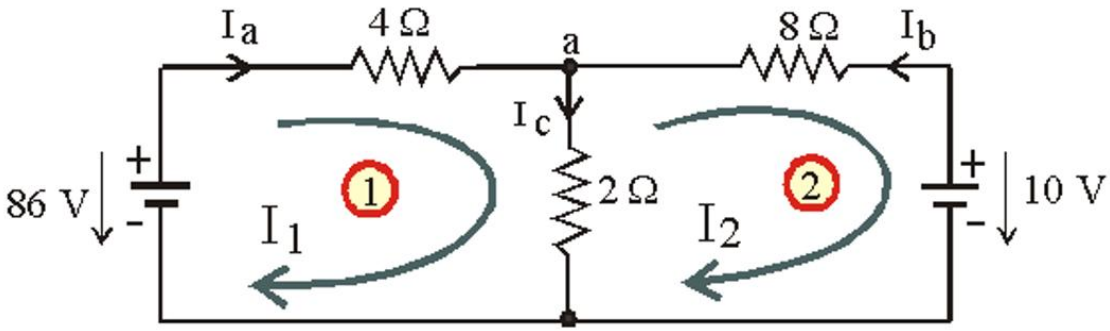
Matrices and Arrays



Matrices and Arrays

EXAMPLE

Use matrix operations in Matlab and dbtain the loop currents in circuit given below



Prepare the formulas from figure

$$\begin{bmatrix} -86 \\ 10 \end{bmatrix} + \begin{bmatrix} 4+2 & -2 \\ -2 & 8+2 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 86 \\ -10 \end{bmatrix} = \begin{bmatrix} 4+2 & -2 \\ -2 & 8+2 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 4+2 & -2 \\ -2 & 8+2 \end{bmatrix}^{-1} \begin{bmatrix} 86 \\ -10 \end{bmatrix}$$

```
E=[86; -10];
R=[6  -2; 2  10];
I=inv(R)*E

I=
15.0000
2.0000
```

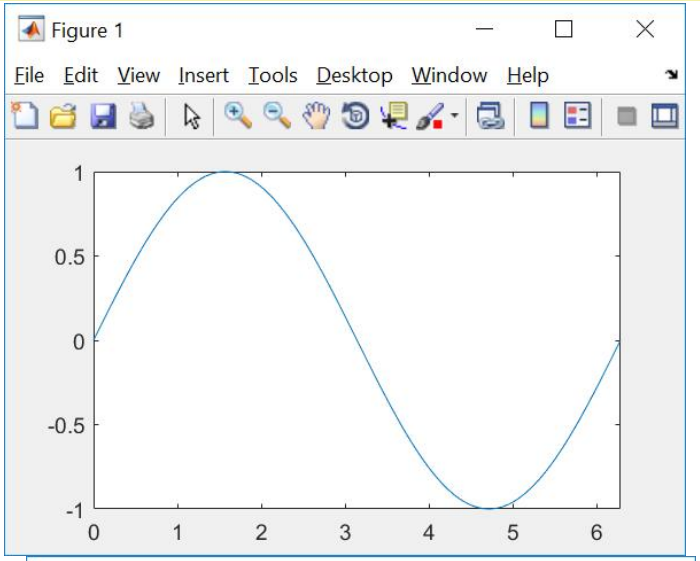
Command Window

```
>> E=[86; -10];
R=[6  -2; 2  10];
I=inv(R)*E

I =

   13.1250
   -3.6250

fx >>
```

Line Plots

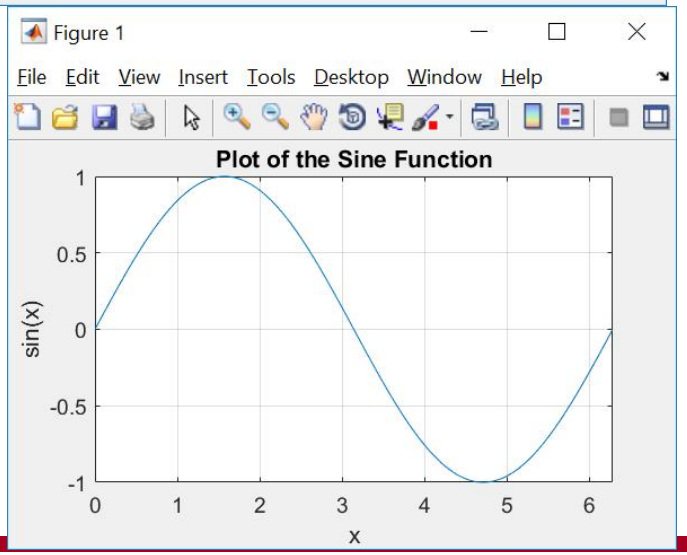
To create two-dimensional line plots, use the plot function. For example, plot the value of the sine function from 0 to 2π:

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)
```



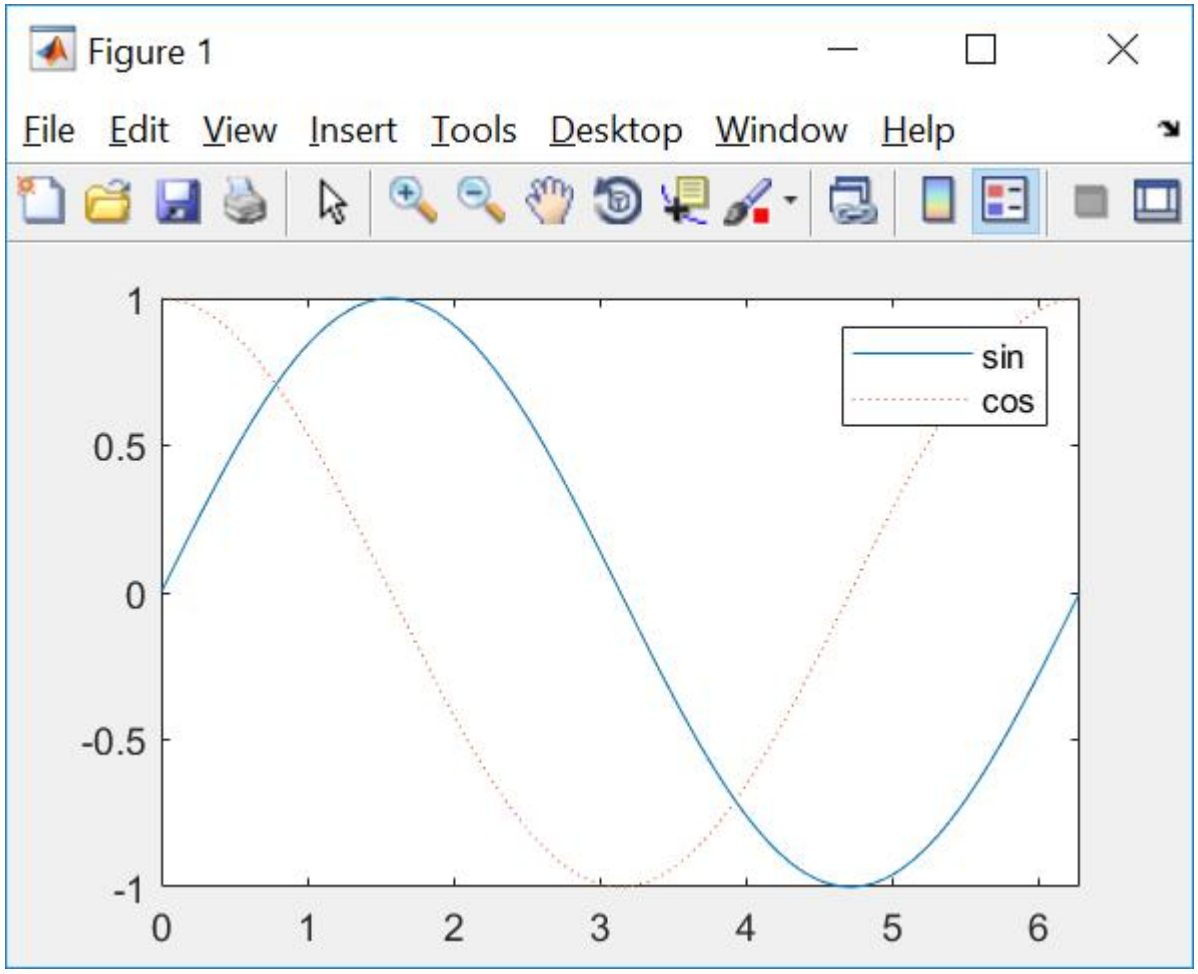
You can label the axes and add a title.

```
xlabel('x')
ylabel('sin(x)')
title('Plot of the Sine Function')
grid
```



Line Plots - Examples

```
x = 0:pi/100:2*pi;
y = sin(x); plot(x,y)

hold on
y2 = cos(x);
plot(x,y2,':')
legend('sin','cos')
hold off
```



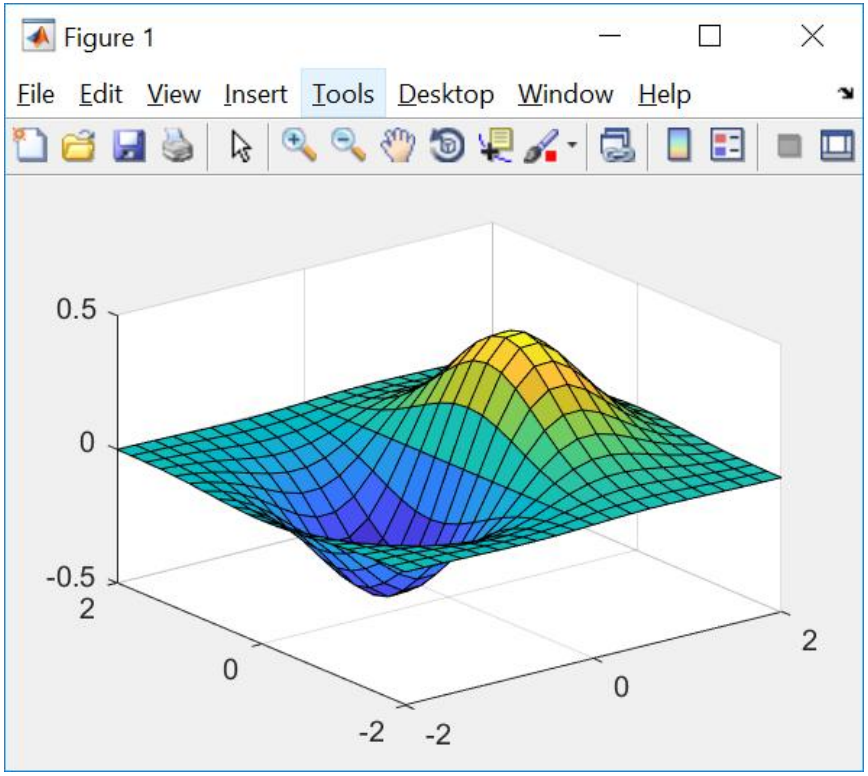
3-D Plots

Three-dimensional plots typically display a surface defined by a function in two variables, ***z = f(x,y)*** .

To evaluate *z*, first create a set of (*x*,*y*) points over the domain of the function using **meshgrid**.

```
[X,Y] = meshgrid(-2:.2:2);
Z = X .* exp(-X.^2 - Y.^2);
```

```
surf(X,Y,Z)
```



Subplots

You can display multiple plots in different subregions of the same window using the subplot function.

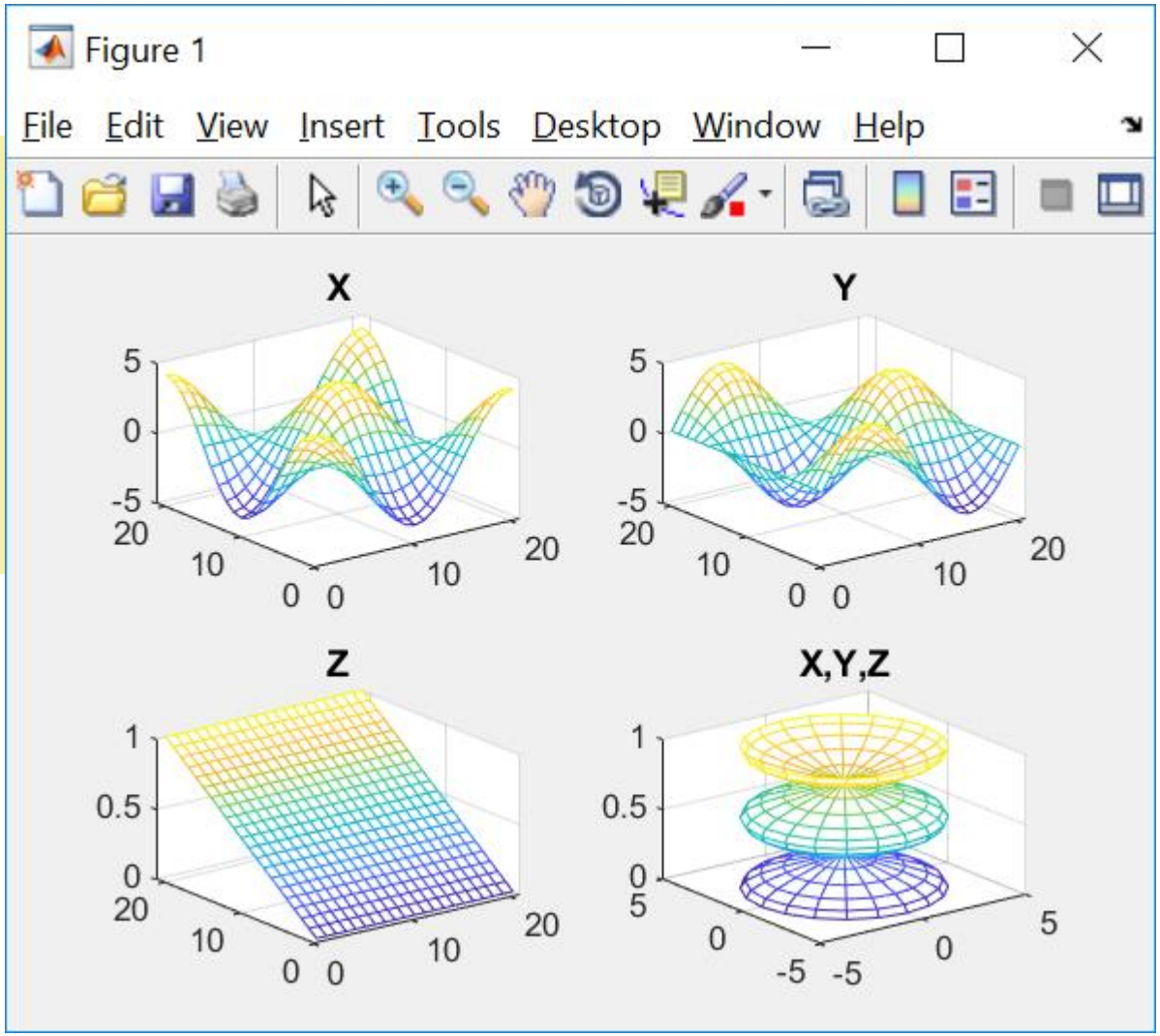
The first two inputs to subplot indicate the number of plots in each row and column. The third input specifies which plot is active. For example, create four plots in a 2-by-2 grid within a figure window.

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));

subplot(2,2,1);   mesh(X);          title('X');
subplot(2,2,2);   mesh(Y);          title('Y');
subplot(2,2,3);   mesh(Z);          title('Z');
subplot(2,2,4);    mesh(X,Y,Z);   title('X,Y,Z');
```

Subplots

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));

subplot(2,2,1);   mesh(X);
subplot(2,2,2);   mesh(Y);
subplot(2,2,3);   mesh(Z);
subplot(2,2,4);    mesh(X,Y,Z);
```

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



NEXT
Simulink Basics