



ME 444

MATLAB® FOR ENGINEERS

Lecturer:

Dr. Nurettin Furkan DOĞAN

Mechanical Engineering Department

4 th Floor Office No: 303

nfdogan@gantep.edu.tr

0342 317 1200- 2569



Office Hours:

Tuesday: 13.30- 15:00

Friday : 10.00- 11.30

CHAPTER 5

PROGRAMMING IN MATLAB-2 **(File management, User-defined Functions)**

- In most cases, the input to a script comes from a data file created by another source.
- It's useful to store the output in an external file that can be processed and/or printed later.
- There are processes on the files:
 - ☐ Read: Reading data from a previously created file
 - ☐ Write: Write (save) data to a file
 - ☐ Add: Adding data to an existing file

File Management: Writing (Saving) Data to a File

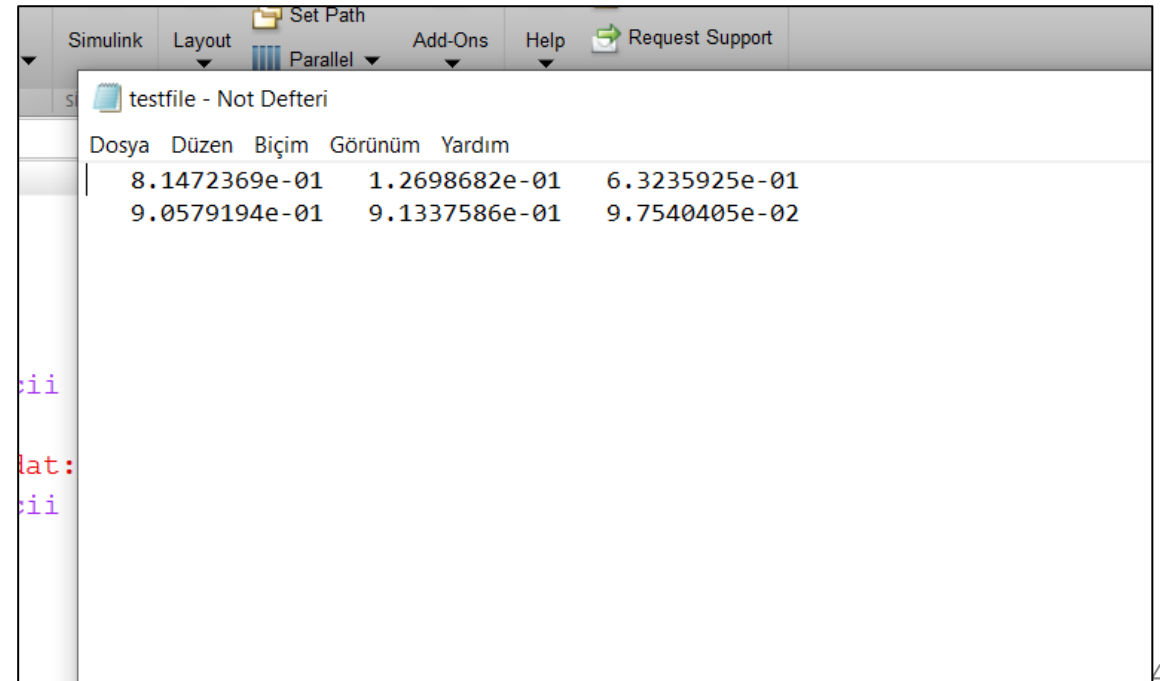
- The **save** command can be used to write or append data from a matrix to a data file.

Format of this command:

```
save filename matrixvariablename -ascii
```

```
>> mymat = rand(2,3)
mymat =
    0.4565    0.8214    0.6154
    0.0185    0.4447    0.7919

>> save testfile.dat mymat -ascii
```



The image shows a MATLAB Command Window titled 'testfile - Not Defteri'. The window displays the contents of the saved file 'testfile.dat' in ASCII format. The data is presented as a 2x3 matrix of floating-point numbers in scientific notation. The window has a menu bar with options: Simulink, Layout, Set Path, Add-Ons, Help, and Request Support. Below the menu bar, there are tabs for 'Dosya', 'Düzen', 'Biçim', 'Görünüm', and 'Yardım'. The main area of the window shows the following data:

Dosya	Düzen	Biçim	Görünüm	Yardım
8.1472369e-01	1.2698682e-01	6.3235925e-01		
9.0579194e-01	9.1337586e-01	9.7540405e-02		

File Management: Writing (Saving) Data to a File

- To append the contents of a matrix variable to an existing file:

```
>> save testfile.dat mat2 -ascii -append
```

```
>> mat2 = rand(3,3)
```

```
mymat =
```

```
    0.9218    0.4057    0.4103
```

```
    0.7382    0.9355    0.8936
```

```
    0.1763    0.9169    0.0579
```

```
>> save testfile.dat mat2 -ascii -append
```

Link Layout Parallel Add-Ons Help Request Support

testfile - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

8.1472369e-01	1.2698682e-01	6.3235925e-01
9.0579194e-01	9.1337586e-01	9.7540405e-02
2.7849822e-01	9.6488854e-01	9.5716695e-01
5.4688152e-01	1.5761308e-01	4.8537565e-01
9.5750684e-01	9.7059278e-01	8.0028047e-01

ci

File Management: Reading Data from a File

- The load command is used to assign the previously saved data to a variable in MATLAB.

```
>> load testfile.dat
```

```
Command Window
>> load testfile.dat
>> who

Your variables are:

testfile

>> testfile
testfile =
    0.8147    0.1270    0.6324
    0.9058    0.9134    0.0975
    0.2785    0.9649    0.9572
    0.5469    0.1576    0.4854
    0.9575    0.9706    0.8003
```

- who:** lists the variable names in the current file.

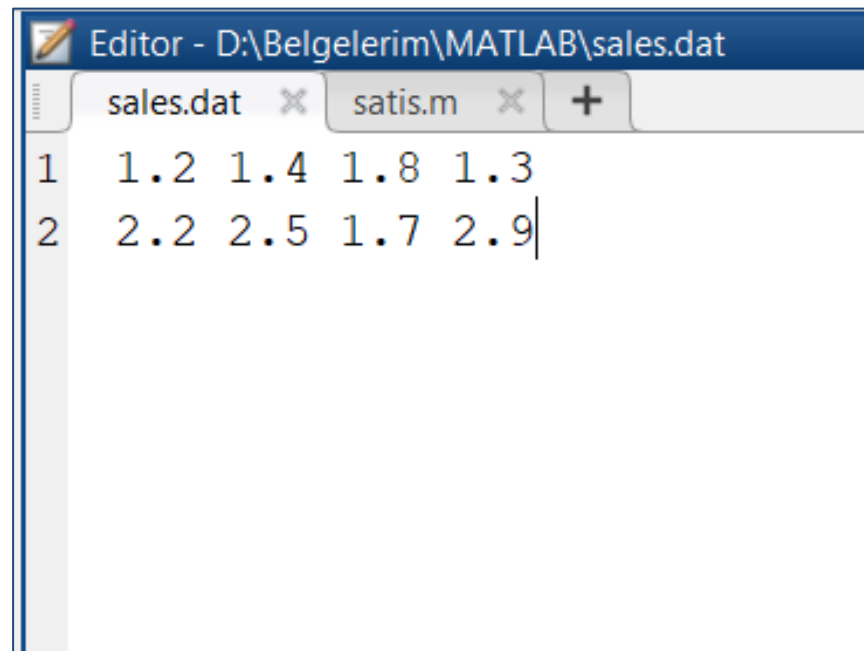
- **Problem:** The sales (in billions of TL) of Company X for each of the four quarters of 2019 are stored in a file named .. salesfigs.dat ..:

```
1.2 1.4 1.8 1.3  
2.2 2.5 1.7 2.9
```

- ✓ First, create this file (just type numbers in Editor and Save As.. sales.dat)
- ✓ Next, load the data from the file into a matrix and split that matrix into 2 vectors.

- **EXAMPLE SOLUTION:**

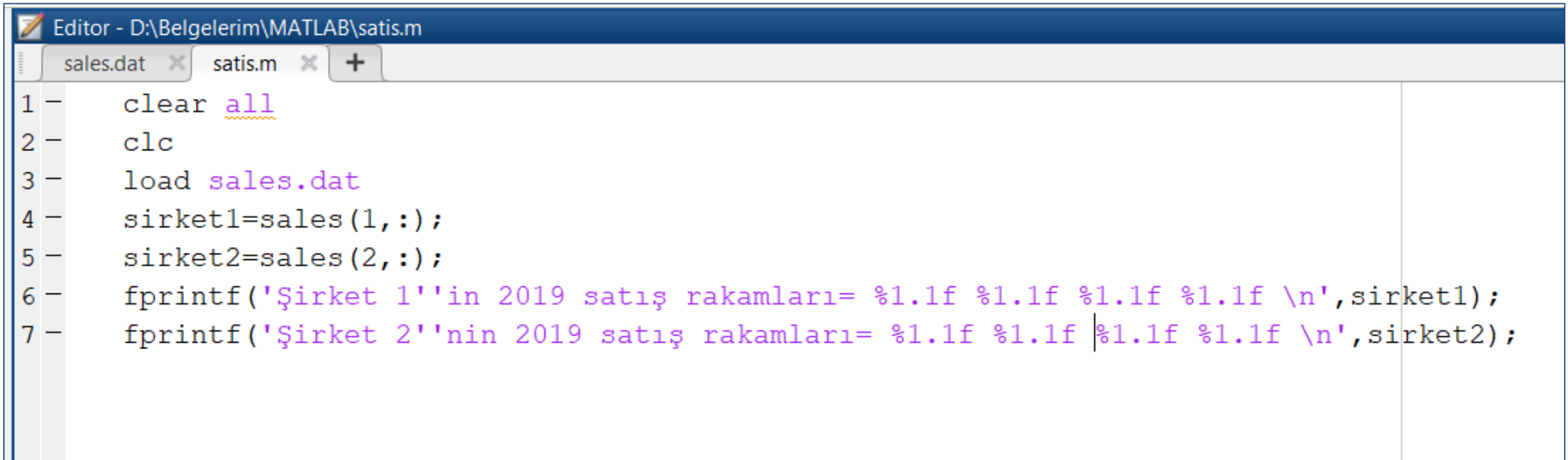
- ✓ First, create this file (just type numbers in Editor and Save As.. sales.dat)

A screenshot of the MATLAB Editor window. The title bar reads 'Editor - D:\Belgelerim\MATLAB\sales.dat'. The window has two tabs: 'sales.dat' and 'satis.m'. The 'sales.dat' tab is active, showing two lines of data: '1 1.2 1.4 1.8 1.3' and '2 2.2 2.5 1.7 2.9'. The cursor is at the end of the second line.

```
1 1.2 1.4 1.8 1.3
2 2.2 2.5 1.7 2.9
```


- **EXAMPLE SOLUTION:**

- ✓ Then, in the script file called sales.m, load the data from the sales.dat file into two separate vectors.

A screenshot of the MATLAB Editor window. The title bar shows 'Editor - D:\Belgelerim\MATLAB\satis.m'. The window has two tabs: 'sales.dat' and 'satis.m'. The 'satis.m' tab is active, showing a script with 7 lines of code. The code is as follows:

```
1 - clear all
2 - clc
3 - load sales.dat
4 - sirket1=sales(1,:);
5 - sirket2=sales(2,:);
6 - fprintf('$irket 1''in 2019 satış rakamları= %1.1f %1.1f %1.1f %1.1f \n',sirket1);
7 - fprintf('$irket 2''nin 2019 satış rakamları= %1.1f %1.1f |%1.1f %1.1f \n',sirket2);
```

- **EXAMPLE SOLUTION:**
- ✓ Command Window image after running the program:

```
Command Window  
Şirket 1'in 2019 satış rakamları= 1.2 1.4 1.8 1.3  
Şirket 2'nin 2019 satış rakamları= 2.2 2.5 1.7 2.9  
fx >>
```

File Management: Reading and Writing Data in MS Excel



- ✓ In MATLAB, data entry to the program can be done through an Excel data file.
- ✓ At the same time, data output from the program can also be done using an Excel file.
- ✓ Detailed information on file input/output operations can be found under the `File I/O` heading in the MATLAB program through the help window.

- ✓ In MATLAB, the `xlsread` command is used to input data from an Excel file to the program.

```
variable_name=xlsread('filename')
```

- ✓ 'filename' is the name of the excel file to be entered. When using this command, the excel file to be entered must be in the same Work Folder as the program we wrote in MATLAB.
- ✓ If the excel file to be entered consists of more than one sheet, the data will be taken from the first sheet.

File Management: Reading and Writing Data in MS Excel

- ✓ If there is more than one sheet in the excel file used and the sheets are specially named, data entry can be made from the desired sheet by specifying the name of the desired sheet.

```
variable_name=xlsread('filename','sheetname')
```

- ✓ Another option is to enter data in the desired row/column range.

```
variable_name=xlsread('filename','sheetname','row/column')
```

- ✓ When writing 'row/column' we need the addressing used to specify the data range in excel. For example; 'C2:E5' contains data from a 4x3 field in excel. That is, it consists of rows 2,3,4, and 5 and the data in columns C,D, and E.

File Management: Reading and Writing Data in MS Excel

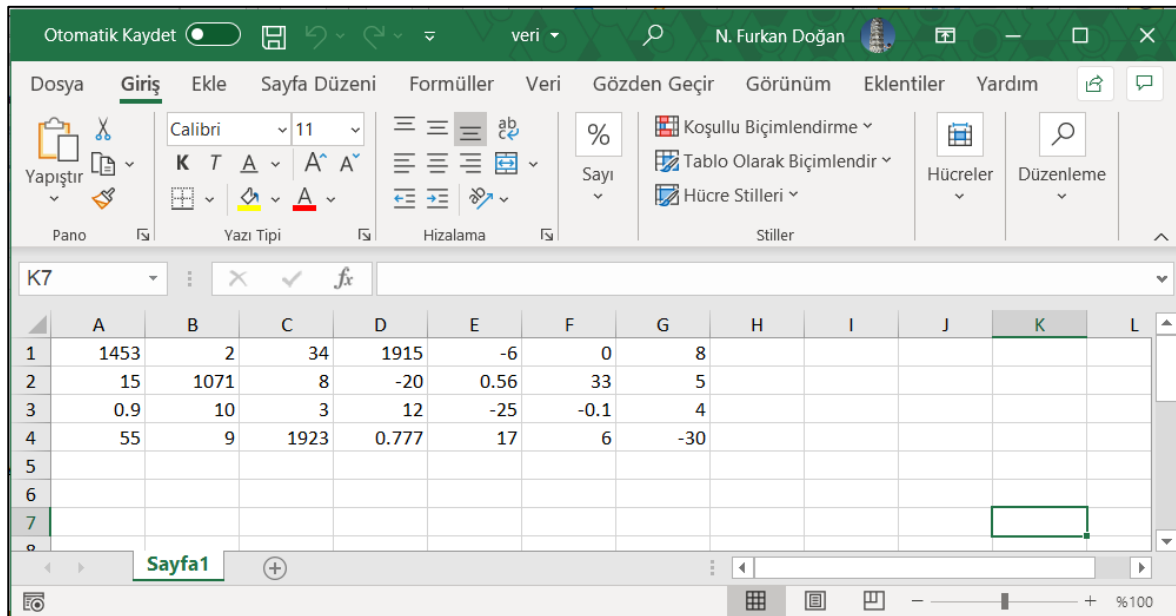
- ✓ To write data to an excel file, the `xlswrite` command is used.

```
xlswrite('filename', variable_name)
```

- ✓ 'filename' is the name of the excel file where the data sent from the program will be written. If there is no file with the specified name in the Work Folder, MATLAB will automatically create an excel file with the specified name.
- ✓ `variable_name` is the name of the variable where the data to be sent is defined.
- ✓ In addition, data can be written to a certain page and data range in the excel file in data writing, as in the data reading process.

File Management: Reading and Writing Data in MS Excel

✓ **Example:** Let's read data from the excel file named data in the picture to MATLAB.



	A	B	C	D	E	F	G	H	I	J	K	L
1	1453	2	34	1915	-6	0	8					
2	15	1071	8	-20	0.56	33	5					
3	0.9	10	3	12	-25	-0.1	4					
4	55	9	1923	0.777	17	6	-30					
5												
6												
7												

```
>> data=xlsread('veri')
```

```
data =
```

```
Columns 1 through 6
```

```
Column 7
```

1453	2	34	1915	-6	0	8
15	1071	8	-20	0.56	33	5
0.9	10	3	12	-25	-0.1	4
55	9	1923	0.777	17	6	-30

Otomatik Kaydet ☒     matrisdosya...  N. Furkan Doğan

Dosya **Giriş** Ekle Sayfa Düzeni Formüller Veri Gözden Geçir Görünüm Ek

- ✓ There are several types; For now we will focus only on the type of function that calculates and returns a value.
- ✓ We give a name to the function we wrote to perform a certain action and save it as .m.
- ✓ To use the function later, just like using a built-in function embedded in MATLAB: the function name is written, the necessary data is sent to the function by writing in parantheses, and after the calculations are made in the function, the result of the operation returns to the place where the function was called.

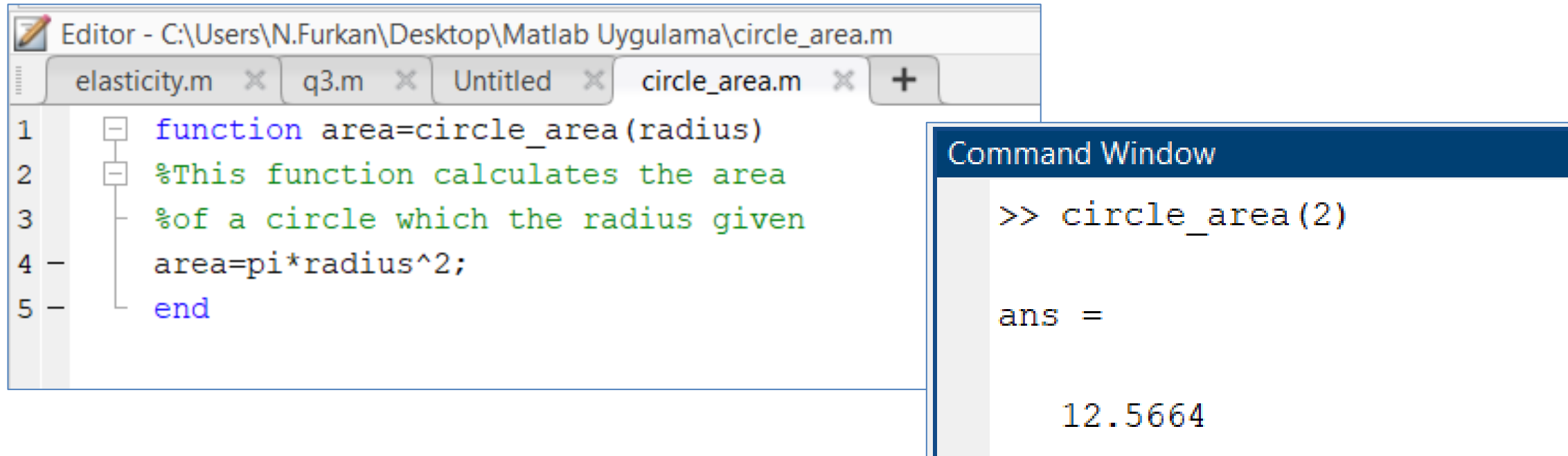
- ✓ When defining the function, it should look like this:

```
function output_data=function_name(input_data)
%comments for the user
computation1;
computation2;
computation3;
output_data=data_value;
end
```

User-Defined Functions

Example: Let's calculate the area of the circle and write a sending function.

- ✓ One input argument required: radius
- ✓ There is one output argument: area



The image shows a MATLAB Editor window with the file `circle_area.m` open. The code defines a function `area=circle_area(radius)` that calculates the area of a circle. The Command Window shows the function being called with `circle_area(2)`, resulting in the output `ans = 12.5664`.

```
Editor - C:\Users\N.Furkan\Desktop\Matlab Uygulama\circle_area.m
elasticity.m x q3.m x Untitled x circle_area.m x +
1 function area=circle_area(radius)
2 %This function calculates the area
3 %of a circle which the radius given
4 area=pi*radius^2;
5 end

Command Window
>> circle_area(2)

ans =

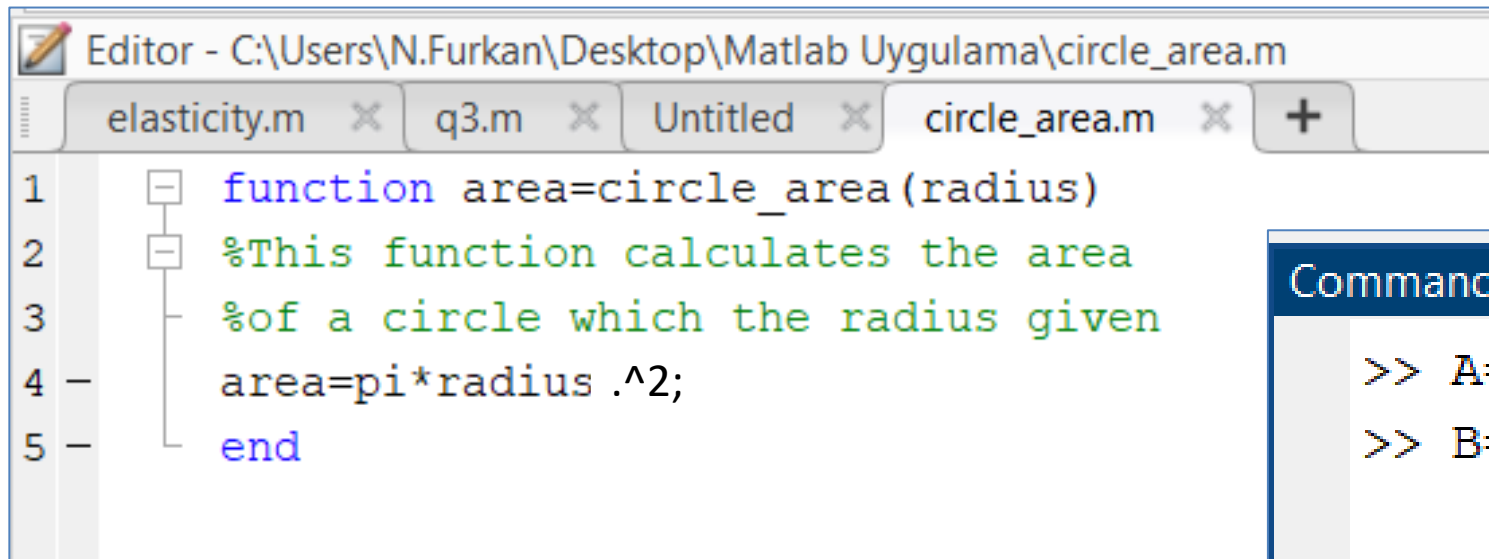
12.5664
```

- When the function is called from the command window, it performs the operation using the input argument and sent the output argument.

User-Defined Functions

- To pass vector elements to the function: the `.^` operator for exponentiation of the elements should be used instead of the `^` exponentiation operator in

```
area=pi*radius.^2;
```



```
Editor - C:\Users\N.Furkan\Desktop\Matlab Uygulama\circle_area.m
elasticity.m x q3.m x Untitled x circle_area.m x +
1 function area=circle_area(radius)
2 %This function calculates the area
3 %of a circle which the radius given
4 area=pi*radius.^2;
5 end
```

Command Window

```
>> A=[1 2 3];
>> B=circle_area(A)
```

```
B =
```

```
3.1416    12.5664    28.2743
```

NOTES ON FUNCTIONS

- You can send multiple input data to the function.
- The variables used inside the function are called "local variables". Local variables can be defined in the function as needed.
- Notice that normally the outputs of user-defined functions are not printed to the screen (it does not contain `disp` or `fprintf!`). It needs a MATLAB program (script code) for printing to the screen.

NOTES ON FUNCTIONS

- Therefore, written MATLAB functions can be called with script codes written in the Editor window.
- The combination of user-written functions with the command lines typed in the editor is called a **program**. From now on, what is meant by the word program is the set of command lines written and saved in the Editor.

General Structure of a Simple Program



script.m

- Read input values
- Call function (function_name) to calculate the result
- Print the result to the screen (or file).

fonk_adi.m

```
function output= func_name(input)  
output = calculated value acc.to input  
end
```

EXAMPLE:

The volume of a hollow sphere is found by $\frac{4}{3}\pi(r_o^3 - r_i^3)$. We want to write a program that will ask the user for radii, call a function to calculate the volume, and print the result to the screen.

MATLAB PROGRAMS

EXAMPLE SOLUTION:

NOTE: If more than one input is sent to the function, the order of the inputs is important. The first input sent is assigned to the first input in the function, the second to the second... etc.

```
Editor - C:\Users\N.Furkan\Desktop\Matlab Uygulama\cylinder.m
cylinder.m x cylinderv.m x +
1 %This program calculates the volume of a hollow cylinder
2 ri=input('Enter the inner radius of the cylinder: ');
3 ro=input('Enter the outer radius of the cylinder: ');
4 volume=cylinderv(ri,ro);
5 fprintf('The volume of the cylinder with wall thickness of %3.1f cm ',ro-ri);
6 fprintf('is %5.2f cm^3. \n',volume);
```

```
Editor - C:\Users\N.Furkan\Desktop\Matlab Uygulam
cylinder.m x cylinderv.m x +
1 function v=cylinderv(x,y)
2 v=4/3*pi*(y^3-x^3);
3 end
```

Command Window

```
>> cylinder
Enter the inner radius of the cylinder: 1
Enter the outer radius of the cylinder: 2
The volume of the cylinder with wall thickness of 1.0 cm is 29.32 cm^3.
```

fx >> |

EXAMPLE:

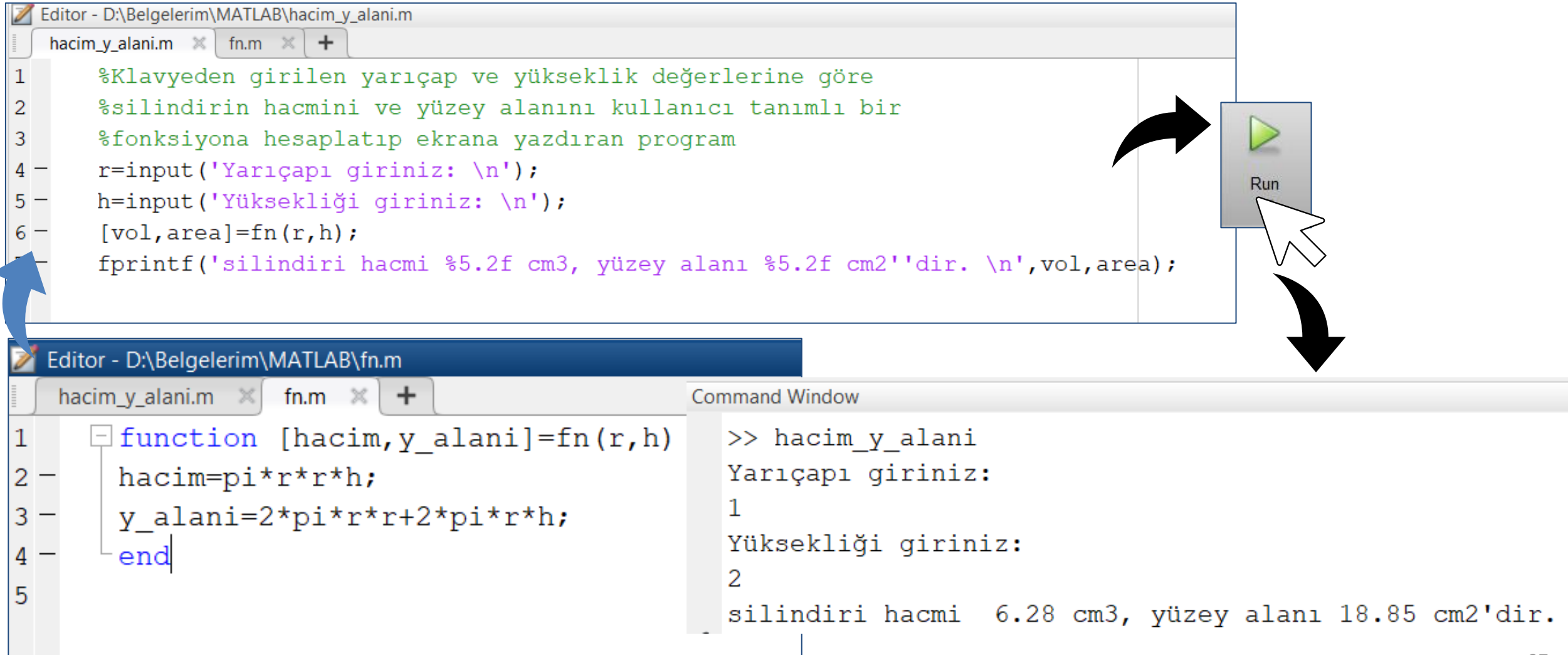
Write a MATLAB program that uses a function called cylinder to calculate the volume and surface area of a cylinder. Ask the user to enter the height and radius of the cylinder from the keyboard.

SOLUTION STEPS:

1. Prepare a script (cylinder.m) that asks the user for the base radius and height of the cylinder, then sends these values as input to the function we will write.
2. Write a function (func1.m) that will calculate the volume and surface area of the cylinder using the input values from the main program and send it as output to the main program.
3. Print the output values from the function in the main program separately.

MATLAB PROGRAMS

EXAMPLE SOLUTION:



The image shows a MATLAB environment with two windows. The top window, titled 'Editor - D:\Belgelerim\MATLAB\hacim_y_alani.m', contains a script that prompts the user for the radius and height of a cylinder, calls a function 'fn' to calculate the volume and surface area, and prints the results. The bottom window, titled 'Editor - D:\Belgelerim\MATLAB\fn.m', contains the function definition for 'fn'. To the right of the top window is a 'Run' button with a green play icon. A mouse cursor is pointing at the 'Run' button, and a large black arrow points from the 'Run' button to the Command Window. A blue arrow points from the first line of the script in the top window to the Command Window.

```
Editor - D:\Belgelerim\MATLAB\hacim_y_alani.m
hacim_y_alani.m x fn.m +
1 %Klavyeden girilen yarıçap ve yükseklik değerlerine göre
2 %silindirin hacmini ve yüzey alanını kullanıcı tanımlı bir
3 %fonksiyona hesaplatıp ekrana yazdıran program
4 r=input('Yarıçapı giriniz: \n');
5 h=input('Yüksekliği giriniz: \n');
6 [vol,area]=fn(r,h);
7 fprintf('silindiri hacmi %5.2f cm3, yüzey alanı %5.2f cm2''dir. \n',vol,area);

Editor - D:\Belgelerim\MATLAB\fn.m
hacim_y_alani.m x fn.m +
1 function [hacim,y_alani]=fn(r,h)
2 hacim=pi*r*r*h;
3 y_alani=2*pi*r*r+2*pi*r*h;
4 end
5

Command Window
>> hacim_y_alani
Yarıçapı giriniz:
1
Yüksekliği giriniz:
2
silindiri hacmi 6.28 cm3, yüzey alanı 18.85 cm2'dir.
```

Next week

Chapter 6

DECISION MAKING