Introduction to Engineering MATLAB – 5 & 6 Script Files - 1

Agenda
- Script files

1

SCRIPT FILE

- A script file is a sequence of MATLAB commands, called a program.
- The file can be edited and saved.
- When the file is run, MATLAB executes the commands in the order they are written just as if they were typed in the Command Window.
- Using a script file is convenient since it can be executed many times without the need to retype the commands (as is needed when working in the Command Window).
- Script files are also called M-files because the extension .m is used when they are saved.

2

In-Class Exercise

- Give an everyday example of when someone would use a script file.
- Example: Calculating the class average on an exam.

3

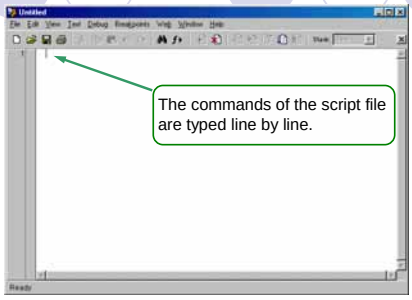
CREATING A SCRIPT FILE

In the command window click on the **File** menu, select **New**, and then select **M-file**.

Once **M-file** is selected, the M-file Editor/Debugger window opens.

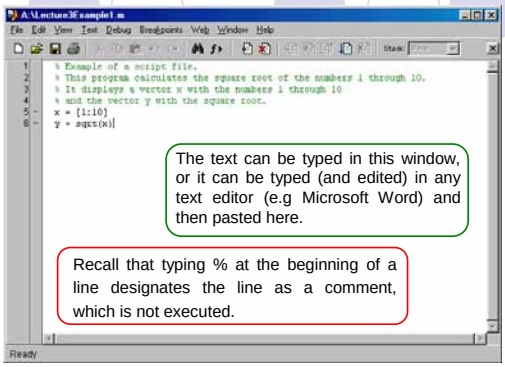
4

The M-file Editor/Debugger window



5

EXAMPLE OF A SCRIPT FILE



6

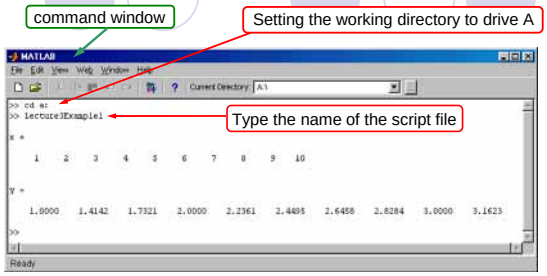
SAVING A SCRIPT FILE

- Once the script file is completed, it must be saved. In our class use **Save As** and save in your J drive.
- The name of the script file follows the rules for names of variables in MATLAB. (Must begin with a letter, can include digits and underscore, up to 31 characters long, don't give the file a name of a variable that is used, or a predefined variable, don't use a name of a MATLAB command or a function.)
- There is no need to add ".m" at the end of a file name. The software will automatically add it for you.

RUNNING A SCRIPT FILE

- A script file is run from the command window.
- To run a script file that is saved in drive J, the MATLAB search path has to be modified to include drive J, or the working directory has to be changed to drive J.
- To change the working directory to drive J type (in the command window):
  cd j:
- To run a script file type the name of the file (without the extension .m) in the command window.
- If saved in a directory within the J drive, type the directory name as well, such as **cd j:\Engineering\Matlab**
  - You can also change the directory using the typical Windows style "browser" at the top of the Matlab window.

EXAMPLE OF RUNNING A SCRIPT FILE



The output that is generated when the script file runs is printed in the command window.

GLOBAL VARIABLES

- Global variables are variables that, once created in one part of MATLAB, are recognized in other parts of MATLAB.
- Variables that are created in the command window are recognized and can be used in a script file.
- Variables that are created in a script file are recognized and valid in the command window.

INPUT TO A SCRIPT FILE

A script file is a program that can be executed with different values of its variables. This can be done in three different ways depending on where and how the variables are defined:

1. The variable is defined in the script file. To run the script file with different variable value, the file is edited and the value of the variable is changed. Then the file is saved, and executed.
2. The variable is defined in the command window. To run the script file with a different value, a new value is given to the variable in the command window. Then the script file is executed.

INPUT TO A SCRIPT FILE

3. The variable is defined in the script file without a specific value. When the script file runs the user is prompted to enter a value from the command window.

   This is done by using the **input** statement:

   x = input('text')

   For example: (string)

   x = input('Please enter a value for x')

Once a number (or a vector, or a matrix) is entered, x has this value.

If you want to enter a *string* for x (instead of a numerical value), you must denote that x is a string variable, such as:

x = input('Enter the material name' , 's')

OUTPUT FROM A SCRIPT FILE

- When a script file runs, output that is generated is displayed in the command window.
- Output is displayed automatically if a statement does not end with a semicolon.
- Output can also be displayed intentionally by using the **disp** command.

The disp COMMAND

disp(A) Displays the content, but not the name, of the variable A.

disp('text') Displays the text (string) that is enclosed within the single quotes.

string

EXAMPLE OF A SCRIPT FILE THAT USES THE input AND disp COMMANDS

```
1   %Example of input to, and output from a script file.
2   %
3   disp 'First Last CID'
4   disp 'Matlab 5, In-Class Example'
5   disp ' '
6   %
7   test1=input('Enter the score of the first test: ');
8   test2=input('Enter the score of the second test: ');
9   test3=input('Enter the score of the third test: ');
10  average_grade=(test1+test2+test3)/3;
11  disp('The average grade is: ')
12  disp(average_grade)
```


RUNNING THE SCRIPT FILE WITH THE input AND disp COMMANDS IN THE COMMAND WINDOW

```
>> example2
First Last CID
Matlab 5, In-Class Example

Enter the score of the first test: 90
Enter the score of the second test: 85
Enter the score of the third test: 92
The average grade is:
    89
```

The grades are entered following the prompt.

CREATE AND DISPLAY A TABLE

```
1   %This script file shows how to create and display a table.
2   %The table includes the population data from Matlab 2.
3   %
4   disp 'First Last CID'
5   disp 'Matlab 5, In-Class Example'
6   disp ' '
7   %Creating a vector of year numbers.
8   yr=[1984 1986 1988 1990 1992 1994 1996];
9   %Creating a vector of population data.
10  pop=[127 130 136 145 158 178 211];
11  %
12  %Substituting the yr vector in the first column of the table matrix.
13  table_yr_pop(:, 1)=yr';
14  %
15  %Substituting the pop vector in the second column of the table matrix.
16  table_yr_pop(:, 2)=pop';
17  %
18  disp('        YEAR      POPULATION (millions)')   %Display the titles.
19  disp('  ')  %Display an empty line.
20  disp(table_yr_pop)    %Display the table.
```


CREATE AND DISPLAY A TABLE

Executing the script file from the previous slide in the command window gives:

```
>> example3
First Last CID
Matlab 5, In-Class Example

        YEAR      POPULATION (millions)

        1984         127
        1986         130
        1988         136
        1990         145
        1992         158
        1994         178
        1996         211
```

In-Class Assignment

- Copy the script file on slide #17 and be sure you can generate the table in slide #18.
- Create a script file that will generate the table to the right.

Time	Temp (°F)
8	36
9	36
10	37
11	38
12	40
1	42
2	44

19

Assignment 5…

- For each problem write a script file and execute it in the command window.
  - For each problem, the first two lines of the script file are:
    - disp 'First Last, CID'
    - disp 'MATLAB 5, Problem Number Page Number'
    - disp ' '
- Submit a printout of the script file, and a printout of the command window. Print each problem separately, and staple together.
  - Problem 32 page 61 in the textbook.
  - Problem 33 page 61 in the textbook.

20

Assignment 6

- For each problem write a script file and execute it in the command window.
  - For each problem, the first two lines of the script file are:
    - disp 'First Last, CID'
    - disp 'MATLAB 5, Problem Number Page Number'
    - disp ' '
- Submit a printout of the script file, and a printout of the command window. Print each problem separately, and staple together.
  - Problem 4 page 161 in the textbook.

21

…Assignment 6 Continued

- Problem 2: Recall the beam bending lab (Bike Lab 2). Write a script file called beamdeflect that prompts the user to enter in the following:
- Beam Material (Note: this will be a string variable)
  - Use the format **input ('Enter beam material: ','s')** to indicate that the input will be in the form of a *string* variable.
- Beam Type (i.e. Steel rectangular, Aluminum rectangular, Aluminum box, etc.) (Note: this is also a string variable)
- Modulus of Elasticity, E, for the material with units of (psi).
  - *Remember:*
    - *Steel: 29*106*
    - *Aluminum: 10.1*106*
    - *Copper: 17*106*
    - *Titanium: 16.5*106*
- The cross section moment of inertia, I, with units of in4
  - Remember that for a rectangular cross-sectional beam: The distances L and s which are the distances in inches from the beam clamped end to the point of load application and dial indicator respectively. *(In Bike Lab 2, these were about 12.5 in and 11.5 in respectively.)*
- Your script file should then calculate the theoretical beam deflection using the formula:

$$\delta = \frac{Fs^2(3L - s)}{6EI}$$

- You should evaluate the deflection for the same loads that you did during lab. These were 0, 2.5, 5, 7.5, 10, 12.5 pound respectively.
- Your program should display the beam material, the beam type, the applied load, and the resulting deflection. Remember, when you display something you should always include the units! Run the program for the steel rectangular bar used in the Beam Bending Lab. Turn in a copy of the script file along with the printout of the command window.
- Make sure that you SAVE your script file. You will need it again for more MATLAB Assignments.

22