SVKM'S NMIMS Deemed to be UNIVERSITY | NAVI MUMBAI

MATLAB
Unit 5-Lecture 15

BTech (CSBS) -Semester VII

6 September 2022, 09:35AM

*few figures/content have been prepared/referred from internet source/books

Introduction to programming

1) Introduction,
2) M-File Scripts,
3) script side-effects,
4) M-File functions,
5) anatomy of a M- File function,
6) input and output arguments,
7) input to a script file,
8) output commands.

Algorithm

- An **algorithm** is a sequence of steps needed to solve a problem.
- In a **modular** approach, problem is broken into seperate steps and then it is refined until results are manageble.
- The basic algorithm involves 3 steps:
  - Get the input: eg. the radius
  - Calculate the result: eg. the area
  - Display the output

This is known as ***top to down*** design

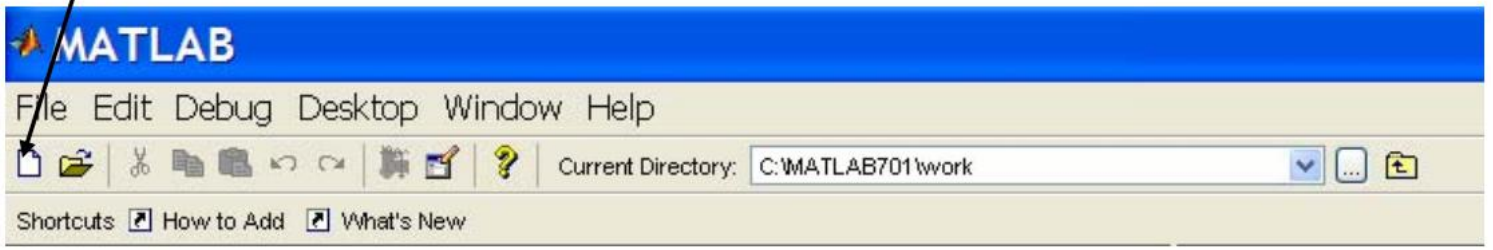
MATLAB script

- Scripts are
  - collection of commands executed in sequence
  - written in the MATLAB editor
  - saved as MATLAB files (.m extension)

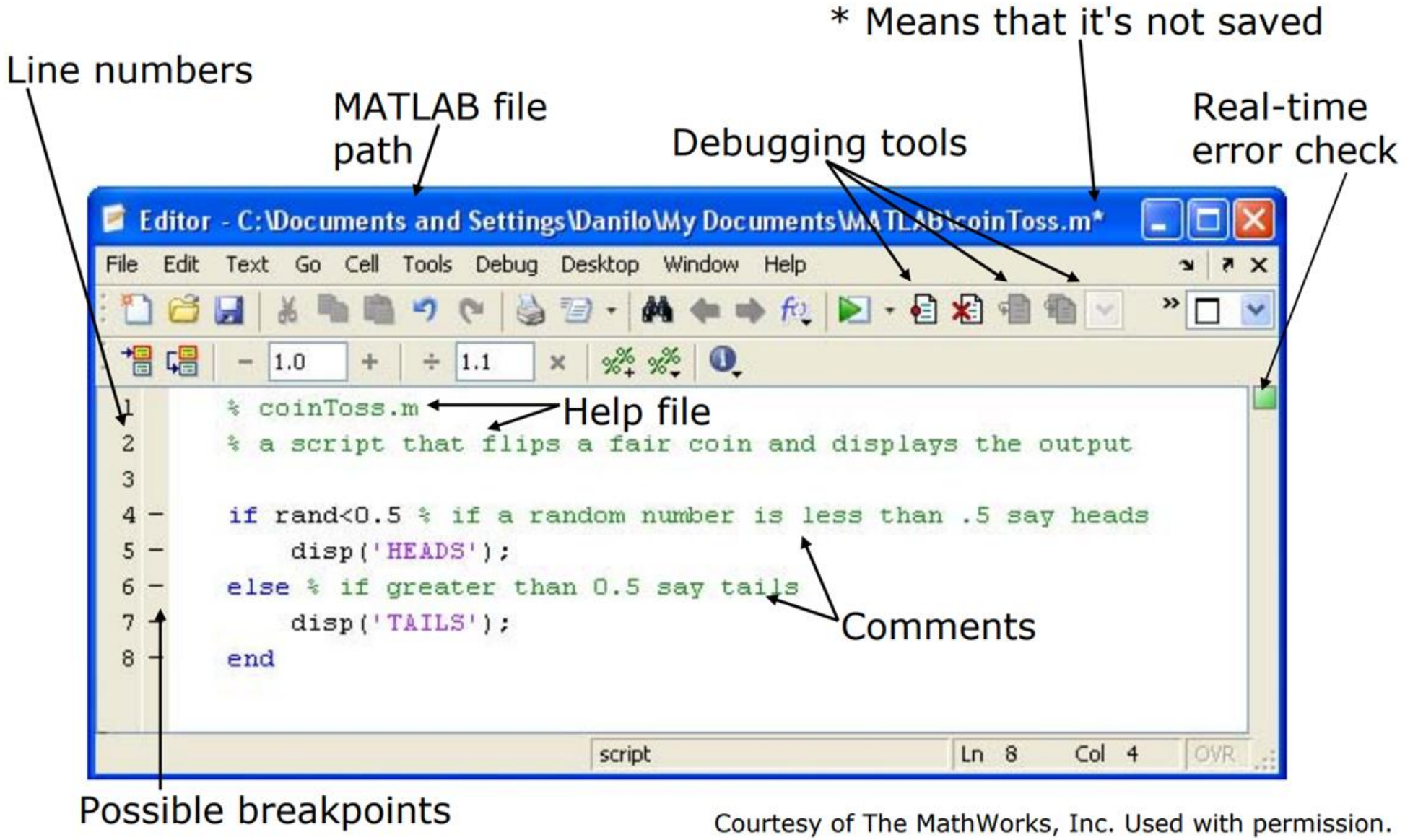
- To create an MATLAB file from command-line

```
» edit helloWorld.m
```

- or click



Script-editor



Courtesy of The MathWorks, Inc. Used with permission.

Script-editor

- **COMMENT!**
  - Anything following a **%** is seen as a comment
  - The first contiguous comment becomes the script's help file
  - Comment thoroughly to avoid wasting time later
- Note that scripts are somewhat static, since there is no input and no explicit output
- All variables created and modified in a script exist in the workspace even after it has stopped running

Script-excercise

```
% This is a MATLAB script file.
% It has been saved as “g13.m".
load g13.dat;                          %Load data file
voltage = g13( : , 4);                 %Extract volts vector
time = .005*[1:length(voltage)];       %Create time vector
plot (time, voltage)                   %Plot volts vs time
xlabel ('Time in Seconds')             % Label x axis
ylabel ('Voltage')                     % Label y axis
title ('Bike Strain Gage Voltage vs Time')
grid                                   %Put a grid on graph
```

Script-Question

Write a script to calculate the circumference of circle ($C=2\pi r$).
Comment the script.

Documentation

circlescript.m

```
% This script calculates the area of a circle

% First the radius is assigned
radius = 5
% The area is calculated based on the radius
area = pi * (radius ^2)
```

```
>> help circlescript
  This script calculates the area of a circle
```

The first comment at the beginning of the script describes what the script does; this is sometimes called a ***block comment***. Then, throughout the script, comments describe different parts of the script (not usually a comment for every line, however!). Comments don't affect what a script does, so the output from this script would be the same as for the previous version.

Example of a script file

Let us write a script file to solve the following system of linear equations1 :

$$\begin{bmatrix} 5 & 2r & r \\ 3 & 6 & 2r-1 \\ 2 & r-1 & 3r \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 3 \\ 5 \end{Bmatrix}$$

or Ax = b. Clearly, A depends on the parameter r. We want to find the solution of the equation for various values of the parameter r. We also want to find, say, the determinant of matrix A in each case.

Example of a script file

```
%---------- This is the script file 'solvex.m' ------------
% It solves equation (4.1) for x and also calculates det(A).

A = [5 2*r r; 3 6 2*r-1; 2 r-1 3*r];  % create matrix A
b = [2;3;5];                          % create vector b
det_A = det(A)                        % find the determinant
x = A\b                               % find x
```

Example of a script file

Let us now execute the script in MATLAB.

```
>> clear all                    % clear the workspace
>> r = 1;                       % specify a value of r
>> solvex                       % execute the script file solvex.m

det_A =

        64

x =
    -0.0312
     0.2344
     1.6875
>> who
```

This is the output. The values of the variables *det_A* and *x* appear on the screen because there is no semi-colon at the end of the corresponding lines in the script file.

Check the variables in the workspace.

Precautions

- NE VER name a script file the same as the name o.f a variable it computes.
- The name of a script file must begin with a letter. The rest of the characters may include digits and the underscore character.
- You may give long names but MATLAB will take only the first 19 character.

eg. proj ecL23C.m, cee213_hw5_1 .m but proj ect.23C.m and cee2 13_hw5.1.m are not valid names.

Function Files

A function file is also an M-file, like a script file, except that the variables in a function file are all local .A function file begins with a function definition line, which has a well-defined list of inputs and outputs. Without this line, the file becomes a script file. The syntax of the function definition line is as follows:

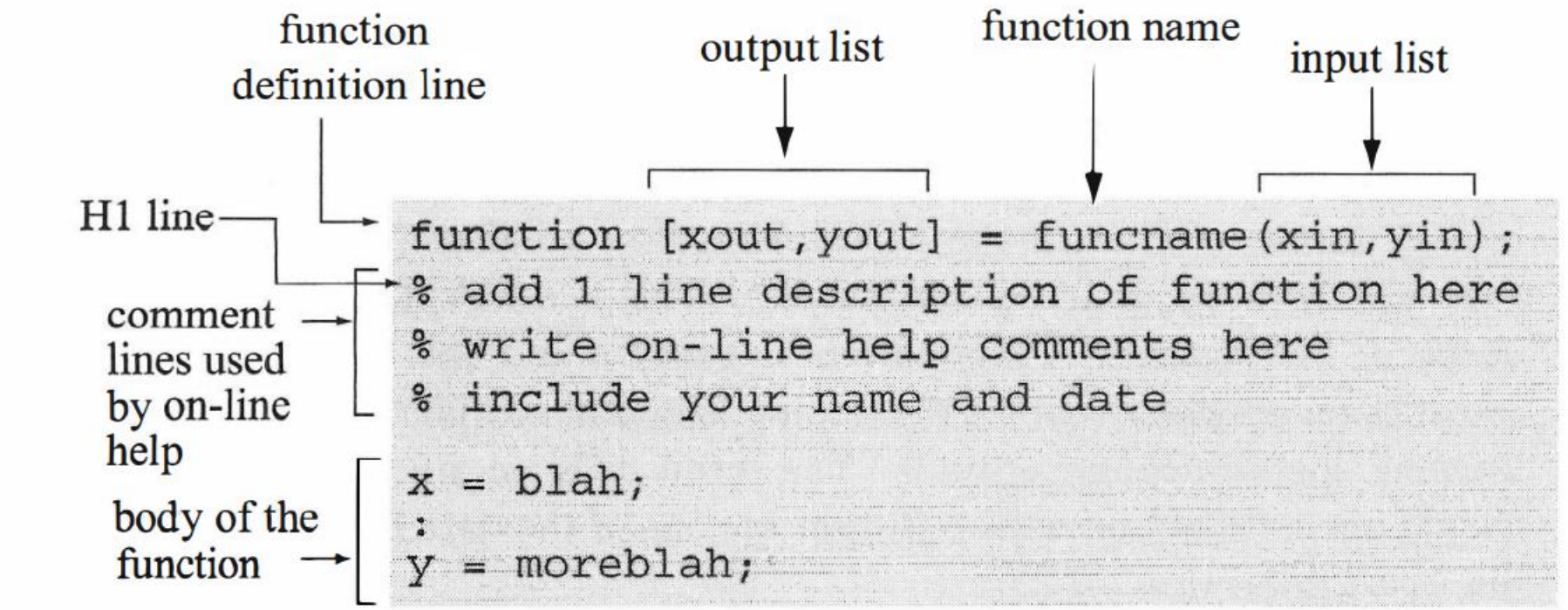
```
function [output variables] = function_name(input variables);
```

Examples of Function Files

Function Definition Line	*File Name*
`function [rho,H,F] = motion(x,y,t);`	motion.m
`function [theta] = angleTH(x,y);`	angleTH.m
`function theta = THETA(x,y,z);`	THETA.m
`function [] = circle(r);`	circle.m
`function circle(r);`	circle.m

Caution: The first word in the function definition line, `function`, *must be typed in lowercase.* A common mistake is to type it as `Function`.

Anatomy of Function Files



Executing Function Files

This is the full syntax of calling a function. Both the output and input list are specified in the call. For example, if the function definition line of a function reads

```
function  [rho ,H,F]   =   mot ion (x ,y,t) ;
```

then all the following commands represent legal call (execution) statements:

Executing Function Files

- `[r,angmom,force]=motion(xt,yt,time);` The input variables *xt*, *yt*, and *time* must be defined before executing this command.
- `[r,h,f]=motion(rx,ry,[0:100]);` The input variables *rx* and *ry* must be defined beforehand; the third input variable *t* is specified in the call statement.
- `[r,h,f]=motion(2,3.5,0.001);` All input values are specified in the call statement.
- `[radius,h]=motion(rx,ry);` Call with partial list of input and output. The third input variable must be assigned a default value inside the function if it is required in calculations. The output corresponds to the first two elements of the output list of `motion`.

Example

```
>> clear all
>> [detA, y] = solvexf(1);   % take r=1 and execute solvexf.m
>> detA                      % display the value of detA
detA =
     64
>> y                         % display the value of y
y =
   -0.0312
    0.2344
    1.6875
>> who
Your variables are:
detA  y
```

Values of *detA* and *y* will be automatically displayed if the semi-colon at the end of the function command is omitted.

Note that only *detA* and *y* are in the workspace; no *A*, *b*, or *x*.