

Ex. No :10 Writing Functions - Creating functions, Calling functions and Setting the MATLAB path

Writing Function

Functions contain one or more sequential commands and can accept inputs and return outputs. To write a program with multiple lines of code, create a named function in a file. Alternatively, if you want to define a one-line function to pass to another function for instance, a mathematical expression to pass to the integral function you can create an anonymous function. Another type of M- file is a function file. Both scripts and functions allow to reuse sequences of commands by storing them in program files. Scripts are the simplest type of program, since they store commands exactly as they are typed at the command line.

Functions provide more flexibility, primarily because functions contain sequential commands and can accept inputs and return outputs. Unlike a script file, all the variables in a function file are local variables, which means their values are available only within the function. Function files are useful when you need to repeat a set of commands several times. They are the building blocks of larger programs. To create a function file, open the Editor /Debugger window.

Named functions must be defined within a program file and not at the command line.

syntax for function:

`function [y1,...,yN] = myfun(x1,...,xM)`

$y_1, \dots, y_N$ are outputs, $x_1, \dots, x_M$ are inputs, and myfun is the function name.

Creating functions

Both scripts and functions allow you to reuse sequences of commands by storing them in program files. Scripts are the simplest type of program, since they store commands exactly as you would type them at the command line. Functions provide more flexibility, primarily because

you can define a function that takes an input value (n) and returns the result (f). For example, this function calculates the factorial of a number (n) and returns the result (f).

```
function f = fact(n)
    f = prod(1:n);
end
```

This type of function must be defined within a file, not at the command line. Often, you store a function in its own file. In that case, the best practice is to use the same name for the function and the file (in this example, `fact.m`), since MATLAB® associates the program with the file name. Save the file either in the current folder or in a folder on the MATLAB search path.

You can call the function from the command line, using the same syntax rules that apply to functions installed with MATLAB. For instance, calculate the factorial of 5.

Example:

```
x = 5;
y = fact(5)
```

Output : `y =`

120

another option for storing functions is to include them at the end of a script file. For instance, create a file named *mystats.m* with a few commands and two functions, *fact* and *perm*. The script calculates the permutation of (3,2).

```
x = 3;
y = 2;
z = perm(x,y)

function p = perm(n,r)
    p = fact(n)/fact(n-r);
end

function f = fact(n)
    f = prod(1:n);
end
```

Call the script from the command line.

```
mystats
z =
```

6

Calling functions and Setting the MATLAB path

Syntax for Function Definition

The first line of every function is the definition statement, which includes the following elements.

function keyword (required)	Use lowercase characters for the keyword.
Output arguments (optional)	1. If your function returns one output, you can specify the output name after the function keyword. function myOutput = myFunction(x) 2. If your function returns more than one output, enclose the output names in square brackets. function [one,two,three] = myFunction(x) 3. If there is no output, you can omit it. function myFunction(x) 4. Or you can use empty square brackets. function [] = myFunction(x)
Function name (required)	Valid function names follow the same rules as variable names. They must start with a letter, and can contain letters, digits, or underscores
Input arguments (optional)	If your function accepts any inputs, enclose their names in parentheses after the function name. Separate inputs with commas. function y = myFunction(one,two,three) If there are no inputs, you can omit the parentheses.

Contents of Functions and Files

The body of a function can include valid MATLAB expressions, control flow statements, comments, blank lines, and nested functions. Any variables that you create within a function are stored within a workspace specific to that function, which is separate from the base workspace.

Program files can contain multiple functions. If the file contains only function definitions, the first function is the main function, and is the function that MATLAB associates with the file name. Functions that follow the main function or script code are called local functions. Local functions are only available within the file.

End Statements

Functions end with either an *end* statement, the end of the file, or the definition line for a local function, whichever comes first. The end statement is required if:

- Any function in the file contains a nested function (a function completely contained within its parent).
- The function is a local function within a function file, and any local function in the file uses the end keyword.
- The function is a local function within a script file.

Although it is sometimes optional, use end for better code readability.

Naming

A call to a function will look for a file with the function's name in the MATLAB path. Therefore, it makes sense to set the file name and the function name to be the same. For example, if your function is called draw_trapeze but the file is named trapeze.m, MATLAB will not accept a call to draw_trapeze. It will, however, accept a call to trapeze.m, and will subsequently run file trapeze regardless of the name you have specified as the function declaration within the file (draw_trapeze).

Function with One Output

Define a function in a file named average.m that accepts an input vector, calculates the average of the values, and returns a single result.

```
function ave = average(x)
    ave = sum(x(:))/numel(x);
end
```

Call the function from the command line.

```
z = 1:99;
ave = average(z)
ave =
    50
```

Function with Multiple Outputs

Define a function in a file named `stat.m` that returns the mean and standard deviation of an input vector.

```
function [m,s] = stat(x)
    n = length(x);
    m = sum(x)/n;
    s = sqrt(sum((x-m).^2/n));
end
```

Call the function from the command line.

```
values = [12.7, 45.4, 98.9, 26.6, 53.1];
[ave,stdev] = stat(values)
ave =
    47.3400
stdev =
    29.4124
```

Multiple Functions in a Function File

Define two functions in a file named `stat2.m`, where the first function calls the second.

```
function [m,s] = stat2(x)
    n = length(x);
    m = avg(x,n);
    s = sqrt(sum((x-m).^2/n));
end
```

```
function m = avg(x,n)
    m = sum(x)/n;
end
```

Function avg is a *local function*. Local functions are only available to other functions within the same file.

Call function stat2 from the command line.

```
values = [12.7, 45.4, 98.9, 26.6, 53.1];  
[ave,stdev] = stat2(values)  
ave =  
    47.3400  
stdev =  
    29.4124
```

Work to be carried out

- a) Find the general solution of the following first order ODE

$$\frac{dy}{dt} - y = e^{2t}.$$

- b) Find the general solution of the following second order ODE

$$\frac{d^2y}{dt^2} + 8\frac{dy}{dt} + 16y = 0.$$

- c) Find the particular solution in a) given the following initial conditions:

$$y(0) = 5.$$

- d) Write a function file that converts temperature in degrees Fahrenheit (°F) to degrees Centigrade (°C). Use **input** and **fprintf** commands to display a mix of text and numbers. Recall the conversion formulation, $C = 5/9 * (F - 32)$
- e) Write a short MATLAB function to find out whether a given number (up to 1,000,000) is a prime number. The function should return true or false. Bear in mind that 1 is not considered a prime number. (Hint: Use the brute force approach and divide the number (K) by all integers from 2 to K-1. Check the remainders for 0s.)
- f) Write a MATLAB function for calculating the Euclidean distance between two two-dimensional arrays A and B given as input arguments. A is of size $N \times n$, and B is

of size $M \times n$. The function should return a matrix D of size $N \times M$ where element $d(i, j)$ is the Euclidean distance between row i of A and row j of B

- g) Write a user-defined MATLAB function, with two input and two output arguments that determines the height in centimeters (cm) and mass in kilograms (kg) of a person from his height in inches (in.) and weight in pounds (lb).
- a. Determine in SI units the height and mass of a 5 ft. 15 in. person who weight 180 lb.
 - b. Determine your own height and weight in SI units.