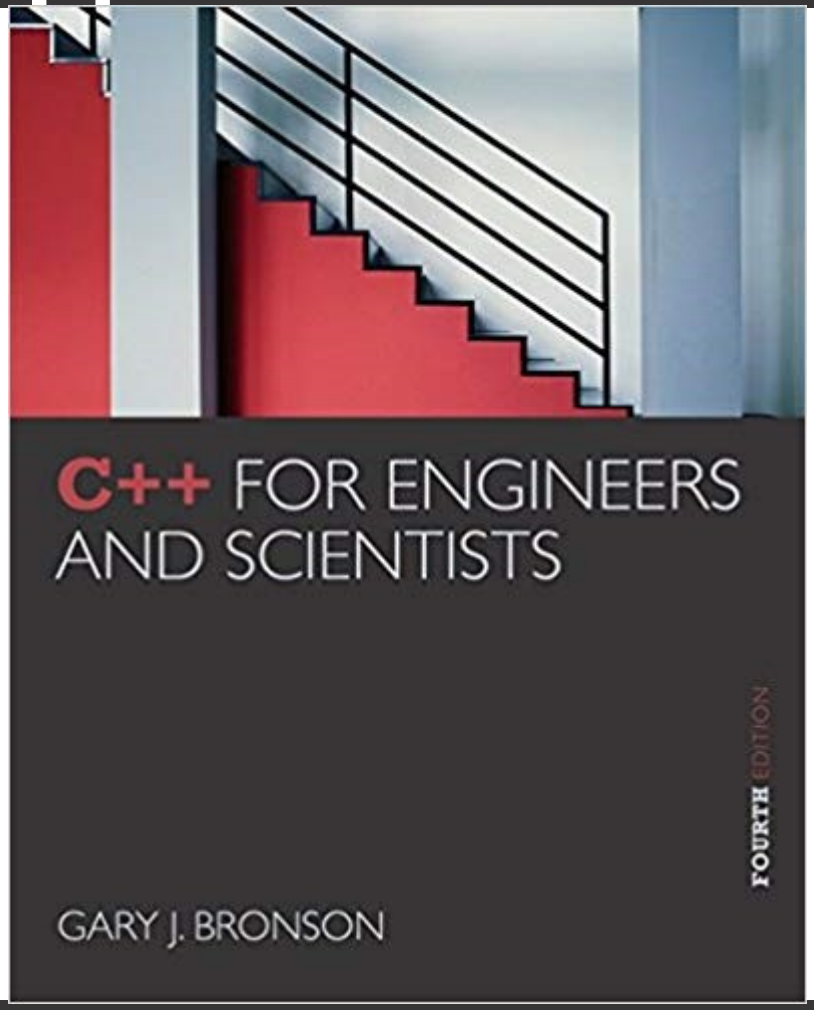
ELEG 1043
Computer Applications in Engineering



Chapter 12:
Matlab Function

C++ FOR ENGINEERS
AND SCIENTISTS

Acknowledgement

- Some of the slides or images are from various sources. The copyright of those materials belongs to their original owners.

Objectives

- In this chapter you will learn about:
  - Basics
  - Subfunctions
  - Nested functions
  - Anonymous functions
  - Function_handles

User-defined functions

- **MATLAB is built with commands & functions:**
  - **both sets are computer codes that accept input from the user and generate output.**
  - **Functions do not use or save variables in the workspace.**

- **Functions and M-file command scripts make technical computing efficient;**

User-defined functions

- **It is necessary, as the programmer, to know exactly**
  - **how a function performs its task;**
  - **otherwise, how to know that the task is completed correctly.**

Concept on function

- **User-defined functions take a certain number of inputs, perform some operation, and give output(s).**
- **For MATLAB built-in functions**
  - **know what they are supposed to do**
  - **know that it does it correctly.**

Syntax for functions

- **Calling a user-defined function:**
  - **my_function(x)**
- **Defining a user-defined function:**
  - **function y = my_function(x)**
  - **x is a value that is the input to my_function.**
  - **my_function performs a functional operation.**
  - **y is a value that is the output of my_function.**
- **Functions must be written in M-files. The M-file must have the same name as the function.**

Naming Function

- **The rule for naming functions is the same as for variables. A function name must**
  - **start with a letter,**
  - **should be meaningful,**
  - **should not use the name of an existing function,**
  - **should not be excessively long.**

Notes on functions

- **It is important to make meaningful variable names to variables inside a function**
  - **Understand what the function does MORE EASILY.**
- **Comments ARE extremely important in functions.**
  - **Comments help both you and anyone who might use the function to understand what it does.**

Function file example

- **Suppose a file called runthis.m contains the text**

```
X = linspace(0,4*pi,1000);
Y = sin(3*X) + 2*cos(5*X);
plot(X,Y)
maxy = max(abs(Y));
disp(['Peak of Y is ' num2str(maxy)]);
```

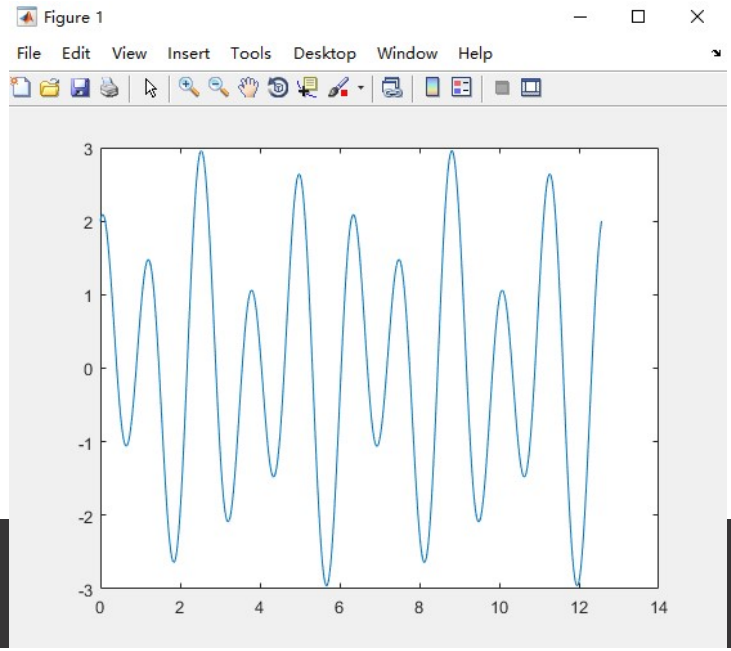
Function file example

- **Then, typing**

  >> runthis

  will cause the 5 lines to execute, resulting in a plot and a message about the peak value of Y.

Peak of Y is 2.9584



Functions

- In mathematics, a function is a rule that assigns to each value of the input, a corresponding output value.
- Consider the function f defined by the rule

  $f(x) = x^2$ for all numbers x.

- Here, x is the input value, and f(x) is the output value.

Functions

- **Equivalent**

$$f(y) = y^2 \text{ for all numbers } y.$$

- **Functions can have MULTIPLE inputs and produce (through multiple rules) MULTIPLE outputs, f1(a,b,c) = 2a+3b, f2(a,b,c) = bc.**

Functions

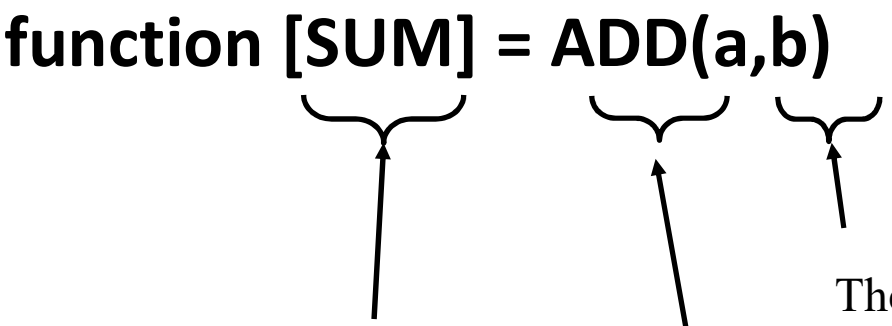
- **A "complex" task (eg., more than two lines of Matlab program code)**
  - **Reuse in a few (or several places)**
  - **A reusable function file.**
- **Then, only need to "call" the function every time you need to execute that task.**

Functions

- **This modularity helps**
  - **break down a huge program task into a collection of smaller tasks**
  - **Easier to design, write, debug and maintain.**
- **Functions are sometimes also called subroutines, methods, etc.**

A MATLAB function file

- **The first line is the function declaration line.**



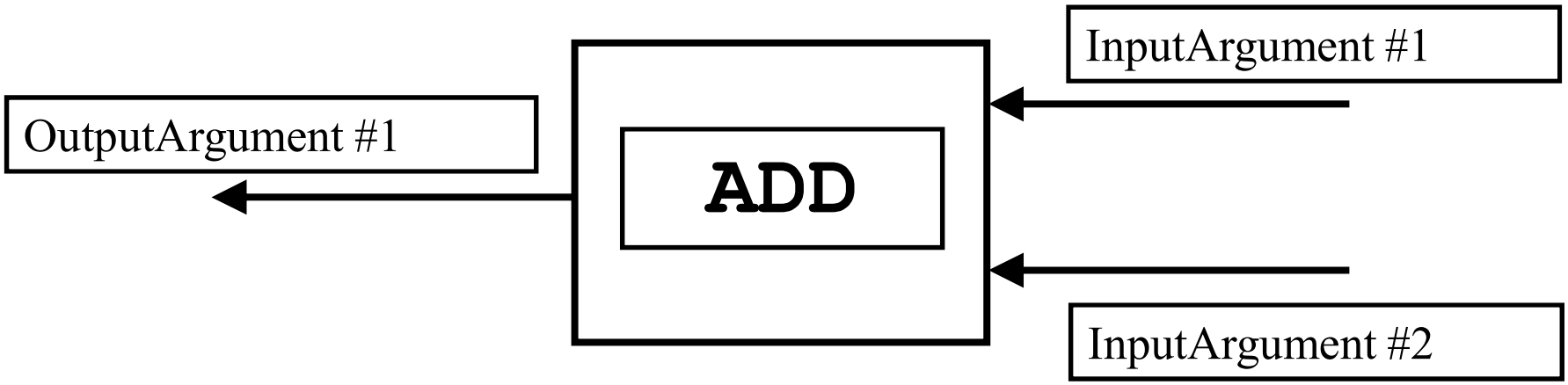
The output variables. This function has two. The function's purpose is to compute these variables, based on the values of the input variables.

The input variables. This function has two. Within the function, the names of the input variables are **a** and **b**.

The input and output variables are also called the input and output arguments.

The function name, this function should be saved in a file called **ADD.m**

ADD function



2-line function in ADD.m

Function declaration line

```
function [SUM] = ADD(a,b)
```

```
SUM = a + b;
```

Logically correct expressions and assignments that compute the output variables using the values of the input variables.

Comments add readability

```
function [SUM] = ADD(a,b)
%Add two numbers
SUM = a + b;
```

Calling a function

```
>>
>> a = 2;
>> b = 3;
>> [SUM] = ADD(a,b);
>>
```

Input/output for functions

- **Several kinds of functions can be created with different combinations of input and/or output:**
  - **Functions with input and output arguments**
  - **Functions with input arguments and no output arguments**
  - **Functions with output arguments and no input arguments**
  - **Functions with neither input arguments nor output arguments**

Input/output for functions

- **Input parameters: 0 or more**
  - **parameter list enclosed by ( )**
  - **if there are no parameters can omit ()**
    - **function show_date**

      today = date
  - **Example with three input parameters**
    - **function volume = box(height, width, depth)**

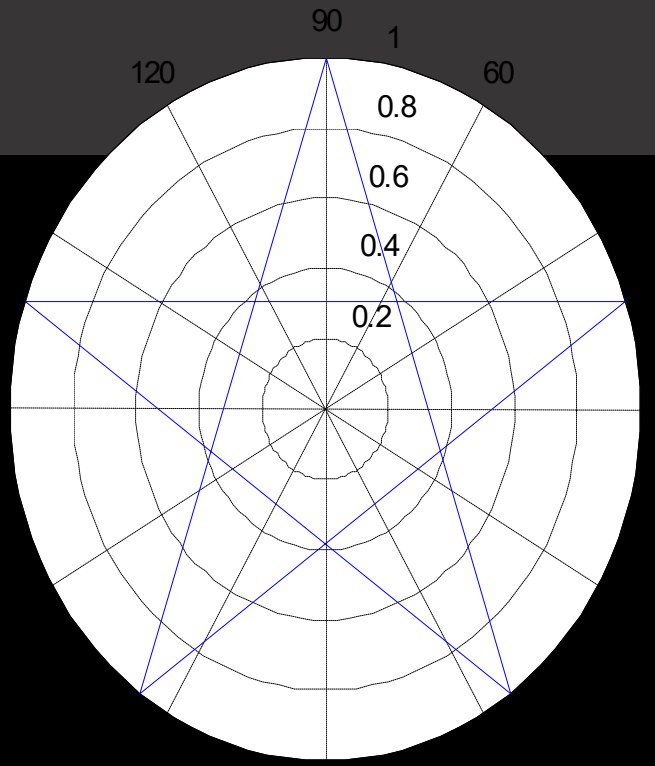
      volume = height.*width.*depth;

Input/output for user defined functions

- **Output parameter 0 or more**
  - **parameter list enclosed by [ ]**
  - **0 or 1 output parameter, can omit [ ]**
  - **2 or more parameters separated by commas**
  - **first output parameter ist set to ans, if function is called without output parameters**
  - **Examples:**
    - **function [areaC, cirumC] = circle(radius);**
    - **function areaSquare = square(sides) ;**

A Function with no input or output

```
function [] = star()
  theta = pi/2 : 0.8 * pi :
          4.8 * pi;
  r = [1 1 1 1 1 1];
  polar(theta, r);
```



Although the star function generates a plot, it does not have any output.

Built-in Functions

- **Exponential**
  - **exp(x)**
  - **sqrt(x)**
- **Logarithmic**
  - **log(x) %natural logarithm ln**
  - **log10(x)**
  - **log2(x)**

Built-in Functions

- **numeric**
  - **fix(x) round to nearest integer towards 0**
  - **round(x) round to nearest integer**
  - **sign(x) +1, 0 or -1**
- **complex**
  - **angle(x) in complex plane**
  - **imag(x)**
  - **real(x)**

Built-in Functions

- **Trigonometric and their inverse**
  - **cos(x) acos(x)**
  - **sin(x) asin(x)**
  - **tan(x) atan(x)**

Built-in Functions for vectors

- **max(x)**
  - **returns largest value in vector x**
- **[a,b] = max(x)**
  - **returns largest value in a and index where found in b**
- **max(x,y)**
  - **x and y arrays of same size, returns vector of same length with lager value from corresponding positions in x and y**
- **same type of functions are available for min**

Exercises

- **Write a function that converts Celsius temperatures into Fahrenheit.**

For example:

(50 degrees Celsius x 1.8) + 32 = 122 degrees Fahrenheit

Summary

- **Function concept and syntax**
- **Different kinds of functions**
  - **Functions with input and output**
  - **Functions with input only**
  - **Functions with output only**
  - **Functions with no input or output**