البرمجة الإجرائية

Lecture No. 9

Functions

ميكاترونيك-سنة أولى-فصل أول

Dr. Eng. Essa Alghannam
Ph.D. Degree in Mechatronics Engineering

2024

Functions in Matlab

- Functions are actually computer routines(programs) to perform specific task.
- Functions are m-files which can be executed by specifying some inputs and supply some desired outputs.
- A function is called, user gives specific input(s) to the function, which generates the required output(s).

Matlab is rich in built-in function

Example of built-in functions

- Our objective is to calculate mean or average of certain vector ‘x’.
- We call the built in function “mean” of Matlab and give ‘x’ as an input to the function.
- The mean is calculated and is stored in variable ‘y’.

x=[1 2 3 4 5 6 7 8 9]

y=mean(x)

$$y = \frac{\sum_{i=1}^{n} x_i}{n}$$

User Defined Functions

```
function function_name
Commands
end
```

Function name should match MATLAB file name. you should save the m-file with a file name same as the function name

```
function function_name(arg)
Commands
end
```

It must have the reserved word 'function' at the beginning of the m-file

```
function out=function_name(arg)
Commands
end
```

Inputs must be specified
arg is function input argument

out is function output

```
function out=function_name(arg 1, arg 2, arg 3)
Commands
end
```

If more than one output, must be in brackets

```
function [out1 out2 out3]=function_name(arg 1, arg 2, arg 3)
Commands
end
```

- No need for return: MATLAB returns variables whose names match those in the function declaration.
- Variable scope: Any non returned variables created within a function disappear after function stops running

```
function printsum(a, b, c)
  disp(a+b+c);
end
```

printsum.m

```
1   function printsum(a, b, c)
2       disp(a+b+c);
3   end
```

```
>> printsum(1, 2, 3)
   6
```

لا يعيد أي قيمة ولا يأخذ أي بارمتر كدخل

```
function clcclear
  clc; clear;
end
```

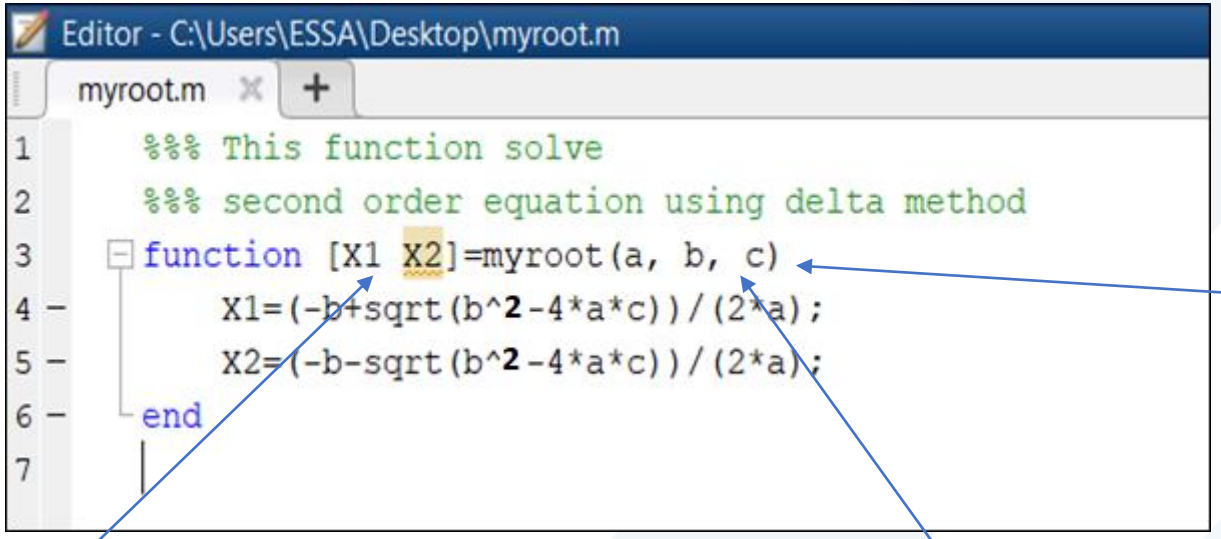
Editor - C:\Users\ESSA\Desktop\c

clcclear.m

```
1   function clcclear
2       clc; clear;
3   end
```

User Defined Functions

- Functions look exactly like scripts, but for **ONE** difference Functions must have a function declaration.



- Some comments about the function declaration

Function declaration

outputs

inputs

Example

- For example for a given quadratic equation we wish to calculate the roots of equation. So we build our own function "call" and use the quadratic formula to solve for x1 and x2.

```
%{
This function is for Solving Quadratic
equations (second degree) using delta method
%}
function [X1,X2]=myroot(a, b, c)
   X1=(-b+sqrt(b^2-4*a*c))/(2*a);
   X2=(-b-sqrt(b^2-4*a*c))/(2*a);
end
```

```
>> a=1;  b=2;  c=-3;
>> [x1 x2]=myroot(a,b,c)
x1 =
  -1.0000 + 0.2165i
x2 =
  -1.0000 - 0.2165is
```

- In our main program the function is called and the inputs 'a', 'b' and 'c' are given to the function which calculates the required roots

Example

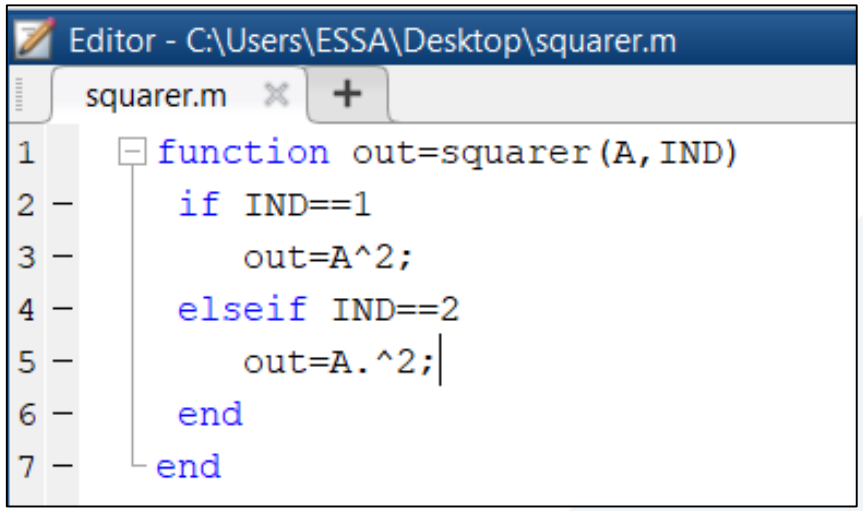
```
%{
This function is for Solving Quadratic
equations (second degree) using delta method
%}
function X=myroot(a, b, c)
  delta=b^2-4*a*c;
  if delta>0
   X(1)=(-b+sqrt(delta))/(2*a);
   X(2)=(-b-sqrt(delta))/(2*a);
  elseif delta==0
    X=-b/(2*a);
  end
end
```

Examples

- Write a function : out=squarer (A, ind)
  - Which takes the square of the input matrix if the input indicator is equal to 1
  - And takes the element by element square of the input matrix if the input indicator is equal to 2

Example

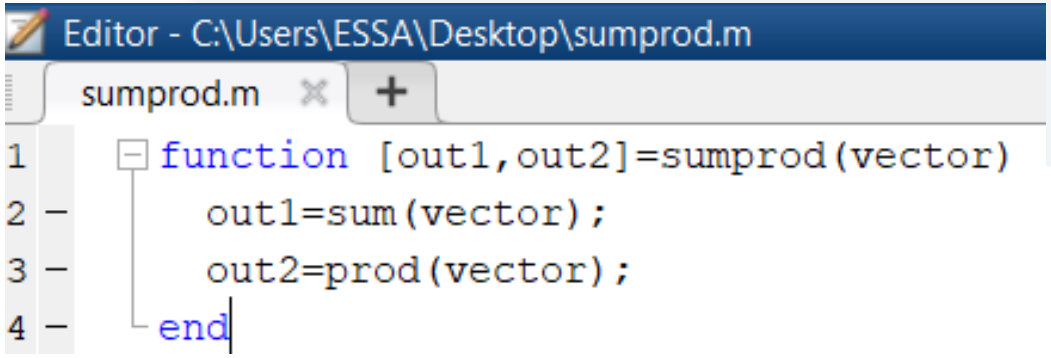
```
function out=squarer(A,IND)
 if IND==1
   out=A^2;
 elseif IND==2
   out=A.^2;
 end
end
```




```
>> myout=squarer([1 2 ; 3 4],1)

myout =

   7   10
  15   22

>> myout=squarer([1 2 ; 3 4],2)

myout =

   1    4
   9   16
>> myout=squarer([1 2 ; 3 4],0)
Output argument "out" (and maybe others) not assigned
during call to "squarer".
```

Example

- Another function which takes an input vector and returns the sum and product of its elements as outputs




```
function [out1,out2]=sumprod(vector)
 out1=sum(vector);
 out2=prod(vector);
end
```

- The function sumprod() can be called from command window or an m-file as

```
>> X=1:8;
>> [S P]=sumprod(X)
S =
   36
P =
   40320
```

```
>> X=1:10;
>> [S P]=sumprod(X)
S =
   55
P =
   3628800
```

Example

ضرب عناصر مصفوفة ببعضها

```
function out=myprod(array)
 [row,column]=size(array);
 out=1;
 for i=1:row
    for j=1:column
       out = out*array(i,j);
    end
 end
end
```

Editor - C:\Users\ESSA\Desktop\myprod.m

myprod.m

```
1  function out=myprod(array)
2      [row,column]=size(array);
3      out=1;
4      for i=1:row
5          for j=1:column
6              out = out*array(i,j);
7          end
8      end
9  end
```

```
>> a=[10 2 3;4 5 6;7 8 9];
>> myprod(a)

ans =

   3628800
```

جَامعة
المَنارة
MANARA UNIVERSITY

Thanks .