EP375 Computational Physics

Topic 3

MATLAB TUTORIAL

SCRIPT & FUNCTIONS

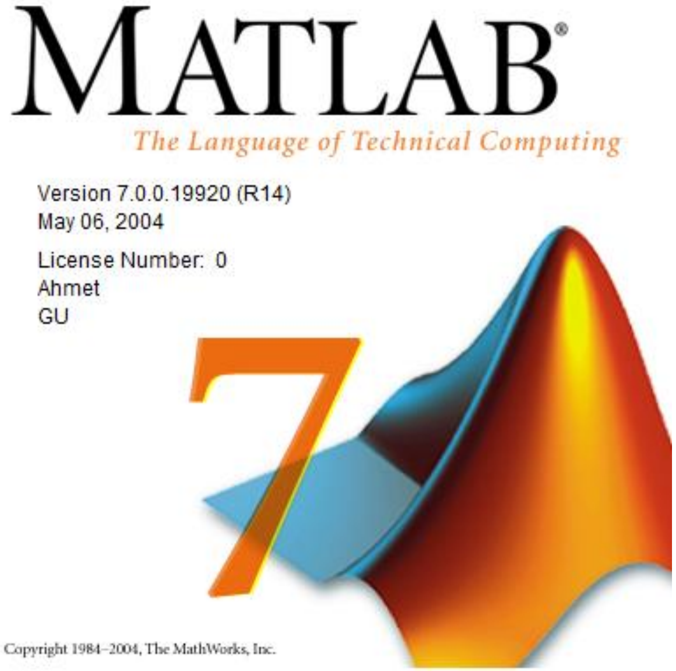
Department of
Engineering Physics

University of Gaziantep

Feb 2014

Content

1. M-Files
2. Functions



MATLAB®

The Language of Technical Computing

Version 7.0.0.19920 (R14)
May 06, 2004

License Number: 0
Ahmet
GU

Copyright 1984–2004, The MathWorks, Inc.

M-Files

We can generate the script file with the **edit** command.

```
>> edit topla.m
```

```
A = input(‘A = ’);
B = input(‘B = ’);
C = A + B;
disp(C)
```

```
>> topla
A = 2
B = 5
C = 7
```

```
>> topla
A = [1 2]
B = [3 4]
C = 4 6
```

Funtions

Function decleration:

```
function [output_arguments] = fname(input_arguments)
    block
end
```

Here

- ***fname*** **is the name of the function**
- ***input*** **and** ***output arguments*** **must be sepatated by commas.**
- **Number of arguments may be zero.**

- To make the function accessible to other programs units, it must be saved under the filename ***fname.m***.

```
>> edit deBroglie.m
```

```
function lambda = deBroglie(p)
  h = 6.6e-34;
  lambda = h/p;
end
```

```
>> deBroglie(0.07)
ans =
       9.428571428571428e-033
```

- A **return** statement can be used to finalize the function.

```
>> edit fact.m
```

```
function f = fact(n)
  if(n<0)
      f = 0;
      return
  end
  f = 1;
  for i=1:n
      f = f * i;
  end
end
```

```
>> x = fact(3)
x = 6

>> x = fact(-4)
x = 0
```

- ***fname.m*** may contain other functions, called *sub-functions.*
- Sub-functions are accessible in function **`fname()`** but are not accessible to other program units.

```
>> edit binom.m
```

```
% Computes binom coefficient defined by
% C(n,r) = n! / (r!*(n-r)!)
function c = binom(n,r)
   c = fact(n)/(fact(r)*fact(n-r));
end
function f = fact(n)
   f = 1;
   for i=1:n
       f = f * i;
   end
end
```

```
>> binom(5,3)
ans = 10
```

- A MATLAB function may return more than one value.

```
>> edit car2pol.m
```

```
% Cartesian to polar transformation
function [r, theta] = car2pol(x,y)
  r = sqrt(x^2 + y^2);
  theta = atand(y/x); % in degrees
end
```

```
>> [r,t] = car2pol(3,4)
r = 5
t = 53.1301
```

```
>> p = car2pol(3,4)
p = 5
```

- The number of input and output arguments used in the function call can be determined by **nargin** and **nargout**.
- Following function returns the valume of a cylinder whose radius r and height h. By default h is assumed to be 1 m.

```
>> edit cylinderVolume.m
```

```
function v = cylinderVolume(r, h)
  if nargin == 1
    h = 1;
  end
  v = 2*pi*r^2 + 2*pi*r*h;
end
```

```
>> cylinderVolume(10,4)
ans = 879.6459
```

```
>> cylinderVolume(10)
ans = 691.1504
```

- A function may return an array of values.
- The following function returns the positive divisors of an integer *n*.

```
>> edit divisors.m
```

```
% Returns positive divisors of integer n
function d = divisors(n)
   j = 1;
   for i=1:n
      if mod(n,i) == 0
          d(j) = i;
          j = j + 1;
      end
   end
end
```

```
>> q = divisors(28)
q =      1      2      4      7     14     28
```

- If the function is not overly complicated, it can also be represented as an **inline** object:

```
>> f = inline('x^2');
>> f(3)
ans = 9
```

```
>> myfunc = inline ('x^2 + y^2', 'x', 'y');
>> myfunc(3,5)
ans = 34
```

HW1:

The average distance to the Sun is about 93 million miles. If the speed of light is about 3x10^8 m/s then how many minutes and seconds does it take Sun light to reach the earth? Solve this problem using MATLAB script.

(Hint:1 mile is about 1.609 kilometers)

HW2:

Write MATLAB script to evaluate the sum

$$1 - 1/3 + 1/5 - 1/7 + 1/9 - \ . \ . \ . \ - 1/1000001$$

with a for loop and time the computation using tic and toc.

HW3:

Write a function named **`function p = isPrime(n)`**

that returns 1 if *n* is a prime number and returns 0 otherwise.

Example function call in MATLAB command line:

```
>> x = isPrime(3)
x = 1
```

HW4:

Write a function named **`function v = primeList(n)`** that

returns list (vector) of prime numbers less than or equal to *n*.

Example function call in MATLAB command line:

```
>> p = primeList(20)
>> p =  2   3   5   7   9   11   13   17   19
```

HW5:

Write a function named **`function lp(n, m)`** to return the largest positive integer power of input the argument n that is less than m where m>n. For example if n=5 and m=1000 largest power less than 1000 is $5^4 = 625$. So the function must return 625.

HW6:

Write a program to perform the following algorithm:

```
S1:  Start
S2:  Input integers a and b
S3:  If a > b then swap a and b
     (i.e. do c = a ; a = b ; b = c)
S4:  If a <= 0 then go to S7
S5:  b = b - a
S6:  Go to S3
S7:  output b
S8:  End
```

HW7:

Series expansion of the sin(x) function is:

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + \ldots$$

Write a function named **`function mySin(x)`** that takes real angle x (in radians) returns the result of the sum of the first 15 terms of the expansion.

HW8:

Write a function of the form **`function y = rmaxval(x)`** to remove the largest element(s) from a vector **`x`**.

Example call in command window:

```
>> a = [1 2 5 0 0 5];
>> b = rmaxval(a)
b = 1 2 0 0
```

References

[1]. http://www.mathworks.com/products/matlab

[2]. Numerical Methods in Engineering with MATLAB,
J. Kiusalaas, Cambridge University Press (2005)

[3]. Numerical Methods for Engineers, 6th Ed.
S.C. Chapra, Mc Graw Hill (2010)