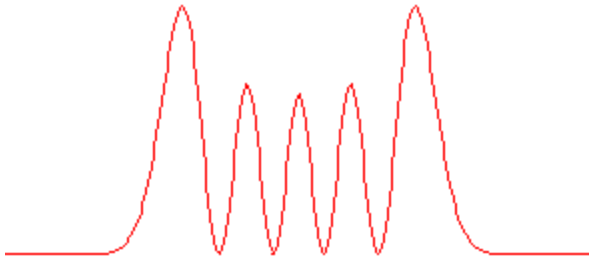
EP547 Computational Methods in QM

Topic 3

Scripts & Functions



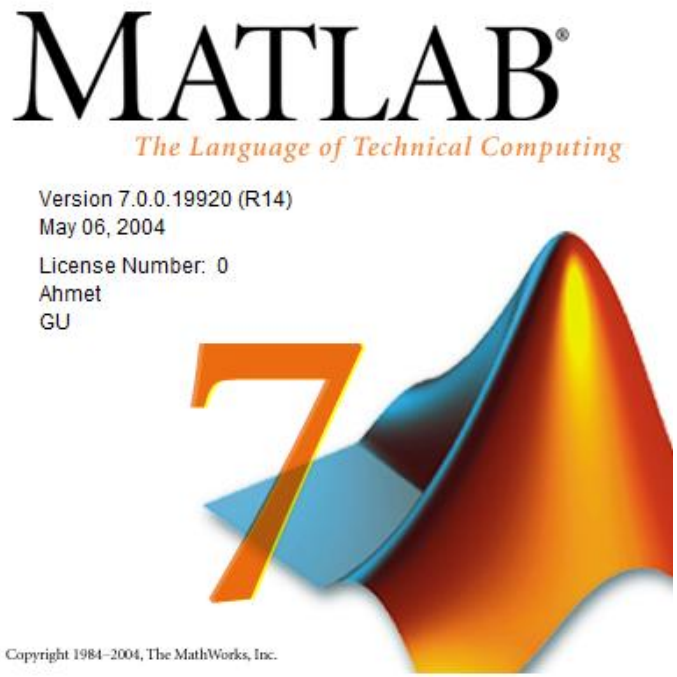
Department of Engineering Physics

University of Gaziantep

Feb 2013

Content

1. **M-Files**
2. **Functions**



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 factorial.m
```

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

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

>> x = factorial(-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
```

- 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
```

- One can call a function in another one.

myfunc.m

```
% A function definition
function y = myfunc(x)
   y = 2*x^2 - 5*x + 16;
end
```

deriv.m

```
% Computes first derivative of a function
% using FDA method
function fda = deriv(f,x)
   h = 0.01;
   fda = (f(x+h)-f(x))/h;
end
```

```
>> myfunc(2)          % myfunc value at x = 2
ans = 14
>> deriv(@myfunc,2)   % derivative of myfunc at x = 2
ans = 3.02
>> deriv(@exp,2)      % derivative of exp(x) at x = 2
ans = 7.4261
```

- 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
```

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)