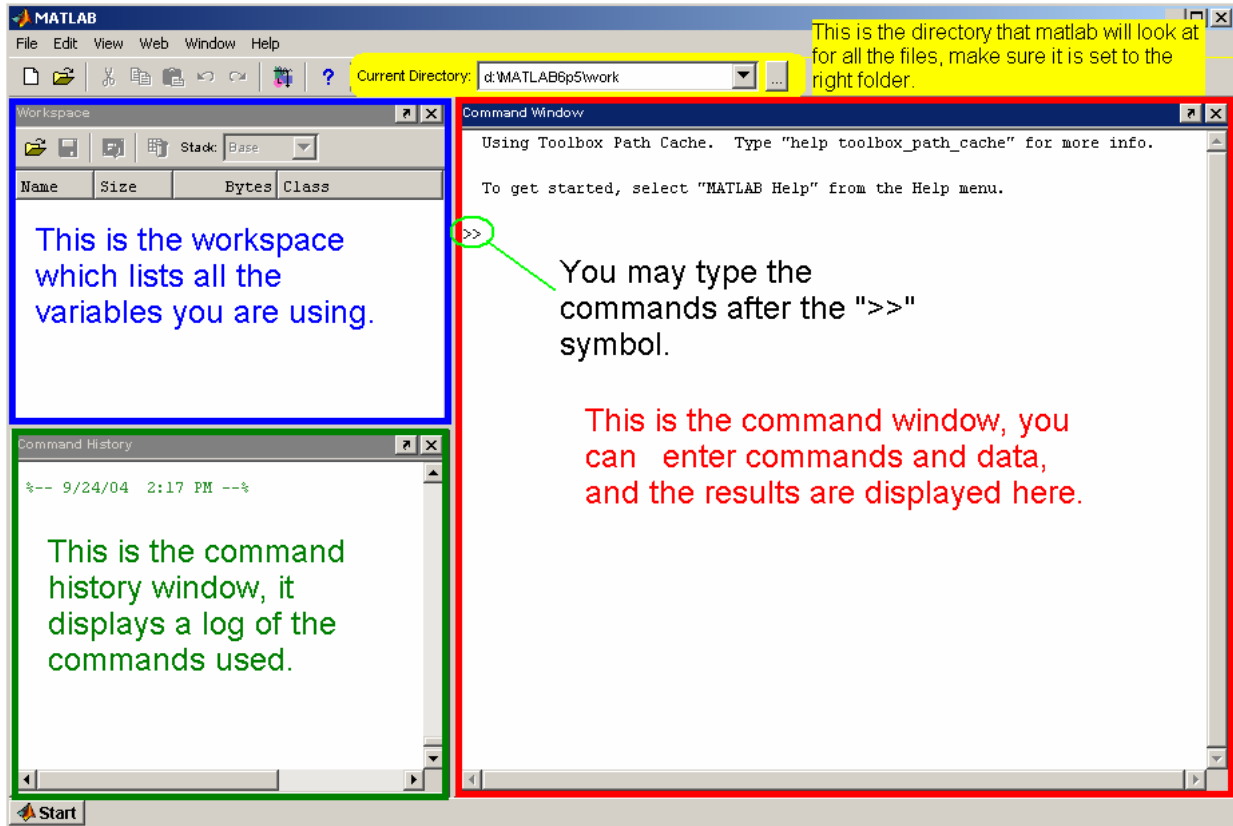


# Introduction to Matlab

## 1. Starting Matlab



- It might be different if you use an older version of Matlab (mine is 7.0.1).
- Make sure you have Signal Processing Toolbox, Control System Toolbox, and Symbolic Math Toolbox. To check, type

```
>> help
```

and you should see these toolboxes listed among all the help pages:

```
signal\signal          - Signal Processing Toolbox
control\control        - Control System Toolbox
toolbox\symbolic       - Symbolic Math Toolbox
```

## 2. Numbers, constants, variables and data types

- Numbers

```
>> 4
```

```
ans =
```

```
4
```

- Constants

```
>> pi
```

```
ans =
    3.1416
>> 3+4i
ans =
    3 + 4.0000i
```

- **MATLAB** variable names must begin with a letter, which may be followed by any combination of letters, digits, and underscores. **MATLAB** distinguishes between uppercase and lowercase characters, so **A** and **a** are not the same variable.

- **Vectors**

```
>> v = [2 3 4 5 6 7]
v =
    2    3    4    5    6    7
>> v(4)
ans =
    5
>> w = 2:0.5:7
w =
Columns 1 through 6
    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000
Columns 7 through 11
    5.0000    5.5000    6.0000    6.5000    7.0000
```

- **Matrices** – the formal name of Matlab is **MA**Trix **LAB**oratory, so Matlab is very efficient in handling matrices

```
>> A = [1 2 3; 4 5 6; 7 8 9]
A =
    1    2    3
    4    5    6
    7    8    9
>> A(1,2)
ans =
    2
```

- **Strings**

```
>> w = 'matlab is simple'
w =
matlab is simple
```

- **Symbolic variables**

```
>> syms x y z
>> x
x =
x
```

### 3. Operations and functions

- Simple operations all work as they should be. Type “help ops” for a list of operators and “help elfun” for a list of elementary mathematical functions. Almost all of them can be applied to scalar, vectors, and matrices.

```
>> ((3+4*5)/(4+5i))^0.44
ans =
    1.6206 - 0.6743i
>> sin([1 2 3 4])
ans =
    0.8415    0.9093    0.1411   -0.7568
>> exp([1 2 3; 4 5 6; 7 8 9])
ans =
  1.0e+003 *
    0.0027    0.0074    0.0201
    0.0546    0.1484    0.4034
    1.0966    2.9810    8.1031
```

- Entry-wise operations are preceded with a “.”

```
>> a = [1 2 3; 4 5 6; 7 8 9]; % <-comment sign
>> b = [10 11 12; 13 14 15; 16 17 18]; % ;<- no echo
>> a*b
ans =
    84    90    96
   201   216   231
   318   342   366
>> a.*b
ans =
    10    22    36
    52    70    90
   112   136   162
```

- Function calls work pretty much the same way. The most powerful aspect of Matlab is a large collection of functions commonly used in many engineering and scientific communities. Here are some of the signal processing commands we are going to use in this course. We will discuss their usages later.

residue	Partial-fraction expansion
fourier, ifourier	Symbolic forward and inverse Fourier Transform
laplace, ilaplace	Symbolic forward and inverse Laplace Transform
tf	Create transfer function for a system
bode	Magnitude and Phase response of a system
pzmap	Pole zero plot of a system
ss	Create state-space model for a system

ss2tf, tf2ss	Conversion between transfer function and state-space representation
lsim	Simulate time response of LTI models
eig	Eigenvalues of matrix
expm	Matrix exponential
c2d	Continuous to discrete time
ztrans, iztrans	Symbolic forward and Inverse Z-transform
residuez	Partial-fraction expansion in z-domain
filter	Discrete-time filtering
conv	Discrete-time convolution
butter	Butterworth filter design
cheby1	Chebyshev 1 filter design
impinvar	Impulse invariance method for analog-to-digital filter conversion
bilinear	Bilinear transformation method for analog-to-digital filter conversion
lp2bp, lp2bs, lp2hp, lp2lp	Frequency transformation
fir1, fir2	FIR filter design
freqs, freqz	Numerical Fourier transform
fft	Fast Fourier Transform

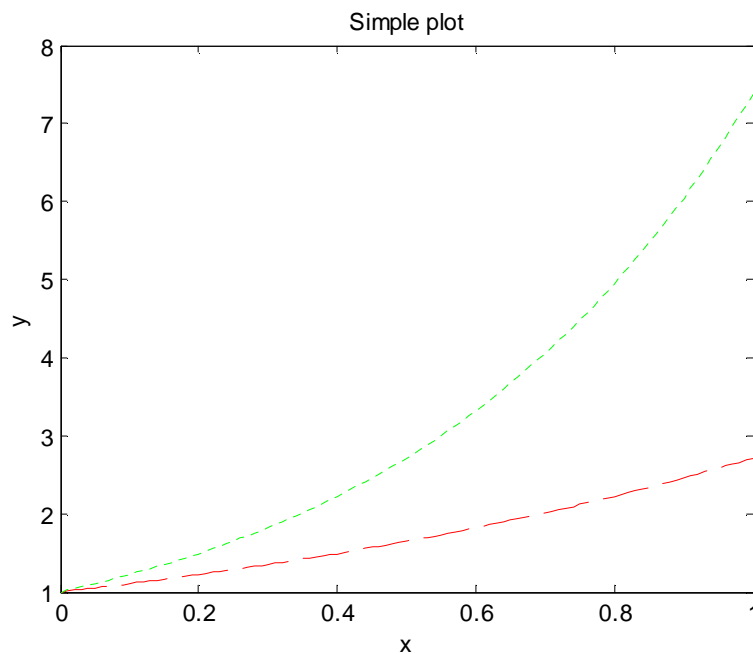
- Symbolic operations – most of the basic operations can be applied directly to symbolic variables. To actual evaluate an expression of symbolic variables, use the “subs” command.

```
>> syms a b c
>> s = sin(a)*b*exp(c)
s =
sin(a)*b*exp(c)
>> a=0.5;b=9;c=1;
>> subs(s)
ans =
    11.7289
>> a=0;b=0;c=0;
>> subs(s)
ans =
    0
```

## 4. Miscellaneous

- Plotting

```
>> x=0:0.01:1; y=exp(x);  
>> x=0:0.01:1;  
>> plot(x,exp(x),'r--',x,exp(2*x),'g:');  
>> xlabel('x');  
>> ylabel('y');  
>> title('Simple plot');
```



- Saving the workspace

```
>> save everything.mat % save workspace to "everything.mat"  
>> load everything.mat % load "everything.mat"
```

- Saving what you type and the matlab output to a text file

```
>> diary foo.txt % output & response goto "foo.txt"  
>> ...  
>> diary off % stop logging
```

- You can also save all your commands in a file and execute it later on. The file should have a ".m" extension