A Crash Course in MatLab

For Masters and other students

Abdul Hanan Sheikh

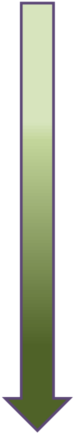
Department of Mathematics, QUEST Nawabshah

2016

Today’s objective

Get comfortable playing with Matlab...

- Interacting with Matlab
- Enter Data
- Operations
- Some Commonly Used Functions
- Making Pretty Pictures
- M-Files and Scripts
- For, While, and If
- Solving ODEs



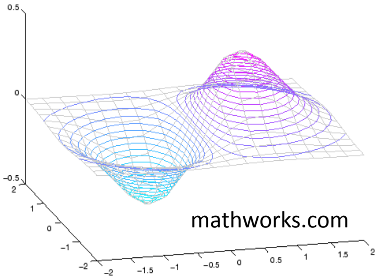
What is Matlab?

hibp.ecse.rpi.edu/~connor

- Matlab = Matrix Laboratory
- Problem-solving environment
- Designed for convenient *numerical* computations (e.g. matrix manipulation, differential eqns, stats, and graphics)
- Developed by Cleve Moler in 1970s as a teaching tool
- Now ubiquitous in education and industry

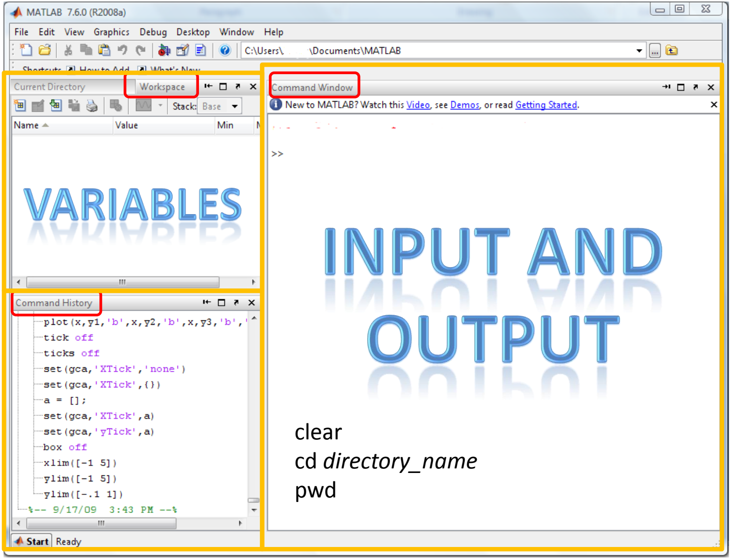
Why Matlab?

- Great tool for simulation and data analysis
- User-friendly interface
- Many easy to use built-in functions and tool boxes
- Easy visualization
- Easy to get help:
  - help *function_name*
  - lookfor *topic*
  - www.mathworks.com



mathworks.com

Interacting with Matlab



VARIABLES

```
plot(x,y1,'b',x,y2,'b',x,y3,'b','
tick off
ticks off
set(gca,'XTick','none')
set(gca,'XTick',{})
a = [];
set(gca,'XTick',a)
set(gca,'yTick',a)
box off
xlim([-1 5])
ylim([-1 5])
ylim([-.1 1])
%-- 9/17/09  3:43 PM --%
```

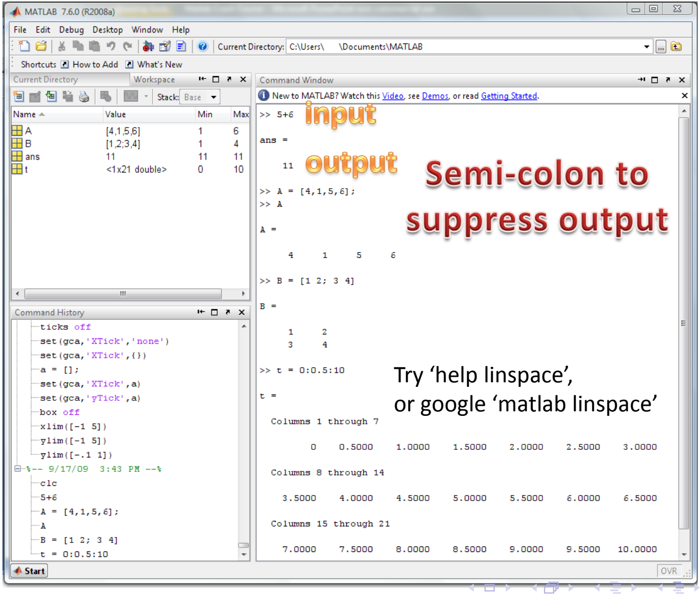
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>>
```

INPUT AND OUTPUT

clear
cd *directory_name*
pwd

Entering Data


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\ \Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Workspace
Stack: Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

Name	Value	Min	Max
A	[4,1,5,6]	1	6
B	[1,2;3,4]	1	4
ans	11	11	11
t	<1x21 double>	0	10

```
ticks off
set(gca,'XTick','none')
set(gca,'XTick',{})
a = [];
set(gca,'XTick',a)
set(gca,'yTick',a)
box off
xlim([-1 5])
ylim([-1 5])
ylim([-.1 1])
%-- 9/17/09  3:43 PM --%
clc
5+6
A = [4,1,5,6];
A
B = [1 2; 3 4]
t = 0:0.5:10
```

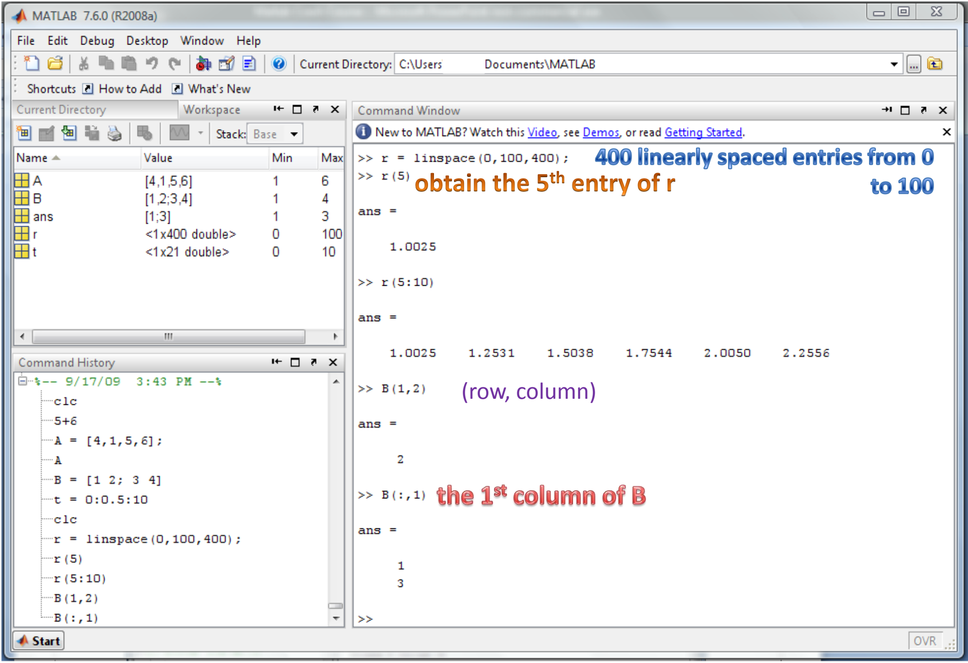
```
>> 5+6

ans =

    11

>> A = [4,1,5,6];
>> A

A =

     4     1     5     6

>> B = [1 2; 3 4]

B =

     1     2
     3     4

>> t = 0:0.5:10

t =

  Columns 1 through 7

         0    0.5000    1.0000    1.5000    2.0000    2.5000    3.0000

  Columns 8 through 14

    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000

  Columns 15 through 21

    7.0000    7.5000    8.0000    8.5000    9.0000    9.5000   10.0000
```

input

output

Semi-colon to suppress output

Try ‘help linspace’,
or google ‘matlab linspace’


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Users Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Stack: Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

Name	Value	Min	Max
A	[4,1,5,6]	1	6
B	[1,2;3,4]	1	4
ans	[1;3]	1	3
r	<1x400 double>	0	100
t	<1x21 double>	0	10

```
%-- 9/17/09  3:43 PM --%
clc
5+6
A = [4,1,5,6];
A
B = [1 2; 3 4]
t = 0:0.5:10
clc
r = linspace(0,100,400);
r(5)
r(5:10)
B(1,2)
B(:,1)
```

```
>> r = linspace(0,100,400);
>> r(5)

ans =

    1.0025

>> r(5:10)

ans =

    1.0025    1.2531    1.5038    1.7544    2.0050    2.2556

>> B(1,2)

ans =

     2

>> B(:,1)

ans =

     1
     3

>>
```

400 linearly spaced entries from 0 to 100

obtain the 5th entry of r

(row, column)

the 1st column of B

Some Frequently Used Commands

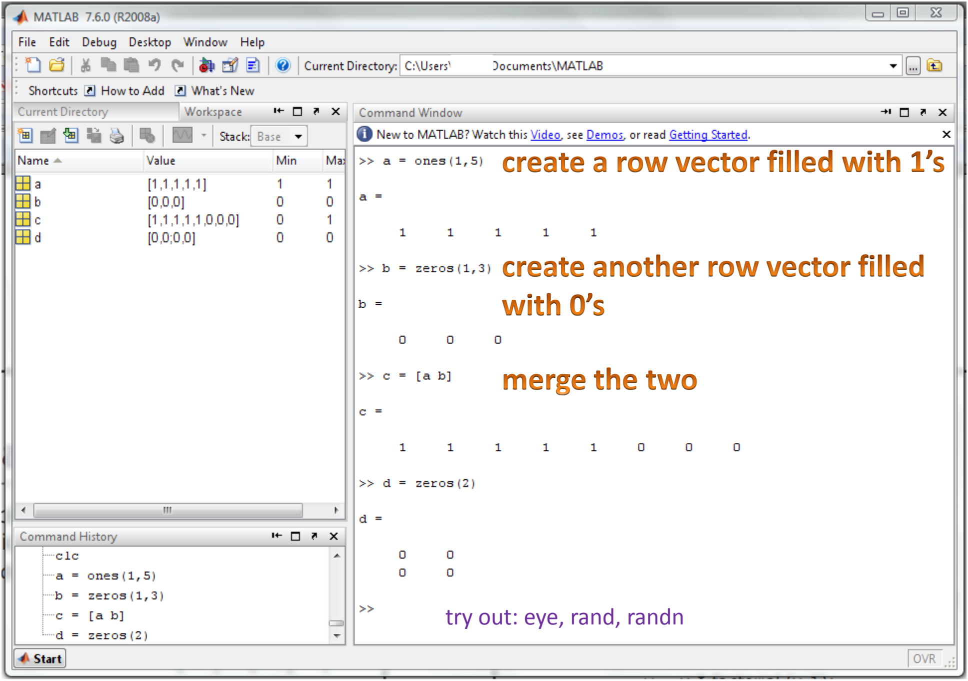
- To show variable: `who` and `whos`
- To get help on any command: `help any_command`
- To get documentation of any command: `doc any_command`
- For clearing screen: `cls`
- For removing variables from memory: clear variable_name

Entering data (in workspace/command line):

- Quite simple 0 : `a = 2`
- Semicolon, stops printing values for variables. : `a = 2;`
- Vectors in brackets [ ] : `vec = [1 2 3];`
- Matrices, as combination of vectors:`mat = [1 2 3;4 5 6;7 8 9]`
- Using existing arrays(vectors and matrices); for e.g. using first row of above matrix `mat(1,:)`

Some Functions:

- To get vectors(or matrices) of elements 0 : `zeros(m,n)`
- To get vectors(or matrices) of elements 1 : `ones(m,n)`
- Sum, Subtract, Multiplication and Division : `+, -, *, /` for all objects.
- Point-wise operations: `.* , ./ and so on`
- Exponential, Logarithm and other functions: `exp, log,`
- Formatting numbers: for e.g. `format short, format long, and others`



MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Users\ [redacted] Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Stack: Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

Name	Value	Min	Max
a	[1,1,1,1,1]	1	1
b	[0,0,0]	0	0
c	[1,1,1,1,1,0,0,0]	0	1
d	[0,0;0,0]	0	0

```
>> a = ones(1,5)

a =

     1     1     1     1     1

>> b = zeros(1,3)

b =

     0     0     0

>> c = [a b]

c =

     1     1     1     1     1     0     0     0

>> d = zeros(2)

d =

     0     0
     0     0

>>
```

create a row vector filled with 1's

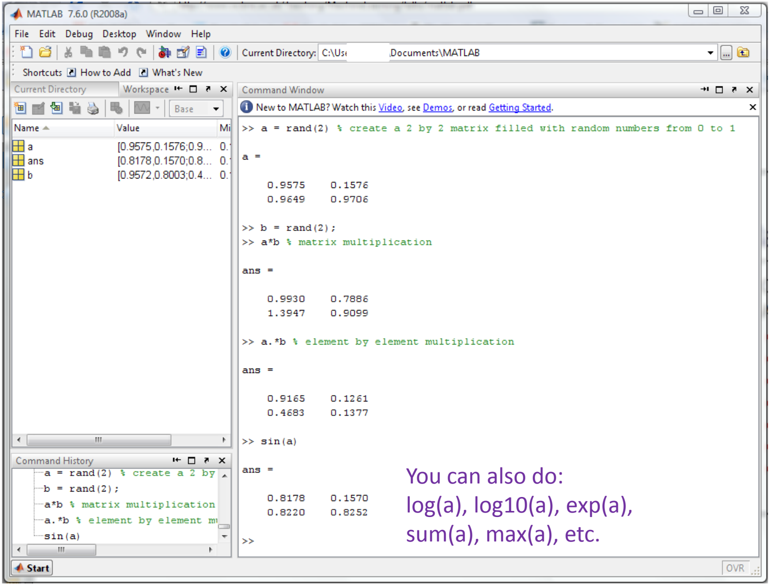
create another row vector filled with 0's

merge the two

try out: eye, rand, randn

```
clc
a = ones(1,5)
b = zeros(1,3)
c = [a b]
d = zeros(2)
```

Operations


MATLAB 7.6.0 (R2008a)
File Edit Debug Desktop Window Help
Current Directory: C:\Use Documents\MATLAB
Shortcuts How to Add What's New
Current Directory Workspace
Base
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Command History
Start
OVR

Name	Value	Mi
a	[0.9575,0.1576;0.9...	0.
ans	[0.8178,0.1570;0.8...	0.
b	[0.9572,0.8003;0.4...	0.

```
>> a = rand(2) % create a 2 by 2 matrix filled with random numbers from 0 to 1

a =

    0.9575    0.1576
    0.9649    0.9706

>> b = rand(2);
>> a*b % matrix multiplication

ans =

    0.9930    0.7886
    1.3947    0.9099

>> a.*b % element by element multiplication

ans =

    0.9165    0.1261
    0.4683    0.1377

>> sin(a)

ans =

    0.8178    0.1570
    0.8220    0.8252

>>
```

Command History:

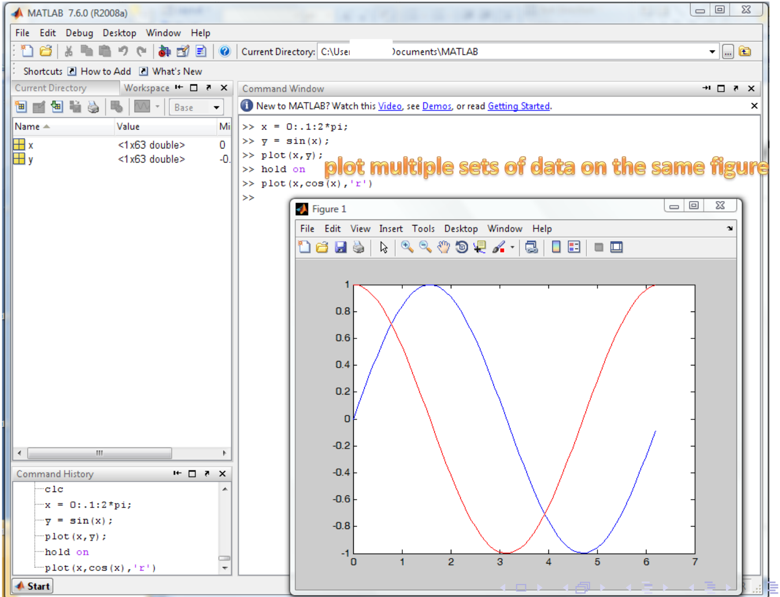
```
a = rand(2) % create a 2 by
b = rand(2);
a*b % matrix multiplication
a.*b % element by element mu
sin(a)
```

You can also do:
log(a), log10(a), exp(a),
sum(a), max(a), etc.

Plotting :

- To start blank figure : `figure`
- plot : `plot(y,x,'OPTIONS')`, where y and x are vectors(or matrices) and options are like line style, line color, etcetera.

Plotting Data / Making Pretty Pictures



plot multiple sets of data on the same figure

```
>> x = 0:.1:2*pi;
>> y = sin(x);
>> plot(x,y);
>> hold on
>> plot(x,cos(x),'r')
>>
```

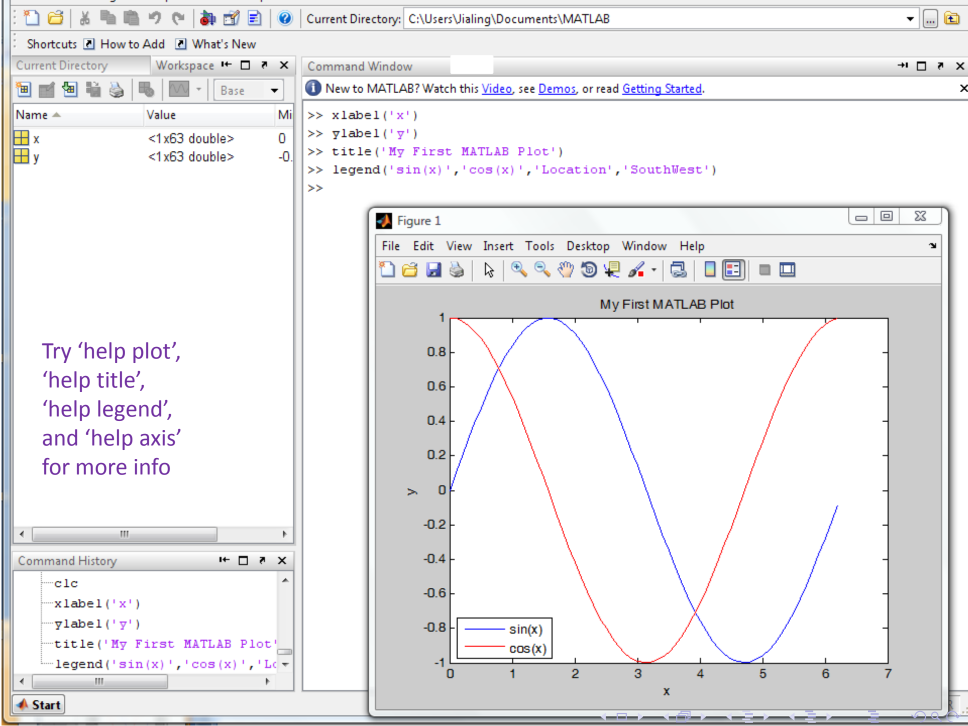
Name	Value	Mi
x	<1x63 double>	0
y	<1x63 double>	-0.

Command History

```
clc
x = 0:.1:2*pi;
y = sin(x);
plot(x,y);
hold on
plot(x,cos(x),'r')
```

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Current Directory: C:\Use Documents\MATLAB



Current Directory: C:\Users\Jialing\Documents\MATLAB

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Name	Value	Mi
x	<1x63 double>	0
y	<1x63 double>	-0.

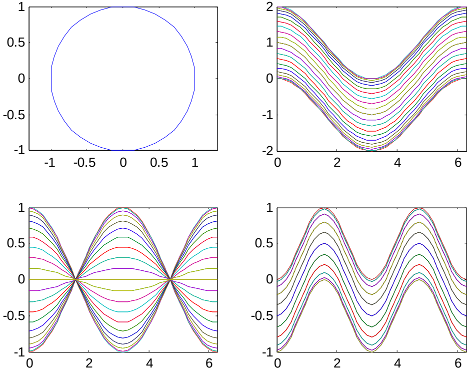
```
>> xlabel('x')
>> ylabel('y')
>> title('My First MATLAB Plot')
>> legend('sin(x)','cos(x)','Location','SouthWest')
>>
```

```
clc
xlabel('x')
ylabel('y')
title('My First MATLAB Plot'
legend('sin(x)','cos(x)','Lc
```

Try ‘help plot’, ‘help title’, ‘help legend’, and ‘help axis’ for more info

More About Plotting

```
t = 0:pi/20:2*pi;
[x,y] = meshgrid(t);   % look up meshgrid

subplot(2,2,1)    % creates a 2x2 array of plots, and plot in the first subplot
plot(sin(t),cos(t))
axis equal   % this is a parametric plot

subplot(2,2,2)
z = sin(x)+cos(y);    % z is a matrix
plot(t,z)
axis([0 2*pi -2 2])     % plotting each column of z
                        % versus t

subplot(2,2,3)
z = sin(x).*cos(y);
plot(t,z)
axis([0 2*pi -1 1])

subplot(2,2,4)
z = (sin(x).^2)-(cos(y).^2);
plot(t,z);
axis([0 2*pi -1 1])
```



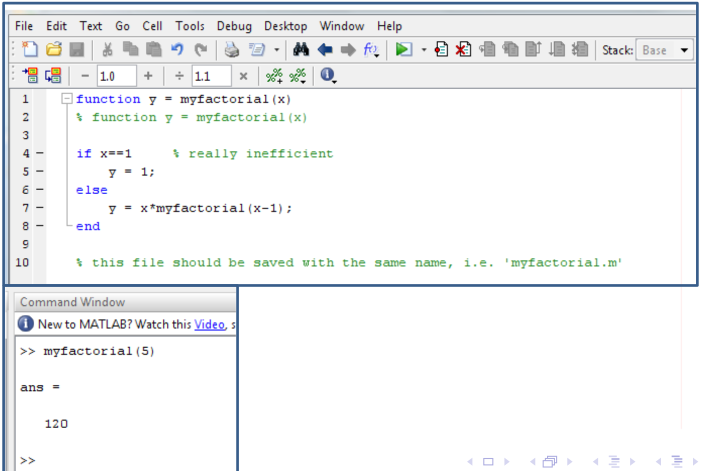
```
% for 3-D plotting, try mesh, surf, surfl, waterfall, etc
```

Creating files and M-Files :

- creat files in editor : `edit newfile.m`
- Functions in m-files : `ones(m,n)`

M-Files and Functions

- Let’s make our own functions
- To start the editor, type ‘edit’

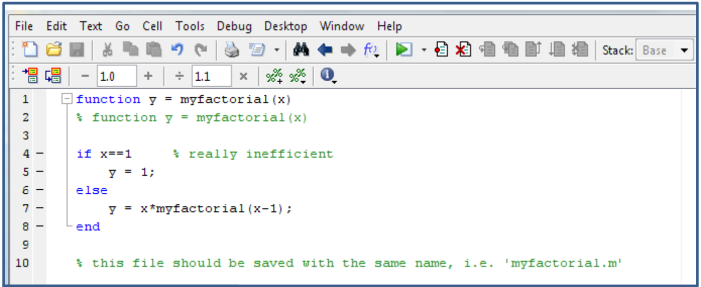


```
function y = myfactorial(x)
% function y = myfactorial(x)

if x==1     % really inefficient
    y = 1;
else
    y = x*myfactorial(x-1);
end

% this file should be saved with the same name, i.e. 'myfactorial.m'
```

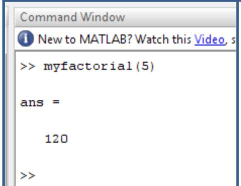
```
>> myfactorial(5)

ans =

   120

>>
```

M-Files and Functions

- Local workspace and Scoping
- To make variables global: global *variable_name*



```
function y = myfactorial(x)
% function y = myfactorial(x)

if x==1     % really inefficient
    y = 1;
else
    y = x*myfactorial(x-1);
end

% this file should be saved with the same name, i.e. 'myfactorial.m'
```



New to MATLAB? Watch this Video, s

```
>> myfactorial(5)

ans =

   120

>>
```

For, if and while loops :

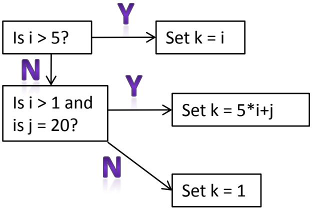
- Several built-in functions for e.g. : `ode23, ode45, ode23s, ode113 etcetera`
- Lot of other ode solver function in external libraries.

For, While, and If

```
for m = 1:100
        num = 1/(m+1)
end
-----------------------------
% find all the powers
% of 2 below 10000
while num < 10000
        num = 2^i;
        v = [v; num];
        i = i+1;
end
-----------------------------
i = 6; j = 21;
if i > 5
        k = i;
elseif (i > 1) & (j == 20)
        k = 5*i+j;
else
        k = 1;
end
```

A for loop

A while loop



• And:	a & b
• Or:	a | b
• Not-equal:	a ~=b
• Equal:	a == b

Solving Differential Equations numerically :

- Several built-in functions for e.g. : `ode23, ode45, ode23s, ode113 etcetera`
- Lot of other ode solver function in external libraries.

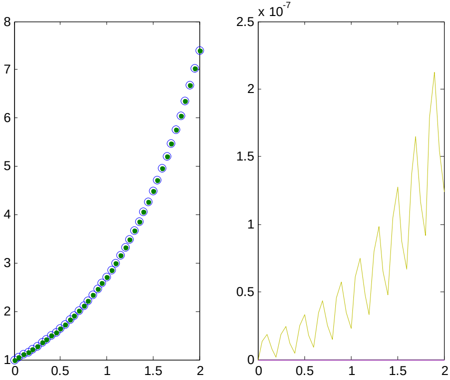
Solving ODEs

- A very simple case: $\frac{dy}{dt} = y(t) \qquad 0 \le t \le 2 \qquad y(0) = 1$

```
function dy = simpleode(t,y)
dy = y;  % save as simpleode.m
```

- Type in command line:

```
[t y] = ode45(@simpleode, [0, 2], [1]);
subplot(1,2,1),plot (t,y,'o',t,exp(t),'.')
subplot(1,2,2),plot(t,(y-exp(t))/exp(t))
```



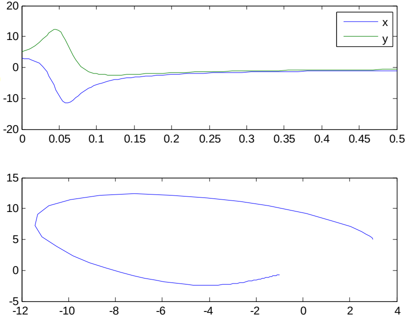
Solving ODEs

- A system of eqns:

$$\frac{dx}{dt} = 2x - y + 3(x^2 - y^2) + 2xy \qquad 0 \le t \le \frac{1}{2}$$

$$\frac{dy}{dt} = x - 3y - 3(x^2 - y^2) + 3xy \qquad y(0) = 3, x(0) = 5$$

```
function xdot = aode(t,y)
% y(1) = x
% y(2) = y
xdot = zeros(2,1); % initialize the xdot vector
xdot = [2*y(1)-y(2)+3*(y(1)^2-y(2)^2)+2*y(1)*y(2)
        y(1)-3*y(2)-3*(y(1)^2-y(2)^2)+3*y(1)*y(2)];
%save as aode.m
```



- Type in command line:

```
[t,y] = ode45(@aode,[0,.5],[3;5]);
subplot(2,1,1),plot(t,y)
subplot(2,1,2),plot(y(:,1),y(:,2)) % plot the phase portrait
```

Solving ODEs

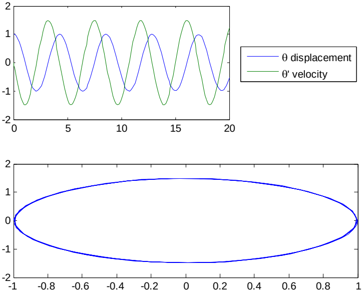
- A second order system: $$\ddot{\theta}+\omega^2\sin\theta=0 \qquad \begin{array}{l}\theta(0)=1\\ \dot{\theta}(0)=0\end{array}$$
- First, convert to a system of two first-order equations, *by hand*.

  let $\begin{array}{l}u_1=\theta\\ u_2=\dot{\theta}\end{array}$, then $\begin{bmatrix}\dot{u}_1\\ \dot{u}_2\end{bmatrix}=\begin{bmatrix}u_2\\ -\omega^2\sin(u_1)\end{bmatrix}$

```
function udot = pend(t,u,omega)
udot = zeros(2,1)
udot = [u(2); -omega^2*sin(u(1))];
%save as pend.m
```

- Type in command line:

```
%omega = 1.56
[t, y] = ode45(@pend,[0 20],[1;0],[],1.56);
subplot(2,1,1),plot(t,y)
subplot(2,1,2),plot(y(:,1),y(:,2)) % plot the phase portrait
```


θ displacement
θ' velocity