MATLAB Tips and Tricks 1/11/07

Note: For all the commands listed here, more details can be found using the help of doc function – see getting help on the next page.

General Tips

	Example
Suppressing output to the screen: use a ';' at the end of a line,	`a = 10^2;`
Adding comments to a program: Start the line with a '%'	`% Comment here`
Special symbols:	
Resetting MATLAB by clearing memory: `clear all`	
Initializing a number from the keyboard: `R=input('Prompt');`	`x=input('Please enter the value for x: ');`
Initializing a string from the keyboard: `R = input('prompt','s')`	`name=input('Please enter your name: ','s');`
Setting the output format: format	`Format compact; % Suppresses extra line-feeds`

Special Symbols

`+`	Addition	`x = 1+2;`
`-`	Subtraction	`x = 1-2;`
`*`	Matrix Multiplication	`x = 1*2;`
`/`	Division	`x = 1/2;`
`^`	Exponentiation	`x = 1^2;`
`pi`	The constant 'pi'	`x = pi;`
`i , j`	Contains the value $\sqrt{-1}$	
`Inf`	Machine infinity	
`NaN`	Not a number, 0/0 will give you this	
`eps`	The smallest difference between two numbers that can be represented by the computer, is machine dependent.	`>> eps` `ans = 2.2204e-16`
`ans`	A special symbol used to store the result of a calculation, it the result is not assigned	`>> 2+3` `ans = 5`

`...`	Continuation of a line, often useful when you have a very long line.	`>> x=2+...`
		`3`
		`x = 5`

Getting Help

There are 4 ways of getting help in MATLAB:

Type `help` or `help` followed by the function name in the command window	`help sin`
The `lookfor` command searches for an exact function name.	`lookfor sin`
Use the help window in MATLAB – the `doc` command brings up a help window	`doc sin`
Use the MATLAB help desk	

Array Manipulation

Initializing variables: `var = expression`	`y=10;` `z=[1 2 3; 4 5 6]; % multi-row matrix`
Generate an evenly sampled vector : `x = xstart:xincrement:xend`	`x=-10:1:10;`
Generate an evenly sampled vector of n values: `x=linspace(xstart,xend,n)`	`x=linspace(-10,10,21);`
Generate a n×n matrix of zeros: `x=zeros(n);`	`x=zeros(10);`
Generate a n×m matrix of zeros: `x=zeros(n,m);`	`x=zeros(10,20);`
Generate a matrix of ones: `x=ones(n,m);`	`x=ones(10,13);`
Generate a n×m identity matrix: `x=eye(n,m);`	

Return the length of an array: `size(arr)`	`>> x=ones(10,13);`
	`>> size(x)`
	`ans = 10  13`

Return the length of the longest dimension of an array: `length(arr)`	`>> x=ones(10,13);`
	`>> length(x)`
	`ans = 13`

Accessing part of an array ':'	`x=[1,2 3; 4 5 6];`
	`>> x(:,1)`
	`ans =`
	`1`
	`4`

Computing the dot product of 2 arrays or squaring a matrix; use '.'	`x.*y` `x.^2`
Make a 2 2D grids for contour and surface plots use: `meshgrid`	`[x,y]= meshgrid(-2:0.1:2,-5:0.1:5);`

Conditionals and looping

The if construct:
`If control_expr_1`
`  Statement 1`
`elseif control_expr_2`
`  Statement 2`
`else`
`  Statement 3`
`end`
	`if x > -10`
	`  y = 10`
	`elseif x == 0`
	`  y = 0`
	`else`
	`  y = -10`
	`end`

The while loop – a block of statements that is repeated indefinitely as long as some conditional is satisfied
`while expression`
`  statement 1`
`  statement 2`
`.`
`.`
`end`
	`while x < 10`
	`  x = x + 1`
	`end`

The for loop – a loop that executes a block of statements a specified number of times
`For index = expr`
`  statement 1`
`  statement 2`
`  .`
`  .`
`end`
	`for n=1:10`
	`  x(n) = n`
	`end`

Plotting

To	Use	examples

Plot 2 vectors.	`plot(x,y)`	`x=-pi:pi/10:pi`
		`y1=sin(x);`
		`plot(x,y1,'r-')`

Add gridlines	`grid` – toggles the grid on/off
	`grid on` – turns the grid on
	`grid off` – turns the grid off
		`x=-pi:pi/10:pi;`
		`y1=sin(x);`
		`plot(x,y1,'r-');`
		`grid on;`

Do multiple overlaid plots on the same axis.	`hold on / hold off`	`x=-pi:pi/10:pi;`
		`y1=sin(x);`
		`y2=cos(x);`
		`plot(x,y1,'r-')`
		`hold on`
		`plot(x,y2,'k--')`
		`hold off`
		`legend('sin x','cos x')`

Do multiple figures.	use `figure(x)` where x is the figure number	`figure(1)`
		`x=-pi:pi/10:pi;`
		`y1=sin(x);`
		`y2=cos(x);`
		`plot(x,y1)`
		`figure(2)`
		`plot(x,y2)`

Do multiple plots.	use `subplot(m,n,p)`	`x=-pi:pi/10:pi;`
		`y1=sin(x);`
		`y2=cos(x);`
		`subplot(2,1,1)`
		`plot(x,y1,'r-')`
		`subplot(2,1,2)`
		`plot(x,y2,'k--')`

Polar plots	`polar(theta,r)`	`theta=0:pi/10:2*pi;`
		`r=theta;`
		`polar(theta,r)`

Adding titles to graphs.	`title('title'),` and `xlabel` and `ylabel` for x and y labels. See 'doc title' for more details, including special formatting	`title('The title')`
		`xlabel('x');`
		`ylabel('y');`

Axis control:	`axis`	`x=-pi:pi/10:pi;`
		`y1=sin(x);`
		`plot(x,y1,'r-');`
		`axis([-pi pi  -1 1]);`

Set tick spaces	The command: set(gca,'Xtick', vector) where vector is specifies the tick location See: http://www.mathworks.com/access/helpdesk/help/techdoc/ref/axes_props.html For y use 'Ytick' and z 'Ztick'	`Set(gca,'Xtick', 0:2;10)` – sets the x grid spacing from 0 to 10 in steps of 2

Do contour or surface plots plots	surf, contour	`[x,y] = meshgrid([-2:0.2:2]);`
		`z = x.*exp(-x.^2-y.^2);`
		`surf(x,y,z)`

Make the aspect ratio equal	axis	`[x,y] = meshgrid([-2:0.2:2]);`
		`z = x.*exp(-x.^2-y.^2);`
		`contour(x,y,z);`
		`axis equal;`