READ THIS BEFORE GOING THROUGH THE REST OF THE MANUAL

Table of Contents


```
Course: Analytical Computational Neuroscience
         Math 635 / Math 430 - Fall 2012
         Fall 2012

TA : Kohitij Kar
```

This is a beginner's guide on how to use MATLAB Go through each step as shown below at your own pace at the MATLAB command window.

The document has been divided into 4 sections:

Section 1 : Vectors and Matrix Manipulations Section 2 : Loop Control and Conditional Statements Section 3 : Functions Section 4 : Graphics

Each section has corresponding exercises. After you have completed each module, use the concepts from the module to solve the following exercises.

Prepared by Kohitij: 8/30/2012

SECTION 1:

Variables

```
a = 1; % initialize a variable 'a' with a numeric value 1
% Using specific mathematical functions.
b = exp(10); % exponential
c = sin(20); %  Sine of argument in radians , also see 'sind', 'asin'

% Tip 1: Always use variables with meaningful names. e.g.
distance = 10; % in meters
time = 15; %   in seconds
speed = distance / time; % in meters/sec
```

Vectors / Arrays

```
% row vectors (voltage and resistance across 5 different membranes)
voltageR = [ 10 12 13 14 15 ];        % in millivolts
resistanceR     = [ 15 18 17 10 20 ]; % in megaohms
currentR    = voltageR./resistanceR; % From Ohms law, i = V/R

% column vectors

voltageC = [10;12;13;14;15]; % It is the same as voltageC = voltageR'; (using the
resistanceC     = [15;18;17;10;20];
currentC    = voltageC./resistanceC;


% Different ways of initializing vectors with values

vector1 = 1:1:10;           % numbers from 1 to 10 in increments of 1
vector2 = linspace(1,51,10); % 10 linearly spaced numbers between 1 and 51
vector3 = rand(1,10);  % row vector of random numbers between 0 and 1
vector4 = zeros(10,1); % column vector of zeros
vector5 = ones(10,1); % column vector
% Tip 2: Use 'help' function , e.g., help rand , to know more about the functions
```

Matrices

```
% Initialize  a MxN matrix,(M = no. of rows, N = no. of columns) with zeros
% say M = 5, N = 4
matrixA  = zeros(5,4);
size(matrixA) % To check size
% Transpose of the matrix
aTranspose  = matrixA';

% Matrix multiplication
% If a matrix of size MxN is multipled with a matrix of size NxP, the
% resulting matrix is of the size MxP
% Always make sure that no. of columns in the first one is equal to the
% number of rows in the second one
```

```
matrixA = rand(4,5);
matrixB = rand(5,8);
multAB = matrixA*matrixB;
```

```
ans =

     5     4
```

SOME REALLY IMPORTANT AND INTERESTING FUNCTIONS:

```
% reshape function
%(goes all rows per column )

% e.g. Say you have a matrix X

X = rand(6,4);
modifiedX = reshape(X,8,3); % also see help reshape


% repmat function

Y = [ 1 2];
repY = repmat(Y,2,2);
```

EXERCISE 1:

```
% 1. Create a 6 by 4 matrix whose first three rows have elements equal to 1,
% and the last three rows elements equal to 2. (HINT : see help ones)
% 2. Construct an 8 by 8 matrix, consisting of 2 by 2 submatrices with the entries
% 3. A cyclist drives 15km/h. Starting at time zero at position zero, where
% is she after 1,4, 7, 10 and 13 hours? [Repeat the Exercise for Functions]
% 4. A car makes a three-stage journey. In the first stage it drives 100km in 2 hr
%  then 150km in 1.5 hrs, then 200km in 3hrs.
%   What were the average speeds over the three stages of the journey?
```

SECTION 2

IF - ELSE STATEMENT

```
% Let's say you have a variable 'cellVoltage'. Now whenever the value of this
% variable crosses a certain level (say -40, 'thresholdVoltage' ) you need to set
% otherwise just increment it by 1.
% These kind of operations can be done by using the if-else statement.

cellVoltage = -39;
thresholdVoltage= -40;
```

```
 if(cellVoltage>=thresholdVoltage)
    cellVoltage=55;
 else
    cellVoltage=cellVoltage+1;
 end

 % note that the cell volage became 55.

% Now set the cell voltage to -42 and try again.

cellVoltage = -42;
thresholdVoltage= -40;

 if(cellVoltage>=thresholdVoltage)
    cellVoltage=55;
 else
    cellVoltage=cellVoltage+1;
 end


 % Note that the cellVoltage increased by 1 this time.
```

WHILE LOOP

```
% A part of the same operation as conducted above can be also implemented
% by using the while loop


while(cellVoltage<thresholdVoltage)
    cellVoltage = cellVoltage +1;
end

% Manually set different value of threshold and cell voltage and practise
% using the while and if-else statemnts.

% Another example of while loop

% Let spikes be a vector containing the spike rates at certain times.
% spike(1) is the spikerate at time 1, spike(2) at time 2 etc.
% To determine the first time at which the spike rate drops below 10,
% a while-loop could be used:

spikes=[100 210 280 150 100 80 70 30 20 10 5 5 1 2 3];
time=1;
while spikes(time)>=10
time=time+1;
end
belowThreshold = time

% Tip 3: Note that here too a vectorized approach is possible, and faster:
belowThreshold = find(spikes<10,1)
```

```
belowThreshold =

    11


belowThreshold =

    11
```

FOR LOOP

```
% The for-loop is a useful looping construct whenever you want to perform
% a certain set of operations for a number of times that is known in advance.
% To do this, you enclose the lines of code that has to be done multiple
% times with the for..end construct.


% Asumme you have 10 time and voltage values. and you need to create a
% vector that contains the times at which the voltage was below 5

voltage = [ 10 20 3 45 1 20 4 25 36 1 ];
time =    [ 1 2 3 4 5 6 7 8 9 10 ];  % it can be also initialised as time = 1:10;
data.voltage = voltage;
data.time = time;


Cntr = 1; % This is a counter for the required time matrix that we need

for i = 1:10
    if(voltage(i)<5);
        reqTime(Cntr) = time(i);
        Cntr = Cntr+1;
    end
end


% The objective of MATLAB Programming is always to minimize the usage of
% for loop. The same above operation could be done in the following way.

reqTime = time(voltage<5);
```

EXERCISE

```
% 1. Write a loop for 1000 iterations that draws random numbers (use the function
%    Exit the loop when the random number lies between 0.5 and 0.55 (see help brea
%    Count the number of draws done before the loop exited.
```

SECTION 3

Functions

```
% As functions form the backbone of structured programming,
% let us look at an example function that determines the surface and
% circumference of a circle, given its radius.

% [USE SEPARATE m-file]

% function [surface,circumference] = circledata(radius)
% % function [surface,circumference] = circledata(radius)
% % A function to determine the surface and circumference
% % of a circle, given its radius.
% %
% % INPUT
% % radius The radius of the circle.
% %
% % OUTPUT
% %surface The surface of the circle.
% % circumference The circumference of the circle.
% %
%
% surface = pi*radius^2;
% circumference = 2*pi*radius;
% end

% The first line defines the name of the function 'circledata',
% its input argument 'radius' and the variables it returns 'surface' and
% 'circumference'. Save this function in an M-file called 'circledata.m'

% Now try :

help circleData
```

```
  function [surface,circumference] = circledata(radius)
  A function to determine the surface and circumference
  of a circle, given its radius.

  INPUT
  radius The radius of the circle.

  OUTPUT
 surface The surface of the circle.
  circumference The circumference of the circle.
```

EXERCISE 3

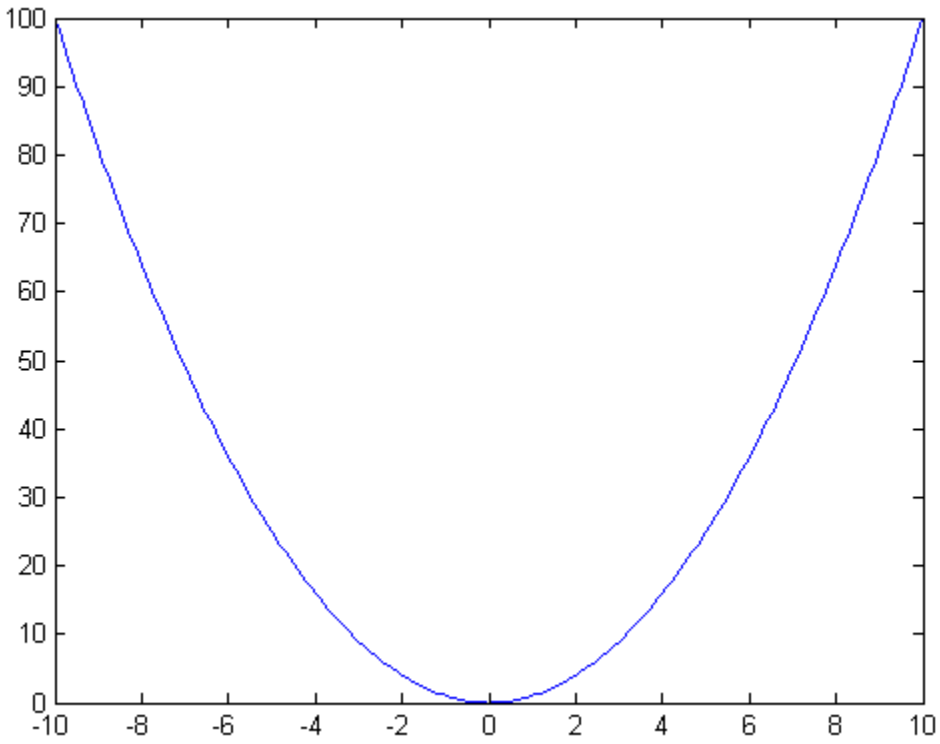
```
% 1. Write a function called mstd that takes a matrix as its input argument
```

```
%    and determines mean and standard deviation of each row.
%    (standard dev is the second output argument, the mean is the first).
```

SECTION 4

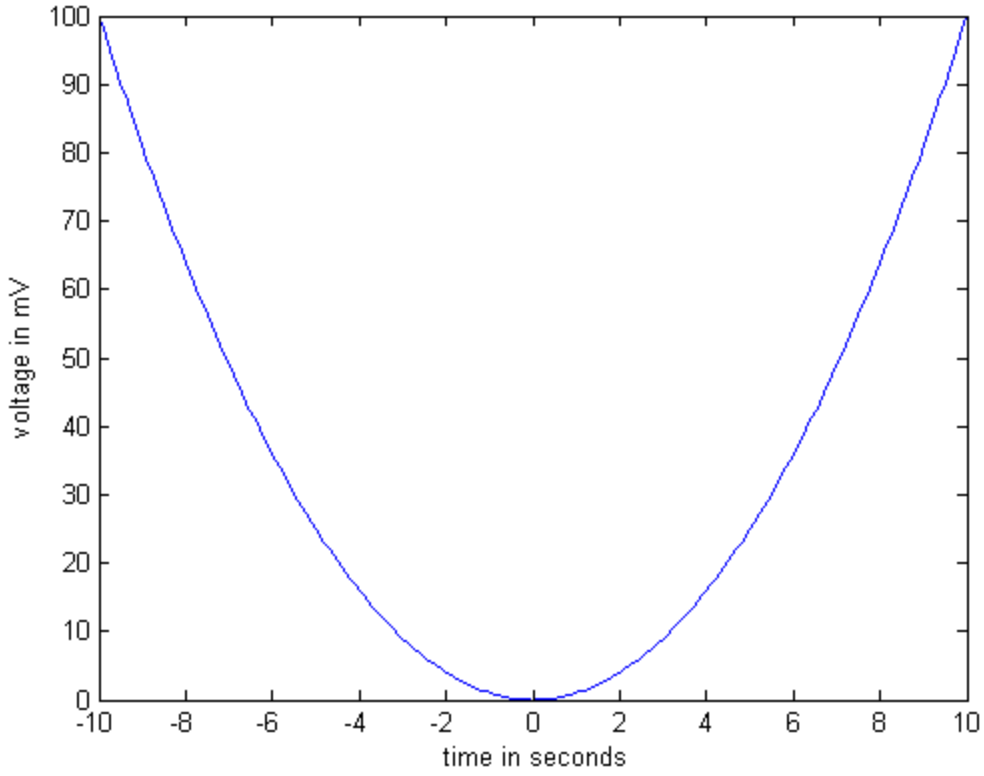
MATLAB Plot function

```
time = -10:0.1:10;
voltage = time.^2;
figure;
plot(time,voltage)
```



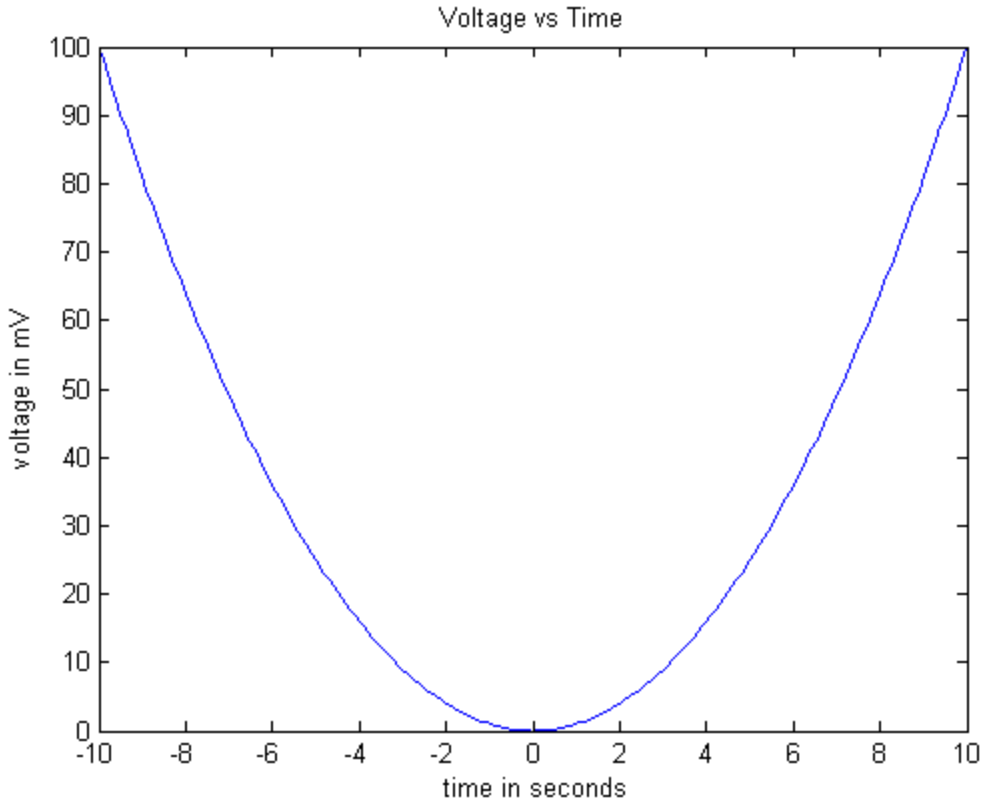
Add x and y labels

```
xlabel(' time in seconds');
ylabel(' voltage in mV ');
```



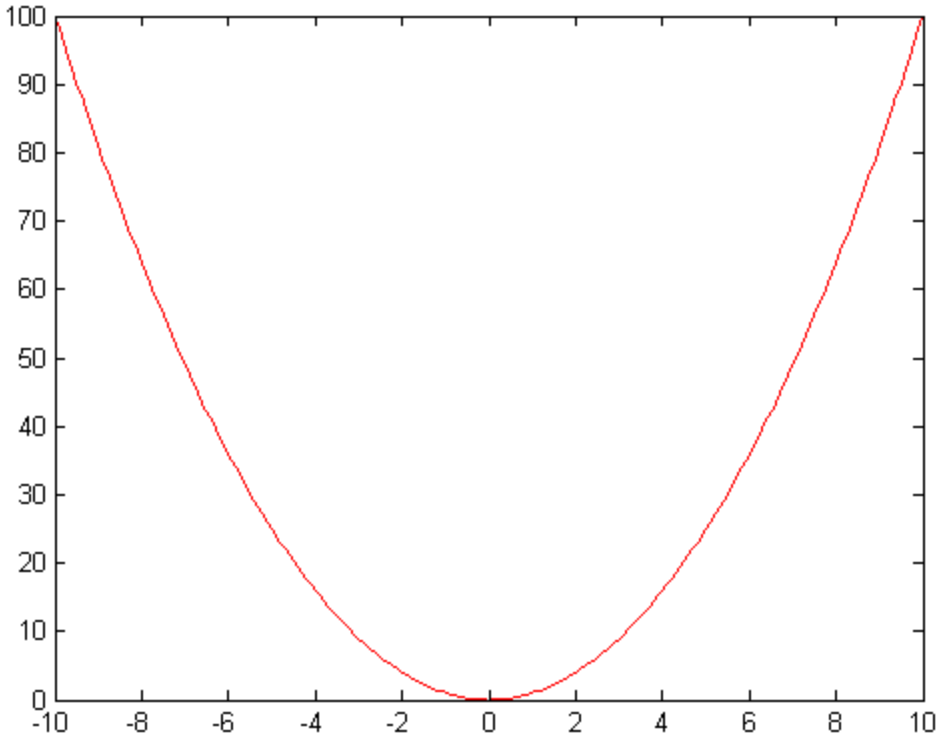
Add title to the plot

```
title(' Voltage vs Time ')
```



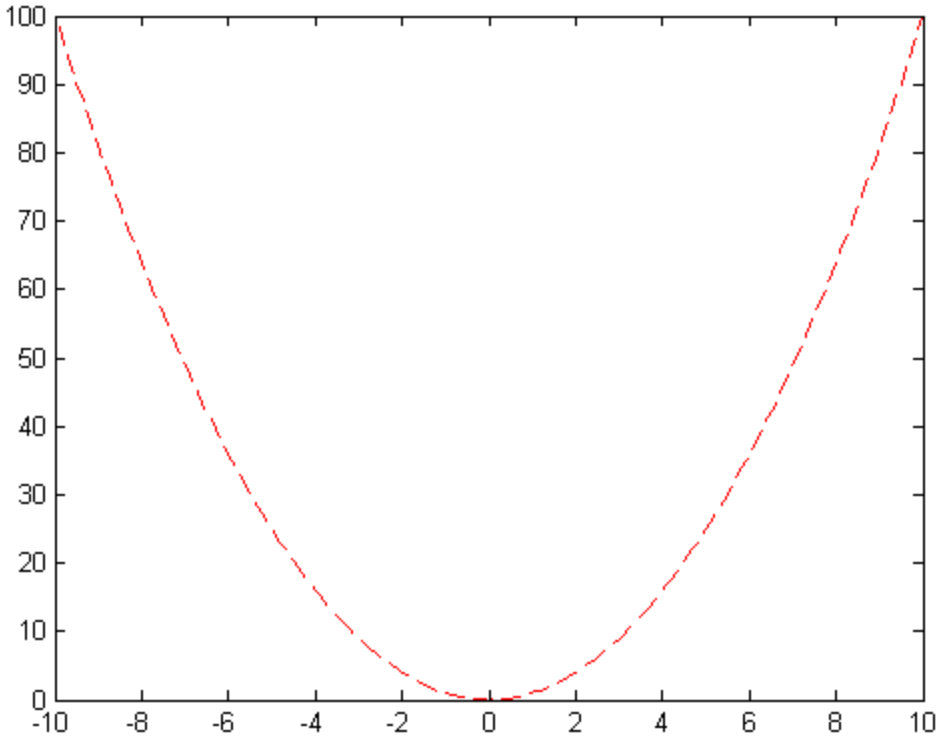
plot the graph in red

```
plot(time,voltage,'r')
```



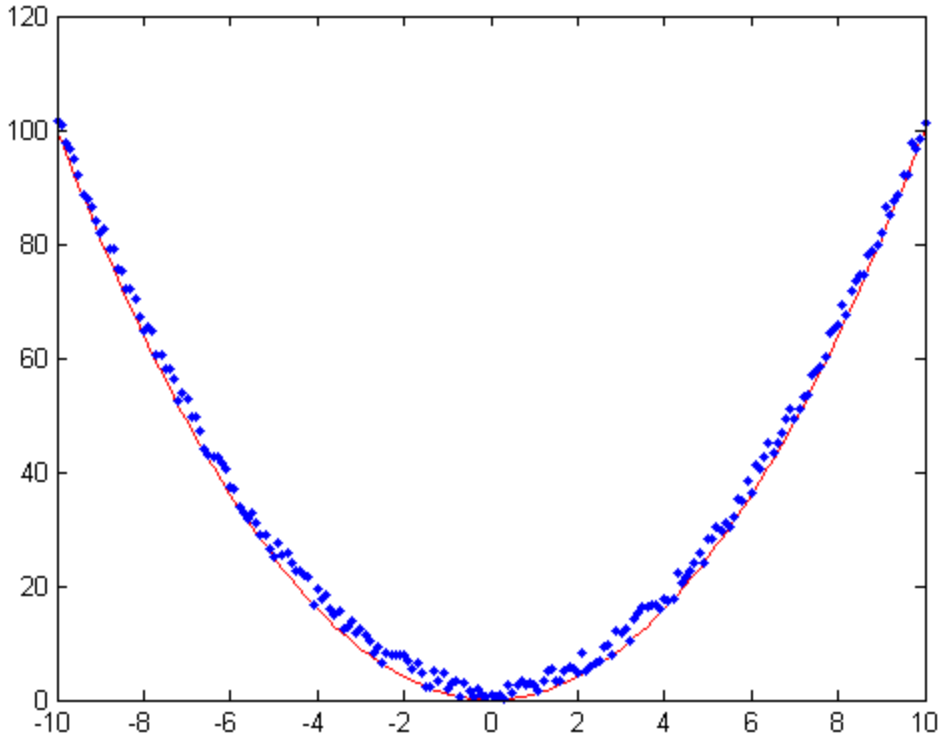
plot the graph as red dashed lines

```
plot(time,voltage,'r--')
```



plot multiples curves on a single plot (using hold on)

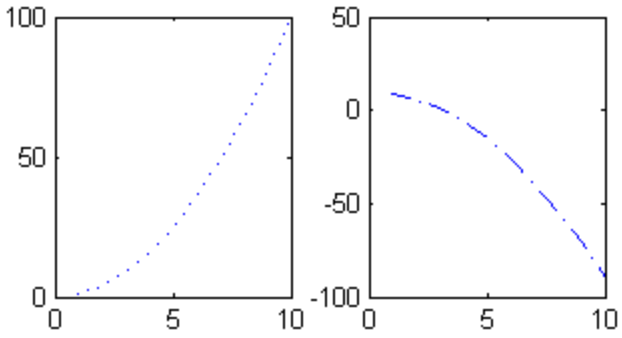
```
noise = 4*rand(1,length(time));
noisyVoltage = voltage + noise;
plot(time,voltage,'r')
hold on;
plot(time,noisyVoltage,'.') % plotted as dots
```

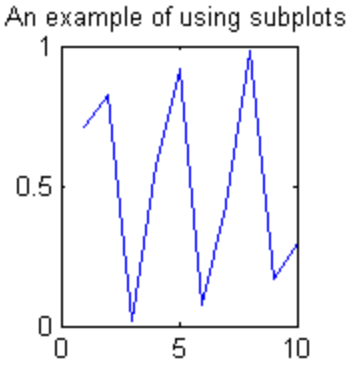


SUBPLOT

```
% Figures can be enhanced by the use of subplot which allows you to display
% multiple subplots in a single figure. The command subplot(2,3,1) for instance,
% generates a figure with 2 rows with 3 axes each and it makes the first axis
% the current one. I.e. all plotting commands will be directed towards the first s

subplot(2,3,1)
plot(1:10,(1:10).^2,':b')
subplot(2,3,2)
plot(1:10,10 - (1:10).^2,'-.b')
subplot(2,3,6)
plot(1:10,sin (1:10).^2,'-b')
title 'An example of using subplots'
```





Other functions for plotting

```
% See 'bar', 'hist', 'line' etc
```

EXERCISE 4

```
% 1. Create a vector that contains membrane voltage of a cell that spikes
% according to a leaky integrate and fire neuron model.
% vrest = -70 mV
% vthr = -50  mV
% vreset = -70 mV
% tau_m = 20   ms
% Rm = 1 Mohm
% Use input current : 10 nA , 20 nA, 30 , 40 nA
% time of simulation = 0 to 1 s
% Plot the voltage trace and the spike train on the same graph
% Write this first as a script.
% Then: Try and make a function that takes in all the parameters as input
% argumnets and generates the plots. This way you can change the input
% currents and check the effect
```

Published with MATLAB® 7.14