

**Armstrong Atlantic State University**  
**Engineering Studies**  
**MATLAB Marina – 2D Plotting Primer**

**Prerequisites**

The 2D Plotting Primer assumes knowledge of the MATLAB IDE, MATLAB help, arithmetic operations, built in functions, scripts, variables, arrays, logic expressions, conditional structures, iteration, functions, debugging, characters and strings, cell arrays, structures, and file input and output. Material on these topics is covered in the MATLAB Marina Introduction to MATLAB module, MATLAB Marina Variables module, MATLAB Marina Arrays module, MATLAB Marina Logic Expressions module, MATLAB Marina Conditional Structures module, MATLAB Marina Iteration module, MATLAB Marina Functions module, MATLAB Marina debugging module, MATLAB Marina Character and Strings module, MATLAB Marina Cell Arrays module, MATLAB Marina Structures module, and MATLAB Marina File Input and Output module.

**Learning Objectives**

1. Be able to generate and appropriately annotate 2D plots using MATLAB.

**Terms**

figure window, 2D plot, parametric plot

**MATLAB Functions, Keywords, and Operators**

figure, plot, subplot, xlabel, ylabel, title, legend, gtext, grid, axis, clf, close

**Figure Windows**

MATLAB plots are displayed in figure windows (figure containers). The MATLAB function `figure` creates a new figure window which can be referenced by its figure number and `figure(number)` creates a figure with the specified figure number or makes the specified figure the current active figure.

The MATLAB function `clf` clears the current figure window and `clf(number)` clears the specified figure window. The MATLAB function `close` closes the current figure window, `close(number)` closes the specified figure window, and `close all` closes all open figure windows.

**Two Dimensional Plotting (2D Plotting)**

Two-dimensional plots can be done using the MATLAB `plot` function. The statement `plot(x,y)` creates a two-dimensional plot using the vector `x` as the x-axis values and the vector `y` as the y-axis values. The vectors `x` and `y` must be the same length. If `plot` is called with a single argument, `plot(y)`, then the values of `y` are plotted versus their vector index. Multiple plots can be placed on the same axes, `plot(t,x1,t,x2)` plots `t` vs. `x1` and `t` vs. `x2` on the same axes. The scale of each axis and the axes ticks are added automatically by MATLAB. Plots

are placed in the active figure window. If there is not an active figure window, the plot command will create and open a figure window (starting with figure 1).

The MATLAB program of Figure 1a generates the plot of Figure 1b,  $f(t) = 1 - e^{-t}$  for  $0 \leq t \leq 1$  seconds. Plots can be titled and labeled using the `title`, `xlabel`, `ylabel`, and `text` functions. The `title` function places a text string centered on the top of the figure. The `xlabel` and `ylabel` functions place labels on the x and y axis respectively. The `text` function allows one to place a text string anywhere in the figure and the `gtext` function allows one to place a text string using the mouse cursor. MATLAB also has functions: `grid` to place a grid on the plot, `axis` to specify the x and y axis limits, and `legend` to create a legend.

```
% Plot of 1-exp(-t), t = 0 to 1 sec
t = (0 : 0.01 : 1);
f = 1 - exp(-t);
figure(1)
plot(t, f)
xlabel('t (sec)');
ylabel('f(t) = 1 - exp(-t)');
grid
```

Figure 1a, MATLAB Program for 2D Plot

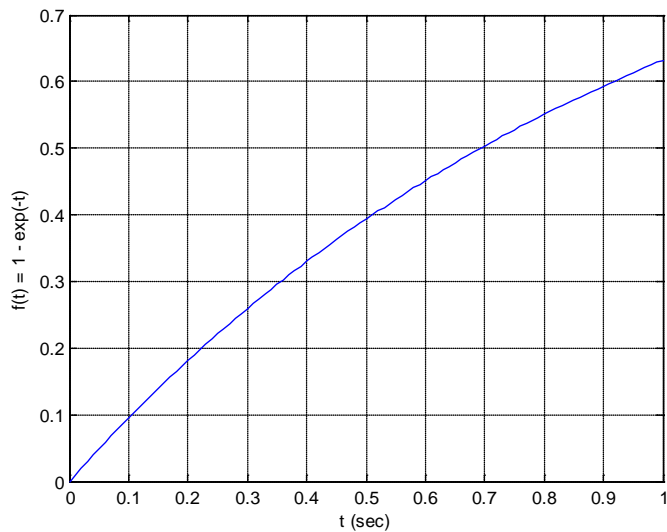


Figure 1b, 2-D Plot

The figure properties can also be modified through the figure Insert and Edit menus. The insert menu has selections corresponding to the MATLAB figure functions as well as options to label the figure with shapes, lines and arrows. The edit menu allows axis ranges, labels, titles to be set through the Axes properties selection. Figures can be copied and pasted in Microsoft Word documents via the Copy Figure selection.

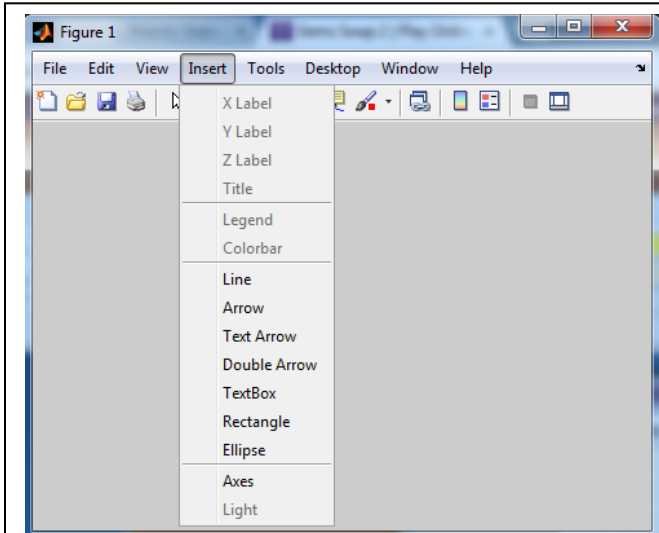


Figure 2a, Figure Insert Menu

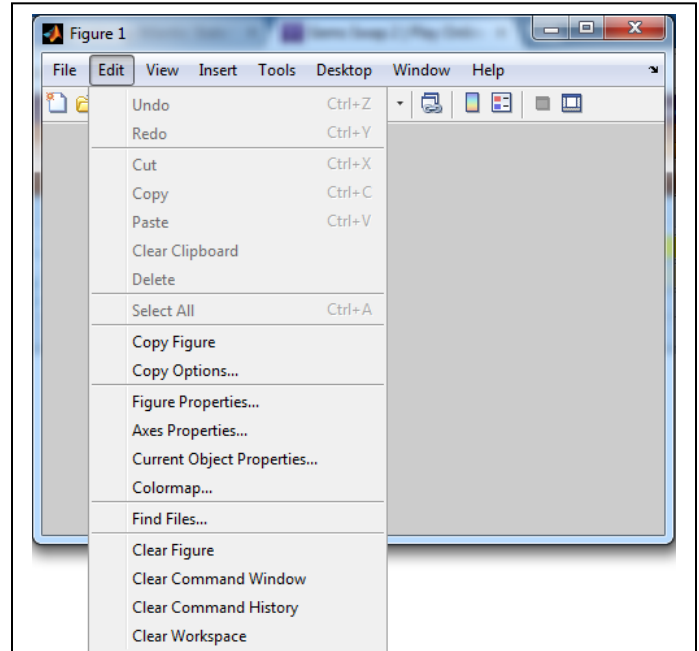


Figure 2b, Figure Edit Menu

The MATLAB program of Figure 4 generates the plots of Figures 3a - 3c. Figure 3a shows the plot of  $s_x$  versus  $s_y$  generated using the MATLAB `plot` function. The line width was specified to be 2 (twice as thick as the default). The plot of Figure 3a is a 2D parametric plot. The x and y axis variables are functions of a separate independent variable  $t$ .

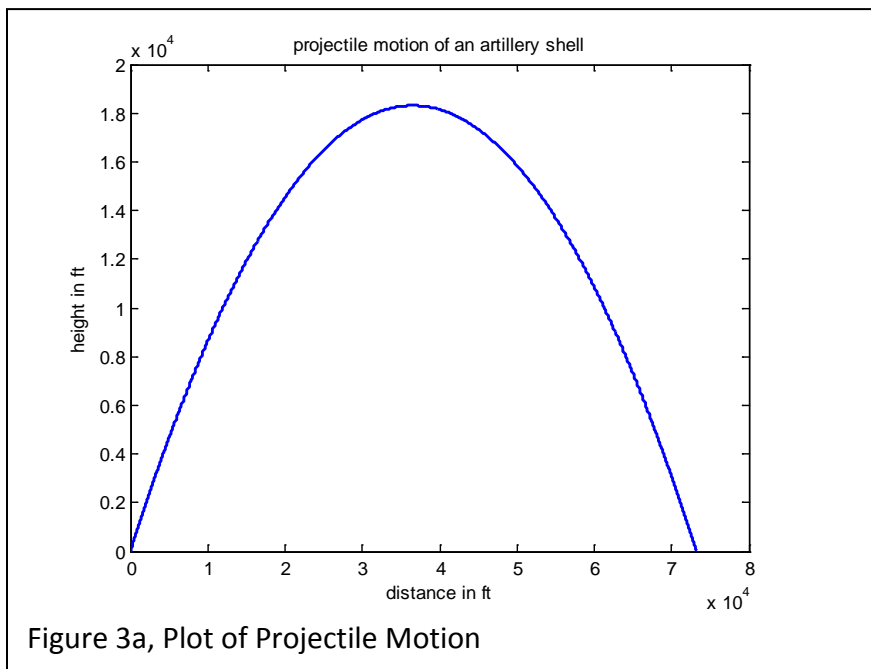


Figure 3a, Plot of Projectile Motion

Figure 3b shows multiple plots on the same axes, notice that MATLAB automatically chooses a new color for each new plot on the same axes (blue for first, green for second). The `grid` on

statement gives the background grid. The `grid off` statement will turn off the background grid.

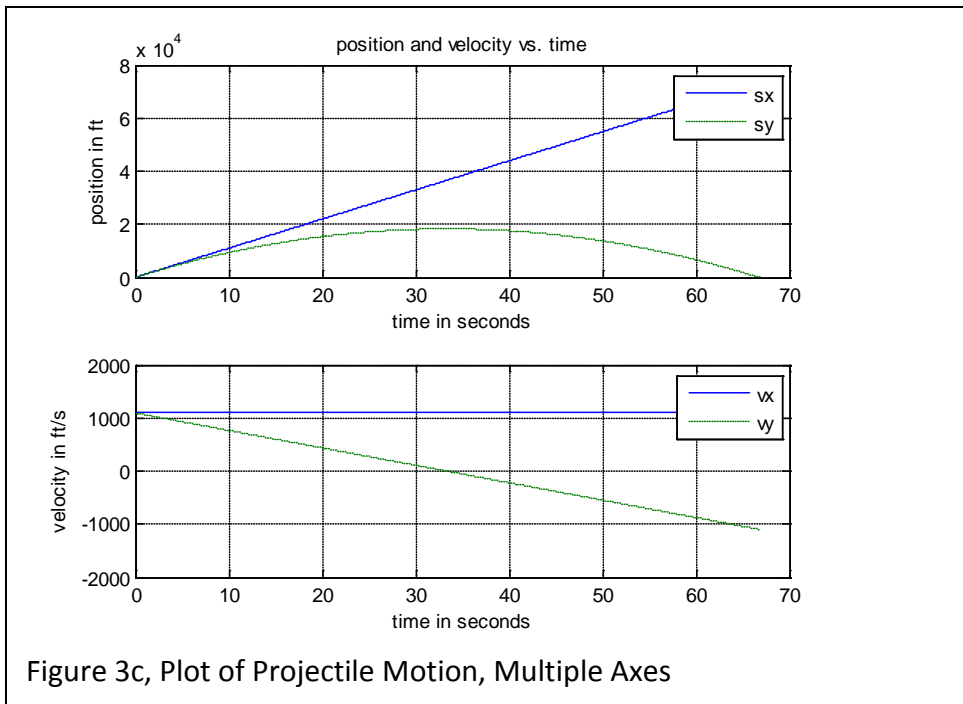
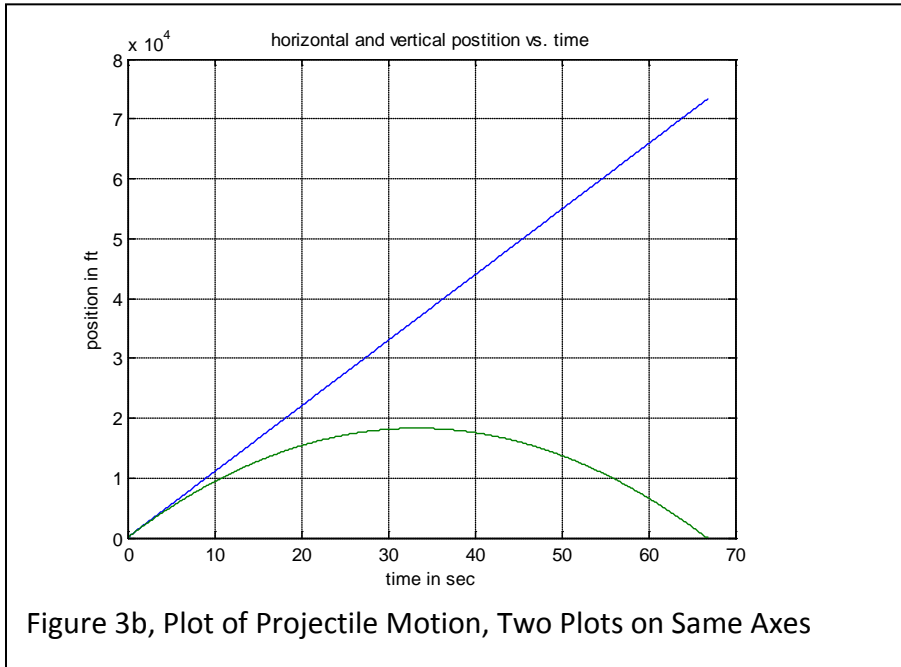


Figure 3c shows two plots on separate axes in the same figure window. Multiple axes in a figure can be created using the `subplot` function; `subplot(m,n,p)` breaks the figure window up into a  $m$  by  $n$  grid and specifies the  $p$ th axes for the plot. The axes are numbered from left to right along each row starting with the top row, then the second row, etc. For example, `subplot(3,1,1)` creates a 3 row by 1 column grid and specifies that the current plot should

be placed in the first row, `subplot(3,1,2)` would specify the second row, and `subplot(3,1,3)` would specify the third row. The legend is created with the MATLAB `legend` function.

The MATLAB `ginput` function allows one to get points from the current axes. The cursor is positioned via the mouse and points are returned when the mouse button or any key on the keyboard is hit (except return). For example, `[x,y] = ginput(2)` gets two points from the current axes and returns two vectors containing the x and y coordinates of the points. The statement `[x,y] = ginput` gets an unlimited number of points until the return key is pressed.

```
% 2D projectile motion
sx0 = 0.0;
sy0 = 0.0;
v0 = 1550.20; % muzzle velocity in ft/s
gx = 0.0;
gy = -32.81; % gravity in ft/s^2
% get the muzzle angle from the user and convert to radians
angle = input('Input the muzzle angle in degrees: ');
theta = angle*pi/180; % barrel angle in radians

% calculate the x and y components of muzzle velocity
vx0 = v0*cos(theta);
vy0 = v0*sin(theta);
% calculate the range for the shell and time to max range
range = -v0^2*sin(2*theta)/gy;
tmax = range/vx0;
t = [0:0.01:tmax];

% calculate the position and velocity of the shell
vx = vx0 + gx*t;
vy = vy0 + gy*t;
sx = sx0 + vx0*t + 0.5*gx*t.*t;
sy = sy0 + vy0*t + 0.5*gy*t.*t;

% plot the projectile motion with a line width of 2
figure(2)
plot(sx,sy,'Linewidth',2);
xlabel('distance in ft');
ylabel('height in ft');
title('projectile motion of an artillery shell');
```

Figure 4 (First Part), MATLAB Program for Plots in Figures 3a – 3c

```

% multiple plots on same graph
figure(3)
plot(t,sx,t,sy);
title('horizontal and vertical postition vs. time');
xlabel('time in sec');
ylabel('position in ft');
grid on;

% multiple plots in same figure window (different graphs)
figure(4)
subplot(2,1,1),plot(t,sx,'-',t,sy,':');
subplot(2,1,1),xlabel('time in seconds');
subplot(2,1,1),ylabel('position in ft');
subplot(2,1,1),grid on;
subplot(2,1,2),plot(t,vx,'-',t,vy,':');
subplot(2,1,2),xlabel('time in seconds');
subplot(2,1,2),ylabel('velocity in ft/s');
subplot(2,1,2),grid on;
subplot(2,1,1),title('position and velocity vs. time');
subplot(2,1,1),legend('sx','sy');
subplot(2,1,2),legend('vx','vy');

```

Figure 4 (Second Part), MATLAB Program for Plots in Figures 3a – 3c

### Other Useful Plotting Functions

Other MATLAB 2D plotting functions are `stem`, `semilogx`, `semilogy`, `loglog`, `bar`, `hist`, `pie`, and `polar`. Other functions that are useful with plots are `hold` and `view`. See MATLAB's help for more info on the different plot types.

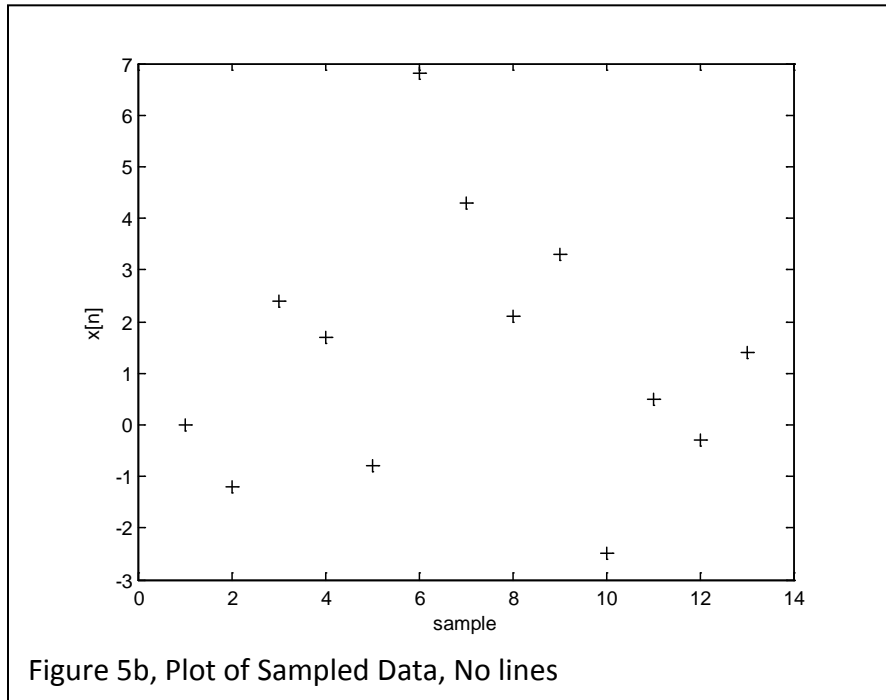
MATLAB allows one to specify the point style, line color, and line style of plots via an optional control string argument of the plot function. The script of Figure 5a generates the plot of Figure 5b. In this case only the data points are plotted with point style of + and point color black. There are no lines joining the data points (no interpolation between points). See MATLAB's help on the different point and line styles of the plot function.

```

% Plot of Sampled Data
xn = [0.0, -1.2, 2.4, 1.7, -0.8, 6.8, 4.3, 2.1, 3.3, -2.5,
0.5, -0.3, 1.4];
figure(2)
plot(xn,'k+')
xlabel('sample');
ylabel('x[n]');

```

Figure 5a, Program for Figure 5b



### Other Useful Annotation Operations

The Figure window toolbar provides many other options for viewing annotating plots. The Figure Insert menu has options for inserting lines, arrows, and textboxes along with axis labels, titles, legends, and shapes.

The Figure window toolbar has icons for zooming, panning, rotating, and various annotation operations. Selecting the toolbar magnifying glass icon with + or – inside allows one to zoom in or out on a region of a plot. Selecting the toolbar hand icon allows one to pan to the left/right and up/down of a plot. The toolbar also has an insert legend icon which allows one to insert a legend and a data cursor icon which allows one to annotate the plot with the values at the location of the data cursor. Keep in mind that any annotation done via the toolbar or menu is for the current Figure window and would need to be redone whenever the plot was generated.

Last modified Monday, September 30, 2013



This work by Thomas Murphy is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](https://creativecommons.org/licenses/by-nc-nd/3.0/).