Matlab Users Group

Plotting Intro

Gordon Parker

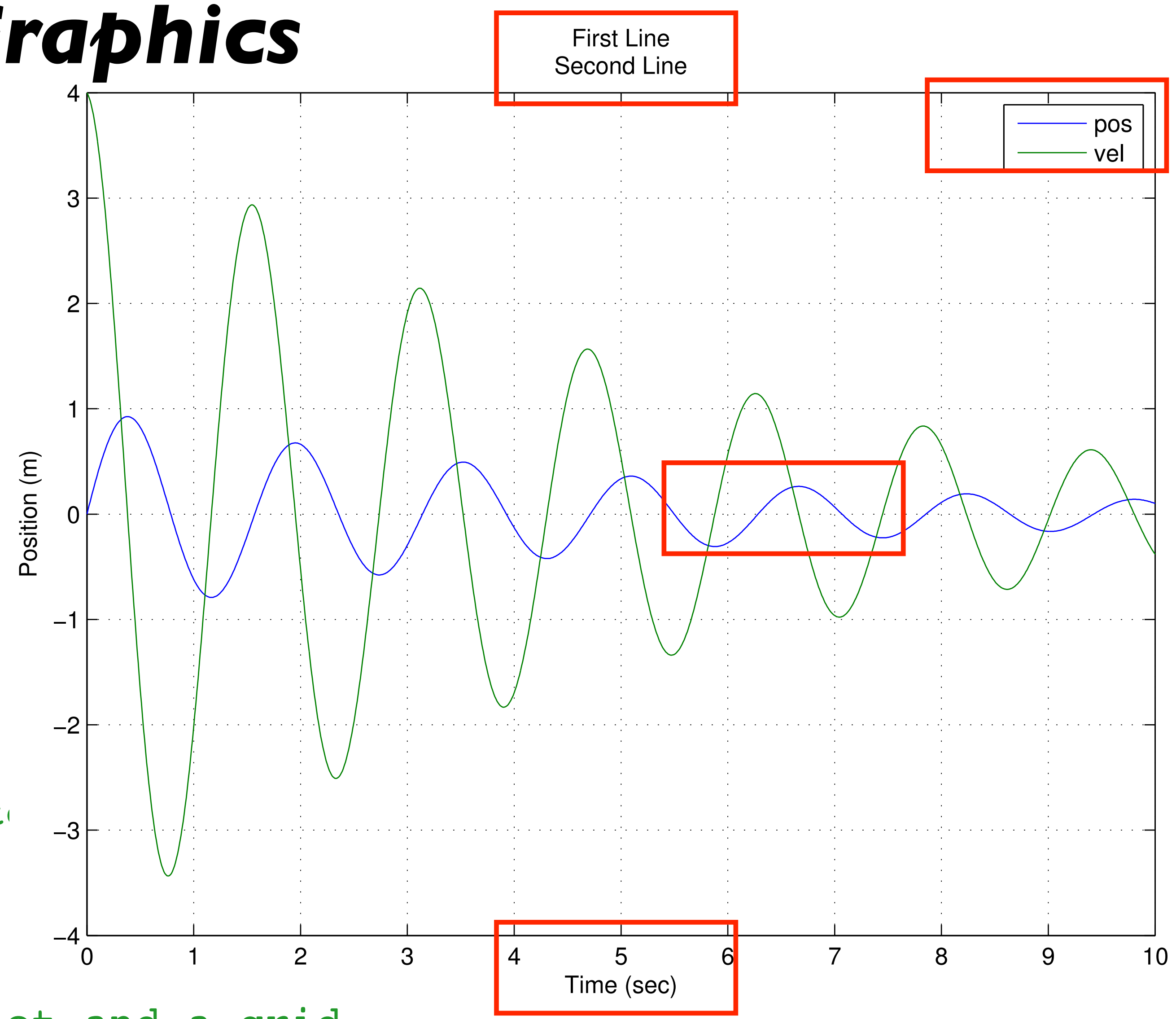
Michigan Tech

Outline

- Basics
- Handle Graphics
- Workflow
- Cool Downloads

Basic Example - No Handle Graphics

Unacceptable for presentations
-- too small and too skinny...



```
%% Plot1.m: Basic Plot Example
% This example is a basic, 2-line plot with
% no handle graphics.

t = 0:.01:10;                     % time vect
pos = sin(4*t).*exp(-.2*t);       % position
vel = 4*cos(4*t).*exp(-.2*t) +...
  -.2*sin(4*t).*exp(-.2*t);       % speed
plot(t,pos,t,vel);grid;           % make a plot and a grid
xlabel('Time (sec)');             % label the x-axis
ylabel('Position (m)');           % label the y-axis
mytitle{1} = 'First Line';        % make a 2-line title cell
mytitle{2} = 'Second Line';
title(mytitle);                   % put on the title
legend('pos','vel',...
  'Location','NorthEast');        % put on a legend
print -depsc fig1.eps             % save the plot as a color .eps file
%publish(plot1.m,'html');         % make a cool document, execute this at
                                  % the command line
```

Plot Colors

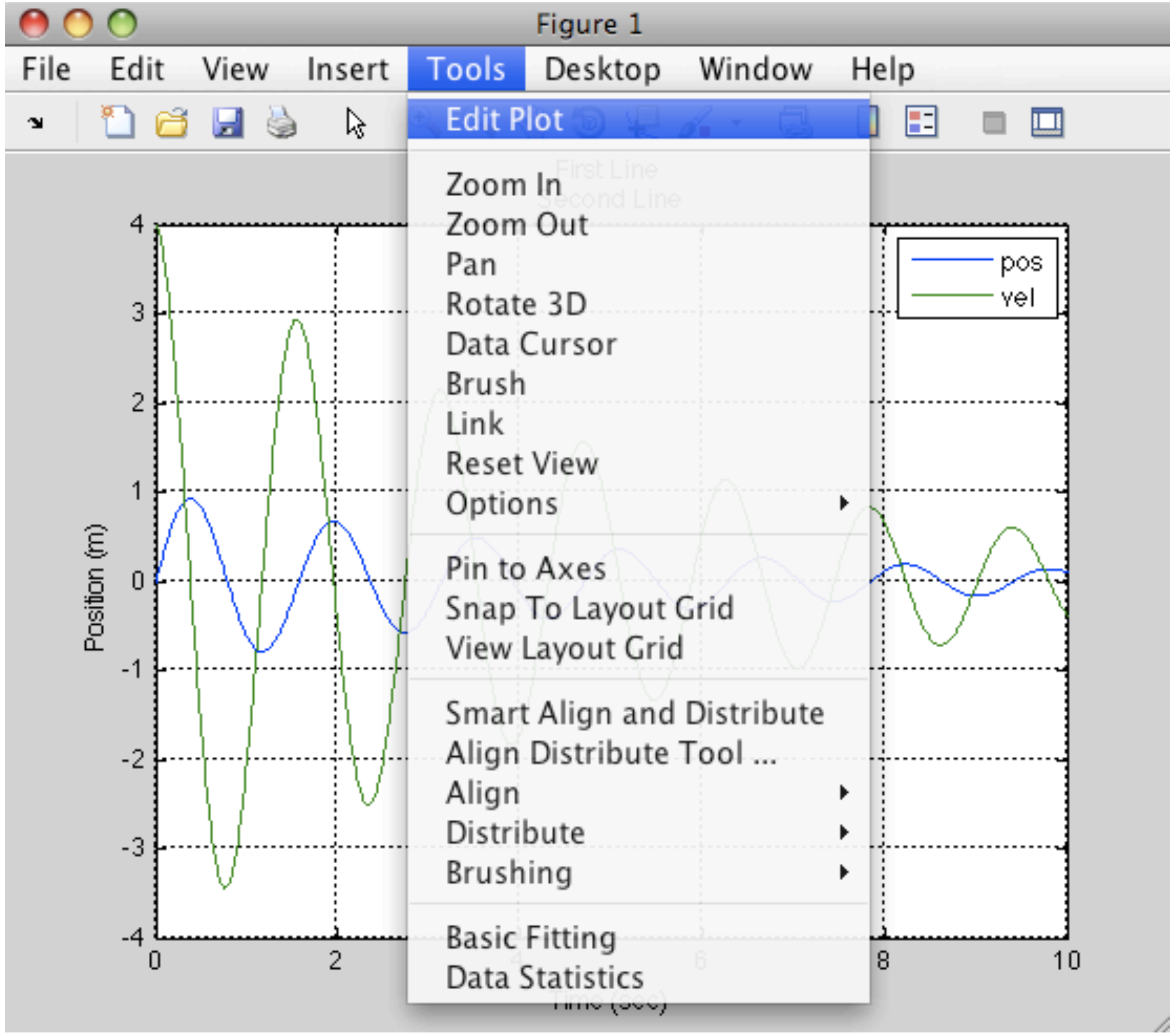
Default MATLAB **plot** color sequence

```
blue    [  0   0 255] / 255
green   [  0 128   0] / 255
red     [255   0   0] / 255
cyan    [  0 191 191] / 255
magenta [191   0 191] / 255
yellow  [191 191   0] / 255
black   [ 64  64  64] / 255
```

Shortcut color notation accessible from **plot**

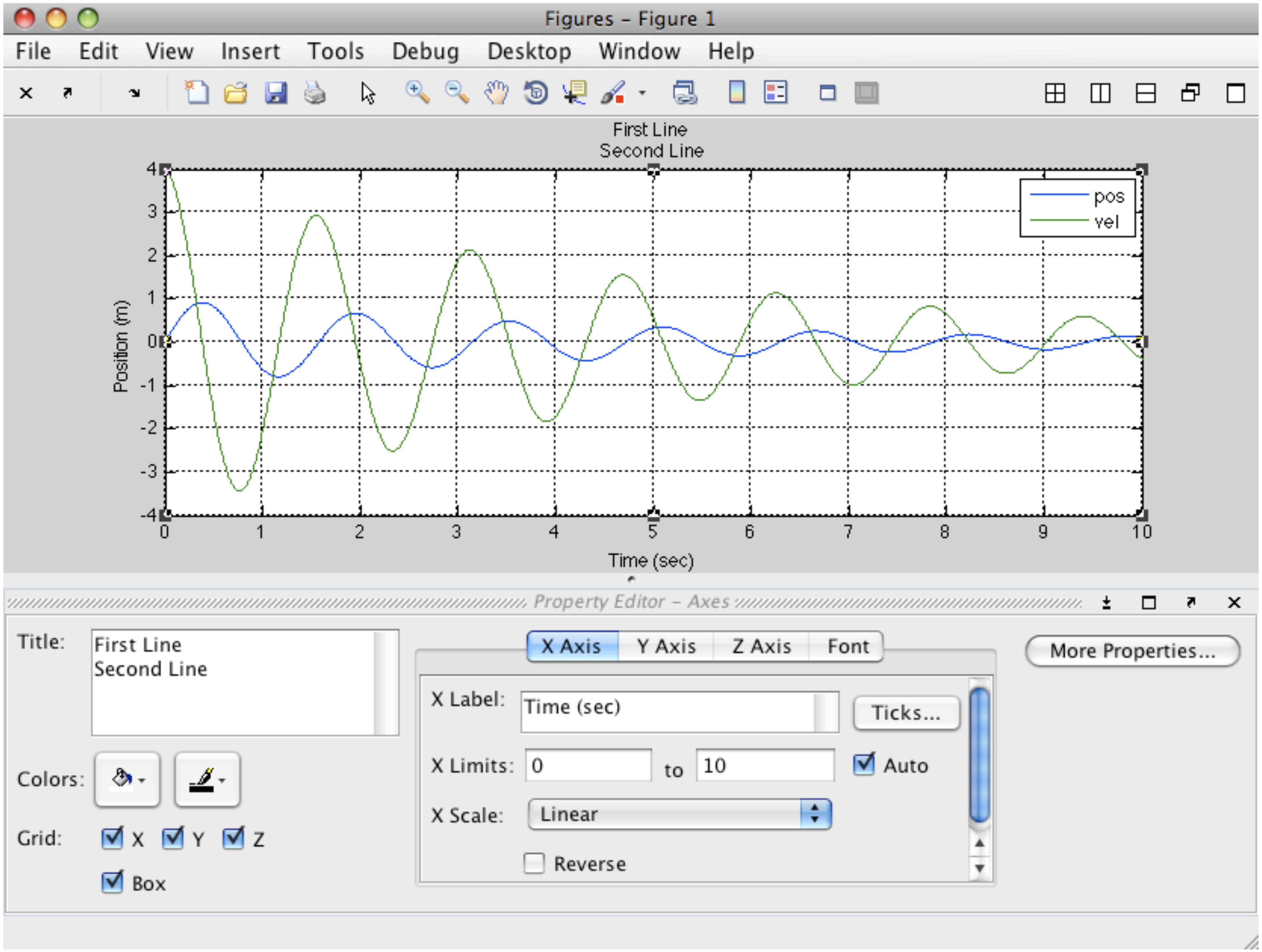
```
‘b’     [  0   0 255] / 255
‘g’     [  0 255   0] / 255
‘r’     [255   0   0] / 255
‘c’     [  0 255 255] / 255
‘m’     [255   0 255] / 255
‘y’     [255 255   0] / 255
‘k’     [  0   0   0] / 255
```

Basic Example - Edit Plot



Michigan Tech

Basic Example - Edit Plot



Title: First Line
Second Line

Colors:

Grid: ✓ X ✓ Y ✓ Z

✓ Box

X Label: Time (sec)

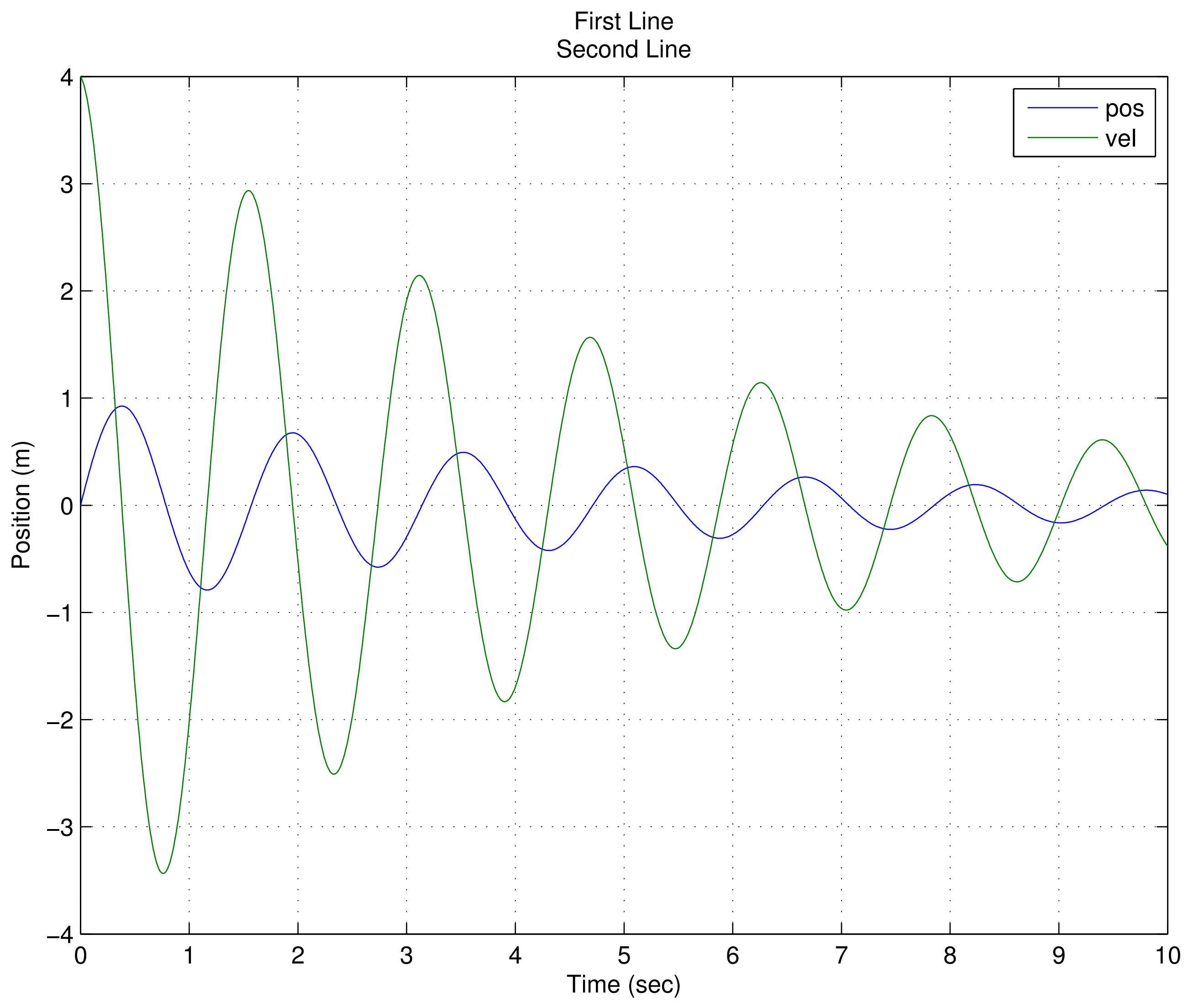
X Limits: 0 to 10 ✓ Auto

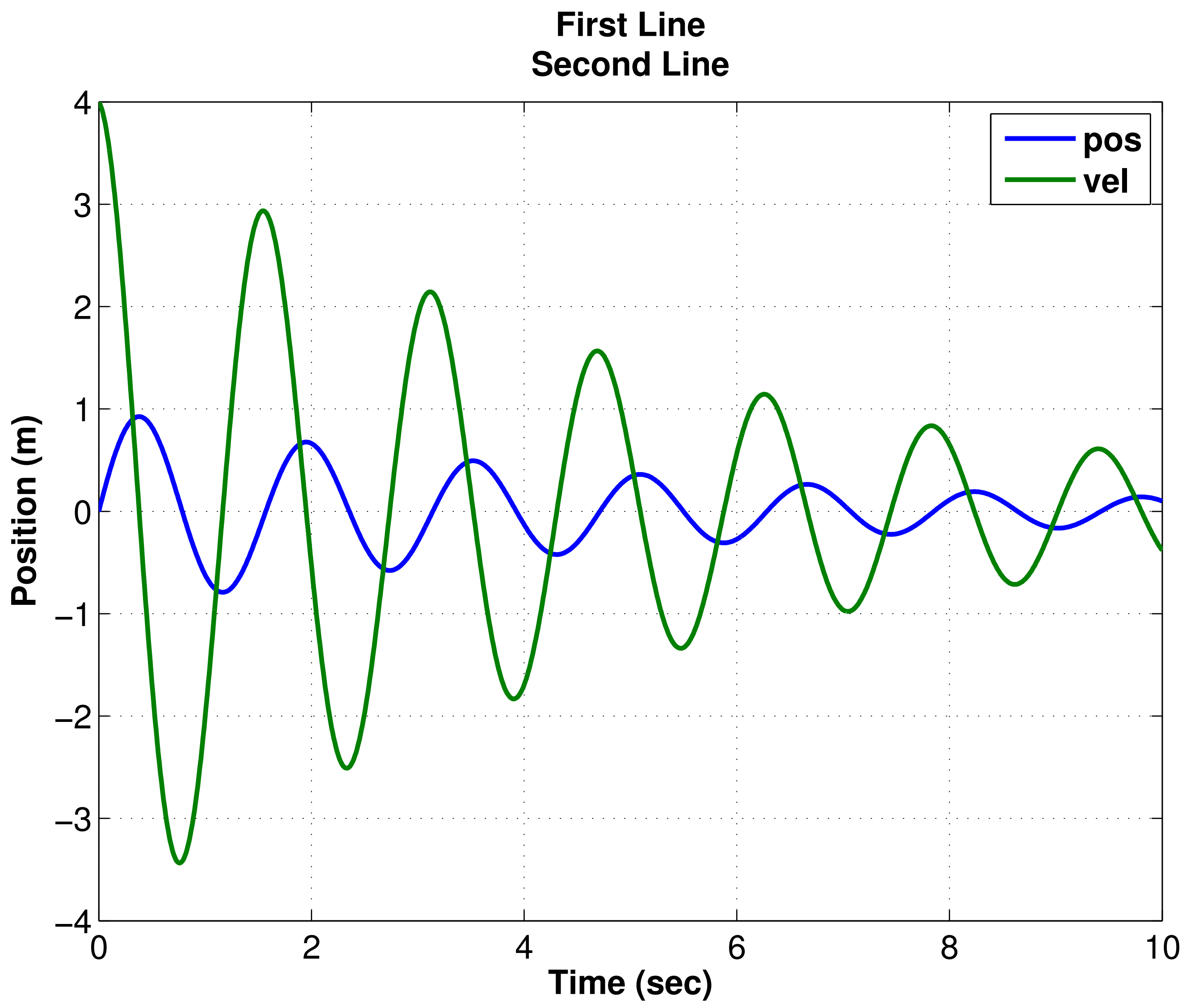
X Scale: Linear

☐ Reverse

Basic Example - Edit Plot

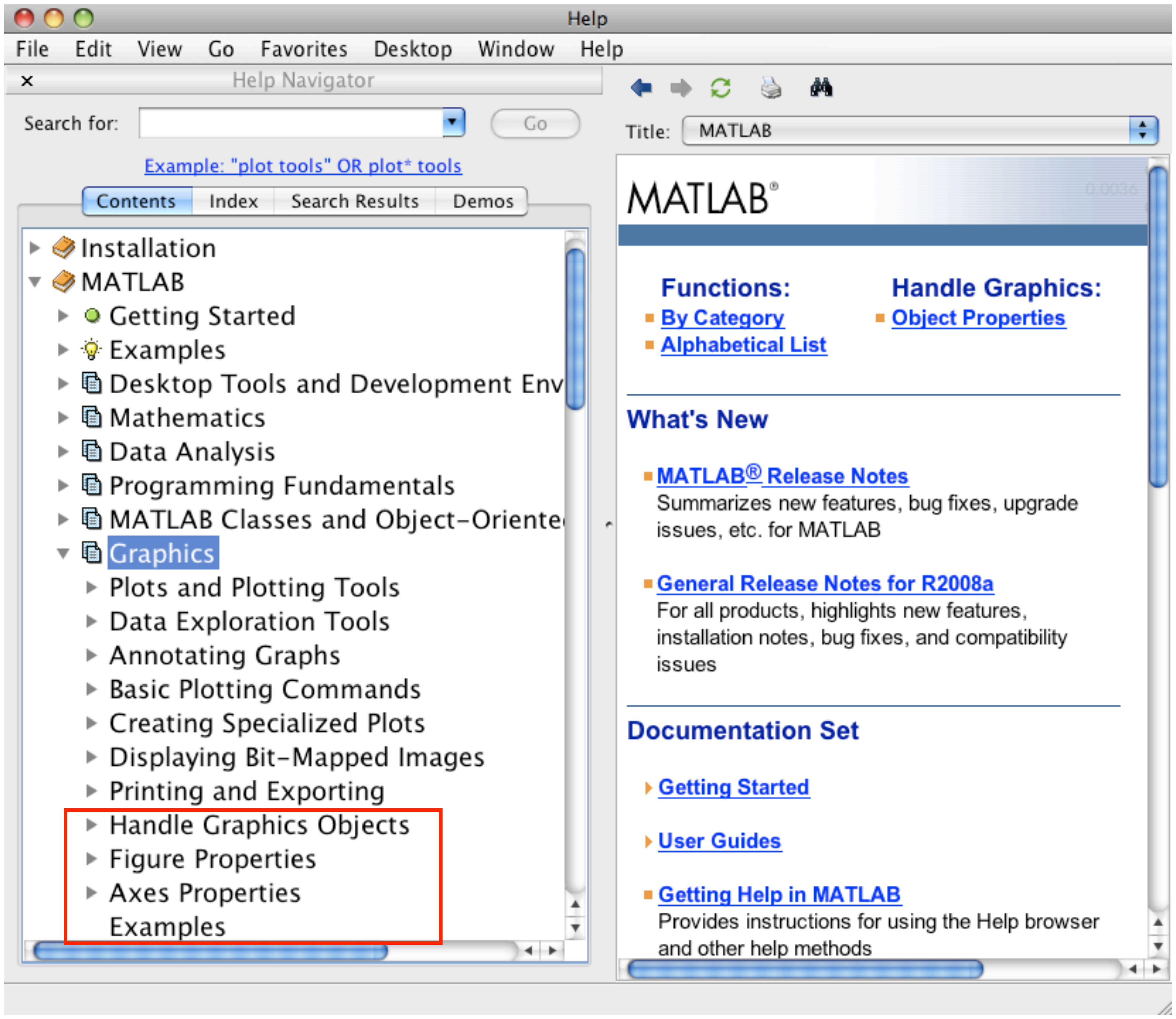
After much clicking and hunting, we have something that is more suitable for public consumption. Pointing and clicking is not a good option if you need to make many plots from lots of data. Automating the pointing / clicking actions is needed.





Handle Graphics

Here's how to get good reference material on handle graphics using "helpdesk."



Help
File Edit View Go Favorites Desktop Window Help
Help Navigator
Search for:
Go
Example: "plot tools" OR plot* tools
Contents Index Search Results Demos
Title: MATLAB

Installation
MATLAB
- Getting Started
- Examples
- Desktop Tools and Development Env
- Mathematics
- Data Analysis
- Programming Fundamentals
- MATLAB Classes and Object-Oriente
- Graphics
  - Plots and Plotting Tools
  - Data Exploration Tools
  - Annotating Graphs
  - Basic Plotting Commands
  - Creating Specialized Plots
  - Displaying Bit-Mapped Images
  - Printing and Exporting
  - Handle Graphics Objects
  - Figure Properties
  - Axes Properties
  - Examples

MATLAB®

Functions:
- By Category
- Alphabetical List

Handle Graphics:
- Object Properties

What's New

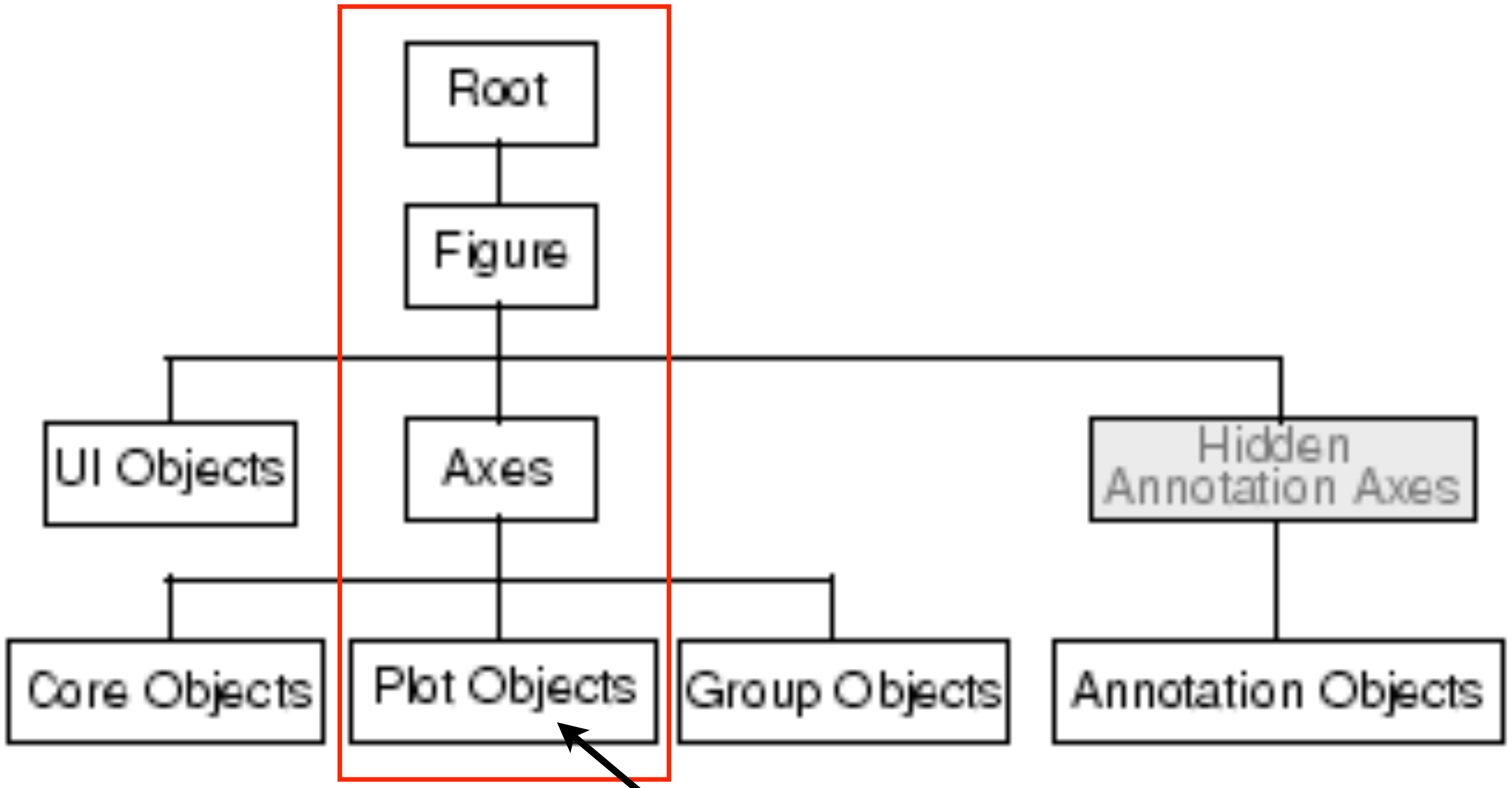
- MATLAB® Release Notes
  Summarizes new features, bug fixes, upgrade issues, etc. for MATLAB
- General Release Notes for R2008a
  For all products, highlights new features, installation notes, bug fixes, and compatibility issues

Documentation Set

- Getting Started
- User Guides
- Getting Help in MATLAB
  Provides instructions for using the Help browser and other help methods

Michigan Tech

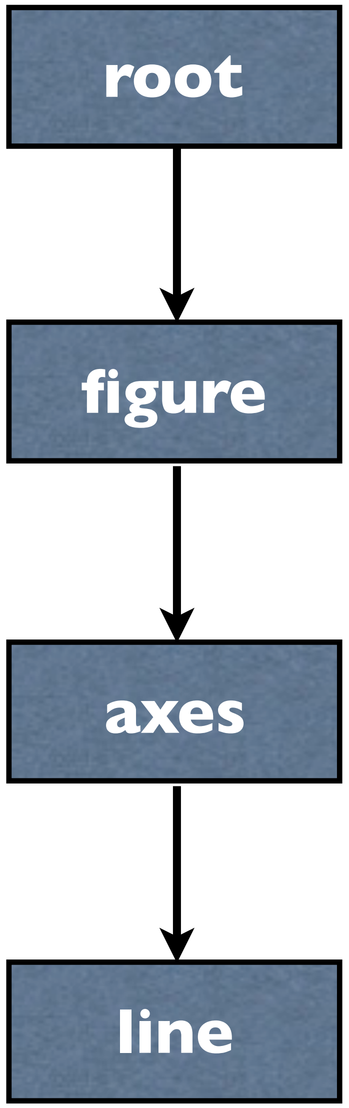
Handle Graphics - Hierarchy



Ex: **line, title**

Handle Graphics - Parents, Children, Set & Get

- The main components of a "plot" are **objects**, each with a **handle** number associated with it
- These components have a **linked-list** relationship.
- **Axes** are the child of a **figure**, but the parent of a **line**.
- Each object has a many, many attributes that can be explored with **get** and changed with **set**.
- Just about every aspect of an object can be modified through the available attributes. In short, if you want a certain look, you can likely achieve it.



Handle Graphics - Parents, Children, Set & Get

```
%% Plot2.m: Introduction to Handle Graphics
% This example uses the same data as plot1.m
% but starts the exploration and parents,
% children, set, and get.
clear all                          % clear workspace
close all                          % remove plots
t = 0:.01:10;                      % time vector
pos = sin(4*t).*exp(-.2*t);        % position
vel = 4*cos(4*t).*exp(-.2*t) +...
  -.2*sin(4*t).*exp(-.2*t);        % speed
h.fig = figure(1);                 % open a fig window
h.axs = axes;                      % put on some axes
h.lin(1) = line(t,pos,'Color','b');% plot pos data
h.lin(2) = line(t,vel,'Color','g');% plot vel data
grid;                              % put up a grid

h.axs                              % show the axes handle
get(h.lin(1),'Parent');            % show the handle of lin(1)'s parent

h.lin(1)                           % show the lin(1) handle
get(h.axs,'Children')              % show the handles of all axes' children
```

Michigan Tech

Handle Graphics - Parents, Children, Set & Get

```
%% PlotEx3.m: Introduction to Handle Graphics
% This example does the same thing as PlotEx2.m, but
% without all the low level commands such as
% axes and line.
clear all                          % clear workspace
close all                          % remove plots
t = 0:.01:10;                      % time vector
pos = sin(4*t).*exp(-.2*t);        % position
vel = 4*cos(4*t).*exp(-.2*t) +...
  -.2*sin(4*t).*exp(-.2*t);        % speed
h.fig = figure(1);                 % open a fig window
h.lin = plot(t,pos,t,vel);         % make the plot
grid;                              % put up a grid

h.axs = gca;                       % extract the axes handle
```

Explore object attirubtes -- try: `get(h.leg), get(h.fig), get (h.lin(1)), get(h.axs), set(h.lin,'LineWidth',3)`

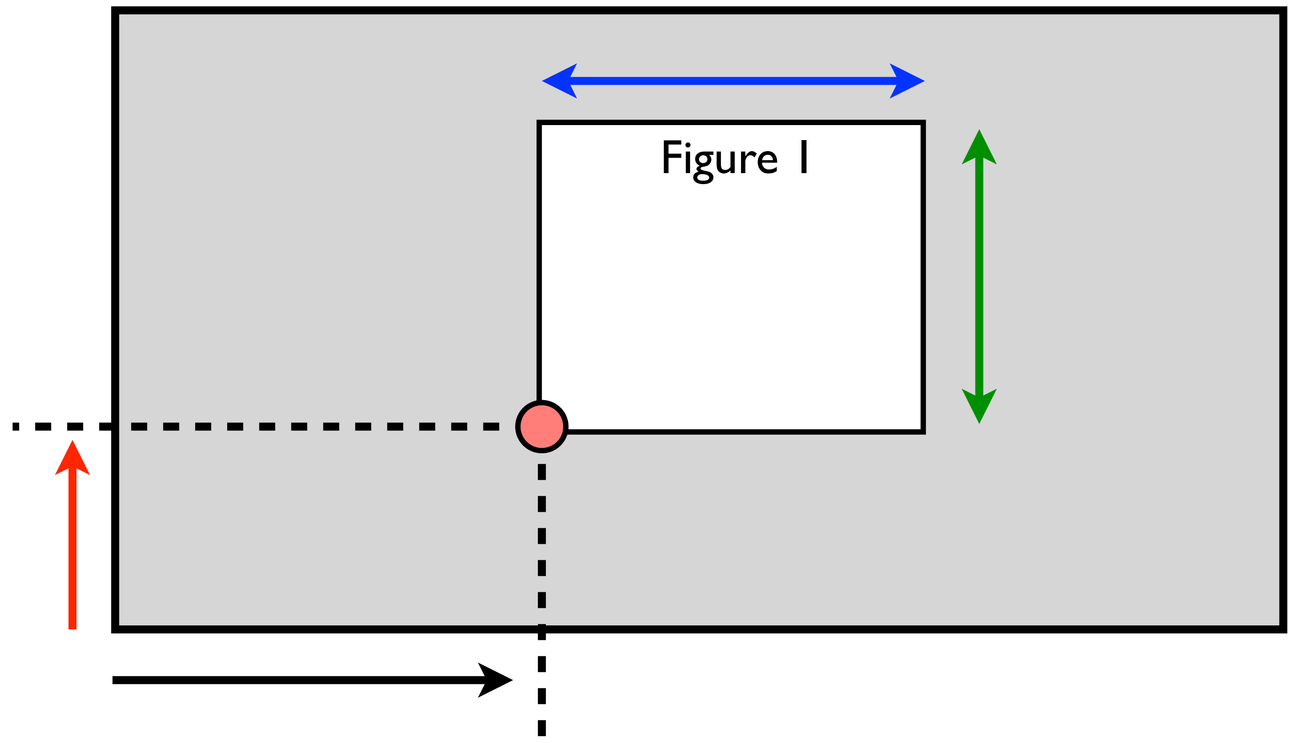
See attribute options using "set" : `set(h.leg)`

Handle Graphics - Figure Attributes

There are about 66 **figure** attributes. Here's my favorite 5.

```
MenuBar     - removes the ability to point/click edit
Color       - adjusts the color of the area around the plot
NumberTitle - removes the title designator at the top
Position    - specify location, width, & height
Visible     - makes the plot invisible or visible
```

Position = [**AA** **BB** **CC** **DD**]



Explore figure attributes with PlotEx4.m

Michigan Tech

Handle Graphics - Axes Attributes

There are about 103 **axes** attributes. Here's my favorite 12.

```
Color         - sets the color of the space behind the axes
FontSize      - sets legend and axis label font size
FontWeight    - sets legend and axis label font weight
GridLineStyle - sets grids to dots, dashes etc
XLim          - x-axis min and max vector
YLim          - y axis min and max vector
XTick         - specify where to put x axis tick marks
YTick         - specify where to put y axis tick marks
XTickLabel    - custom x axis tick labels
YTickLabel    - custom y axis tick labels
XAxisLocation - put x axis labels on top or bottom
YAxisLocation - put y axis labels at right or left
```

Explore axes attributes with PlotEx5.m and PlotEx6.m

Handle Graphics - Line Attributes

There are about 35 **line** attributes. Here's my favorite 11.

```
Color           - color of the line
LineStyle       - line type (dashed, dotted, etc.)
LineWidth       - line width
Marker          - marker (circle, triangle, etc.)
MarkerSize      - marker size
MarkerEdgeColor - marker edge color
MarkerFaceColor - marker fill color
XData           - x axis data values
YData           - y axis data values
Zdata           - z axis data values
Visible         - make the line disappear or reappear
```

Explore axes attributes with PlotEx7.m

Handle Graphics - Text Attributes & Annotation

- There are several different objects that have text associated with them (text, xlabel, ylabel, title, etc.)
- You can imagine the attributes
- My favorite is the LaTeX **interpreter** option
- There's also a host of **annotation** objects. I'm not a big fan of these, as I typically annotate plots based on how their used (presentation, report, etc.). Also, they don't scale naturally, so you have to fiddle with them.

Explore text and annotation attributes with PlotEx8.m

Handle Graphics - Saving Figures & Record Keeping

- use **saveas** to save your figure to a .fig format
- use **load** to bring the figure back
- use **get(h , XData)** etc. to extract the raw data associated with a **line** - this is huge in terms of keeping track of analysis results
- Use **setappdata** to make notes about your figure (or any object). Again, huge for tagging figures, e.g. indicate the "where" and "when" for the data, etc.

Explore text and annotation attributes with PlotEx9.m

Handle Graphics - Special 2-D Plots

- use **subplot** to easily create multiple axes in a figure
- use **semilogx, semilogy, and loglog** to make nice log plots

Explore some special plots with PlotEx10.m

Handle Graphics - User Submitted Functions Sampler

ploty4	- like **plotyy**, but with 4 y-axes
dragplot	- combine lines from different plots
spring	- draw fun springs, nice for 2D animation
fftf	- time-2-freq, filter, then freq-2-time
bubbleplot3	- challenge the COE with a 3D bubbleplot
cascade	- cascade all the open figure windows
freezecolors	- multiple **colormaps** in a figure
quiverS	- quiver plot with an arrow magnitude scale
gridcolor	- manipulate grid line colors
tan_plane	- nice 3D transparent solids
nicebars	- translucent error bars around a line