



WEEK #5

HANDLE GRAPHICS

Customizing MATLAB Graphics

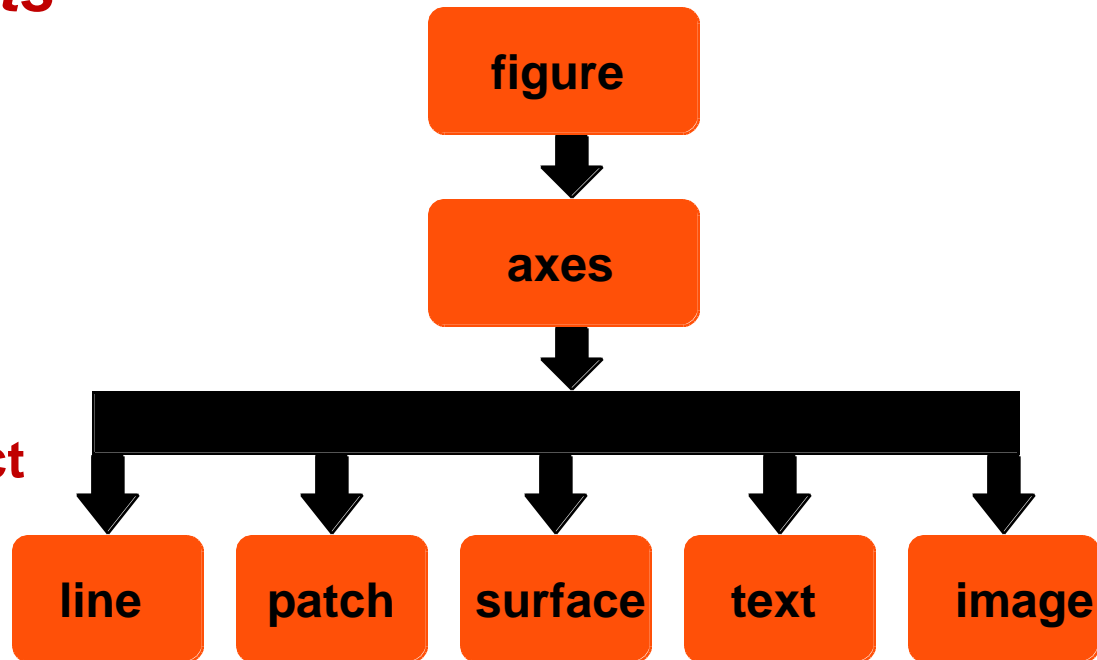
ADVANCED GRAPH CONTROL

- There numerous ways that the appearance of a graph can be controlled
- They include
 - Axis line,color,etc
 - Axis tick mark placement
 - Plot and axis line type and width
 - Plot box control
 - etc

MATLAB GRAPHICS HIERARCHY

- A typical MATLAB graphics is a hierarchy of *graphic objects*

- root
- figure object
- axes object
- line object
- patch object
- surface object
- image object
- text object



ROOT

- The highest level of hierarchy is the root. It refers to the physical computer screen.

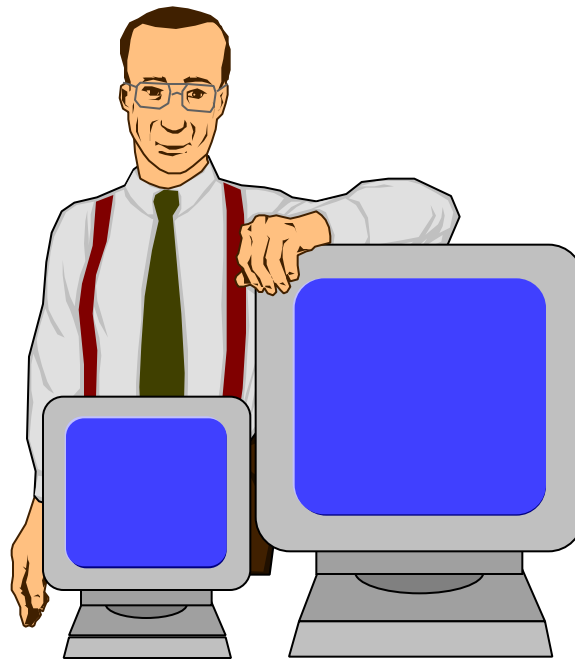
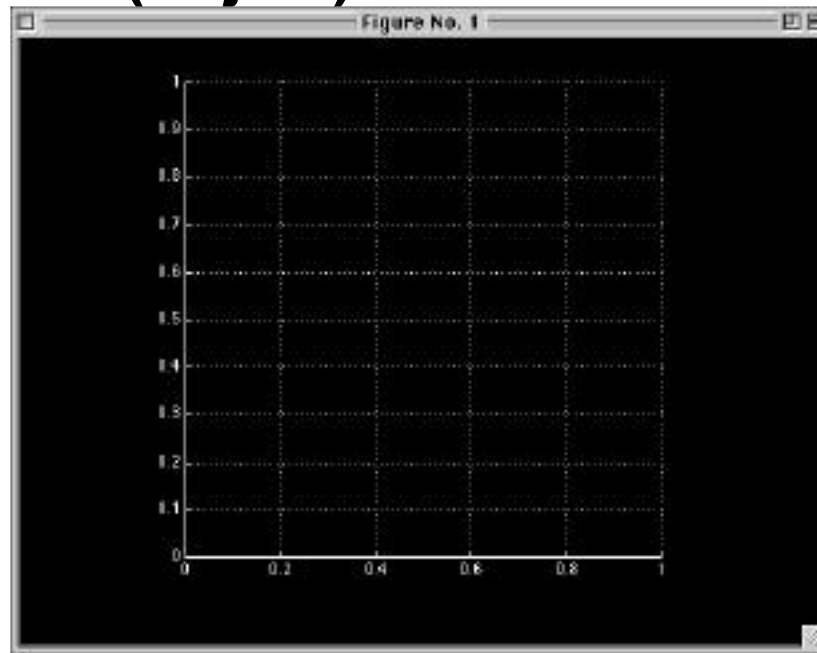


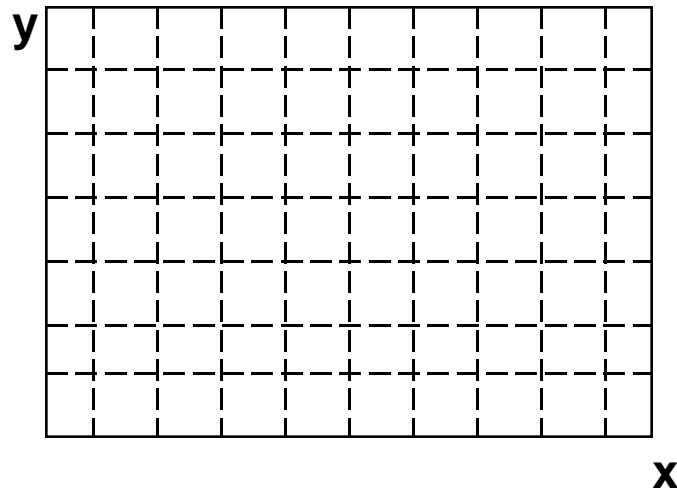
Figure Object

- The next level below the root is the figure object. Any graphics command creates a figure window(object).



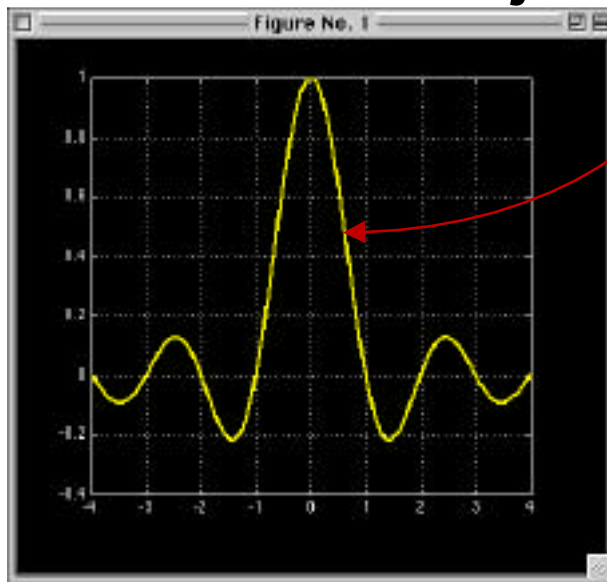
Axes Object

- One level below figure object is called axes object . All plots will appear inside the axes framework. Axes has a broader definition than axis



Line Object

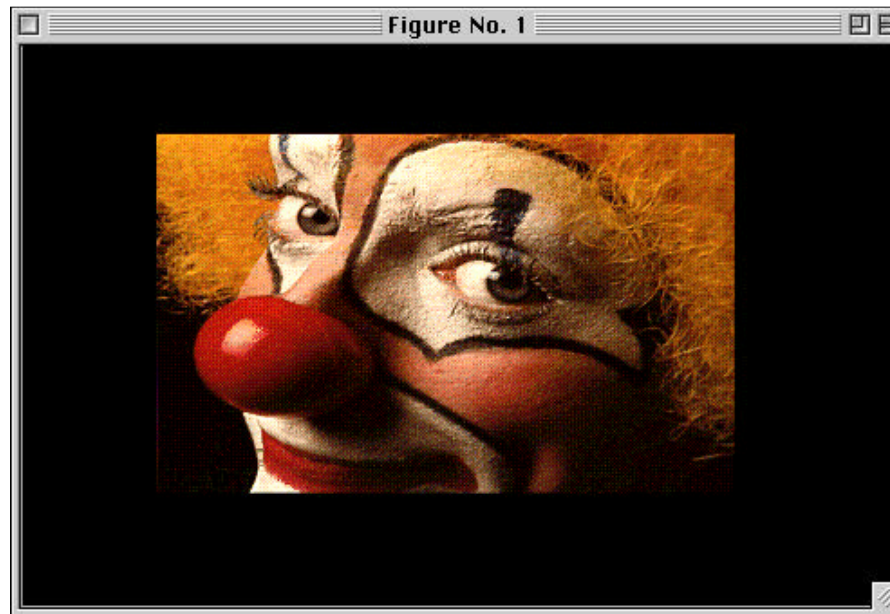
- Once an axes is established, plotting can be done. If you are plotting elements consists of lines (such as those generated by *plot* or *plot3*) then we have a line object



line object

patch, surface, image and text objects

- similar to line objects, graphics that are generated by *patch*, *surf*, *image* or *text* commands define unique graphic objects



Customizing Graphics

- **We can customize, to a very fine degree of detail, almost any aspect of a graphic.**
- **The key to success is to understand which level of hierarchy does each part of the graph belong?**
- **Enter “Handle Graphics”**

Graphics pointers

- Consider a typical graphic session. It consists of one or several figure windows. Inside each figure, there might be one or more graphics. Inside each graph there are lines, surfaces, text etc..
- The first task is the ability to “point” to them
- In MATLAB, each graphics element is identified by a unique pointer called a *handle*

SET AND GET

- In order to retrieve the properties of a plot, we use the combination of *set* and *get*.
- *get* (*h*) retrieves all the existing parameters of graphic objects with handle *h*.
- *set* (*h*) allows us to see the current settings and change them.

`set(fig_handle,'PropertyName','PropertyValue')`



ROOT

- The root is permanently assigned handle 0.
- To find out what the settable properties are, use
 - `get(0)`

Figure Object Handle (gcf)

- Now, let's create a figure object by simply typing *figure*.
- To find the handle to this figure, use *get current figure*, abbreviated by *gcf*
- It turns out that the handle to the current figure, is the figure number itself.
- To find the settable parameters, type *get(gcf)*

Axes Object Handle (gca)

- Now plot something in the figure window. This will create your first axes object inside the figure object.
- To get the handle to the axes just created, use ***h=gca***
- Then use ***get(h)*** or ***set(h)*** to change the settings

Line Object Handle

- In order to change the appearance of a plot, you must record its handle before plotting it. Then use either `get(h)` or `set(h)`

- For example

```
t=-4:0.01:4
```

```
p=sinc(t);
```

```
h1=plot(t,p)
```



Pointing to more than one plot

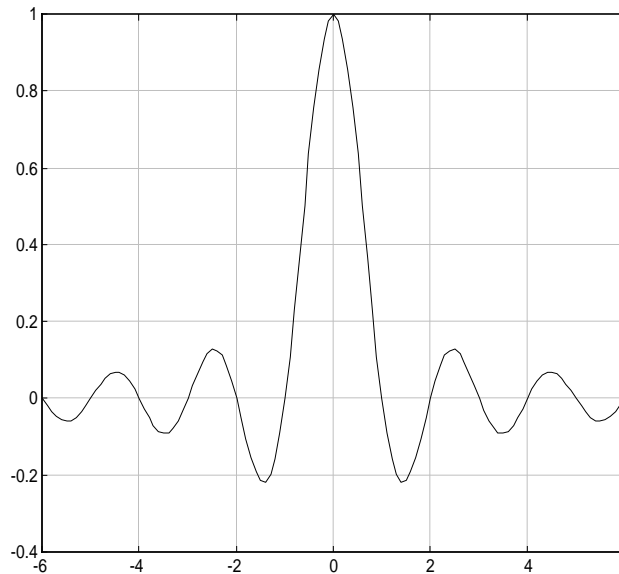
- Let's say you have two or more plots on an axes and wish to address them individually
- Use separate handles for each
 - `h1=plot(t,x)`
 - `h2=plot(t,y)`

Surface Objects

- Handles to surface objects created by various forms of surf command, are similarly created or obtained.
 - $h1 = \text{surf}(x,y,z)$
- The only difference is that set(h) and get(h) will return parameters that are different from line objects.

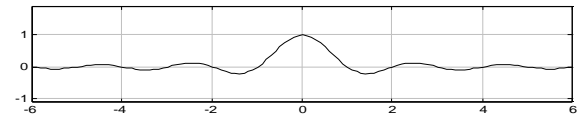
EXAMPLE PLOT

- Throughout this session we will be using the plot of a sinc function.

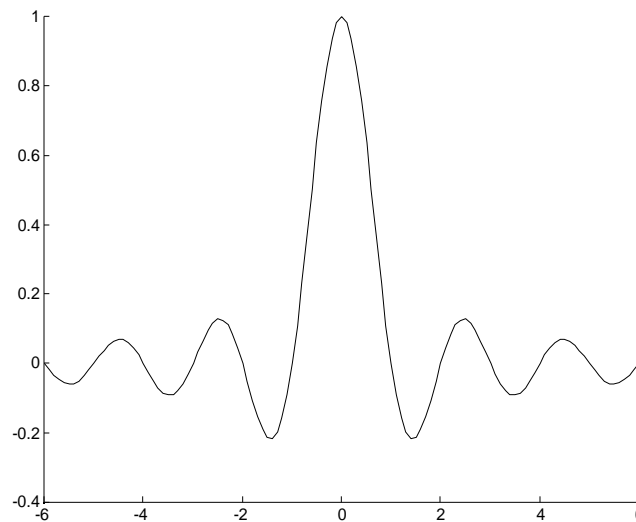


AspectRatio

- Aspect ratio is the height width to width ratio
- Synopsis:
`set(fig_handle,'PropertyName','Property Value')`
- `set(gca, 'aspectratio', [4,1])`

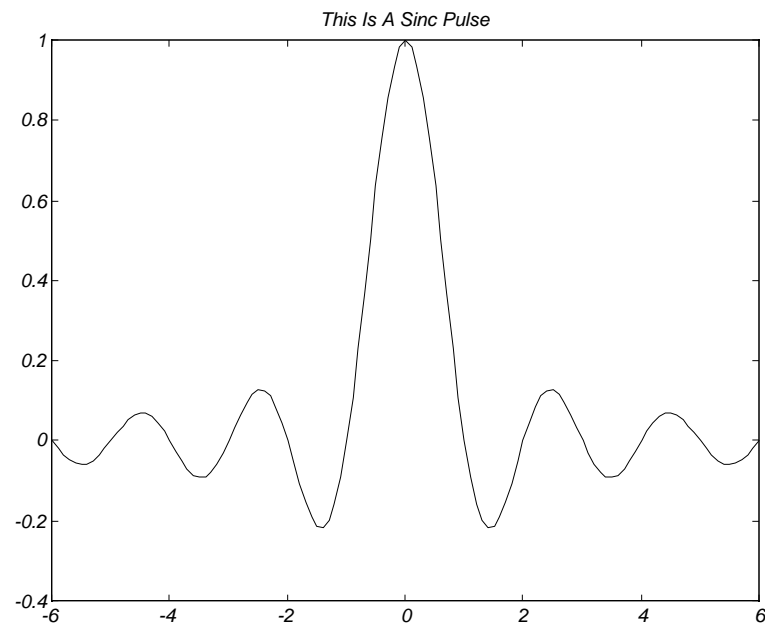


- `set(gca,'box','off')`
- The previous figures had `box=on`.



FontAngle

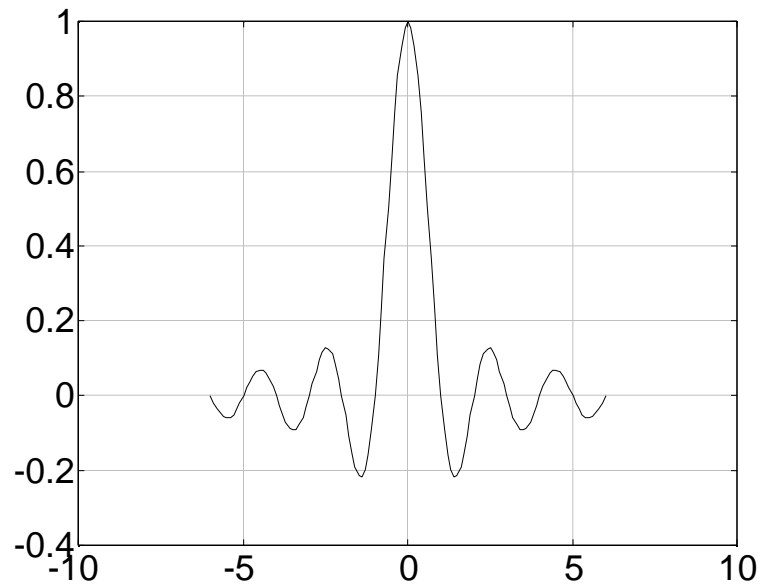
■ `set(gca,'fontangle','italic')`



FontSize

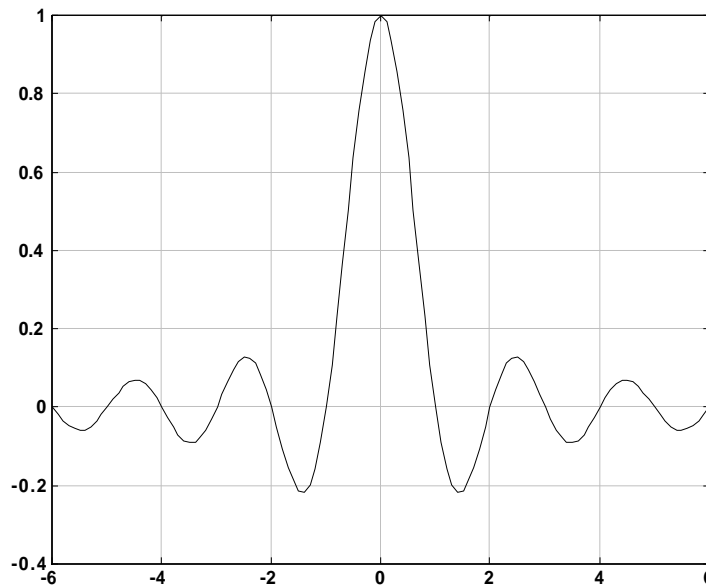
■ `set(gca,'fontsize',[24])`

Default Value: 12



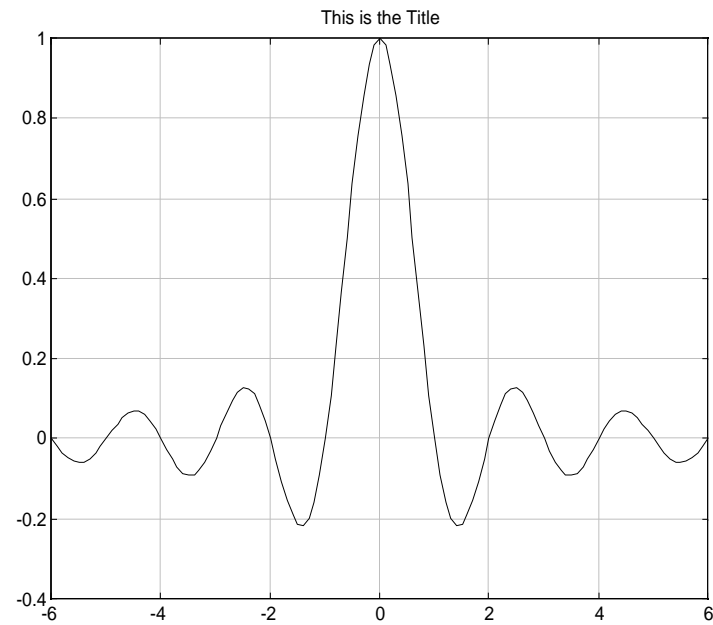
FontWeight

■ `set(gca,'fontweight','bold')` Default=normal



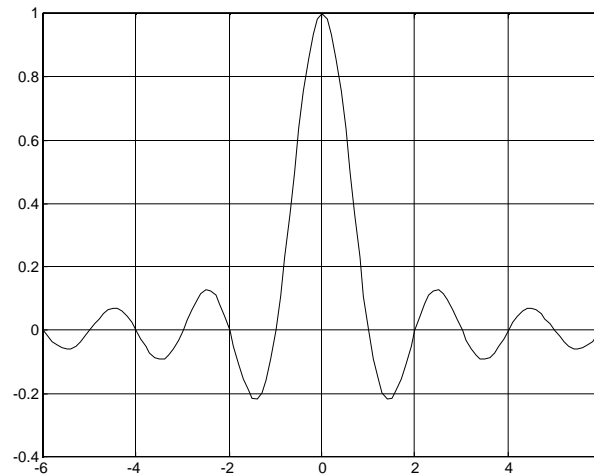
Title

■ `set(gca,'title',text(0,0,'This is the Title'))`



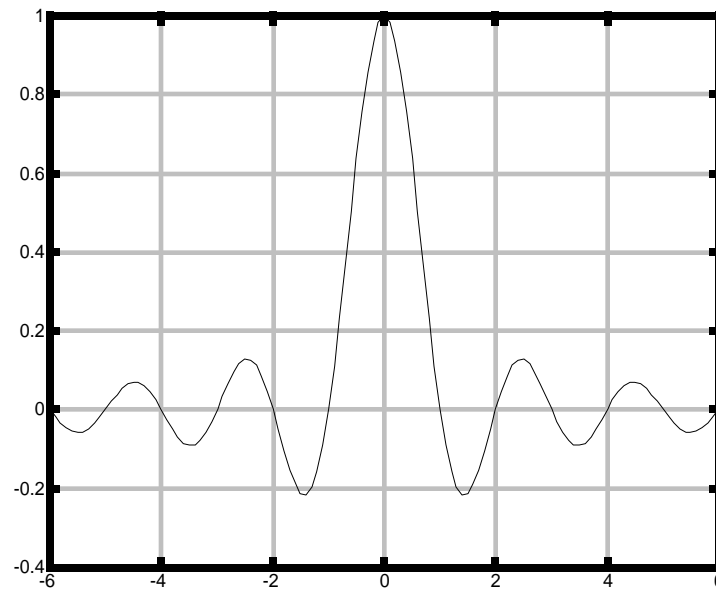
GridLineStyle

- `set(gca,'gridlinestyle','-')` **Solid Lines**
- Dashed, dotted, and dash-dotted lines are also available.



LineWidth

■ `set(gca,'linewidth',[5])`

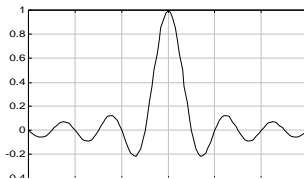


Units

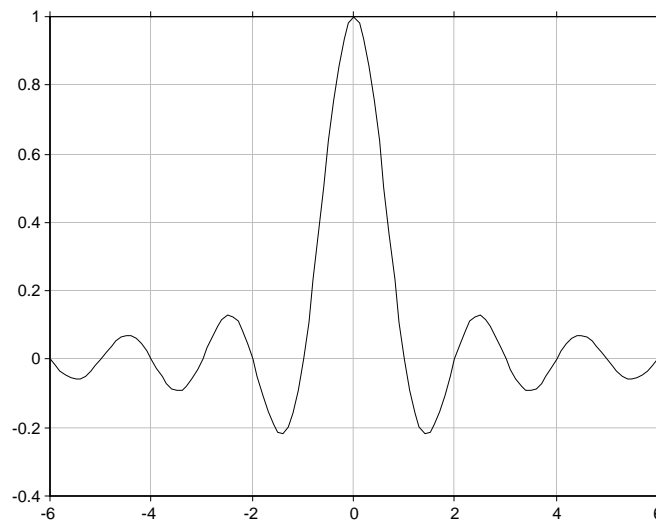
- **set(gca,'units','pixels')** will change the units of measurement to pixel spacings
- **Other available units are:**
 - **inches - inches**
 - **cent - centimeters**
 - **points - each point is equivalent to 1/72 of an inch**
- **normalized (default) - the lower left corner maps to (0,0) and the upper right corner to (1,1)**

Position

- `set(gca,'position',[.1 0 .5 .5])`
- The 'PropertyValue' corresponds to :
[left, bottom, width, height]

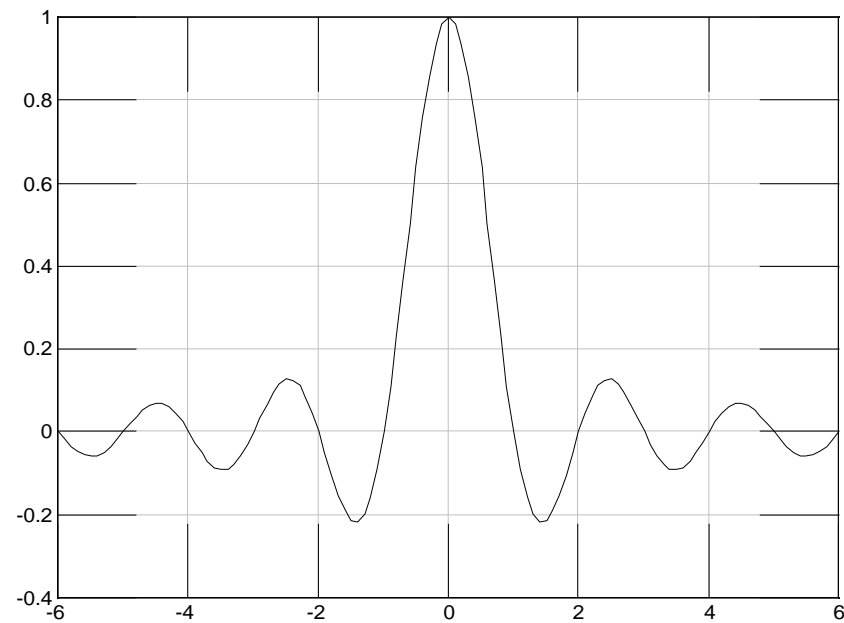


- `set(gca,'tickdir','out')`
- The default for 2-D plots is 'in'.



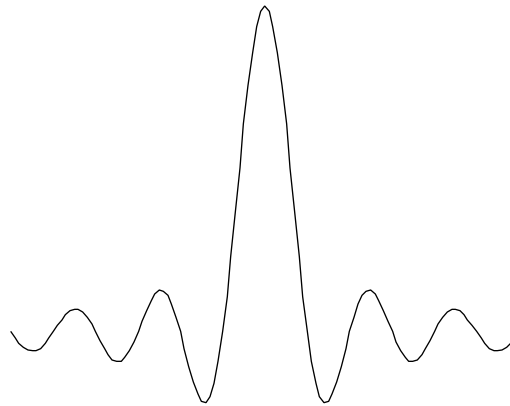
TickLength

■ `set(gca,'ticklength',[.1 .1])`

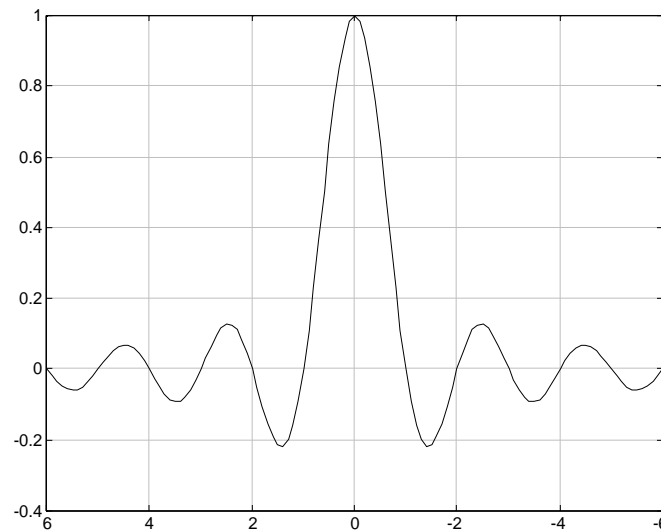


Visible

- `set(gca,'visible','off')`
- The default is 'on'

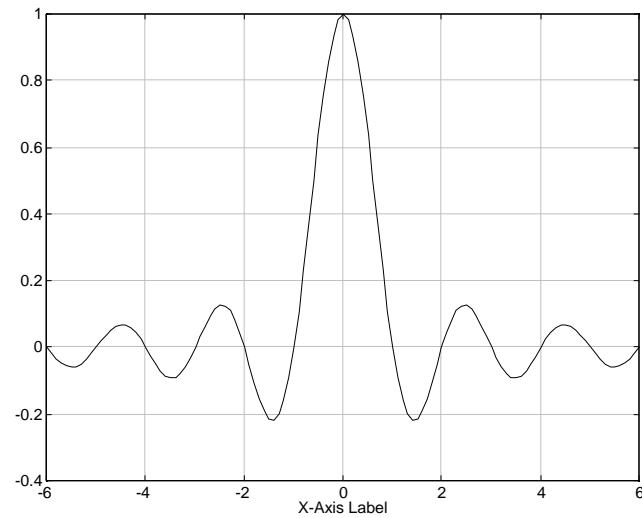


- `set(gca,'xdir','reverse')`
- Default is 'normal'



XLabel

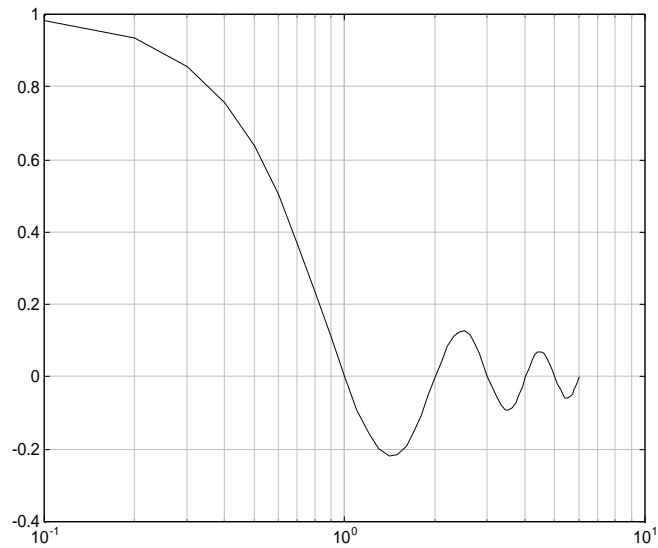
- `set(gca,'xlabel',text(0,0,'X-Axis Label'))`



Xscale

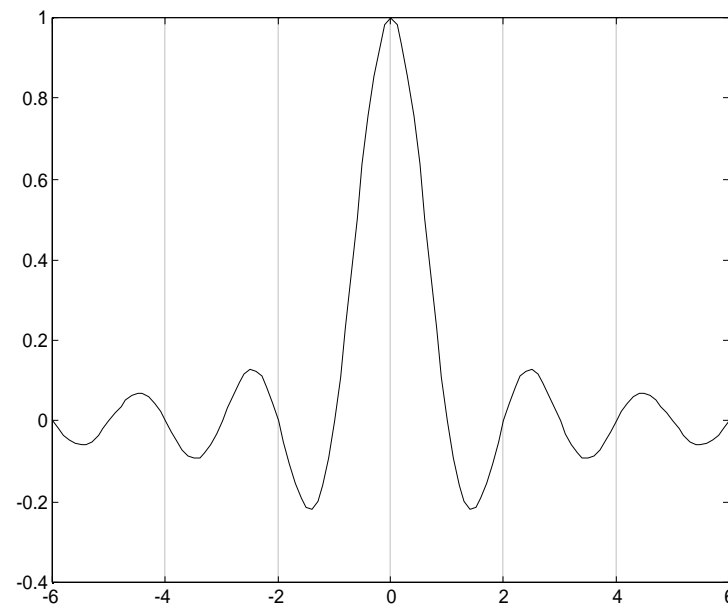
■ `set(gca,'xscale','log')`

Default = linear



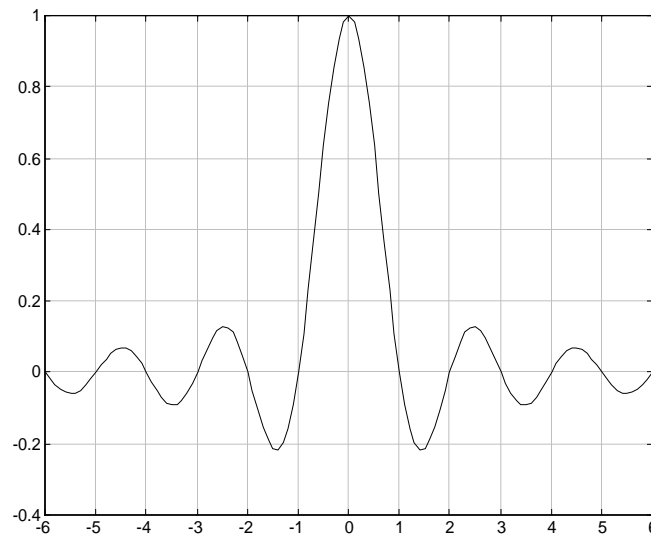
■ `set(gca,'xgrid','on')`

Default = off



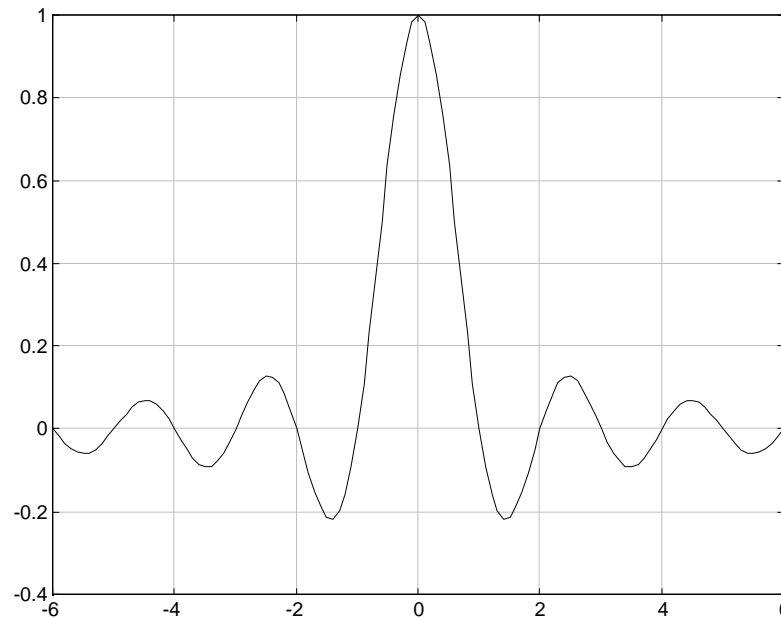
XTick

- `set(gca,'xtick',[-6:1:6])`
- Use `set(gca,'xtick',[])` to turn tick marks off.



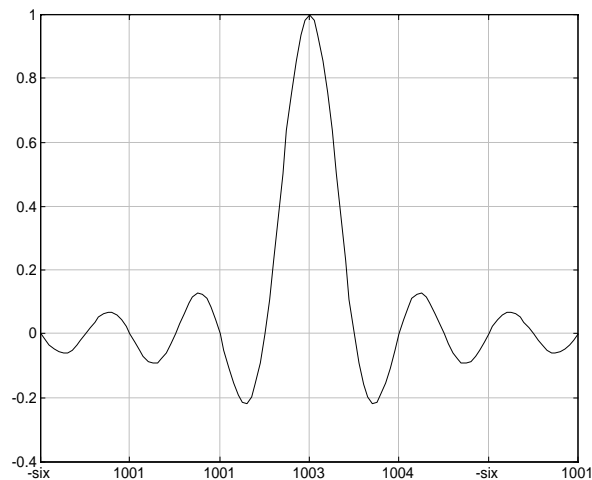
XTickMode

■ `set(gca,'xtickmode','auto')`



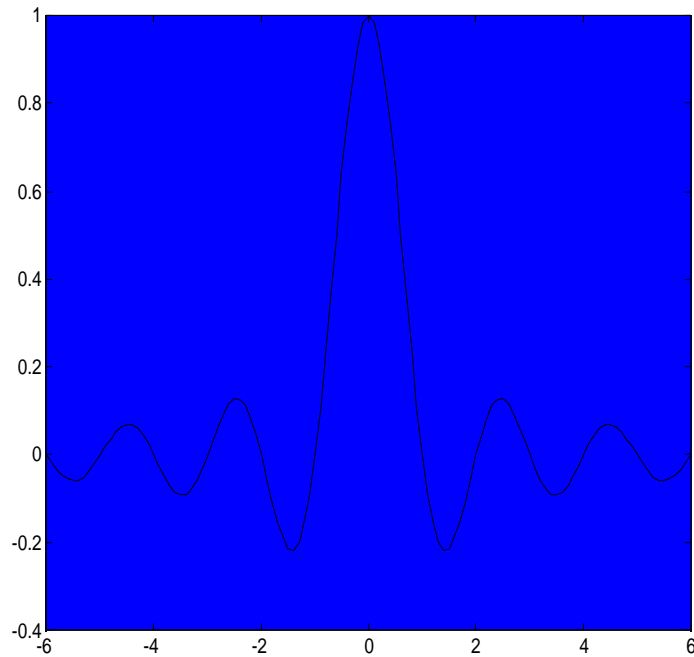
XTickLabels

■ `set(gca,'xticklabels',['-six' ;'1001'
;'1001';'1003';'1004'])`



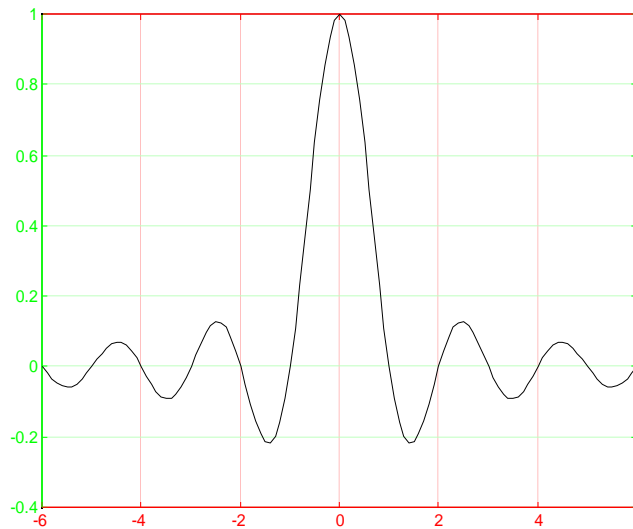
Color

■ `set(gca,'color',[0 0 1])`



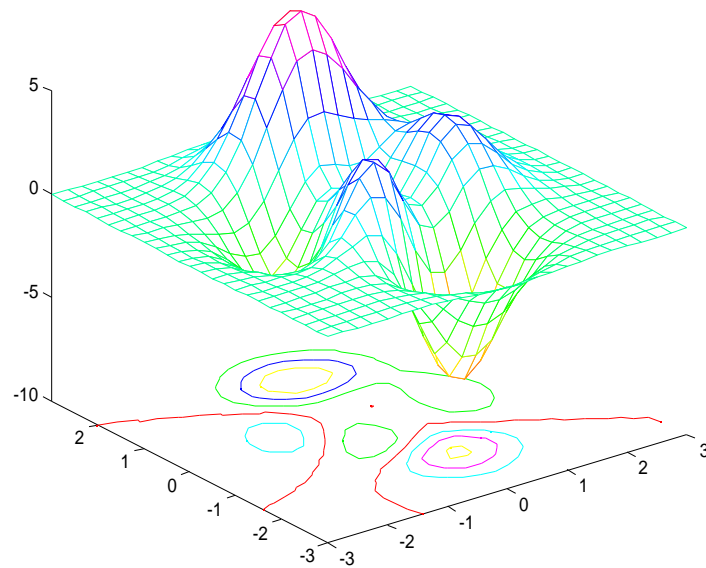
RGB	COLOR
[1 1 0]	yellow
[1 0 1]	magenta
[0 1 1]	cyan
[1 0 0]	red
[0 1 0]	green
[0 0 1]	blue
[1 1 1]	white
[0 0 0]	black

- `set(gca,'xcolor',[1 0 0])`
- `set(gca,'ycolor',[0 1 0])`



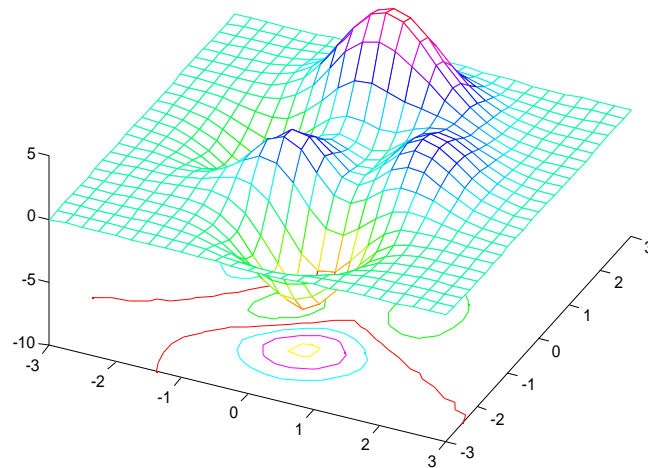
3D Plot

- The **AXES** command can be used to manipulate viewing angles of 3D plots.



View

- `set(gca,'view',[25 50])`
- 25 is the azimuth angle in degrees, 50 is the elevation.



AXIS Commands

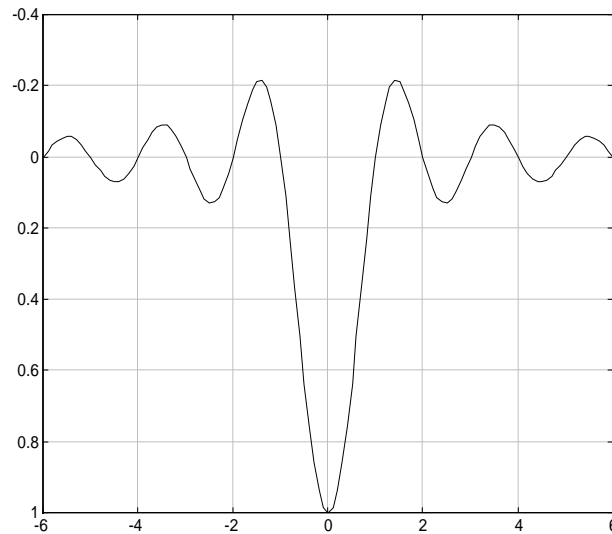
- **AXIS('auto')** returns the axis scaling to its default, automatic mode where, for each plot, $x_{min}=\min(x)$, $x_{max}=\max(x)$, etc.
- **V=AXIS** returns a row vector containing the scaling for the current plot whose contents would be x_{min} , x_{max} , y_{min} , y_{max} , etc.
- **AXIS(AXIS)** freezes the scaling at the current limits, so that if **HOLD** is turned on, subsequent plots will use the same limits.

AXIS - General

- **AXIS('off')** turns off all axis labeling and tick marks.
- **AXIS('on')** turns axis labeling and tick marks back on.
- **[S1,S2,S3]=AXIS('state')** returns strings indicating the current setting of three axis properties:
 - S1 = 'auto' or 'manual'**
 - S2 = 'on' or 'off'**
 - S3 = 'xy' or 'ij'**

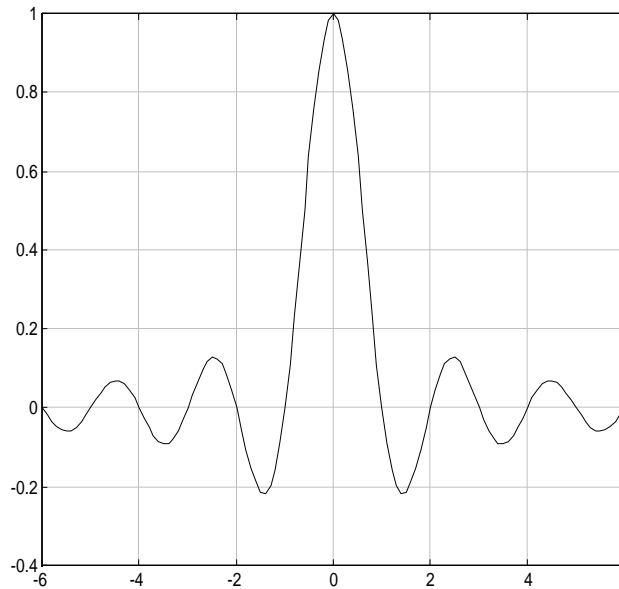
AXIS - Matrix Mode

- **AXIS('ij')** puts MATLAB into its “matrix” axes mode. The coordinate system origin is at the upper left corner.



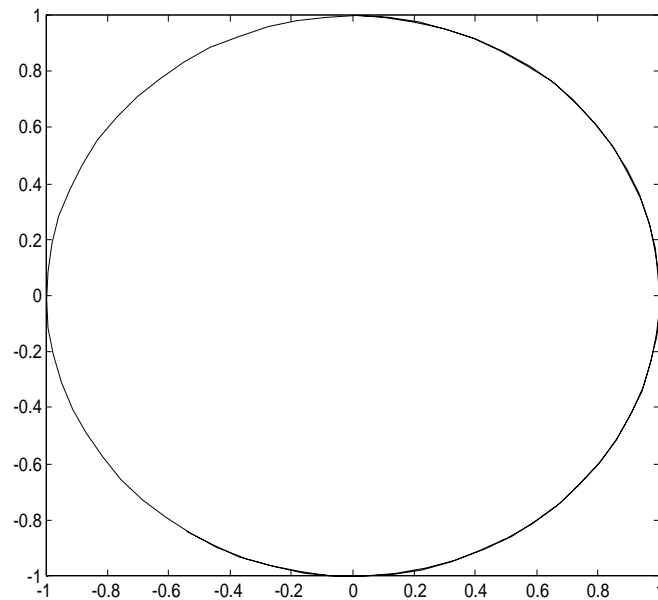
AXIS - Cartesian Mode

- **AXIS('xy')** puts MATLAB into its default “Cartesian” axes mode.



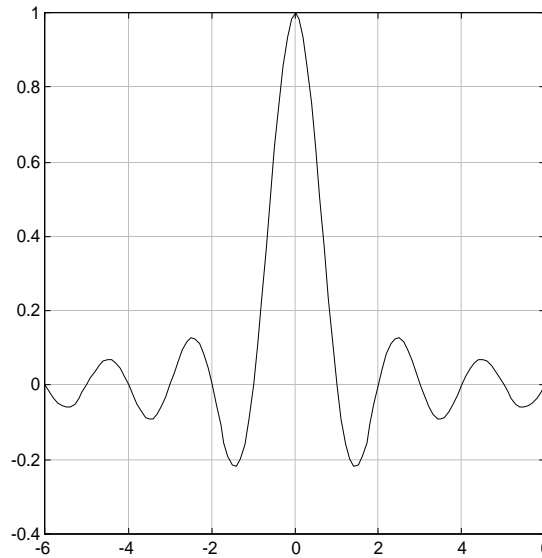
AXIS - Box Size

- `plot(sin(x),cos(x))`
- `axis('equal')`



AXIS - Box Size (con't)

- **AXIS('square')** makes the current axis box square in size.

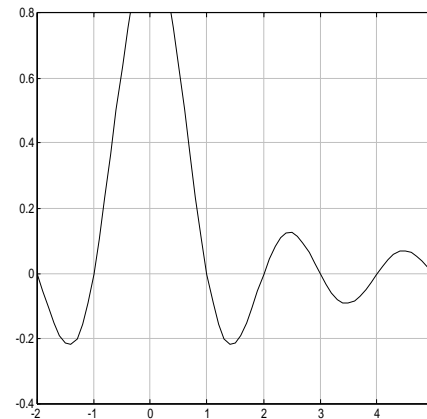
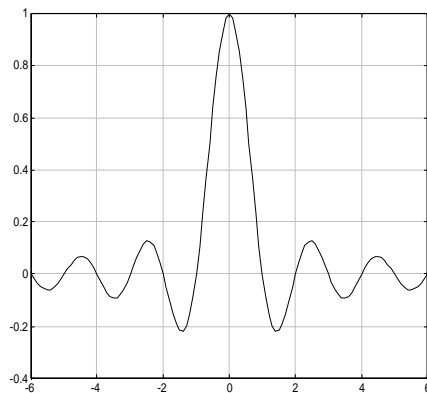


AXIS - Box Size (con't)

- **AXIS('normal')** restores the current axis box to full size and removes any restrictions on the scaling of the units. This undoes the effects of **AXIS('square')** and **AXIS('equal')**.
- **AXIS('image')** sets the aspect ratio and the axis limits so the image in the current axes has square pixels.

AXIS - Scaling

- **AXIS([XMIN XMAX YMIN YMAX])** sets scaling for the x- and y- axes on the current plot.
- **AXIS([-2 5 0 0.8])** will augment the plot as shown on the right. The original is on the left.



IN-CLASS EXERCISE

- **Plot $x=\cos(t)$ and $y=\exp(-t)\cos(10*t)$ on the same axes then do the following customization**
 - label your figure window by your name
 - set axis thickness to 3 points
 - set axis color to blue
 - set gridline style to dots
 - set axis labels to 14 point Helvetica
 - set line style for x to solid and y to dash
 - set line thickness for x and y to 3 points, one in red and one in white

HOMEWORK

- **Using subplot, do the in-class exercise on page 51 by placing the two plots on two different axes. Then**
 - label your figure window by your name
 - set axis color to blue for the top and red for the bottom
 - set background figure to red(top) and blue(bottom)
 - set gridline styles to dots for both
 - set axis labels to 14 point Helvetica for the top and 14pt times for the bottom
 - set line style for x to solid and y to dash
 - set line thickness for x and y to 3 points, one in red and one in white