6.1 Introduction
6.2 Anatomy of 2-D (x–y) Plots in MATLAB
6.3 Line Specifications
6.4 Accessing Plot Properties via the `get` and `set` Functions
6.5 Text Strings
6.6 Overlays, Legends, Subplots, and Multiple Figures
6.7 Special Types of 2-D Plots
6.8 3-D Plots
6.9 Changing Color Palette
6.10 Easy-to-Use Function Plots
6.11 Plot Editing
6.12 Interactive Plotting and Animation
6.13 Requirements to Engineering Plots
Plotting Basics

x=0:pi/20:pi;
y=sin(x);
plot(x,y)

x=linspace(0,pi,20);
y=sin(x);
plot(x,y)

plot(0:pi/20:pi,sin(0:pi/20:pi))

fplot('sin(x)',[0,pi])

plot(sin(0:pi/20:pi))
Plot’s Anatomy

```
figure
axis
line
xlabel
title
```

```
ezplot('sin(x)')
```
Enhancing Plot Readability

title('text') – places text in a title at the top of a plot
axis([xmin xmax ymin ymax]) – sets the minimum and maximum limits of the x- and y-axes. (If you want to specify just one limit but want MATLAB to autoscale the other, use Inf or -Inf for autoscaled limits, or use xlim and ylim functions.) You may also use one of the following:
  axis square – selects the axes’ limits so that the plot will be square
  axis equal – assures that x-and y-axes have same tick mark spacing
  axis tight – sets the limits to the range of the data
  axis manual – freezes the scaling at the current limits
  axis auto – returns axis scaling to its default autoscaling mode
  axis vis3d – freezes aspect ratio properties to enable rotation of 3-D objects and overrides stretch-to-fill

v=axis – returns the current axis scaling in the vector v (so that you might use the same scaling for another plot by calling axis(v))

grid – displays gridlines at the tick marks corresponding to the tick labels. Use grid on to turn on gridlines and grid off to turn off grid lines. When used by itself, grid toggles the grid switch on or off

xlabel('text') – places a text label on the x-axis (abscissa)
ylabel('text') – places a text label on the y-axis (ordinate)
text(x,y,'text') – adds the text in the quotes to location (x,y) on the current axes
gtext('text') – places the text in Figure window at a point specified by the click of the mouse
annotation(annotation_type) – creates the specified annotation type, such as line, arrow, double arrow, text arrow, textbox, ellipse, and rectangle
>> t=linspace(0,2*pi);
>> line(cos(t),sin(t)), grid on

>> title('My Plot')
>> xlabel('x-axis'), ylabel('y-axis')
>> text(0.5,0.5,'Text')

>> annotation('arrow',[0.5 0.7],[0.7 0.8])
>> gtext('Chosen Location')
Stream modifiers:

- \textbf{\texttt{\textbackslash bf}} - bold font
- \textit{\texttt{\textbackslash it}} - italics font
- \textit{\texttt{\textbackslash sl}} - oblique font (rarely available)
- \textit{\texttt{\textbackslash rm}} - normal font
- \textit{\texttt{\textbackslash fontname\{fontname\}}} - specify the name of the font family to use
- \textit{\texttt{\textbackslash fontsize\{fontsize\}}} - specify the font size (in font units)

(The first four modifiers are mutually exclusive. Stream modifiers remain in effect until the end of the string or only within the context defined by braces \{\}.)

The subscript character “\_” and the superscript character “\^” modify the character or substring defined in braces immediately following.

To print the special characters used to define the TeX strings when Interpreter is TeX, prefix them with the backslash “\” character, for instance: \textbackslash, \textbackslash{,} \textbackslash}, \textbackslash\_, \textbackslash^.
<table>
<thead>
<tr>
<th>Character Sequence</th>
<th>Symbol</th>
<th>Character Sequence</th>
<th>Symbol</th>
<th>Character Sequence</th>
<th>Symbol</th>
<th>Character Sequence</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>\alpha</td>
<td>α</td>
<td>\nu</td>
<td>ν</td>
<td>\Gamma</td>
<td>Γ</td>
<td>\equiv</td>
<td>≡</td>
</tr>
<tr>
<td>\beta</td>
<td>β</td>
<td>\xi</td>
<td>ξ</td>
<td>\Delta</td>
<td>∆</td>
<td>\leq</td>
<td>≤</td>
</tr>
<tr>
<td>\gamma</td>
<td>γ</td>
<td>\pi</td>
<td>π</td>
<td>\Theta</td>
<td>Θ</td>
<td>\geq</td>
<td>≥</td>
</tr>
<tr>
<td>\delta</td>
<td>δ</td>
<td>\varpi</td>
<td>σ</td>
<td>\Lambda</td>
<td>Λ</td>
<td>\neq</td>
<td>≠</td>
</tr>
<tr>
<td>\epsilon</td>
<td>ε</td>
<td>\rho</td>
<td>ρ</td>
<td>\Xi</td>
<td>Ξ</td>
<td>\clubsuit</td>
<td>♠</td>
</tr>
<tr>
<td>\zeta</td>
<td>ζ</td>
<td>\sigma</td>
<td>σ</td>
<td>\Pi</td>
<td>Π</td>
<td>\diamondsuit</td>
<td>♦</td>
</tr>
<tr>
<td>\eta</td>
<td>η</td>
<td>\varsigma</td>
<td>ξ</td>
<td>\Sigma</td>
<td>Σ</td>
<td>\heartsuit</td>
<td>♥</td>
</tr>
<tr>
<td>\theta</td>
<td>θ</td>
<td>\tau</td>
<td>τ</td>
<td>\Upsilon</td>
<td>Υ</td>
<td>\spadesuit</td>
<td>♠</td>
</tr>
<tr>
<td>\vartheta</td>
<td>θ</td>
<td>\upsilon</td>
<td>υ</td>
<td>\Phi</td>
<td>Φ</td>
<td>\leftrightarrow</td>
<td>↔</td>
</tr>
<tr>
<td>\iota</td>
<td>ι</td>
<td>\phi</td>
<td>φ</td>
<td>\Psi</td>
<td>ψ</td>
<td>\leftarrow</td>
<td>←</td>
</tr>
<tr>
<td>\kappa</td>
<td>κ</td>
<td>\chi</td>
<td>χ</td>
<td>\Omega</td>
<td>Ω</td>
<td>\uparrow</td>
<td>↑</td>
</tr>
<tr>
<td>\lambda</td>
<td>λ</td>
<td>\psi</td>
<td>ψ</td>
<td>\Im</td>
<td>ℑ</td>
<td>\rightarrow</td>
<td>→</td>
</tr>
<tr>
<td>\mu</td>
<td>μ</td>
<td>\omega</td>
<td>ω</td>
<td>\Re</td>
<td>ℜ</td>
<td>\downarrow</td>
<td>↓</td>
</tr>
</tbody>
</table>
## Special Symbols

<table>
<thead>
<tr>
<th>Character Sequence</th>
<th>Symbol</th>
<th>Character Sequence</th>
<th>Symbol</th>
<th>Character Sequence</th>
<th>Symbol</th>
<th>Character Sequence</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>\otimes</td>
<td>⊗</td>
<td>\forall</td>
<td>∀</td>
<td>\circ</td>
<td>°</td>
<td>\aleph</td>
<td>א</td>
</tr>
<tr>
<td>\cap</td>
<td>∩</td>
<td>\exists</td>
<td>∃</td>
<td>\pm</td>
<td>±</td>
<td>\wp</td>
<td>ω</td>
</tr>
<tr>
<td>\supset</td>
<td>⊃</td>
<td>\ni</td>
<td>∈</td>
<td>\sim</td>
<td>~</td>
<td>\oslash</td>
<td>∅</td>
</tr>
<tr>
<td>\int</td>
<td>∫</td>
<td>\cong</td>
<td>≡</td>
<td>\propto</td>
<td>∝</td>
<td>\supseteq</td>
<td>⊇</td>
</tr>
<tr>
<td>\lfloor</td>
<td>\lfloor</td>
<td>\approx</td>
<td>≈</td>
<td>\partial</td>
<td>∂</td>
<td>\subset</td>
<td>⊂</td>
</tr>
<tr>
<td>\perp</td>
<td>⊥</td>
<td>\cup</td>
<td>∪</td>
<td>\div</td>
<td>⊸</td>
<td>\nabla</td>
<td>∇</td>
</tr>
<tr>
<td>\wedge</td>
<td>∧</td>
<td>\subseteq</td>
<td>⊆</td>
<td>\infty</td>
<td>∞</td>
<td>\ldots</td>
<td>…</td>
</tr>
<tr>
<td>\rceil</td>
<td>\rceil</td>
<td>\in</td>
<td>∈</td>
<td>\cdotp</td>
<td>·</td>
<td>\prime</td>
<td>′</td>
</tr>
<tr>
<td>\vee</td>
<td>\vee</td>
<td>\lceil</td>
<td>\lceil</td>
<td>\neg</td>
<td>¬</td>
<td>\emptyset</td>
<td>∅</td>
</tr>
<tr>
<td>\langle</td>
<td>\langle</td>
<td>\rangle</td>
<td>\rangle</td>
<td>\times</td>
<td>×</td>
<td>\mid</td>
<td>\mid</td>
</tr>
<tr>
<td>\surd</td>
<td>\surd</td>
<td></td>
<td></td>
<td>\copyright</td>
<td>©</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Adding Equations to the Plot

```matlab
fplot('0.25*exp(-.006*t)*sin(0.2*t)',[0 900])
a='\fontsize{12}{\itAe}^{-\alpha \itt} \sin \beta \itt';
b='\alpha << \beta';
title(strcat(a,b))
xlabel('\bf\itTime, \musec')
ylabel('\fontname{Times}\fontsize{12}Amplitude')
```

\[ Ae^{-\alpha t} \sin \beta t \quad \alpha << \beta \]

```latex
\begin{align*}
  \Delta \tau &= \frac{\tau_f}{N-1} \\
  A_1 &= \begin{bmatrix}
    -1 & 0 \\
    1 & -1 \\
    -1 & 0
  \end{bmatrix} \\
  \dot{x}(t) &= \frac{dx}{d\tau} \frac{d\tau}{dt} = x'(\tau) \lambda(\tau)
\end{align*}
```

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
Multiple Plots and Line Specifications

```matlab
x=0:pi/20:pi;
y1=sin(x); y2=cos(x);
plot(x,y1), hold
plot(x,y2)
legend('sin','cos')
```

```matlab
x=0:pi/20:pi;
y1=sin(x); y2=cos(x);
plot(x,y1,x,y2)
legend('sin','cos')
```

```matlab
x=0:pi/20:pi;
y1=sin(x); y2=cos(x);
plot(x,y1,x,y2,'-.sm')
legend('sin','cos')
```
```
legend(....,'Location',LOC) adds a legend in the specified location, LOC, with respect to the axes. LOC may be either a 1x4 position vector or one of the following strings:

'North' ('n')    inside plot box near top
'South' ('s')    inside bottom
'East' ('e')     inside right
'West' ('w')     inside left
'NorthEast' ('ne') inside top right (default)
'NorthWest' ('nw') inside top left
'SouthEast' ('se') inside bottom right
'SouthWest' ('sw') inside bottom left
'NorthOutside' ('no') outside plot box near top
'SouthOutside' ('so') outside bottom
'EastOutside' ('eo') outside right
'WestOutside' ('wo') outside left
'NorthEastOutside' ('neo') outside top right
'NorthWestOutside' ('nwo') outside top left
'SouthEastOutside' ('seo') outside bottom right
'SouthWestOutside' ('swo') outside bottom left
'Best' ('b')     least conflict with data in plot
'BestOutside' ('bo') least unused space outside plot
```
## LineSpec Options

<table>
<thead>
<tr>
<th>Line Types</th>
<th>Data Markers†</th>
<th>Colors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid line</td>
<td>Dot (.)</td>
<td>Black (k)</td>
</tr>
<tr>
<td>Dashed line</td>
<td>Asterisk (*)</td>
<td>Blue (b)</td>
</tr>
<tr>
<td>Dash-dotted line</td>
<td>Cross (x)</td>
<td>Cyan (c)</td>
</tr>
<tr>
<td>Dotted line</td>
<td>Plus sign (+)</td>
<td>Green (g)</td>
</tr>
<tr>
<td></td>
<td>Circle (o)</td>
<td>Magenta (m)</td>
</tr>
<tr>
<td></td>
<td>Square (□)</td>
<td>Red (r)</td>
</tr>
<tr>
<td></td>
<td>Diamond (◇)</td>
<td>White (w)</td>
</tr>
<tr>
<td></td>
<td>Pentagram (star)</td>
<td>Magenta (m)</td>
</tr>
<tr>
<td></td>
<td>Hexagram (star of David)</td>
<td>Red (r)</td>
</tr>
<tr>
<td></td>
<td>Left triangle (◄)</td>
<td>Yellow (y)</td>
</tr>
<tr>
<td></td>
<td>Up triangle (▲)</td>
<td>Yellow (y)</td>
</tr>
<tr>
<td></td>
<td>Right triangle (►)</td>
<td>Yellow (y)</td>
</tr>
<tr>
<td></td>
<td>Down triangle (▼)</td>
<td>Yellow (y)</td>
</tr>
</tbody>
</table>

† Note that circle and all following markers are filled markers (you may fill them with color)

- **LineWidth** – specifies the width of the line (in points)
- **MarkerEdgeColor** – specifies the color of the marker or the edge color for filled markers
- **MarkerFaceColor** – specifies the color of the face of filled markers
- **MarkerSize** – specifies the size of the marker (in points)
Markers and Line Styles

```matlab
>> x=-pi:pi; y=sin(x);
>> line(x,y,'Color','r','Marker','p',
    'LineWidth',3,'MarkerSize',7)

>> x=-pi:pi; y=sin(x);
>> plot(x,y.^2,'--s','MarkerSize',10,...
    'MarkerFaceColor','r')

>> x=linspace(0,pi); plot(x,cos(6*x).*exp(-x),'-.^')

>> x=linspace(0,pi); xm=linspace(0,pi,20);
>> plot(x,cos(6*x).*exp(-x),xm,cos(6*xm).*exp(-xm),'^')
```
Utilizing Two y-axes

```matlab
x=linspace(0,15,200);
y1=10*exp(-0.36*x).*sin(2.7*x);
y2=200*exp(-0.05*x).*sin(0.8*x);
[ax,h1,h2]=plotyy(x,y1,x,y2);
xlabel('Time, s')
set(get(ax(1),'Ylabel'),'String','Short Period Motion ($\alpha, ^{\circ}$)')
set(get(ax(2),'Ylabel'),'String','Phugoid Motion (h, ft)')
seth2,'LineStyle',':'
title('T-37 Longitudinal Dynamics')
```

Prior to R2014b

![Graph showing short period motion and phugoid motion before R2014b](image)

After R2014b

![Graph showing short period motion and phugoid motion after R2014b](image)
Finding a Graphics Object

$$\begin{align*}
> & \text{x = linspace(-pi,pi,50);} \\
> & \text{hp = plot(x,sin(x),x,cos(x),'rp',x,sin(x).*cos(x),'m+:');} \\
> & \text{legend([hp(1) hp(3)],'f_1=sin(x)','f_2=sin(x)*cos(x)',0)} \\
> & \text{h1 = findobj('Marker','p');} \\
> & \text{delete(h1)} \\
> & \text{h2 = findobj('Tag','legend');} \\
> & \text{set(h2,'Color','g')} \\
\end{align*}$$

Comments:

a) the \text{h=findobj} call returns handles of the root object and all its descendants
b) the \text{get(gcf,'children')} call returns two handles, one of which is a legend handle

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
The subplot Function

```
subplot(221), ezplot('sin(x)')
subplot(222), ezplot('cos(x)')
subplot(223), ezplot('sin(x).^2')
subplot(224), ezplot('cos(x).^2')
```

```
subplot(2,2,1), ezplot('sin(x)')
subplot(2,2,2), ezplot('cos(x)')
subplot(2,2,3), ezplot('sin(x).*cos(x)')
```

```
subplot(2,2,1), ezplot('sin(x)')
subplot(2,2,2), ezplot('cos(x)')
subplot(2,2,[3 4]), ezplot('sin(x).*cos(x)')
```
Variety of the 2-D Plots

- loglog
- semilogy
- polar
- area
- pie
- barh
- hist
- errorbar
- compass
- feather
- stem
- stairs
- scatter
- fill
- triplot
Combining the Different-Type Plots

generation=1:5;
number=[4.2 33.6 42.7 9.5 1];
price=[0.1 0.2 1 30 150];
bar(generation,number,'y')
xlabel('Generation')
ylabel('Number manufactured, thousands')
h1=gca;
% Setting the new axes atop the first ones
h2=axes('Position',get(h1,'Position'));
% Adding the second plot to the new axes
semilogy(generation,price,'LineWidth',3)
% Modifying second axes settings
set(h2,'Color','none','YAxisLocation', ...%right', 'XLim',get(h1,'XLim'), ...%XTickLabel',[], 'TickLength',[0 0])
text(3.5,10,'Price','Rotation',58)
ylabel('Price, M$'), title('Jet Fighters')
Two Types of 3-D Plots

**Trajectories**

```matlab
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
grid on, axis square
```

**Surfaces**

```matlab
[x,y]=meshgrid(linspace(0,2*pi,100));
z=sin(x).*cos(y).^2;
plot3(x,y,z)
axis equal
```

```matlab
t=0:.1:10; y=exp(-(.1+i)*t);
stem3(real(y),imag(y),t)
hold on
plot3(real(y),imag(y),t,'r')
hold off, view(-39.5,62)
```

**view (az,el)**
Variety of Surface Plots

- mesh
- meshc
- meshz
- surf
- surfc
- waterfall
Similarities in 2-D and 3-D

quiver
contour
bar

quiver3
contour3
bar3

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
cylinder
h=findobj('Type','surface');
C=rand(size(get(h,'CData')));
set(h,'CData',C)
axis square

cylinder(1,8)
hc=get(gca,'Children')
rotate(hc,[1 1 1],45)
set(hc,'FaceColor','g')
axis equal

t=linspace(0,pi,50);
[X,Y,Z]=cylinder(2+cos(t));
surf(X,Y,Z)
axis square
axis off

sphere(10)
hold on
[x,y,z]=sphere;
surf(x+2,y,z)
axis equal

ellipsoid(0,0,0,2,10,1)
hold, axis equal
[x,y,z]=cylinder(0.5,40);
surf(x,y,z+0.5)
shading interp
view([8,14])
colormap colorcube

ellipsoid(0,0,0,2,2,1), hold
h=get(gca,'Children');
rotate(h,[3 3 3],34)
z=get(h,'Zdata'); shading flat
set(h,'Zdata',z+2), axis equal
hs=mesh([-2 2; -2 2],[0]*ones(2));
set(hs,'FaceColor','c')
Using Different Color Maps and colormap Editor

```matlab
>> colormap('name')
[x,y]=meshgrid(linspace(0,2*pi,100));
z=sin(x).*cos(y).^2;
surf(x,y,z)
colormap('Cool')
axis equal
set(gca,'Visible','off')
```

After R2014b

Prior to R2014b

All rights reserved. No part of this publication may be reproduced, distributed, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
fill3 and patch Functions

```matlab
x = [0 1; 1 0; 0 1];
y = [1 0; 1 1; 0 1];
z = [1 1; 1 1; 0 1];
c = [1 1; 0 0; 1 1];
fill3(x,y,z,c)
xlabel('x'), ylabel('y')
```

```matlab
t=linspace(0,2*pi,11);
x=cos(t); y=sin(t);
x(2:2:10)=.4*cos(t(2:2:10));
y(2:2:10)=.4*sin(t(2:2:10));
patch(x,y,'b')
axis square
```

```matlab
x=[0 0 0;0 1 1 0;0 1 1 0]';
y=[0 1 1 0;0 0 0;1 1 1 1]';
z=[0 0 1 1;0 0 1 1;0 0 1 1]';
patch(x,y,z,-z)
axis square, view([35,35])
zlabel('z'), ylabel('y')
```

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
Creating a 3-D Object Using Patch Graphics

```matlab
fx=[6.6 5.3 2.3 1.5 -2 -2.5 -10 -10 0 5.3 5.3]'; % fuselage
fy=zeros(11,1);
fz=[1.2 1.7 1.9 1.2 1.2 2.6 1.1 0.7 -0.2 0.3 0.8]';
cx=[2.3 0 -2.5 -2 1.5]';        % cockpit canopy
cy=zeros(5,1);
cz=[1.9 3.1 2.6 1.2 1.2]';
wingx=2.3*[1 -3 -3+6 -3+6 -3 -3 -3+6 -3+6 -3 -3 -3 1]'/4; % wing
wingy=[8.5 8.5 7.8 7.8 2.2 2.2 -2.2 -2.2 -7.8 -7.8 -8.5 -8.5]';
wingz=zeros(12,1);
ax=2.3*[3+6 -3 -3 -3+6]'/4;  % right aileron
ay=[7.8 7.8 2.2 2.2]';
az=zeros(4,1);
hsx=1.3*[0 -1 -1 0]'-8.1;           % horizontal stabilizer
hsy=5.1*[1 1 -1 -1]'/2;
hsz=0.7*ones(4,1);
ex=0.6*[0 -1 -1 0]'-9.4; % elevator
ey=5.1*[1 1 -1 -1]'/2;
ez=0.7*ones(4,1);
vsx=[0 -1.7 -2.7 -1.8]'-7.3; % vertical stabilizer
vsy=zeros(4,1);
vsz=[1.65 4.5 4.5 1.25]';
rx=[-2.7 -3.3 -3.1 -1.8]'-7.3; % rudder
ry=zeros(4,1);
rz=[4.5 4.5 1.5 1.25]';
pf=patch(fx,fy,fz,'c'); pw=patch(wingx,wingy,wingz,'c');
pc=patch(cx,cy,cz,'b'); ps=patch([hsx vsx],[hsy vsy],[hsz vsz],'c');
pu=patch([ax ax ex rx],[ay -ay ey ry],[az az ez rz],'m')
haircraft=[pf pc pw ps pu];
axis equal, axis off, view(135,20)
for i=1:1000
    rotate(haircraft,[25,35],1)
    pause(0.001)
end
```

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
Property Editor

Figure 1

Property Editor: `graph2d.lineseries`

- BeingDeleted: off
- BusyAction: queue
- ButtonDownFcn
- Clipping: on
- Color
- CreateFcn
- DeleteFcn
- DisplayName: null
- EraseMode: normal
- HandleVisibility: on
- HitTest: on
- Interruptible: on
-LineStyle: -
- LineWidth: 0.5
- Marker: none
- MarkerEdgeColor: auto
- MarkerFaceColor: none
- MarkerSize: 6.0
- SelectionHighlight: on
- Tag
- UserData: 0x0 double array
- Visible: on

Property Editor: `lineseries`

- Display Name
- Plot Type: Line
- Line: 0.5
- Marker: none
- Marker Size: 6.0
- More Properties...
Saving Plots

print ('-dbitmap','Name')
print ('-dtiffn','Name')
print ('-dbmp','Name')
print ('-djpe','Name')
plot(sin(0:pi/20:pi))
Accessing Graphics Properties via Property Editor
Graphics Objects Hierarchy

```
plot(sin(0:pi/20:pi))
```
```
h_line= plot(sin(0:pi/20:pi));
get(h_line)
```

- **Color**: [0 0 1]
- **EraseMode**: 'normal'
- **LineStyle**: '-'
- **LineWidth**: 0.5000
- **Marker**: 'none'
- **MarkerSize**: 6
- **MarkerEdgeColor**: 'auto'
- **MarkerFaceColor**: 'none'

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, without permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
Exploiting Parent – Children Relationship

Comment: any object has a single parent, but may have several children, e.g. several axes (and legends) within one figure, several lines within one axis.
Changing Graphics Object Properties

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

```
get(h_line)
```

```
Color: [0 0 1]
EraseMode: 'normal'
LineStyle: '-'
LineWidth: 0.5000
Marker: 'none'
MarkerSize: 6
MarkerEdgeColor: 'auto'
MarkerFaceColor: 'none'
... 
```

```
set(h_line,'Color','r','LineWidth',2.5)
```

```
set(h_line,'ydata',d)
```
figure('Renderer','zbuffer')

% Defining a membrane
r = [0:0.05:1]';            % Radius vector
phi = 0:pi/20:2*pi;         % Phi angle vector
x = r*cos(phi);            % x-coordinates of a grid
y = r*sin(phi);            % y-coordinates of a grid
z = besselj(1,3.8316*r)*cos(phi);

% Plotting the membrane
mesh(x,y,z)
xlabel('x-axis'), ylabel('y-axis')
zlabel('z-axis'), axis tight
YY=axis; set(gca,'zlim',YY(5:6))
set(gca,'nextplot','replacechildren');

% Creating movie frames
for j = 1:20
    mesh(x,y,sin(2*pi*j/20)*z,z);
    F(j) = getframe;
end

% Starting the movie
k=questdlg('Ready to watch the movie?','Start the Movie', 'Yes', 'No', 'Yes');

% Playing the movie ten times
if char(k(1))=='Y'
    movie(F,10)
end

% Plotting a sinusoid
x=0:0.1:2*pi;            % Defines the x scale
y=sin(x);                % Computes sin(x)
z=cos(x);                % Computes cos(x)
plot(x,y)                % Plots sin(x) curve
set(gcf,'Renderer','zbuffer');
set(gca,'xlim',[0 2*pi],'ylim',[-1 1]);
set(gca,'XTick',[0:pi:2*pi])
set(gca,'XTickLabel',{0;'\pi';'2\pi'})
xlabel('x'), ylabel('y=f(x)')

% Getting a handle to the line
h_line=get(gca,'children');

% Changing line properties
for i=1:1000
    pause(0.005)
    % Setting the weighting coefficient w
    w=i/1000;
    % Blending sin(x) and cos(x) using w
    d=(1-w)*y+w*z;
    % Changing ydata for the line
    set(h_line,'ydata',d,'EraseMode','none');
end

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, without written permission of the Publisher. For permission to contact the American Institute of Aeronautics and Astronautics, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191.
function R=Euler2DCM(psi,theta,phi)

Rpsi = [ cos(psi) sin(psi) 0; ... 
        -sin(psi) cos(psi) 0; 0 0 1];
Rtheta = [cos(theta) 0 -sin(theta); 0 1 0; ... 
          sin(theta) 0 cos(theta)];
Rphi = [1 0 0; 0 cos(phi) sin(phi); ... 
        0 -sin(phi) cos(phi)]; ...
R = Rphi*Rtheta*Rpsi;

end
Embedded Means for Animations

- `comet` instead of `plot`
- `comet3` instead of `plot3`
% Plotting a sinusoid
x=0:0.1:2*pi;  % Defines the x scale
y=sin(x);      % Computes sin(x)
z=cos(x);      % Computes cos(x)
plot(x,y)       % Plots sin(x) curve
% Setting the Renderer property to zbuffer or Painters works around limitations of getframe with the OpenGL renderer on some Windows systems.
set(gcf,'Renderer','zbuffer');
set(gca,'xlim',[0 2*pi],'ylim',[-1 1]);
set(gca,'XTick',[0:pi:2*pi])
set(gca,'XTickLabel',{'0';'pi';'2pi'})
xlabel('x'), ylabel('y=f(x)')
% Getting a handle to the line
h_line = get(gca,'children');

% Getting a handle to the line
h_line = get(gca,'children');

aviobj = VideoWriter('sin2cos.avi');
open(aviobj)
% Changing line properties
for i=1:100
% Setting the weighting coefficient w
w = i/100;
% Blending sin(x) and cos(x) using w
  d = (1-w)*y + w*z;
% Changing ydata for the line
set(h_line,'ydata',d,EraseMode','normal');
F = getframe;
writeVideo(aviobj,F);
end
close(aviobj);
Exploring Different Codices

Note, that the Indeo5 codec (used by default by the movie2avi and avifile functions) and others were shipped with Windows XP, Windows XP SP2, and prior versions, however they are not shipped with the following operating systems: Windows XP SP1, Windows XP x64, Windows Vista (32/64), and Windows 7 (32/64). You need to install these codecs separately or use no compression.

The frame height and width will be padded to be a multiple of four as required by majority of codices.

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.

After R2014b VideoWriter should be used (movie2avi is removed)
Graphical Input From Mouse

axis([0 10 0 10]), hold on
xy = []; n = 0; % Initially, the list of points is empty
%% Picking up multiple points by clicking left mouse button
% (Right mouse button means you are picking the last point)
but = 1;
while but == 1
    [xi,yi,but] = ginput(1);
    plot(xi,yi,'ro')
    n = n+1;
    xy(:,n) = [xi;yi];
end
%% Interpolating with a spline curve and finer spacing
    t = 1:n; ts = 1: 0.1: n;
    xys = spline(t,xy,ts);
%% Plot the interpolated curve
    plot(xys(1,:),xys(2,:),'b-'); hold off

[X,Y] = ginput(N) gets N points from the current axes and returns the x- and y-coordinates in length N vectors X and Y (data points are entered by pressing a mouse button or any key on the keyboard except carriage return, which terminates the input before N points are entered.)
[X,Y,BUTTON] = ginput(N) returns a third result, BUTTON, that contains a vector of integers specifying which mouse button was used (1,2,3 from left).
Numerical Computing with MATLAB by Cleve Moler

www.mathworks.com/moler/ncmfilelist.html
### Experiments with MATLAB by Cleve Moler

#### Table of Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>fern</td>
<td><img src="image" alt="fern" /></td>
</tr>
<tr>
<td>biorhythm</td>
<td><img src="image" alt="biorhythm" /></td>
</tr>
<tr>
<td>clockex</td>
<td><img src="image" alt="clockex" /></td>
</tr>
<tr>
<td>lifex</td>
<td><img src="image" alt="lifex" /></td>
</tr>
<tr>
<td>rabbits</td>
<td><img src="image" alt="rabbits" /></td>
</tr>
<tr>
<td>pagerank</td>
<td><img src="image" alt="pagerank" /></td>
</tr>
<tr>
<td>wiggle</td>
<td><img src="image" alt="wiggle" /></td>
</tr>
<tr>
<td>t_puzzle</td>
<td><img src="image" alt="t_puzzle" /></td>
</tr>
<tr>
<td>tictactoe</td>
<td><img src="image" alt="tictactoe" /></td>
</tr>
<tr>
<td>backslash</td>
<td><img src="image" alt="backslash" /></td>
</tr>
<tr>
<td>predprey</td>
<td><img src="image" alt="predprey" /></td>
</tr>
<tr>
<td>mandelbrot</td>
<td><img src="image" alt="mandelbrot" /></td>
</tr>
<tr>
<td>durerperm</td>
<td><img src="image" alt="durerperm" /></td>
</tr>
<tr>
<td>waterwave</td>
<td><img src="image" alt="waterwave" /></td>
</tr>
<tr>
<td>expgui</td>
<td><img src="image" alt="expgui" /></td>
</tr>
<tr>
<td>touchtone</td>
<td><img src="image" alt="touchtone" /></td>
</tr>
<tr>
<td>censusgui</td>
<td><img src="image" alt="censusgui" /></td>
</tr>
<tr>
<td>swinger</td>
<td><img src="image" alt="swinger" /></td>
</tr>
<tr>
<td>walker</td>
<td><img src="image" alt="walker" /></td>
</tr>
<tr>
<td>exmlogo</td>
<td><img src="image" alt="exmlogo" /></td>
</tr>
<tr>
<td>exm_book</td>
<td><img src="image" alt="exm_book" /></td>
</tr>
<tr>
<td>helpwin exmgui</td>
<td><img src="image" alt="helpwin exmgui" /></td>
</tr>
<tr>
<td>helpwin exam</td>
<td><img src="image" alt="helpwin exam" /></td>
</tr>
<tr>
<td>close</td>
<td><img src="image" alt="close" /></td>
</tr>
</tbody>
</table>

---

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted, unless for course participation, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the Publisher and/or Author. Contact the American Institute of Aeronautics and Astronautics, Professional Development Programs, Suite 500, 1801 Alexander Bell Drive, Reston, VA 20191-4344.
The End of Chapter 6

Questions?