Chapter 6. 2-D and 3-D Plotting and Animation
Outline

• 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

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

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

fplot('sin(x)',[0,pi])
```
```matlab
ezplot('sin(x)')
```

```matlab
gfigure
gaxis

title
xlabel

title

Default line color: [0.00,0.45,0.74]  
(before R2014b used to be [0,0,1])

Brush/Select Data
Data Tips
Pan
Zoom In
Zoom Out
Restore View

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.
### Plotting Functions Difference

<table>
<thead>
<tr>
<th>Plotting Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>plot(X,Y,…)</code></td>
<td>Creates a 2-D plot of data in <code>Y</code> versus the corresponding values in <code>X</code> (this function creates the figure, axes and line objects)</td>
</tr>
<tr>
<td><code>line(X,Y,…)</code></td>
<td>Plots a line in the current axes using the data in vectors <code>X</code> and <code>Y</code> (or multiple lines if either <code>X</code> and <code>Y</code>, or both are matrices). Unlike the <code>plot</code> function, <code>line</code> adds the line to the current axes without deleting other graphics objects or resetting axes properties.</td>
</tr>
<tr>
<td><code>ezplot(f,…)</code></td>
<td>Plots symbolic expression, equation, or function <code>f(x)</code> or <code>f(x, y)</code>. The default range for <code>x</code> or both <code>x</code> and <code>y</code> is <code>[-2π; 2π]</code></td>
</tr>
<tr>
<td><code>ezplot(funx,funy,…)</code></td>
<td>Plots the parametrically defined planar curve <code>x = funx(t)</code> and <code>y = funy(t)</code> over the default range <code>0 &lt;= t &lt;= 2π</code> (similar function: <code>ezplot3</code>)</td>
</tr>
<tr>
<td><code>fplot(f,…)</code></td>
<td>Plots the curve defined by the function <code>y = f(x)</code> over the default interval <code>[-5; 5]</code> for <code>x</code> (similar functions: <code>fsurf</code>, introduced in R2016a)</td>
</tr>
<tr>
<td><code>fplot(funx,funy,…)</code></td>
<td>Plots the curve defined by <code>x = funx(t)</code> and <code>y = funy(t)</code> over the default interval <code>[-5; 5]</code> for <code>x</code> (similar functions: <code>fplot3</code> and <code>fsurf</code>, both introduced in R2016a)</td>
</tr>
<tr>
<td><code>fimplicit(f,…)</code></td>
<td>Plots the implicit function defined by <code>f(x,y) = 0</code> over the default interval <code>[-5; 5]</code> for <code>x</code> and <code>y</code> (introduced in R2016b) (similar functions <code>fcontour</code> and <code>fimplicit3</code>, introduced in R2016a and R2016b, respectively)</td>
</tr>
</tbody>
</table>
Enhancing Plot Readability

**title('text')**  
- places text in a title at the top of a plot

**title({'First line';'Second line'})**  
- create a two-line title

**sgtitle('text')**  
- adds a title above the grid of subplots in the current figure (introduced in R2018b)

**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 mouse click

**annotation(annotation_type)**  
- creates the specified annotation type, such as line, arrow, double arrow, text arrow, textbox, ellipse, and rectangle

**xticks(ticks)**  
- sets the x-axis tick values (the locations along the x-axis where the tick marks appear). Specify ticks as a vector of increasing values. (Introduced in R2016b.) Similarly, use the yticks and zticks functions

**xticklabels(labels)**  
- sets the x-axis tick labels for the current axes. Specify labels as a string array or a cell array of character vectors. (Introduced in R2016b.) Also, see xtickangle, xtickformat and others

---

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.
Enhancing Plot Readability

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. The grid minor command toggles the visibility of the minor grid lines. (note, not all types of charts support minor grid lines.)
grid, axis, and Annotations

```matlab
>> 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')
```

>> t=linspace(0,2*pi);
>> line(cos(t),sin(t)), axis equal
Text Strings Modifiers

Stream modifiers:

- \bf - bold font
- \it - italics font
- \sl - oblique font (rarely available)
- \rm - normal font
- \fontname\{fontname\} - specify the name of the font family to use
- \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: \ll, \{, \}, \_, \^.
<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>Δ</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>θ</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>ζ</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>i</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>
<th>Character Sequence</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>\otimes</td>
<td>\otimes</td>
<td>\forall</td>
<td>\forall</td>
<td>\circ</td>
<td>\circ</td>
<td>\aleph</td>
<td>\aleph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\cap</td>
<td>\cap</td>
<td>\exists</td>
<td>\exists</td>
<td>\pm</td>
<td>\pm</td>
<td>\wp</td>
<td>\wp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\supset</td>
<td>\supset</td>
<td>\ni</td>
<td>\ni</td>
<td>\sim</td>
<td>\sim</td>
<td>\oslash</td>
<td>\oslash</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\int</td>
<td>\int</td>
<td>\cong</td>
<td>\cong</td>
<td>\propto</td>
<td>\propto</td>
<td>\supseteq</td>
<td>\supseteq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\rfloor</td>
<td>\rfloor</td>
<td>\approx</td>
<td>\approx</td>
<td>\partial</td>
<td>\partial</td>
<td>\subset</td>
<td>\subset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\lfloor</td>
<td>\lfloor</td>
<td>\oplus</td>
<td>\oplus</td>
<td>\bullet</td>
<td>\bullet</td>
<td>\o</td>
<td>\o</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\perp</td>
<td>\perp</td>
<td>\cup</td>
<td>\cup</td>
<td>\div</td>
<td>\div</td>
<td>\nabla</td>
<td>\nabla</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\wedge</td>
<td>\wedge</td>
<td>\subseteq</td>
<td>\subseteq</td>
<td>\infty</td>
<td>\infty</td>
<td>\ldots</td>
<td>\ldots</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\rceil</td>
<td>\rceil</td>
<td>\in</td>
<td>\in</td>
<td>\cdot</td>
<td>\cdot</td>
<td>\prime</td>
<td>\prime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\lor</td>
<td>\lor</td>
<td>\lceil</td>
<td>\lceil</td>
<td>\neg</td>
<td>\neg</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\langle</td>
<td>\langle</td>
<td>\rangle</td>
<td>\rangle</td>
<td>\times</td>
<td>\times</td>
<td>\mid</td>
<td>\mid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\surd</td>
<td>\surd</td>
<td>\copyright</td>
<td>\copyright</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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.
Adding Equations to the Plot

```matlab
fplot('0.25*exp(-0.006*t)*sin(0.2*t)',[0 900])
a='\text{\textit{Ae}}^{\text{\textit{-}}(\alpha \text{\scriptsize{\textit{tt}}})}\text{\textit{sin}\beta}^{\text{\textit{tt}}};
b='\\ \alpha<<\beta';
title(strcat(a,b))
xlabel('\textbf{\textit{Time}} \text{\mu\text{sec}}')
ylabel('\text{\textit{fontname\textit{\text{\textsize{Amplitude}}}}}')
```

You may also try `text(.3,.3,texlabel('\text{\textit{beta12}}e^{\text{-}}(\alpha t)'))`, which uses the `texlabel` function to convert Greek variable names into a character vector that is displayed as Greek letters: $\beta_{12}e^{-\alpha t}$

\[\Delta \tau = \frac{\tau_f}{N-1}\]
\[A_1 = \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ -1 & 0 \end{bmatrix}\]
\[\dot{x}(t) = \frac{dx}{d\tau} \frac{d\tau}{dt} = x'(\tau)\lambda(\tau)\]
Examples of using LaTeX Syntax

The easiest way to visualize what additional capabilities the LaTeX interpreter of MATLAB offers is to click on the Insert equation icon of the Live Editor's Insert tab, which opens the Equation tab.

\[ \sum_{i=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} \]

\[ \lim_{x \to \infty} \exp(-x) = 0 \]

For advanced users
Multiple Plots and Line Specifications

```matlab
x=0:pi/20:pi;
y1=sin(x); y2=cos(x);
plot(x,y1), hold on
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. LocC 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

Legend

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.
# 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>Yellow y</td>
</tr>
<tr>
<td></td>
<td>Hexagram (star of David)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Left triangle (◄)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Up triangle (▲)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Right triangle (►)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Down triangle (▼)</td>
<td></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

```
>> 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,'--sb','MarkerSize',10,...
      'MarkerFaceColor','r')
```

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

```
>> x=linspace(0,pi); xm=linspace(0,pi,20);
>> plot(x,cos(6*x).*exp(-x),xm,cos(6*xm).*exp(-xm),'^')
```
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, ^o)')
set(get(ax(2),'Ylabel'),'String','Phugoid Motion (h, ft)')
set(h2,'LineStyle',':')
title('T-37 Longitudinal Dynamics')

Starting from R2014b

Prior to R2014b
Finding a Graphics Object

```matlab
>> x = linspace(-pi,pi,50);
>> hp = plot(x,sin(x),'b-',x,cos(x),'rp',x,sin(x).*cos(x),'m+:');
>> legend([hp(1) hp(3)],'f_1=sin(x)','f_2=sin(x)*cos(x)','location','se')
```

```matlab
>> h1 = findobj('Marker','p');
>> delete(h1)
>> h2 = findobj('Tag','legend');
>> set(h2,'Color','g')
or
>> h2.Color='g';
```

Comments:
- a) the `h=findobj` call returns handles of the root object and all its descendants
- b) the `get(gcf,'children')` call returns 2x1 graphics array composed of two handles, one of which is a legend handle
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 4]), ezplot('sin(x).*cos(x)')
```
More About the Grids

H=subplot(221); ezplot('sin(x)')

creates a handle to axes object properties. By changing property values, you can modify certain aspects of the axes, including those of the grid, e.g. H.GridColor='r';

<table>
<thead>
<tr>
<th>Axes Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTick, YTick, ZTick</td>
<td>Location of tick marks and major grid lines for each axis direction</td>
</tr>
<tr>
<td>XGrid, YGrid, ZGrid</td>
<td>Display of major grid lines for each axis direction</td>
</tr>
<tr>
<td>XMinorGrid,</td>
<td>Display of minor grid lines for each axis direction</td>
</tr>
<tr>
<td>YMinorGrid,</td>
<td></td>
</tr>
<tr>
<td>ZMinorGrid</td>
<td></td>
</tr>
<tr>
<td>LineWidth</td>
<td>Line width of grid lines, axes box outline, and tick marks</td>
</tr>
<tr>
<td>GridLineStyle</td>
<td>Major grid line style</td>
</tr>
<tr>
<td>MinorGridLineStyle</td>
<td>Minor grid line style</td>
</tr>
<tr>
<td>GridColor</td>
<td>Major grid line color</td>
</tr>
<tr>
<td>MinorGridColor</td>
<td>Minor grid line color</td>
</tr>
<tr>
<td>GridAlpha</td>
<td>Major grid line transparency</td>
</tr>
<tr>
<td>MinorGridAlpha</td>
<td>Minor grid line transparency</td>
</tr>
<tr>
<td>Layer</td>
<td>Location of grid lines in relation to the plotted data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PolarAxes Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThetaTick, RTick</td>
<td>Location of tick marks and major grid lines for each axis direction</td>
</tr>
<tr>
<td>ThetaGrid, RGrid</td>
<td>Display of major grid lines for each axis direction</td>
</tr>
<tr>
<td>ThetaMinorGrid,</td>
<td>Display of minor grid lines for each axis direction</td>
</tr>
<tr>
<td>RMinorGrid</td>
<td></td>
</tr>
<tr>
<td>LineWidth</td>
<td>Width of outline, tick marks, and grid lines</td>
</tr>
<tr>
<td>GridLineStyle</td>
<td>Major grid line style</td>
</tr>
<tr>
<td>MinorGridLineStyle</td>
<td>Minor grid line style</td>
</tr>
<tr>
<td>GridColor</td>
<td>Major grid line color</td>
</tr>
<tr>
<td>MinorGridColor</td>
<td>Minor grid line color</td>
</tr>
<tr>
<td>GridAlpha</td>
<td>Major grid line transparency</td>
</tr>
<tr>
<td>MinorGridAlpha</td>
<td>Minor grid line transparency</td>
</tr>
<tr>
<td>Layer</td>
<td>Location of grid lines in relation to the plotted data</td>
</tr>
</tbody>
</table>

For advanced users
%% Hourly air and dew point temperatures
A=[
60 42
63 42
...
];
%% Hourly sky condition
B=[
"Fair"
"Fair"
"Fair"
...
];
C=nan(length(A),1); C(B~="Fair")=3;

stime=datenum('02-01-2018 00:54');
etime=datenum('02-03-2018 23:54');
t=linspace(stime,etime,3*24);
plot(t,A,'.-.',t,A(:,1)-A(:,2),'m.--'), grid, hold
bar(t,C,'y')
datetick('x','HH:MM PM')
xlabel('Time'), ylabel('Temperature, ^oF')
legend('Air','Dew point','\Delta T','Overcast','Location','E')
Variety of the 2-D Plots

loglog

semilogx

semilogy

area

pie

barh

hist

errorbar

stackedplot

feather

stem

stairs

scatter

fill

triplot

Introduced in R2018b

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.

23 out of 51
% Defining axes using the subplot function
subplot 211
    plot(P(:,1),P(:,2)/1000,'o-.','markersize',3), grid, hold
    plot(P(end-2:end,1),P(end-2:end,2)/1000,'pr','markersize',6)
    xlabel('Year'), ylabel('Population, billion')
    xlim([-8000 2100])
    set(gca,'Color',[0.98 0.98 0.85]);
subplot 212
    plot(P(2:end,1),diff(P(:,2))./diff(P(:,1)),...
         'o-.','markersize',3), grid, hold
    plot(P(end-1:end,1),diff(P(end-2:end,2))./diff(P(end-2:end,1)),...
         'pr','markersize',6)
    xlim([-8000 2100])
    xlabel('Year'), ylabel('Yearly increase, million')
    set(gca,'Color',[0.98 0.98 0.85]);

% Explicitly-defined axes positions (for two insert plots)
ax1=axes('Position',[0.2 0.68 0.5 0.22]);
semilogy(ax1,P(:,1),P(:,2)/1000,'o-.',...
         'markersize',2), grid, hold
semilogy(ax1,P(end-2:end,1),P(end-2:end,2)/1000,pr',...
         'markersize',4)
xlim([-4000 2100]), ylabel('log scale, billion')
ax2=axes('Position',[0.2 0.20 0.5 0.22]);
semilogy(ax2,P(2:end,1),diff(P(:,2))./diff(P(:,1)),...
         'o-.',...
         'markersize',2), grid, hold
semilogy(ax2,P(end-1:end,1),diff(P(end-2:end,2))./diff(P(end-2:end,1)),...
         'pr','markersize',4)
xlim([-4000 2100]), ylabel('log scale, million')

P=[-10000 1
-8000 5
-6500 5
...];
Overlaying One Plot over Another

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')
The Polar Plots

- **polarplot**
- **polarscatter**
- **polarhistogram**
- **compass**

**Introduced in R2016b**

- `rlim` sets or queries the r-axis limits for polar axes
- `rticks` sets or queries the r-axis tick values
- `rticklabels` sets or queries the r-axis tick labels
- `rtickformat` specifies the r-axis tick label format
- `rtickangle` rotates the r-axis tick labels
- `thetalim` sets or queries the theta-axis limits for polar axes
- `thetaticks` sets or queries the theta-axis tick values
- `thetaticklabels` sets or queries the theta-axis tick labels
- `thetatickformat` specifies the theta-axis tick label format
- `polaraxes` creates polar axes

```matlab
theta=0:0.01:2*pi;
rho=10*sin(.5*theta).*cos(.5*theta+pi/2);
pax=polaraxes;
polarplot(theta,rho,'-.','LineWidth',2)
pax.ThetaDir='clockwise';
pax.ThetaZeroLocation='top';
pax.FontSize = 12;
pax.ThetaColor='r';
rticks([1 5 9])
rticklabels({'r=1nmi','r=5nmi','r=9nmi'})
thetaticks(0:90:270)
thetaticklabels({'N','E','S','W'})
```
Two Types of 3-D Plots

Trajectories

t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t,'b')
grid on, axis square

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

Surfaces

[X,Y]=meshgrid(linspace(0,2*pi,100));
Z=sin(X).*cos(Y).^2;
plot3(X,Y,Z)
axis equal

view(az,el)
Variety of the Surface Plots

All these functions allow passing in just the grid vectors as the first two arguments (i.e. not necessarily a full grid matrices)
Similarities in 2-D and 3-D

quiver
contour
bar

quiver3
contour3
bar3
Shapes and Shades

The default is `shading faceted`

```matlab
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

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

e llipsoid(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

e llipsoid(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],...
[-2 -2; 2 2], 2.5*ones(2));
set(hs, 'FaceColor', 'c')
```

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.
Surface and Patch Objects

Lighting

camlight('headlight'), camlight('right'), camlight('left') - creates a light at the camera position, right and up from camera (default), and left and up from camera, respectively.
camlight(az,el), lightangle(az,el) - creates a light at the specified azimuth (az) and elevation (el) with respect to the camera position.
camlight(...,'style') - defines the style argument using one of the following two values:
   local (default) - the light is a point source that radiates from the location in all directions
   infinite - the light shines in parallel rays
lighting flat - produces uniform lighting across each of the faces of the faceted object
lighting gouraud - calculates the vertex normals and interpolates linearly across the curved faces
material shiny - the object has a high specular reflectance relative to the diffuse and ambient light, and the color of the specular light depends only on the color of the light source
material dull - the object reflects more diffuse light and has no specular highlights, but the color of the reflected light depends only on the light source
material metal - the object has a very high specular reflectance, very low ambient and diffuse reflectance, and the color of the reflected light depends on both the color of the light source and the color of the object

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.
Using Different Color Maps and colormap Editor

```
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')
```

Prior to R2014b

```
>> colormappeditor
```

After R2014b
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 1 1]';
y=[0 1 1 0; 0 0 0; 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')
```

Jet

Parula
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 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)
end
pause(0.001)

```

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 (before R2018b)
Property Inspector (since R2018b)

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 (since R2018b)
Saving Plots

```matlab
print('-dbitmap', 'Name')
print('-dtnf', 'Name')
print('-dbmp', 'Name')
print('-djpeg', 'Name')
```
Accessing Graphics
Properties via Property Editor
Graphics Objects Hierarchy

Root (Computer Screen)

Figure

Core Graphics Objects

Hidden Annotation Axes

UI Objects

Parent

Children

Figure

Group Objects

Annotation Objects

Uicontrol
Uimenu
Uicontextmenu
Uipanel

Core Graphics Objects

Uipushtool
Uitoggletool

Image
Light
Line
Patch
Rectangle
Surface
Text

Core Graphics Objects

Prior to R2014b

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'
...
Exploiting Parent – Children Relationship

Figure

Axis

Line

% Figure

`h_figure = get(gcf, 'children')`

% Axis

`h_axis = get(gcf, 'children')`

% Line

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

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

Prior to R2014b

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

Starting from R2014b

```matlab
H.PropertyName = PropertyValue
```

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

```matlab
set(h_line, 'Color', 'r', 'LineWidth', 2.5)
set(h_line, 'ydata', get(h_line, 'ydata')*5)
```

```matlab
h_line.Color = 'r';
h_line.LineWidth = 2.5;
h_line.YData = 5*h_line.YData;
```
Creating Animations

% 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);  % z-coordinates of a grid

% 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
figure('color','w')
quiver3(0,0,0,1.5,0,0,'LineWidth',2), hold on
quiver3(0,0,0,0,1.5,0,'LineWidth',2)
quiver3(0,0,0,0,0,1.5,'LineWidth',2)
    text(1.5,0,0,'n_1'); text(0,1.5,0,'n_2'); text(0,0,1.5,'n_3'
axis([-1 1 -1 1 -1 1]), view(130,30)
xlabel('x_i'), ylabel('y_i'), zlabel('z_i');
R=eye(3);
h(1)=quiver3(0,0,0,R(1,1),R(1,2),R(1,3),'m','Linewidth',3);
h(2)=quiver3(0,0,0,R(2,1),R(2,2),R(2,3),'m','Linewidth',3);
h(3)=quiver3(0,0,0,R(3,1),R(3,2),R(3,3),'m','Linewidth',3);
    ht(1)=text(R(1,1),R(1,2),R(1,3),'bf{b_1}');
    ht(2)=text(R(2,1),R(2,2),R(2,3),'bf{b_2}');
    ht(3)=text(R(3,1),R(3,2),R(3,3),'bf{b_3}');
ha(1)=text(-0.5,0,0, ['phi = 0 ^o']);
ha(2)=text(-0.5,0,0.2, ['\theta = 0 ^{\circ}']);
ha(3)=text(-0.5,0,0.4, ['psi = 0 ^{\circ}']);
for i = 1:200
    psi=4*pi*(i-1)/99; phi=2*pi*(i-1)/99; theta=pi*(i-1)/99;
    R=Euler2DCM(psi,theta,phi);
        for j=1:3
            set(h(j),'UData',R(j,1),'VData',R(j,2),'WData',R(j,3));
            set(ht(j),'Position',[R(j,:)]);
        end
    phid=mod(phi*180/pi,360);     if phid>180, phid=phid-360;     end
    thetad=mod(theta*180/pi,360); if thetad>180, thetad=thetad-360; end
    psid=mod(psi*180/pi,360);
        set(ha(1),'String', ['\phi = ' int2str(phid) ' ^{\circ}']);
        set(ha(2),'String', ['\theta = ' int2str(thetad) ' ^{\circ}']);
        set(ha(3),'String', ['\psi = ' int2str(psid) ' ^{\circ}']);
    pause(0.01)
end
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;

Replaced with angle2dcm in 2017a
% 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');

% 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

% 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);
F = getframe;
writeVideo(aviobj,F);
end

close(aviobj);
Creating Animated Lines

```matlab
x = linspace(0, 4*pi, 400);
y = sin(x) + 0.5*rand(1, length(x));
h = animatedline('Marker', 'p', 'Color', 'b', 'LineStyle', '-.);
axis([min(x), max(x), min(y), max(y)]), grid
xlabel("x-axis"), ylabel("y-axis")
for k = 1:length(x)
    addpoints(h, x(k), y(k)); % Introduced in R2016b
    drawnow
end
```

- `comet` instead of `plot`
- `comet3` instead of `plot3`
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.

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).
<table>
<thead>
<tr>
<th>EXM GUI</th>
<th>EXM GUI</th>
<th>EXM GUI</th>
<th>EXM GUI</th>
<th>EXM GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>fern</td>
<td>biorhythm</td>
<td>clockex</td>
<td>lifex</td>
<td>rabbits</td>
</tr>
<tr>
<td>pagerank</td>
<td>wiggle</td>
<td>t_puzzle</td>
<td>tictactoe</td>
<td>backslash</td>
</tr>
<tr>
<td>predprey</td>
<td>mandelbrot</td>
<td>durerperm</td>
<td>waterwave</td>
<td>expgui</td>
</tr>
<tr>
<td>touchtone</td>
<td>censusgui</td>
<td>swinger</td>
<td>walker</td>
<td></td>
</tr>
</tbody>
</table>

Solve $A^x = b$
The End of Chapter 6

Questions?