MatLab Graphics

2 February 2004
Jacob D. Furst
2D Graphing

- Collect your data
- Prepare a place to put your graph
- Plot your graph
- If desired
  - Modify lines
  - Modify axes
  - Annotate
  - Print
Collect Your Data

• MatLab can plot many different forms of data – easiest to plot two arrays, one with x and one with f(x)
  
x = -pi:pi/100:pi;
fx = sin(x);

• Will also plot a single array as f(x) with the indices of the array as x values
Preparing a Place

• Create a figure

```
figure(1);
```

• Create a place within the figure for your graph – only do this if you plan to have multiple graphs

```
subplot (2, 2, 1);
```

• The first two arguments indicate the number of rows and columns for subplots, the third the position of the graph
Plot Your Graph

• Basic 2D plots
  plot (x, fx);
• Plot multiple lines
  cosx = cos(x);
  plot (x, fx, x, cosx);
• MatLab will plot a matrix as column arrays
• X axis will be labeled 1 to number of columns
Plotting Variants

- loglog plots with both x and y axes logarithmic
- semilogx and semilogy make the x and y axis (respectively) log and the other linear
- plotyy allows you to have two y axes – one on the left and one on the right
What Else?

• MatLab allows you considerable control over the appearance of the graph
• Much of it can be specified interactively, by clicking in the figure window
• Much of it can also be controlled functionally, with MatLab commands
# Line Style

<table>
<thead>
<tr>
<th>Specifier</th>
<th>Line Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>solid line (default)</td>
</tr>
<tr>
<td>--</td>
<td>dashed line</td>
</tr>
<tr>
<td>:</td>
<td>dotted line</td>
</tr>
<tr>
<td>-.</td>
<td>dash-dot line</td>
</tr>
</tbody>
</table>
Example

\[ x = -\pi:pi/10:pi; \]
\[ \text{sine} = \sin(x); \]
\[ \text{cosine} = \cos(x); \]
\[ \text{plot} \ (x, \ \text{sine}, \ ':', \ x, \ \text{cosine}, \ '--)'); \]
# Marker Types

<table>
<thead>
<tr>
<th>Specifier</th>
<th>Marker Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>plus sign</td>
</tr>
<tr>
<td>o</td>
<td>circle</td>
</tr>
<tr>
<td>*</td>
<td>asterisk</td>
</tr>
<tr>
<td>.</td>
<td>point</td>
</tr>
<tr>
<td>x</td>
<td>cross</td>
</tr>
<tr>
<td>s</td>
<td>square</td>
</tr>
<tr>
<td>d</td>
<td>diamond</td>
</tr>
<tr>
<td>^</td>
<td>upward pointing triangle</td>
</tr>
<tr>
<td>v</td>
<td>downward pointing triangle</td>
</tr>
<tr>
<td>&gt;</td>
<td>right pointing triangle</td>
</tr>
<tr>
<td>&lt;</td>
<td>left pointing triangle</td>
</tr>
<tr>
<td>p</td>
<td>five-pointed star (pentagram)</td>
</tr>
<tr>
<td>h</td>
<td>six-pointed star (hexagram)</td>
</tr>
<tr>
<td>Specifier</td>
<td>Color</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>r</td>
<td>red</td>
</tr>
<tr>
<td>g</td>
<td>green</td>
</tr>
<tr>
<td>b</td>
<td>blue</td>
</tr>
<tr>
<td>c</td>
<td>cyan</td>
</tr>
<tr>
<td>m</td>
<td>magenta</td>
</tr>
<tr>
<td>y</td>
<td>yellow</td>
</tr>
<tr>
<td>k</td>
<td>black</td>
</tr>
<tr>
<td>w</td>
<td>white</td>
</tr>
</tbody>
</table>
Example

```matlab
x = -pi:pi/10:pi;
sine = sin(x);
 cosine = cos(x);
plot (x, sine, ':sr', x, cosine, '--ok');
```
Printing/Saving

- Use the `print` function to send your graph to the printer or to save as a file.
- You can save to a file by specifying the file type and name:
  ```
  print -dfileformat filename
  ```
- Some common file format include jpeg, tiff, bitmap and eps.
3D Graphing

- Occasionally, you want to plot a curve in 3D
- Uses the `plot3` function and three arrays (x, y, and z coordinates) to generate a curve in space
- The standard helix:
  
  ```matlab
  t = 0:pi/50:10*pi;
  plot3(sin(t), cos(t), t, '--or');
  ```
Plotting Surfaces

- Often, we’ll want to plot surfaces, rather than curves, in 3D
- MatLab offers both the `surf` and `mesh` functions
- Use `mesh` to create a wireframe surface
- Use `surf` to create a shaded 3D surface
Example

• Create the X and Y coordinate matrices
  \[ [X, Y] = \text{meshgrid}(-\pi:\pi/10:\pi); \]

• Calculate the function values
  \[ Z = \sin(X) \times \sin(Y); \]

• Plot the graph
  \text{mesh}(Z);
  \text{surf}(Z);
Caveat

• If your function is fairly smooth, plot fewer points and let MatLab interpolate the rest
• For example, change our last plot only a little, and look at the results:

```
[X, Y] =meshgrid(-pi:pi/100:pi);
Z = sin (X) .* sin (Y);
mesh (Z);
surf (Z);
```
Parametric Surfaces

- $X$, $Y$ and $Z$ are all matrices, in which the points of the surface are located at $X(i,j)$, $Y(i,j)$ and $Z(i,j)$

```matlab
theta = -pi:pi/10:pi;
phi = (-pi/2:pi/20:pi/2)';
X = cos(phi)*cos(theta);
Y = cos(phi)*sin(theta);
Z = sin(phi)*ones(size(theta));
surf (X, Y, Z);
```