MATLAB for the Sciences

Plotting, Simple Arrays, and Special Functions

Jon M. Ernstberger

January 9, 2008

Plotting in MATLAB

- Perhaps what makes MATLAB so wonderful is the ease of graphical output.
- We all want to see pretty pictures!
- In the past, for graphical output, you had two options:
    - Poor ASCII (DOS-type) graphics.
    - Save program data and export it to another interpreting program.
    - One is unprofessional and the other is time-consuming.
- MATLAB's Java-based implementation makes the plotting much simpler.
- Good pictures can take mediocre results/research and impress people.

Simple Plotting

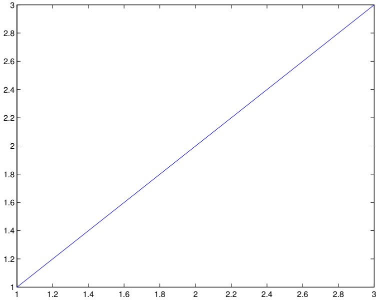
- You need two arrays of numbers to plot.
- Example #1

```
x=[1 2 3];   % pick values for y=x;
y=[1 2 3];
plot(x,y); %plot y versus x
```

Simple Plotting

- You need two arrays of numbers to plot.
- Example #1

```
x=[1 2 3];   % pick values for y=x;
y=[1 2 3];
plot(x,y); %plot y versus x
```



Defining Arrays....Differently

Until now we've explicitly defined an array element-wise. Now, we define the array using *vector notation*. Vector Notation

- Example #2

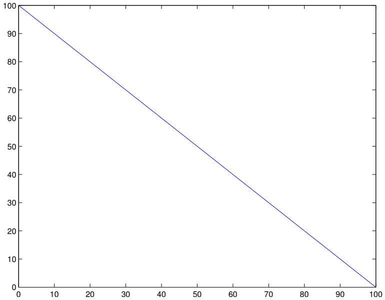
```
x=0:2:100; %x=[0,2,4,...,98,100];
y=100:-2:0; %y=[100,98,96,94,92,90,....6,4,2,0];
```

Defining Arrays....Differently

Until now we've explicitly defined an array element-wise. Now, we define the array using *vector notation*. Vector Notation

- Example #2

```
x=0:2:100; %x=[0,2,4,...,98,100];
y=100:-2:0; %y=[100,98,96,94,92,90,....6,4,2,0];
```



...Array....Differently, cont.

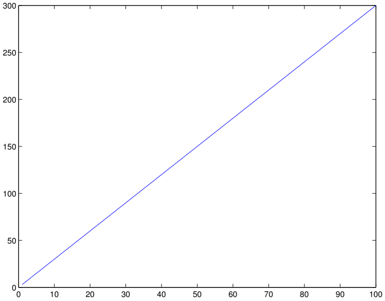
- Example #3

```
x=1:100; %x=[1,2,3,4,5,...,99,100];
y=3*x;   % y=[3,6,9,12,....297,300];
```

...Array....Differently, cont.

- Example #3

```
x=1:100; %x=[1,2,3,4,5,...,99,100];
y=3*x;  % y=[3,6,9,12,....297,300];
```



...Array....Differently, cont.

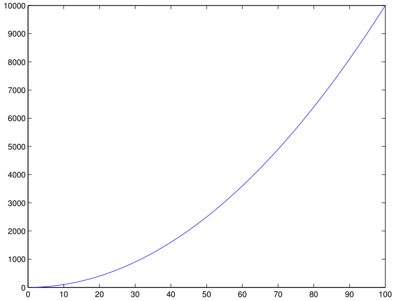
- Example #x4

```
x=1:100;  %x=[1,2,3,4,5,...,99,100];
y=x.^2;   %y=[1,4,9,16,25,...9801,10000];
```

...Array....Differently, cont.

- Example #x4

```
x=1:100; %x=[1,2,3,4,5,...,99,100];
y=x.^2;  %y=[1,4,9,16,25,...9801,10000];
```



MATLAB Functions

- We just learned to do basic plotting. What about plotting with more interesting data?
- There are certain functions which take arrays (vectors) and output vectors.
- Test the following:

```
x=-2*pi:pi/4:2*pi;
y=sin(x);
Figure(1) %trust me on this....
plot(x,y);
```

- How frequently is $\sin(x)$ being evaluated?

MATLAB Functions, cont.

- Now set the x-grid to $\pi/2$. Is this grid more or less *coarse*?
- Now redo the whole thing:

```
x=-2*pi:pi/2:2*pi;
y=sin(x);
Figure(2) %what is this command doing?
plot(x,y);
```

MATLAB Functions, cont.

- Now change the x-grid to $\pi/16$. Is this grid more or less *coarse*?
- Try it again...

```
x=-2*pi:pi/16:2*pi;
y=sin(x);
Figure(3)
plot(x,y);
```

- Compare the three....
- The less coarse (or more *refined*) grid yields the better approximate.
- Remember that these are all approximates.

Plotting

- Until now, the plots are just pictures.
- Graphics are only helpful if they add to a presentation. Otherwise, they detract.
- The information in a picture must be clearly understood, otherwise readers gloss over the content.

Text Labels

- We use the `xlabel` command to generate labels corresponding to what would be the 'x-axis'.
  `xlabel('My text here');`
- Similarly, the `ylabel` command generates labels that correspond to the 'y-axis'.
  `ylabel('The label for the y-axis goes here.');`
- We use the `title` command in a similar fashion to put titles at the top of the figure.

LaTeXand MATLAB

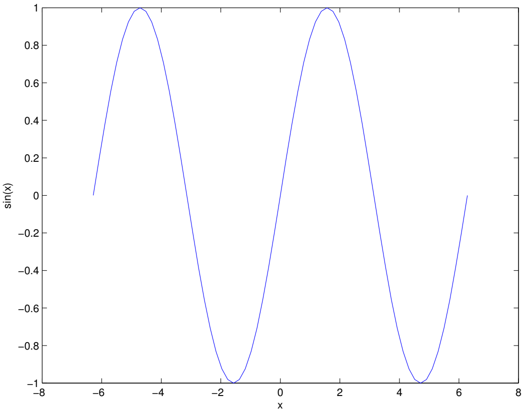
- You can use simple LaTeX commands in MATLAB in your labels and title.
- If you're saving images for use in a LaTeX document, don't use the `title`.

Plotting Example, Part A

- Try this:

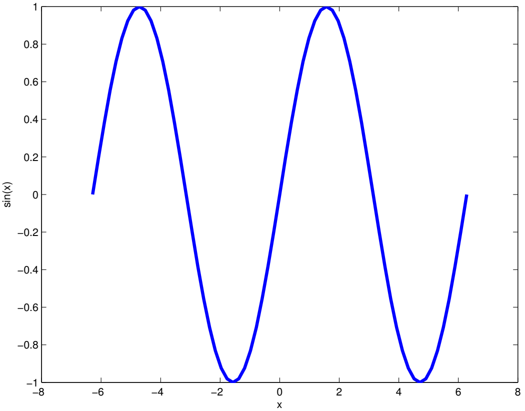
```
x=-2*pi:pi/16:2*pi
plot(x,sin(x));
xlabel('x');
ylabel('sin(x)');
```

- Still hard to see..



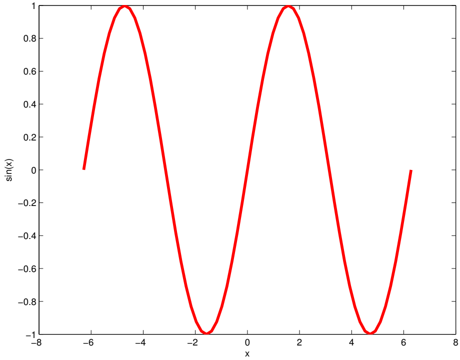
Plotting Example, Part B

- What if I want to increase the line thickness?
- `plot(x,y,‘LineWidth’,3);`



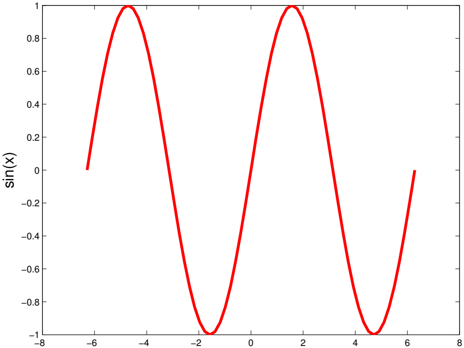
Plotting Example, Part C

- Do a 'help plot'.
- What can I do if I want to change the color of the line to red?



Plotting Example, Font Sizes

- Notice that font sizes are still way too small. I wouldn't want those in a LaTeX file.
- How do we fix this?
    - `xlabel('x','FontSize',16);`
    - Same for the `ylabel` and `title` commands.



Plotting Example, Axis Sizes

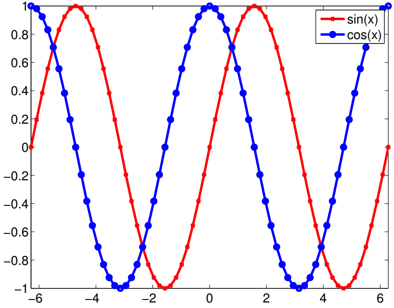
- To fix the font on the axes themselves, we issue the following commands

  `set(gca,'FontSize',16);`
- The gca stands for "Get Current Axes".
- Try it.

Multiple Plots on the Same Figure

- Can we plot multiple images on the same figure? You bet!
- The plot command looks at vector combos.

```
x=-2*pi:pi/16:2*pi;
plot(x,sin(x),'r-x',...
   x,cos(x),'b-o',...
   'LineWidth',3)
set(gca,'FontSize',16)
legend('sin(x)','cos(x)');
axis tight;
```

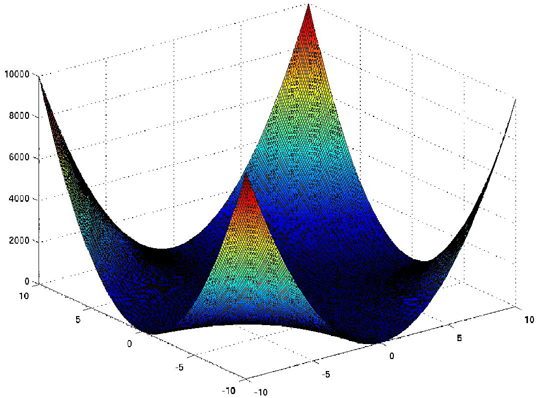


Tips and Tricks

- Use `axis tight` to zoom in the image as much as possible.
- Plot black lines for axes (like the Cartesian Coordinate Plane).
- Use a combination of plot techniques. Use 'r-x' and 'b-o' or something similar to compare actual data points.
- You can use the built-in tools to change all of the properties.
- You can save the image as a .eps file.

3-D Plotting

- MATLAB produces some superb three-dimensional graphics!
- `x=-10:.1:10;` and `y=-10:.1:10;`
- $f(x, y) = x^2y^2$



Ways to Plot in Three Dimensions

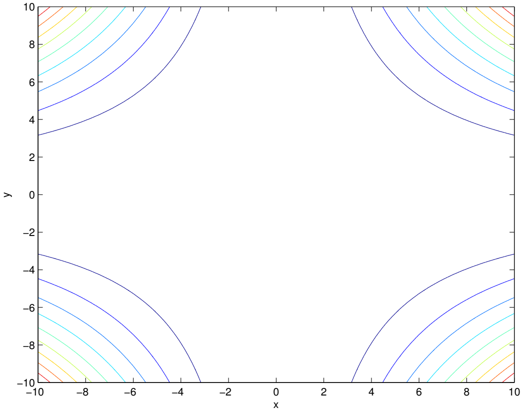
- Contour Plots
- Surface Plots
- Mesh Plots

Contour Plots

- Let's set up our functions

```
x=-10:.1:10;y=x;
Z=(x.^2)'*(y.^2);
contour(x,y,Z);
xlabel('x');
ylabel('y');
```

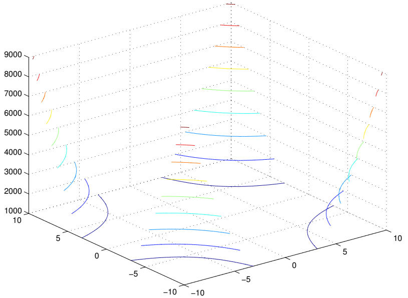
- Two-dimensional plot
- The third dimension is expressed via concentric color-coded curves.

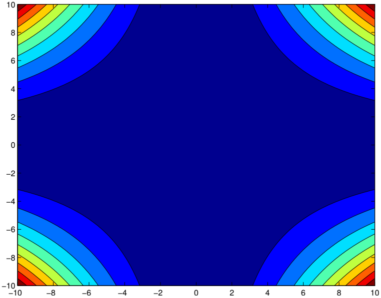


You can click on "Edit" and then "Figure Properties" to change

Contour Plot Variants

- Run the program as `contour3(x,y,Z)` (shown left).
- Run the program as `contourf(x,y,Z)` (shown right).





Bold colors are eye-catching and help readers to see information more clearly.

Surface Plots

- Same information

```
x=-10:10;y=x;
Z=(x.^2)'*(y.^2);
surf(x,y,Z);
xlabel('x');
ylabel('y');
```

- Creates a surface.
- surf
- surfc
- surfl

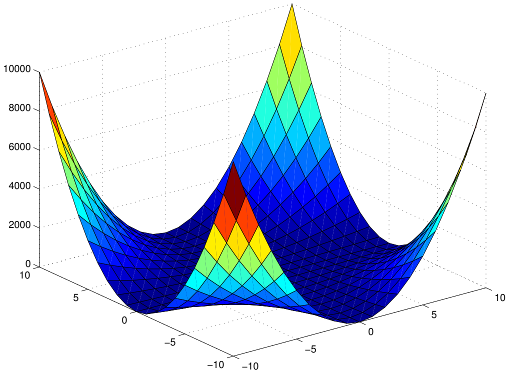


Figure: Three-dimensional surface plot.

Surface Plots

- Same information

```
x=-10:.1:10;y=x;
Z=(x.^2)'*(y.^2);
surfc(x,y,Z);
xlabel('x');
ylabel('y');
```

- Creates a surface.
- surf
- surfc
- surfl

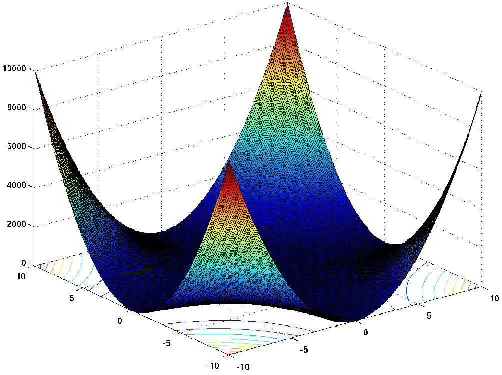


Figure: Surface plot with contours (surfc).

Surface Plots

- Same information

```
x=-10:10;y=x;
Z=(x.^2)'*(y.^2);
surfl(x,y,Z);
xlabel('x');
ylabel('y');
```

- Creates a surface.
- `surf`
- `surfc`
- `surfl`

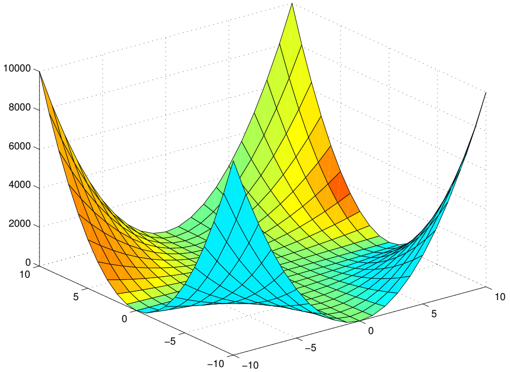


Figure: Surface plot with lighting (`surfl`).

Mesh Plots

- Same information

```
x=-10:10;y=x;
Z=(x.^2)'*(y.^2);
mesh(x,y,Z);
xlabel('x');
ylabel('y');
```

- Creates a surface.
- mesh
- meshc
- meshz
- waterfall

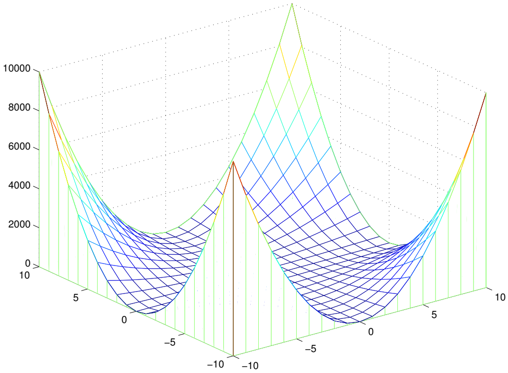


Figure: Three-dimensional mesh plot.

Mesh Plots

- Same information

```
x=-10:.1:10;y=x;
Z=(x.^2)'*(y.^2);
meshc(x,y,Z);
xlabel('x');
ylabel('y');
```

- Creates a surface.
- `mesh`
- `meshc`
- `meshz`
- `waterfall`

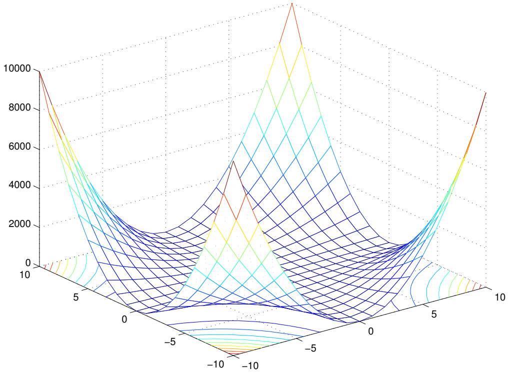


Figure: (meshc).

Mesh Plots

- Same information

```
x=-10:.1:10;y=x;
Z=(x.^2)'*(y.^2);
meshz(x,y,Z);
xlabel('x');
ylabel('y');
```

- Creates a surface.
- `mesh`
- `meshc`
- `meshz`
- `waterfall`

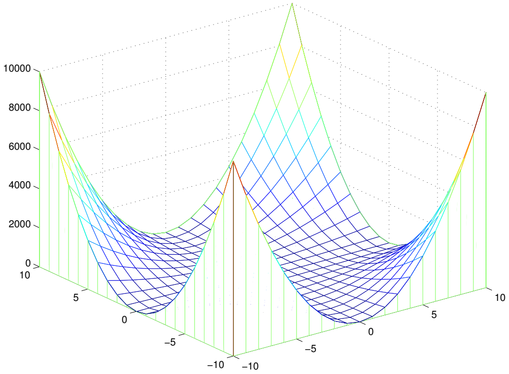


Figure: (meshz).

Mesh Plots

- Same information

```
x=-10:10;y=x;
Z=(x.^2)'*(y.^2);
waterfall(x,y,Z);
xlabel('x');
ylabel('y');
```

- Creates a meshgrid.
- `mesh`
- `meshc`
- `meshz`
- `waterfall`

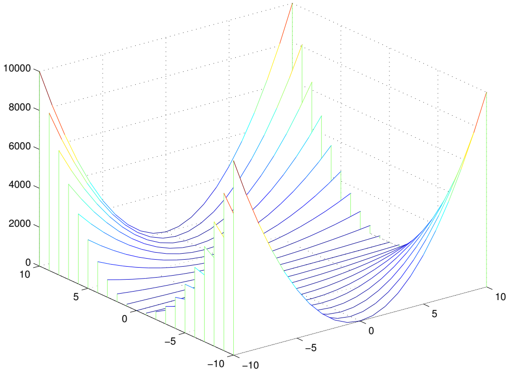
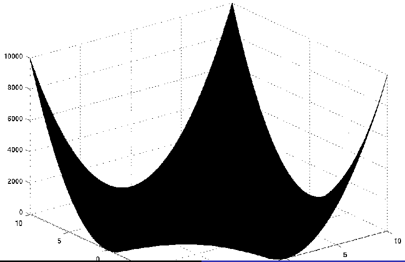


Figure: Waterfall plot (`waterfall`).

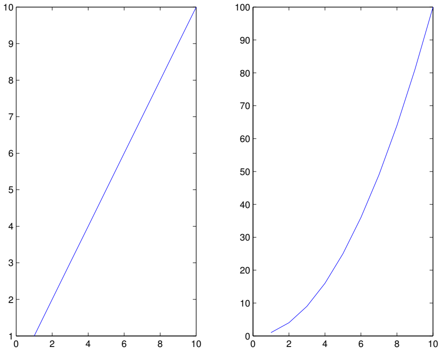
Ideas of Note

- Don't typically save as .fig image. Only MATLAB recognizes this format.
- For 3-D plots, you can rotate the image to get *just* the picture you want.
- Comment the plot command. What are you plotting? Why are you plotting this?
- You can make your plots too accurate!



Subplots

```
x=1:10;
subplot(1,2,1)
y=x;
plot(x,y);
subplot(1,2,2)
y=x.^2;
plot(x,y);
```



This is great for presentations. However, in LaTeX documents couple a figure with an internal tabular environment where the tabular entries are graphics.

Exercises

The results and discussion of the following should be included, presented, and discussed in a LaTeX document.

1. For `x=-10:.01:10;`, plot the tangent of x versus x. What does the MATLAB image depict? What behaviors are you seeing? Clearly label all axes.
2. Plot the graph of

$$f(x, y) = xy^2 e^{-\left(\frac{x^2+y^2}{4}\right)}.$$

Use a three-dimensional graph of appropriate type and color-scheme to relate the contours of the function. Give attention to the grid refinement. Use any rotations or shifts to generate an image which is most easily related.

Exercises, cont.

3. Plot, in the same figure, the second, fourth, and sixth Taylor expansion approximates to $f(x) = \sin(x)$. Use different line styles (solid, dashed, .etc.). Clearly label axes and plotted functions. Which is the best approximate? Thoroughly discuss these results.

4. A simple harmonic oscillator has a closed form solution of position $x(t)$ as a function of time t given by

$$x(t) = A\cos(\omega t + \phi).$$

Choose an amplitude A, frequency ω and phase ϕ and plot the behavior of a simple (undamped) harmonic oscillator. Are there examples of undamped harmonic oscillators that you can discover? Does it make sense to do discuss negative t?