

FF505/FY505
Computational Science

MATLAB Graphics

Marco Chiarandini (marco@imada.sdu.dk)

Department of Mathematics and Computer Science (IMADA)
University of Southern Denmark

1. Graphics
 - 2D Plots
 - 3D Plots

1. Graphics

- 2D Plots

- 3D Plots

Plot measured data (points) or functions (lines)

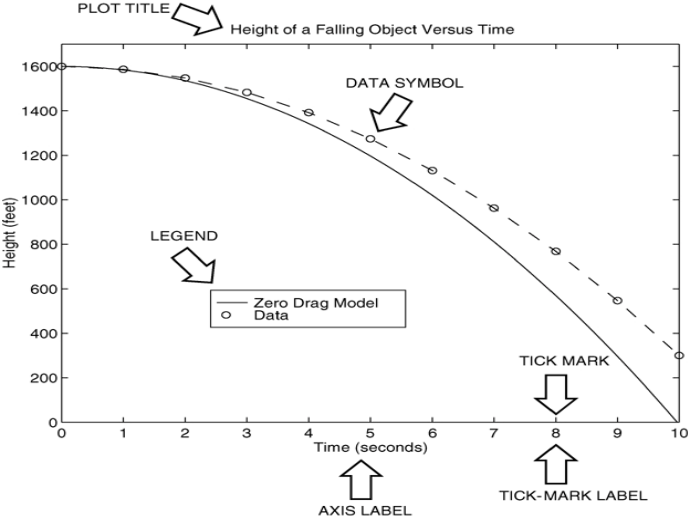
Two-dimensional plots or *xy plots*

```
help graph2d
```

Three-dimensional plots or *xyz plots* or
surface plots

```
help graph3d
```

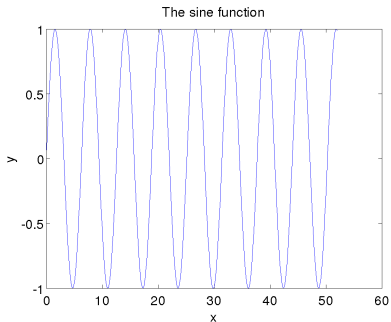
Nomenclature xy plot



1. Graphics
 - 2D Plots
 - 3D Plots

An Example: $y = \sin(x)$

```
x = 0:0.1:52;  
y = sin(x)  
plot(x,y)  
xlabel('x')  
ylabel('y')  
title('The sine function')  
%title('F(\theta)=sin(\theta)')
```



The autoscaling feature in MATLAB selects tick-mark spacing.

To start plot edit mode, click this button.

Use these toolbar buttons to add a legend, text, and arrows.

Use the Edit, Insert, and Tools menus to add objects or edit existing objects in a graph.

Double-click on an object to select it.

Position labels, legends, and other objects by clicking and dragging.

Access object-specific plot edit functions through context-sensitive pop-up menus.

Use these toolbar buttons to add a legend, text, and arrows.

Figure 1

File Edit View Insert Tools Desktop Window Help

Lotka-Volterra Predator-Prey Population Model

Population Size

Time t (Years)

Prey

Predator

Many predators; prey population will decline

Few predators; prey population will increase

Cut

Copy

Delete

Color...

Line Width

Line Style

Marker

Marker Size

Properties...

Show M-code

But better to do this with lines of code, just in case you have to redo the plot.

Saving Figures

The plot appears in the **Figure window**. You can include it in your documents:

1. type

```
print -dpng foo
```

at the command line. This command sends the current plot directly to `foo.png`

```
~> help print
```

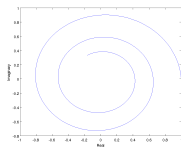
2. from the File menu, select **Save As**, write the name and select file format from **Files of Types** (eg, png, jpg, etc)
.fig format is MATLAB format, which allows to edit
3. from the File menu, select **Export Setup** to control size and other parameters
4. on Windows, copy on clipboard and paste. From Edit menu, Copy Figure and Copy Options

The grid and axis Commands

- grid command to display gridlines at the tick marks corresponding to the tick labels.
grid on to add gridlines;
grid off to stop plotting gridlines;
grid to toggle
- axis command to override the MATLAB selections for the axis limits.
axis([xmin xmax ymin ymax]) sets the scaling for the x- and y-axes to the minimum and maximum values indicated. Note: no separating commas
axis square, axis equal, axis auto

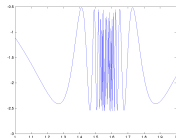
plot complex numbers

```
y=0.1+0.9i, plot(y)
z=0.1+0.9i, n=0:0.01:10,
plot(z.^n, xlabel('Real'), ylabel('Imaginary'))
```



function plot command

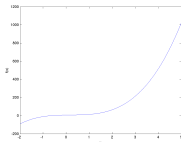
```
f=@(x) (cos(tan(x))-tan(sin(x)));
fplot(f,[1 2])
[x,y]=fplot(function,limits)
```



plotting polynomials

Eg, $f(x) = 9x^3 - 5x^2 + 3x + 7$ for
 $-2 \leq x \leq 5$:

```
a = [9,-5,3,7];
x = -2:0.01:5;
plot(x,polyval(a,x)),xlabel('x'),ylabel('f(x)')
```

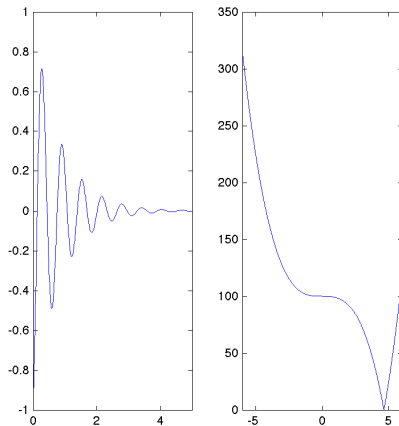


Subplots

subplot command to obtain several smaller **subplots** in the same figure.

subplot(*m,n,p*) divides the Figure window into an array of rectangular **panes** with *m* rows and *n* columns and sets the pointer after the *p*th pane.

```
x = 0:0.01:5;  
y = exp(-1.2*x).*sin(10*x+5);  
subplot(1,2,1)  
plot(x,y),axis([0 5 -1 1])  
x = -6:0.01:6;  
y = abs(x.^3-100);  
subplot(1,2,2)  
plot(x,y),axis([-6 6 0 350])
```



Data Markers and Line Types

Three components can be specified in the string specifiers along with the plotting command. They are:

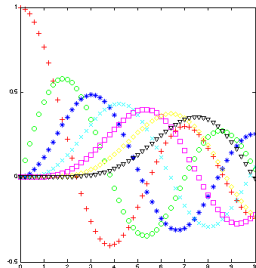
- Line style
- Marker symbol
- Color

```
plot(x,y,u,v,'--') % where the symbols '--' represent a dashed line
plot(x,y,'*','x,y',':') % plot y versus x with asterisks connected with a dotted line
plot(x,y,'g*','x,y','r--') % green asterisks connected with a red dashed line
```

% Generate some data using the besselj

```
x = 0:0.2:10;
y0 = besselj(0,x);
y1 = besselj(1,x);
y2 = besselj(2,x);
y3 = besselj(3,x);
y4 = besselj(4,x);
y5 = besselj(5,x);
y6 = besselj(6,x);

plot(x, y0, 'r+', x, y1, 'go', x, y2, 'b*',
      x, y3, 'cx', ...
      x, y4, 'ms', x, y5, 'yd', x, y6, 'kv');
```



doc LineSpec

Specifier	LineStyle
'-'	Solid line (default)
'--'	Dashed line
'.'	Dotted line
'-.'	Dash-dot line

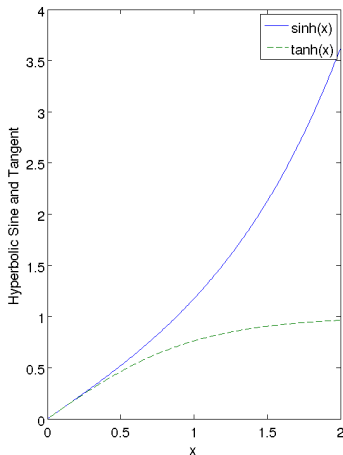
Specifier	Marker Type
'+'	Plus sign
'o'	Circle
'*'	Asterisk
'.'	Point
'x'	Cross
'square' or 's'	Square
'diamond' or 'd'	Diamond
'^'	Upward-pointing triangle
'v'	Downward-pointing triangle
'>'	Right-pointing triangle
'<'	Left-pointing triangle
'pentagram' or 'p'	Five-pointed star (pentagram)
'hexagram' or 'h' ''	Six-pointed star (hexagram)

Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

Labeling Curves and Data

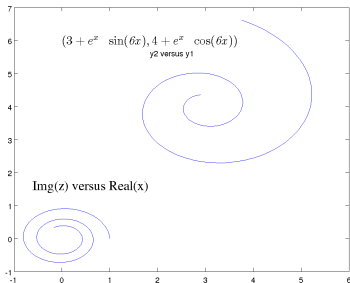
The legend command automatically obtains the line type used for each data set

```
x = 0:0.01:2;  
y = sinh(x);  
z = tanh(x);  
plot(x,y,x,z,'--'),xlabel('x')  
ylabel('Hyperbolic Sine and Tangent')  
legend('sinh(x)', 'tanh(x)')
```



The hold Command and Text Annotations ^{Graphics}

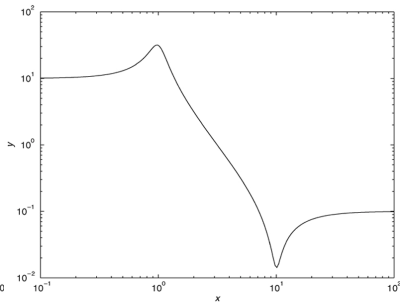
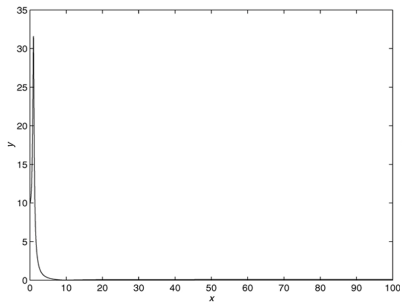
```
x=-1:0.01:1
y1=3+exp(-x).*sin(6*x);
y2=4+exp(-x).*cos(6*x);
plot((0.1+0.9i).^(0:0.01:10)), hold, plot(y1,y2)
gtext('y2 versus y1') % places in a point specified by the mouse
gtext('Img(z) versus Real(x)', 'FontName', 'Times', 'FontSize', 18)
```



```
text('Interpreter', 'latex', ...
    'String', ...
    '$(3+e^{-x}\sin(\{it 6x\}),4+e^{-x}\cos(\{it 6x\}))$', ...
    'Position', [0,6], ...
    'FontSize', 16)
```

Search **Text Properties** in Help
Search **Mathematical symbols, Greek Letter and TeX Characters**

Axes Transformations



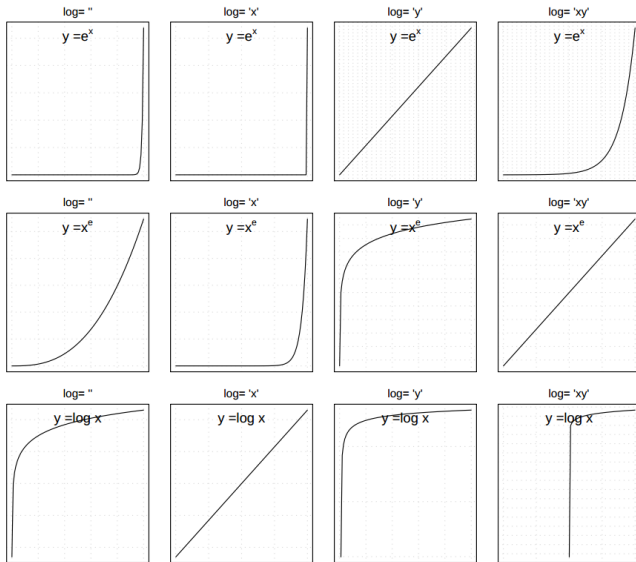
Instead of `plot`, plot with

```
loglog(x,y) % both scales logarithmic.  
semilogx(x,y) % x scale logarithmic and the y scale rectilinear.  
semilogy(x,y) % y scale logarithmic and the x scale rectilinear.
```

Remember:

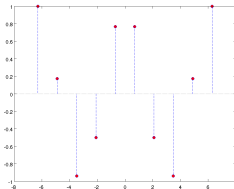
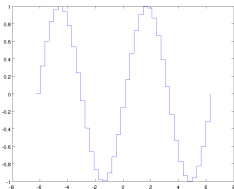
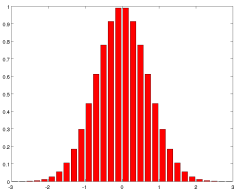
1. You cannot plot negative numbers on a log scale: the logarithm of a negative number is not defined as a real number.
2. You cannot plot the number 0 on a log scale: $\log_{10} 0 = -\infty$.
3. The tick-mark labels on a log scale are the actual values being plotted; they are not the logarithms of the numbers. Eg, the range of x values in the plot before is from $10^{-1} = 0.1$ to $10^2 = 100$.
4. Gridlines and tick marks within a decade are unevenly spaced. If 8 gridlines or tick marks occur within the decade, they correspond to values equal to 2, 3, 4, ..., 8, 9 times the value represented by the first gridline or tick mark of the decade.
5. Equal distances on a log scale correspond to multiplication by the same constant (as opposed to addition of the same constant on a rectilinear scale).

The effect of log-transformation



Specialized plot commands

Command	Description
<code>bar(x,y)</code>	Creates a bar chart of y versus x
<code>stairs(x,y)</code>	Produces a stairs plot of y versus x .
<code>stem(x,y)</code>	Produces a stem plot of y versus x .



Command

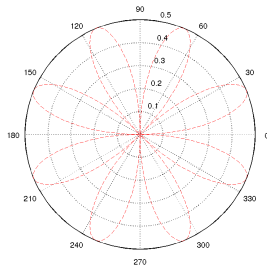
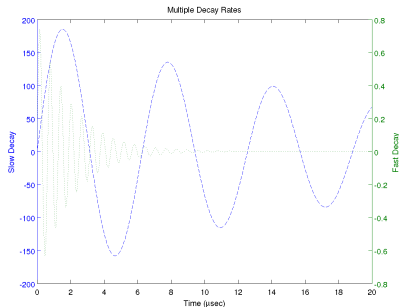
```
plotyy(x1,y1,x2,y2)
```

```
polar(theta,r,'type')
```

Description

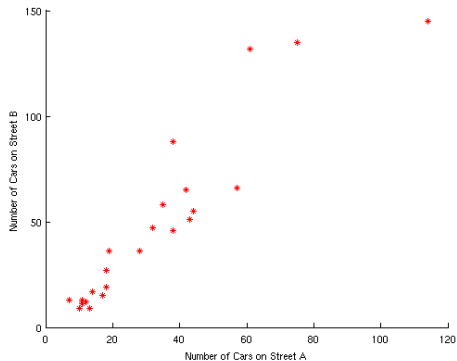
Produces a plot with two y-axes, y1 on the left and y2 on the right

Produces a polar plot from the polar coordinates theta and r, using the line type, data marker, and colors specified in the string type.



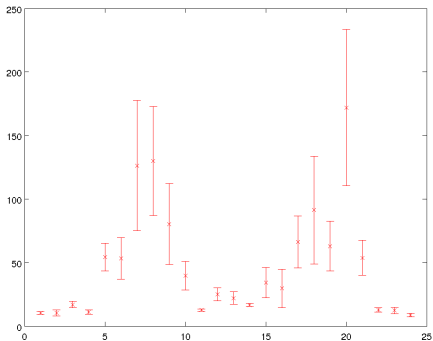
Scatter Plots

```
load count.dat
scatter(count(:,1),count(:,2),
        'r*')
xlabel('Number of Cars on
Street A');
ylabel('Number of Cars on
Street B');
```



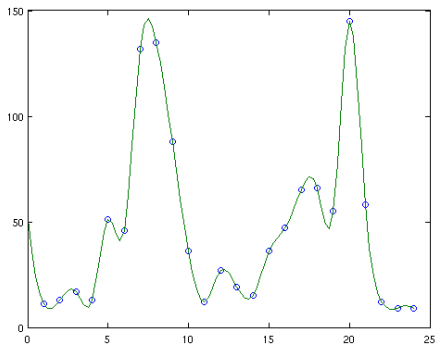
Error Bar Plots

```
load count.dat;  
y = mean(count,2);  
e = std(count,1,2);  
figure  
errorbar(y,e,'xr')
```



Add interpolation

```
x=1:24  
y=count(:,2)  
xx=0:.25:24  
yy=spline(x,y,xx)  
plot(x,y,'o',xx,yy)
```



1. Graphics

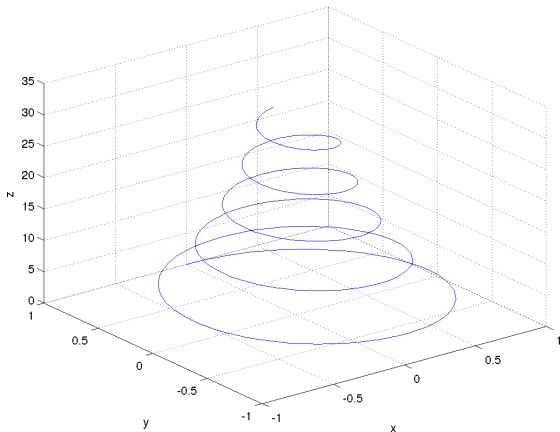
2D Plots

3D Plots

Three-Dimensional Line Plots

Plot in 3D the curve: $x = e^{-0.05t} \sin(t)$, $y = e^{-0.05t} \cos(t)$, $z = t$

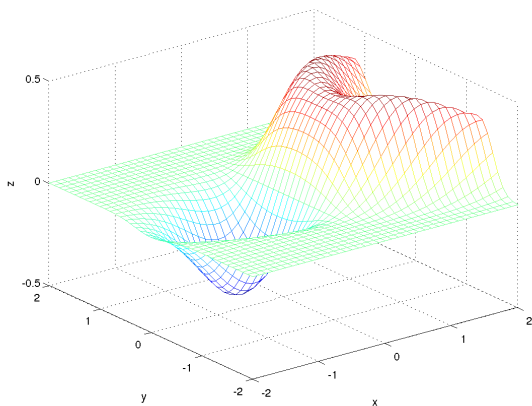
```
t = 0:pi/50:10*pi;  
plot3(exp(-0.05*t).*sin(t), exp(-0.05*t).*cos(t), t)  
xlabel('x'), ylabel('y'), zlabel('z'), grid
```



Surface Plots

Surface plot of the function $z = xe^{-[(x-y^2)^2+y^2]}$, for $-2 \leq x \leq 2$ and $-2 \leq y \leq 2$ with a spacing of 0.1

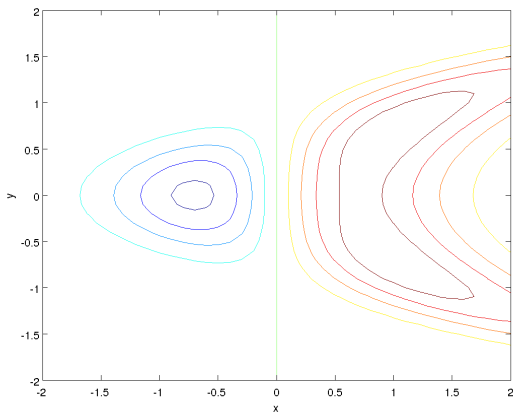
```
[X,Y] = meshgrid(-2:0.1:2);  
Z = X.*exp(-((X-Y.^2).^2+Y.^2));  
mesh(X,Y,Z), xlabel('x'), ylabel('y'), zlabel('z') % or also surf
```



Contour Plots

Contour plot of the function $z = xe^{-[(x-y^2)^2+y^2]}$, for $-2 \leq x \leq 2$ and $-2 \leq y \leq 2$ with a spacing of 0.1

```
[X,Y] = meshgrid(-2:0.1:2);  
Z = X.*exp(-((X-Y.^2).^2+Y.^2));  
contour(X,Y,Z), xlabel('x'), ylabel('y')
```



Three-Dimensional Plotting Functions

Function

`contour(x,y,z)`

`mesh(x,y,z)`

`meshc(x,y,z)`

`meshz(x,y,z)`

`surf(x,y,z)`

`surf(x,y,z)`

`[X,Y] = meshgrid(x,y)`

`[X,Y] = meshgrid(x)`

`waterfall(x,y,z)`

Description

Creates a contour plot.

Creates a 3D mesh surface plot.

Same as mesh but draws contours under the surface.

Same as mesh but draws vertical reference lines under the surface.

Creates a shaded 3D mesh surface plot.

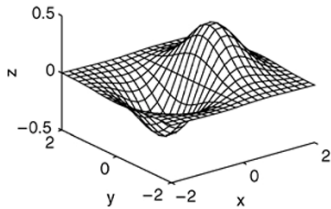
Same as surf but draws contours under the surface.

Creates the matrices X and Y from the vectors x and y to define a rectangular grid.

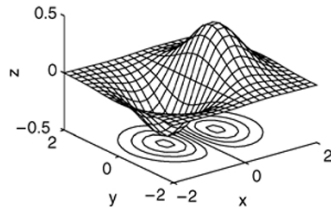
Same as `[X,Y]= meshgrid(x,x)`.

Same as mesh but draws mesh lines in one direction only.

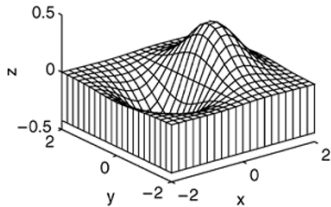
a) mesh, b) meshc, c) meshz, d) waterfall



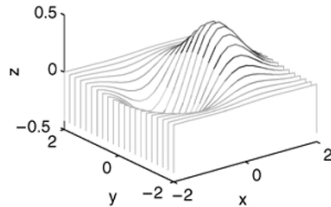
(a)



(b)



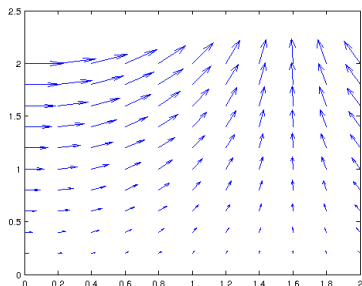
(c)



(d)

Vector fields

Use `quiver` to display an arrow at each data point in x and y such that the arrow direction and length represent the corresponding values of the vectors u and v .



```
[x,y] = meshgrid(0:0.2:2,0:0.2:2);  
u = cos(x).*y;  
v = sin(x).*y;  
  
figure  
quiver(x,y,u,v)
```

- Should the experimental setup from the exploratory phase be redesigned to increase conciseness or accuracy?
- What parameters should be varied? What variables should be measured?
- Should a 3D-plot be replaced by collections of 2D-curves?
- Can we reduce the number of curves to be displayed?
- How many figures are needed?
- Should the x-axis be transformed to magnify interesting subranges?
- Should the x-axis have a logarithmic scale? If so, do the x-values used for measuring have the same basis as the tick marks?

- Make sure that each axis is labeled with the name of the quantity being plotted and its units.
- Make tick marks regularly paced and easy to interpret and interpolate, eg, 0.2, 0.4, rather than 0.23, 0.46
- Use the same scale limits and tick spacing on each plot if you need to compare information on more than one plot.
- Is the range of x-values adequate?
- Do we have measurements for the right x-values, i.e., nowhere too dense or too sparse?
- Should the y-axis be transformed to make the interesting part of the data more visible?
- Should the y-axis have a logarithmic scale?
- Is it misleading to start the y-range at the smallest measured value? (if not too much space wasted start from 0)
- Clip the range of y-values to exclude useless parts of curves?

- Can we use banking to 45° ?
- Are all curves sufficiently well separated?
- Can noise be reduced using more accurate measurements?
- Are error bars needed? If so, what should they indicate? Remember that measurement errors are usually not random variables.
- Connect points belonging to the same curve.
- Only use splines for connecting points if interpolation is sensible.
- Do not connect points belonging to unrelated owners.
- Use different point and line styles for different curves.
- Use the same styles for corresponding curves in different graphs.
- Place labels defining point and line styles in the right order and without concealing the curves.

- Captions should make figures self contained.
- Give enough information to make experiments reproducible.
- Golden ratio rule: make the graph wider than higher [Tuft 1983].
- Rule of 7: show at most 7 curves (omit those clearly irrelevant).
- Avoid: explaining axes, connecting unrelated points by lines, cryptic abbreviations, microscopic lettering, pie charts