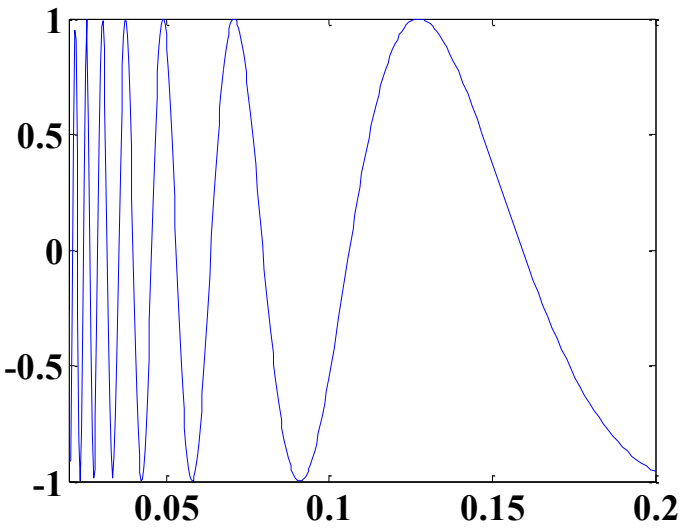
8. Specialized 2-D plotting functions

MATLAB includes a variety of specialized plotting in addition to the ones described above. The following below briefly describes some of the other plotting functions available in MATLAB.

fplot: evaluates a function and plots the results

Example:

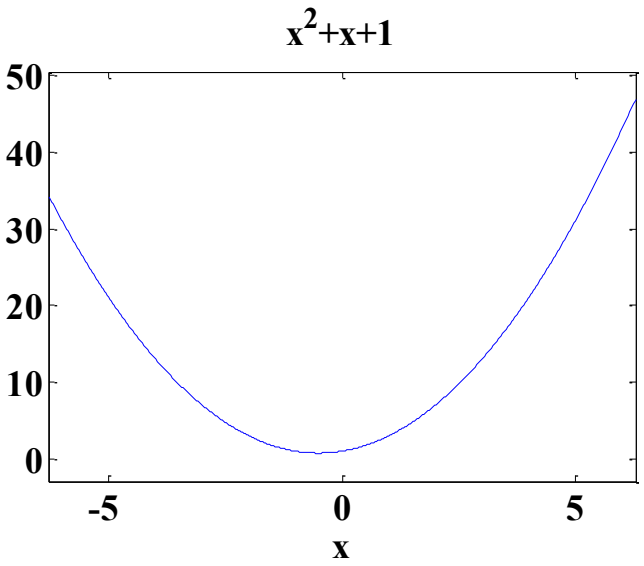
```
fplot('sin(1/x)', [0.02 0.2]);
```



ezplot: simplest way to graph a function (easy plot).

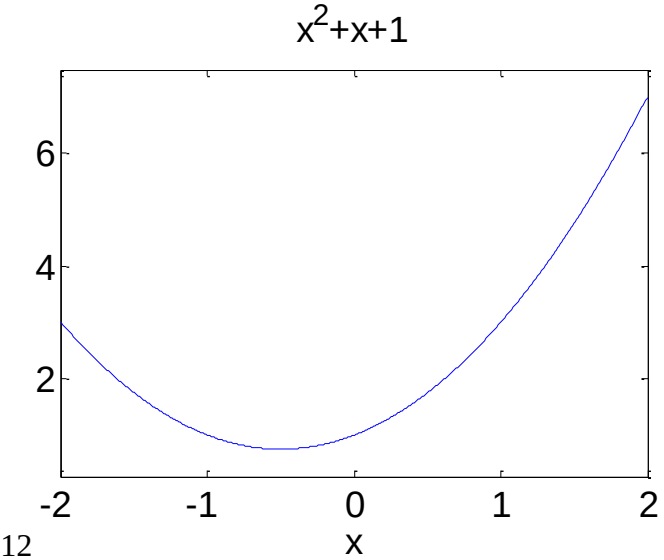
Example:, to graph the function $y=x^2+x+1$, you write:

```
ezplot('x^2+x+1')
```



If you want to sketch the same function in between –2 and 2 you simply write:

```
ezplot('x^2+x+1', [-2, 2])
```

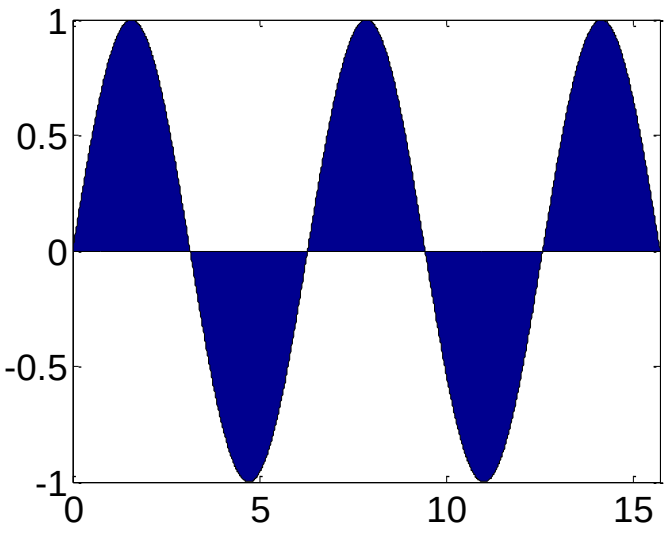


area Filled area plot.

area(X,Y) produces a stacked area plot suitable for showing the contributions of various components to a whole. For vector X and Y, area(X,Y) is the same as plot(X,Y) except that the area between 0 and Y is filled.

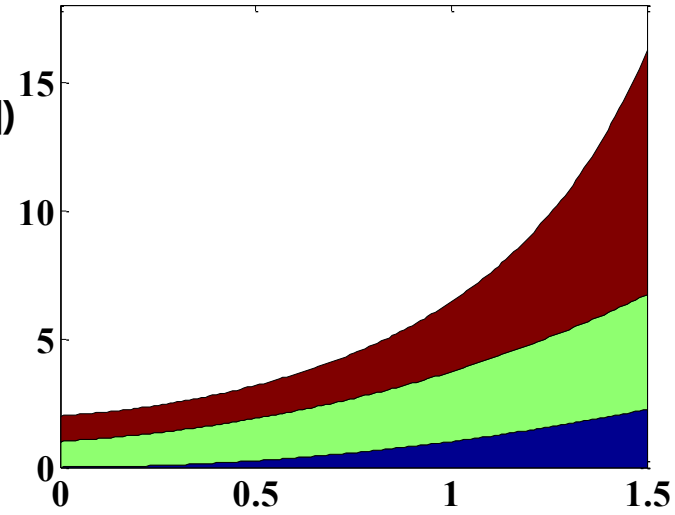
Example:

```
t = 0:.01:5*pi;
 area(t,sin(t))
```



Example:

```
x = 0:0.01:1.5;
area(x,[(x.^2)',(exp(x))',(exp(x.^2))'])
```

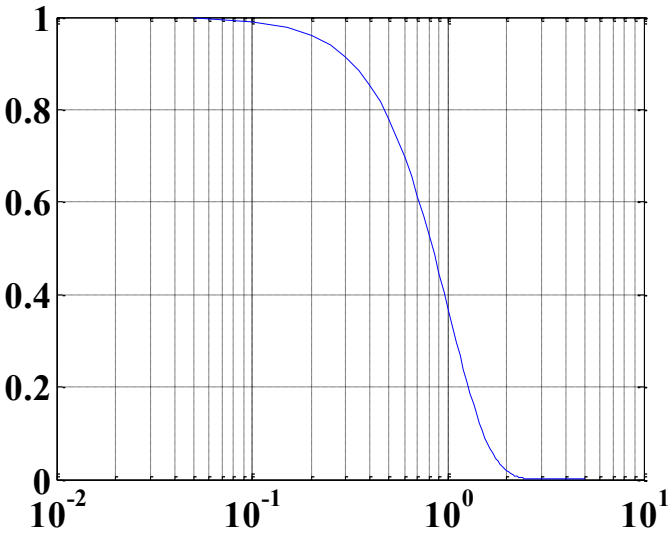


semilogx: log-scaled x axis

semilogx is the same as **plot**, except a logarithmic (base 10) scale is used for the X-axis.

Example:

```
x=0:0.05:5;
y=exp(-x.^2);
semilogx(x,y);
grid
```

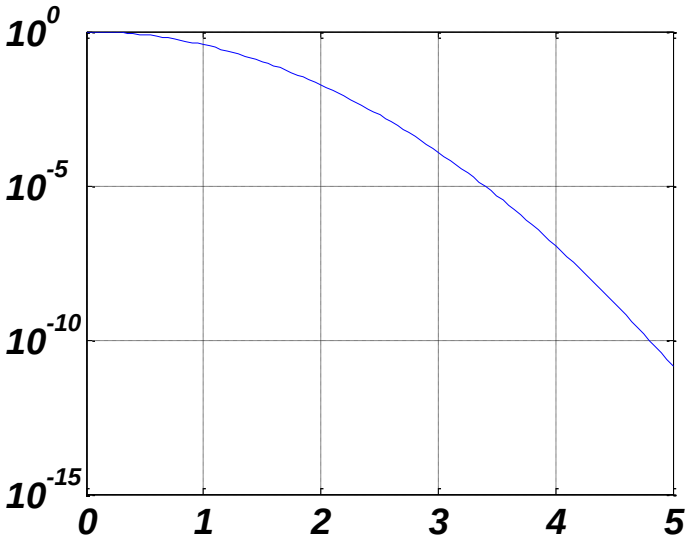


semilogy: log-scaled y axis

semilogy is the same as **plot**, except a logarithmic (base 10) scale is used for the Y-axis.

Example:

```
x=0:0.05:5;
y=exp(-x.^2);
semilogy(x,y);
grid
```

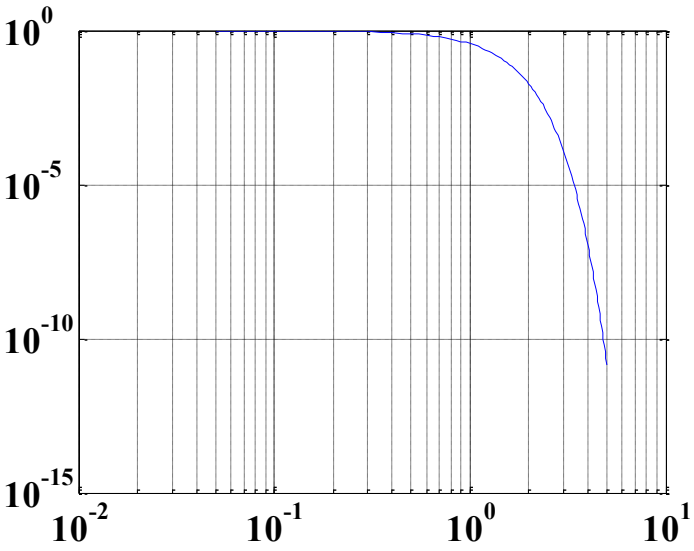


Loglog: Log-log scale plot.

Loglog is the same as **plot**, except logarithmic scales are used for both the X- and Y- axes.

Example:

```
x=0:0.05:5;
y=exp(-x.^2);
loglog(x,y);
grid
```

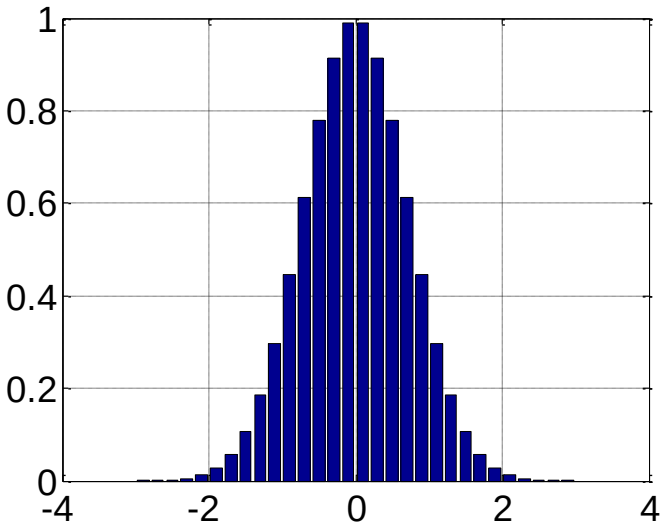


bar: creates a bar graph.

bar(X,Y) draws the columns of the M-by-N matrix Y as M groups of N vertical bars. The vector X must be monotonically increasing or decreasing

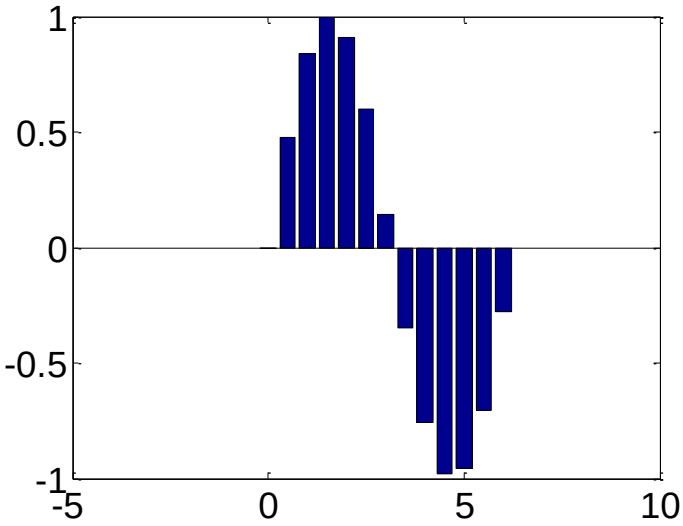
Example:

```
x = -2.9:0.2:2.9;
bar(x,exp(-x.*x));
grid
```



Example:

```
x = 0:0.5:2*pi;
bar(x, sin(x));
```

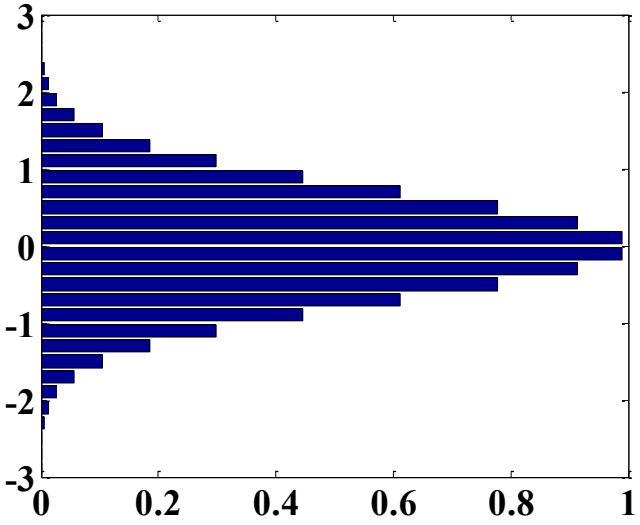


barh: horizontal bar graph

barh(X,Y) draws the columns of the M-by-N matrix Y as M groups of N horizontal bars.

Example:

```
x = -2.9:.2:2.9;
y = exp(-x.*x);
barh(x,y);
```

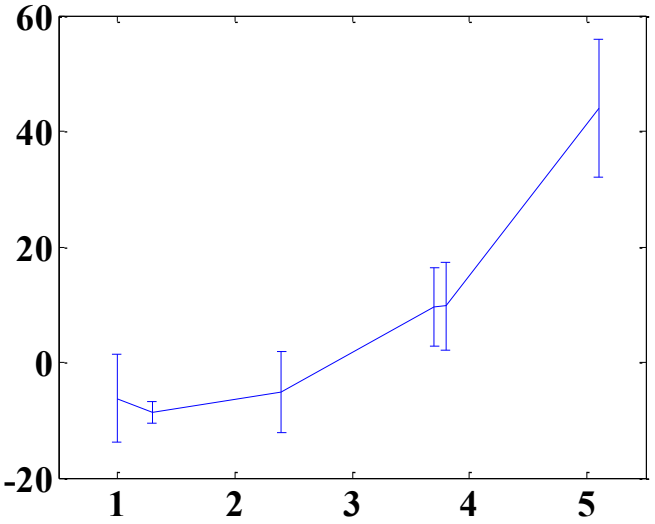


errorbar: creates a plot with error bars

errorbar (X,Y,L) plots the graph of vector X vs. vector Y with error bars specified by the vectors L and U.

Example:

```
x = [ 1.0 1.3 2.4 3.7 3.8 5.1 ];
y = [ -6.3 -8.7 -5.2 9.5 9.8 43.9 ];
coeff = polyfit(x,y,1)
yp=polyval(coeff,x)
e=abs(yp-y)
errorbar(x,y,e); grid
e =
```



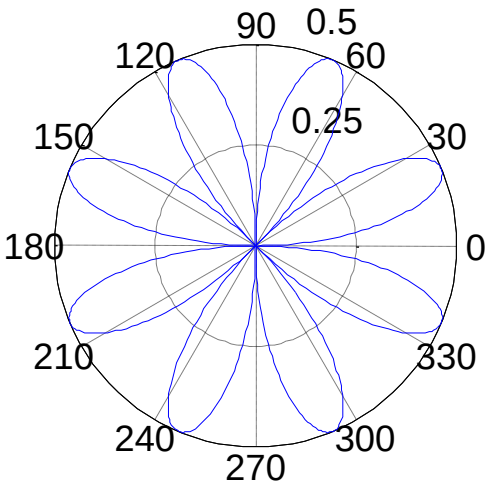
7.6079 1.8509 6.9582 6.8052 7.6242 11.9288

polar: creates a plot in polar coordinates of angles versus radin

polar(theta,rho) makes a plot using polar coordinates of the angle **theta**, in radians, versus the radius **rho**.

Example:

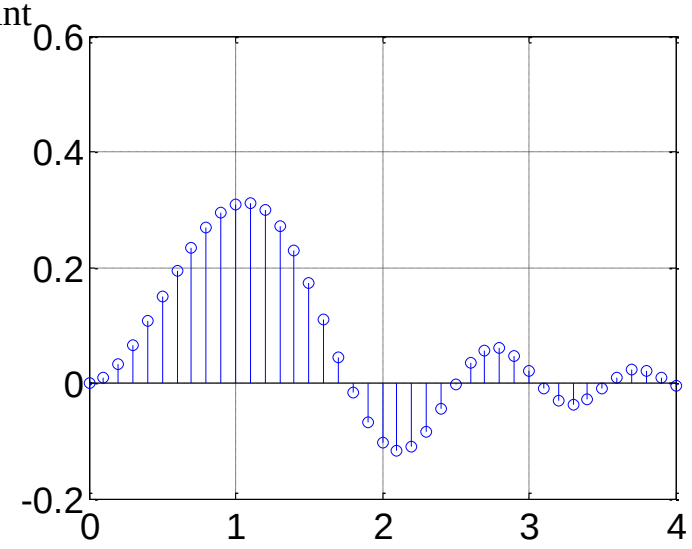
```
t=0:.01:2*pi;
polar(t,sin(2*t).*cos(2*t));
```



stem: generates stems at each data point

Example:

```
x = 0:0.1:4;
y = sin(x.^2).*exp(-x);
stem(x,y);
grid
```

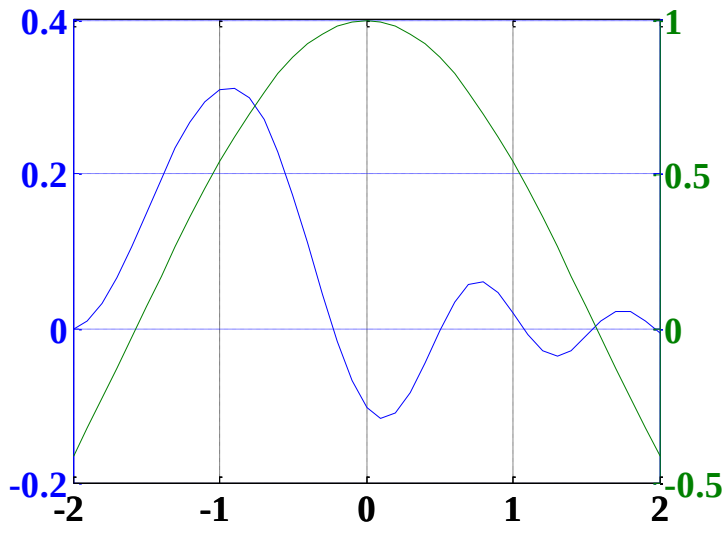


plotyy: graphs with y tick labels on the left and right

plotyy(X1,Y1,X2,Y2) plots Y1 versus X1 with y-axis labeling on the left and plots Y2 versus X2 with y-axis labeling on the right.

Example:

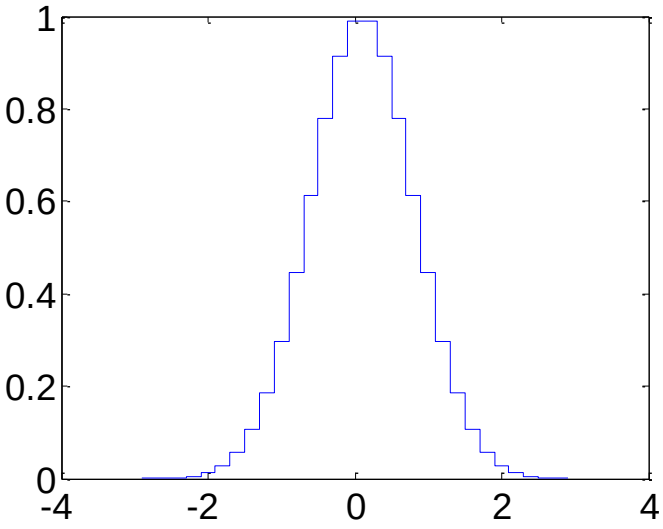
```
x=-2:0.1:2;
y1=sin(x);
y2=cos(x);
plotyy(x,y,x,y2);
grid
```



stairs: creates a graph similar to a bar graph, but without internal lines

Example:

```
x = -2.9:.2:2.9;
y = exp(-x.*x);
stairs(x,y);
title('Stair Chart');
```

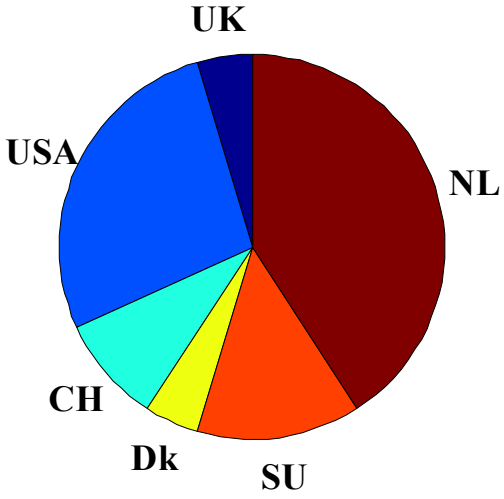


pie: Pie chart.

pie(X) draws a pie plot of the data in the vector **X**. The values in X are normalized via X/sum(X) to determine the area of each slice of pie.

Example:

```
x = [1 6  2 1 3 9];
label = {'UK','USA','CH','Dk','SU','NL'};
pie(x, label)
```



hist: creates a histogram

Example:

```
x=rand(1,100);
hist(x);
grid
```