Scan or click here for more resources

MATLAB Plotting

Plotting is a graphical representation of a data set that shows a relationship between two or more variables. MATLAB plots play an essential role in the field of mathematics, science, engineering, technology, and finance for statistics and data analysis.

There are several functions available in MATLAB to create **2-dimensional** and **3-dimensional** plots.

Adding Title, Labels, Grid Lines and Scaling on the Graph

MATLAB allows you to add title, labels along the x-axis and y-axis, grid lines and also to adjust the axes to spruce up the graph.

- The **xlabel** and **ylabel** commands generate labels along x-axis and y-axis.
- The **title** command allows you to put a title on the graph.
- The **grid on** command allows you to put the grid lines on the graph.
- The **axis equal** command allows generating the plot with the same scale factors and the spaces on both axes.
- The **axis square** command generates a square plot.
- linspace – linspace generates a row vector of 100 linearly equally spaced points between X1 and X2.
- legend - legend(string1, string2, string3, ...) puts a legend on the current plot using the specified strings as labels.

Line, Color, and Mark Style

The style option in the plot command are a character string that consists of one, two, or three characters that specify the color and line style. There are several color, line, and marker-style options:

Color Style-option	Line Style option	Marker Style-option
y yellow	- Solid	+ plus sign
m magenta	-- dashed	0 circle
c cyan	: dotted	* asterisk
r red	-. dash-dot	x x-mark
g green	none no line	. point
b blue		^ up triangle
w white		s square
k black		d diamond, etc.

Creating Multiple Plots

A plot with more than one line can be created in several methods. The following example demonstrates the concept:

Example:

```
1.  a = 0: pi/100:2*pi;
2.   b=sin (a);
3.   c = cos (a);
4.  plot (a, c, 'r:'), legend ('Sin(a)', 'Cos(a)')
```

Setting Axis Scales and Annotation Plots

The **axis** command allows us to set the axis scales. We can provide minimum and maximum values for x and y axes using the axis command in the following method

```
1.  axis ( [xmin xmax ymin ymax] )
```

Example

```
1.  axis ([-5 10 2 22]);     sets x-axis from -5 to 10, y-axis from 2 to 22
2.  axy = [-5 10 2 22]; axis (axy);
3.  ax=[-5 10]; ay=[2 22]; axis ([ax ay]);
```

There are also some predefined string arguments for the **axis** command:

axis ('equal')	Sets equal scale on both axes
axis ('square')	Sets the default rectangular frame to the square
axis ('normal')	Resets the axis to default values
axis ('axis')	Freezes the current axis limits
axis ('off')	Delete the surrounding frame and the tick marks.

Generating Sub-Plots

We can use the **subplot** function to display multiple plots in different sub-regions of the same window. The subplot commands require three integer arguments:

```
1.    subplot (m, n, p)
```

Splits the figure into an m x n matrix. The variable p identifies the part of the window where the next plot will be drawn. For example, if the command

```
1.    subplot (2, 2, 1)
```

MATLAB fplot()

It is used to plot between the specific limit. The function must be of form y=f(x), where x is the vector whose specifies the limits, and y is the vector with the same size as x.

Example

```
1.    f(t)=t sin t, 0≤t≤10π
2.    fplot ('x.*sin(x)',[0 10*pi])
```

MATLAB Semilogx()

It generates a plot of the values of x and y, using the logarithmic scale for x and the linear scale for y.Create a plot with logarithmic scales for the x-axis and linear scales for the y-axis.

```
1.    x=e^-t,y=t,0≤t≤2π
2.    t=linspace (0,2*pi,200);
3.    x=exp(-t);
4.    y=t;
5.    semilogx (x, y), grid
```

MATLAB Semilogy()

It generates a plot of the values of x and y, using a linear scale for x and a logarithmic scale for y.Create a plot with logarithmic scales for the y-axis and linear scales for the x-axis.

```
1.    x=t,y=e^t,0≤t≤2π
2.    t=linspace (0, 2*pi,200);
3.    semilogy(t, exp(t))
4.    grid
```

MATLAB loglog()

It generates a plot of the vectors x and y, using a logarithmic scale for both x and y-axis.Create a plot using logarithmic scales for both the x-axis and the y-axis.

```
x=e^t,y=100+e^2t,0≤t≤2π
t=linspace(0, 2*pi,200);
x=exp(t)
y=100+exp(2*t);
loglog(x, y), grid
```

MATLAB Polar Plots()

MATLAB provides plotting capability with polar coordinates:

polar(theta, r)

creates a polar plot of angle theta (in radians) and radial distance r.

Example

Create a Polar Plot

```
r^2=2sin5t,0≤t≤2π
t=linspace(0, 2*pi,200);
r=sqrt(abs(2*sin(5*t)))
polar(t, r)
```

MATLAB fill()

For plotting filled polygons, we may use the **fill()** function. The vertices are recorded along with the color to be filled.

Example

```
r^2=2sin5t,0≤t≤2π
x=r cost,y=r sint
t=linspace (0, 2*pi,200);
r=sqrt(abs(2*sin(5*t)));
x=r.*cos(t);
y=r.*sin(t);
fill(x, y, 'k');
axis('square')
```

MATLAB Bar()

A bar plot is a plot in which each point is represented by a vertical bar or horizontal bar.

Create Bar Graph

```
r^2=2 sin 5t, 0≤t≤2π
y = r sin t
t=linspace (0, 2*pi,200);
r=sqrt(abs(2*sin(5*t)));
y=r.*sin(t);
bar (t, y)
axis ([0 pi 0 inf]);
```

MATLAB errorbar()

Plot error bars along a curve

```
fapproax=x-x^3/3!,0≤x≤2
error=f approax-sin?x
x=0:.1:2;
apprx2=x-x.^3/6;
er=apprx2-sin(x);
errorbar(x,apprx2,er)
```

MATLAB barh()

This function creates a horizontal bar plot, with the values in x used to label each bar and the values in y used to determine the horizontal length of the bar.

Example

```
World population by continents
cont=char ('Asia','Europe','Africa',...'N.America','S.America');
pop=[333.2;696;694;437;307];
barh(pop)
for i=1:5,
    gtext(cont(i,:));
end
xlabel('Population in millions');
Title (World population (1992)','fontsize', 18)
```

MATLAB plotyy()

It creates graphs with y-axes on both the left and right sides.

Example

```
y_1=e^-x sinx,0≤t≤10
y_2=e^x
```

```
x= 1:1:10;
y1=exp(-x).*sin(x);
y2=exp(x);
Ax=plotyy(x, y1, x, y2);
hy1=get(Ax(1),'ylabel');
hy2= get(Ax(2),'ylabel');
set(hy1,'string','e^-x sin(x)');
set(hy2,'string',' e^ x');
```

MATLAB area()

An area plot shows items in Y as one or more curves and fills the area beneath each curve. When Y is the matrix, the curves are stacked, presenting the relative contribution of each row item to the total height of the curve at each x interval.

Example

```
y=sinx/x,-3π≤x≤3π
x=linspace (-3*pi, 3*pi, 100);
y=-sin(x)./x;
area(x, y)
xlabel('x'),ylabel('sin(x)./x')
hold on
x1=x(46:55);y1=y(46:55);
area(x1,y1,'facecolor','y')
```

MATLAB Pie()

This function creates a pie plot. This function determines the percentage of the total pie corresponding to each value of x, and plots pie slices of that size.

The optional array ***explodes*** *controls whether or not individual pie slices are separated from the remainder of the pie.*

Example

```
World population by continents.
cont= char('Asia','Europe','Africa',....'N.America','S.America');
pop=[3332;696;694;437;307];
pie(pop)
for i=1:5,
    gtext (cont(i,:));
end
Title ('World Population (1992)',....'fontsize', 18)
```

MATLAB hist()

A histogram is a plot presenting the distribution of values within a data set. To develop a histogram, the range of values within the data set is split into evenly spaced bins, and the number of data values falling into each bin is determined.

Example

1. Histogram of 50 randomly distributed numbers between 0 and 1.
2. y=randn (50, 1);
3. hist (y)

MATLAB stem()

A two-dimensional stem plot shows data as lines extending from a baseline along the x-axis. A circle (the default) or another marker whose y-position represents the data value terminates each stem.

Example

1. f=e^-t/5 sint,0≤t≤2π
2. t=linspace (0, 2*pi, 200);
3. f=exp (-.2*t).*sin(t);
4. stem(t, f)

MATLAB Stairs()

Stairstep plots are useful for drawing time history plots of digitally sampled-record systems.

Create a plot

1. r^2=2sin5t,0≤t≤2π
2. y= r sin t
3. t=linspace(0,2*pi,200);
4. r=sqrt(abs(2*sin(5*t)));
5. y=r.*sin(t);
6. stairs(t, y);
7. axis[0 pi 0 inf]);

MATLAB compass()

A compass plot shows direction or velocity vectors as arrows arise from the origin. X, Y, and Z are in the cartesian coordinates and plotted on a circular grid.

Example

```
z=cosθ+isinθ,-π≤θ≤π
th=-pi:pi/5:pi;
zx=cos(th);
zy=sin(th);
z=zx+i*zy;
compass(z)
```

MATLAB comet()

A comet plot is an animated graphs in which a circle traces the data points on the screen.

Example

```
y=t sint,0≤t≤10π
q=linspace(0, 10*pi,2000);
y=q.*sin(q);
comet (q, y)
```

MATLAB contour()

A contour plot shows the isolines of matrix Z.

Example

```
z=-1/2 x^2+xy+y^2
|x|≤5,|y|≤5.
r= -5: .2:5;
[X, Y]=meshgrid (r, r);
Z=-.5*X.^2+X.*Y+Y.^2;
cs= contour (X, Y, Z);
clabel (cs)
```

MATLAB quiver()

A quiver plot present velocity vectors as arrows with components (U, V) at the points (X, Y).

Example

```
z=x^2+y^2-5sin?(xy)
|x|≤2,|y|≤2.
r=-2: .2:2;
[X, Y]= meshgrid (r, r);
Z=X.^2- 5*sin(X.*Y)+Y.^2;
[dx, dy]=gradient (Z,.2,.2);
quiver (X, Y, dx, dy, 2);
```

MATLAB pcolor()

A **pseudocolor** plots is a rectangular array of cells with colors persistent by C.

Example

```
z^2=x^2+y^2-5 sin?(xy)
|x|≤2,|y|≤2.
r=-2: .2:2;
[X, Y]= meshgrid (r, r);
Z=X.^2- 5*sin(X.*Y)+Y.^2;
pcolor(Z), axis('off')
shading interp
```

MATLAB 3D Plots

MATLAB also includes a wide variety of three-dimensional plots that can be useful for displaying certain types of data. In general, three-dimensional plots are helpful for displaying two types of data:

1. *Two variables that are functions of the same independent variable, when you wish to emphasize the importance of the independent variable.*
2. *A single variable that is a function of two independent variables, z = f (x, y).*

*To get the z, first, we have to create a set of (x, y) points using the function **meshgrid**.*

MATLAB plot3()

The plot3 function shows a three-dimensional plot of a set of data points.

Example

Plot of a parametric space curve:

```
 x(t)=t,y(t)=t^2,z(t)=t^3.
0≤t≤1
t= linspace (0, 1,100);
x=t; y=t.^2; z=t.^3;
plot3(x, y, z), grid
xlabel ('x(t)=t')
ylabel ('y(t)=t2')
zlabel ('z(t)=t3')
```

MATLAB fill3()

The fill3 function creates flat-shaded and Gouraud-shaded polygons.

Example

Plot of four filled polygons with three vertices each.

```
X= [0 0 0 0; 1 1 -1 1; 1 -1 -1 -1];
Y= [0 0 0 0; 4 4 4 4; 4 4 4 4];
Z= [0 0 0 0; 1 1 -1 -1; -1 1 1 -1];
fillcolor= rand(3, 4);
fill3 (X, Y, Z, fillcolor)
view (120, 30)
```

MATLAB contour3()

contour3 generates a three-dimensional contour plot of a surface defined on a rectangular grid.

Example

Plot of 3-D contour lines

```
z=-5/(1+x^2+y^2 )
|x|≤3,|y|≤3.
r=linspace (-3, 3, 50);
[x, y]=meshgrid (r, r);
z= -5./(1+x.^2+y.^2);
contour3 (x, y, z)
```

MATLAB surf()

This function creates a surface plot.

Example

```
z=cos x cos y e (-√(x^2+y^2 ))/4
|x|≤5,|y|≤5
u=-5:.2:5;
[X, Y]=meshgrid(u, u);
Z=cos(X). *cos(Y).*...
    exp (-sqrt(X.^2+Y.^2)/4);
surf(X, Y, Z)
```

MATLAB surfc()

surfc develop colored parametric surfaces specified by X, Y, and Z, with the color specified by Z or C.

Example

Display contour plot under surface plot.

```
[X,Y] = meshgrid(1:0.5:10,1:20);
Z = sin(X) + cos(Y);
surfc(X,Y,Z)
```

MATLAB mesh()

mesh create wireframe parametric surfaces specified by X, Y, and Z, with the color specified by C.

Example

```
z=-5/(1+x^2+y^2 )
|x|≤3,|y|≤3
x=linspace (-3, 3, 50);
y=x;
[x, y]=meshgrid(x, y);
z=-5./(1+x.^2+y.^2);
mesh(z)
```

MATLAB meshz()

It creates a mesh plot with a curtain around it.

Example

```
z=sin^2 x+sin^2 y
|x|≤π/2,|y|≤π/2
x= linspace(-pi/2,pi/2,50)
y=x;
[x, y]=meshgrid(x, y);
z= sin(x.^2)+sin(y.^2);
meshz(x, y,z), axis tight
view (-37 .5, 50)
```

MATLAB waterfall()

*The waterfall function draws a mesh equal to the **meshz** function, but it does not generate lines from the columns of the matrices.*

Example

```
z=-5/(1+x^2+y^2 )
|x|≤3,|y|≤3
x=linspace (-3, 3, 50);
y=x;
```

```
[x, y] = meshgrid(x, y);
z=-5. / (1+x. ^2+y. ^2);
waterfall(z)
hidden off
```

MATLAB stem3()

Three-dimensional stem plots shows lines extending from the xy-plane. A circle (the default) or other marker symbols whose z-position describes the data value terminates each stem.

Example

Create a three-dimensional stem plot to visualize the function of two variables.

```
x=t, y=tsin(t)
z=e^t/10-1
for 0≤t≤6π
t=linspace(0,6*pi,200);
x=t; y=t.*sin(t);
z=exp(t/10)-1;
stem3(x, y, z,'filled')
```

MATLAB ribbon()

It creates a ribbon plot.

Example

Create a ribbon plot

2-D curves as ribbons in 3-D.

```
y_1=sin?(t),y_2=e^(-.15t) sin?(t)
y_3=e^(-.8t) sin?(t)
for 0≤t≤5π
t=linspace (0, 5*pi, 100);
y1=sin(t);
y2=exp(-.15*t).*sin(t);
y3=exp(-.8*t).*sin(t);
y=[y1;y2;y3];
rib_width=0.2;
ribbon(t',y',rib_width)
```

MATLAB sphere()

*The sphere function develops the x-, y-, and z-coordinates of a unit sphere for use with **surf** and **mesh**.*

Example

Generate and plot a sphere.

```
sphere(20)
axis('square')
or
[x,y,z]=sphere(20);
surf(x, y, z)
axis('square')
```

MATLAB ellipsoid()

It generates an ellipsoid function.

Example

```
cx=0;cy=0;cz=0;
rx=1;ry=2;rz=0.5;
ellipsoid(cx,cy,cz,rx,ry,rz)
axis('equal')
```

MATLAB cylinder()

cylinder creates x, y, and z coordinates of the unit cylinder. We can draw the cylindrical object using surf or mesh, or draw it immediately by not providing output arguments.

Example

```
r=sin?(3π z)+2
0≤z≤1,0≤θ≤2θ
z=[0: .02:1]';
r=sin(3*pi*z)+2;
cyclinder(r), axis square
```

MATLAB slice()

slice shows orthogonal slice planes through volumetric data.

Example

Slice of the volumetric function

```
f(x, y, z)=cos^2 x+cos^2 y-z^2
```

```
|x|≤3,|y|≤3,|z|≤3 at x=-2  and 2,y=2,and z=-2.5 and 0.
v=[-3: .2: 3];
[x, y, z]=meshgrid (v, v, v);
f= (cos(x).^2+sin(y). ^2-z. ^2);
xv= [-2 2. 5];  yv=2;
zv = [-2.5 0];
slice (x, y, z, f, xv, yv, zv);
```