3D plot of a surface in Matlab

For 3-D shaded ***surface plot*** use command:

surf(Z)

surf(Z,C)

surf(X,Y,Z)

surf(X,Y,Z,C)

surf(...,'PropertyName',PropertyValue)

surf(axes_handles,...)

h = surf(...)

surf(Z) creates a three-dimensional shaded surface from the z components in matrix Z, using x = 1:n and y = 1:m, where [m,n] = size(Z). The height, Z, is a single-valued function defined over a geometrically rectangular grid. Z specifies the color data, as well as surface height, so color is proportional to surface height.

surf(Z,C) plots the height of Z, a single-valued function defined over a geometrically rectangular grid, and uses matrix C, assumed to be the same size as Z, to color the surface. See Coloring Mesh and Surface Plots for information on defining C.

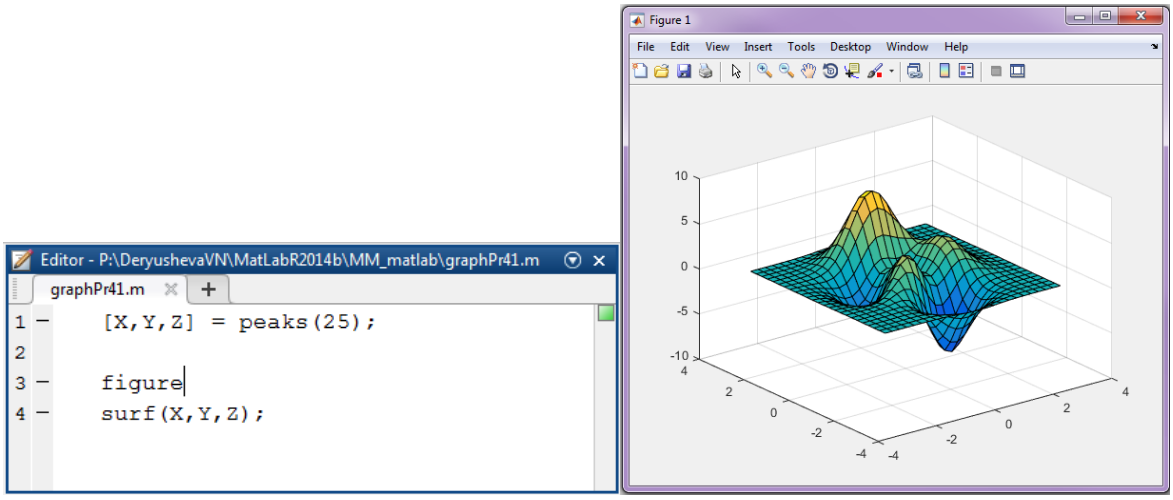
surf(X,Y,Z) uses Z for the color data and surface height. X and Y are vectors or matrices defining the x and y components of a surface. If X and Y are vectors, length(X) = n and length(Y) = m, where [m,n] = size(Z). In this case, the vertices of the surface faces are (X(j), Y(i), Z(i,j)) triples. To create X and Y matrices for arbitrary domains, use the meshgrid function.

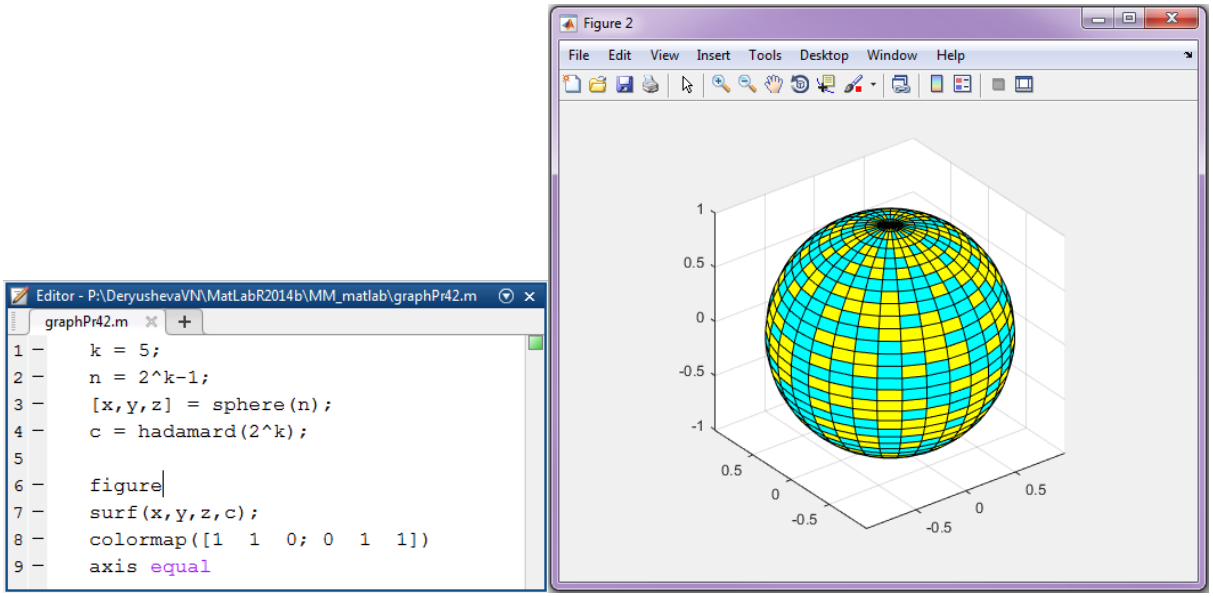
surf(X,Y,Z,C) uses C to define color. MATLAB® performs a linear transformation on this data to obtain colors from the current colormap.

surf(...,'PropertyName',PropertyValue) specifies surface properties along with the data. For a list of properties, see Chart Surface Properties.

surf(axes_handles,...) plots into the axes with handle axes_handle instead of the current axes (gca).

h = surf(...) returns a handle to a chart surface graphics object.





See also: http://www.mathworks.com/help/matlab/ref/surf.html

Mesh plot

mesh(X,Y,Z)

mesh(Z)

mesh(...,C)

mesh(...,'PropertyName',PropertyValue,...)

mesh(axes_handles,...)

h = mesh(...)

mesh(X,Y,Z) draws a wireframe mesh with color determined by Z, so color is proportional to surface height. If X and Y are vectors, length(X) = n and length(Y) = m, where [m,n] = size(Z). In this case, (X(j), Y(i), Z(i,j)) are the intersections of the wireframe grid lines; X and Y correspond to the columns and rows of Z, respectively. If X and Y are matrices, (X(i,j), Y(i,j), Z(i,j)) are the intersections of the wireframe grid lines.

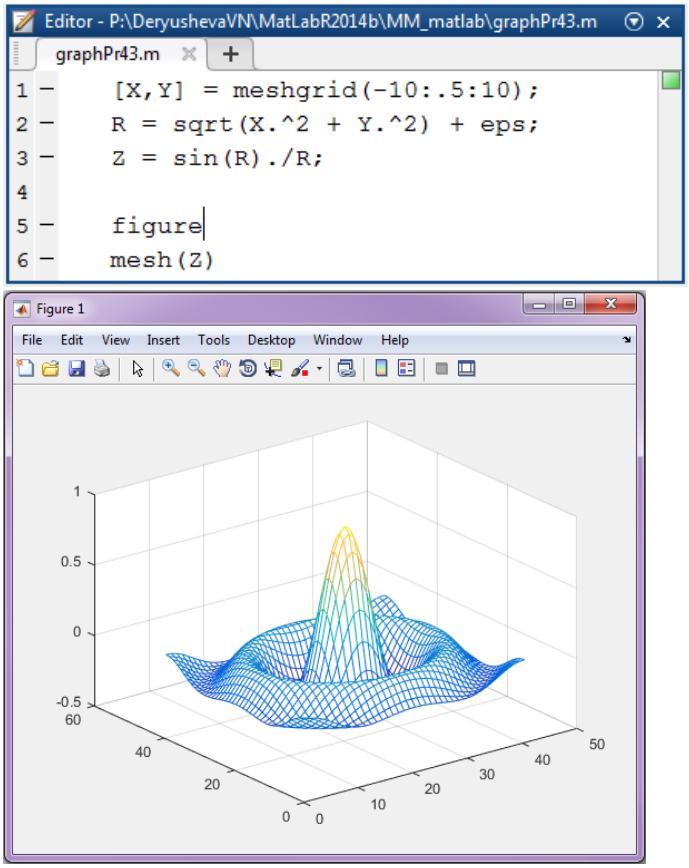
mesh(Z) draws a wireframe mesh using X = 1:n and Y = 1:m, where [m,n] = size(Z). The height, Z, is a single-valued function defined over a rectangular grid. Color is proportional to surface height.

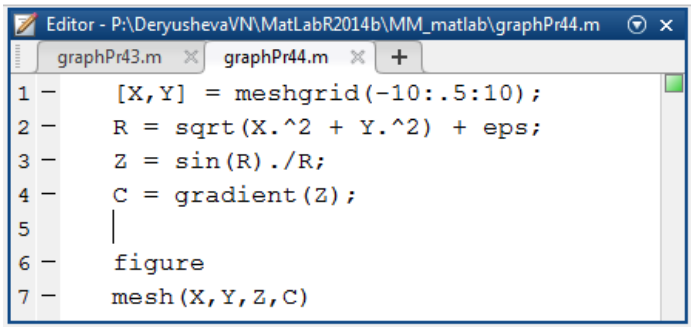
mesh(...,C) draws a wireframe mesh with color determined by matrix C. MATLAB® performs a linear transformation on the data in C to obtain colors from the current colormap. If X, Y, and Z are matrices, they must be the same size as C.

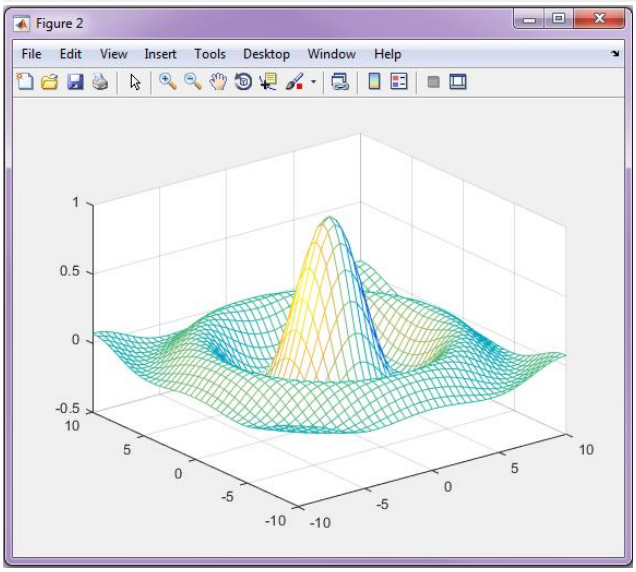
mesh(...,'PropertyName',PropertyValue,...) sets the value of the specified surface property. Multiple property values can be set with a single statement.

mesh(axes_handles,...) plots into the axes with handle axes_handle instead of the current axes (gca).

h = mesh(...) returns a handle to a Chart Surface Properties graphics object.







3-D line plot

plot3(X1,Y1,Z1,...)

plot3(X1,Y1,Z1,LineSpec,...)

plot3(...,'PropertyName',PropertyValue,...)

plot3(axes_handle,...)

h = plot3(...)

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

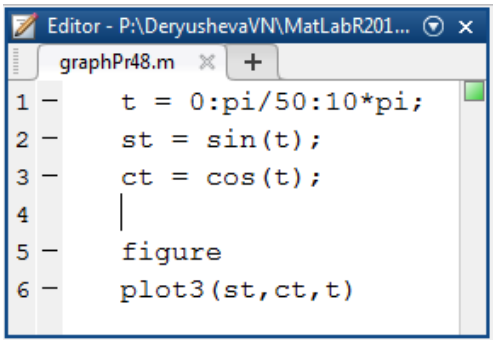
plot3(X1,Y1,Z1,...), where X1, Y1, Z1 are vectors or matrices, plots one or more lines in three-dimensional space through the points whose coordinates are the elements of X1, Y1, and Z1.

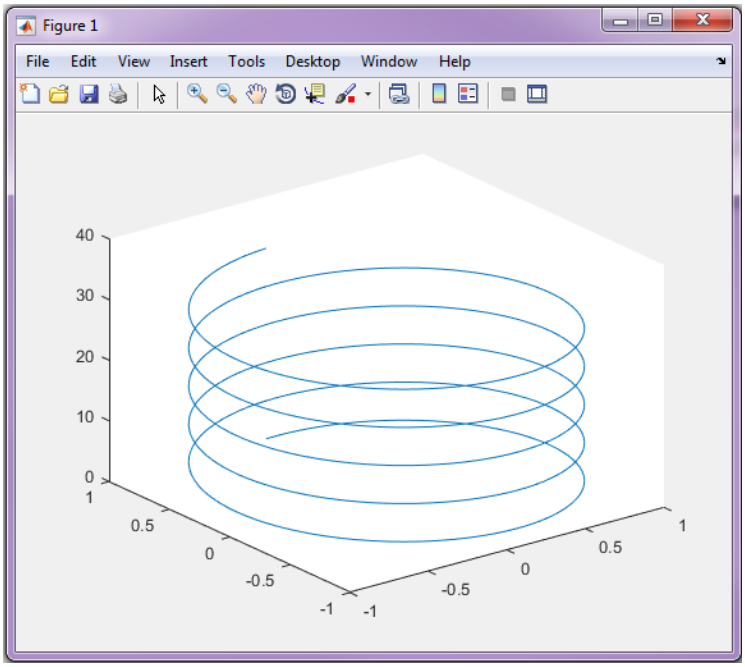
plot3(X1,Y1,Z1,LineSpec,...) creates and displays all lines defined by the Xn,Yn,Zn,LineSpec quads, where LineSpec is a line specification that determines line style, marker symbol, and color of the plotted lines.

plot3(...,'PropertyName',PropertyValue,...) sets line properties to the specified property values for all the charting lines created by plot3. See Chart Line Properties for a description of the properties you can set.

plot3(axes_handle,...) plots into the axes specified by axes_handle instead of into the current axes (gca). The option, axes_handle can precede any of the input combinations in the previous syntaxes.

h = plot3(...) returns a column vector of charting line handles, with one handle per object.





3-D contour plot

contour3(Z)

contour3(Z,n)

contour3(Z,v)

contour3(X,Y,Z)

contour3(X,Y,Z,n)

contour3(X,Y,Z,v)

contour3(...,LineSpec)

contour3(...,Name,Value)

contour3(ax,...)

[C,h] = contour3(...)

contour3 creates a 3-D contour plot of a surface defined on a rectangular grid.

contour3(Z) draws a contour plot of matrix Z in a 3-D view. Z is interpreted as heights with respect to the x-y plane. Z must be at least a 2-by-2 matrix that contains at least two different values. The x values correspond to the column indices of Z and the y values correspond to the row indices of Z. The contour levels are chosen automatically.

contour3(Z,n) draws a contour plot of matrix Z with n contour levels in a 3-D view.

contour3(Z,v) draws a contour plot of matrix Z with contour lines at the values specified in vector v. The number of contour levels is equal to length(v). Specifying the vector v sets the LevelListMode property to manual. To display a single contour line at a particular value, define

v as a two-element vector with both elements equal to the desired contour level. For example, to draw a single contour of level k, use contour3(Z,[k k]).

contour3(X,Y,Z), contour3(X,Y,Z,n), and **contour3(X,Y,Z,v)** draw contour plots of Z using X and Y to determine the x and y values.

If X and Y are vectors, then length(X) must equal size(Z,2) and length(Y) must equal size(Z,1). The vectors must be strictly increasing or strictly decreasing and cannot contain any repeated values.

If X and Y are matrices, then their sizes must equal the size of Z. Typically, you should set X and Y so that the columns are strictly increasing or strictly decreasing and the rows are uniform (or the rows are strictly increasing or strictly decreasing and the columns are uniform).

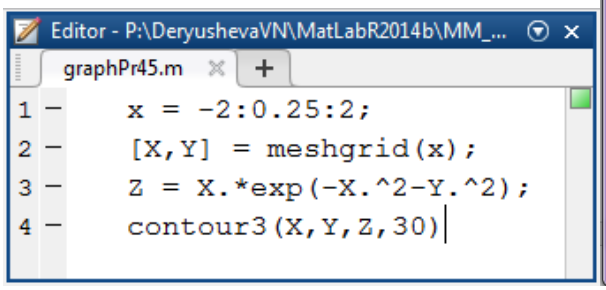
If X or Y is irregularly spaced, then contour3 calculates contours using a regularly spaced contour grid, and then transforms the data to X or Y.

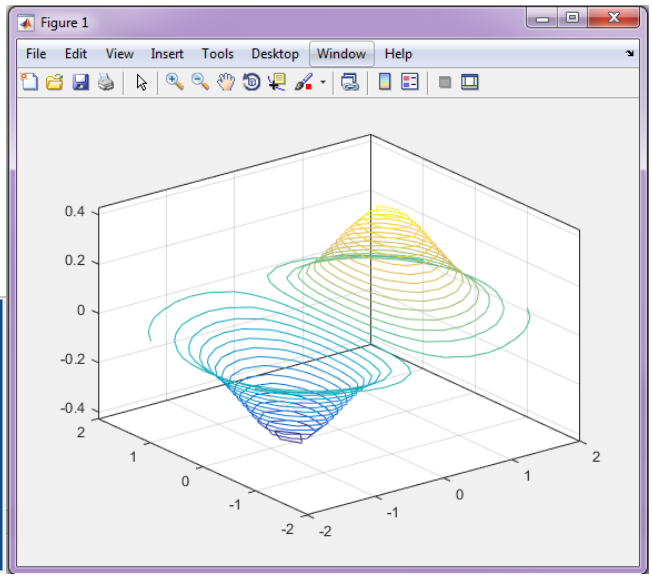
contour3(...,LineSpec) draws the contour lines using the line type and color specified by LineSpec. contour3 ignores marker symbols.

contour3(...,Name,Value) specifies contour properties using one or more property name, property value pairs. Name is the property name and must appear inside single quotes ('). Value is the corresponding value. For example, 'LineWidth',2 sets the contour line width to 2. For a list of contour property names and values, see Contour Properties.

contour3(ax,...) plots into the axes specified by ax instead of into the current axes (gca).

[C,h] = contour3(...) returns the contour matrix C containing the data that defines the contour lines and the contour object h. The clabel function uses the contour matrix to label the contour lines.





Draw contours in volume slice planes

contourslice(X,Y,Z,V,Sx,Sy,Sz)

contourslice(X,Y,Z,V,Xi,Yi,Zi)

contourslice(V,Sx,Sy,Sz)

contourslice(V,Xi,Yi,Zi)

contourslice(...,n)

contourslice(...,cvals)

contourslice(...,[cv cv])

contourslice(...,'method')

contourslice(axes_handle,...)

h = contourslice(...)

contourslice(X,Y,Z,V,Sx,Sy,Sz) draws contours in the x-, y-, and z-axis aligned planes at the points in the vectors Sx, Sy, Sz. The arrays X, Y, and Z define the coordinates for the volume V and must be monotonic and represent a Cartesian, axis-aligned grid (such as the data produced by meshgrid). The color at each contour is determined by the volume V, which must be an m-by-n-by-p volume array.

contourslice(X,Y,Z,V,Xi,Yi,Zi) draws contours through the volume V along the surface defined by the 2-D arrays Xi,Yi,Zi. The surface should lie within the bounds of the volume.

contourslice(V,Sx,Sy,Sz) and contourslice(V,Xi,Yi,Zi) (omitting the X, Y, and Z arguments) assume [X,Y,Z] = meshgrid(1:n,1:m,1:p), where [m,n,p]= size(v).

contourslice(...,n) draws n contour lines per plane, overriding the automatic value.

contourslice(...,cvals) draws length(cval) contour lines per plane at the values specified in vector cvals.

contourslice(...,[cv cv]) computes a single contour per plane at the level cv.

contourslice(...,'method') specifies the interpolation method to use. method can be linear, cubic, or nearest. nearest is the default except when the contours are being drawn along the surface defined by Xi, Yi, Zi, in which case linear is the default. (See interp3 for a discussion of these interpolation methods.)

contourslice(axes_handle,...) plots into the axes with the handle axes_handle instead of into the current axes (gca).

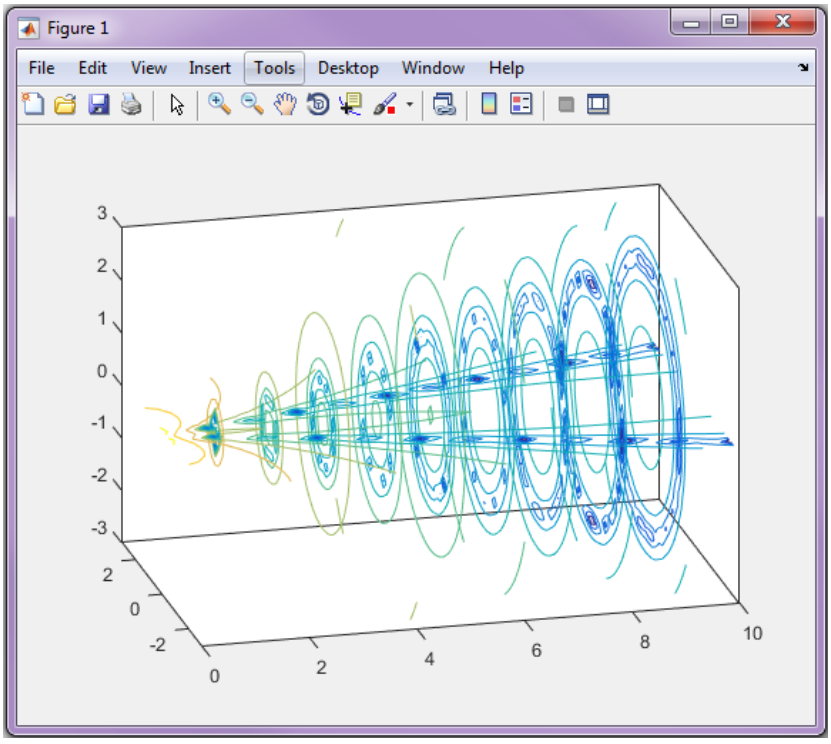
h = contourslice(...) returns a vector of handles to patch objects that are used to implement the contour lines.

Editor - P:\DeryushevaVN\MatLabR2014b\MM_matlab\graphPr47.m

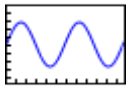
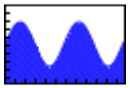
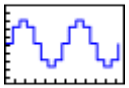
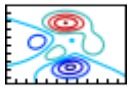
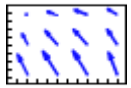
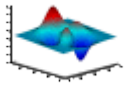
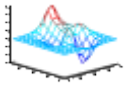
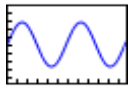
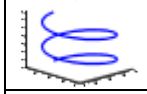
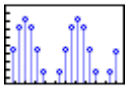
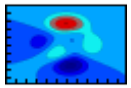
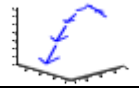
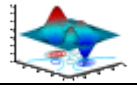
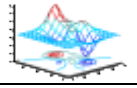
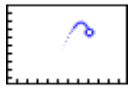
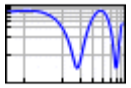
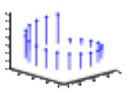
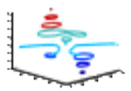
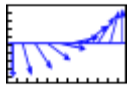
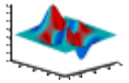
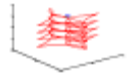
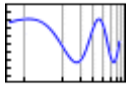
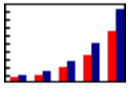
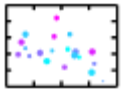
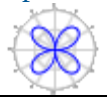
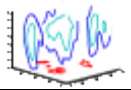
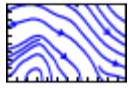
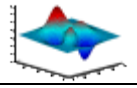
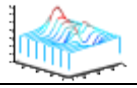
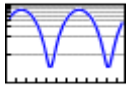
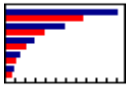
graphPr47.m

```
%Store the matrices X, Y, Z, and V from the flow data set.
[X,Y,Z,V] = flow;
%Create nine contour plots in the y-z plane,
%no plots in the x-z plane, and one plot in the x-y plane
%by specifying Sx as a vector of nine elements,
%Sy as an empty vector, and Sz as a scalar.
Sx = 1:9;
Sy = [];
Sz = 0;
%Draw 10 contour lines between -8 and 2 by specifying cvals as
%a 10-element vector of linearly spaced values between -8 and 2.
cvals = linspace(-8,2,10);
%Create the contour slice plots and set the axis limits.
%Set the data aspect ratio, change the camera position,
%and display the black box outline.
figure
contourslice(X,Y,Z,V,Sx,Sy,Sz,cvals)

axis([0,10,-3,3,-3,3])
daspect([1,1,1])
campos([0,-20,7])
box on
```



Common Graphics Functions in MATLAB http://www.mathworks.com/help/matlab/creating_plots/types-of-matlab-plots.html

ine Plots	Pie Charts, Bar Plots, and Histograms	Discrete Data Plots	Polar Plots	Contour Plots	Vector Fields	Surface and Mesh Plots		Polygons	Animation
plot 	area 	stairs 	polar 	contour 	quiver 	surf 	mesh 	fill 	animatedline 
plot3 	pie 	stem 	rose 	contourf 	quiver3 	surfc 	meshc 	fill3 	comet 
loglog 	pie3 	stem3 	compass 	contour3 	feather 	surfl 	meshz 	patch 	comet3 
semilogx 	bar 	scatter 	ezpolar 	contourslice 	streamslice 	ezsurf 	waterfall 		
semilogy 	barh 	scatter3		ezcontour	streamline	ezsurfc	ezmesh		
errorbar	bar3	spy		ezcontourf	streamribbon	ribbon	ezmeshc		
ezplot	bar3h	plotmatrix			streamtube	pcolor			
ezplot3	histogram				coneplot				

	pareto