



## TOPIC 8

### Computer application for drawing 3D graph

#### Basic 3-D Plotting: The plot3 function

The 3-D analog of the plot function is plot3. If  $x$ ,  $y$ , and  $z$  are three vectors of the same length, `plot3(x,y,z)` generates a line in 3-D through the points whose coordinates are the elements of  $x$ ,  $y$ , and  $z$  and then produces a 2-D projection of that line on the screen. For example, these statements produce a helix:

```
>>t = 0:pi/50:10*pi;  
>>plot3(sin(t),cos(t),t)  
>>axis square; grid on
```

#### Plotting Matrix Data

If the arguments to plot3 are matrices of the same size, lines obtained from the columns of  $X$ ,  $Y$ , and  $Z$  are plotted. For example:

```
>>[X,Y] = meshgrid([-2:0.1:2]);  
>>Z = X.*exp(-X.^2-Y.^2);  
>>plot3(X,Y,Z), grid on
```

#### Functions for Plotting Data Grids

MATLAB graphics defines a surface by the  $z$ -coordinates of points above a rectangular grid in the  $x$ - $y$  plane. The plot is formed by joining adjacent points with straight lines. Surface plots are useful for visualizing matrices that are too large to display in numerical form and for graphing functions of two variables.

MATLAB can create different forms of surface plots. This table lists the various forms.

Function	Used to Create
<code>mesh</code> , <code>surf</code>	Surface plot
<code>meshc</code> , <code>surfc</code>	Surface plot with contour plot beneath it
<code>meshz</code>	Surface plot with curtain plot (reference plane)
<code>pcolor</code>	Flat surface plot (value is proportional only to color)
<code>surf1</code>	Surface plot illuminated from specified direction
<code>surface</code>	Low-level function (on which high-level functions are based) for creating surface graphics object

## Plotting Surfaces

A surface is defined mathematically by a function  $f(x, y)$  corresponding to each value of  $(x, y)$  we compute the height of the function by

$$z = f(x, y):$$

In order to plot this we have to decide on the ranges of  $x$  and  $y$ -suppose  $2 \leq x \leq 4$  and  $1 \leq y \leq 3$ . This gives us a square in the  $(x, y)$ -plane. Next, we need to choose a grid on this domain; Figure 1 shows the grid with intervals 0.5 in each direction.

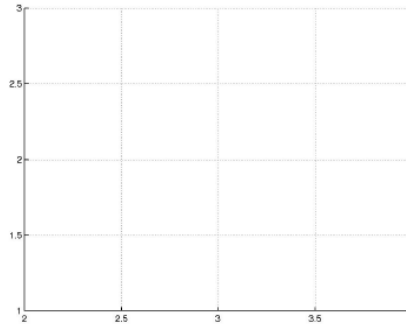


Figure 1: An example of a 2D grid

Finally, we have to evaluate the function at each point of the grid and “plot” it. Suppose we choose a grid with intervals 0.5 in each direction for illustration. The  $x$ - and  $y$ -coordinates of the grid lines are  $x = 2:0.5:4$ ;  $y = 1:0.5:3$ ; in Matlab notation. We construct the grid with **meshgrid**:

```
>> [X,Y] = meshgrid(2:.5:4, 1:.5:3)
X =
 2.0000  2.5000  3.0000  3.5000  4.0000
 2.0000  2.5000  3.0000  3.5000  4.0000
 2.0000  2.5000  3.0000  3.5000  4.0000
 2.0000  2.5000  3.0000  3.5000  4.0000
 2.0000  2.5000  3.0000  3.5000  4.0000
Y =
 1.0000  1.0000  1.0000  1.0000  1.0000
 1.5000  1.5000  1.5000  1.5000  1.5000
 2.0000  2.0000  2.0000  2.0000  2.0000
 2.5000  2.5000  2.5000  2.5000  2.5000
 3.0000  3.0000  3.0000  3.0000  3.0000
```

If we think of the  $i$ th point along from the left and the  $j$ th point up from the bottom of the grid as corresponding to the  $(i, j)$ th entry in a matrix, then  $(X(i,j), Y(i,j))$  are the coordinates of the point. We then need to evaluate the function  $f$  using  $X$  and  $Y$  in place of  $x$  and  $y$ , respectively.

1. To display a function of two variables,  $z = f(x,y)$ , generate  $X$  and  $Y$  matrices consisting of repeated rows and columns, respectively, over the domain of the function. You will use these matrices to evaluate and graph the function.
2. The `meshgrid` function transforms the domain specified by two vectors,  $x$  and  $y$ , into matrices  $X$  and  $Y$ . You then use these matrices to evaluate functions of two variables: The rows of  $X$  are copies of the vector  $x$  and the columns of  $Y$  are copies of the vector  $y$ .

### Example: Illustrating the Use of meshgrid

To illustrate the use of meshgrid, consider the  $\sin(r)/r$  or sinc function. To evaluate this function between -8 and 8 in both x and y, you need pass only one vector argument to meshgrid, which is then used in both directions.

```
>>[X,Y] = meshgrid(-8:.5:8);  
>>R = sqrt(X.^2 + Y.^2) + eps;
```

The matrix R contains the distance from the center of the matrix, which is the origin. Adding eps prevents the divide by zero (in the next step) that produces Inf values in the data.

Forming the sinc function and plotting Z with mesh results in the 3-D surface.

```
>>Z = sin(R)./R;  
>>mesh(X,Y,Z)
```

Plot the mesh graph of the function  $f(x,y) = xe^{-(x^2+y^2)}$  in the interval  $-2 \leq x \leq 2$ , and  $y=x$

### Emphasizing Surface Shape

MATLAB provides a number of techniques that can enhance the information content of your graphs. For example, this graph of the sinc function uses the same data as the previous graph, but employs lighting and view adjustment to emphasize the shape of the graphed function (daspect, axis, view, camlight).

```
>>surf(X,Y,Z,'FaceColor','interp',...  
      'EdgeColor','none',...  
      'FaceLighting','phong')  
>>daspect([5 5 1])  
>>axis tight  
>>view(-50,30)  
>>camlight left
```

### Surface Plots of Nonuniformly Sampled Data

You can use meshgrid to create a grid of uniformly sampled data points at which to evaluate and graph the sinc function. MATLAB then constructs the surface plot by connecting neighboring matrix elements to form a mesh of quadrilaterals.

To produce a surface plot from nonuniformly sampled data, first use griddata to interpolate the values at uniformly spaced points, and then use mesh and surf in the usual way.

### Example – Displaying Nonuniform Data on a Surface

This example evaluates the sinc function at random points within a specific range and then generates uniformly sampled data for display as a surface plot. The process involves these tasks:

- Use `linspace` to generate evenly spaced values over the range of your unevenly sampled data.
- Use `meshgrid` to generate the plotting grid with the output of `linspace`.
- Use `griddata` to interpolate the irregularly sampled data to the regularly spaced grid returned by `meshgrid`.
- Use a plotting function to display the data.

1. Generate unevenly sampled data within the range `[-8, 8]` and use it to evaluate the function:

```
>>x = rand(100,1)*16 - 8;  
>>y = rand(100,1)*16 - 8;  
>>r = sqrt(x.^2 + y.^2) + eps;  
>>z = sin(r)./r;
```

2. The `linspace` function provides a convenient way to create uniformly spaced data with the desired number of elements. The following statements produce vectors over the range of the random data with the same resolution as that generated by the `-8:.5:8` statement in the previous sinc example:

```
>>xlin = linspace(min(x),max(x),33);  
>>ylin = linspace(min(y),max(y),33);
```

3. Now use these points to generate a uniformly spaced grid:

```
>> [X,Y] = meshgrid(xlin,ylin);
```

4. The key to this process is to use `griddata` to interpolate the values of the function at the uniformly spaced points, based on the values of the function at the original data points (which are random in this example). This statement uses a triangle-based cubic interpolation to generate the new data:

```
>>Z = griddata(x,y,z,X,Y,'cubic');
```

5. Plot the interpolated and the nonuniform data to produce:

```
>>mesh(X,Y,Z) %interpolated  
>>axis tight; hold on  
>>plot3(x,y,z,'.','MarkerSize',15) %nonuniform
```

### Parametric Surfaces

The functions that draw surfaces can take two additional vector or matrix arguments to describe surfaces with specific  $x$  and  $y$  data. If  $Z$  is an  $m$ -by- $n$  matrix,  $x$  is an  $n$ -vector, and  $y$  is an  $m$ -vector, then

```
mesh(x,y,Z,C)
```

describes a mesh surface with vertices having color  $C(i,j)$  and located at the points

```
(x(j), y(i), Z(i,j))
```

where  $x$  corresponds to the columns of  $Z$  and  $y$  to its rows.

More generally, if  $X$ ,  $Y$ ,  $Z$ , and  $C$  are matrices of the same dimensions, then

```
mesh(X,Y,Z,C)
```

describes a mesh surface with vertices having color  $C(i,j)$  and located at the points  $(X(i,j), Y(i,j), Z(i,j))$

This example uses spherical coordinates to draw a sphere and color it with the pattern of pluses and minuses in a Hadamard matrix, an orthogonal matrix used in signal processing coding theory. The vectors  $\theta$  and  $\phi$  are in the range  $-\pi \leq \theta \leq \pi$  and  $-\pi/2 \leq \phi \leq \pi/2$ . Because  $\theta$  is a row vector and  $\phi$  is a column vector, the multiplications that produce the matrices  $X$ ,  $Y$ , and  $Z$  are vector outer products.

```
>>k = 5;
>>n = 2^k-1;
>>theta = pi*(-n:2:n)/n;
>>phi = (pi/2)*(-n:2:n)'/n;
>>X = cos(phi)*cos(theta);
>>Y = cos(phi)*sin(theta);

>>Z = sin(phi)*ones(size(theta));
>>colormap([0 0 0;1 1 1])
>>C = hadamard(2^k);
>>surf(X,Y,Z,C)
>>axis square
```

### Hidden Line Removal

By default, MATLAB removes lines that are hidden from view in mesh plots, even though the faces of the plot are not colored. You can disable hidden line removal and allow the faces of a mesh plot to be transparent with the command

```
>>hidden off
```

This is the surface plot with hidden set to off.

### Exercise.

1. Plot the surface defined by the function

$$f(x, y) = (x - 3)^2 - (y - 2)^2 \text{ for } 2 \leq x \leq 4 \text{ and } 1 \leq y \leq 3.$$

2. Repeat the previous example replacing mesh by surf and then by surf1.

Consult the help pages to find out more about these functions.

3. Plot the surface defined by the function  $f = -xye^{-2(x^2+y^2)}$  on the domain  $-2 \leq x \leq 2; -2 \leq y \leq 2$ .

Find the values and locations of the maxima and minima of the function.