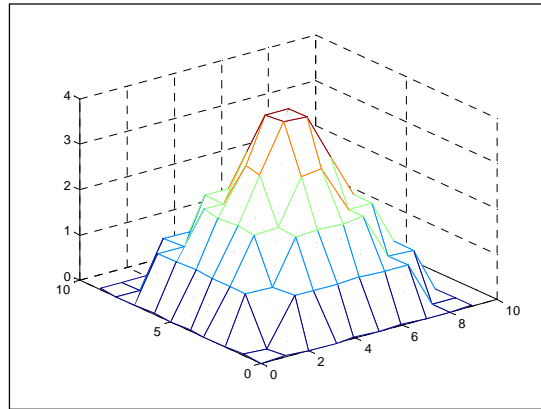


An Example of 3-Dimensional Plotting in Matlab

This example will produce this 3-D plot.



First of all, Matlab stands for matrix laboratory, so it is important to remember that Matlab thinks in matrices. A 3-D plot will consist of an x-y grid with discrete z values for each x-y point.

To generate a 3-D plot with experimental data, the easiest thing to do is to write a script m-file. A Matlab m-file is a file that is executed line by line as if entered in the command window. Writing a script file is the best way because you can set up everything before you plot.

To begin a new script file, click on [File, New, Script](#), or you can just click the [New Script](#) button on the toolbar. Either of these will open the Editor window.

Here's a few Matlab hints. A percent (%) character denotes a comment, a semicolon (;) at the end of a line tells Matlab not to display the results of what it does, and square brackets denote matrices. In a matrix columns are separated by spaces and rows are separated by either carriage returns or semicolons.

Two examples of entering a matrix:

If you input

```
x=[1 2 3;4 5 6;7 8 9]
```

the result is a matrix:

```

x =
  1  2  3
  4  5  6
  7  8  9
    
```

If you input

```

x=[1 2 3
  4 5 6
  7 8 9]
    
```

the result is a matrix:

```

x =
  1  2  3
  4  5  6
  7  8  9
    
```

For a 3-D plot, Matlab needs the x,y,z data. The data needs to be of the form:

$$\bar{x} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad \bar{y} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \quad \bar{z} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}.$$

Notice the format of the y data, if it is too confusing to enter the y data in that way, enter it like the x data and transpose the matrix with the apostrophe (see example.) Remember that Matlab is creating a grid for the z data.

Following is an example m-file to generate a 3-D plot. You can get the same results by typing everything into the command window, but an m-file gives you the flexibility of editing everything before execution.

```
%EXAMPLE.M
%Example of 3-D plot in Matlab
%Experimental data isn't always spaced perfectly, so neither is
%the data for this example
%Here's the x data
x=[0 1.1 2 3.2 4 5 6 7 8 9
    .1 1 2.2 3 4 5 6 7 8 9
    0 1 2.1 3 4 5 6 7 8 9
    0 1 2 3.1 4 5 6 7 8 9
    .1 1 2.1 3 4.1 5 6 7 8 9
    0 1.1 2 3.2 4 5 6 7 8 9
    .1 1 2.2 3 4 5 6 7 8 9
    0 1 2.1 3 4 5 6 7 8 9
    0 1 2 3.1 4 5 6 7 8 9
    .1 1 2.1 3 4.1 5 6 7 8 9];
%Here's the y data
%a
y=[0 1.1 2 3.2 4 5 6 7 8 9
    .1 1 2.2 3 4 5 6 7 8 9
    0 1 2.1 3 4 5 6 7 8 9
    0 1 2 3.1 4 5 6 7 8 9
    .1 1 2.1 3 4.1 5 6 7 8 9
    0 1.1 2 3.2 4 5 6 7 8 9
    .1 1 2.2 3 4 5 6 7 8 9
    0 1 2.1 3 4 5 6 7 8 9
    0 1 2 3.1 4 5 6 7 8 9
    .1 1 2.1 3 4.1 5 6 7 8 9]';
%Here's the z data
z=[0 0 0 0 0 0 0 0 0 0
    0 0 1 1 1 1 1 1 0 0
    0 1 1 2 2 2 2 1 1 0
    0 1 2 2 3 3 2 2 1 0
    0 1 2 3 4 4 3 2 1 0
    0 1 2 3 4 4 3 2 1 0
    0 1 2 2 3 3 2 2 1 0
    0 1 1 2 2 2 2 1 1 0
    0 0 1 1 1 1 1 1 0 0
    0 0 0 0 0 0 0 0 0 0];
%There are several ways to do 3-D plots, here are a few
%plot3(x,y,z)
%mesh(x,y,z)
%surf(x,y,z)
%Note: Matlab is case sensitive, so X and x aren't the same thing.
mesh(x,y,z)
%end EXAMPLE.M
```

To access your script m-file, from the command window screen, under the **Home** tab, in the **Environment** grouping there is a button called **Set Path**. When the **Set Path** box opens, click on the **Add Folder** button and then select the directory where your file is saved. If you are working out of the TEMP directory, type **cd c:\temp** at the command line. If you want to see what directory you are working out of, type **pwd**. Type the name of your m-file to execute it (i.e. example.) Alternatively, you can click **Debug, Run** from your m-file window. For additional help with

Matlab, go to the **Home** tab on the command window and click the **Help** button in the **Resources** grouping, or type **help** and then **graph3d**, **mesh**, **plot3**, **surf**, **contour**, **quiver**, etc. at the Matlab prompt.