

MATLAB Marina: Plotting 3D (Surface and Contour)

Student Learning Objectives

After completing this module, one should:

1. Be able to generate and appropriately annotate surface and contour plots using MATLAB.

Terms

3D plot, parametric plot, surface plot, contour plot

MATLAB Functions, Keywords, and Operators

figure, meshgrid, mesh, surf, contour, xlabel, ylabel, zlabel, title, legend, gtext, grid, axis, clf, close, view, colormap, shading

Surface Plots

Surface plots are used to present three-dimensional data. For 3D surface plots, the x y z coordinates specify points in three space and the 3D surface plot functions generate smooth surfaces connecting adjacent points. Surface plots are typically a plot of a function of two independent variables. The two independent variables specify a 2D grid which the third variable is plotted versus.

The process for creating surface plots consists of three steps:

1. Create a grid of points specifying all the valid points in the x-y plane (all x y locations). Typically this is done by creating two 1D arrays defining the edges of the grid and then using the `meshgrid` function to create 2D arrays defining the underlying grid.
2. Compute the z values for the grid, i.e. evaluate the function describing the surface for the grid.
3. Create the surface plot using one of the MATLAB 3D surface plot functions: `mesh`, `meshc`, `surf`, or `surfc`.

Consider plotting the paraboloid $z = 4 - x^2 - y^2$ over the range $-5 \leq x \leq 5$ and $-5 \leq y \leq 5$. The MATLAB program of Figure 2a generates the paraboloid of Figure 2b.

The `meshgrid` function takes two 1D arrays specifying the edges of the grid for the 3D plot and replicates the rows and columns to produce 2D rectangular matrices specifying the underlying grid for a 3D plot. Figure 3 illustrates the result of the `meshgrid` function.

The `mesh` function generates a surface plot with the surfaces as white facets outlined by colored lines. The line colors are proportional to the z axis values. The `surf` function generates a surface plot with the surfaces as colored facets outlined by black lines. The surface colors are proportional to the z axis values. The functions `meshc` and `surfc` generate mesh and surface plots just as `mesh` and `surf` do but also generate a contour plot below the surface plot.

```
% surface plot of paraboloid
clear; clc; close all;
% 1D arrays defining edges of grid
x = linspace(-5.0,5.0,100);
y = linspace(-5.0,5.0,100);
% 2D arrays defining the underlying grid
[xx, yy] = meshgrid(x,y);
% evaluate the function describing the surface for the grid
zz = 4 - xx.^2 - yy.^2;

figure(1)
mesh(xx,yy,zz);
colormap hsv; colorbar; shading interp;
xlabel("x"), ylabel("y"), zlabel("z");
title("Paraboloid");
```

Figure 2a. MATLAB Program for Surface Plot of Paraboloid

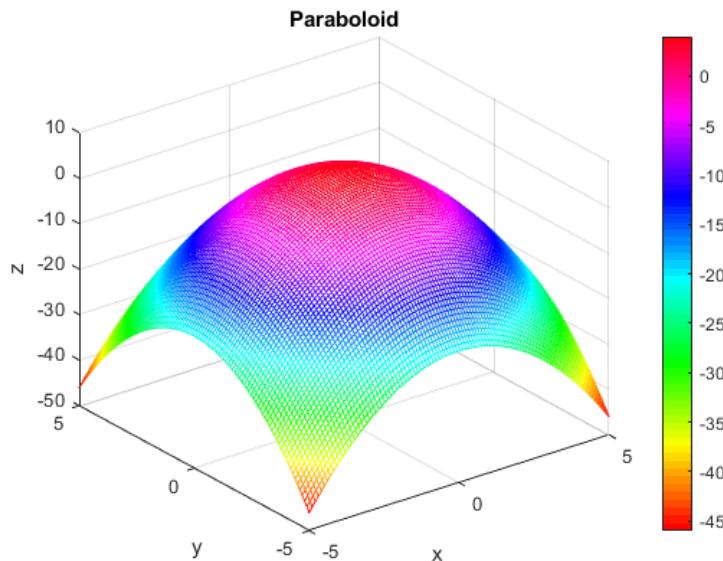


Figure 2b. Surface Plot of Paraboloid with Colorbar

The paraboloid of Figure 2b was generated using rectangular coordinates. Notice that the chosen range does not give all the $z = 0$ points. Some surface plots are easier to specify using polar (r, θ, z) or spherical (ρ, θ, φ) coordinates.

```

x = 1:1:3;
y = -2:1:2;
[xx,yy] = meshgrid(x,y)

xx =
    1     2     3
    1     2     3
    1     2     3
    1     2     3
    1     2     3

yy =
   -2     -2     -2
   -1     -1     -1
    0     0     0
    1     1     1
    2     2     2

```

Figure 3. `meshgrid` Function Results

To plot the paraboloid $z = 4 - x^2 - y^2$ for $z \geq 0$, it is more straightforward to use polar coordinates. The projection of the paraboloid on the x-y plane is a circle of radius 2 described by the equation $x^2 + y^2 = 4$. The height of the paraboloid above the x-y plane is 4. One can think of the paraboloid as a series of circles with different radius piled on top of each other. The radius of the circle for a particular z value is $r = \sqrt{4 - z}$.

The MATLAB program of Figure 4a generates the paraboloid of Figure 4b using polar coordinates for the underlying x-y grid.

```

%% surface plot of paraboloid using polar coordinates
clear; clc; close all;
% 1D arrays defining edges of grid using polar coordinates
theta = linspace(0,2*pi,100);
z = linspace(0.0,4.0,100);
% 2D arrays defining the underlying grid
[zz,thth] = meshgrid(z, theta);
% evaluate the functions describing the surface for the grid
rr = sqrt(4 - zz);
xx= rr.*cos(thth);
yy = rr.*sin(thth);

figure(1)
surf(xx,yy,zz);
colormap hsv; colorbar; shading interp;
xlabel("x"), ylabel("y"), zlabel("z");
title("Paraboloid");

```

Figure 4a. MATLAB Program for Surface Plot of Paraboloid using Polar Coordinates

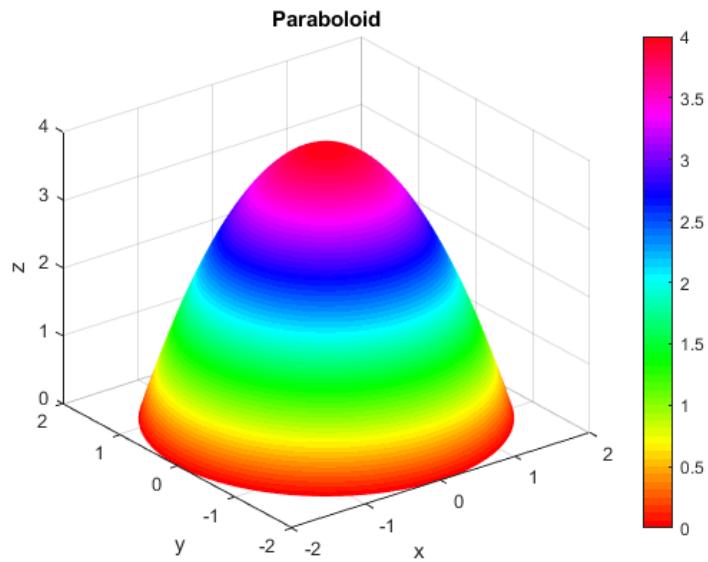


Figure 4b. Surface Plot of Paraboloid

Some other useful MATLAB functions for surface plots are `hidden` and `lightangle`. See MATLAB's help for more information on these functions.

Contour Plots

Contour plots are plots of z plane slices of a 3-dimensional surface. Each slice is a 2D contour. Besides the `meshc` and `surf` functions which generate a contour along with a surface plot, MATLAB has the functions `contour` and `contour3` for 2D and 3D contour plots.

Figures 5b – 5d show examples of a 2D contour, 3D contour, and surface with contour generated using the MATLAB code of Figure 5a. These contour plots use the paraboloid surface $z = 4 - x^2 - y^2$ generated with polar coordinates.

```
% 2D contour of paraboloid
figure(1)
contour(xx,yy,zz)
% 3D contour of paraboloid
figure(2)
contour3(xx,yy,zz)
% surface with contour of paraboloid
figure(3)
meshc(xx,yy,zz), hidden off;
```

Figure 5a. MATLAB Code to Create Contour Plots

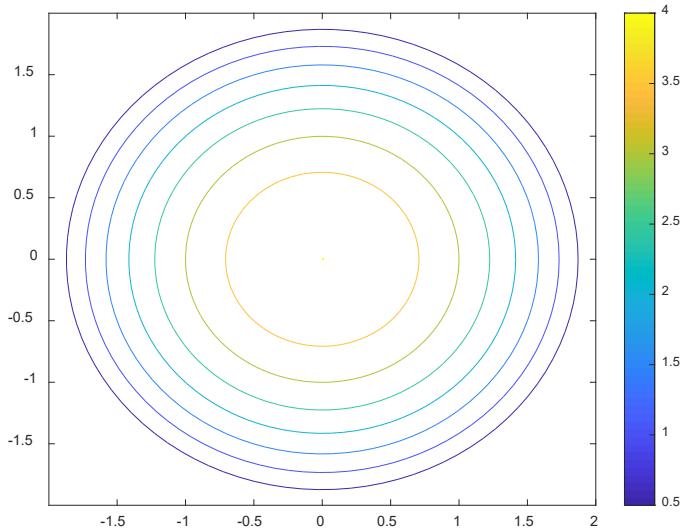


Figure 5b. 2D Contour of Paraboloid

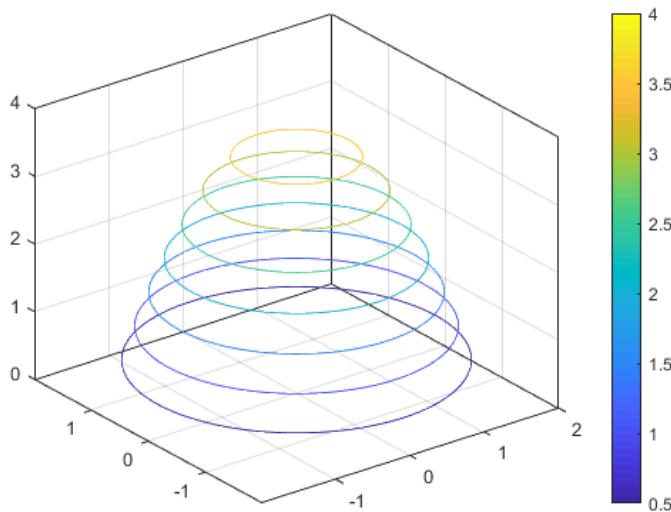


Figure 5c. 3D Contour of Paraboloid

The `hidden off` statement is used for the plot of Figure 5d so that the contour can be seen through the surface plot. The number of contours is determined automatically or can be specified with an optional fourth argument, `contour (xx, yy, zz, N)` or `contour3 (xx, yy, zz, N)` where `N` specifies the number of contours.

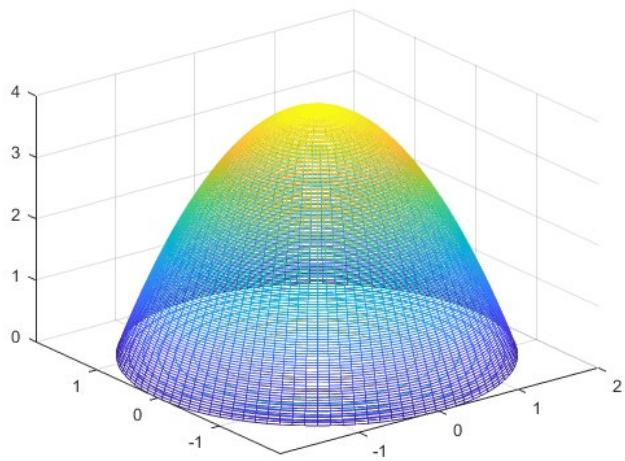


Figure 5d. Surface and Contour of Paraboloid

Last modified Friday, September 25, 2020

[MATLAB Marina](#) is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.