Chapter 10

Three-Dimensional Plots

MATLAB An Introduction With Applications, 6th Edition
Dr. Amos Gilat
The Ohio State University

Slide deck by
Dr. Greg Reese
Miami University

Three-dimensional (3-D) plots useful for presenting related 3-D points. Common cases are

- Scalar or vector function of two independent variables
- Scalar or vector data measurements in 3-D space
- Movement over time 3-D space

MATLAB has many commands for making 3-D plots. Will study

- Line plots
- Wire plots
- Surface plots
- Mesh plots

For more information, see **Plotting and Data Visualization** in Help Window

A *three-dimensional line plot* is a plot obtained by connecting points in 3-D space. MATLAB command is

```
plot3(x,y,z,'line specifiers','PropertyName',property value)
```

x, y, and z are vectors of the coordinates of the points.

(Optional) Specifiers that define the type and color of the line and markers.

(Optional) Properties with values that can be used to specify the line width, and marker's size and edge and fill colors.

- x,y, and z must be same size
- Remaining arguments are same as in 2-D plots (Section 5.1)

If the spatial coordinates of a set of points are each functions of the same independent variable, the coordinates form a set of *parametric equations*.

- Often independent variable is time (t) and the set shows how a particle moves through space over time

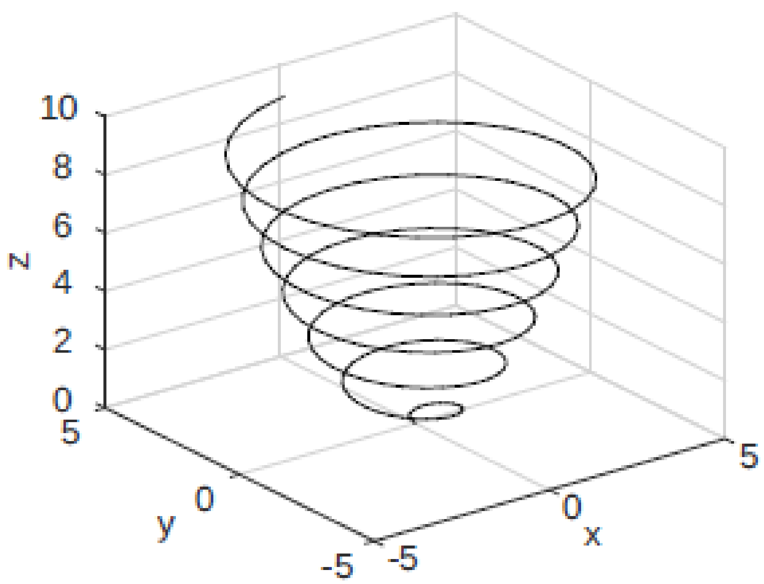
EXAMPLE

Suppose the spatial coordinates vary with time as

$$
\begin{aligned}
x &= \sqrt{t}\sin(2t) \\
y &= \sqrt{t}\cos(2t) \\
z &= 0.5t
\end{aligned}
$$

Make a line plot for $0 \le t \le 6\pi$

```
t=0:0.1:6*pi;
x=sqrt(t).*sin(2*t);
y=sqrt(t).*cos(2*t);
z=0.5*t;
plot3(x,y,z,'k','linewidth',1)
grid on
xlabel('x'); ylabel('y'); zlabel('z')
```



Mesh and surface plots are 3-D plots used to graph functions of the form
$z = f(x,y)$

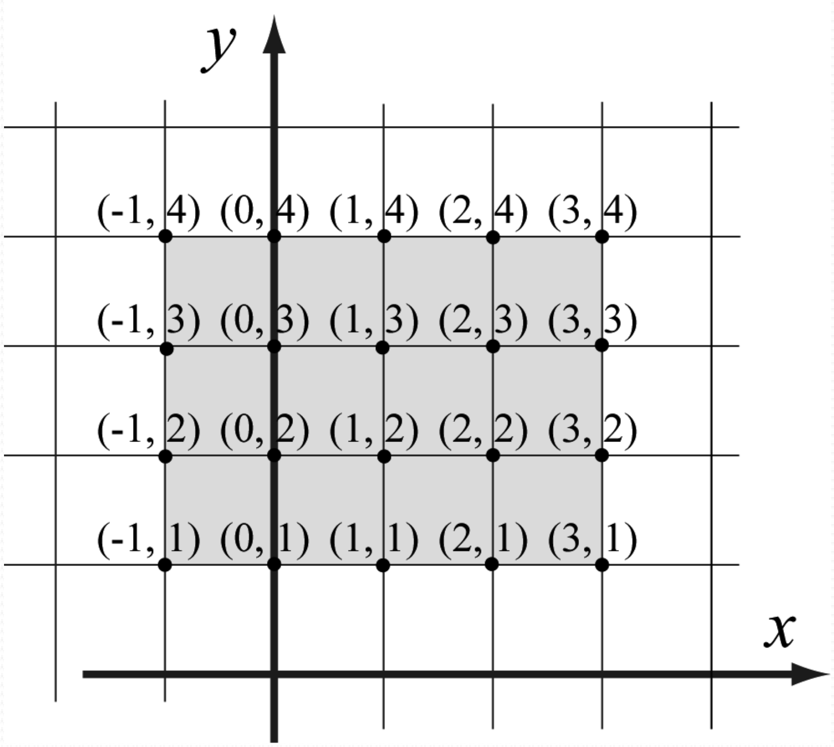
- x and y are independent variables, z is a dependent variable
- A *mesh plot* connects values of z with lines to form the outline of a surface
- A *surface plot* connects lines in a mesh plot with planes to show a solid representation of the surface

Three steps to making mesh or surface plot

1. Create grid in the x-y plane that contains points you're interested in
2. Calculate the value of z at every point of the grid
3. Make the plot

<u>Creating a grid in the *x y* plane (Cartesian coordinates):</u>

The grid is the set of points on which you want to evaluate z. For example



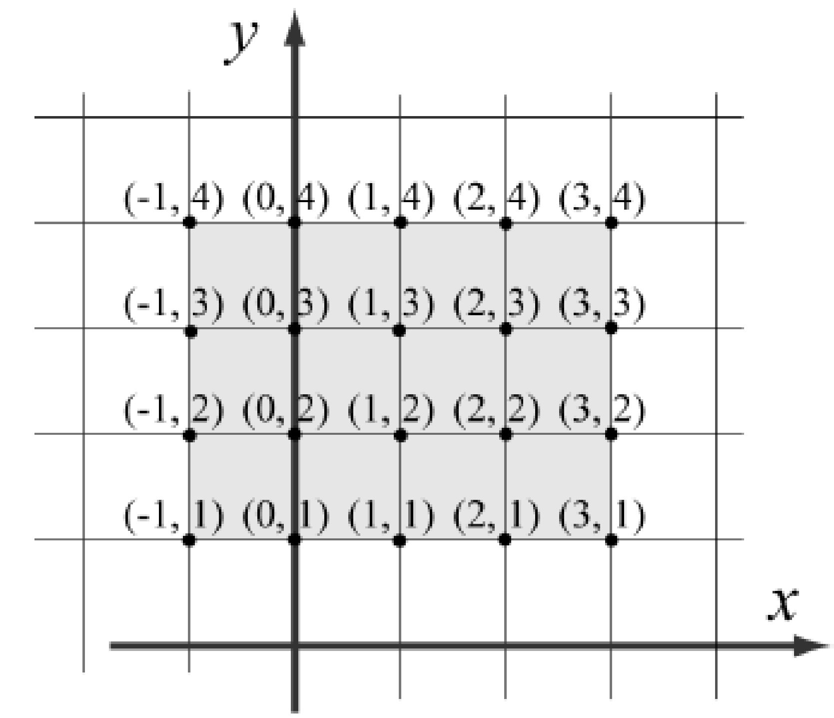
Can define the grid by using two matrices, X and Y

- X has x-coordinates of all grid points
- Y has y-coordinates of all grid points

For grid shown

$$X = \begin{bmatrix} -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{bmatrix}$$

$$\mathrm{Y} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



Note that

- X is made of identical rows because each row of grid has the same x-coordinates
- Y is made of identical columns because each column of grid has same y-coordinates

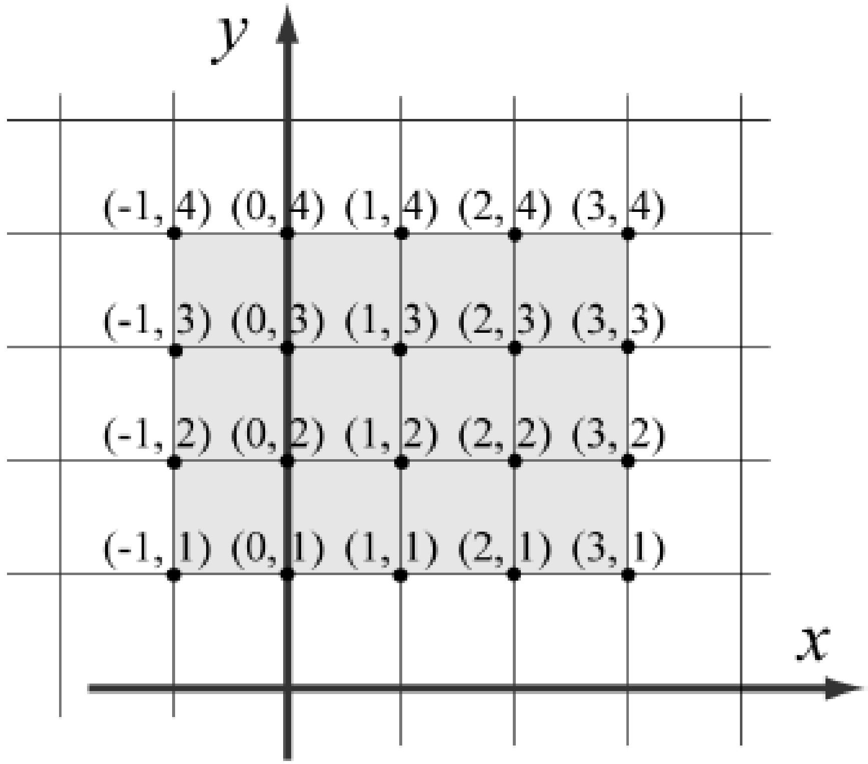
To make matrices, use MATLAB command

```
[X,Y] = meshgrid(x,y)
```

X is the matrix of the x coordinates of the grid points.
Y is the matrix of the y coordinates of the grid points.

x is a vector that divides the domain of x.
y is a vector that divides the domain of y.

```
>> x=-1:3;
>> y=1:4;
>> [X,Y]=meshgrid(x,y)
X =
    -1     0     1     2     3
    -1     0     1     2     3
    -1     0     1     2     3
    -1     0     1     2     3
Y =
     1     1     1     1     1
     2     2     2     2     2
     3     3     3     3     3
     4     4     4     4     4
```



Calculating the value of *z* at each point of the grid:

Calculate value of *z* at each point by using elementwise calculations.

- X and Y must be same dimensions
- Resulting *z* will also be same dimension

For example grid and $z = \frac{xy^2}{x^2+y^2}$

```
>> Z = X.*Y.^2 ./ (X.^2 + Y.^2)
```

```
Z =
   -0.5000         0    0.5000    0.4000    0.3000
   -0.8000         0    0.8000    1.0000    0.9231
   -0.9000         0    0.9000    1.3846    1.5000
   -0.9412         0    0.9412    1.6000    1.9200
```

<u>Making mesh and surface plots</u>:

- To make mesh plot use `mesh(X,Y,Z)`
- To make surface plot use `surf(X,Y,Z)`

EXAMPLE

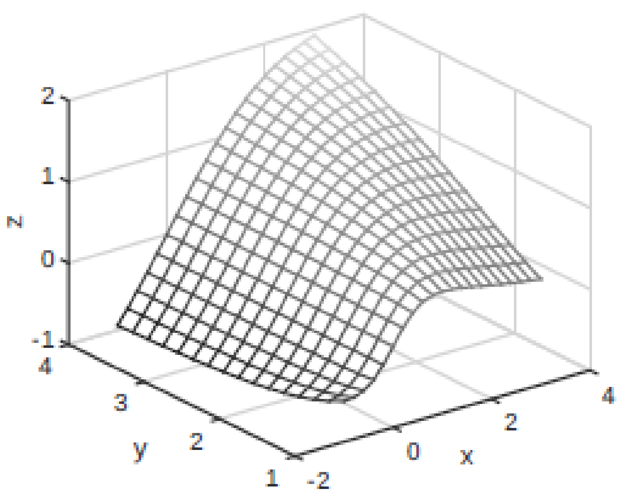
Make mesh and surface plots of $z = \frac{xy^2}{x^2+y^2}$

over domain $-1 \le x \le 3$ and $1 \le y \le 4$

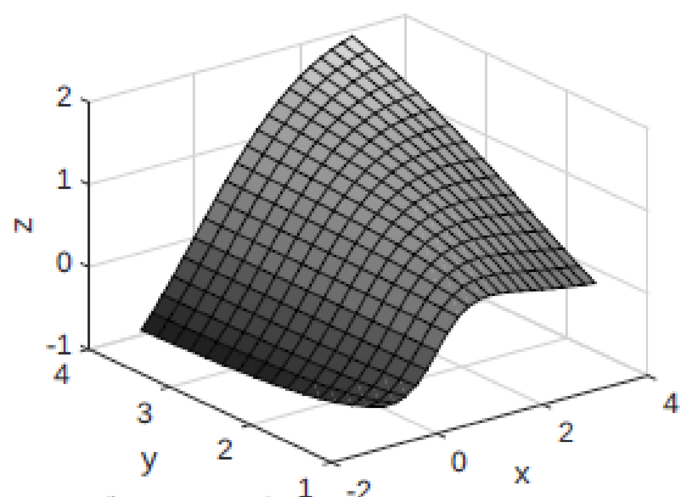
```
x=-1:0.1:3;
y=1:0.1:4;
[X,Y]=meshgrid(x,y);
Z=X.*Y.^2./(X.^2+Y.^2);
mesh(X,Y,Z)
xlabel('x'); ylabel('y'); zlabel('z')
```

Type `surf(X,Y,Z)` for surface plot.

Note that in the program above the vectors x and y have a much smaller spacing than the spacing earlier in the section. The smaller spacing creates a denser grid. The figures created by the program are:



Mesh plot



Surface plot

<u>Additional comments on the mesh command</u>:

- MATLAB colors surface plots with colors that vary with value of *z*
  - Can make color constant by using the Plot Editor in the Figure Window or by using `colormap` command. (See Help on `colormap` for details)
- By default, mesh draws a grid. Issue command `grid off` to prevent grid from appearing
- Can draw box around plot with `box on`

Can also use `mesh(Z)` and `surf(Z)`

- Command uses row indexes on the x-axis and column indexes on the y-axis

Table 10-1 in book shows lots of variations available with `mesh` and `surf` commands

Table 10-2 in book shows some commands for specialized 3-D plots.

Can get more information from Help Window or by using `help` command

Polar coordinates grid in the *x y* plane:

To make 3-D plot of function $z = f(r,\theta)$

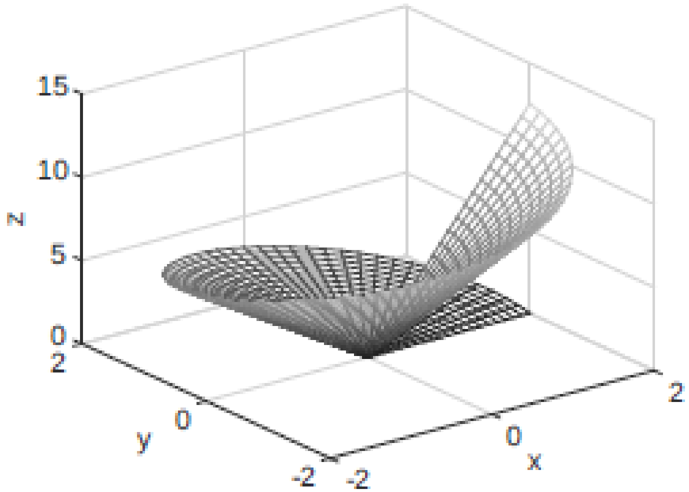
1. Make grid of values of θ and r with `meshgrid`
2. Compute value of z at each grid point
3. Convert grid with polar coordinates to grid with Cartesian coordinates using MATLAB's `pol2cart` command
4. Make 3-D plot using values of z and the Cartesian coordinates

For example, the following script creates a plot of the function $z = r\theta$ over the domain $0 \le \theta \le 360^{\circ}$ and $0 \le r \le 2$.

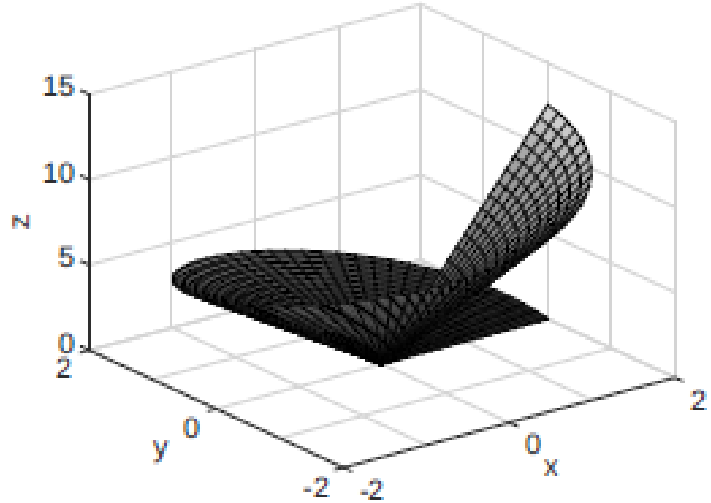
```
[th,r]=meshgrid((0:5:360)*pi/180,0:.1:2);
Z=r.*th;
[X,Y] = pol2cart(th,r);
mesh(X,Y,Z)
```

Type `surf(X,Y,Z)` for surface plot.

The figures created by the program are:



Mesh plot

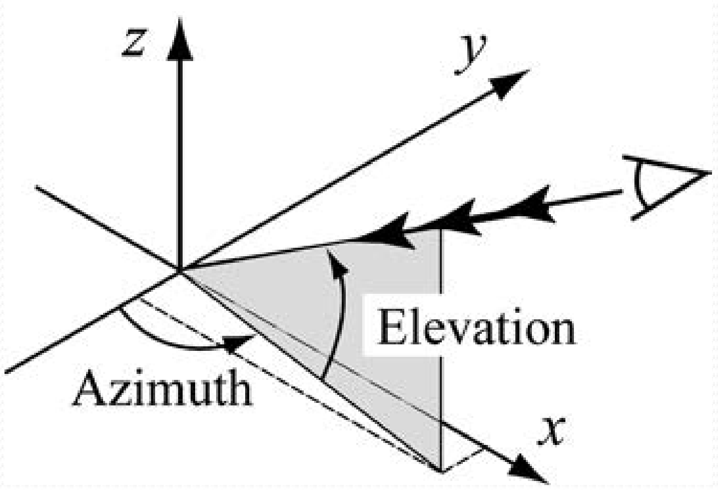


Surface plot

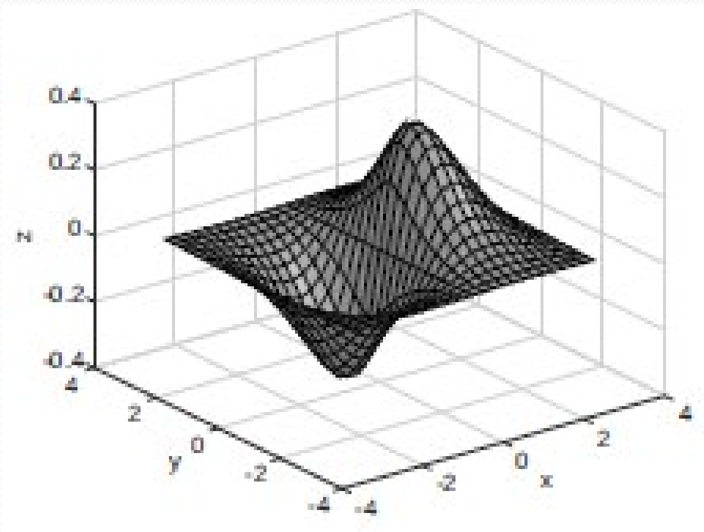
The `view` command controls direction from which you view plot. Command is

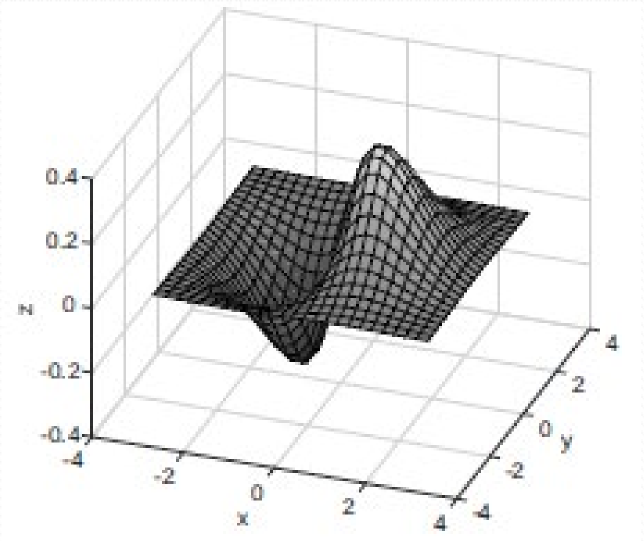
`view(az,el)` or `view([az el])`

- `az` – *azimuth*: angle (in degrees) in *x*-*y* plane measured from negative *y* axis and positive in counterclockwise direction
- `el` – *elevation*: angle of elevation (in degrees) from *x*-*y* plane. Positive in direction of positive z axis



Default view angles are *az* = -37.5° and *el* = 30°





az = -37.5° and *el* = 30°

az = 20° and *el* = 35°

Can project 3-D curve onto 2-D plane by specific settings of azimuth and elevation

Projection plane	*az* value	*el* value
x y (top view)	0	90
x z (side view)	0	0
y z (side view)	90	0

See Fig. 10-5 through Fig. 10-7 for examples of projections

`view` can also set a default view

- `view(2)` sets default to top view (projection onto *x*-*y* plane with $az = 0^o$, and $el = 90^o$
- `view(3)` sets default to standard 3-D view ($az = -37.5^o$, and $el = 30^o$)

Can also set viewing direction by selecting a point in space from which to view plot

- Command has form `view([x y z])`
  - `x`, `y`, and `z` are the coordinates of the point
  - Viewing direction is direction from specified point to origin of coordinate system
  - Viewing direction independent of distance to origin, e.g., view is same with point [6 6 6] as with point [10 10 10]
  - Set top view with [0 0 1]
  - Set side view of *x*-*z* plane from negative *y* with [0 –1 0]