25-Three Dimensional Plots

text: Chapter 9

ECEGR 101

Engineering Problem Solving with Matlab

Professor Henry Louie

Overview

- Three Dimensional Plots
- Meshgrid
- 3D Plotting Functions
- Coloring
- Shading
- Views
- Line Plots
- Movies

Three-Dimensional Plots

- MATLAB supports several varieties of three dimensional plots
- Useful for visualizing functions of two variables
  - z(x,y)

Three Dimensional Plots

- Need to create an m x n matrix of values to plot
- Example: plot z = x + .5y between [1, 5] for x and y

```
for x=1:5
    for y=1:5
        z(y,x)=x + .5*y;
    end
end
```

```
z =
```

x →

y ↓

1.5000	2.5000	3.5000	4.5000	5.5000
2.0000	3.0000	4.0000	5.0000	6.0000
2.5000	3.5000	4.5000	5.5000	6.5000
3.0000	4.0000	5.0000	6.0000	7.0000
3.5000	4.5000	5.5000	6.5000	7.5000

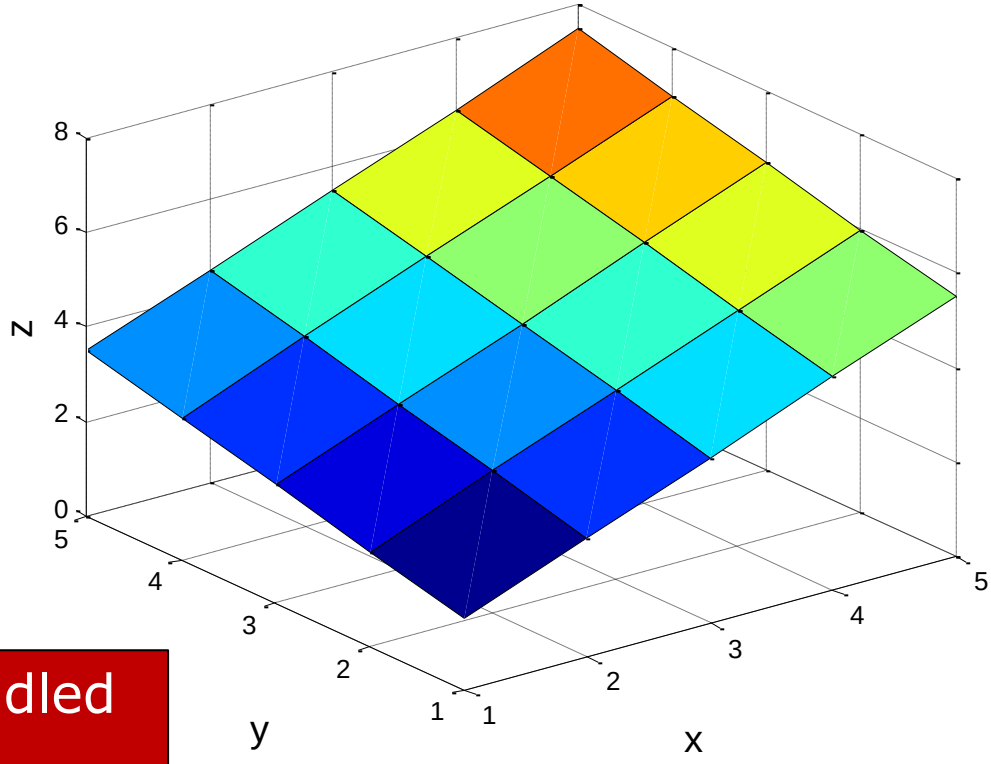
x-axis across the columns
Y-axis is down the rows

Three Dimensional Plots

- Several options of plotting z
  - Mesh
  - Surf
  - Contour

surf

```
for x=1:5
    for y=1:5
        z(y,x)=x + .5*y;
    end
end
surf(z)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
```



Labels and properties are handled just like in 2D plots

surf

- What if we want to plot the same function, but consider x in increments of 0.5 instead of 1?

```
for x=1:.5:5
    for y=1:5
        z(y,x)=x+.5*y;
    end
end
surf(z)
```

Error will result for non-integer values of x.

Surf(z), assumes that x and y coordinates start at 1 and have increments of 1 (similar to plot(y)

surf

- Use surf(x,y,z) instead of surf(z)
  - x: m x n matrix of x-axis values
  - y: m x n matrix of y-axis values
  - z: m x n matrix of z-axis values (evaluated function)

```
x =

    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
```

x increases across columns

```
y =

     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2
     3     3     3     3     3     3     3     3     3
     4     4     4     4     4     4     4     4     4
     5     5     5     5     5     5     5     5     5
```

y increases down rows

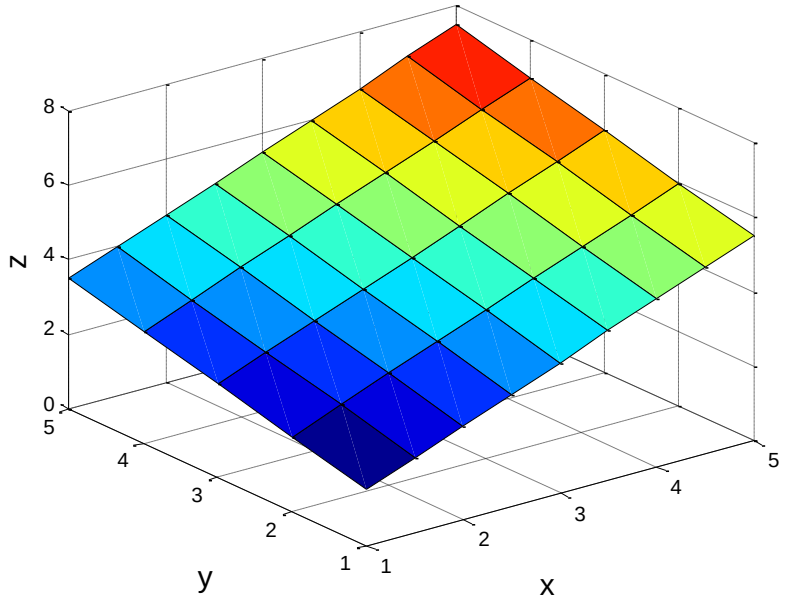
```
x =

    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000

y =

     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2
     3     3     3     3     3     3     3     3     3
     4     4     4     4     4     4     4     4     4
     5     5     5     5     5     5     5     5     5

z =

    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000
    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000
    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000
    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000    7.0000
    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000    7.0000    7.5000
```

z = x + .5y


```
x =

    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000

y =

     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2
     3     3     3     3     3     3     3     3     3
     4     4     4     4     4     4     4     4     4
     5     5     5     5     5     5     5     5     5

z =

    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000
    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000
    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000
    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000    7.0000
    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000    6.5000    7.0000    7.5000
```

z = x + .5y

```
x=[1:.5:5]
x=[x; x; x; x; x]
y=1:5;
y=[y', y', y', y',y',y',y',y',y']
z=x+.5*y
 surf(x,y,z)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
```



Also can use ‘repmat’ to form x, y matrices

Meshgrid

- Creating x and y matrices for 3D plotting can be cumbersome
- Easier method: Meshgrid
- [X,Y] = meshgrid(xgv,ygv) replicates the grid vectors xgv and ygv to produce the coordinates of a rectangular grid (X, Y). The grid vector xgv is replicated numel(ygv) times to form the columns of X. The grid vector ygv is replicated numel(xgv) times to form the rows of Y.

Meshgrid

```
xv=[1:.5:5]
yv=1:5;
[x,y]=meshgrid(xv,yv)
```

```
x =

    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000
    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000

y =

     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2
     3     3     3     3     3     3     3     3     3
     4     4     4     4     4     4     4     4     4
     5     5     5     5     5     5     5     5     5
```

Same result, reduced code.

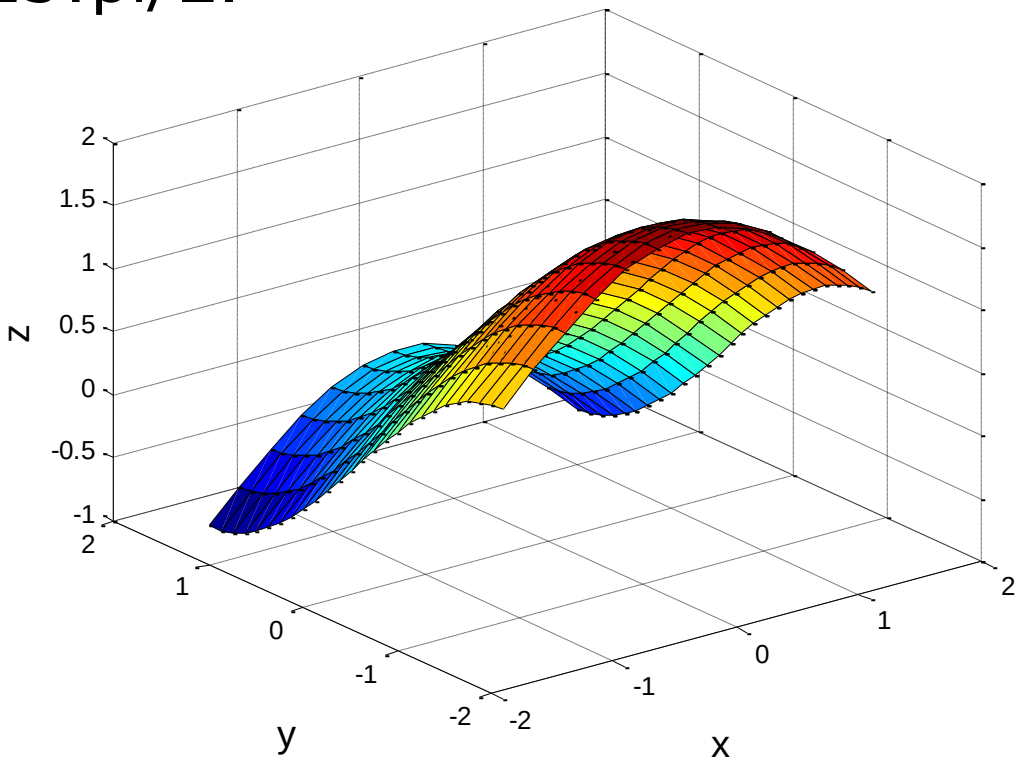
Exercise

Create a surface plot of the function z = cos(x) – sin(y) for x values of x=-pi/2:0.25:pi/2 and for y values of y = -pi/2:0.125:pi/2.

Exercise

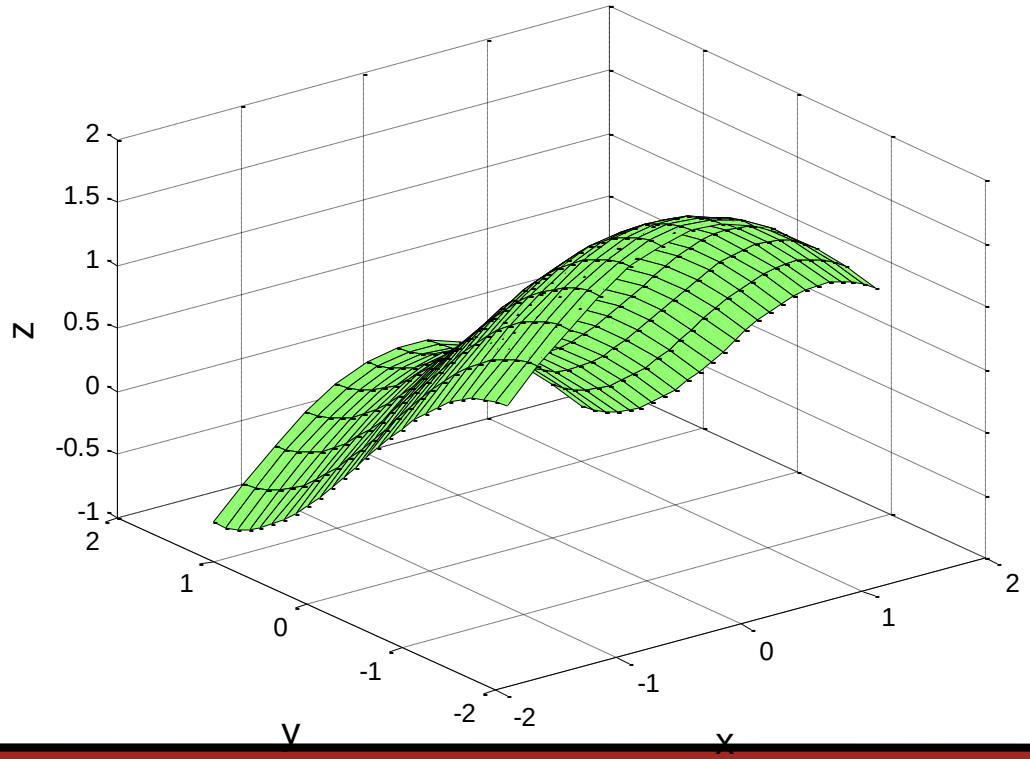
Create a surface plot of the function z = cos(x) – sin(y) for x values of x=-pi/2:0.25:pi/2 and for y values of y = -pi/2:0.125:pi/2.

```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.125:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
surf(x,y,z)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
```



Coloring

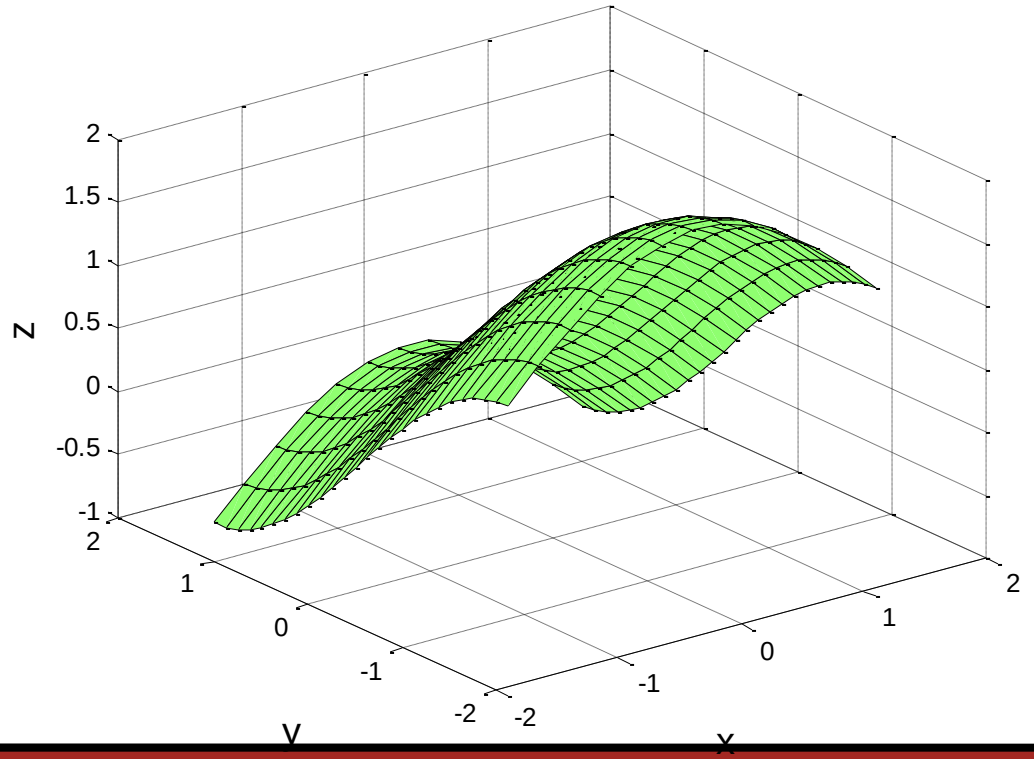
- Default is for the coloring to be associated with the z-axis value (height).
- Use surf(x,y,z,c) to control the coloring



Coloring

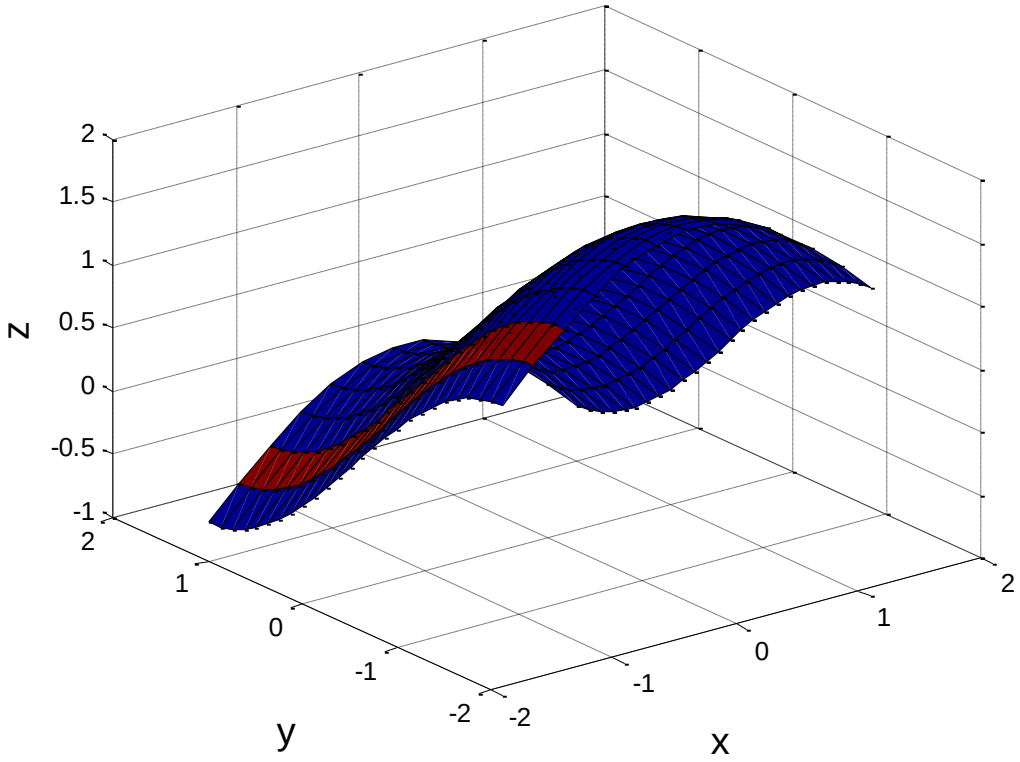
```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.125:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
c=ones(size(z,1),size(z,2))*.95
surf(x,y,z,c)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
```

Matrix c is the size of z, and the value at element (m,n), sets the color for element (m,n).



Coloring

```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.125:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
c=ones(size(z,1),size(z,2))*.95
c(:,2)=1;
surf(x,y,z,c)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
```

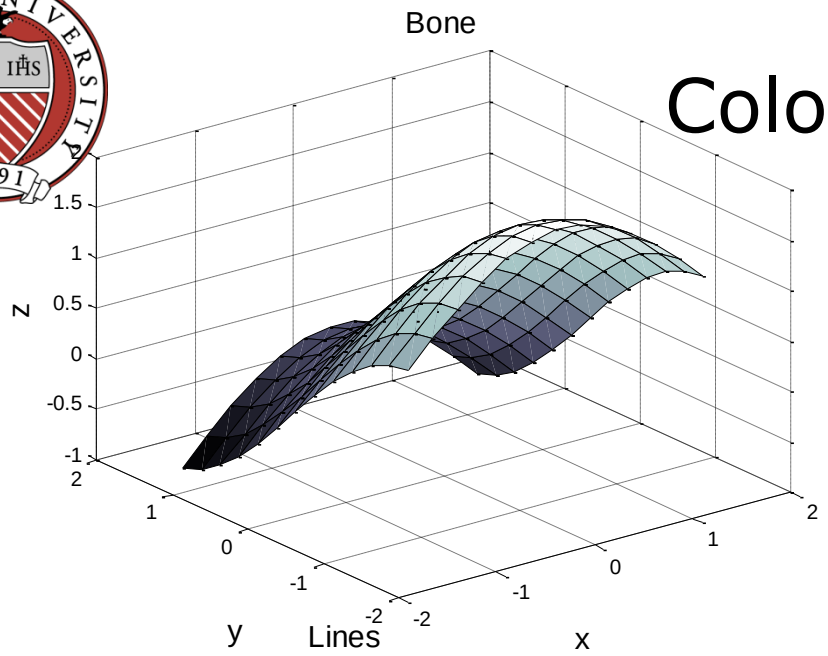


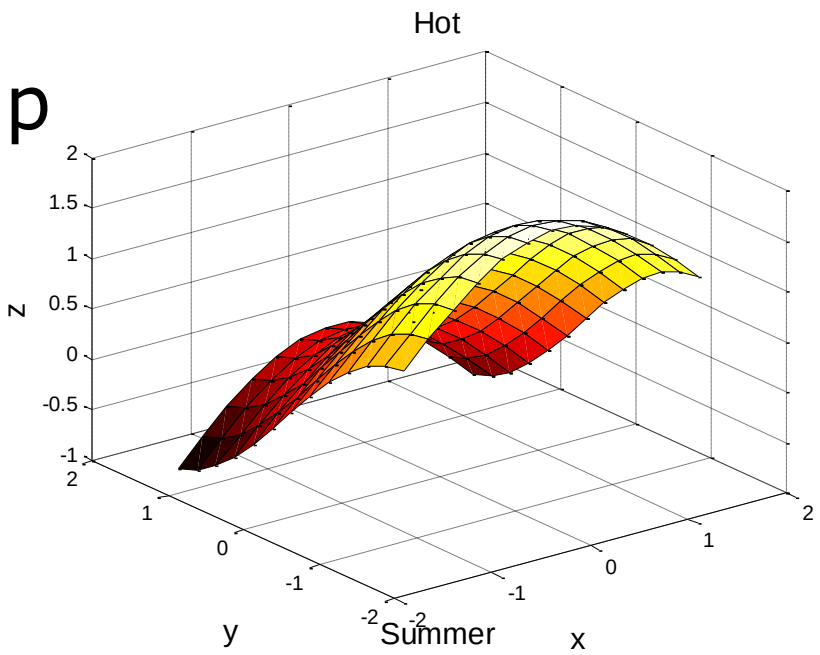
Create a “ribbon” of different color

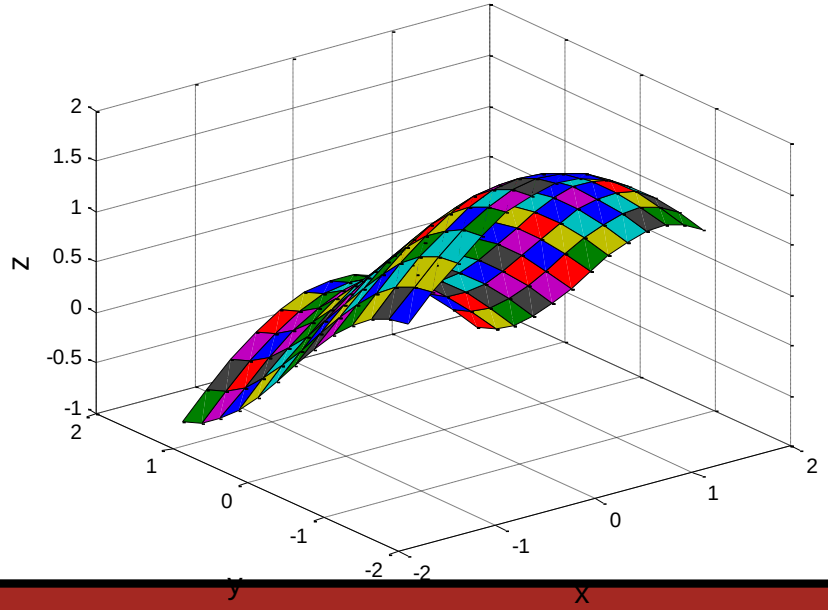
Colormap

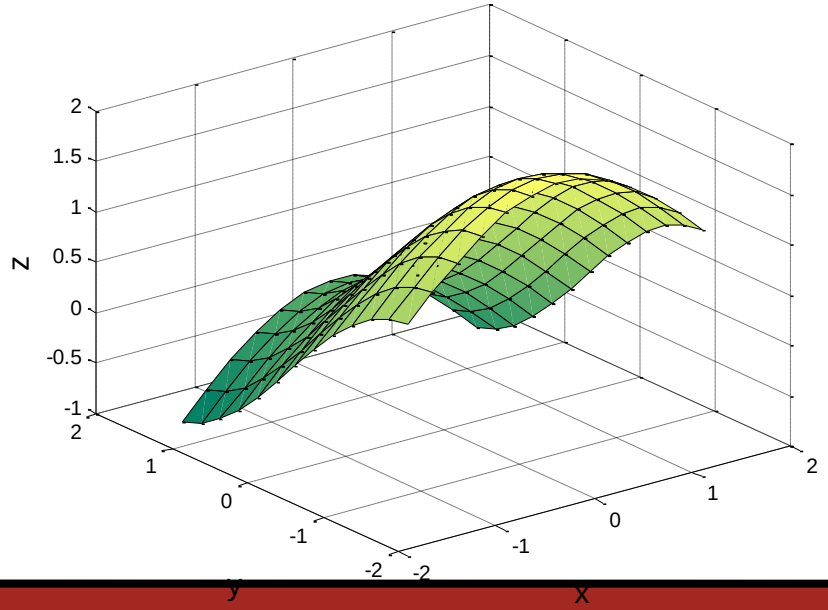
- Matlab has pre-defined colormaps
- Use colormap('name') to change color map
- Example:
  - colormap('hot')

Colormap







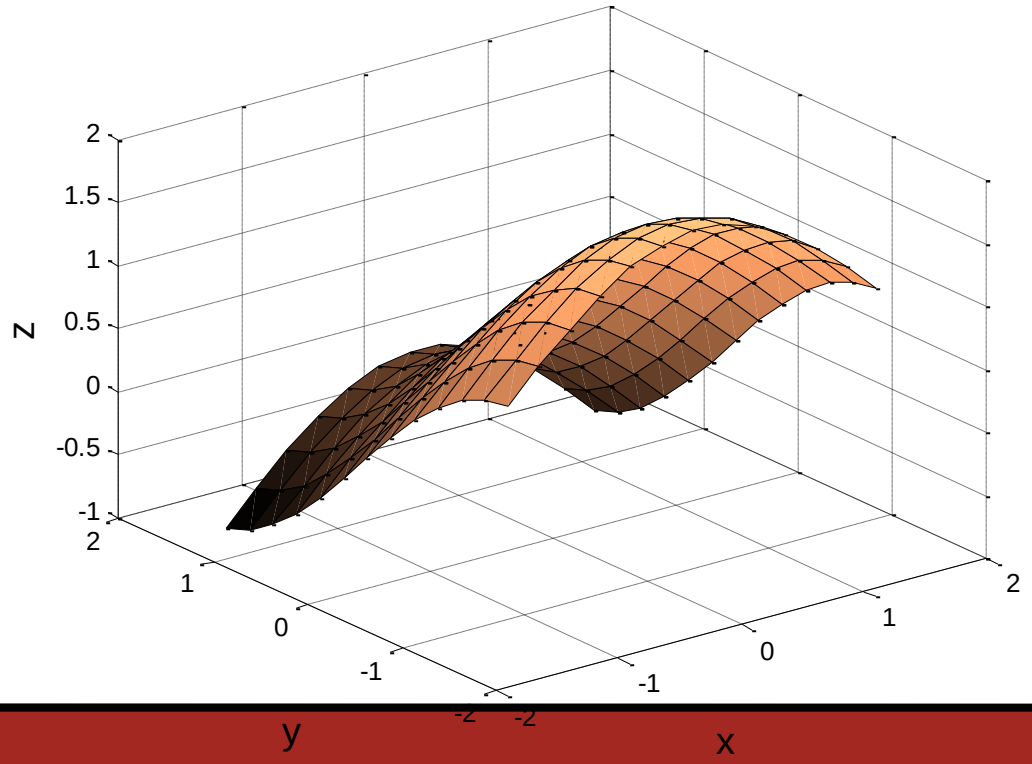


Exercise

- Create a surface plot of the function z = cos(x) – sin(y) for x values of x=-pi/2:0.25:pi/2 and for y values of y = -pi/2:0.25:pi/2. Specify the colormap to be ‘Copper’.

Exercise

- Create a surface plot of the function z = cos(x) – sin(y) for x values of x=-pi/2:0.25:pi/2 and for y values of y = -pi/2:0.25:pi/2. Specify the colormap to be 'Copper'. Copper



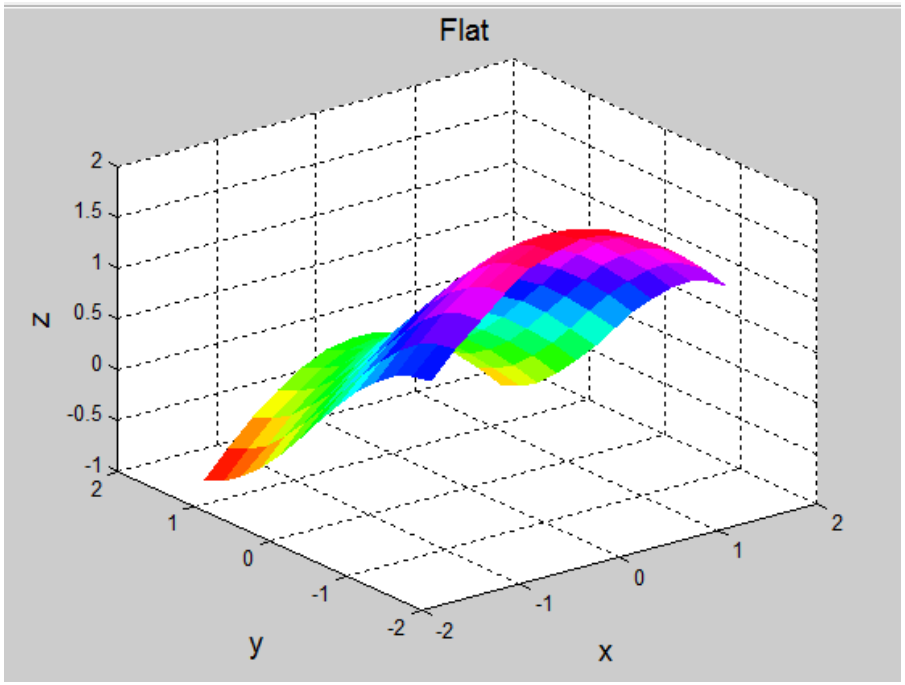
Shading

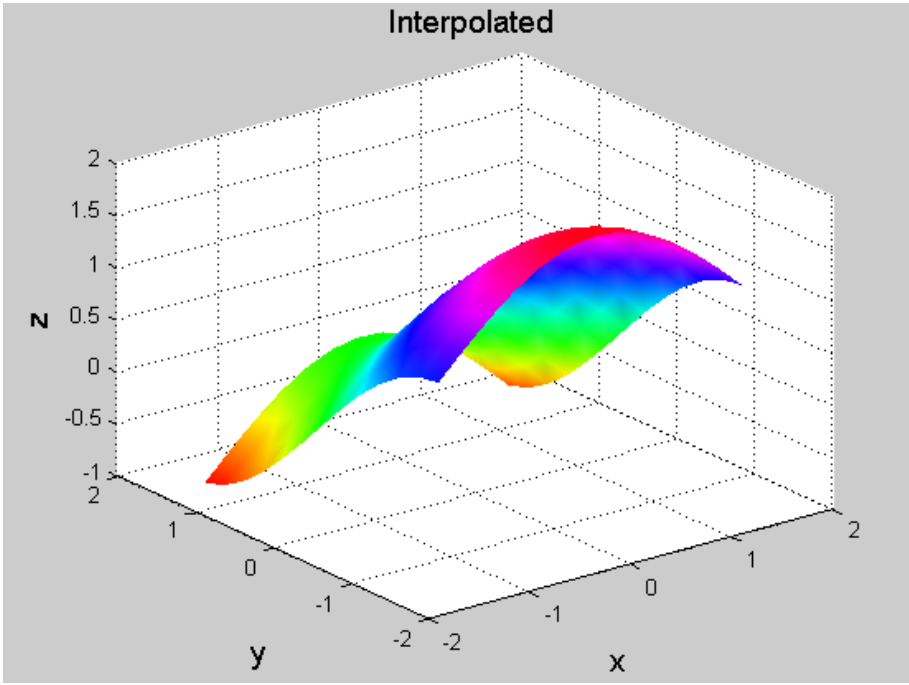
- Shading can also be controlled.
- Default is “faceted”, which draws black lines on the surface
- Other options: “flat” and “interpolated”

Include after creating surface →

```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.25:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
colormap('hsv')
surf(x,y,z)
shading flat
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
title('Flat','fontsize',14)
```

Shading

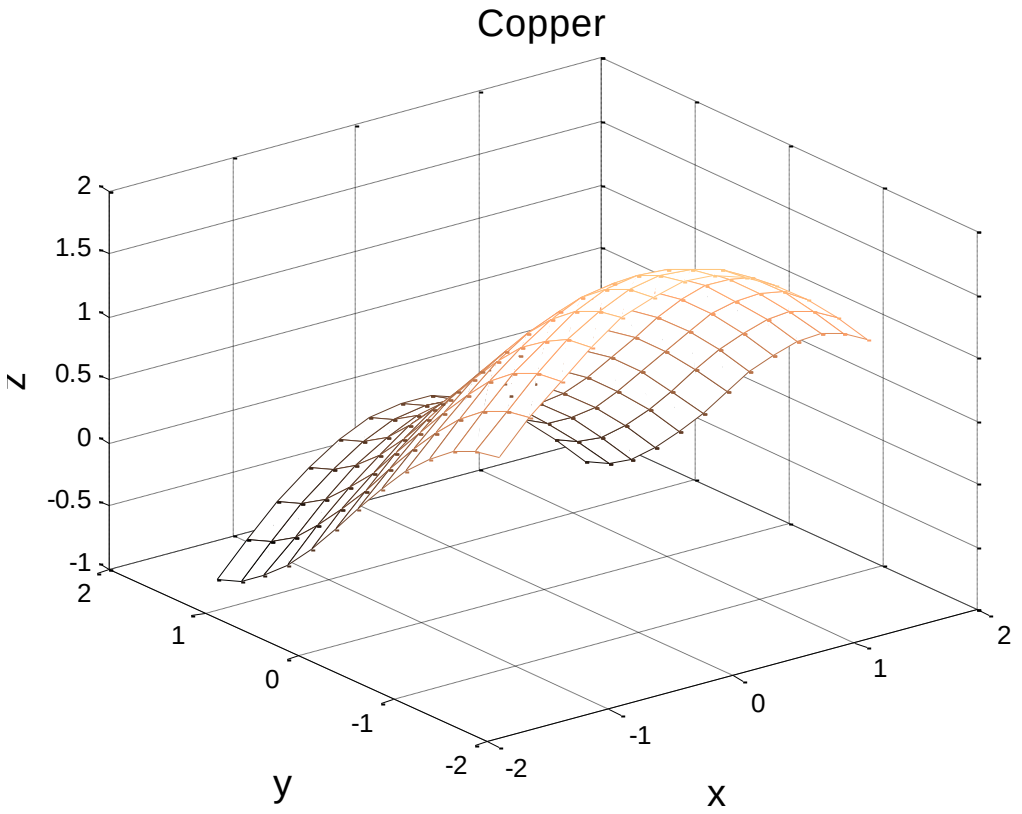




Mesh

- Mesh(x,y,z,c) creates a mesh plot. Inputs x, y and z are optional (same interpretation as in surf)

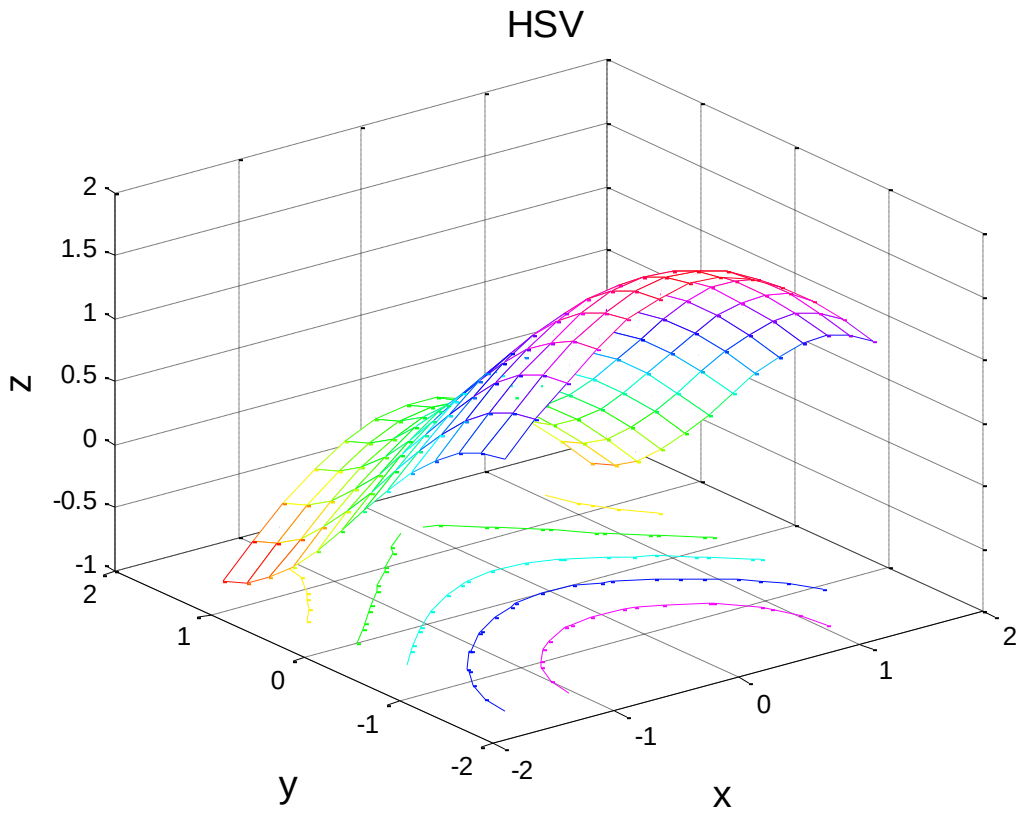
```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.25:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
colormap('Copper')
mesh(x,y,z)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
title('Copper','fontsize',14)
```



Meshc

- Meshc(x,y,z,c) Creates a mesh plot with a contour underneath

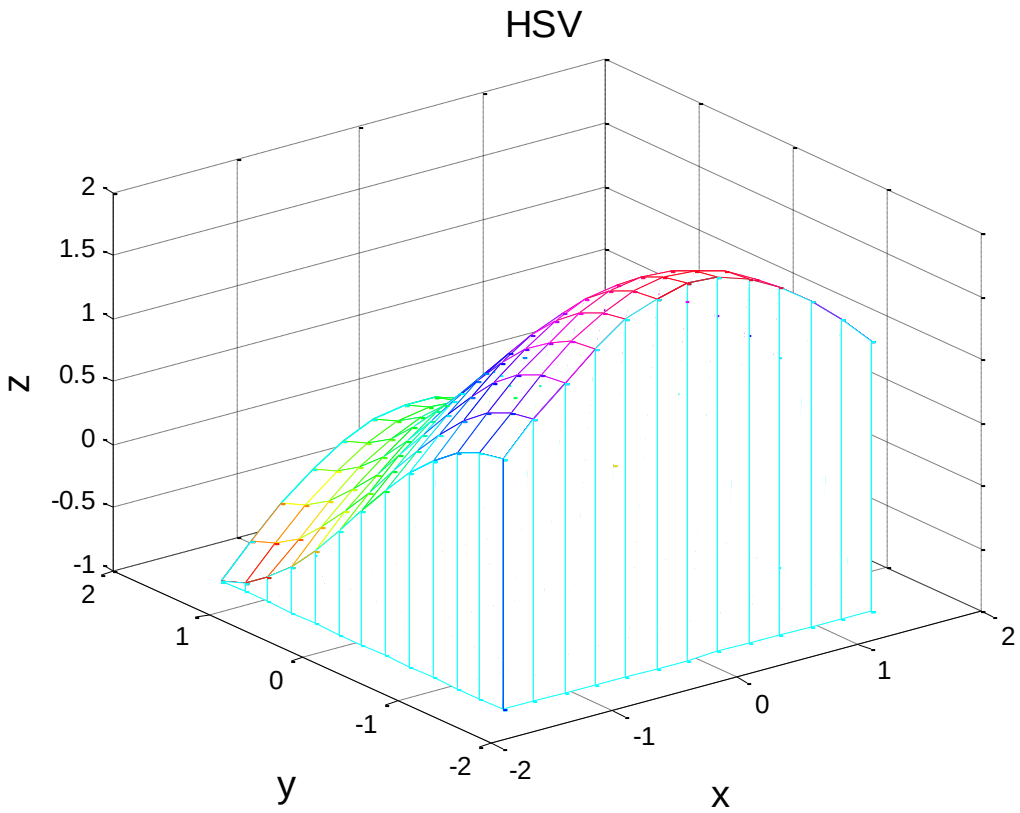
```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.25:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
colormap('hsv')
meshc(x,y,z)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
title('HSV','fontsize',14)
```



Meshz

- Meshz(x,y,z,c): Creates a mesh plot with a curtain

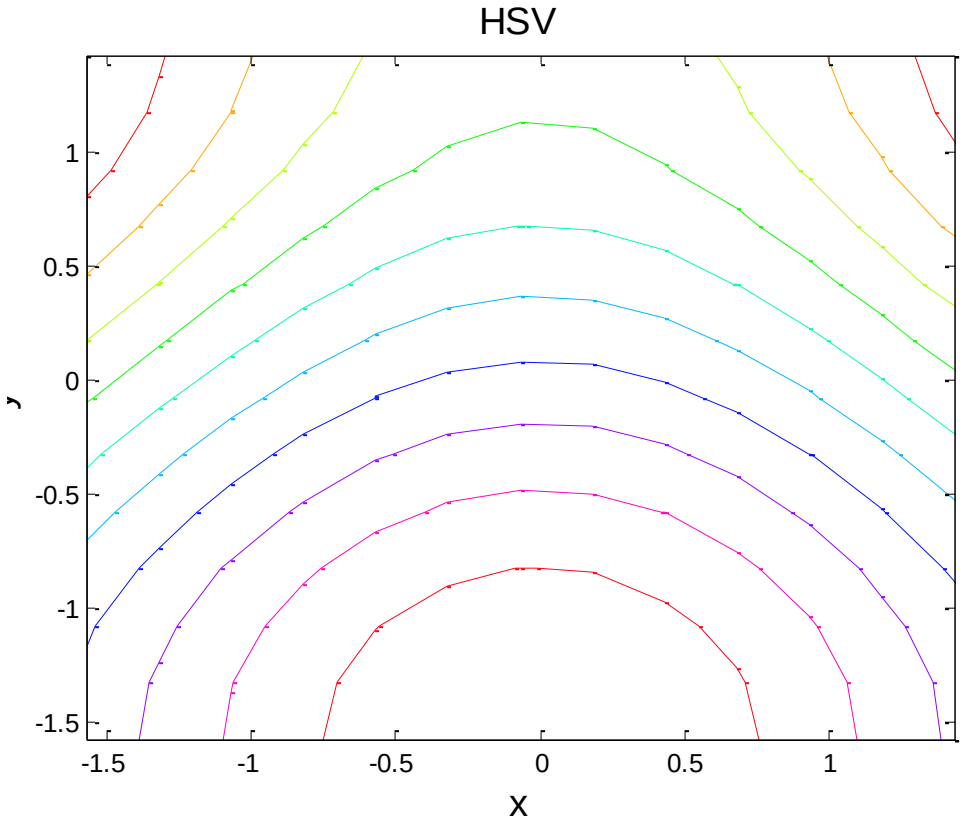
```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.25:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
colormap('hsv')
meshz(x,y,z)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
title('HSV','fontsize',14)
```



Contour

- countour(x,y,z,n): creates a contour plot with n lines

```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.25:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
colormap('hsv')
contour(x,y,z,10)
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
title('HSV','fontsize',14)
```

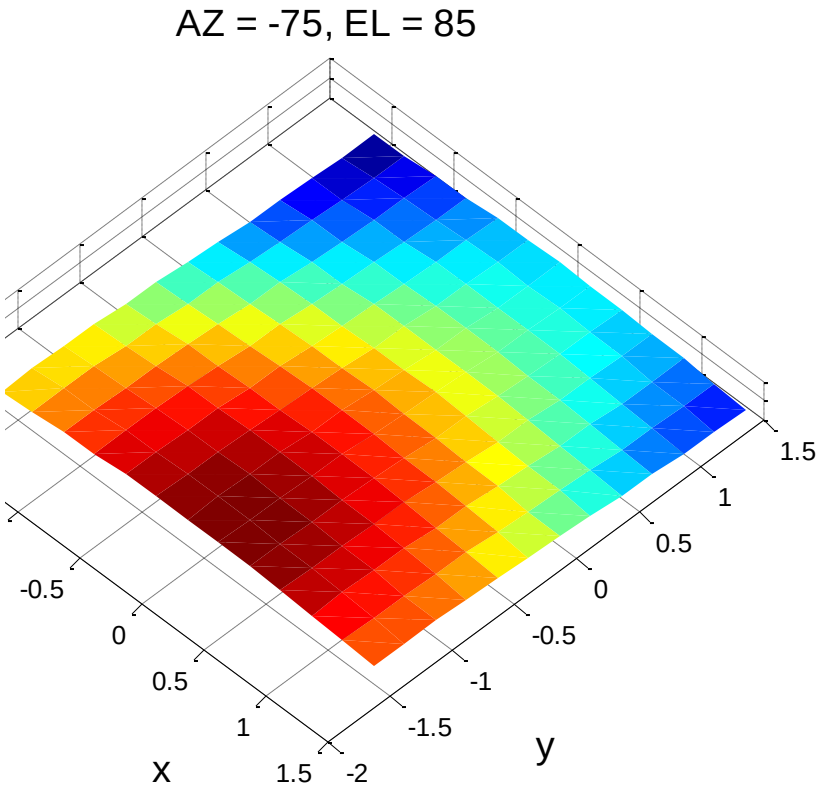


View

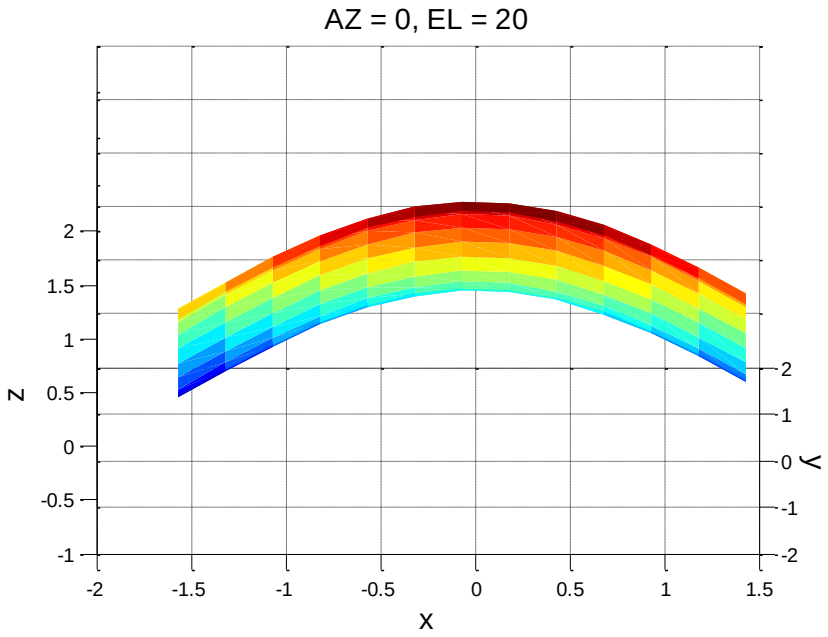
- The viewing angle can be adjusted
- view(AZ,EL) and view([AZ,EL]) set the angle of the view from which an observer sees the current 3-D plot. AZ = -37.5, EL = 30 is the default 3-D view

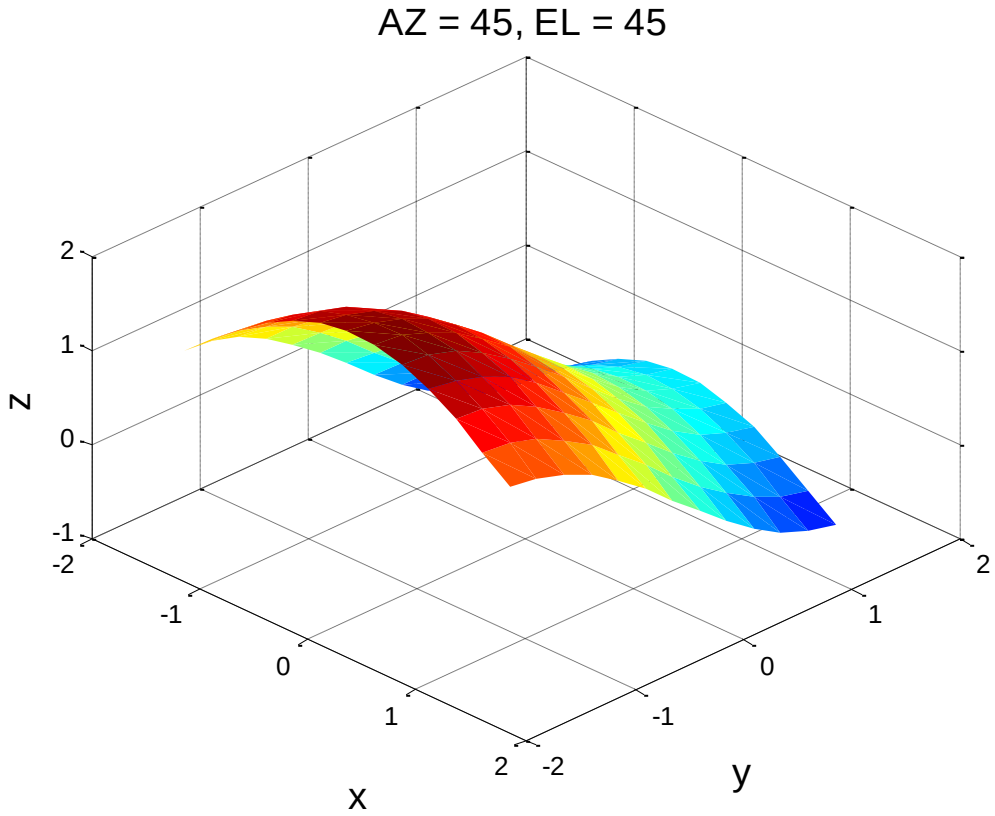
View

```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.25:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
surf(x,y,z)
view(45,85)
shading flat
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
title('AZ = -75, EL = 85','fontsize',14)
```



View

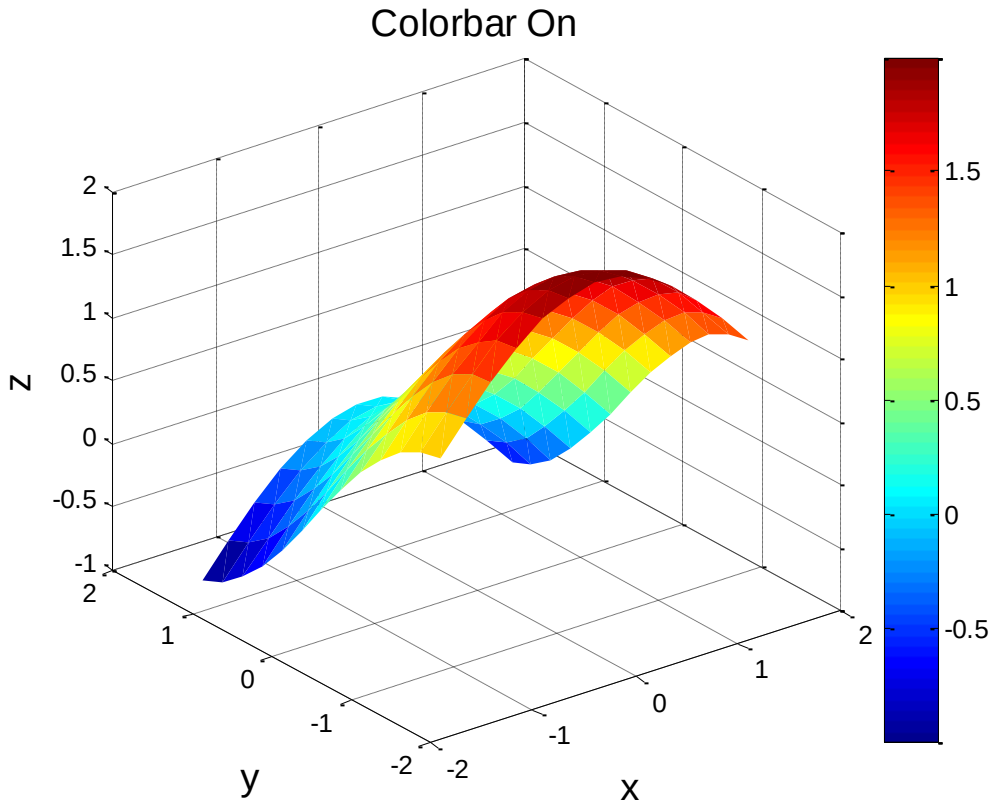

AZ = 0, EL = 20
z
x
y
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
1
0
-1
-2


AZ = 45, EL = 45
z
x
y
2
1
0
-1
-2
-1
0
1
2
-2
-1
0
1
2

Colorbar

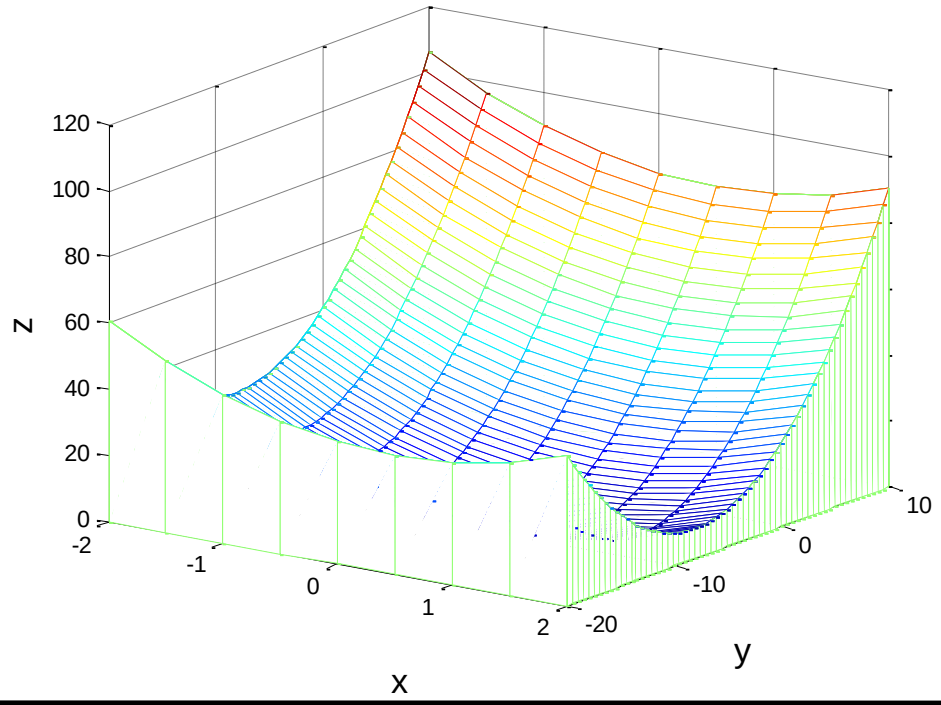
A colorbar can be added using the "colorbar" command

```
xv=[-pi/2:.25:pi/2]
yv=[-pi/2:.25:pi/2];
[x,y]=meshgrid(xv,yv)
z=cos(x) -sin(y)
surf(x,y,z)
colorbar
shading flat
xlabel('x','fontsize',14)
ylabel('y','fontsize',14)
zlabel('z','fontsize',14)
title('Colorbar On','fontsize',14)
```



Exercise

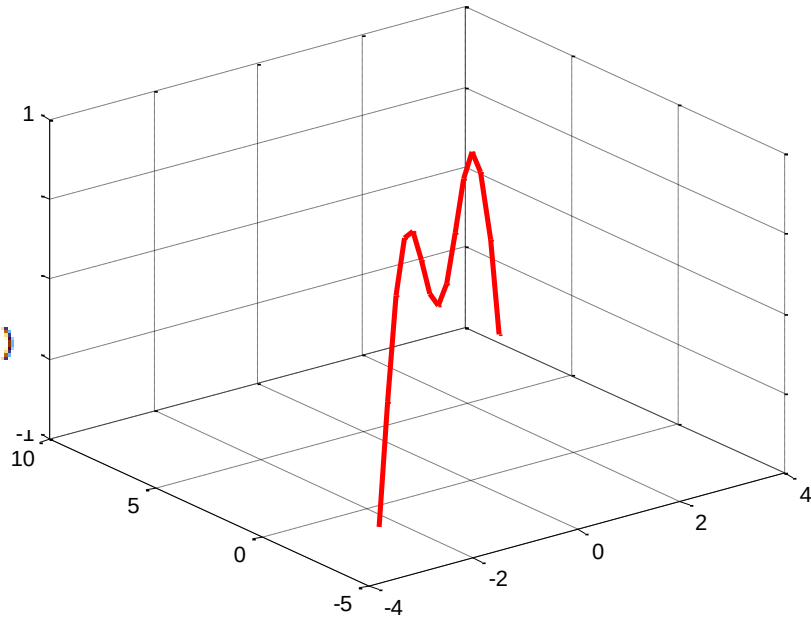
Create a 3D plot (of any kind) of the equation $z = (2x-1)^2 + (.5y +4)^2$ for x values between -2 and 2, and y values between -20 and 10. Change the view using the "view" command



Three-Dimensional Line Plots

- Plot3 is used to create a three-dimensional line plot

```
x=linspace(-3,3,15);
y=linspace(-3,3*2,15);
z=cos(x) -sin(y)
plot3(x,y,z,'color','red','linewidth',2.5)
grid on
```



Movies

- MATLAB can create movies of figures to show how a figure changes overtime
- movie(M,N,FPS) plays the movie at FPS frames per second. The default if FPS is omitted is 12 frames per second. Machines that can't achieve the specified FPS play as fast as they can.
- getframe returns a movie frame. The frame is a snapshot of the current axis

Getframe is usually used within a “for” loop

Example

```
i=0;
for t=0:10:360
    i=i+1;
    y=sind(t);
    plot(t,y,'ro','markersize',10,'MarkerFaceColor','Red')
    set(gca,'XLim',[0 360],'YLim',[-1.1 1.1]);
    grid on
    M(i)=getframe;
end
close all
movie(M,3,10)
```

← Frame is saved as the ith entry in M

Replay move three times at 10 frames per second

Exercise

See what happens when the following commands are entered into Matlab

```
i=0;
for t=0:10:360
    i=i+1;
    y=sind(t);
    plot(t,y,'ro','markersize',10,'MarkerFaceColor','Red')
    set(gca,'XLim',[0 360],'YLim',[-1.1 1.1]);
    grid on
    M(i)=getframe;
end
close all
movie(M,3,10)
```

Other commands

- See also
  - caxis
  - pcolor
  - fill3
  - ribbon
  - waterfall
  - surfl
  - bar3
  - stem3
  - pie3
  - scatter3