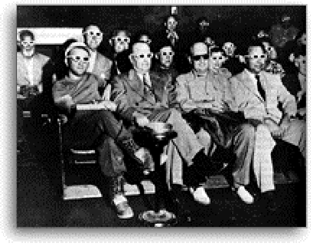
2D Routines in 3D



Outline

- Announcements
  - Wed: 2 Lectures, 8-10
    - bagels or donuts?
  - HW II--due Wed. 5PM
  - Discuss HWI on Wed
- Updated Syllabus
- Representing f(x,y)
- Lines & Surfaces in 3D
- Color & Surfaces
- Example: curtain.m

Syllabus

6. 2D routines in 3D: surfaces
7. Advanced color and lighting
8.Interpolation & unstructured grids
9. GUI’s I
10. GUI’s II
11. Volumetric visualization I
12. VV II & where to go from here

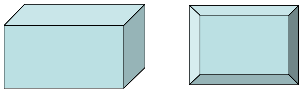
Representing f(x,y)

- Lots of choices for visualizing 2D data, often depends on properties of data

	Structure of x & y	
	Regular Grid	Irregular
2D view	pcolor	colortime
	contour	circleplot
3D view	surface	bar3
	meshgrid	stem3

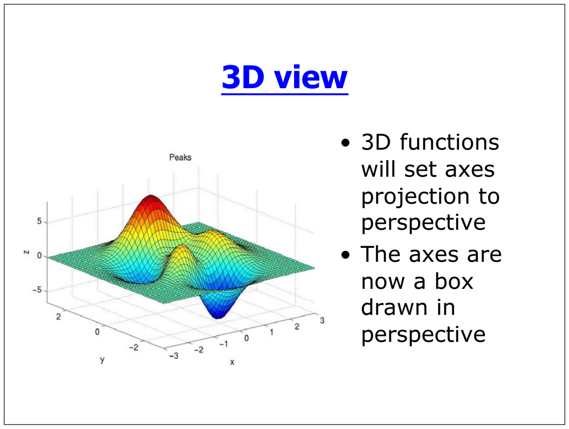
3D views

- 3D views on a computer or painting are just illusions
  - Perspective
    - lines converge towards focal point
    - Color and lighting can enhance perspective
    - Optical illusions are possible



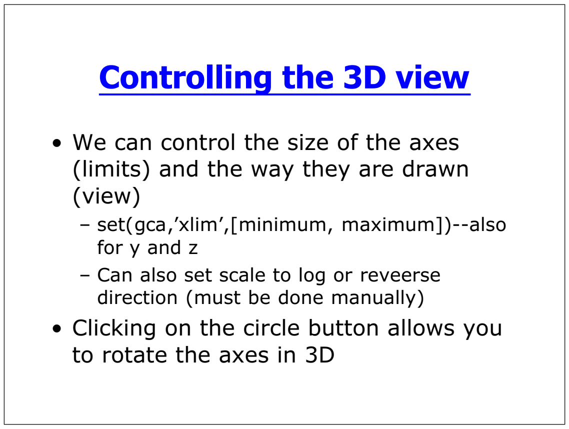
Line Objects in 3D

- h=plot(x,y);get(h,’zdata’)
  - ans=
    - Empty matri:x 1-by-0
- Both patch and line objects have a zdata field. Plot and patch set this to []
- We can plot a line in 3D using plot3(x,y,z)
  - could also set zdata field manually



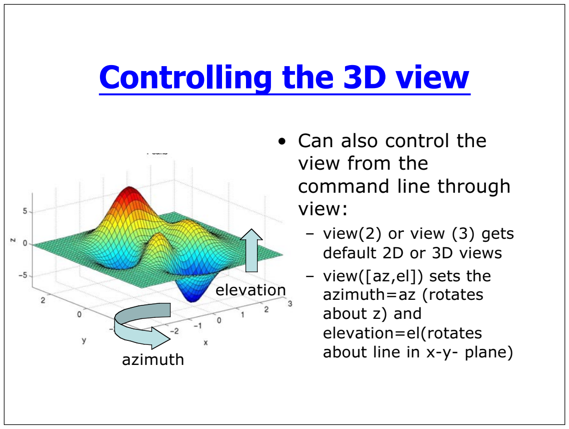
3D view

- 3D functions will set axes projection to perspective
- The axes are now a box drawn in perspective



Controlling the 3D view

- We can control the size of the axes (limits) and the way they are drawn (view)
  - set(gca,’xlim’,[minimum, maximum])--also for y and z
  - Can also set scale to log or reveerse direction (must be done manually)
- Clicking on the circle button allows you to rotate the axes in 3D



Controlling the 3D view

- Can also control the view from the command line through view:
  - view(2) or view (3) gets default 2D or 3D views
  - view([az,el]) sets the azimuth=az (rotates about z) and elevation=el(rotates about line in x-y- plane)

Surfaces in 3D

- Like lines, patch and surface objects have zdata fields.
- surf(X,Y,Z) creates a surface with vertices defined by X,Y, and Z
  - color is proportional to Z
  - facecolor=flat
- mesh(X,Y,Z) is similar, but doesn’t fill polygons
  - edgecolor=flat

Comparing surf and pcolor

- pcolor is a special form of surf

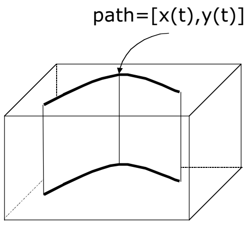
field	pcolor(x,y,Z)	surface(x,y,Z)
xdata	x	x
ydata	y	y
zdata	*[]*	*Z*
cdata	*Z*	*Z*
facecolor	‘faceted’	‘faceted’
view	*orthographic*	*perspective*

- How can we change cdata?

CDATA

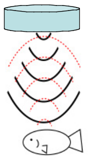
- We can use surface plots to represent 3D functions c=f(x,y,z)
  - x,y, and z define a surface
  - c is represented by color
- surf(X,Y,Z,C) where C is the same size as Z

Example: Curtain plots



- Sample c at every z along an x,y path
- We want to display c on curtain below the path

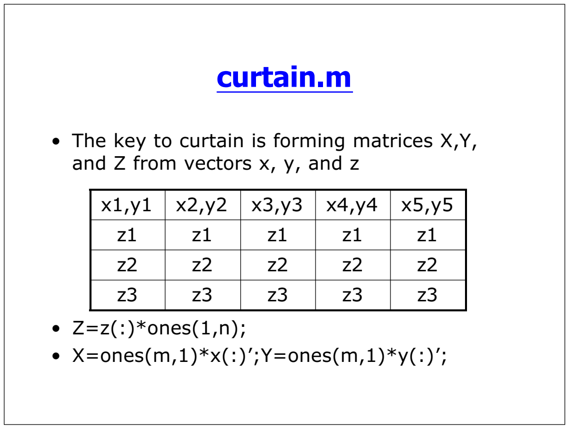
Curtain plots



- The path is the trajectory of a ship
- The ship has an acoutsic transducer that is sending sound into the water
- By measuring the volume of sound returned and when it occurs, we can get c(z)
  - this is a rough measure of the concentration of "stuff" in the water
  - depending on the frequency used, this could be shrimp, fish, or submarines

curtain.m

	name	size	desctiption
Inputs	x	1-by-n	x-locations
	y	1-by-n	y-locations
	z	m-by-1	depth bins
	BS	m-by-n	data
Outputs	h	scalar	handle to the curtain



curtain.m

- The key to curtain is forming matrices X,Y, and Z from vectors x, y, and z

x1,y1	x2,y2	x3,y3	x4,y4	x5,y5
z1	z1	z1	z1	z1
z2	z2	z2	z2	z2
z3	z3	z3	z3	z3

- Z=z(:)*ones(1,n);
- X=ones(m,1)*x(:)’;Y=ones(m,1)*y(:)’;