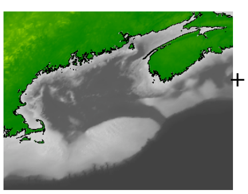
Outline

- Announcements
  - HW II due today
  - HW III available shortly
    - Remember, send me ideas by Wed.
- Syllabus
- Case Study
- Task 1: Animation

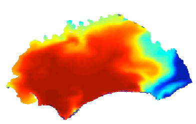
Case Study

- Each frame in animation contains



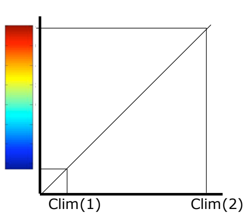
map

+



field

Colormaps



cdata

- Matlab colormaps are m-by-3 matrices, where each row is an RGB vector
- When a color property (face or edge) is set to flat or interp, Matlab will determine the color using Cdata, Clim, and the colormap

Colormaps

- Built in colormaps (help graph3d)
  - map=copper(N);--gets copper colormap with N rows
  - map=colormap--gets current colormap (default is jet)
  - colormap(map);--sets colormap to map
    - map could be a built-in colormap (copper)
- Colormap is a property of the figure, not the axes
  - This means that we can have only one colormap per figure

Creating New Colormaps

- Matlab colormaps are usually adequate, but will need to create your own if:
  - You need more than one map/figure
  - You don't like Matlab's

Creating New Colormaps

- Simplest approach is modify Matlab's
  - map=colormap(gray);map=flipud(map);
    - map will go from black to white rather than white to black
  - brighten lets you "brighten" or "darken" current colormap
- Create your own with interp1
  - v=[1 3 4]'; col=[0.5 0.5 0.5; .75 0 0; 1 1 0];
  - map=interp1(v,col,linspace(1,4,64)', 'cubic');

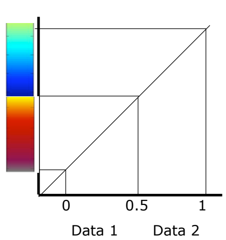
Making the Map

- have matrix z that defines elevation(depth)
  - surf(lon,lat,z) makes 3D surface
- we want to make land use green-yellow colormap
- we want ocean to use gray colormap
- Problem:
  - land ranges from 0-10m,
  - ocean ranges from -4000-0m
  - implies gray portion of colormap should be 40x green!

Making the Map

- An easier solution
  - concatenate colormaps, then
  - rescale ocean to [-1 0]
  - rescale land to [0 1]
  - store rescaled values in matrix c (color)
  - surf(lon,lat,z,c);
    - sets "zdata" field to z
    - sets "cdata" field to c
  - colormap(map);
  - caxis([-1 1]);

Multiple Colormaps



- Working with multiple colormaps gets very complicated
  - requires lots of handle graphics work
- Tips & Things to remember
  - Single Clim-space, so pick something simple [0 1],[-2 1]
  - Transform actual clims to this space

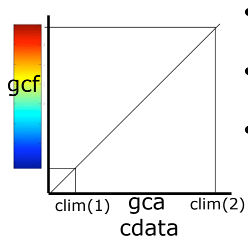
Adding the field

- field is a vector C(:,j) on a triangular mesh
- mesh is defined by arrays
  - nodll(k,:)--[#, lon, lat]
  - ele(k,:)--[#,nod1,nod2,nod3]
- h=plotonmesh(ele,nodll,C(:,j));
  - plots field using patch
  - similar to trimesh

Adding the field

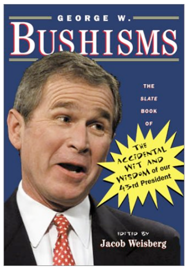
- map as clims [-1 1]
- make color of field [1 2]
  - cd=get(h,'cdata');
  - cd=(cd-min(cd(:)))/(max(cd(:))-min(cd(:)))+1;;
  - set(h,'cdata',cd);
- add default map (jet) to colormap
  - map2=[map;jet(256)];

Summary



- Figure has a colormap
- Axes defines colorlimits
- Surfaces and patches have cdata fields

Opacity



Def. 1: Quality of a body that makes it impervious to light
Def. 2: Obscurity of sense: UNINTELLIGIBLENESS
Def. 3: The quality or state of being mentally obtuse.

Misunderstimated?
Sublimable?
Hopefuller?

"I know how hard it is for you to put food on your family."

"I know the human being and fish can coexist peacefully."

Controlling Opacity

- Opacity is controlled in a similar way to color
  - Uses "Alpha" fields
  - An alpha is a number between 0 and 1
    - 0==transparent, 1==opaque

Controlling Opacity

	Colors		Alphas	
Object	Property	Options	Property	Options
Figure	ColorMap	Matrix of rgb values (jet)	AlphaMap	Vector of alphas (linspace(0,1,64))
Axes	Clim	Controls mapping of Cdata values to colors	Alim	Controls mapping of AlphaData (or FaceVertexAlphaData) to alphas
surface	FaceColor	none, flat, interp, or a color	FaceAlpha	Flat, interp, or an alpha (1)
	Cdata	Matrix specifying color data (for flat or interp)	AlphaData	Matrix specifying alpha data (for flat or interp)
Patch	FaceColor	Same	FaceAlpha	same
	FaceVertexColorData	Color values at vertices (taken from Cdata, if necessary)	FaceVertexAlphaData	Alpha values at vertices (no AlphaData for patches!)

Controlling Opacity

- So, for patches & surfaces we can specify opacity either
  - Directly--by setting facealpha to a value, or
  - Indirectly--by setting facealpha to flat or interp and filling AlphaData (or FaceVertexAlphaData) with data values
    - Can control the appearance by changing figure’s AlphaMap and axes’ Alim

So what?

- Why would you want to control opacity?
  - See inside closed surfaces
  - Represent another dimension of data (next example)
  - It’s cool

Making transparency useful

- Statistical interpolation techniques (like objective analysis) give you a distribution of values and an estimate of their accuracy (error variance)
- Most people will simply plot the interpolated data and ignore the error maps
- Ideally, we would incorporate error into the image so that it is easy to tell which values we believe