Outline

- Announcements
  - Homework I due TODAY. SPM by e-mail
  - Homework II on web
  - No lecture on Monday (Fall Break)
  - HW II due next Friday, HW III out then
- Patches in 2D--pcolor
- Example: NWtopex
- Survey

Drawing patches

- Lots of functions produce patches
  - bar, fill, fill3
- Patches are most flexible 2D objects
- patch is the lowest level function (followed closely by fill)
  - patch(x,y,c)--x and y specify vertex coordinates, c controls the color
    - C=color character ('r', 'b', 'g', etc.)
    - RGB vector
    - "color index"

```
0 1
x=[0 1 0]
y=[0 0 1]
c='r'
```
**Drawing patches**

- \( \text{patch}(X,Y,C) \)--Each column of X, Y, and C is a different polygon,
  - but same object!
  - X and Y must be the same size
    - Each polygon must have same number of vertices (rows)

\[
X = \begin{bmatrix} 0 & 1 & 0; \\ 0 & 1 & 0 \end{bmatrix}'
\]
\[
Y = \begin{bmatrix} 0 & 0 & 1; \\ 0 & 0 & -1 \end{bmatrix}'
\]
\[
C = 'r'
\]

**Key properties of patch objects**

- edgcolor--color of the edges
- facecolor--color inside the patch
- Both of these can be set to a specific color (or none)
- Or, we can prescribe another dimension of data at each vertex and let it control the color

**Patches and data**

- We’re moving from \( y = f(x) \) to \( z = f(x,y) \)
- Typically, x & y are defined on a grid of points:

```
x(1) x(2) x(3) x(4) x(5) x(6)
y(1) y(2) y(3) y(4) y(5) y(6)
```

\( Z(3,4) = f(x(4),y(3)) \)
Colorizing \( z \)

- A standard way of representing 2D data is to make color indicate \( z \)

\[
\begin{array}{c}
\text{min} \quad z \quad \text{max} \\
\end{array}
\]

\text{pcolor}

- \text{pcolor}(x,y,Z) will colorize \( Z \) on grid defined by \( x \) and \( y \)
  - \( Z=m\times n, \ x=1\times n, \ y=m\times 1 \)
- \text{pcolor}(X,Y,Z) will colorize \( Z \) on an irregular grid
  - \( X,Y,Z \) all \( m\times n \)
- \text{h=pcolor(…)} gets the handle.
  - The object is actually a surface object
  - surface objects are nearly identical to patches, but must be constructed from quadrilaterals (a grid)

\text{How it works}

- \text{h=pcolor(eye(3));}

  - shading('faceted') color of cell is set by lower left-hand corner
  - shading('flat') edgcolor='none'
  - shading('interp') interpolates between vertices to get color
Controlling `pcolor`

- `shading(str)` sets `facecolor` property to `str`:
  - `flat`, `faceted` or `interp`
- `colorbar` shows a colorbar
- `caxis([zmin, zmax])` controls the color limits:
  - same as `set(gca,'clim',[zmin, zmax])`
- `colormap(cmap)`—changes the colors. `help graph3d` lists the built-in colormaps:
  - we’ll learn how to “roll-your-own”

Example: NWtopex

- Since water flows down hill, sea-surface height (SSH) indicates currents
- The TOPEX/Poseidon satellite measures SSH with radar

NWtopex

- `load(NWtopex)`—loads `NWtopex.mat` which contains the following arrays:

<table>
<thead>
<tr>
<th>name</th>
<th>size</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lon</td>
<td>1-by-66</td>
<td>longitude (x)</td>
</tr>
<tr>
<td>lat</td>
<td>1-by-31</td>
<td>latitude (y)</td>
</tr>
<tr>
<td>SSH</td>
<td>31-by-66</td>
<td>SSH=z(x,y)</td>
</tr>
<tr>
<td>rkb</td>
<td>256-by-3</td>
<td>new colormap</td>
</tr>
</tbody>
</table>