Patches of Color

Outline

- Announcements
  - Homework I due TODAY. 5PM by e-mail
  - Homework II on web
  - No lecture on Friday
- Polar Exploration
- Patches in 2D--pcolor
- Example: NWtopex
- Survey

Polar Exploration

- The mystery: how can you produce objects without handles?
**Polar Exploration**

- toolbox/matlab/graph2d/polar.m
- `polar` creates a patch (white circle), lines (circular grid, spokes), and text
- the 'HandleVisibility' is set to 'off'-- making them inaccessible
- Gus Lott wins the Polar Explorer award for finding this out

**Patch Review**

- Patches are our workhorse 2D objects
- Lots of functions produce patches
- `patch` is the lowest level functioned (followed closely by `fill`)
  - `patch(x,y,c)`--x and y specify vertex coordinates, c controls the color
  - `patch(X,Y,C)`--Each column of X, Y, and C is a separate patch

```
x=[0 1 0]
y=[0 0 1]
c='r'
```

**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:

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

data lives at points (nodes)
\( 2(3,4)=f(x(4),y(3)) \)
**Colorizing z**

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

![Colorizing z Diagram]

**pcolor**

- `pcolor(x,y,Z)` will colorize Z on grid defined by x and y
  - Z=m-by-n, x=1-by-n, y=m-by-1
- `pcolor(X,Y,Z)` will colorize Z on an irregular grid
  - X, Y, and Z all m-by-n
- `h=pcolor(...)` gets the handle.
  - The object is actually a surface object
  - Surface objects are nearly identical to patches

**How it works**

- `h=pcolor(eye(3));`

```
0 0 1
2 1 0
1 0 0
```

- **shading(‘faceted’)**: color of cell is set by lower left-hand corner
- **shading(‘flat’)**: edgecolor='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 “role-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>