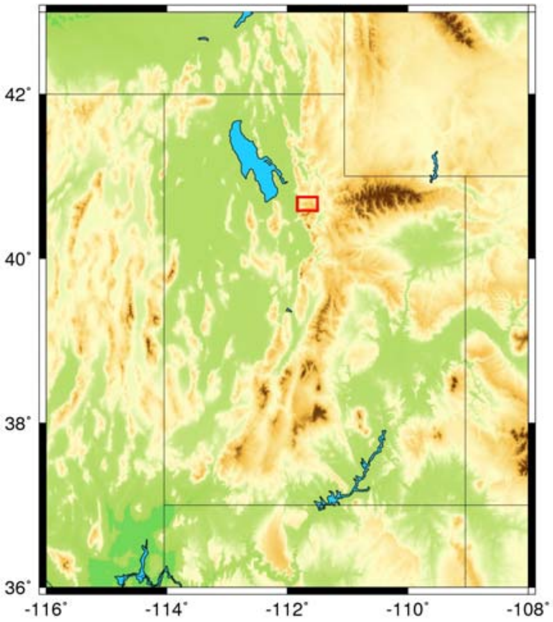
L07 – Generic Mapping Tools (GMT) - Part 3

1. Local Scale Mapping

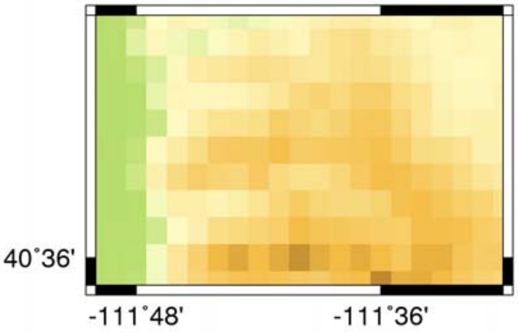
In our last lecture we showed some examples of global scale mapping. We also introduced a global elevation model Etopo1. This file has a single elevation value for each 1 minute × 1 minute grid point. This is an elevation value for grid sizes about 1.85 km × 1.85 km. This is a really fantastic grid size for plotting at global scales, or even on the statewide scale. Consider the next script which shows an example of plotting elevations for the state of Utah.

```
#!/bin/csh

makecpt -CColor_DEM.cpt -Z -T800/5000/100 >! utah.cpt

grdimage ETOPO1_Ice_g_gmt4.grd -R-116/-108/36/43 -JX5i/6i \
  -B2g10000nSeW -Cutah.cpt -V -P -K >! utah.ps
pscoast -R -JX5d/6d -Df -V -P -O -K -W2 -N2  \
  -S33/204/255 >> utah.ps
# draw an inset box
psxy -R -JX -W8/255/0/0 -O -P << eof >> utah.ps
-111.833 40.75
-111.833 40.583
-111.5   40.583
-111.5   40.75
-111.833 40.75
eof

gs -sDEVICE=x11 utah.ps
```

The execution of which results in the following plot:



Note that I use a color palette table derived form **Color_DEM** that was obtained from the ***CPT City*** website and available in this lectures accompanying material on the web site. The resulting image on the scale of the state of Utah is quite nice. But notice the red inset box which approximately covers the area between Mill Creek and Big Cottonwood Canyons. If we zoom into this space what we get looks like this:



As you can see the resolution isn't that pretty at this scale. So, what do we do? Fortunately there are a ton of good data sources for **Digital Elevation Models** (**DEM**), ranging from 90 meters × 90 meters (lateral grid size), down to 2 m × 2 m in certain locations.

Data sources

A fair amount of data is currently available online. However, it is usually on a state by state basis. For example, one can find DEM data for the state of Arizona at: http://sco.az.gov. But, what is available there compared with other states varies quite dramatically. Utah has one of the nicest repositories of DEM data I've seen yet. This is available at: http://gis.utah.gov

To get to the data click on **GIS Data & Resources** > **SGID, Utah GIS Data**

Note that here there are two important types of data resources: **(1) Vector Data**, and **(2) Raster GIS Data**. We will talk about using both of these file sources.

2. Generating netCDF files from DEM files.

We will first look at Raster GIS Data, and work through collecting elevation models and converting this to a ***.grd*** file. Just follow the instructions below which are rather recipe like.

- **Go to GIS Data & Resources > SGID, Utah GIS Data > Raster GIS Data > Elevation/Terrain Data**
- Next go to: **5 Meter Auto-Correlated Elevation Model (DEM)**
- Next go to: **Retrieve 5 meter DEM via Interactive Map**.
- Let's generate an image of the region between Mill Creek and Big Cottonwood Canyons that will look nicer than the above image. Zoom into the region around Salt Lake City, and notice that the map is broken up into squares. Each square represents a different DEM file.

Download the files:

12TVK200800.asc, 12TVK400800.asc, 12TVL200000.asc, and 12TVL400000.asc

- The ***.asc*** files we downloaded are not written in the most convenient manner for converting to a ***.grd*** file with GMT. In the material for this lecture there is a C Shell script called **catascfiles.csh**. This is a script written by myself that will combine all of the files we downloaded into one big file. The final output file is named: **temp01**, and is just a table of easting, northing, and elevation (in meters). The brunt of the work is done with a Fortran 90 code called **Lidar_ASC2XYZ.f90**. We will cover Fortran 90 later in the semester, but for now just assume this code I wrote works.

  So, let's make one big file of ***x***, ***y***, ***z*** positions by typing:

  ```
  >> ./catascfiles.csh
  ```

- Now, let's generate a grid file. Here is an example file on how to do this:

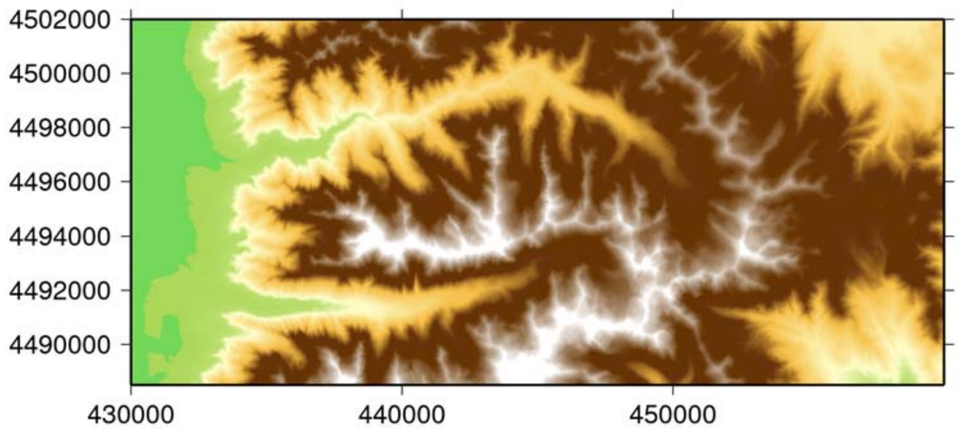
```
#!/bin/csh
#
# Example script for generating .grd files
#
#  Input file is 'temp01' written out from catascfiles.csh
#
#------------------------------------------------------------------#

#------------------------------------------------------------------#
# Set input parameters
#------------------------------------------------------------------#
set ifile = temp01      # input file output from catascfiles.csh
set elev  = f           # output elevation in meters (m) or feet (f)
set cell  = 5           # cellsize (5 meters for the 5-m DEM)
set ofile = Mt_Olympus  # Prefix naming for output files
#------------------------------------------------------------------#


#------------------------------------------------------------------#
# Determine boundaries of grid file
#------------------------------------------------------------------#
echo "Determiniming map bounds..."
minmax -C $ifile >! temp.f1

set xmin = `awk 'NR == 1 {print $1}' temp.f1`
set xmax = `awk 'NR == 1 {print $2}' temp.f1`
set ymin = `awk 'NR == 1 {print $3}' temp.f1`
set ymax = `awk 'NR == 1 {print $4}' temp.f1`

rm temp.f1
#------------------------------------------------------------------#
```

```
#------------------------------------------------------------------#
# Convert elevations to feet if switch is set
#------------------------------------------------------------------#
if ($elev == 'f') then
  echo "Converting elevations from meters to feet..."
  awk '{print $1, $2, ($3*3.2808399)}' $ifile >! temp.f2
else
  cp $ifile temp.f2
endif
#------------------------------------------------------------------#

#------------------------------------------------------------------#
# make .grd file
#------------------------------------------------------------------#
echo "Gridding file..."
xyz2grd temp.f2 -R${xmin}/${xmax}/${ymin}/${ymax}  \
 -I${cell}/${cell} -V -G${ofile}.grd
#------------------------------------------------------------------#
```

- At this point we can generate a plot using grdimage to view our elevations. The following is an example that will get us in the same vicinity as our image above.

```
#!/bin/csh

makecpt -CColor_DEM.cpt -Z -T5000/10500/100 >! mtolympus.cpt

grdimage Mt_Olympus.grd -R430000/459995/4488500/4502000 -JX6i/2.7i \
  -B10000g100000/2000g1000000000nSeW -Cmtolympus.cpt -V -P -K   \
  -X1 -Y2 >! olympus.ps

pscoast -R -JX3d/2d -Df -V -P -O -K -W2 -N2 >> olympus.ps

gs -sDEVICE=x11 olympus.ps
```

This gives us the following image:



This has significantly more detail than our image generated using Etopo1. But, we can make it look better than this.

3. Intensity Files

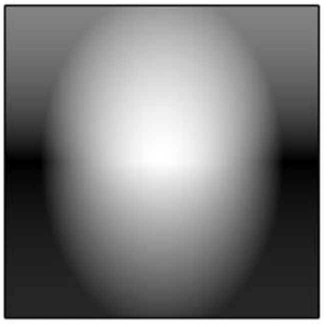
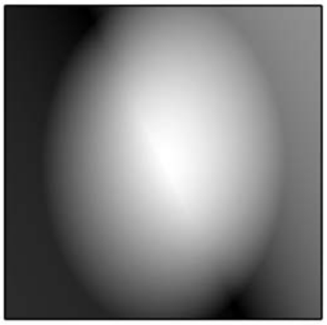
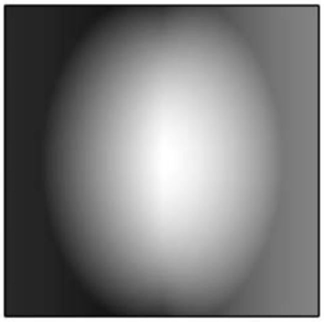
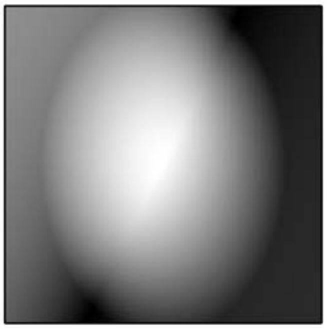
If we really want our elevation images to pop then we need to add some illumination. The easiest way to do this is with the **grdgradient** command. **Grdgradient** is used to calculate the directional derivative (or gradient) of a ***.grd*** file.

Consider our example above where we generated a grid file for the Mt. Olympus Wilderness Area. We could just add the following couple of lines to the end of our script that generated the ***.grd*** file, to also produce shading.

```
#----------------------------------------------------------------------#
# make intensity (.gradients) file
#----------------------------------------------------------------------#
echo "Calculating intensities..."
grdgradient ${ofile}.grd -A0/270 -G${ofile}.gradients -Ne0.6 -V
rm temp.f2
#----------------------------------------------------------------------#
```

What is most important here is the **–A** flag which describes which direction (azimuth in degrees clockwise from true North) do we shine our light from. To see how this works, we can just create a simple ***.grd*** file of a mound. In this case we just create a simple mound in a Cartesian space that is longer in the *y*-direction than in the *x*-direction. These data are located in the file **mound.xyz**. Using these data we can explore different gradient effects. Here is an example:

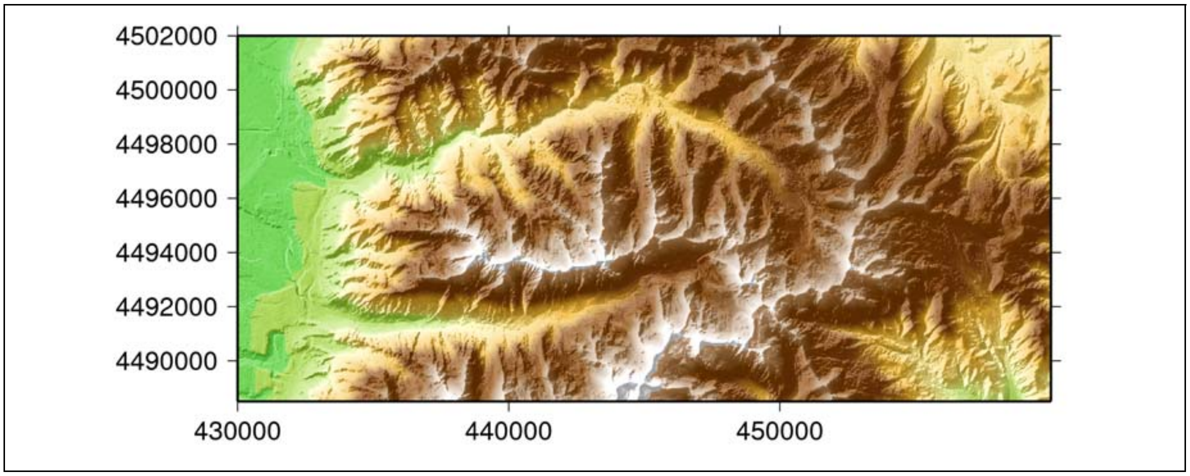
```
#!/bin/csh

xyz2grd mound.xyz -Gmound.grd -R-100/100/-100/100 -I1/1

grdgradient mound.grd -Ne0.6 -A0 -GAzi_0.grd
grdgradient mound.grd -Ne0.6 -A45 -GAzi_45.grd
grdgradient mound.grd -Ne0.6 -A90 -GAzi_90.grd
grdgradient mound.grd -Ne0.6 -A-45 -GAzi_m45.grd

makecpt -T-1000/2000/100 -Z -Cgray >! color.cpt

grdimage mound.grd -R -JX3i/3i -B0 -Ccolor.cpt -P -K -X1 -Y6 \
  -IAzi_0.grd >! mound.ps
grdimage mound.grd -R -JX -B0 -Ccolor.cpt -P -O -K -X4 \
  -IAzi_45.grd >> mound.ps
grdimage mound.grd -R -JX -B0 -Ccolor.cpt -P -O -K -X-4 -Y-4 \
  -IAzi_90.grd >> mound.ps
grdimage mound.grd -R -JX -B0 -Ccolor.cpt -P -O -X4  \
  -IAzi_m45.grd >> mound.ps

rm Azi_0.grd Azi_45.grd Azi_90.grd Azi_m45.grd mound.grd

gs -sDEVICE=x11 mound.ps
```

	
Azimuth = 0°	**Azimuth = 45°**
	
Azimuth = 90°	**Azimuth = -45°**

Finally, once we have calculated the gradients for our Mt. Olympus DEM file, we can re plot the data as follows:

```
#!/bin/csh

makecpt -CColor_DEM.cpt -Z -T5000/10500/100 >! mtolympus.cpt

grdimage Mt_Olympus.grd -R430000/459995/4488500/4502000 -JX6i/2.7i \
  -B10000g100000/2000g1000000000nSeW -Cmtolympus.cpt -V -P -K   \
  -X1 -Y2 –IMt_Olympus.gradients –Sb –E300i >! olympus.ps

pscoast -R -JX3d/2d -Df -V -P -O -K -W2 -N2 >> olympus.ps

gs -sDEVICE=x11 olympus.ps
```

The topography definitely pops more with the shading added! As a note, in this image I used put the lighting at two different azimuths (**-A0/270**). Notice in the image that the topography is dominated by East-West and North-South running ridgelines. Thus putting a light source at both 0° and 270° makes both orientations of ridgelines stand out.



4. Map Scale in GMT

Thus far we've been specifying how big to make our maps in terms of inches along either the *x*- or *y*- axis. For example, we've typed **–JX6i/2.7i** to draw our map of the Mt. Olympus Wilderness area to roughly the same scale in the *x*- and *y*- directions. There is a much easier way to accomplish this.

Everyone in here has looked at topographic quads (at least all geoscience majors). Most commonly these are plotted at the scale **1:24,000**. How do I draw a map to that scale in GMT? Simple, we just change our flag to **–Jx1:24000**. Our map above covers an area that is a little large for such a scale to fit on a sheet of paper, hence it might be better to try something like **-Jx1:200000**. All we did was use a **lower case x** instead of an **upper case X** and GMT will look for map scale instead of absolute map size. Note that this works with all map projections (e.g., -Jr as opposed to –JR).

At this point it would also be useful to add a scale bar to the plot. This is most conveniently done using the **–L** flag in either the **pscoast** or **psbasemap** command. However, this can be challenging when working in the UTM coordinate system as we have been with our DEM data. The key to getting around this is as follows:

1) **Convert map bounds from UTM coordinates to Latitude and Longitude.**

   In our plot of the Mt. Olympus region we used the UTM region:

   ```
   -R430000/459995/4488500/4502000
   ```

   We can use GMTs mapproject command to find the longitude and latitude. Above we have the minimum *x*- and *y*- location given by: 430000, 4488500. To find the **minimum** *lon*- and *lat*- location:

   ```
   echo 430000 4488500 | mapproject –Ju12T/1:1 –F –C –I
   ```

   To find the **maximum** *lon*- and *lat*- location:

```
echo 459995 4502000 | mapproject –Ju12T/1:1 –F –C –I
```

Note that we used the **–Ju** projection, which is the UTM projection. Also note, that with the UTM projection we have to specify the UTM Zone, which for our area of interest is zone **12T**. Also note that we use the **–I** flag which does the inverse transformation. That is, we want to go from UTM coordinates to longitude and latitude.

2) Plot map data with pscoast in UTM coordinate system.

Our command may look something as follows where our latitude and longitude bounds determined from step one are represented by variables **$lon1**, etc.

```
pscoast -Ju12T/1:175000 -R${lon1}/${lon2}/${lat1}/${lat2} \
-Dc -P -O -K -W2 -N1 -A100000 \
-Lf-112.24/40.84/${lat1}/5m+l1:175000+u --LABEL_FONT_SIZE=10 \
>> plot.ps
```

Note that here it is the **–L** flag that we are interested in. The most basic form of the **–L** flag looks like:

```
-Lf${lon_pos}/${lat_pos}/${lat_scale}/${distance}
```

where,

lon_pos = the longitude position on the map where you want the scale located.

lat_pos = the latitude position on the map where you want the scale located.

lat_scale = at which latitude do you want determine the length of your scale bar.

distance = how big do we make the scale.

In its most basic form, we could have just written:

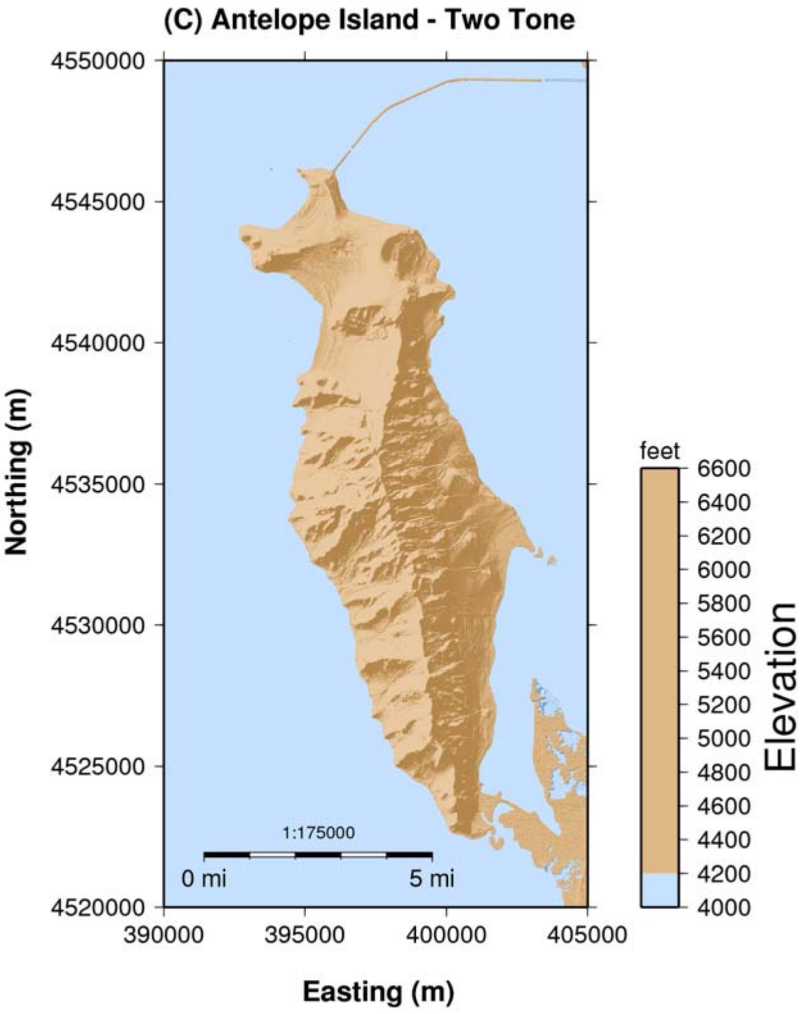
```
-Lf-112.24/40.84/40.84/5m
```

Where the **5m** indicates we want the scale bar to show 5 miles.

6. Homework

In this homework and in the next we will make a variety of plots of Antelope Island. Hence, whatever work you do for this weeks HW should be saved for future use. Download 5-m DEM data covering the Antelope Island region. Convert these elevation data into a netCDF format grid file, and generate a series of maps of Antelope Island. Make three separate plots of the island showing: (A) just colored elevations, (B) elevations colored with intensity shading, (C) bi-modal color (e.g., light brown for land masses) and blue (for water) with intensity shading. Provide a scale bar for the maps for both color (elevation in feet) and distance (in miles). Be sure to label the axes and to label each plot.

Bring a printed copy of your map to the next class. We will vote on the best plot with the winner receiving a prize. As an example, the third plot might look something like this:


(C) Antelope Island - Two Tone
4550000
4545000
4540000
4535000
4530000
4525000
4520000
Northing (m)
390000
395000
400000
405000
Easting (m)
1:175000
0 mi
5 mi
feet
6600
6400
6200
6000
5800
5600
5400
5200
5000
4800
4600
4400
4200
4000
Elevation