Lab-6: General Mapping Tools (GMT), http://gmt.soest.hawaii.edu

Apr.-9, 2010

The MODIS reflectance data R(i,j) are pixels on a grid with “i” representing pixels along- and “j” representing pixels oriented across the satellite track. We also have latitude(i,j) and longitude(i,j). You all extracted L1B data, converted from scaled integers to reflectances values, and have an ascii file with longitude, latitude, and reflectance as three columns. Our next task is to display these data graphically on a grid oriented with longitude and latitude. Added coastlines, pleasing labels, scales, and color are all desired graphic display features.

Hence, you are here tasked to create and execute a file (call it map.g) which will draw coastlines, interpolate data onto a regular lat/lon grid, and display reflectances in a chosen projection that may include Mercator, Polar Steorographic, Peters, and many more.

Formally, GMT is a set of integrated commands, just as cp, mv, nawk, or pico are. So formally, the map.g file is an executable shell script that contains a set of commands each with many “options.” It is these options that make GMT scripts extremely compact, powerful, but also abstract. I use the online manual pages all the time and you will need to use these, too:

http://www.soest.hawaii.edu/gmt/gmt/doc/gmt/html/gmt_man.html

This is definitely for the power-users, not the faint of heart. Each option starts with a “–“ (such as –G255/0/0 for a red color fill in pscoast) and is separated from prior options by a space. Lets start with an example that you will have to modify to suit your specific study area for data display:

1. Create a file called map.g with #! /bin/csh as its first line, make it executable, and add

```
pscoast –R280/310/79/83 –JS295/90/16 –Df -G –K >map.ps
```

which calls “pscoast” (google it) within a rectangle -Rxmin/xmax/ymin/ymax projected as polar steorographic (-JS) centered on longitude 295 (=65W) relative to latitude 90N for 16 inches (or was it cm?) on paper using the coastline at full resolution (-Df), painting land areas with default colors (-G, try different colors by prescribing a mixture of –Gred/blue/yellow such as a light grey produced by –G120/120/120) and places this all into a postscript file called map.ps. Add a little note, too:

```
echo ‘295 83.5 16 0 0 0 My first GMT script’ | pstext –R –JS –R –O –N >>map.ps
```

2. Have a look at the postscript file by issuing the (ghostscript) command

```
gs map.ps
```

that could also be added to the end of the map.g shell script (control-C closes the pop-up window).

3. Add the following lines to the map.g script that will use your InputFile data to generate a regular grid called grid.grd with resolution dx and dy in longitude and latitude

```
set dx = 0.1                                   # grid resolution of longitude set to 0.1 degrees
set dy = 0.05                                  # grid resolution of latitude set to 0.05 degrees (5.5 km)
blockmean InputFile –I$dx/$dy –R >output.dat   #pre-processing data, light smoothing
surface output.dat –Ggrid.grd –I$dx/$dy –R     #heavy duty grid interpolation
```

4. Display the gridded values after creating a color palette file called color.cpt to display reflectances as color representing values between 0 and 0.4 in 0.1 increments (-T option);

```
makecpt –Cjet –T0/0.4/0.1 –Z >color.cpt                # create a color palette
psscale –Ccolor.cpt –D8/10/6/0.2h –O –K >>map.ps       # show the color palette
grdimage grid.grd –Ccolor.cpt –JS –R –O –Qs >>map.ps   #show the grid
```

you will also need to modify the pstext command in #1 by adding the **–K** option. This tells the command that there will be more information to be added to the map.ps file in subsequent commands.