
L06 – Generic Mapping Tools (GMT) - Part 2

1. Plotting Fields - Grdimage

Last time we plotted a couple of maps. This was great, but there wasn't a lot of information on those maps other than where the coastlines are located. The next example shows how to add some additional information. In this case, we are plotting seismic *S*-wave velocities as derived from the tomographic inversion by Jeroen Ritsema (U. Michigan). The values of *S*-wave velocity are given (in % difference from a standard reference model – PREM) in the file [s20rts.grd](#) (this file is supplied on the webpage along with the data sets for use in the homework). Grab the files available for the homework and try the following example.

```
#!/bin/csh
# Generate a color palette table
makecpt -Ccelsius -T-5/5/.1 -Z -I >! color.cpt

# Plot the gridded image
grdimage s20rts.grd -Ccolors.cpt -R120/280/-55/40 -JX6.4i/3.8i \
  -B20g10000f10/10g10000nseW -P -E300 -K >! plot.ps

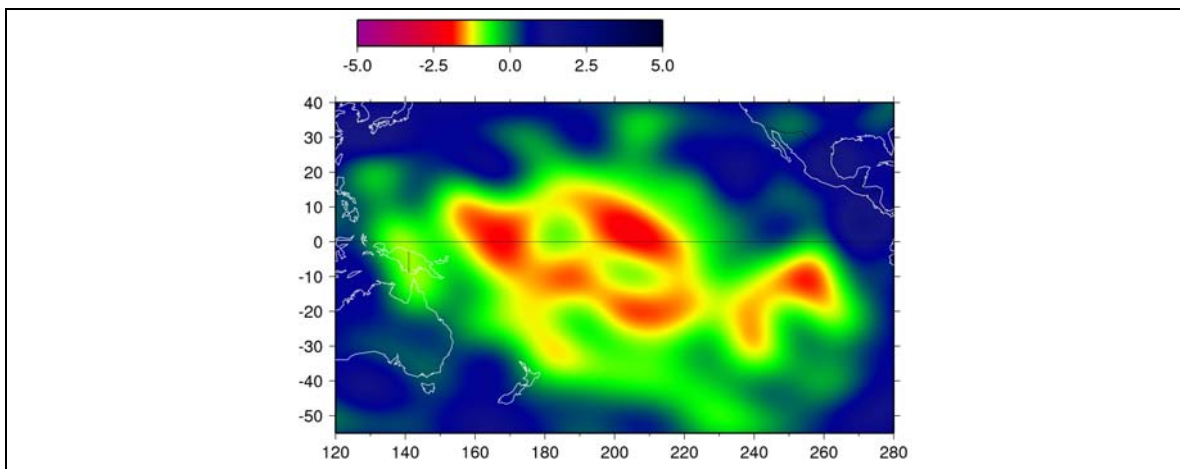
# Add the coastline information
pscoast -R -JX6.4d/3.8d -Dc -W1/2/255/255/255 -P -O -K -A10000 \
  -N1/1/255/255/255 >> plot.ps

# Add a scale bar of the colors
psscale -D2.0i/4.75i/3.5i/.3ih -O -Ccolors.cpt -B2.5 >> plot.ps

rm colors.cpt

gs -sDEVICE=x11 plot.ps
```

The resulting image you should get is shown here:




So, what did we do in creating this image? I am assuming you are expert in searching the man pages by now to determine what all of the flags mean. But, let's discuss a couple of the main points in creating such an image:

- 1) We need data in a specific type of gridded format. For this example we specifically needed our data in longitude, latitude, *S*-wave velocity format. In this case it was specified in a special kind of file called a **netCDF** or **grid** file. This is the supplied file called **s20rts.grd**. Later on in this lecture we will talk about how to create this type of file.
- 2) We needed a file that provided a mapping between seismic wave velocities and the color used to plot them. This is called a **color palette table** or **.cpt** file. Here we used the file **colors.cpt**. The next section of this lecture shows how to generate such a table.
- 3) Once we have a **.grd** and **.cpt** file we can create a colored image using the GMT command **grdimage**.
- 4) Note that we can now plot coastline information with **pscoast**, but that it has to overlay the color information or else we won't be able to see it.
- 5) Finally, we should *always* add a scale bar to our plot so we know what the colors mean. This is done with GMT's **psscale** command.


2. Color Palette Tables

Now, that we've seen an example of plotting a gridded image let's discuss how the coloring of this image is done. Note in the above example that we specified a file: **colors.cpt**. The **.cpt** extension is used to denote a color palette table. Here is an example of a cpt file designed specifically for plotting the topography of Scotland:

<pre># scotland.cpt # # by Eric Gaba -1750 121 178 222 -1500 121 178 222 -1500 132 185 227 -1250 132 185 227 -1250 141 193 234 -1000 141 193 234 -1000 150 201 240 -750 150 201 240 -750 161 210 247 -500 161 210 247 -500 172 219 251 -200 172 219 251 -200 185 227 255 -100 185 227 255 -100 200 235 255 -50 200 235 255 -50 216 242 254 0 216 242 254 0 172 208 165 50 172 208 165 50 148 191 139 100 148 191 139 100 168 198 143 200 168 198 143 200 189 204 150 400 189 204 150 400 209 215 171 600 209 215 171 600 239 235 192 800 239 235 192 800 222 214 163 1000 222 214 163 1000 202 185 130 1200 202 185 130 1200 192 154 83 1400 192 154 83 B 0 0 0 F 255 255 255 N 255 0 0</pre>	
---	--

What does this mean? Looking at the first line we have:

```
-1750 121 178 222 -1500 121 178 222
```

What this says is that we want to color elevations between -1750 to -1500 (so this is actually below sea level) with the RGB color: **R**=121, **G**=178, **B**=222. This color looks like: .

The next line is:

```
-1500 132 185 227 -1250 132 185 227
```

Which simply states that we want to color elevations between -1500 and -1250 with the color: **R**=132, **G**=185, **B**=227. So the cpt file is just a mapping between elevations and the color used to represent those elevations.

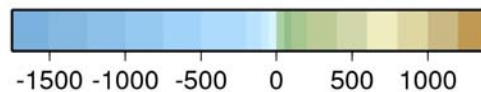
The final three lines of the cpt file are special:

```
B 0 0 0
F 255 255 255
N 255 0 0
```

Note that the color palette table in our example [scotland.cpt](#) is valid for elevations between -1750 and 1400 m. These three lines state:

- 1) **B** – Elevations are less than the defined range. In this example if the elevation occurs in the map with a value < -1750 m, then color that space as black (**R**=0, **G**=0, **B**=0).
- 2) **F** – Elevations are greater than the defined range. In this example if the elevation is > 1400 m then color that space as white (**R**=255, **G**=255, **B**=255).
- 3) **N** – If no elevation data is given for that space (i.e., elevation is given as **NaN - Not-A-Number**) then color that space as red (**R**=255, **G**=0, **B**=0).

This is an example of a categorical type of cpt. What this means is that we do not interpolate color for a given block of data. That is, in our example, for the range of elevation between -1750 and -1500 m, we use a single color. Hence, a scale bar for this type of cpt looks like discrete blocks as shown here:



However, we can also make a continuous cpt, where color is linearly interpolated between data ranges. To do this, just consider the first line of our cpt file:

```
-1750 121 178 222 -1500 121 178 222
```

Now, what would happen if we wrote:

```
-1750 121 178 222 1400 192 154 83
```

This line by itself could be a cpt file. Try it out by creating a cpt file with just that line, and plotting a color bar using psscale.

As you can see we can make cpt files by just plain hand editing. But, luckily GMT comes with a number of preloaded cpts. To see their names just type **makecpt**. The makecpt utility is great for generating cpts with custom bounds.

For example, what if I wanted to make a continuous color palette table using the GMT base file relief for topography, but having the elevations limited to the range between +/- 1000 feet. We can do this easily by:

```
>> makecpt -Crelief -Z -T-1000/1000/25 >! color.cpt
```

A couple of flags are of special note:

- 1) **-Z**: This flag says to make a continuous cpt file.
- 2) **-I**: This flag says reverse the colors

As a final note, there is an excellent resource called **CPT City** located at:

<http://soliton.vm.bytemark.co.uk/pub/cpt-city/>

There are literally hundreds of color palette tables contributed to this web site, many by professional cartographers, so it's often useful to peruse the site for interesting color schemes. Note that the colors here are given in a variety of formats, but the **cpt** format is the same as that used in GMT.

3. Generating Grid Files (xyz2grd)

In general our data sets aren't arranged in **.grd** files. But luckily GMT comes supplied with a nice command to create them: **xyz2grd**.

To make a **.grd** file one typically needs to have data ordered in a table of *x*, *y*, and *z* values. For example, if I have global data set of temperatures on the Earth's surface at a specific time, then I would want the table to look like:

lon 1	lat 1	T 1
lon 2	lat 2	T 2
lon 3	lat 3	T 3
...
lon N	lat N	T N

If I named this input file as **temperature.xyz**, and my longitude and latitude points were spaced on a $1^\circ \times 1^\circ$ interval, then I could generate a **.grd** file as follows:

```
>> xyz2grd temperature.xyz -Gtemps.grd -R0/360/-90/90 -I1/1 -V
```

Here, the **-G** flag says that we want to create a grid file named **temps.grd**. The **-I** flag tells us the spacing between *x* and *y* grid points. In our example we suggested a $1^\circ \times 1^\circ$ interval which is why this flag takes on its current values. The **-V** flag states we want xyz2grd to spit out as much

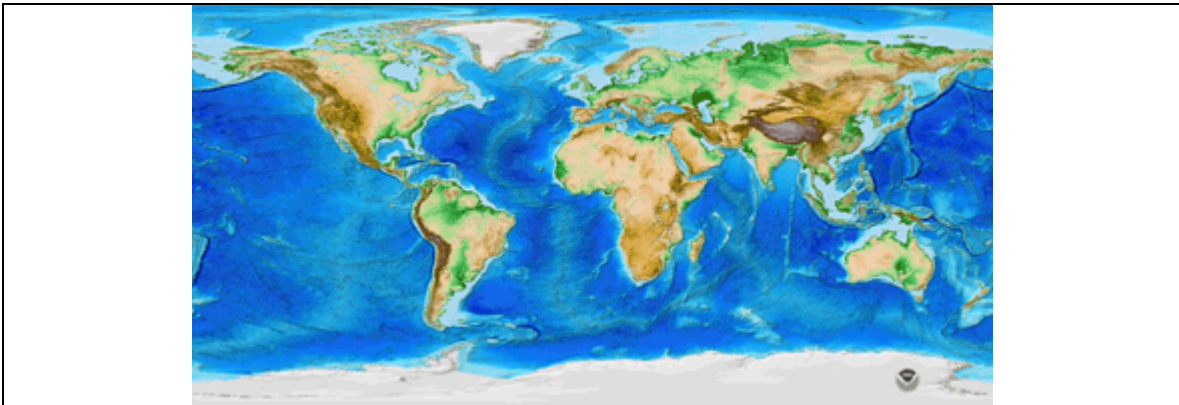
information as possible while generating the **.grd** file. This can be really useful in trying to debug problems.

4. Global Datasets

The National Oceanic and Atmospheric Administration (NOAA) manage a number of really nice global datasets, which can often be downloaded in netCDF grid file format.

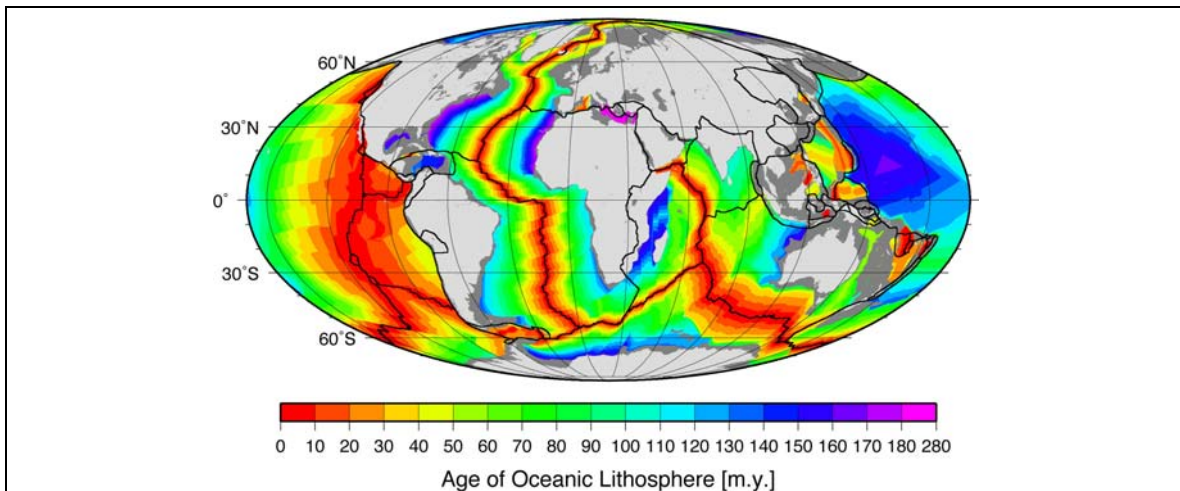
The Etopo series of surface topography and bathymetry are probably the most useful data set you will find on-line.

<http://www.ngdc.noaa.gov/mgg/global/global.html>



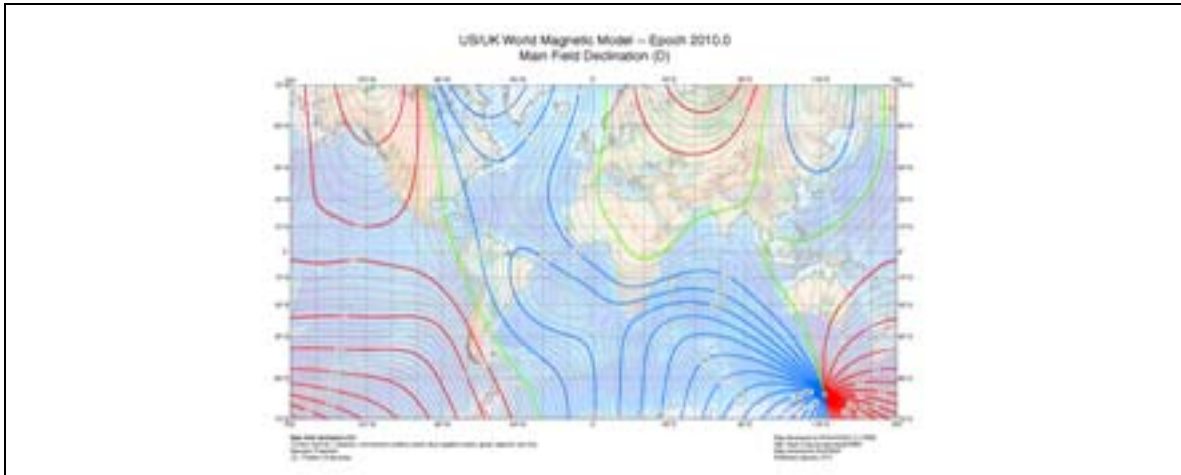
Other interesting data sets include ocean crustal ages:

http://www.ngdc.noaa.gov/mgg/ocean_age/ocean_age_2008.html



and Geomagnetic data:

<http://www.ngdc.noaa.gov/geomag/geomag.shtml>



5. What are .grd files?

In the older GMT examples, the authors always referred to their gridded data sets as **.grd** files, or grid files. However, I notice that now they more commonly refer to the files as **.nc** files. There is no change in the file format, they are still of the form called netCDF.

NetCDF (Network Common Data Form) is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.

<http://www.unidata.ucar.edu/software/netcdf/>

This data format is embraced much more in the atmospheric sciences discipline than it is in the geophysical community. But, in the geophysical community everyone still thinks they should design their own data format for every thing they do. Many of the old timers still like to cling to their data as well. But, if we were to adopt a common format for sharing data, netCDF would probably be the format.

If you really want details on the netCDF format see my other handout that is on the web page. Here is a reiteration of the important points:

NetCDF is an array based data structure for storing multidimensional data. A netCDF file is written with an ASCII header and stores the data in a binary format. The space saved by writing binary files is an obvious advantage, particularly because one does not need worry about the byte order. Any byte-swapping is automatically handled by the netCDF libraries and a netCDF binary file can thus be read on any platform. Some features associated with using the netCDF data format are as follows:

Coordinate systems: support for N -dimensional coordinate systems.

- X-coordinate (e.g., lat)
- Y-coordinate (e.g., lon)
- Z-coordinate (e.g., elevation)
- Time dimension
- Other dimensions

Variables: Support for multiple variables.

- E.g., S -wave velocity, P -wave velocity, density, stress components...

Geometry: Support for a variety of grid types (implicit or explicit).

- Regular grid (implicit)
- Irregular grid
- Points

Self-Describing: Dataset can include information defining the data it contains.

- Units (e.g., km, m/sec, gm/cm³,...)
- Comments (e.g., titles, conventions used, names of variables (e.g., P -wave velocity), names of coordinates (e.g., km/sec),...)

As a special point, netCDF comes with a utility called `ncdump` that will allow you to dump the contents of a netCDF file into an ASCII readable format. It also allows you to check just the header information. Try it:

```
>> ncdump -h netcdf_file
```

GMT also comes with a similar utility:

```
>> grdinfo netcdf_file
```

6. Homework

1) Write a C Shell script utility that one can use to visualize a color palette table. That is it should use the `psscale` command to generate a plot that one can quickly see what the `.cpt` file looks like. If we name the script `viewcpt` then we should be able to run it as:

```
>> viewcpt cptfile
```

where `cptfile` is the name of the color palette table we are interested in seeing.

2) Using the `Etopo1` grid file create a world map to emulate the first plot shown in Section 4.

3) Make a global plot showing elevation of land masses using the `Etopo1` elevations. Instead of using bathymetry for the oceans use the ocean ages and overlay the plate boundaries we used in the last homework. This problem will require clipping. Refer to the GMT Example #17 for hints on how to do this. Hint: Elevations will look best here as gray scale.

4) The webpage contains a file: [Tomography_TXBW.xyz](#). This file has longitude, latitude, and variations of S -wave velocity based on the tomographic inversion of Steve Grand (University of Texas at Austin). The velocities are given for a layer just above the core mantle boundary. Convert this file into a [.grd](#) file and plot it in two panels centered on (a) the central Pacific Ocean region and (b) Africa. It is customary to plot the lowest values (low velocities) with red colors and the highest values (high velocities) with blue colors. Make the color palette table a continuous table. Also plot world hot spot locations as triangles (to emulate volcanoes) on the maps. Be sure to include a scale bar for the colors. Is there any relation between deep mantle seismic velocities and hot spots locations?