

---

## L05 – Generic Mapping Tools (GMT) - Part 1

---

### 1. What is GMT?

It's not Greenwich Mean Time (there is a story there) According to the GMT webpage itself:

*GMT is an open source collection of ~60 tools for manipulating geographic and Cartesian data sets (including filtering, trend fitting, gridding, projecting, etc.) and producing Encapsulated PostScript File (EPS) illustrations ranging from simple x-y plots via contour maps to artificially illuminated surfaces and 3-D perspective views. GMT supports ~30 map projections and transformations and comes with support data such as GSHHS coastlines, rivers, and political boundaries.*



GMT is a set of programs that allows you to make plots, and most importantly for geoscience people it has a ton of tools for manipulating and creating maps. It is also scriptable. That is, we make plots using GMT by writing shell scripts. This is awesome because it means that once we write a shell script to create a certain type of plot we can use this script over and over to make the same kind of plot on all kinds of different data. Plots can even be made automatically. E.g., once we do some kind of data processing, we can have the code launch a GMT script and make plots for you. You can go away, let your codes run, come back after lunch, or a weekend (not if you're a grad student you should be here every day!) and have plots sitting on your desktop waiting for you. Beautiful!

Another key feature of using GMT is that you have total control over every aspect of the plot. You can make just about any style of plot you can dream up, even if no one has ever dreamed up of that style of plot before. Try doing that with Excel!

All right, for a few examples let's go to the GMT web page. It's a good site to know because it has the entire GMT manual online:

<http://gmt.soest.hawaii.edu> > Once here, click on Examples. (Actually most of these examples suck if you ask me, but it gives at least a quick idea of what can be done).

### Referenceing GMT:

If you use GMT, and most people in the geosciences do for generating figures, then it is appropriate to acknowledge the fact in your publications. Typically one says something as follows in the acknowledgements, "figures were drawn using the Generic Mapping Tools (Wessel and Smith, 1998)."

Wessel, P. and W.H.F. Smith (1998), New, improved version of the Generic Mapping Tools released, *Eos Trans. AGU*, **79**, 579.

## 2. Getting Started with GMT

OK, so plotting with GMT is definitely not like anything you are probably used to right now. So, you can't learn how to swim without jumping in the water, so let's dive in:

```
#!/bin/csh

# Example of a simple location map

pscoast -R0/360/-90/90 -JG-111/45/4.5i -Bg30 -Dc -A8000 \
  -G10/10/10 -W3/10/10/10 -P -K >! globe.ps

psxy -R -JG -W6/255/0/0 -P -O -Am << END >> globe.ps
-100 40
-100 50
-120 50
-120 40
-100 40
END

gs -sDEVICE=x11 globe.ps
```

If you type up the above script and execute it, you should get a [ghostscript](#) window that pops up with an image that looks as follows:



The image is just that of the Earth, with the continents filled in black and a red box drawn around some area of interest. It's actually a useful script and if you look at several of my publications (and those of coauthors of mine who now use the same basic format) you will see that I use a small image just like this to show my study region.

Now, that you've done that, let's take a quick look through the GMT commands we used. Note that to find out information on these commands you should always consult the man pages:

```
>> man pscoast
```

```
>> man psxy
```

**pscoast** and **psxy** are two of the most important commands you will use. Let's dissect them.

## 2.1 pscoast

The **pscoast** command is what we use to plot land features (or water) on maps. It can plot coastlines, rivers, lakes, and political boundaries. The command we typed was:

```
pscoast -R0/360/-90/90 -JG-111/45/4.5i -Bg30 -Dc -A8000 \
  -G10/10/10 -W3/10/10/10 -P -K >! globe.ps
```

The options we used are described as follows:

Flag	Options	Purpose
<b>-R</b>	<b>0/360/-90/90</b>	The <b>-R</b> flag sets the region of interest. Here I am plotting a map of the entire globe, so I set the region of interest as being between longitude 0° and 360° and latitude from -90° to 90°.
<b>-J</b>	<b>G-111/45/4.5i</b>	The <b>-J</b> flag sets what map projection you want to use. Here we select G, which says use an Orthographic projection. (for an overview of the map projections available go to the <i>GMT webpage &gt; DOCS &gt; GMT Technical Reference and Cookbook &gt; 6. GMT Map Projections</i> ). Different map projections have different input options: -JG requires 3 numbers: -111 says center the map on a longitude of -111°, 45 means center map on a latitude of 45°, and 4.5i means make the size of the map 4.5i across.  So, now is a good time to play around with the script above. Try changing the center latitude and longitude and rerun the script.
<b>-B</b>	<b>g30</b>	The <b>-B</b> flag tells us how often to draw latitude and longitude lines. Here the g says how often to draw gridlines. Try the script as g5 and see what happens.
<b>-D</b>	<b>c</b>	This flag tells us what resolution of coastline data to use. Here the c stands for crude. You can also try <b>full</b> , <b>high</b> , <b>intermediate</b> , and <b>low</b> .
<b>-A</b>	<b>8000</b>	In the example this flag says: do not plot features smaller than 8000 km <sup>2</sup> .  Just completely remove the <b>-A</b> command and see what happens.  Now change <b>-D</b> option to <b>-Df</b> and see what happens. What does this do to the file size?
<b>-G</b>	<b>10/10/10</b>	This option tells us what color we want to use to plot the land masses (see the section below on the R/G/B color space). To get an idea how this works try plotting using <b>-G0/255/0</b> .

<b>-W</b>	<b>3/10/10/10</b>	This says we want to draw solid lines around our coastlines. Here the 10/10/10 says what RGB color to use and the 3 says use a line thickness of 3.  Try it as: -W5/0/0/255 (you probably want to have the -A8000 flag turned on for this one)
<b>-P</b>		This flag says the plot should be in Portrait mode (i.e., page size 8.5" × 11"). Leaving out the -P option would say plot in landscape mode (11" × 8.5").
<b>-K</b>		<b><i>This is an extremely important flag!</i></b> Many of your problems using GMT will center on forgetting to put this flag in, or putting it in when it is not needed. What it says is that I will provide more GMT commands later, so <b><i>do not</i></b> close the postscript file yet!

Note, that we used the command all as follows:

```
pscoast -R0/360/-90/90 -JG-111/45/4.5i -Bg30 -Dc -A8000 \
  -G10/10/10 -W3/10/10/10 -P -K >! globe.ps
```

There are two important things to note:

- 1) At the end of the first line we used a “\” symbol (a backslash). This says, we are not finished typing the command, but that we ran out of room on the first line. The backslash says to continue this command on the next line.
- 2) The output of the pscoast command goes into **>! globe.ps**. This says (remember our C Shell conventions) to force the creation of a new file called **globe.ps** and put all of the output in this file. Since this is the first GMT command we used in this script, we used the **>!** redirection to ensure that we opened up a new file.

Wow, so that’s a lot of options with the pscoast command. But, if you noticed in looking at the man page there are even more that we didn’t even touch! At first, it will likely seem really arbitrary, but after a little while you will generally just remember what all of the flags are and be able to create plots quickly.

## 2.2 psxy

Another command you will really get to know is psxy. This command is used to plot lines, symbols, or polygons on a map. In our example we used it to plot a red box around our study region. The command we used was:

```
psxy -R -JG -W6/255/0/0 -P -O -Am << END >> globe.ps
-100 40
-100 50
-120 50
-120 40
-100 40
END
```

Let's go through the options we used for this command also:

Flag	Options	Purpose
<b>-R</b>		The <b>-R</b> flag always serves the same purpose, i.e., to specify the region of interest. In this case, we do not add a region. This is because we specified the region with the <code>pscoast</code> command already. Hence, GMT will use the same region specified above.
<b>-J</b>	<b>G</b>	We want to plot our box on the same map as specified with <code>pscoast</code> , hence we just leave the option as <b>-JG</b> and GMT knows to keep the projection as is.
<b>-W</b>	<b>6/255/0/0</b>	Draw the box with a red line (255/0/0) with a thickness of 6 pts.
<b>-P</b>		Portrait mode.
<b>-O</b>		<i><b>Muy importante!</b></i> In <code>pscoast</code> we used the <b>-K</b> flag which said, we will add more to the plot later. The <b>-O</b> option is now used, saying let's overlay the results of this <code>psxy</code> command on top of whatever was supplied in previous commands. Note, we do not now use a <b>-K</b> command because we don't want to add anything else to plot later.
<b>-A</b>	<b>m</b>	This command controls how to draw connecting lines, i.e., as great circle arcs or not. In this case, I wanted the lines to follow lat/lon lines so the <b>m</b> option did this.

So, what is this odd redirection we did with this command?

```
<< END >> globe.ps
lon lat
lon lat
...
END
```

1) Note, we use **>> globe.ps**. This is important as we want to append the results of the `psxy` command into our file **globe.ps**.

2) We also used: **<< END**. This states that we are going to redirect anything between the current line and the **END** statement into the `psxy` command. In this case we have a list of longitudes and latitudes we want `psxy` to connect into a box.

Another option is to put the lons and lats into a file. For example, if we placed our locations in a file named **box.xy**. Then we could run `psxy` as:

```
psxy box.xy -R -JG -W6/255/0/0 -P -O -Am >> globe.ps
```

### 3. Interlude - The RGB color space

Specifying color in GMT is always done with **Red/Green/Blue** values. The amount of red, green, or blue varies from 0 – 255. So, **255/0/0** indicates to use the maximum amount of red and no green or blue, i.e., make the color pure red. **0/0/0** indicates not to use any color values, i.e., black. **255/255/255** indicates using the maximum amount of each color, i.e., make it white.

Color is often specified as shown above as either a line color (generally specified with the **-W** flag) or as a fill color (specified with the **-G** flag). But, in a later lecture we will discuss Color Palette Tables, which also use the RGB color space.

As a final note, it is often useful to have a webpage bookmarked that shows you colors and the equivalent RGB value. There are plenty of sites on the web (e.g., google **rgb colors**). An example is given below:

<http://cloford.com/resources/colours/500col.htm>

cadetblue 3		<a href="#">cadetblue 3</a>	122 197 205
cadetblue 4		<a href="#">cadetblue 4</a>	83 134 139
turquoise 1		<a href="#">turquoise 1</a>	0 245 255
turquoise 2		<a href="#">turquoise 2</a>	0 229 238

This is really nice, for example scrolling through the colors I think that cadetblue 4 may be useful for plotting the fill colors of lakes. To use this color I can quickly see that: Red=83, Blue=134, and Green=139.

### 4. Interlude #2 – What are postscript files?

You may have noticed by now that we have output the names of our plot files with the **.ps** extension. This is because our GMT output is in the **postscript** format.

Postscript is a language that describes what goes on a printed page. Its development was intimately tied to the development of the first laser printers and still remains a standard. The postscript language was actually created by John Warnock (a UU graduate who co-founded Adobe and for whom the Warnock building is named) and Charles Geschke around 1982.

Note that postscript files are just text files that you could actually edit (not recommended) with your favorite text editor (open one up and check it out). As another note, it is usually advisable to store your plots in postscript format if you are thinking about using them in a scientific publication. As we will later see, this makes them exceptionally easy to edit in Adobe Illustrator. Note that the encapsulated postscript format (.eps) which is an export option in Matlab is the same thing.

## 5. The **-K** and **-O** flags

These flags are often the most confusing to new users of GMT but they are truly quite simple to understand. The basic rules are:

- 1) If I am going to add more to my plot with additional GMT commands, then I need to use the **-K** flag, which states: *I will append more to this plot later.*
- 2) Any time I am appending to a plot that I have already drawn to (i.e., on the previous line I used the **-K** flag) then I must use the **-O** flag, which states: *I am overlaying onto a plot that has already been started.*
- 3) The very last command used in generating a plot must **NOT** have a **-K** flag. This is because the postscript file needs to be closed with an **end** statement in the postscript language. The **-K** flag keeps the file open for more plotting instructions, and does not close the file. If you try to send your postscript file to the printer and it doesn't print out the most common culprit to your printing problems is that you didn't close out the file.

Here are some examples of proper usage of these flags.

**Example 1:** GMT Plot with a single command.

```
# For a single command don't use either the -K or -O flag
gmtcommand -flags >! plot.ps
```

**Example 2:** GMT Plot with two commands.

```
# Use the -K flag on the first line
gmtcommand -flags -K >! plot.ps

# Use the -O flag on the last line, but not the -K flag
gmtcommand -flags -O >> plot.ps
```

**Example 3:** GMT Plot with three or more commands.

```
# Use the -K flag on the first line
gmtcommand -flags -K >! plot.ps

# Use both -K and -O flags on all but the last command
gmtcommand -flags -K -O >> plot.ps
...

# Just use the -O flag on the last line
gmtcommand -flags -O >> plot.ps
```

## 6. X-Y Plots

GMT is definitely a powerful tool for mapping applications. However, one of its most powerful aspects as a researcher is in its potential for creating simple 2D plots. The reason I think it is so powerful is because (1) It is scriptable, meaning from one single script I can plot multiple data sets in exactly the same way with minimal effort, and (2) GMT provides complete control over every aspect of how the plot looks. These are huge advantages over plotting programs that require you to click your way through options and only offer minimal control over how the plot looks (e.g., Excel).

To wrap up this lecture let's just look at a really quick plot of some data.

```
#!/bin/csh
#-----#
# First we will create a data set to plot
#
# Here we are going to create a wave packet example.
# This is just a combination of two sinusoids with a 100
# and 20 sec period
# i.e., angular frequency = 2*pi/ T
#-----#

@ time = 0
while ($time <= 1000)

if ($time == 0) then
    echo $time >! temp.xy
else
    echo $time >> temp.xy
endif

@ time = $time + 1
end

# now let's calculate some amplitude values
awk '{print $1, (5*sin(-$1*.063) + 2.5*sin(-$1*.314))}' temp.xy
>! timeseries.xy
rm temp.xy

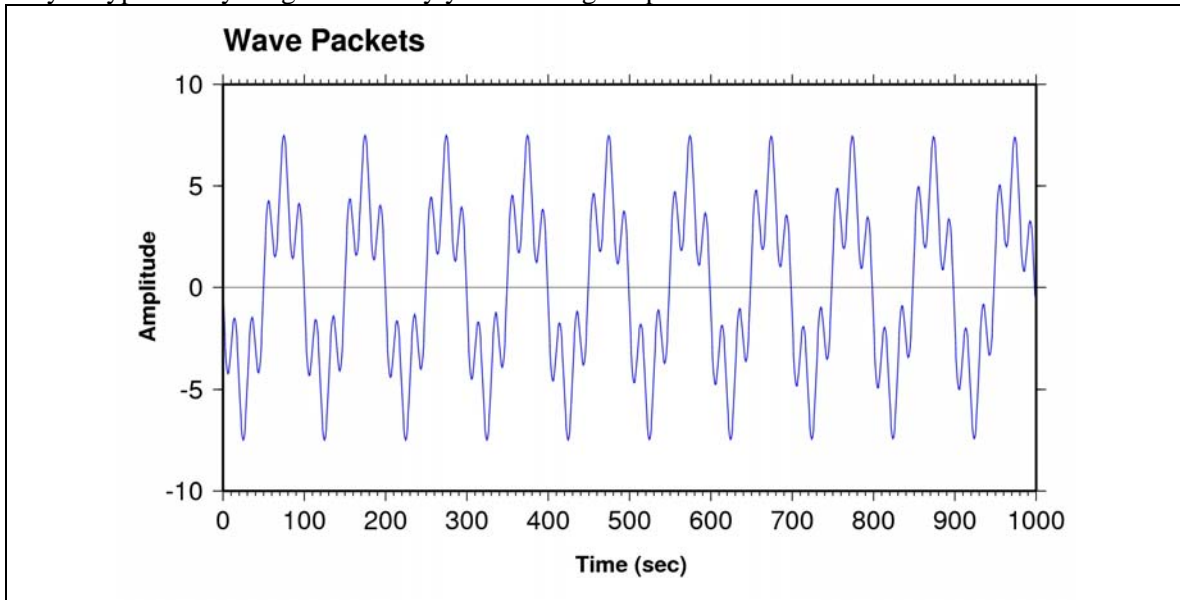
# plot the time series
psxy timeseries.xy -JX6i/3i -R0/1000/-10/10 -W2/0/0/255 -P \
    -B100g10000f10/5g10000nseW -X1.5 -Y6.0 -K >! plot.ps

# add some text describing the plot and axes
pstext -R0/8/0/11 -JX8i/11i -P -O -N << END >> plot.ps
0.0 3.25 16 0 1 1 Wave Packets
2.6 -0.6 12 0 1 1 Time (sec)
-0.5 1.1 12 90 1 1 Amplitude
END

gs -sDEVICE=x11 plot.ps
```



If you typed everything in correctly you should get a plot that looks like this:



Note that we use `psxy` as our primary command for generating this plot. I will not go into detail as to what all of the options mean. It's better just to get some practice on your own. Hence, it's time for the homework!

## 7. Homework

- 1) Create a global map showing the major plate boundaries (an x,y table of plate boundaries are given on the web page) using a Mollweide projection and centered on the Pacific Plate.
- 2) Create a map that will plot the locations of all Global Seismic Network (GSN) stations using a Winkel projection. Put the station code (e.g., AAK) next to the plot symbol for the station. The station names and locations are provided in a file on the web page.
- 3) The data file: `envelope.xy` is given on the web page. This data file contains the envelope of a seismogram recorded at Eilson Airforce Base in Alaska with the direct P-wave arrival aligned at 0 sec. The envelope is given here to show the exponential decay of the coda waves. Make a plot of this seismogram in two separate panels. In the top panel show the raw seismogram as provided in the file `envelope.xy`. Anytime a quantity displays exponential behavior it is customary to plot this as the natural log of the amplitudes instead of the raw amplitude. Such a plot would then show a linear decay instead of exponential decay, and the exponential decay can then be estimated by fitting a straight line to the linear region. Make the bottom panel of the plot should be the natural log of the amplitude of the seismogram, and show which region immediately following the direct P-arrival can be fit with a straight line.
- 4) This will be one of the most useful scripts you will ever write. When doing any type of scientific work one always generates tabular data (i.e., x, y pairs of data). It is really nice to quickly be able to take a glance at these data without having to write a special GMT script every time. For a finalized plot of data you probably will want to write a special script, but sometimes its really, really useful to just be able to take a quick glance at it. Write a generic script that will plot a 2D group of data. Make an option for plotting solid lines or symbols. Name the script

**xyplot**. The plot should contain a title that contains the name of the file being plotted. The script should run as:

```
>> xyplot input.xy s
```

where, if the **s** option is given, the data will be plotted with filled in circles.

**Hint:** Use GMT's **minmax** and **gmtmath** commands to help in determining the plot range, and spacing between tickmarks for the axes.