



GMT

Generic Mapping Tools or
Gravity, Magnetism and Topography

Lecture #1

Mapping and Plotting with GMT

GMT 4.5.1

- Began as a set of subroutines to write Postscript commands
- Grew with Paul's and Walter's Ph.D. Theses
- Encompassed map projections (30!)
- Data Analysis
- Cross-Over Errors



Paul Wessel, our hero

GMT resources

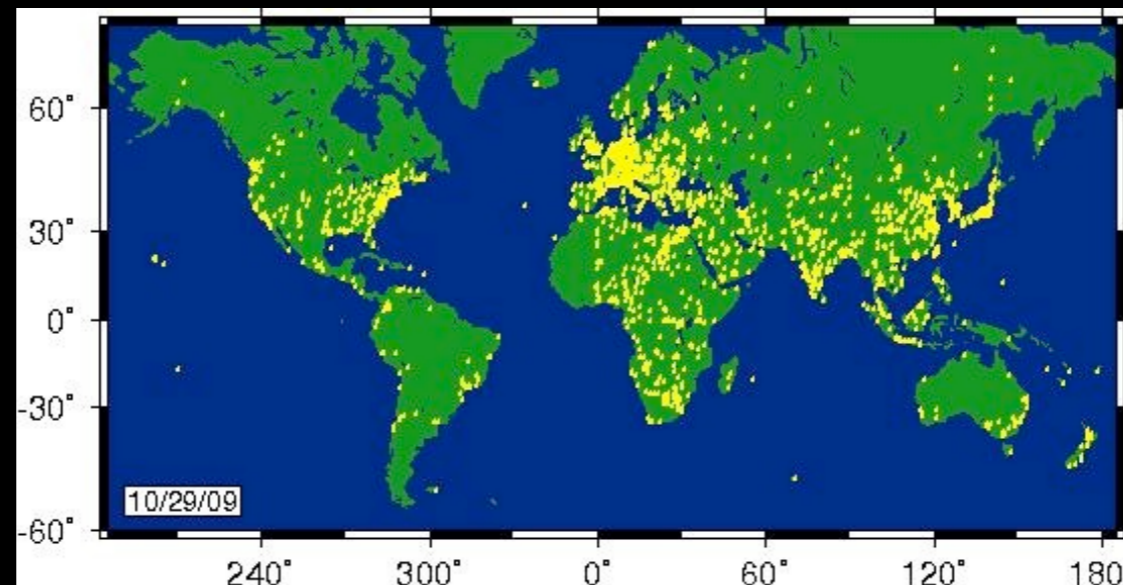
- Web site - gmt.soest.hawaii.edu
- User Group
- Open source project
- Windows, OS X, Unix and OS/2
- Extensive documentation (html and pdf)
- Examples and “cookbook”
- Ancillary data sets included (eg. coastlines)

Postscript

- Vector graphic language
- Rasterizes for output to various devices
- Scale set by dots-per-inch (dpi)
- Typically 300-1200

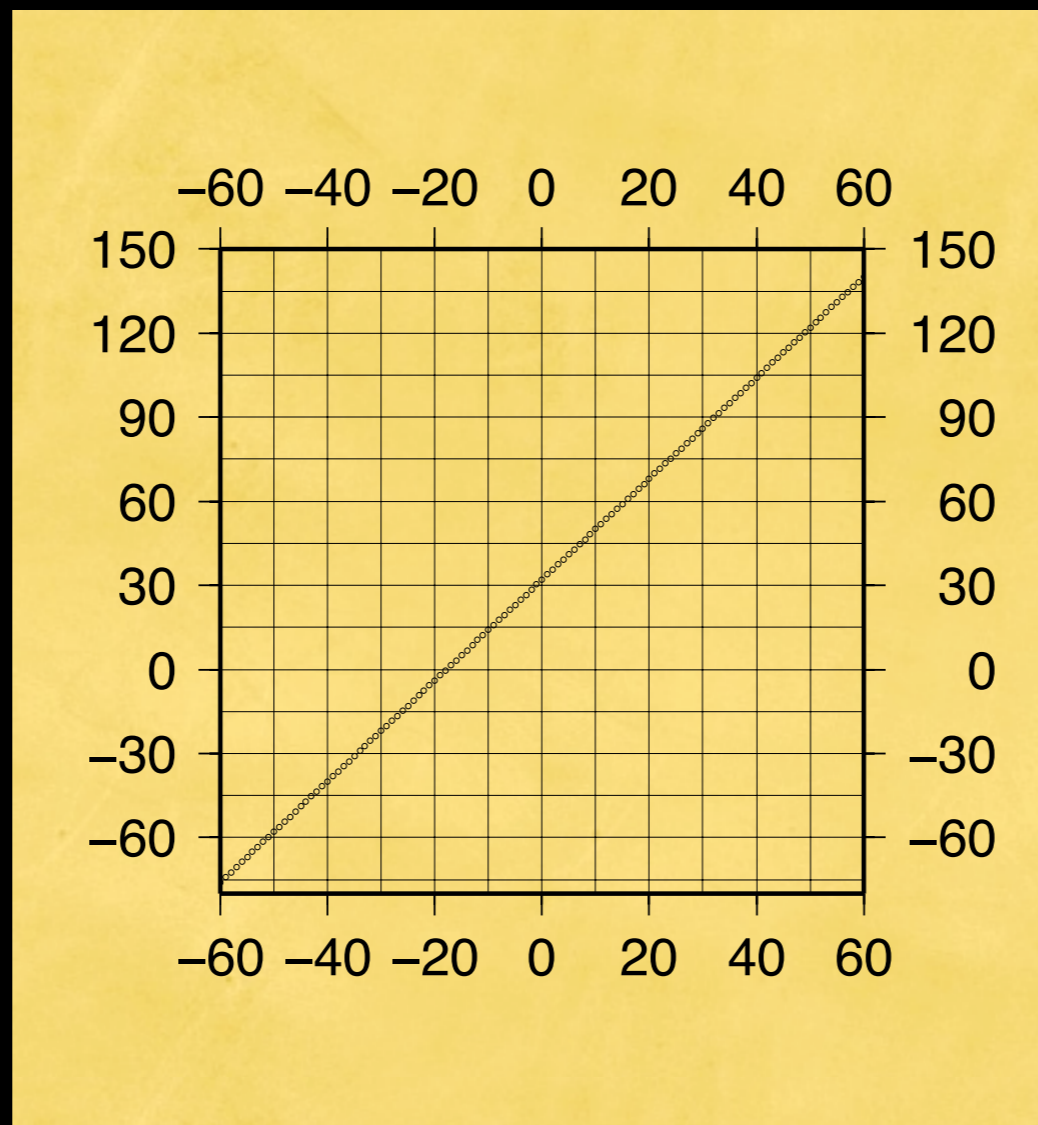
How does GMT work?

- Scripted language for vector graphics
- Facilitates automated plotting
- Relates graphic space to the data space
- Sequential commands create a plot or map



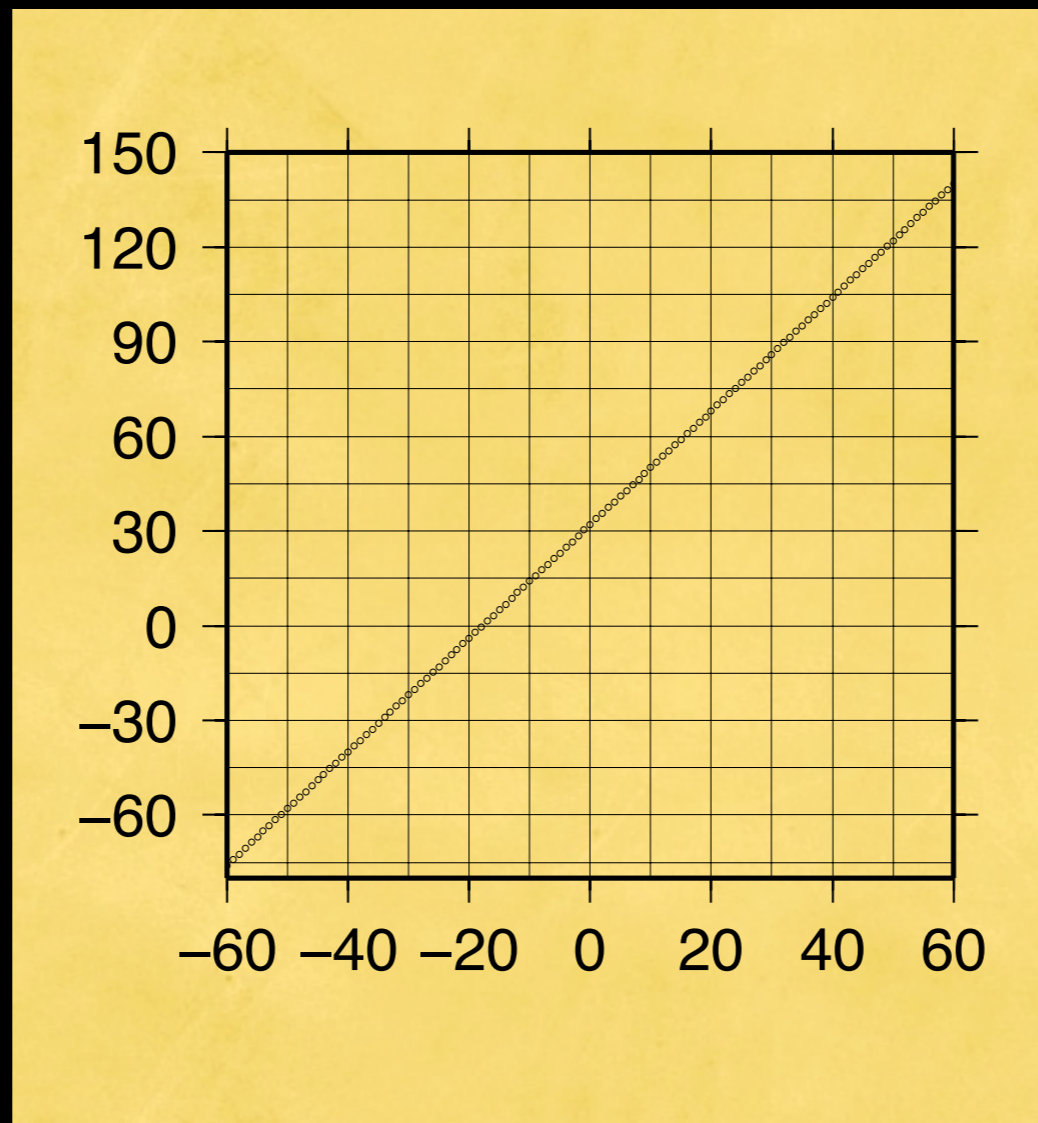
Create a simple plot.

```
pwd |  
awk '{for (i = -60; i < 61; ++i) {printf"%d %.2f\n",i,32+9/5*i}}' |  
psxy -R-60/60/-80/150 -jX6.0 -Sc0.05 -Ba20g|0/a30g|5 > test.ps
```



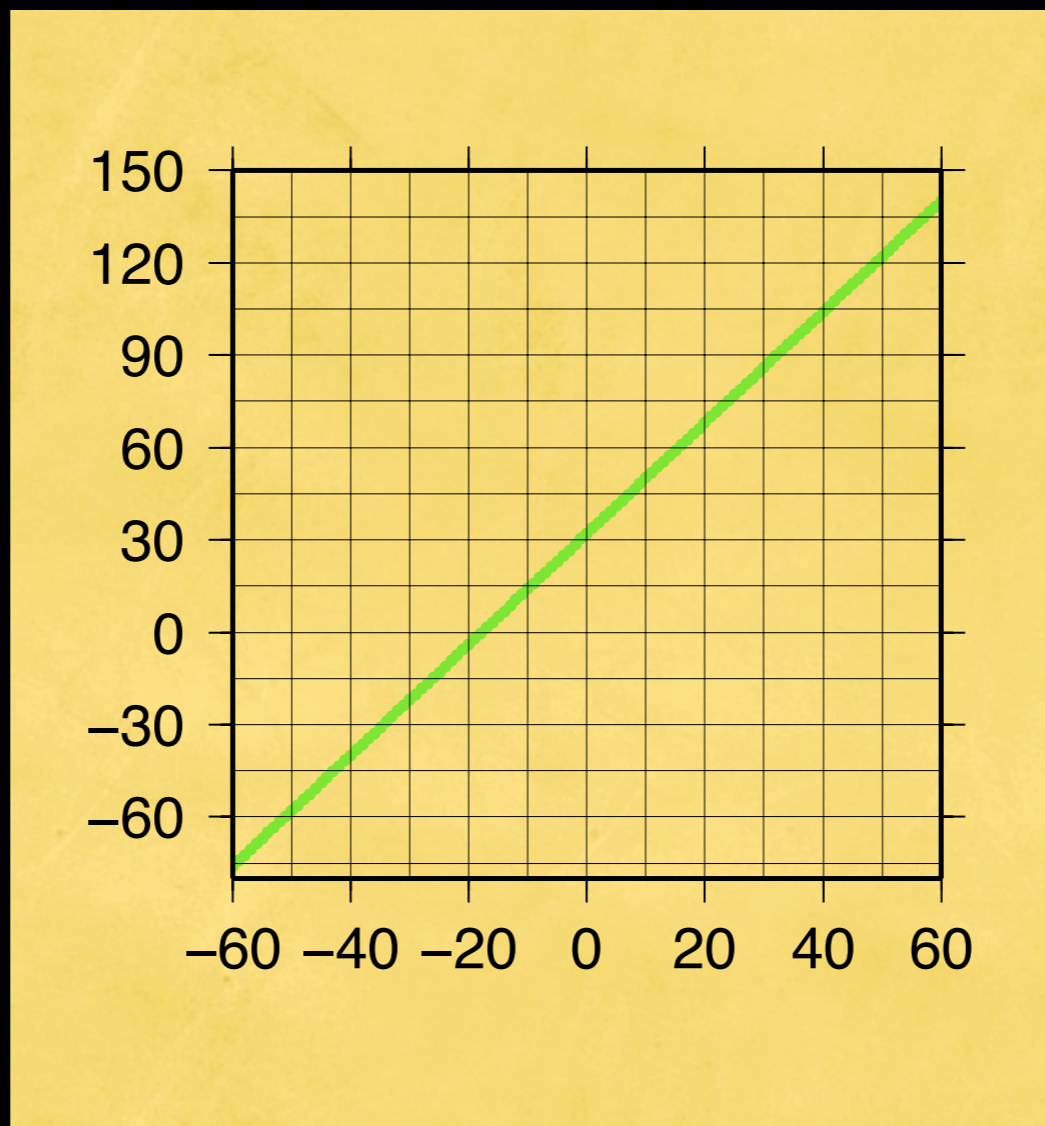
Change it

Change -Ba20gI0/a30gI5 to -Ba20gI0/a30gI5WeSn



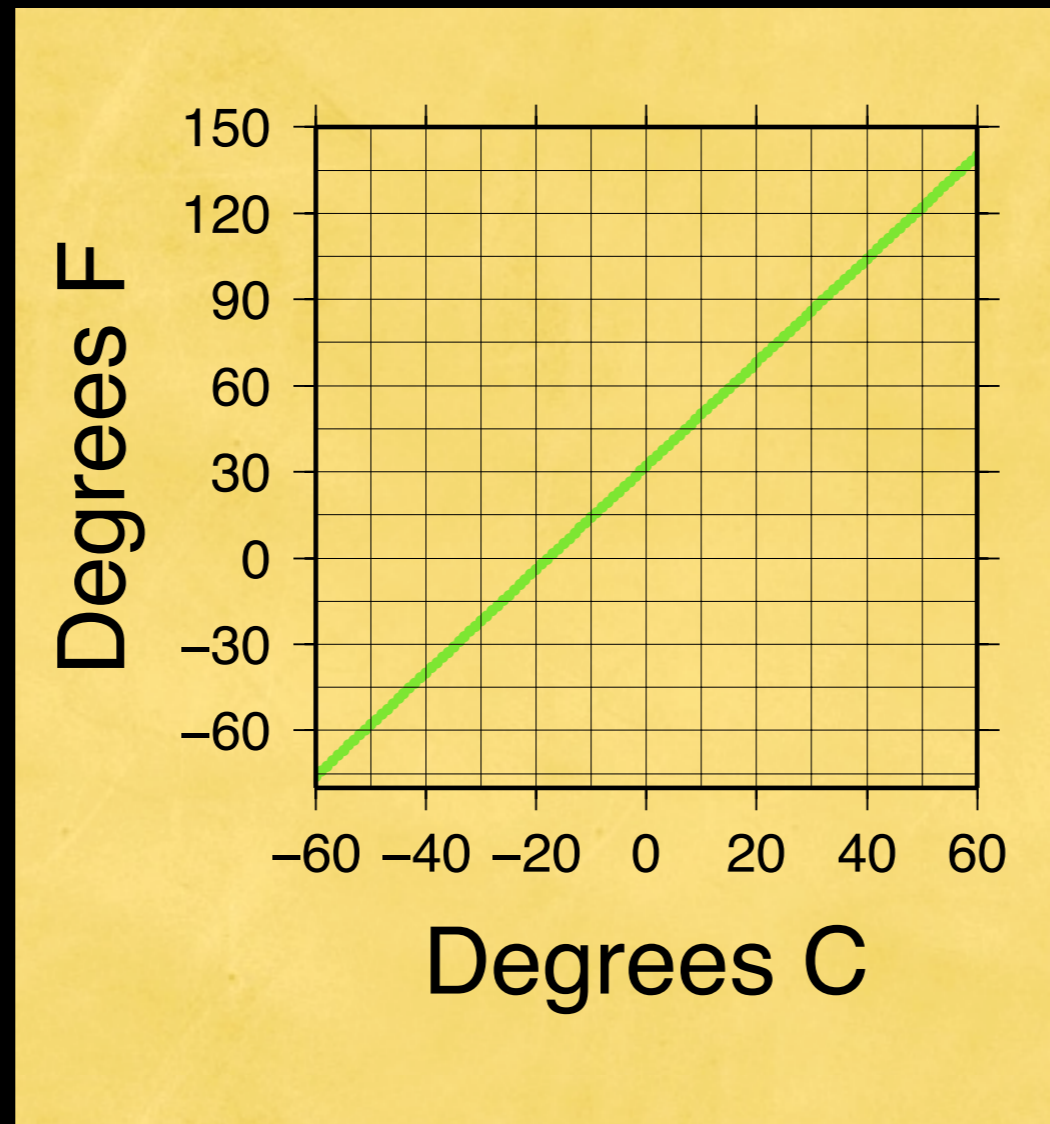
Again

Change -Sc0.10 to -Sc0.10 -G125/230/50



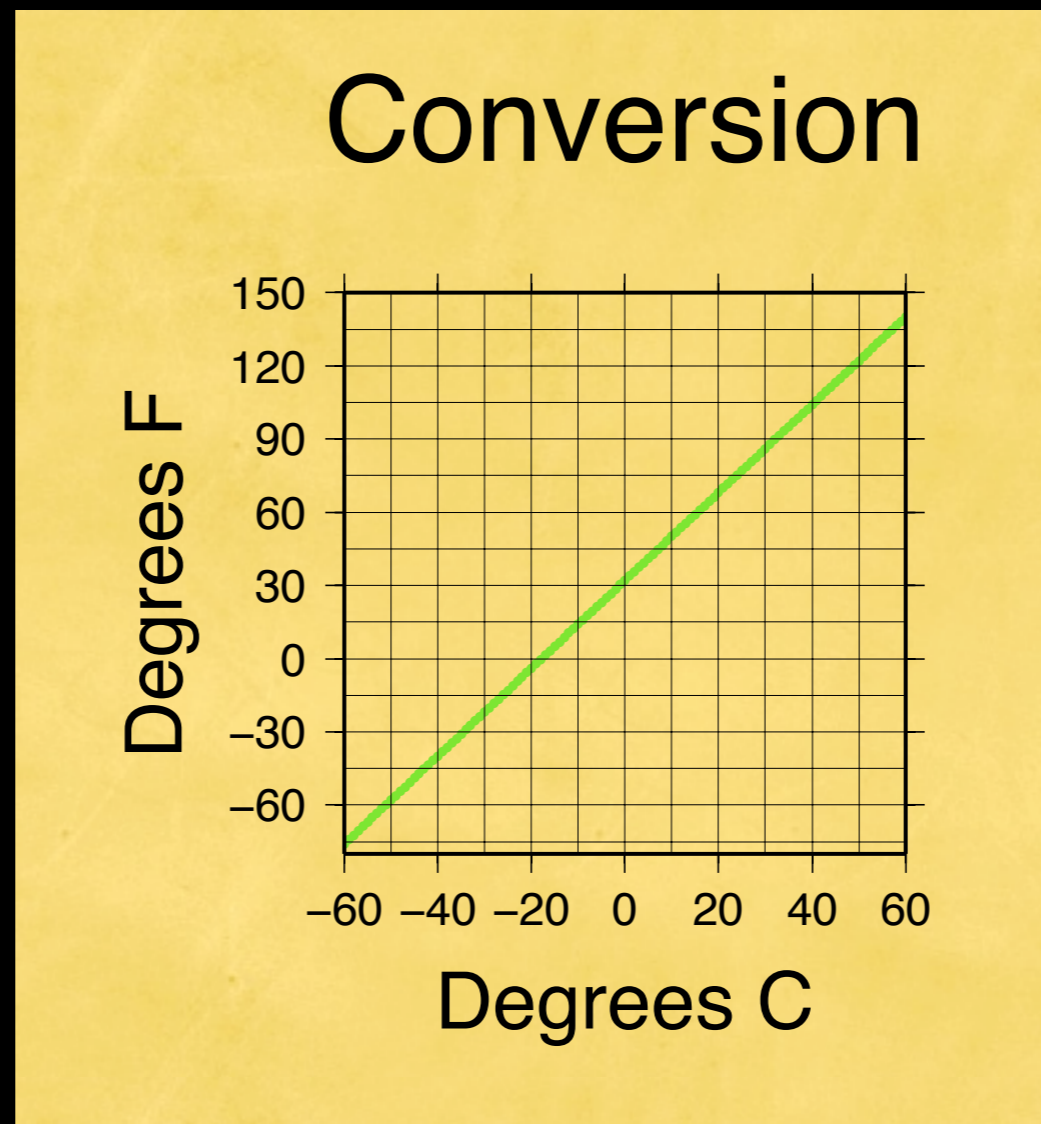
Again

Change -Ba20gI0/a30gI5WeSn to -Ba20gI0:"Degrees C":/a30gI5:"Degrees F":WeSn



Again

Change -Ba20g10:"Degrees C":/a30g15:"Degrees F":WeSn to
-Ba20g10:"Degrees C":/a30g15:"Degrees F"::Conversion:WeSn

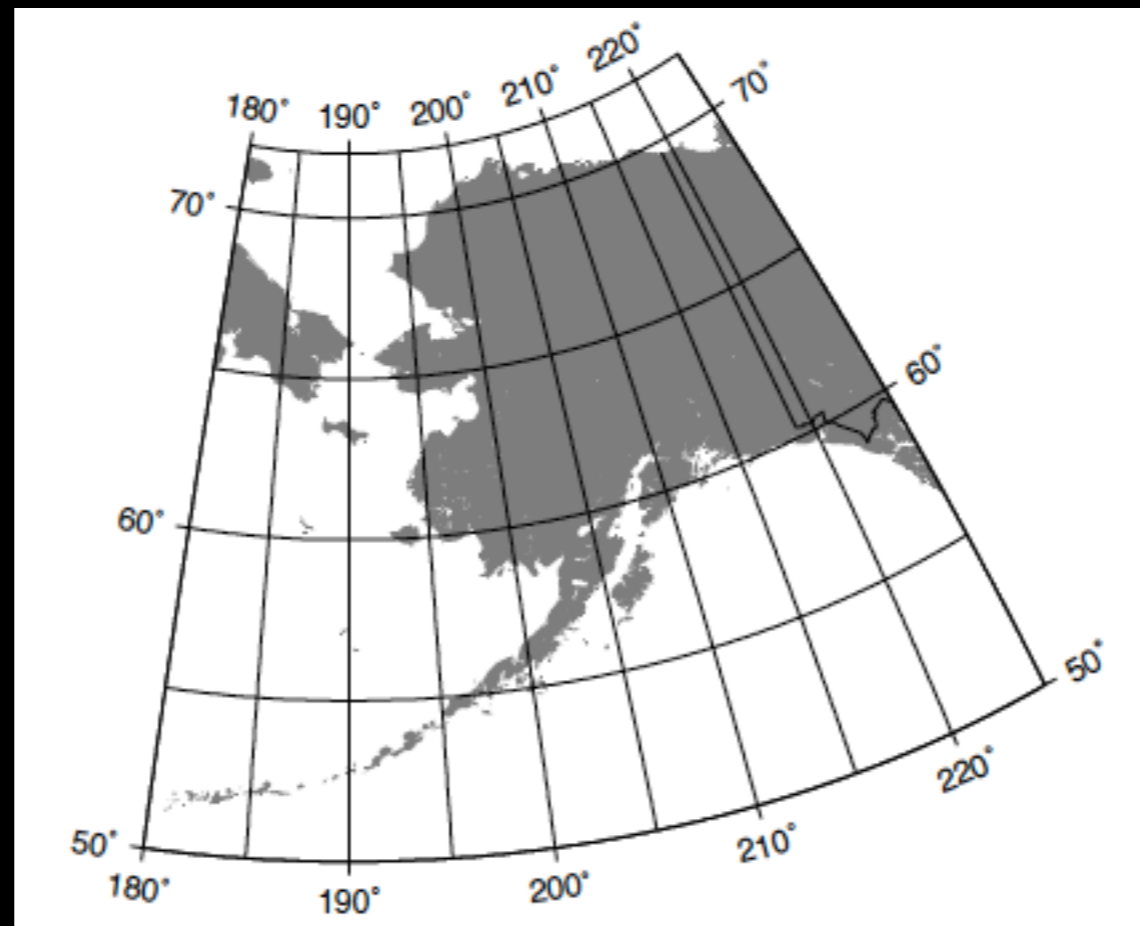


Graphic Programs

- **grdcontour** Contouring of 2-D gridded data
- grdimage Produce images from 2-D gridded data
- grdvector Plot vector fields from 2-D gridded data
- grdview 3-D perspective imaging of 2-D gridded data
- **psbasemap** Create a basemap frame
- psclip Use polygon files as clipping paths
- **pscoast** Plot coastlines, filled continents, rivers, and political borders
- pscontour Direct contouring or imaging of xyz-data by triangulation
- pshistogram Plot a histogram
- psimage Plot Sun rasterfiles on a map
- pslegend Plot legend on a map
- psmask Create overlay to mask specified regions of a map
- psrose Plot sector or rose diagrams
- psscale Plot grayscale or colorscale
- **pstext** Plot textstrings
- pswiggle Draw anomalies along track
- **psxy** Plot symbols, polygons, and lines in 2-D
- psxyz Plot symbols, polygons, and lines in 3-D

Create a simple map

```
pscoast -jc | 90/68/0.50 -R | 80/225/50/72 -Di -NI -G | 25 -Bg5a | 0 > Alaska.ps
```



The 17 standardized GMT command line switches

- B** Defines tickmarks, annotations, and labels for basemaps and axes
- H Specifies that input/output tables have header record(s)
- J** Selects a map projection or coordinate transformation
- K** Allows more plot code to be appended to this plot later
- O** Allows this plot code to be appended to an existing plot
- P Selects Portrait plot orientation [Default is landscape]
- R** Defines the extent of the map/plot region
- U Plots a time-stamp, by default in the lower left corner of page
- V Selects verbose operation; reporting on progress
- X Sets the x-coordinate for the plot origin on the page
- Y Sets the y-coordinate for the plot origin on the page
- b Selects binary input and/or output
- c Specifies the number of plot copies
- f Specifies the data format on a per column basis
- g Identify data gaps based on supplied criteria
- m Specifies data in multiple segment format
- : Assumes input geographic data are (*lat,lon*) and not (*lon,lat*)

Command line on 1st plot

```
pscoast -Jc 190/68/0.50 -R 180/225/50/72 -Di -NI -G 125 -Bg5a 10 > test.ps
```

-Jc specifies the Cassini projection, centered at 190 longitude, 68 latitude and scale at 0.50 inches per map unit

-R defines the map area in WESN form

-Di says to use the intermediate resolution coastline

-NI says to plot international boundaries

-G says to fill the continents with a medium gray

-Bg5a 10 says to put a grid on the map at every 5 map units and annotate it every 10 map units

It looks easy, but...

-B Sets map boundary annotation and tickmark intervals. The format of *tickinfo* is `[pls]xinfo[/yinfo[/zinfo]][:."Title":][Wl w][E]e][S]s][N]n][Z]z[+]`. The leading `p` [Default] or `s` selects the primary or secondary annotation information. Each of the *?info* segments are textstrings of the form `info:" Axis label"[:="prefix":][:,"unit label":]`. The *info* string is made up of one or more concatenated substrings of the form `[which]stride[+-phase][u]`. The optional *which* can be either a for annotation tick spacing [Default], `f` for frame tick spacing, and `g` for gridline spacing. If frame interval is not set, it is assumed to be the same as annotation interval. *stride* is the desired stride interval. The optional *phase* shifts the annotation interval by that amount. The optional *u* indicates the unit of the *stride* and can be any of `Y` (year, plot with 4 digits), `y` (year, plot with 2 digits), `O` (month, plot using `PLOT DATE FORMAT`), `o` (month, plot with 2 digits), `U` (ISO week, plot using `PLOT DATE FORMAT`), `u` (ISO week, plot using 2 digits), `r` (Gregorian week, 7-day stride from start of week `TIME WEEK START`), `K` (ISO weekday, plot name of day), `D` (date, plot using `PLOT DATE FORMAT`), `d` (day, plot day of month 0-31 or year 1-366, via `PLOT DATE FORMAT`), `R` (day, same as `d`, aligned with `TIME WEEK START`), `H` (hour, plot using `PLOT CLOCK FORMAT`), `h` (hour, plot with 2 digits), `M` (minute, plot using `PLOT CLOCK FORMAT`), `m` (minute, plot with 2 digits), `C` (second, plot using `PLOT CLOCK FORMAT`), `c` (second, plot with 2 digits). Note for geographic axes `m` and `c` instead mean arc minutes and arc seconds. All entities that are language-specific are under control by `TIME LANGUAGE`. To specify separate `x` and `y` ticks, separate the substrings that apply to the `x` and `y` axes with a slash `/` (If a 3-D basemap is selected with `-E` and `-Jz`, a third substring pertaining to the vertical axis may be appended.) For linear/log/power projections (`-JxX`): Labels for each axis can be added by surrounding them with colons `:`. If the first character in the label is a period, then the label is used as plot title; if it is a comma `,` then the label is appended to each annotation; if it is an equal sign `=` the the prefix is prepended to each annotation (start label/prefix with `-` to avoid space between annotation and item); else it is the axis label. If the label consists of more than one word, enclose the entire label in double quotes (e.g., `"my label"`). If you need to use a colon `:` as part of your label you must specify it using its octal code `(\072)`. By default, all 4 boundaries are plotted (referred to as `W, E, S, N`). To change the default, append the code for only those axes you want (e.g., `WS` for standard lower-left `x-` and `y-` axis system). Upper case (e.g., `W`) means draw axis/tickmarks AND annotate it, whereas lower case (e.g., `w`) will only draw axis/tickmarks. (If a 3-D basemap is selected with `-E` and `-Jz`, append `Z` or `z` to control the appearance of the vertical axis. Append `+` to draw the outline of the cube defined by `-R`. Note that for 3-D views the title, if given, will be suppressed.) For non-geographical projections: Give negative scale (in `-Jx`) or axis length (in `-JX`) to change the direction of increasing coordinates (i.e., to make the `y`-axis positive down). For `log10` axes: Annotations can be specified in one of three ways: (1) *stride* can be `1, 2, 3`, or `-n`. Annotations will then occur at `1, 1-2-5`, or `1-2-3-4-...-9`, respectively; for `-n` we annotate every `n`'t magnitude. This option can also be used for the frame and grid intervals. (2) An `l` is appended to the *tickinfo* string. Then, `log10` of the tick value is plotted at every integer `log10` value. (3) A `p` is appended to the *tickinfo* string. Then, annotations appear as 10 raised to `log10` of the tick value. For power axes: Annotations can be specified in one of two ways: (1) *stride* sets the regular annotation interval. (2) A `p` is appended to the *tickinfo* string. Then, the annotation interval is expected to be in transformed units, but the annotation value will be plotted as untransformed units. E.g., if *stride* = 1 and *power* = 0.5 (i.e., `sqrt`), then equidistant annotations labeled `1-4-9...` will appear.

These GMT parameters can affect the appearance of the map boundary: `ANNOT MIN ANGLE`, `ANNOT MIN SPACING`, `ANNOT FONT PRIMARY`, `ANNOT FONT SECONDARY`, `ANNOT FONT SIZE PRIMARY`, `ANNOT FONT SIZE SECONDARY`, `ANNOT OFFSET PRIMARY`, `ANNOT OFFSET SECONDARY`, `BASEMAP AXES`, `BASEMAP FRAME RGB`, `BASEMAP TYPE`, `DEGREE FORMAT`, `FRAME PEN`, `FRAME WIDTH`, `GRID CROSS SIZE PRIMARY`, `GRID PEN PRIMARY`, `GRID CROSS SIZE SECONDARY`, `GRID PEN SECONDARY`, `HEADER FONT`, `HEADER FONT SIZE`, `LABEL FONT`, `LABEL FONT SIZE`, `LINE STEP`, `OBLIQUE ANNOTATION`, `PLOT CLOCK FORMAT`, `PLOT DATE FORMAT`, `TIME FORMAT PRIMARY`, `TIME FORMAT SECONDARY`, `TIME LANGUAGE`, `TIME WEEK START`, `TICK LENGTH`, `TICK PEN`, and `Y AXIS TYPE`; see the `gmtdefaults` man page for details.

gmtdefaults

```
gmtset `cat $workingdir/gmt.plot.parameters` D_FORMAT %.0f MEASURE_UNIT inch
```

```
gmtset D_FORMAT %.6f
```

USE gmtdefaults -D to see default settings

USE gmtdefaults -L to see your current settings

Try it again with a script

create with text editor

“chmod +x” to make it executable

```
#!/bin/csh
# plot a Cassini projected map of Alaska with a user-defined scale

set workingdir = `pwd`
gmtset `cat $workingdir/gmt.plot.parameters`

set scale = $1

set w_lon = 180
set e_lon = 225
set s_lat = 50
set n_lat = 72

set map_box = -R$w_lon/$e_lon/$s_lat/$n_lat
set filename = alaska.ps

set pro_lon = `echo $w_lon $e_lon | awk '{printf "%.2f",($1+$2)/2.0}'`
set pro_lat = `echo $n_lat $s_lat | awk '{printf "%.2f",($1+$2)/2.0}'`

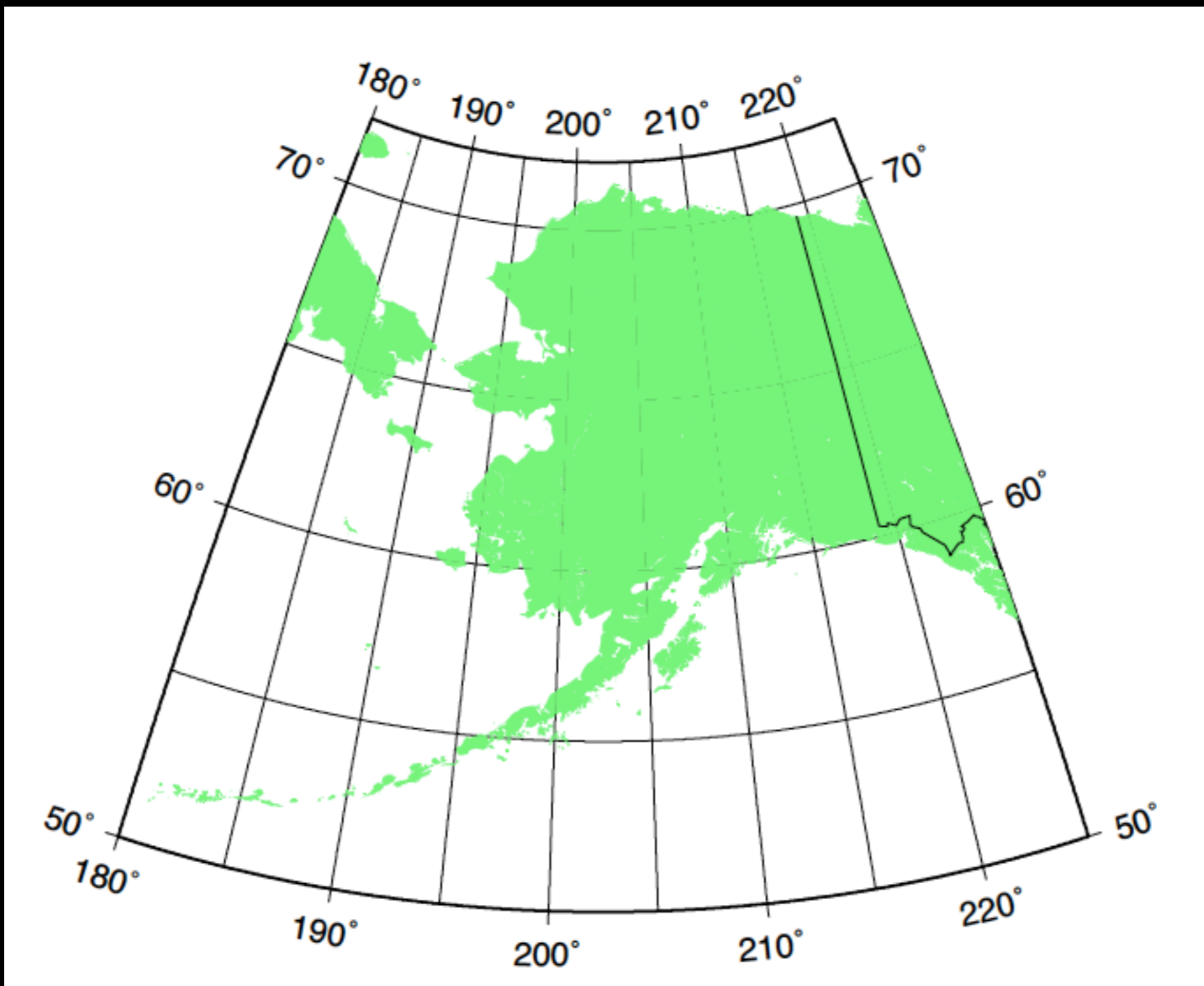
set map_pro = -Jc$pro_lon/$pro_lat/$scale
```

the commands

```
psbasemap $map_pro $map_box -Bg5a10 -K -P > $filename
```

```
pscoast $map_pro $map_box -Di -NI -G125/255/125 -O >>  
$filename
```

the map.....

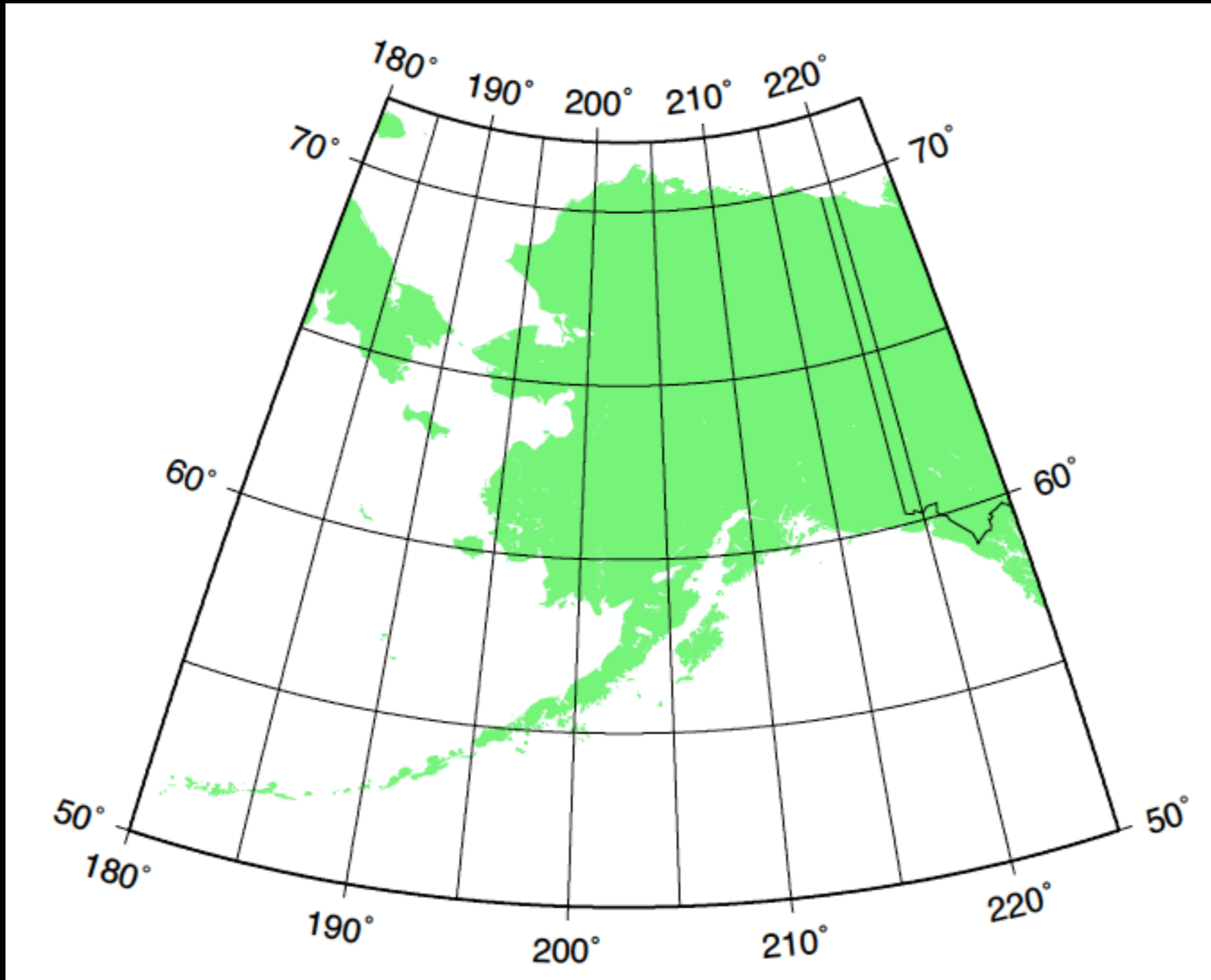


sequence is important

```
pscoast $map_pro $map_box -Di -NI -G 125/255/125 -K -P > $filename
```

```
psbasemap $map_pro $map_box -Bg5a 10 -O >> $filename
```

a somewhat different map



add some data points?

```
pscoast $map_pro $map_box -Di -NI -G125/255/125 -K -P > $filename
```

```
psxy $map_pro $map_box -Sc0.25 -G255/125/125 -K -O >> $filename << END
```

```
210 70
```

```
210 65
```

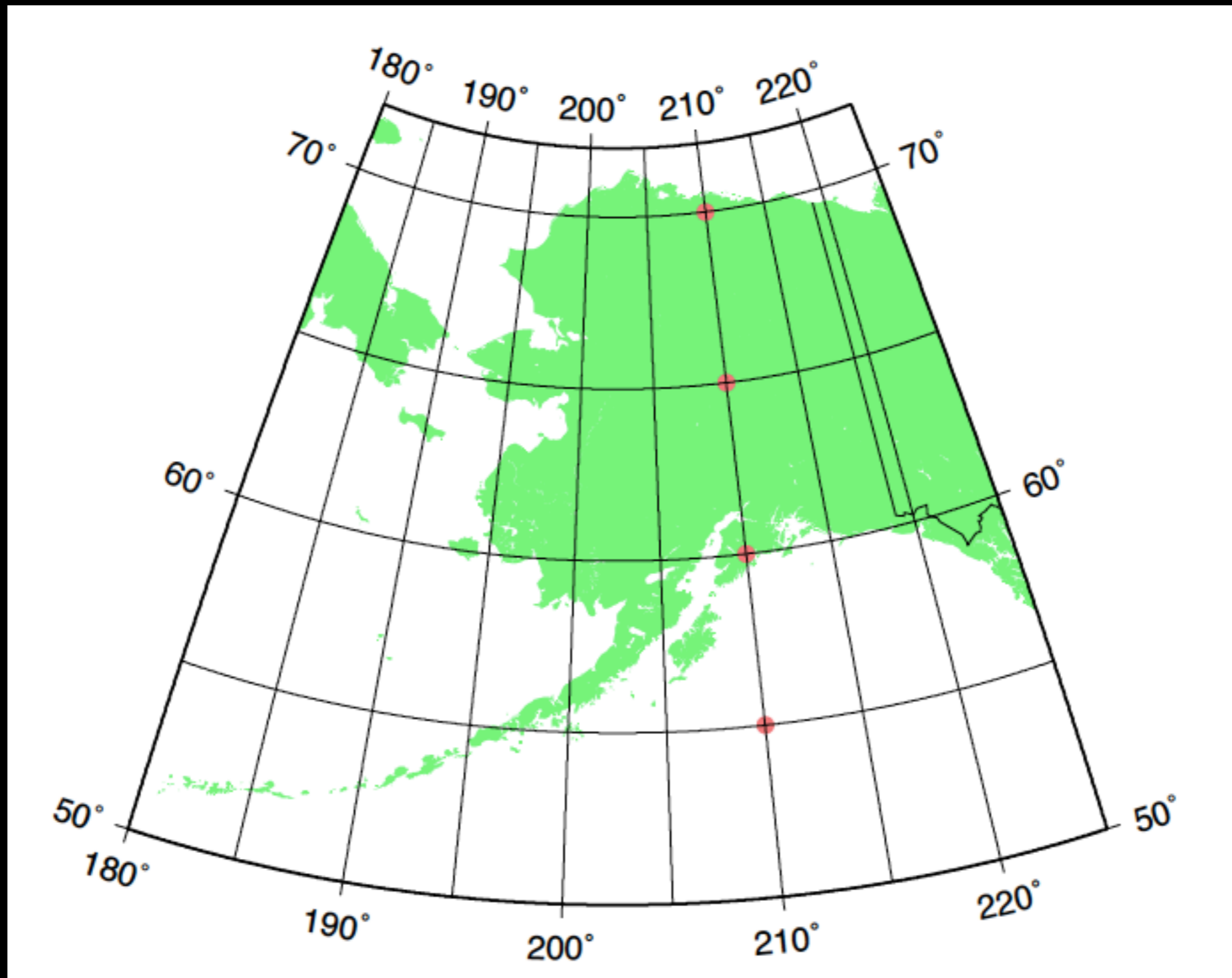
```
210 60
```

```
210 55
```

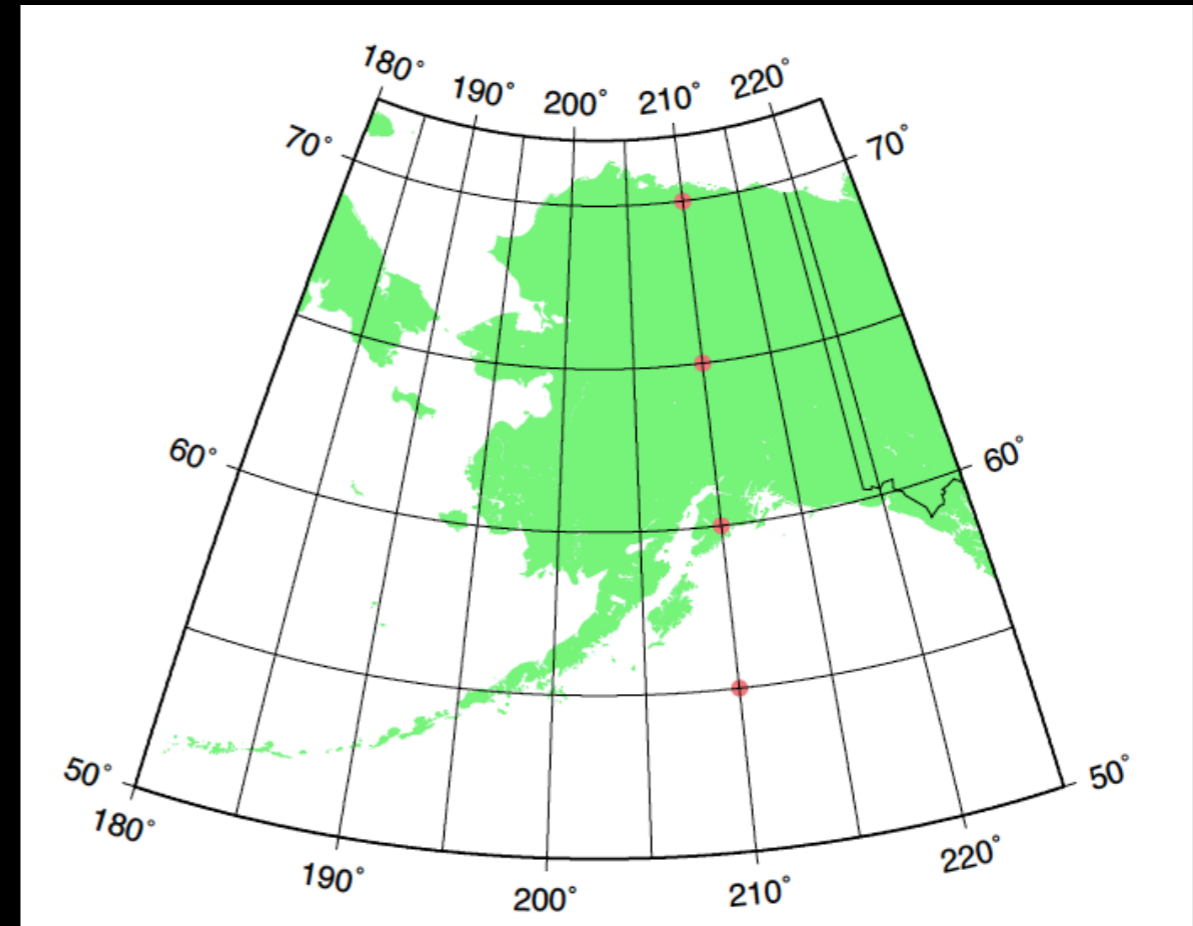
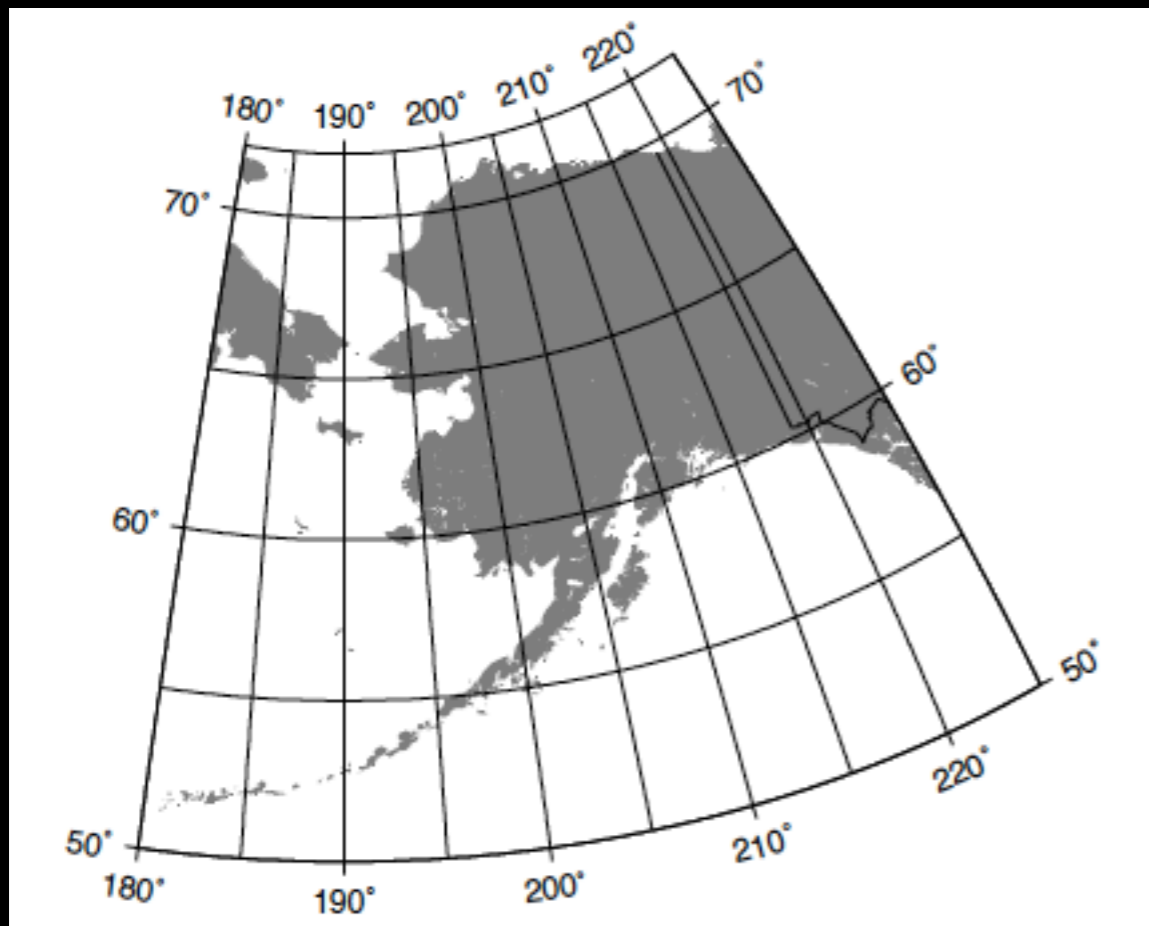
```
END
```

```
psbasemap $map_pro $map_box -Bg5a10 -O >> $filename
```

map with data points



so where did we go?



what are -K and -O?

-O suppresses the header, which defines a number of aliases to make a more compact file

```
%!PS-Adobe-3.0
%%BoundingBox: 0 0 595 842
%%Title: GMT v4.2.0 Document from pscoast
%%Creator: GMT
%%For: bernardcoakley
%%DocumentNeededResources: font Helvetica
%%CreationDate: Thu Oct 29 06:47:11 2009
%%LanguageLevel: 1
%%DocumentData: Clean7Bit
%%Orientation: Portrait
%%Pages: 1
%%EndComments
```

```
%%BeginProlog
250 dict begin
/A /setgray load def
/B /setdash load def
/C /setrgbcolor load def
/D /rlineto load def
/E {dup stringwidth pop} bind def
/F /fill load def
```

-K suppresses the trailer

a lot of text Postscript precedes these last few lines;

```
I O D
S
%%PageTrailer
S -0 -0 T 4.16667 4.16667 scale 0 A
showpage

%%Trailer

end
%%EOF
```

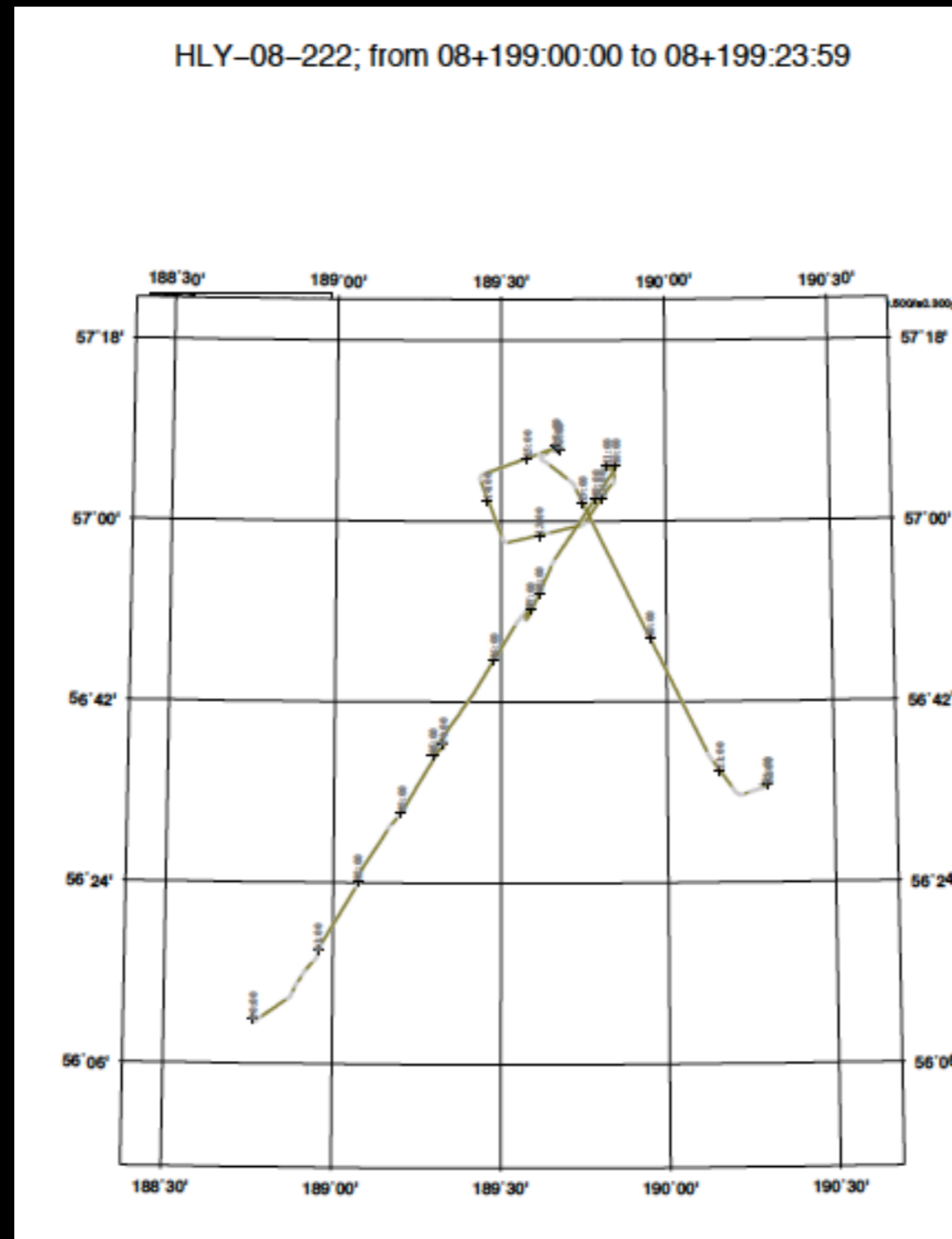
The simplest state uses neither -K or -O and results in a complete plot with both header and trailer.

By using -K and -O we can use GMT commands in series to create a single plot.

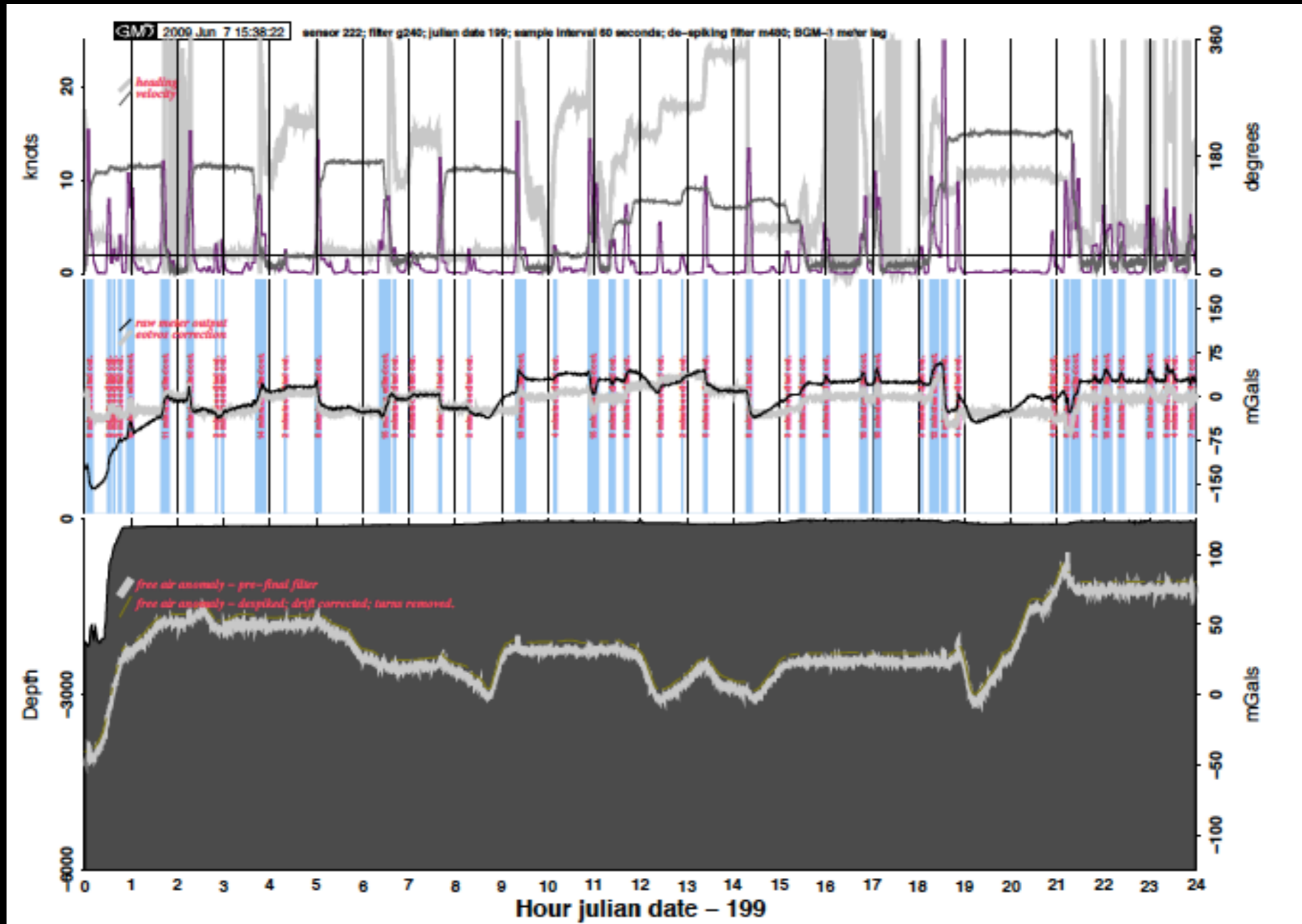
most common errors in GMT scripts

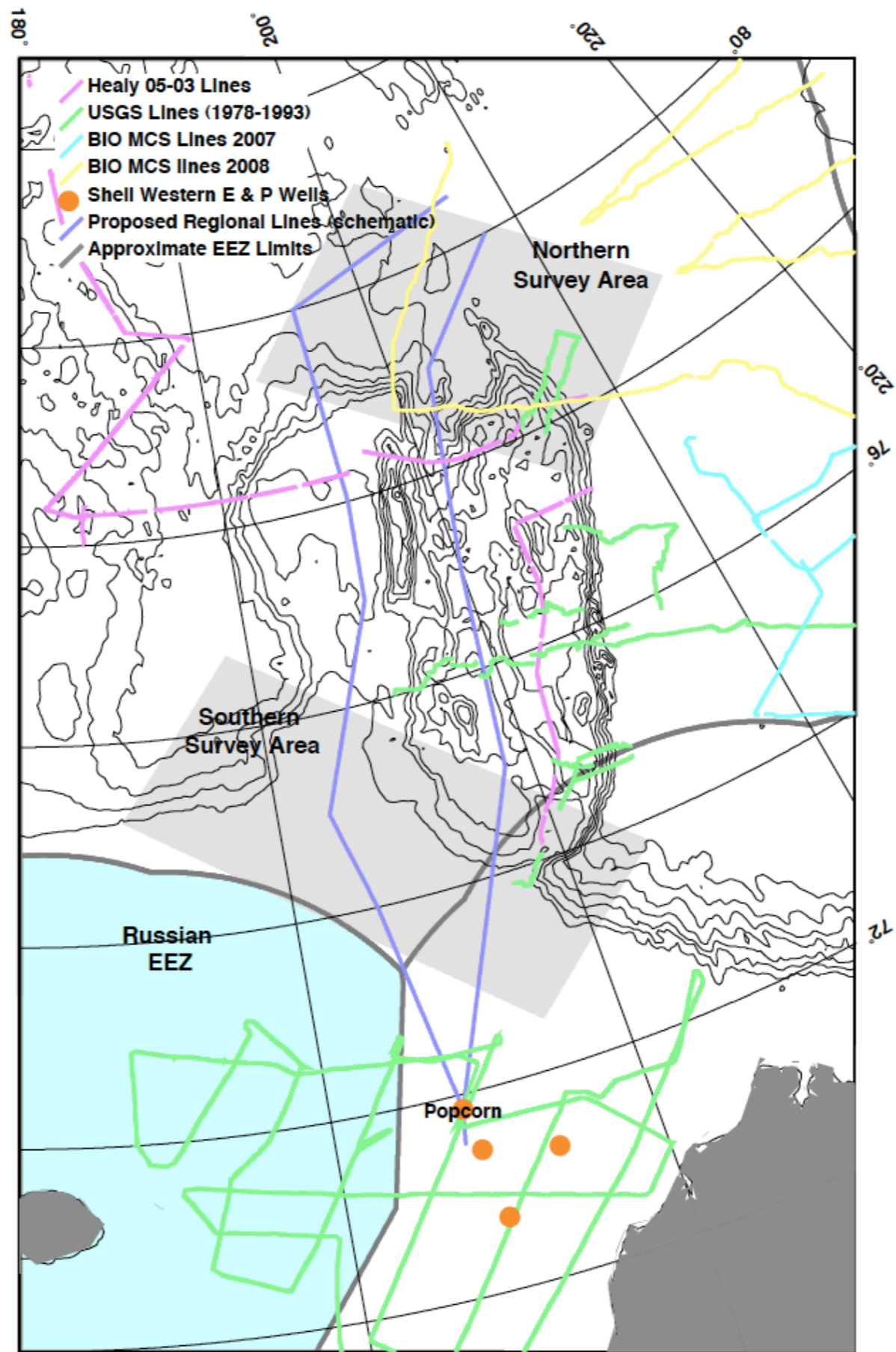
- misuse of -K and -O options
- using “>” instead of “>>” as a redirect
- not specifying -R or -J correctly
- incorrect sequencing of commands
- trying to plot incorrectly scaled data
- trying to write the whole script at once

USCGC Ship track

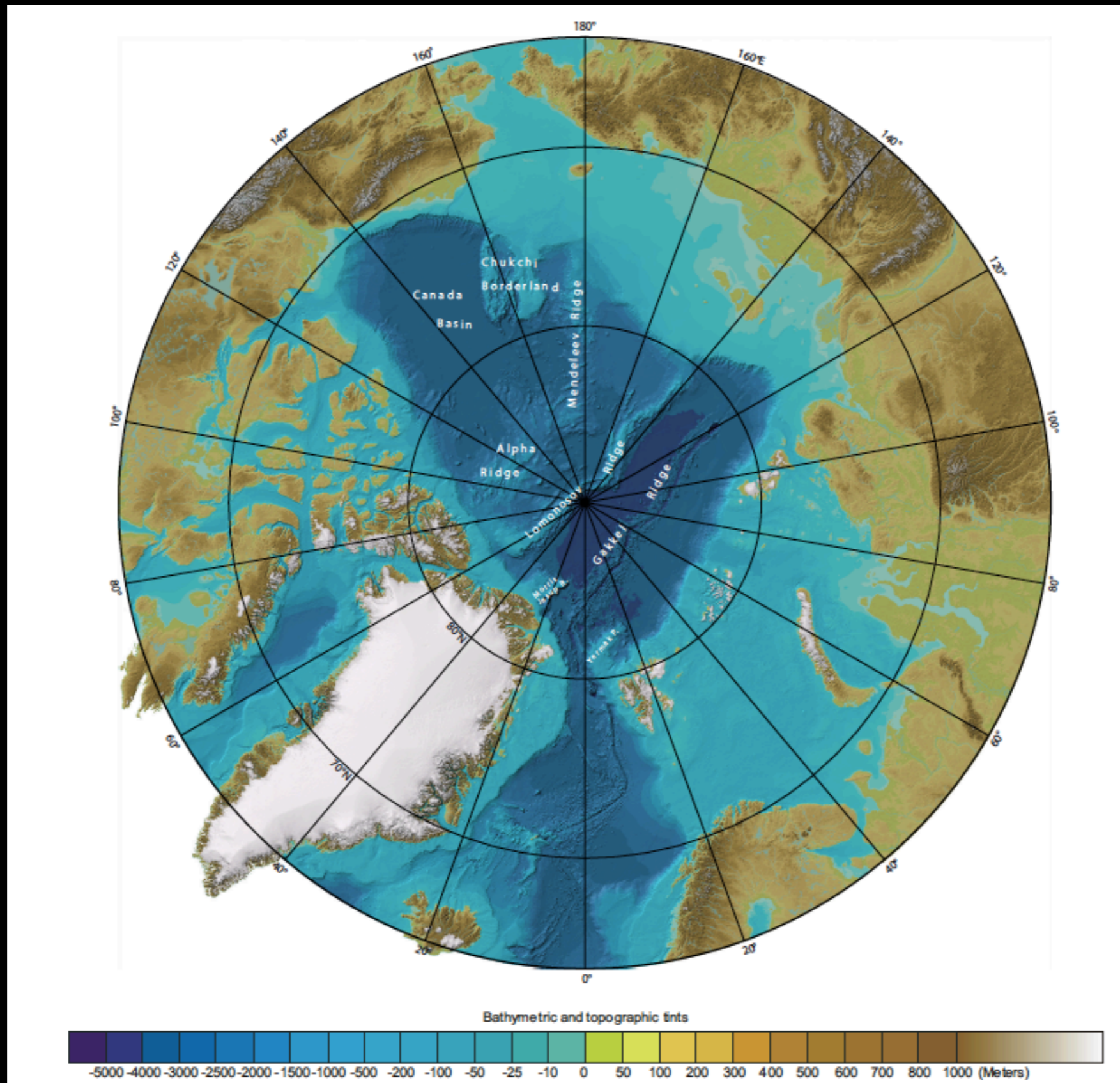


Gravity data from USCGC Healy





some people are really good at this...



Assignment -

Write a script to plot some data on a map.

Write a 2nd script to do an x, y plot of the same data.

suggestion - use awk to scale or select column data for your plots

Next time, data analysis
and gridding, a lot of gridding with GMT