Final Projects - Posters

Printing: Send to Andrew Phillips (andrew.g.phillips@nmt.edu) with the information he needs, listed on webpage: https://nmtearth.com/wide-format-plotter/

Size: 3 or 4 ft wide, 5-7 ft long (unless you are re-using the poster for another meeting, which in that case, print in meeting-required size)

Note that you are printing for this class (so you do not need a separate account number)

Timing: Poster session during lab on Dec 4

Andrew needs time to print the posters and he will not be available the afternoon of Dec 1. He requests getting poster files to him no later than 10 am on Dec 1.

GMT Part 1:
Overview and Simple Figures

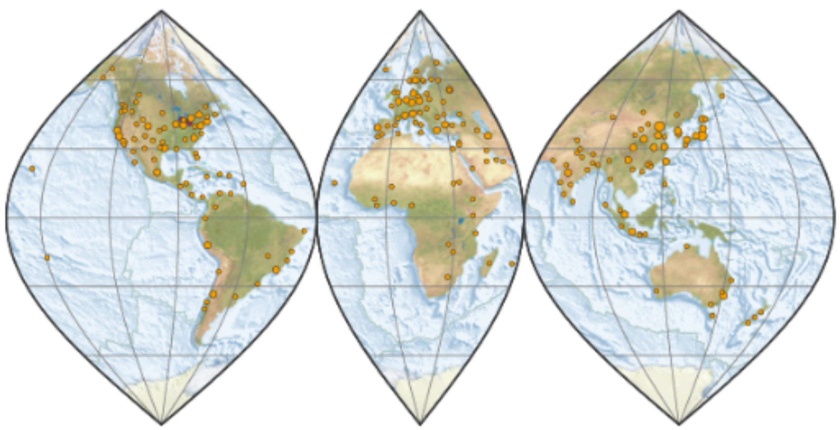
November 20, 2017

What is GMT?

GMT is an open source collection of about 80 command-line tools for manipulating geographic and Cartesian data sets (including filtering, trend fitting, gridding, projecting, etc.) and producing PostScript illustrations ranging from simple x-y plots via contour maps to artificially illuminated surfaces and 3D perspective views; the GMT supplements add another 40 more specialized and discipline-specific tools. GMT supports over 30 map projections and transformations and requires support data such as GSHHG coastlines, rivers, and political boundaries and optionally DCW country polygons. GMT is developed and maintained by Paul Wessel, Walter H. F. Smith, Remko Scharroo, Joaquim Luis and Florian Wobbe, with help from a global set of volunteers, and is supported by the National Science Foundation. It is released under the GNU Lesser General Public License version 3 or any later version.

THE GENERIC MAPPING TOOLS

The GMT World Domination



HTTP downloads (1981) · FTP downloads (0) · Subversion checkouts (7) · 1–9 · 10–99 · Total: 1988 · 100–999 · 1000+ · 2017–07–27 06:20:02 UTC

Considering its flexibility at no charge, people worldwide are using GMT in their work and at home. Most users of GMT are Earth or ocean scientists, but there are apparently no limits to the kind of applications that may benefit from GMT: We know GMT is used in medical research, engineering, physics, mathematics, social and biological sciences, and by geographers, fisheries institutes, oil companies, a wide range of government agencies, and last but not least innumerable hobbyists.

The map above illustrates the spreading of the current GMT release around the world based on web traffic. Each colored circle in the map above represents a 15x15 arc minute block with one or more users who downloaded GMT. Download geolocation is based on MaxMind's freely available GeoLite data.

Latest Releases

GMT-5.4.2 (2017-06-25)
↳Documentation (HTML)
GMT-4.5.16 (2017-06-25)
↳Documentation

Resources

Installing instructions
Building instructions
Mailing Lists
Tidbits
Podcasts
References

Wiki

Start page
Index by title
Index by date

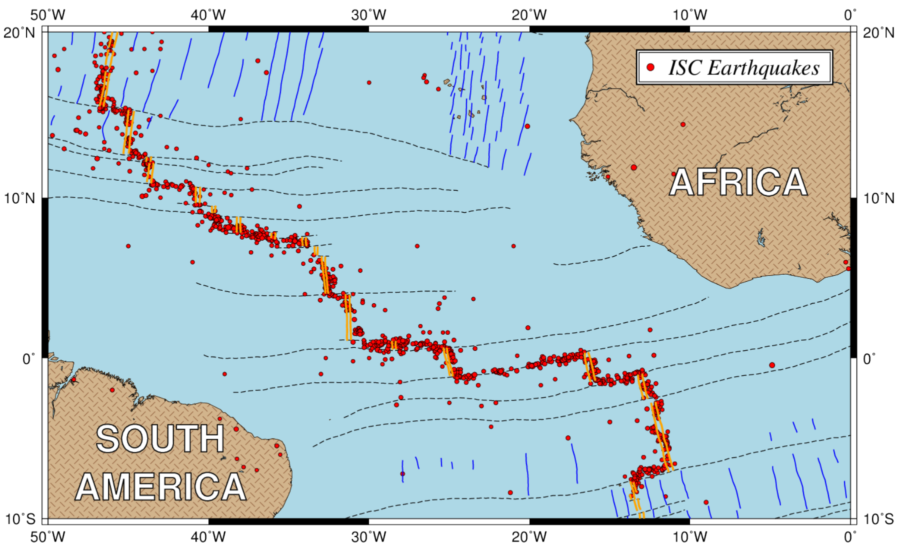
From GMT webpage:
http://gmt.soest.hawaii.edu

You will spend time at this page!!

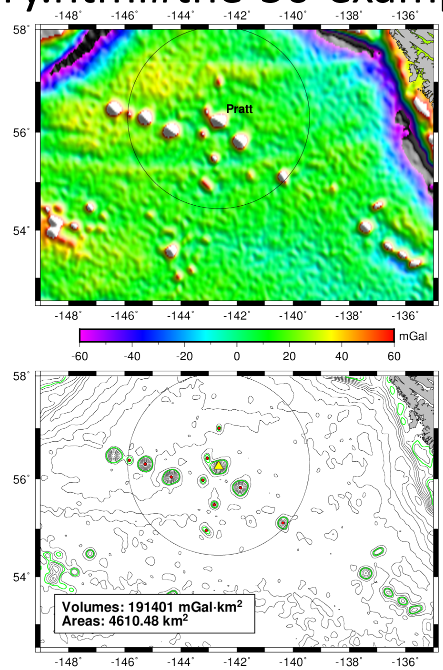
Samples (from the GMT webpage)

Bonus – you can get the entire script for these too….

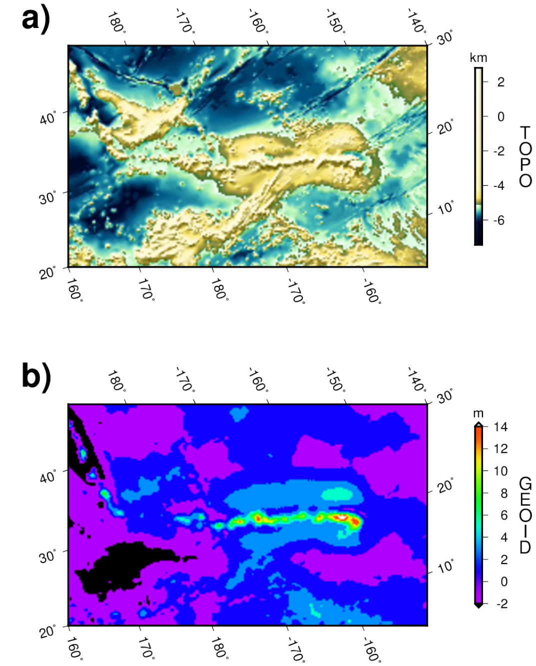
http://gmt.soest.hawaii.edu/doc/5.4.2/Gallery.html#the-50-examples



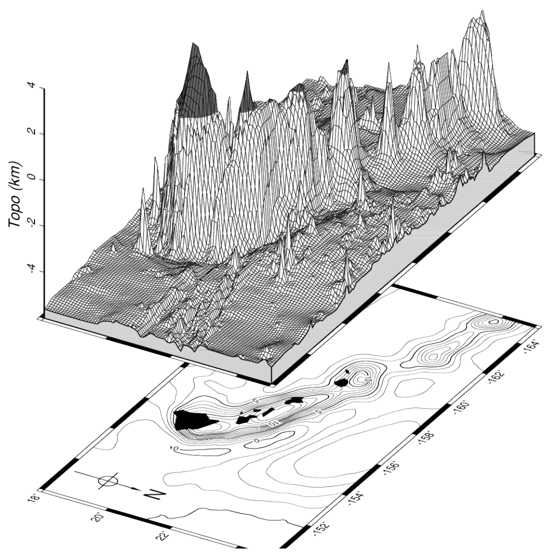
HAWAIIAN RIDGE

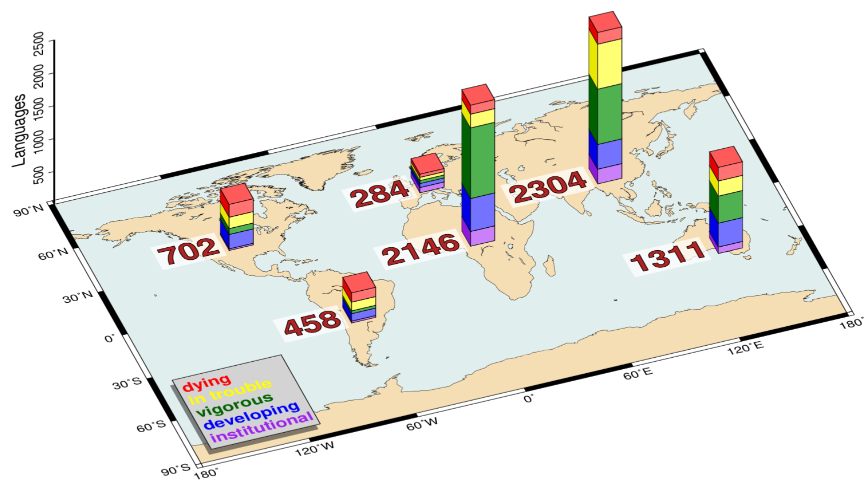


HAWAIIAN TOPO AND GEOID



World Languages By Continent





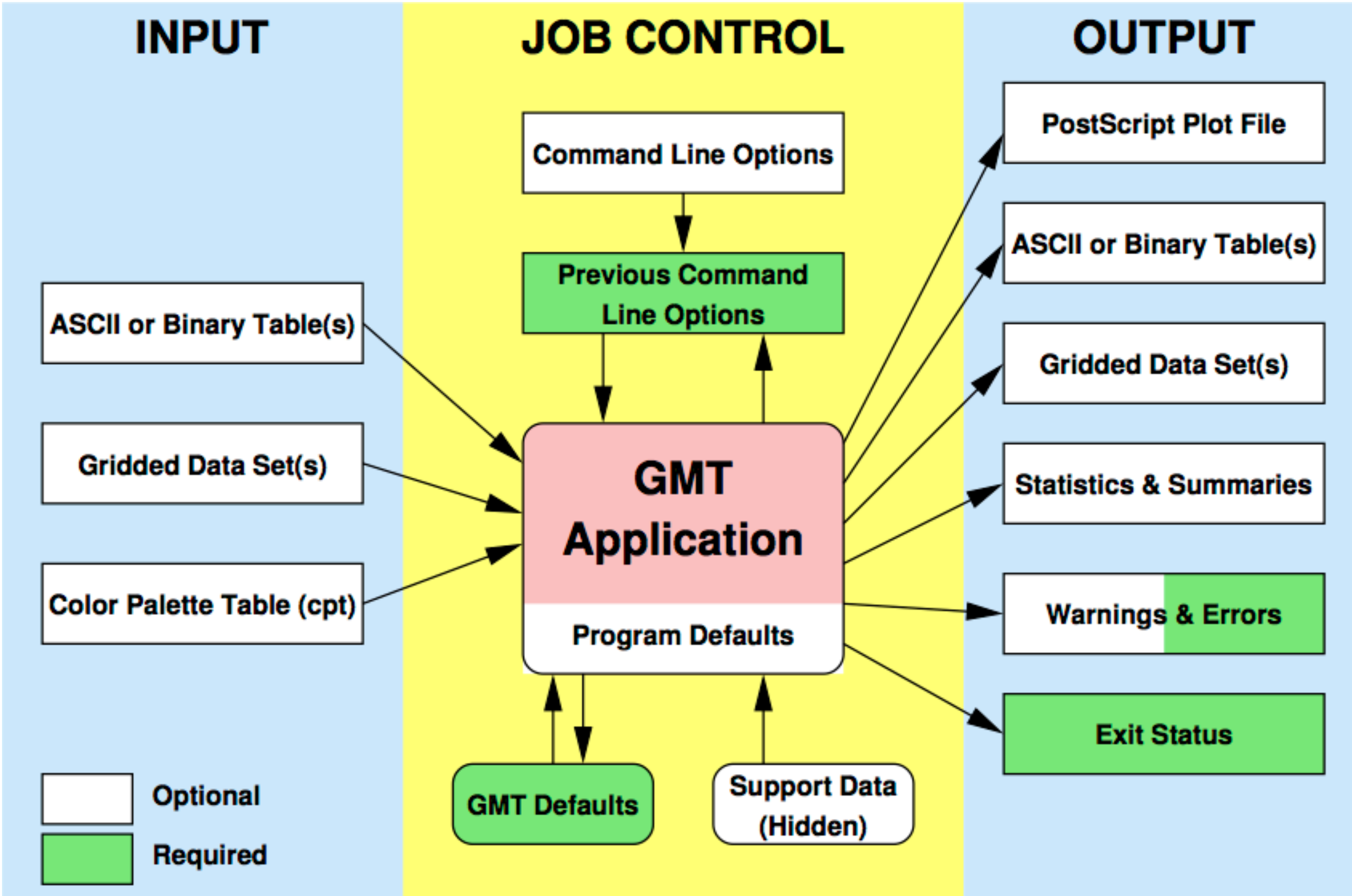
Next 2 weeks of GMT

- Today: Basics and Creating Simple Plots
  - Accessing commands
  - Parameter setting
  - Writing scripts
  - Basic maps and x-y plots
- Next week: More complexity
  - Contouring
  - Other datasets

GMT Background

- Started in 1987 by grad students at Lamont
- Wider use in 1990s after publications in EOS/AGU
- Current version is GMT 5
  - Major upgrade to previous versions, recommend starting here rather than with older versions
- Follows UNIX style – break down tasks into small components, build final product through combining several modules
- Command-line usage, used as with any other UNIX tools
  - ~80 tools with many customizable parameters
- Available on any type of UNIX OS (Linux, MacOS, Cygwin)

How does GMT run?



From GMT tutorial pages…

GMT Modules

- Basic modules for plotting 1D and 2D data

grdcontour	Contouring of 2-D gridded data sets
grdimage	Produce images from 2-D gridded data sets
grdvector	Plotting of 2-D gridded vector fields
grdview	3-D perspective imaging of 2-D gridded data sets
psbasemap	Create a basemap plot
psclip	Use polygon files to define clipping paths
pscoast	Plot (and fill) coastlines, borders, and rivers on maps
pscontour	Contour or image raw table data by triangulation
pshistogram	Plot a histogram
psimage	Plot Sun raster files on a map
pslegend	Plot a legend on a map
psmask	Create overlay to mask out regions on maps
psrose	Plot sector or rose diagrams
psscale	Plot gray scale or color scale on maps
psternary	Plot data on ternary diagrams
pstext	Plot text strings on maps
pswiggle	Draw table data time-series along track on maps
psxy	Plot symbols, polygons, and lines on maps
psxyz	Plot symbols, polygons, and lines in 3-D

Types of Map Projections

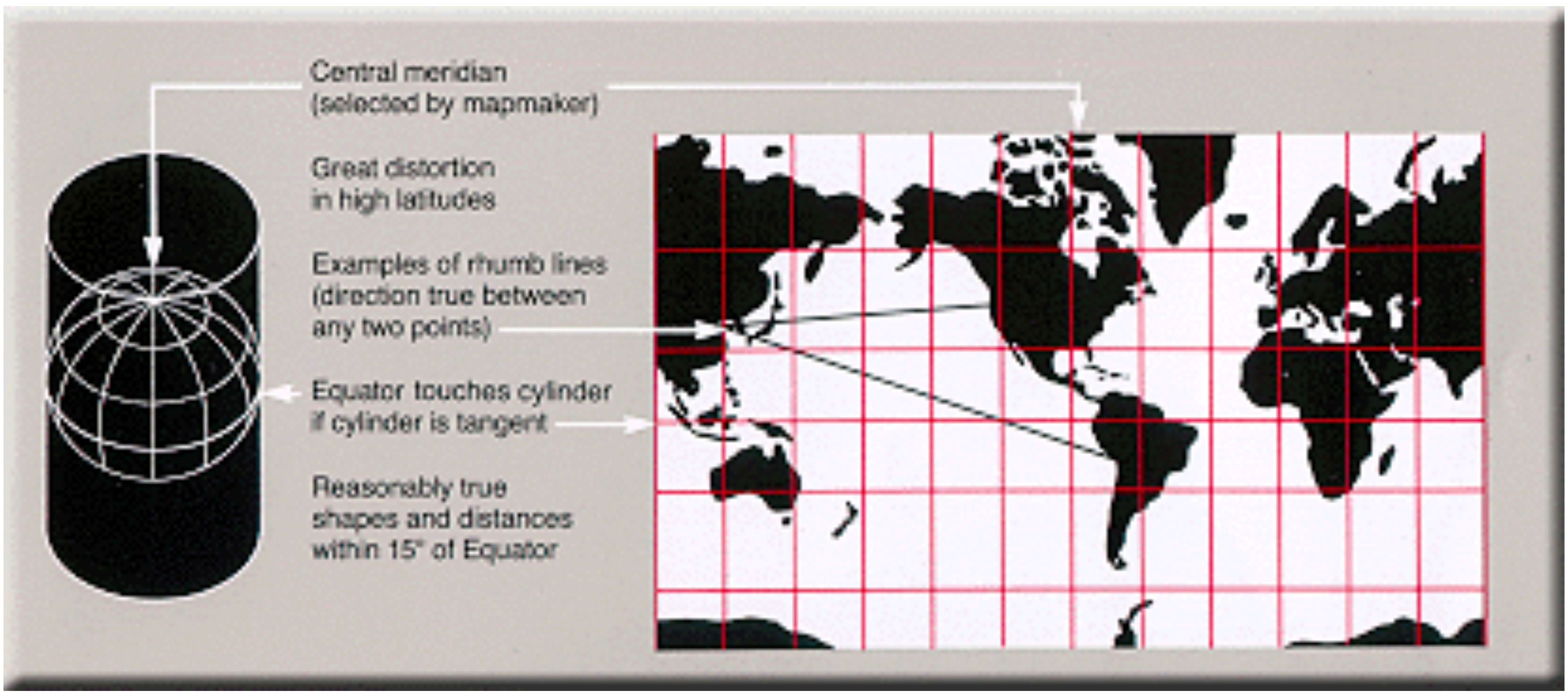
- Need to project spherical surface (globe) onto flat map
  - Will get distortions
  - Different projections will handle distortion in different ways

- GMT has over 30 different projections, falling in these basic categories
  - Conic map projections
  - Azimuthal map projections
  - Cylindrical map projections
  - Miscellaneous projections

https://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj.html

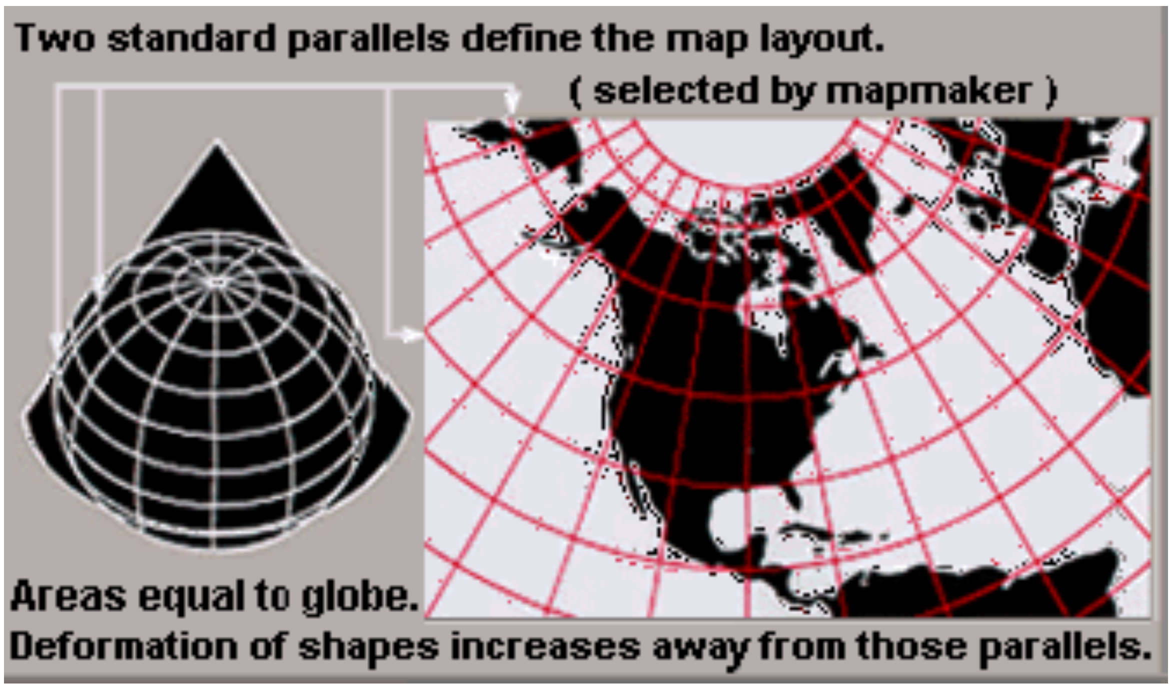
Some examples:

- Mercator: conformal (maintains shapes) and cylindrical. No distortion at equator, grows large at higher latitudes



Some examples:

- Conic: equal area projection with meridians equally spaced radii about a center point



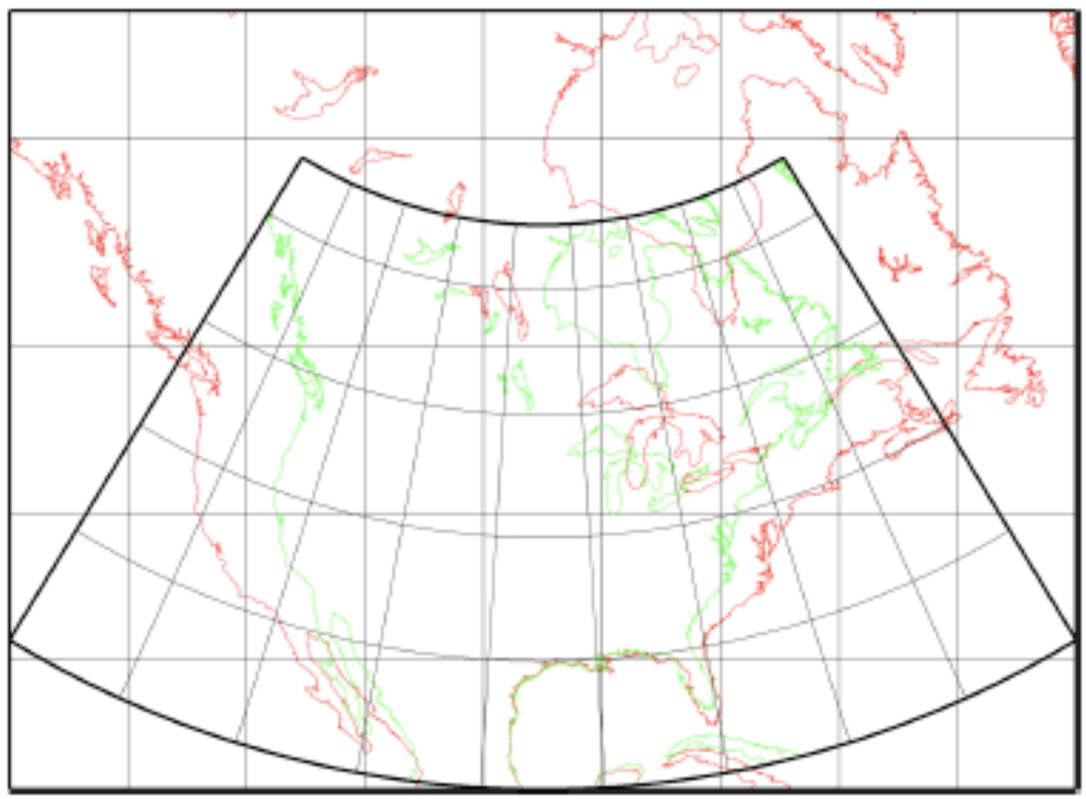
Albers conic

Lots more available….

What projection to use?

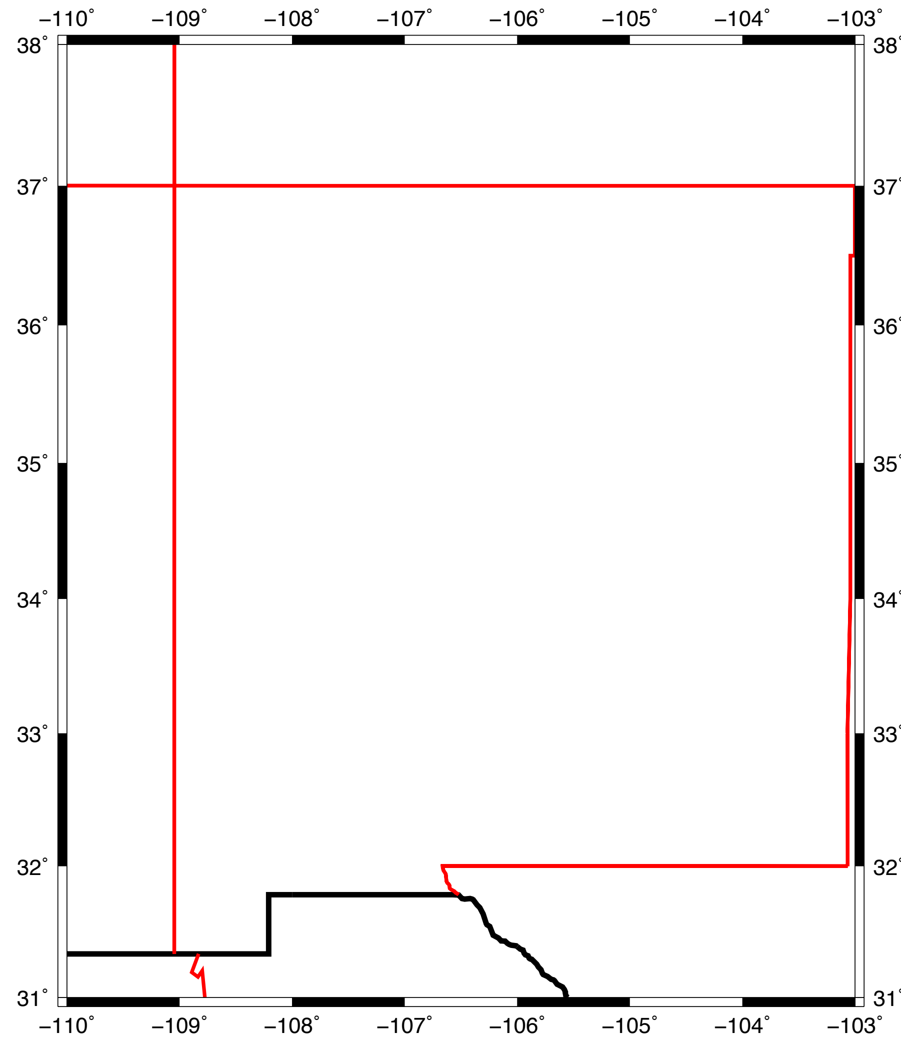
- Region near equator (tropics) – use cylindrical
- Near poles – use azimuthal projection
- In between – conical projection

GMT – red is Mercator, green is Lambert conic



Basic Map

>>> gmt pscoast -JM6i -R-110/-103/31/38 -Ba -Dh -N1/3 -N2/2,red -P > nm_lecture.ps



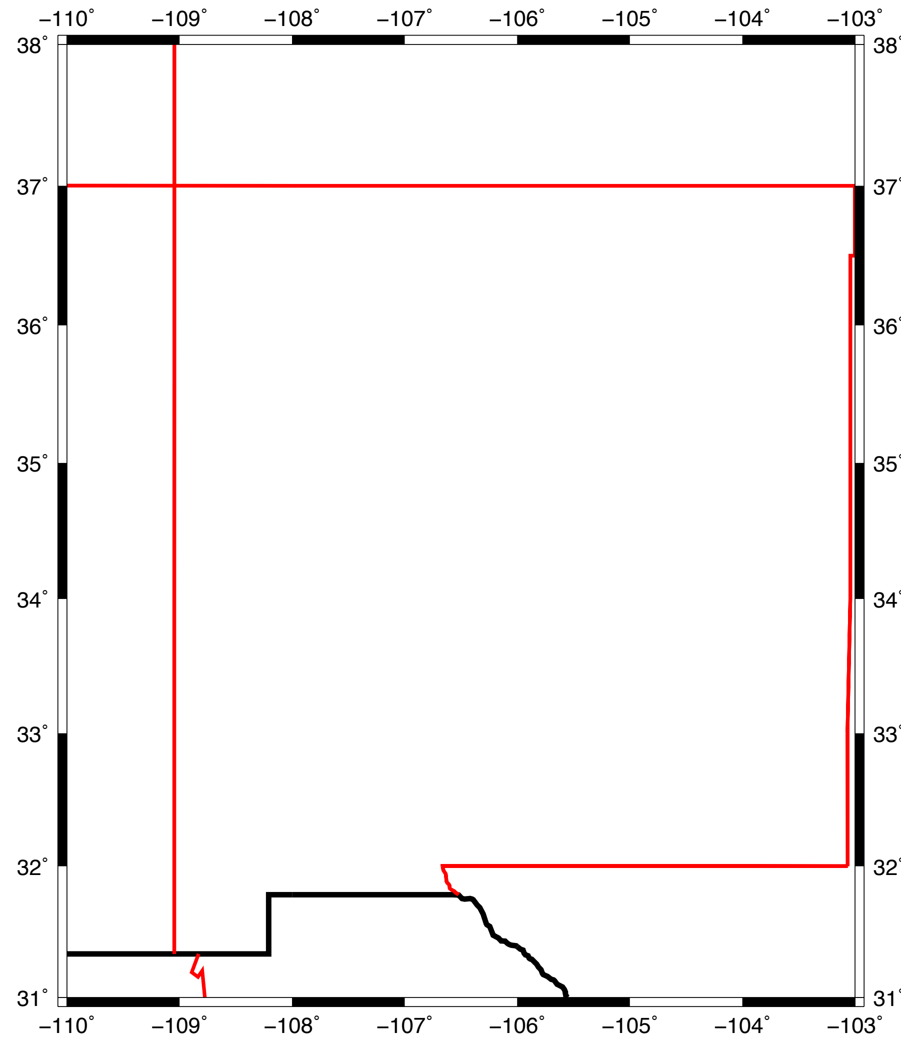
Basic Map

>>> gmt pscoast -JM6i -R-110/-103/31/38 -Ba -Dh -N1/3 -N2/2,red -P > nm_lecture.ps

>>> gmt pscoast   calls the pscoast module, -… is parameter setting, > sends to output file

Parameters:
-J defines projection (Mercator, 6 inch size)
-R defines geographic boundaries of map
-B defines annotation of map (default increment)
-D defines resolution of boundaries
-N defines how to plot political boundaries
-P defines orientation of map, here portrait

> nm_lecture.ps   creates new file and writes output of the command to that file



Common Options

STANDARDIZED COMMAND LINE OPTIONS

-B*information*	Specify map frame and axes parameters (…)
-J*parameters*	Select map projection (…)
-K	Append more PS later (…)
-O	This is an overlay plot (…)
-P	Select Portrait orientation (…)
-R*west/east/south/north*[*/zmin/zmax*][**+r**]	Specify Region of interest (…)
-U[[*just*]/*dx*/*dy*/][*label*]	Plot time-stamp on plot (…)
-V	Run in verbose mode (…)
-X[**a**|**c**|**r**]*off*[**u**]	Shift plot origin in *x*-direction (…)
-Y[**a**|**c**|**r**]*off*[**u**]	Shift plot origin in *y*-direction (…)
-a*name=col,…*	Associates aspatial data with columns (…)
-b[**i**|**o**][*ncol*][**t**]	Select binary input or output (…)
-d[**i**|**o**]*nodata*	Replace columns with *nodata* with NaN (…)
-e[~]*"pattern"* | **-e**[~]/*regexp*/[**i**]	Filter data records that match the given pattern (…)
-f[**i**|**o**]*colinfo*	Set formatting of ASCII input or output (…)
-g[**+**]**x**|**X**|**y**|**Y**|**d**|**D***gap*[**u**]	Segment data by detecting gaps (…)
-h[**i**|**o**][*n_headers*]	ASCII [*I*|*O*] tables have header record[s] (…)
-i*columns*	Selection of input columns (…)
-o*columns*	Selection of output columns (…)
-n[*type*][**+a**][**+b***BC*] [**+c**][**+t***threshold*]	Set grid interpolation mode (…)
-p*azim*[*elev*[*/zlevel*]][**+w***lon0/lat0*[*/z0*]][**+v***x0/y0*]	Control 3-D perspective view (…)
-r	Sets grid registration (…)
-s[**z**|*cols*]	Control treatment of NaN records (…)
-t*transparency*	Set layer PDF transparency (…)
-x[[-]*n*]	Set number of cores in multi-threaded modules (…)
-:[**i**|**o**]	Expect *y*/*x* input rather than *x*/*y* (…)

You will need the manual pages!!!!! (reading patiently + carefully)

This does tell you that something like this:

-Ba1f0.25WeSn

Means to annotate every 1 degree and put ticks in the frame every 0.25 degree, with annotations plotted only on the left (W) and bottom (S) of the plot

```
-B Specify both (1) basemap frame settings and (2) axes parameters.
   Frame settings are modified via an optional single invocation of
     -B[<axes>][+b][+g<fill>][+n][+o<lon>/<lat>][+t<title>]
   Axes parameters are specified via one or more invocations of
     -B[p|s][x|y|z]<info>

   1. Frame settings control which axes to plot, frame fill, title, and type of gridlines:
     <axes> is a combination of W,E,S,N,Z and plots those axes only [Default is WESNZ (all)].
     Use lower case w,e,s,n,z just to draw and tick (but not annotate) those axes.
     For 3-D plots the Z|z[<corners>][+b] controls the vertical axis.  The <corners> specifies
     at which corner(s) to erect the axis via a combination of 1,2,3,4; 1 means lower left corner,
     2 is lower right, etc., in a counter-clockwise order. [Default automatically selects one axis].
     The optional +b will erect a 3-D frame box to outline the 3-D domain [no frame box]. The +b
     is also required for x-z or y-z gridlines to be plotted (if such gridlines are selected below).
     Append +g<fill> to paint the inside of the map region before further plotting [no fill].
     Append +n to have no frame and annotations whatsoever [Default is controlled by WESNZ/wesnz].
     Append +o<plon>/<plat> to draw oblique gridlines about this pole [regular gridlines].
     Note: the +o modifier is ignored unless gridlines are specified via the axes parameters (below).
     Append +t<title> to place a title over the map frame [no title].
   2. Axes settings control the annotation, tick, and grid intervals and labels.
     The full axes specification is
       -B[p|s][x|y|z]<intervals>[+l|L<label>][+p<prefix>][+u<unit>]
     Alternatively, you may break this syntax into two separate -B options:
       -B[p|s][x|y|z][+l|L<label>][+p<prefix>][+u<unit>]
       -B[p|s][x|y|z]<intervals>
     There are two levels of annotations: Primary and secondary (most situations only require primary).
     The -B[p] selects (p)rimary annotations while -Bs specifies (s)econdary annotations.
     The [x|y|z] selects which axes the settings apply to.  If none are given we default to xy.
     To specify different settings for different axes you must repeat the -B axes option for
     each dimension., i.e., provide separate -B[p|s]x, -B[p|s]y, and -B[p|s]z settings.
     To prepend a prefix to each annotation (e.g., $ 10, $ 20 ...), add +p<prefix>.
     To append a unit to each annotation (e.g., 5 km, 10 km ...), add +u<unit>.
     To label an axis, add +l<label>.  Use +L to enforce horizontal labels for y-axes.
     Use quotes if any of the <label>, <prefix> or <unit> have spaces.
     Geographic map annotations will automatically have degree, minute, seconds units.
     The <intervals> setting controls the annotation spacing and is a textstring made up of one or
     more substrings of the form [a|f|g][<stride>[+-<phase>][<unit>]], where the (optional) a
     indicates annotation and major tick interval, f minor tick interval, and g grid interval.
     Here, <stride> is the spacing between ticks or annotations, the (optional)
     <phase> specifies phase-shifted annotations/ticks by that amount, and the (optional)
     <unit> specifies the <stride> unit [Default is the unit implied in -R]. There can be
     no spaces between the substrings; just append items to make one very long string.
     For custom annotations or intervals, let <intervals> be c<intfile>; see man page for details.
     The optional <unit> modifies the <stride> value accordingly.  For geographic maps you may use
       d: arc degree [Default].
       m: arc minute.
       s: arc second.
     For time axes, several units are recognized:
       Y: year - plot using all 4 digits.
       y: year - plot only last 2 digits.
       O: month - format annotation according to FORMAT_DATE_MAP.
       o: month - plot as 2-digit integer (1-12).
       U: ISO week - format annotation according to FORMAT_DATE_MAP.
       u: ISO week - plot as 2-digit integer (1-53).
       r: Gregorian week - 7-day stride from chosen start of week (Monday).
       K: ISO weekday - format annotation according to FORMAT_DATE_MAP.
       k: weekday - plot name of weekdays in selected language [us].
       D: day - format annotation according to FORMAT_DATE_MAP, which also determines whether
                we should plot day of month (1-31) or day of year (1-366).
       d: day - plot as 2- (day of month) or 3- (day of year) integer.
       R: Same as d but annotates from start of Gregorian week.
       H: hour - format annotation according to FORMAT_CLOCK_MAP.
       h: hour - plot as 2-digit integer (0-23).
       M: minute - format annotation according to FORMAT_CLOCK_MAP.
       m: minute - plot as 2-digit integer (0-59).
       S: second - format annotation according to FORMAT_CLOCK_MAP.
       s: second - plot as 2-digit integer (0-59; 60-61 if leap seconds are enabled).
     Cartesian axes takes no units.
     When <stride> is omitted, a reasonable value will be determined automatically, e.g., -Bafg.
     Log10 axis: Append l to annotate log10 (value) or p for 10^(log10(value)) [Default annotates value].
     Power axis: Append p to annotate value at equidistant pow increments [Default is nonlinear].
     See psbasemap man pages for more details and examples of all settings.
```

Scripting with GMT

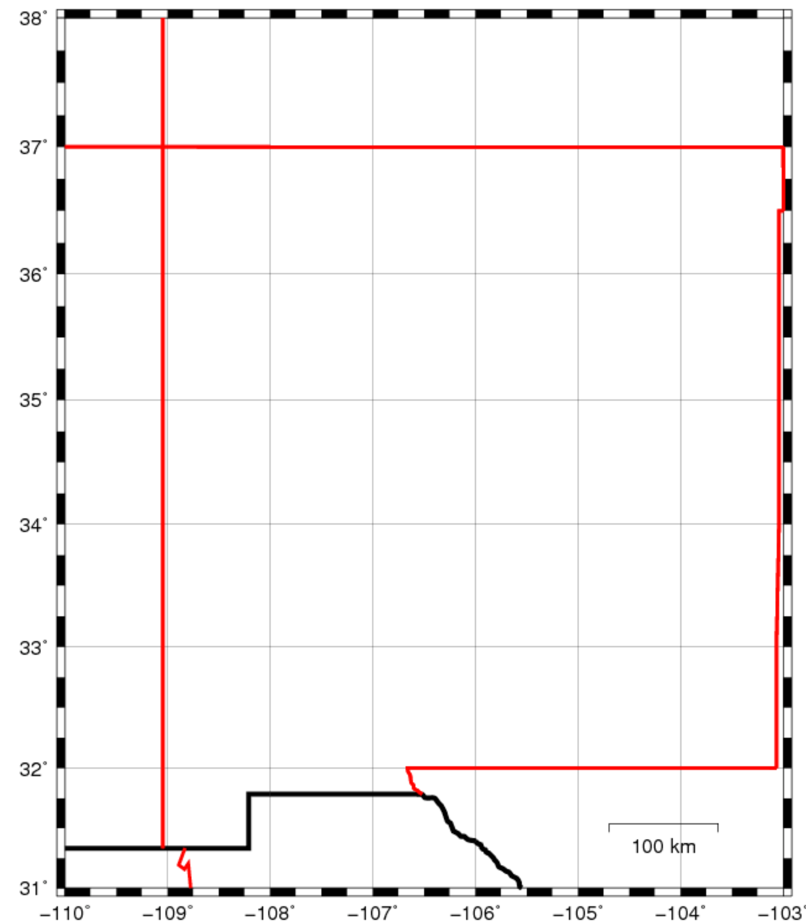
- Best to build GMT commands into a script
- Allows you to define variables to be used in parameter setting (cut down on typing), reproduce figures, easily change/update them with new data, etc

```
#!/bin/bash

#Sue's sample script for making a NM basemap
#Uses GMT5 syntax

rm gmt.history

##############################
# SETUP REGIONAL PARAMETERS, PROJECTION, AND OUTPUT

#set regional bounding box
w_lon=-110
e_lon=-103
n_lat=38
s_lat=31

#set projection and scale
proj=M6i

#set output file
outfile=nm_basic.ps

###############################
#REST OF MAP

#defines a basemap using region of state of NM, plot latitude and longitude at
#0.25 degree increments and annotate every degree on the left and bottom of the frame
gmt psbasemap -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Ba1g1f0.25WeSn -Lx5i/0.5i+c32+w100k -P -K > $outfile

#plot national boundaries in black, state boundaries in red
gmt pscoast -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Dh -N1/3 -N2/2,red -P -O  >> $outfile

#view output file using ghostscript
gs $outfile
```

Make executable using chmod +x then simply type filename on command line

Can get fancier by setting variables through user input, making it more interactive

```
#!/bin/bash

#Sue's sample script for making a NM basemap
#Uses GMT5 syntax

rm gmt.history

############################
# SETUP REGIONAL PARAMETERS, PROJECTION, AND OUTPUT

#set regional bounding box
w_lon=-110
e_lon=-103
n_lat=38
s_lat=31

#set projection and scale
proj=M6i

#set output file
outfile=nm_basic.ps

##############################
#REST OF MAP

#defines a basemap using region of state of NM, plot latitude and longitude at
#0.25 degree increments and annotate every degree on the left and bottom of the frame
gmt psbasemap -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Ba1g1f0.25WeSn -Lx5i/0.5i+c32+w100k -P -K > $outfile

#plot national boundaries in black, state boundaries in red
gmt pscoast -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Dh -N1/3 -N2/2,red -P -O  >> $outfile

#view output file using ghostscript
gs $outfile
```



New things added:

psbasemap: creates the basemap with annotation, gridlines, and simple scale bar

Then pscoast command to plot the geographic (here national/state) boundaries

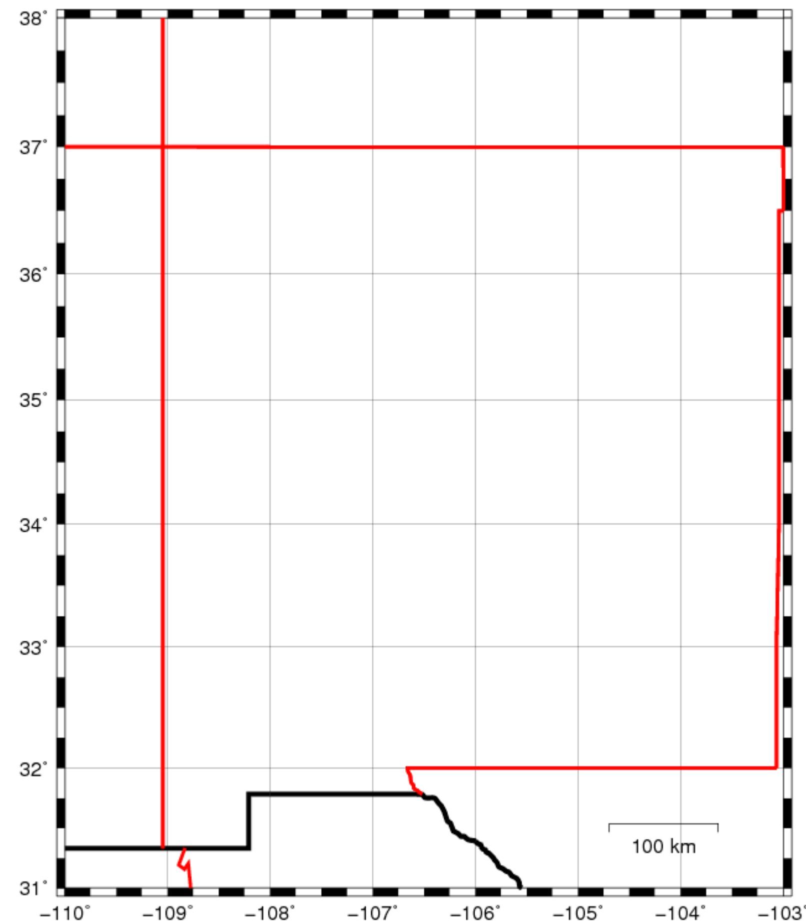
Remember – order of commands matters (think layering)

Adding Data to Plots

- psxy

```
#!/bin/bash

#Sue's sample script for making a NM basemap
#Uses GMT5 syntax

rm gmt.history

##############################
# SETUP REGIONAL PARAMETERS, PROJECTION, AND OUTPUT

#set regional bounding box
w_lon=-110
e_lon=-103
n_lat=38
s_lat=31

#set projection and scale
proj=M6i

#set output file
outfile=nm_basic.ps

###############################
#REST OF MAP

#defines a basemap using region of state of NM, plot latitude and longitude at
#0.25 degree increments and annotate every degree on the left and bottom of the frame
gmt psbasemap -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Ba1g1f0.25WeSn -Lx5i/0.5i+c32+w100k -P -K > $outfile

#plot national boundaries in black, state boundaries in red
gmt pscoast -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Dh -N1/3 -N2/2,red -P -O -K >> $outfile

#plot a few seismic stations in green triangles
gmt psxy -J$prog -R$w_lon/$e_lon/$s_lat/$n_lat -St0.3 -Ggreen -P -O << END >> $outfile
34.9500 -106.4600
33.9525 -106.7340
34.0722 -106.9460
32.4914 -104.5150
END

#view output file using ghostscript
gs $outfile
```

```
#!/bin/bash

#Sue's sample script for making a NM basemap
#Uses GMT5 syntax

rm gmt.history

#############################
# SETUP REGIONAL PARAMETERS, PROJECTION, AND OUTPUT

#set regional bounding box
w_lon=-110
e_lon=-103
n_lat=38
s_lat=31

#set projection and scale
proj=M6i

#set output file
outfile=nm_basic.ps

##############################
#REST OF MAP

#defines a basemap using region of state of NM, plot latitude and longitude at
#0.25 degree increments and annotate every degree on the left and bottom of the frame
gmt psbasemap -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Ba1g1f0.25WeSn -Lx5i/0.5i+c32+w100k -P -K > $outfile

#plot national boundaries in black, state boundaries in red
gmt pscoast -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Dh -N1/3 -N2/2,red -P -O -K >> $outfile

#plot a few seismic stations in green triangles
gmt psxy -J$prog -R$w_lon/$e_lon/$s_lat/$n_lat -St0.3 -Ggreen -P -O << END >> $outfile
34.9500 -106.4600
33.9525 -106.7340
34.0722 -106.9460
32.4914 -104.5150
END

#view output file using ghostscript
gs $outfile
```



But where are my green triangles?

Read the manual pages!!!!

- Format is important.....

PSXY(1) Generic Mapping Tools PSXY(1)

NAME

psxy - Plot lines, polygons, and symbols on maps

SYNOPSIS

psxy files **-J**parameters **-R**west/east/south/north[**r**] [**-A**[**m**|**p**]] [**-B**[**p**|**s**]parameters] [**-C**cptfile] [**-D**dx/dy] [**-E**[**x**|**y**|**X**|**Y**][**n**][cap][/[-|+]pen]] [**-G**fill] [**-H**[**i**][nrec]] [**-I**intens] [**-K**] [**-L**] [**-N**] [**-M**[flag]] [**-O**] [**-P**] [**-S**[symbol][size]] [**-U**[just/dx/dy/][**c**|label]] [**-V**] [**-W**[-|+][pen]] [**-X**[**a**|**c**|**r**][x-shift[**u**]]] [**-Y**[**a**|**c**|**r**][y-shift[**u**]]] [**-:**[**i**|**o**]] [**-c**copies] [**-bi**[**s**|**S**|**d**|**D**[ncol]|**c**[var1/...]]] [**-f**colinfo]

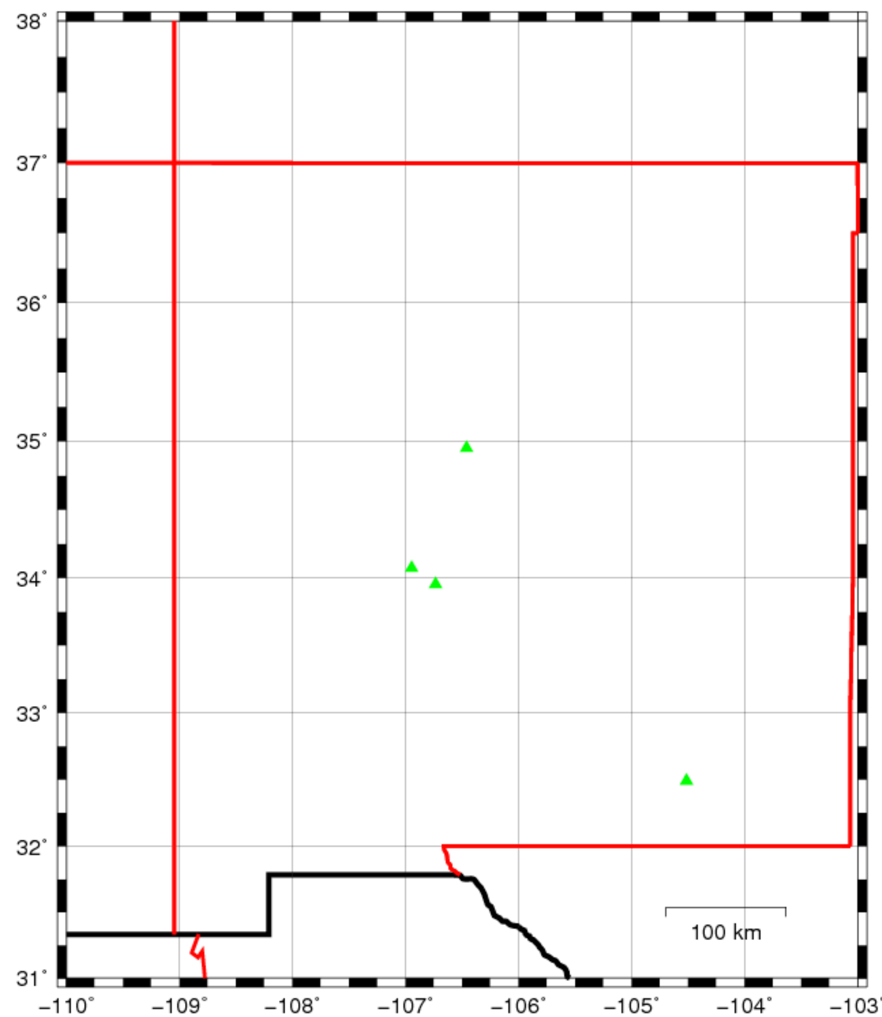
DESCRIPTION

psxy reads (x,y) pairs from files [or standard input] and generates PostScript code that will plot lines, polygons, or symbols at those locations on a map. If a symbol is selected and no symbol size given, then **psxy** will interpret the third column of the input data as symbol size. Symbols whose size is <= 0 are skipped. If no symbols are specified then the symbol code (see **-S** below) must be present as last column in the input. Multiple segment files may be plotted using the **-M** option. If **-S** is not used, a line connecting the data points will be drawn instead. To explicitly close polygons, use **-L**. Select a fill with **-G**. If **-G** is set, **-W** will control whether the polygon outline is drawn or not. If a symbol is selected, **-G** and **-W** determines the fill and outline/no outline, respectively. The PostScript code is written to standard output.

...

-: Toggles between (longitude,latitude) and (latitude,longitude) input and/or output. [Default is (longitude,latitude)]. Append **i** to select input only or **o** to select output only. [Default affects both].

```
#!/bin/bash

#Sue's sample script for making a NM basemap
#Uses GMT5 syntax

rm gmt.history

############################
# SETUP REGIONAL PARAMETERS, PROJECTION, AND OUTPUT

#set regional bounding box
w_lon=-110
e_lon=-103
n_lat=38
s_lat=31

#set projection and scale
proj=M6i

#set output file
outfile=nm_basic.ps

##############################
#REST OF MAP

#defines a basemap using region of state of NM, plot latitude and longitude at
#0.25 degree increments and annotate every degree on the left and bottom of the frame
gmt psbasemap -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Ba1g1f0.25WeSn -Lx5i/0.5i+c32+w100k -P -K > $outfile

#plot national boundaries in black, state boundaries in red
gmt pscoast -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Dh -N1/3 -N2/2,red -P -O -K >> $outfile

#plot a few seismic stations in green triangles
gmt psxy -J$prog -R$w_lon/$e_lon/$s_lat/$n_lat -St0.3 -Ggreen -: -P -O << END >> $outfile
34.9500 -106.4600
33.9525 -106.7340
34.0722 -106.9460
32.4914 -104.5150
END

#view output file using ghostscript
gs $outfile
```

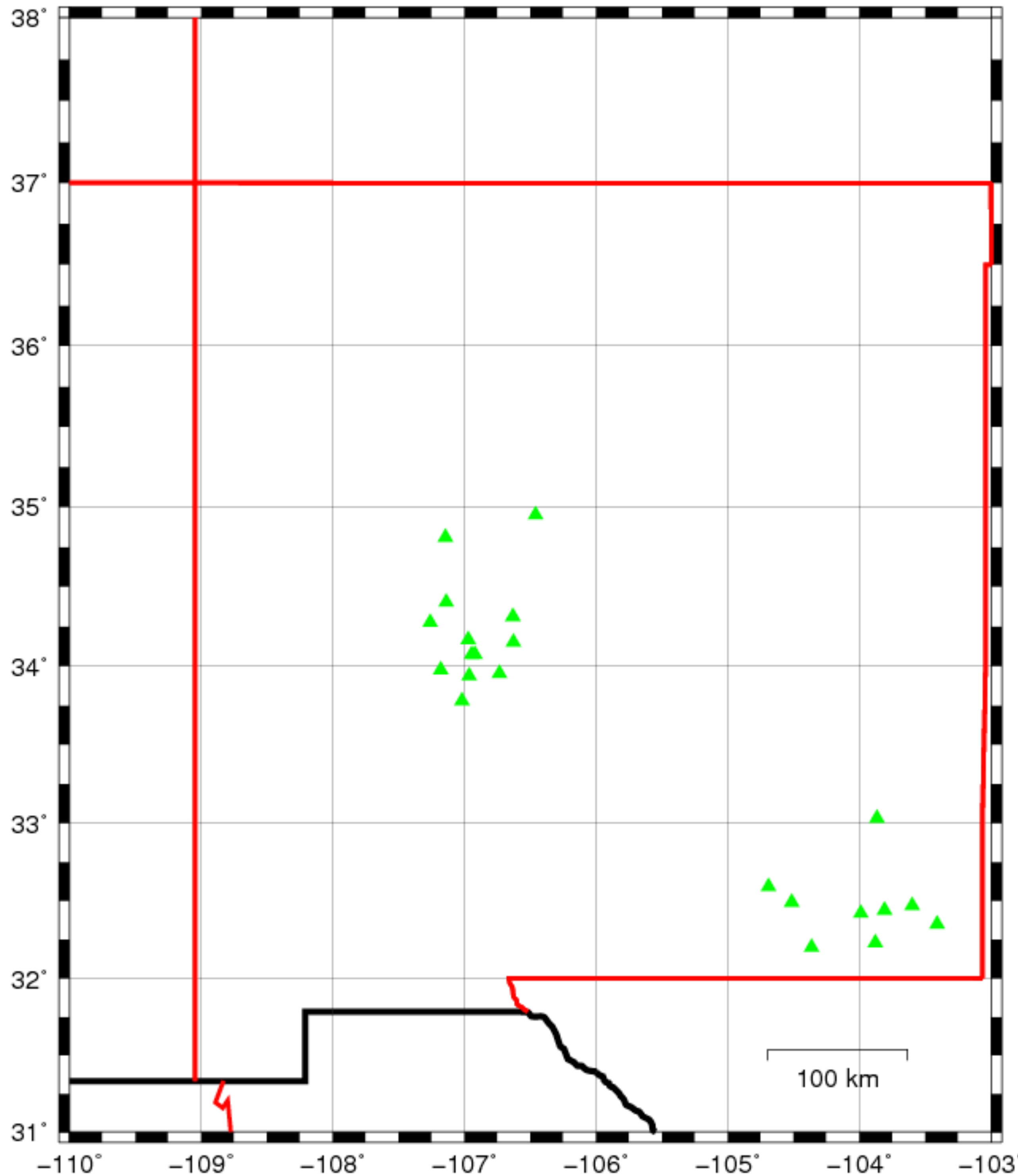


Other ways to add data….

- Long datasets already contained in a file… just read that in
- Remember format is important for psxy
- Use other tools at your disposal to format properly

```
#!/bin/bash

#Sue's sample script for making a NM basemap
#Uses GMT5 syntax

rm gmt.history

#############################
# SETUP REGIONAL PARAMETERS, PROJECTION, AND OUTPUT

#set regional bounding box
w_lon=-110
e_lon=-103
n_lat=38
s_lat=31

#set projection and scale
proj=M6i

#set output file
outfile=nm_basic.ps

##############################
#REST OF MAP

#defines a basemap using region of state of NM, plot latitude and longitude at
#0.25 degree increments and annotate every degree on the left and bottom of the frame
gmt psbasemap -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Ba1g1f0.25WeSn -Lx5i/0.5i+c32+w100k -P -K > $outfile

#plot national boundaries in black, state boundaries in red
gmt pscoast -J$proj -R$w_lon/$e_lon/$s_lat/$n_lat -Dh -N1/3 -N2/2,red -P -O -K >> $outfile

#plot seismic stations in green triangles
awk '{print $3, $2}' stations.txt | gmt psxy -J$prog -R$w_lon/$e_lon/$s_lat/$n_lat -St0.3 -Ggreen -h1 -P -O >> $outfile

#view output file using ghostscript
gs $outfile
```

```
Name Lat Lon Elevation Type Number
ANMO 34.9500 -106.4600 1820 1 2
BAR 34.1500 -106.6280 2121 1 3
BMT 34.2750 -107.2600 1987 1 4
CAR 33.9525 -106.7340 1658 1 5
CBET 32.4200 -103.9900 1042 1 6
CL2B 32.2300 -103.8800 2121 1 7
CL7 32.4400 -103.8100 1032 1 8
CPRX 33.0308 -103.8670 1356 1 9
DAG 32.5913 -104.6910 1277 1 10
GDL2 32.2003 -104.3640 1213 1 11
HTMS 32.4700 -103.6000 1192 1 12
LAZ 34.4020 -107.1390 1878 1 13
LEM 34.1660 -106.9720 1698 1 1
LPM 34.3117 -106.6320 1737 1 14
MLM 34.8100 -107.1450 2088 1 15
SBY 33.9752 -107.1810 3230 1 16
SMC 33.7787 -107.0190 1560 1 17
SRH 32.4914 -104.5150 1276 1 18
SSS 32.3500 -103.4100 1072 1 19
Y22A 33.9370 -106.9650 1674 1 20
Y22D 34.0739 -106.9210 1436 1 21
WTX 34.0722 -106.9460 1555 1 22
~
```


38˚
37˚
36˚
35˚
34˚
33˚
32˚
31˚
100 km
−110˚
−109˚
−108˚
−107˚
−106˚
−105˚
−104˚
−103˚

Other types of plots - xy

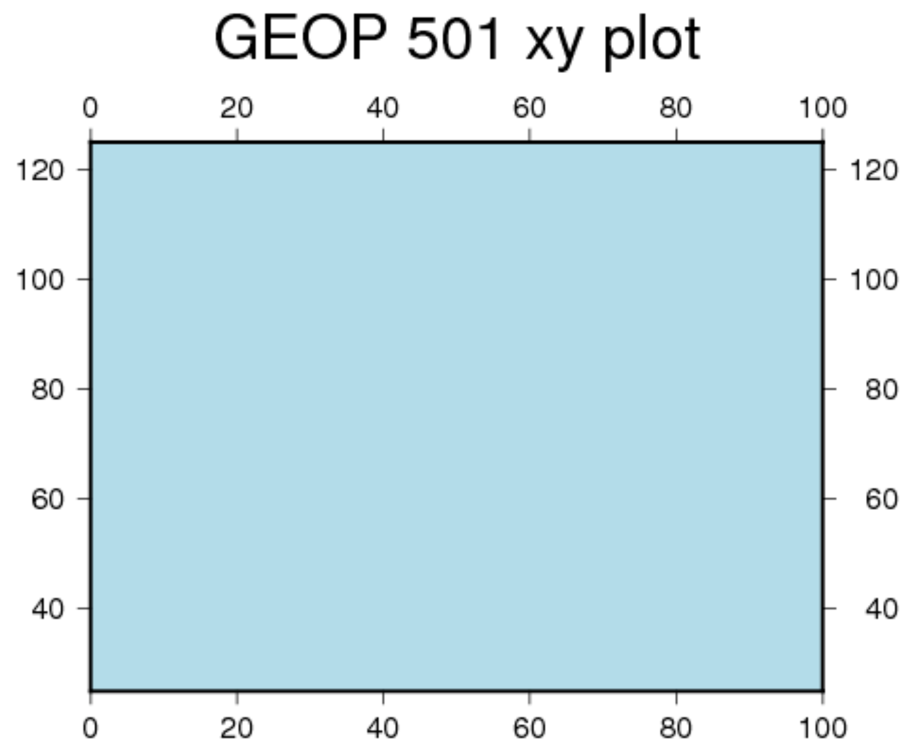
- Need to make “professional” quality x-y plots? Don’t use excel! Even matlab quality isn’t great
- –JX projection: linear projection
- Allows for setting of basemap axes in linear, log, exponential, time
- Then add data using psxy (expects dataset in x,y format, but can be changed with -:)

```
>>> gmt psbasemap -R0/100/25/125 -JX4i/3i -Ba -B+glightblue+t"GEOP 501 xy plot" -P > test_linear.ps
```

Next steps:

Put in a script, add a psxy command to plot some data

Note that GMT has some other really useful tools….



gmtregress

gmtregress - Linear regression of 1-D data sets

Synopsis

gmtregress [*table*] [-A*min*/*max*/*inc*] [-C*level*] [-**Ex**|**y**|**o**|**r**] [-F*flags*] [-**N1**|**2**|**r**|**w**] [-**S**[**r**]] [-T*min*/*max*/*inc* | -T*n*] [-**W**[**w**][**x**][**y**][**r**]] [-**V**[*level*]] [-**a**flags] [-**b**binary] [-**d**nodata] [-**e**regexp] [-**g**gaps] [-**h**headers] [-**i**flags] [-**o**flags]

Note: No space is allowed between the option flag and the associated arguments.

Description

gmtregress reads one or more data tables [or *stdin*] and determines the best linear regression model *y* = *a* + *b** *x* for each segment using the chosen parameters. The user may specify which data and model components should be reported. By default, the model will be evaluated at the input points, but alternatively you can specify an equidistant range over which to evaluate the model, or turn off evaluation completely. Instead of determining the best fit we can perform a scan of all possible regression lines (for a range of slope angles) and examine how the chosen misfit measure varies with slope. This is particularly useful when analyzing data with many outliers. Note: If you actually need to work with log10 of *x* or *y* you can accomplish that transformation during read by using the **-i** option.

«

Just one example…

Common GMT errors

- -K and –O problems
  - -K Don't close PostScript, use when more commands will follow
    - Use –K on all but last line of GMT script
  - -O Don't initialize PostScript, use when appending to pre-existing file (think overlay)
    - Use –O on all but first line of GMT script
- Using > instead of >> in later lines
- Problems with –R and –J definitions

Next time – More complex plotting and gridding in GMT

Will be the last lecture + lab of the course
Following week will be final poster session