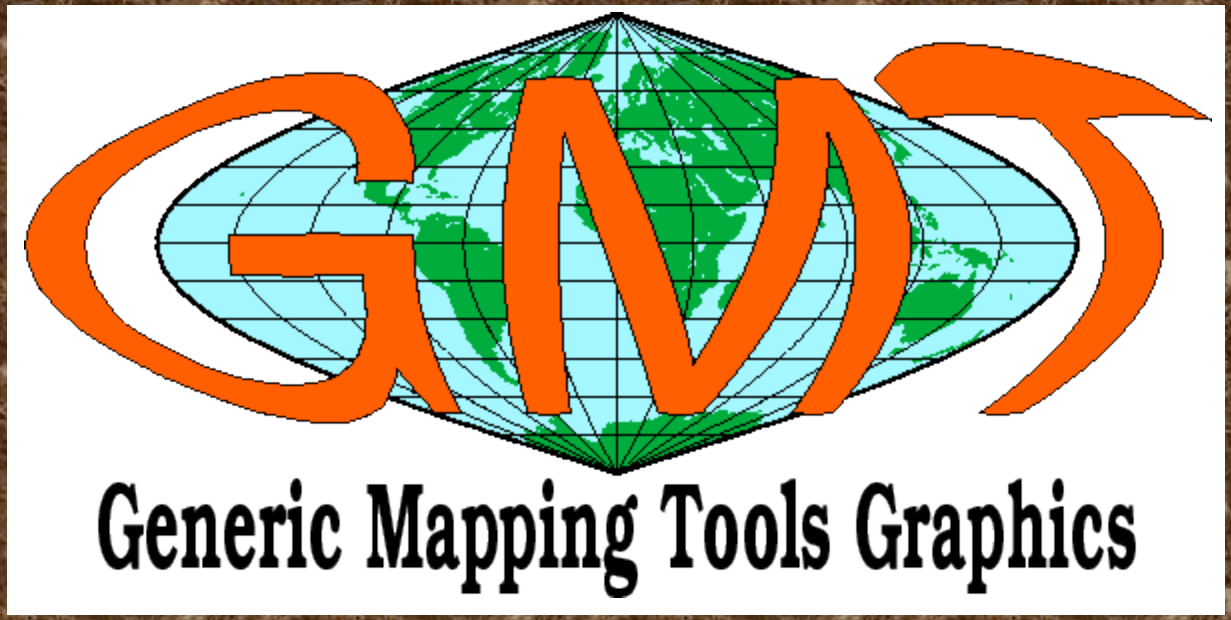
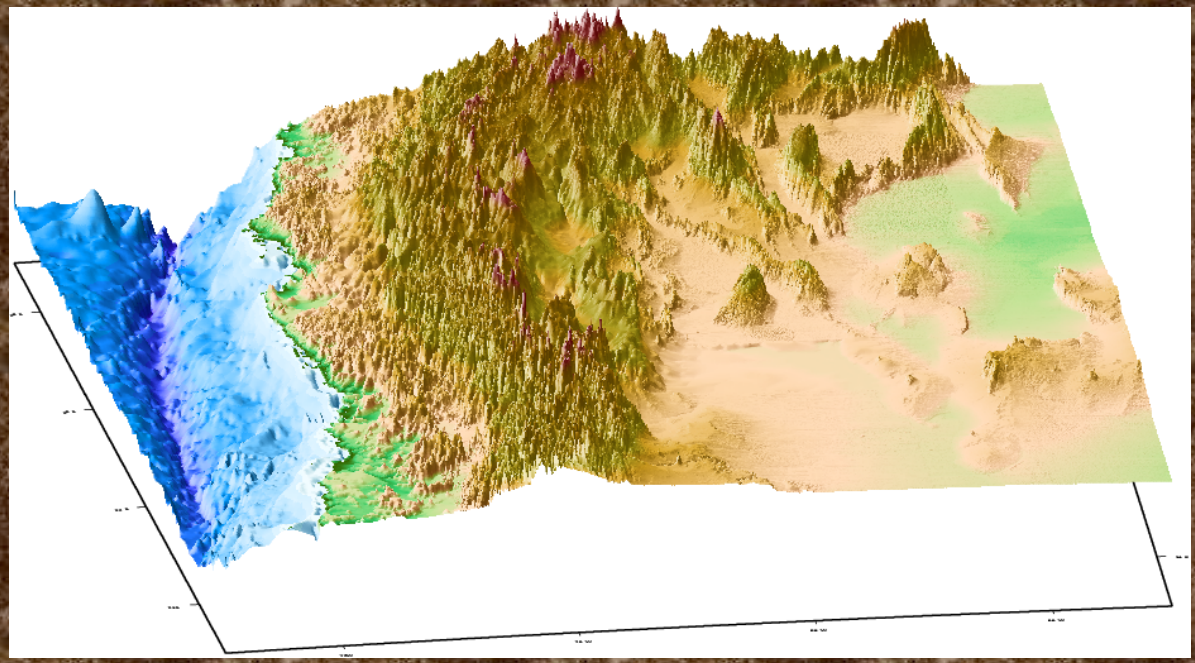


More advanced - topography

GENERIC MAPPING TOOLS (GMT)

We want to plot
Earthquakes
Moment Tensors
Digitized geologic data
Topography/Bathymetry
Other Geophys. Data
Roads, Cities, etc.

What tools there are to handle these data sets - GMT is one of them.



Another example

Non-simple input data format

We look in the file

```
1975061019 3539818n 682 64w2567  864                                                  /              1
 rrd P 2 75 61019 413.86122327430.36 S 3 41       1956160              3199.M  194   0  0  0  0
 cup PD1 75 61019 4 9.46 93528222.26 S 3 48       1566 80       D     -20320    68   0  0  0   0
 csj P 3 75 61019 414.26124428230.46 S 4 41       1983320             4499.   156   0  0  0   0
 pwp P 3 75 61019 412.66103826826.26 S 4 48       171399.M            15599.  213   0  0  0   0
 mtp PC0 75 61019 412.76115926627.06 S 3 41       1875 16       ?      0320     3   0  0  0   0
 abv PC1 75 61019 4 5.56 645 1213.66 S 3 48       1151 80       ?      7320   -58   0  0  0   0
                  10
19750617 445237218n4581 65w1307  515                                                  /              2
 rrd PC0 75 617 44536.53 755216           48      1309 16       ?     -29          0  0  0  0  0
 abv P 3 75 617 44543.83 908 94           48      1527320             484          0  0  0  0  0
 mtp P 0 75 617 44538.43 856207           48      1454 16              15          0  0  0  0  0
 pwp P 0 75 617 44538.13 768200           48      132                 114          0  0  0  0  0
 csj P 0 75 617 44534.83 73622048.32 S 4 48       1282 16              -599.    369  0  0  0  0
 cup P 0 75 617 44532.48 522196           48       976 16             -101         0  0  0  0  0
                  10
```

What's this?

First line – looks like earthquake location information

Pick it apart

```
1975061019 3539818n 682 64w2567  864                                  /          1
```

Year, month, day, hour, minute, second, lat (in degrees, N/S, min, seconds =DMS format), lon (DMS format), other stuff that we can't guess

Is all run together

Next batch of lines looks like phase information then line with "10"

```
rrd P 2 75 61019 413.86122327430.36 S 3 41        1956160                3199.M  194   0   0   0   0
cup PD1 75 61019 4 9.46 93528222.26 S 3 48        1566 80        D      -20320     68   0   0   0    0
csj P 3 75 61019 414.26124428230.46 S 4 41        1983320                4499.   156   0   0   0    0
pwp P 3 75 61019 412.66103826826.26 S 4 48        171399.M               15599.  213   0   0   0    0
mtp PC0 75 61019 412.76115926627.06 S 3 41        1875 16        ?       0320      3   0   0   0    0
abv PC1 75 61019 4 5.56 645 1213.66 S 3 48        1151 80        ?       7320    -58   0   0   0    0
                  10
```

GMT wants lat, long

Fact that fields are run together is a problem.

Have to pick input lines apart by column.

Have to select lines with earthquake location and ignore those with phase info

```
1975061019 3539818n 682 64w2567  864                                    /  .          1
 rrd P 2 75 61019 413.86122327430.36 S 3 41       1956160              3199.M  194  0  0  0  0
```

```
#!/bin/sh -f
nawk 'substr($0,19,1) == "n" || substr($0,19,1) == "s" \
{ print (substr($0,19,1) == "s" ? "-" : "") \
substr($0,17,2)+(substr($0,20,2)+substr($0,22,2)/60)/60, \
(substr($0,27,1) == "w" ? "-" : "" ) \
substr($0,24,3)+(substr($0,28,2)+substr($0,30,2)/60)/60}' \
timesortedallc.pha
```

```
#!/bin/sh -f
#make a simple map with point data

LATMIN=10
LATMAX=30
LONMIN=-80
LONMAX=-55
SCALE=0.6
MEDYELLOW=255/255/192
LTBLUE=192/192/255
RED=255/0/0
DONTCLOSE=-K
DONTINIT=-O
CONTINUE="-K -O"
INVLATLON="-:"

pscoast -R$LONMIN/$LONMAX/$LATMIN/$LATMAX -Jm${SCALE} \
-B10 -G$MEDYELLOW -S$LTBLUE $DONTCLOSE -P > $0.ps
psxy -R -Jm${SCALE} -Sc0.2 -G$RED -W1/0 $DONTINIT \
$INVLATLON << END >> $0.ps
`preqs2gmt.sh`
END
```

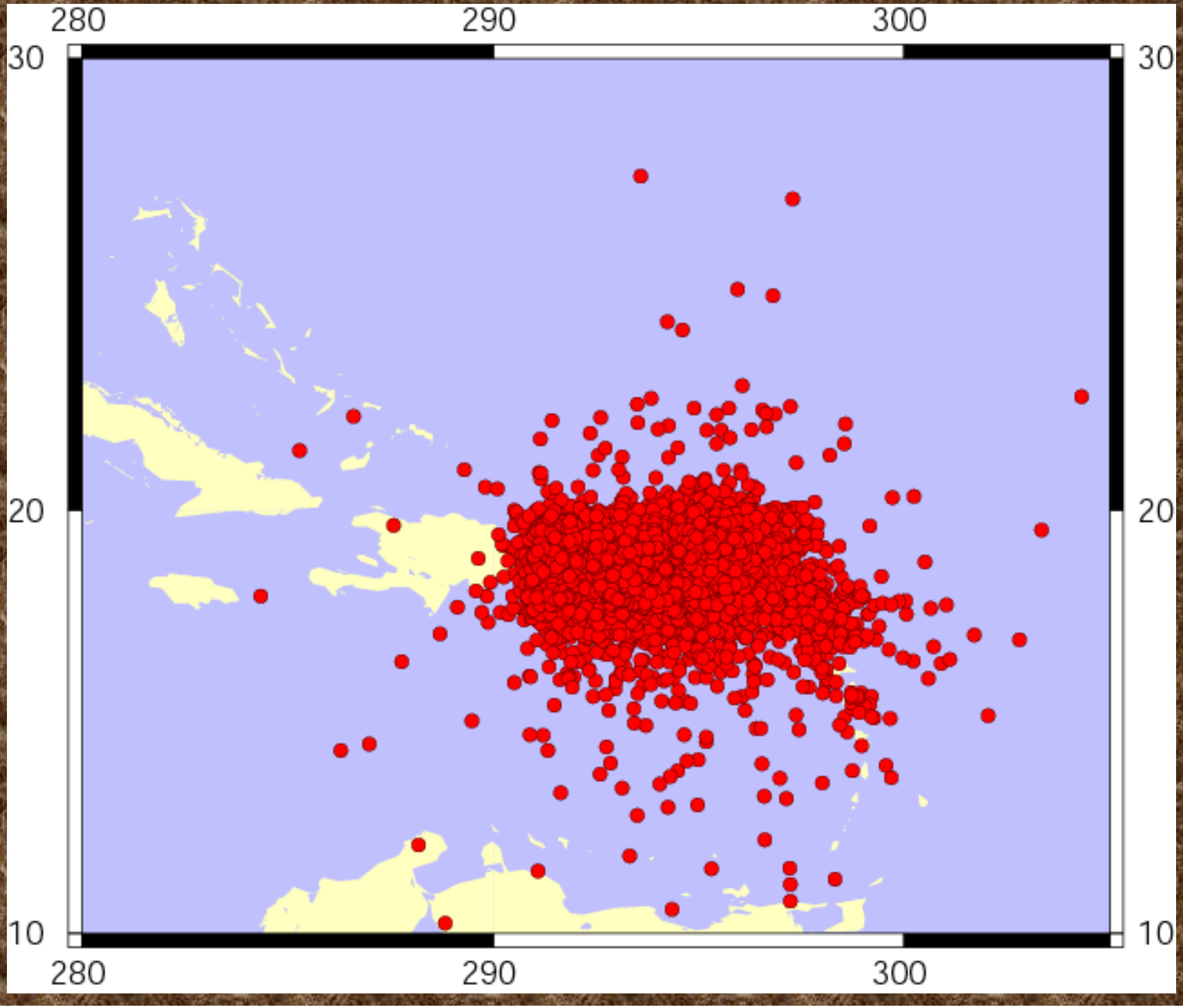
Set it up

pscoast to draw background map

psxy to draw the earthquakes (red circles with black outline)

preqs2gmt.sh to prepare data on the fly

result



Example of GMT man page – expanded for understanding

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.

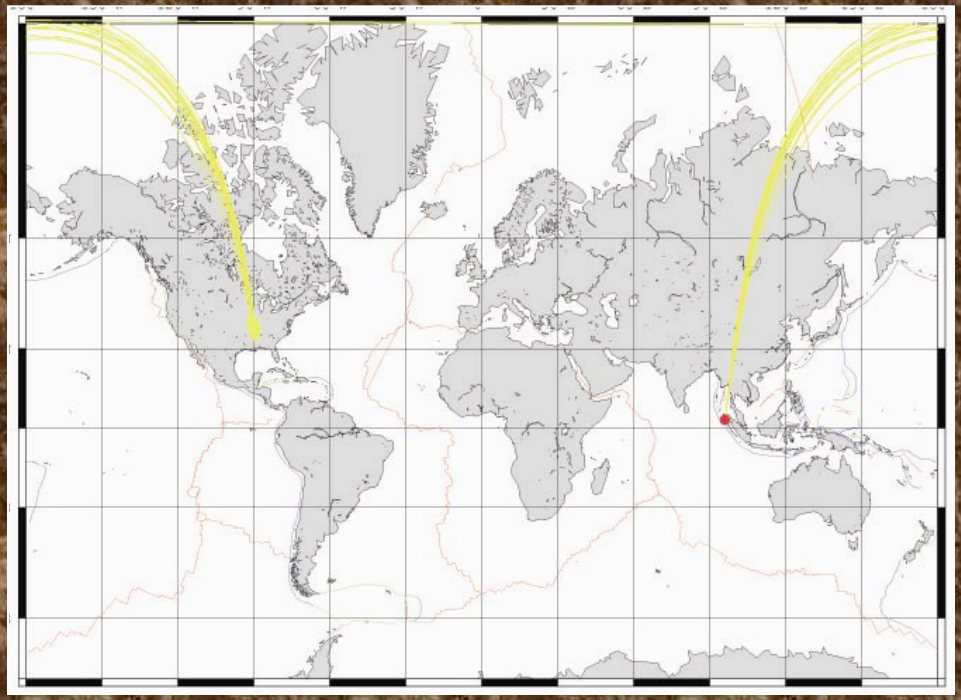
Example of GMT man page – expanded for understanding

psxy . . .

Symbols whose size is <= 0 are skipped.

If no symbols are specified then the symbol code (see -S below) must be present as (specify symbol in) last column in the input.

Multiple segment files (lift pen) may be plotted using the -M option. If -S (symbol plotting) is not selected, a (great circle) line connecting the data points will be drawn instead.



To explicitly close polygons, use -L.

Shade with -G. If -G is set, -W (line width and color) will control whether the polygon outline is drawn or not.

If a symbol is selected, -G and -W determines the fill color and outline/no outline, respectively.

The PostScript code is written to standard output (screen!).

Things you can plot with psxy - Point or line data with symbols

star
Bar
Circle
Diamond
Ellipse
front [w/ various symbols such as thrust fault barbs, warm front symbol, etc.]
Hexagon
Invtriangle
Letter
Point
Square
Triangle
Vector
Wedge
cross

Make focal mechanisms - use GMT filter (program/routine) psmeca

make/obtain input file - see psmeca documentation for large number of ways to define focal mechanism data

```
35.59  -90.48 12  220 65  150 4.5975 -0.25 -0.25
35.86  -89.95 16  220 75  150 4.0727 -0.25 0.25
36.37  -89.51 7.5 350 84  145 4.2020 -0.25 0.25
36.54  -89.68 9   85  60  -20 3.7118 0 0.5
36.56  -89.83 8   90  67.5   20  4.1068 -0.25 -0.25
36.64  -90.05 15  304 78  -28 4.6309 0 -0.5
37.16  -89.58 15  140 75  50  4.2547 0.25 0
37.22  -89.31 1.5 280 70  -20 3.5783 -0.25 0.25
37.36  -89.19 16  30  70  170 3.8250 0.25 0.25
37.44  -90.44 15  350 60  135 4.0126 0.25 0.25
37.48  -90.94 5   260 40  -70 4.5728 0.25 -0.25
37.91  -88.37 22  0   46  79  5.2612 -0.35 0.1
38.55  -88.07 15  310 70  0   4.3154 -0.25 -0.25
38.71  -87.95 10  135 70  15  4.9309 -0.25 0.25
```

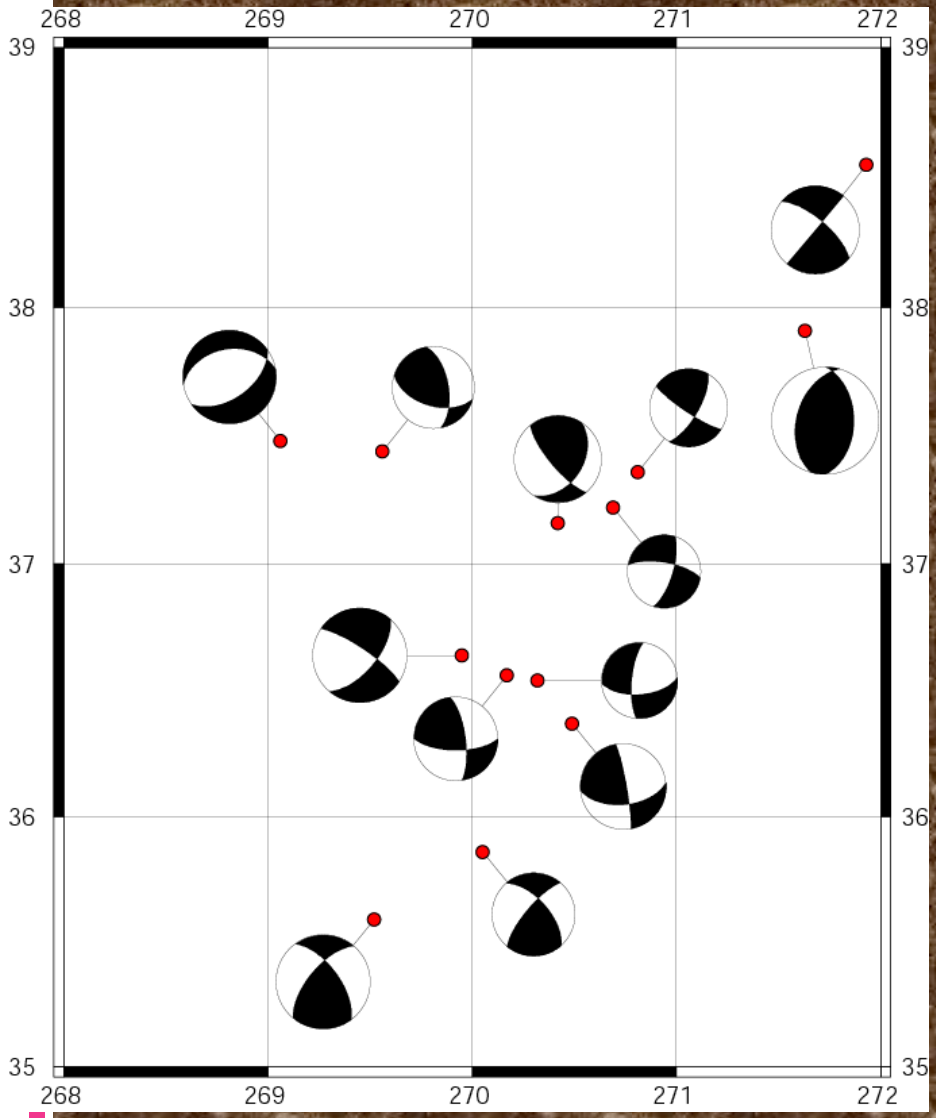
Make map with focal mechanisms (psmeca) and earthquake locations (psxy)

```
#!/bin/sh -f
REG=-92/-88/35/39
psmeca -R$REG << END -Jm4. -Bg1f1a1 -P -Sa2./0/0 -CP -: -K > $0.ps
`nawk '{print $1, $2, $3, $4, $5, $6, $7, $1+$8, $2+$9}'
practice_data.dat`
END
psxy -R$REG practice_data -Jm4. -Sc0.25 -: -G255/0/0 -W3/0 -O >> $0.ps
```

-S for focal mechanism input format definition

-C for plotting beach ball away from earthquake location and connecting it to point at earthquake location with a line

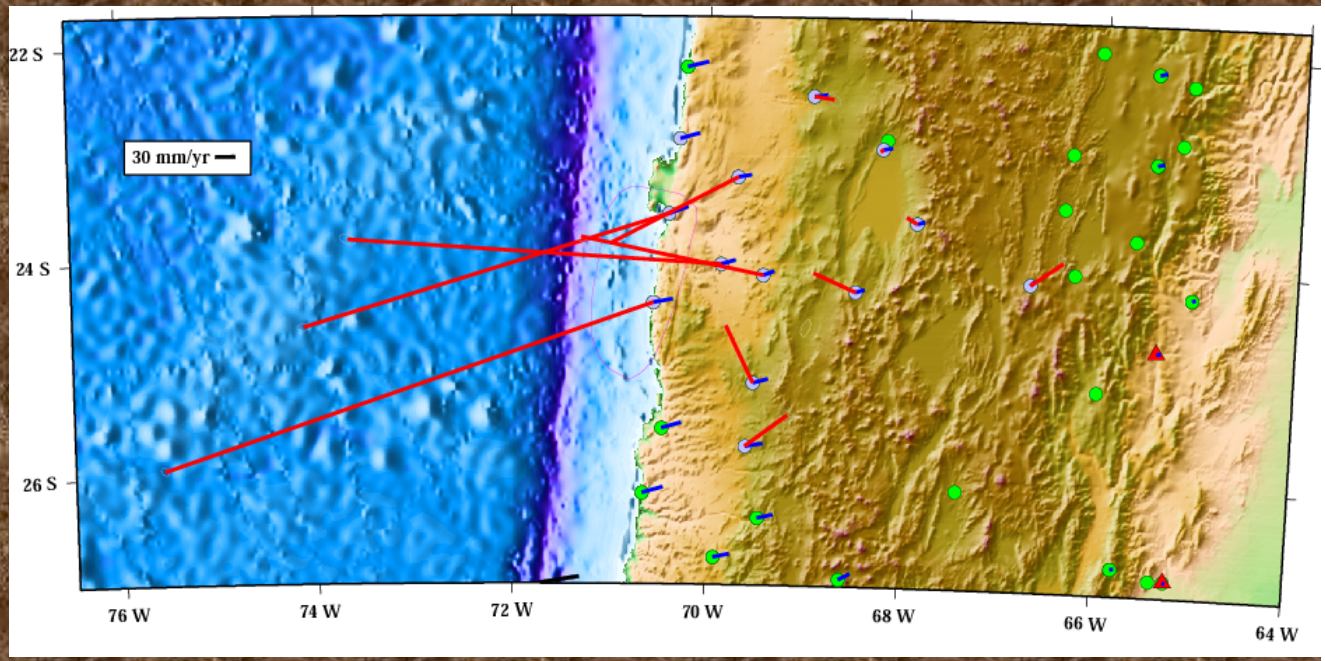
```
35.59  -90.48 12  220 65  150 4.5975 -0.25 -0.25
`nawk '{print $1, $2, $3, $4, $5, $6, $7, $1+$8, $2+$9}'
```



Uses "offsets" specified in columns 8 and 9 to reposition the focal mechanism.

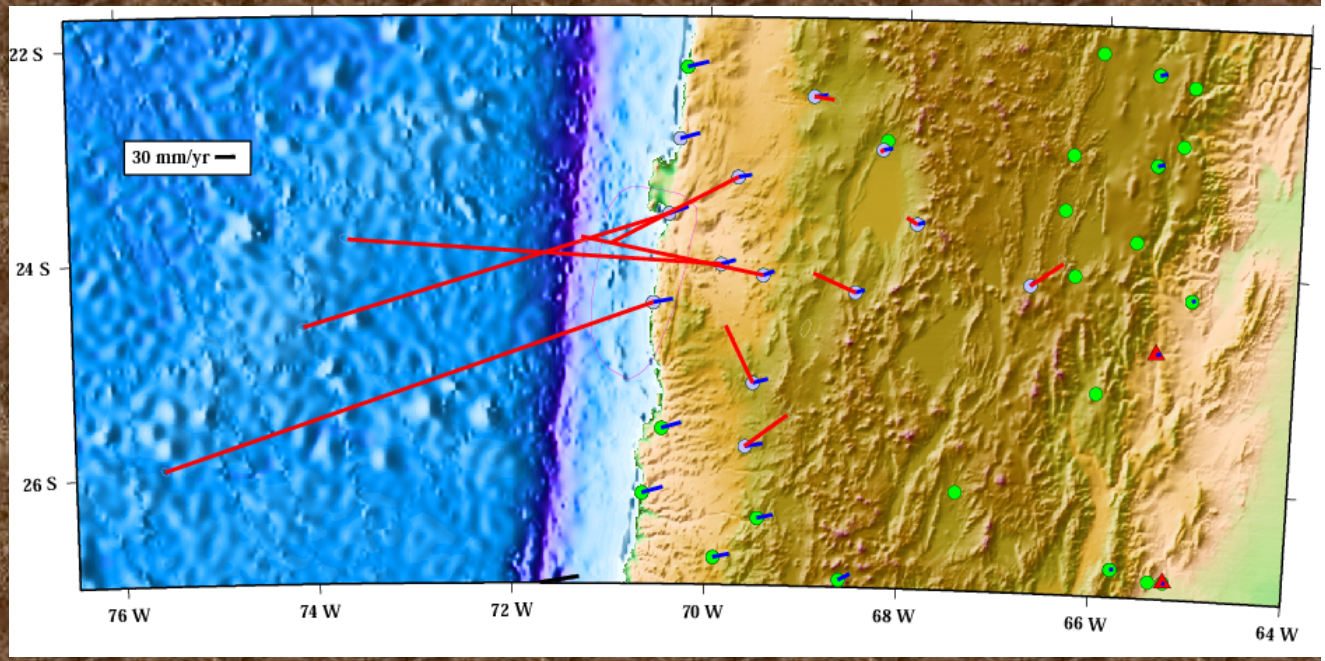
You could put the lat, long you wanted in cols 8 and 9, but why calculate all of them by hand?

You have to specify the offsets for each beachball depending on how things look, no easy way to do automatically.

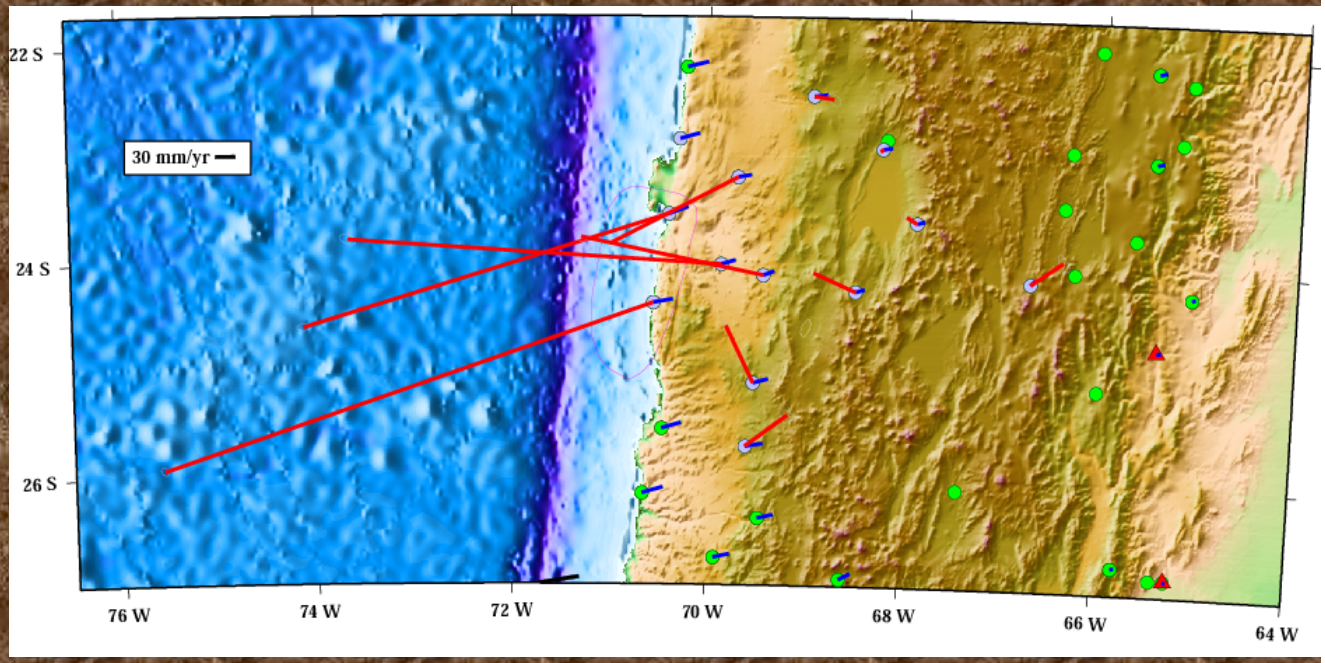


Plot

- Velocity vectors with error ellipses
- Anisitropy bars
- Rotational wedges
- Strain crosses




```
psvelo -R -$PROJ$SCALE -Sr$VELLEN/0.95/0 -W1/$PURPLE -G$PURPLE
\
$VELARROW $CONTINUE $VBSE andaman_nicobar_coseis.dat \
>> $OUTPUTFILE
```



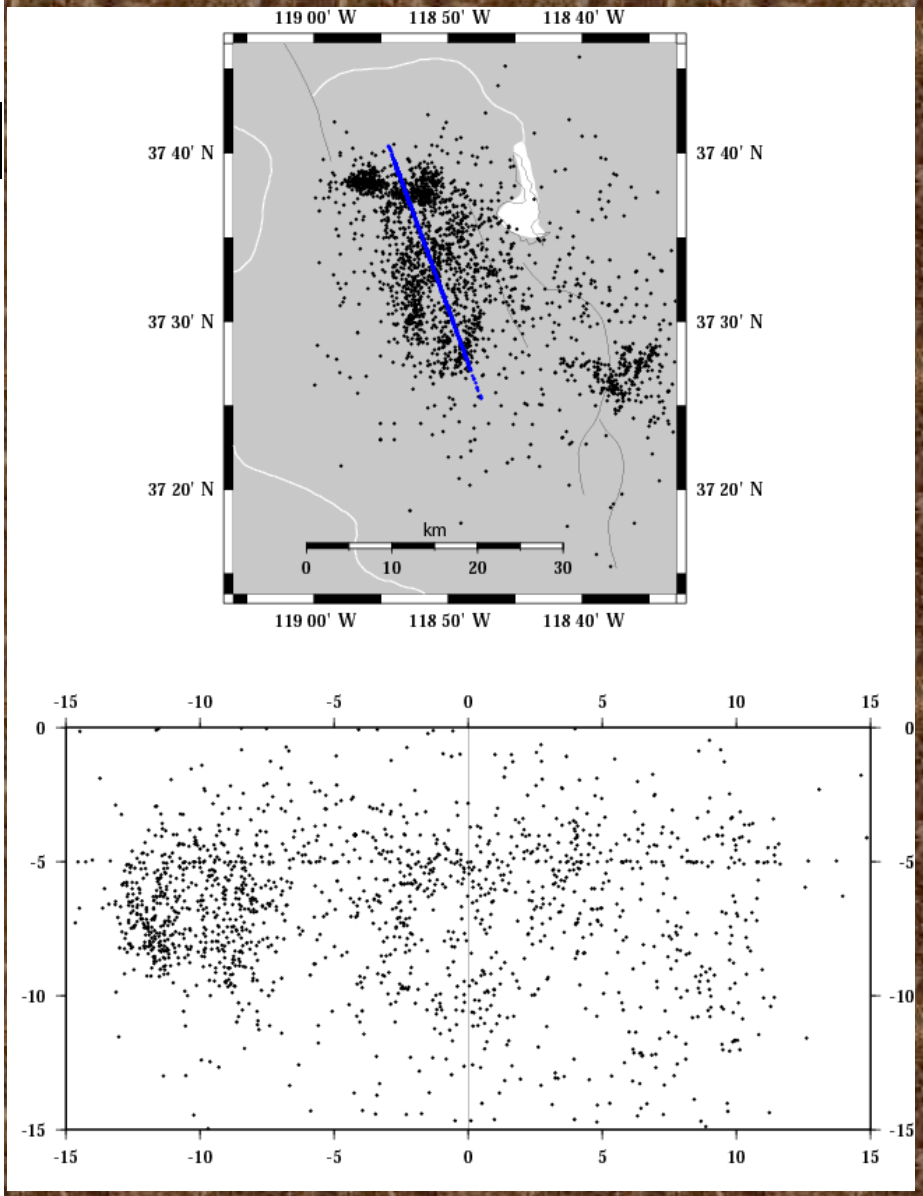
Various ways to define vector data

(ve, vw, or mag, az)

Vector length, error ellipse confidence for plot, label font size

Arrow shaft width, head length and width

Data - Lat lon vlat vlon 1siglat 1siglon corr



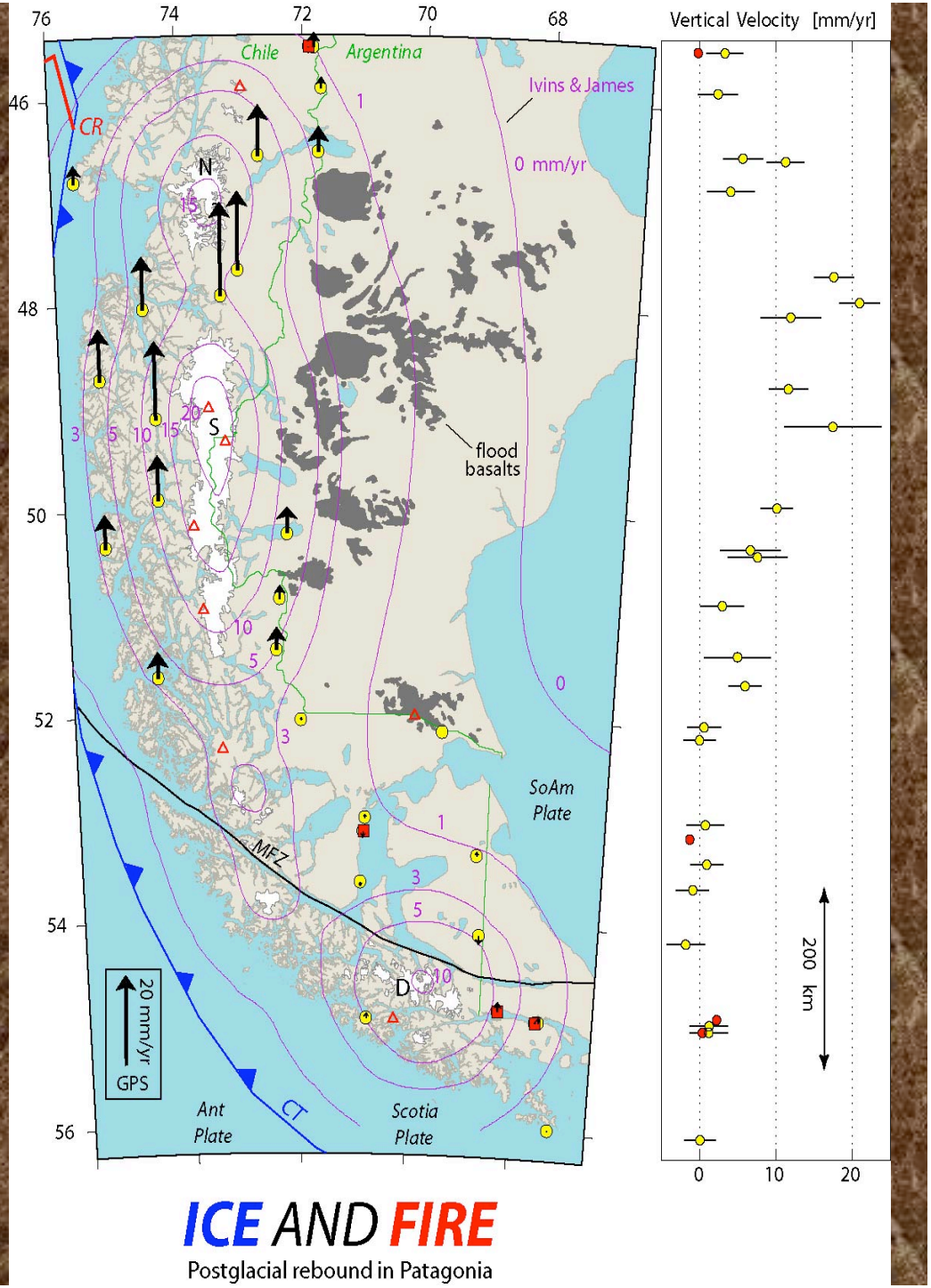
Make a cross section

(2 parts, draw map, draw cross section)

Data and non working version of shell script from
http://www-geology.ucdavis.edu/~gps/GMT/LONG_VALLEY/hypocenter.html

```
# Set PARAMETERS FOR CROSS-SECTION PLOT
center="-118.85/37.55"
azimuth="160.0"
#3.  DEFINE A BOX
width="-5/5"
length="-15/15"
\rm LV_seismicity.tmp
nawk '{print $1,$2, $3}' LV_seismicity.dat | project -C${center}\ -A$
{azimuth} -Q -W${width} -L${length} -V > LV_seismicity.tmp

# PLOT CROSS-SECTION HYPOCENTERS ON MAP
nawk '{print $6,$7}' LV_seismicity.tmp | psxy  -J${projection} \
-R${range} -P -M -Sc0.03 -G0/0/255 -O -V -K >> ${psfile}
# PLOT CROSS-SECTION BOX
# SET PARAMETERS TO PLOT
brange="-15/15/-15/0"
bprojection="x0.2/0.2"
btick="a5f5g0/a5f5g0"
psxy box_dim -R${brange} -J${bprojection} -B${btick} -W1 -P -O \
-K -X-1.25 -Y-4 -V >> ${psfile}
# PLOT HYPOCENTERS ON CROSS-SECTION
nawk '{print $4, $3*(-1.0)}' LV_seismicity.tmp | psxy -P -M \
-J${bprojection} -R${brange} -Sc0.03 -G0/0/0 -O -V >> ${psfile}
```

(Make and then) Plot contours



ICE AND FIRE

Postglacial rebound in Patagonia

First grid data for gmt.

```
echo make pgr contours
PGRFILE=pgr5e18
SPACING=4m
xyz2grd $SAMDATA/$PGRFILE -G$SAMDATA/$PGRFILE.grd -ISPACING /
-: -R$REGION
grdinfo $SAMDATA/$PGRFILE.grd
```

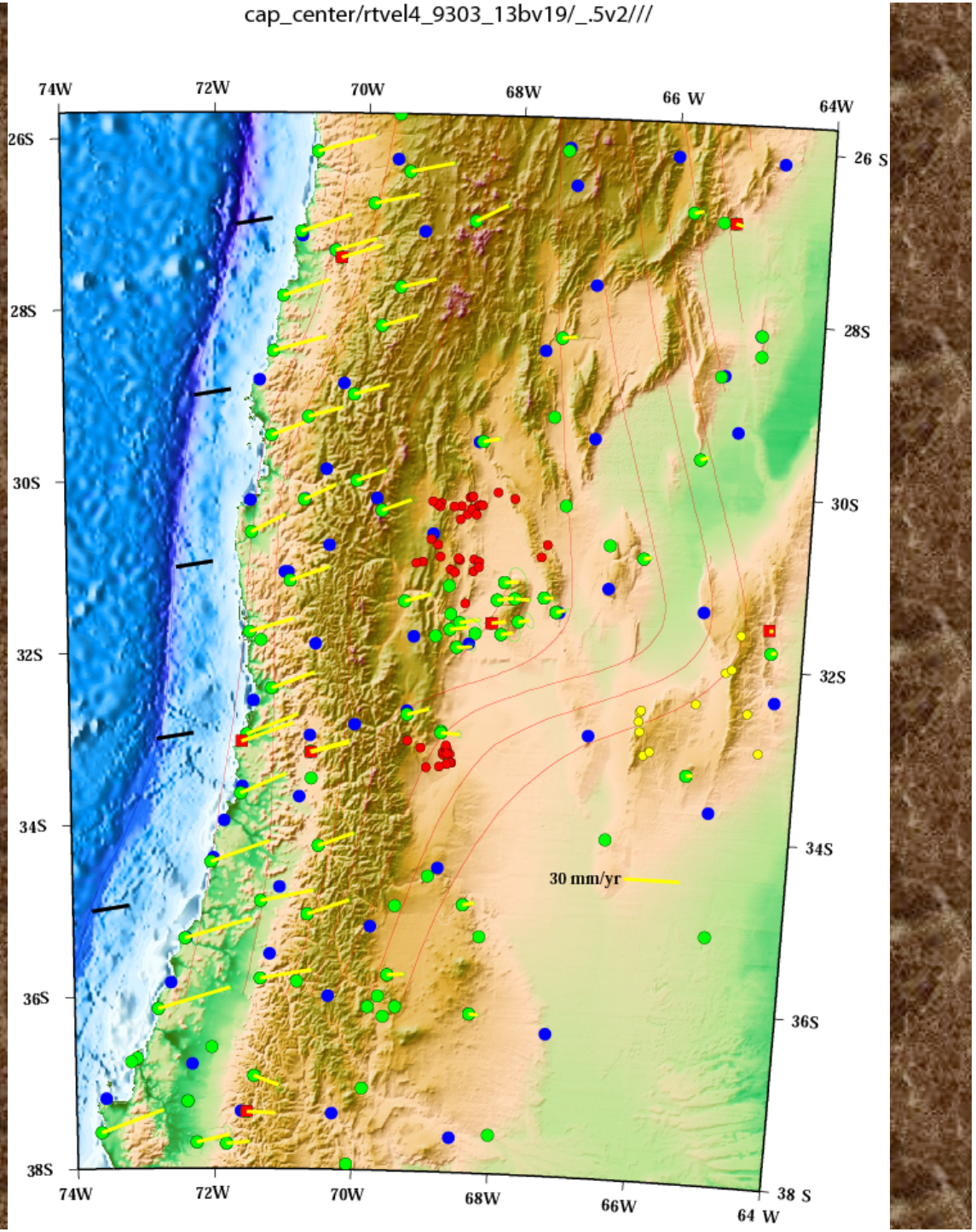
Then contour.

```
grdcontour $SAMDATA/$PGRFILE.grd -C1 -Jx1.0 -D$PGRFILE.con -
M /
-R$REGION > /dev/null

#have to hand edit the contour file to do 2 things -- as made
the first point in each contour
#is stuck on the end of the new contour seperator line - have
to add <cr>, also does VERY bizzare
#stuff with > for segment seperator, change to $ and works
fine.
#exit

psxy -R$REGION -$PROJ/$SCALE -M -W$LINETHICK/$ICECOLOR /
$CONTINUE $PGRFILE.con $VBSE >> $OUTPUTFILE
```

Returning to making pertty MAPS?

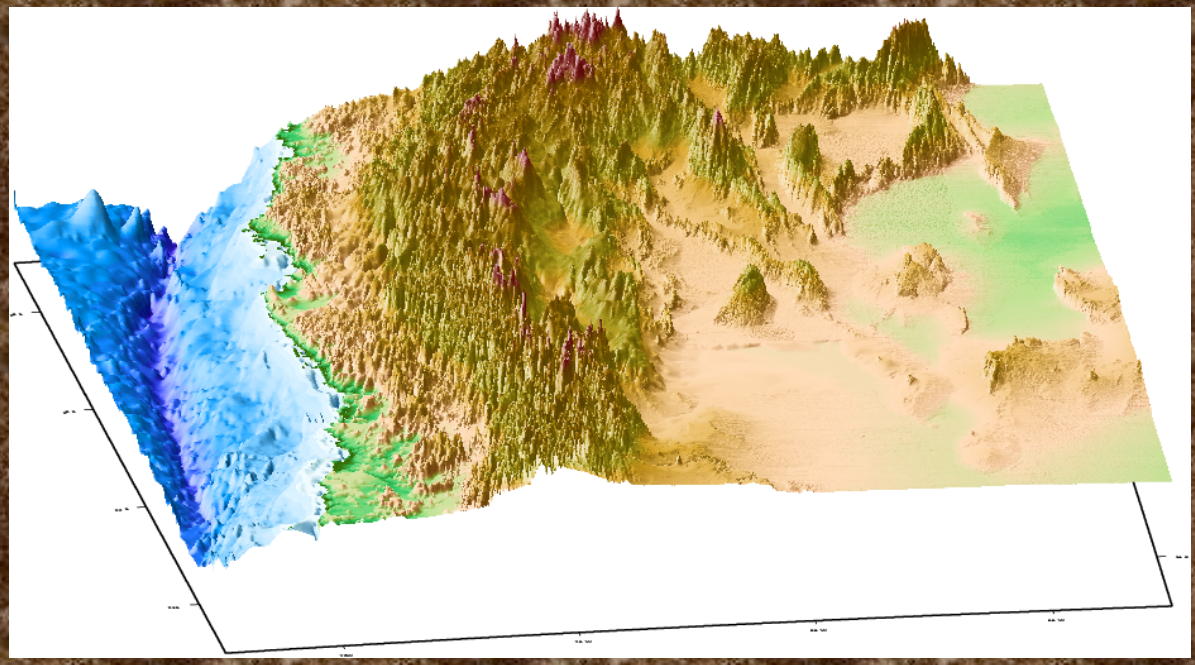
How to do:

- color or b&w topo with shaded topo
- how to combine topo and bathymetry



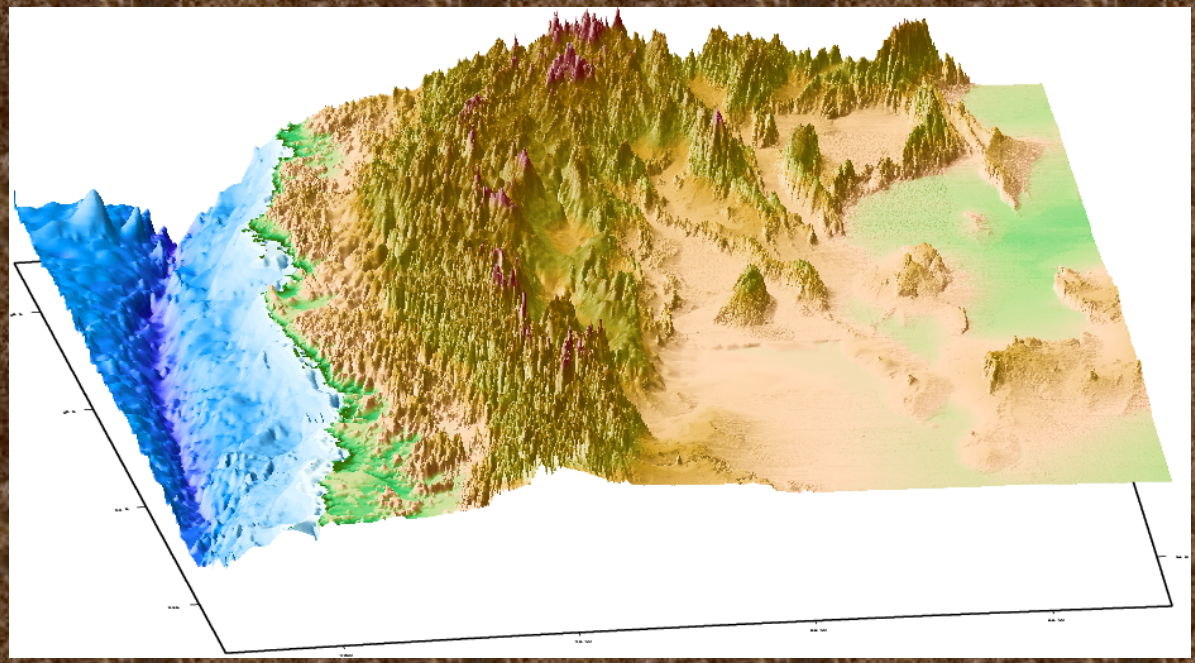
First – have to find data – what's available

DEM's (Digital Elevation Models) of world – several resolutions, several kinds of data (GTOPO-30, ETOPO-5 , SRTM, seasat, obs/ predicted bath, gravity)



Where to get them?

(We have some online at CERI – makes it easy.
Have not fully figured out SRTM yet.)



use grdraster to extract a subregion from the global bathymetry data set and make a new grid file for GMT.

grdraster is not part of "standard" GMT. Is a "supplemental" GMT program.

There are a bunch (order 35-40) of such supplemental GMT programs like this around.

Many are written by others (not Smith and Wessell) and become "attached" to GMT and can be found on the GMT web page, but they are not officially part of GMT.

psmeca and psvelo (to draw focal mechanisms and vector fields) are in this class.

use grdraster to extract a subregion from the global bathymetry data set and make a new grid file for GMT.

$GRDRASTERREGION has same format at the REGION definition (min lon/max lon/min lat/max lat) and been previously set up to define the region

```
echo do seafloor
DATASET=10
DATAGRID=-I2m/2m
grdraster $DATASET -G${ROOTNAME}_2mtopo.grd $DATAGRID \
-R$GRDRASTERREGION -V
echo done with 2m topo grdraster
```

Let's look at the documentation first

Typing grdraster all by itself dumps the man page.

- reports available data sets, unit, data coverage area, spacing and registration (pixel or grid – not important for now, except that when combining data sets they have to be the same).

```
alpaca/smalley 142:> grdraster
grdraster 3.4.3 - Extract a region from a raster and save in a grdfile
usage: grdraster <file number> -R<west/east/south/north>[r] \
[-G<grdfilename>] [-I<dx>[m][/<dy>[m]]][-bo[s][<n>]]
        <file number> (#) corresponds to one of these:

#  Data Description                    Unit    Coverage      Spacing Registration
-------------------------------------------------------------------------------
1 "ETOPO5 global topography"          "m"     -R0/359:55/-90/90      -I5m     G
2 "US Elevations from USGS"           "m"     -R234/294/24/50        -I0.5m   P
3 "Geo/Seasat grav from Haxby"        "mGal"  -R0/359:55/-90/90      -I5m     G
4 "Geo/Seasat geoid from Haxby"       "m"     -R0/359:55/-90/90      -I5m     G
5 "Sea floor age from Cande"          "Ma"    -R0/359:55/-90/90      -I5m     P
6 "Sea floor age from Muller et al., 1998" "Ma" -R0/360/-72/90 -I6m     G
7 "Sea floor age errors Muller et al., 1997" "Ma" -R0/360/-72/72 -I6m   G
8 "1=land, 0=sea bitmask"             "T/F"   -R0/360/-90/90        -I5m     P
9 "USGS/SS ETOPO30s"    "m"                   -R0/360/-90/90        -I0.5m   P
10 "2min Observed/Predicted Topo"     "m"   -R0/360/-72/72        -I2m     P
11 "et30wbath"    "m"    -R-78/-63/-25/-12          -I0.5m        P
-------------------------------------------------------------------------------
```

First use grdraster to extract a subregion from the global data set

```
echo do seafloor
DATASET=10
DATAGRID=-I2m/2m
grdraster $DATASET -G${ROOTNAME}_2mtopo.grd $DATAGRID \
-R$GRDRASTERREGION -V
echo done with 2m topo grdraster
```

We have selected the 2m predicted sea floor topography - data set 10.

We have set the grid to the proper sample spacing (get from previous slide w/ data set properties).

First use grdraster to extract a subregion from the global data set

```
echo do seafloor
DATASET=10
DATAGRID=-I2m/2m
grdraster $DATASET -G${ROOTNAME}_2mtopo.grd $DATAGRID \
-R$GRDRASTERREGION -V
echo done with 2m topo grdraster
```

We are going to put the data into a file called ${ROOTNAME}_2mtopo.grd

Now we do the same for the land topographic data, using GTOPO-30, which only has data for land.

```
echo do topo
DATASET=9
DATAGRID=-I30c/30c
grdraster $DATASET -G${ROOTNAME}_topo.grd $DATAGRID \
-R$GRDRASTERREGION -V
echo done with gtopo grdraster
```

Now we select the ETOTO-30 topography - data set 9.

Notice that the grid has a different sample spacing than the bathymetry, otherwise this code snippet is the same.

Now we do the same for the land topographic data, using GTOPO-30, which only has data for land.

```
echo do topo
DATASET=9
DATAGRID=-I30c/30c
grdraster $DATASET -G${ROOTNAME}_topo.grd $DATAGRID \
-R$GRDRASTERREGION -V
echo done with gtopo grdraster
```

The data will go into a file called ${ROOTNAME}_topo.grd

We now have two complimentary data sets, one for topography and one for bathymetry and we have to combine them.

Unfortunately, they have different sample spacing.

So we have to resample one of the data sets – lets do it to the sea floor (since it has the lower resolution – we will therefore be interpolating).

Use grdsample to resample the bathymetry as defined by DATAGRID and put in a new file ${ROOTNAME}_30stopo.grd

```
echo prep and merge bathy
DATAGRID=-I30c/30c
grdsample ${ROOTNAME}_2mtopo.grd -G${ROOTNAME}_30stopo.grd $DATAGRID \
-F -R$GRDRASTERREGION -V
```

Now we use grdmath to combine (AND) the two data sets (they have distinguishing values in the dataless points).

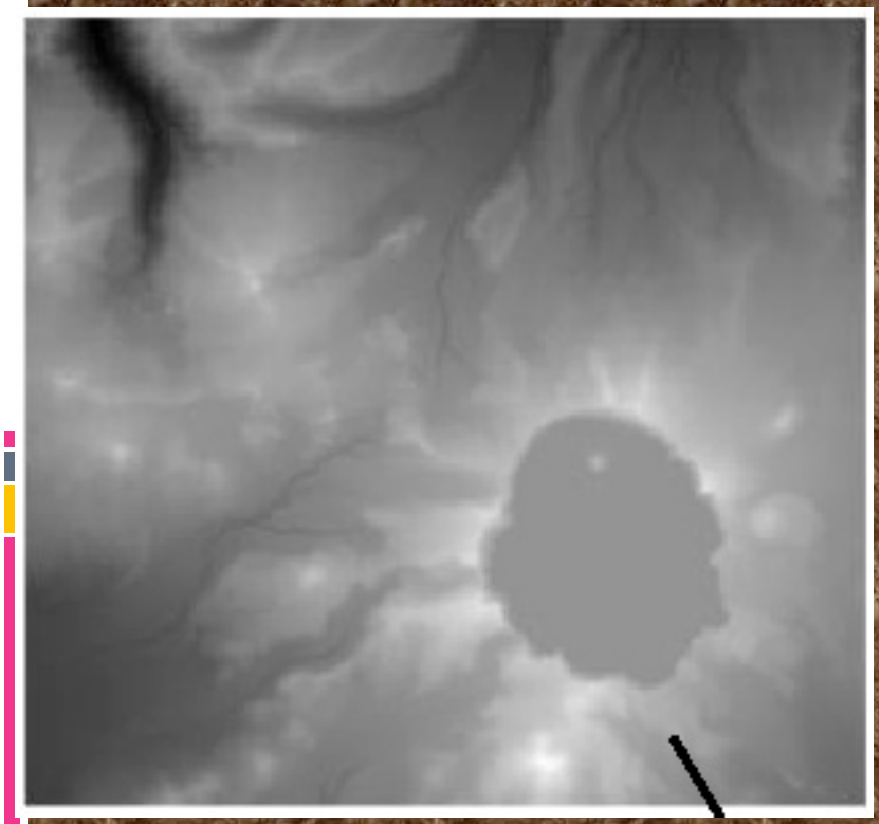
grdmath uses a stack and RPN - Reverse Polish Notation)

```
grdmath -F -V ${ROOTNAME}_topo.grd ${ROOTNAME}_30stopo.grd AND = \
${ROOTNAME}_topobath.grd
echo done with merge bathy
```

And put the new topo file in ${ROOTNAME}_topobath.grd

We are now done selecting the topographic and bathymetric data,

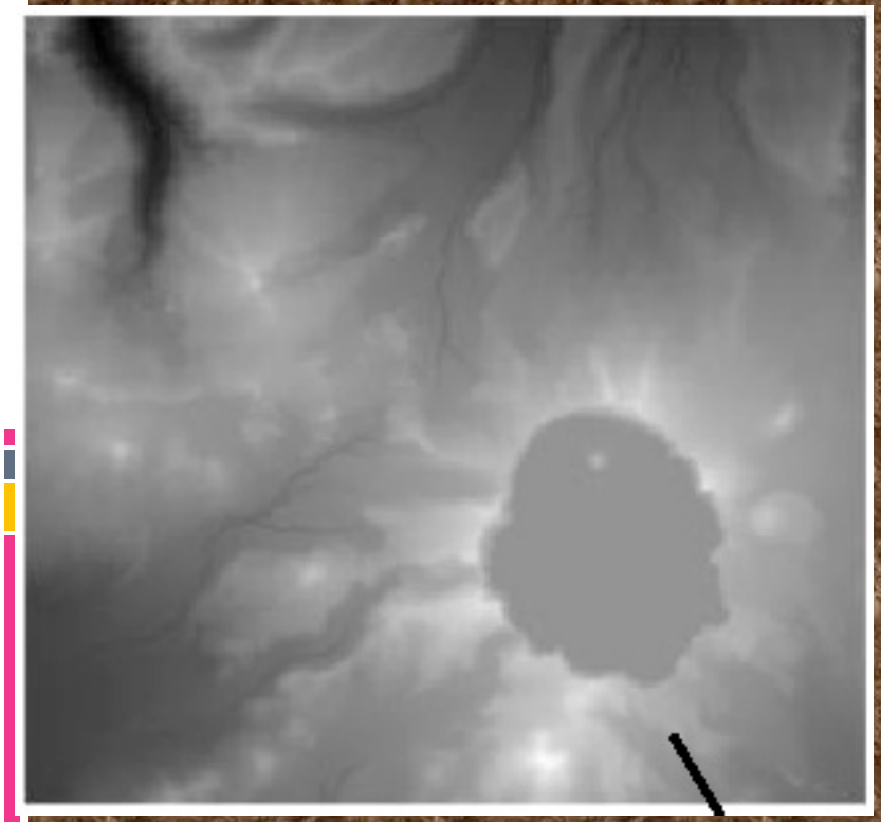
which is used to give the coloring or grayscale.

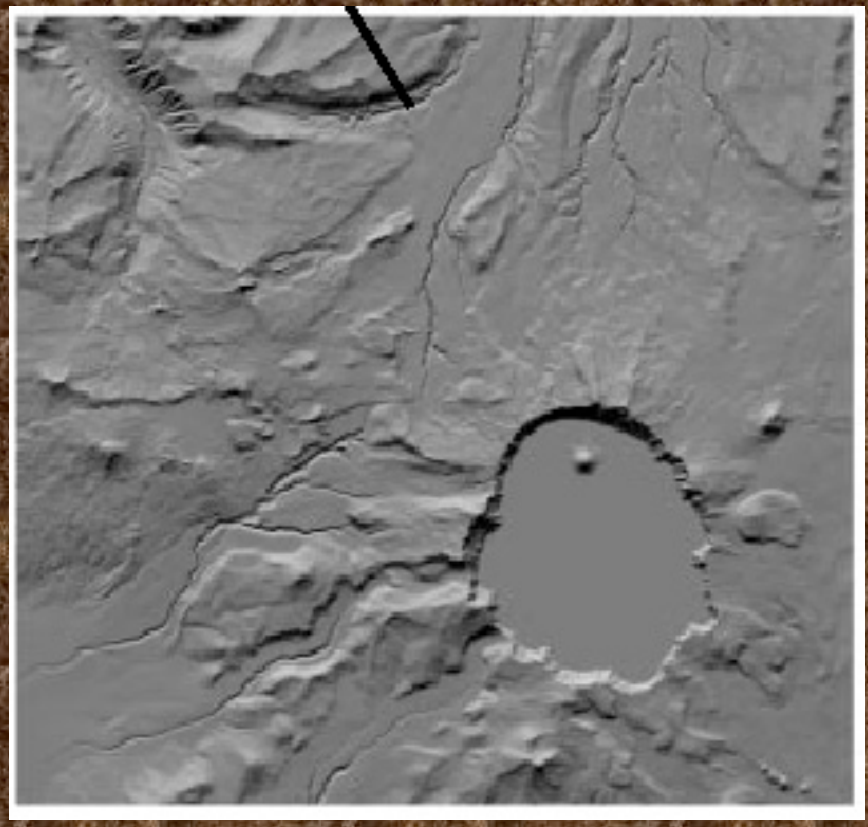


It is very hard, however, for the brain to interpret this view of the data.

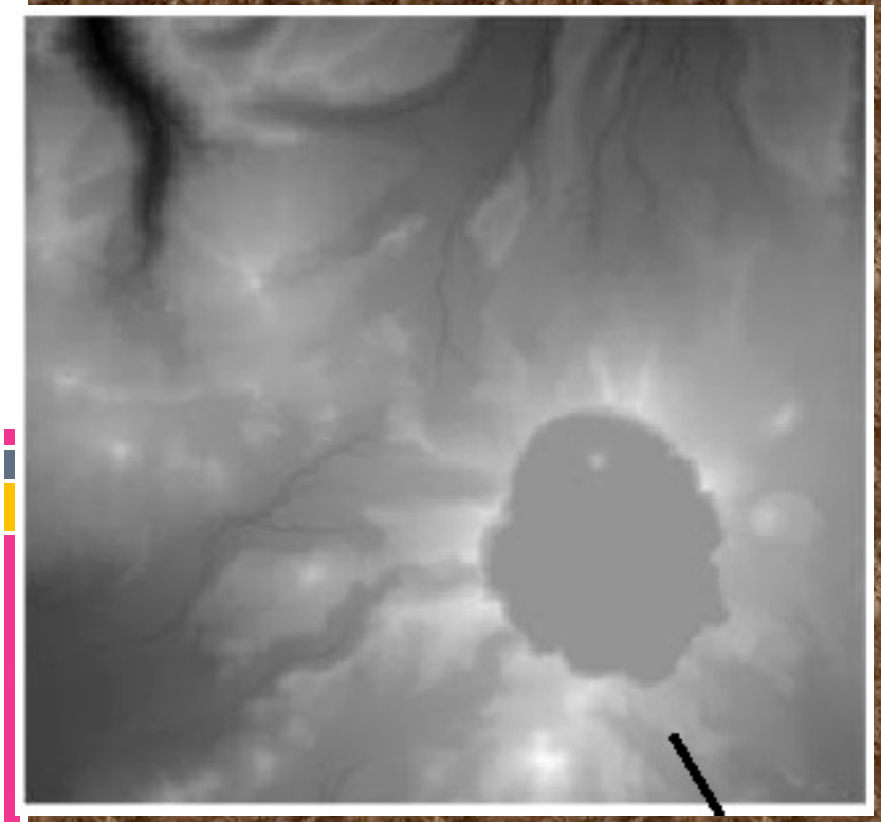
What is this?

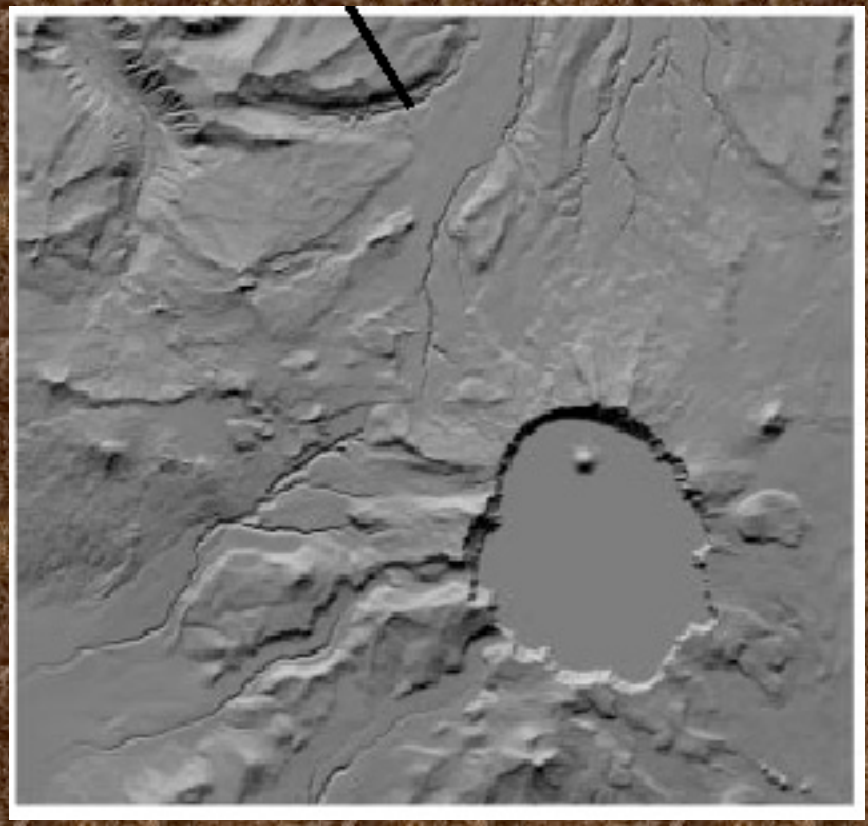
One needs to add shadows (shading) for the brain to "get the picture" (and even then there are some problems.)





We will therefore "illuminate" the topography and generate an intensity filter to be added to the color or grayscale image.

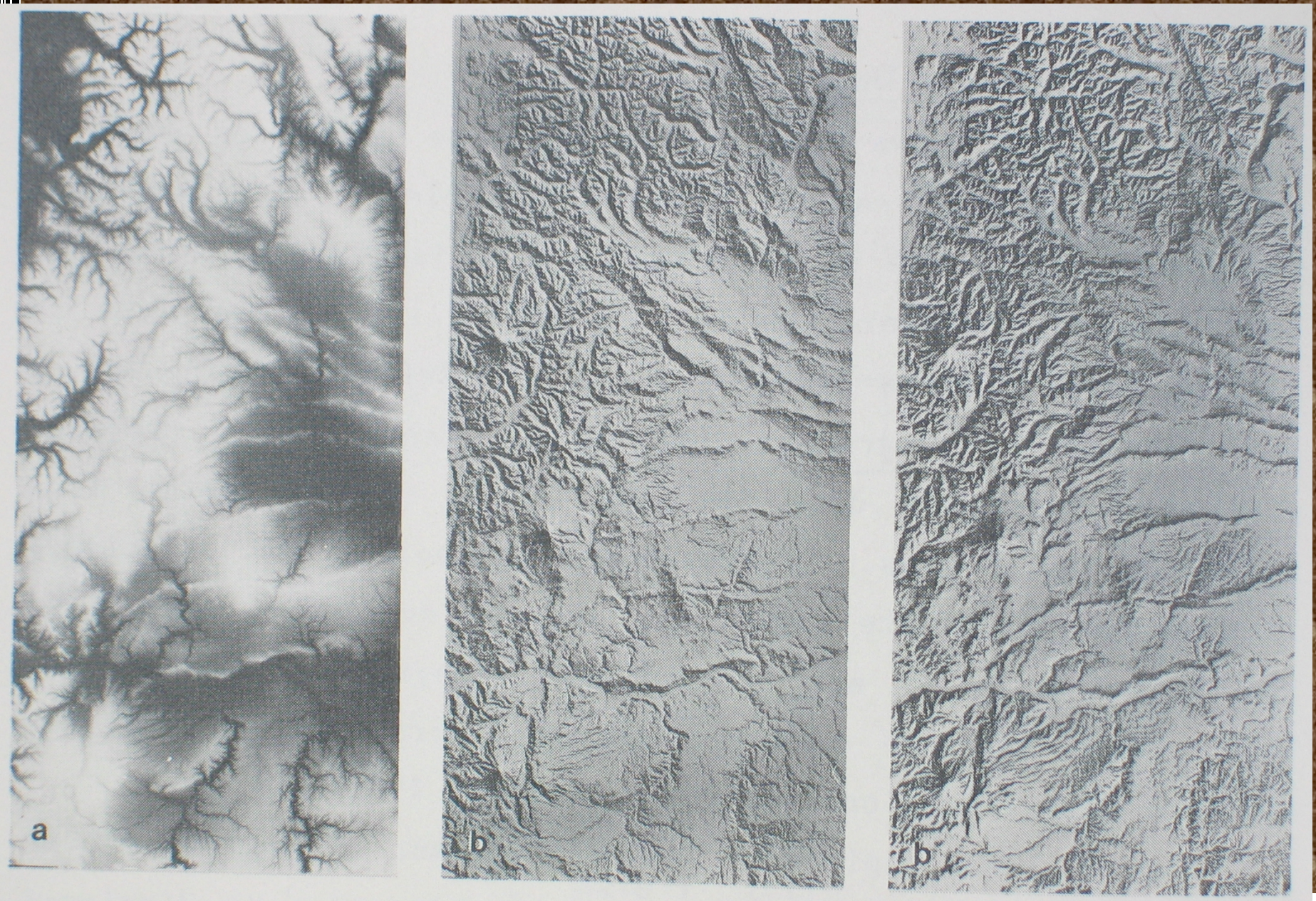




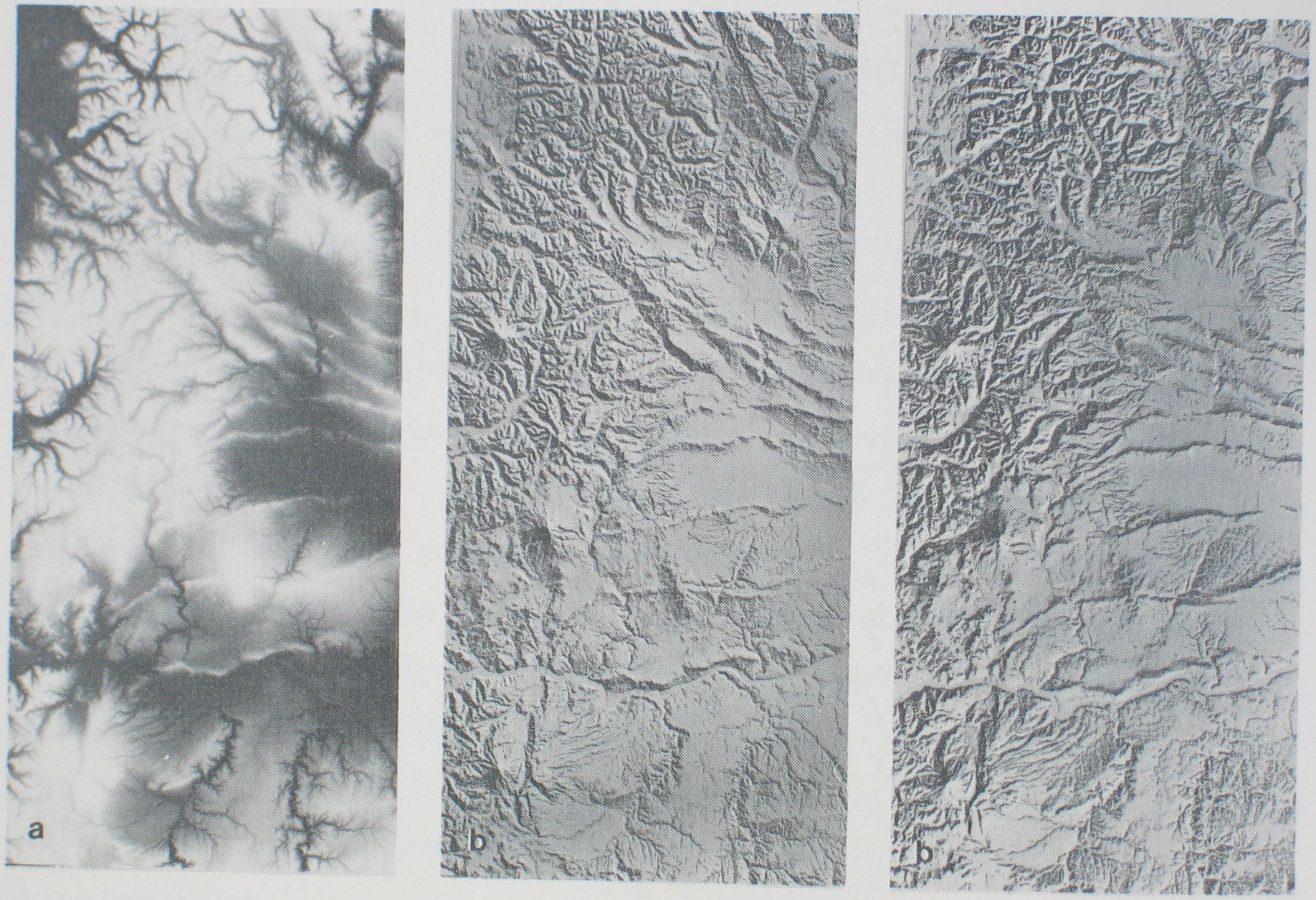
"Raw" data
grey scale into topo

shaded topo "illuminated" from
upper right

lower right



The left is an image of the data (altitude), two on the right are nice visual pictures but do not show the altitude.



GMT has a routine to do this grdgradient.

I'll also illuminate the ocean floor and the topography from slightly different angles - to bring out the "best" of both.

After generating the illumination, we have to combine the two files using grdmath.

Output files will have .intns as extension.

```
NORM=-Nt
BATHILLUM=270
TOPOILLUM=315
grdgradient ${ROOTNAME}_topo.grd -A$TOPOILLUM \
-G${ROOTNAME}_topo.intns $NORM -V

grdgradient ${ROOTNAME}_30stopo.grd -A$BATHILLUM \
-G${ROOTNAME}_30stopo.intns $NORM -V

grdmath -F -V ${ROOTNAME}_topo.intns ${ROOTNAME}_30stopo.intns AND = \
${ROOTNAME}_topobath.intns

INTNSFILE=${ROOTNAME}_topobath
```

So now we have two grid files

One with the topographic data
One with the shading

Now we're ready to plot them together to make the map.

Finally we make our first contribution to the map (PostScript output file) using grdimage.

grdimage can combine the coloring of the data, based on the CPT file, with the shading (which comes from the slopes of the data).

grdimage can combine the coloring of the data, based on the CPT file, with the shading (which comes from the slopes of the data).

```
echo color topo
CPTFILE=/gaia/opt/gmt/share/GMT_globe.cpt
grdimage $INTNSFILE.grd -I$INTNSFILE.intns -C$CPTFILE -R$REGION -$PROJ
$SCALE $GRID -K -X$XOFFSET -Y$YOFFSET -V $ORIENT > $OUTPUTFILE
echo done with color topo
```

The CPT file is the color table file. GMT has a bunch of them predefined (look in the directory referenced above).

grdimage can combine the coloring of the data, based on the CPT file, with the shading (which comes from the slopes of the data).

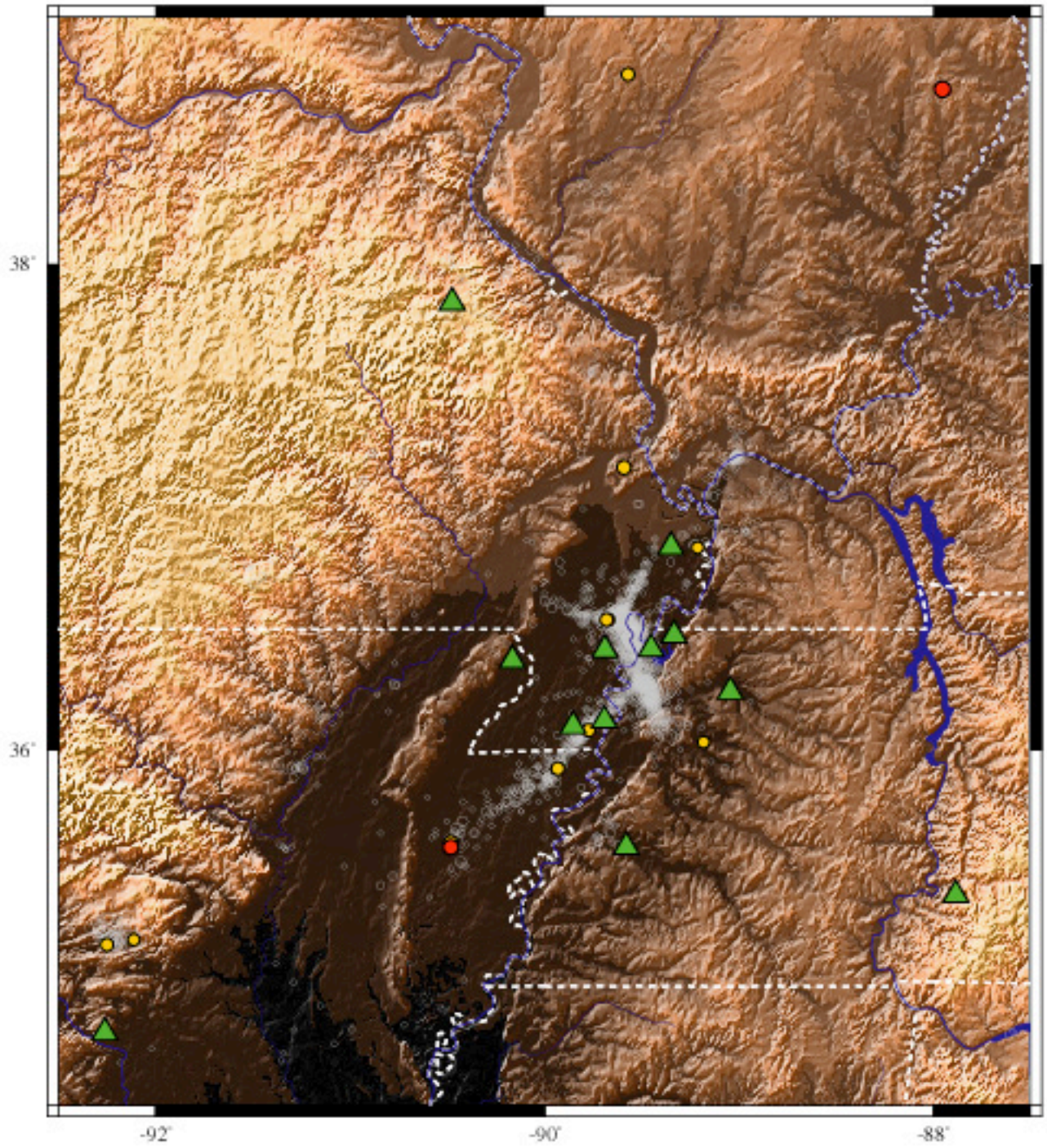
```
echo color topo
CPTFILE=/gaia/opt/gmt/share/GMT_globe.cpt
grdimage $INTNSFILE.grd -I$INTNSFILE.intns -C$CPTFILE -R$REGION -$PROJ
$SCALE $GRID -K -X$XOFFSET -Y$YOFFSET -V $ORIENT > $OUTPUTFILE
echo done with color topo
```

The CPT file is the color table file. GMT has a bunch of them predefined (look in the directory referenced above).

GMT uses the R/G/B model for color

You can also make your own CPT files
(if you have lots of time)

Now we can add other data (notice -K) --- earthquakes, GPS vectors, focal mechanisms, etc.



"copper" cpt file

Now we can add other data - earthquakes, GPS vectors, focal mechanisms, etc.

```
psmeca -R -$PROJ$SCALE -Sd0.2/0/0 -G$RED $CONTINUE -L -W0.5/$BLACK \
india.cmt >> $OUTPUTFILE
```

Again being lazy, I don't like to have to keep track of the last GMT call (to keep track of whether or not I need the -O) so I use $CONTINUE.

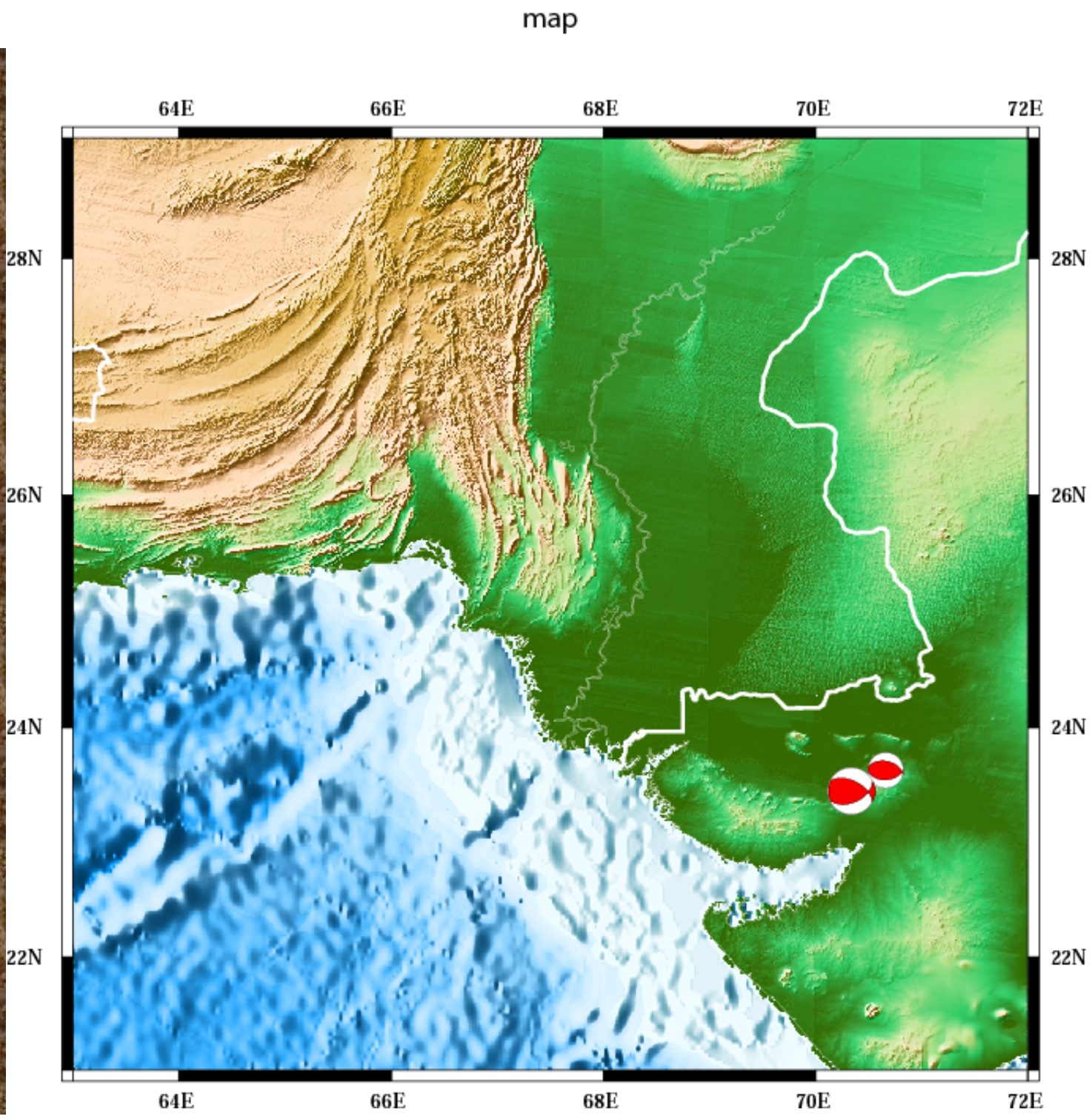
Then I check the output file for a showpage when I'm done – and write the PostScript myself when I need it.

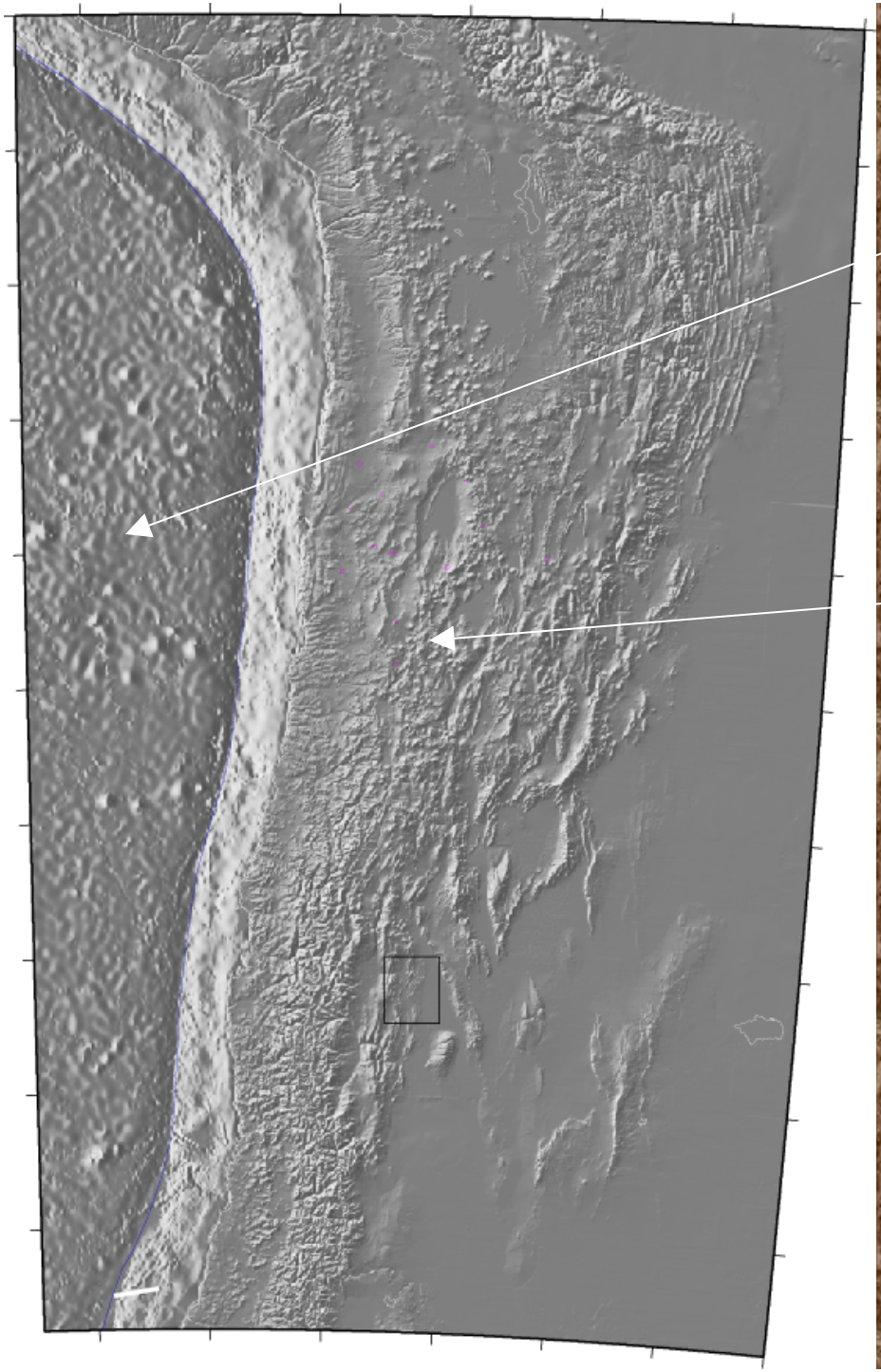
```
echo done with figure - clean up
SHOWPAGE=`tail -1 $OUTPUTFILE | nawk '{print $1}'`
echo check SHOWPAGE -${SHOWPAGE}-
if [ $SHOWPAGE != showpage ]
then
 echo add showpage
 echo showpage >> $OUTPUTFILE
fi
```

We then have to erase all the temporary files we made.

```
if [ $CLEAN = yes ]
then
 echo yes - clean up
 if [ $TOPO != notopo ]
 then

  \rm ${ROOTNAME}.cpt
  \rm ${ROOTNAME}.grd
  \rm ${ROOTNAME}.intns

  \rm ${ROOTNAME}_topo.grd
  \rm ${ROOTNAME}_topo.intns
  \rm ${ROOTNAME}_2mtopo.grd
  \rm ${ROOTNAME}_2mtopo.intns
  \rm ${ROOTNAME}_30stopo.grd
  \rm ${ROOTNAME}_30stopo.intns
  \rm ${ROOTNAME}_topobath.grd
  \rm ${ROOTNAME}_topobath.intns

 fi

 \rm ${ROOTNAME}.nawk
 \rm ${ROOTNAME}.tmp

fi
```

So here's our pretty MAP!





ETOPO -5

global

(5 min)

GTOPO-30

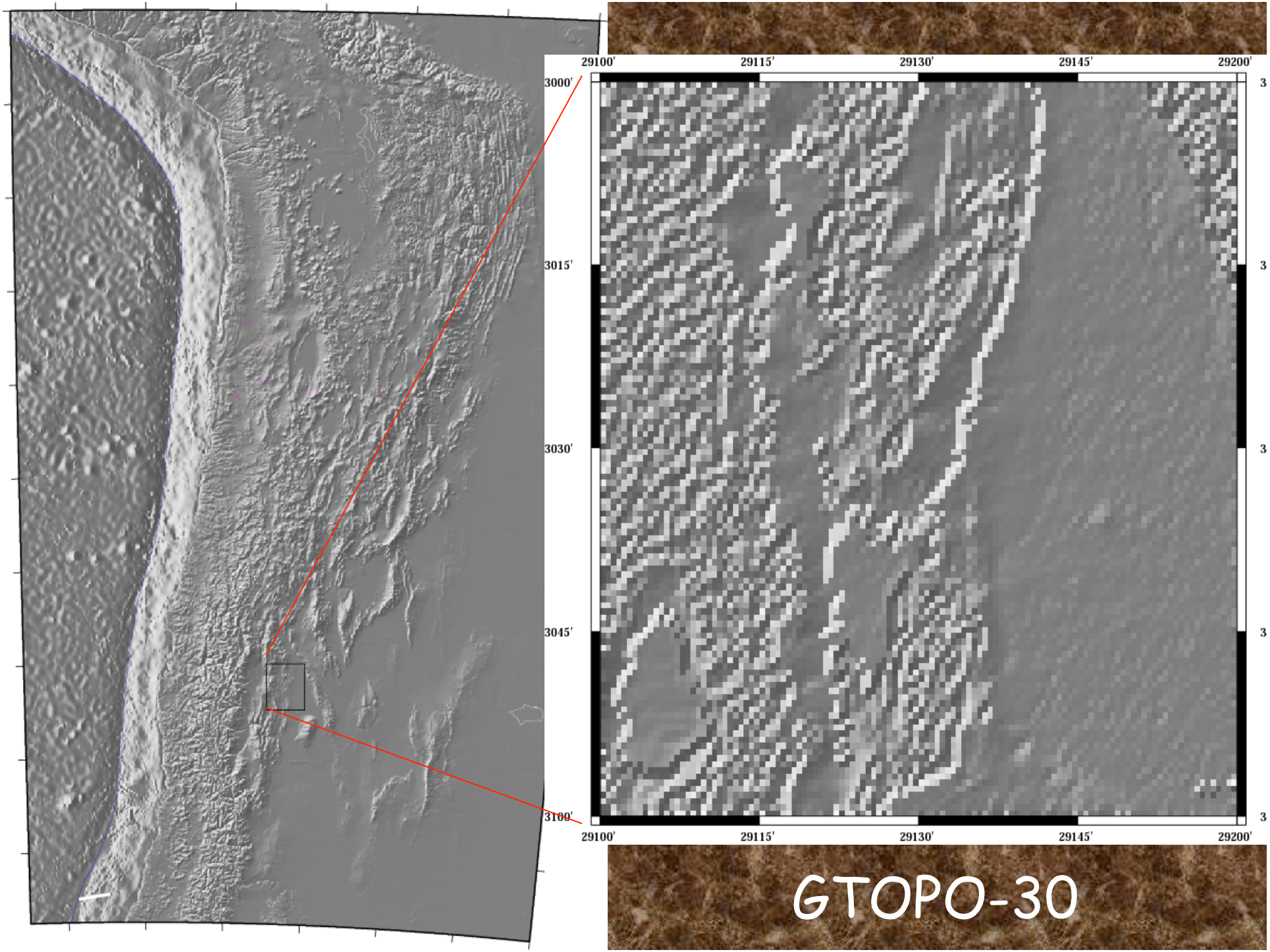
Land only

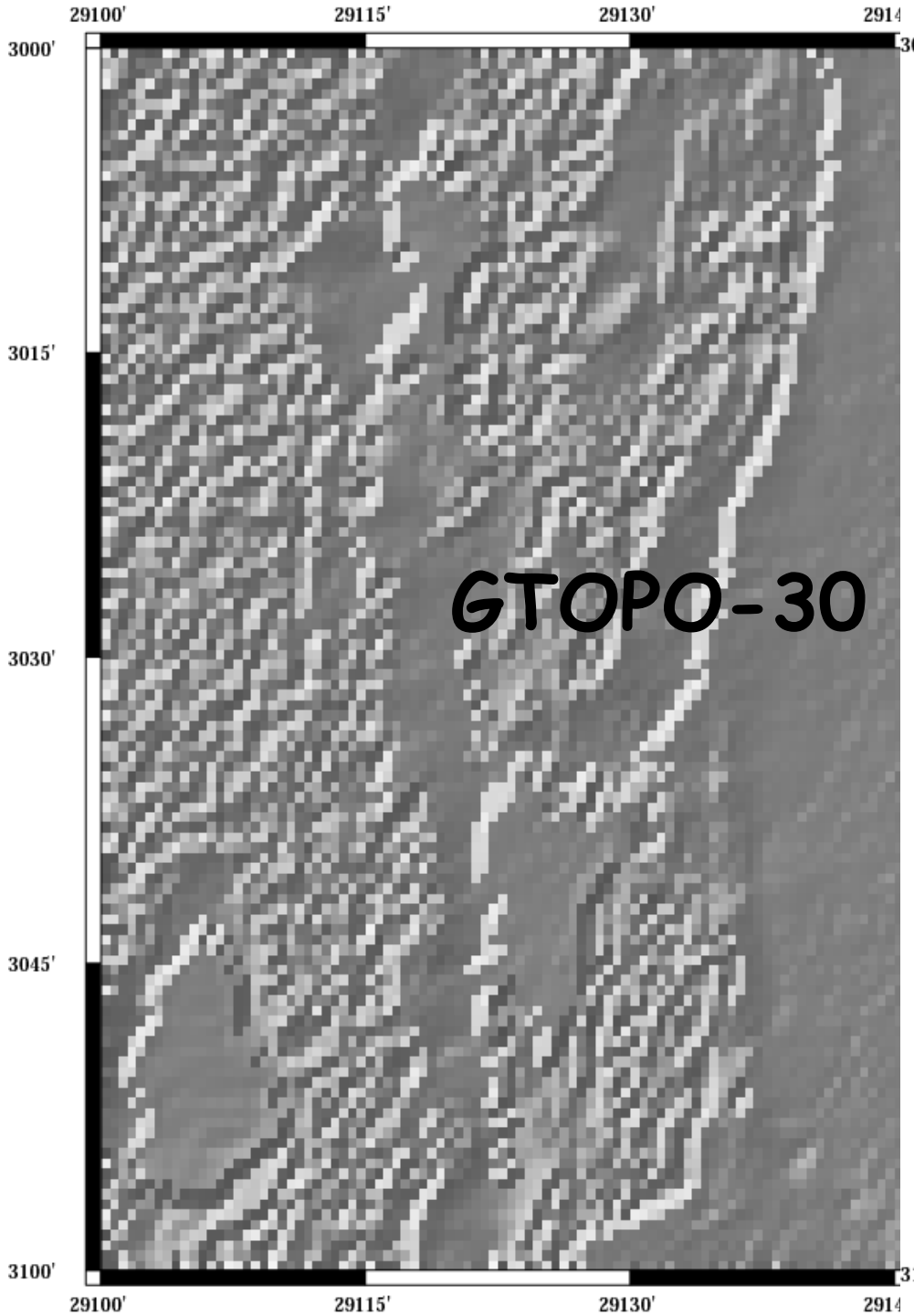
(30 sec)

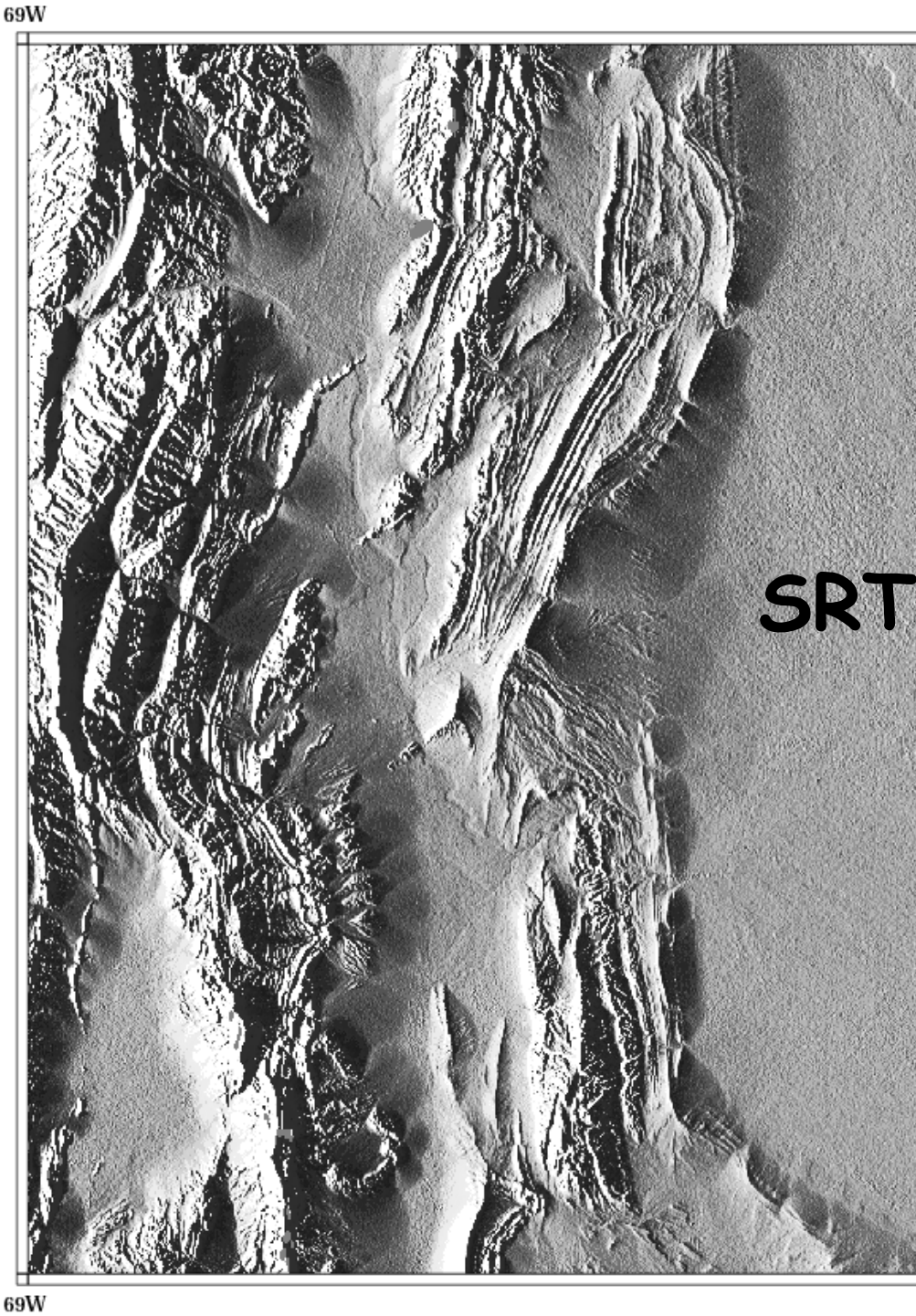
SRTM

Land only

(3 sec)


29100'
29115'
29130'
29145'
29200'
3000'
3015'
3030'
3045'
3100'
GTOPO-30


29100'
29115'
29130'
2914
3000'
3015'
3030'
3045'
3100'
GTOPO-30


69W
30S
31S
SRT

Plotting a single srtm file

```
#!/bin/sh
\rm tst.grd
grdgradient tile_31_69.grd -A270 -Gtst.intens -Ne0.6 -V \
grd2cpt tst.intens -Cgray > $0.cpt
grdimage tst.intens -Itst.intens -R-69/-68/-31/-30 -Jm7 \
-B1g1a -P -C$0.cpt > $0.ps
```

Plotting multiple 1x1 degree tiles possible, but more slightly complicated (see me).

I can't get SRTM data into grdraster format input file (any volunteers?)

NOTE:

GMT uses the NETCDF data base package for DEMs (and some other stuff).

Another "free" UNIX package.

This has to be installed and maintained separately (and is done so by Mitch).

One has to put the SRTM files one downloads from NASA, the USGS or other source into NETCDF files (this is the hard part).

Have covered lots of stuff,

but even more stuff has not been covered

- there are 60 GMT and 35+ Supplemental programs!

Plus power of UNIX to manipulate them.

General GMT shell script will look something like this

Call to set up base map – this may or may not plot any data

Series of GMT calls to add various kinds of data

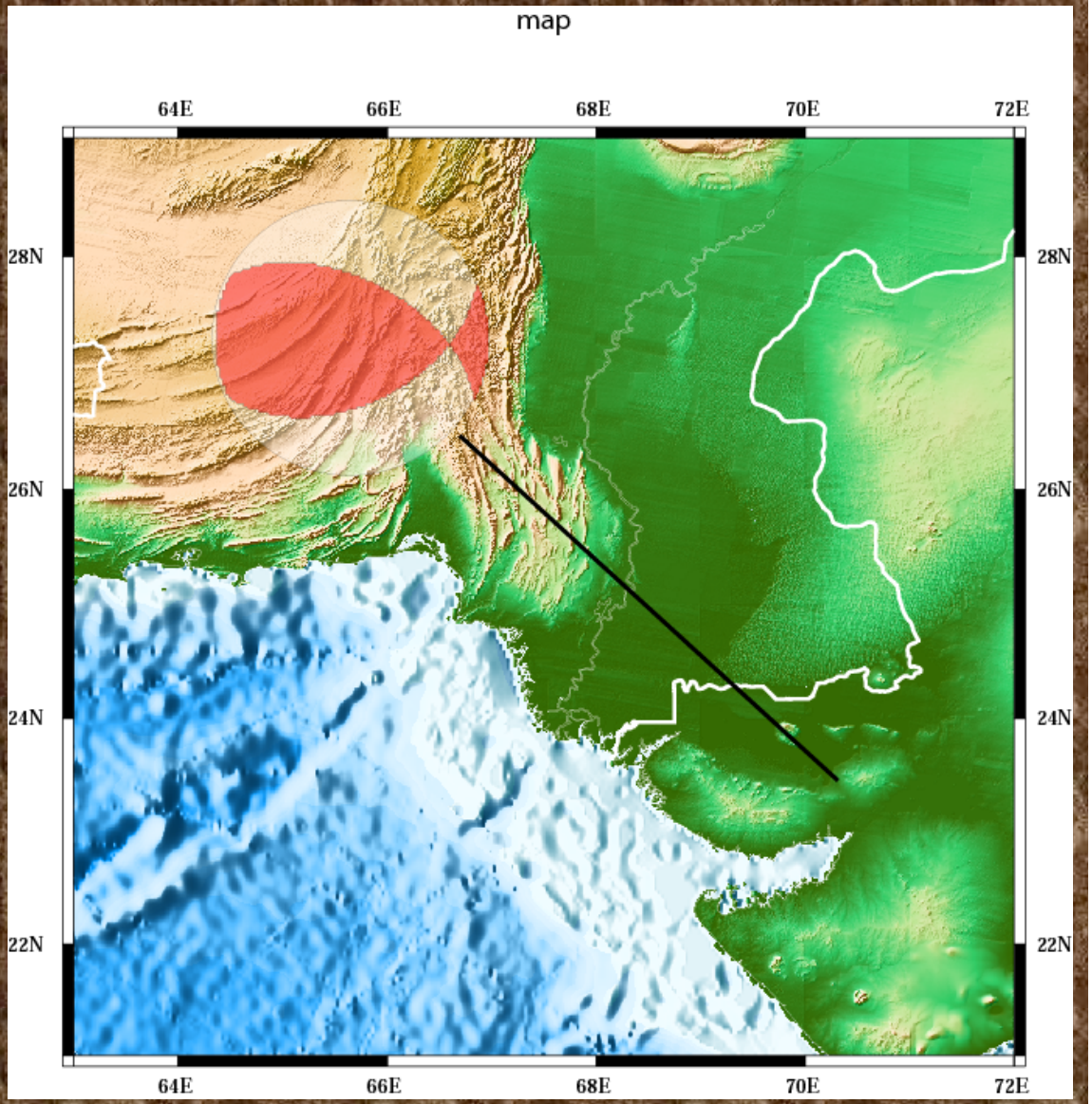
Last GMT call "closes" file

Majority of work is in manipulating the data files using all the standard UNIX tools.

Finally, you can put the finishing touches on your figure with Adobe Illustrator (which works with PostScript files)

Change line thicknesses, colors; annotate; etc. Make focal mechanism transparent. Past in other stuff.

Lots well documented problems going over to Adobe – principally with annotation/ text.



Why is GMT so popular?

The price is right!
(But there's also no such thing as a free lunch!)

Offers unlimited flexibility since it can be called from the command line, inside scripts, and from user programs.

Has attracted many users because of its high quality PostScript output.

"Easily" installs on almost any (including windows) computer.

GMT Colors

Color is specified using Red/Green/Blue (RGB) or Grayscale color

WHITE=255
LTGRAY=192
VLTGRAY=225
EXTGRAY=250
GRAY=128
BLACK=0

RED=250/0/0
DKRED=196/50/50
BLUE=0/0/255
GREEN=0/255/0
YELLOW=255/255/50

ORANGE=255/192/50
PURPLE=255/50/255
CYAN=50/255/255
LTBLUE=192/192/250
VLTBLUE=225/250/250
LTRED=250/225/225
PINK=255/225/255
BROWN=160/64/32

GMT Defaults

There are about 100 parameters which can be adjusted individually to modify the appearance of plots or affect the manipulation of data. Each as a default value.

GMT defaults are kept in a file called ~/.gmtdefaults4. There are tons of them and you can find out what they are and what the mean reading the man page for gmtdefaults.

When a program is run, it initializes all parameters to the GMT defaults, then tries to open the file .gmtdefaults4 in the current directory. If not found, it looks in a sub-directory ~/.gmt, and finally in your home directory itself.

If successful, the program will read the contents and set the default values to those provided in the file.

If a script works for the author and not for you, your defaults are probably different.

If a script works for the author and not for you, your defaults are probably different.

To view your current gmtdefault setting

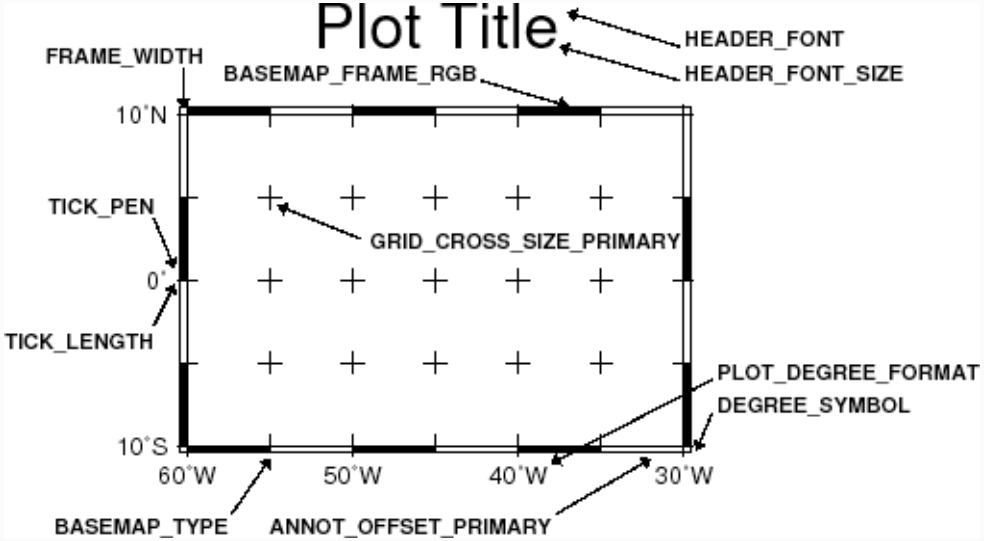
```
%gmtdefaults –L
```

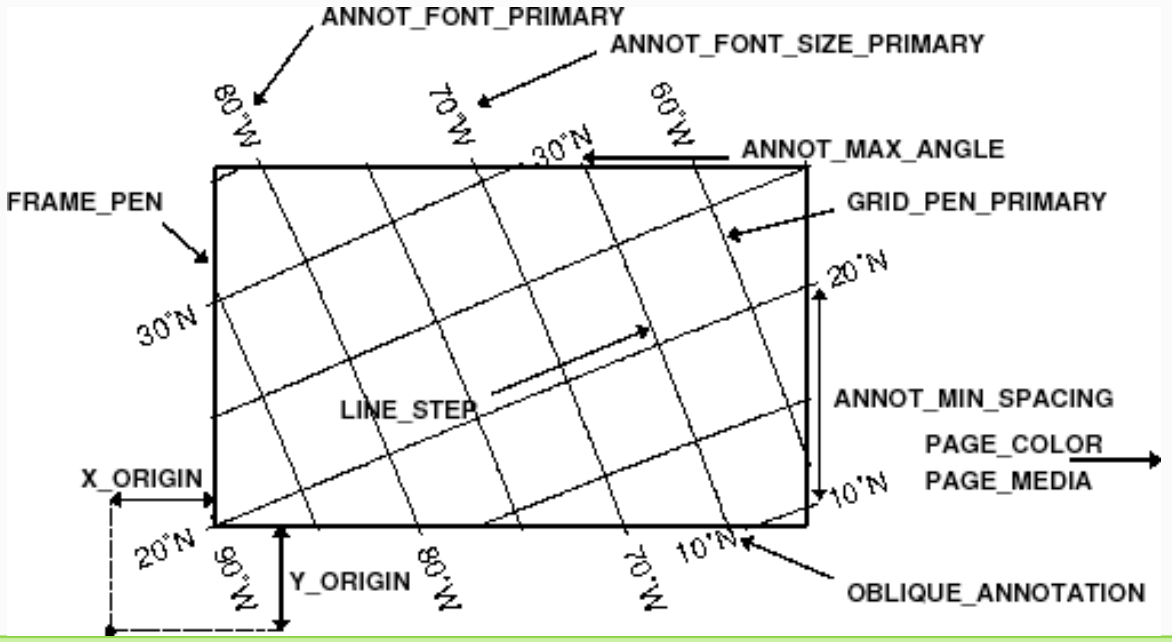
To view the list of options for each default parameter

```
%man gmtdefaults
```

Plotting Defaults

```
example of start
of .gmtdefaults4


#
#       GMT-SYSTEM 4.2.1
Defaults file
#
#-------- Plot Media
Parameters --
PAGE_COLOR
= 255/255/255
PAGE_ORIENTATION          =
landscape
PAPER_MEDIA
= letter
#--- Basemap Annotation
Parameters --
ANNOT_MIN_ANGLE
= 20
ANNOT_MIN_SPACING
= 0
ANNOT_FONT_PRIMARY        =
Helvetica
ANNOT_FONT_SIZE
= 14p
ANNOT_OFFSET_PRIMARY      =
0.075i
```





Changing the defaults

You can edit your local copy of .gmtdefaults4 using nedit or vim

You can explicitly reset a default within a script using the command gmtset

```
#!/bin/sh
gmtset PAPER_MEDIA  letter
```