A ridiculously short guide to GMT

GMT is a command-line orientated **G**eneral **M**apping **T**ool which is especially useful for plotting 'geo-data', mostly because of it's included global data bases of rivers, coast-lines and borders, and it's many implemented projection methods for data on a spherical surface.

If not already familiar with it, using GMT is a good reason and way to learn basic Linux commands, the C-Shell, and some (G)awk.

Availability and installation

GMT is freeware and can be downloaded from http://gmt.soest.hawaii.edu. An interactive install-form manages installation under Linux. For Windows installations, the best option is to install the freely available Cygwin software, which creates a Linux environment (including csh and gawk, and then install as under Linux. Running GMT under Windows without Cygwin is not advised due to the very limited possibilities of DOS batch files. Usage under Linux is highly recommended!

After installation, the variable GMTHOME has to be set adequately, the directory of GMT binaries has to be included into the system path, and the path to GMT man-pages should be included into the MANPATH. This can be done in the shell configuration file .tcshrc (executed each time tcsh is started), or better in the login configuration file .login (or .rlogin for remote login - executed each time you [remotely] login), all located in the user's home-directory, which e.g. on cagniard in Physics 605 does then take the form:

```
setenv GMTHOME /localdisk/sys/local/GMT3.4.1
set path = (/localdisk/sys/local/GMT3.4.1/bin .  $path)
setenv MANPATH /localdisk/sys/local/GMT3.4.1/man:  $MANPATH
```

Documentation

For Technical Reference and Cookbook (with good examples in section 6), a Tutorial, and the Man-Pages Online, see http://gmt.soest.hawaii.edu/gmt/gmt_services.html. When using GMT, calling the respective man-page from the terminal by typing man <GMT-command> will often help.

Usage

GMT is command-line driven, i.e. it could be run completely by typing everything in a terminal. But using shell-scripts instead for reproducibility is certainly the choice. A shell script (I recommend csh or tcsh, respectively – which is identical, since csh is linked to tcsh) must start with a line like:

```
#!/bin/csh
```

and must be set executable by typing e.g. chmod +x <script.csh> (type ls -al <script.csh> to see permissions. Run the script by just typing <script.csh> or, if the actual directory is not included in path ./<script.csh>. Defining variables within the script – e.g. the region of interest, or the name of the output file – can be wise to keep an overview and make changes easy.

Postscript-output is sent from stdout to the respective file by ">" for the first ps-output, and by "≫" for the followings. Options "-K" and "-O", i.e. "more postscript code will appended later", resp. "Overlay plot mode" have to be set accordingly to finally create a consistent PS file (first: -K, then: -K -O, finally: -O. See tutorial).

Running gmtdefaults -L > .gmtdefaults will create a file with a list of the variable settings used. This file can modified as wanted and is read by all GMT plot routines (for the meaning of the variables, see man gmtdefaults). Alternatively (recommended), including lines like

```
gmtset BASEMAP_TYPE = FANCY
```

in the script will work as well. Copying the script file to a new directory and executing it will then result in exactly the same plot.

One of the most useful commands is pscoast, which creates maps with coastlines, rivers and borders of a certain area of interest. To plot data in a 2-D plane, use psxy, which is also capable of plotting vectors or ellipses. pstext will plot texts of a specified size, format and angle on an arbitrary position in a 2-D plot. By default, GMT will put all plots on top of each other, e.g. first create a shaded relief map (grdimage), add geographic data like coastlines (pscoast), and then maybe station locations (psxy) with labels (pstext).

Here's a little example, which plots the UofA ABC map for MT:

```
#!/bin/csh

# variables
set outfile = map_ABC.ps
set r = "-125/-118/48/51.5"
set pro = "3"
# some settings
gmtset BASEMAP_TYPE = FANCY
gmtset PAGE_ORIENTATION = landscape
gmtset DEGREE_FORMAT = 3
gmtset FRAME_WIDTH = .04i
gmtset PAPER_MEDIA = letter
# plot topo information (see also below)
grdimage topo_ABC.grd -Itopo_ABC_i.grd -Jm$pro -CGMT_topo.cpt -R$r -K >
$outfile
# plot coastline data
pscoast -A0/0/4 -Ia -P -R$r -Jm$pro -Ba2f1/a2f1::WSen -S255 -Df -N1/3ta
-W1 -K -O >> $outfile
# plot station locations...
gawk '{print $2,$3}' ../coords/abc_sites.dat | psxy -R$r -Jm$pro -O
-Sc0.25 -L -G0 -K >> $outfile
gawk '{print $2,$3}' ../coords/abc_sites.dat | psxy -R$r -Jm$pro -O
-Sc0.2 -L -G255/0/0 -K >> $outfile
# ...with labels
gawk '{print $2" "$3" 8 90 0 LM "$1}' ../coords/abc_sites.dat | pstext
-Jm$pro -R$r -O >> $outfile
# proper boundary box
ps2epsi $outfile
```

Gawk interprets ASCII files row-wise, and can be very useful as shown above (see GNU website for a manual: http://www.gnu.org/software/gawk/manual/gawk.html. The pipe "|" redirects stdout from one program to stdin of the next – a classical Linux feature. The few lines in the example should illustrate the usage.

To look at the generated postscript files, use gv under Linux, or gsview under Windows (gs has to be installed first). Often, EPS-output files don't have the boundary box where we would like to have it. The Linux-command ps2epsi can often help in such cases.

The generated postscript file(s) can also contain several completely independent figures. Therefore, when one figure is finished, the next plotting routine can read an option -X and/or -Y to shift the plot origin on the paper by a chosen offset. This will be the origin for all following plots until the next occurrence of -X/-Y.

For much more and much more comprehensive information, see GMT Technical Reference, Tutorial & Man Pages.

Get & plot topography data

GTopo30

DEM (**D**igital **E**levation **M**odel) data of land topography in a 30 arc second resolution (0.9 km), organized in 33 tiles (see figure), are downloadable from the USGS: http://edcdaac.usgs.gov/gtopo30/gtopo30.html. The unzipped and untarred files contain the actual data files *.DEM (50-60 MByte), which have are in Sun-format with wrong byte-order for Intel-machines (PCs), and have to be converted via:

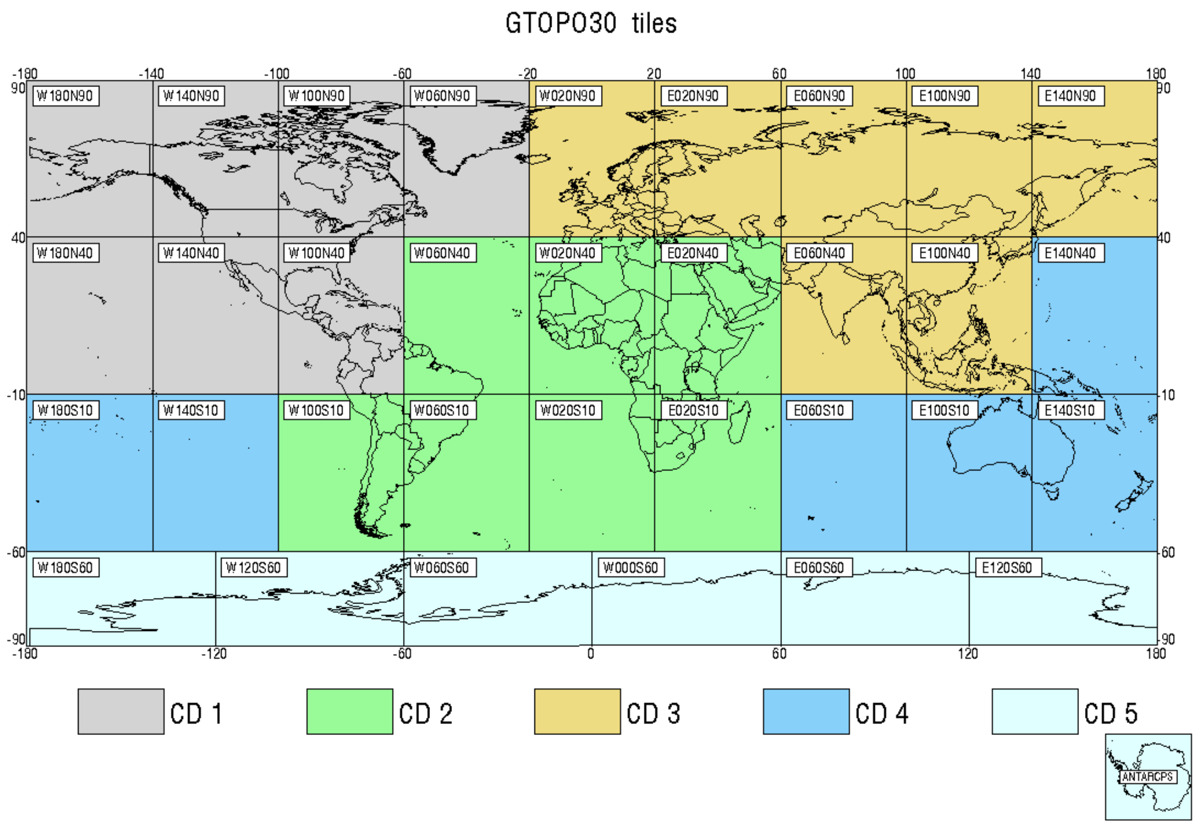
```
dd if=<infile.DEM> of=<outfile.DEM> conv=swab
```

To read the DEM files into GMT, put appropriate information into a file called grdraster.info, located usually in $GMTHOME/share/dbase by adding lines like

```
23 "GTOPO30 W100S10" "m" -R-100/-60/-60/-10 -I0.5m P i 1 0 -9999
/path/W100S10.DEM.
```

The data can then be read via grdraster and finally displayed as shown below:

```
# read DEM file
grdraster 23 -R<coordinates> -V -Gtopo.grd
# eventually resample to lower resolution (here:  8 Minutes)
grdsample topo.grd -Gtopo_out.grd -I8m
# calculate gradient for shaded relief plot
grdgradient topo_out.grd -Nt1 -A270 -Gtopo_grad.grd
```



Tiles of GTopo30 topography data.

```
# finally create image plot
grdimage topo_out.grd -Itopo_grad.grd -J<projection> -C<colormap-file>
-R<area-of-interest > out.ps
```

Colormap-files have typically the extension .cpt, and can be created by makecpt or grd2cpt. Useful colormaps – also for global topography data – can be found in $GMTHOME/share, i.e. are installed with the GMT program package.

GTopo30 plus estimated seafloor topography

ASCII files containing global topography and bathymetry data (estimated from satellite altimetry) with a ∼1 arc minute resolution can be downloaded from http://topex.ucsd.edu/cgi-bin/get_data.cgi. Gridding of the data via surface:

```
# create grid from ASCII-data
surface xyz.dat -Gxyz.grd -I1m -R<area-of-interest> -V
# rest as above.
```

SRTM

SRTM = **S**huttle **R**adar **T**opography **M**ission. Information on the project at: http://www.jpl.nasa.gov/srtm. Data resolution is 1 arc second for US and 3 arc seconds for other countries. Data are distributed in 1deg x 1deg tiles, which have to be unzipped, transformed and merged (using grdpaste). Data can be downloaded from the USGS: ftp://edcsgs9.cr.usgs.gov/pub/data/srtm.
Very helpful is the site: http://www.geo.arizona.edu/∼/rfr/Software/SRTM/, which also contains powerful perl scripts to do all the conversions. See the Readme file there. To plot data, either use plotgrd also found on this site, or, once the combined grid file is created, type:

```
# eventually cut out area of interest
grdcut srtm.grd -Gsrtm_cut.grd -R -V
# create grid with zeros above sea and ones on land
grdlandmask -R<area-of-interest> -Df -I3c -N0/1 -Gmask.grd -V
# multiply the two grids
grdmath srtm_cut.grd mask.grd MUL = srtm_final.grd
# gradient for shaded relief plots
grdgradient srtm_final.grd -Nt1 -A270 -Gsrtm_grad.grd
# eventually downsample, since files are presumably HUGE!!!
grdsample srtm_final.grd -Gsrtm_final_12.grd -I12c -V
grdsample srtm_grad.grd -Gsrtm_grad_12.grd -I12c -V
# and finally plot
grdimage srtm_final_12.grd -Isrtm_grad_12.grd -J<projection>
-C<colormap-file> -R<area-of-interest> >out.ps
```

It is advised to comment lines out once intermediate files are already created.