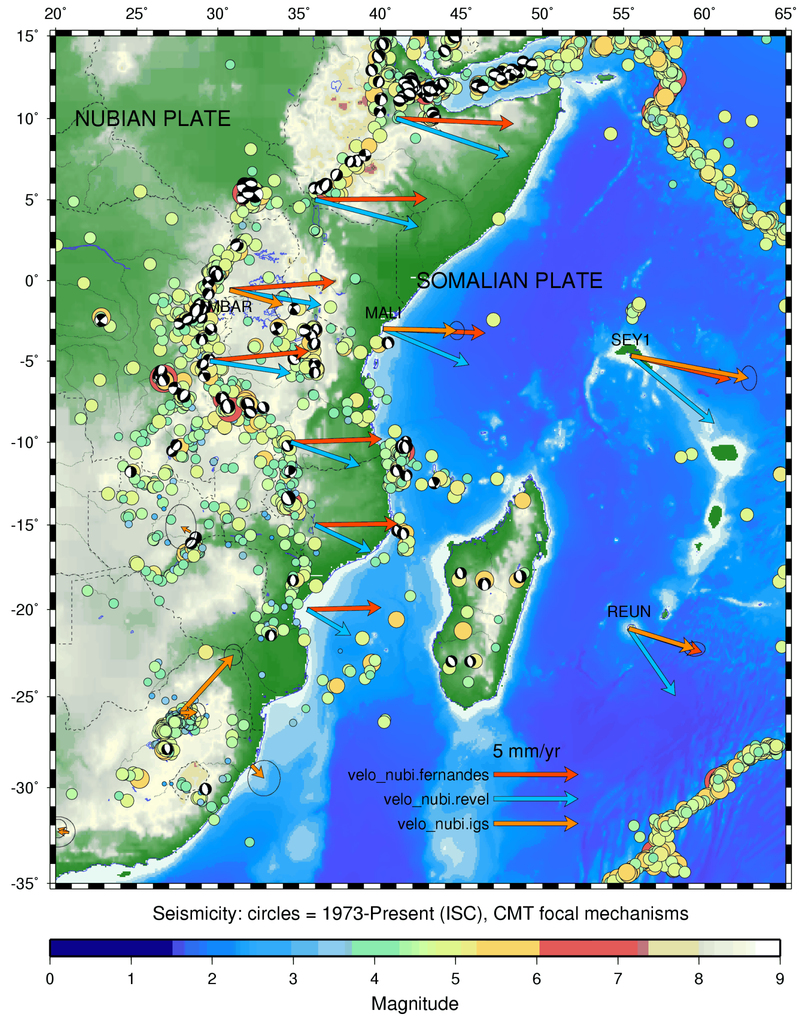
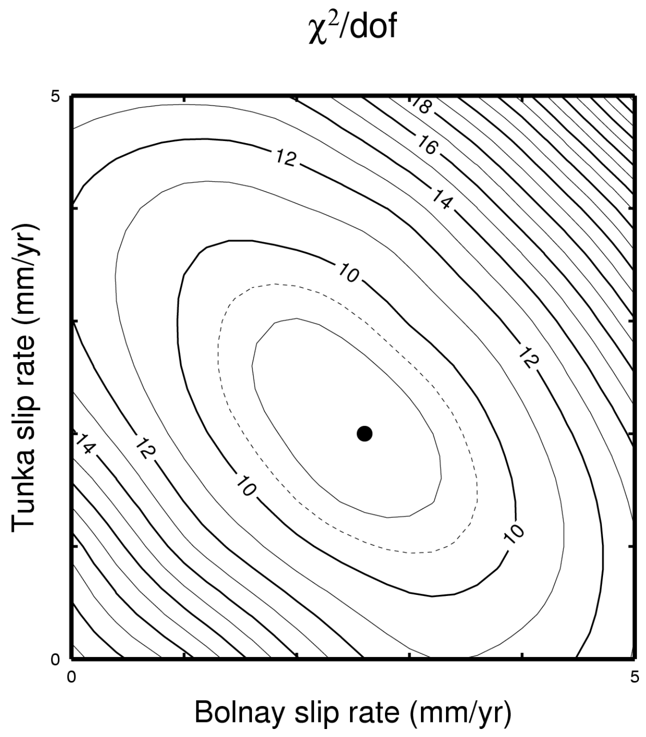
GMT: The Generic Mapping Tools

- GMT = a software **package** to create high-quality **postscript** maps and graphs in various projections.
- Output includes standard x-y-plots as well as complicated maps combined with other **geographical referenced data**.
- “Around 6000 scientists and engineers worldwide are using GMT in their work”
- GMT is a highly effective way for creating customized, professional looking maps or graphs.
- More information and on-line manual: **http://gmt.soest.hawaii.edu/**

Example output





GMT: The Generic Mapping Tools

- GMT comes as a set of more than 50 programs and tools, each of them performing a specific task.
- Most of the time, only 5-6 of these programs are used to plot maps or simple graphs.
- GMT programs are either called from the **command-line** or from **shell-scripts**.
- GMT commands can be **called** from you code (C, Fortran, etc.) or from shell-scripts

Your first GMT map

At the command prompt, type:

```
pscoast -R0/360/-70/70 -Jm1.2e-2i -Ba60f30/a30f15
        -Dc -G240 -W1/0 -P > GMT_mercator.ps
```

To display the resulting map, type: `gv GMT_mercator.ps`

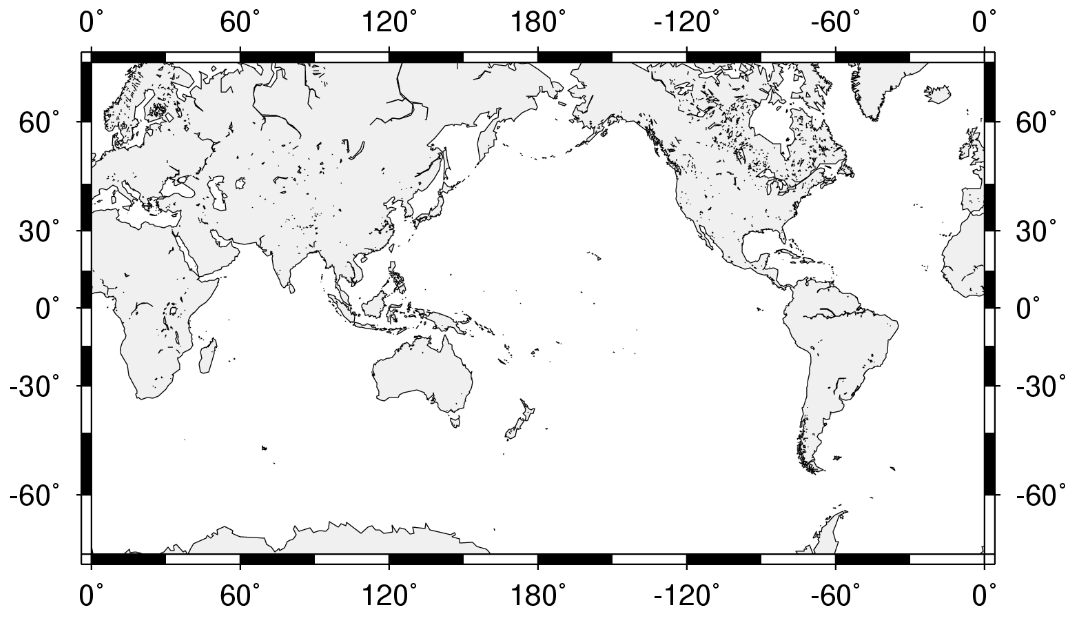


Figure 1: My first GMT map!

What did we just type?

- A GMT command to plot coastlines: `pscoast`
- Followed by a series of arguments in the form `-...`:
  - `-R0/360/-70/70` = select frame between longitudes 0/360 and latitudes -70/70
  - `-Jm1.2e-2i` = use Mercator projection (`m`) and a scale of 0.012 degree per inch
  - `-Ba60f30/a30f15` = annotate longitude borders every 60 degrees, latitude border every 30 degrees, fill longitude borders every 30 degrees, latitude border every 15 degree.
  - `-Dc` = use a crude resolution for plotting coastlines
  - `-G240` = color landmasses in light grey (0=black, 255=white)
  - `-W1/0` = draw coastlines with a 1 point-wide line (i.e. extra thin) in black
  - `-P` = plot in portrait mode (GMT default is landscape)

Displaying postscript

- There are several standard tools to display postscript, usually available on most unix systems:
  - **ghostview**: `gs`
  - **ghostscript**: `gv`
  - **ImageMagick**: `display`
- Note that GMT commands can be directly *“piped”* into `gv` for instance:

```
pscoast -R0/360/-70/70 -Jm1.2e-2i -Ba60f30/a30f15
        -Dc -G240 -W1/0 -P | gv -
```

| (vertical bar) means that the output of GMT is directly fed into (= “piped” into) gv.

Unix

- UNIX is an operating system, i.e. it manages the way the computer work by driving the processor, the on-board memory, the disk drives, keyboards, video monitors, etc. and by performing useful tasks for the users
- UNIX was created in the late 1960s as a multiuser, multitasking system for use by programmers.
- The philosophy behind the design of UNIX was to provide simple, yet powerful utilities that could be pieced together in a flexible manner to perform a wide variety of tasks.

Unix: basic commands

- login, logout, work environment
- Current directory? `pwd`
- Creating a new directory: `mkdir directory`
- Changing directory:
  - Go to home directory: `cd` or `cd ~user_name`
  - Go to directory /home/users/ecalais/work: `cd /home/users/ecalais/work`
  - Go to directory one level below: `cd ..`
- List the content of a directory: `ls`
  - List all files (includind those starting with a .): `ls -a`
  - Show details (ownership, date, etc): `ls -l`

Unix: basic commands

- Create empty file: `touch file1`
- Copying a file: `cp file1 file2`
- Moving (= renamimg) a file: `mv file2 file3`
- Removing a file: `rm file3`
- Viewing files:
  - `cat file_name`
  - `more file_name`
- Editing files:
  - `vi file_name`, `emacs file_name`
  - `edit file_name` (opens a new window)
- Manual pages: `man unix_command`

Unix: basic commands

- Connect to remote computer: `ssh username@remote.domain`
- Transfer files between computers by `ftp`:
  - Establish connection with: `ftp computername.domain`
  - For secure connection use: `sftp computername.domain`
  - “Anonymous” ftp: `ftp computername.domain`, username = `anonymous`, password `your_email_address`
  - Change directory on the server: `cd directory`
  - Change directory on the host: `lcd directory`
  - Transfer in binary mode: `binary`
  - Download a file: `get file`
  - Upload a file: `put file`

Unix: variables

```
set day = 1
echo $day
echo $day > junk
echo $day > /dev/null
@ day = $day + 1
echo $day >> junk
cat junk
```

Note that:

- > redirects the output of a command to a file. If the file did not exist, it is created. If the file already existed, it is overwritten!
- >> appends the output of a command to a file. If the file did not exist, it is created. If the file already existed, the output is appended.

Unix: if

```
set day = 2
if ($day == 2) then
      echo you win
else
      echo you loose
endif
```

Try with day = 1 ...

Unix: while / foreach

```
set day = 1
while ($day < 10)
  echo This is file $day > file.$day
  @ day ++
end

foreach f (*)
  echo This is file: $f
end
```

Unix: grep

```
echo TOTO > junk
echo TATA >> junk
echo TITI >> junk
cat junk

grep TATA junk
grep TATA junk | awk ’{print substr($1,1,2)}’
set TA = ‘grep TATA junk | awk ’{print substr($1,1,2)}’‘
echo $TA
```

Unix: background/foreground processes, kill

```
gv
ˆC             (control-C)

gv
ˆZ             (control-Z)
bg
jobs -l
kill job_number

gv &
jobs -l
kill job_number
```

Unix: background/foreground processes, kill

```
gv &
ps -elf
ps -elf | more
ps -elf | grep ecalais
ps -elf | grep gv
kill job_number
```

Running CSH scripts

- Run your script: `csh my_script.csh`
- Make your script executable and run it:

```
ls -al my_script.csh
chmod +x my_script.csh
ls -al my_script.csh
my_script.csh
```

Your first GMT script

- Create a script file *gmt1.csh* with the following content:

```
pscoast -R0/360/-70/70 -Jm1.2e-2i -Ba60f30/a30f15
        -Dc -G240 -W1/0 -P > GMT_mercator.ps
gv GMT_mercator.ps &
```

- Run it using: `csh gmt1.csh`
- Or make it executable first: `chmod +x gmt1.csh`
- And then run it: `gmt1.csh`

Your second GMT script

Let’s plot the same map as before twice on the same page, shifted vertically by 4 inches. You GMT script `gmt2.csh` looks like:

```
pscoast -R0/360/-70/70 -Jm1.2e-2i -Ba60f30/a30f15
        -Dc -G240 -W1/0 -P -K > GMT_mercator.ps
pscoast -R -Jm -Ba60f30/a30f15 -Dc -G240 -W1/0
        -O >> GMT_mercator.ps
gv GMT_mercator.ps &
```

Run your script using: `csh gmt2.csh`

Or make it executable first: `chmod +x gmt2.csh`

And then run it: `gmt2.csh`

Your second GMT script

Note that:

- The contents of -R and -J do not need to be repeated
- The first line **creates** file `GMT_mercator.ps` (with `>`), the second line **appends** to that file (with `>>`)
- `-K` means that more code will be added later: therefore, every GMT command, **except the last one**, must have `-K`
- `-O` means overlay on top of previous command: therefore, every GMT command, **except the first one**, must have `-O`
- `-P` (for portrait mode) does not need to be repeated

Assignment

Using a csh script, create on the same page 4 maps of North America ($20 < lat < 65$ and $-140 < lon < -50$) using:

- A Mercator projection, grey land masses, white oceans, black coastline with crude resolution, lat/lon borders annotated every 20 degrees and filled every 5 degrees
- Same as above, but light brown land masses, light blue oceans, intermediate resolution coastlines, a 1500 km long map scale located in the bottom right corner of the map
- Same as above, with all major rivers in blue pen, state boundaries in dashed solid black, country borders in solid red, coastline in dark blue.
- Same as above, using a Lambert projection, without the map scale, with a title, and the lat/lon annotations along the S and E sides only.

Assignment

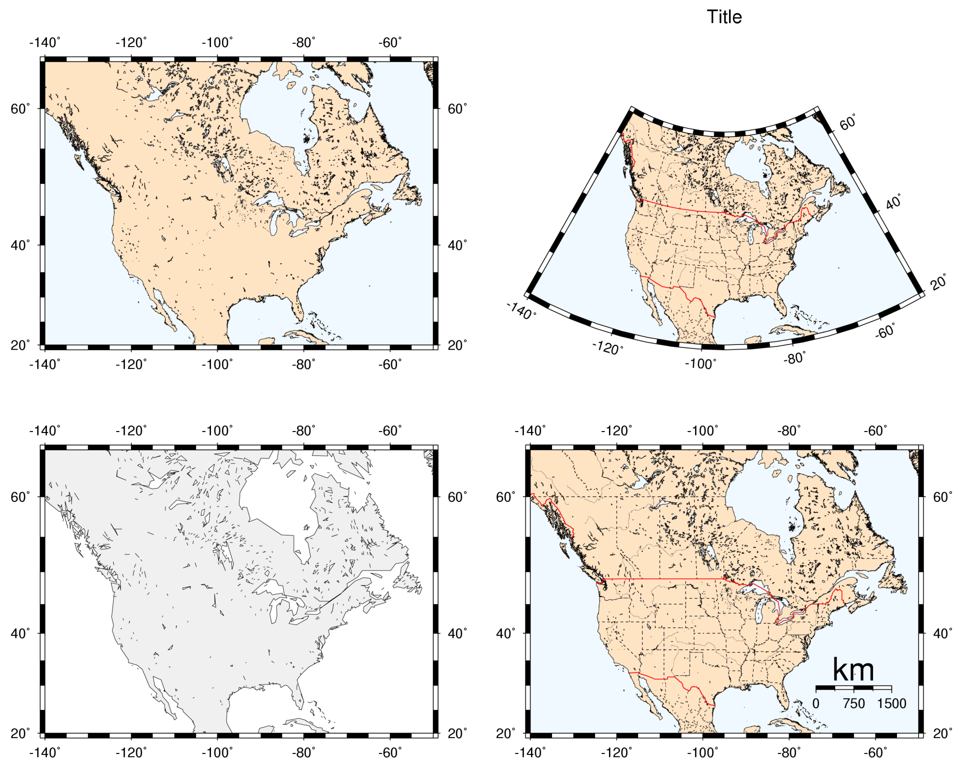


Figure 2: Your output should look like this...