Department of Earth and Planetary Sciences

Northwestern University

Winter 2016

An Overview of

CATANIM

v1.0

Amir Salaree

Contents

1 Foreword

Movies are great since they can nail the most intricate ideas in your mind through clever visualization. This characteristic of movies has long been known and studied from the dawn of cinema to modern days. Movies are awesome for education (imagine yourself trying to teach things like wave propagation, Darcy's law, RLC circuits, or subjects as simple as kinematics of free falls of a bouncing ball to a bright young mind who just hates spending much time on a topic) and are amazing tools in research when you are looking for a comprehensive view of a phenomenon (think of boundry condition problems or manifold embedding).

As a seismologist, I have always been tempted to make good movies and figures to communicate what I am interested in. With seismic mapping and fault geometry as one of my research interests, I became involved in studying earthquake/aftershock distributions in crustal profiles within faulting regimes. Well, making seismicity maps in 3D is easy. All you need is an XYZ dataset of your events and a program (such as GMT or some of Python libraries) to make a map. It is less straight-forward however,to look at seismicity in *time*. Therefore, I found it interesting to think of some way to make a fast-forward view of seismicity in time – something that I can make once and then keep using forever.

CATANIM (Catalog Animator) is a first step in automating the tedious and cumbersome process of creating an animated map visualizing the sequence of earthquakes in a given geographic area. Anyone who has studied earthquake aftershocks will agree that the $3-$D and $4-$D visualization of aftershock sequences is crucial in understanding seismic cycles. Writing scripts and codes to do the animation, even though very simple in essence, requires some scrutiny and a great deal of patience. Although the visualization of such datasets is getting more and more straightforward everyday with the advent of complex MATLAB/Python libraries, it still suffers from a sometimes annoying weakness: *speed*. Don't get me wrong! These are state-of-the-art algorithms written by the brilliant for the smart, but the overall process is doomed to slow down by the nature of scripting languages.[1]

[1] I am sure someone will say: "no! I know a very fast animating software". But that psuedo-speed is usually within the animating process and not in the sifting algorithms.

Well, this was my first motive in compiling the CATANIM package. I wrote the first draft of the code trying to tackle the annual distribution of events at the Mid-Continent Rift (MCR) during the instrumental era and making a movie of the earthquake occurrences in time. However, I thought why stop there? So, I extended the viewing intervals from annual to monthly and then – what the heck?! – daily snapshots. The code was very very simple and all it needed to happen was some time to actually do the chore. The coding step was tiring – remember, I was doing it in FORTRAN – but I was going to do it once and for all.

Alright, let's talk business! There are a bunch of pros and cons while using CATANIM. The package – being written in FORTRAN – is a victim to formatting, that is the – one and only – input file must be in a special format (which I call the *cat* format) and is the standard output of many of my catalog codes. I guess this is the price you pay for the life you choose.[2] Also, this package is not stand-alone. You will need some third party programs. These programs are free to acquire and I guess are commonly used by the scientific community – in fact, many systems already have them ready to use.

Nevertheless, CATANIM should be easy to configure on any UNIX system and benefits from the speed of FORTRAN. I also have included several output formats for different purposes: educational, publication, fun, etc.

A few – among the many – new options that still can be added to the package are *arbitrary snapshot intervals*, *cross-section view for event depths*, *spatial anaylses*, and *seismic hazard/risk evaluation*. I will always be open to suggestions...

I feel immensely indebted to Seth Stein who gave me the first motivation for doing this and to Chuck Ammon who first taught me (among several other seismologists during an IRIS workshop at the University of Indianna at Bloomington) the concept and advantages of good data visualization.

Happy Animating!

Amir Salaree

Winter 2016

[2] I love the Godfather movies after all!

2 Getting It to Work

You will need the following software components to properly compile and run the CATANIM package:

1. UNIX-based system (already tested on RedHat and some of Macintosh distributions).
2. A version of GCC-GFORTRAN compiler (really anything more recent than FORTRAN-90).
3. A version of ImageMagick (it usually exists as a part of some Linux distributions; there are multiple – simple – ways to get a copy[3]).
4. GMT 4.X (slight modifications are needed if you are running the GMT 5.X or more recent versions.[4])
5. FFMPEG: This is needed for making the movie (there are various methods by which you can acquire this program[5]).
6. GNUPLOT: This is needed for some of the statistical plots (it usually exists as a part of some Linux distributions; there are multiple –simple– ways to get a copy[6]).

It may seem a lot, but many of the above requirements already exist on a student's computer.

2.1 Compiling

In the main directory (where you have this text file) run the *compile* script:

```
[user@computer CATANIM]$ ./compile
```

You may have to change the permission settings to be able to run the script (e.g. chmod 776 compile).

[3] See `http://www.imagemagick.org`
[4] Consult the GMT online documentation
[5] See `www.ffmpeg.org`
[6] See `www.gnuplot.info`

2.1.1 Adding to the Working Environment

You can edit your environment to call the program easier. In *bash*, you can add the following line to the ~/*.bashrc* file:

```
PATH=$PATH:/path/to/the/bin/directory/
export PATH
```

2.2 Input and Outputs

The only input the program needs is a catalog file containing the temporal and spatial information of the earthquake as well as their magnitude. This must be in a special "*cat*" format:

- 15 header lines
- event number,year,month,day,hour,minute,second,latitude,longitude,depth,magnitude
  (i5, i6, 2i4, i5, 1x, i2, 1x, f6.3, 2f9.3, 2f7.2)
  + There are blank lines after every 5 lines of event data.

See the following example:

```
                                                                  USGS CATALOG
                                                                 Rectangular Window

Minimum Latitude =   24.000   ,  Maximum Latitude =   42.000
Minimum Longitude =   43.000  ,  Maximum Longitude =   65.000
Minimum Depth =    0.000      ,  Maximum Depth =   1000.000
Minimum Magnitude =  0.0      ,  Maximum Magnitude =  10.0
Minimum Year = 2006           ,  Maximum Year = 2015

--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------

  No  YEAR MON DAY       TIME        LAT      LON     DEP     MAG TYPE            LOCATION                          EVENT TYPE

   1  2006   1   6   17:41:45.330   26.895   54.529   9.20   4.50  mb      southern Iran                              earthquake
   2  2006   1   6   20:07:01.700   31.613   50.521  10.00   3.40  mb      western Iran                               earthquake
   3  2006   1  12   21:22:45.900   32.851   49.332  15.00   4.60  mb      western Iran                               earthquake
   4  2006   1  12   21:32:32.700   26.886   55.903   7.40   4.40  mb      southern Iran                              earthquake
   5  2006   1  14   23:23:08.000   35.050   56.850  18.00   4.40  mb      northeastern Iran                          earthquake

   6  2006   1  16   09:18:06.000   34.760   45.040  10.00   4.50  mb      Iran-Iraq border region                    earthquake
   7  2006   1  17   18:16:49.830   40.506   48.847  10.00   3.20  mb      Azerbaijan                                 earthquake
```

```
.
.
 2396  2014  12  30   22:10:17.900   28.729   51.893   7.10   4.40  mb      67km W of Firuzabad, Iran                  earthquake

  ---------------------------------------------------
 823262   events read.
   2396 events matched your search criteria.
```

This is a very simple format to set up. Please contact me if you need a version of the catalog program.[7]

After successfully running the *catanim* code, several file types will be created.

I. events.XYYYYMMDD.xy

These are the files containing the events on the day XYYYYMMDD. Value of X can be either "m" (for 'minus' years [=BC]) or "p" (for 'positive' years [=AD]). Obviously, there will be empty files for days with no recorded events.

II. gmt.XYYYYMMDD

GMT scripts to make postscript images of the epicenters.

III. cat.sec This is the UNIX time of individual events and could be useful when studying temporally compact distributions – say, 1000 or more events in a day.

IV. gnuplot.stat.in This gnuplot script can be used to create an statistical summary of the dataset.

V. log.yr, log.mn, log.dy These files (not so useful on themselves) include the current year, month and day respectively.

VI. (mag.yr,stat.yr) or (mag.mn,stat,mn) or (mag.dy,stat.dy) Depending on the snapshot intervals you pick (yearly, monthly, or daily), you will end up with a pair from above.

VII. XYYYYMMDD.ps and XYYYYMMDD.eps Postscript images of the epicenter snapshots. Both ".ps" and ".eps" formats are produced to satisfy different documentation needs.

VIII. EVENTS.XXXXX.jpg These are ".jpg" versions of the postscript images. XXXXX is a sequential value between "00000"" and "99999".

[7] amir@earth.northwestern.edu

IX. EVENTS.avi and EVENTS.mp4 The ".avi" and ".mp4" versions of the movie. You may want to choose one of the two as there is a chance that hardware acceleration is disabled on your machine or you simply may not have the required codecs.

3 How It Works

The program actually runs in two steps. It assumes that you already have a catalog that you want to plot (see 2.2).

3.1 Sifting Part:

After compiling the catalog file, simply run:

```
[user@computer CATANIM]$ catanim filein
```

where *filein* is the catalog. You will see some basic statistics of your catalog. Then you will be prompt to choose a snapshot interval from *year*, *month*, *day* and *anything else*[8] for which you should a number between 1 and 4.

After that, you will be given the choice of running the code for the entire catalog or a specific time window. If you type in "1", then the code will start the analysis/movie at the beginning of your catalog and end it on its final day. However, if you choose "0", you will be asked for the beginning and the end of your working period. Just remember that the program only works with full year integers (e.g. 1994 rather than 94). For BC (negative) dates (as in your catalog files), use negative years.[9].

[8] The last option is not functional yet.

[9] This of course if you have the luxury of a good paleoseismic catalog

3.2 Statistics:

After a successful run, you can run the gnuplot script to visualize the distribution of events. Simply run:

```
[user@computer CATANIM]$ gnuplot < gnuplot.stat.in
```

A postscript file named “stat.ps” will be created (see the example in Fig. 1). You may want to change some of the parameters in the gnuplot script to either make it compatible with your version of gnuplot or to modify it according to what you need. You can make JPG, EPS, etc. versions of this figure using numerous types of third party softwares. A possible upgrade to the package could be combining this graph with the epicenter movie. You can also use the “events.mag” and “events.dep” files to make histograms of the magnitude and depth distributions.

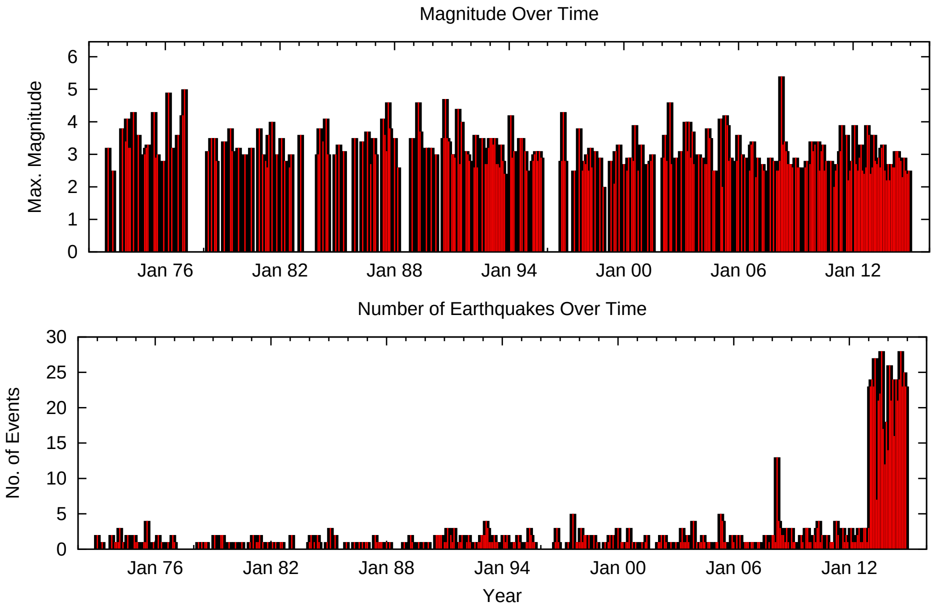


Figure 1: Top: Monthly maximum magnitude of the events between 1973 and 2014 for the MCR from the USGS catalog. Bottom: Monthly number of events in the same time period.

3.3 Making the Movie

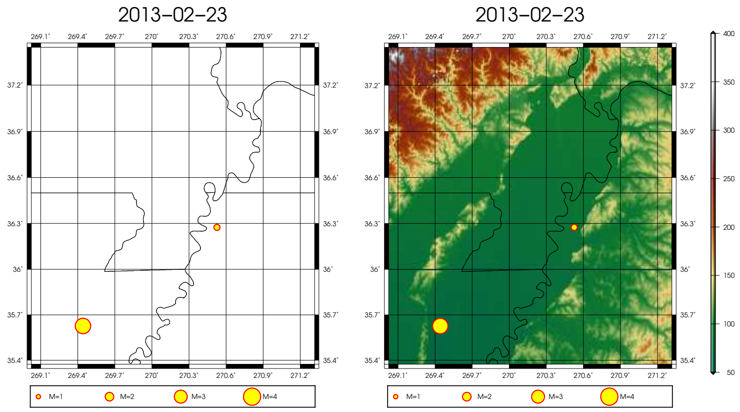
So far, you have not made any maps or movies. To do so, run the "catmovie" script in the "bin" directory of the package to make a video of your snapshots. If you have added "bin" directory to your environment just type:

```
[user@computer CATANIM]$ catmovie
```

The final map is not envisioned to have embedded topography (due to the hassle of getting a suitable raster to work on everyone's machine), but this would be an easy fix either in the main code or the CATANIM script. I would add something like

```
[user@computer CATANIM]$ grdimage topo.grd -Ctopo.cpt -R$range -J$proj -P -K -Y7 >$out
```

at the beginning of the GMT scripts. Of course, you should have your own CPT file. If you don't have one at hand, you can always play around with the *makecpt* command in GMT to create a file suitable for your region of interest. You can also change the legend.xy file to suit your projection. Fig. 2 is an example snapshot from a movie I made for the MCR. You can always add other features such as faults and cities to your map by editing the GMT scripts which is easily done through the controller script (see 3.3.1).



(a) (b)

Figure 2: (a) A snapshot of a movie I made for MCR. (b) The same map with embedded topography.

3.3.1 How to Adapt the CATMOVIE Controller

While you may run the CATMOVIE script as is, you may also need to change some of the details to get better maps/movie and as a result will need to have some idea about the structure of CATANIM.

CATANIM is a very simple script with a – deliberately – very simple structure. Some useful variables are:

- **proj** Sets the map projection you will use. The original projection is Mercator "m1i", and is by default changed to "m0.2i" for an $\sim 20^\circ \times \sim 20^\circ$ region. You can change the value of the *proj* according to your region of interest.
- **space** Sets the gridline spacing of the map. The original spacing is 1° and is by default changed to 3° for an $\sim 20^\circ \times \sim 20^\circ$ region. You can change the value of the *proj* according to your region of interest.
- **margin** Sets the thickness of event margins. The original thickness is 3 points and is by default changed to 6. You can change the value of the *proj* according to your resolution of interest.

3.3.2 Important Notes

1. If you are using GMT 5.X rather than GMT 4.X, uncomment lines 22–25 in the CATMOVIE script.
2. Some basic knowlege of GMT and bash scripting may be needed to make a good movie. As such, you may have to make some changes to the GMT files from the CATMOVIE cotrol script. For example, you may want to change "legend" resolution, or map projection which are possible by making a few changes to the CATMOVIE file.
3. Postscript (EPS and PS) files are created for the entire time period of your choice and there are also JPG versions (see 2.2).
4. You should now have AVI and MP4 versions of the movie. You may want to choose one of

the two as there is a chance that hardware acceleration is disabled on your machine or you simply may not have the required codecs. Deinterlacing of frames may cause some issues with WMP[10] family softwares while the VLC family might function properly.

5. You may want to delete the JPG and either of the postscript files that you don't use since they might eat up to 3GB of your hard drive.
6. You may face some hassle playing the movie on one machine and not the other with some nasty error message from the player software complaining about the supported resolution. This is due to the graphical settings of that machine. You can avoid this by making a lower quality movie. To do this, change the "res" variable on the 51st line of the catmovie script (res=200 or res=100 are good starting points).
7. You may end up with an static image when you play the created movie. This sometimes happens when your hardwar and/or software don't agree with the frame rate and/or interlacing of the movie maker program. You can play around with the "FP" variable on the 56nd line of the catmovie script (FP=8 is a good start).

3.4 Comments on Speed

CATANIM should be very fast for *usual* animating processes, that is if the catalog size (number of events and days) is not super large. It also will depend on how fast the processing unit of the machine on which you are running the code is. Also, I would expect that the actual movie making process will take longer than the event sifting. For instance, for a year catalog with $\sim 11,000$ events while the sifting step is done in no time (~ 10 seconds), the animating step takes much longer (~ 10 minutes). At the same time, the sifting process for a 6-year long catalog with $\sim 63,000$ events can take up to ~ 6 minutes (Fig. 3).

[10] Windows Media Player

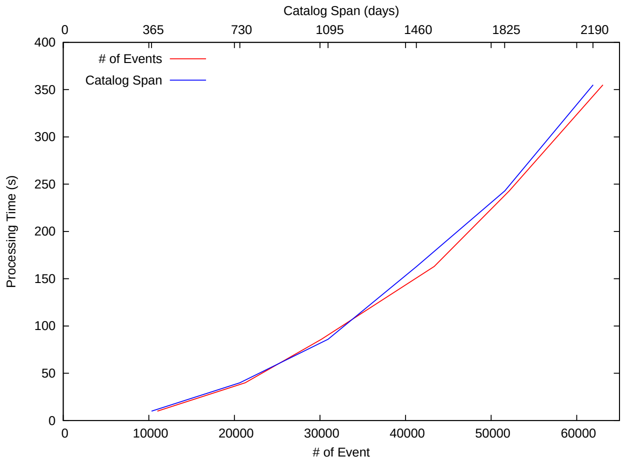


Figure 3: Examples of processing times for various catalog population. Obviously the distribution is not uniform due to catalog incompleteness.

Overall, making such movies is an inherently slow process and unless some better algorithms are developed, one should spend some time finding its hiccups and intricacies.

3.5 Free Style Comments on Movie Making Sofware

Making movies is fun! There are tons of programs that create movies. These programs range from free (usually Linux style) quick and dirty ones to several-grand commercial ones developed by movie business professionals. Which one you will need depends, to a large extent, on what you want to do and the video format you are thinking of. If you want to make a high quality video and have some extra bucks, you may think of some Adobe products (such as After Effects®) or similar softwares.

Sometimes, specially when making easy-to-load web pages, it is best to make an animated GIF image which does not need any specific codecs, instead of a full-fledged video. “Gimp”, “gifsicle” and ”ImageMagick” are some easy-to-use examples. Also “ghostscript” has some built-in functions to make GIF videos.

FFMPEG (which I have used here) is a handy tool to make animations. Check out the documentation for FFMPEG to get some idea of its capabilities. You will probably be amazed by how much this tiny little package can do.

4 Epilogue

This code, however handy or crude it might be, is simply a first attempt made in my spare times. Many of its aspects could be improved, changed or removed and there is always room for new options.

Looking forward to hearing about your suggestions and complaints!