**A guideline to generate stochastic earthquake scenarios for tsunami modelling   
[the R-PTHA code version]**

Prepared by Ryan (29 October 2020)

In this tutorial, I will give you a tutorial on how I prepare a set of stochastic scenarios using the PTHA code from Geoscience Australia. The code is freely available from this link: <https://github.com/GeoscienceAustralia/ptha.> You can follow the guideline given to compile and build the code.

There are three main steps:

1. Subfaults preparation
2. Generate unit sources displacement
3. Generate stochastic scenarios

I use those three steps above to generate a set of tsunami Greens Functions on the JAGURS code, which is also freely available from this link: <https://github.com/jagurs-admin/jagurs>. This Greens Function would be useful for analysing off-shore tsunami hazard from thousands scenarios. Alternatively, you can just run forward modelling for all (or selected) the stochastic scenarios. Compiling JAGURS is a little bit tricky due to the libraries requirement. I will show you on how I compiled JAGURS on my Ubuntu Mate 20.04 machine at another chance.

Software and/or program needed:

* R- for the PTHA code
* QGIS

Let’s do it!

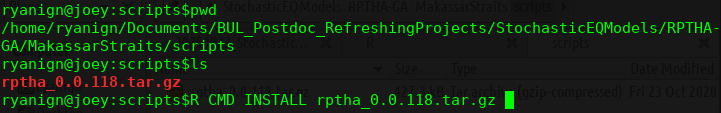
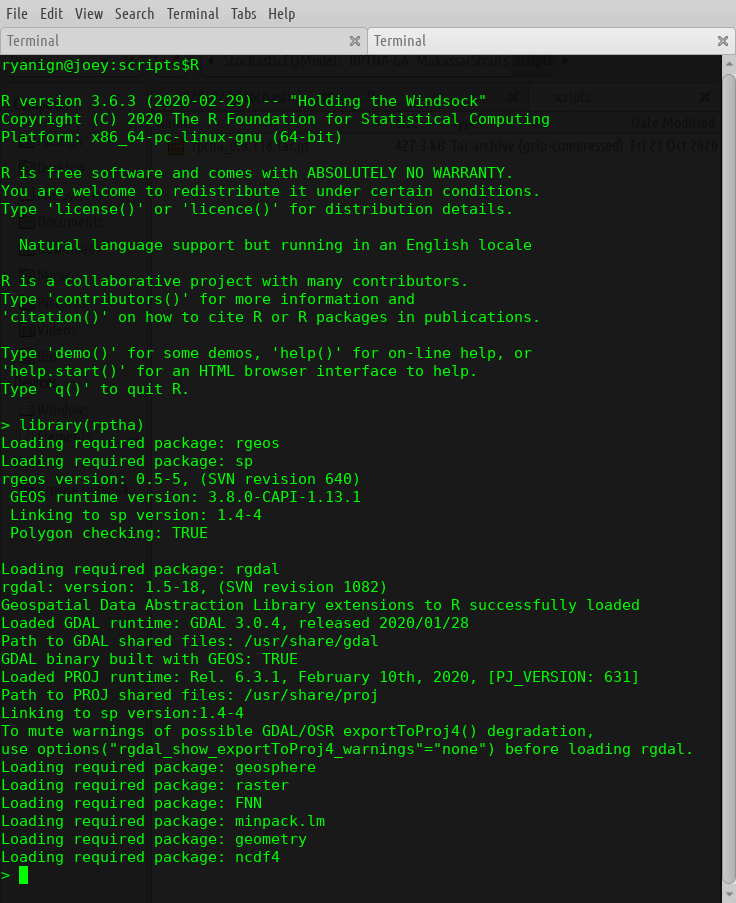
For this tutorial, I will use the Makassar Thrust Fault from the central segment ([Irsyam et al., 2020](https://journals.sagepub.com/doi/full/10.1177/8755293020951206?casa_token=yfVMNlqPmCUAAAAA%3AwsWdN_lSa0IHs1KGOoRYCRs57PnyXMdIMiKS9oSoRsCKTXCwBRH6p6USAl-AIt1myFR_kyqD0K9AQA&)) as an example.

**General preparation**

1. Create a working directory (e.g. MakassarStraits). Inside this, create *scripts* directory to store scripts needed.

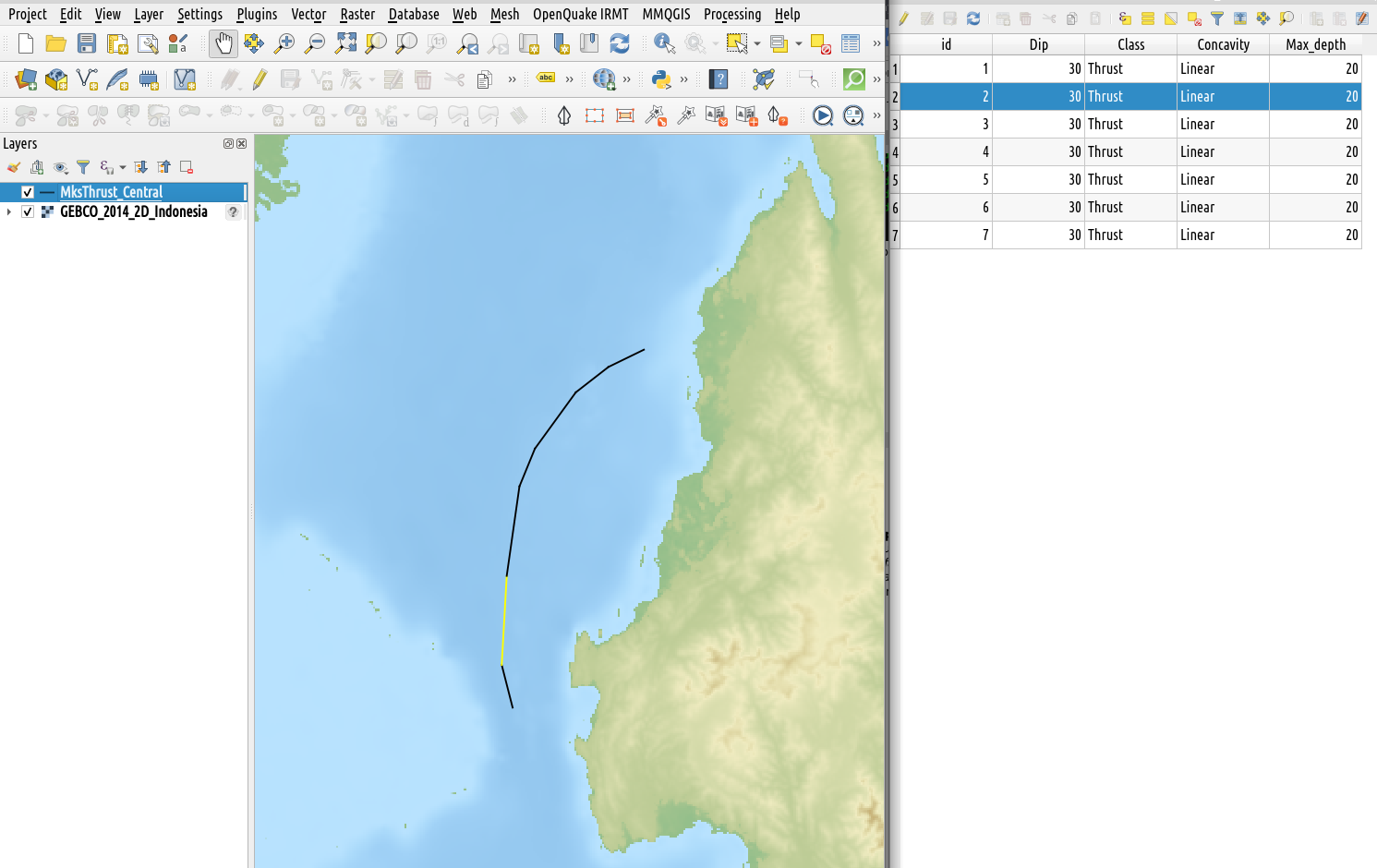
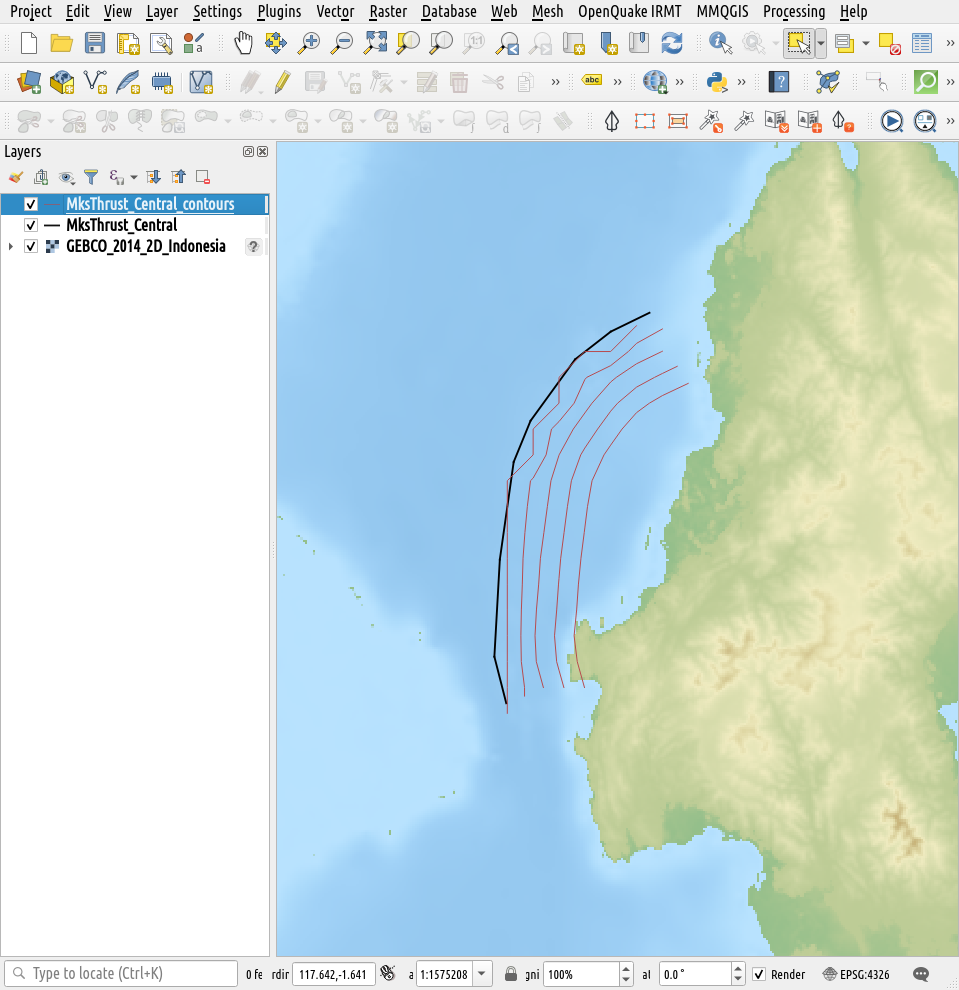
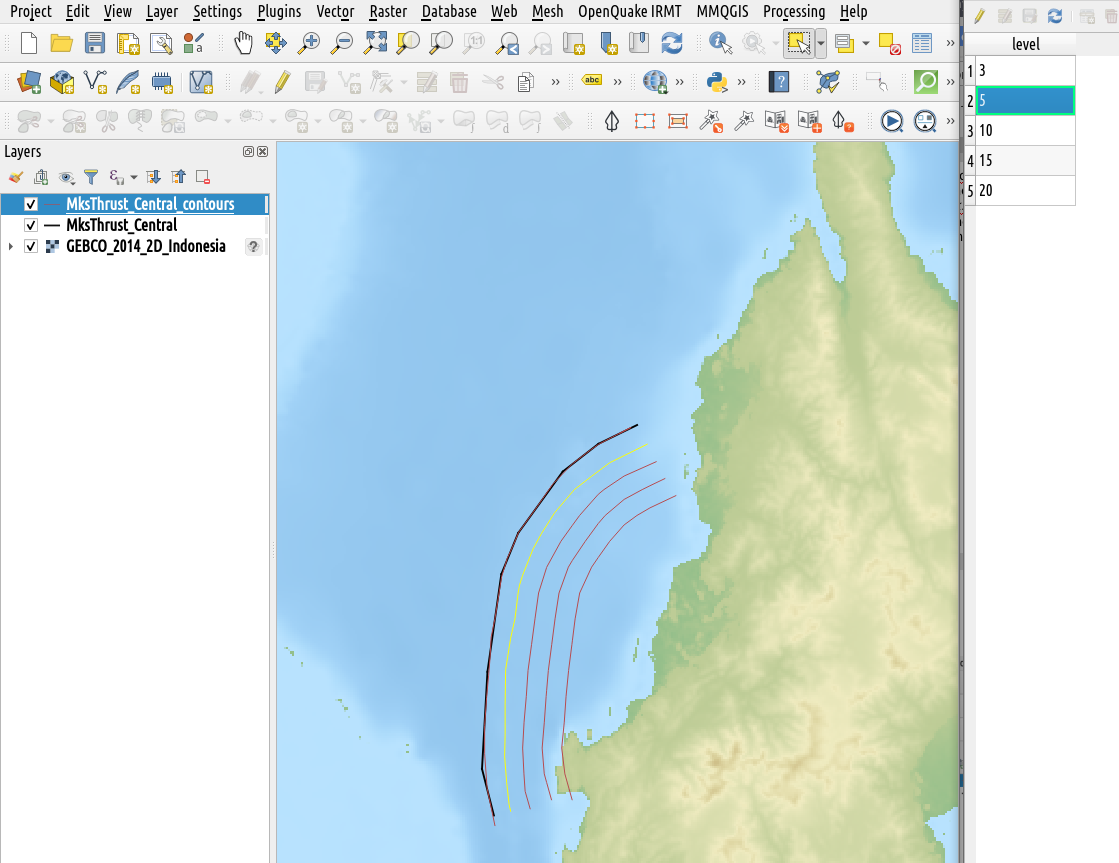
* Inside *scripts* you will need:
  + 01\_run\_convert\_traces\_to\_contours.R
  + 02\_make\_initial\_downdip\_lines.R
  + 03\_produce\_unit\_sources.R
  + 04a\_stochastic\_tsunami\_sources.R
  + collate\_source\_traces.R

1. Compile the PTHA code inside the *scripts* directory.

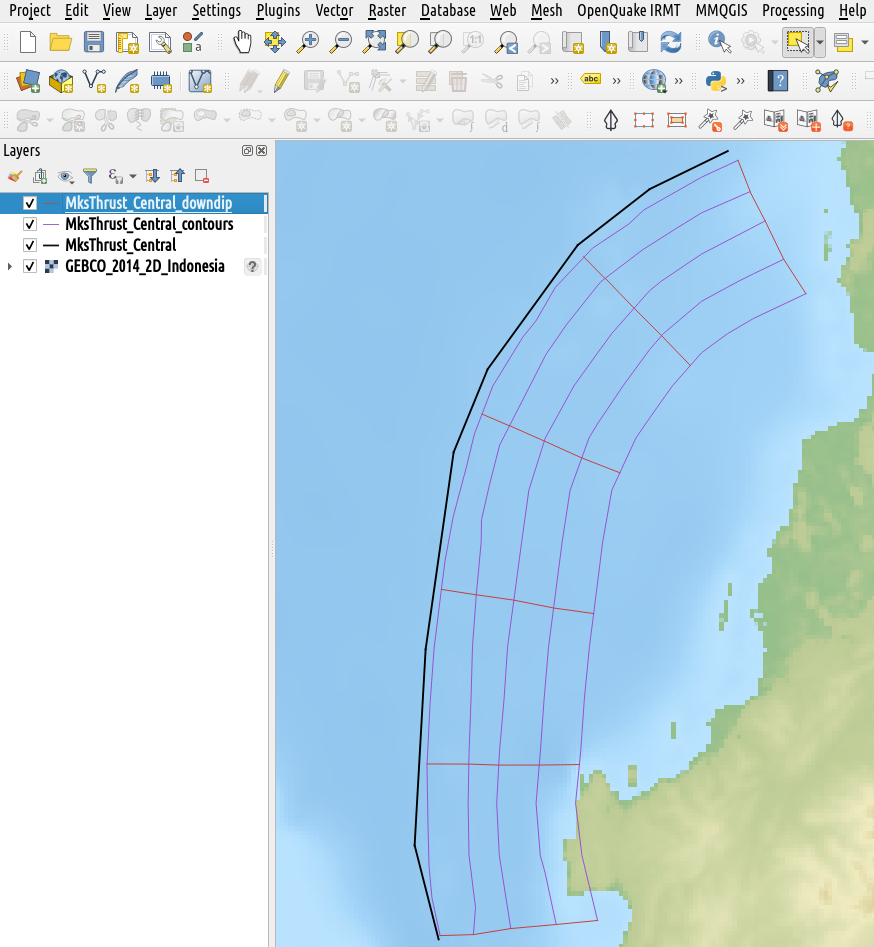
* If you already have rptha\_X.X.XXX.tar.gz from others, you could just do command below from your terminal:
  + $R CMD INSTALL rptha\_X.X.XXX.tar.gz
  + 
* When it finished without error, check whether the code is successfully compiled.
  + $R <enter>
  + >>library(rptha) <enter>
  + 
  + If you got something like above, it means no problem so far.

**Subfaults preparation**

1. Create a fault contours shapefile (if you don’t have it)

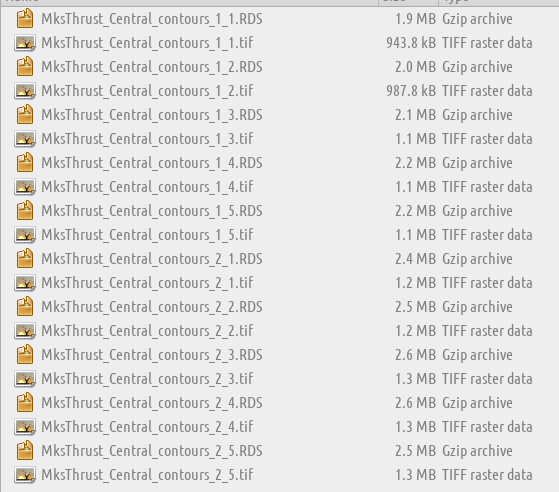
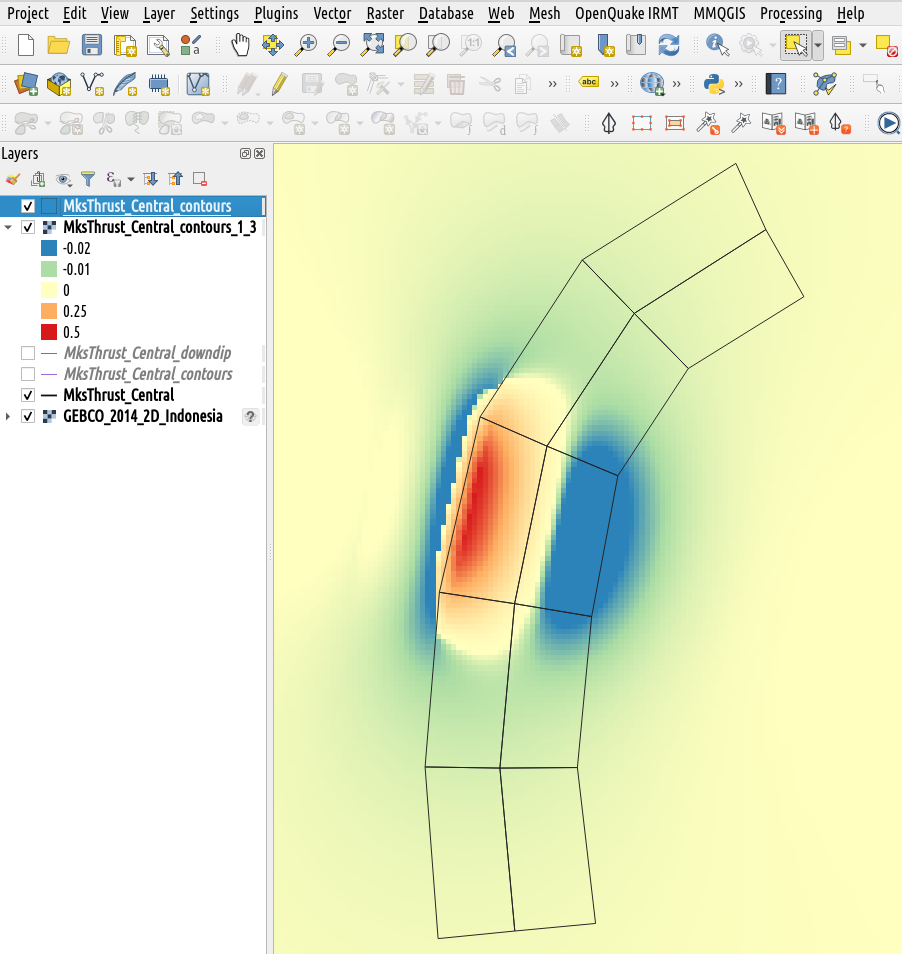
* Create *fault\_traces* directory inside *MakassarStraits*
* Inside *fault\_traces* you need MksThrust\_Central.shp as follows
* 
* The MksThrust\_Central.shp at least should have attribute table such as on the right figure.
* Edit 01\_run\_convert\_traces\_to\_contours.R; important variables:
  + sources\_traces = c(“../fault\_traces/MksThrust\_Central.shp”) << location of the fault\_traces shp
  + output\_dir = “../contours”
* Run the script from terminal:
  + $Rscript 01\_run\_convert\_traces\_to\_contours.R <enter>
  + If there is no error, you will have a new directory: *contours* inside *MakssarStraits,* and MksThrust\_Central\_contours.shp
* Open the contours.shp on QGIS. You will have a slightly dodgy contours at the first contour (shown below). You may edit it manually.
  + 
* You may edit the contours manually to example below. The contour.shp needs attribute table of “level” in string format. **Alternatively**, if you have contour.shp from other sources (e.g. SLAB2.0), you can directly use it and adapt the format as needed.
  + 

1. Create down-dip lines to approximately create the subfaults

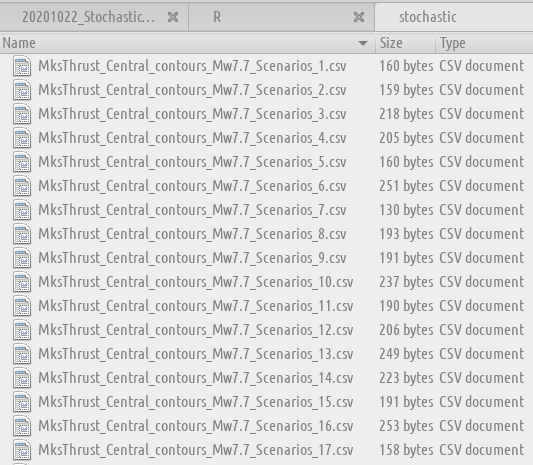
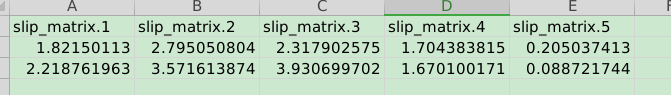
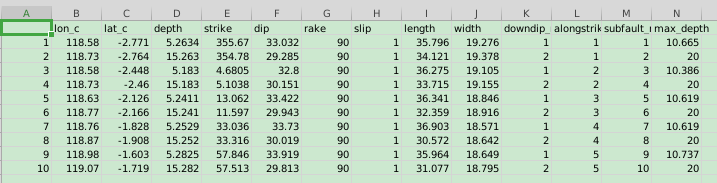
* Edit 02\_make\_initial\_downdip\_lines.R:
  + source\_shapefile = “../contours/MksThrust\_Central\_contours.shp”
  + out\_shapefile = “../downdip\_lines/MksThrust\_central\_downdip.shp”
  + desired\_unit\_source\_length = 40 << to set up the subfault length (in km) along the strike
  + 
* Run the script from terminal:
  + Create *downdip\_lines* directory inside *MakassarStraits*
  + $Rscript 02\_make\_initial\_downdip\_lines.R <enter>
  + If no error, you will have MksThrust\_Central\_downdip.shp inside *downdip\_lines*
* Open MksThrust\_Central\_downdip.shp and manually edit as needed
  + 
  + If you don’t like the subfaults, you can:
    - repeat step 1 and 2
    - I recommend to manually edit the contours.shp and regenerate the downdip\_lines.shp

**Generate unit sources displacement**

In this step, you will generate earthquake deformation from each subfaults according to the Okada (1985) formula.

* Create *outputs* directory
* Edit 03\_produce\_unit\_sources.R:
  + all\_sourcezone\_shapefiles = Sys.glob('../contours/\*.shp')
  + all\_sourcezone\_downdip\_shapefiles = Sys.glob('../downdip\_lines/\*.shp')
  + desired\_subfault\_length = 40
  + desired\_subfault\_width = 20
  + sourcezone\_rake = rep(90,...) << change 90 to -90 if you need a normal mechanism, or as needed
  + tsunami\_source\_cellsize = 0.01 #in degree
  + output\_base\_dir = “../outputs/
  + Other variables can read from the comments or study all the scripts :)
* Run the script from terminal:
  + $Rscript 03\_produce\_unit\_sources.R 1 << change 1 into number of faults if there are multi-sources.
* If no error, you will have new directories and all\_discretized\_sources.RDS inside *outputs*:
  + *Unit\_source\_data/MksThrust\_Central\_contours/*
    - Inside this, you will have a bunch of .tif and .RDS files.
    - The .tif file is earthquake displacement for each subfault.
    - In this case, you will have 10 subfaults: 5x along strike, and 2x along dip
    - MksThrust\_Central\_contours\_<row number along dip>\_<column number along strike>.tif
    - 
    - 
  + *unit\_source\_grid*
    - Polygon of the subfaults
  + all\_discretized\_sources.RDS
* In the final run, you may need to increase the tsunami\_source\_cellsize and change the desired\_subfault\_length and desired\_subfault\_width.

**Generate stochastic scenarios**

* Create *stochastic* directory
* Edit 04a\_combine\_tsunami\_sources.R. This is a simplification version from the original example to be able export the slip distribution.
  + unit\_source\_dirname = “../outputs/Unit\_source\_data” <.tif files location>
  + sourcename = “MksThrust\_Central\_contours”
  + desired\_Mw = 7.7 <random as needed>
  + target\_location = c(lon,lat) <you can fix the peak slip location or put it randomly>
  + number\_of\_sffm = 100 <number of scenarios you want to generate>
  + You can edit between line 64 and 76. There are several options for the *relation* (earthquake scaling)
* Run the script from terminal:
  + $Rscript 04a\_stochastic\_tsunami\_sources.R
* If no error, you will have a bunch of .csv files inside *stochastic* directory.
  + Slip distribution:
    - 
    - 
      * It shows slip (in metre) for #row and #column (A,B,etc.) of the subfault
  + subfaults\_MksThrust\_Central\_contours.csv
    - Subfaults configuration:
    - 
* Every time you do this step, you will get different results even using the same parameters (or you didn’t change the script)

**Additional**

Depends on your needs and the final goal, after this you can do whatever you want:

1. You can convert all the .tif files into a .grd file as needed for the JAGURS then generate tsunami Greens Function. Then use the stochastic scenarios to analyse off-shore tsunami heights. Even, you can continue doing a PTHA.
2. Alternatively, by combining the .tif files and the slip distribution from each stochastic scenarios, you can generate a total earthquake displacement (e.g. in grd file) then use it as initial condition for your tsunami modelling.

Here how I do the b) option on Python. It’s a “dirty” script.

1. Create *ori\_displacement\_grid* inside *outputs* directory
2. Create *scenarios\_grid* inside *stochastic*
3. Edit combine\_unitsources\_and\_slip.py
4. Run combine\_unitsources\_and\_slip.py
5. You will have grd files inside *scenarios\_grid*, then continue as needed.
6. 