SHERIFS

***S**eismic **H**azard and **E**arthquake **R**ates **I**n **F**ault **S**ystems*

Thomas Chartier, Oona Scotti and Hélène Lyon-Caen

Version 1.1

The SHERIFS program is a code developed in the framework of the PhD thesis of Thomas Chartier under the supervision of Oona Scotti (IRSN) and Hélène Lyon-Caen (ENS).

Seismic Hazard and Earthquake Rates In Fault Systems (SHERIFS) is a computer code written in python that allows computing earthquake rates on faults given a geometry of a fault system and of the background, a list of either single or multi-fault earthquake rupture scenarios (FtF) and specified rules to set the moment rate target for the fault system. The underlying iterative approach used in the code is to first estimate the moment rate available for each fault and then to apply a set of rules that allows slip rate increments of each fault to be consumed in rupture scenarios allowed in the model depending on the picked magnitude until their slip-rate budget is exhausted.

Previous versions

The first version of the code was distributed to the participants of the SHERIFS training (IRSN, Paris, France in December 2017). The present Version 1.1 of SHERIFS is published under a GNU Affero Global Public license together with a publication explaining the main features of SHERIFS (Chartier et al, submitted).

The initial version of the code was used to model the fault system of the West Corinth rift published in NHESS special issue (Chartier et al 2017). The code has evolved since. The following are the main aspects that have been implemented after the 2017 publication:

- The code now allows considering background seismicity. The user needs to define a zone surrounding the fault system.

- There is a new option in the code that allows correlating the sampling of the slip-rate on neighboring faults. In this way the slip rate budget of two faults that break together in many rupture scenarios will be sampled in the same way. If one of these faults is sampled in the upper part of the slip-rate distribution, it is admissible that the neighboring fault that breaks very often with this fault is similarly in its upper part of the distribution. This option can be turned on or off.
- The code allows to define magnitude-frequency distribution MFD that deviate from the classical Gutenberg-Richter assumption (e.g. characteristic distribution, Youngs and Coppersmith 1985)
- A modified version of the Youngs and Coppersmith equation has been implemented.

New in Version 1.1 (since the 2019 article)::

- More text files are generated during the Visualization to allow the users to generate more easily their own figures.
- For advanced users, in Smpling_analysis.py, comparison between the model and the data are automatically generated. See code for more details.

Installing SHERIFS Version 1.1

The current version of SHERIFS needs to be run using python 3.6.

We suggest installing **Anaconda** that includes a lot of the dependencies required by SHERIFS. Installing Anaconda also installs **spyder** which is the easiest way to run SHERIFS. You will also need to install *basemap*, the library for plotting maps. Documentation is available online but if you are using anaconda you can run the following line in a terminal (warning! About 200Mo to download):

conda install -c conda-forge basemap

Then run this line to have high resolution coast lines:

conda install -c conda-forge basemap-data-hires

If basemap is still not loading, you might have to close and reopen your python console or spyder.

Before running SHERIFS, you should run **test_SHERIFS.py**. This code ensures that you have all the python libraries necessary for running SHERIFS. If you have everything necessary for running SHERIFS, running test_SHERIFS.py will display a window confirming that all python libraries are correctly installed. The window might appear in the back of other opened windows, so it is worth looking around. If this is the case, the windows opened by SHERIFS might also appear in the back.

If you have a python error saying a library is missing, please install the library using conda install or pip, if you are not using anaconda (documentation on installing libraries is available online with a quick google search – python install *name_of_the_library* -).

Tips for non-frequent python users:

- if the code is crashing, it will display exactly where it stopped, in spyder, you can click on the line number of the SHERIFS code where it crashed. If you use print(*name_of_variable*) the line before it crashed, you can have an idea of where the problem is coming from. Most crashes are due to problems in the input files (format not respected, wrong name of a fault…). But if it is a python issue, most of the big issues have already been encountered and solved by other people online. A copy and paste of the error in google will likely lead you to the answer to your problem.

- for any questions or bugs you cannot solve, please use the google group : https://groups.google.com/forum/#!forum/sherifs . If not already asked and answered, your question will be answered shortly.

Running SHERIFS Version 1.1

The SHERIFS code allows end-users to build the fault model thanks to an interactive user-friendly interface. The files structure, the logic tree structure and the input files can be easily modified for a re-run if different parameters need to be tested.

The code is written so as to build Openquake-compatible input files (OpenQuake V2.6). The user of SHERIFS should consult Openquake for further details about the hazard computation.

Flow chart

1_SHERIFS.py

2_ Visualization.py

3_Weighting.py

The required input files and formats are listed below.

Input files formats

Faults_geometry.txt

This file contains the trace of the faults for the simple faults and the position of each point for complex faults.

The first line contains the column labels.

__list of the column:__

model name = name of the model the fault belongs to

fault name = name of the fault (one fault name should not contain the name of another fault)

longitude and latitude = The points of the faults are listed in rows, they need to be ordered or the fault will have loops.

type ('sf' or depth of the point) For describing a simple fault geometry (see OpenQuake definitions) input 'sf'; for a complex fault geometry, the user inputs the depth of the point - the complex fault geometry is described by at least by two edges (list of points) of uniform depth (top and bottom). Additional edges of intermediate depth can be added for more detailed geometries.

model_name	fault_name	longitude	latitude	type
Example_Model	F1	21.8461481	38.32598913	sf
Example_Model	F1	21.84745563	38.32597968	sf
Example_Model	F1	21.8494889	38.32590774	sf
Example_Model	F1	21.85130491	38.32589457	sf
Example_Model	F1	21.85341081	38.32582203	sf
Example_Model	F1	21.85508153	38.32580986	sf
Example_Model	F1	21.85711547	38.32579501	sf
Example_Model	F1	21.8590754	38.32566621	sf
Example_Model	F1	21.86096404	38.32565235	sf
Example_Model	F1	21.86372436	38.32563205	sf
Example_Model	F1	21.86495924	38.32562295	sf
Example_Model	F1	21.86684855	38.32566622	sf
Example_Model	F1	21.86815744	38.32577099	sf
Example_Model	F1	21.86997412	38.32581475	sf
Example_Model	F1	21.87135565	38.32591895	sf
Example_Model	F1	21.87288247	38.32602204	sf
Example_Model	F1	21.87433665	38.32612566	sf
Example_Model	F1	21.87600737	38.32611319	sf
Example_Model	F1	21.87788911	38.32552684	sf
Example_Model	F1	21.87883066	38.32529088	sf
Example_Model	F1	21.88042596	38.32505	sf
Example_Model	F1	21.88238235	38.3246347	sf
Example_Model	F1	21.88521178	38.32432724	sf
Example_Model	F1	21.8865919	38.32431681	sf
Example_Model	F1	21.88855313	38.32430196	sf
Example_Model	F1	21.89000589	38.32429094	sf
Example_Model	F1	21.89139093	38.32468101	sf
Example_Model	F1	21.89306652	38.32506885	sf
Example_Model	F1	21.89423507	38.32557498	sf

Faults_properties.txt

This file contains the geometry and kinematics of the faults. All the parameters required by Openquake are requested.

The first line contains the column labels. Each row is a fault in a model.

__list of the column:__

model name = name of the model the fault belongs to **fault name** = name of the fault (one fault name should not contain the name of another fault)

dip = dip of the fault

oriented = orientation of the dip (important for the "right hand rule" of OpenQuake)

mechanism = fault mechanism (N, S, R or value of the rake)

upper_sismo_depth = upper limit of the fault, (km) following the OpenQuake definition

lower_sismo_depth = lower limit of the fault, (km) following the OpenQuake definition

slip_rate_min = lower limit of the slip-rate distribution **slip_rate_moy** = mean value of the distribution **slip_rate_max** = higher value of the distribution

slip-rate is picked in a uniform distribution. (sample 1 is always the mean value)

Domain = seismotectonic model used as a key in OpenQuake to attribute to each seismogenic source the correct GMPE.

Shear modulus = Shear modulus applied to this fault (in GPa, typical value 30 GPa)

model_name	fault_name	dip	oriented	mechanism
Example_Model	F1	60	N	N
Example_Model	F2	55	N	N
Example_Model	F3	60	N	N
Example_Model	F4	60	N	N
Example_Model	F5	60	N	N

upper_sismo_depth	lower_sismo_depth
0	6
0	7
0	7
0	7
0	7

slip_rate_min	slip_rate_moy	slip_rate_max
4.8	5.	5.2
3.	3.2	3.4
3.8	4	4.2
0.7	0.9	1.1
3.3	3.5	3.7

Domain	shear_modulus
Active_Shallow_Crust	30
Active_Shallow_Crust	30
Active_Shallow_Crust	30
Active_Shallow_Crust	30
Active_Shallow_Crust	30

Model_used lon lat Example_Model 21.77 38.450 Example_Model 22.018 38.400 Example_Model 22.330 38.300 Example_Model 22.330 38.163 Example_Model 22.018 38.210 Example_Model 21.77 38.311	**Background_properties.txt** This file contains the description of the source parameters for the background region used in OpenQuake. The details of this parameters are explained in the OpenQuake manual. New lines can be added if more options need to be in the model.
Example_Model upperSeismoDepth 0. Example_Model lowerSeismoDepth 8. Example_Model ruptAspectRatio 1. Example_Model nodalPlane 0.7 270. 60. -90. Example_Model nodalPlane 0.3 90. 60. -90. Example_Model hypoDepth 0.2 2. Example_Model hypoDepth 0.3 4. Example_Model hypoDepth 0.3 6. Example_Model hypoDepth 0.2 8.	**Background_geometry.txt** This file contains the description of the geometry of the background zone.

Step by step building of the hazard model

Edit 1-SHERIFS.py file

- **Find a name for your hazard calculation and link to the proper input files you want to use:** *here the Run is called Example and the files concerning the Example are targeted.*

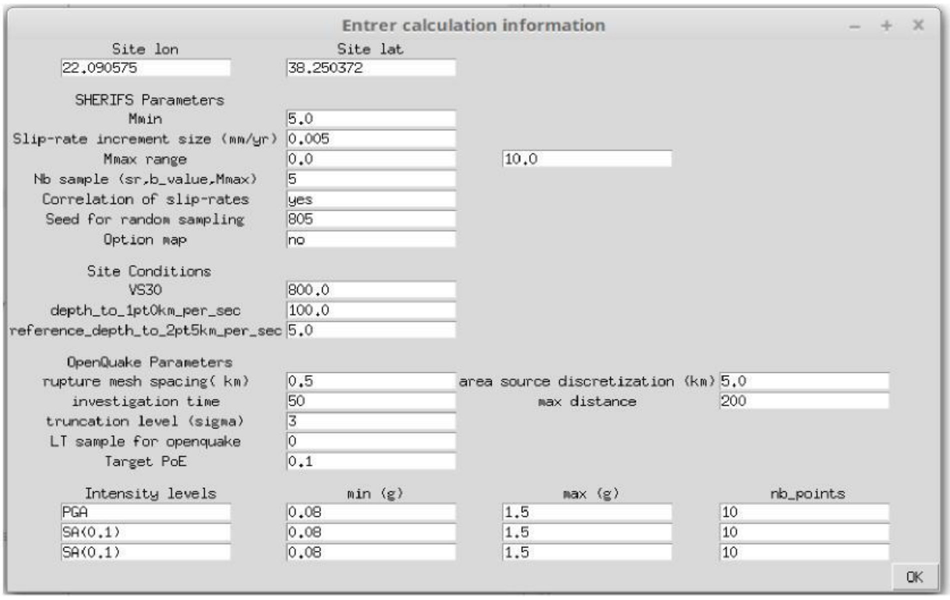
1_SHERIFS.py | 2_Hazard_model_visualisation.py | 3_Weighting_the_Logic_Tree.py

```python
# -*- coding: utf-8 -*-

"""
SHERIFS
Seismic Hazard and Earthquake Rates In Fault Systems

Version 1.0

The Seismic Hazard and Earthquake Rates In Fault Systems (SHERIFS) program, is an open source collection of
tools for calculating the rates of earthquakes on each fault of a fault system
allowing complex Fault to Fault ruptures following the methodology presented
in Chartier et al 2017. It includes a set of tools for checking and visualizing input and outputs
and producing JPEG illustrations.It is released under the GNU Lesser General Public License.
The SHERIFS program is a code developed in the framework of the PhD thesis of Thomas Chartier
under the supervision of Oona Scotti (IRSN) and Hélène Lyon-Caen (ENS).


@author: Thomas Chartier
contact : chartier@geologie.ens.fr
"""

import time
import os
import sys
debut = time.time()

path_actuel=os.path.dirname(os.path.abspath(__file__))
path_dossier = path_actuel + '/lib'
sys.path.append(path_dossier)
from GMPE_Logic_Tree_Creator import GMPE_Logic_Tree_Creator
from Sources_Logic_Tree_Creator import Sources_Logic_Tree_Creator
from OQ_job_Creator import OQ_job_Creator

print ('\nRunning SHERIFS version 1.0\n')
'''############################'''
'''       Input files        '''
'''############################'''

Run_Name = 'Example'
File_geom = 'data/Example/Faults_geometry.txt'
File_prop = 'data/Example/Faults_properties.txt'
File_bg = 'data/Example/Background_geometry.txt'
file_prop_bg = 'data/Example/Background_properties.txt'

# if rerunning the calcuation do you want to overwrite the existing source model files
overwrite_files = True

use_host_model = False
host_model_file = 'data/Example/host_source_model.xml'

if not os.path.exists(str(Run_Name)):
    os.makedirs(str(Run_Name))
if not os.path.exists(str(Run_Name) + '/results'):
     os.makedirs(str(Run_Name) + '/results')

Domain_in_model = []

OQ_job_Creator = OQ_job_Creator(Run_Name) # ask the info about the run and create the job.ini file
```

Run 1-SHERIFS.py : windows asking for the information concerning the run and the logic tree will appear one after the other. Hereafter is an explanation of what are the requested information.

These windows will help building the file structure, the logic tree structure and the input files for OpenQuake. All these files can be easily modified for a re-run with a few modifications afterwards. Going over the whole graphical interface is not necessary after building the basic structure of the model and its logic tree. (see below for more information)

Fill in the first window with the general information of the calculation



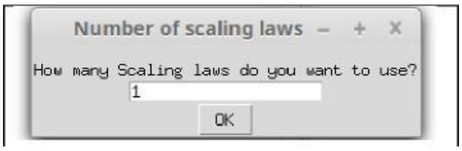
Details of the information:

Coordinate of the calculation site (won't be used if "option map" is activated). Can easily be changed later on in the OpenQuake input file job.ini.
The SHERIFS parameters **Mmin** : Minimal magnitude for the PSHA calculation **Slip-rate increment size (mm/yr):** A small size of slip-rate increment will lead to more precise use of the slip-rate budget of each fault but will lead to longer computing time for creating the source models **Mmax range**: imposes the Mmax of the fault system to be in this range. Nb sample: Number of sampling in the distributions of slip-rate of the faults, b_value of the model and Mmax (uncertainty within a scaling law) **Correlation of slip-rates**: Correlate the sampling of faults that can break together in several multi-fault rupture scenarios. **Seed for random sampling**: Random sampling seed used in python and OQ. **Option map**: Do you want to do a hazard map? If yes, another window will open where the user needs to specify the corners of the map. If no, the calculation will be done for a single site with the coordinates provided in the first entry.
VS30 in m/s **Z100** in m Z2.5 in km

Openquake parameters
Rupture mesh spacing (km): in km. Refers to the level of detail of each modeled earthquake rupture.
Investigation time: Investigation time for the hazard calculation.
Truncation level (sigma): Truncation level of the GMPEs in number of sigmas.
LT sample for OpenQuake: if 0, all Branches of the logic tree are explored. Otherwise, this parameter is the number of Monte-Carlo exploration of the logic tree.
Target PoE: Target Probability of exceedance for UHS calculation
Area source discretization(km): in km. Distance between each modelled hypocenter.
Max distance(km): Maximum distance to incorporate seismic sources.

Intensity levels used for the PSHA calculation.
Default calculates PGA and PSA(0.1s). if two lines are the same, only one is calculated. (in this example only PGA and PSA(0.1s) will be calculated.

Choose the number of scaling laws you want to use



if you want to use different options within the same scaling law, this is considered as a different scaling law by the program and you need to ask for additional scaling laws.

For example, WC94 using Area and WC94 using the length as a metric counts as two different scaling laws.

Describe the scaling laws

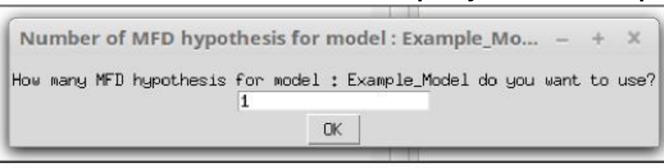
You need to describe which law you want to use (only Wells and Coppersmith 94, Leonard 2010, Thingbaijam 2017 and Bakun 2004 available so far).

You need to choose if you want to use the mechanic specific option (default = yes), this option is only usable with WC94. This option tells you which database is used : earthquakes with all focal mechanisms together or distinguished according to their mechanism as your source (Normal, Strike-slip or Reverse). Finally , you choose if you want to use the law based on the area or the length of the fault (area is default).

list the seismotectonic models you want to use

Each model is a vision of the geometry of the faults in the fault network or of their activity as well at the geometry of the background.

Choose the number of MFD shapes you want to explore.



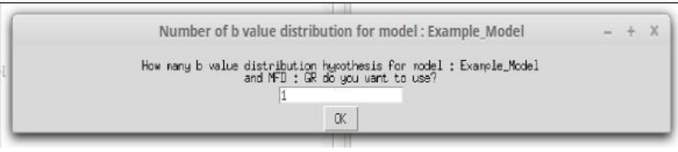
- **Choose the name of the MFD shape to explore.**

Only **GR**, **YC** and **YC_modified** distributions are coded so far.

For the case of the YC_modified distribution, the Mf parameter has to be modified in EQ_on_faults.py.

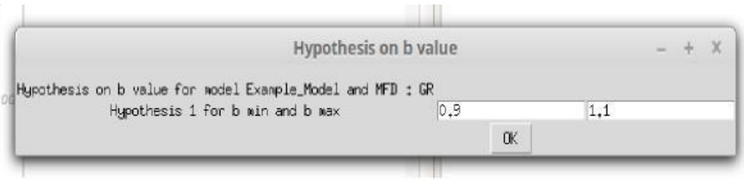
For each MFD :

- **Choose the number of b value distributions you wish to explore for each specific model and a specific MFD distribution.**



Each hypothesis is not one single value of b, but a distribution.

Set the limits of the b value distributions for the MFD shape

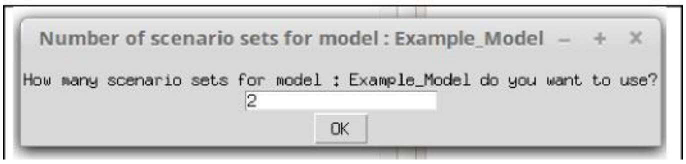


These values are the lower and upper bounds of a centered triangular distribution that will be used to randomly pick the b value of the target MFD for the fault network.

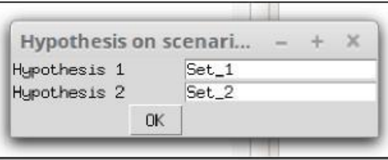
Set the number of scenario sets to explore for each specific model.

Each scenario set hypothesis is a list of all possible multi-fault (FtF) ruptures allowed in the model.

Default value is 1 -

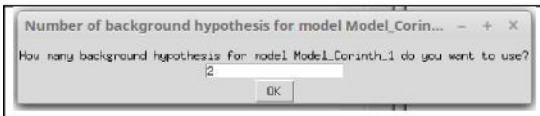


Name each hypothesis

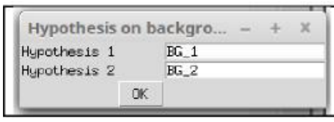


The name of the scenarios should not contain spaces.

Set the number of background hypothesis you wish to use



Name the different background hypothesis

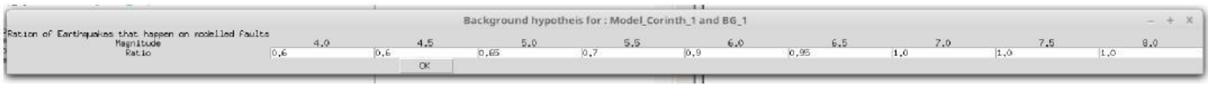


Each hypothesis will be a different opinion on how the seismicity will be shared between modeled faults and background seismicity.

<u>**Same steps are repeated for each model explored in**</u>

<u>**the logic tree.**</u> **For each background hypothesis:**

- **Define the ratio of how much of the seismicity is on the faults and how much is in the background**

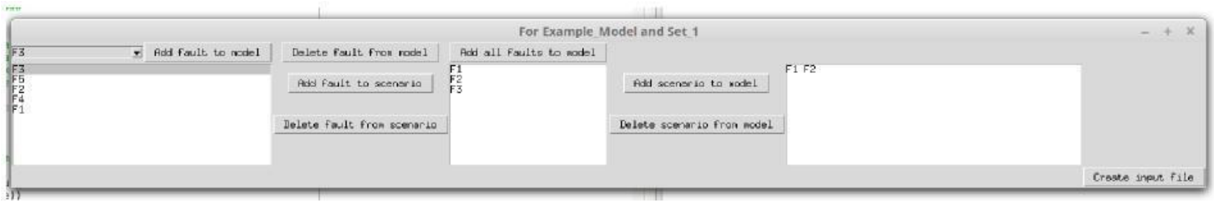


Under each magnitude, define the ratio of earthquakes of this magnitude that are expected to happen on modeled faults. In the example, 60% of M4 are believed to be on modeled faults, 90% of the M6 are on the modeled faults and all M7 and more are believed to be on the modeled faults.

The python code will use this value and linearly interpolate the value of the magnitudes in between.

This approach allows an easy exploration of the epistemic uncertainty related to the background seismicity that is strongly linked to expert judgment. It is strongly encouraged to use statistical and analytical techniques in order to set these parameters.

- **Select the faults and the multi-fault ruptures scenarios possible in the model**



- *Select each fault to include in the model (you can directly select all faults clicking on "add all faults to model"*
- *Construct each possible scenario by selecting a fault in the list and clicking on " add fault to scenario"*
- *If you made a mistake, you can delete the fault from the scenario by selecting it and clicking on "delete fault from scenario"*
- *Once your scenario is ready, you can add it to the scenario list by clicking "add scenario to model".*

- *If you made a mistake in a scenario, you can delete it by selecting it and clicking on "delete scenario from model"*

If a scenario is already on the list, an error message will appear. This message only appears if the faults in the scenario are selected in the same order. The order for selecting the fault doesn't matter for the calculation but I suggest that faults in a scenario should always be selected by scrolling down along the list.

**If you realize later on that you did a mistake, don't worry, it is possible to rectify it fairly easily later. (see output files part)*

***If your model has a large number of faults and scenarios and you have your own routine to create a list of possible scenario it is possible to use your routine easily (see output files part for more detail).*

*** We suggest starting with a basic exploration of a few scenarios rather than trying to do the whole logic tree in the first run. It is always possible and even easier to add and run more branches later on!!*

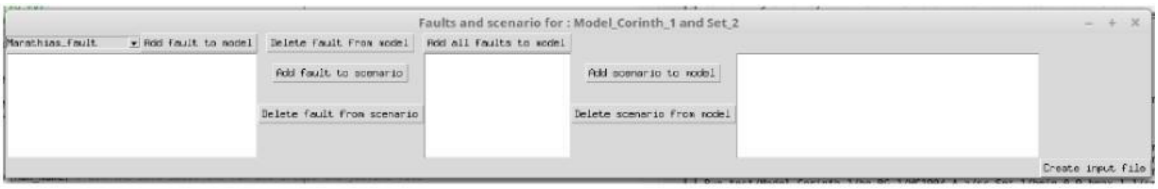
!!!!!!!!!!! Once you are ready, press "create input file" !!!!!!!!!!!!!

... The python code creates OpenQuake input files and many different log files for each branch of the logic tree (including the random sampling of the slip-rates, the Mmax and the b value)….

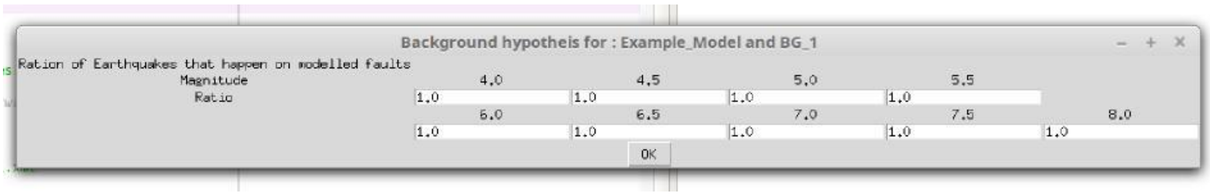
```
Console 3/A
invalid value encountered in true_divide
  shape_mfd_i = (moment_rate_in_bin)/sum(moment_rate_in_bin)
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 32
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 2
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3253.0
1%
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 26
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 3
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3370.0
1%
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 22
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 4
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3154.0
1%
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 32
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 5
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3287.0
1%
25%
50%
- target set -
-target filled-
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 33
```

For each model, the name of the branch and the sample number are written. The number of dsr (slip-rate increment) to spend is displayed. The user can follow the advancement of the calculations since the code displays when 1%,25%,50%,73% and 90% of the slip-rate budget is spent. When the rate of the three largest bins of magnitude is limited, the target is set and the code writes 'target set'. At the end of the calculation for one branch, the code writes the ratio between the target shape and the actual shape. 1.0 is a perfect score. If the ratio is not good enough given the error accepted (indicated by the user in the 1_SHERIFS.py file), the model is ran another time with a smaller dsr.

- **If there is an alternative "scenario set" branch, then the code will ask you to fill in the possible FtF rupture for this specific branch.**



- **If there is an alternative "background hypothesis" branch, then the code will ask you to define the partition of earthquakes on and off the faults.**

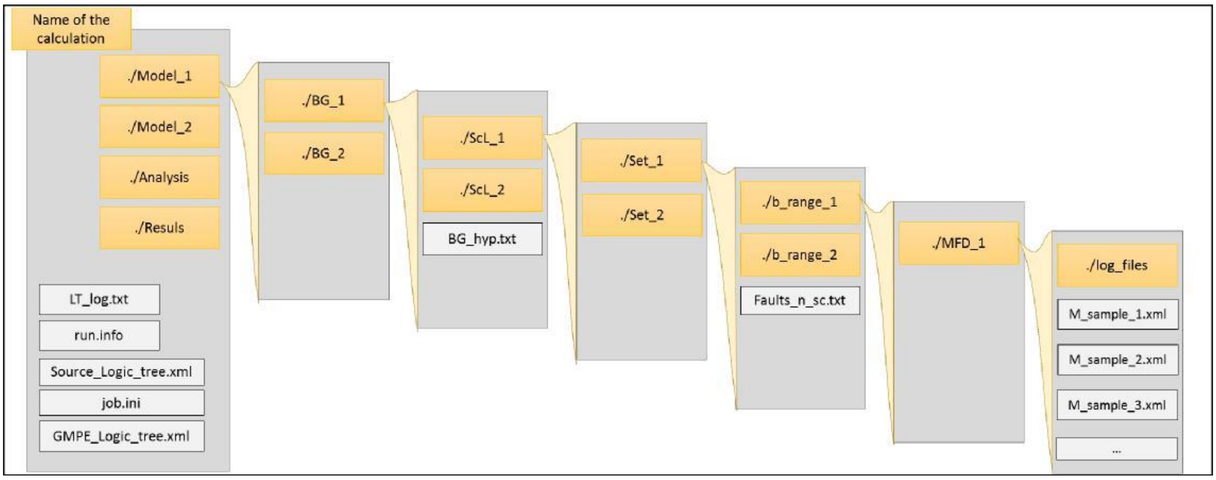


The computation goes on for the different branches until all hypotheses are explored.

- **Once the whole logic tree is explored, a window will open where you need to choose the GMPEs to explore in the PSHA calculation and their respective weights in the calculation. It will create the file GMPE_logic_tree.xml that can easily be modified later before running the hazard calculation with OpenQuake.**

Output files

Folder structure



The whole hazard calculation is in the folder with the name of the calculation.

In this folder, there are three OpenQuake files (Source_logic_tree.xml, Job.ini and GMPE_logic_tree.xml if the option build the GMPE logic tree was selected), two files containing the information set by the user and the folders for each model explored.

Each level of the logic tree has its own level of the folder structure. **<u>Description of the created files</u>:**

LT_log.txt

```
Models

Model_Corinth_1
Model_Corinth_2 Scaling
laws
```

```
WC1994 Area m    le2010
Area m
MFD b value

MFD_GR      bmin_1.0_bmax_1.1     bmin_0.9_bmax_1.0

MFD_YC      bmin_1.1_bmax_1.2     bmin_1.2_bmax_1.3

Background

bg_BG_1
            bg_
BG_2
Scenario set

sc_Set_1 sc_Set_2
```

For advance users: This file can be modified to add hypothesis and then SHERIFS.py can be run again to overwrite the files. Hypotheses are on the same line separated by a tab except for the MFD hypothesis that are in a row with the attached b value hypothesis in line separated by a tab. Be careful when editing by hand and check your results.

All lines must finish with a tab to avoid problems.

run.info

```
Information on run : Corinth_risk

Option map:         no

Vs30   :       800.0
Site Z1000 :        100.0
Site Z2500   :      5.0
nb_LT_samp :        0
rup_mesh :    0.5
source_discr :      5.0
investigation_time :      50
                                 0.1
Probability of exceedance :

trunc_lvl   :       3            :      20
Mmin   :      5.0
Random seed :       805
SR correl   :       True
SR increment size   (mm/yr)   :  0.001
Mmax range :        0     10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
intensity_i   :     PGA    0.01   1.5    10
```

This file contains the information entered in the first window of SHERIFS, see above for more detail. In the same manner as LT_log.txt, it can be modified for running again SHERIFS.py.

bg_ratio.txt

```
0.5
0.6
0.7
0.75
0.8
0.9
```

1.0
1.0
1.0

This file contains the ratio of earthquakes on the faults for magnitude 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5 and 8.0.

This file can also be modified for a rerun. **In such a case, the file must be modified in every single folder.**

faults_n_scenario.txt

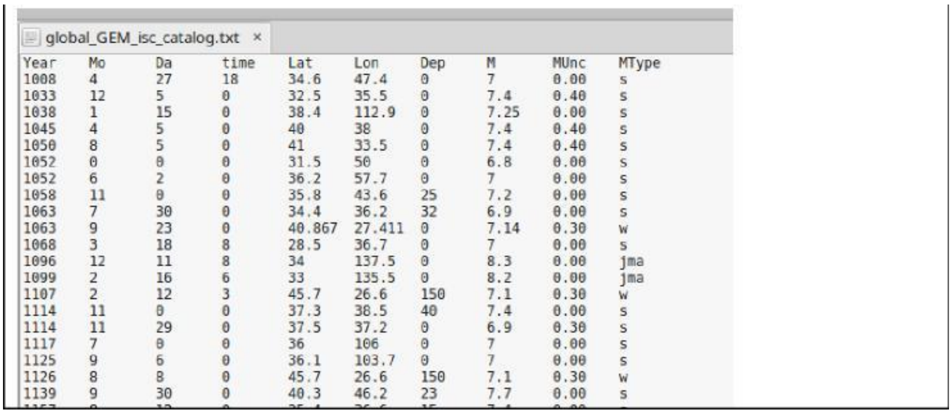
First line (list of the faults in the model)	F1 F2 F3 F5
Next line, rupture scenarios	F1 F2 F1 F2 F3 F1 F2 F3 F5 F2 F3 F2 F3 F5 F3 F5

The first line of this file is the list of faults in the model; the following lines are the possible FtF ruptures in the model. This file can also be modified manually before re-running SHERIFS.py. Filling all the possible FtF rupture of a model with the graphical interface can be a drag, if your model has a lot of possible FtF ruptures, it is much easier to change it afterwards. But as always, be careful when editing manually the files.

Each fault name is separated by a space. No tab and no newline at the end.

Reality check of your hazard models

Choose the earthquake catalog you wish to use and format it as following



global_GEM_isc_catalog.txt

Year	Mo	Da	time	Lat	Lon	Dep	M	MUnc	MType
1008	4	27	18	34.6	47.4	0	7	0.00	s
1033	12	5	0	32.5	35.5	0	7.4	0.40	s
1038	1	15	0	38.4	112.9	0	7.25	0.00	s
1045	4	5	0	40	38	0	7.4	0.40	s
1050	8	5	0	41	33.5	0	7.4	0.40	s
1052	0	0	0	31.5	50	0	6.8	0.00	s
1052	6	2	0	36.2	57.7	0	7	0.00	s
1058	11	0	0	35.8	43.6	25	7.2	0.00	s
1063	7	30	0	34.4	36.2	32	6.9	0.00	s
1063	9	23	0	40.867	27.411	0	7.14	0.30	w
1068	3	18	8	28.5	36.7	0	7	0.00	s
1096	12	11	8	34	137.5	0	8.3	0.00	jma
1099	2	16	6	33	135.5	0	8.2	0.00	jma
1107	2	12	3	45.7	26.6	150	7.1	0.30	w
1114	11	0	0	37.3	38.5	40	7.4	0.00	s
1114	11	29	0	37.5	37.2	0	6.9	0.30	s
1117	7	0	0	36	106	0	7	0.00	s
1125	9	6	0	36.1	103.7	0	7	0.00	s
1126	8	8	0	45.7	26.6	150	7.1	0.30	w
1139	9	30	0	40.3	46.2	23	7.7	0.00	s

The catalog should be in the shape of the example (first line is the name of the column).
If the uncertainty of magnitudes in the catalog is not specified, input a default value.

lines to change or verify in Visualisation.py :

Name of the run

File_bg : file containing the geometry of the background. Same file used for creating the models. The geometry of the background will be used to extract the earthquake catalog for comparisons.

File_fault_geometry : ...

File_fault_data : (optional) file with information of earthquake rate of a specific fault. If information is available on a specific fault (rate of historical or instrumental earthquake, or rate of paleoearthquake located on this fault).

Structure :

Model	fault_name	type	M	sigma_M	rate	sigma_rate
Model _1	Fault_name	pal	6.4	0.4	0.003	0.002
Model _1	Fault_name	cat	6.	0.	0.006	0.001

…

Catalog_file : name of the catalog file you want to use. By default, the SHEEC catalog is used but it can be modified.

Completeness_file : File containing an estimation of the completeness for each magnitude. It is possible to explore several completeness hypotheses.

Structure:

Name_completeness 4.0,4.4 4.5,4.9 5.0,5.4 5.5,5.9 6.0,6.4 6.5,6.9 7.0,7.4 7.5,7.9 8.0,8.4

Weight 1996 1962 1958 1904 1725 1725 1725 1725 1725

Name_completeness 4.0,4.4 4.5,4.9 5.0,5.4 5.5,5.9 6.0,6.4 6.5,6.9 7.0,7.4 7.5,7.9 8.0,8.4

Weight 1996 1962 1958 1904 1725 1725 1725 1725 1725

Sub_area_file : If you want to extract a sub region of your model to compare the model rate to the catalog, define the coordinate of this zone in this file.

Structure:

Model_name Sub_area_name lat,lon lat,lon lat,lon ...

Model_2 ...

(! no empty line, tab at the end of each line!)

Coordinate of the **llcr** (lowerleftcorner)and **urcr** (upperrigthcorner) for setting the rectangle for the maps.

In order to be able to visualize more rapidly different aspects of the model, Booleans can be turned on to activate the visualization of different parts of the model.

Plot_mfd	Plot the MFD for each node of the logic tree.
Plot_mfd_detailed	Plot the MFD for sub selection of the logic tree combining different hypothesis.
Plot_Mmax	Plot the distribution of maximum magnitude in the model.
Plt_as_rep	For each node of the logic tree, give the proportion of aseismic slip in the models.
Plot_rup_freq	Calculate the rupture rate for each fault of the model and for eventual subareas defined by the user.
plot_moment_rate	Plot the moment rate in different branches of the model and compare it to the moment rate in the catalog.
Visual_FtF	Draw map of all the FtF ruptures for each model hypothesis and each scenario set hypothesis.

The created figures and text files are located in the folder /analysis.

./analysis/txt_files/ contains text files containing the MFD of each branch and each source of the logic tree, the partitioning of the slip-rate between the single ruptures and the complex ruptures for each fault source and more...

- list of the files and short description, see code for exact detail of what is being done, any user is of course welcome to generate more files:

branch_cumMFD.txt : cumulative MFD of each branch of the logic tree.

- ◦ Branch_vs_catalog.txt : ration between the rate in the catalog and the modelled rate for each branch
- ◦ faults_MFD.txt : very rough file containing the array for the MFD of each individual source of the each individual branch of the logic tree.
- ◦ In V1.1 : lT_metrics.txt : each line is a branch of the logic tree with the mean slip-rate of the faults, and the scores when comparing to the catalog, the Mmax score, the MNS score and the paleo score. (see sample_analysis.py for more details)
- ◦ mean_parameters_faults.txt : for each model, set and fault, gives the mean slip-rate and the mean Mmax.
- ◦ slip_rate_sampling.txt : for each branch of the model and each fault, gives the slip-rate randomly selected for this sample. Sample 1 is always the mean slip-rate.
- ◦ slip_rep_on_faults_data : for each branch of the logic tree and fault, explains how the slip-rate has been used. The first columns of the files are describing the branch, then the name of the fault the first number is the percentage of slip budget spent on single fault ruptures, the second number is for ruptures involving two faults, then three faults and so on.. the last number is the NMS.
- ◦ slip_rep... : similar but average for whole parts of the logic tree corresponding to the name of the file

./analysis/figures/ contains a list of folders containing the different figures created.

/analyse_branch contains cumulative MFDs for each branch of the logic tree and the ratio of aseismic slip in the model. Exploring the folders and files in this folder will allow selecting some branches of the logic tree. In the folder Model, you will find more detail.

/catalog contains the catalog MFD and maps as well as the number of earthquake of each magnitude in the catalog. If subareas have been used, the rates from the catalog are available in a sub folder.

/compare moment_rate contains figures of box plots of the modeled moment rate, the model moment rate if no NMS was considered and the moment rate calculated from the catalog.

/FtF contains the map visualization of each FtF scenario taken as input in each FtF set hypothesis of the logic tree. This folder also contains the map of the maximum magnitude for each fault in the system, the map of mean slip-rate for each input model, the mean slip-rate that was considered as seismic on each fault and the map of NMS slip ratio on each fault. The name of the file details which branches of the logic tree are taken into account in order to calculate the average value presented for each fault on the map.

/mfd contains the MFD of the whole logic tree.

/Mmax contains statistics on the maximum magnitude in the logic tree, for the whole logic tree and in sub folders, for specific branches of the logic tree. This folder also contains the distribution of lengths and areas of the considered ruptures. If there is a gap in those distribution (some large rupture are considered but no intermediate ones, it can cause a lot of NMS slip in the SHERIFS calculations).

In V1.1 : In the folder named after each model, and the subfolders named after each scenario set, you will find a figure for each fault showing the number of ruptures considered on this fault section able to generate a magnitude larger or equal to a given magnitude. If for a fault, some very large magnitudes are considered, but no intermediate ones, it is likely that there is a problem in the SHERIFS calculation and that more FtF rupture need to be added in the intermediate range.

/rupture_rate_on_each_fault contains the MFD of each individual fault of fault section as well as the mfd of the sub_area if some have been defined. The plotted rates are the rates of each rupture including the fault section. Concerning the sub area, the rate of each FtF completely contained in the subarea is included and the rate of FtF rupture only partially rupturing in the subarea are multiplied by the ratio of number of sections of the FtF scenario in the subarea over the total number of sections of the FtF scenario.

In V1.1:This folder also contains txt file presenting the Magnitude of Median moment rate Mmmr for each fault for several branches of the logic tree. The Mmmr is the magnitude for which half of the moment rate of this fault is released in larger magnitude and half of the moment rate of this fault is released in smaller magnitudes. It's a good indicator of how much characteristic or GR the MFD of the fault is.

/sampling_analysis contains figures illustrating the fit of each branch to the catalog. Advanced SHERIFS users with python coding skills can use this tool to test hypotheses of the logic tree and sampling of parameters.

In V1.1: The model_performance.png figure exposes how well a model performs against data and if the MSN on some faults of the model is acceptable. (More details to be added once this feature is complete)

Weight the logic tree

Once you have looked at your model, performed consistency checks against the data, you are ready to set the weight for each branch of the logic tree.

The sum of the branches will be calculated and displayed. If this sum is not exactly one, the user is required to change manually the weights in the openquake logic tree input file.

Once the logic Tree is weighted, you can run the Openquake Engine.

(Running OpenQuake : see github.com/gem/oq-engine

command line : >> oq engine --run job.ini

>> oq engine --exports-outputs #run ./results

>> oq export hcurves/all)

Références :

Chartier, T., Scotti, O., Lyon-Caen, H. and Boiselet, A., 2017. Methodology for earthquake rupture rate estimates of fault networks: example for the western Corinth rift, Greece. Natural Hazards and Earth System Sciences, 17(10), pp.1857-1869.

Chartier, T., Scotti, O.,and Lyon-Caen, H., *in press* SRl. SHERIFS – Open-Source Code for Computing Earthquake Rates in Fault Systems and Constructing Hazard Models

Gutenberg, B., and C. F. Richter 1944 "Frequency of earthquakes in California." Bulletin of the Seismological Society of America 34.4 : 185-188.

Youngs, R. R., and K. J. Coppersmith. (1985)"Implications of fault slip rates and earthquake recurrence models to probabilistic seismic hazard estimates." Bulletin of the Seismological society of America 75.4 : 939-964.