User Manual SHERIFS

Thomas Chartier, Oona Scotti, Hélène Lyon-Caen Version 1.3

Last Updated January 2021

Contents

1 Introduction

The SHERIFS program is a code first developed in the framework of the PhD thesis of Thomas Chartier under the supervision of Oona Scotti (IRSN) and Hélène lyon-Caen (ENS). Since version 1.3, developments of the code have been realized at GEM. 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 potential earthquake ruptures and specified rules to set the moment rate target for the fault system. The underlying 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 allow the slip rate of each fault to be consumed in either single or multi-fault rupture scenarios (FtF) allowed in the model depending on the picked magnitude until their slip-rate budget is exhausted.

Updates to this manual are available at: https://github.com/tomchartier/SHERIFS/

2 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 [1] (Chartier et al 2017). The code has evolved since. The following are the main aspects that have been implemented after the 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 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 Sampling_analysis.py, comparison between the model and the data are automatically generated. See code for more details.

New in version 1.3 (2020 developments):

- Greatly improved computational speeds by saving intermediate results that are shared by different branches of the logic tree.
- SHERIFS now supports Geojson as input which allows for compatibility with the Global fault database and the output results coming from Oiler (GEM products).
- An automatic calculation of the possible rupture in a fault system according to the distance between faults.
- A general clean up and reorganisation of the libraries.
- The graphical user interface is no longer supported, running using Spyder is still possible but the command line is now preferred

3 Installing SHERIFS Version 1.3

for now, we suggest to download from Github. Basemap is not needed anymore a required library for the figures. In the future, SHERIFS will start using GMT instead. In the meanwhile, SHERIFS relies on the use of geojson files and QGIS.

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

- if when running, python doesn't find a module, you can install it byt running in the command line: pip install name_of_the_command or conda install name_of_the_command (if you are a anaconda user)

- 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.

4 Running SHERIFS Version 1.3

The SHERIFS code allows end-users to build the fault model with complex multifault ruptures. 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 to build Openquake-compatible input files (OpenQuake V3.12). The user of SHERIFS should consult Openquake for further details about the hazard computation.

Flow chart

(optional : 0_build_ruptures.py)

1_SHERIFS.py

2_Visualization.py (under revision)

3_Weighting.py (under revision)

To run SHERIFS from the command line

Make sure you are in the folder containing SHERIFS.

Run :

python 0_build_ruptures.py path_to_sherifs_in.toml

python 1_SHERIFS.py path_to_sherifs_in.toml

4.1 The SHERIFS input parameter file

4.2 Input file formats - using geojson files

4.2.1 Fault info : in a geojson

The geometry is described in a LineString describing the trace at the surface. SHERIFS will describe a kite fault surface between the min and max seismogenic depth.

Description of the parameters:

si = name of the section, can be a string of an integer

model = 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) **sr** = slip-rate in mm/yr in the direction of slip **e_sr** = error on the slip-rate in mm/yr in the direction of slip

dip = dip of the fault

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

rake = value of the rake for this section

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

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

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)

Two parameters nb_rup and max_rup_length are also included in the file by the reprocessing tools but optional and are not directly used within SHERIFS.

4.2.2 Background geojson file

A geojson with a MultiPolygon describes the background geometry. Several backgrounds can be included in this file, the ones chosen for a given model is identified by the property "model" of the feature. It is important that the model property correspond with the one used in the logic tree file and the fault property file.

4.2.3 Mmax geojson file - optional

This file is used in the precomputational phase to define the maximum magnitude or length that rupture can reach in different regions described by polygons.

4.2.4 Shear modulus correction geojson file - optional

This file is used in the precomputational phase to modify the shear modulus of a large number of faults in specific regions.

4.3 Input file formats - using txt files

This is the former input format for SHERIFS. It is still supported, but we suggest switching to geojson, it is far easier to use.

4.3.1 Faults_geometry.txt

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

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.

4.3.2 Faults_properties.txt

model_name	fault_name	dip	oriented	mechanism	upper_sismo_depth	lower_sismo_depth	slip_rate_min	slip_rate_moy	slip_rate_max	Domain	shear_modulus
Example_Model	F1	60	N	N	0	6	4.8	5.	5.2	Active_Shallow_Crust	30
Example_Model	F2	55	N	N	0	7	3.	3.2	3.4	Active_Shallow_Crust	30
Example_Model	F3	60	N	N	0	7	3.8	4	4.2	Active_Shallow_Crust	30
Example_Model	F4	60	N	N	0	7	0.7	0.9	1.1	Active_Shallow_Crust	30
Example_Model	F5	60	N	N	0	7	3.3	3.5	3.7	Active_Shallow_Crust	30

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.

Description of the columns:

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)

4.3.3 Background_geometry.txt

```
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
```

This file contains the description of the geometry of the background zone.

4.4 Input file formats - common files when using txt or geojson

4.4.1 Background_properties.txt

```
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.
```

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.

4.4.2 LT.toml

This toml files contains a list of hypotheses for each node of the logic tree.

Models = alternative fault models that should be explored. this is useful for exploring uncertainties in the geometries of the faults, for example.

MFD_shape = Hypotheses on the target shape of the MFD. Each hypothesis contains the name of the MFD shape and the needed parameters to constrain it.

Background = List of the background hypotheses to explore. The name needs to correspond to the name in the bg_seismicity.txt file.

scenario_set = List of the scenario sets to explore. This list needs to correspond to the scenarii defined in the rupture.txt files.

Scaling_Laws = List of the scaling laws to use with associated parameters such as the rake dependency and the dimension to use (Area or Length).

4.4.3 sherifs.in

Run_Name : name of the run (needs to be the same as run.info)

File_geom : path to file of fault geometry

File_prop : path to file of fault property

(if a geojson file is used they are the same)

File_bg : path to file of background geometry

file_prop_bg : path to file of background property

host_model_file : xml of the host source file (optional)

bgf : path to folder, or xml file with the smoothed seismicity background

overwrite_files : if True, every file is overwritten

use_host_model : tag True when using host model

fit_quality : default is 5 meaning the resulting mfd is withing 5% of the target shape

Please see the example file for the full list of parameters

4.4.4 bg_seismicity.txt

bg ”name of the bg branch”

0.2 0.2 0.3 0.3 0.4 0.6 0.75 0.95 0.999

add new lines for new hypotheses.

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.

4.5 Faults_n_scenario.txt

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

4.6 Running SHERIFS

In order to run SHERIFS with the command line, move to the SHERIFS repo and run the following line:

python 1_SHERIFS.py path_to_sherifs_in.txt

... 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.

5 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.

(For running OpenQuake : see `github.com/gem/oq-engine`)

OQ command lines :

oq engine –run job.ini

oq engine –exports-outputs #run ./results

oq export hcurves/all

References

[1] T. Chartier, O. Scotti, H. Lyon-Caen, and A. Boiselet. Methodology for earthquake rupture rate estimates of fault networks: example for the western corinth rift, greece. *Natural Hazards and Earth System Sciences*, 17(10):1857–1869, 2017.