



OPENQUAKE

calculate share explore



OPENQUAKE ENGINE USER INSTRUCTION MANUAL

Version 1.8.0

Hands-on-instructions on the different types of calculations you can carry out with the OpenQuake Engine software



USER
INSTRUCTIONS



HAZARD
SCIENCE



RISK
SCIENCE

OpenQuake: calculate, share, explore

The OpenQuake-engine User Instruction Manual

globalquakemodel.org/openquake

Authors

Helen Crowley¹, Damiano Monelli², Marco Pagani¹, Vitor Silva³, Graeme Weatherill³, Anirudh Rao³

¹ GEM Foundation
via Ferrata, 1
20133 Pavia
Italy

² GEM Model Facility
SED - ETHZ
Sonneggstrasse, 5
CH-8092 Zurich
Switzerland

³ GEM Model Facility
via Ferrata, 1
20133 Pavia
Italy

Email address for all the authors:

<name.surname>@globalquakemodel.org

Citation

Please cite this document as:

Crowley, H., Monelli, D., Pagani, M., Silva, V., Weatherill, G., and Rao, A. (2016). The OpenQuake-engine User Manual. *Global Earthquake Model (GEM) Technical Report 2016-02*. doi: 10.13117/GEM.OPENQUAKE.MAN.ENGINE.1.8, 164 pages.

Disclaimer

The OpenQuake-engine User Manual is distributed in the hope that it will be useful, but without any warranty: without even the implied warranty of merchantability or fitness for a particular purpose. While every precaution has been taken in the preparation of this document, in no event shall the authors of the Manual and the GEM Foundation be liable to any party for direct, indirect, special, incidental, or consequential damages, including lost profits, arising out of the use of information contained in this document or from the use of programs and source code that may accompany it, even if the authors and GEM Foundation have been advised of the possibility of such damage. The Manual provided hereunder is on as “as is” basis, and the authors and GEM Foundation have no obligations to provide maintenance, support, updates, enhancements, or modifications.

License

This Manual is distributed under the Creative Commons License Attribution- NonCommercial- ShareAlike 4.0 International ([CC BY-NC-SA 4.0](#)). You can download this Manual and share it with others as long as you provide proper credit, but you cannot change it in any way or use it commercially.

Contents

Preface	11
I Introduction	13
1 OpenQuake-engine Background	15
1.1 Running the OpenQuake-engine	15
1.2 Concurrent Computing with OpenQuake	17
II Hazard	19
2 Introduction to the Hazard Module	21
2.1 Source typologies	21
2.1.1 Source typologies for modelling distributed seismicity	22
2.1.1.1 Point sources	22
2.1.1.2 Grid sources	24
2.1.1.3 Area sources	25
2.1.2 Fault sources with floating ruptures	27
2.1.2.1 Simple faults	27
2.1.2.2 Complex faults	29
2.1.3 Fault sources without floating ruptures	31
2.1.3.1 Characteristic faults	31
2.1.3.2 Non-Parametric Fault	33
2.2 Magnitude-frequency distributions	36
2.3 Magnitude-scaling relationships	39
2.3.1 Relationships for shallow earthquakes in active tectonic regions	39
2.3.2 Magnitude-scaling relationships for subduction earthquakes	39
2.3.3 Magnitude-scaling relationships stable continental regions	39

2.3.4	Miscellaneous Magnitude-Scaling Relationships	40
2.4	Calculation workflows	40
2.4.1	Classical Probabilistic Seismic Hazard Analysis	40
2.4.2	Event-Based Probabilistic Seismic Hazard Analysis	41
2.4.3	Scenario based Seismic Hazard Analysis	41
3	Using the Hazard Module	43
3.1	Defining Logic Trees	43
3.1.0.1	<i>Logic trees as described in the nrml schema</i>	45
3.2	The Seismic Source System	50
3.2.1	The Seismic Source Logic Tree	50
3.2.2	The Seismic Source Model	52
3.3	The Ground Motion System	53
3.3.0.1	<i>The Ground Motion Logic Tree</i>	53
3.4	Configuration file	54
3.4.1	Classical PSHA	54
3.4.2	Seismic hazard disaggregation	58
3.4.3	Event based PSHA	59
4	Hazard Calculations and Results	61
4.1	Running OpenQuake-engine for hazard calculations	61
4.2	Exporting results from a hazard calculation	62
4.3	Description of hazard outputs	63
4.3.1	Outputs from Classical PSHA	63
4.3.2	Outputs from Hazard Disaggregation	65
4.3.3	Outputs from Event Based PSHA	67

5	Demonstrative Examples	71
5.1	Classical PSHA Demos	71
5.1.1	Classical PSHA with different source typologies	72
5.1.2	Classical PSHA with non trivial logic trees	76
5.2	Hazard Disaggregation Demos	83
5.3	Event Based PSHA Demos	84
III	Risk	89
6	Introduction to the Risk Module	91
6.1	Scenario Damage Assessment	92
6.2	Scenario Risk Assessment	93
6.3	Classical Probabilistic Seismic Damage Analysis	94
6.4	Classical Probabilistic Seismic Risk Analysis	95
6.5	Event-Based Probabilistic Seismic Risk Analysis	96
6.6	Retrofit Benefit-Cost Ratio Analysis	98
7	Risk Input Models	99
7.1	Exposure models	99
7.2	Fragility models	105
7.3	Consequence models	108
7.4	Vulnerability models	110
8	Using the Risk Module	113
8.1	Scenario Damage Calculator	114
8.2	Scenario Risk Calculator	118

8.3	Classical Probabilistic Seismic Risk Calculator	119
8.4	Classical Probabilistic Seismic Damage Calculator	119
8.5	Event-Based Probabilistic Seismic Risk Calculator	119
8.6	Retrofit Benefit-Cost Ratio Calculator	120
8.7	Running the Risk Calculators	121
9	Risk Calculations and Results	123
9.1	Exporting risk results	123
9.2	Description of the outputs	124
9.2.1	Loss statistics	124
9.2.2	Loss maps	125
9.2.3	Damage distribution	127
9.2.4	Collapse maps	130
9.2.5	Loss exceedance curves	131
9.2.6	Retrofitting Benefit/cost ratio maps	131
9.2.7	Loss disaggregation	133
9.2.8	Event loss tables	134
10	Demonstrative Examples	137
10.1	Scenario Damage Demos	138
10.2	Scenario Risk Demos	139
10.3	Classical Probabilistic Seismic Damage Demos	140
10.4	Classical Probabilistic Seismic Risk Demos	140
10.5	Event-Based Probabilistic Seismic Risk Demos	141
10.6	Retrofit Benefit-Cost Ratio Demos	142
	Bibliography	143

Glossary 147

List of Terms 147

Preface

The goal of this manual is to provide a comprehensive and transparent description of the features of the OpenQuake-engine (v1.7). This manual is designed to be readable by someone with basic understanding of Probabilistic Seismic Hazard and Risk Analysis, but no previous knowledge of the OpenQuake- engine is assumed.

The OpenQuake-engine is an effort promoted and actively developed by the Global Earthquake Model, a public-private partnership initiated by the Global Science Forum of the Organisation for Economic Co-operation and Development (OECD)¹.

The OpenQuake-engine is the result of an effort carried out jointly by the Information Technology and Scientific teams working at the GEM Secretariat. It is freely distributed under an Afferro GPL license (<http://www.gnu.org/licenses/agpl-3.0.html>).

¹A short description of the process promoted by OECD is available here:
<http://www.oecd.org/science/sci-tech/theglobalearthquakemodellgem.htm>

Part I

Introduction

1. OpenQuake-engine Background

OpenQuake-engine (oq-engine) is the seismic hazard and risk calculation software developed by the Global Earthquake Model. By following current standards in software developments like test-driven development and continuous integration, OpenQuake-engine aims at becoming an open, and community-driven tool for seismic hazard and risk analysis.

The source code of the OpenQuake-engine is available on a public web-based repository at the following address:

<http://github.com/gem/oq-engine>.

1.1 Running the OpenQuake-engine

An oq-engine analysis is launched from the command line of a terminal.

A schematic list of the options that can be used for the execution of the oq-engine can be obtained with the following command:

```
user@ubuntu:~$ oq-engine --help
```

The result is the following:

```
usage: oq-engine [-h]
                  [--version]
                  [--log-file LOG_FILE]
                  [--log-level debug,info,progress,warning,error,critical]
                  [--no-distribute]
                  [--run CONFIG_FILE] [--yes]
                  [--config-file CONFIG_FILE]
                  [--make-html-report YYYY-MM-DD|today]
                  [--upgrade-db]
                  [--version-db]
                  [--what-if-I-upgrade]
```

```

[--run-hazard CONFIG_FILE]
[--list-hazard-calculations]
[--delete-calculation CALCULATION_ID]
[--delete-uncompleted-calculations]
[--run-risk CONFIG_FILE]
[--hazard-output-id HAZARD_OUTPUT_ID]
[--hazard-calculation-id HAZARD_CALCULATION_ID]
[--list-risk-calculations]
[--list-outputs CALCULATION_ID]
[--show-view CALCULATION_ID VIEW_NAME]
[--exports EXPORT_FORMATS]
[--export-output OUTPUT_ID TARGET_DIR]
[--export-outputs CALCULATION_ID TARGET_DIR]

OpenQuake Seismic Hazard and Risk Analysis Engine

optional arguments:
  -h, --help            show this help message and exit

General:
  --version             Display version information
  --log-file LOG_FILE, -L LOG_FILE
                        Location to store log messages; if not specified, log
                        messages will be printed to the console (to stderr)
  --log-level           debug,info,progress,warn,error,critical
                        -l debug,info,progress,warn,error,critical
                        Defaults to "info"
  --no-distribute, --nd
                        Disable calculation task distribution and run the
                        computation in a single process. This is intended for
                        use in debugging and profiling.
  --run CONFIG_FILE     Run a job with the specified configuration file; you
                        can also pass a comma-separated pair of files for
                        hazard and risk
  --yes, -y              Automatically answer "yes" when asked to confirm an
                        action
  --config-file CONFIG_FILE
                        Custom openquake.cfg file, to override default
                        configurations
  --make-html-report YYYY-MM-DD|today, -r YYYY-MM-DD|today
                        Build an HTML report of the computation at the given
                        date

Database:
  --upgrade-db          Upgrade the openquake database
  --version-db           Show the current version of the openquake database
  --what-if-I-upgrade   Show what will happen to the openquake database if you
                        upgrade

Hazard:
  --run-hazard CONFIG_FILE, --rh CONFIG_FILE
                        Run a hazard job with the specified config file
  --list-hazard-calculations, --lhcc
                        List hazard calculation information
  --delete-calculation CALCULATION_ID, --dc CALCULATION_ID
                        Delete a calculation and all associated outputs

```

```
--delete-uncompleted-calculations, --duc
Delete all the uncompleted calculations

Risk:
--run-risk CONFIG_FILE, --rr CONFIG_FILE
Run a risk job with the specified config file
--hazard-output-id HAZARD_OUTPUT_ID, --ho HAZARD_OUTPUT_ID
Use the desired hazard output as input for the risk
job
--hazard-calculation-id HAZARD_CALCULATION_ID, --hc HAZARD_CALCULATION_ID
Use the desired hazard job as input for the risk job
--list-risk-calculations, --lrc
List risk calculation information

Export:
--list-outputs CALCULATION_ID, --lo CALCULATION_ID
List outputs for the specified calculation
--show-view CALCULATION_ID VIEW_NAME, --sv CALCULATION_ID VIEW_NAME
Show a view of the specified calculation
--exports EXPORT_FORMATS
Comma-separated string specifying the export formats,
in order of priority
--export-output OUTPUT_ID TARGET_DIR, --eo OUTPUT_ID TARGET_DIR
Export the desired output to the specified directory
--export-outputs CALCULATION_ID TARGET_DIR, --eos CALCULATION_ID TARGET_DIR
Export all the calculation outputs to the specified
directory
```

1.2 Concurrent Computing with OpenQuake

The OpenQuake-engine supports concurrent computing on both standalone computers and computer clusters.

The OpenQuake-engine works by splitting a computation into a number of tasks which are then processed in parallel. The user has the ability to control the splitting procedure, at least to a certain extent, by setting the parameter ‘concurrent_tasks’ in the job.ini file. The engine will try to produce a number of tasks close to ‘concurrent_tasks’: it could be more, it could be less. The details of the algorithm used can change depending on the release of the engine and this is why they are not documented here. Instead, we will document how you can set the parameter to a sensible value.

For instance, suppose you have a standard PC with an i7 processor with 8 hyperthreaded cores, i.e. 4 real cores. You could set:

```
concurrent_tasks = 16
```

and then each hyperthreaded core would process around 2 tasks, which is a reasonable value. If your computation consumes a lot of memory, you could increase ‘concurrent_tasks’, thus producing more tasks of smaller size, requiring less memory.

If you don’t set the parameter, a default value is used. Currently the default is set to 8 times the number of cores in your controller machine. This default value for ‘concurrent_tasks’

is likely to change in the future and you should not rely on it if you are using a computer cluster. If you are not using a cluster, the default value should be a reasonable choice.

Now, suppose you have a cluster with a controller node and 10 workers, each of which has 8 hyperthreaded cores, making for 80 cores in total. In this scenario you could set:

```
concurrent_tasks = 160
```

If you did not set the parameter, the default (assuming 8 cores on the controller machine) would be $8 * 8 = 64$ tasks, which is not enough. The number of available cores on the workers is 80, so 16 cores will remain unused. Our suggestion is to provide a value:

```
concurrent_tasks = 2 * number of (hyperthread) cores in the workers
```

or more, if the computation has memory issues. With more tasks, less memory is used, but more data is transferred and the computation becomes slower.

If ‘concurrent_tasks’ is set to zero, the parallelization is disabled and the job is executed by using a single core. This is useful when debugging errors.

Part II

Hazard

Source typologies

Source typologies for modelling distributed seismicity

Fault sources with floating ruptures

Fault sources without floating ruptures

Magnitude-frequency distributions

Magnitude-scaling relationships

Relationships for shallow earthquakes in active tectonic regions

Magnitude-scaling relationships for subduction earthquakes

Magnitude-scaling relationships stable continental regions

Miscellaneous Magnitude-Scaling Relationships

Calculation workflows

Classical Probabilistic Seismic Hazard Analysis

Event-Based Probabilistic Seismic Hazard Analysis

Scenario based Seismic Hazard Analysis

2. Introduction to the Hazard Module

The hazard component of the OpenQuake-engine builds on top of the OpenQuake hazard library (`oq-hazardlib`), a Python-based library containing tools for PSHA calculations.

The web repository of this library is available at the following address:

<http://github.com/gem/oq-hazardlib>.

In this section we briefly illustrate the main properties of the hazard component of the OpenQuake-engine. In particular, we will describe the main typologies of sources supported and the main calculation workflows available.

2.1 Source typologies

An `oq`-engine seismic source input model contains a list of sources belonging to a finite set of possible typologies. Each source type is defined by a set of parameters - called source data - which are used to specify the source geometry and the properties of seismicity occurrence.

Currently the `oq`-engine supports the following source types:

- Sources for modelling distributed seismicity:
 - Point source - The elemental source type used to model distributed seismicity. Grid and area sources (described below) are different containers of point sources.
 - Area source - So far, the most frequently adopted source type in national and regional PSHA models.
 - Grid source - A replacement for area sources admitting spatially variable seismicity occurrence properties.
- Fault sources with floating ruptures:
 - Simple fault source - The simplest fault model in the OpenQuake-engine. This source is habitually used to describe shallow seismogenic faults.
 - Complex fault source - Often used to model subduction interface sources with a complex geometry.

- Fault sources with ruptures always covering the entire fault surface:
 - Characteristic fault source - A typology of source where ruptures always fill the entire fault surface.

The OpenQuake-engine contains some basic assumptions for the definition of these source typologies:

- In the case of area and fault sources, the seismicity is homogeneously distributed over the source;
- Seismicity temporal occurrence follows a Poissonian model.

The above sets of sources may be referred to as “parametric” sources, that is to say that the generation of the Earthquake rupture forecast is done by the OpenQuake engine based on the parameters of the sources set by the user. In some cases, particularly if the user wishes for the temporal occurrence model to be non-Poissonian (such as the lognormal or Brownian Passage Time models) a different type of behaviour is needed. For this OpenQuake-engine supports a Non-parametric source in which the Earthquake rupture forecast is provided explicitly by the user as a set of ruptures and their corresponding probabilities of occurrence.

2.1.1 Source typologies for modelling distributed seismicity

2.1.1.1 Point sources

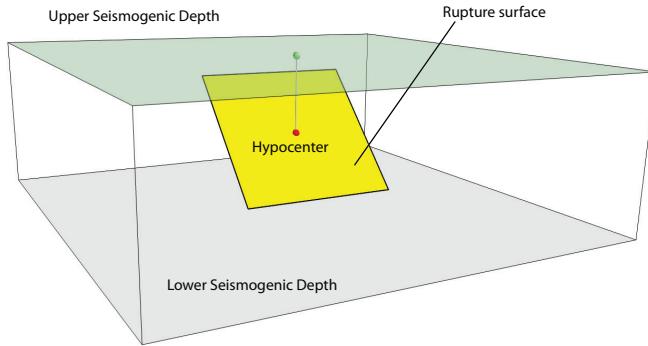


Figure 2.1 – Single rupture

The point source is the elemental source type adopted in the OpenQuake-engine for modelling distributed seismicity. The OpenQuake-engine always performs calculations considering finite ruptures, even in the case of point sources.

These are the basic assumptions used to generate ruptures with point sources:

- Ruptures have a rectangular shape
- Rupture hypocenter is located in the middle of the rupture
- Ruptures are limited at the top and at the bottom by two planes parallel to the topographic surface and placed at two characteristic depths named upper and lower seismogenic depths, respectively (see Figure 2.1)

Source data

For the definition of a point source the following parameters are required (Figure 2.1 shows some of the parameters described below, together with an example of the surface of a generated rupture):

- The coordinates of the point (i.e. longitude and latitude) [decimal degrees]
- The upper and lower seismogenic depths [km]
- One magnitude-frequency distribution
- One magnitude-scaling relationship
- The rupture aspect ratio
- A distribution of nodal planes i.e. one (or several) instances of the following set of parameters:
 - strike [degrees]
 - dip [degrees]
 - rake [degrees]
- A magnitude independent depth distribution of hypocenters [km].

Figure 2.2 shows ruptures generated by a point source for a range of magnitudes. Each rupture is centered on the single hypocentral position admitted by this point source. Ruptures are created by conserving the area computed using the specified magnitude-area scaling relationship and the corresponding value of magnitude.

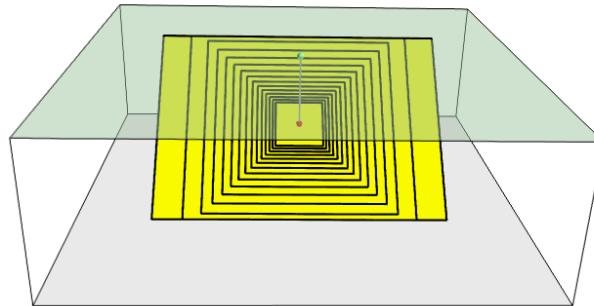


Figure 2.2 – Point source with multiple ruptures. Note the change in the aspect ratio once the rupture width fills the entire seismogenic layer.

Below we provide the excerpt of an .xml file used to describe the properties of a point source:

```

1 <pointSource id="1" name="point" tectonicRegion="Stable Continental Crust">
2   <pointGeometry>
3     <gml:Point>
4       <gml:pos>-122.0 38.0</gml:pos>
5     </gml:Point>
6     <upperSeismoDepth>0.0</upperSeismoDepth>
7     <lowerSeismoDepth>10.0</lowerSeismoDepth>

```

```

8   </pointGeometry>
9   <magScaleRel>WC1994</magScaleRel>
10  <ruptAspectRatio>0.5</ruptAspectRatio>
11  <truncGutenbergRichterMFD aValue="-3.5" bValue="1.0" minMag="5.0"
12    maxMag="6.5" />
13  <nodalPlaneDist>
14    <nodalPlane probability="0.3" strike="0.0" dip="90.0" rake="0.0" />
15    <nodalPlane probability="0.7" strike="90.0" dip="45.0" rake="90.0" />
16  </nodalPlaneDist>
17  <hypoDepthDist>
18    <hypoDepth probability="0.5" depth="4.0" />
19    <hypoDepth probability="0.5" depth="8.0" />
20  </hypoDepthDist>
21</pointSource>

```

The red part shows the the parameters used to describe the geometry of the point source, the blue part is the description of the magnitude-frequency distribution, the green text shows the nodal plane distribution and the text in magenta illustrates the hypocentral depth distribution.

The text in black describes the parameters needed to generate the ruptures such as the magnitude-scaling relationship and the aspect ratio.

Note that in this example, ruptures occur on two possible nodal planes and two hypocentral depths. Figure 2.3 shows the ruptures generated by the point source specified above.

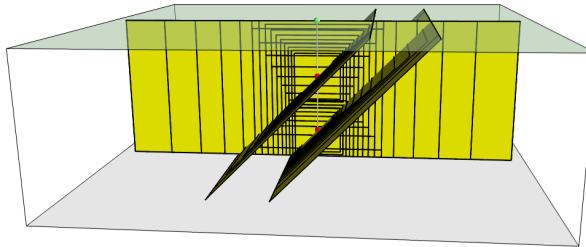


Figure 2.3 – Ruptures produced by the source created using the information in the example .xml file described at page 24.

2.1.1.2 Grid sources

A grid source is simply a collection of point sources distributed over a regular grid (usually equally spaced in longitude and latitude).

In probabilistic seismic hazard analysis a grid source can be considered a model alternative to area sources, since they both model distributed seismicity.

Grid sources are generally used to reproduce more faithfully the spatial pattern of seismicity depicted by the earthquakes occurred in the past; in some models (e.g. Petersen

et al. (2008)) only events of low and intermediate magnitudes are considered.

Grid sources are generally computed using seismicity smoothing algorithms (Frankel, 1995; Woo, 1996, amongst many others).

The use of smoothing algorithms to produce grid sources brings some advantages compared to area sources, since (1) it removes most of the unavoidable degree of subjectivity due to the definition of the geometries of the area sources and (2) it produces a spatial pattern of seismicity that is usually closer to what observed in the reality. Nevertheless, in many cases smoothing algorithms require an a-priori definition of some setup parameters that expose the calculation to a certain degree of partiality.

Grid sources are modeled in oq-engine simply as a set of point sources; in other words, a grid source is just a long list of point sources specified as described in the previous section (see page 22).

2.1.1.3 Area sources

Area sources are usually adopted to describe the seismicity occurring over wide areas where the identification and characterization - i.e. the unambiguous definition of position, geometry and seismicity occurrence parameters - of single fault structures is difficult.

From a computation standpoint, area sources are comparable to grid sources since they are both represented in the engine by a list of point sources.

The oq-engine using the source data parameters (see below) creates an equally spaced in distance grid of point sources where each point has the same seismicity occurrence properties (i.e. rate of events generated).

Below we provide a brief description of the parameters necessary to completely describe an area source.

Source data

- A polygon defining the external border of the area (i.e. a list of Longitude-Latitude [degrees] tuples) The current version of the OQ-engine doesn't support the definition of internal borders.
- The upper and lower seismogenic depths [km]
- One glsmfd
- One glmsr
- The rupture aspect ratio
- A distribution of nodal planes i.e. one (or several) instances of the following set of parameters
 - strike [degrees]
 - dip [degrees]
 - rake [degrees]
- A magnitude independent depth distribution of hypocenters [km].

Below we provide the excerpt of an .xml file used to describe the properties of an area

source:

```

1 <areaSource id="1" name="Quito" tectonicRegion="Active Shallow Crust">
2   <areaGeometry>
3     <gml:Polygon>
4       <gml:exterior>
5         <gml:LinearRing>
6           <gml:posList>
7             -122.5 37.5
8             -121.5 37.5
9             -121.5 38.5
10            -122.5 38.5
11          </gml:posList>
12        </gml:LinearRing>
13      </gml:exterior>
14    </gml:Polygon>
15    <upperSeismoDepth>0.0</upperSeismoDepth>
16    <lowerSeismoDepth>10.0</lowerSeismoDepth>
17  </areaGeometry>
18  <magScaleRel>PeerMSR</magScaleRel>
19  <ruptAspectRatio>1.5</ruptAspectRatio>
20  <incrementalMFD minMag="6.55" binWidth="0.1">
21    <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4 6.108288E-4
22      5.080653E-4</occurRates>
23 </incrementalMFD>
24 <nodalPlaneDist>
25   <nodalPlane probability="0.3" strike="0.0" dip="90.0" rake="0.0"/>
26   <nodalPlane probability="0.7" strike="90.0" dip="45.0" rake="90.0"/>
27 </nodalPlaneDist>
28 <hypodepthDist>
29   <hypodepth probability="0.5" depth="4.0" />
30   <hypodepth probability="0.5" depth="8.0" />
31 </hypodepthDist>
32 </areaSource>
```

The red text describes the parameters used to describe the geometry of the area source; the blue part is the description of the magnitude-frequency distribution; the green text displays the nodal plane distribution; and the text in magenta illustrates the hypocentral depth distribution.

The text in gray describes the parameters required to generate the ruptures such as the magnitude-scaling relationship and the aspect ratio.

The ruptures generated by the area source described in the example above are controlled by two nodal planes and have hypocenters at localized at two distinct depths.

2.1.2 Fault sources with floating ruptures

Fault sources in the oq-engine are classified according to the method adopted to distribute ruptures over the fault surface. Two options are currently supported:

- With the first option, ruptures with a surface lower than the whole fault surface are floated so as to cover as much as possible homogeneously the fault surface. This model is compatible with all the supported magnitude-frequency distributions.
- With the second option, ruptures always fill the entire fault surface. This model is compatible with magnitude-frequency distributions similar to a characteristic model (à la Schwartz and Coppersmith, 1984).

In this subsection we discuss the different fault source types that support floating ruptures. In the next subsection we will illustrate the fault typology available to model a characteristic rupturing behaviour.

2.1.2.1 Simple faults

Simple Faults are the most common source type used to model shallow faults; the “simple” adjective relates to the geometry description of the source which is obtained by projecting the fault trace (i.e. a polyline) along a characteristic dip direction.

The parameters used to create an instance of this source type are described in the following paragraph.

Source data

- A fault trace (usually a polyline). It is a list of longitude-latitude tuples [degrees]
- A frequency-magnitude distribution
- A magnitude-scaling relationship
- A representative value of the dip angle (specified following the Aki-Richards convention; see Aki and Richards (2002)) [degrees]
- Rake angle (specified following the Aki-Richards convention; see Aki and Richards (2002)) [degrees]
- Upper and lower depth values limiting the seismogenic interval [km]

For near-fault probabilistic seismic hazard analysis, two additional parameters are needed for characterising seismic sources:

- A hypocentre list. It is a list of the possible hypocentral positions, and the corresponding weights, e.g., `alongStrike="0.25" downDip="0.25" weight="0.25"`. Each hypocentral position is defined in relative terms using as a reference the upper left corner of the rupture and by specifying the fraction of rupture length and rupture width.
- A slip list. It is a list of the possible rupture slip directions [degrees], and their corresponding weights. The angle describing each slip direction is measured counter-clockwise using the fault strike direction as reference.

In near-fault PSHA calculations, the hypocentre list and the slip list are mandatory. The weights in each list must always sum to one. The available GMPE which currently supports the

near-fault directivity PSHA calculation in OQ- engine is the ChiouYoungs2014NearFaultEffect GMPE developed by Chiou and Youngs (2014) (associated with an Active Shallow Crust tectonic region type).

Below we provide two examples of simple fault source files. The first is an excerpt of an xml file used to describe the properties of a simple fault source and the second example shows the excerpt of an xml file used to describe the properties of a simple fault source that can be used to perform a PSHA calculation taking into account directivity effects.

```

1 <areaSource id="1" name="Quito" tectonicRegion="Active Shallow Crust">
2   <areaGeometry>
3     <gml:Polygon>
4       <gml:exterior>
5         <gml:LinearRing>
6           <gml:posList>
7             -122.5 37.5
8             -121.5 37.5
9             -121.5 38.5
10            -122.5 38.5
11          </gml:posList>
12        </gml:LinearRing>
13      </gml:exterior>
14    </gml:Polygon>
15    <upperSeismoDepth>0.0</upperSeismoDepth>
16    <lowerSeismoDepth>10.0</lowerSeismoDepth>
17  </areaGeometry>
18  <magScaleRel>PeerMSR</magScaleRel>
19  <ruptAspectRatio>1.5</ruptAspectRatio>
20  <incrementalMFD minMag="6.55" binWidth="0.1">
21    <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4 6.108288E-4
22      5.080653E-4</occurRates>
23  </incrementalMFD>
24  <nodalPlaneDist>
25    <nodalPlane probability="0.3" strike="0.0" dip="90.0" rake="0.0"/>
26    <nodalPlane probability="0.7" strike="90.0" dip="45.0" rake="90.0"/>
27  </nodalPlaneDist>
28  <hypoDepthDist>
29    <hypoDepth probability="0.5" depth="4.0" />
30    <hypoDepth probability="0.5" depth="8.0" />
31  </hypoDepthDist>
32 </areaSource>
```

Below is an excerpt of a simple fault source xml file for near-fault directivity PSHA calculations:

```

1 <simpleFaultSource id="1" name="Mount Diablo Thrust"
2   tectonicRegion="Active Shallow Crust">
3   <simpleFaultGeometry>
```

```

4      <gml:LineString>
5          <gml:posList>
6              -121.82290 37.73010
7              -122.03880 37.87710
8      </gml:posList>
9      </gml:LineString>
10     <dip>45.0</dip>
11     <upperSeismoDepth>10.0</upperSeismoDepth>
12     <lowerSeismoDepth>20.0</lowerSeismoDepth>
13 </simpleFaultGeometry>
14 <magScaleRel>WC1994</magScaleRel>
15 <ruptAspectRatio>1.5</ruptAspectRatio>
16 <incrementalMFD minMag="5.0" binWidth="0.1">
17     <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4 6.108288E-4
18         5.080653E-4</occurRates>
19 </incrementalMFD>
20 <rake>30.0</rake>
21 <hypoList>
22     <hypo alongStrike="0.25" downDip="0.25" weight="0.25"/>
23     <hypo alongStrike="0.25" downDip="0.75" weight="0.25"/>
24     <hypo alongStrike="0.75" downDip="0.25" weight="0.25"/>
25     <hypo alongStrike="0.75" downDip="0.75" weight="0.25"/>
26 </hypoList>
27 <slipList>
28     <slip weight="0.333"> 0.0 </slip>
29     <slip weight="0.333"> 45.0 </slip>
30     <slip weight="0.334"> 90.0 </slip>
31 </slipList>
32 </simpleFaultSource>
```

As with the previous examples, the red text highlights the parameters used to specify the source geometry, the parameters in green describe the rupture mechanism, the text in blue describes the magnitude-frequency distribution and the gray text describes the rupture properties.

2.1.2.2 Complex faults

A complex fault differs from simple fault just by the way the geometry of the fault surface is defined and the fault surface is later created. The input parameters used to describe complex faults are, for the most part, the same used to describe the simple fault typology.

In case of complex faults the dip angle is not requested while the fault trace is substituted by two fault edges limiting at the top and bottom the fault surface. Additional curves lying over the fault surface can be specified to complement and refine the description of the fault surface geometry.

Usually, we use complex faults to model intraplate megathrust faults such as the big subduction structures active in the Pacific (Sumatra, South America, Japan) but this source

typology can be used also to create - for example - listric fault sources with a realistic geometry.

```

1 <complexFaultSource id="1" name="Cascadia Megathrust"
2   tectonicRegion="Subduction Interface">
3   <complexFaultGeometry>
4     <faultTopEdge>
5       <gml:LineString>
6         <gml:posList>
7           -124.704 40.363 0.5493260E+01
8           -124.977 41.214 0.4988560E+01
9           -125.140 42.096 0.4897340E+01
10      </gml:posList>
11    </gml:LineString>
12  </faultTopEdge>
13  <intermediateEdge>
14    <gml:LineString>
15      <gml:posList>
16        -124.704 40.363 0.5593260E+01
17        -124.977 41.214 0.5088560E+01
18        -125.140 42.096 0.4997340E+01
19      </gml:posList>
20    </gml:LineString>
21  </intermediateEdge>
22  <intermediateEdge>
23    <gml:LineString>
24      <gml:posList>
25        -124.704 40.363 0.5693260E+01
26        -124.977 41.214 0.5188560E+01
27        -125.140 42.096 0.5097340E+01
28      </gml:posList>
29    </gml:LineString>
30  </intermediateEdge>
31  <faultBottomEdge>
32    <gml:LineString>
33      <gml:posList>
34        -123.829 40.347 0.2038490E+02
35        -124.137 41.218 0.1741390E+02
36        -124.252 42.115 0.1752740E+02
37      </gml:posList>
38    </gml:LineString>
39  </faultBottomEdge>
40 </complexFaultGeometry>
41 <magScaleRel>WC1994</magScaleRel>
42 <ruptAspectRatio>1.5</ruptAspectRatio>
43 <truncGutenbergRichterMFD aValue="-3.5" bValue="1.0" minMag="5.0"
44   maxMag="6.5" />
45 <rake>30.0</rake>
```

46 </complexFaultSource>

As with the previous examples, the red text highlights the parameters used to specify the source geometry, the parameters in green describe the rupture mechanism, the text in blue describes the magnitude-frequency distribution and the gray text describes the rupture properties.

2.1.3 Fault sources without floating ruptures

2.1.3.1 Characteristic faults

The characteristic fault source is a particular typology of fault created with the assumption that its ruptures will always cover the entire fault surface. As such, no floating is necessary on the surface. The characteristic fault may still take as input a magnitude frequency distribution. In this case, the fault surface can be represented either as a simple fault source surface or as a complex fault source surface or as a combination of rectangular ruptures as represented in Figure 2.4. Multiple surfaces containing mixed geometry types are also supported.

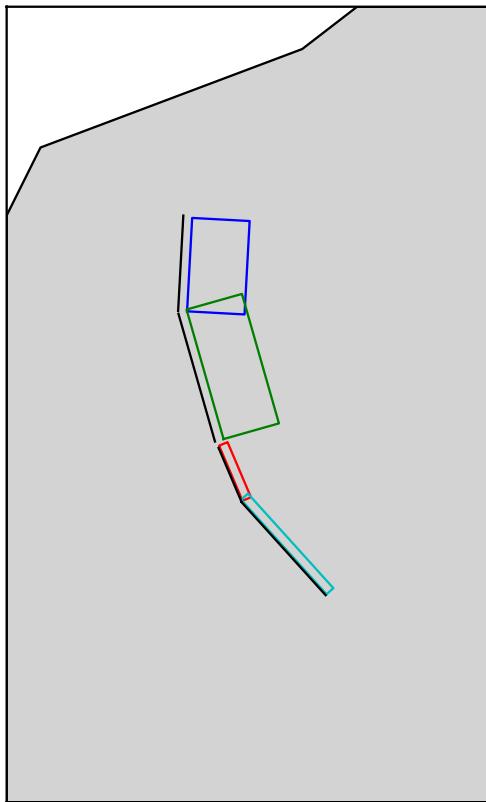


Figure 2.4 – Geometry of a multi-segmented characteristic fault composed of four rectangular ruptures as modelled in OpenQuake.

Source data

- The characteristic rupture surface is defined through one of the following options:
 - A list of rectangular ruptures (“planar surfaces”)
 - A simple fault source geometry
 - A complex fault source geometry
- A frequency-magnitude distribution.
- Rake angle (specified following the Aki-Richards convention; see Aki and Richards (2002)).
- Upper and lower depth values limiting the seismogenic interval.

A comprehensive example enumerating the possible rupture surface configurations is shown below.

```

1  <characteristicFaultSource
2    id="5"
3    name="characteristic source, simple fault"
4    tectonicRegion="Volcanic">
5      <truncGutenbergRichterMFD aValue="-3.5" bValue="1.0"
6        minMag="5.0" maxMag="6.5" />
7      <rake>30.0</rake>
8      <surface>
9        <!-- Characteristic Fault with a simple fault surface -->
10       <simpleFaultGeometry>
11         <gml:LineString>
12           <gml:posList>
13             -121.82290 37.73010
14             -122.03880 37.87710
15           </gml:posList>
16         </gml:LineString>
17         <dip>45.0</dip>
18         <upperSeismoDepth>10.0</upperSeismoDepth>
19         <lowerSeismoDepth>20.0</lowerSeismoDepth>
20       </simpleFaultGeometry>
21     </surface>
22   </characteristicFaultSource>
23   <characteristicFaultSource
24     id="6"
25     name="characteristic source, complex fault"
26     tectonicRegion="Volcanic">
27     <incrementalMFD minMag="5.0" binWidth="0.1">
28       <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4</occurRates>
29     </incrementalMFD>
30     <rake>60.0</rake>
31     <surface>
32       <!-- Characteristic source with a complex fault surface -->
33       <complexFaultGeometry>
34         <faultTopEdge>
35           <gml:LineString>
```

```

36      <gml:posList>
37          -124.704 40.363 0.5493260E+01
38          -124.977 41.214 0.4988560E+01
39          -125.140 42.096 0.4897340E+01
40      </gml:posList>
41      </gml:LineString>
42  </faultTopEdge>
43  <faultBottomEdge>
44      <gml:LineString>
45          <gml:posList>
46              -123.829 40.347 0.2038490E+02
47              -124.137 41.218 0.1741390E+02
48              -124.252 42.115 0.1752740E+02
49          </gml:posList>
50          </gml:LineString>
51      </faultBottomEdge>
52  </complexFaultGeometry>
53 </surface>
54 </characteristicFaultSource>
55 <characteristicFaultSource
56     id="7"
57     name="characteristic source, multi surface"
58     tectonicRegion="Volcanic">
59     <truncGutenbergRichterMFD aValue="-3.6" bValue="1.0"
60                               minMag="5.2" maxMag="6.4" />
61     <rake>90.0</rake>
62     <surface>
63         <!-- Characteristic source with a collection of planar surfaces -->
64         <planarSurface strike="0.0" dip="90.0">
65             <topLeft lon="-1.0" lat="1.0" depth="21.0" />
66             <topRight lon="1.0" lat="1.0" depth="21.0" />
67             <bottomLeft lon="-1.0" lat="-1.0" depth="59.0" />
68             <bottomRight lon="1.0" lat="-1.0" depth="59.0" />
69         </planarSurface>
70         <planarSurface strike="20.0" dip="45.0">
71             <topLeft lon="1.0" lat="1.0" depth="20.0" />
72             <topRight lon="3.0" lat="1.0" depth="20.0" />
73             <bottomLeft lon="1.0" lat="-1.0" depth="80.0" />
74             <bottomRight lon="3.0" lat="-1.0" depth="80.0" />
75         </planarSurface>
76     </surface>
77 </characteristicFaultSource>
```

2.1.3.2 Non-Parametric Fault

The non-parametric fault typology requires that the user indicates the rupture properties (rupture surface, magnitude, rake and hypocentre) and the corresponding probabilities of the rupture. The probabilities are given as a list of floating point values that correspond to the probabilities of 0, 1, 2,.....N occurrences of the rupture within the specified investigation

time. Note that there is not, at present, any internal check to ensure that the investigation time to which the probabilities refer corresponds to that specified in the configuration file. As the surface of the rupture is set explicitly, no rupture floating occurs, and, as in the case of the characteristic fault source, the rupture surface can be defined as either a single planar rupture, a list of planar ruptures, a simple fault source geometry, a complex fault source geometry, or a combination of different geometries.

An comprehensive example enumerating the possible configurations is shown below:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <nrmrl
3 xmlns="http://openquake.org/xmlns/nrml/0.5"
4 xmlns:gml="http://www.opengis.net/gml"
5 >
6   <sourceModel name="Test Source Model">
7     <nonParametricSeismicSource id="1"
8       name="A Non Parametric Source" tectonicRegion="Some TRT">
9       <singlePlaneRupture probs_occur="0.544 0.456">
10      <magnitude>8.3</magnitude>
11      <rake>90.0</rake>
12      <hypocenter depth="26.101" lat="40.726" lon="143.0"/>
13      <planarSurface>
14        <topLeft depth="9.0" lat="41.6" lon="143.1"/>
15        <topRight depth="9.0" lat="40.2" lon="143.91"/>
16        <bottomLeft depth="43.202" lat="41.252" lon="142.07"/>
17        <bottomRight depth="43.202" lat="39.852" lon="142.91"/>
18      </planarSurface>
19    </singlePlaneRupture>
20    <multiPlanesRupture probs_occur="0.9244 0.0756">
21      <magnitude>6.9</magnitude>
22      <rake>0.0</rake>
23      <hypocenter depth="7.1423" lat="35.296" lon="139.31"/>
24      <planarSurface>
25        <topLeft depth="2.0" lat="35.363" lon="139.16"/>
26        <topRight depth="2.0" lat="35.394" lon="138.99"/>
27        <bottomLeft depth="14.728" lat="35.475" lon="139.19"/>
28        <bottomRight depth="14.728" lat="35.505" lon="139.02"/>
29      </planarSurface>
30      <planarSurface>
31        <topLeft depth="2.0" lat="35.169" lon="139.34"/>
32        <topRight depth="2.0" lat="35.358" lon="139.17"/>
33        <bottomLeft depth="12.285" lat="35.234" lon="139.45"/>
34        <bottomRight depth="12.285" lat="35.423" lon="139.28"/>
35      </planarSurface>
36    </multiPlanesRupture>
37    <simpleFaultRupture probs_occur="0.157 0.843">
38      <magnitude>7.8</magnitude>
39      <rake>90.0</rake>
40      <hypocenter depth="22.341" lat="43.624" lon="147.94"/>
```

```
41      <simpleFaultGeometry>
42          <gml:LineString>
43              <gml:posList>
44                  147.96 43.202
45                  148.38 43.438
46                  148.51 43.507
47                  148.68 43.603
48                  148.76 43.640
49              </gml:posList>
50          </gml:LineString>
51          <dip>30.0</dip>
52          <upperSeismoDepth>14.5</upperSeismoDepth>
53          <lowerSeismoDepth>35.5</lowerSeismoDepth>
54      </simpleFaultGeometry>
55  </simpleFaultRupture>
56  <complexFaultRupture probs_occur="0.157 0.843">
57      <magnitude>7.8</magnitude>
58      <rake>90.0</rake>
59      <hypocenter depth="22.341" lat="43.624" lon="147.94"/>
60      <complexFaultGeometry>
61          <faultTopEdge>
62              <gml:LineString>
63                  <gml:posList>
64                      148.76 43.64 5.0
65                      148.68 43.603 5.0
66                      148.51 43.507 5.0
67                      148.38 43.438 5.0
68                      147.96 43.202 5.0
69                  </gml:posList>
70              </gml:LineString>
71          </faultTopEdge>
72          <faultBottomEdge>
73              <gml:LineString>
74                  <gml:posList>
75                      147.92 44.002 35.5
76                      147.81 43.946 35.5
77                      147.71 43.897 35.5
78                      147.5 43.803 35.5
79                      147.36 43.727 35.5
80                  </gml:posList>
81              </gml:LineString>
82          </faultBottomEdge>
83      </complexFaultGeometry>
84  </complexFaultRupture>
85  </nonParametricSeismicSource>
86  </sourceModel>
87 </nrml>
```

2.2 Magnitude-frequency distributions

The magnitude-frequency distributions currently supported by the oq-engine are the following:

A discrete incremental magnitude-frequency distribution It is the simplest distribution supported. It is defined by the minimum value of magnitude (representing the mid point of the first bin) and the bin width. The distribution itself is simply a sequence of floats describing the annual number of events for different bins. The maximum magnitude admitted by this magnitude-frequency distribution is just the sum of the minimum magnitude and the product of the bin width by the number annual rate values.

Below we provide an example of the xml that should be incorporated in a seismic source description in order to define this Magnitude-Frequency Distribution (MFD). An additional example for this distribution can be also found at page 28.

```
<incrementalMFD minMag="5.05" binWidth="0.1">
    <occurRates>0.15 0.08 0.05 0.03 0.015</occurRates>
</incrementalMFD>
```

The magnitude-frequency distribution obtained with the above parameters is represented in Figure 2.5.

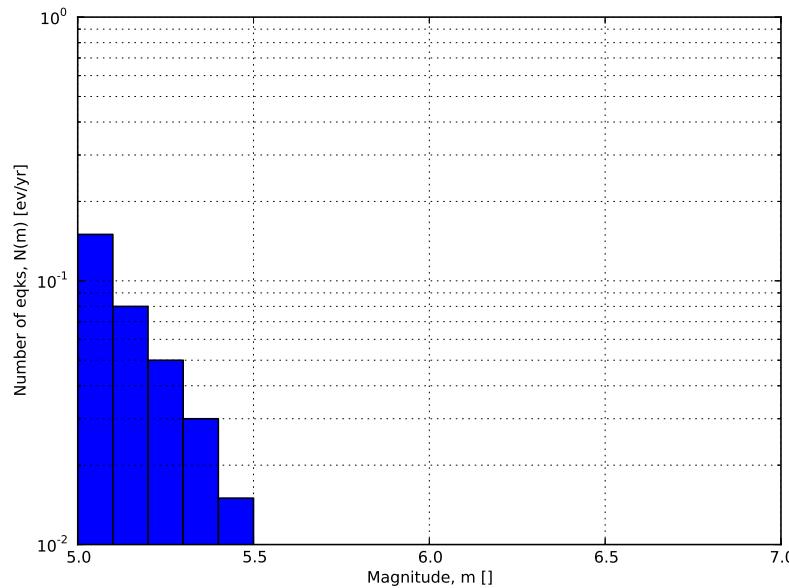


Figure 2.5 – Example of an incremental magnitude-frequency distribution.

A double truncated Gutenberg-Richter distribution This distribution is described by means of a minimum `minMag` and maximum magnitude `maxMag` and by the a and b values of the Gutenberg-Richter relationship.

The syntax of the xml used to describe this magnitude-frequency distribution is rather compact as demonstrated in the following example:

```
<truncGutenbergRichterMFD aValue="5.0" bValue="1.0" minMag="5.0"
    maxMag="6.0"/>
```

Figure 2.6 shows the magnitude-frequency distribution obtained using the parameters of the considered example.

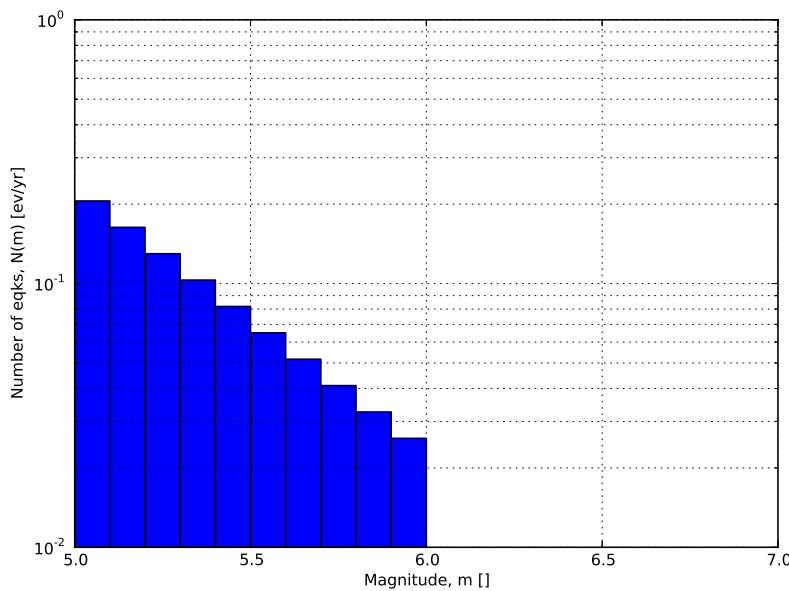


Figure 2.6 – Example of a double truncated Gutenberg-Richter magnitude-frequency distribution.

Hybrid Characteristic earthquake model (à la Youngs and Coppersmith, 1985) The hybrid characteristic earthquake model, presented by Youngs and Coppersmith, 1985, distributes seismic moment proportionally between a characteristic model (for larger magnitudes) and an exponential model. The rate of events is dependent on the magnitude of the characteristic earthquake, the b-value and the total moment rate of the system (Figure 2.7). However, the total moment rate may be defined in one of two ways. If the total moment rate of the source is known, as may be the case for a fault surface with known area and slip rate, then the distribution can be defined from the total moment rate (in N-m) of the source directly. Alternatively, the distribution can be defined from the rate of earthquakes in the characteristic bin, which may be preferable if the distribution is determined from observed seismicity behaviour. The option to define the distribution according to the total moment rate is input as:

```
<YoungsCoppersmithMFD minmag="5.0" bValue="1.0" binWidth="0.1"
    characteristicMag="7.0" totalMomentRate="1.05E19"/>
```

whereas the option to define the distribution from the rate of the characteristic events is given as:

```
<YoungsCoppersmithMFD minmag="5.0" bValue="1.0" binWidth="0.1"
characteristicMag="7.0" characteristicRate="0.005"/>
```

Note that in this distribution the width of the magnitude bin must be defined explicitly in the model.

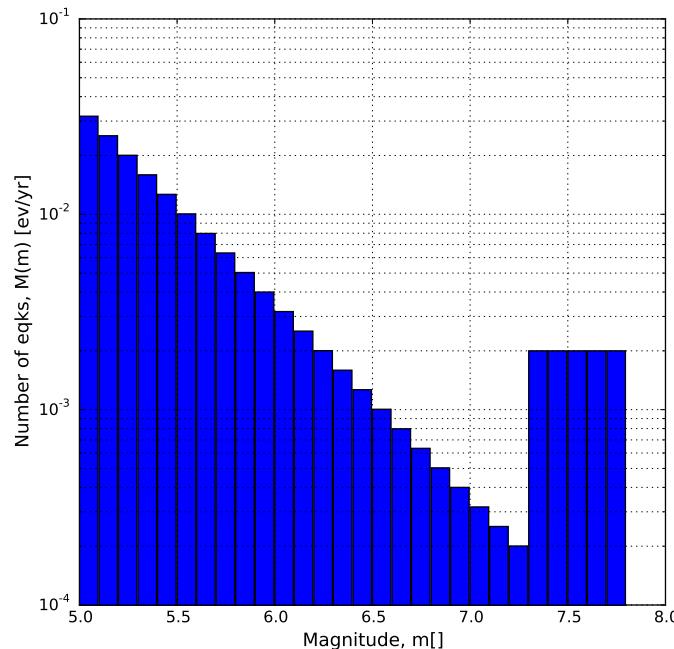


Figure 2.7 – Youngs and Coppersmith, 1985 magnitude-frequency distribution.

“Arbitrary” Magnitude Frequency Distribution The arbitrary magnitude frequency distribution is another non-parametric form of MFD, in which the rates are defined explicitly. Here, the magnitude frequency distribution is defined by a list of magnitudes and their corresponding rates of occurrence. There is no bin-width as the rates correspond exactly to the specific magnitude. Unlike the evenly discretised MFD, there is no requirement that the magnitudes be equally spaced. This distribution (illustrated in Figure 2.8) can be input as:

```
<arbitraryMFD>
<occurRates>0.12 0.036 0.067 0.2</occurRates>
<magnitudes>8.1 8.47 8.68 9.02</magnitude>
</arbitraryMFD>
```



Figure 2.8 – “Arbitrary” magnitude-frequency distribution.

2.3 Magnitude-scaling relationships

We provide below a list of the magnitude-area scaling relationships implemented in the oq-hazardlib:

2.3.1 Relationships for shallow earthquakes in active tectonic regions

- Wells and Coppersmith, 1994 - One of the most well known magnitude scaling relationships, based on a global database of historical earthquake ruptures. The implemented relationship is the one linking magnitude to rupture area, and is called with the keyword WC1994

2.3.2 Magnitude-scaling relationships for subduction earthquakes

- Strasser et al., 2010 - Defines several magnitude scaling relationships for interface and in-slab earthquakes. Only the magnitude to rupture-area scaling relationships are implemented here, and are called with the keywords StrasserInterface and StrasserIntraslab respectively.

2.3.3 Magnitude-scaling relationships stable continental regions

- EPRI, 2011 - Defines a single magnitude to rupture-area scaling relationship for use in the central and eastern United States: $Area = 10.0^{M_W - 4.336}$. It is called with the keyword CEUS2011

2.3.4 Miscellaneous Magnitude-Scaling Relationships

- PeerMSR defines a simple magnitude scaling relation used as part of the Pacific Earthquake Engineering Research Center verification of probabilistic seismic hazard analysis programs: $Area = 10.0^{M_w - 4.0}$.
- PointMSR approximates a ‘point’ source by returning an infinitesimally small area for all magnitudes. Should only be used for distributed seismicity sources and not for fault sources.

2.4 Calculation workflows

The hazard component of the OpenQuake-engine can compute seismic hazard using various approaches. Three types of analysis are currently supported:

- *Classical Probabilistic Seismic Hazard Analysis (PSHA)*, allowing calculation of hazard curves and hazard maps following the classical integration procedure (Cornell, 1968, McGuire (1976)) as formulated by Field et al., 2003.
- *Event-Based Probabilistic Seismic Hazard Analysis*, allowing calculation of ground-motion fields from stochastic event sets. Traditional results - such as hazard curves - can be obtained by post-processing the set of computed ground-motion fields.
- *Scenario Based Seismic Hazard Analysis (SSHA)*, allowing the calculation of ground motion fields from a single earthquake rupture scenario taking into account ground-motion aleatory variability.

Each workflow has a modular structure, so that intermediate results can be exported and analyzed. Each calculator can be extended independently of the others so that additional calculation options and methodologies can be easily introduced, without affecting the overall calculation workflow.

2.4.1 Classical Probabilistic Seismic Hazard Analysis

Input data for the classical Probabilistic Seismic Hazard Analysis (PSHA) consist of a PSHA input model provided together with calculation settings.

The main calculators used to perform this analysis are the following:

1. Logic Tree Processor

The Logic Tree Processor (LTP) takes as an input the PSHA Input Model and creates a Seismic Source Model. The LTP uses the information in the Initial Seismic Source Models and the Seismic Source Logic Tree to create a Seismic Source Input Model (i.e. a model describing geometry and activity rates of each source without any epistemic uncertainty).

Following a procedure similar to the one just described the Logic Tree Processor creates a Ground Motion model (i.e. a data structure that associates to each tectonic region considered in the calculation a Ground Motion Prediction Equation (GMPE)).

2. Earthquake Rupture Forecast Calculator

The produced Seismic Source Input Model becomes an input information for the Earthquake Rupture Forecast (ERF) calculator which creates a list earthquake ruptures admitted by the source model, each one characterized by a probability of occurrence over a specified time span.

3. *Classical PSHA Calculator*

The classical PSHA calculator uses the ERF and the Ground Motion model to compute hazard curves on each site specified in the calculation settings.

2.4.2 Event-Based Probabilistic Seismic Hazard Analysis

Input data for the Event-Based PSHA - as in the case of the Classical PSHA calculator - consists of a PSHA Input Model and a set of calculation settings.

The main calculators used to perform this analysis are:

1. *Logic Tree Processor*

The Logic Tree Processor works in the same way described in the description of the Classical PSHA workflow (see Section 2.4.1 at page 40).

2. *Earthquake Rupture Forecast Calculator*

The Earthquake Rupture Forecast Calculator was already introduced in the description of the PSHA workflow (see Section 2.4.1 at page 40).

3. *Stochastic Event Set Calculator*

The Stochastic Event Set Calculator generates a collection of stochastic event sets by sampling the ruptures contained in the ERF according to their probability of occurrence. A Stochastic Event Set (SES) thus represents a potential realisation of the seismicity (i.e. a list of ruptures) produced by the set of seismic sources considered in the analysis over the time span fixed for the calculation of hazard.

4. *Ground Motion Field Calculator*

The Ground Motion Field Calculator computes for each event contained in a Stochastic Event Set a realization of the geographic distribution of the shaking by taking into account the aleatory uncertainties in the ground- motion model. Eventually, the Ground Motion Field calculator can consider the spatial correlation of the ground-motion during the generation of the Ground Motion Field (GMF).

5. *Event-based PSHA Calculator*

The event-based PSHA calculator takes a (large) set of ground-motion fields representative of the possible shaking scenarios that the investigated area can experience over a (long) time span and for each site computes the corresponding hazard curve.

This procedure is computationally intensive and is not recommended for investigating the hazard over large areas.

2.4.3 Scenario based Seismic Hazard Analysis

In case of SSHA, the input data consist of a single earthquake rupture model and one or more ground-motion models. Using the Ground Motion Field Calculator, multiple realizations of

ground shaking can be computed, each realization sampling the aleatory uncertainties in the ground-motion model.

The main calculators used to perform this analysis are:

1. *Ground Motion Field Calculator*

The Ground Motion Field Calculator was already introduced during the description of the event based PSHA workflow (see Section 2.4.2 at page 41).

Defining Logic Trees

The Seismic Source System

The Seismic Source Logic Tree

The Seismic Source Model

The Ground Motion System

Configuration file

Classical PSHA

Seismic hazard disaggregation

Event based PSHA

3. Using the Hazard Module

This Chapter summarises the structure of the information necessary to define a PSHA input model to be used with the OpenQuake-engine.

Input data for probabilistic based seismic hazard analysis (Classical, Event based, Disaggregation, and UHS) are organised into:

- A general configuration file.
- A file describing the Seismic Source System, that is the set of initial source models and associated epistemic uncertainties needed to model the seismic activity in the region of interest.
- A file describing the Ground Motion System, that is the set of ground motion prediction equations, per tectonic region type, needed to model the ground motion shaking in the region of interest.

Figure 3.1 summarises the structure of a PSHA input model for the OpenQuake-engine and the relationships between the different files.

3.1 Defining Logic Trees

The main components of a logic tree structure in the OpenQuake-engine are the following:

branch : the simplest component of a logic tree structure. A branch represent a possible interpretation of a value assignment for a specific type of uncertainty. It is fully described by the tuple (parameter or model, weight).

branching set : it is a key component in the logic tree structure used by the oq-engine. It groups a set of branches i.e. alternative interpretations of a parameter or a model. Each branching set is defined by:

- An ID
- An uncertainty type (for a comprehensive list of the types of uncertainty currently supported see page 46)

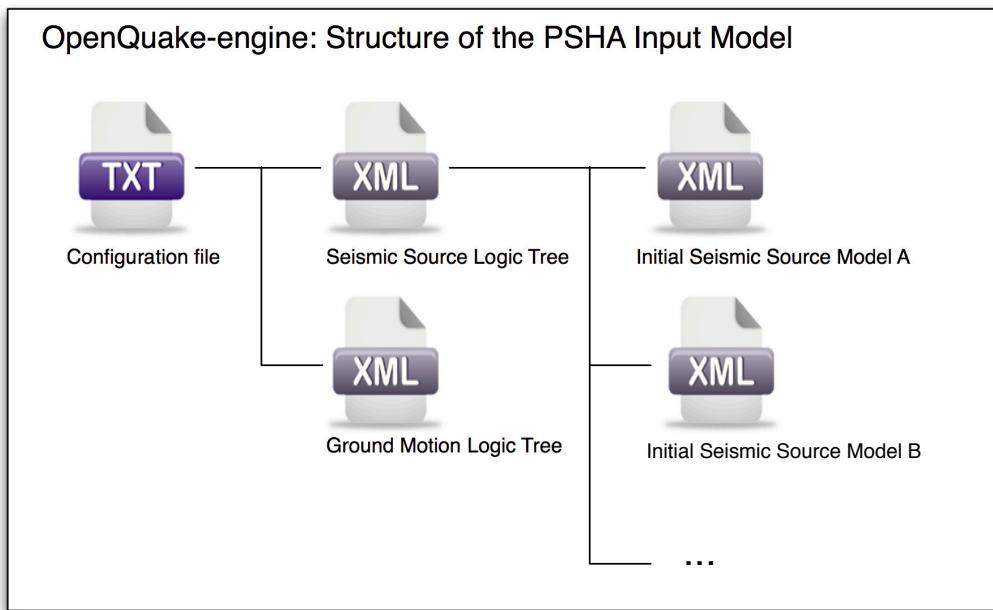


Figure 3.1 – PSHA Input Model structure

- One or more branches

This set of uncertainties can be applied to the whole initial seismic source input model or just to a subset of seismic sources. The sum of the weights/probabilities assigned to the set of branches always correspond to one.

branching level : it is the largest container. It is not used in modelling uncertainty, but it is useful in maintaining a logic and an order in the structure of the tree.

Below we provide a simple schema illustrating the skeleton of xml file containing the description of a logic tree:

```

<logicTreeBranchingLevel branchingLevelID=ID>
    <logicTreeBranchSet branchSetID=ID
        uncertaintyType=TYPE>
        <logicTreeBranch>
            <uncertaintyModel>VALUE</uncertaintyModel>
            <uncertaintyWeight>WEIGHT</uncertaintyWeight>
        </logicTreeBranch>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>
  
```

As it appears from this example, the structure of a logic tree is a set of nested elements. A schematic representation of the elemental components of a logic tree structure is provided in Figure 3.2. A branching level identifies the position where branching occurs while a branch set identifies a collection of branches (i.e. individual branches) whose weights sum to 1.



Figure 3.2 – Generic Logic Tree structure as described in terms of branching levels, branch sets, and individual branches.

3.1.0.1 Logic trees as described in the nrml schema

In the NRML schema, a logic tree structure is defined through the `logicTree` element:

```
<logicTree logicTreeID="ID">
  ...
</logicTree>
```

A `logicTree` contains as a sequence of `logicTreeBranchingLevel` elements. The position in the sequence of a `logicTreeBranchingLevel` specifies the level of the tree where it is located. That is, the first `logicTreeBranchingLevel` element in the sequence represents the first level in the tree, the second element the second level in the tree, and so on.

```
<logicTree logicTreeID="ID">
  <logicTreeBranchingLevel branchingLevelID="ID_1">
    ...
  </logicTreeBranchingLevel>
  <logicTreeBranchingLevel branchingLevelID="ID_2">
    ...
  </logicTreeBranchingLevel>
  ....
```

```

<logicTreeBranchingLevel branchingLevelID="ID_N">
    ...
</logicTreeBranchingLevel>
</logicTree>

```

There are no restrictions on the number of tree levels that can be defined.

A logicTreeBranchingLevel is defined as a sequence of logicTreeBranchSet elements where each logicTreeBranchSet defines a particular epistemic uncertainty inside a branching level.

A branch set has two required attributes: branchSetID and uncertaintyType. The latter defines the type of epistemic uncertainty this branch set is describing.

```

<logicTree logicTreeID="ID">
    ...
    <logicTreeBranchingLevel branchingLevelID="ID_#">
        <logicTreeBranchSet branchSetID="ID_1"
            uncertaintyType="UNCERTAINTY_TYPE">
            ...
        </logicTreeBranchSet>
        <logicTreeBranchSet branchSetID="ID_2"
            uncertaintyType="UNCERTAINTY_TYPE">
            ...
        </logicTreeBranchSet>
        ...
        <logicTreeBranchSet branchSetID="ID_N"
            uncertaintyType="UNCERTAINTY_TYPE">
            ...
        </logicTreeBranchSet>
    </logicTreeBranchingLevel>
    ...
</logicTree>

```

Possible values for the uncertaintyType attribute are:

- gmpeModel: indicates epistemic uncertainties on ground motion prediction equations
- sourceModel: indicates epistemic uncertainties on source models
- maxMagGRRelative: indicates relative (i.e. increments) epistemic uncertainties to be added (or subtracted, depending on the sign of the increment) to the Gutenberg-Richter maximum magnitude value.
- bGRRelative: indicates relative epistemic uncertainties to be applied to the Gutenberg-Richter b value.

- `abGRAbsolute`: indicates absolute (i.e. values used to replace original values) epistemic uncertainties on the Gutenberg-Richter a and b values.
- `maxMagGRAbsolute`: indicates (absolute) epistemic uncertainties on the Gutenberg-Richter maximum magnitude.
- `incrementalMFDAbsolute`: indicates (absolute) epistemic uncertainties on the incremental magnitude frequency distribution (i.e. alternative rates and/or minimum magnitude) of a specific source (can only be applied to individual sources)
- `simpleFaultGeometryAbsolute`: indicates alternative representations of the simple fault geometry for an individual simple fault source
- `simpleFaultDipRelative`: indicates a relative increase or decrease in fault dip for one or more simple fault sources
- `simpleFaultDipAbsolute`: indicates alternative values of fault dip for one or more simple fault sources
- `complexFaultGeometryAbsolute`: indicates alternative representations of complex fault geometry for an individual complex fault source
- `characteristicFaultGeometryAbsolute`: indicates alternative representations of the characteristic fault geometry for an individual characteristic fault source

There are no restrictions on the number of branch sets that can be defined inside a branching level.

A `branchSet` is defined as a sequence of `logicTreeBranch` elements, each specified by an `uncertaintyModel` element (a string identifying an uncertainty model; the content of the string varies with the `uncertaintyType` attribute value of the `branchSet` element) and the `uncertaintyWeight` element (specifying the probability/weight associated to the `uncertaintyModel`):

```
<logicTree logicTreeID="ID">
  ...
    <logicTreeBranchingLevel branchingLevelID="ID_#">
      ...
        <logicTreeBranchSet branchSetID="ID_#"
          uncertaintyType="UNCERTAINTY_TYPE">
          <logicTreeBranch branchID="ID_1">
            <uncertaintyModel>
              UNCERTAINTY_MODEL
            </uncertaintyModel>
            <uncertaintyWeight>
              UNCERTAINTY_WEIGHT
            </uncertaintyWeight>
          </logicTreeBranch>
        ...
      </logicTreeBranchSet>
    </logicTreeBranchingLevel>
  </logicTree>
```

```

<logicTreeBranch branchID="ID_N">
    <uncertaintyModel>
        UNCERTAINTY_MODEL
    </uncertaintyModel>
    <uncertaintyWeight>
        UNCERTAINTY_WEIGHT
    </uncertaintyWeight>
</logicTreeBranch>
</logicTreeBranchSet>
...
</logicTreeBranchingLevel>
...
</logicTree>

```

Depending on the `uncertaintyType` the content of the `<uncertaintyModel>` element changes:

- if `uncertaintyType="gmpeModel"`, the uncertainty model contains the name of a ground motion prediction equation (a list of available GMPEs are given in Appendix ??), e.g.:

```
<uncertaintyModel>GMPE_NAME</uncertaintyModel>
```

- if `uncertaintyType="sourceModel"`, the uncertainty model contains the paths to a source model file, e.g.:

```
<uncertaintyModel>SOURCE_MODEL_FILE_PATH</uncertaintyModel>
```

- if `uncertaintyType="maxMagGRRelative"`, the uncertainty model contains the increment to be added (or subtracted, depending on the sign) to the Gutenberg-Richter maximum magnitude:

```
<uncertaintyModel>MAX_MAGNITUDE_INCREMENT</uncertaintyModel>
```

- if `uncertaintyType="bGRRelative"`, the uncertainty model contains the increment to be added (or subtracted, depending on the sign) to the Gutenberg-Richter b value:

```
<uncertaintyModel>B_VALUE_INCREMENT</uncertaintyModel>
```

- if `uncertaintyType="abGRAbsolute"`, the uncertainty model must contain one a and b pair:

```
<uncertaintyModel>A_VALUE B_VALUE</uncertaintyModel>
```

- if `uncertaintyType="maxMagGRAbsolute"`, the uncertainty model must contain one Gutenberg-Richter maximum magnitude value:

```
<uncertaintyModel>MAX_MAGNITUDE</uncertaintyModel>
```

- if `uncertaintyType="incrementalMFDAbsolute"`, the uncertainty model must contain an instance of the incremental MFD node:

```
<uncertaintyModel>
  <incrementalMFD
    minMag="MIN MAGNITUDE"
    binWidth="BIN WIDTH">
    <occurRates>RATE_1 RATE_2 ... RATE_N</occurRates>
  </incrementalMFD>
</uncertaintyModel>
```

- if `uncertaintyType="simpleFaultGeometryAbsolute"` then the uncertainty model must contain a *valid* instance of the `simpleFaultGeometry` node as described in section 2.1.2.1
- if `uncertaintyType="simpleFaultDipRelative"` then the uncertainty model must specify the number of degrees to increase (positive) or decrease (negative) the fault dip. Note that if this increase results in an adjusted fault dip greater than 90° or less than 0° an error will occur.

```
<uncertaintyModel>DIP_INCREMENT</uncertaintyModel>
```

- if `uncertaintyType="simpleFaultDipAbsolute"` then the uncertainty model must specify the dip angle (in degrees)

```
<uncertaintyModel>DIP</uncertaintyModel>
```

- if `uncertaintyType="complexFaultGeometryAbsolute"` then the uncertainty model must contain a *valid* instance of the `complexFaultGeometry` source node as described in section 2.1.2.2
- if `uncertaintyType="characteristicFaultGeometryAbsolute"` then the uncertainty model must contain a *valid* instance of the `characteristicFaultGeometry` source node, as described in section 2.1.3.1

There are no restrictions on the number of `logicTreeBranch` elements that can be defined in a `logicTreeBranchSet`, as long as the uncertainty weights sum to 1.0.

The `logicTreeBranchSet` element offers also a number of optional attributes allowing for complex tree definitions:

- `applyToBranches`: specifies to which `logicTreeBranch` elements (one or more), in the previous branching level, the branch set is linked to. The linking is established

by defining the IDs of the branches to link to:

```
applyToBranches="branchID1 branchID2 .... branchIDN"
```

The default is the keyword ALL, which means that a branch set is by default linked to all branches in the previous branching level. By specifying one or more branches to which the branch set links to, non-symmetric logic trees can be defined.

- `applyToSources`: specifies to which source in a source model the uncertainty applies to. Sources are specified in terms of their IDs:

```
applyToSources="srcID1 srcID2 .... srcIDN"
```

- `applyToSourceType`: specifies to which source type the uncertainty applies to. Only one source typology can be defined (`area`, `point`, `simpleFault`, `complexFault`), e.g.:

```
applyToSources="area"
```

- `applyToTectonicRegionType`: specifies to which tectonic region type the uncertainty applies to. Only one tectonic region type can be defined (`Active Shallow Crust`, `Stable Shallow Crust`, `Subduction Interface`, `Subduction IntraSlab`, `Volcanic`), e.g.:

```
applyToTectonicRegionType="Active Shallow Crust"
```

3.2 The Seismic Source System

The Seismic Source System contains the model (or the models) describing position, geometry and activity of seismic sources of engineering importance for a set of sites as well as the possible epistemic uncertainties to be incorporated into the calculation of seismic hazard.

3.2.1 The Seismic Source Logic Tree

The structure of the Seismic Source Logic Tree consists of at least one branching level. This branching level is the one used to define the initial seismic source input model (or a number of initial seismic source models, see Figure 3.1).

The example provided below shows the simplest Seismic Source Logic Tree structure that can be defined in a PSHA input model for oq-engine. It's a logic tree with just one branching level containing one branch set with one branch used to define the initial seismic source model (its weight will be equal to one).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <nrm1 xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
```

```

4   <logicTree logicTreeID="lt1">
5     <logicTreeBranchingLevel branchingLevelID="bl1">
6       <logicTreeBranchSet uncertaintyType="sourceModel"
7         branchSetID="bs1">
8         <logicTreeBranch branchID="b1">
9           <uncertaintyModel>seismic_source_model.xml
10          </uncertaintyModel>
11          <uncertaintyWeight>1.0</uncertaintyWeight>
12        </logicTreeBranch>
13      </logicTreeBranchSet>
14    </logicTreeBranchingLevel>
15  </logicTree>
16 </nrml>

```

The optional branching levels will contain rules that modify parameters of the sources in the initial seismic source model.

For example, if the epistemic uncertainties to be considered are source geometry and maximum magnitude, the modeller can create a logic tree structure with three initial seismic source models (each one exploring a different definition of the geometry of sources) and one branching level accounting for the epistemic uncertainty on the maximum magnitude.

Below we provide an example of such logic tree structure:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <nrmr xmlns:gml="http://www.opengis.net/gml"
3    xmlns="http://openquake.org/xmlns/nrml/0.5">
4    <logicTree logicTreeID="lt1">
5
6      <logicTreeBranchingLevel branchingLevelID="bl1">
7        <logicTreeBranchSet uncertaintyType="sourceModel"
8          branchSetID="bs1">
9            <logicTreeBranch branchID="b1">
10           <uncertaintyModel>seismic_source_model_A.xml
11           </uncertaintyModel>
12           <uncertaintyWeight>0.2</uncertaintyWeight>
13         </logicTreeBranch>
14         <logicTreeBranch branchID="b2">
15           <uncertaintyModel>seismic_source_model_B.xml
16           </uncertaintyModel>
17           <uncertaintyWeight>0.3</uncertaintyWeight>
18         </logicTreeBranch>
19         <logicTreeBranch branchID="b3">
20           <uncertaintyModel>seismic_source_model_C.xml
21           </uncertaintyModel>
22           <uncertaintyWeight>0.5</uncertaintyWeight>

```

```

23      </logicTreeBranch>
24    </logicTreeBranchSet>
25  </logicTreeBranchingLevel>
26
27  <logicTreeBranchingLevel branchingLevelID="bl2">
28    <logicTreeBranchSet branchSetID="bs21"
29      uncertaintyType="maxMagGRRelative">
30      <logicTreeBranch branchID="b211">
31        <uncertaintyModel>+0.0</uncertaintyModel>
32        <uncertaintyWeight>0.6</uncertaintyWeight>
33      </logicTreeBranch>
34      <logicTreeBranch branchID="b212">
35        <uncertaintyModel>+0.5</uncertaintyModel>
36        <uncertaintyWeight>0.4</uncertaintyWeight>
37      </logicTreeBranch>
38    </logicTreeBranchSet>
39  </logicTreeBranchingLevel>
40
41  </logicTree>
42 </nrml>

```

Note that the uncertainty on the maximum magnitude is specified in terms of relative increments with respect to the initial maximum magnitude defined for each source in the initial seismic source models.

3.2.2 The Seismic Source Model

The structure of the xml file representing the seismic source model corresponds to a list of sources, each one modelled using one out of the five typologies currently supported. Below we provide a schematic example of a seismic source model:

```

<sourceModel gml:id="ID">
  ...
  <areaSource gml:id="SOURCE_ID">
    <gml:name>SOURCE_NAME</gml:name>
    <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
    ...
  </areaSource>
  ...
  <pointSource gml:id="SOURCE_ID">
    <gml:name>SOURCE_NAME</gml:name>
    <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
    ...
  </pointSource>
  ...

```

```

<simpleFaultSource gml:id="SOURCE_ID">
    <gml:name>SOURCE_NAME</gml:name>
    <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
    ...
</simpleFaultSource>
...
<complexFaultSource gml:id="SOURCE_ID">
    <gml:name>SOURCE_NAME</gml:name>
    <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
    ...
</complexFaultSource>
...
</sourceModel>

```

3.3 The Ground Motion System

The Ground Motion System defines the models and the possible epistemic uncertainties related to ground motion modelling to be incorporated into the calculation.

3.3.0.1 The Ground Motion Logic Tree

The structure of the ground-motion logic tree consists of a list of ground motion prediction equations for each tectonic region used to characterise the sources in the PSHA input model.

The example below shows a simple ground-motion logic tree. This logic tree assumes that all the sources in the PSHA input model belong to “Active Shallow Crust” and uses for calculation the Chiou and Youngs (2008) GMPE.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <nrml xmlns:gml="http://www.opengis.net/gml"
3      xmlns="http://openquake.org/xmlns/nrml/0.5">
4      <logicTree logicTreeID='lt1'>
5          <logicTreeBranchingLevel branchingLevelID="bl1">
6              <logicTreeBranchSet uncertaintyType="gmpeModel"
7                  branchSetID="bs1"
8                  applyToTectonicRegionType="Active Shallow Crust">
9
10             <logicTreeBranch branchID="b1">
11                 <uncertaintyModel>
12                     ChiouYoungs2008
13                 </uncertaintyModel>
14                 <uncertaintyWeight>1.0</uncertaintyWeight>
15             </logicTreeBranch>
16
17         </logicTreeBranchSet>

```

```

18      </logicTreeBranchingLevel>
19    </logicTree>
20  </nrml>

```

3.4 Configuration file

The configuration file is the primary file controlling both the definition of the input model as well as parameters governing the calculation. We illustrate in the following different examples of the configuration file addressing different typologies of seismic hazard calculation.

3.4.1 Classical PSHA

In the following we describe the overall structure and the most typical parameters of a configuration file to be used for the computation of a seismic hazard map using a classical PSHA methodology.

Calculation type and model info

```

1 [general]
2 description = A demo OpenQuake-engine .ini file for classical PSHA
3 calculation_mode = classical
4 random_seed = 1024

```

In this section the user specifies the following parameters:

- **description**: a parameter that can be used to designate the model
- **calculation_mode**: it is used to set the kind of calculation. In this case it corresponds to `classical`. Alternative options for the `calculation_mode` are described later in this manual.
- **random_seed**: is used to control the random generator so that when Monte Carlo procedures are used calculations are replicable (if the same `random_seed` is used you get exactly the same results).

Geometry of the area (or the sites) where hazard is computed

This section is used to specify where the hazard will be computed. Two options are available:

The first option is to define a polygon (usually a rectangle) and a distance (in km) to be used to discretize the polygon area. The polygon is defined by a list of longitude-latitude tuples.

An example is provided below:

```

5 [geometry]
6 region = 10.0 43.0, 12.0 43.0, 12.0 46.0, 10.0 46.0
7 region_grid_spacing = 10.0

```

The second option allows the definition of a number of sites where the hazard will be computed. An example is provided below:

```
5 [geometry]
6 sites = 10.0 43.0, 12.0 43.0, 12.0 46.0, 10.0 46.0
```

If the list of sites is too long the user can specify the name of a csv file as shown below:

```
5 [geometry]
6 sites_csv = <name_of_the_csv_file>
```

The format of the csv file containing the list of sites is a sequence of points (one per row) specified in terms of the longitude, latitude tuple. An example is provided below:

```
1 179.0,90.0
2 178.0,89.0
3 177.0,88.0
```

Logic tree sampling

The oq-engine provides two options for processing the whole logic tree structure. The first option uses Montecarlo sampling; the user in this case specifies a number of realizations.

In the second option all the possible realizations are created. Below we provide an example for the latter option. In this case we set the `number_of_logic_tree_samples` to 0. oq-engine will perform a complete enumeration of all the possible paths from the roots to the leaves of the logic tree structure.

```
9 [logic_tree]
10 number_of_logic_tree_samples = 0
```

If the seismic source logic tree and the ground motion logic tree do not contain epistemic uncertainties the engine will create a single PSHA input.

Generation of the earthquake rupture forecast

```
11 [erf]
12 rupture_mesh_spacing = 5
13 width_of_mfd_bin = 0.1
14 area_source_discretization = 10
```

This section of the configuration file is used to specify the level of discretization of the mesh representing faults, the grid used to delineate the area sources and, the magnitude-frequency distribution. Note that the smaller is the mesh spacing (or the bin width) the larger are (1) the precision in the calculation and (2) the computation demand.

Parameters describing site conditions

```

17 [site_params]
18 reference_vs30_type = measured
19 reference_vs30_value = 760.0
20 reference_depth_to_2pt5km_per_sec = 5.0
21 reference_depth_to_1pt0km_per_sec = 100.0

```

In this section the user specifies local soil conditions. The simplest solution is to define uniform site conditions (i.e. all the sites have the same characteristics).

Alternatively it is possible to define spatially variable soil properties in a separate file; the engine will then assign to each investigation location the values of the closest point used to specify site conditions.

```

17 [site_params]
18 site_model_file = site_model.xml

```

The file containing the site model has the following structure:

```

<?xml version="1.0" encoding="utf-8"?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
  <siteModel>
    <site lon="10.0" lat="40.0" vs30="800.0"
          vs30Type="inferred"
          z1pt0="19.367196734" z2pt5="0.588625072259" />
    <site lon="10.1" lat="40.0" vs30="800.0"
          vs30Type="inferred"
          z1pt0="19.367196734" z2pt5="0.588625072259" />
    <site lon="10.2" lat="40.0" vs30="800.0"
          vs30Type="inferred"
          z1pt0="19.367196734" z2pt5="0.588625072259" />
    <site lon="10.3" lat="40.0" vs30="800.0"
          vs30Type="inferred"
          z1pt0="19.367196734" z2pt5="0.588625072259" />
    <site lon="10.4" lat="40.0" vs30="800.0"
          vs30Type="inferred"
          z1pt0="19.367196734" z2pt5="0.588625072259" />
    ...
  </siteModel>
</nrmr>

```

If the closest available site with soil conditions is at a distance greater than 5 km from the investigation location, a warning is generated.

Calculation configuration

```

22 [calculation]
23 source_model_logic_tree_file = source_model_logic_tree.xml
24 gsim_logic_tree_file = gmpe_logic_tree.xml
25 investigation_time = 50.0
26 intensity_measure_types_and_levels = {"PGA": [0.005, ..., 2.13]}
27 truncation_level = 3
28 maximum_distance = 200.0

```

This section of the oq-engine configuration file specifies the parameters that are relevant for the calculation of hazard. These include the names of the two files containing the Seismic Source System and the Ground Motion System, the duration of the time window used to compute the hazard, the ground motion intensity measure types and levels for which the probability of exceedence will be computed, the level of truncation of the Gaussian distribution of the logarithm of ground motion used in the calculation of hazard and the maximum integration distance (i.e. the distance within which sources will contribute to the computation of the hazard).

Output

```

32 [output]
33 export_dir = outputs/
34 # given the specified 'intensity_measure_types_and_levels'
35 mean_hazard_curves = true
36 quantile_hazard_curves = 0.1 0.5 0.9
37 uniform_hazard_spectra = false
38 individual_curves = false
39 poes_hazard_maps = 0.1

```

The final section of the configuration file is the one that contains the parameters controlling the types of output to be produced. Providing an export directory will tell OpenQuake where to place the output files when the --exports flag is used when running the program. Setting `mean_hazard_curves` to true will result in a specific output containing the mean curves of the logic tree, likewise `quantile_hazard_curves` will produce seperate files containing the quantile hazard curves at the quantiles listed (0.1, 0.5 and 0.9 in the example above, leave blank or omit if no quantiles are required). Setting `uniform_hazard_spectra` to true will output the uniform hazard spectra at the same probabilities of exceedence (poes) as those specified by the later option `poes_hazard_maps`. The probabilities specified here correspond to the set investigation time.

By default, OpenQuake will export the results for each of the logic tree end branches. If the logic tree contains a large number of end branches the process of exporting the results from each end branch can add a significant amount of computation time - and result in a

large volume of disk spaced being used. If instead the user wishes only to export the hazard results for the set mean and quantiles then setting the `individual_curves` option should be set to false.

3.4.2 Seismic hazard disaggregation

In this section we describe the structure of the configuration file to be used to complete a seismic hazard disaggregation. Since only a few parts of the standard configuration file need to be changed we can use the description given in Section 3.4.1 at page 54 as a reference and we emphasize herein major differences.

Calculation type and model info

```
[general]
description = A demo .ini file for PSHA disaggregation
calculation_mode = disaggregation
random_seed = 1024
```

The calculation mode parameter in this case is set as `disaggregation`.

Geometry of the area (or the sites) where hazard is computed

```
[geometry]
sites = 11.0 44.5
```

In the section it is necessary to specify the geographic coordinates of the site (or sites) where the disaggregation will be performed.

Disaggregation parameters

```
[disaggregation]
poes_disagg = 0.02, 0.1
mag_bin_width = 1.0
distance_bin_width = 25.0
coordinate_bin_width = 1.5
num_epsilon_bins = 3
```

With the disaggregation settings shown above we'll disaggregate the intensity measure levels with 10% and 2% probability of exceedance using the `investigation_time` and the intensity measure types defined in the “Calculation configuration” section of the OpenQuake configuration file (see page 56).

The parameters `mag_bin_width`, `distance_bin_width`, `coordinate_bin_width` control the level of discretization of the disaggregation matrix computed. `num_epsilon_bins` indicates the number of bins used to represent the contributions provided by different values of `epsilon`.

If the user is interested in a specific type of disaggregation, we suggest to use a very coarse gridding for the parameters that are not necessary. For example, if the user is interested in a magnitude-distance disaggregation, we suggest the use of very large value for the coordinate_bin_width and to set num_epsilon_bins equal to 1.

3.4.3 Event based PSHA

In the following we describe the sections of the configuration file that are required to complete event based PSHA calculations

Calculation type and model info

This part is almost identical to the corresponding one described in Section 3.4.1. Note the setting of the calculation_mode parameter which now corresponds to event_based.

```

1 [general]
2 description = A demo OpenQuake-engine .ini file for classical PSHA
3 calculation_mode = event_based
4 random_seed = 1024

```

Event based parameters

This section is used to specify the number of stochastic event sets to be generated for each logic tree realisation (each stochastic event set represents a potential realisation of seismicity during the investigation_time specified in the calculation_configuration part). Additionally, in this section the user can specify the spatial correlation model to be used in case for the generation of ground motion fields.

```

[event_based_params]
ses_per_logic_tree_path = 5
ground_motion_correlation_model = JB2009
ground_motion_correlation_params = "vs30_clustering": True

```

The acceptable flags for the parameter vs30_clustering are False and True, with a capital F and T respectively. 0 and 1 are also acceptable flags.

Output

This part substitutes the Output part described in the configuration file example described in the Section 3.4.1 at page 54.

```

[output]
export_dir = /tmp/xxx
ground_motion_fields = true
# post-process ground motion fields into hazard curves,
# given the specified 'intensity_measure_types_and_levels'
hazard_curves_from_gmfs = true
mean_hazard_curves = true

```

```
quantile_hazard_curves = 0.15, 0.5, 0.85  
poes_hazard_maps = 0.1, 0.2
```

The option `hazard_curves_from_gmf`s instructs the user to use the event-based ground motion values to provide hazard curves indicating the probabilities of exceeding the intensity measure levels set previously in the `intensity_measure_types_and_levels` option.

[Running OpenQuake-engine for hazard calculations](#)

[Exporting results from a hazard calculation](#)

Description of hazard outputs

[Outputs from Classical PSHA](#)

[Outputs from Hazard Disaggregation](#)

[Outputs from Event Based PSHA](#)

4. Hazard Calculations and Results

In this Chapter we provide a description of the main commands available for running hazard with the oq-engine and the file formats used to represent the results of the analyses.

A general introduction on the use of OpenQuake-engine is provided in Section 1.1 at page 15. The reader is invited to consult this part before diving into the following sections.

4.1 Running OpenQuake-engine for hazard calculations

The execution of a hazard analysis using the OpenQuake-engine is straightforward. Below we provide an example of the simplest command that can be used to launch a hazard calculation. It consists in the invocation of oq-engine together with the --run or --rh (run-hazard) option, and the name of a configuration file (in the example below it corresponds to job.ini):

```
user@ubuntu:~$ oq-engine --run job.ini
```

The amount of information prompted during the execution of the analysis can be controlled through the --log-level flag as shown in the example below:

```
user@ubuntu:~$ oq-engine --rh job.ini --log-level debug
```

In this example we ask the engine to provide an extensive amount of information (usually not justified for a standard analysis). Alternative options are: `debug`, `info`, `progress`, `warn`, `error`, `critical`.

4.2 Exporting results from a hazard calculation

There are two alternative ways to get results from the OpenQuake-engine: directly through the calculation or by exporting them from the internal oq-engine database once a calculation is completed.

The first option is defined at the OpenQuake-engine invocation through the flag `--exports xml`, as shown in the example below:

```
user@ubuntu:~$ oq-engine --run job.ini --exports xml
```

This will export the results to the `results` directory specified in the `job.ini` file.

The second option allows the user to export the computed results or just a subset of them whenever they want. In order to obtain the list of results of the hazard calculations stored in the oq-engine database the user can utilize the `--lhc` command ('list hazard calculations') to list the hazard calculations:

```
user@ubuntu:~$ oq-engine --lhc
```

The execution of this command will produce a list similar to the one provided below (the numbers in red are the calculations IDs):

```
user@ubuntu:~$ oq-engine --lhc
job_id | latest_job_status | last_update | description
1 | failed | 2013-03-01 09:49:34 | Classical PSHA
2 | successful | 2013-03-01 09:49:56 | Classical PSHA
3 | failed | 2013-03-01 10:24:04 | Classical PSHA
4 | failed | 2013-03-01 10:28:16 | Classical PSHA
5 | failed | 2013-03-01 10:30:04 | Classical PSHA
6 | successful | 2013-03-01 10:31:53 | Classical PSHA
7 | failed | 2013-03-09 08:15:14 | Classical PSHA
8 | successful | 2013-03-09 08:18:04 | Classical PSHA
```

Subsequently the user can get the list of result stored for a specific hazard analysis by using the `--list-outputs`, or `--lo`, command, as in the example below (note that the number in blue emphasizes the result ID):

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
3 | hazard_curve | hc-rlz-6
```

and finally extract an xml file for a specific hazard result:

```
user@ubuntu:~$ oq-engine --export-outputs <result_id> <output_folder>
```

4.3 Description of hazard outputs

The results generated by the OpenQuake-engine are fundamentally of two distinct typologies differentiated by the presence (or absence) of epistemic uncertainty in the PSHA input model.

When epistemic uncertainty is incorporated into the calculation, the OpenQuake-engine calculators (e.g. Classical PSHA, Event Based PSHA, Disaggregation, UHS) produce a set of results (i.e. hazard curves, ground motion fields, disaggregation matrices, UHS, for each logic-tree realisation) which reflects epistemic uncertainties introduced in the PSHA input model. The user can specify that only the mean results and desired quantiles are exported by using the optional configuration `individual_curves=False` in the `job.ini` file.

For each logic tree sample, results are computed and stored. Calculation of results statistics (mean, standard deviation, quantiles) are supported by all the calculators, with the exception of the disaggregation calculator.

4.3.1 Outputs from Classical PSHA

By default, the classical PSHA calculator computes and stores hazard curves for each logic tree sample considered.

When the PSHA input model doesn't contain epistemic uncertainties the results is a set of hazard curves (one for each investigated site). The command below illustrates how is possible to retrieve the group of hazard curves obtained for a calculation with a given identifier `<calc_id>` (see Section 4.2 for an explanation about how to obtain the list of calculations performed with their corresponding ID):

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
3 | hazard_curve | hc-rlz-6
```

In this case the oq-engine computed a group of hazard curves with result ID equal to 3. On the contrary, if the parameter `number_of_logic_tree_samples` in the configuration file is different than zero, then N hazard curves files are generated. The example below shows this case:

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
5 | hazard_curve | hc-rlz-10
6 | hazard_curve | hc-rlz-7
7 | hazard_curve | hc-rlz-8
8 | hazard_curve | hc-rlz-9
9 | hazard_curve | hc-rlz-11
10 | hazard_curve | hc-rlz-12
```

If we export from the database the hazard curves contained in one of the items above using the following command:

```
user@ubuntu:~$ oq-engine --export-outputs <output_id> <output_directory>
```

we obtain a nrml formatted file as represented in the example in the inset below:

```
<?xml version='1.0' encoding='UTF-8'?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <hazardCurves sourceModelTreePath="b1|b212"
    gsimTreePath="b2" IMT="PGA" investigationTime="50.0">
    <IMLs>0.005 0.007 0.0098 ... 1.09 1.52 2.13</IMLs>
    <hazardCurve>
      <gml:Point>
        <gml:pos>10.0 45.0</gml:pos>
      </gml:Point>
      <poEs>1.0 1.0 1.0 ... 0.000688359310522 0.0 0.0</poEs>
    </hazardCurve>
    ...
    <hazardCurve>
      <gml:Point>
        <gml:pos>lon lat</gml:pos>
      </gml:Point>
      <poEs>poe1 poe2 ... poeN</poEs>
    </hazardCurve>
  </hazardCurves>
</nrmr>
```

Notwithstanding the intuitiveness of this file, let's have a brief overview of the information included.

The overall content of this file is a list of hazard curves, one for each investigated site, computed using a PSHA input model representing one possible realisation obtained using the complete logic tree structure.

The attributes of the `hazardCurves` element (see text in red) specify the path of the logic tree used to create the seismic source model (`sourceModelTreePath`) and the ground motion model (`gsimTreePath`) plus the intensity measure type and the investigation time used to compute the probability of exceedance.

The `IMLs` element (in green in the example) contains the values of shaking used by the engine to compute the probability of exceedance in the investigation time. For each site this file contains a `hazardCurve` element which has the coordinates (longitude and latitude in decimal degrees) of the site and the values of the probability of exceedance for all the intensity measure levels specified in the `IMLs` element.

If in the configuration file the calculation of mean hazard curves and hazard curves

corresponding to one or several percentiles have been specified, the list of outputs that we should expect from the OpenQuake-engine corresponds to:

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
17 | hazard_curve | hc-rlz-17
18 | hazard_curve | hc-rlz-18
19 | hazard_curve | hc-rlz-13
20 | hazard_curve | hc-rlz-14
21 | hazard_curve | hc-rlz-15
22 | hazard_curve | hc-rlz-16
23 | hazard_curve | quantile(0.5)-curves-PGA
24 | hazard_map | hazard-map(0.1)-PGA-rlz-17
25 | hazard_map | hazard-map(0.1)-PGA-rlz-18
26 | hazard_map | hazard-map(0.1)-PGA-rlz-13
27 | hazard_map | hazard-map(0.1)-PGA-rlz-14
28 | hazard_map | hazard-map(0.1)-PGA-rlz-15
29 | hazard_map | hazard-map(0.1)-PGA-rlz-16
30 | hazard_map | hazard-map(0.1)-PGA-quantile(0.5)
```

In this example the oq-engine produced hazard curves and hazard maps for six logic tree realisations plus median hazard curves and the median hazard map (both highlighted in red).

The following inset shows a sample of the nrml file used to describe a hazard map:

```
<?xml version='1.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <hazardMap sourceModelTreePath="b1" gsimTreePath="b1"
              IMT="PGA" investigationTime="50.0" poE="0.1">
    <node lon="119.596690957" lat="21.5497682591" iml="0.204569990197"/>
    <node lon="119.596751048" lat="21.6397004197" iml="0.212391638188"/>
    <node lon="119.596811453" lat="21.7296325803" iml="0.221407505615"/>
    ...
  </hazardMap>
</nrml>
```

4.3.2 Outputs from Hazard Disaggregation

The oq-engine output of a disaggregation analysis corresponds to the combination of a hazard curve and a multidimensional matrix containing the results of the disaggregation.

The example below shows the list of disaggregation results obtained for four logic tree realisations.

For each realisation, disaggregation has been completed for two intensity measure levels corresponding to different probabilities of exceedence in the specified investigation time.

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
19 | hazard_curve | hc-rlz-3
20 | hazard_curve | hc-rlz-3
21 | hazard_curve | hc-rlz-4
22 | hazard_curve | hc-rlz-4
23 | disagg_matrix | disagg(0.02)-rlz-3-SA(0.025)-POINT(10.1 40.1)
24 | disagg_matrix | disagg(0.1)-rlz-3-SA(0.025)-POINT(10.1 40.1)
25 | disagg_matrix | disagg(0.02)-rlz-3-PGA-POINT(10.1 40.1)
26 | disagg_matrix | disagg(0.1)-rlz-3-PGA-POINT(10.1 40.1)
27 | disagg_matrix | disagg(0.02)-rlz-4-SA(0.025)-POINT(10.1 40.1)
28 | disagg_matrix | disagg(0.1)-rlz-4-SA(0.025)-POINT(10.1 40.1)
29 | disagg_matrix | disagg(0.02)-rlz-4-PGA-POINT(10.1 40.1)
30 | disagg_matrix | disagg(0.1)-rlz-4-PGA-POINT(10.1 40.1)
```

In the following inset we show an example of the nrml file used to represent the different disaggregation matrices (highlighted in red) produced by OpenQuake-engine:

```
<?xml version='2.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
    <disaggMatrices sourceModelTreePath="b1" gsimTreePath="b1" IMT="PGA"
                    investigationTime="50.0" lon="10.1" lat="40.1"
                    magBinEdges="5.0, 6.0, 7.0, 8.0"
                    distBinEdges="0.0, 25.0, 50.0, 75.0, 100.0"
                    lonBinEdges="9.0, 10.5, 12.0"
                    latBinEdges="39.0, 40.5"
                    pdfBinEdges="-3.0, -1.0, 1.0, 3.0"
                    tectonicRegionTypes="Active Shallow Crust">
        <disaggMatrix type="Mag" dims="3" poE="0.1"
                      iml="0.033424622602">
            <prob index="0" value="0.987374744394"/>
            <prob index="1" value="0.704295394366"/>
            <prob index="2" value="0.0802318409498"/>
        </disaggMatrix>
        <disaggMatrix type="Dist" dims="4" poE="0.1"
                      iml="0.033424622602">
            <prob index="0" value="0.700851969171"/>
            <prob index="1" value="0.936680387051"/>
```

```

<prob index="2" value="0.761883595568"/>
<prob index="3" value="0.238687565571"/>
</disaggMatrix>
<disaggMatrix type="TRT" dims="1" poE="0.1"
    iml="0.033424622602">
    <prob index="0" value="0.996566187011"/>
</disaggMatrix>
<disaggMatrix type="Mag,Dist" dims="3,4" poE="0.1"
    iml="0.033424622602">
    <prob index="2,3" value="0.0"/>
</disaggMatrix>
<disaggMatrix type="Mag,Dist,pdf" dims="3,4,3" poE="0.1"
    iml="0.033424622602">
    <prob index="0,0,0" value="0.0785857271425"/>
    ...
</disaggMatrix>
<disaggMatrix type="Lon,Lat" dims="2,1" poE="0.1"
    iml="0.033424622602">
    <prob index="0,0" value="0.996566187011"/>
    <prob index="1,0" value="0.0"/>
</disaggMatrix>
<disaggMatrix type="Mag,Lon,Lat" dims="3,2,1" poE="0.1"
    iml="0.033424622602">
    <prob index="0,0,0" value="0.987374744394"/>
    <prob index="0,1,0" value="0.0"/>
    <prob index="1,0,0" value="0.704295394366"/>
    <prob index="1,1,0" value="0.0"/>
    <prob index="2,0,0" value="0.0802318409498"/>
    <prob index="2,1,0" value="0.0"/>
</disaggMatrix>
<disaggMatrix type="Lon,Lat,TRT" dims="2,1,1" poE="0.1"
    iml="0.033424622602">
    <prob index="0,0,0" value="0.996566187011"/>
    <prob index="1,0,0" value="0.0"/>
</disaggMatrix>
</disaggMatrices>
</nrml>

```

4.3.3 Outputs from Event Based PSHA

The Event Based PSHA calculator computes and stores stochastic event sets and the corresponding ground motion fields.

This calculator can also produce hazard curves and hazard maps exactly in the same way as done using the Classical PSHA calculator.

The inset below shows an example of the list of results provided by the oq-engine at the end of an event-based PSHA calculation:

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
31 | ses | ses-coll-rlz-19
32 | gmf | gmf-rlz-19
33 | ses | ses-coll-rlz-20
34 | gmf | gmf-rlz-20
35 | hazard_curve | hazard-curve-rlz-19-SA(0.1)
36 | hazard_curve | hazard-curve-rlz-20-SA(0.1)
37 | hazard_curve | hazard-curve-rlz-19-PGA
38 | hazard_curve | hazard-curve-rlz-20-PGA
39 | hazard_curve | mean curve for SA(0.1)
40 | hazard_curve | quantile curve (poe >= 0.15) for imt SA(0.1)
41 | hazard_curve | quantile curve (poe >= 0.50) for imt SA(0.1)
42 | hazard_curve | quantile curve (poe >= 0.85) for imt SA(0.1)
43 | hazard_curve | mean curve for PGA
44 | hazard_curve | quantile curve (poe >= 0.15) for imt PGA
45 | hazard_curve | quantile curve (poe >= 0.50) for imt PGA
46 | hazard_curve | quantile curve (poe >= 0.85) for imt PGA
```

This list in the inset above contains two sets of stochastic events (in red) and two sets of ground motion fields (in blue).

The whole group of stochastic event set and ground motion fields can be exported immediately using the results with id 35 and 25, respectively.

Below is an example showing a nrml file containing a collection of stochastic event sets (2 ruptures):

```
<?xml version='1.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <stochasticEventSetCollection sourceModelTreePath="b1">
    <stochasticEventSet id="12" investigationTime="50.0">
      <rupture id="rlz=00|ses=0001|src=10138|i=0049-00"
               magnitude="4.55" strike="90.0" dip="90.0"
               rake="90.0" tectonicRegion="Active Shallow Crust">
        <planarSurface>
          <topLeft lon="12.233903801" lat="43.256198599"
                  depth="11.3933265259"/>
          <topRight lon="12.263958243" lat="43.2562025344"
                    depth="11.3933265259"/>
          <bottomLeft lon="12.233903801" lat="43.256198599"
                      depth="12.6066734741"/>
```

```

<bottomRight lon="12.263958243" lat="43.2562025344"
              depth="12.6066734741"/>
</planarSurface>
</rupture>
<rupture id="rlz=00|ses=0001|src=12412|i=0121-00"
          magnitude="4.65" strike="135.0" dip="90.0"
          rake="90.0" tectonicRegion="Active Shallow Crust">
<planarSurface>
  <topLeft lon="11.45858812" lat="42.7429056814"
            depth="11.3208667302"/>
  <topRight lon="11.4822820715" lat="42.7256333907"
            depth="11.3208667302"/>
  <bottomLeft lon="11.45858812" lat="42.7429056814"
            depth="12.6791332698"/>
  <bottomRight lon="11.4822820715" lat="42.7256333907"
            depth="12.6791332698"/>
</planarSurface>
</rupture>
</stochasticEventSet>
</stochasticEventSetCollection>
</nrml>

```

The text in red shows the part which describes the id of the generated stochastic event set and the investigation time covered.

The text in green emphasises the portion of the text used to describe a rupture. The information provided describes entirely the geometry of the rupture as well as its rupturing properties (e.g. rake, magnitude). The rupture ID is a string that represents each rupture uniquely (including the case where the same rupture is sampled multiple times). The exact format may be subject to change from time-to-time; however, the ID will usually contain information regarding the specific logic tree branch sample (realisation), the stochastic event set counter, the ID of the source from which the rupture was samples and a unique number indicating the rupture identifier within the earthquake rupture forecast of the specific source.

This is an example of a nrml file containing one ground motion field:

```

<?xml version='1.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <gmfCollection sourceModelTreePath="b1" gsimTreePath="b1">
    <gmfSet investigationTime="50.0" stochasticEventSetId="12">
      <gmf IMT="PGA" ruptureId="rlz=00|ses=0001|src=10138|i=0049-00">
        <node gmv="0.0105891230432" lon="11.1240023202"
              lat="43.5107462335"/>
        <node gmv="0.00905803920023" lon="11.1241875202"
              lat="43.5107462335"/>
      </gmf>
    </gmfSet>
  </gmfCollection>
</nrml>

```

```
    lat="43.6006783941"/>
<node gmv="0.00637664420977" lon="11.1243735810"
      lat="43.6906105547"/>
<node gmv="0.00476533134789" lon="11.1245605075"
      lat="43.7805427153"/>
<node gmv="0.00452594698469" lon="11.1247483046"
      lat="43.8704748759"/>
...
<node gmv="0.00017301076646" lon="11.3782630185"
      lat="44.5129482397"/>
</gmf>
</gmfSet>
</gmfCollection>
</nrm1>
```

Classical PSHA Demos

Classical PSHA with different source typologies

Classical PSHA with non trivial logic trees

Hazard Disaggregation Demos

Event Based PSHA Demos

5. Demonstrative Examples

A number of hazard calculation demos are provided with the oq-engine installation, showing different examples of input and configuration files, for different use cases.

This is the list of demos which illustrate how to use the oq-engine for various seismic hazard analysis:

- AreaSourceClassicalPSHA
- CharacteristicFaultSourceCase1ClassicalPSHA
- CharacteristicFaultSourceCase2ClassicalPSHA
- CharacteristicFaultSourceCase3ClassicalPSHA
- ComplexFaultSourceClassicalPSHA
- Disaggregation
- EventBasedPSHA
- LogicTreeCase1ClassicalPSHA
- LogicTreeCase2ClassicalPSHA
- LogicTreeCase3ClassicalPSHA
- PointSourceClassicalPSHA
- SimpleFaultSourceClassicalPSHA

5.1 Classical PSHA Demos

A number of demos have been designed to show how to perform a classical PSHA calculation using the different available source typologies and how to define non-trivial logic trees. It should be noted that the input files that will be illustrated are valid not only for a classical PSHA calculation but also for event based and disaggregation analysis.

All the classical PSHA demos illustrating the different source typologies (all demos but the ones about Logic Tree definition) share the same GSIM logic tree file, which for clarity is provided below.

Since this logic tree consideres only one tectonic region (i.e. Active Shallow Crust) all the seismic sources will belong be considered active shallow crust sources.

```

<?xml version="1.0" encoding="UTF-8"?>

<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
  <logicTree logicTreeID='lt1'>

    <logicTreeBranchingLevel branchingLevelID="bl1">
      <logicTreeBranchSet
        uncertaintyType="gmpeModel"
        applyToTectonicRegionType="Active Shallow Crust"
        branchSetID="bs1">

        <logicTreeBranch branchID="b1">
          <uncertaintyModel>
            BooreAtkinson2008
          </uncertaintyModel>
          <uncertaintyWeight>1.0</uncertaintyWeight>
        </logicTreeBranch>

      </logicTreeBranchSet>
    </logicTreeBranchingLevel>

  </logicTree>
</nrml>
```

5.1.1 Classical PSHA with different source typologies

This section discusses the following examples:

- AreaSourceClassicalPSHA
- CharacteristicFaultSourceCase1ClassicalPSHA
- CharacteristicFaultSourceCase2ClassicalPSHA
- CharacteristicFaultSourceCase3ClassicalPSHA
- ComplexFaultSourceClassicalPSHA
- PointSourceClassicalPSHA
- SimpleFaultSourceClassicalPSHA

The configuration file (see below) is defined to compute hazard curves for several intensity measure types (PGV, PGA and Spectral acceleration at different periods), hazard maps and uniform hazard spectra for different probabilities of exceedance:

```
[general]
description = ...
calculation_mode = classical
random_seed = 23

[geometry]
region = ...
region_grid_spacing = 5.0

[logic_tree]
number_of_logic_tree_samples = 0

[erf]
rupture_mesh_spacing = 2
width_of_mfd_bin = 0.1
area_source_discretization = 5.0

[site_params]
reference_vs30_type = measured
reference_vs30_value = 600.0
reference_depth_to_2pt5km_per_sec = 5.0
reference_depth_to_1pt0km_per_sec = 100.0

[calculation]
source_model_logic_tree_file = source_model_logic_tree.xml
gsim_logic_tree_file = gmpe_logic_tree.xml
investigation_time = 50.0
intensity_measure_types_and_levels ={  
    "PGV": [2, 4, 6 ,8, 10, ...],  
    "PGA": [0.005, 0.007, ...],  
    "SA(0.025)": [...],  
    "SA(0.05)": [...],  
    "SA(0.1)": [...],  
    "SA(0.2)": [...],  
    "SA(0.5)": [...],  
    "SA(1.0)": [...],  
    "SA(2.0)": [...]}  
truncation_level = 3  
maximum_distance = 200.0
```



Figure 5.1 – Hazard maps (for PGA, 10% in 50 years) as obtained from the different oq-engine source typologies. (a) Point Source. (b) Area source. The solid black line represents the area boundary. (c) Simple Fault Source. The dashed line represents the fault trace, while the solid line the fault surface projection. (d) Complex Fault Source. The solid line represent the fault surface projection (d)

```
[output]
export_dir = ...
mean_hazard_curves = false
quantile_hazard_curves =
hazard_maps = true
uniform_hazard_spectra = true
poes = 0.1 0.02
```

Hazard maps for the different demos are shown in Figure 5.1 and Figure 5.2.

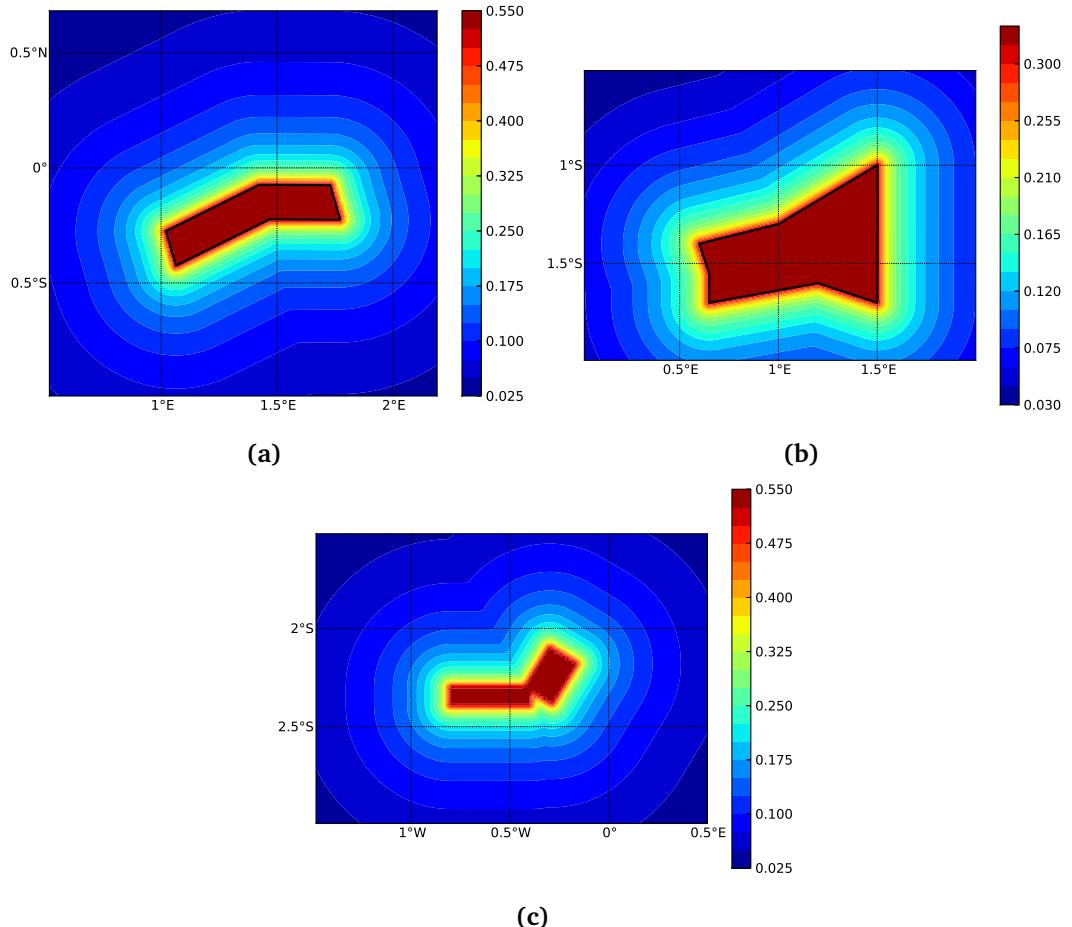


Figure 5.2 – Hazard maps (for PGA, 10% in 50 years) as obtained from characteristic fault sources with simple fault geometry (e), complex fault geometry (f), and collection of planar surfaces (g)

5.1.2 Classical PSHA with non trivial logic trees

Three demos are provided to illustrate how the logic tree formalism can be used to express epistemic uncertainties in seismic hazard analysis.

LogicTreeCase1ClassicalPSHA shows an example of logic tree defining two alternative source models, with sources belonging to two different tectonic region types, and with two alternative GMPEs for each tectonic region type. The source model logic tree is therefore defined in the following way:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <logicTree logicTreeID="lt1">

    <logicTreeBranchingLevel branchingLevelID="bl1">

      <logicTreeBranchSet uncertaintyType="sourceModel"
                           branchSetID="bs1">
        <logicTreeBranch branchID="b1">
          <uncertaintyModel>
            source_model_1.xml
          </uncertaintyModel>
          <uncertaintyWeight>0.5</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b2">
          <uncertaintyModel>
            source_model_2.xml
          </uncertaintyModel>
          <uncertaintyWeight>0.5</uncertaintyWeight>
        </logicTreeBranch>
      </logicTreeBranchSet>

    </logicTreeBranchingLevel>

  </logicTree>
</nrmr>
```

The two source models are defined in two separate files: `source_model_1.xml` and `source_model_2.xml` each one associated to a corresponding weight (0.5 for both).

The GSIM logic tree file contains the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>

<nrmrl xmlns:gml="http://www.opengis.net/gml"
         xmlns="http://openquake.org/xmlns/nrml/0.5">
    <logicTree logicTreeID='lt1'>

        <logicTreeBranchingLevel branchingLevelID="bl1">
            <logicTreeBranchSet uncertaintyType="gmpeModel"
                applyToTectonicRegionType="Active Shallow Crust"
                branchSetID="bs1">
                <logicTreeBranch branchID="b11">
                    <uncertaintyModel>
                        BooreAtkinson2008
                    </uncertaintyModel>
                    <uncertaintyWeight>0.5</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b12">
                    <uncertaintyModel>
                        ChiouYoungs2008
                    </uncertaintyModel>
                    <uncertaintyWeight>0.5</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>

        <logicTreeBranchingLevel branchingLevelID="bl2">
            <logicTreeBranchSet uncertaintyType="gmpeModel"
                applyToTectonicRegionType="Stable Continental Crust"
                branchSetID="bs2">
                <logicTreeBranch branchID="b21">
                    <uncertaintyModel>
                        ToroEtAl2002</uncertaintyModel>
                    <uncertaintyWeight>0.5</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b22">
                    <uncertaintyModel>
                        Campbell2003</uncertaintyModel>
                    <uncertaintyWeight>0.5</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>
    </logicTree>
```

```

        </logicTreeBranchSet>
    </logicTreeBranchingLevel>

    </logicTree>
</nrml>

```

The source model contains sources belonging to Active Shallow Crust and Stable Continental Crust, therefore the GSIM logic tree defines two branching levels, one for each considered tectonic region type. Moreover for each tectonic region a branch set with two GMPEs is defined: Boore and Atkinson 2008 and Chiou and Youngs 2008 for Active Shallow Crust and Toro et al. 2003 and Campbell 2003 for Stable Continental Crust. By processing the above logic tree files using the logic tree path enumeration mode (enabled by setting in the configuration file `number_of_logic_tree_samples = 0`) hazard results are computed for 8 logic tree paths (2 source models x 2 GMPEs for Active x 2 GMPEs for Stable).

LogicTreeCase2ClassicalPSHA defines a single source model consisting of only two sources (area and simple fault) belonging to different tectonic region types (Active Shallow Crust and Stable Continental Region) and both characterized by a truncated Gutenberg-Richter distribution. The logic tree defines uncertainties for G-R a and b values (three possible pairs for each source), maximum magnitude (three values for each source) and uncertainties on the GMPEs for each tectonic region type (two GMPE per region type).

To accommodate such a structure the GSIM logic tree is defined in the following way:

```

<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
    <logicTree logicTreeID="lt1">

        <logicTreeBranchingLevel branchingLevelID="bl1">
            <logicTreeBranchSet uncertaintyType="sourceModel"
                                branchSetID="bs1">
                <logicTreeBranch branchID="b11">
                    <uncertaintyModel>
                        source_model.xml
                    </uncertaintyModel>
                    <uncertaintyWeight>1.0</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>

        <logicTreeBranchingLevel branchingLevelID="bl2">

```

```
<logicTreeBranchSet uncertaintyType="abGRAbsolute"
    applyToSources="1"
    branchSetID="bs21">
    <logicTreeBranch branchID="b21">
        <uncertaintyModel>4.6 1.1</uncertaintyModel>
        <uncertaintyWeight>0.333</uncertaintyWeight>
    </logicTreeBranch>
    <logicTreeBranch branchID="b22">
        <uncertaintyModel>4.5 1.0</uncertaintyModel>
        <uncertaintyWeight>0.333</uncertaintyWeight>
    </logicTreeBranch>
    <logicTreeBranch branchID="b23">
        <uncertaintyModel>4.4 0.9</uncertaintyModel>
        <uncertaintyWeight>0.334</uncertaintyWeight>
    </logicTreeBranch>
</logicTreeBranchSet>
</logicTreeBranchingLevel>

<logicTreeBranchingLevel branchingLevelID="bl3">
    <logicTreeBranchSet uncertaintyType="abGRAbsolute"
        applyToSources="2"
        branchSetID="bs31">
        <logicTreeBranch branchID="b31">
            <uncertaintyModel>3.3 1.0</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b32">
            <uncertaintyModel>3.2 0.9</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b33">
            <uncertaintyModel>3.1 0.8</uncertaintyModel>
            <uncertaintyWeight>0.334</uncertaintyWeight>
        </logicTreeBranch>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>

<logicTreeBranchingLevel branchingLevelID="bl4">
    <logicTreeBranchSet uncertaintyType="maxMagGRAbsolute">
```

```

        applyToSources="1"
        branchSetID="bs41">
    <logicTreeBranch branchID="b41">
        <uncertaintyModel>7.0</uncertaintyModel>
        <uncertaintyWeight>0.333</uncertaintyWeight>
    </logicTreeBranch>
    <logicTreeBranch branchID="b42">
        <uncertaintyModel>7.3</uncertaintyModel>
        <uncertaintyWeight>0.333</uncertaintyWeight>
    </logicTreeBranch>
    <logicTreeBranch branchID="b43">
        <uncertaintyModel>7.6</uncertaintyModel>
        <uncertaintyWeight>0.334</uncertaintyWeight>
    </logicTreeBranch>
</logicTreeBranchSet>
</logicTreeBranchingLevel>

<logicTreeBranchingLevel branchingLevelID="bl5">
    <logicTreeBranchSet uncertaintyType="maxMagGRAbsolute"
        applyToSources="2"
        branchSetID="bs51">
        <logicTreeBranch branchID="b51">
            <uncertaintyModel>7.5</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b52">
            <uncertaintyModel>7.8</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b53">
            <uncertaintyModel>8.0</uncertaintyModel>
            <uncertaintyWeight>0.334</uncertaintyWeight>
        </logicTreeBranch>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>

</logicTree>
</nrml>
```

The first branching level defines the source model. For each source, two branching


```
<logicTreeBranch branchID="b11">
    <uncertaintyModel>
        source_model.xml
    </uncertaintyModel>
    <uncertaintyWeight>1.0</uncertaintyWeight>
</logicTreeBranch>
</logicTreeBranchSet>
</logicTreeBranchingLevel>

<logicTreeBranchingLevel branchingLevelID="bl2">
    <logicTreeBranchSet uncertaintyType="bGRRelative"
        branchSetID="bs21">
        <logicTreeBranch branchID="b21">
            <uncertaintyModel>+0.1</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b22">
            <uncertaintyModel>0.0</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b23">
            <uncertaintyModel>-0.1</uncertaintyModel>
            <uncertaintyWeight>0.334</uncertaintyWeight>
        </logicTreeBranch>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>

<logicTreeBranchingLevel branchingLevelID="bl3">
    <logicTreeBranchSet uncertaintyType="maxMagGRRelative"
        branchSetID="bs31">
        <logicTreeBranch branchID="b31">
            <uncertaintyModel>0.0</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b32">
            <uncertaintyModel>+0.5</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b33">
```

```
<uncertaintyModel>+1.0</uncertaintyModel>
<uncertaintyWeight>0.334</uncertaintyWeight>
</logicTreeBranch>
</logicTreeBranchSet>
</logicTreeBranchingLevel>

</logicTree>
</nrml>
```

After the first branching level defining the source model, two additional branching levels are defined, one defining relative uncertainties on b value (bGRRelative applied consistently to all sources in the source model) and the second uncertainties on maximum magnitude (maxMagGRRelative). Similar to the other cases, two GMPEs are considered for each tectonic region type and therefore the total number of logic tree path is 36 (1 source model x 3 b value increments x 3 maximum magnitude increments x 2 GMPE for Active x 2 GMPEs for Stable).

5.2 Hazard Disaggregation Demos

An example of disaggregation calculation is given considering a source model consisting of two sources (area and simple fault) belonging to two different tectonic region types.

The calculation is defined with the following configuration file:

```
[general]
description = ...
calculation_mode = disaggregation
random_seed = 23

[geometry]
sites = 0.5 -0.5

[logic_tree]
number_of_logic_tree_samples = 0

[erf]
rupture_mesh_spacing = 2
width_of_mfd_bin = 0.1
area_source_discretization = 5.0

[site_params]
```

```

reference_vs30_type = measured
reference_vs30_value = 600.0
reference_depth_to_2pt5km_per_sec = 5.0
reference_depth_to_1pt0km_per_sec = 100.0

[calculation]
source_model_logic_tree_file = source_model_logic_tree.xml
gsim_logic_tree_file = gmpe_logic_tree.xml
investigation_time = 50.0
intensity_measure_types_and_levels = "PGA": [...]
truncation_level = 3
maximum_distance = 200.0

[disaggregation]
poes_disagg = 0.1
mag_bin_width = 1.0
distance_bin_width = 10.0
coordinate_bin_width = 0.2
num_epsilon_bins = 3

[output]
export_dir = ...

```

Disaggregation matrices are computed for a single site (located between the two sources) for a ground motion value corresponding to a probability value equal to 0.1 (poes_disagg = 0.1). Magnitude values are classified in one magnitude unit bins (mag_bin_width = 1.0), distances in bins of 10 km (distance_bin_width = 10.0), coordinates in bins of 0.2 degrees (coordinate_bin_width = 0.2). 3 epsilon bins are considered (num_epsilon_bins = 3).

5.3 Event Based PSHA Demos

A demo showing an example of Event Based PSHA calculation is provided with the following configuration file:

```

[general]
description = Event Based PSHA using Area Source
calculation_mode = event_based
random_seed = 23

```

```
[geometry]
sites = 0.5 -0.5

[logic_tree]
number_of_logic_tree_samples = 0

[erf]
rupture_mesh_spacing = 2
width_of_mfd_bin = 0.1
area_source_discretization = 5.0

[site_params]
reference_vs30_type = measured
reference_vs30_value = 600.0
reference_depth_to_2pt5km_per_sec = 5.0
reference_depth_to_1pt0km_per_sec = 100.0

[calculation]
source_model_logic_tree_file = source_model_logic_tree.xml
gsim_logic_tree_file = gmpe_logic_tree.xml
investigation_time = 50.0
intensity_measure_types_and_levels = "PGA": [...]
truncation_level = 3
maximum_distance = 200.0

[event_based_params]
ses_per_logic_tree_path = 100
ground_motion_correlation_model =
ground_motion_correlation_params =

[output]
export_dir = ...
ground_motion_fields = true
hazard_curves_from_gmfs = true
mean_hazard_curves = false
quantile_hazard_curves =
hazard_maps = true
poes = 0.1
```

The source model consist of one source (area). 100 stochastic event sets are generated



Figure 5.4 – A stochastic event set generated with the event based PSHA demo. The area source defines a nodal plane distribution which distributes events among vertical and dipping (50 degrees) faults with equal weights. Vertical ruptures are then distributed equally in the range 0-180 degrees while the dipping ones in the range 0-360, both with a step of 45 degrees.

(`ses_per_logic_tree_path = 100`) (an example can be seen in Figure 5.4). Ground motion fields are computed (`ground_motion_fields = true`, Figure 5.5) and also hazard curves from ground motion fields are extracted (`hazard_curves_from_gmfss = true`). The corresponding hazard maps for 0.1 probability are also calculated (`hazard_maps = true`)



Figure 5.5 –Ground motion fields (PGA) with no spatial correlations (a) and with spatial correlation (b)

Part III

Risk

Scenario Damage Assessment

Scenario Risk Assessment

Classical Probabilistic Seismic Damage Analysis

Classical Probabilistic Seismic Risk Analysis

Event-Based Probabilistic Seismic Risk Analysis

Retrofit Benefit-Cost Ratio Analysis

6. Introduction to the Risk Module

The seismic risk results are calculated using the OpenQuake risk library (oq-risklib), an open-source suite of tools for seismic risk assessment and loss estimation. This library is written in the Python programming language and available in the form of a “developers” release at the following location: <http://github.com/gem/oq-risklib>.

The risk component of the OpenQuake-engine can compute both scenario-based and probabilistic seismic damage and risk using various approaches. The following types of analysis are currently supported:

- **Scenario Damage Assessment**, for the calculation of damage distribution statistics for a portfolio of buildings from a single earthquake rupture scenario taking into account aleatory and epistemic ground-motion variability.
- **Scenario Risk Assessment**, for the calculation of individual asset and portfolio loss statistics due to a single earthquake rupture scenario taking into account aleatory and epistemic ground-motion variability. Correlation in the vulnerability of different assets of the same typology can also be taken into consideration.
- **Classical Probabilistic Seismic Damage Analysis**, for the calculation of damage state probabilities over a specified time period, and probabilistic collapse maps, starting from the hazard curves computed following the classical integration procedure (Cornell, 1968, McGuire (1976)) as formulated by Field et al., 2003.
- **Classical Probabilistic Seismic Risk Analysis**, for the calculation of loss curves and loss maps, starting from the hazard curves computed following the classical integration procedure (Cornell, 1968, McGuire (1976)) as formulated by Field et al., 2003.
- **Event-Based Probabilistic Seismic Risk Analysis**, for the calculation of event-loss tables starting from stochastic event sets. Other results such as loss-exceedance curves, probabilistic loss maps, average annual losses, and insured loss statistics can be obtained by post- processing the event-loss tables.

- **Retrofit Benefit-Cost Ratio Analysis**, which is useful in estimating the net-present value of the potential benefits of performing retrofitting for a portfolio of assets (in terms of decreased losses in seismic events), measured relative to the upfront cost of retrofitting.

Each calculation workflow has a modular structure, so that intermediate results can be exported and analyzed. Moreover, each calculator can be extended independently of the others so that additional calculation options and methodologies can be easily introduced, without affecting the overall calculation workflow. Each workflow is described in more detail in the following sections.

6.1 Scenario Damage Assessment

The scenario damage calculator computes damage distribution statistics for all assets in a given exposure model for a single specified earthquake rupture. Damage distribution statistics include the mean and standard deviation of damage fractions for different damage states. This calculator requires the definition of a finite rupture model, an exposure model and a fragility model; the main results are the damage distribution statistics per asset, aggregated damage distribution statistics per taxonomy, aggregated damage distribution statistics for the region, and collapse maps, which contain the spatial distribution of the number or area of collapsed buildings throughout the region of interest.

The rupture characteristics—i.e. the magnitude, hypocenter and fault geometry—are modelled as deterministic in the scenario calculators. Multiple realizations of different possible ground motion fields (GMFs) due to the single rupture are generated, taking into consideration both the inter-event variability of ground motions, and the intra-event residuals obtained from a spatial correlation model for ground motion residuals. The use of logic trees allows for the consideration of uncertainty in the choice of a ground motion model for the given tectonic region.

As an alternative to computing the GMFs with OpenQuake, users can also provide their own sets of GMFs as input to the scenario damage calculator.

For each GMF realization, damage fractions (the fraction of buildings in each damage state) are estimated for every asset in the exposure model using the provided fragility model, and finally the damage distribution statistics (i.e., the mean damage fractions and standard deviation of damage fractions for all damage states) across all realizations are calculated. The calculator also provides aggregated damage distribution statistics for the portfolio, such as mean damage fractions and standard deviation of damage fractions for each taxonomy in the exposure model, and the mean damage fractions and standard deviation of damage fractions for the entire region of study.

The required input files required for running a scenario damage calculation and the resulting output files are depicted in Figure 6.1.

Starting with OpenQuake-engine v1.7, consequence model files can also be provided



Figure 6.1 – Scenario Damage Calculator input/output structure.

as inputs for a scenario damage calculation in addition to fragility models files, in order to estimate consequences based on the calculated damage distribution. The user may provide one consequence model file corresponding to each loss type (amongst structural, nonstructural, contents, and business interruption) for which a fragility model file is provided. Whereas providing a fragility model file for at least one loss type is mandatory for running a Scenario Damage calculation, providing corresponding consequence model files is optional.

6.2 Scenario Risk Assessment

The scenario risk calculator computes loss statistics for all assets in a given exposure model for a single specified earthquake rupture. Loss statistics include the mean and standard deviation of ground-up losses and insured losses for each loss type considered in the analysis. Loss statistics can currently be computed for five different loss types using this calculator: structural losses, nonstructural losses, contents losses, downtime losses, and occupant fatalities. This calculator requires the definition of a finite rupture model, an exposure model and a vulnerability model for each loss type considered; the main results are the loss statistics per asset and mean loss maps.

The rupture characteristics—i.e. the magnitude, hypocenter and fault geometry—are modelled as deterministic in the scenario calculators. Multiple realizations of different possible ground motion fields (GMFs) due to the single rupture are generated, taking into consideration both the inter-event variability of ground motions, and the intra-event residuals obtained from a spatial correlation model for ground motion residuals. The use of logic trees allows for the consideration of uncertainty in the choice of a ground motion model for the given tectonic region.

As an alternative to computing the GMFs with OpenQuake, users can also provide their own sets of GMFs as input to the scenario risk calculator.

For each GMF realization, a loss ratio is sampled for every asset in the exposure model using the provided probabilistic vulnerability model, taking into consideration the correlation model for vulnerability of different assets of a given taxonomy. Finally loss statistics, i.e., the mean loss and standard deviation of loss for both ground-up losses and insured losses across all realizations, are calculated for each asset. Mean loss maps are also generated by this calculator, describing the mean ground-up losses and mean insured losses caused by the scenario event for the different assets in the exposure model.

The required input files required for running a scenario risk calculation and the resulting output files are depicted in Figure 6.2.

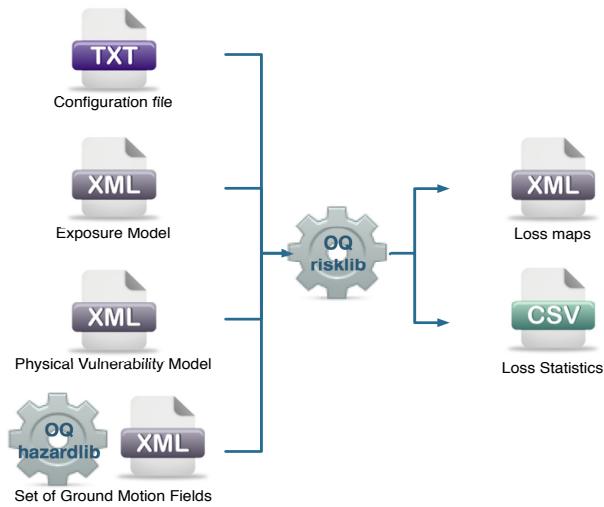


Figure 6.2 – Scenario Risk Calculator input/output structure.

6.3 Classical Probabilistic Seismic Damage Analysis

The classical PSHA-based damage calculator integrates the fragility functions for an asset with the seismic hazard curve at the location of the asset, to give the expected damage distribution for the asset within a specified time period. The calculator requires the definition of an exposure model, a fragility model for each taxonomy represented in the exposure model, and hazard curves calculated in the region of interest. The main results of this calculator are the expected damage distribution for each asset, which describe the probability of the asset being in different damage states, and collapse maps for the region, which describe the probability of collapse for different assets in the portfolio over the specified time period. Damage distribution aggregated by taxonomy or of the total portfolio (considering all assets in the exposure model) can not be extracted using this calculator, as the spatial correlation of the ground motion residuals is not taken into consideration.

The hazard curves required for this calculator can be calculated by the OpenQuake engine for all asset locations in the exposure model using the classical PSHA approach (Cornell, 1968; McGuire, 1976).

The required input files required for running a classical probabilistic damage calculation and the resulting output files are depicted in Figure 6.3.



Figure 6.3 – Classical PSHA-based Damage Calculator input/output structure.

6.4 Classical Probabilistic Seismic Risk Analysis

The classical PSHA-based risk calculator convolves through numerical integration, the probabilistic vulnerability functions for an asset with the seismic hazard curve at the location of the asset, to give the loss distribution for the asset within a specified time period. The calculator requires the definition of an exposure model, a vulnerability model for each loss type of interest with vulnerability functions for each taxonomy represented in the exposure model, and hazard curves calculated in the region of interest. Loss curves and loss maps can currently be calculated for five different loss types using this calculator: structural losses, nonstructural losses, contents losses, downtime losses, and occupant fatalities. The main results of this calculator are loss exceedance curves for each asset, which describe the probability of exceedance of different loss levels over the specified time period, and loss maps for the region, which describe the loss values that have a given probability of exceedance over the specified time

Unlike the probabilistic event-based risk calculator, an aggregate loss curve (considering all assets in the exposure model) can not be extracted using this calculator, as the correlation of the ground motion residuals and vulnerability uncertainty is not taken into consideration in this calculator.

The hazard curves required for this calculator can be calculated by the OpenQuake engine for all asset locations in the exposure model using the classical PSHA approach (Cornell, 1968; McGuire, 1976). The use of logic-trees allows for the consideration of model uncertainty in the choice of a ground motion prediction equation for the different tectonic region types in the region. Unlike what was described in the previous calculator, a total

loss curve (considering all assets in the exposure model) can not be extracted using this calculator, as the correlation of the ground motion residuals and vulnerability uncertainty is not taken into consideration.

The required input files required for running a classical probabilistic risk calculation and the resulting output files are depicted in Figure 6.4.



Figure 6.4 – Classical PSHA-based Risk Calculator input/output structure.

6.5 Event-Based Probabilistic Seismic Risk Analysis

This calculator employs an event-based Monte Carlo simulation approach to probabilistic risk assessment in order to estimate the loss distribution for individual assets and aggregated loss distribution for a spatially distributed portfolio of assets within a specified time period. The calculator requires the definition of an exposure model, a vulnerability model for each loss type of interest with vulnerability functions for each taxonomy represented in the exposure model, and a set of ground motion fields representative of the seismicity of the region over the specified time period. Loss curves and loss maps can currently be calculated for five different loss types using this calculator: structural losses, nonstructural losses, contents losses, downtime losses, and occupant fatalities. The main results of this calculator are loss exceedance curves for each asset, which describe the probability of exceedance of different loss levels over the specified time period, and loss maps for the region, which describe the loss values that have a given probability of exceedance over the specified time period. Aggregate loss exceedance curves can be also be produced using this calculator; these describe the probability of exceedance of different loss levels for all assets in the portfolio. Finally, event loss tables can be produced using this calculator; these tables describe the total loss across the portfolio for each seismic event in the stochastic event set.

This calculator relies on the probabilistic event-based hazard calculator, which simulates the seismicity of the chosen time period T by producing a *stochastic event set* (also known as

a *synthetic catalog*). For each rupture generated by a source, the number of occurrences in the given time span T is simulated by sampling the corresponding probability distribution as given by $P_{rup}(k|T)$. A stochastic event set is therefore a *sample* of the full population of ruptures as defined by a seismic source model. Each rupture is present zero, one or more times, depending on its probability. Symbolically, we can define a stochastic event set (*SES*) as:

$$SES(T) = \{k \times rup, k \sim P_{rup}(k|T) \quad \forall rup \text{ in } Src \quad \forall Src \text{ in } SSM\} \quad (6.1)$$

where k , the number of occurrences, is a random sample of $P_{rup}(k|T)$, and $k \times rup$ means that rupture rup is repeated k times in the stochastic event set.

For each rupture or event in the stochastic event sets, a spatially correlated ground motion field (GMF) realisation is generated, taking into consideration both the inter-event variability of ground motions, and the intra-event residuals obtained from a spatial correlation model for ground motion residuals. The use of logic trees allows for the consideration of uncertainty in the choice of a seismic source model, and in the choice of GMPE models for the different tectonic regions.

For each GMF realization, a loss ratio is sampled for every asset in the exposure model using the provided probabilistic vulnerability model, taking into consideration the correlation model for vulnerability of different assets of a given taxonomy. Finally loss exceedance curves are computed for both ground-up losses and insured losses.

The required input files required for running a probabilistic stochastic event-based risk calculation and the resulting output files are depicted in Figure 6.5



Figure 6.5 – Probabilistic Event-based Risk Calculator input/output structure.

6.6 Retrofit Benefit-Cost Ratio Analysis

This calculator represents a decision-support tool for deciding whether the employment of retrofitting measures to a collection of existing buildings is advantageous from an economical point of view. For this assessment, the expected losses considering the original and retrofitted configuration of the buildings are estimated, and the economic benefit due to the better seismic design is divided by the retrofitting cost, leading to the benefit/cost ratio. These loss curves are computed using the previously described Classical PSHA- based Risk calculator. The output of this calculator is a benefit/cost ratio for each asset, in which a ratio above one indicates that employing a retrofitting intervention is economically viable.

In Figure 6.6, the input/output structure for this calculator is depicted.



Figure 6.6 – Retrofitting Benefit /Cost Ratio Calculator input /output structure.

For further information regarding the theoretical background of the methodologies used for each calculator, users are referred to the OpenQuake- engine Book (Risk).

7. Risk Input Models

The following subsections describe the basic inputs required for a risk calculation, including exposure models, fragility models, consequence models, and vulnerability models. In addition, each risk calculator also requires the appropriate hazard inputs computed in the region of interest. Hazard inputs include hazard curves for the classical probabilistic damage and risk calculators, ground motion fields for the scenario damage and risk calculators, or stochastic event sets for the probabilistic event based calculators.

7.1 Exposure models

All risk calculators in the OpenQuake-engine require an exposure model that needs to be provided in the NRML format. The information included in an exposure model comprises a metadata section listing general information about the exposure, followed by a cost conversions section that describes how the different areas, costs, and occupancies for the assets will be specified, followed by data regarding each individual asset in the portfolio.

A minimal exposure model comprising a single asset is shown in the example below:

Let us take a look at each of the sections in the above example file in turn. The first part of the file contains the metadata section:

The information in the metadata section is common to all of the assets in the portfolio and needs to be incorporated at the beginning of every exposure model file. There are a number of parameters that compose the metadata section, which is intended to provide general information regarding the assets within the exposure model. These parameters are described below:

- **id:** a unique key used to identify the exposure model

- **category**: an optional string used to define the type of assets being stored (e.g: buildings, lifelines)
- **taxonomySource**: an optional attribute used to define the taxonomy being used to classify the assets
- **description**: a brief string with further information about the exposure model

Next, let us look at the part of the file describing the area and cost conversions:

Notice that the `costType` element defines a `name`, a `type`, and a `unit` attribute.

The NRML schema for the exposure model allows the definition of a structural cost, a nonstructural components cost, a contents cost, and a business interruption or downtime cost for each asset in the portfolio. Thus, the valid values for the `name` attribute of the `costType` element are the following:

- **structural**: used to specify the structural replacement cost of assets
- **nonstructural**: used to specify the replacement cost for the nonstructural components of assets
- **contents**: used to specify the contents replacement cost
- **business_interruption**: used to specify the cost that will be incurred per unit time that a damaged asset remains closed following an earthquake

The exposure model shown in the example above defines only the structural values for the assets. However, multiple cost types can be defined for each asset in the same exposure model.

The `unit` attribute of the `costType` element is used for specifying the currency unit for the corresponding cost type. Note that the OpenQuake-engine itself is agnostic to the currency units; the `unit` is thus a descriptive attribute which is used by OpenQuake-engine to annotate the results of a risk assessment. This attribute can be set to any valid Unicode string.

The `type` attribute of the `costType` element specifies whether the costs will be provided as an aggregated value for an asset, or per building or unit comprising an asset, or per unit area of an asset. The valid values for the `type` attribute of the `costType` element are the following:

- **aggregated**: indicates that the replacement costs will be provided as an aggregated value for each asset
- **per_asset**: indicates that the replacement costs will be provided per building or unit comprising each asset
- **per_area**: indicates that the replacement costs will be provided per unit area for each asset

If the costs are to be specified `per_area` for any of the `costTypes`, the `area` element will also need to be defined in the conversions section. The `area` element defines a `type`, and a `unit` attribute.

The `unit` attribute of the `area` element is used for specifying the units for the area of an asset. The OpenQuake-engine itself is agnostic to the area units; the `unit` is thus a descriptive attribute which is used by the OpenQuake-engine to annotate the results of a risk assessment. This attribute can be set to any valid Unicode string.

The `type` attribute of the `area` element specifies whether the area will be provided as an aggregated value for an asset, or per building or unit comprising an asset. The valid values for the `type` attribute of the `area` element are the following:

- `aggregated`: indicates that the area will be provided as an aggregated value for each asset
- `per_asset`: indicates that the area will be provided per building or unit comprising each asset

The way the information about the characteristics of the assets in an exposure model are stored can vary strongly depending on how and why the data was compiled. As an example, if national census information is used to estimate the distribution of assets in a given region, it is likely that the number of buildings within a given geographical area will be used to define the dataset, and will be used for estimating the number of collapsed buildings for a scenario earthquake. On the other hand, if simplified methodologies based on proxy data such as population distribution are used to develop the exposure model, then it is likely that the built up area or economic cost of each building typology will be directly derived, and will be used for the estimation of economic losses.

Finally, let us look at the part of the file describing the set of assets in the portfolio to be used in seismic damage or risk calculations:

Each asset definition involves specifying a set of mandatory and optional attributes concerning the asset. The following set of attributes can be assigned to each asset based on the current schema for the exposure model:

- `id`: mandatory; a unique key used to identify the given asset, which is used by the OpenQuake-engine to relate each asset with its associated results.
- `taxonomy`: mandatory; specifies the building typology of the given asset. The taxonomy strings can be user-defined, or based on an existing classification scheme such as
- `number`: mandatory; the number of independent units or individual structures comprising a given asset
- `location`: mandatory; specifies the longitude (between -180° to 180°) and latitude (between -90° to 90°) of the given asset, both specified in decimal degrees¹.
- `area`: area of the asset, at a given location. As mentioned earlier, the area is a mandatory attribute only if any one of the costs for the asset is specified per unit area.

¹Within the OpenQuake-engine, longitude and latitude coordinates are internally rounded to a precision of 5 digits after the decimal point.

- **costs**: specifies a set of costs for the given asset. The replacement value for different cost types must be provided on separate lines within the **costs** element. As shown in the example above, each cost entry must define the type and the value. Currently supported valid options for the cost type are: **structural**, **nonstructural**, **contents**, and **business_interruption**.
- **occupancies**: mandatory only for probabilistic or scenario damage calculations. Each entry within this element specifies the number of occupants for the asset for a particular period of the day. As shown in the example above, each occupancy entry must define the **period** and the **occupants**. Currently supported valid options for the **period** are: **day**, **transit**, and **night**. Currently, the number of occupants for an asset can only be provided as an aggregated value for the asset.

For the purposes of performing a retrofitting benefit/cost analysis, it is also necessary to define the retrofitting cost (**reco**). The combination between the possible options in which these three attributes can be defined leads to four ways of storing the information about the assets. For each of these cases a brief explanation and example is provided in this section.

Example 1

This example illustrates an exposure model in which the aggregated cost (structural, non-structural, contents and business interruption) of the buildings of each taxonomy for a set of locations is directly provided. Thus, in order to indicate how the various costs will be defined, the following information needs to be stored in the exposure model file:

In this case, the cost type of each component as been defined as aggregated. Once the way in which each cost is going to be defined has been established, the values for each asset can be stored according to the following format:

Each asset is uniquely identified by its **id**, (e.g. loss exceedance curves). Then, a pair of coordinates (latitude and longitude) for a **location** where the asset is assumed to exist is defined. Each asset must be classified according to a **taxonomy**, so that the OpenQuake-engine is capable of employing the appropriate vulnerability function or fragility function in the risk calculations. Finally, the cost values of each type are stored within the **costs** attribute. In this example, the aggregated value for all units (within a given asset) at each location is provided directly, so there is no need to define other attributes such as **number** or **area**. This mode of representing an exposure model is probably the simplest one.

Example 2

In this example an exposure model containing the number of units (buildings) and the associated costs per unit of each building typology is presented.

For this case, the cost type has been set to `per_asset`. Then, the information from each asset can be stored following the format below:

In this example, the various costs for each asset is not provided directly, as happened in the previous example. In order to carry out the risk calculations in which the economic cost of each asset is required, the OpenQuake-engine multiplies, for each asset, the number of units (buildings) by the “per asset” replacement cost. Note that in this case, there is no need to specify the attribute `area`.

Example 3

This example is comprised of an exposure model containing the built up area of each building typology for a set of locations, and the associated costs are provided per unit area.

In order to compile an exposure model with this structure, it is required to set the cost type to `per_area`. In addition, it is also necessary to specify if the `area` that is being stored represents the aggregated area of number of units within an asset, or the average area of a single unit. In this particular case, the `area` that is being stored is the aggregated built up area per asset, and thus this attribute was set to `aggregated`.

Once again, the OpenQuake-engine needs to carry out some calculations in order to compute the different costs per asset. In this case, this value is computed by multiplying the aggregated built up area of each building typology by the associated cost per unit of area. Notice that in this case, there is no need to specify the attribute `number`.

Example 4

This example is comprised of an exposure model containing the number of buildings for each location, the average built up area per building unit and the associated costs per unit area.

Similarly to what was described in the previous example, the various costs type also need to be established as `per_area`, but the type of area is now defined as `per_asset`.

In this example, the OpenQuake-engine will make use of all the parameters to estimate the various costs of each asset, by multiplying the number of buildings by its average built up area, and then by the respective cost per unit of area.

Example 5

In this example, additional information will be included, which is required for other risk analysis besides loss estimation, such as the calculation of insured losses or benefit/cost analysis. For the former assessment, it is necessary to establish how the insured limit and deductible is going to be defined, according to the format below.

In this example, both the insurance limit and the deductible were defined as a fraction of the costs, by setting the attribute `isAbsolute` to `false`. Then, for each type of cost, the limit and deductible value can be stored for each asset.

On the other hand, a user could define one or both of these parameters as the absolute threshold, by setting the aforementioned attribute to `true`. This is shown in the example file below:

Moreover, in order to perform a benefit/cost assessment, it is also fundamental to indicate the retrofitting cost. This parameter is handled in the same manner as the structural cost, and it should be stored according to the following structure:

Despite the fact that for the demonstration of how the insurance parameters and retrofitting cost can be stored the per building type of cost structure described in Example 1 was used, it is important to mention that any of the other cost storing approaches can also be employed (Examples 2–4).

Example 6

The OpenQuake-engine is also capable of estimating human losses, based on a number of occupants within an asset, at a certain time of the day. This example describes how this parameter is defined for each asset. In addition, this example also serves the purpose of presenting an exposure model in which three cost types have been defined following different structures.

As previously mentioned, in this example only three costs are being stored, and each one follows a different approach. The `structural` cost is being defined as the aggregate replacement cost for all of the buildings comprising the asset (Example 1), the `nonstructural` value is defined as the replacement cost per unit area where the area is defined per building comprising the asset (Example 4), and the `contents` and `business_interruption` values are provided per building comprising the asset (Example 2). The occupants at different times of the day are also provided as aggregated values for all of the buildings comprising

the asset.

Scripts to convert an exposure model in CSV format or as Excel or ASCII files into NRML are also under development, and can be found at the OpenQuake platform at the following address: https://platform.openquake.org/risk_input_preparation_toolkit/.

7.2 Fragility models

This section describes the schema currently used to store fragility models, which are required for the Scenario Damage Calculator and the Classical Probabilistic Seismic Damage Calculator. A fragility model defines a set of fragility functions, describing the probability of exceeding a set of limit, or damage, states. These fragility function can be defined in two ways: discrete or continuous.

For discrete fragility functions, sets of probabilities of exceedance (one set per limit state) are defined for a list of intensity measure levels, as illustrated in Figure 7.1.

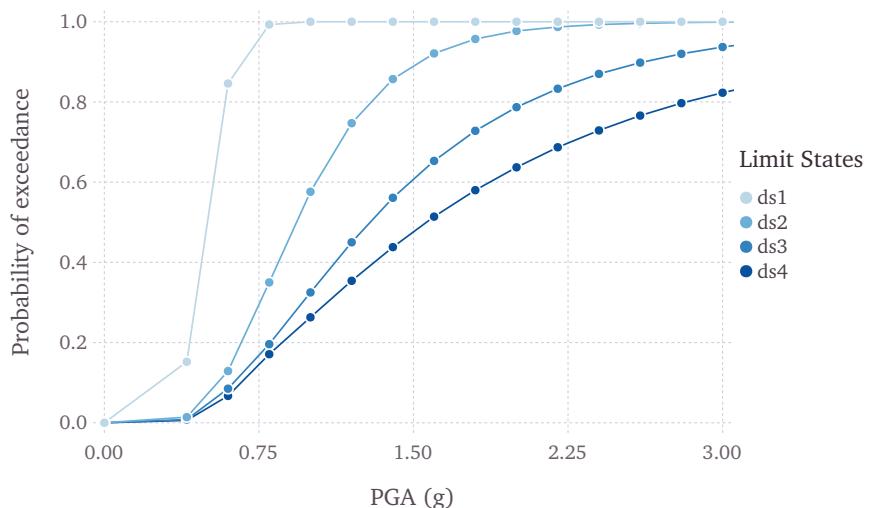


Figure 7.1 – Graphical representation of a discrete fragility model

The fragility functions can also be defined as continuous functions, through the use of cumulative lognormal distribution functions. In Figure 7.2, a continuous fragility model is presented.

An example fragility model comprising one discrete fragility function and one continuous fragility function is shown below:

The initial portion of the schema contains general information that describes some general aspects of the fragility model.

- **id:** a unique key used to identify the fragility model

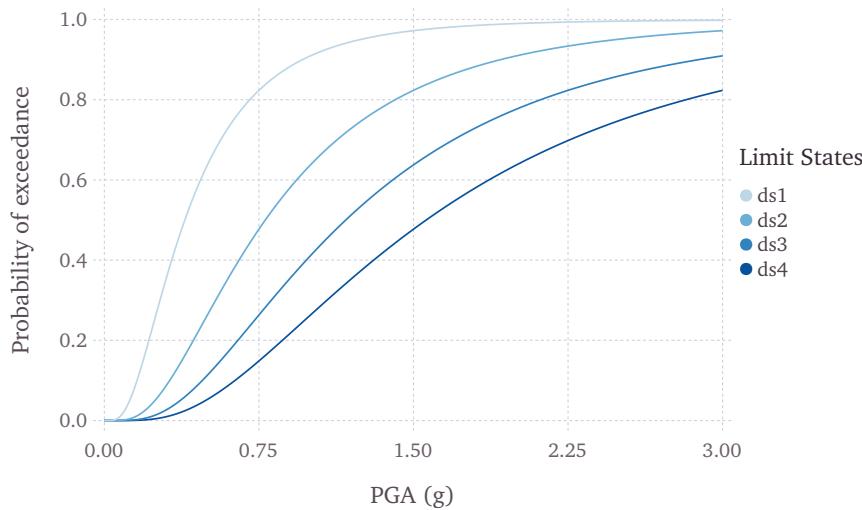


Figure 7.2 – Graphical representation of a continuous fragility model

- **assetCategory:** an optional string used to specify the type of assets for which fragility functions will be defined in this file (e.g: buildings, lifelines)
- **lossCategory:** valid strings for this attribute are “structural”, “nonstructural”, “contents”, and “business_interruption”
- **description:** a brief string with further information about the exposure model

The information in the metadata section is common to all of the functions in the fragility model and needs to be included at the beginning of every fragility model file. The parameters are described below:

- **description:** a brief string with further information about the fragility model, for example, which building typologies are covered or the source of the functions in the fragility model
- **limitStates:** this field is used to define the number and nomenclature of each limit state. Four limit states are employed in the example above, but it is possible to use any number of discrete states, as long as a fragility curve is always defined for each limit state. The limit states must be provided as a set of strings separated by whitespaces between each limit state. Please ensure that there is no whitespace within the name of any individual limit state.

In order to perform probabilistic or scenario damage calculations, it is necessary to define a fragility function for each building typology present in the exposure model. The fragility functions can be defined using either a discrete or a continuous format, and the fragility model file can include a mix of both types of fragility functions.

The following snippet from the above fragility model example file defines a discrete

fragility function:

The following attributes are needed to define a discrete fragility function:

- **id**: a unique key used to identify the taxonomy for which the function is being defined. This key is used to relate the fragility function with the relevant asset in the exposure model.
- **format**: for discrete fragility functions, this attribute should be set to “discrete”
- **imls**: this attribute specifies the list of intensity levels for which the limit state probabilities of exceedance will be defined. In addition, it is also necessary to define the intensity measure type (**imt**). Optionally, a **noDamageLimit** can be specified, which defines the intensity level below which the probability of exceedance for all limit states is taken to be zero.
- **poes**: this field is used to define the probabilities of exceedance (**poes**) for each limit state for this fragility function. It is also necessary to specify which limit state the exceedance probabilities are being defined for using the attribute **ls**. The probabilities of exceedance for each limit state must be provided on a separate line; and the number of exceedance probabilities for each limit state defined by the **poes** attribute must be equal to the number of intensity levels defined by the attribute **imls**. Finally, the number and names of the limit states in each fragility function must be equal to the number of limit states defined earlier in the metadata section of the fragility model using the attribute **limitStates**.

The following snippet from the above fragility model example file defines a continuous fragility function:

The following attributes are needed to define a continuous fragility function:

- **id**: a unique key used to identify the taxonomy for which the function is being defined. This key is used to relate the fragility function with the relevant asset in the exposure model.
- **format**: for continuous fragility functions, this attribute should be set to “continuous”
- **shape**: for continuous fragility functions using the lognormal cumulative distribution, this attribute should be set to “logncdf”. At present, only the lognormal cumulative distribution function can be used for representing continuous fragility functions.
- **imls**: this element specifies various aspects related to the intensity measure used by the fragility function. The range of intensity levels for which the continuous fragility functions are valid are specified using the attributes **minIML** and **maxIML**. In addition, it is also necessary to define the intensity measure type **imt**. Optionally, a **noDamageLimit** can be specified, which defines the intensity level below which the probability of exceedance for all limit states is taken to be zero.

- `params`: this field is used to define the parameters of the continuous curve for each limit state for this fragility function. For a lognormal cumulative distribution function, the two parameters required to specify the function are the mean and standard deviation of the intensity level. These parameters are defined for each limit state using the attributes `mean` and `stddev` respectively. The attribute `ls` specifies the limit state for which the parameters are being defined. The parameters for each limit state must be provided on a separate line. The number and names of the limit states in each fragility function must be equal to the number of limit states defined earlier in the metadata section of the fragility model using the attribute `limitStates`.

Several methodologies to derive fragility functions are currently being evaluated by Global Earthquake Model (GEM) and have been included as part of the Risk Modeller's Toolkit, the code for which can be found on a public repository at GitHub at the following address: <http://github.com/gemscienctools/rmtk>.

Scripts to convert fragility functions in CSV format or as Excel or ASCII files into NRML are also under development, and can be found at the OpenQuake platform at the following address: https://platform.openquake.org/risk_input_preparation_toolkit/.

7.3 Consequence models

Starting from OpenQuake-engine v1.7, the Scenario Damage calculator also accepts consequence models in addition to fragility models, in order to estimate consequences based on the calculated damage distribution. The user may provide one consequence model file corresponding to each loss type (amongst structural, nonstructural, contents, and business interruption) for which a fragility model file is provided. Whereas providing a fragility model file for at least one loss type is mandatory for running a Scenario Damage calculation, providing corresponding consequence model files is optional.

This section describes the schema currently used to store consequence models, which are optional inputs for the Scenario Damage Calculator. A consequence model defines a set of consequence functions, describing the distribution of the loss (or consequence) ratio conditional on a set of discrete limit (or damage) states. These consequence function can be currently defined in OpenQuake-engine by specifying the parameters of the continuous distribution of the loss ratio for each limit state specified in the fragility model for the corresponding loss type, for each taxonomy defined in the exposure model.

An example consequence model is shown below:

The initial portion of the schema contains general information that describes some general aspects of the consequence model. The information in this metadata section is common to all of the functions in the consequence model and needs to be included at the beginning of every consequence model file. The parameters are described below:

- **id:** a unique key used to identify the consequence model
- **assetCategory:** an optional string used to specify the type of assets for which fragility functions will be defined in this file (e.g: buildings, lifelines)
- **lossCategory:** valid strings for this attribute are “structural”, “nonstructural”, “contents”, and “business_interruption”
- **description:** a brief string with further information about the consequence model, for example, which building typologies are covered or the source of the functions in the consequence model
- **limitStates:** this field is used to define the number and nomenclature of each limit state. Four limit states are employed in the example above, but it is possible to use any number of discrete states, as long as a fragility curve is always defined for each limit state. The limit states must be provided as a set of strings separated by whitespaces between each limit state. Please ensure that there is no whitespace within the name of any individual limit state.

The following snippet from the above consequence model example file defines a consequence function using a lognormal distribution to model the uncertainty in the consequence ratio for each limit state:

The following attributes are needed to define a consequence function:

- **id:** a unique key used to identify the taxonomy for which the function is being defined. This key is used to relate the consequence function with the relevant asset in the exposure model.
- **dist:** for vulnerability function which use a continuous distribution to model the uncertainty in the conditional loss ratios, this attribute should be set to either “LN” if using the lognormal distribution, or to “BT” if using the Beta distribution ².
- **params:** this field is used to define the parameters of the continuous distribution used for modelling the uncertainty in the loss ratios for each limit state for this consequence function. For a lognormal distribution, the two parameters required to specify the function are the mean and standard deviation of the consequence ratio. These parameters are defined for each limit state using the attributes **mean** and **stddev** respectively. The attribute **ls** specifies the limit state for which the parameters are being defined. The parameters for each limit state must be provided on a separate line. The number and names of the limit states in each consequence function must be equal

²Note that in OpenQuake-engine v1.7, the uncertainty in the consequence ratios is ignored, and only the mean consequence ratios for the set of limit states is considered when computing the consequences from the damage distribution. Consideration of the uncertainty in the consequence ratios will be included in future releases of the OpenQuake-engine.

to the number of limit states defined in the corresponding fragility model using the attribute `limitStates`.

7.4 Vulnerability models

In this section, the NRML schema for the vulnerability model is described in detail. In order to do so, a graphical representation of a vulnerability model (mean loss ratio for a set of intensity measure levels) is illustrated in Figure 7.3.

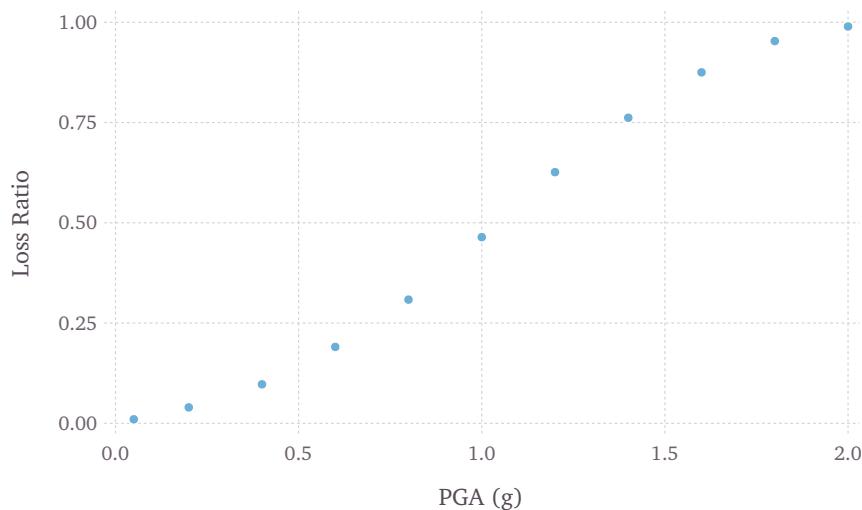


Figure 7.3 – Graphical representation of a vulnerability model.

Note that although the uncertainty for each loss ratio is not represented in Figure 7.3, it can be considered in the input NRML file, by means of a coefficient of variation per loss ratio and a probabilistic distribution, which can currently be set to lognormal (LN) or Beta (BT). An example of a vulnerability function that models the uncertainty in the loss ratio at different intensity levels using a lognormal distribution is illustrated below in Figure 7.4.

An example vulnerability model comprising three vulnerability functions is shown below. This vulnerability model contains one function that uses the lognormal distribution to represent the uncertainty in the loss ratio at different intensity levels, one function that uses the Beta distribution, and one function that is defined using a discrete probability mass distribution.

The initial portion of the schema contains general information that describes some general aspects of the vulnerability model. The information in this metadata section is common to all of the functions in the vulnerability model and needs to be included at the beginning of every vulnerability model file. The parameters are described below:

- `id`: a unique key used to identify the vulnerability model

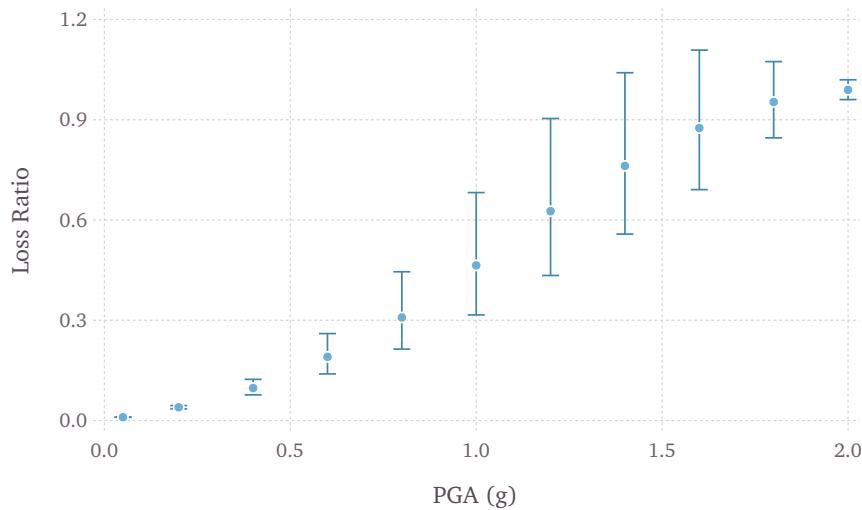


Figure 7.4 – Graphical representation of a vulnerability function that models the uncertainty in the loss ratio using a lognormal distribution. The mean loss ratios and coefficients of variation are illustrated for a set of intensity levels.

- **assetCategory:** an optional string used to specify the type of assets for which vulnerability functions will be defined in this file (e.g: buildings, lifelines)
- **lossCategory:** valid strings for this attribute are “structural”, “nonstructural”, “contents”, “business_interruption”, and “occupants”
- **description:** a brief string with further information about the vulnerability model, for example, which building typologies are covered or the source of the functions in the vulnerability model

In order to perform probabilistic or scenario risk calculations, it is necessary to define a vulnerability function for each building typology present in the exposure model. The vulnerability functions require the user to specify the distribution of the loss ratio for a set of intensity levels. The loss ratio distributions can be defined using either a discrete or a continuous format, and the vulnerability model file can include a mix of both types of vulnerability functions. It is also possible to define a vulnerability function using a set of deterministic loss ratios corresponding to a set of intensity levels (i.e., ignoring the uncertainty in the conditional loss ratios).

The following snippet from the above vulnerability model example file defines a vulnerability function modelling the uncertainty in the conditional loss ratios using a (continuous) lognormal distribution:

The following attributes are needed to define a vulnerability function which uses a

continuous distribution to model the uncertainty in the conditional loss ratios:

- **id**: a unique key used to identify the taxonomy for which the function is being defined. This key is used to relate the vulnerability function with the relevant asset in the exposure model.
- **dist**: for vulnerability function which use a continuous distribution to model the uncertainty in the conditional loss ratios, this attribute should be set to either “LN” if using the lognormal distribution, or to “BT” if using the Beta distribution.
- **imls**: this attribute specifies the list of intensity levels for which the parameters of the conditional loss ratio distributions will be defined. In addition, it is also necessary to define the intensity measure type (**imt**).
- **meanLRs**: this field is used to define the mean loss ratios for this vulnerability function for each of the intensity levels defined by the attribute **imls**. The number of mean loss ratios defined by the **meanLRs** attribute must be equal to the number of intensity levels defined by the attribute **imls**.
- **covLRs**: this field is used to define the coefficient of variation for the conditional distribution of the loss ratios for this vulnerability function for each of the intensity levels defined by the attribute **imls**. The number of coefficients of variation of loss ratios defined by the **covLRs** attribute must be equal to the number of intensity levels defined by the attribute **imls**. The uncertainty in the conditional loss ratios can be ignored by setting all of the **covLRs** for a given vulnerability function to zero.

Several methodologies to derive vulnerability functions are currently being evaluated by GEM and have been included as part of the Risk Modeller’s Toolkit, the code for which can be found on a public repository at GitHub at the following address <http://github.com/gemscienctools/rmtk>.

Scripts to convert vulnerability functions in CSV format or as Excel or ASCII files into NRML are also under development, and can be found at the OpenQuake platform at the following address: https://platform.openquake.org/risk_input_preparation_toolkit/.

[Scenario Damage Calculator](#)

[Scenario Risk Calculator](#)

[Classical Probabilistic Seismic Risk Calculator](#)

[Classical Probabilistic Seismic Damage Calculator](#)

[Event-Based Probabilistic Seismic Risk Calculator](#)

[Retrofit Benefit-Cost Ratio Calculator](#)

[Running the Risk Calculators](#)

8. Using the Risk Module

This Chapter summarises the structure of the information necessary to define the different input data to be used with the OpenQuake-engine risk calculators. Input data for scenario-based and probabilistic seismic damage and risk analysis using OpenQuake-engine are organised into:

- An exposure model file in the NRML format, as described in Section 7.1.
- A file describing the vulnerability model (Section 7.4) for loss calculations, or a file describing the fragility model (Section 7.2) for damage calculations. Optionally, a file describing the consequence model (Section 7.3) can also be provided in order to calculate losses from the estimated damage distributions.
- A general calculation configuration file.
- Hazard inputs. These include hazard curves for the classical probabilistic damage and risk calculators, ground motion fields for the scenario damage and risk calculators, or stochastic event sets for the probabilistic event based calculators. As of OpenQuake-engine v1.7, in general, there are five different ways in which hazard calculation parameters or results can be provided to the OpenQuake-engine in order to run the subsequent risk calculations:
 - Use a single configuration file for running the hazard and risk calculations sequentially
 - Use separate configuration files for running the hazard and risk calculations sequentially
 - Use a configuration file for the risk calculation along with all hazard outputs from a previously completed, compatible OpenQuake-engine hazard calculation
 - Use a configuration file for the risk calculation along with a specific hazard output from a previously completed, compatible OpenQuake-engine hazard calculation
 - Use a configuration file for the risk calculation along with hazard input files in

the OpenQuake NRML format

The file formats for exposure, fragility, consequence, and vulnerability models have been described earlier in Chapter 6. The configuration file is the primary file that provides the OpenQuake-engine information regarding both the definition of the input models (e.g. exposure, site parameters, fragility, consequence, or vulnerability models) as well as the parameters governing the risk calculation.

Information regarding the configuration file for running hazard calculations using OpenQuake-engine can be found in Section 3.4. Some initial parameters of the configuration file common to all of the risk calculators are presented below. The remaining parameters that are specific to each risk calculator are discussed in subsequent sections.

- **description:** a parameter that can be used to include some information about the type of calculations that are going to be performed.
- **calculation_mode:** this parameter specifies the type of calculation to be run. Valid options for the `calculation_mode` for the risk calculators are: `scenario_damage`, `scenario_risk`, `classical_damage`, `classical_risk`, `event_based_risk`, and `classical_bcr`.
- **exposure_file:** this parameter is used to specify the path to the exposure model file.

Depending on the type of risk calculation, other parameters besides the aforementioned ones may need to be provided. We illustrate in the following sections different examples of the configuration file for the different risk calculators.

8.1 Scenario Damage Calculator

For this calculator, the parameter `calculation_mode` should be set to `scenario_damage`.

Example 1

This example illustrates a scenario damage calculation which uses a single configuration file to first compute the ground motion fields for the given rupture model and then calculate damage distribution statistics based on the ground motion fields. A minimal job configuration file required for running a scenario damage calculation is shown below:

The general parameters `description` and `calculation_mode`, and `exposure_file` have already been described earlier. The other parameters seen in the above example configuration file are described below:

- **rupture_model_file:** a parameter used to define the path to the earthquake rupture model file describing the scenario event.

- `rupture_mesh_spacing`: a parameter used to specify the mesh size (in km) used by OpenQuake-engine to discretize the rupture. Note that the smaller the mesh spacing, the greater will be (1) the precision in the calculation and (2) the computational demand.
- `structural_fragility_file`: a parameter used to define the path to the structural fragility model file.

In this case, the ground motion fields will be computed at each of the locations of the assets in the exposure model. The above calculation can be run using the command line:

```
user@ubuntu:~$ oq-engine --run job.ini
```

After the calculation is completed, a message similar to the following will be displayed:

```
Calculation 2680 completed in 13 seconds. Results:  
id | output_type | name  
5069 | datastore | dmg_by_asset_and_collapse_map  
5070 | datastore | dmg_by_taxon  
5071 | datastore | dmg_total
```

Note that one or more of the following parameters can be used in the same job configuration file to provide the corresponding fragility model files:

- `structural_fragility_file`: a parameter used to define the path to a structural fragility model file
- `nonstructural_fragility_file`: a parameter used to define the path to a non-structural fragility model file
- `contents_fragility_file`: a parameter used to define the path to a contents fragility model file
- `business_interruption_fragility_file`: a parameter used to define the path to a business interruption fragility model file

It is important that the `lossCategory` parameter in the metadata section for each provided fragility model file (“structural”, “nonstructural”, “contents”, or “business_interruption”) should match the loss type defined in the configuration file by the relevant keyword above.

Example 2

This example illustrates a scenario damage calculation which uses separate configuration files for the hazard and risk parts of a scenario damage assessment. The first configuration file contains input models and parameters required for the computation of the ground motion fields due to a given rupture. The second configuration file contains input models and parameters required for the calculation of the damage distribution for a portfolio of assets due to the ground motion fields.

Now, the above calculations described by the two configuration files “job_hazard.ini” and “job_damage.ini” can be run sequentially using the command line as follows:

```
user@ubuntu:~$ oq-engine --run job_hazard.ini,job_damage.ini
```

The hazard and risk calculations can also be run separately. In that case, the calculation id for the hazard calculation or the output id for the specific ground motion fields output generated by the hazard calculation should be provided to the OpenQuake-engine while running the risk calculation using the options `--hazard-calculation-id` (or `--hc`) and `--hazard-output-id` (or `--ho`) respectively. This is shown below:

```
user@ubuntu:~$ oq-engine --run job_hazard.ini
```

In the configuration file illustrated above, the list of sites at which the ground motion values will be computed is provided in a CSV file, specified using the `sites_csv` parameter. The sites used for the hazard calculation need not be the same as the locations of the assets in the exposure model used for the following risk calculation. In such cases, it is recommended to set a reasonable search radius (in km) using the `asset_hazard_distance` parameter for the OpenQuake-engine to look for available hazard values, as shown in the `job_damage.ini` example file above.

- `asset_hazard_distance`: this parameter indicates the maximum allowable distance between an asset and the closest hazard input. Hazard inputs can include hazard curves or ground motion intensity values. If no hazard input is found within this distance, the asset is skipped and a message is provided mentioning the id of the asset that is affected by this issue. If this parameter is not provided, the OpenQuake-engine assumes the maximum allowable distance as 5 km.

After the hazard calculation is completed, a message similar to the one below will be displayed in the terminal:

```
Calculation 2681 completed in 4 seconds. Results:  
id | output_type | name  
5072 | datastore | gmfs
```

In the example above, the calculation id of the hazard calculation is 2681. There is only one output from this calculation, i.e., the ground motion fields. The output id for the gmfs generated by the above calculation is 5027.

The risk calculation for computing the damage distribution statistics for the portfolio of assets can now be run using either:

```
user@ubuntu:~$ oq-engine --run job_damage.ini --hc 2681
```

or

```
user@ubuntu:~$ oq-engine --run job_damage.ini --ho 5072
```

After the calculation is completed, a message similar to the one listed above in Example 1 will be displayed.

In order to retrieve the calculation id of a previously run hazard calculation, the option `--list-hazard-calculations` (or `--lhc`) can be used to display a list of all previously run hazard calculations:

job_id	status	last_update	description
2609	successful	2015-12-01 14:14:14	Mid Nepal earthquake
...			
2681	successful	2015-12-12 10:00:00	Scenario hazard example

The option `--list-outputs` (or `--lo`) can be used to display a list of all outputs generated during a particular calculation. For instance,

```
user@ubuntu:~$ oq-engine --lo 2681
```

will produce the following display:

id	output_type	name
5072	datastore	gmfs

Example 3

This example illustrates a scenario damage calculation which uses a file listing a precomputed set of ground motion fields. These ground motion fields can be computed using the OpenQuake-engine or some other software. The ground motion fields can be provided in the NRML format as presented in Section 4.3.3, or as a CSV file. The damage distribution is computed based on the provided ground motion fields.

- `gmfs_file`: a parameter used to define the path to the ground motion fields file in the NRML format

The above calculation can be run using the command line:

```
user@ubuntu:~$ oq-engine --run job.ini
```

8.2 Scenario Risk Calculator

In order to run this calculator, the parameter `calculation_mode` needs to be set to `scenario_risk`. Most of the job configuration parameters required for running a scenario risk calculation are the same as those described in the previous section for the scenario damage calculator. The remaining parameters specific to the scenario risk calculator are illustrated below.

```
...
structural_vulnerability_file = structural_vulnerability_model.xml
nonstructural_vulnerability_file = nonstructural_vulnerability_model.xml
contents_vulnerability_file = contents_vulnerability_model.xml
business_interruption_vulnerability_file = downtime_vulnerability_model.xml
occupants_vulnerability_file = occupants_vulnerability_model.xml

asset_correlation = 0.7
master_seed = 3
insured_losses = true
```

- `structural_vulnerability_file`: this parameter is used to specify the path to the structural vulnerability model file
- `nonstructural_vulnerability_file`: this parameter is used to specify the path to the nonstructuralvulnerability model file
- `contents_vulnerability_file` : this parameter is used to specify the path to the contents vulnerability model file
- `business_interruption_vulnerability_file` : this parameter is used to specify the path to the business interruption vulnerability model file
- `occupants_vulnerability_file`: this parameter is used to specify the path to the occupants vulnerability model file
- `asset_correlation` if the uncertainty in the loss ratios has been defined within the vulnerability model, users can specify a coefficient of correlation that will be used in the Monte Carlo sampling process of the loss ratios, between the assets that share the same taxonomy. If the `asset_correlation` is set to one, the loss ratio residuals will be perfectly correlated. On the other hand, if this parameter is set to zero, the loss ratios will be sampled independently. Any value between zero and one will lead to increasing levels of correlation. If this parameter is not defined, the OpenQuake-engine will assume zero correlation in the vulnerability.
- `master_seed`: this parameter is used to control the random number generator in the loss ratio sampling process. If the same `master_seed` is defined at each calculation run, the same random loss ratios will be generated, thus allowing reproducibility of the results.

- `insured_losses`: this parameter specifies whether insured losses should be calculated (true) or not (false).

8.3 Classical Probabilistic Seismic Risk Calculator

In order to run this calculator, the parameter `calculation_mode` needs to be set to `classical_risk`. With this calculator it is also possible to extract loss maps, so the parameter `conditional_loss_poes` needs to be defined as explained in the previous sub-section. The remaining parameters are illustrated below:

```
...
structural_vulnerability_file = struct_vul_model.xml
nonstructural_vulnerability_file = nonstruct_vul_model.xml
contents_vulnerability_file = cont_vul_model.xml
business_interruption_vulnerability_file = bus_int_vul_model.xml
occupants_vulnerability_file = occ_vul_model.xml

lrem_steps_per_interval = 2
conditional_loss_poes = 0.01, 0.05, 0.10
```

- `lrem_steps_per_interval`: this parameter controls the number of intermediate values between consecutive loss ratios (as defined in the vulnerability model) that are considered in the risk calculations. A larger number of loss ratios than those defined in each vulnerability function should be considered, in order to better account for the uncertainty in the loss ratio distribution. If this parameter is not defined in the configuration file, the OpenQuake-engine assumes the `lrem_steps_per_interval` to be equal to 5. More details are provided in the OpenQuake-engine Book (Risk).

8.4 Classical Probabilistic Seismic Damage Calculator

In order to run this calculator, the parameter `calculation_mode` needs to be set to `classical_damage`. Similar to the Scenario Damage calculator, there is only one parameter specific to this calculator, which is the fragility model file path, as presented below.

```
...
fragility_file = fragility_model.xml
```

8.5 Event-Based Probabilistic Seismic Risk Calculator

The parameter `calculation_mode` needs to be set to `event_based_risk` in order to use this calculator. Similarly to that described for the Scenario Risk Calculator, a Monte Carlo

sampling process is also employed within this module to take into account the loss ratio uncertainty. Hence, the parameters `asset_correlation` and `master_seed` need to be defined as previously described. This calculator is also capable of estimating insured losses and therefore, the `insured_losses` attribute can be specified as well. The parameter “`risk_investigation_time`” specifies the time period for which the event loss tables and loss exceedance curves will be calculated. If this parameter is not provided in the risk job configuration file, the time period used is the same as that specified in the hazard calculation.

The remaining parameters are presented below.

```

...
structural_vulnerability_file = struct_vul_model.xml
nonstructural_vulnerability_file = nonstruct_vul_model.xml
contents_vulnerability_file = cont_vul_model.xml
business_interruption_vulnerability_file = bus_int_vul_model.xml
occupants_vulnerability_file = occ_vul_model.xml

asset_correlation = 0.7
master_seed = 3
insured_losses = true

loss_curve_resolution = 20
conditional_loss_poes = 0.01, 0.05, 0.10

```

- `loss_curve_resolution`: since this calculator uses an event-based approach, a large number of levels of loss (and associated probabilities of exceedance) is computed (one per event) for each asset. The oq-risklib will use this large set of results to extrapolate a loss curve, whose number of points are controlled by this parameter. By default, the OpenQuake-engine assumes the `loss_curve_resolution` equal to 20.
- `risk_investigation_time`: this parameter specifies the time period for which the event loss tables and loss exceedance curves will be calculated. If this parameter is not provided in the risk job configuration file, the time period used is the same as that specified in the hazard calculation.
- `conditional_loss_poes`: this parameter is used to define the probabilities of exceedance at which loss maps are to be produced.

8.6 Retrofit Benefit-Cost Ratio Calculator

As previously explained, this calculator uses loss exceedance curves which are calculated using the Classical PSHA-based Risk calculator. The `calculation_mode` should be set to `classical_bcr` and the calculator-specific part of the configuration file should be defined as presented below.

```
...  
structural_vulnerability_file = structural_vulnerability_model.xml  
vulnerability_retrofitted_file = retrofitted_vulnerability_model.xml  
  
lrem_steps_per_interval = 2  
  
interest_rate = 0.05  
asset_life_expectancy = 50
```

- `vulnerability_retrofitted_file`: this parameter is used to specify the path to the vulnerability model file containing the vulnerability functions for the retrofitted assets;
- `interest_rate`: this parameter represents the interest rate and it serves the purposes of taking into account the variation of building value throughout time;
- `asset_life_expectancy`: this variable defines the life expectancy, or design life, of the assets.

8.7 Running the Risk Calculators

Using the command line interface, risk calculations can be launched and the resulting outputs can be extracted. This section describes all the currently implemented commands and presents examples for each of the calculators.

This is the parameter that will be used when launching the risk calculations to indicate which hazard input should be employed. In the case of the scenario-based calculators, there is only a single hazard input (one or a set of ground motion fields). For the remaining calculators, where probabilistic seismic hazard is used, it is possible to have multiple hazard inputs due to the employment of logic trees, as described in Section 3.1.0.1. In the following illustration, a set of hazard results produced using the Classical PSHA calculator is presented.

```
Calculation 4 results:  
id | output_type | name  
32 | hazard_curve | hc-rlz-32-PGA  
33 | hazard_curve | hc-rlz-33-PGA  
34 | hazard_curve | hc-rlz-34-PGA  
35 | hazard_curve | hc-rlz-35-PGA  
36 | hazard_curve | mean curve for PGA  
37 | hazard_curve | quantile curve (poe>= 0.15) for imt PGA  
38 | hazard_curve | quantile curve (poe>= 0.85) for imt PGA
```

In this case, since the logic tree had four branches, four sets of hazard curves were produced, each one with its own id. In addition, mean and quantile hazard curves were also produced. A user may choose to run risk calculations using results from one of the branches or mean/quantile curves. To do so, the id of the respective hazard result should be employed when launching the risk calculations, as depicted below.

```
user@ubuntu:~$ oq-engine --run job_risk.ini --hazard-output-id <hazard_output_id>
```

or simply:

```
user@ubuntu:~$ oq-engine --run job_risk.ini --ho <hazard_output_id>
```

On the other hand, a user might also want to run the risk calculations considering all the hazard results from a certain calculation run. In this case, rather than providing the `hazard-output-id`, users need to provide the id of the hazard calculation as follows.

```
user@ubuntu:~$ oq-engine --run job_risk.ini --hazard-calculation-id <hazard_calculation_id>
```

or simply:

```
user@ubuntu:~$ oq-engine --run job_risk.ini --hc <hazard_calculation_id>
```

Exporting risk results

Description of the outputs

- Loss statistics
- Loss maps
- Damage distribution
- Collapse maps
- Loss exceedance curves
- Retrofitting Benefit/cost ratio maps
- Loss disaggregation
- Event loss tables

9. Risk Calculations and Results

9.1 Exporting risk results

To obtain a list of all the risk calculations, the following command can be employed.

```
user@ubuntu:~$ oq-engine --list-risk-calculations
```

or simply:

```
user@ubuntu:~$ oq-engine --lrc
```

Which will display a list of risk calculations as presented below.

calc_id	num_jobs	latest_job_status	last_update	description
1	1	successful	2013-04-02 08:50:30	Scenario Damage
2	1	failed	2013-04-03 09:56:17	Scenario Risk
3	1	successful	2013-04-04 10:45:32	Scenario Risk
4	4	successful	2013-04-04 10:48:33	Classical PSHA Risk

Then, in order to display a list of the risk outputs from a given job, the following command can be used:

```
user@ubuntu:~$ oq-engine --list-outputs <risk_calculation_id>
```

or simply:

```
user@ubuntu:~$ oq-engine --lo <risk_calculation_id>
```

which will display a list of outputs for the calculation requested, as presented below:

```
Calculation 4 results:
id | output_type | name
29 | loss_curve | loss curves. type=structural, hazard=32
30 | loss_map | loss maps. type=structural poe=0.1, hazard=32
```

Then, in order to export all of the risk calculation outputs in the appropriate xml format, the following command can be used.

```
user@ubuntu:~$ oq-engine --export-outputs <risk_calculation_id> <output_directory>
```

or simply:

```
user@ubuntu:~$ oq-engine --eos <risk_calculation_id> <output_directory>
```

If, instead of exporting all of the outputs from a particular calculation, only particular output files need to be exported, this can be achieved by using the --export-output option and providing the id of the required output:

```
user@ubuntu:~$ oq-engine --export-output <risk_output_id> <output_directory>
```

or simply:

```
user@ubuntu:~$ oq-engine --eo <risk_output_id> <output_directory>
```

9.2 Description of the outputs

This section describes how the different risk outputs are being stored using the Natural Hazards risk Markup Language (NRML). For each output, the various attributes are discussed, and example schema is provided.

9.2.1 Loss statistics

This output is produced by the Scenario Risk calculator and is comprised by a mean total loss and associated standard deviation. These results are stored in a comma separate value (.csv) file as follows:

```
Mean,Standard Deviation
8717775315.66,2047771108.36
```

9.2.2 Loss maps

A loss map contains the spatial distribution of the losses throughout the region of interest. This result can be produced by the Scenario Risk calculator (representing the losses from a single event), or from the Probabilistic Event-based Risk or Classical PSHA-based Risk calculators (representing the expected losses from probabilistic seismic hazard). In the former case, the loss map is comprised of a mean loss and respective standard deviation for each asset, whilst for the latter, a single value is provided, representing the expected loss for a given return period (or probability of exceedance for a certain time span, or investigation interval). In the following example, a loss map due to a single earthquake is presented.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
  <lossMap lossCategory="buildings" unit="EUR">
    <node>
      <gml:Point>
        <gml:pos>83.31 29.46</gml:pos>
      </gml:Point>
      <loss assetRef="a1" mean="53.3" stdDev="109.25"/>
      <loss assetRef="a2" mean="386.0" stdDev="695.7"/>
      <loss assetRef="a3" mean="303.1" stdDev="447.4"/>
      <loss assetRef="a4" mean="298.9" stdDev="453.7"/>
    </node>
    ...
    <node>
      <gml:Point>
        <gml:pos>83.33 28.71</gml:pos>
      </gml:Point>
      <loss assetRef="a997" mean="277.3" stdDev="100.8"/>
      <loss assetRef="a998" mean="219.6" stdDev="123.5"/>
      <loss assetRef="a999" mean="576.3" stdDev="210.9"/>
    </node>
  </lossMap>
</nrmr>
```

- **lossCategory:** the type of losses that are being stored. This parameter is taken from the vulnerability model that was used in the loss calculations (e.g. fatalities, economic loss);
- **unit:** this attribute is used to define the units in which the losses are being measured (e.g. EUR);

- **node:** each loss map is comprised by various nodes, each node possibly containing a number of assets. The location of the node is defined by a latitude and longitude in decimal degrees within the field `gml:Point`. The mean loss (`mean`) and associated standard deviation (`stdDev`) for each asset (identified by the parameter `assetRef`) is stored in the `loss` field.

For the probabilistic loss maps (expected losses for a given return period), a set of additional parameters need to be considered as depicted in the following example.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
<lossMap investigationTime="50" poE="0.1" sourceModelTreePath="b1"
           gsimTreePath="b1" lossCategory="buildings" unit="EUR">
  <node>
    <gml:Point>
      <gml:pos>83.31 29.46</gml:pos>
    </gml:Point>
    <loss assetRef="a1" value="696.1"/>
    <loss assetRef="a2" value="4201.4"/>
    <loss assetRef="a3" value="2666.0"/>
    <loss assetRef="a4" value="1291.8"/>
  </node>
  ...
  <node>
    <gml:Point>
      <gml:pos>83.33 28.71</gml:pos>
    </gml:Point>
    <loss assetRef="a997" value="4077.3"/>
    <loss assetRef="a998" value="2466.4"/>
    <loss assetRef="a999" value="4434.5"/>
  </node>
</lossMap>
</nrmr>
```

- `investigationTime`: time span used to compute the probability of exceedance;
- `poE`: parameter specifying the probability of exceedance (e.g. 0.1);
- `sourceModelTreePath`: this is a parameter indicating the path used to create the seismic source model;
- `gsimTreePath`: this parameter designates the ground motion model;

- **node:** this attribute follows an identical structure as seen in the previous example, but only a single loss (**value**) is provided per asset.

9.2.3 Damage distribution

The damage distribution is part of the outputs from the Scenario Damage calculator, and can be provided in three ways: per asset, per taxonomy or the total damage distribution. In the following illustration, an example of the NRML schema for the damage distribution per asset is presented:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
  <dmgDistPerAsset>
    <damageStates>
      no_damage
      slight
      moderate
      complete
    </damageStates>
    <DDNode>
      <gml:Point>
        <gml:pos>83.31 29.46</gml:pos>
      </gml:Point>
      <asset assetRef="a1">
        <damage ds="no_damage" mean="486.6" stddev="130.1"/>
        <damage ds="slight" mean="118.8" stddev="9.9"/>
        <damage ds="moderate" mean="130.3" stddev="20.3"/>
        <damage ds="complete" mean="186.5" stddev="90.8"/>
      </asset>
      <asset assetRef="2">
        <damage ds="no_damage" mean="877.08" stddev="257.9"/>
        <damage ds="slight" mean="171.3" stddev="13.2"/>
        <damage ds="moderate" mean="161.5" stddev="014.5"/>
        <damage ds="complete" mean="563.8" stddev="223.6"/>
      </asset>
    </DDNode>
    ...
    <DDNode>
      <gml:Point>
        <gml:pos>83.91 28.19</gml:pos>
```

```

</gml:Point>
<asset assetRef="999">
    <damage ds="no_damage" mean="21.5" stddev="16.6"/>
    <damage ds="slight" mean="15.5" stddev="8.7"/>
    <damage ds="moderate" mean="39.1" stddev="17.3"/>
    <damage ds="complete" mean="493.5" stddev="53.1"/>
</asset>
</DDNode>
</dmgDistPerAsset>
</nrml>

```

- **damageStates**: this field serves the purposes of storing the set of damage states, as defined in the fragility model employed in the calculations;
- **DDNode**: this attribute is used to store the damage distribution of a number of assets, at a given location (defined within the attribute gml:Point). For each asset, the mean number of buildings (mean) and associated standard deviation (stddev) in each damage state is defined.

The Scenario Damage calculator can also estimate the total number of buildings with a certain taxonomy, in each damage state. This distribution of damage per building taxonomy is depicted in the following example.

```

<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
    <dmgDistPerAsset>
        <damageStates>
            no_damage
            slight
            moderate
            complete
        </damageStates>
        <DDNode>
            <taxonomy>W</taxonomy>
            <damage ds="no_damage" mean="456450.2" stddev="26376.62"/>
            <damage ds="slight" mean="88102.3" stddev="3283.9"/>
            <damage ds="moderate" mean="103564.6" stddev="3487.1"/>
            <damage ds="complete" mean="275891.1" stddev="26676.8"/>
        </DDNode>
        ...
        <DDNode>

```

```

<taxonomy>RC</taxonomy>
<damage ds="no_damage" mean="4484.2" stddev="460.9"/>
<damage ds="slight" mean="932.4" stddev="106.7"/>
<damage ds="moderate" mean="1691.7" stddev="177.9"/>
<damage ds="complete" mean="7659.5" stddev="799.3"/>
</DDNode>
</dmgDistPerAsset>
</nrml>

```

In the damage distribution per taxonomy, each DDNode contains the statistics of the number of buildings in each damage state, belonging to a given building class as specified in the taxonomy attribute. Finally, a total damage distribution can also be calculated, which contains the mean and standard deviation of the total number of buildings in each damage state, as illustrated below.

```

<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
<totalDmgDist>
  <damageStates>
    no_damage
    slight
    moderate
    complete
  </damageStates>
  <damage ds="no_damage" mean="456450.2" stddev="26376.62"/>
  <damage ds="slight" mean="88102.3" stddev="3283.9"/>
  <damage ds="moderate" mean="103564.6" stddev="3487.1"/>
  <damage ds="complete" mean="275891.1" stddev="26676.8"/>
</totalDmgDist>
</nrml>

```

Similarly, the Classical PSHA-based Damage calculator provides the expected damage distribution per asset as a csv file, as illustrated below:

asset_ref	no_damage	slight	moderate	extreme	complete
a1	4.4360E-06	6.3482E-03	3.4851E-01	4.7628E-01	1.6884E-01
a2	1.0391E-05	9.1856E-03	3.7883E-01	4.6140E-01	1.5056E-01
.					
.					

```
.
a998, 6.9569E-02, 6.4106E+00, 7.4108E+01, 5.7563E+01, 1.7848E+01
a999, 1.2657E-01, 8.1294E+00, 7.6249E+01, 5.4701E+01, 1.6792E+01
```

9.2.4 Collapse maps

Collapse maps are part of the Scenario Damage calculator outputs. These results provide the spatial distribution of the number of the collapsed buildings throughout the area of interest. An example of the schema is presented below.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrmr/0.5">

<collapseMap>
  <CMNode>
    <gml:Point>
      <gml:pos>83.31 29.46</gml:pos>
    </gml:Point>
    <cf assetRef="a1" mean="227.1" stdDev="95.8"/>
    <cf assetRef="a2" mean="703.2" stdDev="240.2"/>
    <cf assetRef="a3" mean="199.5" stdDev="63.3"/>
    <cf assetRef="a4" mean="357.8" stdDev="136.1"/>
  </CMNode>
  ...
  <CMNode>
    <gml:Point>
      <gml:pos>83.33 28.71</gml:pos>
    </gml:Point>
    <cf assetRef="a997" mean="239.4" stdDev="102.0"/>
    <cf assetRef="a998" mean="733.0" stdDev="253.2"/>
    <cf assetRef="a999" mean="207.4" stdDev="66.5"/>
  </CMNode>
</collapseMap>
</nrmr>
```

This schema follows the same structure of the loss maps presented previously. Thus, the results for a number of assets at a given location are stored within the field CMNode. This field is associated with a location (defined within the gml:Point attribute) and it contains the mean number of collapses (mean) and respective standard deviation (stdDev) for each asset (identified by the parameter assetRef).

9.2.5 Loss exceedance curves

Loss exceedance curves can be calculated using the Classical PSHA-based Risk or Probabilistic Event-based Risk calculators, and they provide the probability of exceeding a set of levels of loss, within a given time span (or investigation interval). Similarly to what has been described for the probabilistic loss maps, also here it is necessary to define the parameters `investigationTime`, `sourceModelTreePath`, `gsimTreePath` and `unit`. Then, the set of loss exceedance curves are stored as presented in the following example.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
<lossCurves investigationTime="50" sourceModelTreePath="b1"
  gsimTreePath="b1" unit="EUR">
  <lossCurve assetRef="a1">
    <gml:Point>
      <gml:pos>83.31 29.46</gml:pos>
    </gml:Point>
    <poE> 0.970 0.297 0.137 0.019 0.005 0.001</poE>
    <losses> 235 477 989 4102 7444 15631</losses>
    <lossRatios> 0.02 0.03 0.06 0.26 0.48 1.0</lossRatios>
  </lossCurve>
  ...
  <lossCurve assetRef="a999">
    <gml:Point>
      <gml:pos>83.33 28.71</gml:pos>
    </gml:Point>
    <poE> 0.99 0.714 0.112 0.020 0.004 0.001</poE>
    <losses>58 402 819 3664 8001 13540</losses>
    <lossRatios> 0.02 0.04 0.07 0.32 0.59 1.0</lossRatios>
  </lossCurve>
</lossCurves>
</nrmr>
```

Each `lossCurve` is associated with a location (defined within the `gml:Point` attribute) and a reference to the asset (`assetRef`) whose loss is being represented. Then, three lists of values are stored: the probabilities of exceedance (`poE`), levels of absolute loss (`losses`) and percentages of loss (`lossRatios`).

9.2.6 Retrofitting Benefit/cost ratio maps

Ratio maps from the Retrofitting Benefit/Cost Ratio calculator require loss exceedance curves, which can be calculated using the Classical PSHA-based Risk calculator. For this reason, the

parameters `sourceModelTreePath` and `gsimTreePath` are also included in this NRML schema, so the whole calculation process can be traced back. The results for each asset are stored as depicted below:

```

<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
  <bcrMap interestRate="0.05" assetLifeExpectancy="50.0">
    sourceModelTreePath="b1" gsimTreePath="b1" unit="EUR">
    <node>
      <gml:Point>
        <gml:pos>83.31 29.46</gml:pos>
      </gml:Point>
      <bcr assetRef="a1" ratio="1.96" aalOrig="1466.9"
            aalRetr="253.0"/>
      <bcr assetRef="a2" ratio="2.33" aalOrig="937.9"
            aalRetr="215.7"/>
      <bcr assetRef="a3" ratio="1.32" aalOrig="492.0"
            aalRetr="83.7"/>
      <bcr assetRef="a4" ratio="0.76" aalOrig="294.1"
            aalRetr="57.9"/>
    </node>
    ...
    <node>
      <gml:Point>
        <gml:pos>83.33 28.71</gml:pos>
      </gml:Point>
      <bcr assetRef="a997" ratio="0.84" aalOrig="18323.1"
            aalRetr="7340.7"/>
      <bcr assetRef="a998" ratio="1.36" aalOrig="152027.6"
            aalRetr="29123.5"/>
      <bcr assetRef="a999" ratio="0.83" aalOrig="60727.3"
            aalRetr="12676.1"/>
    </node>
  </bcrMap>
</nrml>
```

- `interestRate`: this parameter represents the inflation rate of the economic value of the assets;
- `assetLifeExpectancy`: a parameter specifying the life expectancy (or design life)

- of the assets considered for the calculations;
- **node:** this schema follows the same node structure already presented for the loss maps, however, instead of losses for each asset, the benefit/cost ratio (**ratio**), the average annual loss considering the original vulnerability (**aalOrig**) and the average annual loss for the retrofitted (**aalRetr**) configuration of the assets are provided.

9.2.7 Loss disaggregation

Currently the loss disaggregation can only be performed using the Probabilistic Event-based Risk calculator. Thus, the parameters `sourceModelTreePath` and `gsimTreePath` have been included in the NRML schema. Additional information can also be comprised such as the `investigationTime`, `lossCategory` and `unit` of the losses. The OpenQuake-engine can perform loss disaggregation in terms of magnitude/distance or latitude/longitude. An example of the former type of output is presented below.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.5">
  <lossFraction investigationTime ="50.00" sourceModelTreePath="b1"
                 gsimTreePath="b1" lossCategory="buildings" unit="EUR"
                 variable="magnitude_distance">
    <map>
      <node lon="85.07916667" lat="27.4625">
        <bin value="5.00,5.50|220.00,240.0000"
             absoluteLoss="150000" fraction="0.75"/>
        <bin value="6.50,7.00|480.00,500.0000"
             absoluteLoss="50000" fraction="0.25"/>
      </node>
      ...
      <node lon="85.67" lat="27.58">
        <bin value="5.00,5.50|220.00,240.00"
             absoluteLoss="100000" fraction="0.50"/>
        <bin value="6.50,7.00|480.00,500.00"
             absoluteLoss="50000" fraction="0.25"/>
        <bin value="7.00,7.50|350.00,370.00"
             absoluteLoss="50000" fraction="0.25"/>
      </node>
    </map>
  </nrml>
```

- **variable:** The type of disaggregation is defined by this attribute, and it can assume

- the value `magnitude_distance` or `coordinate`;
- `bin`: Each `bin` is identified by edges of the corresponding pair of parameter that it represents (e.g. lower and upper bounds of a given combination of magnitude and distance, as illustrated in the previous example). Then, the aggregated losses associated with this pair of parameters are stored in the field `absoluteLoss`, and their percentage with respect to the overall loss are defined on the field `fraction`.

9.2.8 Event loss tables

Unlike what was described for the other outputs, the event loss tables are exported using a comma separated value (.csv) file format. In this structure, each row contains the rupture id, magnitude and aggregated loss (sum of the losses from the collection of assets within the region of interest), for each event within the stochastic event sets. The following example depicts an example of this output.

```
Rupture,Magnitude,Aggregate Loss
1,8.25,79197
2,8.25,74478
3,7.75,46458
4,7.75,45153
5,7.75,42569
6,8.25,40649
7,7.75,38868
8,7.75,37707
9,7.75,37141
...
```

The asset event loss tables provide calculated losses for each of the assets in the exposure model, for each event within the stochastic event sets. In these tables, each row contains the rupture id, the asset id, magnitude and asset loss. Note that only assets that sustain nonzero losses in a rupture are listed in the asset event loss table. An example of this output type is provided below:

```
Rupture, Asset, Magnitude, Loss
col=00|ses=0239|src=1727-356|rup=001-01, asset_10, 4.7, 13785407.3791
col=00|ses=0239|src=1727-356|rup=001-01, asset_11, 4.7, 17124507.6341
col=00|ses=0239|src=1727-356|rup=001-01, asset_12, 4.7, 16654913.6026
.
.
.
```

col=01 ses=6905 src=1120-2 rup=011-01, asset_53, 6.7, 18506576.0384
col=01 ses=6905 src=1120-2 rup=011-01, asset_58, 6.7, 12095498.7066

[Scenario Damage Demos](#)

[Scenario Risk Demos](#)

[Classical Probabilistic Seismic Damage Demos](#)

[Classical Probabilistic Seismic Risk Demos](#)

[Event-Based Probabilistic Seismic Risk Demos](#)

[Retrofit Benefit-Cost Ratio Demos](#)

[List of Terms](#)

10. Demonstrative Examples

The following sections describe the set of demos that have been compiled to demonstrate some of the features and usage of the risk calculators of the OpenQuake-engine. These demos can be found in a public repository in GitHub at the following link <https://github.com/gem/oq-risklib/tree/master/demos/risk>. Furthermore, a folder containing all of these demonstrative examples is provided when an OATS (OpenQuake Alpha Testing Service) account is requested, and it is also part of the OpenQuake-engine virtual image package.

These examples are purely demonstrative and are not intended to represent accurately the seismicity, vulnerability or exposure characteristics of the region of interest, but simply to provide example input files that can be used as a starting point for users planning to employ the OpenQuake-engine in seismic risk and loss estimation studies.

It is also noted that in the demonstrative examples presented in this section, illustrations about the various messages from the engine displayed in the command line interface are presented. These messages often contain information about the calculation id and output id, which will certainly be different for each user.

Following is the list of demos which illustrate how to use the oq-engine for various scenario-based and probabilistic seismic risk analyses:

- ClassicalBCR
- ClassicalDamage
- ClassicalRisk
- EventBasedRisk
- ScenarioDamage
- ScenarioDamageCsq
- ScenarioRisk

These seven demos use Nepal as the region of interest. An example exposure model has been developed for this region, comprising 9,144 assets distributed amongst 2,221

locations (due to the existence of more than one asset at the same location). A map with the distribution of the number of buildings throughout Nepal is presented in Figure 10.1.

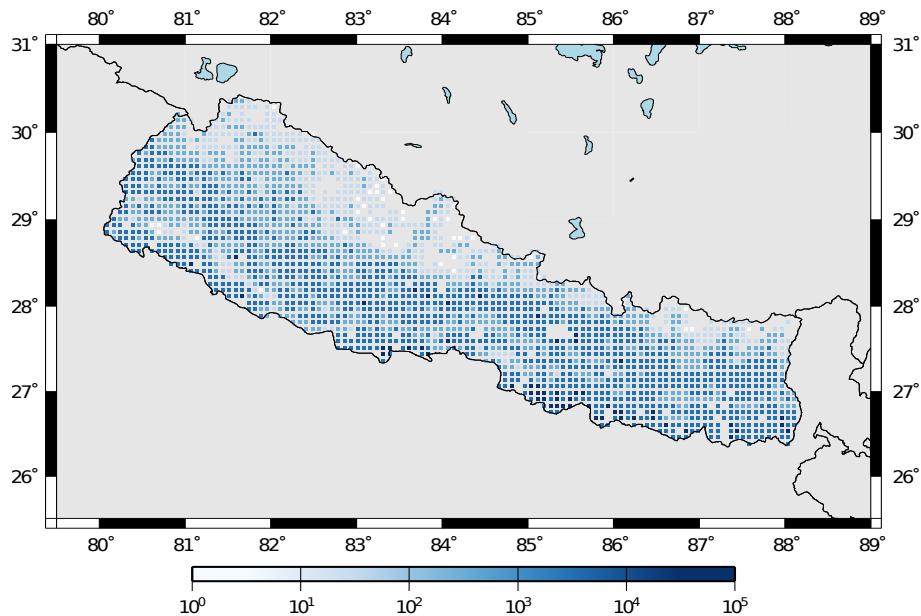


Figure 10.1 – Distribution of number of buildings in Nepal.

The building portfolio was organised into four classes for the rural areas (adobe, dressed stone, unreinforced fired brick, wooden frames), and five classes for the urban areas (the aforementioned typologies, in addition to reinforced concrete buildings). For each one of these building typologies, vulnerability functions and fragility functions were collected from the literature. These input models are only for demonstrative purposes and for further information about the building characteristics of Nepal, users are advised to contact the National Society for Earthquake Technology of Nepal (NSET - <http://www.nset.org.np/>).

The following sections include instructions not only on how to run the risk calculations, but also on how to produce the necessary hazard inputs. Thus, each demo comprises the configuration file, exposure model and fragility or vulnerability models fundamental for the risk calculations. Each demo folder also a configuration file and the input models to produce the relevant hazard inputs.

10.1 Scenario Damage Demos

The same rupture described in the Scenario Risk demo was used for this demo. The workflow to produce the set of ground motion fields described in the previous section is also valid herein. Then, to run the Scenario Damage demo, users should move to the folder where the required files have been placed and employ following command:

```
user@ubuntu:~$ oq-engine --run job_damage.ini --hazard-output-id 20
```

and the following outputs will be produced:

```
Calculation 12 results:  
id | output_type | name  
23 | collapse_map | Collapse Map per Asset  
24 | dmg_dist_per_asset | Damage Distribution per Asset  
25 | dmg_dist_per_taxonomy | Damage Distribution per Taxonomy  
26 | dmg_dist_total | Damage Distribution Total
```

10.2 Scenario Risk Demos

A rupture of magnitude 7 Mw in the central part of Nepal was considered within this demo. The characteristics of this rupture (geometry, dip, rake, hypocentre, upper and lower seismogenic depth) have been defined in the `rupture.xml` file, whilst the hazard calculation settings have been established on the `job_hazard.ini` file. In order to calculate the set of ground motion fields due to this rupture, users should navigate to the folder where the demo files are located, and use the following command:

```
user@ubuntu:~$ oq-engine --run job_hazard.ini
```

which will produce the following hazard result:

```
Calculation 10 results:  
id | output_type | name  
20 | gmf_scenario | gmf_scenario
```

Then, this hazard input can be used for the risk calculations using the following command:

```
user@ubuntu:~$ oq-engine --run job_risk.ini --hazard-output-id 20
```

leading to the following results:

```
Calculation 11 results:  
id | output_type | name  
21 | aggregate_loss | Aggregate Loss type=structural  
22 | loss_map | loss maps. type=structural
```

10.3 Classical Probabilistic Seismic Damage Demos

The same hazard input as described in the Classical PSHA-based Risk demo was used for this demo. Thus, the workflow to produce the set of hazard curves described in 10.4 is also valid herein. Then, to run the Classical PSHA-based Damage demo, users should navigate to the folder containing the demo input models and configuration files and employ the following command:

```
user@ubuntu:~$ oq-engine --run job_hazard.ini
```

which will produce the following sample hazard output:

Calculation 5 results:		
id	output_type	name
420	Hazard Curve	Hazard Curve rlz-102-PGA
419	Hazard Curve (multiple imts)	hc-multi-imt-rlz-102

In this demo, the damage distribution for each asset in the exposure model is produced. The following command launches the risk calculations:

```
user@ubuntu:~$ oq-engine --run job_risk.ini --hazard-output-id 419
```

and the following sample outputs are obtained:

Calculation 6 results:		
id	output_type	name
421	Damage Per Asset	Damage distribution for hazard=419

10.4 Classical Probabilistic Seismic Risk Demos

The seismic source model developed within the Global Seismic Hazard Assessment Program (GSHAP) was used with the Chiou and Youngs, 2008 ground motion prediction equation to produce the hazard input for this demo. No uncertainties are considered in the seismic source model and since only one GMPE is being considered, there will be only one possible path in the logic tree. Therefore, only one set seismic hazard curves will be produced. To do so, the following command needs to be employed:

```
user@ubuntu:~$ oq-engine --run job_hazard.ini
```

which will produce the following hazard output:

```
Calculation 13 results:  
id | output_type | name  
27 | hazard_curve | hc-rlz-70
```

In this demo, loss exceedance curves for each asset and two probabilistic loss maps (for probabilities of exceedance of 1% and 10%) are produced. The following command launches these risk calculations:

```
user@ubuntu:~$ oq-engine --run job_risk.ini --hazard-output-id 27
```

and the following outputs are expected:

```
Calculation 14 results:  
id | output_type | name  
28 | loss_curve | loss curves. type=structural, hazard=27  
29 | loss_map | loss maps. type=structural poe=0.1, hazard=27  
30 | loss_map | loss maps. type=structural poe=0.01, hazard=27
```

10.5 Event-Based Probabilistic Seismic Risk Demos

This demo uses the same probabilistic seismic hazard assessment (PSHA) model described in the previous example. However, instead of hazard curves, sets of ground motion fields are required. To trigger the hazard calculations the following command needs to be used:

```
user@ubuntu:~$ oq-engine --run job_hazard.ini
```

and the following results are expected:

```
Calculation 15 results:  
id | output_type | name  
414 | Ground Motion Field | GMF rlz-101  
412 | Stochastic Event Set | SES Collection 0  
413 | Stochastic Event Set | SES Collection 1
```

Again, since there is only one branch in the logic tree, only one set of ground motion fields will be used in the risk calculations, which can be launched through the following command:

```
user@ubuntu:~$ oq-engine --run job_risk.ini --hc 15
```

leading to the following outputs:

```
Calculation 16 results:  
id | output_type | name  
415 | Event Loss Table | Event Loss Table type=structural, hazard=414  
416 | Event Loss Asset | Event Loss Asset type=structural, hazard=414  
417 | Loss Curve | loss curves. type=structural, hazard=414  
418 | Aggregate Loss Curve | aggregate loss curves. loss_type=structural  
hazard=414||gmf||GMF rlz-101
```

10.6 Retrofit Benefit-Cost Ratio Demos

The loss exceedance curves used within this demo are produced using the Classical PSHA-based Risk calculator. Thus, the process to produce the seismic hazard curves described in the respective section (10.4) can be employed here. Then, the risk calculations can be initiated using the following command:

```
user@ubuntu:~$ oq-engine --run job_bcr.ini --hazard-output-id 27
```

which should produce the following output:

```
Calculation 17 results:  
id | output_type | name  
37 | bcr_distribution | BCR Distribution for hazard 27
```

Bibliography

Books

Aki, K. and P. G. Richards (2002). *Quantitative Seismology*. Sausalito, California: University Science Books (cited on pages 27, 32).

Articles

- Abrahamson, N. A. and W. Silva (2008). "Summary of the Abrahamson & Silva NGA Ground-Motion Relations". In: *Earthquake Spectra* 24.1, pages 67–97.
- Akkar, S. and J. J. Bommer (2010). "Empirical equations for the prediction of PGS, PGV, and spectral accelerations in Europe, the Mediterranean Region, and the Middle East". In: *Seismological Research Letters* 81.2, pages 195–206. doi: [10.1785/gssrl.81.2.195](https://doi.org/10.1785/gssrl.81.2.195).
- Akkar, S. and Z. Çağnan (2010). "A Local Ground-Motion Predictive Model for Turkey, and Its Comparison with Other Regional and Global Ground-Motion Models". In: *Bulletin of the Seismological Society of America* 100.6, pages 2978–2995.
- Atkinson, G. M. and D. M. Boore (2003). "Empirical Ground-Motion Relations for Subduction-Zone Earthquakes and Their Application to Cascadia and Other Regions". In: *Bulletin of the Seismological Society of America* 93.4, pages 1703–1729.
- Atkinson, G. M. and D. M. Boore (2006). "Earthquake Ground-Motion Prediction Equations for Eastern North America". In: *Bulletin of the Seismological Society of America* 96.6, pages 2181–2205.
- Boore, D. M. and G. M. Atkinson (2008). "Ground-Motion Prediction Equations for the Average Horizontal Component of PGA, PGV, and 5%-Damped PSA at Spectral Periods between 0.01 s and 10.0 s". In: *Earthquake Spectra* 24.1, pages 99–138.
- Campbell, K. W. and Y. Bozorgnia (2003). "Updated Near-Source Ground-Motion (Attenuation) Relations for the Horizontal and Vertical Components of Peak Ground Acceleration and Acceleration Response Spectra". In: *Bulletin of the Seismological Society of America* 93, pages 314–331.
- Cauzzi, C. and E. Faccioli (2008). "Broadband (0.05 s to 20 s) prediction of displacement response spectra based on worldwide digital records". In: *Journal of Seismology* 12, pages 453–475.
- Chiou, B. S.-J. and R. R. Youngs (2008). "An NGA Model for the Average Horizontal Component of Peak Ground Motion and Response Spectra". In: *Earthquake Spectra* 24, pages 173–215 (cited on pages 53, 140).
- Chiou, B. S.-J. and R. R. Youngs (2014). "Update of the Chiou and Youngs NGA model for the average horizontal component of peak ground motion and response spectra". In: *Earthquake Spectra* 30.3, pages 1117–1153 (cited on page 28).

- Cornell, C. A. (1968). "Engineering Seismic Risk Analysis". In: *Bulletin of the Seismological Society of America* 58, pages 1583–1606 (cited on pages 40, 91, 94, 95).
- Field, E. H., T. H. Jordan, and C. A. Cornell (2003). "OpenSHA - A developing Community-Modeling Environment for Seismic Hazard Analysis". In: *Seismological Research Letters* 74, pages 406–419 (cited on pages 40, 91).
- Frankel, A. (1995). "Mapping Seismic Hazard in the Central and Eastern United States". In: *Seismological Research Letters* 66.4, pages 8–21 (cited on pages 25, 149).
- Lin, P-S. and C.-T. Lee (2008). "Ground-Motion Attenuation Relationships for Subduction-Zone Earthquakes in Northeastern Taiwan". In: *Bulletin of the Seismological Society of America* 98, pages 220–240.
- Sadigh, K., C.-Y. Chang, J. Egan, F. Makdisi, and R. R. Youngs (1997). "Attenuation relationships for shallow crustal earthquakes based on California strong motion data". In: *Seismological Research Letters* 68, pages 180–189.
- Scherbaum, F. and N. M. Kuehn (2011). "Logic tree branch weights and probabilities: Summing up to one is not enough". In: *Earthquake Spectra* in press (cited on page 147).
- Schwartz, D. P. and K. J. Coppersmith (1984). "Fault Behaviour and Characteristic Earthquakes: Examples from the Wasatch and San Andreas fault zones". In: *Journal of Geophysical Research* 89.B7, pages 5681–5698 (cited on page 27).
- Strasser, F. O., M. C. Arango, and J. J. Bommer (2010). "Scaling of the Source Dimensions of Interface and Intraslab Subduction-zone Earthquakes with Moment Magnitude". In: *Seismological Research Letters* 81 (cited on page 39).
- Wells, D. L. and K. J. Coppersmith (1994). "New Empirical Relationships among Magnitude, Rupture Length, Rupture Width, Rupture Area, and Surface Displacement". In: *Bulletin of the Seismological Society of America* 84.4, pages 974–1002 (cited on page 39).
- Woo, G. (1996). "Kernel Estimation Methods for Seismic Hazard Area Source Modeling". In: *Bulletin of the Seismological Society of America* 86.2, pages 353–362 (cited on page 25).
- Youngs, R. R. and K. J. Coppersmith (1985). "Implications of fault slip rates and earthquake recurrence models to probabilistic seismic hazard estimates". In: *Bulletin of the Seismological Society of America* 75, pages 939–964 (cited on pages 37, 38).
- Youngs, R., S. Chiou, W. Silva, and J. R. Humphrey (1997). "Strong Ground Motion Attenuation Relationships for Subduction Zone Earthquakes". In: *Seismological Research Letters* 68, pages 58–73.
- Zhao, J. X., J. Zhang, A. Asano, Y. Oyno, T. Oouchi, T. Takahashi, H. Ogawa, K. Irikura, H. K. Thio, P. G. Somerville, Y. Fukushima, and Y. Fukushima (2006). "Attenuation Relations of Strong Ground Motion in Japan Using Site Classification Based on Predominant Period". In: *Bulletin of the Seismological Society of America* 96, pages 898–913. doi: [10.1785/0120050122](https://doi.org/10.1785/0120050122).

Other Sources

- EPRI, E. P. R. I. (2011). *Technical Report: Central and Eastern United States Seismic Source Characterisation for Nuclear Facilities*. Report. EPRI, Palo Alto, CA. U. S. DoE, and U. S. NERC (cited on page 39).
- Faccioli, E., A. Bianchini, and M. Villani (2010). "New ground motion prediction equations for $T > 1$ s and their influence on seismic hazard assessment". In: *Proceedings of the University of Tokyo Symposium on Long-Period Ground Motion and Urban Disaster Mitigation*.
- McGuire, K. K. (1976). *FORTRAN computer program for seismic risk analysis*. Open-File report 76-67. 102 pages. United States Department of the Interior, Geological Survey (cited on pages 40, 91, 94, 95).
- Petersen, M. D., A. D. Frankel, S. C. Harmsen, C. S. Mueller, K. M. Haller, R. L. Wheeler, R. L. Wesson, Y. Yzeng, O. S. Boys, D. M. Perkins, N. Luco, E. H. Field, C. J. Wills, and K. S. Rukstales (2008). *Documentation for the 2008 Update of the United States National Seismic Hazard Maps*. Open File Report 2008-1128. U.S. Department of the Interior, U.S. Geological Survey (cited on page 24).
- Toro, G. R. (2002). "Modification of the Toro Et Al. (1997) Attenuation Equations for Large Magnitudes and Short Distances". url: http://riskeng.com/PDF/atten_toro_extended.pdf.

Glossary

List of Terms

area source

A source type usually adopted to model distributed seismicity. In an area source the seismicity occurrence rate is assumed uniform over the source area; this produces an hazard pattern with a plateau of constant hazard inside the polygon delimiting the area source and values of hazard that tend to decrease as we move away from the border of the source.

asset

An asset is an element with a certain value, which can include buildings or population. For example, an asset can include an individual building at a given location, or a number of buildings that are grouped, co-located at a single location and classified with the same taxonomy.

branch

The simplest element in a logic tree; it belongs to a branch set where it represents one possible option among a finite number of alternatives. A branch is associated with a weight value (Scherbaum and Kuehn, 2011) if the branch set represents the epistemic uncertainty on a parameter or a model when the branch set is used to specify alternative models (e.g. district MFDs).

branch set

The structure describing the epistemic uncertainty on a specific parameter or model included in a logic tree structure. It ensembles a number of branches, each one representing a discrete alternative.

branching level

It indicates the position where a branch set or a branch is located in a logic tree structure. For example, the first branching level of the seismic source logic tree always contains one or several initial seismic source input models.

characteristic fault source

A fault source typology where ruptures always cover the entire fault surface.

complex fault source

A source typology usually adopted to model subduction interface faults.

consequence function

the distribution of the consequence (or loss) ratio conditional on a set of discrete limit states, defined for a particular taxonomy.

consequence model

A set of consequence functions used to model the consequence ratios of all the taxonomies in the exposure model.

dip

The dip is the steepest angle of descent of the fault plane relative to a horizontal plane; it is measured in degrees [0,90].

earthquake rupture

A 3D surface - representing a portion or the entire fault surface - over which a slip event (i.e. an earthquake) occurs.

earthquake rupture forecast

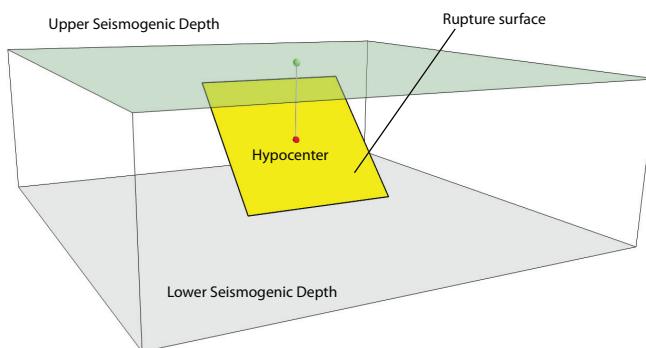
A list of all possible ruptures generated by all the sources included in a seismic source model. Each element in the list contains: the rupture geometry and the rupture probability of occurrence in a given time span. See also the definition available on the [OpenSHA website](#).

exposure model

A set of assets grouped according to their geographical location, taxonomy and value.

fault trace

A curve representing the intersection between the surface containing the fault surface (or its prolongation) and the topographic surface .

**fragility function**

the probability of exceeding a set of limit states, given an intensity measure level. These functions can be discrete or continuous.

fragility model

A set of fragility functions used to model the fragility of all the assets in the exposure model.

frequency-magnitude distribution

See magnitude-frequency distribution.

grid source

A source typology usually adopted to model distributed seismicity. It is routinely produced by a seismicity smoothing algorithm (one of the most famous algorithm is the one proposed by Frankel (1995)).

ground-motion logic tree

A method used to systematically describe the epistemic uncertainties related to the ground motion models used in the computation of hazard using a specific PSHA input model.

initial seismic source input model

It is the ensemble of information needed to fully describe the seismic sources composing a seismic source input model. The initial seismic source input model is included in the first branching level of a seismic source logic tree.

magnitude-frequency distribution

A distribution describing the frequency of earthquakes with a specific magnitude. It can be continuous or discrete. One frequency-magnitude distribution frequently adopted in PSHA is the double truncated Gutenberg-Richter distribution.

magnitude-scaling relationship

An empirical relationship linking the magnitude with a parameter describing the size of the corresponding rupture (e.g. the area of the rupture or the rupture length).

non-parametric source

A source typology in which the earthquake rupture forecast is described explicitly by a set of ruptures and the corresponding probabilities of occurrence.

point source

The elemental source typology used in OpenQuake-engine to model distributed seismicity.

probabilistic seismic hazard analysis

A methodology to compute seismic hazard by taking into account the potential contributions coming from all the sources of engineering importance for a specified site.

PSHA input model

An object containing the information necessary to describe the seismic source and the ground motion models - plus the related epistemic uncertainties.

rake

The rake is the direction in which a hanging wall block moves during a rupture, measured relative to fault strike on the plane of the fault.

rupture model

An object containing the information necessary to describe a earthquake rupture, such as magnitude, hypocenter location, strike, dip, rake, and seismogenic depths.

seismic source data

An object containing the information necessary to completely describe a PSHA seismic source i.e. seismic source type, position, geometry and seismicity occurrence model.

seismic source input model

An object containing a list of seismic source data. In the OpenQuake-engine a seismic source model doesn't contain epistemic uncertainty.

seismic source logic tree

Logic tree structure defined to describe in structured and systematic way the epistemic uncertainties characterizing the seismic source model. The first branching level in the logic tree by definition contains one or several alternative initial seismic source input model.

simple fault source

A source typology usually adopted to model shallow structures with an uncomplicated geometry.

strike

The strike direction correspond to the angle between the north and the direction you take so that when you walk along the fault trace the fault dips on your right.

taxonomy

Scheme used to classify the assets. For buildings, a classification scheme has been proposed by GEM which considers a number of attributes including lateral load resisting system and its material, height, year of construction. The taxonomy is currently used to link the assets in the exposure model to the relevant vulnerability function or fragility function.

vulnerability function

A function that describes the probability distribution of loss ratio, conditioned on an intensity measure level. Currently only discrete vulnerability functions are supported.

vulnerability model

A set of vulnerability functions used to model the physical vulnerability of all the assets in the exposure model.