



OPENQUAKE
calculate share explore

OPENQUAKE ENGINE USER INSTRUCTION MANUAL

Version 1.5.0

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



“OpenQuake: Calculate, share, explore”

The OpenQuake-engine User Manual

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

© 2015 GEM FOUNDATION

GLOBALQUAKEMODEL.ORG/OPENQUAKE

Citation

Please cite this document as:

Crowley, H., Monelli, D., Pagani, M., Silva, V., Weatherill, G., and Rao, A. (2015). The OpenQuake-engine User Manual. *Global Earthquake Model (GEM) Technical Report 2015-09*. doi: 10.13117/GEM.OPENQUAKE.MAN.ENGINE.1.5/01, 146 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 an “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](https://creativecommons.org/licenses/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.

Fifth printing, September 2015

Contents

Preface	11
I Introduction	13
1 OpenQuake-engine Background	15
1.1 Running the OpenQuake-engine	15
1.2 Concurrent Computing with OpenQuake	18
II Hazard	21
2 Introduction to the Hazard Module	23
2.1 Source typologies	23
2.1.1 Source typologies for modelling distributed seismicity	24
2.1.1.1 <i>Point sources</i>	24
2.1.1.2 <i>Grid sources</i>	26
2.1.1.3 <i>Area sources</i>	27
2.1.2 Fault sources with floating ruptures	28
2.1.2.1 <i>Simple faults</i>	28
2.1.2.2 <i>Complex faults</i>	31

2.1.3	Fault sources without floating ruptures	32
2.1.3.1	<i>Characteristic faults</i>	32
2.2	Magnitude-frequency distributions	32
2.3	Magnitude-scaling relationships	34
2.3.1	Relationships for shallow earthquakes in active tectonic regions	34
2.4	Calculation workflows	35
2.4.1	Classical Probabilistic Seismic Hazard Analysis	36
2.4.2	Event-Based Probabilistic Seismic Hazard Analysis	36
2.4.3	Scenario based Seismic Hazard Analysis	37
3	Using the Hazard Module	39
3.1	Defining Logic Trees	39
3.1.0.1	<i>Logic trees as described in the nrml schema</i>	40
3.2	The Seismic Source System	45
3.2.1	The Seismic Source Logic Tree	45
3.2.2	The Seismic Source Model	47
3.3	The Ground Motion System	47
3.3.0.1	<i>The Ground Motion Logic Tree</i>	47
3.4	Configuration file	48
3.4.1	Classical PSHA	48
3.4.2	Seismic hazard disaggregation	51
3.4.3	Event based PSHA	52
4	Hazard Calculations and Results	55
4.1	Running OpenQuake-engine for hazard calculations	55
4.2	Exporting results from a hazard calculation	55
4.3	Description of hazard outputs	56
4.3.1	Outputs from Classical PSHA	57
4.3.2	Outputs from Hazard Disaggregation	59
4.3.3	Outputs from Event Based PSHA	61

5	Demonstrative Examples	65
5.1	Classical PSHA Demos	65
5.1.1	Classical PSHA with different source typologies	66
5.1.2	Classical PSHA with non trivial logic trees	70
5.2	Hazard Disaggregation Demos	77
5.3	Event Based PSHA Demos	78
III	Risk	81
6	Introduction to the Risk Module	83
6.1	Introduction	83
6.2	Calculation workflows	83
6.2.1	Scenario Risk Calculator	83
6.2.2	Scenario Damage Calculator	84
6.2.3	Probabilistic Event-based Risk Calculator	84
6.2.4	Classical PSHA-based Risk Calculator	85
6.2.5	Classical PSHA-based Damage Calculator	86
6.2.6	Retrofitting Benefit/Cost Ratio Calculator	86
7	Using the Risk Module	89
7.1	Input data definition	89
7.1.1	Exposure model definition	89
7.1.2	Physical vulnerability model definition	98
7.1.3	Fragility model definition	102
7.1.4	Configuration file	105
7.1.4.1	<i>Scenario Risk Calculator</i>	106
7.1.4.2	<i>Scenario Damage Calculator</i>	106
7.1.4.3	<i>Probabilistic Event-based Risk Calculator</i>	107
7.1.4.4	<i>Classical PSHA-based Risk Calculator</i>	109
7.1.4.5	<i>Classical PSHA-based Damage Calculator</i>	109
7.1.4.6	<i>Retrofitting Benefit/Cost Ratio Calculator</i>	109

8	Risk Calculations and Results	111
8.1	Running OpenQuake-engine for risk calculations	111
8.2	Description of the outputs	113
8.2.1	Loss statistics	114
8.2.2	Loss maps	114
8.2.3	Damage distribution	116
8.2.4	Collapse maps	119
8.2.5	Loss exceedance curves	119
8.2.6	Retrofitting Benefit/cost ratio maps	120
8.2.7	Loss disaggregation	122
8.2.8	Event loss tables	123
9	Demonstrative Examples	125
9.1	Scenario Risk demo	126
9.2	Scenario Damage demo	127
9.3	Classical PSHA-based Risk demo	127
9.4	Classical PSHA-based Damage demo	128
9.5	Probabilistic Event-based demo	129
9.6	Retrofitting Benefit/cost ratio demo	129
A	Ground Shaking Intensity Models	133
A.1	GMPEs for shallow earthquakes in active tectonic regions	133
A.2	GMPEs for subduction sources	134
A.3	GMPEs for stable continental regions	135
	Bibliography	137
	Books	137
	Articles	137

Reports 139

Index 141

Preface

The goal of this manual is to provide a comprehensive and transparent description of the features of the OpenQuake-engine (v1.5). 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 Affero GPL license (more information is available at this link: <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/theglobalearthquakemodelgem.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 [--help]
               [--version]
               [--log-file LOG_FILE]
               [--log-level debug,info,progress,warn,error,critical]
               [--no-distribute]
               [--run CONFIG_FILE]
               [--list-inputs INPUT_TYPE]
               [--yes]
               [--config-file CONFIG_FILE]
               [--make-html-report YYYY-MM-DD|today]
               [--lite]
               [--upgrade-db]
               [--version-db]
```

```

[--what-if-I-upgrade]
[--run-hazard CONFIG_FILE]
[--list-hazard-calculations]
[--delete-hazard-calculation HAZARD_CALCULATION_ID]
[--delete-uncompleted-calculations]
[--run-risk CONFIG_FILE]
[--hazard-output-id HAZARD_OUTPUT]
[--hazard-calculation-id HAZARD_CALCULATION_ID]
[--list-risk-calculations]
[--delete-risk-calculation RISK_CALCULATION_ID]
[--list-outputs CALCULATION_ID]
[--show-view CALCULATION_ID VIEW_NAME]
[--list-hazard-outputs CALCULATION_ID]
[--list-risk-outputs CALCULATION_ID]
[--exports EXPORT_FORMATS]
[--export-output OUTPUT_ID TARGET_DIR]
[--export-hazard-output OUTPUT_ID TARGET_DIR]
[--export-risk-output OUTPUT_ID TARGET_DIR]
[--export-outputs CALCULATION_ID TARGET_DIR]
[--export-stats CALCULATION_ID TARGET_DIR OUTPUT_TYPE]
[--export-hazard-outputs HAZARD_CALCULATION_ID TARGET_DIR]
[--export-risk-outputs RISK_CALCULATION_ID TARGET_DIR]
[--save-hazard-calculation HAZARD_CALCULATION_ID DUMP_DIR]
[--load-hazard-calculation DUMP_DIR]
[--load-gmf GMF_FILE]
[--load-curve CURVE_FILE]
[--list-imported-outputs]

```

OpenQuake Seismic Hazard and Risk Analysis Engine

optional arguments:

```
-h, --help          show this help message and exit
```

General:

```

--version           Display version information
--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, -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
--list-inputs, --li INPUT_TYPE
                    List inputs of a specific input type
--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, -r YYYY-MM-DD|today
                    Build an HTML report of the computation at the given
                    date
--lite             Use lite version of the calculator, if possible

```


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, --rh CONFIG_FILE
                    Run a hazard job with the specified config file
--list-hazard-calculations, --lhc
                    List hazard calculation information
--delete-hazard-calculation, --dhc HAZARD_CALCULATION_ID
                    Delete a hazard calculation and all associated outputs
--delete-uncompleted-calculations, --duc
                    Delete all the uncompleted calculations
```

Risk:

```
--run-risk, --rr CONFIG_FILE
                    Run a risk job with the specified config file
--hazard-output-id, --ho HAZARD_OUTPUT
                    Use the desired hazard output as input for the risk
                    job
--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
--delete-risk-calculation, --drc RISK_CALCULATION_ID
                    Delete a risk calculation and all associated outputs
```

Export:

```
--list-outputs, --lo CALCULATION_ID
                    List outputs for the specified calculation
--show-view, --sv CALCULATION_ID VIEW_NAME
                    Show a view of the specified calculation
--list-hazard-outputs, --lho CALCULATION_ID
                    List outputs for the specified calculation
                    [deprecated]
--list-risk-outputs, --lro CALCULATION_ID
                    List outputs for the specified calculation
                    [deprecated]
--exports EXPORT_FORMATS
                    Comma-separated string specifying the export formats,
                    in order of priority
--export-output, --eo OUTPUT_ID TARGET_DIR
                    Export the desired output to the specified directory
--export-hazard-output, --eh OUTPUT_ID TARGET_DIR
                    Export the output to the specified directory
                    [deprecated]
--export-risk-output, --er OUTPUT_ID TARGET_DIR
                    Export the output to the specified directory
                    [deprecated]
--export-outputs, --eos CALCULATION_ID TARGET_DIR
                    Export all the calculation outputs to the specified
                    directory
--export-stats, --es CALCULATION_ID TARGET_DIR OUTPUT_TYPE
                    Export the statistical outputs to the specified
                    directory
--export-hazard-outputs, --eho HAZARD_CALCULATION_ID TARGET_DIR
                    Export all the outputs to the specified directory
```

```

[deprecated]
--export-risk-outputs, --ero RISK_CALCULATION_ID TARGET_DIR
Export all the outputs to the specified directory
[deprecated]

Save/Load:
--save-hazard-calculation, --shc HAZARD_CALCULATION_ID DUMP_DIR
Save a hazard calculation to a new created directory
--load-hazard-calculation DUMP_DIR
Load a hazard calculation from a saved import. Only
SES outputs currently supported.

Import:
--load-gmf GMF_FILE Load gmf from a file. Only single-source gmf are
supported currently. The file can be xml or tab-
separated.
--load-curve CURVE_FILE
Load hazard curves from an XML file
--list-imported-outputs
List outputs which were imported from a file, not
calculated from a job

```

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

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 :

- 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.

2.1.1 Source typologies for modelling distributed seismicity

2.1.1.1 Point sources

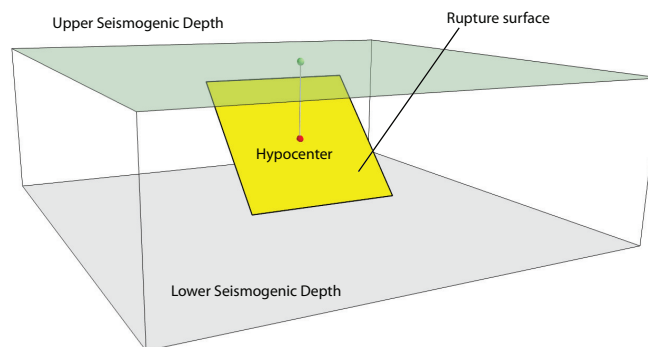


Figure 2.1 – *Single rupture*

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.

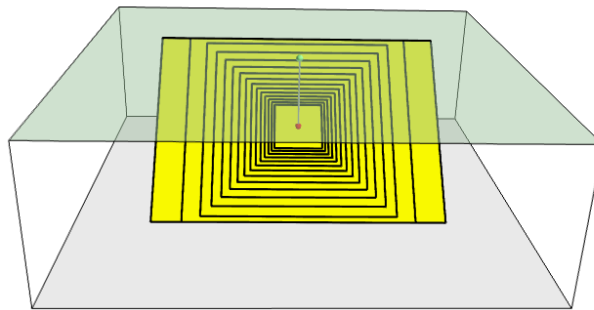


Figure 2.2 – Point source with multiple ruptures. Note the change in the aspect ratio once the rupture width fills the entire seismogenic layer.

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 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.

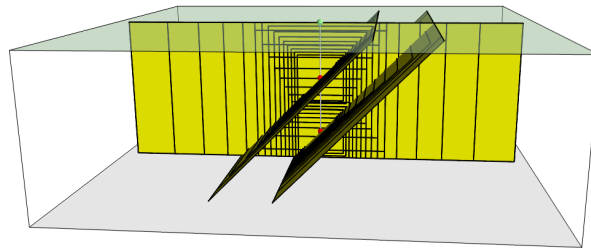


Figure 2.3 – Ruptures produced by the source created using the information in the example .xml file described at page 26.

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 24).

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 glsmr
- 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 magnitude-frequency 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 counterclockwise 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.

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.

Source data

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

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

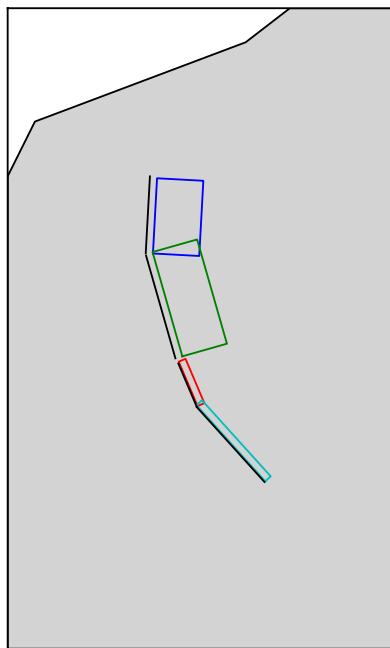


Figure 2.4 – *Geometry of a multi-segmented characteristic fault composed of four rectangular ruptures as modelled in OpenQuake.*

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 30.

```
<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.

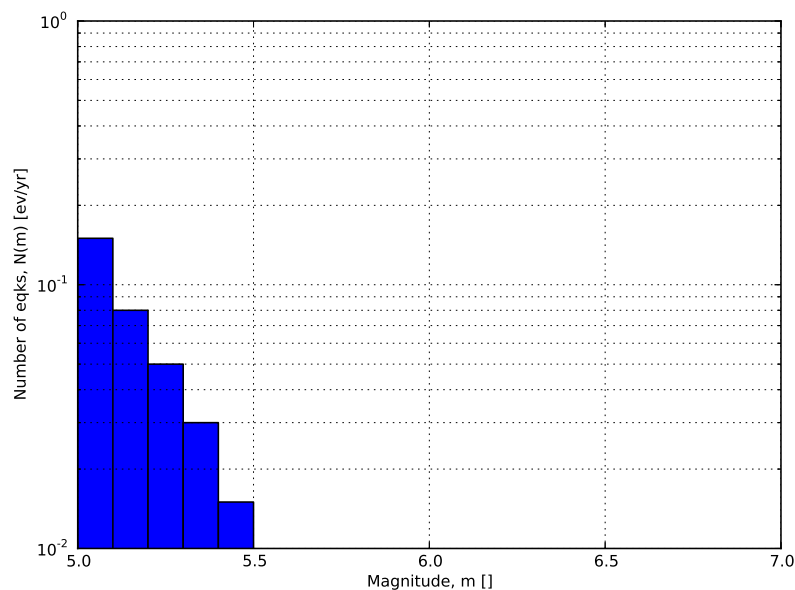


Figure 2.5 – Example of an incremental magnitude-frequency distribution.

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.

2.3 Magnitude-scaling relationships

2.3.1 Relationships for shallow earthquakes in active tectonic regions

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

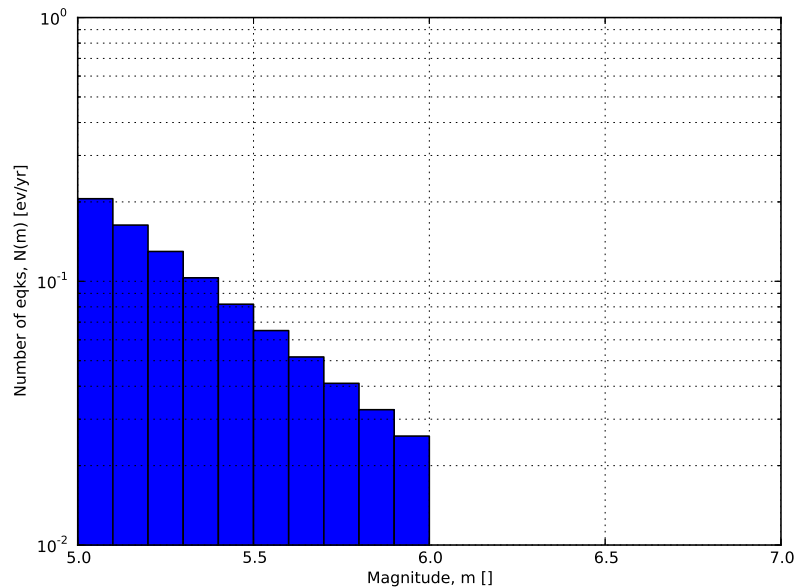


Figure 2.6 – Example of a double truncated Gutenberg-Richter magnitude-frequency distribution.

- 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.

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 36).

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 36).

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 36).

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 42)
- One or more branches

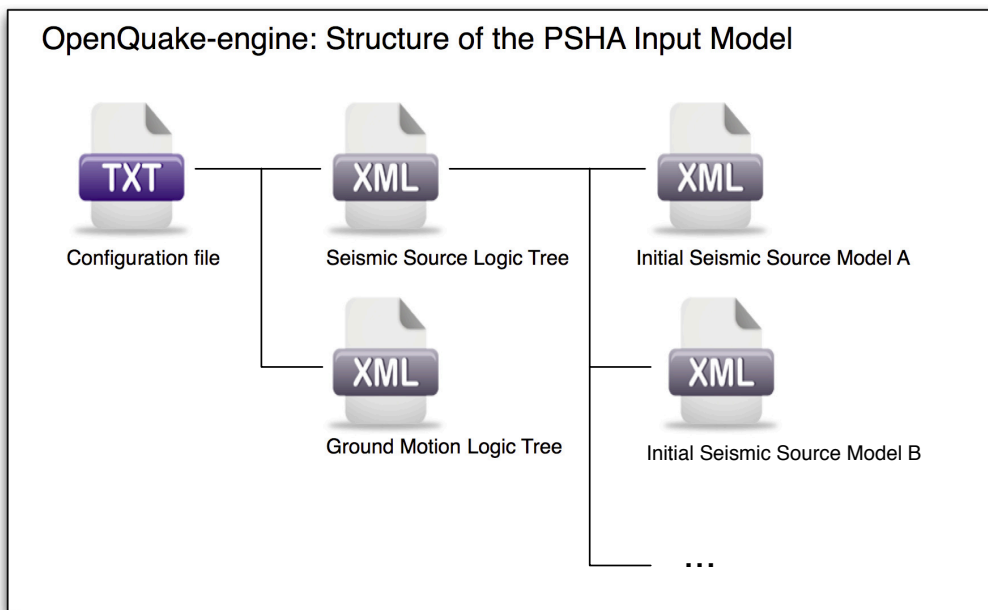


Figure 3.1 – PSHA Input Model structure

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.

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:

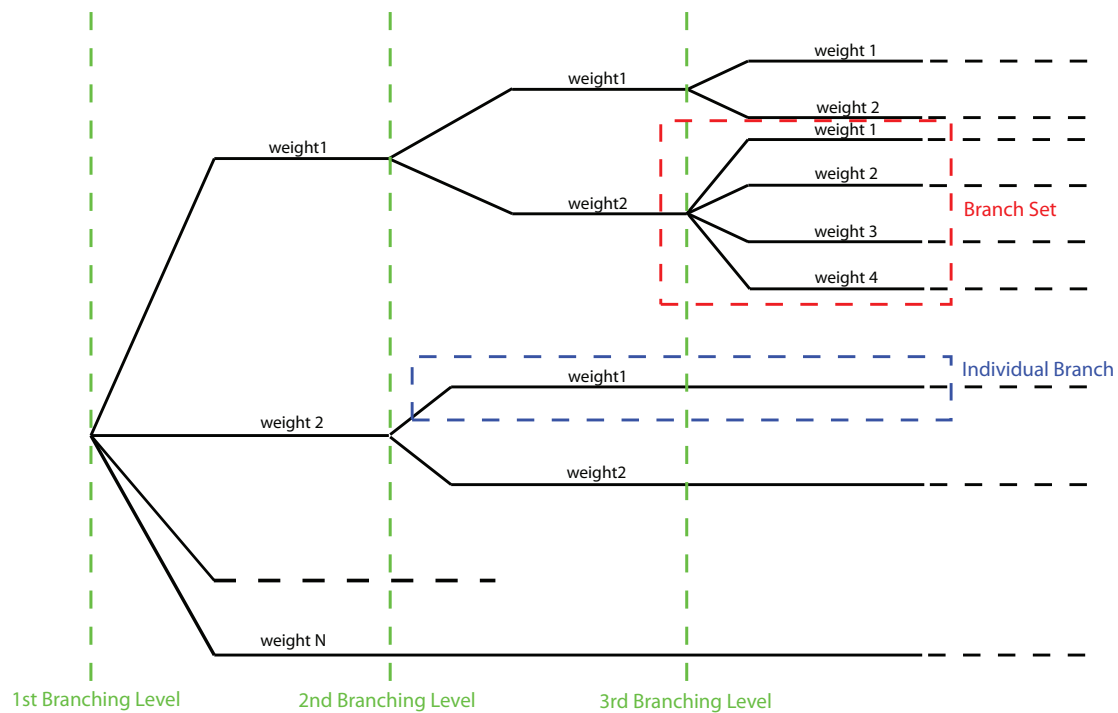


Figure 3.2 – Generic Logic Tree structure as described in terms of branching levels, branch sets, and individual branches.

```
<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.

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>
      ...
      <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 A), 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>
```

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 <nrml xmlns:gml="http://www.opengis.net/gml"
3   xmlns="http://openquake.org/xmlns/nrml/0.4">
4   <logicTree logicTreeID="lt1">
5     <logicTreeBranchingLevel branchingLevelID="b11">
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 <nrml xmlns:gml="http://www.opengis.net/gml"
3   xmlns="http://openquake.org/xmlns/nrml/0.4">
4   <logicTree logicTreeID="lt1">
5
6     <logicTreeBranchingLevel branchingLevelID="b11">
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="b12">
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/0.4">
4   <logicTree logicTreeID='lt1'>
5     <logicTreeBranchingLevel branchingLevelID="b11">
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"?>
<nrml 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" />
  </siteModel>
</nrml>

```

```

    <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>
</nrml>

```

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 quantile_hazard_curves =
36 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.

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 48 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 51).

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 48.

```
[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
```


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 `-rh` option which stands for “run hazard” and the name of a configuration file (in the example below it corresponds to `job.ini`):

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

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 following command:

```
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
calc_id | num_jobs | latest_job_status | last_update | description
1 | 1 | failed | 2013-03-01 09:49:34 | Classical PSHA
2 | 1 | successful | 2013-03-01 09:49:56 | Classical PSHA
3 | 1 | failed | 2013-03-01 10:24:04 | Classical PSHA
4 | 1 | failed | 2013-03-01 10:28:16 | Classical PSHA
5 | 1 | failed | 2013-03-01 10:30:04 | Classical PSHA
6 | 1 | successful | 2013-03-01 10:31:53 | Classical PSHA
7 | 1 | failed | 2013-03-09 08:15:14 | Classical PSHA
8 | 1 | successful | 2013-03-09 08:18:04 | Classical PSHA
```

Subsequently the user can get the list of result stored for a specific hazard analysis as in the example below (note that the number in blue emphasizes the result ID):

```
user@ubuntu:~$ oq-engine --lho <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 --eh <result_id> <path_to_the_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.

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 --lho <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 --lho <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 --eh <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'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
  xmlns="http://openquake.org/xmlns/nrml/0.4">
  <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>
</nrml>

```

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 --lho <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/0.4">
  <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 --lho <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/0.4">
  <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 --lho <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/0.4">
  <stochasticEventSetCollection sourceModelTreePath="b1"
    gsimTreePath="b1">
    <stochasticEventSet id="12" investigationTime="50.0">
      <rupture id="533" 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="535" 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).

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/0.4">
  <gmfCollection sourceModelTreePath="b1" gsimTreePath="b1">
    <gmfSet investigationTime="50.0" stochasticEventSetId="12">
      <gmf IMT="PGA" ruptureId="533">
        <node gmv="0.0105891230432" lon="11.1240023202"
              lat="43.5107462335"/>
        <node gmv="0.00905803920023" lon="11.1241875202"
              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>
</nrml>
```


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 considers 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/0.4">
  <logicTree logicTreeID='lt1'>

    <logicTreeBranchingLevel branchingLevelID="b11">
      <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
```

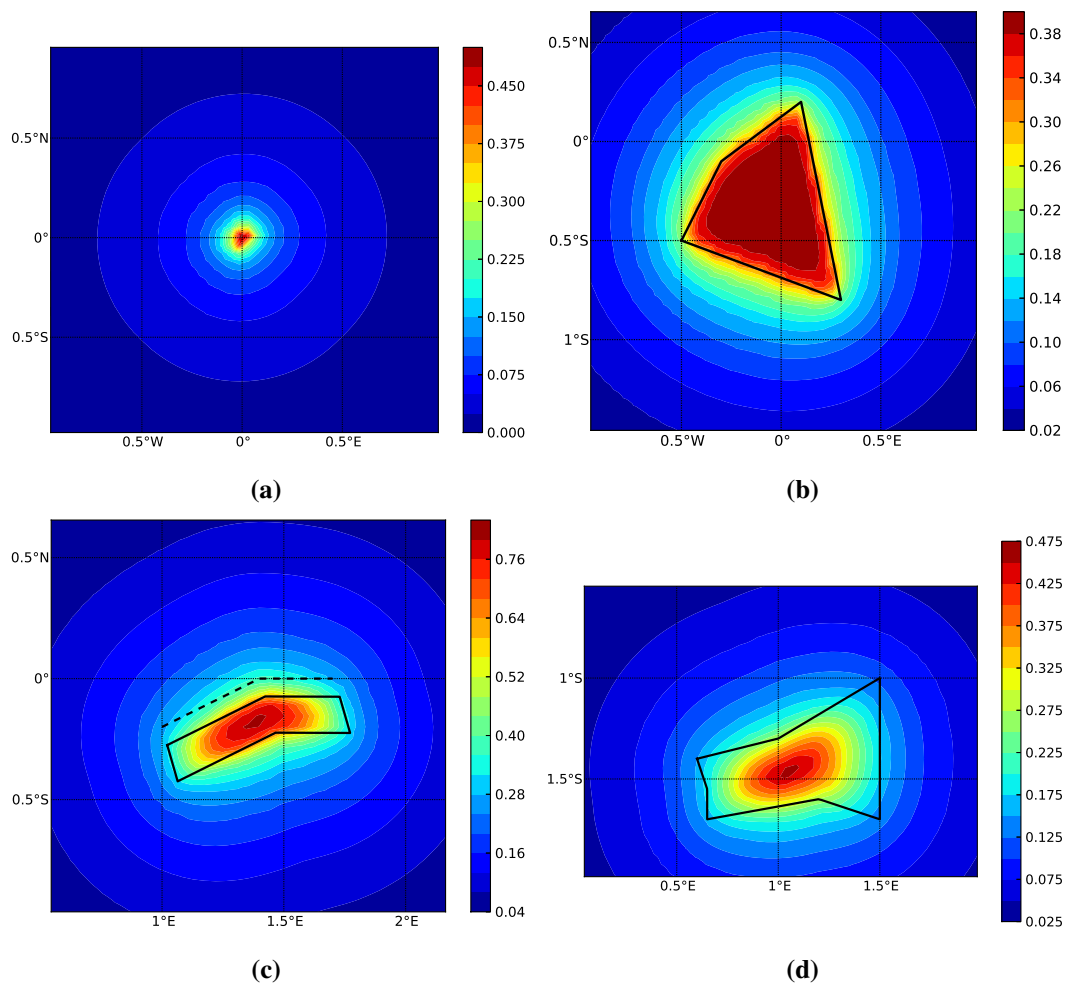


Figure 5.1 – Hazard maps (for PGA, 10% in 50 years) as obtained from the different oq-engine source typologies. (a) Point Source. (b) Area source. The solid black line represents the area boundary. (c) Simple Fault Source. The dashed line represents the fault trace, while the solid line the fault surface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

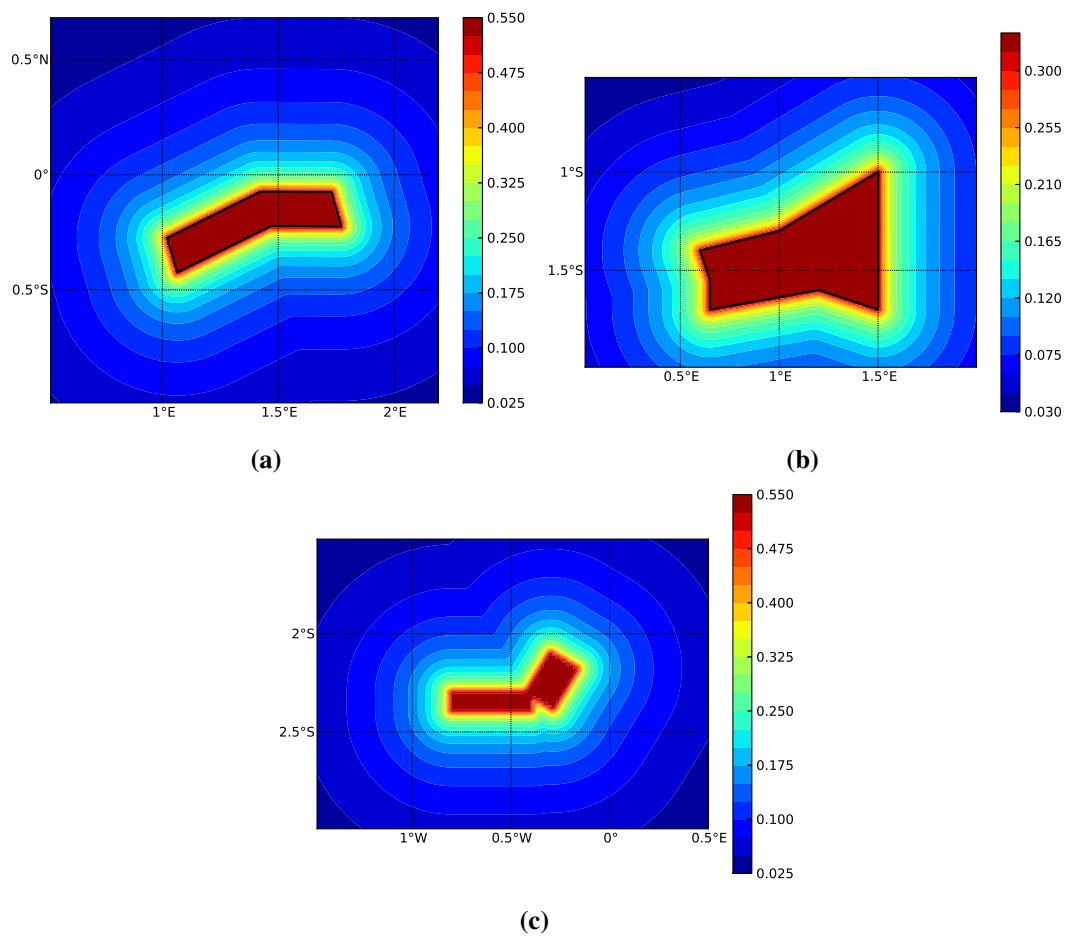


Figure 5.2 – Hazard maps (for PGA, 10% in 50 years) as obtained from characteristic fault sources with simple fault geometry (e), complex fault geometry (f), and collection of planar surfaces (g)

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"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
  <logicTree logicTreeID="lt1">

    <logicTreeBranchingLevel branchingLevelID="b11">

      <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>
</nrml>
```

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"?>
```

```
<nrml xmlns:gml="http://www.opengis.net/gml"
  xmlns="http://openquake.org/xmlns/nrml/0.4">
  <logicTree logicTreeID='lt1'>

    <logicTreeBranchingLevel branchingLevelID="b11">
      <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="b12">
      <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>
```

```
</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/0.4">
  <logicTree logicTreeID="lt1">

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

    <logicTreeBranchingLevel branchingLevelID="b12">
      <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="b13">
    <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="b14">
    <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>

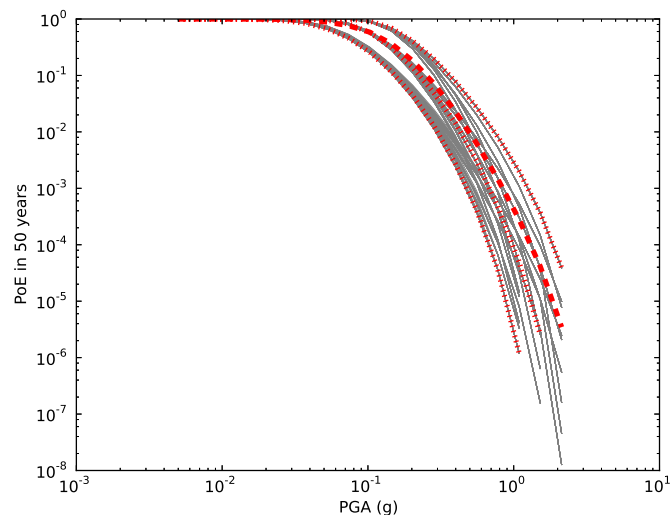
<logicTreeBranchingLevel branchingLevelID="b15">
    <logicTreeBranchSet uncertaintyType="maxMagGRAbsolute"
        applyToSources="2"
        branchSetID="bs51">
        <logicTreeBranch branchID="b51">
            <uncertaintyModel>7.5</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </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 levels are created, one defining uncertainties on G-R a and b values (defined by setting `uncertaintyType="abGRAbsolute"`) and G-R maximum magnitude (`uncertaintyType="maxMagGRAbsolute"`).

It is important to notice that each branch set is applied to a specific source by defining the attribute `applyToSources`, followed by the source ID. The GSIM logic tree file is the same as used for `LogicTreeCase1ClassicalPSHA`. By setting in the configuration file `number_of_logic_tree_samples = 0`, hazard results are obtained for 324 paths (1 source model x 3 (a, b) pairs for source 1 x 3 (a, b) pairs for source 2 x 3 max magnitude values for source 1 x 3 max magnitude values for source 2 x 2 GMPEs for Active Shallow Crust X 2 GMPEs for Stable Continental Crust), see Figure 5.3.



(a)

Figure 5.3 – Hazard curves as obtained from the *LogicTreeCase2* demo. Solid gray lines represent individual hazard curves from the different logic tree path (a total of 324 curves). The red dashed line represents the mean hazard curve, while the red dotted lines depict the quantile levels (0.15, 0.5, 0.95).

LogicTreeCase3ClassicalPSHA illustrates an example of logic tree defining relative uncertainties on G-R maximum magnitude and b value. A single source model is considered containing two sources belonging to different tectonic region types and both characterized by a G-R magnitude frequency distribution. The source model logic tree is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
  xmlns="http://openquake.org/xmlns/nrml/0.4">
  <logicTree logicTreeID="lt1">

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

    <logicTreeBranchingLevel branchingLevelID="b12">
```

```

    <logicTreeBranchSet uncertaintyType="bGRRRelative"
      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="b13">
    <logicTreeBranchSet uncertaintyType="maxMagGRRRelative"
      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 (bGRRRelative applied consistently to all sources in the source model) and the second uncertainties on maximum magnitude

(maxMagGRRrelative). 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 (`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_gmfs = true`). The corresponding hazard maps for 0.1 probability are also calculated (`hazard_maps = true`)

Part III

Risk

Introduction

Calculation workflows

- Scenario Risk Calculator
- Scenario Damage Calculator
- Probabilistic Event-based Risk Calculator
- Classical PSHA-based Risk Calculator
- Classical PSHA-based Damage Calculator
- Retrofitting Benefit/Cost Ratio Calculator

6. Introduction to the Risk Module

6.1 Introduction

The seismic risk results are being 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, that can be executed through a command line interface. The code of the library can be found on a public repository at GitHub at the following address <http://github.com/gem/oq-risklib>.

This section provides a brief description of the calculators currently implemented in oq-risklib, and an initial presentation of the input and output files is provided. In the following sections, the contents and structure of these files are discussed in detail. For further information regarding the methodologies behind each calculator, users are referred to the OpenQuake-engine Book (Risk).

6.2 Calculation workflows

The oq-engine is currently comprised of five risk calculation workflows: two that calculate losses and damage distributions due to a single earthquake, another two that calculate seismic risk using probabilistic seismic hazard, and a fifth one that uses loss exceedance curves to assess whether retrofitting measures would be economically viable or not.

6.2.1 Scenario Risk Calculator

This calculator computes loss maps and loss statistics due to a single seismic event, for a collection of assets. The hazard input can be a single ground motion field (e.g. the median distribution of ground motion in the region of interest) or a set of ground motion fields allowing the characterisation of the inter- and intra-event variability from the GMPE. It is noted that the hazard input can either be calculated using the hazard component of OpenQuake-engine (oq-hazardlib), or provided to the risk component in an external file following the respective

Natural hazards' Risk Markup Language (NRML) schema (see [oq-nrmlib](#)). A vulnerability model is combined with the distribution of the ground motions at each asset location to calculate the loss distribution for each asset, as well as the statistics of the total loss throughout the region of interest. The required input files and resulting output files are depicted in Figure 6.1.

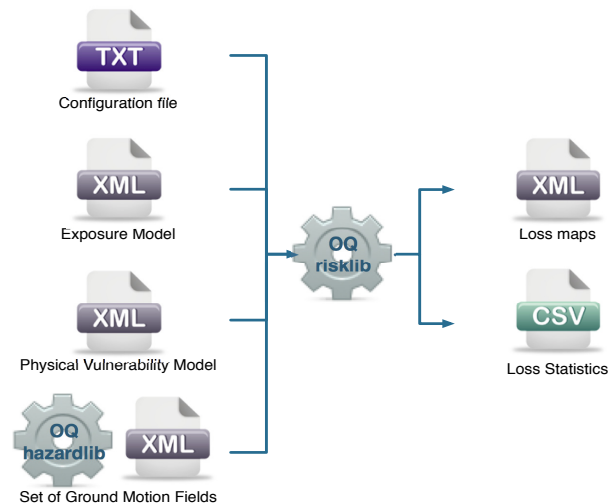


Figure 6.1 – Scenario Risk Calculator input/output structure.

6.2.2 Scenario Damage Calculator

This calculator is capable of assessing the damage distribution due to a single scenario earthquake, for a collection of assets. Similarly to the previous calculator, in order to perform the necessary risk calculations one or a set of ground motion fields are required, which can be derived using the `oq-hazardlib`, or introduced in the OpenQuake-engine using the appropriate NRML schema. In this calculator, a fragility model is combined with the distribution of ground motion at the location of each asset, to estimate the number or area of buildings in each damage state. The damage distribution can be extracted per asset, per building typology (taxonomy) or considering all of the assets simultaneously (total damage distribution). In addition, this calculator also provides collapse maps, which contain the spatial distribution of the number or area of collapsed buildings throughout the region of interest. The input/output structure for this calculator is presented in Figure 6.2.

6.2.3 Probabilistic Event-based Risk Calculator

In this calculator, loss exceedance curves and loss maps for various return periods can be calculated, based on probabilistic seismic hazard, with an event-based approach. A large number of stochastic event sets are generated, and the associated ground motion fields for each event are used together with a vulnerability model to compute the individual (per asset) and total (sum of all the losses per event) losses. Then, this distribution of losses is employed to derive a loss exceedance curve per asset, as well as a total loss exceedance curve representative of the complete building portfolio. Furthermore, `oq-risklib` can also compute loss maps for various

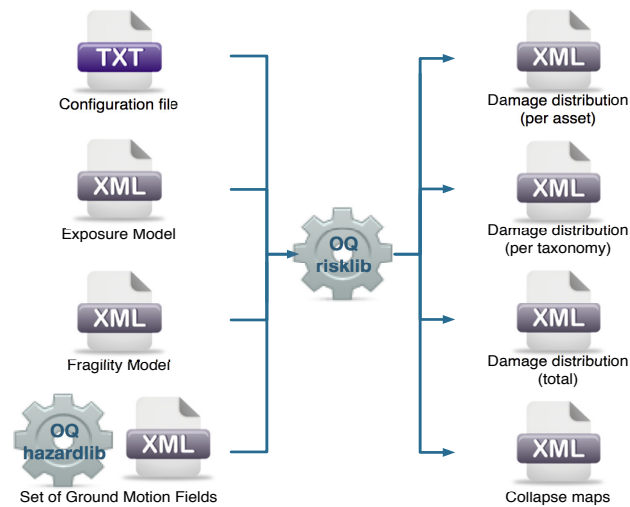


Figure 6.2 – Scenario Damage Calculator input/output structure.

return periods by interpolating each individual loss curve with the respective probability of exceedance. In Figure 6.3, the input/output scheme of this calculator is illustrated.

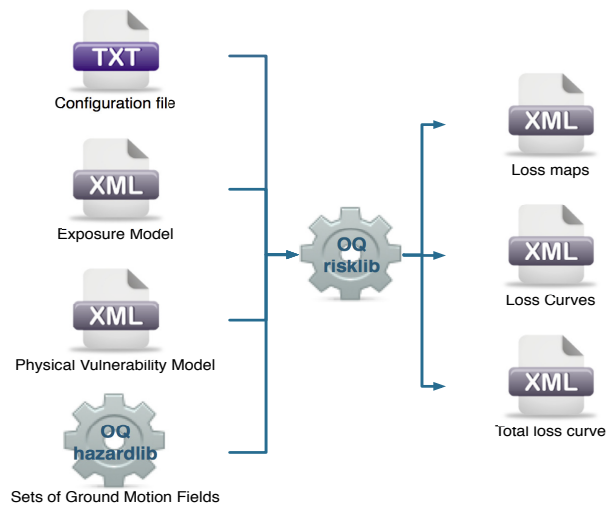


Figure 6.3 – Probabilistic Event-based Risk Calculator input/output structure.

6.2.4 Classical PSHA-based Risk Calculator

In this calculator, probabilistic seismic hazard is employed to calculate a loss exceedance curve for each asset, through the usage of seismic hazard curves. A convolution between the vulnerability function and the hazard curve at location of the asset is performed, leading to the probability of exceeding a set of loss ratios. Each loss ratio is multiplied by the asset value to obtain the final loss exceedance curve. Furthermore, probabilistic loss maps can be extracted by interpolating the loss curves at each location by various probabilities of exceedance. 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 input and output files involved in this calculator are presented in Figure 6.4.

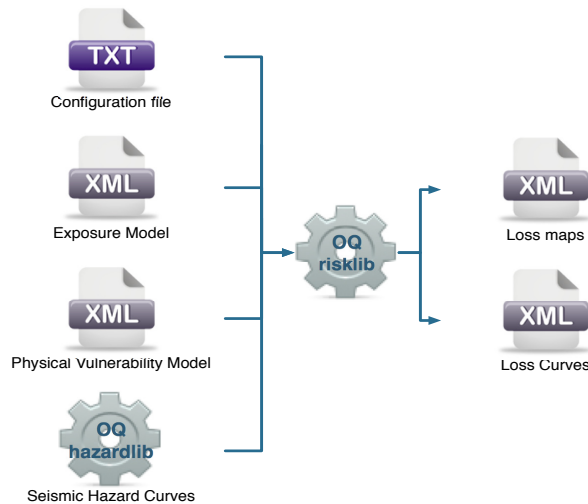


Figure 6.4 – Classical PSHA-based Risk Calculator input/output structure.

6.2.5 Classical PSHA-based Damage Calculator

The classical PSHA-based damage calculator convolves through numerical integration, the damage state fragility functions for an asset with the seismic hazard curve at the location of the asset. The main results of this calculator are damage distribution statistics for each asset, which describe the expected fraction of buildings in each damage state. Furthermore, a probabilistic collapse map can be extracted giving the probability of collapse for each asset within 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 input and output files involved in this calculator are presented in Figure 6.5.

6.2.6 Retrofitting Benefit/Cost Ratio Calculator

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.

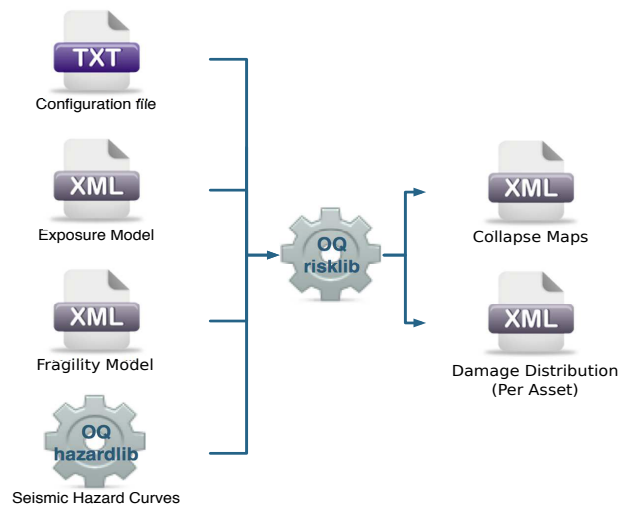


Figure 6.5 – *Classical PSHA-based Damage Calculator input/output structure.*

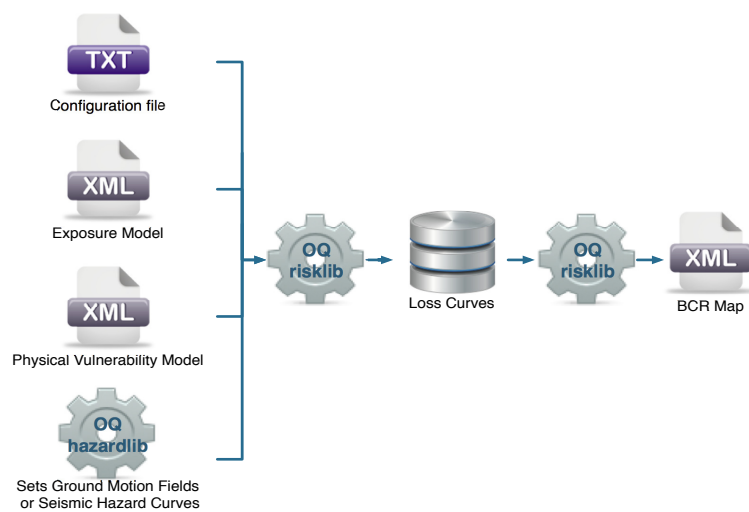


Figure 6.6 – *Retrofitting Benefit/Cost Ratio Calculator input/output structure.*

Input data definition

Exposure model definition
Physical vulnerability model definition
Fragility model definition
Configuration file

7. Using the Risk Module

7.1 Input data definition

7.1.1 Exposure model definition

All risk calculators in the OpenQuake-engine require an exposure model that needs to be stored in NRML. There are a number of parameters that compose the metadata, and provides general information regarding the assets within the exposure model, as described below:

- **id**: a unique key used to identify the exposure model;
- **category**: a string used to define the type of assets being stored (e.g: buildings, population, contents);
- **taxonomySource**: attribute used to define the taxonomy being used to classify the assets;
- **description**: brief string with further information about the exposure model;

This information is common to all the assets and needs to be incorporated at the beginning of the exposure model file according to the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns="http://openquake.org/xmlns/nrml/0.4">
  <exposureModel id="exposure_model"
    category="buildings"
    taxonomySource="GEM taxonomy">
    <description>Buildings in Pavia</description>
  ...
```

The NRML schema for the exposure model allows the definition of various types of costs (structural cost, nonstructural cost, contents cost, business interruption cost). Further explanations regarding the quantities that are currently being used to define the exposure elements can be found in the OpenQuake-engine Book (Risk).

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. Thus, the following set of attributes exist within the schema for the exposure model:

- **number:** number of units of a given asset at a given location;
- **area:** area of the asset, at a given location;
- **cost:** structural replacement cost of the asset at a given location;

The set of required attributes depends on what and how a user wants to store the information about the assets in the exposure model. While the attribute `number` might be a rather simple parameter, the other two (`area` and `cost`) can be ambiguous, as different ways to define them might be used. With regards to the attribute `area`, one can either choose to provide the aggregated built up area of the assets per location or the average built up area for a single building unit (noting that an asset might be made up of a number of individual buildings). Similarly, the `cost` can also be defined as the aggregated structural replacement cost, the cost of replacing a single unit or even the structural replacement cost per unit of area. 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 is comprised of an exposure model in which the aggregated cost (structural, nonstructural, 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:

```
...
<conversions>
  <costTypes>
    <costType name="structural" type="aggregated" unit="EUR">
    <costType name="non_structural" type="aggregated" unit="EUR" />
    <costType name="business_interruption" type="aggregated"
      unit="EUR"/>
    <costType name="contents" type="aggregated" unit="EUR"/>
  </costTypes>
</conversions>
...
```

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:

```

...
<assets>
  <asset id="asset_01" taxonomy="RC/DMRF-D/LR">
    <location lon="9.15" lat="45.17" />
    <costs>
      <cost type="structural" value="1500"/>
      <cost type="non_structural" value="2500"/>
      <cost type="contents" value="1200"/>
      <cost type="business_interruption" value="400"/>
    </costs>
  </asset>
...
  <asset id="asset_99" taxonomy="RC/DMRF-D/HR">
    <location lon="9.15" lat="45.12" />
    <costs>
      <cost type="structural" value="2500"/>
      <cost type="non_structural" value="2100"/>
      <cost type="contents" value="1900"/>
      <cost type="business_interruption" value="40"/>
    </costs>
  </asset>
</assets>
</exposureModel>
</nrml>

```

Each asset is uniquely identified by its id, which is used by the OpenQuake-engine to relate each asset with the associated results (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.¹ 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.

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

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.

```

...
<conversions>
  <costTypes>
    <costType name="structural" type="per_unit" unit="EUR">
    <costType name="non_structural" type="per_unit" unit="EUR" />
    <costType name="business_interruption" type="per_unit"
      unit="EUR"/>
    <costType name="contents" type="per_unit" unit="EUR"/>
  </costTypes>
</conversions>
...

```

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

```

...
<assets>
  <asset id="asset_01" number="10" taxonomy="RC/DMRF-D/LR">
    <location lon="9.15" lat="45.17" />
    <costs>
      <cost type="structural" value="150"/>
      <cost type="non_structural" value="250"/>
      <cost type="contents" value="120"/>
      <cost type="business_interruption" value="40"/>
    </costs>
  </asset>
...
  <asset id="asset_99" number="20" taxonomy="RC/DMRF-D/HR">
    <location lon="9.15" lat="45.12" />
    <costs>
      <cost type="structural" value="125"/>
      <cost type="non_structural" value="105"/>
      <cost type="contents" value="95"/>
      <cost type="business_interruption" value="20"/>
    </costs>
  </asset>
</assets>

```

```
</exposureModel>
</nrml>
```

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 unit” 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 per area.

```
...
<conversions>
  <area type="aggregated" unit="square meters"/>
  <costTypes>
    <costType name="structural" type="per_area" unit="EUR">
    <costType name="non_structural" type="per_area" unit="EUR" />
    <costType name="business_interruption" type="per_area"
      unit="EUR"/>
    <costType name="contents" type="per_area" unit="EUR"/>
  </costTypes>
</conversions>
...
```

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 store 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.

```
...
<assets>
  <asset id="asset_01" area="50" taxonomy="RC/DMRF-D/LR">
    <location lon="9.15" lat="45.17" />
    <costs>
      <cost type="structural" value="100"/>
      <cost type="non_structural" value="200"/>
      <cost type="contents" value="90"/>
      <cost type="business_interruption" value="10"/>
    </costs>
```

```

</asset>
...
<asset id="asset_99" area ="60" taxonomy="RC/DMRF-D/HR">
  <location lon="9.15" lat="45.12" />
  <costs>
    <cost type="structural" value="150"/>
    <cost type="non_structural" value="250"/>
    <cost type="contents" value="50"/>
    <cost type="business_interruption" value="30"/>
  </costs>
</asset>
</assets>
</exposureModel>
</nrml>

```

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 area.

```

...
<conversions>
  <area type="per_asset" unit="square meters"/>
  <costTypes>
    <costType name="structural" type="per_area" unit="EUR">
    <costType name="non_structural" type="per_area" unit="EUR" />
    <costType name="business_interruption" type="per_area"
      unit="EUR"/>
    <costType name="contents" type="per_area" unit="EUR"/>
  </costTypes>
</conversions>
...

```

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_unit`.

```

...
<assets>

```

```

<asset id="asset_01" number="5" area="50" taxonomy="RC/DMRF-D/LR">
  <location lon="9.15" lat="45.17" />
  <costs>
    <cost type="structural" value="100"/>
    <cost type="non_structural" value="200"/>
    <cost type="contents" value="90"/>
    <cost type="business_interruption" value="10"/>
  </costs>
</asset>
...
<asset id="asset_99" number="8" area="60" taxonomy="RC/DMRF-D/HR">
  <location lon="9.15" lat="45.12" />
  <costs>
    <cost type="structural" value="150"/>
    <cost type="non_structural" value="250"/>
    <cost type="contents" value="50"/>
    <cost type="business_interruption" value="30"/>
  </costs>
</asset>
</assets>
</exposureModel>
</nrml>

```

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 define, according to the format below.

```

...
<conversions>
  <costTypes>
    <costType name="structural" type="aggregated" unit="EUR">
    <costType name="non_structural" type="aggregated" unit="EUR" />
    <costType name="business_interruption" type="aggregated"
      unit="EUR"/>
    <costType name="contents" type="aggregated" unit="EUR"/>
  </costTypes>

```

```

<deductible isAbsolute="false"/>
<insuranceLimit isAbsolute="false"/>
</conversions>
...

```

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`. 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`. Then, for each type of cost, the limit and deductible value can be stored for each asset. 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.

```

...
<assets>
  <asset id="asset_01" taxonomy="RC/DMRF-D/LR">
    <location lon="9.15" lat="45.17" />
    <costs>
      <cost type="structural" value="1500" deductible=".05"
            insuranceLimit="0.9" retrofitted="200"/>
      <cost type="non_structural" value="2500" deductible=".1"
            insuranceLimit="0.8"/>
      <cost type="contents" value="1200" deductible=".2"
            insuranceLimit="0.6"/>
      <cost type="business_interruption" value="400" deductible=".1"
            insuranceLimit="0.5"/>
    </costs>
  </asset>
...
  <asset id="asset_99" taxonomy="RC/DMRF-D/HR">
    <location lon="9.15" lat="45.12" />
    <costs>
      <cost type="structural" value="2500" deductible=".1"
            insuranceLimit="0.9" retrofitted="300"/>
      <cost type="non_structural" value="2100" deductible=".05"
            insuranceLimit="0.7"/>
      <cost type="contents" value="1900" deductible=".2"
            insuranceLimit="0.7"/>
      <cost type="business_interruption" value="40" deductible=".05"
            insuranceLimit="0.9">
    </costs>
  </asset>

```



```

</assets>
</exposureModel>
</nrml>

```

Despite the fact that for the demonstration of how the insurance parameters and retrofitting cost can be stored it was used the aggregated type of cost (structure described in example 1), it is important to mention that any of the other storing approaches can also be employed (example 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. In this example, it is demonstrated 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.

```

...
<conversions>
  <area type="aggregated" unit="square meters"/>
  <costTypes>
    <costType name="structural" type="per_unit" unit="EUR">
    <costType name="non_structural" type="per_area" unit="EUR" />
    <costType name="contents" type="aggregated" unit="EUR"/>
  </costTypes>
</conversions>
...

```

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 replacement cost per unit (example 2), the non_structural cost is established as the cost per area (example 3), and the contents cost is provided directly as the aggregated value per asset (example 1). The information about each asset is presented bellow, along with the number of occupants at different times of the day.

```

...
<assets>
  <asset id="asset_01" number="5" area ="500" taxonomy="RC/DMRF-D/LR">
    <location lon="9.15" lat="45.17" />
    <costs>
      <cost type="structural" value="1000"/>
      <cost type="non_structural" value="250"/>
      <cost type="contents" value="5000"/>
    </costs>
  </asset>
</assets>

```

```

    </costs>
    <occupancies>
      <occupancy occupants="10" period="day"/>
      <occupancy occupants="50" period="night"/>
    </occupancies>
  </asset>
  ...
  <asset id="asset_99" number="8" area ="800" taxonomy="RC/DMRF-D/HR">
    <location lon="9.15" lat="45.12" />
    <costs>
      <cost type="structural" value="2000"/>
      <cost type="non_structural" value="400"/>
      <cost type="contents" value="4000"/>
    </costs>
    <occupancies>
      <occupancy occupants="20" period="day"/>
      <occupancy occupants="30" period="night"/>
    </occupancies>
  </asset>
</assets>
</exposureModel>
</nrml>

```

The number of occupants for each asset is stored under the `occupancies` field, as part of the occupancy instance. The number and type of periods of the day is not a fixed variable, and a user can provide as many as needed (e.g. morning, afternoon, night, transit, 9am-17pm). The descriptions used to define each period are used to specify the time of the day for which the human losses should be estimated in the Scenario Risk calculator.

The way this information is being stored is constantly being modified, as further feedback from users and experts is received. Hence, it is important to understand which version of NRML the engine is using, in order to avoid incompatibility issues. NRML is currently v0.4 and documentation about each release can be found on GitHub (see [oq-nrmlib](#)). Several examples of exposure models containing different types of information are presented below.

7.1.2 Physical vulnerability model definition

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 7.1, and the equivalent NRML file is then presented. Note that although the uncertainty for each loss ratio is not represented in the aforementioned figure, it has been 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 or beta. This model

is comprised of two discrete vulnerability functions and uses spectral acceleration for a given period of vibration as the intensity measure type.

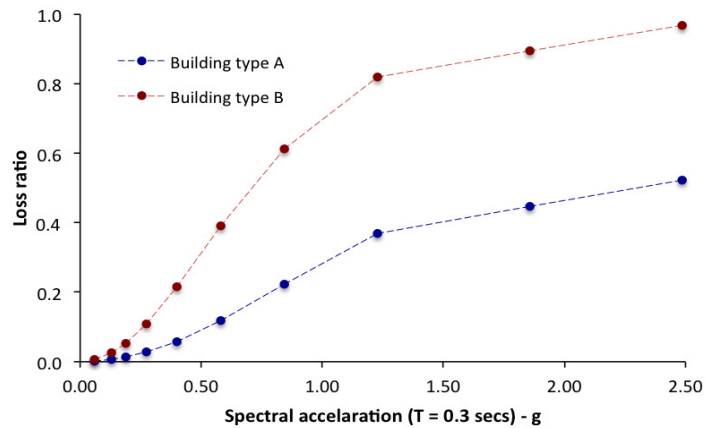


Figure 7.1 – Graphical representation of a vulnerability model.

Each component of the associated NRML file is presented herein:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrm1 xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrm1/0.4">
<vulnerabilityModel>
  <discreteVulnerabilitySet vulnerabilitySetID="OpenQuake2013"
    assetCategory="buildings" lossCategory="economic loss">
    ...
```

At the top of the NRML schema, the following metadata are being stored:

- **vulnerabilitySetID**: A unique key used to identify the vulnerability model instance within the OpenQuake-engine;
- **assetCategory**: An attribute that describes the asset typology (e.g.: population, buildings, contents);
- **lossCategory**: An attribute that describes the type of loss being modelled for the asset-Category (e.g. fatalities, structural replacement cost, contents replacement cost).

```
...
  <IML IMT = "SA(0.3)"> 0.061 0.129 0.188 0.273 0.398 0.579
    0.843 1.227 1.856 2.485 </IML>
  ...
```

Within this component, an attribute specifying the intensity measure type (e.g.: Sa, PGA, MMI) is defined, followed by the list of intensity measure levels. This set of values is common to all of the vulnerability functions in the model.

```

...
<discreteVulnerability vulnerabilityFunctionID="typeA"
probabilisticDistribution="LN">
  <lossRatio> 0.002 0.007 0.014 0.028 0.058 0.118
  0.223 0.370 0.446 0.523 </lossRatio>
  <coefficientsVariation> 0.012 0.058 0.079 0.159 0.265
  0.244 0.211 0.152 0.088 0.082 </coefficientsVariation>
</discreteVulnerability>
<discreteVulnerability vulnerabilityFunctionID="typeB"
probabilisticDistribution="LN">
  <lossRatio> 0.006 0.025 0.052 0.108 0.215 0.391
  0.613 0.820 0.894 0.967 </lossRatio>
  <coefficientsVariation> 0.010 0.054 0.082 0.167 0.285
  0.278 0.261 0.132 0.084 0.021 </coefficientsVariation>
</discreteVulnerability>
</discreteVulnerabilitySet
</vulnerabilityModel>
</nrml>

```

Finally, for each discrete vulnerability function the following parameters are required:

- **vulnerabilityFunctionID** : A unique key that is used to relate each vulnerability function with the assets in the exposure model;
- **probabilisticDistribution** : An attribute that establishes the type of probabilistic distribution used to model the uncertainty in loss ratio. At the moment, the OpenQuake-engine supports lognormal (LN) and beta (BT) distributions;
- **lossRatio** : A set of mean loss ratios (one for each intensity measure level defined previously). These values can represent different losses such as fatality rates (ratio between the number of fatalities and total population exposed) or so-called damage ratio (ratio between the repair cost and the replacement cost of a given structure).
- **coefficientsVariation** : A set of coefficients of variation (one per loss ratio) that describes the uncertainty in the loss ratio. If users do not want to consider the uncertainty, this set of parameters can be set to zero, and the OpenQuake-engine assumes each loss ratio as a deterministic value.

In the previously described vulnerability model all of the vulnerability functions were defined in terms of a single intensity measure type (Sa for 0.3 seconds). However, the current version of the engine also allows the employment of a vulnerability model that is comprised of vulnerability functions that each use distinct intensity measure types. In the following example, the schema of a vulnerability model in which three intensity measure types were used (PGA, PGV and Sa for 0.3 seconds) is presented.

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

```

<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
  <vulnerabilityModel>
    <discreteVulnerabilitySet vulnerabilitySetID="Nepal13_PGA"
      assetCategory="buildings" lossCategory="economic loss">
      <IML IMT = "PGA"> 0.1 0.2 0.4 0.7 1.0 1.3 </IML>
      <discreteVulnerability vulnerabilityFunctionID="RC1"
        probabilisticDistribution="LN">
        <lossRatio> 0.02 0.1 0.3 0.6 0.8 0.9 </lossRatio>
        <coefficientsVariation> 0.7 0.5 0.3 0.2 0.1 0.05
        </coefficientsVariation>
      </discreteVulnerability>
    </discreteVulnerabilitySet>
    <discreteVulnerabilitySet vulnerabilitySetID="Nepal13_PGV"
      assetCategory="buildings" lossCategory="economic loss">
      <IML IMT = "PGV"> 5 20 40 60 80 100 </IML>
      <discreteVulnerability vulnerabilityFunctionID="RC2"
        probabilisticDistribution="LN">
        <lossRatio> 0.05 0.2 0.3 0.4 0.5 0.6 </lossRatio>
        <coefficientsVariation> 0.6 0.3 0.2 0.1 0.05 0.05
        </coefficientsVariation>
      </discreteVulnerability>
    </discreteVulnerabilitySet>
    <discreteVulnerabilitySet vulnerabilitySetID="Nepal13_SA"
      assetCategory="buildings" lossCategory="economic loss">
      <IML IMT = "SA(0.3)"> 0.1 0.3 0.6 0.9 1.2 1.5 </IML>
      <discreteVulnerability vulnerabilityFunctionID="RC3"
        probabilisticDistribution="LN">
        <lossRatio> 0.01 0.06 0.12 0.17 0.24 0.33 </lossRatio>
        <coefficientsVariation> 1.5 1.1 1.0 0.9 0.8 0.5
        </coefficientsVariation>
      </discreteVulnerability>
    </discreteVulnerabilitySet>
  </vulnerabilityModel>
</nrml>

```

Several methodologies to derive vulnerability functions are currently being evaluated by Global Earthquake Model (GEM) and will be a part of a set of modelling tools. Scripts to convert vulnerability functions stored in Excel or ASCII files into NRML have already been developed, and can be found at the GEM Science tools repository at GitHub (<http://github.com/GEMScienceTools>).

7.1.3 Fragility model definition

This section describes the schema currently used to store fragility models, which are required for the Scenario Damage Calculator. A fragility model can be comprised of several fragility functions, describing the probability of exceeding a set of limit, or damage, states. These fragility functions can be defined in two ways: discrete or continuous. In the former manner, sets of probabilities of exceedance (one set per limit state) are defined for a list of intensity measure levels, as illustrated in Figure 7.2.

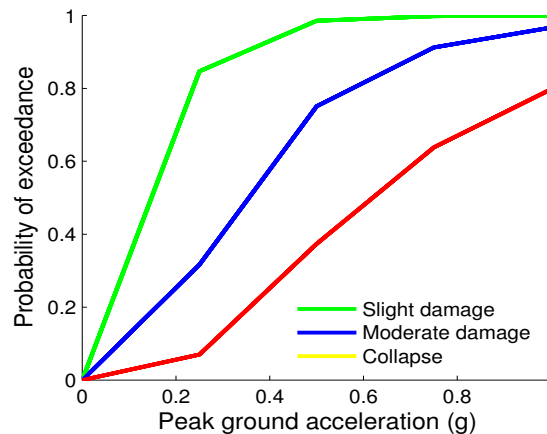


Figure 7.2 – Graphical representation of a discrete fragility model.

Similarly to what has been described for the vulnerability models, the NRML schema for this input also has some attributes that are common to all of the fragility functions comprising the model. This initial portion of the schema is depicted below:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
  <<fragilityModel format="discrete">>
    <description "Fragility Model for RC" </description>
    <limitStates
      slight
      moderate
      collapse
    </limitStates>
    ...
  <</fragilityModel>>
</nrml>
```

- **description:** represents an attribute that can be used to include some information about the fragility model, for example, what building typologies are being covered or the source of the fragility model;
- **limitStates:** this field is used to define the number and nomenclature of each limit state. Despite the fact that three limit states are being employed in this example, it is possible to use any number of states, as long as a fragility curve is always defined for each limit state.

```

...
<ffs noDamageLimit= 0.05>
  <taxonomy RC </taxonomy>
  <IML IMT="PGA" imlUnit="g"> 0.0 0.25 0.50 0.75 1.00 </IML>

  <ffd ls="slight">
    <poes> 0.0 0.85 0.98 0.99 1.00 < /poes>
  </ffd>
  <ffd ls="moderate">
    <poes> 0.0 0.32 0.75 0.91 0.97 < /poes>
  </ffd>
  <ffd ls="collapse">
    <poes> 0.0 0.07 0.37 0.64 0.80 < /poes>
  </ffd>
</ffs>
</fragilityModel>
</nrml>

```

For each building typology, a set of limit state curves need to be stored within the field `ffs` (fragility function set). The following attributes are currently being employed to define this input:

- `noDamageLimit`: this attribute defines the intensity measure level below which the probability of exceedance for all curves is zero;
- `taxonomy`: a unique key that is used to relate each fragility function with the relevant assets in the exposure model;
- `IML`: this attribute serves the purposes of defining the list of intensity measure levels for which the limit state curves are defined. In addition, it is also necessary to define the intensity measure type (`IMT`) being used and the respective units (`imlUnit`);
- `ffd`: this field (fragility function discrete) is used to define the probabilities of exceedance (`poes`) of each limit state curve. It is also necessary to include which limit state is being defined in the attribute `ls`.

As previously mentioned, the user may choose to define the fragility functions in a continuous manner, through the use of cumulative lognormal functions. In Figure 7.3, a continuous fragility model is presented.

The NRML schema to store these functions has an initial structure similar to that described for the discrete fragility models. Then, the continuous limit state curves are stored as illustrated below:

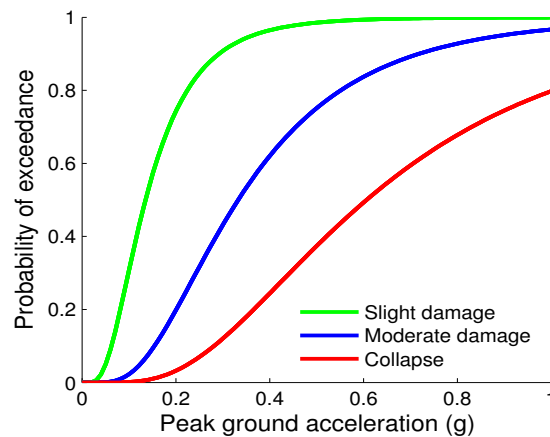


Figure 7.3 – Graphical representation of a continuous fragility model.

```

...
<ffs noDamageLimit= 0.05>
  <taxonomy RC </taxonomy>
  <IML IMT="PGA" minIML="0.0" maxIML="1.0" imlUnit="g" ></IML>
  <ffd ls="slight">
    <params mean="0.16" stddev="0.11" />
  </ffd>
  <ffd ls="moderate">
    <params mean="0.40" stddev="0.26" />
  </ffd>
  <ffd ls="collapse">
    <params mean="0.73" stddev="0.48" />
  </ffd>
</ffs>
</fragilityModel>
</nrml>

```

Again, the set of limit state curves for each building typology needs to be stored within the field `ffs` (fragility function set), through the definition of the following attributes:

- `noDamageLimit`: this attribute defines the intensity measure level below which the probability of exceedance for all curves is zero;
- `type`: this parameter defines the type of probabilistic distribution being used to define the limit state curves. Currently the engine only supports lognormal distributions, however, the capability of considering other types of distributions (e.g. normal, exponential) will be developed in the future;
- `taxonomy`: a unique key that is used to relate each fragility function with the relevant assets in the exposure model;

- **IML**: in this field, the intensity measure type (IMT) and associated units (`imlUnit`) for the limit state curves is defined, along with the minimum (`minIML`) and maximum (`maxIML`) intensity measure levels enclosing the range of applicability of the set of fragility functions;
- **ffc**: this field (fragility function continuous) is used to define the mean (`mean`) and standard deviation (`stddev`) of the cumulative lognormal function. In addition, the limit state for the curve being defined needs to be specified in the attribute `ls`.

7.1.4 Configuration file

The configuration file (or `job.ini` file) represents the location where the paths to the input files, the parameters controlling the risk calculations and the type of outputs are defined. Some initial parameters common to all the risk calculators are presented below. The remaining parameters that are specific to each risk calculator are discussed in subsequent sections. For additional information about how each parameter is being used within the methodologies implemented in the oq-engine, users advised to consult the OpenQuake-engine Book (Risk).

```
[general]
description = Scenario Risk Nepal
calculation_mode = scenario_risk

exposure_file = exposure_model.xml
region_constraint = 78.0 31.5,89.5 31.5,89.5 25.5,78 25.5
asset_hazard_distance = 10
...
```

- **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 sets the type of calculations. The key word for each risk calculator is described in the following sections;
- **exposure_file**: this parameter is used to specify the path to the exposure model file;
- **region_constraint**: this field is used to define the polygon enclosing the region of interest. Assets outside of this region will not be considered in the risk calculations. This region is defined using pairs of coordinates (longitude and latitude in decimal degrees) that indicate the vertexes of the polygon;
- **asset_hazard_distance**: this parameter indicates the maximum allowable distance between an asset and the closest hazard input. 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.

Depending on the type of calculations, other parameters besides the aforementioned ones need to be provided, as will be described in the following sections.

7.1.4.1 Scenario Risk Calculator

In order to run this calculator, the parameter `calculation_mode` needs to be set to `scenario_risk`. The remaining parameters are illustrated bellow.

```
...
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
```

- `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 non-structural vulnerability 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;
- `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 assumes no correlation in the vulnerability;
- `master_seed`: this parameter is used to control the random generator in the loss ratio sampling process. This way, if the same `master_seed` is defined at each calculation run, the same random loss ratios will be generated, thus allowing replicability of the results;
- `insured_losses`: this parameter is used to define if insured losses should be calculated (`true`) or not (`false`).

7.1.4.2 Scenario Damage Calculator

For this calculator, the parameter `calculation_mode` needs to be defined as `scenario_damage`. There is only one parameter specific to this calculator, which is the fragility model file path, as

presented below.

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

- `fragility_file`: a parameter used to define the path to the fragility model file.

7.1.4.3 Probabilistic Event-based 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 should 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 Probabilistic Event-based Risk Calculator can disaggregate the losses based on magnitude/distance and location (longitude/latitude) of the events. In order to do so, it is necessary to define the bin width of each of these parameters, as illustrated in the following example.

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  
  
sites_disagg = 85.07917, 27.4625  
mag_bin_width = 0.5  
distance_bin_width = 20  
coordinate_bin_width = 0.5  
  
specific_assets = asset_11 asset_132 asset_303 asset_611  
  
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;
- `sites_disagg`: list of locations (pairs of longitude and latitude) where the loss disaggregation should be carried out. Notice that in order to perform the loss disaggregation, assets needs to exist at those locations;
- `mag_bin_width`: this parameter specifies the with of the magnitude bins (in Mw);
- `distance_bin_width`: this parameter specifies the with of the distance bins (in km);
- `coordinate_bin_width`: this parameter specifies the with of the coordinates bins (in decimal degrees);
- `specific_assets`: this parameter specifies the set of assets for which the asset event loss table will be generated;

The definition of the parameters for the loss disaggregation follow the same rules established for the seismic hazard disaggregation described in section 3.4.2.

Note regarding the new event-based risk calculators

Starting from this release some of the scientific calculators are saving their inputs and outputs in a single HDF5 file, called the datastore. The HDF5 format is a well known standard in the scientific community, can be read/written by a variety of programming languages and with different tools and it is a state-of-the-art technology when it comes to managing large numeric datasets. The change to the HDF5 technology provides huge performance benefits compared to the earlier approach used by the engine, which involved storing arrays in PostgreSQL.

In OpenQuake 1.5 the event based calculators based on Postgres (both hazard and risk) are officially deprecated. They are still present and work as before, but they are being replaced with new versions of the calculators based on the HDF5 technology. The actual removal of the old calculators is scheduled for OpenQuake 1.6. The change will have no impact on regular users, who will simply notice a definite improvement in erformance. Nonetheless, the change will affect power users who are performing queries on the OpenQuake database, since there will be nothing left in the database once we remove the old calculators.

In order to make the transition easier, OpenQuake 1.5 already includes the new versions of the event based calculators based on HDF5, so it is possible to use them right now. The new calculators can be run in OpenQuake 1.5 with the command:

```
user@ubuntu:~$ oq-engine --lite --run job_haz.ini,job_risk.ini
```

If you do not pass the `-lite` flag the old calculators will be run by default. In future releases of the engine, the remaining calculators based on Postgres will be progressively replaced by the new calculators based on HDF5. At the end of this process, which will be spread over several upcoming releases, the `-lite` flag will be removed. All of the old calculators relying on the database will be replaced internally by the newer “lite” versions based on HDF5 and the old calculators will not be available anymore. The OpenQuake database will only contain accessory information (essentially a table with the users and references to the outputs of each user) but nothing relevant for the scientific computation.

7.1.4.4 Classical PSHA-based 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).

7.1.4.5 Classical PSHA-based 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
```

7.1.4.6 Retrofitting 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.

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

8. Risk Calculations and Results

8.1 Running OpenQuake-engine for risk calculations

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. One of the first tasks that needs to be performed is the definition of the seismic hazard input. As mentioned in Section 6.2, the risk calculations can use the results produced by the hazard component of the OpenQuake-engine. Moreover, for the two scenario-based calculators, users also have the option of loading a set of ground motion fields that might have been produced using the OpenQuake-engine, or other software. In order to load ground motion fields based on a single earthquake event, it is fundamental to ensure that the ground motion values have been stored according to the appropriate NRML schema, as presented in Section 4.3.3 . Then, the following command can be used:

```
user@ubuntu:~$ oq-engine --load-gmf <gml_directory>
```

Whether a user chooses to load pre-computed ground motion fields, or calculate this input using the hazard component of the OpenQuake-engine, a unique id is associated to the set of ground motion fields, as depicted below.

```
Calculation 3 results:  
id | output_type | name  
12 | gmf_scenario | gmf_scenario
```

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-risk job.ini --hazard-output-id
<hazard_output_id>
```

or simply:

```
user@ubuntu:~$ oq-engine --rr job.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-risk job.ini --hazard-calculation-id
<hazard_calculation_id>
```

or simply:

```
user@ubuntu:~$ oq-engine --rr job.ini --co <hazard_calculation_id>
```

For further information about consulting the id of hazard results or calculations, users are referred to section 6.2. 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-risk-outputs <risk_calculation_id>
```

or simply:

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

Which will display a list of risk calculations 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 the risk calculation outputs in the appropriate xml format, the following command can be used.

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

or simply:

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

8.2 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.

8.2.1 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

8.2.2 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"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
<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>
```

```
</nrml>
```

- **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"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
<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>
</nrml>
```

- **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.

8.2.3 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/0.4">
  <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/0.4">
  <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/0.4">
  <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

```

8.2.4 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"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
<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>
</nrml>
```

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`).

8.2.5 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"?>
<nrml 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>
</nrml>
```

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).

8.2.6 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/0.4">
```



```

<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.

8.2.7 Loss disaggregation

Currently the loss disaggregation can only be performed using the Probabilistic Event-based Risk calculator. Thus, the parameters `sourceModelTreePath` and `gsmTreePath` 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/0.4">
<lossFraction investigationTime ="50.00" sourceModelTreePath="b1"
      gsmTreePath="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`.

8.2.8 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
```


9. Demonstrative Examples

This section describes the set of demos that have been compiled to exercise the OpenQuake-engine. These demos can be found in a public repository in GitHub at the following link <http://github.com/gem/oq-engine/tree/master/demos>. 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 do not intend 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 benchmark 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.

The five demos use Nepal as the region of interest. An example exposure model has been developed for this region, comprising 9144 assets distributed amongst 2221 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 9.1.

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/>).

This section includes instructions not only on how to run the risk calculations, but also on how to produce the necessary hazard input. Thus, each demo comprises the configuration file, exposure model and fragility/vulnerability models fundamental for the risk calculations, but also

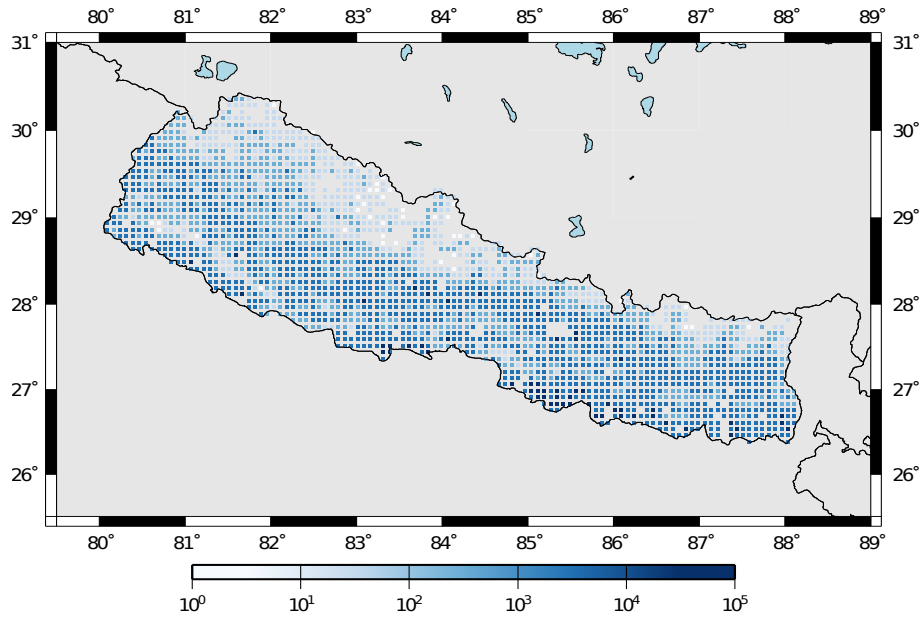


Figure 9.1 – Distribution of number of buildings in Nepal.

a configuration file and associated input models to produce the hazard input.

9.1 Scenario Risk demo

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_haz_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 --rh job_gmfq.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 --rr 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
```

9.2 Scenario Damage demo

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 --rr 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
```

9.3 Classical PSHA-based Risk demo

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 --rh 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 --rr 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
```

9.4 Classical PSHA-based Damage demo

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 9.3 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 --rh 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 --rr 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
```


9.5 Probabilistic Event-based demo

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 --rh 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 --rr 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
```

9.6 Retrofitting Benefit/cost ratio demo

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 (9.3) can be employed here. Then, the risk calculations can be initiated using the following command:

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


Appendices

A. Ground Shaking Intensity Models

We provide below a list of the ground motion prediction equations implemented in the oq-hazardlib. All the implemented GMPE use moment magnitude as the reference magnitude. For each GMPE, the oq-engine name, a short description, and the corresponding reference are given.

A.1 Ground motion prediction equations for shallow earthquakes in active tectonic regions

- AbrahamsonSilva2008
A ground motion prediction equation developed in the context of the NGA West project ¹. The model is applicable to magnitudes 5.0-8.5, distances 0-200 km, and spectral periods 0-10 sec (Abrahamson and Silva, 2008).
- AkkarBommer2010
A ground motion prediction equation developed using mostly data from Europe and the Middle East. The dataset used to derive these equations contains events with moment magnitude between 5.0 and 7.6 and distances up to 100 km (Akkar and Bommer, 2010).
- AkkarCagnan2010
A ground motion prediction equation for shallow earthquakes in active tectonic regions developed using data from the Turkish strong-motion database. Equations are valid for a distance range of 0–200 km and are derived for moment magnitudes between 5.0 and 7.6 (Akkar and Çağnan, 2010).
- BooreAtkinson2008
A ground motion prediction equation for shallow earthquakes in active tectonic regions developed in the context of the NGA West project. The model is applicable to magnitude in the range 5.0-8.0, distances < 200 km, and spectral periods 0-10 (Boore and Atkinson, 2008).

¹<http://peer.berkeley.edu/ngawest>

- **CauzziFaccioli2008**
A ground motion prediction equation derived from global data base of shallow crustal earthquakes (vast majority coming from Japan) with magnitudes in range 5-7.2 and distances < 150.0 (Cauzzi and Faccioli, 2008).
- **ChiouYoungs2008**
A ground motion prediction equation for shallow earthquakes in active tectonic regions developed in the context of the NGA West ². The model is supposed to be applicable for magnitude in range 4.0-8.5 (if strike-slip), 4.0-8.0 (if normal or reverse) and distances 0-200 km.
- **FaccioliEtAl2010**
Based on the same functional form of Cauzzi and Faccioli, 2008 but using closest distance to the rupture instead of hypocentral distance (Faccioli et al., 2010)
- **SadighEtAl1997**
A ground motion prediction based primarily on strong motion data from California and applicable for magnitude in range 4.0-8.0 and distances < 100 km (Sadigh et al., 1997).
- **ZhaoEtAl2006Asc**
A ground motion prediction equation for active shallow crust events developed using mostly japanese strong ground motion recordings (Zhao et al., 2006).

A.2 Ground motion prediction equations for subduction sources

- **AtkinsonBoore2003SInter, AtkinsonBoore2003SSlab**
Ground motion prediction equations for subduction interface and in-slab events obtained using a global dataset of subduction earthquakes with moment magnitude between 5.0 and 8.3 (Atkinson and Boore, 2003).
- **LinLee2008SInter, LinLee2008SSlab**
Ground motion prediction equations for subduction interface and in-slab events created using strong motion data included in the the Taiwanese database (Lin and Lee, 2008).
- **YoungsEtAl1997SInter, YoungsEtAl1997SSlab**
One of the most well known ground motion prediction equations for subduction earthquakes. Published in 1997, is still currently used for the calculation of the ground motion in subduction tectonic environments. This GMPE covers events of moment magnitude greater than 5.0 occurred at a distance between 5 and 500 km. The source-site distance metric is the closest distance between the site and rupture (r_{rup}). (Youngs et al., 1997)
- **ZhaoEtAl2006SInter, ZhaoEtAl2006SSlab**
Ground motion prediction equations for subduction interface and in-slab developed using mostly japanese strong ground motion recordings (Zhao et al., 2006).

² <http://peer.berkeley.edu/ngawest>

A.3 Ground motion prediction equations for stable continental regions

- AtkinsonBoore2006
A ground motion prediction equation for Eastern North America derived from a stochastic finite fault model (Atkinson and Boore, [2006](#)).
- Campbell2003
Ground motion prediction equation calibrated for Eastern North America applicable for events with magnitude greater than 5.0 and distances < 70 km (Campbell and Bozorgnia, [2003](#)).
- ToroEtAl2002
Ground motion prediction equation for rock sites in central and eastern North America based on the prediction of a stochastic ground-motion model. The model is applicable for magnitudes in range 5.0-8.0 and distances in 1-500 km (Toro, [2002](#)).

Bibliography

Books

Aki, K. and P. G. Richards (2002). *Quantitative Seismology*. Sausalito, California: University Science Books (cited on pages 29, 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 (cited on page 133).

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/gssr1.81.2.195](https://doi.org/10.1785/gssr1.81.2.195) (cited on page 133).

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 (cited on page 133).

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 (cited on page 134).

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 (cited on page 135).

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 (cited on page 133).

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 (cited on page 135).
- 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 (cited on page 134).
- 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 48, 127).
- 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 29).
- Cornell, C. A. (1968). “Engineering Seismic Risk Analysis”. In: *Bulletin of the Seismological Society of America* 58, pages 1583–1606 (cited on page 35).
- 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 page 35).
- Frankel, A. (1995). “Mapping Seismic Hazard in the Central and Eastern United States”. In: *Seismological Research Letters* 66.4, pages 8–21 (cited on page 26).
- 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 (cited on page 134).
- 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 (cited on page 134).
- 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 28).
- 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 35).
- 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 26).
- 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 (cited on page 134).
- 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) (cited on page 134).

Other Sources

- 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* (cited on page 134).
- McGuire, K. K. (1976). *FORTTRAN computer program for seismic risk analysis*. Open-File report 76-67. 102 pages. United States Department of the Interior, Geological Survey (cited on page 35).
- 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 26).
- 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 (cited on page 135).

A

Area source *see* Source type

C

Characteristic fault *see* Source type
Complex fault *see* Source type

G

Grid source *see* Source type

I

Input

- Configuration file 47, 48
- Ground motion logic tree 47
- Ground motion system 47

O

OpenQuake-engine

- hazard 23
- Hazard calculation workflows 35

- Classical PSHA, 36
- Event-based PSHA, 36
- Scenario-based SHA, 37

P

Point source *see* Source type

R

Running OpenQuake

- hazard 55
- introduction 15

S

Simple fault *see* Source type

Source type 23

area 27

fault

- characteristic, 32
- complex geometry, 31
- simple geometry, 28

grid 26

point 24

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 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..

dip

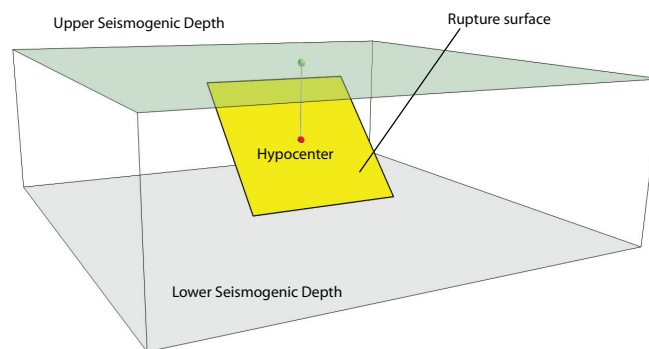
The dip is the steepest angle of descent of the fault plane relative to a horizontal plane; it is measured in degrees $[0,90]$.

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..

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

See magnitude-frequency distribution..

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)..

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..

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..

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..

