



OPENQUAKE

calculate share explore

OPENQUAKE ENGINE USER INSTRUCTION MANUAL

Version 1.0.0 Beta

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



“OpenQuake: Calculate, share, explore”

OpenQuake Engine User Instruction Manual

Authors:

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

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

Disclaimer

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

The current version of the manual has been revised only by members of the GEM model facility and it must be considered a draft copy.

License

This Manual is distributed under the Creative Common License Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0) (see link below). You can download this Book and share it with others as long as you provide proper credit, but you cannot change it in any way or use it commercially

<http://creativecommons.org/licenses/by-nc-nd/3.0/>

Contents

I Introduction	7
1. OpenQuake engine background	9
1.1. OpenQuake-engine introduction	9
1.2. Running the OpenQuake-engine	9
II Hazard	13
2. Introduction to the hazard module	15
2.1. Source typologies	15
2.1.1. Source typologies for modelling distributed seismicity	16
2.1.1.1. Point sources	16
2.1.1.2. Grid sources	19
2.1.1.3. Area sources	20
2.1.2. Fault source with floating ruptures	21
2.1.2.1. Simple faults	22
2.1.2.2. Complex faults	23
2.1.3. Fault source types without floating ruptures	25
2.1.3.1. Characteristic faults	25
2.1.4. Supported magnitude-frequency distributions	25
2.2. Calculation workflows	28
2.2.1. Classical Probabilistic Seismic Hazard Analysis	29
2.2.2. Event-Based Probabilistic Seismic Hazard Analysis	29
2.2.3. Scenario based Seismic Hazard Analysis	30

Contents

3. Using the hazard module	31
3.1. Input Data definition	31
3.1.1. Defining Logic Trees in the OpenQuake engine	31
3.1.1.1. Logic trees as described in the nrml schema	34
3.1.2. The Seismic Source System	38
3.1.2.1. The Seismic Source Logic Tree	38
3.1.2.2. The Seismic Source Model	40
3.1.3. The Ground Motion System	41
3.1.3.1. The Ground Motion Logic Tree	41
3.1.4. Configuration file	42
3.1.4.1. Calculation of a hazard map and hazard curves using the classical PSHA methodology	42
3.1.4.2. Seismic hazard disaggregation	46
3.1.4.3. Event based PSHA	47
4. Hazard calculation and results provided	49
4.1. Running OpenQuake-engine for hazard calculations	49
4.1.1. Getting results	50
4.2. Description of outputs	51
4.2.1. Output from Classical PSHA	51
4.3. Output from Event Based PSHA	54
4.4. Output from Disaggregation	56
5. Demonstrative examples	59
5.1. OpenQuake Hazard Demos	59
5.1.1. Classical PSHA Demos	60
5.1.1.1. Classical PSHA with different source typologies	60
5.1.1.2. Classical PSHA with non trivial logic trees	62
5.1.1.3. Event Based PSHA	72
5.1.1.4. Disaggregation	73
III Risk	77
6. Introduction to the risk module	79
6.1. Introduction	79
6.2. Calculation workflows	79
6.2.1. Scenario Risk Calculator	80
6.2.2. Scenario Damage Calculator	80
6.2.3. Probabilistic Event-based Risk Calculator	81
6.2.4. Classical PSHA-based Risk Calculator	82

6.2.5. Retrofitting Benefit/Cost Ratio Calculator	82
7. Using the risk module	85
7.1. Input data definition	85
7.1.1. Exposure model definition	85
7.1.2. Physical vulnerability model definition	93
7.1.3. Fragility model definition	97
7.1.4. Configuration file	101
7.1.4.1. Scenario Risk Calculator	102
7.1.4.2. Scenario Damage Calculator	102
7.1.4.3. Probabilistic Event-based Risk Calculator	103
7.1.4.4. Classical PSHA-based Risk Calculator	103
7.1.4.5. Retrofitting Benefit/Cost Ratio Calculator	104
8. Risk calculations and results	107
8.1. Running OpenQuake engine for risk calculations	107
8.2. Description of the outputs	110
8.2.1. Loss statistics	110
8.2.2. Loss maps	110
8.2.3. Damage distribution	113
8.2.4. Collapse maps	115
8.2.5. Loss exceedance curves	116
8.2.6. Retrofitting Benefit/cost ratio maps	117
9. Demonstrative examples	121
9.1. Scenario Risk demo	122
9.2. Scenario Damage demo	123
9.3. Classical PSHA-based Risk demo	124
9.4. Probabilistic Event-based demo	124
9.5. Retrofitting Benefit/cost ratio demo	125
IV Appendixes	127
A. Supported ground motion prediction equations	129
A.1. Ground motion prediction equations for shallow earthquakes in active tectonic regions	129
A.2. Ground motion prediction equations for subduction sources	130
A.3. Ground motion prediction equations for stable continental regions	131

Contents

B. Supported magnitude-scaling relationships	133
B.1. Magnitude-scaling relationships for shallow earthquakes in active tectonic regions	133

PART I

Introduction

CHAPTER 1

OpenQuake engine background

1.1. OpenQuake-engine introduction

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 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://gitub.com/gem/oq-engine>.

1.2. Running the OpenQuake-engine

The execution of an analysis using the OpenQuake-engine 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:~$ openquake --help
```

The result is the following:

```
usage: openquake [-h] [--version] [--log-file LOG_FILE]
                  [--log-level debug,info,progress,warn,error,critical]
                  [--no-distribute] [--list-inputs INPUT_TYPE] [--yes]
                  [--run-hazard CONFIG_FILE] [--list-hazard-calculations]
                  [--list-hazard-outputs HAZARD_CALCULATION_ID]
```

1. OpenQuake engine background

```
[--export-hazard OUTPUT_ID TARGET_DIR]
[--delete-hazard-calculation HAZARD_CALCULATION_ID]
[--run-risk CONFIG_FILE] [--hazard-output-id HAZARD_OUTPUT]
[--hazard-calculation-id HAZARD_CALCULATION_ID]
[--list-risk-calculations]
[--list-risk-outputs RISK_CALCULATION_ID]
[--export-risk OUTPUT_ID TARGET_DIR]
[--delete-risk-calculation RISK_CALCULATION_ID]
[--exports xml]

OpenQuake Seismic Hazard and Risk Analysis Engine

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

General:
  --version             Display version information
  --log-file LOG_FILE, -L LOG_FILE
                        Location to store log messages; if not specified, log
                        messages will be printed to the console (to stderr)
  --log-level debug,info,progress,warn,error,critical, -l debug,info,progress,warn,error,critical
                        Defaults to "progress"
  --no-distribute, --nd
                        Disable calculation task distribution and run the
                        computation in a single process. This is intended for
                        use in debugging and profiling.
  --list-inputs INPUT_TYPE, --li INPUT_TYPE
                        List inputs of a specific input type
  --yes, -y              Automatically answer "yes" when asked to confirm an
                        action

Hazard:
  --run-hazard CONFIG_FILE, --rh CONFIG_FILE
                        Run a hazard job with the specified config file
  --list-hazard-calculations, --lhc
                        List hazard calculation information
  --list-hazard-outputs HAZARD_CALCULATION_ID, --lho HAZARD_CALCULATION_ID
                        List outputs for the specified hazard calculation
  --export-hazard OUTPUT_ID TARGET_DIR, --eh OUTPUT_ID TARGET_DIR
                        Export the desired output to the specified directory
  --delete-hazard-calculation HAZARD_CALCULATION_ID, --dhc HAZARD_CALCULATION_ID
                        Delete a hazard calculation and all associated outputs

Risk:
  --run-risk CONFIG_FILE, --rr CONFIG_FILE
                        Run a risk job with the specified config file
  --hazard-output-id HAZARD_OUTPUT, --ho HAZARD_OUTPUT
                        Use the desired hazard output as input for the risk
                        job
  --hazard-calculation-id HAZARD_CALCULATION_ID, --hc HAZARD_CALCULATION_ID
                        Use the desired hazard calculation as input for the
                        risk job
  --list-risk-calculations, --lrc
                        List risk calculation information
  --list-risk-outputs RISK_CALCULATION_ID, --lro RISK_CALCULATION_ID
                        List outputs for the specified risk calculation
  --export-risk OUTPUT_ID TARGET_DIR, --er OUTPUT_ID TARGET_DIR
                        Export the desired risk output to the specified
                        directory
  --delete-risk-calculation RISK_CALCULATION_ID, --drc RISK_CALCULATION_ID
                        Delete a risk calculation and all associated outputs
```

List and export:
--exports xml Automatically export all calculation results to the
 specified format. Only "xml" is supported currently.

PART II

Hazard

CHAPTER 2

Introduction to the hazard module

The hazard component of the OpenQuake-engine builds on top of the [Open-Quake hazard library \(oq-hazardlib\)](#), a python-based library containing tools for PSHA calculation. The web repository of this library is available at the following address: <http://gitub.com/gem/oq-hazardlib>.

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

2.1. Source typologies

A general [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 - essential for specifying source geometry and seismicity occurrence characteristics. 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 basically two 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 vari-

2. Introduction to the hazard module

able seismicity occurrence properties.

- Fault sources with floating ruptures:
 - **Simple fault source** - The simplest fault model in OpenQuake. This typology is habitually used to describe shallow seismogenic faults.
 - **Complex fault source** - Often used to model subduction interface sources with a complex geometry.
- Fault sources with ruptures always covering the entire fault surface:
 - **Characteristic fault source** - A typology of source where ruptures always fill the entire fault surface.

In the OpenQuake-engine there are some basic assumptions accepted in the definition of these source typologies such as:

- 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

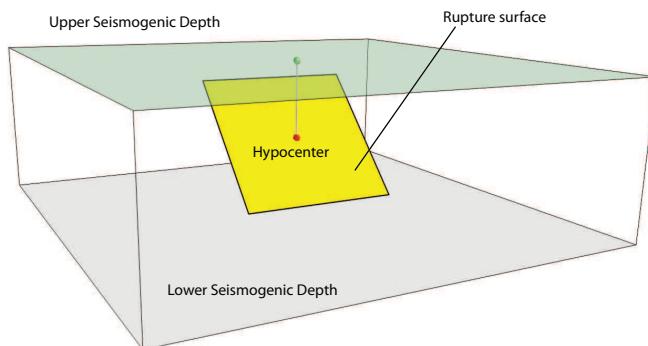


Figure 2.1. – Single rupture

Point source is the elemental source type adopted in the OpenQuake engine to model distributed seismicity. These are the basic assumptions used to generate

2.1. Source typologies

finite ruptures with point sources:

- ruptures have a rectangular shape
- rupture's 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 each point source (i.e. grid node) the following parameters are requested (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 for a range of magnitudes. Each rupture is centered at the only hypocenter 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.

Below we provide the excerpt of an .xml file used to describe the properties of a point source included in a seismic source model.

```
1 <pointSource id="1" name="point" tectonicRegion="Stable Continental Crust">
2   <pointGeometry>
```

2. Introduction to the hazard module

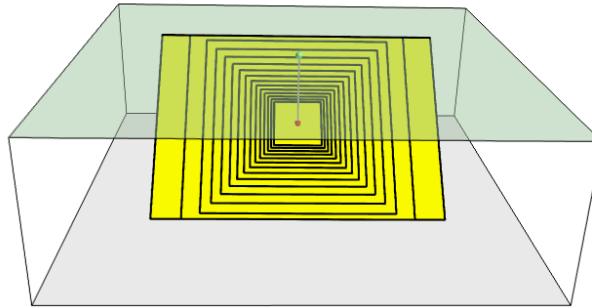


Figure 2.2. – Point source with multiple ruptures. Note the change in the aspect ratio once the rupture width fills the entire seismogenic layer.

```

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 in the sample xml above shows the the parameter used to describe the geometry of the area source, the blue part is the description of the magnitude-frequency distribution, the green text shows the nodal plane distribution and the magenta one the hypocentral depth distribution. The text in black describes paramters needed to generate the ruptures such as the **magnitude-scaling relationship** and the aspect ratio.

In this example, ruptures occur on two possible nodal planes and two hypocentral depths. Figure 2.3 shows all the possible ruptures generated by the point

source specified above.

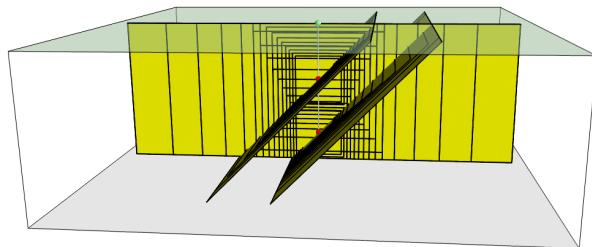


Figure 2.3. – Ruptures produced by the source created using the information in the example .xml file described at page 17.

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. corollary Grid source is a typology used to reproduce more faithfully the spatial variability of the seismicity occurrence process; usually only events of low and intermediate magnitudes are considered. Grid sources are generally computed using seismicity smoothing algorithms [??, 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, some smoothing algorithms require an a-priori definition of some setup parameters that expose the calculation to a certain degree of partiality.

Grid sources are modelled in **oq-engine** simply as a set of point sources; in other words, a grid source is just a long list point sources specified as described in the previous section (see page 19).

2. Introduction to the hazard module

2.1.1.3. Area sources

Area sources 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 tuples) The current version of OQ doesn't support the definition of internal borders. [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].

Below we provide the excerpt of an .xml file used to describe the properties of an area source included in a seismic source model.

```
1 <areaSource id="1" name="Quito" tectonicRegion="Active Shallow Crust">
2   <areaGeometry>
```

2.1. Source typologies

```
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 part in the sample xml above shows the the parameter used to describe the geometry of the area source, the blue part is the description of the magnitude-frequency distribution, the green text shows the nodal plane distribution and the magenta one the hypocentral depth distribution. The text in gray describes paramters needed to generate the ruptures such as the **magnitude-scaling relationship** and the aspect ratio.

The ruptures generated inside this area source follow to possible nodal planes and have hypocenters at two depths.

2.1.2. Fault source with floating ruptures

Fault sources in the **oq-engine** are classified according to the method adopted to distribute ruptures over the fault surface. Two are the option currently sup-

2. Introduction to the hazard module

ported:

- 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 almost all the possible magnitude-frequency distributions.
- With the second option, ruptures always fill the entire fault surface. This model is compatible mainly with characteristic models (à la ?).

In this Section we discuss fault source types that support floating ruptures in the following one we will illustrate the typology of fault 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 basically obtained by projecting a 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's 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 ?) [degrees];
- Rake angle (specified following the Aki-Richards convention; see ?) [degrees];
- Upper and lower depth values limiting the seismogenic interval [km];

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

2.1. Source typologies

```
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 </simpleFaultSource>
```

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

2.1.2.2. Complex faults

Complex fault differ from simple fault just by the way the geometry of the fault surface is defined and by the process 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 - realistic listric fault sources.

2. Introduction to the hazard module

```
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>
```

2.1. Source typologies

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

2.1.3. Fault source types without floating ruptures

2.1.3.1. Characteristic faults

The characteristic fault source is a particular typology of fault created following the assumption that ruptures will always cover the entire fault surface.

In this case, the fault surface can be represented as a **simple fault source** surface or a **complex fault source** surface or 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 ?)
- Upper and lower depth values limiting the seismogenic interval.

2.1.4. Supported magnitude-frequency distributions

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

A discrete incremental magnitude-frequency distribution

It's the simplest distribution supported. It's defined by the minimum value of magnitude (representing the mid point of the first bin) and by the

2. Introduction to the hazard module

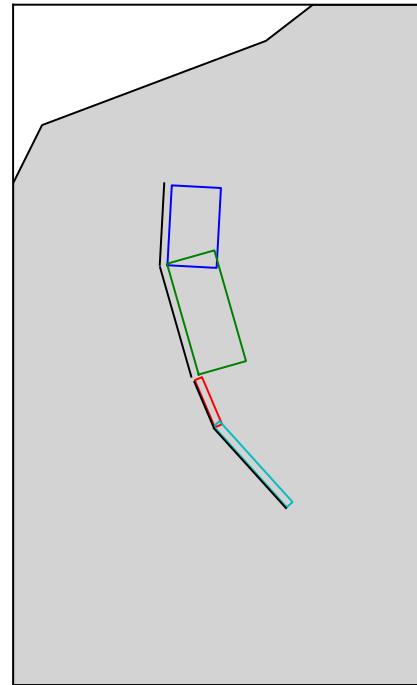


Figure 2.4. – Geometry of a multi-segmented characteristic fault composed of four rectangular ruptures as modelled in OpenQuake.

bin width. The distribution itself is simply a sequence of floats describing the annual number of events for different bins.

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 [23](#).

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

This is the magnitude-frequency distribution obtained with the above settings is represented in Figure [2.5](#).

A double truncated Gutenberg-Richter distribution

2.1. Source typologies

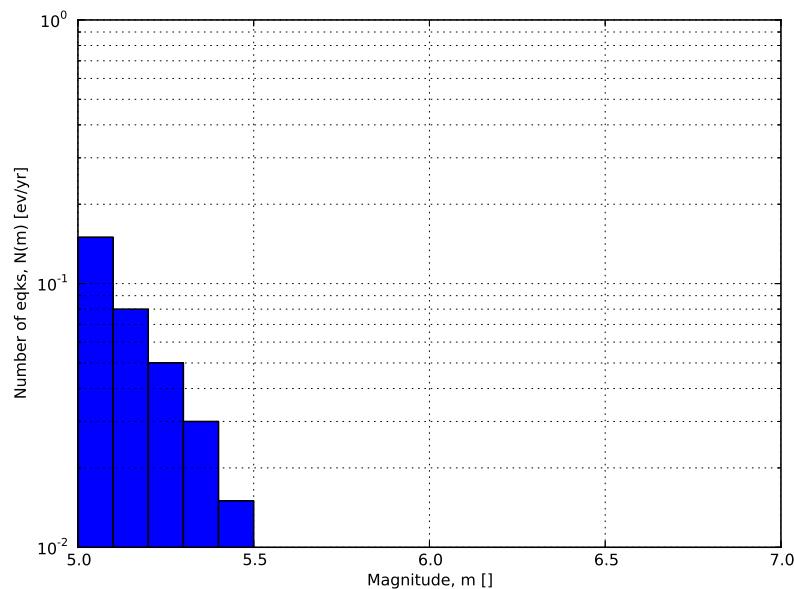


Figure 2.5. – Example of an incremental magnitude-frequency distribution.

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 necessary to specify the annual rate of occurrence equal or larger than a given magnitude value.

The synthax 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 the magnitude-frequency distribution obtained using the parameters of the considered example.

2. Introduction to the hazard module

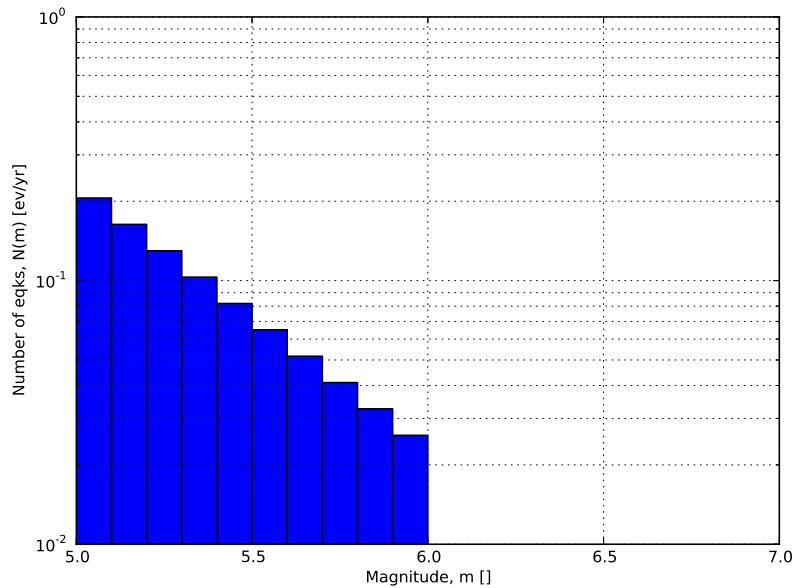


Figure 2.6. – Example of a double truncated Gutenberg-Richter magnitude-frequency distribution.

2.2. Calculation workflows

The hazard component of the OpenQuake-engine can compute seismic hazard following 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 (? , ?) as formulated by ? .
- *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 analysed. Each calculator can be extended independently of the

2.2. Calculation workflows

others so that additional calculation options and methodologies can be easily introduced, without affecting the overall calculation workflow.

2.2.1. Classical Probabilistic Seismic Hazard Analysis

Input data for the classical PSHA consist of an Input Model provided together with a set of calculation settings.

The main calculators used to perform this analysis are:

1. *Logic Tree Processor*

The Logic Tree Processor (LTP) takes as an input the **Probabilistic Seismic Hazard Analysis (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 - i.e. sampling epistemic uncertainties - 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 is then used as input for the Earthquake Rupture Forecast (ERF) calculator which creates a list earthquake rupture produced 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.

2.2.2. Event-Based Probabilistic Seismic Hazard Analysis

Input data for the Event-Based PSHA - as in the case of the Classical **PSHA** calculator - consist 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 2.2.1 at page 29).

2. *Introduction to the hazard module*
 2. *Earthquake Rupture Forecast Calculator*
The Earthquake Rupture Forecast Calculator was already introduced in the description of the PSHA workflow (see section [2.2.1](#) at page [29](#)).
 3. *Stochastic Event Set Calculator*
The Stochastic Event Set Calculator generates a collection of Stochastic Event sets by sampling each rupture contained in the ERF according to its probability of occurrence. A Stochastic Event Set (SES) represents a potential realisation of the seismicity (i.e. a list of ruptures) over the time span fixed for the calculation of hazard (normally corresponding to 50 years) by the set of seismic sources considered in the analysis.
 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.

2.2.3. Scenario based Seismic Hazard Analysis

In case of **SSHA**, the input data consist of a single earthquake rupture model and a single ground-motion model. 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 [2.2.2](#) at page [29](#)).

CHAPTER 3

Using the hazard module

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

3.1. Input Data definition

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 files.

3.1.1. Defining Logic Trees in the OpenQuake engine

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

3. Using the hazard module

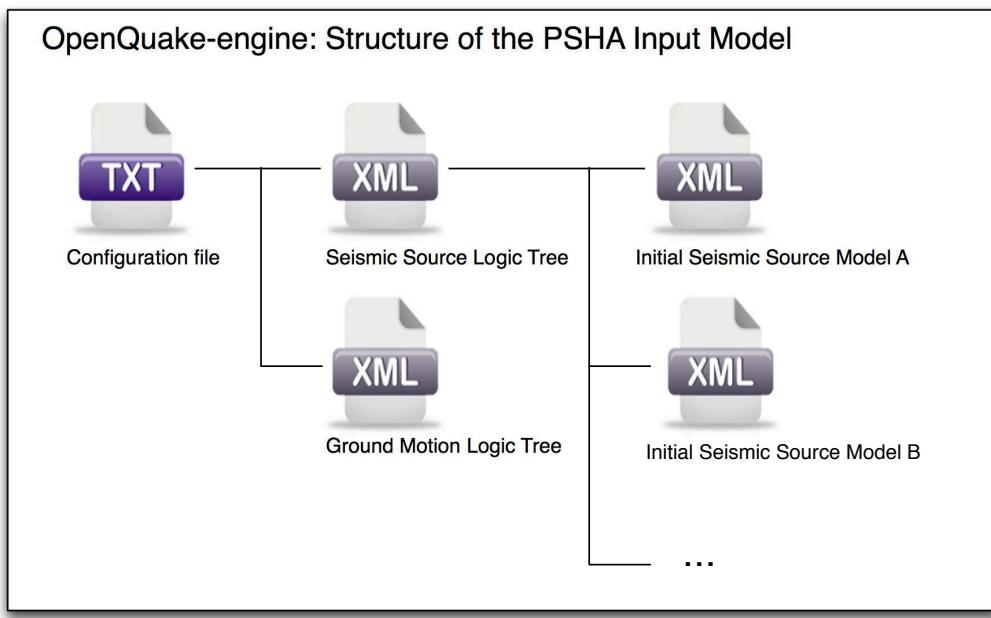


Figure 3.1. – *PSHA Input Model structure*

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

branching set : it's 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 [35](#))
- One or more branches

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

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

3.1. Input Data definition

Below we provide a simple schema illustrating the skeleton of the **oq-engine** logic tree:

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

As it clearly appears from this example the structure of a logic tree is a set nested elements, from the largest to the smallest containers.

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

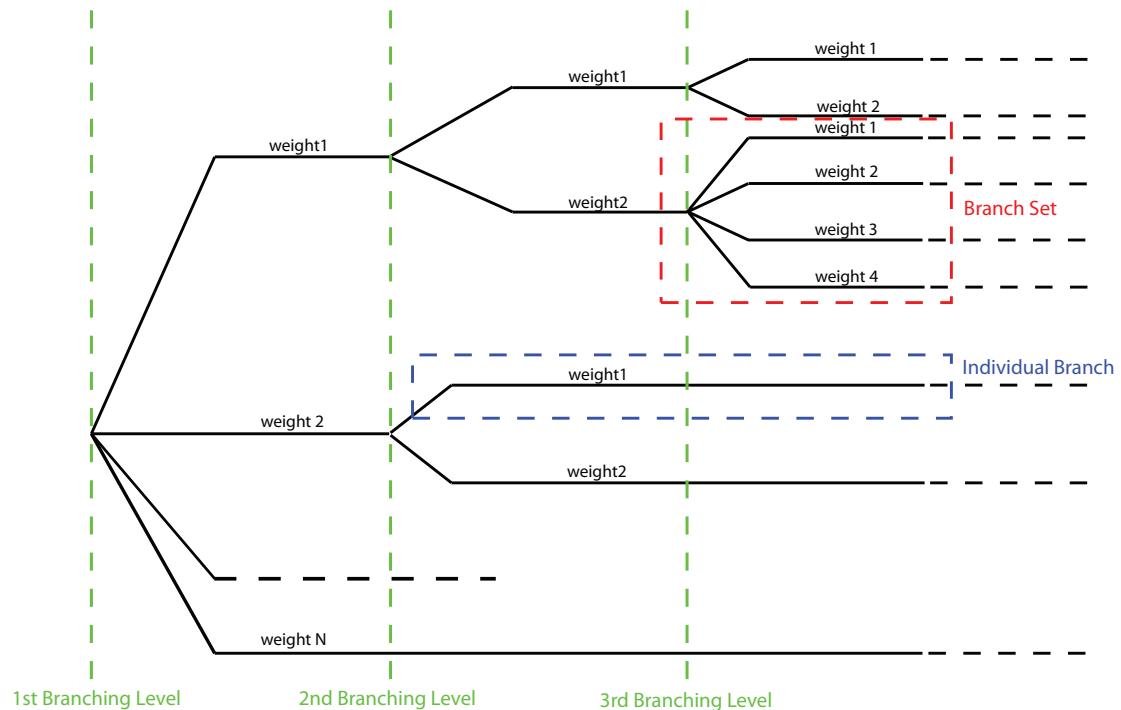


Figure 3.2. – Generic Logic Tree structure as described in terms of branching levels, branch sets, and individual branches.

3. Using the hazard module

3.1.1.1. Logic trees as described in the nrml schema

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

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

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

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

No restrictions are present on the number of tree levels that can be defined.

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

A branch set has two required attributes branchSetID and uncertaintyType (defining the type of epistemic uncertainty the branch set is modelling)

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

3.1. Input Data definition

```
</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: identifying epistemic uncertainties on ground motion prediction equations
- sourceModel: identifying epistemic uncertainties on source models
- maxMagGRRelative: identifying 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: identifying relative epistemic uncertainties to be applied to the Gutenberg-Richter b value.
- abGRAbsolute: identifying absolute (i.e. values used to replace original values) epistemic uncertainties on the Gutenberg-Richter a and b values.
- maxMagGRAbsolute: identifying (absolute) epistemic uncertainties on the Gutenberg-Richter maximum magnitude.

No restrictions are given 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">
    ...
```

3. Using the hazard module

```
<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 at page 129), 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

3.1. Input Data definition

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="abGРАbsolute", the uncertainty model must contain one a and b pair:

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

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

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

No restrictions are given 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.

3. Using the hazard module

- `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"
```

3.1.2. The Seismic Source System

The Seismic Source System contains 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.

3.1.2.1. 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**. This consists of a logic tree with just one branching level containing one **branch set** with one branch used to define the initial seismic source model (its weight will be equal to one).

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <nrm1 xmlns:gml="http://www.opengis.net/gml"
3     xmlns="http://openquake.org/xmlns/nrml/0.4">
```

3.1. Input Data definition

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

The optional branching levels will contain rules that modify parameters of the sources in the initial seismic source model so as to take into account the epistemic uncertainties.

For example, if the principal 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="bl1">
7       <logicTreeBranchSet uncertaintyType="sourceModel"
8         branchSetID="bs1">
9         <logicTreeBranch branchID="b1">
10        <uncertaintyModel>seismic_source_model_A.xml
11        </uncertaintyModel>
12        <uncertaintyWeight>0.2</uncertaintyWeight>
13      </logicTreeBranch>
14      <logicTreeBranch branchID="b2">
15        <uncertaintyModel>seismic_source_model_B.xml
16        </uncertaintyModel>
17        <uncertaintyWeight>0.3</uncertaintyWeight>
18      </logicTreeBranch>
```

3. Using the hazard module

```
19      <logicTreeBranch branchID="b3">
20          <uncertaintyModel>seismic_source_model_C.xml
21          </uncertaintyModel>
22          <uncertaintyWeight>0.5</uncertaintyWeight>
23      </logicTreeBranch>
24  </logicTreeBranchSet>
25 </logicTreeBranchingLevel>

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

40
41 </logicTree>
42 </nrml>
```

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

3.1.2.2. 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 as demonstrated in the following schematic example.

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

3.1. Input Data definition

```
...
<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>
```

3.1.3. The Ground Motion System

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

3.1.3.1. 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 fairly 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 ? **GMPE**.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <nrm1 xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
```

3. Using the hazard module

```
4   <logicTree logicTreeID='lt1'>
5     <logicTreeBranchingLevel branchingLevelID="bl1">
6       <logicTreeBranchSet uncertaintyType="gmpeModel"
7         branchSetID="bs1"
8         applyToTectonicRegionType="Active Shallow Crust">
9
10      <logicTreeBranch branchID="b1">
11        <uncertaintyModel>
12          ChiouYoungs2008
13        </uncertaintyModel>
14        <uncertaintyWeight>1.0</uncertaintyWeight>
15      </logicTreeBranch>
16
17    </logicTreeBranchSet>
18  </logicTreeBranchingLevel>
19 </logicTree>
20 </nrml>
```

3.1.4. 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.

3.1.4.1. Calculation of a hazard map and hazard curves using the classical PSHA methodology

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:

3.1. Input Data definition

- `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`. We'll describe alternative options later on.
- `random_seed`: is used to control the random generator so that when montecarlo procedures are used calculations are replicable (if the same `random_seed` is used).

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

This section is used to specify where hazard will be computed. Two option are available.

The first one consists on defining a polygon (usually a rectangle) and a distance (in km) 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 # km
8 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
7
8
```

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

```
5 [geometry]
6 sites_csv = <name_of_the_csv_file>
7
8
```

- **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 realisations.

3. Using the hazard module

In the second option all the possible realisations are created. Below we provide an example for the latter option. In this example 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.

- **Parameters controlling the construction of the earthquake rupture forecast**

```
11 [erf]
12 # km
13 rupture_mesh_spacing = 5
14 width_of_mfd_bin = 0.1
15 # km
16 area_source_discretization = 10
```

This section of the configuration file is used to specify the level of discretization of the mesh representing faults, of the grid used to delineate the area sources and, of the magnitude-frequency distribution. Note that the lower is the mesh spacing (or the bin width) the higher are the precision in the representation and 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 soil conditions). Alternatively it's possible to define spatially variable soil properties in a separate file; the engine will then assign to each investigation site the appropriate characteristics.

```
17 [site_params]
18 site_model_file = ../_site_model/site_model.xml
```

3.1. Input Data definition

```
19  
20  
21
```

The file containing the site model has the following structure:

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

- **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 # years  
26 investigation_time = 50.0  
27 intensity_measure_types = PGA, SA(0.1)  
28 intensity_measure_types_and_levels = {"PGA": [0.005, ..., 2.13]}  
29 truncation_level = 3  
30 # km  
31 maximum_distance = 200.0
```

This section of the **oq-engine** configuration file specifies the parameters

3. Using the hazard module

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 = out/
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 typology of output to be produced.

3.1.4.2. 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'll use the description given in Section 3.1.4.1 at page 42 as a reference and we'll emphasize in the following 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
```

3.1. Input Data definition

In the section it will be necessary to specify the geographic coordinates of the site (or the 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
# decimal degrees
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 45).

3.1.4.3. Event based PSHA

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

1. **Calculation type and model info**

This part is almost identical to the corresponding one described in section 3.1.4.1. 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
```

2. **Event based**

This is an additional part 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

3. Using the hazard module

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
```

3. Output

This part substitutes the Output part described in configuration file example described in section [3.1.4.1](#) at page [42](#).

```
[output]
export_dir = /tmp/xxx
complete_logic_tree_ses = true
complete_logic_tree_gmf = true
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
```

CHAPTER 4

Hazard calculation and results provided

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 to the use of OpenQuake is provided in Section 1.2 at page 9. The reader is invited to consult this part before moving into the details of this section.

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 which consists in the invocation of `openquake` 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:~$ openquake --rh job.ini
```

The amount of information prompted during the execution of the analysis is controlled by means of the `--log-level` flag as shown in the example below:

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

4. Hazard calculation and results provided

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.

4.1.1. Getting results

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 activated at the OpenQuake engine invocation by the flag --exports xml as shown in the example below

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

The second option, allows the user to export the computed results at their's disposal. In order to get the list of results of the hazard calculations stored in the **oq-engine** database the user can utilize the following command:

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

4.2. Description of outputs

and finally extract an xml file for a specific hazard result

```
user@ubuntu:~$ openquake --eh <result_id> <path_to_the_output_folder>
```

4.2. Description of 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 'distribution' of results (hazard curves, ground motion fields, disaggregation matrices, UHS) which is specular to the 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 included in the hazard module of the OpenQuake-engine calculators (the only exception being disaggregation).

4.2.1. Output 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 will consist on 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 4.1.1 for an explanation about how obtaining the list of calculations performed with their corresponding ID):

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

In this case `oq-engine` produced a group of hazard curves with result ID equal to 3. On the contrary, if the parameter `number_of_logic_tree_samples` in the

4. Hazard calculation and results provided

configuration file is different than zero, N hazard curves files are generated as shown in the example below:

```
user@ubuntu:~$ openquake --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 through the following command

```
user@ubuntu:~$ openquake --eh <output_id> <output_directory>
```

we will 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

4.2. Description of outputs

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 above) contains the values of shaking for which the engine computed the probability of exceedance in the investigation time. For each site this file comprises 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 OpenQuake corresponds to:

```
user@ubuntu:~$ openquake --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 **oq-engine** produced hazard curves and hazard maps for six logic tree realisations plus median hazard curves (highlighted in red) and a median hazard map (also highlighted in red).

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

4. Hazard calculation and results provided

map.

```
<?xml version='1.0' encoding='UTF-8'?>
<nrmrl 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>
</nrmrl>
```

4.3. Output from Event Based PSHA

The Event Based PSHA calculator computes and stores stochastic event sets and the corresponding ground motion fields. In case, this calculator can also compute hazard curves and hazard maps exactly in the same way is done using the Classical PSHA calculator.

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

```
user@ubuntu:~$ openquake --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 | complete_lt_ses | complete logic tree SES
36 | complete_lt_gmf | complete logic tree GMF
37 | hazard_curve | hazard-curve-rlz-19-SA(0.1)
38 | hazard_curve | hazard-curve-rlz-20-SA(0.1)
39 | hazard_curve | hazard-curve-rlz-19-PGA
40 | hazard_curve | hazard-curve-rlz-20-PGA
41 | hazard_curve | mean curve for SA(0.1)
42 | hazard_curve | quantile curve (poe >= 0.15) for imt SA(0.1)
43 | hazard_curve | quantile curve (poe >= 0.5) for imt SA(0.1)
44 | hazard_curve | quantile curve (poe >= 0.85) for imt SA(0.1)
45 | hazard_curve | mean curve for PGA
```

4.3. Output from Event Based PSHA

```
46 | hazard_curve | quantile curve (poe >= 0.15) for imt PGA
47 | hazard_curve | quantile curve (poe >= 0.5) for imt PGA
48 | hazard_curve | quantile curve (poe >= 0.85) for imt PGA
```

This list contains two sets of stochastic events (in red) 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.

In the remaining part of this Section we show an example of a file containing a stochastic event set and a file containing a ground motion field.

This is an example showing a nrml file containing two a collection of stochastic event sets

```
<?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"/>
```

4. Hazard calculation and results provided

```
<bottomRight lon="11.4822820715" lat="42.7256333907"
              depth="12.6791332698"/>
    </planarSurface>
  </rupture>
</stochasticEventSet>
</stochasticEventSetCollection>
</nrml>
```

This is an example showing a nrml file containing a collection of ground motion fields:

```
<?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="11.124373581"
              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="0.000173010769646" lon="11.3782630185"
              lat="44.5"/>
      </gmf>
    </gmfSet>
  </gmfCollection>
</nrml>
```

4.4. Output from Disaggregation

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

4.4. Output from 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:~$ openquake --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.

```
<?xml version='2.0' encoding='UTF-8'?>
<nrmr 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"
    epsBinEdges="-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">
```

4. Hazard calculation and results provided

```
<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,Eps" 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>
```

CHAPTER 5

Demonstrative examples

5.1. OpenQuake Hazard Demos

With the [oq-engine](#) installation a number of demos are provided showing practical examples of input and configuration files definition for different use cases. The following demos illustrate how to use [oq-engine](#) for various seismic hazard analysis:

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

5. Demonstrative examples

5.1.1. 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.

5.1.1.1. Classical PSHA with different source typologies

All the classical PSHA demos illustrating the different source typologies (all demos but the ones about Logic Tree definition) share the following GSIM logic tree file:

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

<nrm1 xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
  <logicTree logicTreeID='lt1'>

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

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

      </logicTreeBranchSet>
    </logicTreeBranchingLevel>

  </logicTree>
</nrm1>
```

5.1. OpenQuake Hazard Demos

Each source in the source model file is defined as belonging to Active Shallow Crust, and therefore the GSIM Logic Tree file defines for simplicity a single GMPE (Boore and Atkinson 2008) which is associated to an Active Shallow Crust tectonic region type.

The configuration file 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
```

5. Demonstrative examples

```
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

[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 show in figure 5.1 and 5.2.

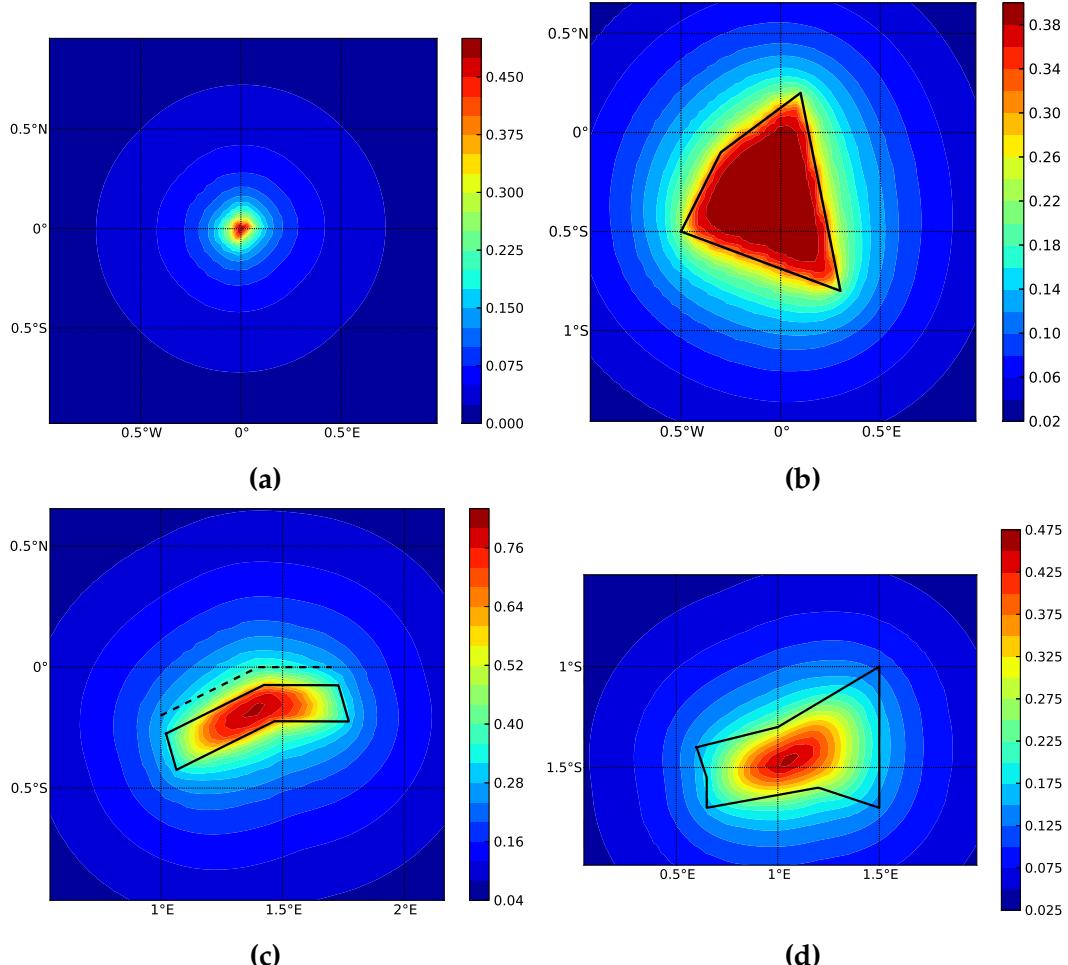
5.1.1.2. Classical PSHA with non trivial logic trees

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"?>
<nrm1 xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
    <logicTree logicTreeID="lt1">
```

5.1. OpenQuake Hazard Demos



5. Demonstrative examples

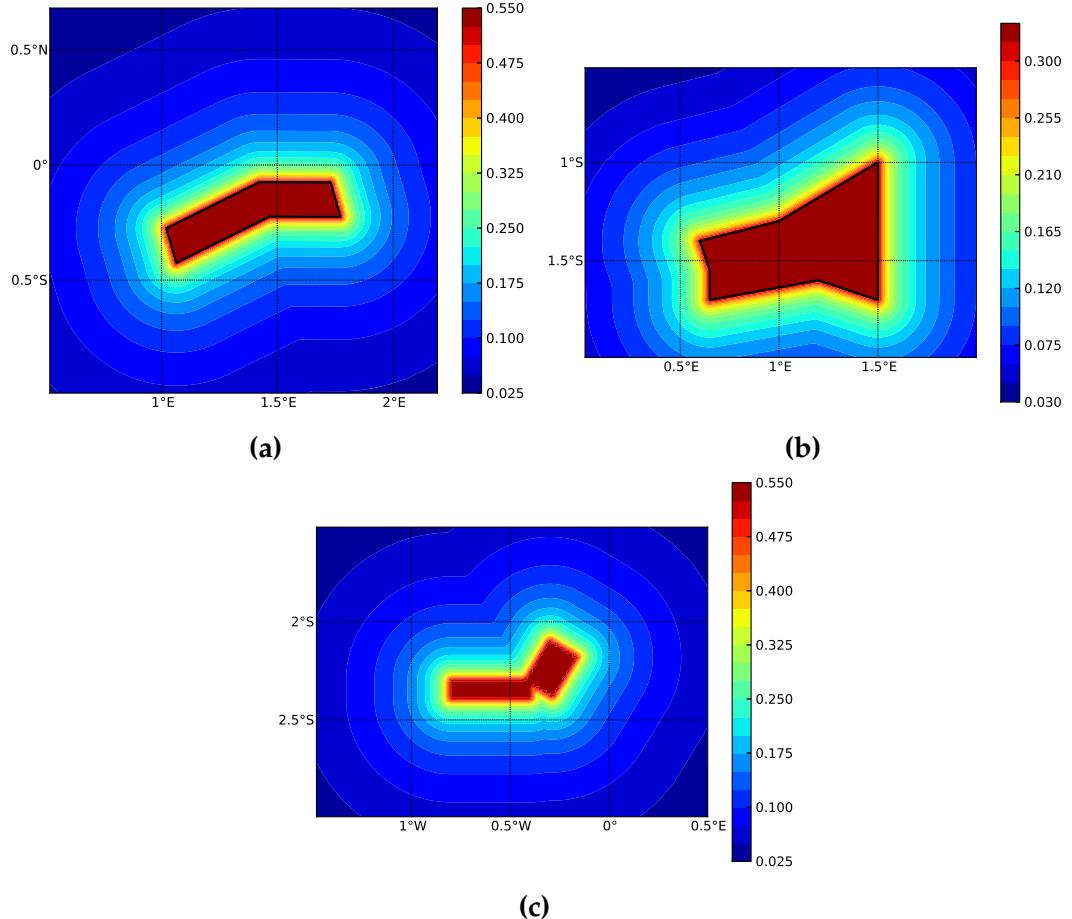


Figure 5.2. – Hazard maps (for PGA, 10% in 50 years) as obtained from characteristic fault sources with simple fault geometry (e), complex fault geometry (f), and collection of planar surfaces (g)

5.1. OpenQuake Hazard Demos

```
<logicTreeBranchingLevel branchingLevelID="bl1">

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

</logicTreeBranchingLevel>

</logicTree>
</nrml>
```

The two source models are defined in two different source model files `source_model_1.xml` and `source_model_2.xml` each associated to the corresponding weight (0.5 in both cases).

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="bl1">
            <logicTreeBranchSet uncertaintyType="gmpeModel"
                applyToTectonicRegionType="Active Shallow Crust"
                branchSetID="bs1">
                <logicTreeBranch branchID="b1">
                    <uncertaintyModel>
                        BooreAtkinson2008
                    </uncertaintyModel>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>
    </logicTree>
</nrml>
```

5. Demonstrative examples

```
</uncertaintyModel>
<uncertaintyWeight>0.5</uncertaintyWeight>
</logicTreeBranch>
<logicTreeBranch branchID="b12">
    <uncertaintyModel>
        ChiouYoungs2008
    </uncertaintyModel>
    <uncertaintyWeight>0.5</uncertaintyWeight>
</logicTreeBranch>
</logicTreeBranchSet>
</logicTreeBranchingLevel>

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

</logicTree>
</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 type 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 obtained

5.1. OpenQuake Hazard Demos

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, b values (three possible pairs for each source) and maximum magnitude (three values for each source) and uncertainties on the GMPEs for each tectonic region type (two GMPE per region type).

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

```
<?xml version="1.0" encoding="UTF-8"?>
<nrm1 xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
  <logicTree logicTreeID="lt1">

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

    <logicTreeBranchingLevel branchingLevelID="bl2">
      <logicTreeBranchSet uncertaintyType="abGRAbsolute"
                           applyToSources="1"
                           branchSetID="bs21">
        <logicTreeBranch branchID="b21">
          <uncertaintyModel>4.6 1.1</uncertaintyModel>
          <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b22">
          <uncertaintyModel>4.5 1.0</uncertaintyModel>
```

5. Demonstrative examples

```
<uncertaintyWeight>0.333</uncertaintyWeight>
</logicTreeBranch>
<logicTreeBranch branchID="b23">
    <uncertaintyModel>4.4 0.9</uncertaintyModel>
    <uncertaintyWeight>0.334</uncertaintyWeight>
</logicTreeBranch>
</logicTreeBranchSet>
</logicTreeBranchingLevel>

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

<logicTreeBranchingLevel branchingLevelID="bl4">
    <logicTreeBranchSet uncertaintyType="maxMagGРАbsolute"
        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>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>
```

5.1. OpenQuake Hazard Demos

```
<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 Logic-TreeCas1ClassicalPSHA. 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.

5. Demonstrative examples

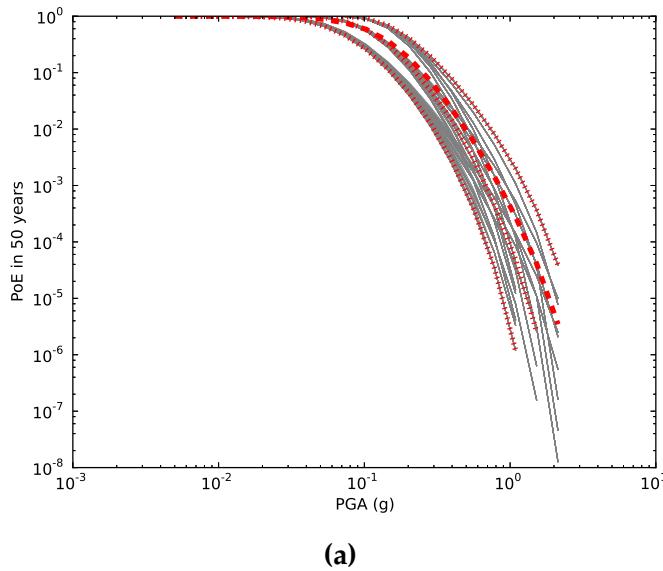


Figure 5.3. – Hazard curves as obtained from the LogicTreeCase2 demo. Solid gray lines represent individual hazard curves from the different logic tree path (a total of 324 curves). The red dashed line represents the mean hazard curve, while the red dotted lines depict the quantile levels (0.15, 0.5,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

LogicTreeCas3ClassicalPSHA 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"?>
<nrmrl 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>
    </logicTree>
```

5.1. OpenQuake Hazard Demos

```
</logicTreeBranch>
</logicTreeBranchSet>
</logicTreeBranchingLevel>

<logicTreeBranchingLevel branchingLevelID="b12">
    <logicTreeBranchSet uncertaintyType="bGRRelative"
        branchSetID="bs21">
        <logicTreeBranch branchID="b21">
            <uncertaintyModel>+0.1</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b22">
            <uncertaintyModel>0.0</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b23">
            <uncertaintyModel>-0.1</uncertaintyModel>
            <uncertaintyWeight>0.334</uncertaintyWeight>
        </logicTreeBranch>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>

<logicTreeBranchingLevel branchingLevelID="b13">
    <logicTreeBranchSet uncertaintyType="maxMagGRRelative"
        branchSetID="bs31">
        <logicTreeBranch branchID="b31">
            <uncertaintyModel>0.0</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b32">
            <uncertaintyModel>+0.5</uncertaintyModel>
            <uncertaintyWeight>0.333</uncertaintyWeight>
        </logicTreeBranch>
        <logicTreeBranch branchID="b33">
            <uncertaintyModel>+1.0</uncertaintyModel>
            <uncertaintyWeight>0.334</uncertaintyWeight>
        </logicTreeBranch>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>
```

5. Demonstrative examples

```
</logicTree>  
</nrml>
```

After the first branching level defining the source model, two additional branching levels are defined, one defining relative uncertainties on b value (bGRRelative applied consistently to all sources in the source model) and the second uncertainties on maximum magnitude (maxMagGRRelative). Similarly 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)

5.1.1.3. Event Based PSHA

A demo showing an example of Event Based 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
```

5.1. OpenQuake Hazard Demos

```
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 = PGA
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`). Corresponding hazard maps for 0.1 probability are additionally calculated (`hazard_maps = true`)

5.1.1.4. Disaggregation

An example of disaggregation calculation is given considering a source model consisting of two sources (area and simple fault) belonging to two different tec-

5. Demonstrative examples

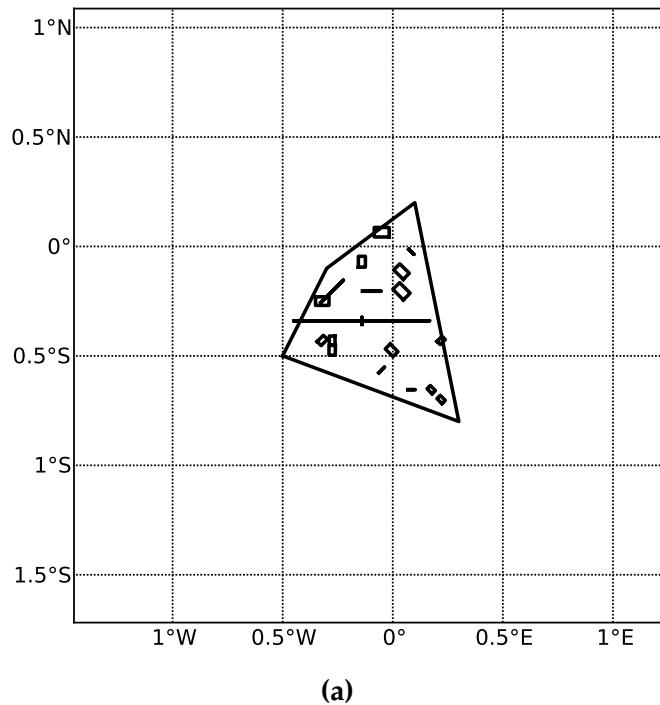


Figure 5.4. – A stochastic event set generated in the event based PSHA demo. The area source defines a nodal plane distribution which distributes events among vertical and dipping (50 degrees) faults with equal weights. Vertical ruptures are then distributed equally in the range 0-180 degrees while the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

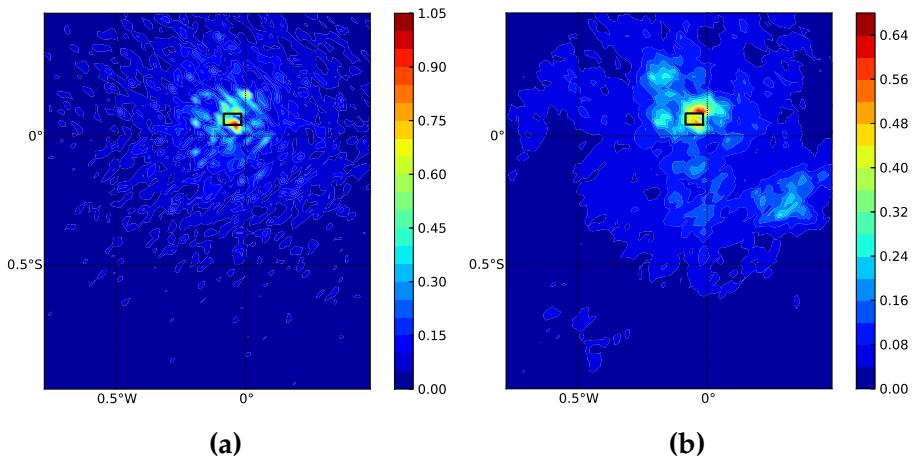


Figure 5.5. – Ground motion fields (PGA) with no spatial correlations (a) and with spatial correlation (b)

5.1. OpenQuake Hazard Demos

tonic region types. The calculation is described by 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]
```

5. Demonstrative examples

```
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).

PART III

Risk

CHAPTER 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://gitub.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. Introduction to the risk module

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-n-rmlib](#)). 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.

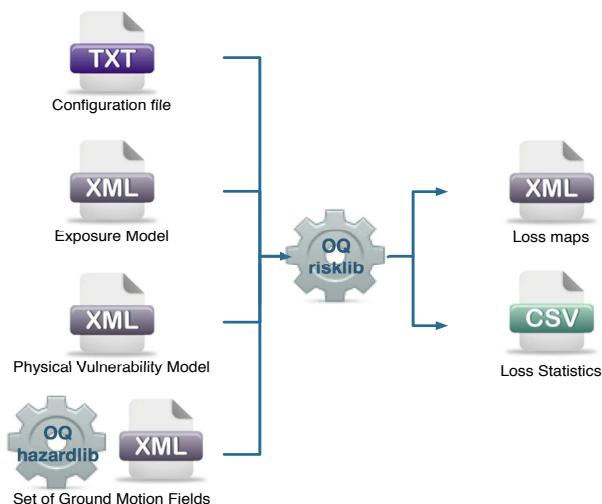


Figure 6.1. – Scenario Risk Calculator input/output structure.

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

6.2. Calculation workflows

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.

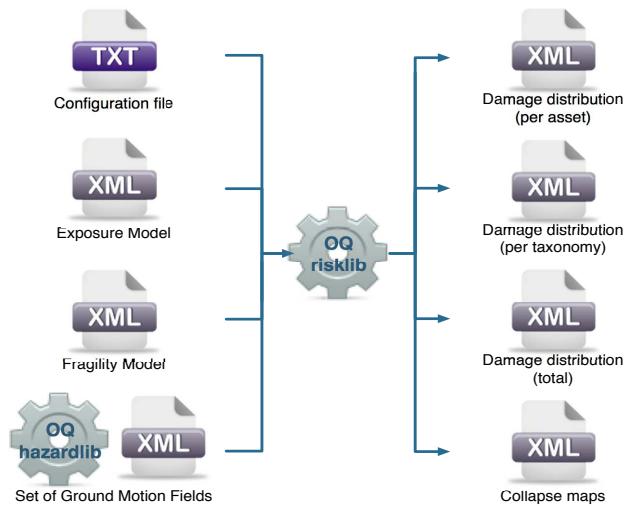


Figure 6.2. – Scenario Damage Calculator input/output structure.

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 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.

6. Introduction to the risk module

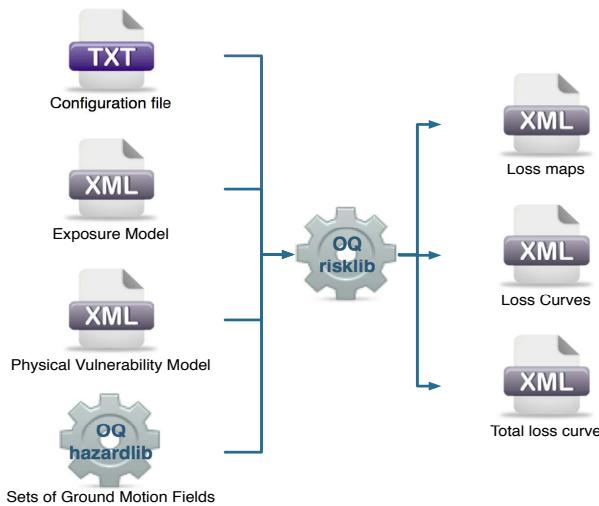


Figure 6.3. – Probabilistic Event-based Risk Calculator input/output structure.

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.

6.2.5. 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

6.2. Calculation workflows

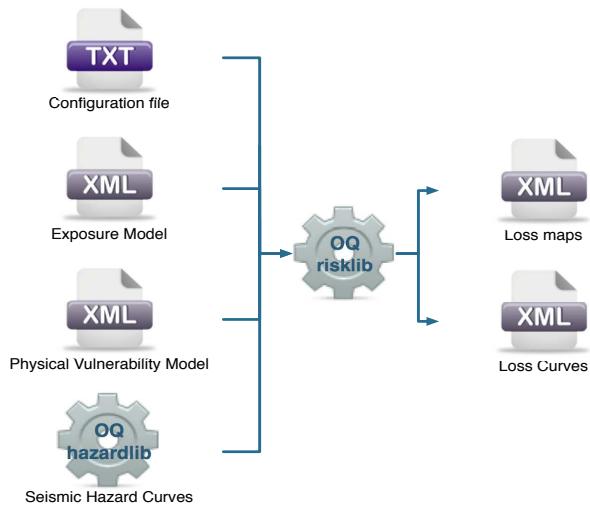


Figure 6.4. – Classical PSHA-based Risk Calculator input/output structure.

by the retrofitting cost, leading to the benefit/cost ratio. These loss curves can be computed using either the previously described Probabilistic Event-based Risk or the Classical PSHA-based Risk calculators. 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 6.5, the input/output structure for this calculator is depicted.

6. Introduction to the risk module

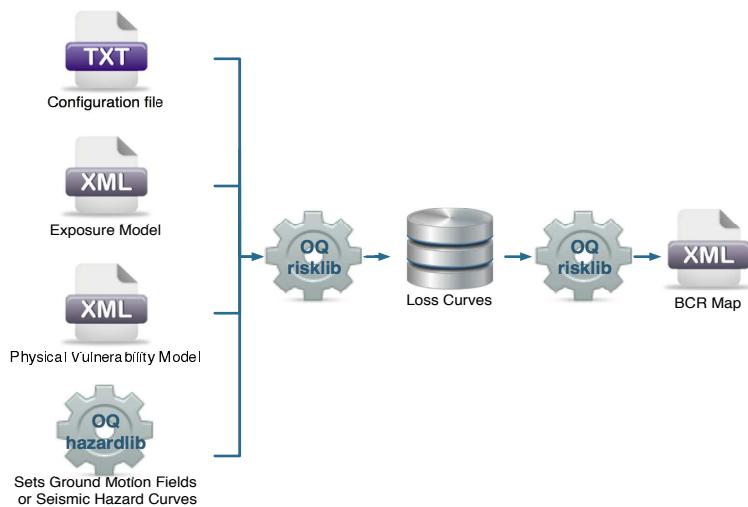


Figure 6.5. – Retrofitting Benefit/Cost Ratio Calculator input/output structure.

CHAPTER 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. The following metadata, that is common to all of the **assets** stored in a given **exposure model**, needs to be provided, as described below:

- **id**: a unique key used to identify the **exposure model**;
- **assetCategory**: a string used to define the type of **assets** being stored (e.g: buildings, population, contents);
- **areaType**: flag defining the way the area is being provided, as explained later in this section;
- **areaUnits**: attribute defining the units used to measure the area;
- **stcoType**: flag defining the way the structural replacement cost is being provided, as explained later in this section;
- **stcoUnits**: attribute defining the units used to measure the structural replacement cost;
- **recoType**: flag defining the way the retrofitting cost is being provided, as explained later in this section;

7. Using the risk module

- recoUnits: attribute defining the units used to measure the retrofitting cost;
- description: brief string with further information about the **exposure model**;
- taxonomySource: attribute used to define the **taxonomy** being used to classify the **assets**;

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 value of each building typology will be directly derived, and will be used for the estimation of economic losses. Thus, the following set of attributes were included in the latest version of 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;
- stco: structural replacement cost of the **asset** at a given location;
- reco: retrofitting cost of the **asset** at a given location.

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 **stco** 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**). This parameter is being handled in the same manner as the structural replacement cost. Thus, it can be defined as the aggregated retrofitting cost, the average cost per unit of area, or as the average cost for a single building unit.

7.1. Input data definition

To establish the way these three attributes are being defined within the **exposure model**, three flags have been introduced in the schema model: areaType, stcoType and recoType.

Note that since the attribute number always represents the same quantity (number of units of a given **asset** in a given location), no flag providing further information about this attribute was introduced. Further information regarding the parameters that are currently being used to define the exposure elements can be found in the OpenQuake Engine Book (Risk). In order to clarify this methodology, several examples are presented later in this section.

Finally, a pair of coordinates (longitude and latitude) needs to be provided to define the location of each asset. 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-nrmllib](#)). Several examples of **exposure models** containing different types of information are presented below. Users will notice that some of the attributes are marked with a "gml" tag, which means that these parameters must follow the Geography Markup Language grammar, developed by the Open Geospatial Consortium.

Example 1

This example is comprised of an **exposure model** in which the aggregated economic value of the buildings of each building typology for a set of locations is directly provided.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrmr xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
<exposureModel gml:id="ep">
    <exposureList gml:id="Nepal_13_1" assetCategory="buildings"
                  stcoUnit="EUR" stcoType="aggregated" recoUnit="EUR"
                  recoType="aggregated">
        <gml:description>Buildings in Nepal</gml:description>
        <taxonomySource>Nepal taxonomy</taxonomySource>
        <assetDefinition gml:id="asset1">
            <site>
                <gml:Point srsName="epsg:4326">
                    <gml:pos>15.22 38.80</gml:pos>
```

7. Using the risk module

```
        </gml:Point>
    </site>
    <taxonomy>URM</taxonomy>
    <stco> 500.0 </stco>
    <reco> 50.0 </reco>
</assetDefinition
...
<assetDefinition gml:id="asset999">
    <site>
        <gml:Point srsName="epsg:4326">
            <gml:pos>15.23 38.95</gml:pos>
        </gml:Point>
    </site>
    <taxonomy>RC</taxonomy>
    <stco> 700.0 </stco>
    <reco> 70.0 </reco>
</assetDefinition>
</exposureList>
</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 site is defined. This position (pos) and the associated geographical projection (srsName) need to be provided in the Point attribute. Each asset must to 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 structural replacement value of the asset is stored in the stco attribute. In this case, the aggregated economic value for all units of 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.

Example 2

This example is comprised of an **exposure model** containing the number of units (buildings) at each location, and the associated structural replacement and

7.1. Input data definition

retrofitting cost per unit of each building typology.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.4">
<exposureModel gml:id="ep">
    <exposureList gml:id="Nepal_13_2" assetCategory="buildings"
                  stcoUnit="EUR" stcoType="per_asset" recoUnit="EUR"
                  recoType="per_asset">
        <gml:description>Buildings in Nepal</gml:description>
        <taxonomySource>Nepal taxonomy</taxonomySource>
        <assetDefinition gml:id="asset1">
            <site>
                <gml:Point srsName="epsg:4326">
                    <gml:pos>15.22 38.80</gml:pos>
                </gml:Point>
            </site>
            <taxonomy>URM</taxonomy>
            <number> 10 </number>
            <stco> 50.0 </stco>
            <reco> 5.0 </reco>
        </assetDefinition>
        ...
        <assetDefinition gml:id="asset999">
            <site>
                <gml:Point srsName="epsg:4326">
                    <gml:pos>15.23 38.95</gml:pos>
                </gml:Point>
            </site>
            <taxonomy>RC</taxonomy>
            <number> 5 </number>
            <stco> 140.0 </stco>
            <reco> 14.0 </reco>
        </assetDefinition>
    </exposureList>
</exposureModel>
</nrml>
```

In this example, the structural replacement and retrofitting value of each asset is not provided directly, as happened in the previous example. In order to carry out the risk calculations in which the structural and retrofitting value of each

7. Using the risk module

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 structural replacement and retrofitting cost per area.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrm1 xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.4">
<exposureModel gml:id="ep">
    <exposureList gml:id="Nepal_13_3" assetCategory="buildings"
                  stcoUnit="EUR/m2" stcoType="per_area" recoUnit="EUR/m2"
                  recoType="per_area" areaUnit="m2" areaType="aggregated">
        <gml:description>Buildings in Nepal</gml:description>
        <taxonomySource>Nepal taxonomy</taxonomySource>
        <assetDefinition gml:id="asset1">
            <site>
                <gml:Point srsName="epsg:4326">
                    <gml:pos>15.22 38.80</gml:pos>
                </gml:Point>
            </site>
            <taxonomy>URM</taxonomy>
            <area> 10000 </area>
            <stco> 400.0 </stco>
            <reco> 40.0 </reco>
        </assetDefinition>
        ...
        <assetDefinition gml:id="asset999">
            <site>
                <gml:Point srsName="epsg:4326">
                    <gml:pos>15.23 38.95</gml:pos>
                </gml:Point>
            </site>
            <taxonomy>RC</taxonomy>
            <area> 20000 </area>
            <stco> 500.0 </stco>
```

7.1. Input data definition

```
<reco> 50.0 </reco>
</assetDefinition>
</exposureList>
</exposureModel>
</nrml>
```

Once again, the OpenQuake engine needs to carry out some calculations in order to compute the economic value per asset. In this case, this value is computed by multiplying the aggregated built up area for each building typology by the associated replacement or retrofitting 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 structural replacement and retrofitting cost per area.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.4">
<exposureModel gml:id="ep">
    <exposureList gml:id="Nepal13_04" assetCategory="buildings"
                  stcoUnit="EUR/m2" stcoType="per_area" recoUnit="EUR/m2"
                  recoType="per_area" areaUnit="m2" areaType="per_asset">
        <gml:description>Buildings in Nepal</gml:description>
        <taxonomySource>Nepal taxonomy</taxonomySource>
        <assetDefinition gml:id="asset1">
            <site>
                <gml:Point srsName="epsg:4326">
                    <gml:pos>15.22 38.80</gml:pos>
                </gml:Point>
            </site>
            <taxonomy>URM</taxonomy>
            <number> 10 </number>
            <area> 100.0 </area>
            <stco> 400.0 </stco>
            <reco> 40.0 </reco>
        </assetDefinition>
    ...

```

7. Using the risk module

```
<assetDefinition gml:id="asset999">
  <site>
    <gml:Point srsName="epsg:4326">
      <gml:pos>15.23 38.95</gml:pos>
    </gml:Point>
  </site>
  <taxonomy>RC</taxonomy>
  <number> 5 </number>
  <area> 150.0 </area>
  <stco> 500.0 </stco>
  <reco> 50.0 </reco>
</assetDefinition>
</exposureList>
</exposureModel>
</nrml>
```

In this example, the OpenQuake engine will make use of all the parameters to estimate the economic and retrofitting value for each asset, by multiplying the number of buildings by its average built up area, and then by the respective replacement or retrofitting cost per unit of area.

Example 5

This example is comprised of an **exposure model** containing population distribution.

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
       xmlns="http://openquake.org/xmlns/nrml/0.4">
<exposureModel gml:id="ep">
  <exposureList gml:id="Nepal13_5" assetCategory="population">
    <gml:description>Population in Nepal</gml:description>
    <taxonomySource>Nepal taxonomy</taxonomySource>
    <assetDefinition gml:id="asset1">
      <site>
        <gml:Point srsName="epsg:4326">
          <gml:pos>15.22 38.80</gml:pos>
        </gml:Point>
      </site>
      <taxonomy>VF1</taxonomy>
```

7.1. Input data definition

```
<number> 200 </number>
</assetDefinition>
...
<assetDefinition gml:id="asset999">
    <site>
        <gml:Point srsName="epsg:4326">
            <gml:pos>15.23 38.95</gml:pos>
        </gml:Point>
    </site>
    <taxonomy>VF2</taxonomy>
    <number> 100 </number>
</assetDefinition>
</exposureList>
</exposureModel>
</nrml>
```

In this final example, the OpenQuake engine will require population distribution instead of structural replacement cost, hence, only the attribute `number` needs to be provided. Notice that in order for the engine to recognize that human losses are going to be estimated rather than economic losses, the attribute `assetCategory` needs to be set to `population`.

Scripts capable of converting information about the assets stored in Excel or ASCII files into NRML have been developed by the scientists working at **Global Earthquake Model (GEM)** and can be found at the GEM Science tools repository at GitHub (<http://github.com/GEMScienceTools>).

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 log-normal 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.

7. Using the risk module

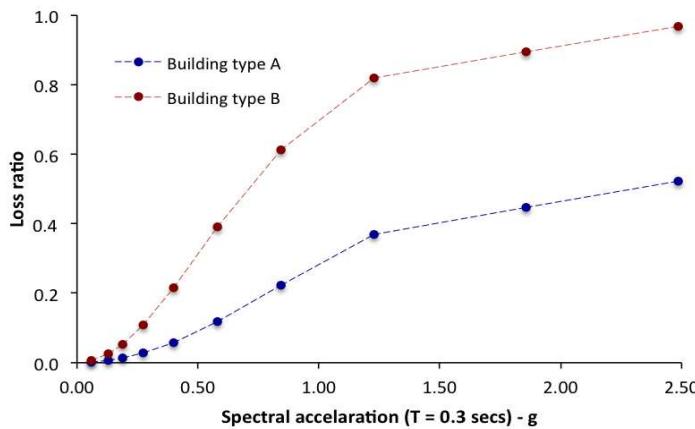


Figure 7.1. – Graphical representation of a vulnerability model.

Each component of the associated NRML file is presented herein:

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

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 assetCategory (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.:

7.1. Input data definition

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.

7. Using the risk module

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"?>
<nrm1 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>
```

7.1. Input data definition

```
</discreteVulnerabilitySet  
</vulnerabilityModel>  
</nrml>
```

Several methodologies to derive vulnerability functions are currently being evaluated by 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.

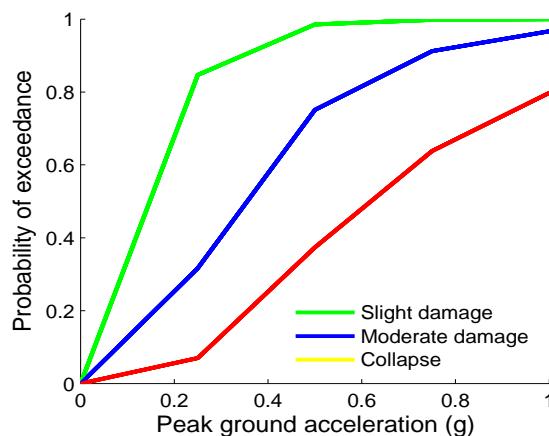


Figure 7.2. – Graphical representation of a discrete fragility model.

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:

7. Using the risk module

```
<?xml version="1.0" encoding="UTF-8"?>
<nrmr 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 damage
        moderate damage
        collapse
    </limitStates>
    ...

```

- **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 damage">
        <poes> 0.0 0.85 0.98 0.99 1.00 </poes>
    </ffd>
    <ffd ls="moderate damage">
        <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

7.1. Input data definition

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.

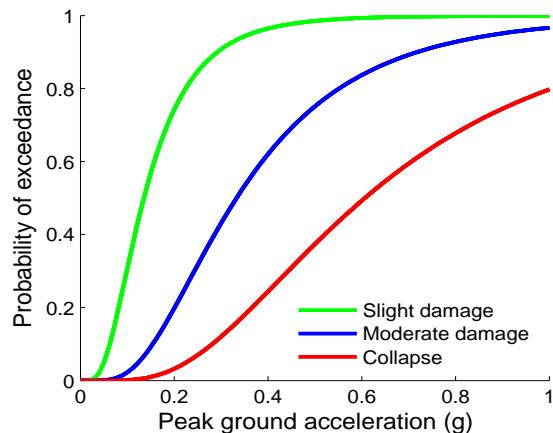


Figure 7.3. – Graphical representation of a continuous fragility model.

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:

7. Using the risk module

```
...
<ffs noDamageLimit= 0.05>
    <taxonomy RC </taxonomy>
    <IML IMT="PGA" minIML="0.0" maxIML="1.0" imlUnit="g" ></IML>
        <ffd ls="slight damage">
            <params mean="0.16" stddev="0.11" />
        </ffd>
        <ffd ls="moderate damage">
            <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

exposure_file = exposure_model.xml
region_constraint = 78.0 31.5,89.5 31.5,89.5 25.5,78 25.5
maximum_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;
- **maximum_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. Using the risk module

7.1.4.1. Scenario Risk Calculator

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

```
...  
vulnerability_file = vulnerability_model.xml  
asset_correlation = 0.7  
master_seed = 3
```

- `vulnerability_file`: this parameter is used to specify the path to the **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.

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. Input data definition

7.1.4.3. Probabilistic Event-based Risk Calculator

The parameter `calculation_mode` needs to be set to `event_based` 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. The remaining parameters are presented below.

```
...  
vulnerability_file = vulnerability_model.xml  
asset_correlation = 0.7  
master_seed = 3  
  
loss_curve_resolution = 15  
conditional_loss_poes = 0.01, 0.05, 0.1
```

- `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;
- `conditional_loss_poes`: this parameter is used to define the probabilities of exceedance at which loss maps are to be produced.

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`. 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 parameter is illustrated below.

```
...  
vulnerability_file = vulnerability_model.xml  
  
lrem_steps_per_interval = 2  
conditional_loss_poes = 0.01, 0.05, 0.1
```

7. Using the risk module

- `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. Retrofitting Benefit/Cost Ratio Calculator

As previously explained, this calculator uses loss exceedance curves which can be calculated using the Classical PSHA-based Risk or the Probabilistic Event-based Risk calculators. Therefore, depending on which calculator a user chooses to employ, the configuration file will be different. If the Classical PSHA-based Risk calculator is employed, then the `calculation_mode` should be set to `classical_bcr` and the calculator-specific part of the configuration file should be defined as presented below.

```
...
vulnerability_file = vulnerability_model.xml
vulnerability_retrofitted_file = vulnerability_model_retrof.xml

lrem_steps_per_interval = 2

interest_rate= 0.005
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.

Alternatively, if a user decides to employ the Probabilistic Event-based Risk calculator for the calculation of the loss curves, then the `calculation_mode` should

7.1. Input data definition

be set to event_based_bcr and the remaining portion of the configuration file should be defined as follows.

```
...
vulnerability_file = vulnerability_model.xml
vulnerability_retrofitted_file = vulnerability_model_retrof.xml
asset_correlation = 0.7
master_seed = 3

loss_curve_resolution = 20

interest_rate= 0.005
asset_life_expectancy = 50
```


CHAPTER 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. Then, the following command can be used:

```
user@ubuntu:~$ openquake --loadgmf <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
```

8. Risk calculations and results

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. 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:~$ openquake --run-risk job.ini --hazard-output-id  
<hazard_output_id>
```

or simply:

```
user@ubuntu:~$ openquake --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:~$ openquake --run-risk job.ini --hazard-calculation-id  
<hazard_calculation_id>
```

or simply:

8.1. Running OpenQuake engine for risk calculations

```
user@ubuntu:~$ openquake --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:~$ openquake --list-risk-calculations
```

or simply:

```
user@ubuntu:~$ openquake --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:~$ openquake --list-risk-outputs <risk_calculation_id>
```

or simply:

```
user@ubuntu:~$ openquake --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 Curve set for hazard 32
30	loss_map	Loss Map Set with poe 0.1 for hazard 32

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

8. Risk calculations and results

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

or simply:

```
user@ubuntu:~$ openquake --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.

8.2. Description of the outputs

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

8. Risk calculations and results

```
<?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"?>
<nrmr 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"/>
    </asset>
  </DDNode>
</nrmr>
```

8. Risk calculations and results

```
<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"/>
```

8.2. Description of the outputs

```
<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 bellow.

```
<?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>
```

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>
```

8. Risk calculations and results

```
<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.

8.2. Description of the outputs

```
<?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 or Probabilistic Event-based Risk calculators. For this reason, the parameters sourceModelTreePath and gsimTreePath are also included in this NRML schema, so the whole calculation process can be tracked back. The results for each asset are being stored as depicted below.

8. Risk calculations and results

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

- **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;

8.2. Description of the outputs

- 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 (**aal0rig**) and the average annual loss for the retrofitted (**aalRetr**) configuration of the **assets** are provided.

CHAPTER 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

9. Demonstrative examples

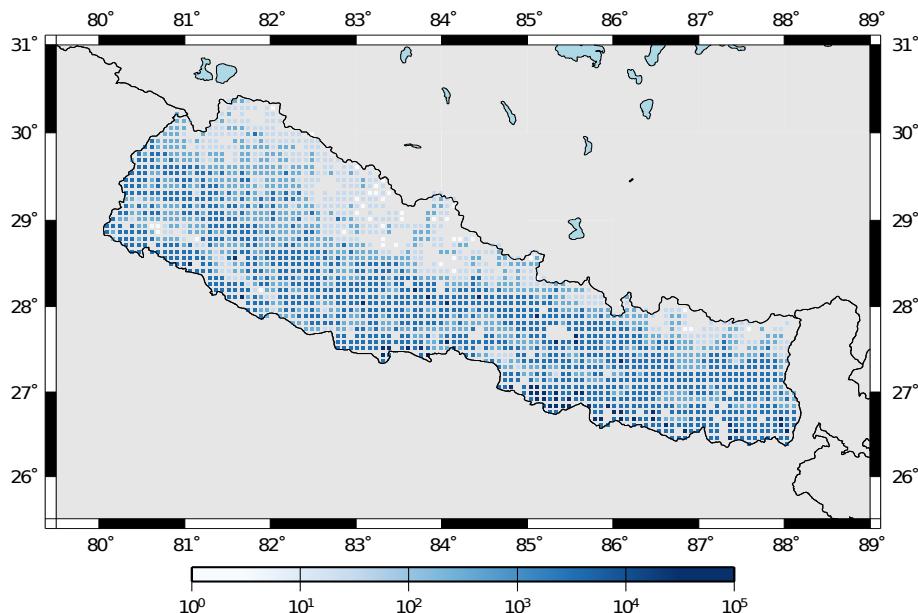


Figure 9.1. – Distribution of number of buildings in Nepal.

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 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:~$ openquake --rh job_gmfq.ini
```

9.2. Scenario Damage demo

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:~$ openquake --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  
22 | loss_map | Loss Map
```

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:~$ openquake --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. Demonstrative examples

9.3. Classical PSHA-based Risk demo

The seismic source model developed within the Global Seismic Hazard Assessment Program (GSHAP) was used with the ? 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:~$ openquake --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:~$ openquake --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 Curve set for hazard 27  
29 | loss_map | Loss Map Set with poe 0.1 for hazard 27  
30 | loss_map | Loss Map Set with poe 0.01 for hazard 27
```

9.4. 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:

9.5. Retrofitting Benefit/cost ratio demo

```
user@ubuntu:~$ openquake --rh job_hazard.ini
```

and the following results are expected:

```
Calculation 15 results:  
id | output_type | name  
31 | gmf | gmf-rlz-72  
32 | ses | ses-coll-rlz-72
```

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:~$ openquake --rr job_risk.ini --hazard-output-id 31
```

leading to the following outputs:

```
Calculation 16 results:  
id | output_type | name  
33 | loss_curve | Loss Curve set for hazard 31  
34 | loss_map | Loss Map Set with poe 0.1 for hazard 31  
35 | loss_map | Loss Map Set with poe 0.01 for hazard 31  
36 | agg_loss_curve | Aggregated loss curve for hazard 31
```

9.5. 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:~$ openquake --rr job_bcr.ini --hazard-output-id 27
```

which should produce the following output:

9. Demonstrative examples

```
Calculation 17 results:
```

id	output_type	name
37	bcr_distribution	BCR Distribution for hazard 27

PART IV

Appendices

APPENDIX A

Supported ground motion prediction equations

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 project¹. The model is applicable to magnitudes 5-8.5, distances 0-200 km, and spectral periods 0-10 sec (?).
- 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 and 7.6 and distances up to 100 km (?).
- AkkarCagnan2010
A ground motion prediction equation for shallow earthquakes in active tectonic regions developed using data from the Turkish strong-motion

¹<http://peer.berkeley.edu/ngawest>

A. Supported ground motion prediction equations

database. Equations are valid for a distance range of 0–200 km and are derived for moment magnitudes between 5 and 7.6 (?).

- 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 5-8, distances < 200 km, and spectral periods 0-10 (?).
- 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 (?).
- 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 4-8.5 (if strike-slip), 4-8 (if normal or reverse) and distances 0-200 km.
- FaccioliEtAl2010
Based on the same functional form of ? but using closest distance to the rupture instead of hypocentral distance (?)
- SadighEtAl1997
A ground motion prediction based primarily on strong motion data from California and applicable for magnitude in range 4-8 and distances < 100 km (?).
- ZhaoEtAl2006Asc
A ground motion prediction equation for active shallow crust events developed using mostly Japanese strong ground motion recordings (?).

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 (?).

² <http://peer.berkeley.edu/ngawest>

A.3. Ground motion prediction equations for stable continental regions

- LinLee2008SInter, LinLee2008SSlab
Ground motion prediction equations for subduction interface and in-slab events created using strong motion data included in the the Taiwanese database (?).
- 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 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}). (?)
- ZhaoEtAl2006SInter, ZhaoEtAl2006SSlab
Ground motion prediction equations for subduction interface and in-slab developed using mostly japanese strong ground motion recordings (?).

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 (?).
- Campbell2003
Ground motion prediction equation calibrated for Eastern North America applicable for events with magnitude greater than 5 and distances < 70 km (?).
- 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 5-8 and distances in 1-500 km (?).

APPENDIX B

Supported magnitude-scaling relationships

B.1. Magnitude-scaling relationships for shallow earthquakes in active tectonic regions

We provide below a list of the magnitude-area scaling relationships implemented in the [oq-hazardlib](#).

- ? - One of the most well known magnitude scaling relationship based on a global database of historical earthquake ruptures. The implemented relationship is the one linking magnitude to rupture area.

