Testing procedures adopted in the development of the hazard component of the OpenQuake-engine

Contents

1. Introduction

The current document describes the testing procedures adopted in the development of the hazard component of the OpenQuake-engine (OQ-engine), the open source hazard and risk software developed by the Global Earthquake Model initiative.

Nowadays seismic hazard analysis serves different needs coming from a variety of users and applications. These may encompass engineering design, assessment of earthquake risk to portfolios of assets within the insurance and reinsurance sectors, engineering seismological research, and effective mitigation via public policy in the form of urban zoning and building design code formulation.

Decisions based on seismic hazard results may have impacts on population, properties and capitals, possibly with important repercussions on our day-to-day life. For these reasons, it is recommendable that the generation of hazard models and their calculation is based on well-recognized, state-of-the-art and tested techniques, requirements that must be reconciled with the need to regularly incorporate recent advances given the progress carried out within the scientific community. The features described below contribute to fulfill these requirements:

- Software should have a modular and flexible structure capable to incorporate new features and - as a consequence - offer to the users the most recent and advanced techniques. In very general terms, modularity is the level to which a component of a system can be moved, replaced or reused. In software design, modularity means the separation of the software into smaller independent components that can be implemented, maintained and tested easily and efficiently.
- Software should have and extensive test coverage which captures possible errors and avoids regressions (i.e. unexpected behaviors introduced by new features). Software testing (Myers et al., 2012) is an important, complex and vast discipline which helps developing methods and processes aimed at certifying the extent to which a computer code behaves according to the original design and user specifications.

The OQ-engine includes different levels of modularity. The first is the one separating the engine itself into a number of libraries (see Figure 1.1) each one containing well identified knowledge, objects and methods (e.g. the OQ-hazardlib includes objects and methods needed to compute probabilistic seismic hazard). The second one pertains to the data model adopted in the development of each library as a result of the abstraction process. According to Berkes (2012) scientific software must be:
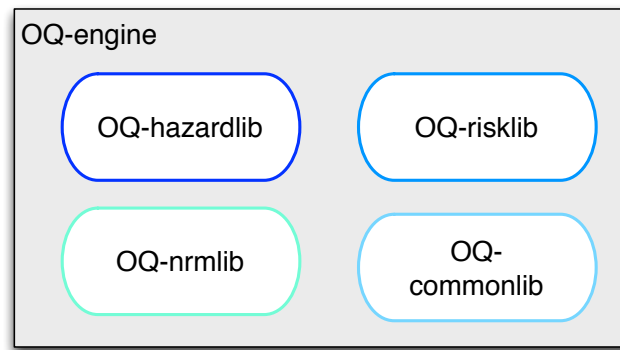
Figure 1.1 – *A schematic describing the main components of the OpenQuake-engine software.*

- Error proof
- Flexible and able to accommodate different methods
- Reproducible and re-usable.

1.1 Testing and Quality Assurance

Despite the distinction between software testing (in some cases also called Quality Control) and Software Quality Assurance (SQA) is vague and partly open to personal judgment, it's clear that SQA is a more comprehensive and overarching process than software testing. SQA aims at the definition of the best processes that should be used to provide guarantees that user expectations will be met. Software testing focuses instead on detecting software faults by inspecting and testing the product at different stages of development.

1.1.1 Testing

Software testing can be implemented at different stages of the development process, with varying strategies to approach the problem. The OQ-engine and the associated libraries are developed following an agile paradigm. This development strategy is organized in a way that the creation of the real code is completed in parallel and fully integrated with the software testing process.

The software engineering community provides a wide range of testing levels and typologies. In the current document we consider just a portion of them with the specific intent to illustrate the standards used in the development of the OQ-engine and particularly of its hazard component.

1.1.2 Quality Assurance

From the IEEE "Standard for Software Quality Assurance Processes": *Software quality assurance is a set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of suitable quality for their intended purposes. A key attribute of SQA is the objectivity of the SQA function with respect to the project. The SQA function may also be organizationally independent of the project; that is, free from technical, managerial, and financial pressures from the project.* In this document we are not covering topics related to SQA since this would go beyond its scopes.

1.2 Document structure

The document is organized into four main chapters and two appendixes.

In the current chapter we provide a very brief and general introduction to software testing with a focus on the testing of scientific software.

In the second chapter we describe the module, or unit, testing, the acceptance tests adopted in the development of the OQ-engine and we discuss some examples.

In the third and fourth chapters we illustrate tests comparing the results computed with the OQ-engine against the ones computed using different probabilistic seismic hazard analysis software.

Appendix A provides details on the PEER tests implemented in the OpenQuake hazard library (OQ-hazardlib).

2. Unit testing

This chapter provides an introduction to the module (unit) testing procedures (Myers et al., 2012) and describes estensively example of tests implemented in the OQ-engine.

2.1 Unit-Testing: An Overview

At the first level of the code quality assurance process is the practice of "unit-testing". This process is a central tenet of test-driven software development and is widely established as a means of "best-practice". Before looking closely at the OpenQuake-engine approach to unit-testing it is important to establish what are the precise objective of the unit-testing process and the benefits (and limitations) that it brings.

Correctness of Implementation

This objective is obviously the primary goal of unit-testing, to ensure that each function of the code is operating in the manner expected by the developer. "Correctness", in this case, requires that the function produces both the correct output, but also if there are cases in which function may fail then the means of failure should be predictable. The following is a relatively simple example of how a unit-test relates to a function:

Consider a simple function to multiply two numbers and take the logarithm of the result. A relevant analogy may be that of a magnitude scaling relation calculation, in which both a rupture length and rupture width are required, and the logarithm of the area may be needed by the function itself. In this circumstance a negative value in either of the two inputs would result in a calculation error. This could be coded in the following manner.

```python
def get_log_area(length, width):
    if (length < 0) or (width < 0):
        raise valueError("Both inputs must be positive")
    else:
        return log10(length * width)
```

From the description above it is evident that the user requirements inform the manner in which the function should behave (i.e. negative values cannot be tolerated). To ensure that the function is operating correctly, we wish to write a set of tests that will confirm the behaviour is correct:

1. 1. If both a and b are equal to 10.0, then the function should return 2.0
2. If $a = -1$ and $b = 10$ the function should raise an error reporting the stated message "Both inputs must be positive".
3. If $a = 10$ and $b = -1$ the function should raise an error reporting the stated message "Both inputs must be positive".
4. If $a = -1$ and $b = -1$ the function should raise an error reporting the stated message "Both inputs must be positive".

A unit-test for this function is an additional function that will check that both cases are satisfied, and will report an error if not.

A comprehensive unit-test suite for a software may fulfil two objectives: **line coverage** and **parameter coverage**. The former should ensure that, in as far as possible, every line (or statement) in the code is executed as some point in the testing process. The latter should ensure that the behaviour of the function is predictable when supplied with "unusual" parameters. In the above example, both objectives are satisfied by the tests. The first test will result in a positive valued "area", thus executing the second branch of the logical path, the second test will result in a negative area and will execute the first logical branch. Therefore all lines of the code are covered and the line coverage is complete. We also see that in this simple example there are four possible cases: i) a is positive and b is positive, ii) a is positive and b is negative, ii) a is negative and b is positive, and iv) both a and b are negative. Only the first case is valid, therefore the first test ensures that they provide the correct answer (usually verified by independent means), whilst the remaining tests should ensure that the function raises the correct error. Thus the full parameter space of the input is ensured.

The above case is, of course, trivial; however, as shall be seen in due course, this same process can be applied in more complex contexts. Furthermore, the same unit-testing approach can be applied not only to individual components within the PSHA calculation, but also to full calculations, essentially verifying that the hazard curve produced by the full PSHA calculator is in agreement with that produced independently (sometimes by hand).

Identifies Problems Prior to Software Release

This advantage is largely self-explanatory, but for many software projects this can reduce the possibility of requiring *a posteriori* fixes to the code (patches). By compiling a comprehensive suite of unit-tests, and following a software development and release process that should automatically run the tests at the point of packaging, this should ensure that new features added to the software cannot inadvertently break other components.

Facilities Improvements in Performance

In the creation of software intended to perform demanding scientific calculations, like those commonly associated with PSHA, the issue of computational performance and efficiency is a major one. There is a continuing need to improve the speed and reduce the work required to undertake the PSHA calculation. To implement improvements it is necessary to ensure that optimisations do not modify the outputs of the calculation, only the speed at which they are performed. The unit-testing is absolutely fundamental to this process as optimisation cannot be undertaken readily without a means to ensure the calculation outputs have not changed. This point was a critical motivation behind the transition from the OpenSHA basis of the OpenQuake hazard calculation engine prior to version 1.0, to the current OpenQuake hazard library.

2.2   Continuous Integration

OpenQuake is developed and packaged within a "continuous integration" system (https://ci.openquake.org/), which used the open-source software "Jenkins" (http://jenkins-ci.org/). Continuous integration is

used in large software projects to run a full test suite of the complete software, either at fixed time intervals or, as in the current case, when any new code is committed to the repository. The continuous integration system will do the following:

1. Run the full set of unit-tests for all code in all of the linked repositories. This will include the main (or "master") branch of the software repository, i.e. the one that will be used for packaging of the software, as well as some development branches.

2. Run a test of the software installation. This test will install the software on a dedicated platform and check that the installation of the software is successful. This test also ensures that if changes occur in the dependency packages, and these changes affect or compromise the installation and operation of the software, these problems are recognised immediately.

3. The software will also run standard Python tests for quality of code, compilation of documentation etc.

4. Several long-running tests may also be run. These implement larger scale seismic hazard and risk calculations designed to test the overall performance of the engine.

If at any point the tests should fail, the OpenQuake development team will be notified automatically. This should ensure that software that is failing any of the tests will remain on the main branch of the repository for the minimum amount of time possible. Furthermore, if the continuous integration tests fail, the new code will not be integrated into the nightly package of the software.

2.3 The OpenQuake Hazard Library: Unit Tests

The unit-test suite for the OpenQuake hazard library consists of three types of tests: i) simple tests for individual functions to verify the correctness of implementation ("component testing"), ii) simple tests of the full calculators for PSHA ("method testing"), and iii) "acceptance" tests, which provide a basic quality assurance check for the each of the three main calculators.

2.3.1 Component Testing

The unit-testing at the component level breaks the functions into simple calculations whose results can be verified by hand. These tests, similar in nature to that illustrated previously, provide the majority of the line and parameter coverage needed to ensure a robust code. To illustrate the comprehensive nature of the coverage we consider the example of the functions to undertake calculations of geodetic distance between two points, which can be found here: https://github.com/gem/oq-hazardlib/blob/master/openquake/hazardlib/geo/geodetic.py. Whilst not necessarily a complex function in itself, the distance between two points on the Earth's surface is a critical component of the software that is frequently called at several points of the PSHA process. Therefore, it is critical that the function operates correctly and its behaviour under extreme cases is understood. Thus, this relatively simple function is verified in the following cases:

- `test_LAX_to_JFK` Checks that a correct geodetic distance is calculated for two known locations on the Earth. This value is verified against an implementation of the algorithm provided by an online geodetic calculation tool.
- `test_on_equator` Checks that the correct distance is provided for two points located on the equator.
- `test_along_meridian` Checks that the correct distance is returned for two points located along the same meridian.
- `test_one_point_on_pole` Verifies the distance calculations for two points assuming one point is located at the geographic pole.

- `test_small_distance` Verifies that two points separated by a distance within the floating point error are considered to be separated by zero km
- `test_opposite_points` Verifies the correct distance between two points in different longitudinal hemispheres (i.e. checks that the distance crosses the international dateline correctly).
- `test_array` Verifies the correct distances between two set of points
- `test_one_to_many` Verifies the correct distances between one point and a set of points.

The test suite for this one function is illustrative of several key components of the unit-testing. First is the use of an independent tool to provide the expected values of the calculation under simple conditions. Second is the use of "extreme cases" such as polar locations, or across the International Dateline. These ensure that the function can be global in application.

The nature of the interdependencies between the functions also means that one a functions own unit-test is verified, the function can then form the basis for testing other conditions. So for example, the geodetic distance tools also contain a method to calculate the minimum distance between a collection of points and a single point. Rather than requiring new expected distances for the different conditions, the geodetic distance function can then be used to construct tests for functions that utilise it. This makes the testing process more efficient, and reduced the need to write large numbers of tests in order to ensure correct behaviour of the function.

### 2.3.2 Ground Motion Prediction Equation (GMPE) Testing

The implementation process for ground motion prediction equations requires careful consideration, as it is in this area that new features may be expected to be added regularly, and where contributions from third parties are more likely to be incorporated into the software. Furthermore, in most cases the expected values of the functions may only be obtained from independent implementations of the GMPE. These expected values take the form of test tables, which are simple comma-separated value (csv) files that provide the expected values and standard deviations of the ground motion prediction equation for an exhaustive combination of parameters for the predictor variables. These should be sufficient to ensure that every part of the GMPE is covered within the test. An example test table is shown in Figure 2.1.

rup_mag	rup_rake	dist_rjb	site_vs30	result_type	damping	pga	pgv	0.0100	0.0200	0.0300
5	-90	0	180	MEAN	5	1.49601781e-01	8.37557932e+00	1.52156355e-01	1.58795879e-01	1.70210885e-01
5	-90	1	180	MEAN	5	1.47903599e-01	8.28996294e+00	1.50423093e-01	1.56955838e-01	1.68179118e-01
5	-90	2	180	MEAN	5	1.43085213e-01	8.04655171e+00	1.45505601e-01	1.51737565e-01	1.62421471e-01
5	-90	5	180	MEAN	5	1.17737222e-01	6.75328291e+00	1.19647558e-01	1.24355023e-01	1.32322974e-01
5	-90	10	180	MEAN	5	7.61992828e-02	4.57644574e+00	7.73244998e-02	7.97912614e-02	8.38439990e-02
5	-90	20	180	MEAN	5	3.59883616e-02	2.35829212e+00	3.64414906e-02	3.71850105e-02	3.83466727e-02
5	-90	50	180	MEAN	5	9.65199384e-03	7.53451965e-01	9.74580840e-03	9.78355926e-03	9.84122886e-03
5	-90	100	180	MEAN	5	3.17550009e-03	2.90883054e-01	3.20031015e-03	3.17448272e-03	3.14065152e-03
5	-90	200	180	MEAN	5	1.00938000e-03	1.09500168e-01	1.01546574e-03	9.95551038e-04	9.69732082e-04
5	-90	0	300	MEAN	5	1.67252769e-01	7.24113461e+00	1.70032072e-01	1.78131433e-01	1.92998896e-01
5	-90	1	300	MEAN	5	1.65118988e-01	7.16015408e+00	1.67857380e-01	1.75822115e-01	1.90442661e-01
5	-90	2	300	MEAN	5	1.59075525e-01	6.93017897e+00	1.61698530e-01	1.69284590e-01	1.83211051e-01
5	-90	5	300	MEAN	5	1.27599511e-01	5.71599408e+00	1.29633909e-01	1.35315177e-01	1.45760346e-01
5	-90	10	300	MEAN	5	7.77695947e-02	3.71798421e+00	7.89225604e-02	8.18670506e-02	8.73396462e-02
5	-90	20	300	MEAN	5	3.34057502e-02	1.79579640e+00	3.38462325e-02	3.47685057e-02	3.65703890e-02
5	-90	50	300	MEAN	5	8.14246877e-03	5.37497854e-01	8.23083166e-03	8.33039755e-03	8.58648935e-03
5	-90	100	300	MEAN	5	2.60178559e-03	2.03416151e-01	2.62548172e-03	2.62680638e-03	2.66678039e-03
5	-90	200	300	MEAN	5	8.18506981e-04	7.60354513e-02	8.24547330e-04	8.15494704e-04	8.15529413e-04

Figure 2.1 – *Example GMPE test table used by OpenQuake*

To ensure the most objective testing strategy, we aim for the test tables to match the GMPE

creator's own implementation of the GMPE, in as far as possible. Therefore we prefer to solicit input from the authors of the GMPE. This will often take one of two forms. We ask that the authors can provide test tables, in a convenient format, or that they provide their own software implementation of the GMPE, from which we will then generate the test tables. Input from the GMPE authors is highly desirable within this process as it can help resolve issues that are perhaps ambiguous within the original publications of the GMPE and it can identify errors and bugs in the author's own implementation. The full workflow for GMPE implementation in OpenQuake is shown in Figure 2.2.
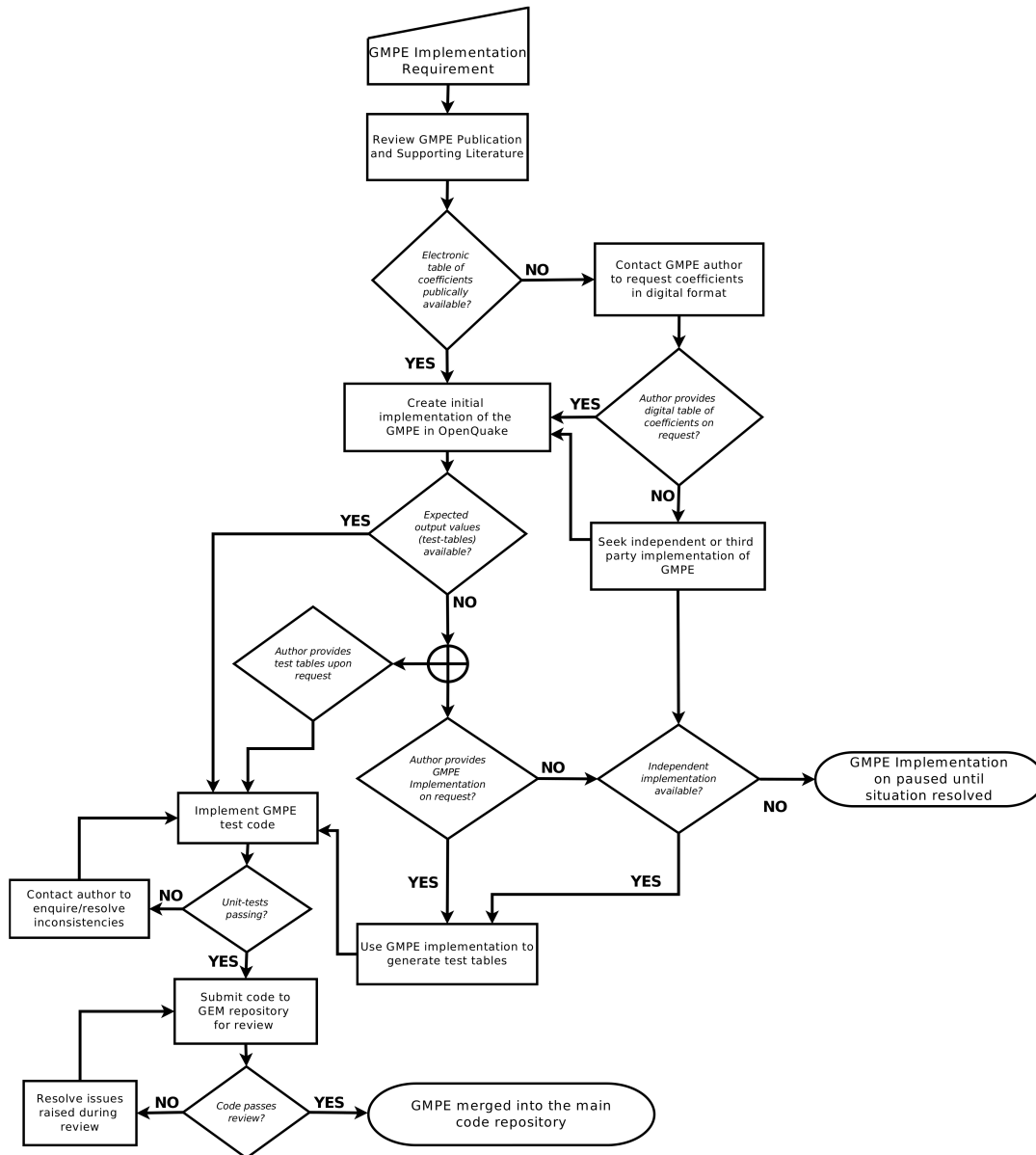
Figure 2.2 – *OpenQuake GMPE Implementation Process*

The GMPE unit-tests themselves are designed to be simple for the user to create once the test tables are provided. Ideally the expected values should match the implementation values to within the test precision (typically permitting a difference of 10^{-7}). In some cases, however, it may not be possible to match the desired level of precision and therefore the tests permit

the maximum discrepancy level (as a percentage) to be specified. Discrepancies may arise due to rounding of the coefficients within published tables, but ideally the tolerable discrepancy between an expected and predicted value should be not more than one tenth of one percent.

As is shown from Figure 2.2, once the GMPE test tables are created, the GMPE implementation should then be checked against the unit-tests. If discrepancies cannot obviously be resolved the author may then be contacted for clarification. Once the unit-tests pass the code is then submitted for review by (typically) one or more of the software development team and one or more of the scientific team. This may help identify issues such as inefficiencies or unclear code. Once the submission is accepted by both the scientific and IT reviewer the code is merged into the main repository. This will then trigger a full test from the continuous integration system described previously.

2.4 Acceptance Tests

In addition to the individual unit-tests on functions, the OpenQuake hazard library contains three code "acceptance" tests. These are tests that are designed to exercise the full workflow of the classical, event-based and disaggregation calculators. Further comprehensive tests of all of the main OpenQuake calculators are also found in the test suite of the main OpenQuake-engine, and these shall be elaborated upon in section ??.

2.4.1 Classical PSHA Acceptance Tests

Some of the tests cases taken from the PEER tests suite (Thomas et al., 2010), are used as the basis for the unit tests of the classical PSHA calculation engine. A description of the tests implemented is provided in Appendix A.

2.4.2 Event-Based PSHA Acceptance Tests

The event-based acceptance tests are designed to verify that for a sufficiently long stochastically generated catalogue originating from an area source, 10^6 years in this case, the normalised rate of events in each magnitude frequency bin is approximately equal to that of the expected magnitude frequency distribution. This is expanded to check out source to site distance filtering by considering a second source beyond the expected soure-to-site distance filter range.

2.4.3 Deaggregation Acceptance Tests

The disaggregation calculator is tested in two separate places. A first unit-test evaluates the disaggregation of hazard for a simple case in which the probabilities in the disaggregations bins have been calculated, by hand, for a simple rupture model. This unit-test will fulfil the requirements of line and parameter coverage, including edge cases such as if the rupture crosses the international dateline, or if no ruptures contribute to the hazard at the site. A second unit-test using a more realistic source and GMPE combination are then implemented. In this case it is not possible to calculate the probabilities by hand, but instead the OpenQuake results are used as the expected values. This test is circular in nature, and is intended simply as a means to ensure that changes to the code do not alter the results of the disaggregation calculator.

2.5 Summary

In this section we have outlined both the process and the key benefits of developing comprehensive unit-tests for OpenQuake, as well as outlining the operation of the continuous integration system, which should ensure that code with the potential to break the tests cannot be packaged and released. The unit-tests themselves have not been discussed in detail as nearly one thousand

tests are executed during the unit-test process. However, to view the comprehensive set of tests reader is encouraged to refer to the full test-suite, which is open and available on the OpenQuake code repository (https://github.com/gem/oq-hazardlib/tree/master/openquake/hazardlib/tests). Furthermore, we have also discussed how OpenQuake development tries to facilitate correct implementation of features such as ground motion prediction equations. For relatively simple conditions, a selection of PEER tests (Thomas et al., 2010) are built into the testing process, making OpenQuake unique amongst other hazard software in integrating the verification into the development process. The following chapters will expand in greater detail upon the additional hazard curve benchmark tests, which both follow and expand upon the PEER testing process.

3. Other PSHA codes: simple cases

This chapter presents a number of benchmarks used to compare the OpenQuake-engine hazard results against solutions provided by alternative software. Classical PSHA results (i.e. hazard curves and map values) generated by the OpenQuake-engine are compared against reference solutions computed using the the suite of seismic hazard analysis computer programs developed by the United States Gelogical Survey - National Seismic Hazard Program (USGS-NSHMP).

3.1   Comparison against the USGS-NSHMP computer programs

The USGS-NSHMP develped a number of Fortran-based computer programs used to implement the U.S. national seismic hazard model (Petersen et al., 2008). Source codes are available on the internet (http://earthquake.usgs.gov/hazards/products/conterminous/2008/software/) together with input files containing the seismic hazard model definition. Input files are available for the three main source typologies defined in the U.S. national seismic hazard model: distributed seismicity, crustal faults and subdution faults. The USGS-NSHMP seismic hazard software offers therefore an excellent opportunity to test the OpenQuake-engine algorithms for the different seismic source typologies.

3.1.1   Benchmarks for distributed seismicity

As a benchmark for the modeling of distributed seismicity we selected a gridded seismicity model developed for the State of California (Figure 3.1). The model defines spatially variable occurrence rates over a regular grid (0.1 by 0.1 degrees) obtained using a smoothed seismicity approach (Frankel, 1995). Occurrence rates are defined from minimum magnitude equal to 5. Maximum magnitude is equal to 7 in areas away from faults. Close to faults, the maximum magnitude is assumed equal to the minimum between 7 and the fault's maximum magnitude. Occurrence rates follow a double truncated Gutenberg-Richter magnitude frequency distribution with $b_{GR} = 0.8$. We compare hazard curves on a set of 21 locations, equally spaced, defining a profile connecting San Francisco (-122.42W, 37.78N) to Los Angeles (-118.25E, 34.05N). We computed hazard curves (probabilities of exceedance in 50 years) for peak ground acceleration using the Boore and Atkinson, 2008 GMPE. This GMPE requires as distance metrics the Joyner-Boore distance (R_{JB}). We first computed hazard curves under the assumption of *point ruptures*. That is ruptures are treated as having no spatial extension. Under this assumption R_{JB}
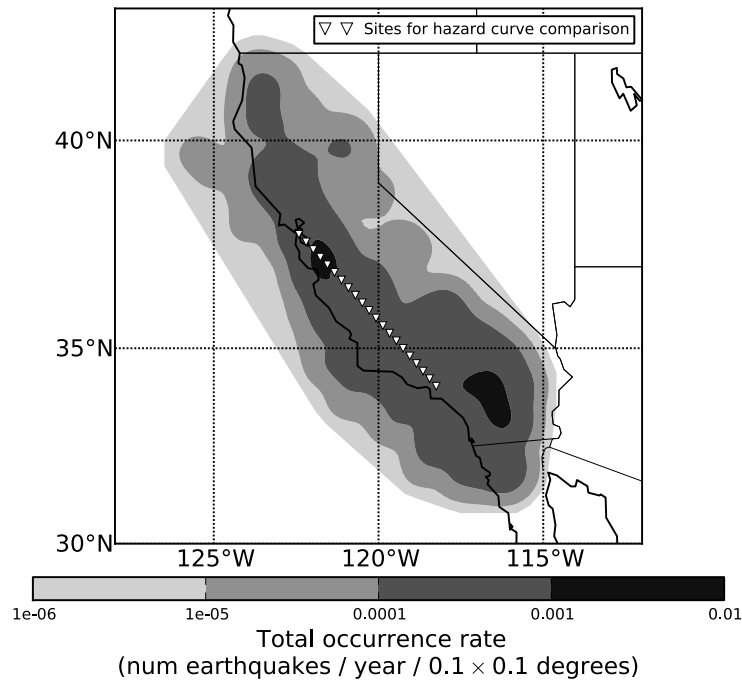
Figure 3.1 – *Gridded seismicity model used as benchmark for testing distributed seismicity in the OQ-engine. Hazard curves are compared for a set of 21 sites defining a profile connecting San Francisco to Los Angeles.*

converges to epicentral distance. We computed hazard curves using the OpenQuake-engine and the USGS-NSHMP code for gridded seismicity (*hazgridXnga3.f*). Results of the comparison (as PGA values for different return periods along the profile) are given in Figure 3.2. Comparison of hazard curves at a single site is also given in Figure 3.3. For most of the locations, results are in agreement. For few sites, larger discrepancies are observed that increase with increasing return periods.

A second test is perfomed but considering extended ruptures. For simplicity, ruptures are assumed vertical and aligned along a single strike direction ($0°$). All ruptures are assumed to be strike-slip. In the OQ-engine calculation, the rupture extension is modeled by considering the *magnitude-area* scaling relationship of Wells and Coppersmith, 1994 ($Area = 10^{-3.42+0.90*M}$) and assuming an aspect ratio equal to 1. In the *hazagridXnga3.f* rupture extension is modeled based on the *magnitude-length* scaling relationship of Wells and Coppersmith, 1994 ($Length = 10^{-3.22+0.69*M}$). Results of the comparison are given in Figures 3.4 and 3.5. Discrepancies are much larger than under the *point ruptures* approximations, at all return periods. This can be motivated by the fact that the two software adopt different modeling strategies. The OQ-engine uses a magnitude-area scaling relationship which implies that rupture lenght may be increased if, for given area and rupture aspect ratio, the resulting rupture width is larger than the seismogenic layer thickness. On the contrary, in the USGS-NSHMP approach, rupture extension is constrained by a magnitude-lenght scaling relationship. For the same rupture magnitude, the OQ-engine may predict shorter distances and thus provide higher hazard values.

3.1.2 Benchmarks for crustal faults

As benchmarks for crustal faults, we considered two fault models for the State of California. One based on type-A faults (Petersen et al., 2008), which defines only characteristic faul sources
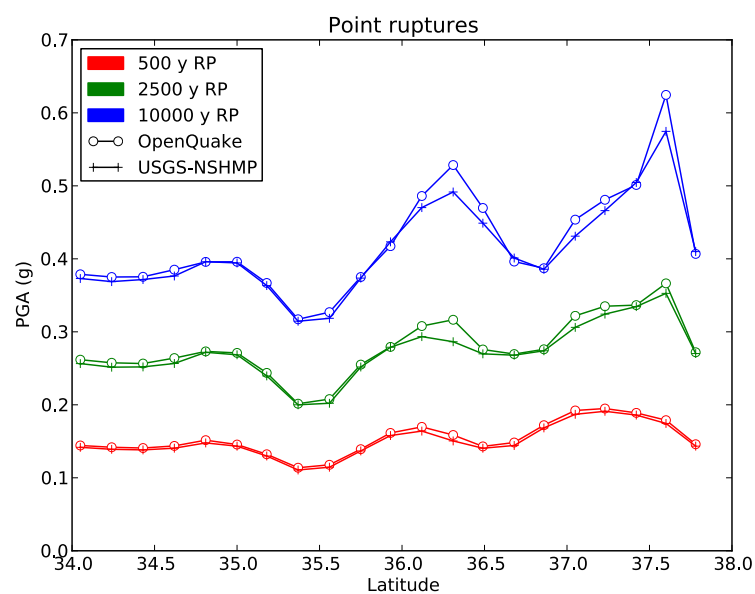
Figure 3.2 – *Comparison of hazard map values for different return periods (RP) along the profile assuming point ruptures*
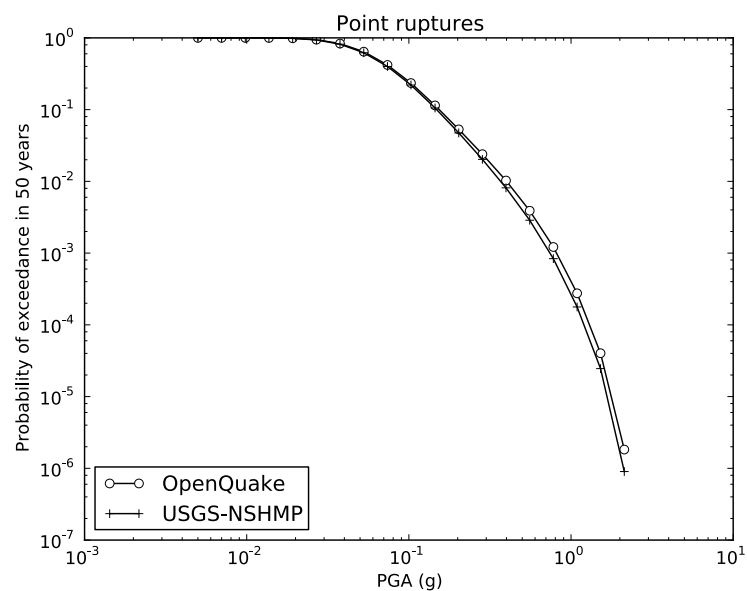
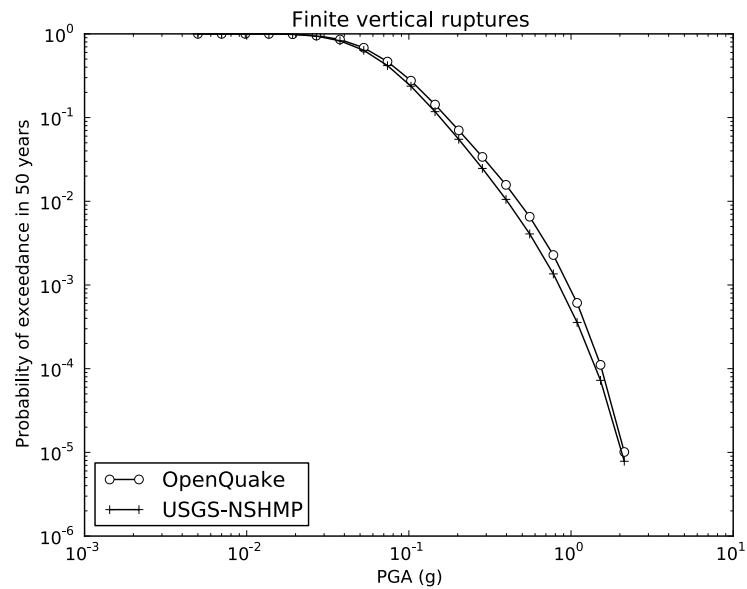**Figure 3.3** – *Hazard curve comparison for site with coordinates -120.7E, 36.31N assuming point ruptures*
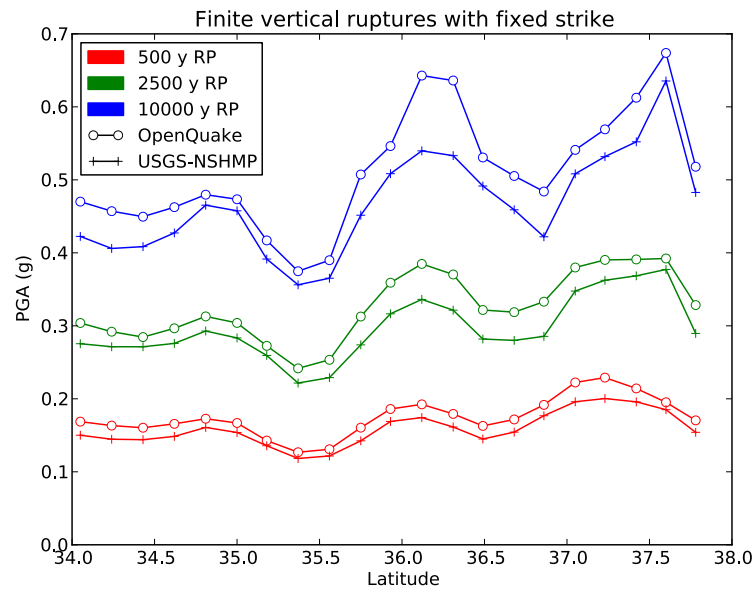
Figure 3.4 – *Comparison of hazard map values for different return periods (RP) along the profile assuming finite vertical ruptures with fixed strike.*

Figure 3.5 – *Hazard curve comparison for site with coordinates -120.7E 36.31N assuming finite vertical ruptures with fixed strike.*

(Figure 3.6) and one based on type-B faults (Figure 3.7), which contain mostly Gutenberg-Richter fault sources.
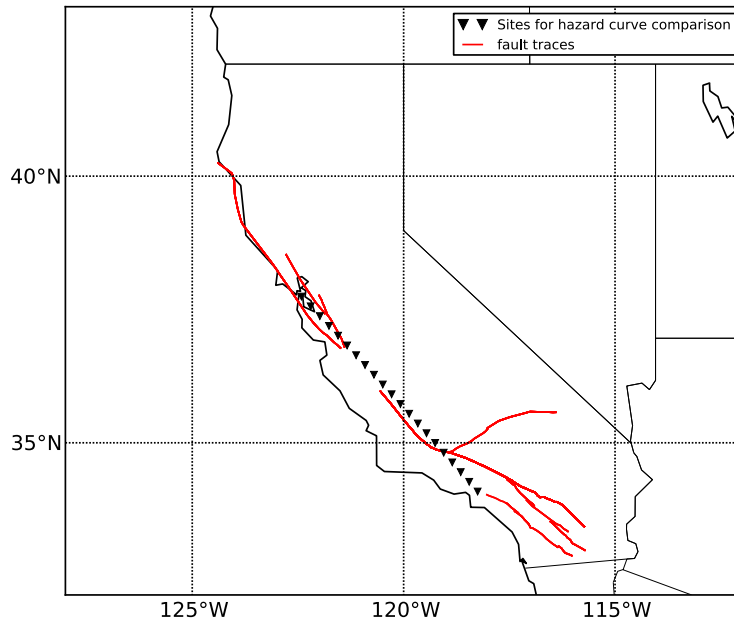
Figure 3.6 – *Type-A fault model*

Characteristic faults are defined through a Normal (i.e. Gaussian) magnitude-frequency distribution. For each magnitude-bin, the entire fault surface is assumed to rupture. In a characteristic model no scaling relationship is required, and hence no floating ruptures are modeled. On the contrary, in a Gutenberg-Richter fault, ruptures are floated along the fault surface (both along strike and dip) and rupture extension is modeled in terms of a magnitude-length scaling relationship. We compare hazard curves for the same set of locations as for the distributed seismicity model. Results of the comparison are given in Figures 3.8 and 3.9 for the type-A fault model, and 3.10, and 3.11 for the type-B fault model. For the type-A fault model, the agreement between the USGS-NSHMP and OpenQuake-engine solutions is very good. When considering instead the results from the type-B fault model some discrepancies are visibile, especially for return periods longer than 500 years, altough the level of agreement is still acceptable. The lack of a complete agreement in the type-B fault model solutions may indicate the effect of differences in the rupture floating algorithm between the two software. Indeed, the very good agreement in the case of the type-A fault model (which does not consider any rupture floating) confirms that the modeling of rupture extension and location on a fault surface plays an important role in the hazard calculation procedure.

3.1.3 Benchmarks for subduction faults

The 2008 U.S. national seismic hazard model (**pertersen2008**) defines fault sources with complex geometries to model large subdaction interface faults in the Cascadia region. We thus selected a fault model for the Cascadia region as a benchmark for the modeling of complex fault sources. Similarly to what done for crustal sources, we considered a characteristic model (where earthquakes always rupture the entire fault surface) and an unsegmented model where ruptures (whose extension is constrained by a scaling relationship) are allowed to move along the entire fault surface. The characteristic model defines a single event of magnitude ($M_w = 9.2$) which breaks a complex fault surface depicted in Figure 3.12. The unsegmented model defines the
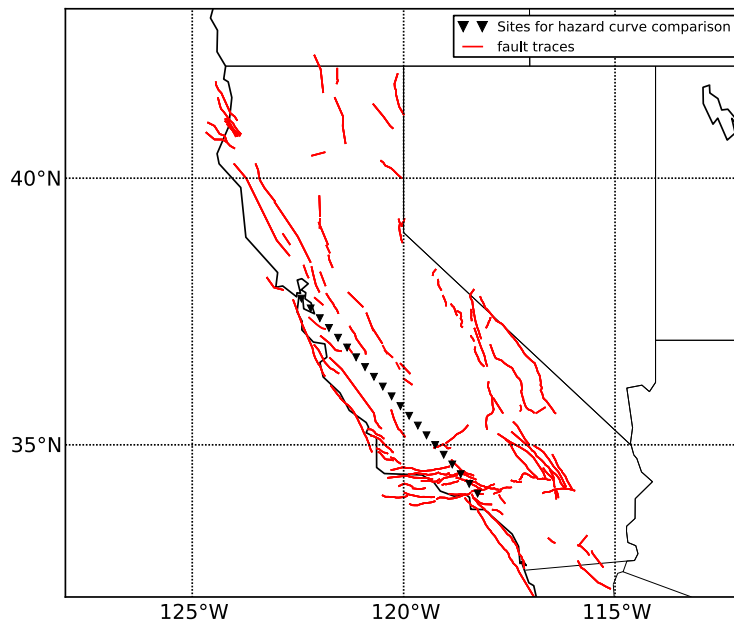
Figure 3.7 – *Type-B fault model*

same fault geometry, however earthquake ruptures are generated from minimum magnitude 8.3 to maximum magnitude 8.7. Hazard curves are compared along two profiles, aligned along the E-W and N-S directions. When considering the characteristic model, the agreement between the solutions provided by the two software is very good (Figures 3.13, 3.14). On the contrary, when considering the unsegmented model (Figures 3.15 and 3.16) large discrepancies are evident especially along the N-S profile. The large discrepancies can be understood again in terms of the differences between the rupture floating algorithm in the two software. In particular, the USGS-NSHMP code define rupture extension along length only (using a magnitude-lenght scaling relationship), and assume each rupture to extend along dip until the fault bottom edge. That is rupture floating occurs only along the strike direction and not along dip. In the OQ-engine instead, rupture extension and shape is determined by a magnitude-area scaling relationship and rupture aspect-ratio, and each rupture can be moved both along strike and along dip. The bulge in the hazard map profiles as visibile in Figure 3.16 can be explained in terms of a large number of ruptures model by the OQ-engine in the inner part of the fault surface. In the USGS-NSHMP approach, the number of ruptures along the fault surface is constant, as also demonstrated by the constant hazard values along the N-S profile.
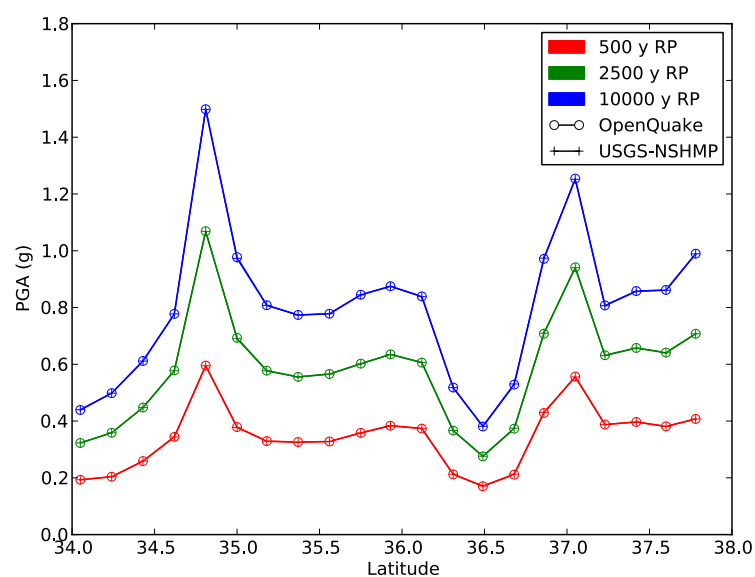
Figure 3.8 – *Hazard map comparison for Type-A fault model.*
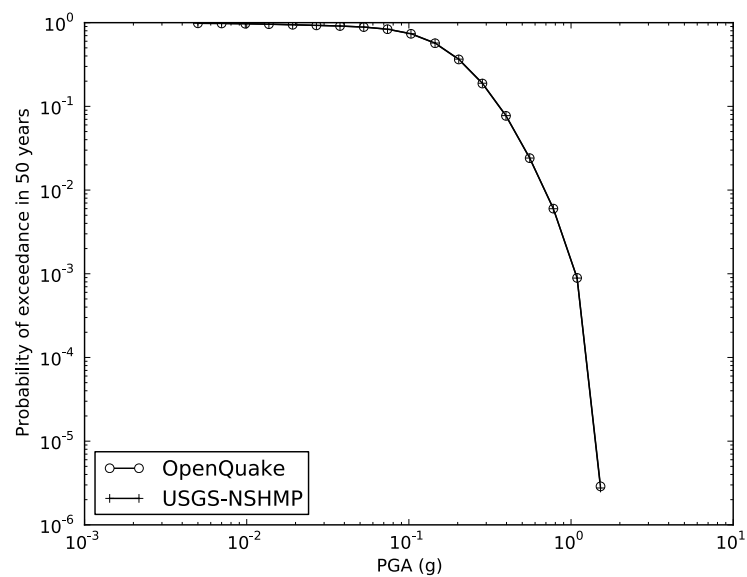
Figure 3.9 – *Hazard curve comparison for site with coordinates -120.49E 36.12N using Type-A fault model*
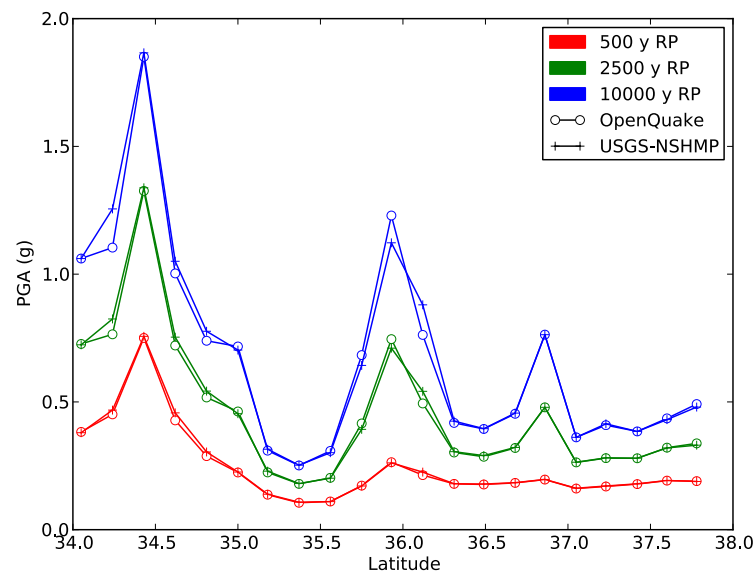
Figure 3.10 – *Hazard map comparison for Type-B fault model*
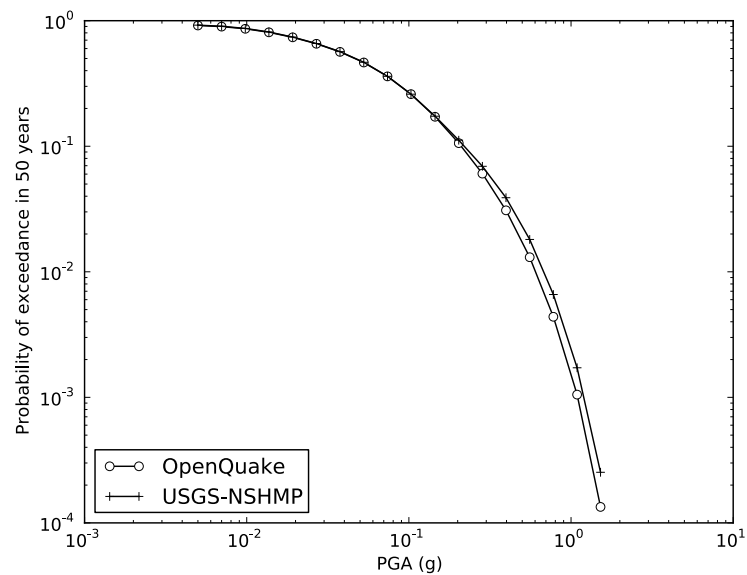
Figure 3.11 – *Hazard curve comparison for site with coordinates -120.49E 36.12N using Type-B fault model*
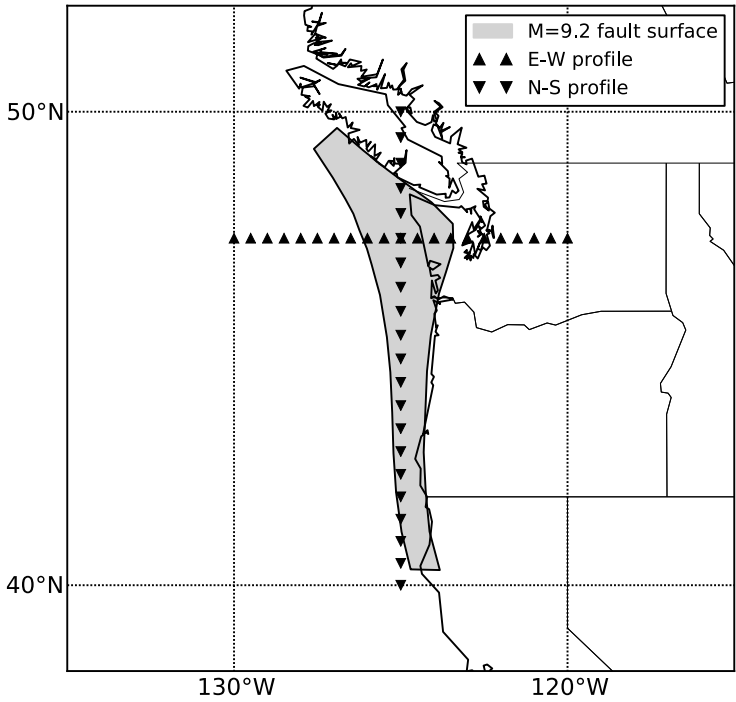
Figure 3.12 – *Cascadia characteristic fault model (M$_w$=9.2)*
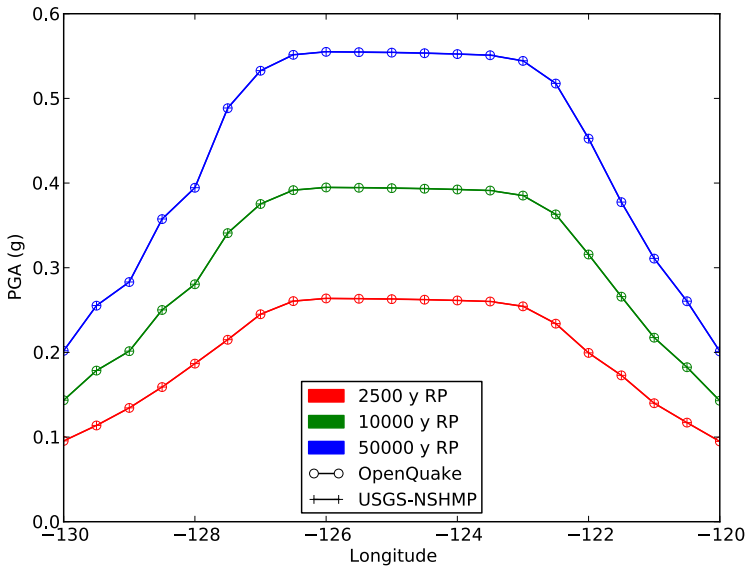
Figure 3.13 – *Hazard map comparison along EW profile*
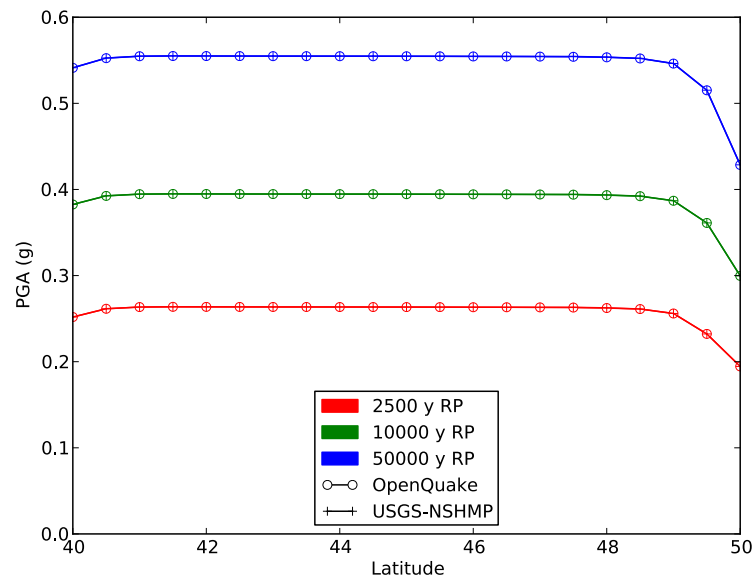
Figure 3.14 – *Hazard map comparison along NS profile*
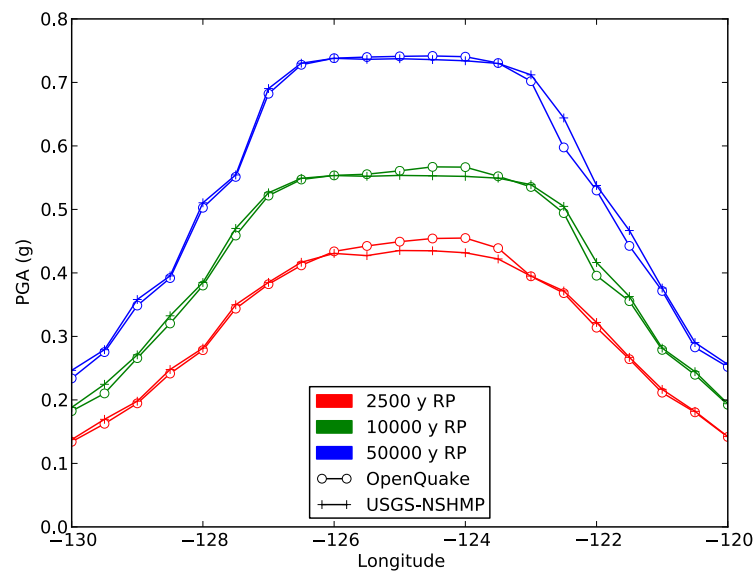
Figure 3.15 – *Hazard map comparison along EW profile using Cascadia unsegmented model*
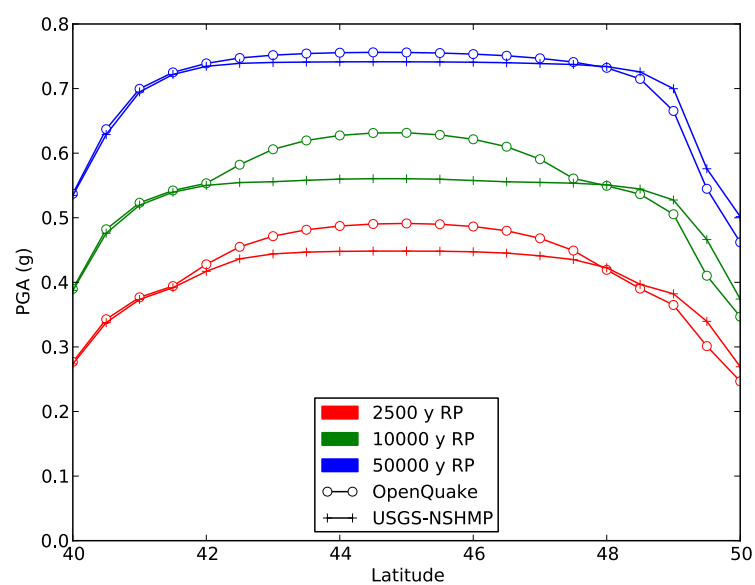
Figure 3.16 – *Hazard map comparison along NS profile using Cascadia unsegmented model*

4. Other PSHA codes: real cases

4.1 The PEER project

AA

4.2 test

BB

A. The OQ-engine PEER acceptance tests

In 2001 the Pacific Earthquake Engineering Center (PEER) started a project aimed at validating and comparing results provided by a number of public and private Probabilistic Seismic Hazard Analysis (PSHA) codes (for a comprehensive list see Thomas et al., 2010). For this purpose two set cases were created to check the way codes were implementing fundamental steps in the PSHA calculation mechanics. The major differences observed were appointed to differences in the numerical procedures adopted in the different software. The test acceptance level adopted corresponds to 10 percent in probability for a chosen value of ground motion intensity.

A.1 Test case set 1

The test case set 1 was designed to test basic elements of the software such as (Thomas et al., 2010):

- modeling of ruptures on fault planes
- implementation of Magnitude-Frequency Distributions (MFDs)
- modeling of area sources
- modeling of ground motion variability

Figure A.1 shows the geometry of the sources adopted in the first test case set.

A.1.1 Description of test cases

The OQ-engine contains some of the test cases included in the PEER set 1. Below we provide a brief description for each of them.

AULT 1

FAULT 2

Cross-sectional view of Fault 2

25 km

25 km

Site 1

1 km

11 km

Fault 2

e: Strike Slip
0 degrees
epths: 0 - 12 km

Fault Type: Reverse
Dip: 60 degrees west
Fault Plane Depths: 1 - 12 km

TES FOR FAULTS 1 & 2

AREA 1 WITH SITES

50 km

10 km   10 km

2   1   7

25 km

12.5 km

4

10 km

5

r =100 km

1

50 km

2

3

25 km

4

fault, at midpoint along strike
km west of fault, at midpoint along strike
km west of fault, at midpoint along strike
fault, at southern end
km south of fault along strike
fault, northern end
km east of fault, at midpoint along strike

Site 1: At center of area
Site 2: 50 km from center (radially)
Site 3: On area boundary
Site 4: 25 km from boundary

N

Figure A.1 – *Geometry of the sources adopted in the PEER test case set 1 and corresponding position of the sites.*

Test case 2 This test considers uniform slip and considers a single rupture with an extension
smaller than the whole surface of the hosting fault. The goal is to test results also
considering possible edge effects. The fault is a vertical strike-slip fault with a Gutenberg-
Richter b-value equal to 0.9 and a slip rate of 2 mm/yr giving an annual recurrance of
$0.0160425 \; yr^{-1}$. The MFD is a degenerate distribution localized at magnitude 6 while the
Ground Motion Prediction Equation (GMPE) is the Sadigh et al. (1997) one with $\sigma^B = 0$.
The following scaling relations are used, which in combination give an expected aspect
ratio equal to 2.0.

$$\log_{10} A \quad = \quad M_W - 4.0 \tag{A.1}$$

$$\log_{10} W \quad = \quad 0.5 M_W - 2.15 \tag{A.2}$$

$$\log_{10} L \quad = \quad 0.5 M_W - 1.85 \tag{A.3}$$

where A, W and L are the rupture area, width and length respectively.

Figure A.2 – *Comparison between the results provided by Thomas et al. (2010) for the test case 2
set 1 (grey dashed lines) and the OQ-hazardlib (green dots).*

Test case 5 This case considers the same fault plane as in **Set 1 Case 2** albeit with an exponential magnitude frequency distribution with M_{MIN} and M_{MAX} of 5.0 and 6.5 respectively and a b-value of 0.9. The same slip rate is assumed, which translates into an a-value of 3.1292. All other inputs are the same.
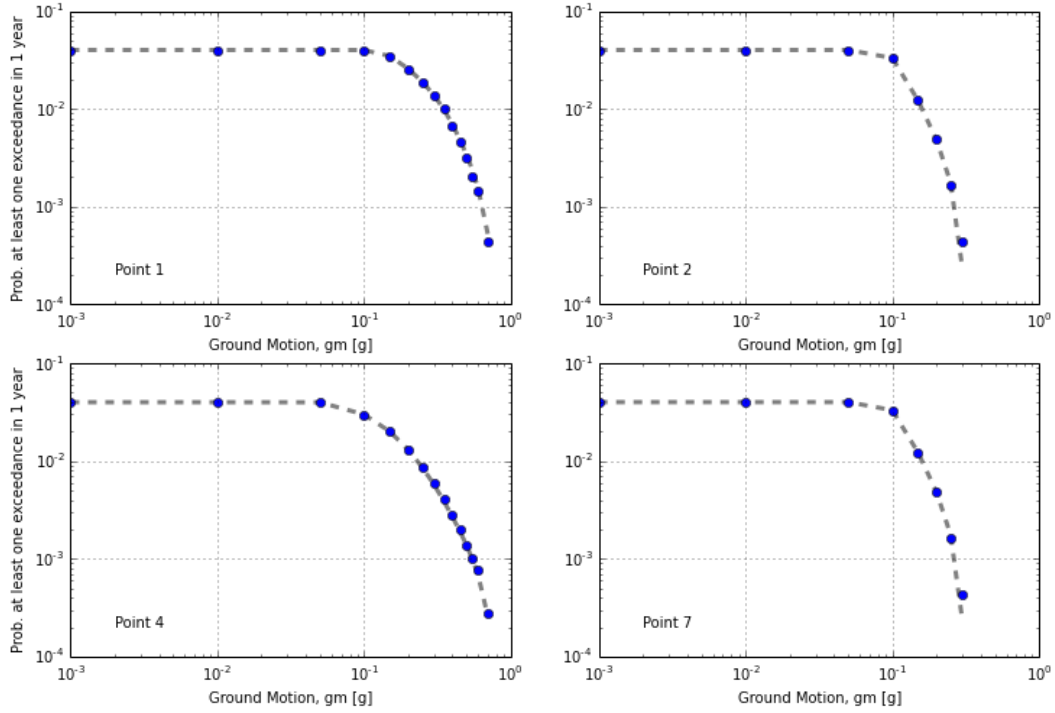
Figure A.3 – *Comparison between the results provided by Thomas et al. (2010) for the test case 5 set 1 (grey dashed lines) and the OQ-hazardlib (green dots). The sites considered have the following indexes: 1, 2, 3, 4 (see Figure A.1)*

Test case 10 This test considers one uniform area source with a truncated exponential model with M_{min} of 5.0, M_{max} 6.5, b-value of 0.9 and an annual rate $M_W \geq M_{min}$ of 0.0395. As OpenQuake defines finite rupture planes for each of the points considered in the area source, the scaling relation was fixed such that the area of the finite rupture was equal to 1.0 km. Hypocentral depth is fixed at 5 km. The preferred GMPE is Sadigh et al., 1997 for rock, with sigma set to 0.0. The expected values for the unit tests are those provided in the appendix of Thomas et al., 2010 (page A - 15), which represent the mean values of the distribution of estimates from the software considered. As these are not solved by hand, the test are considered to pass when the following condition is satisfied for all values:

$$|calculated - expected| \leq (atol + rtol * |expected|) \tag{A.4}$$

where *atol* and *rtol* are the absolute and relative difference between two terms, set to 10^{-4} and 10^{-1} respectively.

Figure A.4 – *Comparison between the results provided by Thomas et al. (2010) for the test case 10 set 1 (grey dashed lines) and the OQ-hazardlib (green dots). The sites considered have the following indexes: 1, 2, 3, 4 (see Figure A.1).*

Test case 11 The same area source and GMPE are considered as for **Set 1 Case 10**; however, the hypocentral depth is distributed uniformly between 5 km and 10 km. All other conditions are the same.
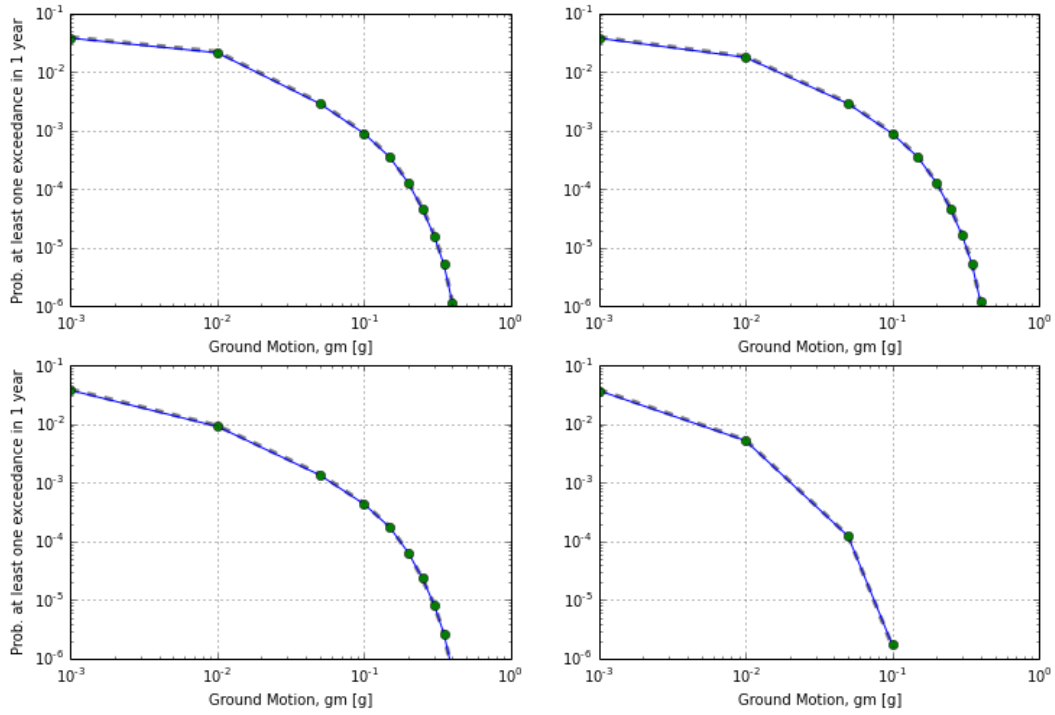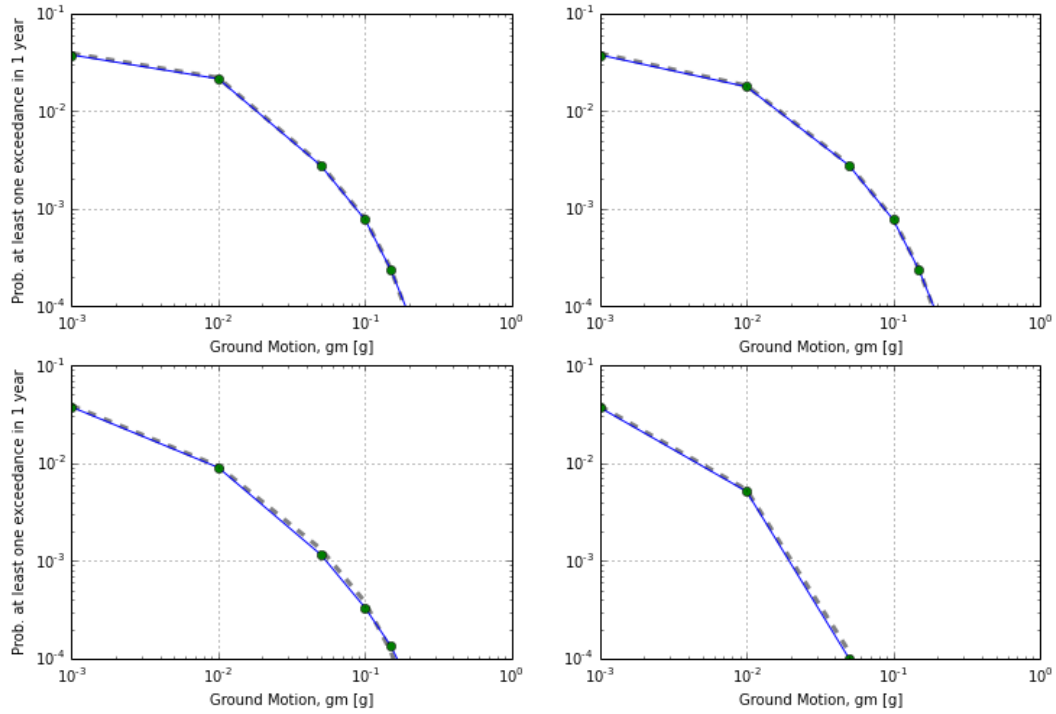

Figure A.5 – *Comparison between the results provided by Thomas et al. (2010) for the test case 11 set 1 (grey dashed lines) and the OQ-hazardlib (green dots). The sites considered have the following indexes: 1, 2, 3, 4 (see Figure A.1).*

A. The OQ-engine benchmark tests

In this section we illustrate the benchmark tests implemented in the OQ-engine. These tests are organised in two main sections. In the first one we consider simple testing cases based on a single seismic source, in the second set we will describe tests designed to check the OQ-engine calculations in case of hazard models based on a logic-tree.

All the tests described in this appendix are accessible at this url `https://github.com/gem/oq-engine/tree/master/qa_tests/hazard/`.

A.1 Theoretical background

Assuming seismicity following a *Poissonian* temporal occurrence model, the calculation of the *probability of (at least one) exceedance* of a ground motion value x in a time span T, for an intensity measure IM, can be computed by first calculating the annual rate of exceedance:

$$\lambda(IM \geq x) = \sum_{i=1}^{N_{sources}} \lambda_i(m \geq M_{min}) \int_{M_{min}}^{M_{max}^i} \int_{r=0}^{\infty} P(IM \geq x|m,r) f_i(m) f_i(r) dr dm \tag{A.1}$$

where:
- $N_{sources}$: total number of sources in the source model
- $\lambda_i(m \geq M_{min})$: annual rate of ruptures in the i-th source with magnitude greater than or equal to M_{min}
- M_{max}^i: maximum magnitude in the i-th source
- $P(IM \geq x|m,r)$: conditional probability for an intensity measure IM to exceede an intensity measure level (x), given magnitude value (m) and distance (r)
- $f_i(m)$: probability density function for magnitude, for the i-th source
- $f_r(r)$: probability density function for distance, for the i-th source

and then computing the Poissonian probability of at least one exceedance using the equation:

$$P(IM \geq x) = 1 - e^{-\lambda(IM \geq x)T} \tag{A.2}$$

Assuming ground motion distribution to follow (in log scale) a truncated normal distribution, we can compute $P(IM \geq x|m,r)$ as:

$$P(IM \geq x|m,r) = 1 - \frac{\Phi(\frac{ln(x) - \overline{ln(IM;m,r)}}{\sigma}) - \Phi(-n_{trunc})}{\Phi(n_{trunc}) - \Phi(-n_{trunc})} \tag{A.3}$$

where Φ is the normal cumulative distribution function. In the following tests, a truncation level of 2 (that is $n_{trunc} = 2$) will be used. In this case, rounding to 4 digits, we obtain $\Phi(2) = 0.9772$, $\Phi(-2) = 0.0228$, and $\Phi(2) - \Phi(-2) = 0.9545$. So we can rewrite equation A.3 as:

$$P(IM \geq x|m,r) = 1 - \frac{\Phi(\frac{ln(x) - \overline{ln(IM;m,r)}}{\sigma}) - 0.0228}{0.9545} \tag{A.4}$$

In case a truncation level of 0 is considered, we can write $P(IM \geq x|m,r)$ as:

$$P(IM \geq x|m,r) = H(\overline{ln(IM;m,r)} - ln(x)) \tag{A.5}$$

where H is the Heaviside (or step) function. That is $P(IM \geq x|m,r) = 1$ if $\overline{ln(IM;m,r)} \geq ln(x)$ and 0 otherwise.

A.1.1 GMPE

The GMPE utilized for the calculation of the tests is Sadigh et al. (1997), for PGA on rock sites, for strike slip events (rake = 0) with magnitude less or equal to 6.5. The mean of the logarithm of PGA is predicted by the following equation:

$$\overline{ln(IM;m,r)} = -0.624 + m - 2.1ln(r + exp(1.29649 + 0.25m)) \tag{A.6}$$

where r is the closest distance to the rupture. The standard deviation is given by:

$$\sigma = 1.39 - 0.14m \tag{A.7}$$

A.1.2 Magnitude Scaling Relationship

The magnitude scaling relationship utilized in the QA tests for calculating median rupture area from magnitude value is the one defined in the PEER tests, that is:

$$A = 10^{m-4} \tag{A.8}$$

where A is the rupture area (in squared km).

A.2 Tests

A.3 Hazard curve calculation with different single source types

Hazard curve calculation with single point source, single magnitude MFD, and with ground motion truncation level = 2.0

NOTE: This test is meant to exercise the hazard curve calculator (both classical and event-based) with a point source

Let's consider a source model consisting of a single point source, generating a single rupture of magnitude 4, with aspect ratio (length/width) equal to 1, vertical (dip=90.0), and with an annual occurrence rate ($\lambda(m = 4)$) equal to 1. This means that the probability density function for magnitude is a dirac delta function centered at m = 4, that is:

$$f(m) = \delta(m - 4.0) \tag{A.9}$$

Let's further assume that the rupture hypocenter (located at the centroid of the rupture surface) is located at 4 km depth, and let's assume upper seismogenic depth at 3.5 km, and lower seismogenic depth at 4.5 km.

Let's assume the site of interest to be right on top of the point source (that is same latitude and longitude).

The rupture area (from eq. A.8) is 1 squared km, and given that the aspect ratio is 1, the rupture length and width are equal to 1 km. Given that the rupture hypocenter is assumed to be in the centre of the rupture surface, the top of rupture depth is 3.5 km. Given that the site is sitting just right on top of the point source, the closest distance from site to rupture is 3.5 km. Given that there is a single rupture scenario, and this is the only possible distance value, the probability density function for the closest distance to rupture is a dirac delta function centered at r = 3.5, that is:

$$f(r) = \delta(r - 3.5) \tag{A.10}$$

Substituting equations A.3 and A.10 in equation A.1, and considering that there is only one source in the source model, and, assuming $M_{min} = 4$, $\lambda_i(m \geq M_{min}) = \lambda(m = 4) = 1$, equation A.1 simplifies to:

$$\lambda(IM \geq x) = P(IM \geq x | 4.0, 3.5) \tag{A.11}$$

The mean of the logarithm of PGA, as predicted by equation A.6 is (rounded to 4 digits):

$$\overline{ln(IM;m,r)} = -0.624 + 4.0 - 2.1ln(3.5 + exp(1.29649 + 0.25 * 4.0)) = -2.0802 \tag{A.12}$$

and the corresponding standard deviation (as from eq. A.7) is:

$$\sigma = 1.39 - 0.14 * 4.0 = 0.83 \tag{A.13}$$

The rates of exceedance for PGA levels equal to 0.1, 0.4, 0.6 are:

$$\lambda(PGA \geq 0.1) = 1 - \frac{\Phi(\frac{ln(0.1) + 2.0802}{0.83}) - 0.0228}{0.9545} = 0.6107$$

$$\lambda(PGA \geq 0.4) = 1 - \frac{\Phi(\frac{ln(0.4) + 2.0802}{0.83}) - 0.0228}{0.9545} = 0.0605$$

$$\lambda(PGA \geq 0.6) = 1 - \frac{\Phi(\frac{ln(0.6) + 2.0802}{0.83}) - 0.0228}{0.9545} = 0.0069$$

$$\tag{A.14}$$

The corresponding probabilities of exceedance in a time period of 1 year ($T = 1.0$) are:

$$P(PGA \geq 0.1) = 1 - exp(-0.6107 * 1.0) = 0.4570$$

$$P(PGA \geq 0.4) = 1 - exp(-0.0605 * 1.0) = 0.0587$$

$$P(PGA \geq 0.6) = 1 - exp(-0.0069 * 1.0) = 0.0069$$

$$\tag{A.15}$$

Hazard curve calculation with single point source, truncated GR MFD, and ground motion truncation level = 0

NOTE: This test is meant to exercise the hazard curve calculator (both classical and event-based) with a point source having a Gutenberg Richter magnitude frequency distribution

This test assumes a source model consisting of a single point source, with a magnitude frequency distribution (MFD) defined as a truncated Gutenbergh-Richter with the following parameters: $a = 2.0$, $b = 1.0$, $M_{min} = 4.0$, $M_{max} = 7.0$. The point source is assumed to generate ruptures at a single hypocentral depth of 0.5 km, with an aspect ratio of 1. The upper seismogenic depth is set to 0.0 (the lower seismogenic depth can be set to an arbitrary value, clearly larger than the hypocentral depth).

Starting from the minimum magnitude all ruptures generated by the source will reach the earth surface. Indeed, for magnitude $M = 4$, rupture area is 1 squared km (from equation A.8) and assuming aspect ratio of 1, rupture width is 1, and having set rupture hypocenter to 0.5 km, the rupture will reach the surface. Assuming site of interest to be right on top the source location, for all ruptures the closest distance to the site of interest is 0.0 km. This means that the probability density function for the closest distance to rupture is a dirac delta function centered at r = 0.0, that is:

$$f(r) = \delta(r) \tag{A.16}$$

The probability density function for a truncated GR can be written as:

$$f(m) = \frac{b\,ln(10)\,10^{-b(m-M_{min})}}{1 - 10^{-b(M_{max}-M_{min})}} \tag{A.17}$$

Assuming a truncation level equal to 0, the probability of the logarithm of PGA exceeding a level x can be computed using equation A.5.

Substituting equations A.16, A.17, and A.5 in equation A.1, and considering that the source model consists of only one source, we can write:

$$\lambda(PGA \geq x) = \lambda(m \geq M_{min}) \int_{M_{min}}^{M_{max}} H(-0.624 + m - 2.1(1.29649 + 0.25m) - ln(x))$$

$$\frac{b\,ln(10)\,10^{-b(m-M_{min})}}{1 - 10^{-b(M_{max}-M_{min})}} dm =$$

$$\lambda(m \geq M_{min}) \frac{b\,ln(10)}{1 - 10^{-b(M_{max}-M_{min})}} \int_{M_{min}}^{M_{max}} H(0.475m - 3.346629 - ln(x)) 10^{-b(m-M_{min})} dm =$$

$$\lambda(m \geq M_{min}) \frac{b\,ln(10)}{1 - 10^{-b(M_{max}-M_{min})}} \left[-\frac{10^{-b(m-M_{min})}}{b\,ln(10)} \right]_{max\{M_{min}, \frac{3.346629+ln(x)}{0.475}\}}^{M_{max}} \tag{A.18}$$

if $M_{max} \leq \frac{3.346629+ln(x)}{0.475}$, otherwise the result of the equation is 0.

if $M_{min} \geq \frac{3.346629+ln(x)}{0.475}$, we can simplify equation A.18 as:

$$\lambda(PGA \geq x) = \frac{\lambda(m \geq M_{min})}{1 - 10^{-b(M_{max}-M_{min})}} (1 - 10^{-b(M_{max}-M_{min})}) \tag{A.19}$$

if $M_{min} \leq \frac{3.346629+ln(x)}{0.475}$, we can write equation A.18 as:

$$\lambda(PGA \geq x) = \frac{\lambda(m \geq M_{min})}{1 - 10^{-b(M_{max}-M_{min})}} (10^{-b(\frac{3.346629+ln(x)}{0.475} - M_{min})} - 10^{-b(M_{max}-M_{min})}) \tag{A.20}$$

The rate of exceedance for PGA level equal to 0.1 is (considering that $(3.346629 + ln(0.1))/0.475 \approx$ 2.2... and therefore using equation A.19):

$$\lambda(PGA \geq 0.1) = \frac{10^{-2}}{1 - 10^{-3}}(1 - 10^{-3}) = 10^{-2} \tag{A.21}$$

and the corresponding probability of exceedance in a time period of 1 year is (rounding to 5 digits):

$$P(PGA \geq 0.1) = 1 - exp(-10^{-2} * 1.0) = 0.00995 \tag{A.22}$$

The rate of exceedance for PGA level equal to 0.4 is (considering that $(3.346629 + ln(0.4))/0.475 =$ 5.11650 and therefore using equation A.20):

$$\lambda(PGA \geq 0.4) = \frac{10^{-2}}{1 - 10^{-3}}(10^{-(5.11650-4.0)} - 10^{-3}) = 0.00076 \tag{A.23}$$

and the corresponding probability of exceedance in a time period of 1 year is (rounding to 5 digits):

$$P(PGA \geq 0.4) = 1 - exp(-0.00076 * 1.0) = 0.00076 \tag{A.24}$$

The rate of exceedance for PGA level equal to 0.6 is (considering that $(3.346629 + ln(0.6))/0.475 =$ 5.97011 and therefore using equation A.20):

$$\lambda(PGA \geq 0.6) = \frac{10^{-2}}{1 - 10^{-3}}(10^{-(5.97011-4.0)} - 10^{-3}) = 9.7 \cdot 10^{-5} \tag{A.25}$$

and the corresponding probability of exceedance in a time period of 1 year is (rounding to 5 digits):

$$P(PGA \geq 0.6) = 1 - exp(-9.7 \cdot 10^{-5} * 1.0) = 9.7 \cdot 10^{-5} \tag{A.26}$$

The rate of exceedance for PGA level equal to 1.0 is 0, considering that $(3.346629 + ln(1.0))/0.475 =$ 7.045531 and therefore greater than the considered maximum magnitude: and the corresponding probability of exceedance in a time period of 1 year is also 0:

$$P(PGA \geq 1.0) = 1 - exp(-0 * 1.0) = 0 \tag{A.27}$$

Hazard curve calculation with single area source, single magnitude MFD, and ground motion truncation level = 0

NOTE: This test is meant to exercise the hazard curve calculator (both classical and event-based) with an area source

This test assumes a source model consisting of a single area source, whose polygon defines a circle of 5 km radius. The MFD contains a single magnitude ($M = 4$) associated to an annual rate of exceedance ($\lambda(m = 4.0)$) equal to 1. The hypocentral depth of the area source is set to 0.5 and the upper seismogenic depth to 0 km. Aspect ratio is assumed equal to 1. Under this conditions all ruptures reach the surface. The lower seismogenic depth can be set to a value greater or equal then 1.0 km. In this case all ruptures have a square shape and have uniform length of 1 km.

With these assumptions, the probability density function for magnitude is a dirac delta function as given in equation A.3.

Assuming a site located in the centre of the area source, and indicating with l the rupture length and with R the radius of the area source, the probability density function for the closest distance to the rupture is given by:

$$f(r) = \begin{cases} \frac{2r}{R^2} + \frac{l}{2R^2} & \text{if } r \leq R - \frac{l}{2} \\ \frac{1}{R} & \text{if } R - \frac{l}{2} < r \leq R \\ 0 & \text{if } r > R \end{cases} \tag{A.28}$$

Assuming a truncation level equal to 0, we can write equation A.1 (considering a single source and $\lambda(m \geq M_{min}) = \lambda(m = 4.0) = 1$) as:

$$\lambda(IM \geq x) = \int_0^{R-\frac{l}{2}} \left[\frac{2r}{R^2} + \frac{l}{2R^2} \right] H(-0.624 + 4.0 - 2.1ln(r + exp(1.29649 + 0.25 \cdot 4.0)) - ln(x))dr +$$

$$\frac{1}{R} \int_{R-\frac{l}{2}}^{R} H(-0.624 + 4.0 - 2.1ln(r + exp(1.29649 + 0.25 \cdot 4.0)) - ln(x))dr \tag{A.29}$$

The Heaviside function is non-zero if and only if $-0.624 + 4.0 - 2.1ln(r + exp(1.29649 + 0.25 \cdot 4.0)) - ln(x) \geq 0$, that is $r \leq e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649}$. If $R \leq e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649}$, then we can solve equation A.29 as:

$$\lambda(IM \geq x) = \frac{1}{R^2}(R - \frac{l}{2})^2 + \frac{l}{2R^2}(R - \frac{l}{2}) + \frac{l}{2R} \tag{A.30}$$

if $R - \frac{l}{2} < e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649} < R$, then we can solve equation A.29 as:

$$\lambda(IM \geq x) = \frac{1}{R^2}(R - \frac{l}{2})^2 + \frac{l}{2R^2}(R - \frac{l}{2}) + \frac{1}{R}(e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649} - R + \frac{l}{2}) \tag{A.31}$$

if $0 < e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649} < R - \frac{l}{2}$, then we can solve equation A.29 as:

$$\lambda(IM \geq x) = \frac{1}{R^2}(e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649})^2 + \frac{l}{2R^2}(e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649}) \tag{A.32}$$

The rate of exceedance of PGA value equal to 0.1 is (considering that $R = 5 < e^{\frac{3.376 - ln(0.1)}{2.1}} - e^{2.29649} = 5.00145$, and therefore using equation A.30) :

$$\lambda(IM \geq 0.1) = \frac{1}{5^2}(5 - 0.5)^2 + \frac{1}{2 \cdot 5^2}(5 - 0.5) + \frac{1}{2 \cdot 5} = 1.0 \tag{A.33}$$

and the corresponding probability of exceedance in a time span of 1 year is:

$$P(IM \geq 0.1) = 1 - exp(-1 \cdot 1) = 0.63212 \tag{A.34}$$

The rate of exceedance of PGA value equal to 0.12 is (considering that $e^{\frac{3.376-ln(0.12)}{2.1}} - e^{2.29649} = 3.75902 < R - \frac{l}{2} = 4.5$ and therefore using equation A.32):

$$\lambda(IM \geq 0.12) = \frac{1}{5^2}(e^{\frac{3.376-ln(0.12)}{2.1}} - e^{2.29649})^2 + \frac{1}{2 \cdot 5^2}(e^{\frac{3.376-ln(0.12)}{2.1}} - e^{2.29649}) = 0.640389 \quad \text{(A.35)}$$

and the corresponding probability of exceedance in a time span of 1 year is:

$$P(IM \geq 0.12) = 1 - exp(-0.640389 \cdot 1) = 0.47291 \quad \text{(A.36)}$$

The rate of exceedance of PGA value equal to 0.2 is (considering that $e^{\frac{3.376-ln(0.2)}{2.1}} - e^{2.29649} = 0.80123 < R - \frac{l}{2} = 4.5$ and therefore using equation A.32):

$$\lambda(IM \geq 0.2) = \frac{1}{5^2}(e^{\frac{3.376-ln(0.2)}{2.1}} - e^{2.29649})^2 + \frac{1}{2 \cdot 5^2}(e^{\frac{3.376-ln(0.2)}{2.1}} - e^{2.29649}) = 0.04170 \quad \text{(A.37)}$$

and the corresponding probability of exceedance in a time span of 1 year is:

$$P(IM \geq 0.2) = 1 - exp(-0.04170 \cdot 1) = 0.04084 \quad \text{(A.38)}$$

Hazard curve calculation with single simple fault source, single magnitude MFD, and ground motion truncation level = 0

NOTE: This test is meant to exercise the hazard curve calculator (both classical and event-based) with a simple fault source

This test assumes a source model consisting of a single simple fault source, characterized by a single magnitude MFD ($M = 4$) with annual occurrence rate ($\lambda(m = 4)$) equal to 1. The fault source has an upper seismogenic depth of 0 km, and a lower seismogenic depth of 1 km. Dip is 90 degrees. Aspect ratio is assumed 1.

Under these assumptions the probability density function for magnitude is given by equation , while the probability density function for the closest distance to the rupture (assuming the site to be in the middle point of the fault trace) is:

$$f(r) = \begin{cases} \frac{l}{L-l}\delta(r) & \text{if } r = 0 \\ \frac{2}{L-l} & \text{if } 0 < r \leq \frac{L}{2} - l \end{cases} \tag{A.39}$$

where l is the rupture length, and L is the fault length.

Assuming truncation level equal to 0, we can compute the rate of exceedance of an intensity measure level as:

$$\lambda(PGA \geq x) = \frac{l}{L-l} + \frac{2}{L-l}min(\frac{L}{2} - l, e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649}) \tag{A.40}$$

Assuming L = 10, and given that the rupture length $l = 1$ (because aspect ratio is 1), for x = 0.1, the rate of exceedance is, considering that $\frac{L}{2} - l = 4 < e^{\frac{3.376 - ln(0.1)}{2.1}} - e^{2.29649} = 5.00145$:

$$\lambda(PGA \geq 0.1) = \frac{1}{1/9} + \frac{2}{9}4 = 1 \tag{A.41}$$

and therefore the probability of exceedance in 1 year is:

$$P(PGA \geq 0.1) = 1 - exp(-1 \cdot 1) = 0.63212 \tag{A.42}$$

The rate of exceedance for x = 0.12 ($\frac{L}{2} - l = 4 > e^{\frac{3.376 - ln(0.12)}{2.1}} - e^{2.29649} = 3.7590$) is:

$$\lambda(PGA \geq 0.12) = \frac{1}{9} + \frac{2}{9}3.7590 = 0.9464 \tag{A.43}$$

and the corresponding probability of exceedance in 1 year is:

$$P(PGA \geq 0.12) = 1 - exp(-0.9464 \cdot 1) = 0.61186 \tag{A.44}$$

The rate of exceedance for x = 0.2 ($\frac{L}{2} - l = 4 > e^{\frac{3.376 - ln(0.2)}{2.1}} - e^{2.29649} = 0.801227$) is:

$$\lambda(PGA \geq 0.2) = \frac{1}{9} + \frac{2}{9}0.801227 = 0.28916 \tag{A.45}$$

and the corresponding probability of exceedance in 1 year is:

$$P(PGA \geq 0.2) = 1 - exp(-0.28916 \cdot 1) = 0.25110 \tag{A.46}$$

Hazard curve calculation with single complex fault source, single magnitude MFD, and ground motion truncation level = 0

NOTE: This test is meant to exercise the hazard curve calculator (both classical and event-based) with a complex fault source

This test assumes a complex fault source composed of top and bottom edges. Both the top and bottom edges have the same length, and same latitudes and longitudes, but they are shifted in depth. Both the two edges consist of two line segments, one horizontal (L_1), and a second , of length L_2, inclined of an angle α with respect to the earth surface. The shift in depth between the top and bottom edges is of 1 km.

The source is associated to a single magnitude MFD, defining a single magnitude value equal to 4, and with corresponding annual rate $\lambda(m = 4)$, equal to 1. The aspect ratio is assumed 1. The rupture area is 1 squared km (as predicted by equation A.8). Ruptures floating along the first section have a square shape, but as soon as they start propagating along the inclined section, they start acquiring the shape of a parallelogram. This means that while on the first section the rupture length is 1 km, on the second section the rupture length is $l = \frac{A}{w \cdot sin(90-\alpha)}$, where A is the rupture area, w is the fault width (that is the shift between top and bottom edges), and α is the inclination of the second section with respect to the horizontal direction.

The probability density function for magnitude is given by equation A.3. Assuming a site located at the beginning of the first segment (and assuming the first segment of the top edge starting at the earth surface), the probability density function for the closest distance to the rupture is:

$$f(r) = \begin{cases} \frac{1}{L_1+L_2-l} & \text{if } 0 \le r \le L_1 \\ \frac{r}{(L_1+L_2-l)\sqrt{L_1^2 cos^2\alpha - L_1^2 + r^2}} & \text{if } L_1 < r \le r_{max} \end{cases} \tag{A.47}$$

where:

$$r_{max} = \sqrt{(L_2-l)^2 + L_1^2 + 2L_1(L_2-l)cos\alpha} \tag{A.48}$$

Assuming $L_1 = 3.0$ km, $L_2 = 3.0$ km, $\alpha = 30.0$ degrees, and $w = 1$, the maximum closest distance to the rupture is 4.68973.

Assuming truncation level = 0, the rate of exceedance of a PGA level x is (from equation A.1):

$$\lambda(PGA \ge x) = \int_0^{min(r_{max}, e^{\frac{3.376 - ln(x)}{2.1}} - e^{2.29649})} f(r)dr \tag{A.49}$$

Considering x = 0.1, $r_{max} = 4.68973 < e^{\frac{3.376 - ln(0.1)}{2.1}} - e^{2.29649} = 5.00145$, and therefore:

$$\lambda(PGA \ge 0.1) = \frac{L_1}{L_1+L_2-l} + \frac{1}{L_1+L_2-l}\left[\sqrt{L_1^2 cos^2\alpha - L_1^2 + r^2}\right]_{L_1}^{r_{max}} =$$

$$\frac{L_1}{L_1+L_2-l} + \frac{1}{L_1+L_2-l}\left[\sqrt{L_1^2 cos^2\alpha - L_1^2 + (L_2-l)^2 + L_1^2 + 2L_1(L_2-l)cos\alpha} - L_1 cos\alpha\right] =$$

$$\frac{L_1}{L_1+L_2-l} + \frac{1}{L_1+L_2-l}\left[\sqrt{(L_1 cos\alpha + (L_2-l))^2} - L_1 cos\alpha\right] =$$

$$\frac{L_1}{L_1+L_2-l} + \frac{1}{L_1+L_2-l}[L_1 cos\alpha + L_2 - l - L_1 cos\alpha] = \tag{A.50}$$

The corresponding probability of exceedance in 1 year is therefore:

$$P(PGA \ge 0.1) = 1 - exp(-1 \cdot 1) = 0.632120 \tag{A.51}$$

Considering x = 0.12, $r_{max} = 4.68973 > e^{\frac{3.376 - ln(0.12)}{2.1}} - e^{2.29649} = 3.7590 > L_1 = 3.0$, and therefore:

$$\lambda(PGA \geq 0.12) = \frac{L_1}{L_1 + L_2 - l} + \frac{1}{L_1 + L_2 - l}\left[\sqrt{L_1^2 cos^2\alpha - L_1^2 + 3.7590^2} - L_1 cos\alpha\right] =$$

$$\frac{3}{3 + 3 - \frac{2}{\sqrt{3}}} + \frac{1}{3 + 3 - \frac{2}{\sqrt{3}}}\left[\sqrt{9 \cdot \frac{3}{4} - 9 + 3.7590^2} - \frac{3\sqrt{3}}{2}\right] = 0.79431 \tag{A.52}$$

The corresponding probability of exceedance in 1 year is therefore:

$$P(PGA \geq 0.12) = 1 - exp(-0.79431 \cdot 1) = 0.54811 \tag{A.53}$$

Considering x = 0.2, $r_{max} = 4.68973 > e^{\frac{3.376 - ln(0.2)}{2.1}} - e^{2.29649} = 0.801227 < L_1 = 3.0$, and therefore:

$$\lambda(PGA \geq 0.2) = \frac{0.801227}{L_1 + L_2 - l} = \frac{0.801227}{3 + 3 - \frac{2}{\sqrt{3}}} = 0.16536 \tag{A.54}$$

The corresponding probability of exceedance in 1 year is therefore:

$$P(PGA \geq 0.2) = 1 - exp(-0.16536 \cdot 1) = 0.15241 \tag{A.55}$$

Hazard curve calculation with source model consisting of multiple sources

NOTE: This test is meant to exercise the parallelization strategy of the hazard curve calculator (both classical and event-based). Given that the source model consists of multiple sources, a task for each source can be defined, and therefore the task creation and aggregation of the results can be tested.

This test assumes a source model consisting of 2 seismic sources, as described in tests 4 and 5, that is a simple fault source and a complex fault source. For the simple fault source as described in test 4, the rates of exceedance for PGA levels of 0.1, 0.12 and 0.2 are 1.0, 0.9464, 0.28916, respectively. For the complex fault source as described in test 6, the rate of exceedance for the same PGA levels are: 1.0, 0.79431, 0.16536.

Using equation A.1 the rates of exceedance in case of a source model consisting of the two above mentioned sources are: (1+1)=2.0, (0.9464 + 0.79431) = 1.74071, (0.28916+0.16536) = 0.45452. The corresponding probabilities of exceedance in a period of 1 year are:

$$
\begin{aligned}
P(PGA \geq 0.10) &= 1 - exp(-2 \cdot 1) = 0.86466 \\
P(PGA \geq 0.12) &= 1 - exp(-1.74071 \cdot 1) = 0.82460 \\
P(PGA \geq 0.20) &= 1 - exp(-0.45452 \cdot 1) = 0.36525
\end{aligned}
$$

$$(A.56)$$

A.4 Hazard curve calculation with logic-trees

Hazard curve calculation with logic tree containing multiple source model

NOTE: This test is meant to exercise the hazard curve calculator (both classical and event-based) when considering a non-trivial logic tree (that is a logic tree with more than one path). The test should check that the correct solution for each path (when using Path Enumeration) is obtained and that the mean hazard curve is correctly computed. This test should also check that Monte Carlo Sampling and Path Enumeration should provide the same mean hazard curve.

This test assumes a logic tree defining 2 source models. Source model 1 consists of a simple and a complex fault sources, as described in test 6, while source model 2 consists of a single simple fault source as described in test 4.

For PGA levels equal to 0.1, 0.12, 0.2, the probabilities of exceedance from source model 1 are:

$$
\begin{aligned}
P(PGA \geq 0.10) &= 0.86466 \\
P(PGA \geq 0.12) &= 0.82460 \\
P(PGA \geq 0.20) &= 0.36525
\end{aligned}
$$

(A.57)

the probabilities of exceedance from source model 2 are instead:

$$
\begin{aligned}
P(PGA \geq 0.10) &= 0.63212 \\
P(PGA \geq 0.12) &= 0.61186 \\
P(PGA \geq 0.20) &= 0.25110
\end{aligned}
$$

(A.58)

Assuming source model 1 to be assigned to probability value of 0.7 and source model 2 to be assigned to probability value of 0.3, the mean hazard curve is:

$$
\begin{aligned}
P(PGA \geq 0.10) &= 0.7*0.86466 + 0.3*0.63212 = 0.794898 \\
P(PGA \geq 0.12) &= 0.7*0.82460 + 0.3*0.61186 = 0.760778 \\
P(PGA \geq 0.20) &= 0.7*0.36525 + 0.3*0.25110 = 0.331005
\end{aligned}
$$

(A.59)

Hazard curve calculation with logic tree containing single source model and a and b Gutenberg Richter absolute uncertainties

NOTE: This test is meant to exercise the 'uncertaintyType="abGRAbsolute" ' option in the logic tree construction.

This test assumes a logic tree defining a single source model. The source model contains a single point source defining a Gutenberg Richter magnitude frequency distribution, as defined in test 2. The logic tree defines absolute uncertainties on the Gutenberg Richter a and b values:

$$
\begin{aligned}
probability &= 0.2 - a = 2.2, b = 0.8 \\
probability &= 0.6 - a = 2.0, b = 1.0 \\
probability &= 0.2 - a = 1.8, b = 1.2
\end{aligned}
$$

$$(A.60)$$

For the case $a = 2.2$ and $b = 0.8$, the rates of exceedance for PGA levels of 0.1, 0.4, 0.6, 1.0 are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= \frac{10^{2.2-0.8\cdot4.0}}{1 - 10^{-0.8(7.0-4.0)}}(1 - 10^{-0.8(7.0-4.0)}) = 0.1 \\
\lambda(PGA \geq 0.4) &= \frac{10^{2.2-0.8\cdot4.0}}{1 - 10^{-0.8(7.0-4.0)}}(10^{-0.8(5.11650-4.0)} - 10^{-0.8(7.0-4.0)}) = 0.012439 \\
\lambda(PGA \geq 0.6) &= \frac{10^{2.2-0.8\cdot4.0}}{1 - 10^{-0.8(7.0-4.0)}}(10^{-0.8(5.97011-4.0)} - 10^{-0.8(7.0-4.0)}) = 0.002265 \\
\lambda(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$(A.61)$$

The corresponding probabilities are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.095163 \\
P(PGA \geq 0.4) &= 0.012362 \\
P(PGA \geq 0.6) &= 0.002262 \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$(A.62)$$

For the case $a = 2.0$, $b = 1$, the rates of exceedance, (and the corresponding probabilities) are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= 10^{-2} \\
\lambda(PGA \geq 0.4) &= 0.00076 \\
\lambda(PGA \geq 0.6) &= 9.7 \cdot 10^{-5} \\
\lambda(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$(A.63)$$

The corresponding probabilities are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.009950 \\
P(PGA \geq 0.4) &= 0.00076 \\
P(PGA \geq 0.6) &= 9.99995 \cdot 10^{-6} \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$(A.64)$$

For the case $a = 1.8$ and $b = 1.2$, the rates of exceedance are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= \frac{10^{1.8-1.2 \cdot 4.0}}{1-10^{-1.2(7.0-4.0)}}(1-10^{-1.2(7.0-4.0)}) = 0.001 \\
\lambda(PGA \geq 0.4) &= \frac{10^{1.8-1.2 \cdot 4.0}}{1-10^{-1.2(7.0-4.0)}}(10^{-1.2(5.11650-4.0)} - 10^{-1.2(7.0-4.0)}) = 4.5490 \cdot 10^{-5} \\
\lambda(PGA \geq 0.6) &= \frac{10^{1.8-1.2 \cdot 4.0}}{1-10^{-1.2(7.0-4.0)}}(10^{-1.2(5.97011-4.0)} - 10^{-1.2(7.0-4.0)}) = 4.07366 \cdot 10^{-6} \\
\lambda(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$(A.65)$$

The corresponding probabilities are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.0009995 \\
P(PGA \geq 0.4) &= 4.5489 \cdot 10^{-5} \\
P(PGA \geq 0.6) &= 4.07365 \cdot 10^{-6} \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$(A.66)$$

Hazard curve calculation with logic tree containing single source model and absolute uncertainties on Gutenberg Richter Maximum Magnitude

NOTE: This test is meant to exercise the 'uncertaintyType="maxMagGRAbsolute" ' option in the logic tree construction

This test assumes a logic tree defining a single source model. The source model contains a single point source defining a Gutenberg Richter magnitude frequency distribution, as defined in test 2. The logic tree defines absolute uncertainties on the Gutenberg Richter maximum magnitude:

$$probability = 0.5 - Mmax = 7.0$$
$$probability = 0.5 - Mmax = 7.5$$

$$\text{(A.67)}$$

For the case $Mmax = 7.0$, the probabilities of exceedance are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.00995 \\
P(PGA \geq 0.4) &= 0.00076 \\
P(PGA \geq 0.6) &= 9.7 \cdot 10^{-5} \\
P(PGA \geq 1.0) &= 0
\end{aligned}
$$

$$\text{(A.68)}$$

For the case $Mmax = 7.5$, the rates of exceedance are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= \frac{10^{-2}}{1 - 10^{-3.5}}(1 - 10^{-3.5}) = 0.01 \\
\lambda(PGA \geq 0.4) &= \frac{10^{-2}}{1 - 10^{-3.5}}(10^{-(5.11650-4.0)} - 10^{-3.5}) = 0.000762 \\
\lambda(PGA \geq 0.6) &= \frac{10^{-2}}{1 - 10^{-3.5}}(10^{-(5.97011-4.0)} - 10^{-3.5}) = 0.000104 \\
\lambda(PGA \geq 1.0) &= \frac{10^{-2}}{1 - 10^{-3.5}}(10^{-(7.5-4.0)} - 10^{-3.5}) = 0.0
\end{aligned}
$$

$$\text{(A.69)}$$

The corresponding probabilities of exceedance in 1 year are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.00995 \\
P(PGA \geq 0.4) &= 0.00076 \\
P(PGA \geq 0.6) &= 0.000104 \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$\text{(A.70)}$$

Hazard curve calculation with logic tree containing source model and relative uncertainties on Gutenberg Richter b value

NOTE: This test is meant to exercise the 'uncertaintyType="bGRRelative" ' option in the logic tree construction

This test assumes a logic tree defining a single source model. The source model contains a single point source defining a Gutenberg Richter magnitude frequency distribution, as defined in test 2. The logic tree defines relative uncertainties on the Gutenberg Richter b value:

$$
\begin{aligned}
probability &= 0.5 \quad - \quad \delta b = 0.0 \\
probability &= 0.5 \quad - \quad \delta b = +0.4
\end{aligned}
$$

(A.71)

For the case $\delta b = 0.0$, the probabilities of exceedance are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.00995 \\
P(PGA \geq 0.4) &= 0.00076 \\
P(PGA \geq 0.6) &= 9.7 \cdot 10^{-5} \\
P(PGA \geq 1.0) &= 0
\end{aligned}
$$

(A.72)

For the case $\delta b = +0.4$, the a value of the Gutenberg Richter magnitude frequency distribution is changed to conserve the total moment rate. The new a value is 4.243. The rates of exceedance are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= \frac{10^{4.243-1.4 \cdot 4.0}}{1-10^{-1.4(7-4)}}(1 - 10^{-1.4(7-4)}) = 0.04395 \\
\lambda(PGA \geq 0.4) &= \frac{10^{4.243-1.4 \cdot 4.0}}{1-10^{-1.4(7-4)}}(10^{-1.4(5.11650-4.0)} - 10^{-1.4(7-4)}) = 0.0012 \\
\lambda(PGA \geq 0.6) &= \frac{10^{4.243-1.4 \cdot 4.0}}{1-10^{-1.4(7-4)}}(10^{-1.4(5.97011-4.0)} - 10^{-1.4(7-4)}) = 7.394 \cdot 10^{-5} \\
\lambda(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

(A.73)

The corresponding probabilities of exceedance in 1 year are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.043 \\
P(PGA \geq 0.4) &= 0.0012 \\
P(PGA \geq 0.6) &= 7.394 \cdot 10^{-5} \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

(A.74)

Hazard curve calculation with logic tree containing single source model and relative uncertainties on Gutenberg Richter Maximum Magnitude

NOTE: This test is meant to exercise the 'uncertaintyType="maxMagGRRelative" ' option in the logic tree construction. It also allows to check the calculation of mean and quantile hazard curves.

This test assumes a logic tree defining a single source model. The source model contains a single point source defining a Gutenberg Richter magnitude frequency distribution, as defined in test 2. The logic tree defines relative uncertainties on the Gutenberg Richter maximum magnitude:

$$
\begin{aligned}
probability &= 0.2 &-& \quad \delta M = +0.5 \\
probability &= 0.6 &-& \quad \delta M = 0.0 \\
probability &= 0.2 &-& \quad \delta M = -0.5
\end{aligned}
$$

$$(A.75)$$

For the case $\delta M = +0.5$, the new a value is 1.7438. The corresponding rates of exceedance for PGA levels of 0.1, 0.4, 0.6, 1.0 are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= \frac{10^{1.7438-1.0\cdot4.0}}{1-10^{-1.0(7.5-4.0)}}(1-10^{-1.0(7.5-4.0)}) = 0.0055 \\
\lambda(PGA \geq 0.4) &= \frac{10^{1.7438-1.0\cdot4.0}}{1-10^{-1.0(7.5-4.0)}}(10^{-1.0(5.11650-4.0)}-10^{-1.0(7.5-4.0)}) = 0.00042 \\
\lambda(PGA \geq 0.6) &= \frac{10^{1.7438-1.0\cdot4.0}}{1-10^{-1.0(7.5-4.0)}}(10^{-1.0(5.97011-4.0)}-10^{-1.0(7.5-4.0)}) = 5.77\cdot10^{-5} \\
\lambda(PGA \geq 1.0) &= 0.0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (A.76)
\end{aligned}
$$

The corresponding probabilities are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.0055 \\
P(PGA \geq 0.4) &= 0.00042 \\
P(PGA \geq 0.6) &= 5.77\cdot10^{-5} \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
$$

$$(A.77)$$

For the case $\delta M = 0.0$, the probabilities of exceedance are equal to the one provided in test 2, that is:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.00995 \\
P(PGA \geq 0.4) &= 0.00076 \\
P(PGA \geq 0.6) &= 9.7\cdot10^{-5} \\
P(PGA \geq 1.0) &= 0
\end{aligned}
$$

$$(A.78)$$

For the case $\delta M = -0.5$, the new a value is 2.261. The corresponding rates of exceedance for PGA levels of 0.1, 0.4, 0.6, 1.0 are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= \frac{10^{2.261-1.0\cdot4.0}}{1-10^{-1.0(6.5-4.0)}}(1-10^{-1.0(6.5-4.0)}) = 0.018 \\
\lambda(PGA \geq 0.4) &= \frac{10^{2.261-1.0\cdot4.0}}{1-10^{-1.0(6.5-4.0)}}(10^{-1.0(5.11650-4.0)}-10^{-1.0(6.5-4.0)}) = 0.0013 \\
\lambda(PGA \geq 0.6) &= \frac{10^{2.261-1.0\cdot4.0}}{1-10^{-1.0(6.5-4.0)}}(10^{-1.0(5.97011-4.0)}-10^{-1.0(6.5-4.0)}) = 0.00014 \\
\lambda(PGA \geq 1.0) &= 0.0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (A.79)
\end{aligned}
$$

The corresponding probabilities are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.018 \\
P(PGA \geq 0.4) &= 0.0013 \\
P(PGA \geq 0.6) &= 0.00014 \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
\tag{A.80}
$$

The mean probabilies of exceedance are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.2*0.0055 + 0.6*0.00995 + 0.2*0.018 = 0.01067 \\
P(PGA \geq 0.4) &= 0.2*0.00042 + 0.6*0.00076 + 0.2*0.0013 = 0.0008 \\
P(PGA \geq 0.6) &= 0.2*5.77 \cdot 10^{-5} + 0.6*9.7 \cdot 10^{-5} + 0.2*0.00014 = 9.774e - 05 \\
P(PGA \geq 1.0) &= 0.2*0.0 + 0.6*0.0 + 0.2*0.0 = 0.0
\end{aligned}
\tag{A.81}
$$

The probabilies of exceedance for quantile level 0.1 are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 0.0055 \\
P(PGA \geq 0.4) &= 0.00042 \\
P(PGA \geq 0.6) &= 5.77 \cdot 10^{-5} \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
\tag{A.82}
$$

The probabilies of exceedance for quantile level 0.9 are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= \frac{0.018 - 0.00995}{1.0 - 0.8}(0.9 - 0.8) + 0.00995 = 0.013975 \\
P(PGA \geq 0.4) &= \frac{0.0013 - 0.00076}{1.0 - 0.8}(0.9 - 0.8) + 0.00076 = 0.00103 \\
P(PGA \geq 0.6) &= \frac{0.00014 - 9.7 \cdot 10^{-5}}{1.0 - 0.8}(0.9 - 0.8) + 9.7 \cdot 10^{-5} = 0.0001185 \\
P(PGA \geq 1.0) &= 0.0
\end{aligned}
\tag{A.83}
$$

Hazard curve calculation with source model consisting of multiple sources belonging to different tectonic region types and therefore requiring different GMPEs

NOTE: This test is meant to check that GMPEs are correctly associated to each source based on the tectonic region type.

This test considers a source model consisting of two point sources. The first point source is the one described in test 1, and the associated rates of exceedance (using Sadight et al 1997 as GMPE, and truncation level equal to 2) are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= 0.6107 \\
\lambda(PGA \geq 0.4) &= 0.0605 \\
\lambda(PGA \geq 0.6) &= 0.0069
\end{aligned}
$$

(A.84)

The second source is again a point source generating a single rupture of magnitude 4.5 associated to an annual occurrence rate equal to 1. The rupture aspect ratio is assumed equal to 1, and the area predicted by the Peer scaling relationship is therefore $10^{0.5}$. The rupture length and width are equal to $10^{0.25}$. The rupture is assumed vertical ($dip = 90.0$) and the hypocenter is set at $\frac{10^{0.25}}{2}$, so that the rupture reaches the surface. This rupture is associated to the Boore and Atkinson 2008 GMPE, which uses R_{JB}. Assuming the site of interest to be on the same location of the point source, $R_{JB} = 0$.

Assuming $m = 4.5$, $R_{JB} = 0$, $Vs_{30} = 760.0$, and $rake = 0$ (that is strike slip event), the median PGA is:

$$
\overline{ln(PGA; m = 4.5, r = 0)} = -1.86841
$$

(A.85)

and the total standard deviation is:

$$
\sigma = 0.564
$$

(A.86)

The rates of exceedance for PGA levels equal to 0.1, 0.4, 0.6 are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= 1 - \frac{\Phi(\frac{ln(0.1)+1.86841}{0.564}) - 0.0228}{0.9545} = 0.79266 \\
\lambda(PGA \geq 0.4) &= 1 - \frac{\Phi(\frac{ln(0.4)+1.86841}{0.564}) - 0.0228}{0.9545} = 0.02409 \\
\lambda(PGA \geq 0.6) &= 0.0
\end{aligned}
$$

(A.87)

The rate of exceedance for PGA = 0.6 is zero because $\frac{ln(0.6)+1.86841}{0.564} = 2.4$ which is larger than the truncation level, that is 2.0. The corresponding probabilities of exceedance in a time period of 1 year ($T = 1.0$) considering the two sources are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 1 - exp(-(0.6107 + 0.79266) * 1.0) = 0.75423 \\
P(PGA \geq 0.4) &= 1 - exp(-(0.0605 + 0.02409) * 1.0) = 0.08111 \\
P(PGA \geq 0.6) &= 1 - exp(-(0.0069 + 0.0) * 1.0) = 0.00688
\end{aligned}
$$

(A.88)

Hazard curve calculation (event based approach) with ground motion correlation included

NOTE: This test is meant to check hazard curve calculation on a single site with ground motion correlation - that is using event based approach.

This test consider a source model consisting of a single point source as defined in test 12 (point source with $m = 4.5$). The GMPE used is Boore and Atkinson 2008. The tests assumes event based approach to be used with spatial correlation (Jayaram and Backer 2009 model).

The rates of exceedance predicted by the source model are:

$$
\begin{aligned}
\lambda(PGA \geq 0.1) &= 0.79266 \\
\lambda(PGA \geq 0.4) &= 0.02409 \\
\lambda(PGA \geq 0.6) &= 0.0
\end{aligned}
$$

(A.89)

and the corresponding probabilities of exceedance are:

$$
\begin{aligned}
P(PGA \geq 0.1) &= 1 - exp(-0.79266 * 1.0) = 0.54736 \\
P(PGA \geq 0.4) &= 1 - exp(-0.02409 * 1.0) = 0.02380 \\
P(PGA \geq 0.6) &= 1 - exp(-0.0 * 1.0) = 0
\end{aligned}
$$

(A.90)

Bibliography

Books

Myers, G. J., C. Sandler, and T. Badgett (2012). *The art of software testing*. Wiley and Sons, Inc. (cited on pages 5, 9).

Articles

Boore, D. M. and G. M. Atkinson (Feb. 2008). "Ground-Motion Prediction Equations for the Average Horizontal Component of PGA, PGV, and 5%-Damped PSA at Spectral Periods between 0.01 s and 10.0 s". In: *Earthquake Spectra* 24.1, pages 99–138 (cited on page 17).

Frankel, A. (1995). "Mapping Seismic Hazard in the Central and Eastern United States". In: *Seismological Research Letters* 66.4, pages 8–21 (cited on page 17).

Sadigh K., C. -Y., J. A. Chang, F. Egan, Makdisi, and R. R. Youngs (1997). "Attenuation relationships for shallow crustal earthquakes based on California strong motion data". In: *Seismological Research Letters* 68, pages 180–189 (cited on pages 33, 35).

Wells, D. L. and K. J. Coppersmith (1994). "New Empirical Relationships among Magnitude, Rupture Length, Rupture Width, Rupture Area, and Surface Displacement". In: *Bull. Seism. Soc. Am.* 84.4, pages 974–1002 (cited on page 18).

Other Sources

Berkes, P. (2012). *Writing robust scientific code with testing (and Python)*. Euroscipy. URL: http://archive.euroscipy.org/talk/6634 (cited on page 5).

Petersen, M. D., A. D. Frankel, S. C. Harmsen, C. S. Mueller, K. M. Haller, R. L. Wheeler, R. L. Wesson, Y. Yzeng, O. S. Boys, D. M. Perkins, N. Luco, E. H. Field, C. J. Wills, and K. S. Rukstales (2008). *Documentation for the 2008 Update of the United States National Seismic Hazard Maps*. Open File Report 2008-1128. U.S. Department of the Interior, U.S. Geological Survey (cited on pages 17, 18).

Thomas, P., I. Wong, and N. A. Abrahamson (May 2010). *Verification of Probabilistic Seismic Hazard Analysis Computer Programs*. PEER Report 2010/106. College of Engineering, University of California, Berkeley: Pacific Earthquake Engineering Centre (cited on pages 14, 15, 31, 33–36).