OpenQuake Engine: An Open Hazard (and Risk) Software for the Global Earthquake Model

by M. Pagani, D. Monelli, G. Weatherill, L. Danciu, H. Crowley, V. Silva, P. Henshaw, L. Butler, M. Nastasi, L. Panzeri, M. Simionato, and D. Vigano

INTRODUCTION

Since its inception in the 1960s, probabilistic seismic-hazard analysis (PSHA) (Cornell, 1968; McGuire, 2004, 2008) has emerged as the principal methodology for assessing the potential hazard posed by earthquake ground motion in a broad range of contexts. Seismic-hazard analysis serves different needs coming from a wide spectrum 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.

End users of seismic-hazard analyses from different sectors of industry may often have specific requirements in terms of the types of results and, as a consequence, in terms of the methodologies preferred for calculation. A large majority of studies for the analysis of structural and geotechnical systems require the calculation of a target response spectrum derived from PSHA results (e.g., Lin *et al.*, 2013). Often the calculation of uniform hazard spectra is performed in conjunction with a disaggregation analysis, which in the simplest cases highlights the combinations of magnitude and distances, providing the largest contributions to a specific level of hazard for a particular intensity measure type, such as the spectral acceleration for a period close to the fundamental elastic period of a structure (Bazzurro and Cornell, 1999; Pagani and Marcellini, 2007). In contrast, in the insurance sector it is more common to use stochastic methodologies (e.g., Weatherill and Burton, 2010; Musson, 2012) to produce multiple realizations of the likely earthquake activity that may be pertinent to a portfolio of assets. Monte Carlo–based methods can provide results in a form that offers a practical comparison with past events and can better account for the temporal and spatial variability of earthquake shaking occurring on a distributed set of sites, for example, by taking into account the correlation of ground-motion intraevent residuals (Crowley and Bommer, 2006; Park *et al.*, 2007). At the level of public policy, many countries now utilize PSHA as the primary methodology for the generation of national seismic-hazard maps. Often, these hazard maps are adopted in national building codes either for a direct definition of ground-motion parameters that delimit the scale and shape of the design spectrum or simply for the classification of the territory into a finite number of zones for which different levels of seismic-resistant design are required.

The maturity and importance of the PSHA methodology and the relevance of its results have led to the emergence of numerous software applications to implement the hazard calculation. The first computer codes for PSHAs date back to the 1970s with the releases of EQRISK (McGuire, 1976) and FRISK (McGuire, 1978). In the subsequent decades, several research groups released open and/or freely available software. Examples are the family of SeisRisk software (Bender and Perkins, 1982, 1987), the more recent Fortran codes produced by the U.S. Geological Survey (USGS) National Seismic-Hazard Mapping Project (hereinafter NSHMP), CRISIS (Ordaz *et al.*, 2013), EQHAZ (Assatourians and Atkinson, 2013), EQRM (Robinson *et al.*, 2005, 2006), and OpenSHA (Field *et al.*, 2003). Such a variety of codes provided an important contribution to the progress of seismic-hazard assessment and consequently to the improvement of societal resilience in many countries around the world.

In this active and diverse community of PSHA modelers and users, the Global Earthquake Model (GEM) initiative (Crowley *et al.*, 2013) is promoting the creation of open and transparent tools for seismic-hazard and risk assessment (additional information is available on the GEM website, http://www.globalquakemodel.org/; last accessed March 2014). The end goal is to provide robust, transparent, reliable, and extensible software, serving and reflecting the needs of a wide spectrum of users. The development of open-source software in this framework (Lees, 2012) is therefore a prerequisite, which allows seismic-hazard and risk practitioners to scrutinize and contribute to the methodologies/algorithms adopted for the calculations.

The fundamental motivations that inspired the creation of the OpenQuake Engine, the hazard and risk software developed by GEM (hereinafter OQ-engine), are those of reproducibility, testing, and community-based development process. Reproducibility, one of the central tenets of the scientific process, is an expanding requirement in the scientific software development. As noted by Ince *et al.* (2012, p. 485), "anything less than release of the actual source code is an indefensible approach for any scientific result that depends on computation, because not releasing such code raises needless, and needlessly confusing, roadblocks to reproducibility." The concept of

 doi: 10.1785/0220130087

reproducibility in this context is thus inextricably connected to the requirement of full and open access to source code. Furthermore, the emergence of open and reproducible scientific software can provide an important platform that allows developers of proprietary software a means of validating and testing their own code. This ensures that the consumers of such software can have increased confidence in the scientific validity of the results, whereas the owners of proprietary code can provide effective user support in a manner that is difficult for open-source projects to deliver.

This paper presents the main features of the OQ-engine hazard component, expanding in detail upon the manner in which they are developed and the cases in earthquake hazard modeling that they are designed to address. We describe the design and development of the OQ-engine, placing particular focus on the key features that make it a suitable platform for the widest range of hazard applications. As quality assurance (QA) and testing in hazard analysis, and in scientific computing in general, are becoming critical elements of any practical application (e.g., Bommer *et al.*, 2013), this paper concentrates on describing the manner in which the OQ-engine, and its development process, fulfills these important requirements.

In the initial sections, we describe how the design and development of the software was established to meet the requirements of reproducibility and QA, demonstrating the rigorous process by which the source code is tested and the scientific validity of the results ensured at every stage of the development. The focus will then move toward the objectives of scalability and global portability, and the steps taken to ensure that the OQ-engine can implement hazard models in a range of different applications and is naturally portable across all of the different tectonic environments.

OPEN PSHA SOFTWARE: SUMMARY OF CURRENT STATUS

At present, the scientific and engineering community have an open access to a limited number of computer codes to perform PSHA. During the preparatory phases of the OQ-engine development, an appraisal of available hazard software was undertaken, identifying the principal points of strength for each of them (Danciu *et al.*, 2010). The survey was updated in early 2013, focusing on recent seismic-hazard software that is freely and/or openly available in the scientific literature, such as EQHAZ (Assatourians and Atkinson, 2013), as well as updates to previously reviewed codes, including CRISIS (Ordaz *et al.*, 2013), EQRM (Robinson *et al.*, 2005, 2006), OpenSHA (Field *et al.*, 2003), and the latest version of the NSHMP Fortran software. Some of these codes are freely available on the Internet, whereas others require registration or contact with the respective development teams.

The functionalities provided by the different software differ mostly in terms of the seismogenic source typologies supported and the methodologies adopted for modeling the distribution in space and time of the earthquake ruptures. For example, CRISIS (Ordaz *et al.*, 2013), the USGS software for the NSHMP, OpenSHA (Field *et al.*, 2003), and OpenQuake implement PSHA according to what we describe herein as the classical PSHA. This refers to the PSHA methodology as articulated by Cornell (1968) and implemented by McGuire (1976), which also forms the basis for proprietary software such as FRISK88M (www.riskeng.com, last accessed October 2013). Alternatively, EQHAZ (Assatourians and Atkinson, 2013) and EQRM (Robinson *et al.*, 2006) implement event-based PSHA, which utilizes Monte Carlo sampling to generate synthetic catalogs of earthquakes for the estimation of PSHA, a process described in further detail in, for example, Musson (1999). OpenSHA and OQ-engine are both capable of supporting either approach, thus allowing for the greatest flexibility in tailoring the seismic-hazard calculations to specific contexts while ensuring consistency and compatibility between the two approaches.

A second relevant difference between the different software collected is the conceptual framework adopted for the development. The OQ-engine was designed to be modular and flexible. For example, the hazard and risk algorithms have been organized in two separate libraries, oq-hazardlib (source code accessible at https://github.com/gem/oq-hazardlib; last accessed March 2014) and oq-risklib (source code accessible at https://github.com/gem/oq-risklib; last accessed March 2014), whereas a third library is responsible for the definition of the data model (oq-nrmllib; https://github.com/gem/oq-nrmllib, last accessed March 2014). This modular structure favors easier software development/maintenance from an Information Technology (IT) perspective but, more importantly, allows access to the core hazard and risk algorithms and to exercise and investigate them without the need to interact with other parts of the software.

This design decouples the internal components of the hazard calculations through well-defined interfaces, promotes extensibility, efficient testability, and scientific operability.

OQ-ENGINE DEVELOPMENT: HISTORY, PROCESS, AND QUALITY ASSURANCE

The development of the hazard component of the OQ-engine started in 2010. The initial version was based on OpenSHA (Field *et al.*, 2003), an object-oriented, Java-based, state-of-the-art library for the hazard calculation currently used, for example, in the development of the Uniform California Earthquake Rupture Forecast (UCERF) models (Field *et al.*, 2009).

The core of the OQ-engine was developed in Python. For an open-source scientific project, Python has many advantages because it is released with an open-source license and has an extensive set of scientific and numerical libraries that make it an attractive environment for interactive development between scientists and IT developers. From the beginning of 2011, the OQ-engine incorporated the initially envisaged OpenSHA calculation workflows directly into the software. The challenges associated with increasing the extensibility and maintenance of the OQ-engine, however, hindered the direct linkage with the Java-based library. The latter was the catalyst for the development of a new Python-based seismic-hazard library, ensuring easier integration with the overall OQ-engine software architecture.

Given the preceding experience and its renowned reputation as state-of-the-art tool in seismic-hazard assessment, it is logical that the design of this new library was guided by the calculation structure contained within OpenSHA (Field *et al.*, 2003). Its conceptual design clearly played a role in the design of the new OQ-engine hazard library, most notably in the philosophy of having an object-oriented framework that deconstructs the hazard calculation into its component set of connectable pieces, which can then be assembled into applications of substantial complexity. Furthermore, well-tested algorithms, such as the one for modeling the movement of ruptures across regular (or simple) fault surfaces and the probabilistic approach for the calculation of hazard curves, have been implemented in the new library.

Code Review and Unit Testing

The OQ-engine development process conforms to current software development practices, such as code review, unit testing, and extensive code documentation (i.e., comment lines within the code). Unit testing refers to the practice of creating tests for each unit of the source code (for instance, functions or classes) that examine its behavior for correctness (e.g., given certain input data the function must return certain expected results) and robustness (e.g., given certain invalid input data the function must return a certain error message). Although inevitably incurring a significant initial time investment, unit testing provides the basis for any rigorous QA. Unit testing creates a safer development process, which, as much as possible, avoids regressions and guarantees reproducibility.

The OQ-engine code is hosted and actively developed using a public web-based repository (see http://www.github.com/gem; last accessed March 2014), which facilitates a transparent development process. Every line of code contributed by a developer must undergo a review process by other members of the development team. The review will include careful scrutiny of the operation, efficiency, and completeness of documentation of the code itself, while also ensuring that changes to the code do not introduce errors or erroneous values when implementing realistic test calculations. Submissions that do not satisfy these conditions are rejected. Such an approach not only helps improve the source code quality and reliability, but also allows software knowledge to be distributed over the development team and does not rest solely with a particular member.

Quality Assurance Tests

Although the process of unit testing should ensure correctness in the behavior of the component modules within the software, they do not necessarily ensure that whole calculators themselves are correct in the output that is produced. To this end, the OQ-engine includes a suite of QA tests that are meant to verify the correctness of the different calculators end-to-end: starting from a set of input files, the values as provided by the output files generated by a calculator are checked against expected results. Current QA tests have been defined using problems for which analytical solutions can be found or solutions that can be computed by hand or with an alternative algorithm. The OQ-engine currently includes four tests from the PSHA software test suite developed by the Pacific Earthquake Engineering Research Center (PEER) tests (Thomas *et al.*, 2010) as in-built QA tests: two for simple fault and two for area source. Figure 1 shows an example of the comparison between reference and results computed using the OQ-engine.

These rigorous testing procedures proved particularly valuable during a Senior Seismic Hazard Analysis Committee) (SSHAC) level 3 PSHA project (Bommer *et al.*, 2013), in which the OQ-engine has been utilized for the calculation of hazard curves for a number of logic-tree paths that then have been compared against the results as computed by FRISK88M. The tests revealed a satisfactory agreement, which, considering the complexity of the input model and the differences in the algorithms used by the two softwares, confirms the reliability of the results provided by the OQ-engine.

MAIN FEATURES OF THE OQ-ENGINE

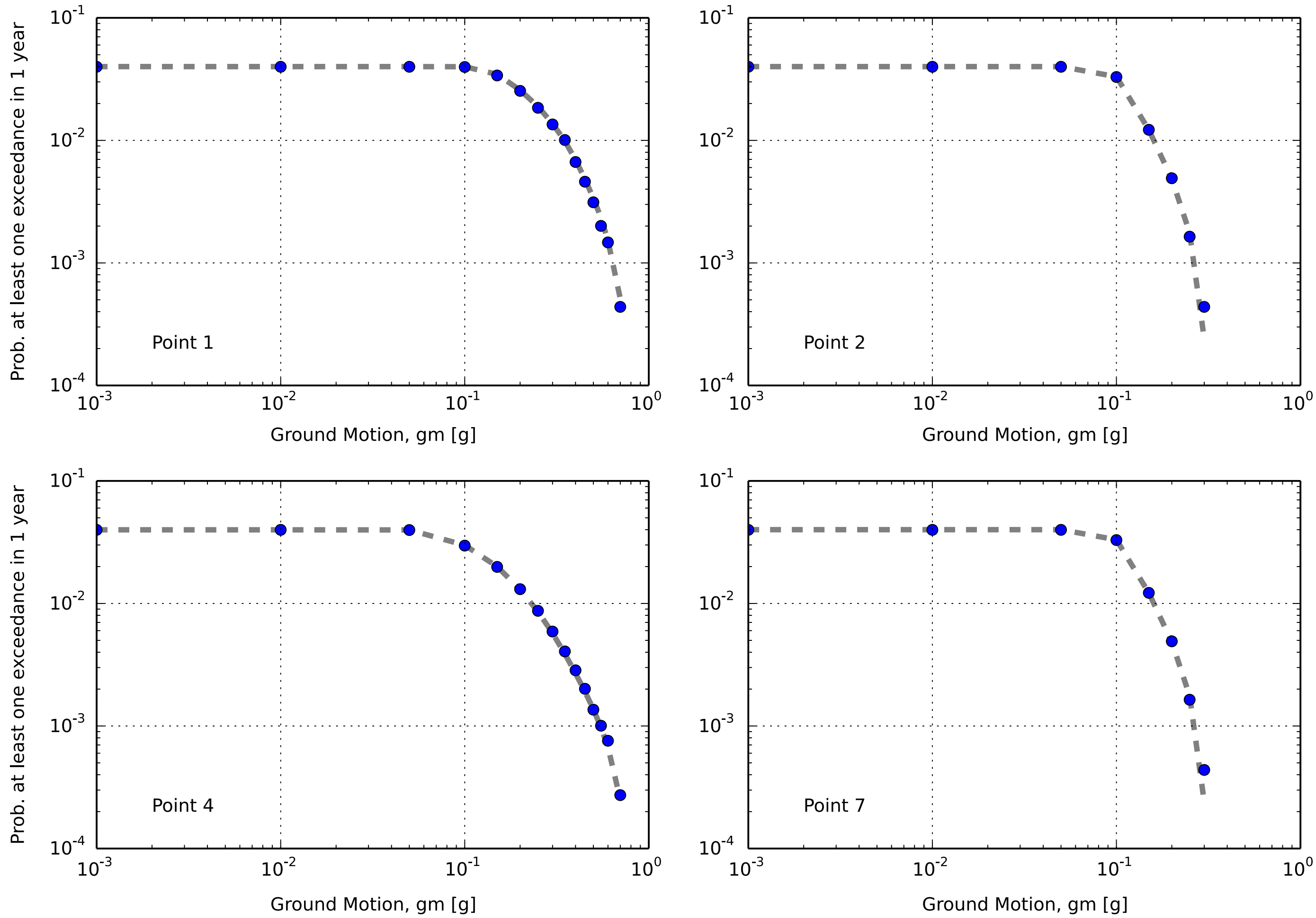
A Data Model for the Representation of Input and Output Information: The Natural Hazard Risk Markup Language (NRML)

One major objective of the hazard component of GEM is the creation of a community-contributed, harmonized collection of seismic-hazard models, which collectively cover the globe. The construction of a harmonized collection is possible, however, only if the PSHA input models can be represented using a standard schema[1] (Pagani *et al.*, 2011).

The current lack of homogeneity in the description of seismogenic zones and their associated uncertainties can severely limit the portability of a seismic-hazard input model to calculation software other than the one for which it was originally implemented. As a consequence, results are often tied to particular software. This poses the question as to what extent the results depend on the hazard model rather than on the software used for the calculation and, more importantly, it puts limits on reproducibility (i.e., results can be obtained if and only if a particular software is used).

To allow for a clear separation between data and algorithms, and to promote an accurate definition of the information describing input and output information of hazard and risk models, a customized XML schema called Natural hazard Risk Markup Language (NRML) has been introduced. This schema builds on top of the eXtensible Markup Language (XML), an open standard developed by an international community, which is widely used as a data exchange format. Examples of standards based on XML are the geographic markup language developed by the Open Geospatial Consortium (OGC) and, in the seismological community, QuakeML (Schorlemmer *et al.*, 2011).

[1]The term "schema" refers to the formal language for describing components of hazard and risk models, including definition of the types of data, the corresponding structure, the rules governing the validity of the data, and the nature of the interconnection between the different elements of the hazard and risk models.



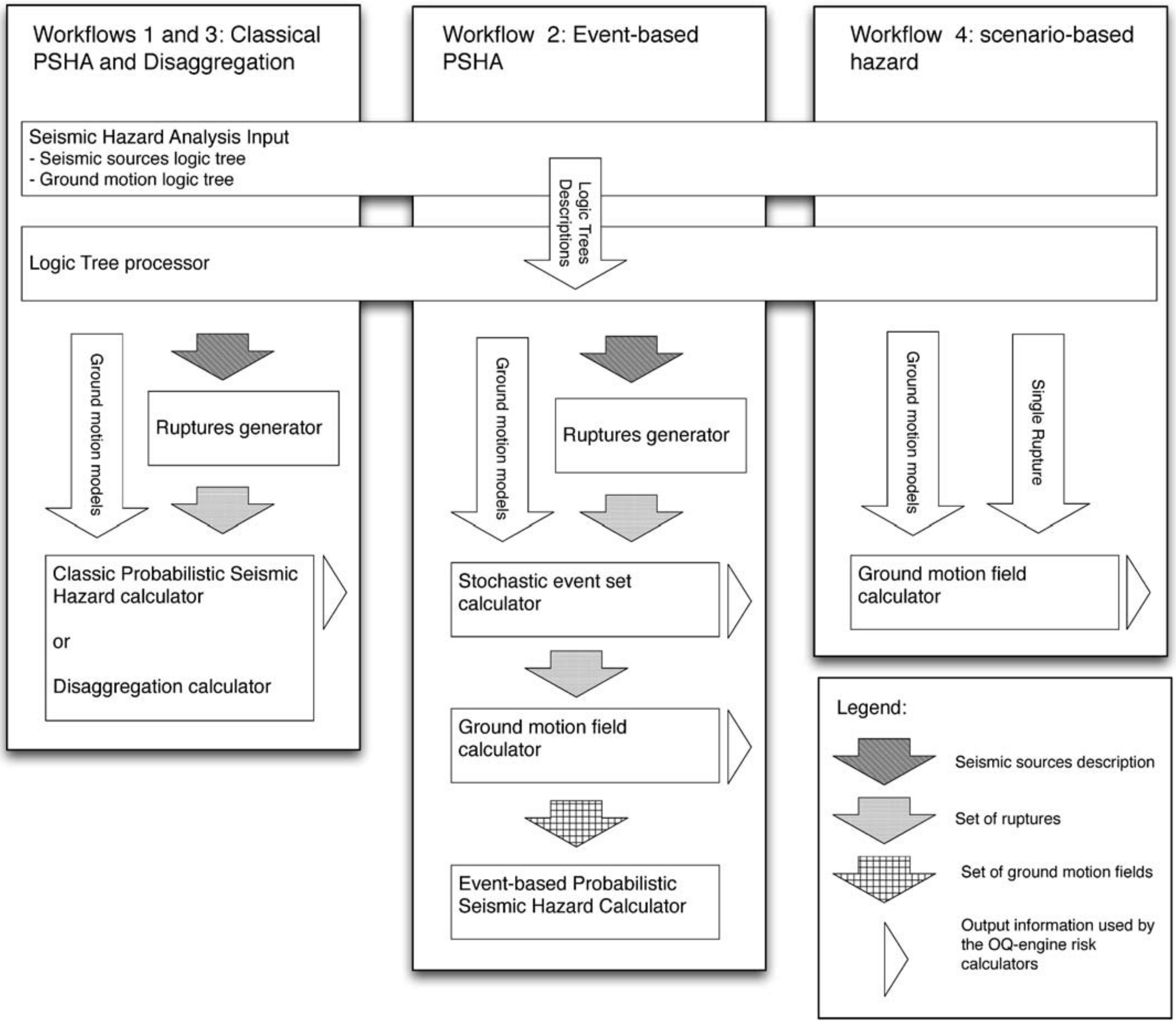
▲ **Figure 1.** Quality assurance example in the OQ-engine. Comparison between the results provided by Thomas *et al.* (2010) set 1 test 5 (dashed lines) and the ones computed with the OQ-engine (dots). See also figures between 3.34 and 3.40 in Thomas *et al.* (2010).

The current NRML schema—the description of which is included in the oq-nrmllib library (available at https://github.com/gem/oq-nrmllib, last accessed on October 2013)—supports the description of the seismic source model and ground-motion model, as well as of the logic-tree structure used to describe their respective epistemic uncertainties. All the outputs produced by the OQ-engine can also be represented using objects described in the NRML schema. The details of the NRML data schema definition are described in the OQ-engine documentation (available at http://www.globalquakemodel.org/openquake/support/documentation/; last accessed March 2014). We wish to emphasize that having a community-driven format for describing seismic-hazard models is a necessary step to make seismic-hazard assessment a more reproducible process. A set of demonstrative PSHA input models for the OQ-engine are available on the GEM web repository (https://github.com/gem/oq-engine/tree/master/demos; last accessed October 2013)

A Single Software for Multiple-Use Cases

A key motivation for the development of the OQ-engine is to offer a single software that is able to serve multiple-use cases. OQ-engine's envisaged users include research scientists interested in performing seismic-hazard assessment on national, regional, or global scale, analysts working in the insurance/reinsurance industries focusing on loss estimates, public institutions performing urban scale loss analysis, and engineering companies working on site-specific hazard analyses. In order to support such a wide spectrum of users, OQ-engine has to offer multiple calculation workflows responding to different needs, various seismic source modeling strategies, and different capabilities to scale from single-site to global analysis. To satisfy the requests of different use cases, the hazard module of OQ-engine currently provides the following calculators, the workflows for which are illustrated in Figure 2:

- *Classical PSHA Calculator*: calculates hazard curves, hazard maps, and uniform hazard spectra by solving the PSHA integration procedure, as proposed by Field *et al.* (2003). This is the usual approach adopted in regional/national-scale hazard assessment, as well as in site-specific studies. Using the risk component of the OQ-engine, the computed hazard curves can be combined with a vulnerability and exposure model to derive asset-specific loss exceedance



▲ **Figure 2.** Illustration of the main hazard calculation workflows supported by the OQ-engine. From left to right, main calculators and data flow used in classical PSHA and disaggregation, event-based PSHA, and scenario-based hazard.

curves and loss maps for various return periods. Such analyses are useful for comparative risk assessment between assets at different locations, or to understand the areas where mitigation actions should be concentrated. Crowley and Bommer (2006) suggest this methodology tends to overestimate losses at high return periods for portfolios of structures and recommend the use of methods capable to account for the spatial correlation of ground motion residuals.

- *Event-Based PSHA Calculator*: computes stochastic event sets (i.e., synthetic catalogs of earthquake ruptures) and ground-motion fields for each rupture, possibly taking into account the spatial correlation of within-event residuals. This is essentially a Monte Carlo–based PSHA calculator (Musson, 1999). The computed synthetic catalogs can be used for comparisons against a real catalog, whereas hazard curves and hazard maps can be derived from post-processing the ground-motion fields (Ebel and Kafka, 1999). Ground-motion fields are essential input for loss estimations, whereby loss exceedance curves and loss maps are calculated for a collection of assets by combining a vulnerability and exposure model with these sets of ground-motion fields. Because the spatial correlation of the ground-motion residuals can be taken into account in this calculator, the losses to each asset can be summed per ground-motion field, and a total loss exceedance curve representative of the whole collection of assets can be derived. These results are important for deriving reliable estimates of the variance of the total losses.

- *Disaggregation Calculator*: given a PSHA model, it computes the earthquake scenarios contributing the most to a given hazard level at a specific site (Bazzurro and Cornell, 1999). Currently this is done following the classical PSHA methodology; this functionality will be added to the event-based calculator in subsequent development phases.
- *Scenario Calculator*: given an earthquake rupture and a ground-shaking model, a set of ground-motion fields can be computed. This is a typical use case for urban-scale loss analysis. This set of ground-motion fields can be employed with a fragility/vulnerability model to calculate distribution of damage/losses for a collection of assets. Such results are of importance for emergency management planning and for raising societal awareness of risk.

Currently, all the OQ-engine calculators generate ruptures with probabilities defined according to the specified temporal occurrence model. For the widest possible global application, the Poisson model is currently supported. Plans for introducing other temporal occurrence models, such as the lognormal or Brownian passage time models, are already under evaluation. The support for a generalized PSHA calculator that does not rely on a specific temporal occurrence model would offer the possibility to potentially deal with any earthquake rupture forecast model.

For further information regarding the risk calculators, the reader can refer to Silva *et al.* (2013) and the OQ-engine scientific documentation available online (http://www.globalquakemodel.org/openquake/support/documentation/, last accessed October 2013).

Flexibility in Modeling Seismic Sources

The OQ-engine was designed to implement PSHA models that follow different strategies or modeling approaches. Regions characterized by low-to-moderate seismicity (e.g., Northern Europe, Canada, and Australia) usually develop models that focus mostly on distributed seismicity sources. Typically, these are represented in the form of regions with uniform activity parameters (often called area sources or source zones) or gridded seismicity in which earthquake occurrence parameters are specified over a regular grid of locations (e.g., eastern United States; Petersen *et al.*, 2008). In regions where active fault data are available (e.g., western United States, Japan, New Zealand), fault sources may introduce further complexities when attempting to model efficiently the propagation of each rupture across the fault surface. Zones close to subduction sources (e.g., United States Pacific Northwest, South America, Indonesia, Japan) require more complex modeling of large subduction interface events.

To accommodate the modeling needs in various tectonic contexts, OQ-engine supports five seismogenic source typologies: point source, area source, simple fault source, complex fault source, and characteristic fault source. The point source defines a geographical location on the Earth's surface, below which earthquake ruptures are generated for all magnitude values admitted by a magnitude–frequency distribution (hereinafter MFD). Sizes and shapes of synthetic ruptures are determined from a magnitude–area scaling relationship and an aspect ratio value (fault length/width) and are constrained within the seismogenic layer (defined in terms of upper and lower seismogenic depths). Ruptures can be distributed according to multiple nodal planes and multiple hypocentral depths. Figure 3 shows some examples of the possible ruptures that can be created by a point source with different properties, such as MFD, rupture plane distribution, and hypocentral depth distribution. Collections of point sources can be used to implement gridded seismicity models in which occurrence parameters vary from point to point.

An area source is defined by the boundary of a geographical region and is modeled by uniformly distributing the original occurrence rates among a collection of point sources that are equally distributed on a regular grid within the polygon area. In the current implementation, the boundaries of the area sources are porous, therefore allowing ruptures generated from points inside the source to extend beyond the limits of the zone.

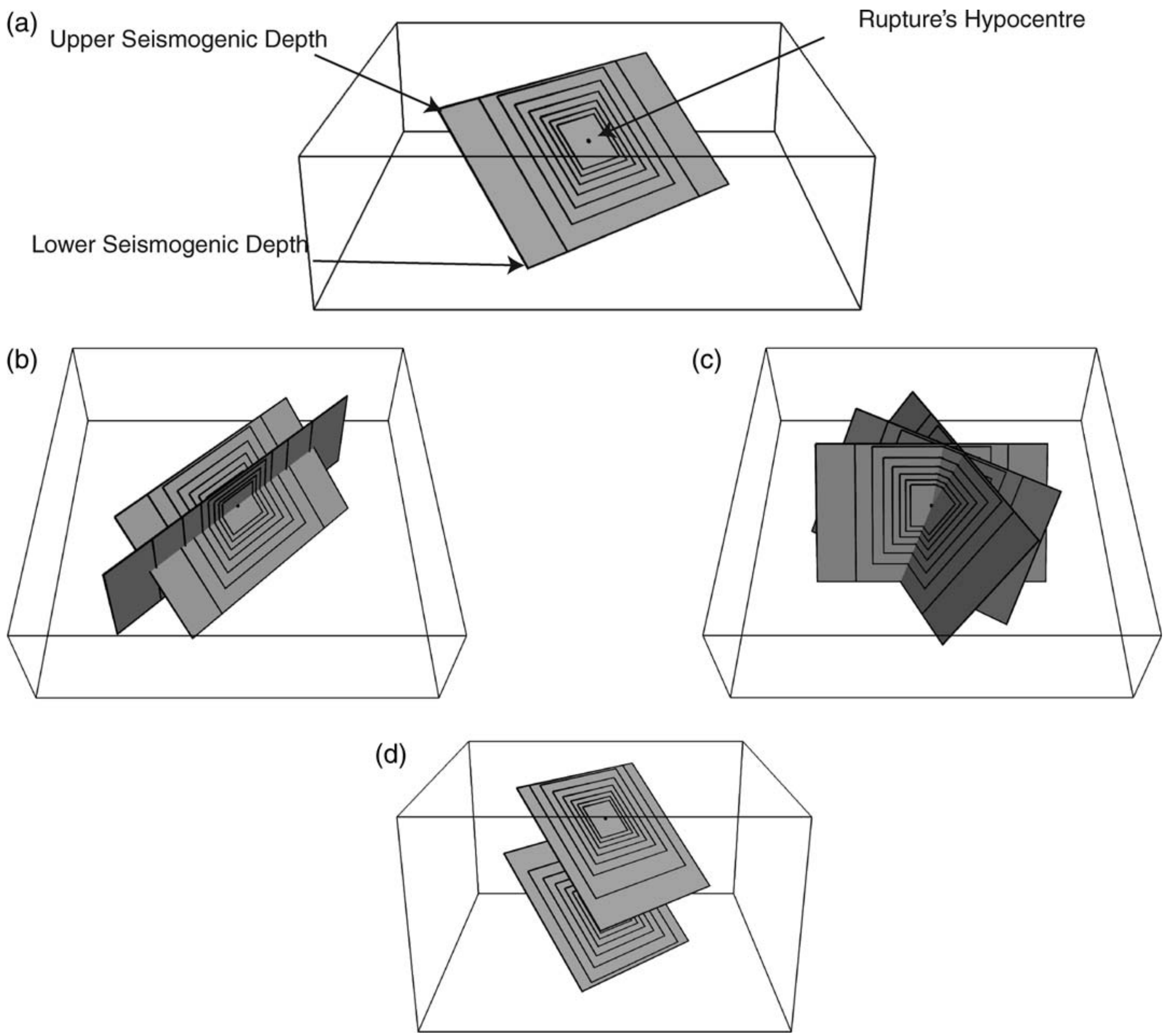
The simple and complex fault sources (see Fig. 4) are intended to model seismicity occurring on active faults with different levels of complexity in the fault geometry. The former is defined in terms of a fault trace, dip angle, and upper and lower seismogenic depths, whereas the latter is simply defined by a sequence of fault edges and therefore allows for the modeling of irregular fault planes (for instance with varying dip and width). Both simple and complex fault sources generate ruptures for each magnitude value in an MFD according to a magnitude–area scaling relationship and an aspect ratio and place them on all possible locations on the fault surface. This process is usually indicated with the term "floating the rupture".

Finally, the characteristic fault source typology is meant to represent seismicity occurring on a generic fault surface with seismic events rupturing the entire fault surface independently of their magnitude values. In other words, it can be used to model individual faults or fault segments that produce earthquakes of almost the same size (Schwartz and Coppersmith, 1984).

These source typologies have been tested in the OQ-engine with preliminary PSHA input models for a number of countries or regions around the world, most notably for the PSHA input models for conterminous United States and Japan (2012 version), produced by the USGS-NSHMP (Petersen *et al.*, 2008) and the National Research Institute for Earth Science and Disaster Prevention (Fujiwara *et al.*, 2009), respectively.

Modeling Distributed Seismicity by Means of Finite Ruptures

The use of distributed seismicity sources in modern PSHA calculations, when used in conjunction with most modern GMPEs, requires comprehensive definitions of the finite-rupture surface for each earthquake, which presents a considerable challenge to hazard modelers. A critical point of scientific divergence for many different seismic-hazard codes is in the rendering of physically consistent virtual rupture planes within a distributed seismicity source. Some codes utilize empirical conversions between different source-to-site distance metrics,



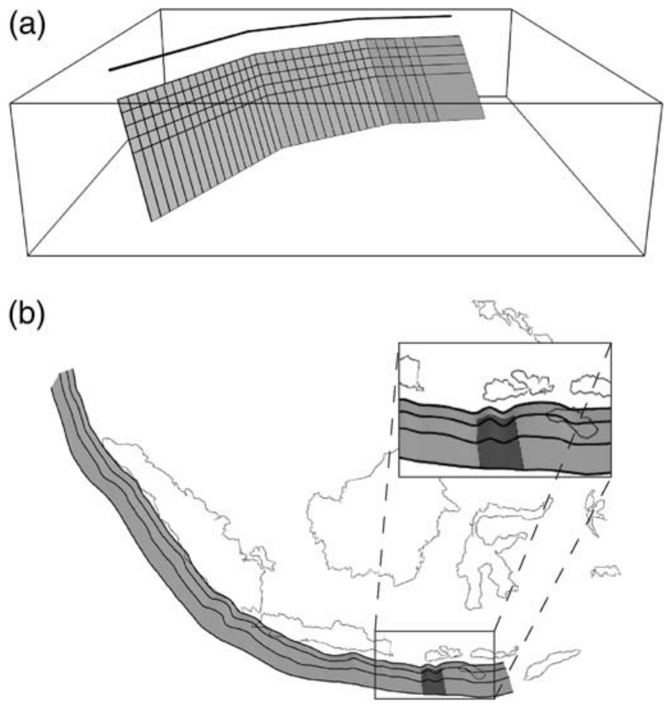
▲ **Figure 3.** Graphical representation of the earthquake ruptures as generated by a point source. (a) Given a geographical location on the Earth's surface, ruptures are generated underneath according to a scaling relationship and aspect ratio value and forced to not exceed the upper and lower seismogenic depths. Ruptures can be distributed over multiple (b) dips, (c) strikes, and (d) hypocentral depths.

thus allowing point source distances to be converted to finite-rupture distances. The inconsistencies of this approach are well described by Bommer and Akkar (2012). Furthermore, such conversions still provide little information to constrain other finite-rupture properties, such as depth to the top of rupture or definition of the hanging wall, in a manner that is physically consistent with the geometry of a rupture plane. Other seismic-hazard codes instead prefer to generate synthetic finite-rupture planes, preferably with physical properties (such as strike, dip, aspect ratio) defined in a manner that is consistent with the seismotectonics of the zone (Fig. 3). This approach is preferred in OQ-engine; therefore, distributed seismicity sources require additional seismotectonic information to aid in the characterization of the synthetic planes. It must also be recognized that the adoption of synthetic planes also incurs a considerable computational cost.

The generation of finite ruptures from a seismogenic source requires each source to be associated with a magnitude scaling relation, in order to determine the physical dimensions of the rupture according to the magnitude.

GMPE Implementation: A Flexible, Test-Driven Approach

The incorporation of new ground-motion prediction equations (GMPEs) is one of the major challenges for long-term maintenance of PSHA software because it will often be necessary to introduce GMPEs that are not supported natively by the



▲ **Figure 4.** (a) Example of a simple fault-source geometry. The fault surface is obtained by translating the fault trace from the upper seismogenic depth to the lower seismogenic depth in a direction perpendicular to the average fault strike and with an inclination equal to the dip angle. For a given magnitude value, a rupture surface is created according to a scaling relationship and aspect ratio values. The rupture surface is then placed uniformly over the fault surface to simulate all possible locations. (b) Example of complex fault source geometry modeling of the the Sumatra–Java subduction interface fault as obtained from the Slab 1.0 subduction fault model (Hayes *et al.*, 2012). The inset shows the detail of a modeled rupture.

software. In the past, a flexible way to easily incorporate new GMPEs was to define tables specifying the median ground motion and standard deviation values as a function of magnitude and distance (e.g., Bender and Perkins, 1987). However, in some cases the complexity of current GMPEs hinders the use of such tables and moves the focus from portability to the need for correctly reproducing the published ground-motion models.

The implementation of GMPEs within the OQ-engine demonstrates the power and flexibility of object-oriented design in seismic-hazard calculations. Each new GMPE can be implemented by extending a base class (or template), which ensures that the GMPE specifies the common information required for use within a hazard calculation, and the principal methods that it must contain (i.e., a method to compute mean and standard deviation once the independent—or predictor—variables are completely specified or a method to compute the probability of exceedance of a certain value of a specific intensity measure type). The definition of predictor variables also follows a predefined schema that incorporates three main groups: site, rupture, and distance parameters. The currently supported site parameters are V_{S30}, V_{S30}-type (measured or inferred), depth to 1.0 km/s V_S horizon, and depth to 2.5 km/s V_S horizon. Rupture parameters are magnitude (moment), dip angle, rake angle, top of rupture depth, hypocentral depth, and rupture width. Supported distances are closest distance to rupture (R_{rup}), Joyner–Boore distance (R_{JB}, the closest distance from the site to the surface projection of the rupture), R_X distance (the horizontal distance from the site to the top edge of the rupture, measured perpendicular to strike), hypocentral distance, and epicentral distance. The effectiveness and flexibility of this approach have been already demonstrated with the implementation of a number of the most recent GMPEs used for hazard analysis in different tectonic regimes (Abrahamson and Shedlock, 1997). An up-to-date list of available GMPEs can be found in the online documentation (http://docs.openquake.org/oq-hazardlib/gsim/index.html; last accessed March 2014). New GMPEs are continually being added to the software, but already the set contains many widely used models for active shallow tectonic regions (including Cauzzi and Faccioli, 2008; Akkar and Bommer, 2010; and the Next Generation Attenuation [NGA-West] models), subduction regions (including Youngs *et al.*, 1997; Atkinson and Boore, 2003; Zhao *et al.*, 2006), and stable continental regions (including Toro *et al.*, 1997; Campbell, 2003; Atkinson and Boore, 2006).

In the OQ-engine, a key requirement in the GMPE implementation process is the definition of test tables and their use for QA testing. A test table contains median or standard deviation values computed for many combinations of predictor variable values. Ideally, a suite of test tables should exhaust all possible combinations of the predictor variables admitted by the functional form of a GMPE. In the OQ-engine, once test tables follow a predefined format (i.e., a comma-separated values [CSV] file), an automatic testing framework facilitates checking the correct implementation/coding of a GMPE. In the implementation of GMPEs, the ideal case is to use test tables produced from codes originally provided by the GMPE authors themselves. If this is not possible, test tables are generated from alternative software implementation. The suite of test tables currently implemented in the OQ-engine is openly available on the OQ-engine web repository.

The rigorous implementation and testing procedure for the implementation of GMPEs described is one of several factors that complicate the use of user-defined tables to characterize the GMPE. The adoption of user-defined tables could weaken the objectives of the development and usage of OQ-engine from a QA perspective. As GMPEs continue to adopt more complex functions to model the earthquake physics, the process of representing the models using tables of common predictor variables becomes more complex and prone to inconsistencies with the characterization of the source and site models, and it leaves room for the user to introduce errors into the PSHA calculation process. Furthermore, the use of GMPE tables requires persistent interpolation of the ground-motion values for each rupture, compromising the efficiency of the hazard calculation.

In future versions of the OQ-engine, nonetheless, we will explore the possibility of allowing the user to define GMPEs through tables, assuming we will be able to overcome the limitations just explained. The rigorous implementation and testing process previously described in this paper remains, at the moment, the option that is most likely to maintain both the quality and efficiency of the GMPE implementation inside the hazard calculation.

A Logic-Tree-Driven Approach

It is a critical requirement of any seismic-hazard assessment that the hazard model incorporates the characterization of epistemic uncertainties (by considering alternative source models, activity parameters, ground-motion prediction equations) by means of a logic-tree structure. A logic tree is therefore an integral part of a PSHA model. The OQ-engine offers a framework, based on the NRML format, to programmatically define logic trees describing epistemic uncertainties in the source and ground-motion models. To operate on such information, OQ-engine contains a logic-tree processor (LTP) that is responsible for creating the overall logic-tree structure, harvesting the tree, creating automatically the input data corresponding to a tree path and executing the associated hazard analysis calculation. More precisely, the LTP can work in two distinct modes: enumeration or Monte Carlo sampling. The former allows for the direct execution of seismic-hazard calculations for all the paths in a logic tree, and this is clearly a viable option when the tree size is relatively small, whereas the latter option is to sample randomly, according to their weights, a number of tree paths for which the hazard analysis is then executed. This is the approach that can be used to process very large tree structures for which it is not feasible to perform calculations for all possible branch combinations. In this regard, the novelty brought by the OQ-engine is the capability to formally and programmatically define a logic tree (basically through the definition of an input file following the NRML format), without the need of having external tools that prepare the input data for the different logic-tree paths and that aggregate the different results. In other words, the processing of a logic tree becomes an integral part of a PSHA calculation with OQ-engine, and again this goes in the direction of having PSHA calculations as automated as possible and therefore more reproducible.

Scalability: Extending PSHA to the Most Complex Models

Current state-of-the-art PSHA models can pose significant challenges from a calculation viewpoint. Alternative modeling approaches, epistemic uncertainties in the source model (geometry, occurrence parameters) and GMPEs, aleatory uncertainties in rupture modelling (variability in rupture orientation, hypocentral depth, faulting style) can create potentially very large earthquake rupture forecasts requiring significant computation times. Regional-scale models may also require calculations over large collections of sites. To allow for the implementation of complex models, and to make more efficient use of now common multicore machines and computational cluster environments, OQ-engine natively supports parallelized calculations.

The parallelization scheme currently adopted executes in parallel the calculation for each seismic source in the source model and then aggregates the results. Moreover, in case of analysis involving multiple logic-tree paths, the associated computations can be also run in parallel. Designing and tuning the parallelization scheme to get optimal performances is not a trivial task, and we recognize that the future development of the OQ-engine will require further efforts toward this direction.

CONCLUSIONS AND FUTURE DIRECTIONS

We have described the development process adopted and the main features of the OQ-engine, the hazard and risk calculation engine developed by the GEM initiative. The OQ-engine software is completely accessible and downloadable through an online repository (see http://www.github.com/gem; last accessed March 2014), in agreement with open-source software practice. The OQ-engine is developed by taking full advantage of unit-testing procedures, that is, each unit of code has a corresponding set of routines that check its correct behavior, and QA tests.

Although the current OQ-engine already represents a comprehensive software suite for seismic-hazard and risk analyses, it provides an opportunity for many new feature developments in the future. Anticipated features in future versions of the engine will include the support of time-dependent earthquake recurrence models and more sophisticated methodologies for handling local site effects than the use of simple parameters (such as V_{S30}, $Z_{1.0}$, etc.) embedded in current GMPEs.

ACKNOWLEDGMENTS

The authors acknowledge the contribution of a large number of developers, who over the last three years played a part in the development of the current and previous versions of the OpenQuake Engine. A particular thanks to A. Gritsay, who worked on the development of the oq-hazardlib. Scientists working on a number of GEM regional programs, and in particular the European Union-funded Seismic Hazard HARmonisation in Europe (SHARE) program, and scientists who voluntarily worked with beta versions of the OpenQuake Engine provided useful feedback and comments that helped in the improvement and debugging of code. The genesis of the OpenQuake software benefitted from a collaboration with the creators and developers of OpenSHA. The authors therefore wish to acknowledge the contribution of the U.S. Geological Survey (USGS), and in particular Ned Field, Peter Powers, and Kevin Milner, in the development of the OpenQuake Engine. The authors would like to express their gratitude to Gail Atkinson, an anonymous reviewer and Associate Editor Zhigang Peng for their valuable comments and suggestions, which helped to improve the quality of the paper. Comments and recommendations from Julian Bommer (GEM OpenQuake Seismic-Hazard Advisory Panel Chair) considerably improved the quality of the manuscript. The authors are grateful for his contribution.

REFERENCES

Abrahamson, N. A., and K. M. Shedlock (1997). Overview, *Seismol. Res. Lett.* **68,** no. 1, 9–23.

Akkar, S., and J. J. Bommer (2010). Empirical equations for the prediction of PGA, PGV, and spectral accelerations in Europe, the Mediterranean region, and the Middle East, *Seismol. Res. Lett.* **81,** no. 2, 195–206, doi: 10.1785/gssrl.81.2.195.

Assatourians, K., and G. M. Atkinson (2013). EqHaz: An open-source probabilistic seismic-hazard code based on the Monte Carlo simulation approach, *Seismol. Res. Lett.* **84,** no. 3, 516–524, doi: 10.1785/0220120102.

Atkinson, G. M., and D. M. Boore (2003). Empirical ground-motion relations for subduction-zone earthquakes and their application to Cascadia and other regions, *Bull. Seismol. Soc. Am.* **93,** no. 4, 1703–1729.

Atkinson, G. M., and D. M. Boore (2006). Earthquake ground-motion prediction equations for eastern North America, *Bull. Seismol. Soc. Am.* **96,** no. 6, 2181–2205, doi: 10.1785/0120050245.

Bazzurro, P., and C. A. Cornell (1999). Disaggregation of seismic hazard, *Bull. Seismol. Soc. Am.* **89,** no. 2, 501–520.

Bender, B., and D. M. Perkins (1982). SEISRISK II: A computer program for seismic hazard estimation, *U.S. Geol. Surv. Open-File Rept. 82-293*.

Bender, B., and D. M. Perkins (1987). SEISRISK III: A computer program for seismic hazard estimation, *U.S. Geol. Surv. Bull. 1772*.

Bommer, J. J., and S. Akkar (2012). Consistent source-to-site distance metrics in ground-motion prediction equations and seismic source models for PSHA, *Earthq. Spectra* **28,** no. 1, 1–15.

Bommer, J. J., F. O. Strasser, M. Pagani, and D. Monelli (2013). Quality assurance for logic-tree implementation in probabilistic seismic hazard analysis for nuclear applications: A practical example, *Seismol. Res. Lett.* **84,** 938–945.

Campbell, K. W. (2003). Prediction of strong ground motion using the hybrid empirical method and its use in the development of ground-motion (attenuation) relations in eastern North America, *Bull. Seismol. Soc. Am.* **93,** no. 3, 1012–1033.

Cauzzi, C., and E. Faccioli (2008). Broadband (0.05 s to 20 s) prediction of displacement response spectra based on worldwide digital records, *J. Seismol.* **12,** no. 4, 453–475.

Cornell, C. A. (1968). Engineering seismic risk analysis, *Bull. Seismol. Soc. Am.* **58,** no. 5, 1583–1606.

Crowley, H., and J. J. Bommer (2006). Modelling seismic hazard in earthquake loss models with spatially distributed exposure, *Bull. Earthq. Eng.* **4,** 249–273, doi: 10.1007/s10518-006-9009-y.

Crowley, H., R. Pinho, M. Pagani, and N. Keller (2013). Assessing global earthquake risks: The Global Earthquake Model (GEM) initiative, in *Handbook of Seismic Risk Analysis and Management of Civil Infrastructure Systems*, S. Tesfamariam and K. Goda (Editors), Woodhead Publishing Limited, 920 pp.

Danciu, L., M. Pagani, D. Monelli, and S. Wiemer (2010). GEM1 Hazard: Overview of PSHA software. *GEM Technical Report 2010-2*, GEM Foundation, Pavia, Italy (available at http://www.globalquakemodel.org/resources/publications/technical-reports/gem1-executive-summary/; last accessed March 2014).

Ebel, J. E., and A. L. Kafka (1999). A Monte Carlo approach to seismic hazard analysis, *Bull. Seismol. Soc. Am.* **89,** no. 4. 854–866.

Field, E. H., T. E. Dawson, K. R. Felzer, A. D. Frankel, V. Gupta, T. H. Jordan, T. Parsons, M. D. Petersen, R. S. Stein, R. J. Weldon II, and C. J. Wills (2009). Uniform California Earthquake Rupture Forecast, Version 2 (UCERF 2), *Bull. Seismol. Soc. Am.* **99,** no. 4, 2053–2107, doi: 10.1785/0120080049.

Field, N., T. A. Jordan, and C. A. Cornell (2003). OpenSHA: A developing community-modeling environment for seismic hazard analysis, *Seismol. Res. Lett.* **74,** no. 4, 406–419.

Fujiwara, H., S. Kawai, S. Aoi, N. Morikawa, S. Senna, N. Kudo, M. Ooi, K. X. Hao, K. Wakamatsu, Y. Ishikawa, T. Okumura, T. Ishii, S. Matsushima, Y. Hayakawa, N. Toyama, and A. Narita (2009). *Technical Reports on National Seismic Hazard Maps for Japan*, Technical Note of the National Research Institute for Earth Science and Disaster Prevention n. 336.

Hayes, G. P., D. J. Wald, and R. L. Johnson (2012). Slab1.0: A three-dimensional model of global subduction zone geometries, *J. Geophys. Res.* **117,** no. B01302, doi: 10.1029/2011JB008524.

Ince, D. C., L. Hutton, and J. Graham-Cumming (2012). The case for open computer program, *Nature* **482,** 485–488.

Lees, J. M. (2012). Open and free: Software and scientific reproducibility, *Seismol. Res. Lett.* **83,** no. 5, 751–752.

Lin, T., S. C. Harmsen, J. W. Baker, and N. Luco (2013). Conditional spectrum computation incorporating multiple causal earthquakes and ground-motion prediction models, *Bull. Seismol. Soc. Am.* **103,** 1103–1116, doi: 10.1785/0120110293.

McGuire, R. (1976). Fortran program for seismic risk analysis, *U.S. Geol. Surv. Open-File Rept. 76-67*.

McGuire, R. (1978). FRISK: Computer program for seismic risk analysis using faults as earthquake sources, *U.S. Geol. Surv. Open-File Rept. 78-1007*.

McGuire, R. (2004). *Seismic Hazard and Risk Analysis*, Earthquake Engineering Research Institute, Oakland, California.

McGuire, R. (2008). Probabilistic seismic hazard analysis: Early history, *Earthq. Eng. Struct. Dyn.* **37,** 329–338.

Musson, R. M. W. (1999). Probabilistic seismic hazard maps for the Balkan region, *Annali di Geofisc.* **42,** no. 6, 1109–1124.

Musson, R. M. W. (2012). PSHA validated by quasi observational means, *Seismol. Res. Lett.* **83,** no. 1, 130–134, doi: 10.1785/gssrl.83.1.130.

Ordaz, M., F. Martinelli, V. D'Amico, and C. Meletti (2013). CRISIS2008: A flexible tool to perform probabilistic seismic hazard assessment, *Seismol. Res. Lett.* **84,** no. 3, 495–504, doi: 10.1785/0220120067.

Pagani, M., and A. Marcellini (2007). Seismic-hazard disaggregation: A fully probabilistic methodology, *Bull. Seismol. Soc. Am.* **97,** no. 5, 1688–1701, doi: 10.1785/0120060126.

Pagani, M., L. Danciu, D. Monelli, J. Woessner, S. Wiemer, and E. H. Field (2011). Toward a unified PSHA input data model and code: A proposal based on the GEM1 experience, in *Proc. 14th European Conference on Earthquake Engineering*, Ohrid, Macedonia, Paper SS5-2, CD-ROM 1.

Park, J., P. Bazzurro, and J. W. Baker (2007). Modeling spatial correlation of ground motion intensity measures for regional seismic hazard and portfolio loss estimation, in *Applications of Statistics and Probability in Civil Engineering*, J. Kanda, T. Takada, and H. Furuta (Editors), Taylor & Francis Group, London, United Kingdom, 602 pp.

Petersen, M. D., A. D. Frankel, S. C. Harmsen, C. S. Mueller, K. M. Haller, R. L. Wheeler, R. L. Wesson, Y. Zeng, O. S. Boyd, 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, *U.S. Geol. Surv. Open-File Rept. 2008-1128 version 1.1*, 128 pp.

Robinson, D., T. Dhu, and J. Schneider (2006). Practical probabilistic seismic risk analysis: A demonstration of capability, *Seismol. Res. Lett.* **77,** no. 4, 453–459.

Robinson, D., G. Fulford, and T. Dhu (2005). EQRM: Geoscience Australia's Earthquake Risk Model: Technical Manual: Version 3.0, *GA Record 2005/01*, Geoscience Australia, Canberra, 148 pp.

Schorlemmer, D., F. Euchner, P. Kästli, and J. Saul (2011). QuakeML: Status of the XML-based seismological data exchange format, *Ann. Geophys.* **54,** no. 1, 59–65.

Schwartz, D. P., and K. J. Coppersmith (1984). Fault behaviour and characteristic earthquakes: Examples from the Wasatch and San Andreas fault, *J. Geophys. Res.* **89,** no. B7, 5681–5698.

Silva, V., H. Crowley, M. Pagani, D. Monelli, and R. Pinho (2013). Development of the OpenQuake Engine, the Global Earthquake Model's open-source software for seismic risk assessment, *Nat. Hazards*, doi: 10.1007/s11069-013-0618-x.

Thomas, P., I. Wong, and N. A. Abrahamson (2010). Verification of probabilistic seismic hazard analysis computer programs, *Pacific Earthq. Eng. Res. Center Rept. 2010/106*, 176 pp.

Toro, G. R., N. A. Abrahamson, and J. F. Schneider (1997). Model of strong ground motions from earthquakes in central and eastern North America: Best estimates and uncertainties, *Seismol. Res. Lett.* **68,** 41–57.

Youngs, R. R., S.-J. Chiou, W. J. Silva, and J. R. Humphrey (1997). Strong ground motion attenuation relationship for subduction zone earthquakes, *Seismol. Res. Lett.* **68,** no. 1, 58–73.

Weatherill, G., and P. W. Burton (2010). An alternative approach to probabilistic seismic hazard analysis in the Aegean region using Monte Carlo simulation, *Tectonophysics* **492,** nos. 1/4, 253–278, doi: 10.1016/j.tecto.2010.06.022.

Zhao, J. X., J. Zhang, A. Asano, Y. Ohno, T. Oouchi, T. Takahashi, H. Ogawa, K. Irikura, H. K. Thio, P. G. Somerville, Y. Fukushima, and Y. Fukushima (2006). Attenuation relations of strong ground motion in Japan using site classification based on predominant period, *Bull. Seismol. Soc. Am.* **96,** no. 3. 898–913, doi: 10.1785/0120050122.

M. Pagani
G. Weatherill
GEM Hazard Team
GEM Foundation
via Ferrata, 1
27100 Pavia, Italy
marco.pagani@globalquakemodel.org

D. Monelli
L. Danciu
GEM Hazard Team
SED-ETHZ
Sonneggstrasse, 5
8092 Zurich, Switzerland

H. Crowley
V. Silva
GEM Risk Team
GEM Foundation
via Ferrata, 1
27100 Pavia, Italy

P. Henshaw
M. Nastasi
L. Panzeri
M. Simionato
D. Vigano
GEM IT Team
GEM Foundation
via Ferrata, 1
27100 Pavia, Italy

L. Butler
GEM IT Team
SED-ETHZ
Sonneggstrasse, 5
8092 Zurich, Switzerland