COMPLEMENTARY COMPONENTS OF OPENQUAKE AND SHAKEMAP

C. B. Worden(1), M. G. Hearne(2), D. J. Wald(3), M. Pagani(4), E. M. Thompson(5), S. A. Verros(6), G. Weatherill(7)

(1) *Geophysicist, Synergetics Inc. under contract to the U.S. Geological Survey, cbworden@usgs.gov*
(2) *Physical Scientist, U.S. Geological Survey, mhearne@usgs.gov*
(3) *Geophysicist, U.S. Geological Survey, wald@usgs.gov*
(4) *Hazard Coordinator, GEM Foundation, marco.pagani@globalquakemodel.org*
(5) *Geophysicist, U.S. Geological Survey, emthompson@usgs.gov*
(6) *Dept. of Applied Mathematics, Colorado School of Mines, sverros@mymail.mines.edu*
(7) *Seismic Hazard Modeler, GEM Foundation, graeme.weatherill@globalquakemodel.org*

Abstract

In 2017, the U.S. Geological Survey anticipates releasing an updated version of ShakeMap. ShakeMap version 4.0 will represent a major departure from all previous versions of ShakeMap. All of the important computational modules are being refactored into the Python programming language, and make use of the tools in the widely available Python "scientific distributions" (e.g., https://www.scipy.org/stackspec.html). The core ShakeMap code, approaching 15 years old, was overdue for a major overhaul to more organically incorporate (or eliminate) the many extensions that had been added over its lifetime, and to facilitate several new demands from ShakeMap's expanded role as a global provider of post-earthquake information and earthquake scenarios, and as the input to loss-modeling software.

One of the significant factors driving the rewrite of ShakeMap into the Python language is the availability of the library of ground motion prediction equations (GMPEs) and other tools incorporated into the OpenQuake (OQ) hazard library (oq-hazardlib). The OQ hazard library provides a broad range of well-tested, high-performance, open-source global GMPEs. Due to constraints imposed by the software architecture of earlier implementations of ShakeMap, the development and validation of GMPE modules is time consuming and difficult, which restricted the quantity and timeliness of the available modules. The oq-hazardlib provides a broad array of current GMPE and related hazard modules, as well as a framework for easily adding new modules (whether by the Global Earthquake Model [GEM] or ShakeMap staff), jumpstarting our efforts to re-implement ShakeMap. The OpenQuake hazard library also provides supporting functions for using the GMPE modules, including a set of software classes for computing the various distance measures required by the GMPEs. The ShakeMap fault model, however, was somewhat more general than allowed for by the oq-hazardlib planar surface modules, so we have sub-classed the oq-hazardlib "surface" class and implemented our own high-performance module. The open-source, cooperative nature of the OQ project allows us to contribute our new module back to the OQ repository, and thus make it available to other users.

In addition to the GEM OpenQuake hazard library, there are a number of other reasons to use Python in an application like ShakeMap. The dynamic nature of the language means that development time is much reduced, allowing a small team to generate useful code in a short amount of time. Also, there is an active scientific computing Python community that has created many tools that solve common problems, including an array object for vectorized operations, input/output routines for common data formats, and plotting/mapping libraries. These tools again help to reduce development time and effort.

Keywords: ShakeMap, OpenQuake, Python

1. Introduction

ShakeMap [1], developed by the U.S. Geological Survey (USGS), facilitates communication of earthquake information beyond simply magnitude and location. By rapidly mapping earthquake ground motions, ShakeMap portrays the distribution and severity of shaking. This information is critical input for other software, such as Prompt Assessment of Global Earthquakes for Response (PAGER), also developed by the USGS, for gauging the extent of the areas affected, determining which areas are potentially hardest hit, and allowing for rapid estimation of losses. ShakeMap has been in continuous development since 1999, and is implemented primarily in the Perl and C programming languages with heavy reliance on the Generic Mapping Tools (GMT) [2] programs and libraries.

ShakeMap was originally written for use at the Southern California Seismic Network. Over time, it was adopted by many national and international seismic networks as the hazard mapping tool of choice. It is now in operation at all regional seismic networks within the United States that participate in the Advanced National Seismic System, and is the basis of the Global ShakeMap System at the USGS's National Earthquake Information Center in Golden, Colorado. As of mid-2016, most U.S. regional or national seismic systems run ShakeMap version 3.5. The varied nature of its national and international installations has required extensive modifications to the original source code. Additional uses of ShakeMap, such as for scenario earthquakes [3] and the ShakeMap Atlas [4], have also required ongoing modification of the code. As time passed, the code has become increasingly difficult to maintain, challenging to use by anyone besides the developers, and was in need of significant refactorization, documentation, and modernization. In addition, other factors made the existing version of ShakeMap nearly unsustainable:

- To achieve acceptable performance, ShakeMap uses GMT's 'gmtmath' and 'grdmath' programs to perform computations on arrays using Reverse Polish Notation. This approach, while computationally efficient, makes the code difficult to develop, maintain, and debug.
- Because of the small size of the ShakeMap development team, it is difficult to keep up with the rapid pace of developments in the field of earthquake science and engineering. Advances in ground motion prediction equations (GMPEs), directivity, site amplification, finite fault inversions, and intensity measure types and components, all require significant effort to implement, test, and maintain.
- ShakeMap has many software dependencies that can be difficult to install and maintain. Installation frequently requires the intervention of system administrators. Aside from Perl and GMT, the operator must install and configure a MySQL database, ImageMagick, Ghostscript, a C compiler, and numerous Perl modules.
- ShakeMap requires the operator to set up more than a dozen configuration files. Many of these files contain legacy parameters that could be eliminated except that, for support reasons, they are problematic to remove.

Considering the above difficulties with the existing version of the ShakeMap software, the USGS ShakeMap team decided to refactor the entire system and release it as a new version. Version 4.0 of ShakeMap is being developed from a complete redesign of the ShakeMap code base, while maintaining the scientific standards of the original. After considering the requirements of the new ShakeMap system, we elected to do the development in the Python programming language. Python was chosen for a number of reasons:

- Our development team had no high-level language in common. We variously have experience with Perl, R, Matlab, and Python. Python, being the most general-purpose modern language with a very active scientific development community, was an early standout. It is also structurally similar enough to the other high-level languages (especially when the Numpy scientific computing package is added) that it was easy for our non-Python programmers to learn.

- Python has been described as “executable pseudocode”—meaning that it is easy to prototype new programs, often with little more effort than sketching out the basic steps of a module. This feature can result in cleaner code that is easy to develop and maintain.
- Python comes with a huge collection of standard libraries that perform many of the functions that require the installation of external modules in the current Perl-based ShakeMap. For instance, Python’s standard libraries come with modules for parsing and writing JSON and XML file formats, sending email, and handling network connections—all ShakeMap requirements. In addition, Python comes with a built in database, SQLite, allowing us to eliminate one of the biggest and most cumbersome external dependencies in the current ShakeMap system: the MySQL database.
- While we will not eliminate all external dependencies, Python’s powerful packaging tools, *pip* and *conda*, allow us to package ShakeMap and all of its dependencies into a simple installation script that creates a virtual environment in users’ unprivileged account, eliminating the need for an administrator’s involvement in the installation process.
- While Python scripts may not be faster than the equivalent Perl script, the Python module Numpy allows for very efficient array calculations, which ShakeMap is able to exploit for most of its heavy numerical processing. Numpy code is also clean and readable, making the resulting programs more maintainable.
- For more sophisticated processing (e.g., interpolation, optimization) Python’s Scipy library contains a large assortment of well-tested, performance-optimized scientific tools.
- By choosing Python, we are developing in the same environment as the Global Earthquake Model (GEM) group’s OpenQuake, allowing us, as discussed later in this paper, to leverage their work, and to contribute some of our development back to the OpenQuake project.
- The USGS ShakeCast, PAGER, and ‘Did You Feel It?’ (DYFI?) systems are also being re-engineered in parallel as Python-based packages. The common programming base should promote code reuse and efficiencies in development resources.

The GEM Foundation is a non-profit organization dedicated to understanding and helping to reduce earthquake risk. GEM’s OpenQuake platform is a software suite for evaluating earthquake hazard and risk. Within the OpenQuake platform, the oq-hazardlib is a set of software tools for computing earthquake hazard for a variety of source types and tectonic environments (http://github.com/gem/oq-hazardlib). The oq-hazardlib contains modules implementing a large number of GMPEs, as well as tools for defining seismic sources, rupture surfaces, area-magnitude scaling relationships, and other constructs useful for assessing seismic hazard. GEM’s dedicated team of developers works primarily in the Python language in an open-source environment.

To date, we have made significant progress on many of the modules necessary for a complete ShakeMap 4.0 system, and anticipate a release sometime in 2017. In this paper, we will discuss our recent developments, as well as ways in which we are leveraging the resources of the ShakeMap and OpenQuake development teams to speed development and improve our products.

2. Development Philosophy

From its beginnings, ShakeMap has been free, open-source software. We have also worked to make sure that all of ShakeMap’s dependencies are free software and, wherever possible, open source. For ShakeMap V4.0, we are committed to maintaining our free, open-source tradition. In addition, our code base exists in a public repository on GitHub, so that our development process can be as transparent as possible. We are working to minimize external dependencies, making ShakeMap easier to install and maintain. Finally, we are using the open-source tool Sphinx to produce standardized documentation, both at the system and API level.

All of these policies are consistent with the GEM group’s development philosophy. The OpenQuake development is run as an open-source project, with source code that is managed on GitHub for maximum transparency. It is extremely important that the OpenQuake code be reliable and reproducible, so the software is subjected to continuous integration practices that require strict unit testing, long-form program verification, and complete validation of the daily builds. The ShakeMap team has adopted unit testing and continuous integration practices, and intend to begin long-form verification as the ShakeMap programs take shape.

3. ShakeMap Integration with OpenQuake

The development and maintenance of numerical modules for ShakeMap is hampered by the use of legacy technology, as well as by the small size of our development team coupled with our operational responsibilities. The OpenQuake development team, however, does not have these limitations and has thus implemented an extensive set of hazard modules. OpenQuake and ShakeMap share a fundamental need to estimate the ground motions generated by earthquakes, where the source may be given as a point or an extended rupture (fault). For OpenQuake, these ground motions form the basis for probabilistic hazard estimates; for ShakeMap, the ground motions form the deterministic basis that underlies the ground motion interpolation scheme [5]. The following sections outline a number of areas in which we can exploit this commonality and leverage the efforts of our respective development teams to the benefit of the broader hazard and risk modeling communities.

3.1 Ground Motion Prediction Equations (GMPEs)

GMPE implementation around the globe entails an enormous and redundant investment in time for numerous hazard modelers to recode and validate the increasingly complex GMPE algorithms and their coefficients, a process that often requires communications with the developers. These efforts are potentially as error-prone as they are inefficient. However, the oq-hazardlib modules have been extensively validated against the GMPE authors' own data, and are rigorously tested in OpenQuake's continuous integration environment (https://ci.openquake.org/). By developing ShakeMap V4.0 to use the oq-hazardlib GMPEs, we have leveraged the work of the OpenQuake team, which is required to maintain a library of modern GMPEs for a wide variety of tectonic environments, while providing OpenQuake with an additional user group, which enhances the validation of their modules through additional test cases and bug reports. In addition, any new GMPE implemented by ShakeMap operators can be contributed back to the OpenQuake project, further expanding the reach of the OpenQuake system. ShakeMap also relies on intensity prediction equations (IPEs) and ground motion intensity conversion equations (GMICEs), which is also code that can be folded back into the OpenQuake product.

3.2 New Fault Class

OpenQuake's oq-hazardlib provides a variety of methods for specifying rupture surfaces, the most general of which allows for the definition of surfaces of almost arbitrary complexity. ShakeMap, however, requires a fault class that is slightly more geometrically flexible than oq-hazardlib's simple fault class (oq-hazardlib requires faults to consist of rectangular segments) but required that the distance calculations be more efficient than is possible with the most general oq-hazardlib fault class (which is represented as a meshed surface). Our compromise between computational efficiency and geometric flexibility is to define a fault class for specifying faults as a series of planar sub-faults, each of which is represented as an arbitrary quadrilateral, with the restriction that the top and bottom edge of each sub-fault be parallel to the ground surface. The performance characteristics are important because ShakeMap requires fault distances (e.g., distance-to-rupture, "Joyner-Boore" distance) for tens of thousands of surface points per event, which must be computed within the expected ShakeMap production time. To satisfy these requirements, we implemented a new fault class, including the necessary geometric primitives. This new rupture class can also improve computational efficiency during the calculation of seismic hazard with OpenQuake-engine simple-fault sources.

In addition to the traditional fault distance measures, we also implemented the Next Generation Attenuation (NGA) project's Generalized Coordinates 2 (GC2) [6] for the new fault class. The GC2 coordinates allow for easy and stable computation of some of the newer distance metrics required by the NGA GMPEs, such as "R_X" and "R_{Y0}" for complex multiple segment faults. These additions allow the new fault class to produce the appropriate "distance context" required by the OpenQuake GMPEs. Fig. 1 shows four different fault distances (R_{JB}, R_{RUP}, R_X, R_{Y0}) for the Little Salmon fault. The geometry of the fault is taken from UCERF3 [7]. Although this fault consists of three main sections, the full UCERF3 description includes 48 quadrilaterals to allow for variable depth to the top of the rupture due to fault creep. Larger ruptures in UCERF3 are described by more than 500 quadrilaterals. The new ShakeMap fault class is able to handle these ruptures correctly and efficiently.

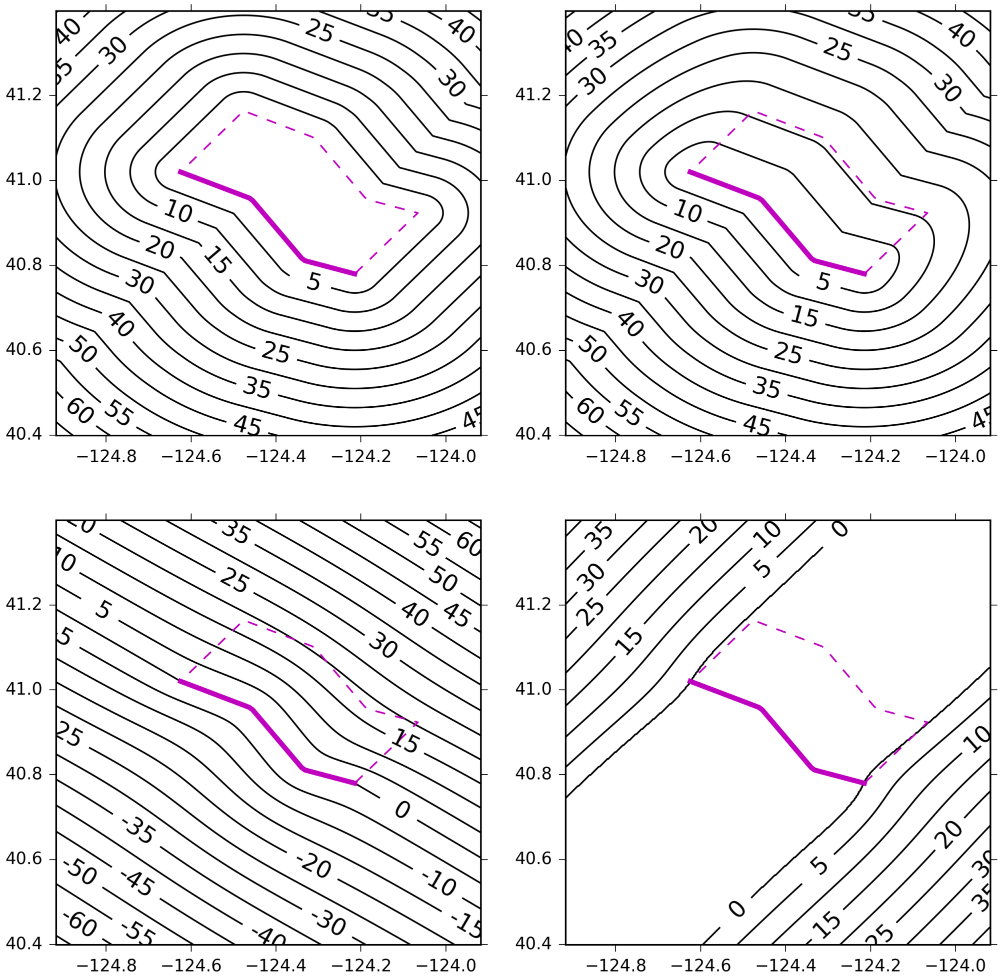


Fig. 1 – Fault distance metrics to the Little Salmon fault in UCERF3 [7]. The surface trace of the fault is shown as a solid magenta line, the surface projection of the downdip portion of the fault is shown as a dashed magenta line. R_{JB} (top left), R_{RUP} (top right), R_X (bottom left), R_{Y0} (bottom right). Distances are in kilometers.

3.3 Directivity

The ShakeMap team has also implemented two source directivity models, specifically, those by Rowshandel [8] and Bayless and Somerville [9], which make use of the new fault class (see Sec. 3.2, above). Separately, the OpenQuake team has implemented directivity models by Shahi and Baker [10] and Spudich and Chiou [11]. Together, these four models comprise the current major directivity models, and may be applied to a variety of fault representations. Fig. 2 shows an example of a directivity factor for a simple fault using Rowshandel's model [8]. The directivity models are generally applied to GMPEs with the following equation:

$$\log(y_D) = \log(y) + f_D \tag{1}$$

where y is the intensity measure predicted by a GMPE, f_D is the modification term predicted by the directivity model, and y_D is the modified intensity measure that includes the effects of direcivity. Thus, $\exp(f_D)$ is the amplification factor due to directivity, which is what we display in Fig. 2. Extensive testing and validation is required for the use of the directivity functions, especially for multiple segments faults. Initial results indicate that not all directivity models scale gracefully with segmentation.

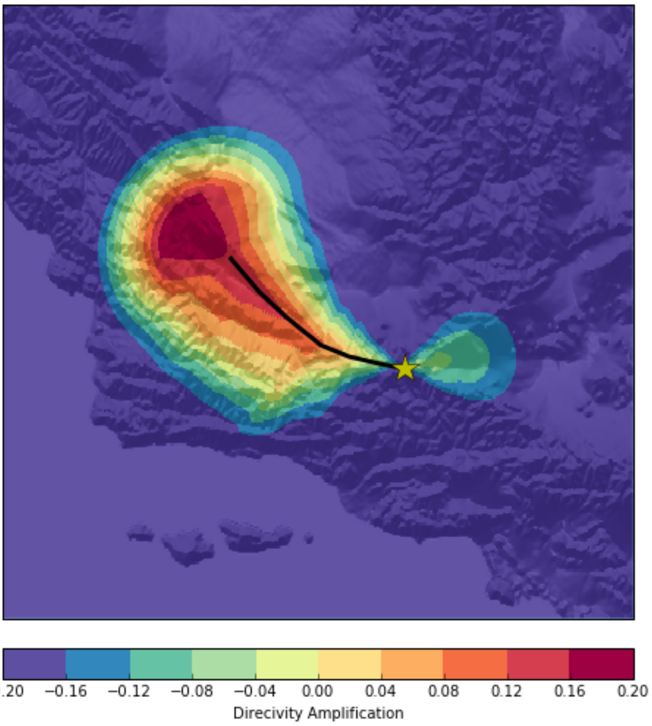


Fig. 2 – Plot of directivity amplification for a multi-segment fault. The epicenter of the rupture is shown as a yellow star; the fault is a black line. These values are computed for a = 0 (only includes the directivity effects of slip direction, not rupture direction) and model type I (the geometric effect is computed by summing only positive contribution subfaults, ignoring the negative contribution subfaults). See 2013 Spudich PEER report [12] for the details of these parameters.

3.4 Distance measures and uncertainty

Frequently, one is required to estimate ground motions for a source, when the source is only specified by a point. In the case of ShakeMap, this may be because a finite-fault model is not yet available, such as in the first versions of ShakeMap for a given significant event, or the earthquake may be of moderate magnitude, in which case no finite fault model may be forthcoming. For OpenQuake, it is commonly required to deal with gridded seismicity where no fault models are available. In both ShakeMap and OpenQuake, unless the GMPE developers provided coefficients to be used with point-source (hypocentral) distances, and few do, using a simple point source is inappropriate and misrepresents both the likely distance to the fault as well as the uncertainty in the ground motions.

To account for the unknown fault orientation (in ShakeMap) or the unconstrained orientation (in oq-hazardlib), we have developed routines that convert from point-source distances to median finite-fault distances and quantify the additional variance of the resulting ground motions [13]. The functions can be tailored to use different sets of assumptions about magnitude-area scaling relationships, rupture aspect ratios, and fault dip distributions (see Fig. 3). Once the assumptions are configured, the functions provide an efficient means of accounting for the unknown fault geometry in ShakeMap and OpenQuake settings.

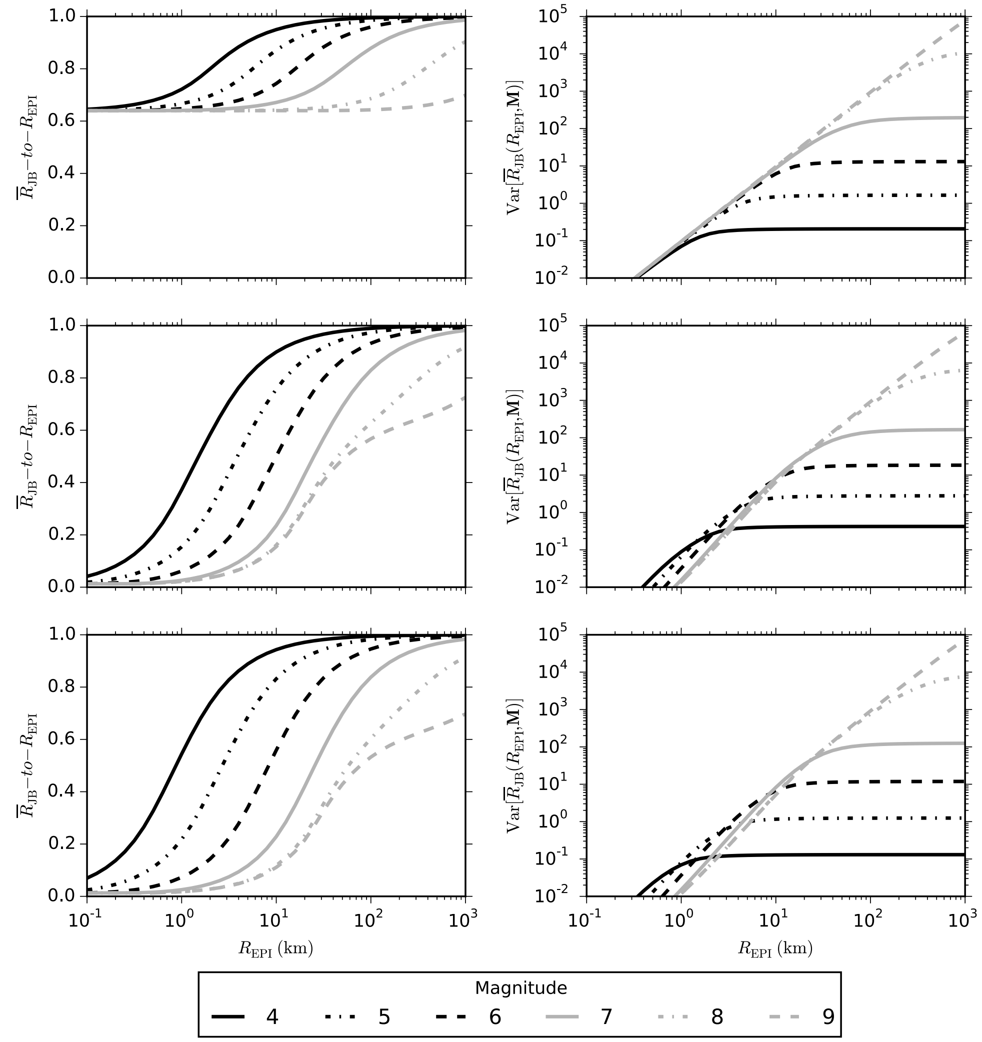
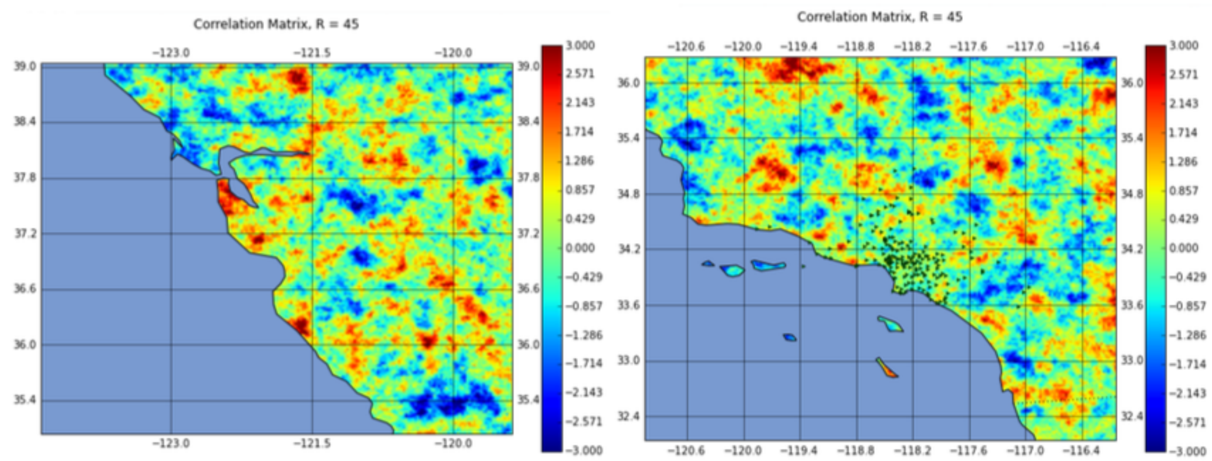


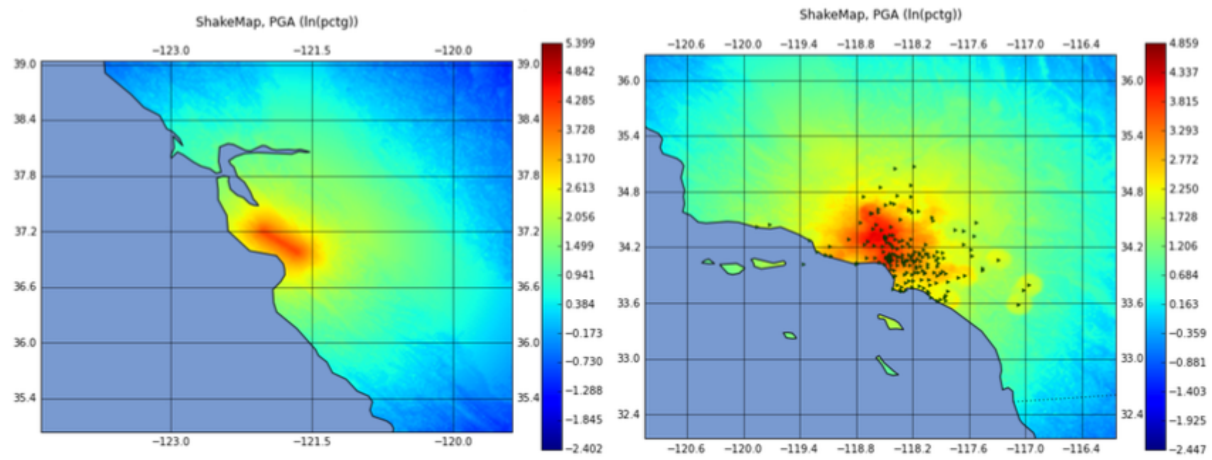
Fig. 3 – R_{JB}-to-R_{EPI} ratio (left) and *Variance[$R_{JB}(R_{EPI}, M)$]* (right) curves for strike-slip (top), normal (middle), and reverse mechanisms (bottom), assuming average dips of 90°, 50°, and 40° respectively. (from Thompson and Worden [13])

3.5 Random Spatial Variability

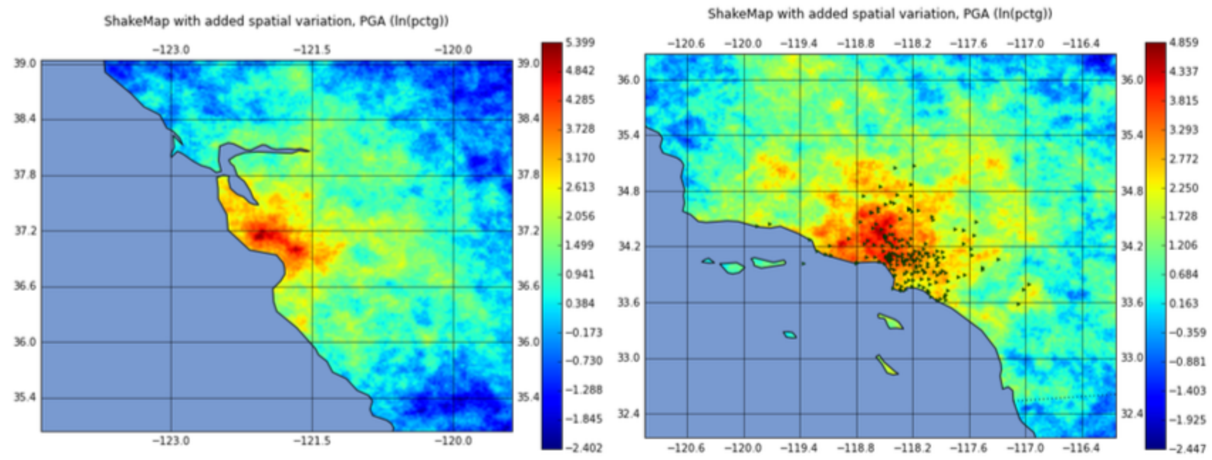
The oq-hazardlib implements functionality to generate realizations of spatially correlated random variability applied to predictive ground motions. This variability may be conditioned upon observed ground motions, and follows the method outlined by Park and others [14]. The oq-hazardlib functions were effective and efficient for a moderate number of output points; however, for ShakeMap, where the number of grid points commonly exceed tens of thousands, the straightforward implementation resulted in unmanageable memory requirements. Additionally, thousands of realizations are typically employed for each scenario event, further motivating additional optimization. Verros [15] implemented the "successive simulation" method suggested by Park and others [14], which dramatically reduced the memory requirements while incurring only a minor computational penalty. Verros further went on to parallelize the computations, which made it practical to compute many realizations of spatially correlated random fields for large areas at fine resolution. Fig. 4 shows examples of adding spatially correlated random variability to ShakeMaps for two significant earthquakes. The addition of spatially correlated random variability has significant effects on the losses computed from ShakeMaps, and is important in reconciling the losses from well-constrained ShakeMaps with those where data are sparse or unavailable (see Fig. 5).



(a) **X**, Loma Prieta 0 stations (b) **X**, Northridge 185 stations



(c) ShakeMap, Loma Prieta 0 stations (d) ShakeMap, 185 stations



(e) Variable ShakeMap, Loma Prieta 0 stations (f) Variable ShakeMap, 185 stations

Fig. 4 – One instance of the generated correlation matrix for the Loma Prieta, Calif., earthquake (a,c,e) (1989) without employing any seismic stations and the Northridge, Calif., earthquake (b,d,f) (1994) conditioned on 185 stations. (from Verros [15])

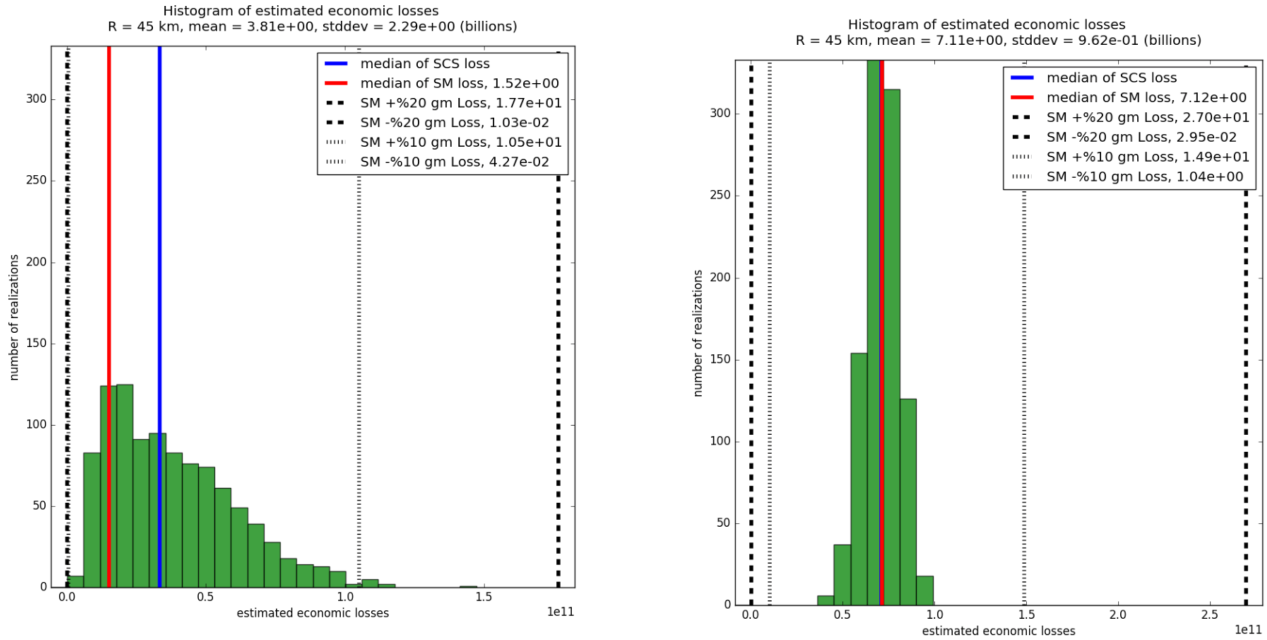


Fig. 5 – Losses from the January 17, 1994 Northridge, Calif., earthquake (in hundreds of billions of dollars) without (left) and with station data (right). Note that the losses from ShakeMap without station data are significantly underestimated (left, red line), while the median of the losses from 1000 realizations that include spatial variability (left, blue line) are considerably closer to the losses when the map is well constrained by data (right, red line). Computations were made using 2016 exposures. (from Verros [15])

3.6 Site Amplification

The application of site amplifications to ground motions has always been a significant part of the ShakeMap process. Recorded ground motions are corrected (that is, de-amplified) to “rock” (that is, to a site condition with a relatively high shear wave velocity such as 760 m/s), the ground motions are then interpolated, and then the interpolated ground motions are re-amplified according to the estimated site conditions (usually based on geology or topography). OpenQuake has embarked on the process of allowing for arbitrary site amplification functions in their hazard calculations. These functions allow the use of empirical amplification factors, which can reduce the overall uncertainty in the prediction of ground motions [16]. Fig. 6 shows the relative reduction in intraevent uncertainty with various sources of site amplification. A more general implementation of site amplification (than V_{S30}) within OpenQuake and ShakeMap would allow for better inclusion of site-specific empirical as well as transfer functions that account for basin effects, directivity effect, and their coupling from 3D simulation-based amplification factors [17].

3.7 V_{S30}

The USGS ShakeMap group developed a global map of time-averaged shear-wave velocity to 30 meters (V_{S30}). This map is useful for computing site amplification terms found in many GMPEs. The model is based on a background map of V_{S30} from topographic slope [18, 19], and embeds more detailed regional maps where such maps are available (e.g., California, Japan, Taiwan) Because site amplification has become an important factor in risk calculations, the OpenQuake group has adopted the USGS V_{S30} map, and is contributing to its support and expansion. The USGS global Vs30 server [20], employing default 30 arc-second topographic slope-based V_{S30} estimates, has been superseded by our repository [21] of functions that allow the user to do more customized calculations of V_{S30}, allowing both selection of geographic domain and slope resolution. It is anticipated that these maps will be further collaboratively enhanced and supported via the ShakeMap and OpenQuake groups.

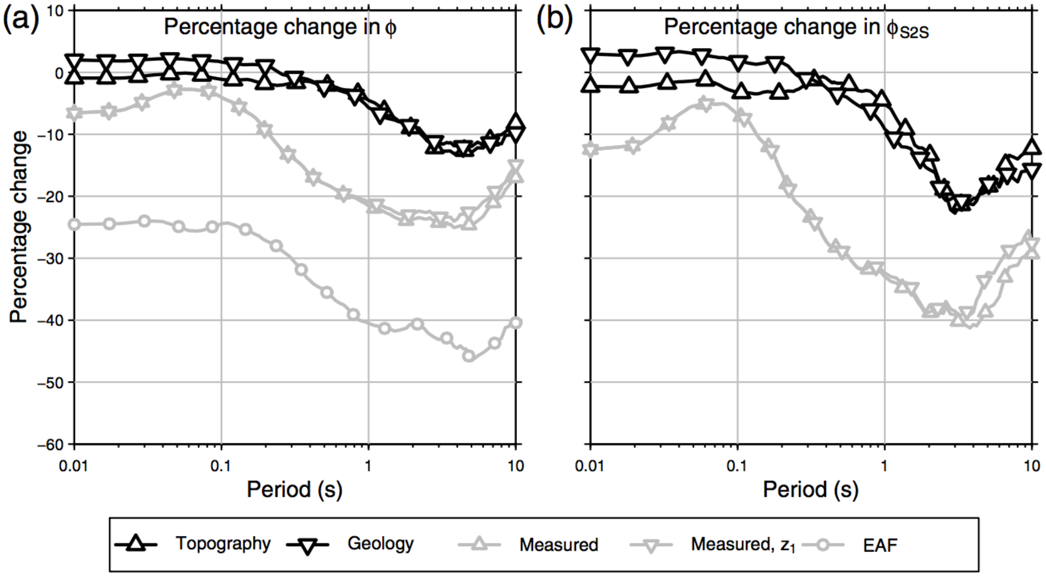


Fig. 6 – Reduction in intraevent standard deviation (a) and site-to-site standard deviation (b) from various site amplification sources. (From Thompson and Wald [16]).

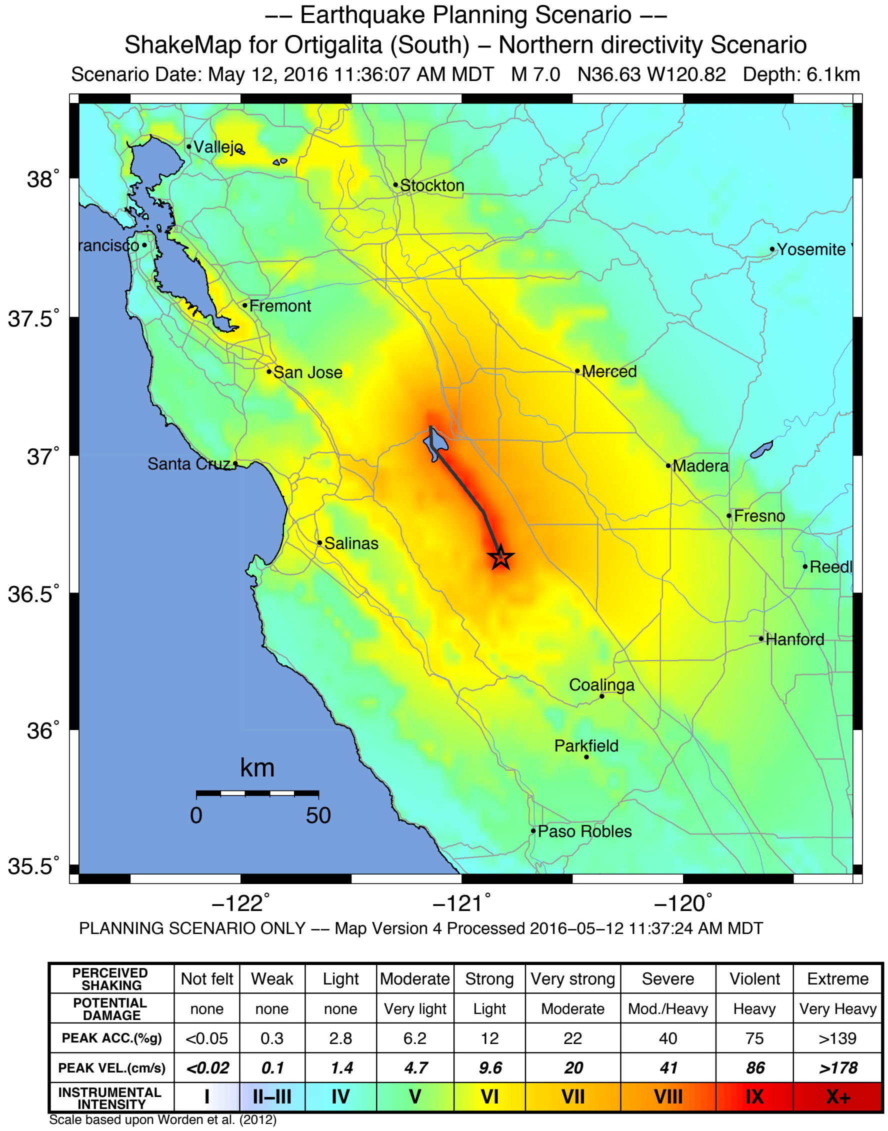
3.7 Earthquake Scenarios

The USGS often receives requests for scenario ShakeMaps. A ShakeMap scenario is a specific realization of the ground motion and impacts from a conceivable earthquake, such as those defined by the National Seismic Hazard Mapping Project (NSHMP), rendered as a ShakeMap and accompanied by standard ShakeMap products, including input for the Federal Emergency Management Agency's Hazus loss-estimation software[22]. There are, however, differences in the shaking calculations between ShakeMap and the 2014 National Seismic Hazard Map [23]. To reconcile some of these differences, we needed the full suite of GMPEs used for the NSHMP's 2014 map [24]. The OpenQuake team had recently completed implementation of those GMPEs, which, with our new fault class (described in Sec. 3.2, above), allowed us to compute ground motions for a large set of possible earthquake scenarios from UCERF3 [7]. Fig. 7 shows an example of one of the new scenarios.

4. Conclusions

The USGS's ShakeMap software is undergoing a much needed update, and the resulting system has been designated ShakeMap V4.0. The new ShakeMap software is being entirely rewritten in the Python programming language and is being developed openly in the GitHub source-control framework. We anticipate a first release of the new version in 2017. In addition to ShakeMap, USGS ShakeCast, PAGER, and DYFI? systems are all being refactored and updated in Python, are publicly accessible via GitHub source control and hosting, and include documentation generated in a mark-down language (specifically, Sphinx), to allow for continuous development and documentation. As in the past, any significant technical or scientific enhancements will be vetted and documented via peer review.

At the same time, the GEM Foundation has continued the development of its innovative hazard and risk modelling software, OpenQuake-engine. We identified a number of areas of overlap between the ShakeMap and OpenQuake systems—primarily in the modelling of ground motions and the representation of finite faults—and are working to integrate the software wherever possible. The result is a more robust ShakeMap system with a much wider array of ground motion models available, and new capabilities and efficiencies in the OpenQuake-engine software. In aggregate, the openly available algorithms and applications described herein span the realm from ground motion estimation for probabilistic and scenario-based seismic hazards, to near-real-time post-earthquake capabilities, which we anticipate will facilitate their use in scientific and risk-reduction studies and applications around the globe.



-- Earthquake Planning Scenario --
ShakeMap for Ortigalita (South) – Northern directivity Scenario
Scenario Date: May 12, 2016 11:36:07 AM MDT M 7.0 N36.63 W120.82 Depth: 6.1km

PLANNING SCENARIO ONLY -- Map Version 4 Processed 2016-05-12 11:37:24 AM MDT

PERCEIVED SHAKING	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
POTENTIAL DAMAGE	none	none	none	Very light	Light	Moderate	Mod./Heavy	Heavy	Very Heavy
PEAK ACC.(%g)	<0.05	0.3	2.8	6.2	12	22	40	75	>139
PEAK VEL.(cm/s)	<0.02	0.1	1.4	4.7	9.6	20	41	86	>178
INSTRUMENTAL INTENSITY	I	II–III	IV	V	VI	VII	VIII	IX	X+

Scale based upon Worden et al. (2012)

Fig. 7 – Example scenario ShakeMap for a M7.0 earthquake on the Ortigalita fault in California. The map makes use of the new fault class, directivity, and multiple weighted GMPEs for ground motion estimates.

5. Acknowledgements

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

5. References

[1] Worden CB, Wald DJ (2016): ShakeMap Manual. February 01, 2016. Accessed May 13, 2016. http://usgs.github.io/shakemap, doi:10.5066/F7D21VPQ

[2] Wessel P, Smith WHF, Scharroo F, Luis JF, and Wobbe F (2013): Generic Mapping Tools: Improved version released. *EOS Trans*. AGU, 94, 409-410.

[3] Thompson EM, Field N, Luco N, Petersen MD, Powers PM, Wald DJ, Worden CB (2016): Systematic generation of USGS earthquake scenarios, *Seismological Research Letters*, Vol. 87, No. 2B.

[4] Garcia D, Mah RT, Johnson KL, Hearne MG, Marano KD, Lin KW, Wald DJ, Worden CB, So E (2012): ShakeMap Atlas 2.0: An improved suite of recent historical earthquake ShakeMaps for global hazard analyses and loss models. *Proc. 15th World Conf. on Eq. Eng.*, Lisbon, 10pp.

[5] Worden CB, Wald DJ, Allen TI, Lin K, Garcia D, Cua G (2010): A revised ground-motion and intensity interpolation scheme for ShakeMap. *Bull. Seism. Soc. Am.* **100** (6), 3083-3096.

[6] Spudich P, Chiou B. (2015): Strike-parallel and strike-normal coordinate system aaround geometrically complicated rupture traces—Use by NGA-West2 and Further Improvements. *U.S. Geological Survey Open-File Report 2015-1028.*

[7] Field EH, Biasi GP, Bird P, Dawson TE, Felzer KR, Jackson DD, Johnson KM, Jordan TH, Madden C, Michael AJ, Milner KR, Page MT, Parsons T, Powers PM, Shaw BE, Thatcher WR, Weldon RJ II, Zeng Y (2013): Uniform California earthquake rupture forecast, version 3 (UCERF3)—The time-independent model: U.S. Geological Survey Open-File Report 2013–1165, 97 p., California Geological Survey Special Report 228, and Southern California Earthquake Center Publication 1792, http://pubs.usgs.gov/of/2013/1165/.

[8] Rowshandel, B (2013): Rowshandel's NGA-West2 directivity model. Chapter 3 of PEER Report No. 2013/09, P. Spudich (Editor), Pacific Earthquake Engineering Research Center, Berkeley, Calif.

[9] Bayless J, Somerville P (2013): Bayless-Somerville directivity model. Chapter 3 of PEER Report No. 2013/09, P. Spudich (Editor), Pacific Earthquake Engineering Research Center, Berkeley, Calif.

[10] Shahi SK, Baker JW (2013): Shahi-Baker directivity model. Chapter 4 of PEER Report No. 2013/09, P. Spudich (Editor), Pacific Earthquake Engineering Research Center, Berkeley, Calif.

[11] Spudich P, Chiou, BS-J (2013): The Spudich and Chiou NGA-West2 directivity model. Chapter 5 of PEER Report No. 2013/09, P. Spudich (Editor), Pacific Earthquake Engineering Research Center, Berkeley, Calif..

[12] Spudich P (editor) (2013): PEER Report No. 2013/09, Pacific Earthquake Engineering Research Center, Berkeley, Calif.

[13] Thompson E, Worden CB (2017): Estimating fault distances without a fault. *Proc. 16th World Conf. on Eq. Eng.*, Santiago, 10pp.

[14] Park J, Bazzurro P, Baker JW (2007): Modeling spatial correlation of ground motion intensity measures for regional seismic hazard and portfolio loss estimation. *Applications of Statistics and Probability in Civil Engineering*, *Kanda, Takada & Furuta (eds).*

[15] Verros SA (2016): A class of efficient algorithms for stochastic seismic ground motions. *Masters Thesis,* Colorado School of Mines, Department of Applied Mathematics and Statistics, 151 pp.

[16] Thompson E, Wald DJ (2016): Uncertainty in V_{S30}-based site response. *Bull. Seism. Soc. Am.,* **106** (2), 453-463.

[17] Moschetti MP, Luco N, Baltay Sundstrom A, Boyd O, Frankel A, Graves R, Petersen M, Thompson E, Jordan T, Callaghan S, Goulet C, Milner K, Maechling P (2017): Incorporating long-period (T>1 s) ground motions from 3-D simulations in the U.S. National Seismic Hazard Model, 16WCEE.

[18] Wald DJ, Allen TI (2007): Topographic slope as a proxy for seismic site conditions and amplification. *Bull. Seism. Soc. Am.* **97** (5), 1379-1395.

[19] Allen TI, Wald DJ (2009): On the use of high-resolution topographic data as a proxy for seismic site conditions (V_{S30}). *Bull. Seism. Soc. Am.,* **99** (2A), 935-943.

[20] http://earthquake.usgs.gov/hazards/apps/vs30/

[21] https://github.com/usgs/earthquake-global_vs30

[22] https://www.fema.gov/hazus

[23] Thompson EM, Wald DJ, Worden CB (2016): A strategy for generating systematic USGS earthquake scenarios. U.S. Geological Survey Open-File Report, in press.

[24] Petersen, M.D., Moschetti, M.P., Powers, P.M., Mueller, C.S., Haller, K.M., Frankel, A.D., Zeng, Yuehua, Rezaeian, Sanaz, Harmsen, S.C., Boyd, O.S., Field, Ned, Chen, Rui, Rukstales, K.S., Luco, Nico, Wheeler, R.L., Williams, R.A., and Olsen, A.H., 2014, Documentation for the 2014 update of the United States national seismic hazard maps: U.S. Geological Survey Open-File Report 2014–1091, 243 p.,*http://dx.doi.org/10.3133/ofr20141091*.