

The SCEC Unified Community Velocity Model Software Framework

by Patrick Small, David Gill, Philip J. Maechling, Ricardo Taborda, Scott Callaghan, Thomas H. Jordan, Kim B. Olsen, Geoffrey P. Ely, and Christine Goulet

ABSTRACT

Crustal seismic-velocity models and datasets play a key role in regional 3D numerical earthquake ground-motion simulation, full waveform tomography, and modern physics-based probabilistic earthquake-hazard analysis, as well as in other related fields, including geophysics and earthquake engineering. Most of these models and datasets, often collectively identified as Community Velocity Models (CVMs), synthesize information from multiple sources and are delivered to users in variable formats, including computer applications that allow for interactive querying of material properties, namely P - and S -wave velocities and density ρ . Computational users often require massive and repetitive access to velocity models and datasets, and such access is often impractical and difficult due to a lack of standardized methods and procedures. To overcome these issues and to facilitate access by the community to these models, the Southern California Earthquake Center developed the Unified CVM (UCVM) software framework, an open-source collection of tools that enables users to access one or more seismic-velocity models, while providing a standard query interface. Here, we describe the research challenges that motivated the development of UCVM, its software design, development approach, and basic capabilities, as well as a few examples of seismic-modeling applications that use UCVM.

INTRODUCTION

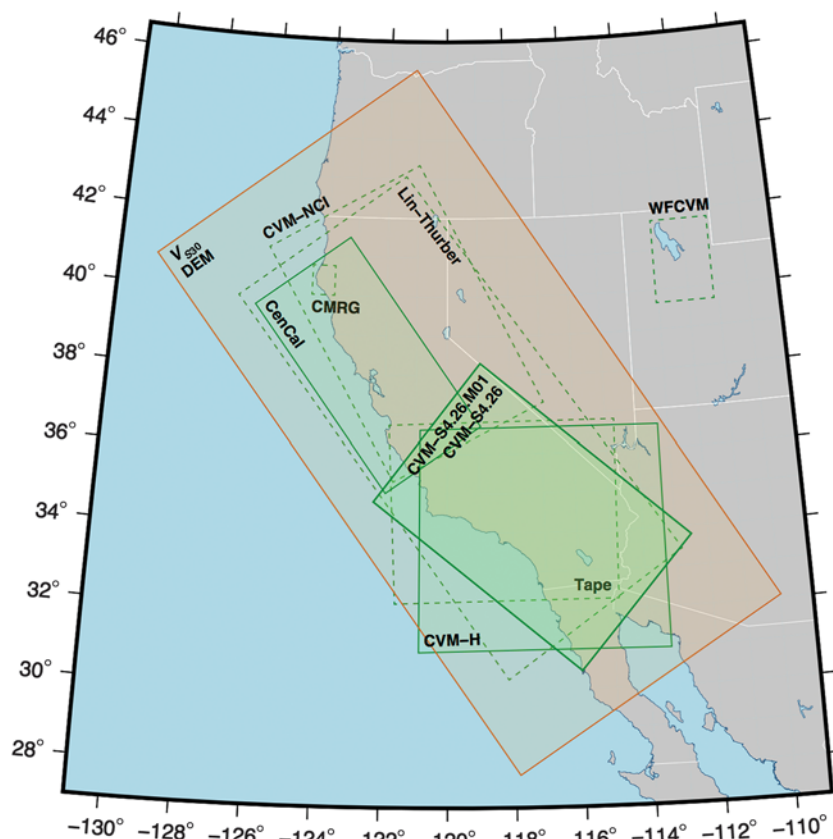
Crustal structure models are numerical representations of the Earth's subsurface geometry and material properties, where the latter are commonly represented by the P - and S -wave velocities (V_P and V_S , respectively) and density (ρ). There are multiple versions of these models—often simply called velocity models—available for different regions, including California, Utah, and the central United States (e.g., Kohler *et al.*, 2003; Süs and Shaw, 2003; Brocher *et al.*, 2006; Magistrale *et al.*, 2006; Ramírez-Guzmán *et al.*, 2012; Shaw *et al.*,

2015). Among these models, those adopted and maintained by specific communities such as the Southern California Earthquake Center (SCEC) or the U.S. Geological Survey (USGS) are referred to as Community Velocity Models (CVMs). Realistic velocity models integrate data from multiple sources (e.g., seismic imaging, geologic mapping, gravity profiles, oil wells, and geotechnical boreholes) to capture the wide range of spatial scales that are important for their multidisciplinary base of users (e.g., geophysicists, seismologists, and engineers).

In contrast to image- or map-based geologic models, which serve primarily for inspection, CVMs provide access to digital datasets in the form of software programs with query interfaces. In recent years, these models have proven to be particularly helpful to regional seismic-wave propagation research and physics-based seismic-hazard assessment (e.g., Aagaard *et al.*, 2008; Bielak *et al.*, 2010; Cui *et al.*, 2010; Graves *et al.*, 2011). However, the type of massive and repetitive individual access to CVMs necessary for applications such as constructing high-resolution regional simulation meshes became impractical and difficult due to lack of standardization among existing models, which limit integration and comparisons. In response to this, SCEC developed the Unified CVM (UCVM) software framework, an open-source collection of tools that enables users to access one or more models, while providing a standard query interface. This article describes the development of UCVM and its main capabilities, and presents examples of how its use benefited science in earthquake simulation, seismic tomography, and hazard analysis.

THE UNIFIED COMMUNITY VELOCITY MODEL

UCVM was developed as part of SCEC's earthquake system science research program (Jordan and Maechling, 2003) and originally designed to support the (forward and inverse) simulation needs of SCEC with regard to models available for California. Although other models from different regions can be registered into the code, we limit our description here to the case of CVMs available for California (and portions of neighboring states). Current models for this region vary from 1D models (e.g., Hadley and Kanamori, 1977), to low- (e.g., Lin *et al.*, 2010) and multiresolution 3D models (e.g., Kohler *et al.*, 2003; Brocher *et al.*, 2006; Shaw *et al.*, 2015), and are based on synthesized information from different datasets (e.g., Magistrale *et al.*, 2000) or on tomography studies that assimilate



▲ **Figure 1.** Map showing coverage region for various models in the western United States. Models registered into UCVM are indicated with continuous lines, others with dashed lines. This map also includes the coverage region for the 2D topography and V_{S30} (outer continuous rectangle) maps overlaid upon the 3D velocity models registered in UCVM (see also Table 1).

recorded data (e.g., Tape *et al.*, 2010; Lee, Chen, Jordan, *et al.*, 2014). Figure 1 shows the coverage regions of several models currently available for the western United States and highlights some of those integrated into UCVM.

Each model in Figure 1 is distributed as its own unique collection of computer readable files or as a computer program. Existing models differ on the material properties they provide (even when overlapping), their coverage area, and their geographical projections and resolution. They also vary from georeferenced grids (e.g., Lee, Chen, Jordan, *et al.*, 2014) to arbitrary precision rule-based programs (e.g., Magistrale *et al.*, 2000; Kohler *et al.*, 2003). Because of these differences and because users may want to query more than one model, UCVM was designed to bridge the gap between CVMs (and other auxiliary datasets) and to meet the particular needs of computational users.

In order for UCVM to operate on these models and datasets, they have to be registered within the framework. Registration of a model or dataset consists of creating an appropriate interface to facilitate the communication between the UCVM tools and the models. Table 1 provides a list of the models registered into UCVM and supported in the current software distribution. In this form, UCVM conceals each

model or dataset behind a generic interface. Once a velocity model or dataset has been registered, a client can use the UCVM framework utilities to query information from the models at any geographic point within the coverage region of the model(s). CVM has been designed to be easily modified to cover any arbitrary region of the Earth's surface, provided adequate velocity and elevation models exist.

In the following sections, we describe the standard query interface implemented in UCVM and how the framework builds upon this interface in several ways. UCVM users can extend the range of registered models, with background models, and through tiling. UCVM also provides algorithms that can add a geotechnical layer (GTL) to underlying velocity models, and offers programs with specialized query features to explore the models and to extract 3D datasets for use in ground-motion simulations. Figure 2 shows the schematic software architecture of the UCVM framework and how these tools build upon and around its standard query interface.

THE UCVM STANDARD QUERY INTERFACE

The primary functionality provided by UCVM is the ability to query a collection of models for material properties through a standardized interface. UCVM users can query models via a command-line program called `ucvm_query` or through a C language application programming interface (API). The `ucvm_query` program is, in fact, a simplified front end to the C API. The interface's input parameters include a model identifier (indicating which model should be accessed) and the requested data points' coordinates (longitude and latitude in World Geodetic System 1984 [WGS-84] decimal degrees, and depth or elevation in meters). The output format includes information about the original input coordinates and the values of V_p , V_s , and ρ , along with other information as explained later with an example. The standard output format returns undefined properties and error conditions in a standard way, too. Zero is returned for V_p , V_s , and ρ in the case of points outside a model's coverage area in the air or in water.

UCVM implements a standard interface by converting the standard query into the native format of each model and then by reformatting the native results back into the standard output. In some cases, such as models defined as regular grid datasets without a query interface, UCVM uses an algorithm for accessing the dataset and linearly interpolates the grid points surrounding the arbitrary location queried by the user.

Data points are queried through the UCVM interface by geographic latitude and longitude (based on the default WGS-84 datum) and a vertical Z coordinate (in meters).

Table 1 Seismic-Velocity Models Registered into the Unified Community Velocity Model (UCVM) and Accessible through Its Standard Interface			
Model Name, Version, and Reference	Model Description	Coverage Coordinates	UCVM Abbreviations
SCEC CVM-H, v.15.1 (Süss and Shaw, 2003; Shaw <i>et al.</i> , 2015)	Southern California 3D velocity model defined on regular mesh without a GTL that uses query by elevation developed by SCEC and Harvard Structural Geology groups	–120.8620, 30.9565; –113.3329, 30.9565; –113.3329, 36.6129; –120.8620, 36.6129	cvmh
SCEC CVM-S4 (Kohler <i>et al.</i> , 2003)	Southern California 3D velocity model defined as rule-based system with a GTL that uses query by depth developed by SCEC, Caltech, and USGS groups	Irregular area	cvms
SCEC CVM-S4.26 (Lee, Chen, Jordan, <i>et al.</i> , 2014)	Southern California 3D velocity model defined on regular mesh without a GTL that uses query by depth based on CVM-S4 as starting model improved using full 3D tomography	–116.0000, 30.4499; –122.3000, 34.7835; –118.9475, 38.3035; –112.5182, 33.7819	cvms5
SCEC CVM-S4.26.M01 (Lee, Chen, Jordan, <i>et al.</i> , 2014)	Southern California 3D velocity model defined on regular mesh with query by depth that adds a GTL to CVM-S4.26	Same as CVM-S4.26	cvmsi
USGS Bay Area Model, Hi-res and Lo-res etree v.08.3.0 (Brocher <i>et al.</i> , 2006)	San Francisco Bay Area and central California 3D velocity model defined on regular mesh with a GTL that uses query by depth	–126.3532, 39.6806; –123.2732, 41.4849; –118.9445, 36.7022; –121.9309, 35.0090	cencal
Modified Hadley–Kanamori 1D (Hauksson, 2010)	Southern California region 1D velocity model in nine layers with query by depth that defines V_P and scaling relationship to calculate V_S and density values	Arbitrary region	1d
Northridge region 1D model (Graves and Pitarka, 2010)	Northridge region 1D velocity model defined in 18 layers with query by depth that was derived from velocity profiles at SCSN stations	Arbitrary region	bbp1d
Topography and V_{S30} model (Gesch <i>et al.</i> , 2002; Wills and Clahan, 2006; Wald and Allen, 2007)	Digital elevation model for California with V_{S30} values based on site classification or topography V_{S30} relationship	–129.7500, 40.7500; –121.0950, 45.7772; –109.9633, 32.0737; –117.8794, 27.8247	ucvm
CVM-H, Community Velocity Model - Harvard; SCEC, Southern California Earthquake Center; SCSN, Southern California Seismic Network; USGS, U.S. Geological Survey.			

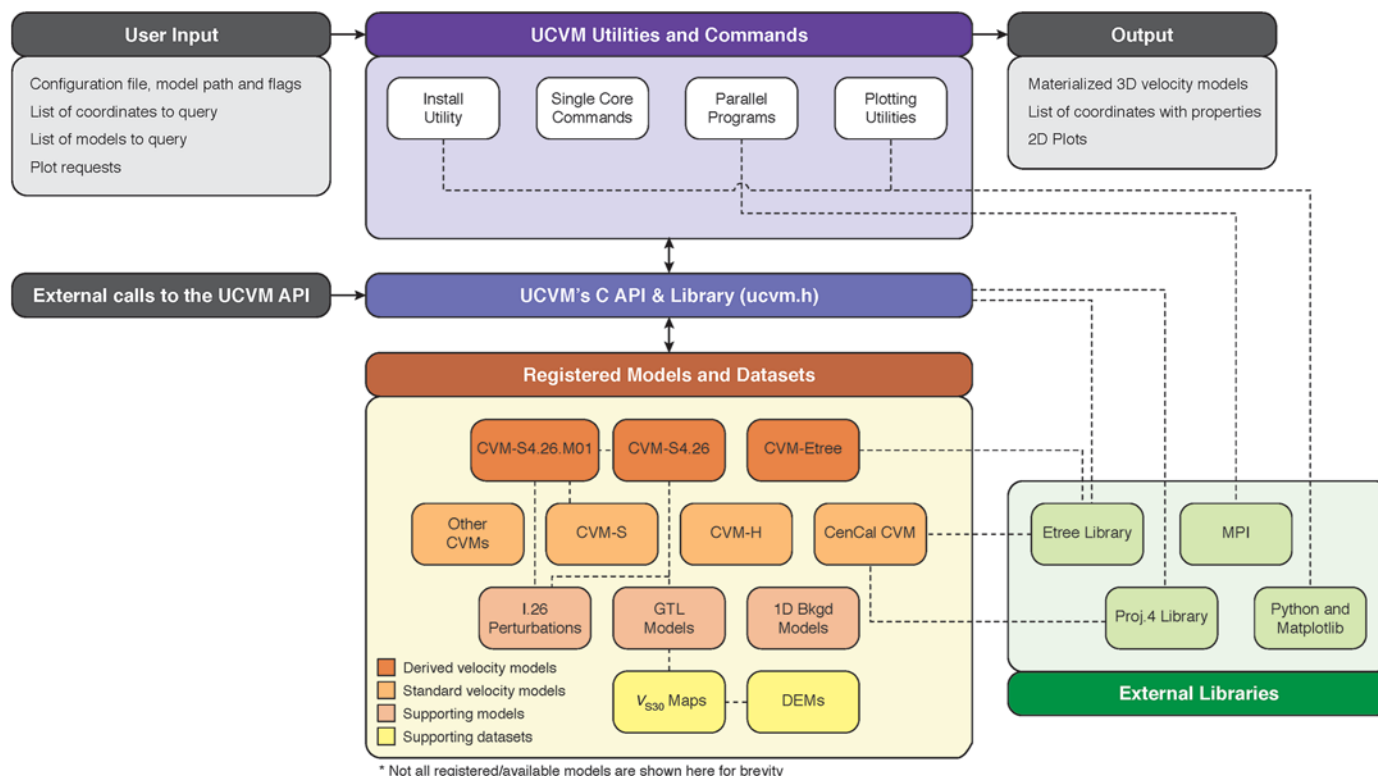
The framework supports queries by depth or elevation by allowing the user to define the orientation and reference of the Z axis through an input configuration parameter. If querying by depth, Z is positive downward with respect to the free surface. If querying by elevation, Z is positive upward relative to mean sea level. To support the latter for models that do not include elevation information, UCVm includes a high-resolution digital elevation model (DEM) and uses a cartographic projection library (PROJ4; see [Data and Resources](#)). The DEM is synthesized from the USGS National Elevation Dataset (Gesch *et al.*, 2002; Gesch, 2007) and the ETOPO1 Global Relief Model (Amante and Eakins, 2009).

OPTIONAL BACKGROUND AND GEOTECHNICAL LAYER MODELS

A general principle of UCVm is that the UCVm query interface should return the same properties as in the original queried

model. However, UCVm can modify the material properties to satisfy specific needs and return both the original and modified material properties in the standard output format.

The first case to consider is that of queries outside the model's coverage region. In this case, UCVm can supplement the properties found in the original model by accessing a background model. Some models, including the SCEC CVM-S4, and the updated model CVM-S4.26.M01, revert to a background 1D velocity model for points outside its defined region by default. However, other models such as CVM-H and CVM-S4.26 are bounded to a particular coverage volume, and they provide no information outside it. For these models, UCVm provides a modified version of the 1D Hadley–Kanamori model (Hauksson, 2010) as background model and returns material properties based on this model if the user query falls outside the model's predefined region. This 1D model was originally defined for the Transverse Ranges and is commonly used for additional regions in southern California.



▲ **Figure 2.** Overview of the UCVM software architecture. The top left and right frames indicate components at the level of user or client interaction for input and output, respectively. The top-center and middle-left frames display UCVM utilities and commands directly accessible to users or clients with access to the models through the box at the center indicating the lower-level UCVM application API and library upon which UCVM operations rest. Underlying this is a selection of the registered models supported by UCVM and to the bottom-right are the library dependencies of the framework.

Users need to use scientific judgment to decide whether this background model is appropriate for a particular region of interest and their specific application.

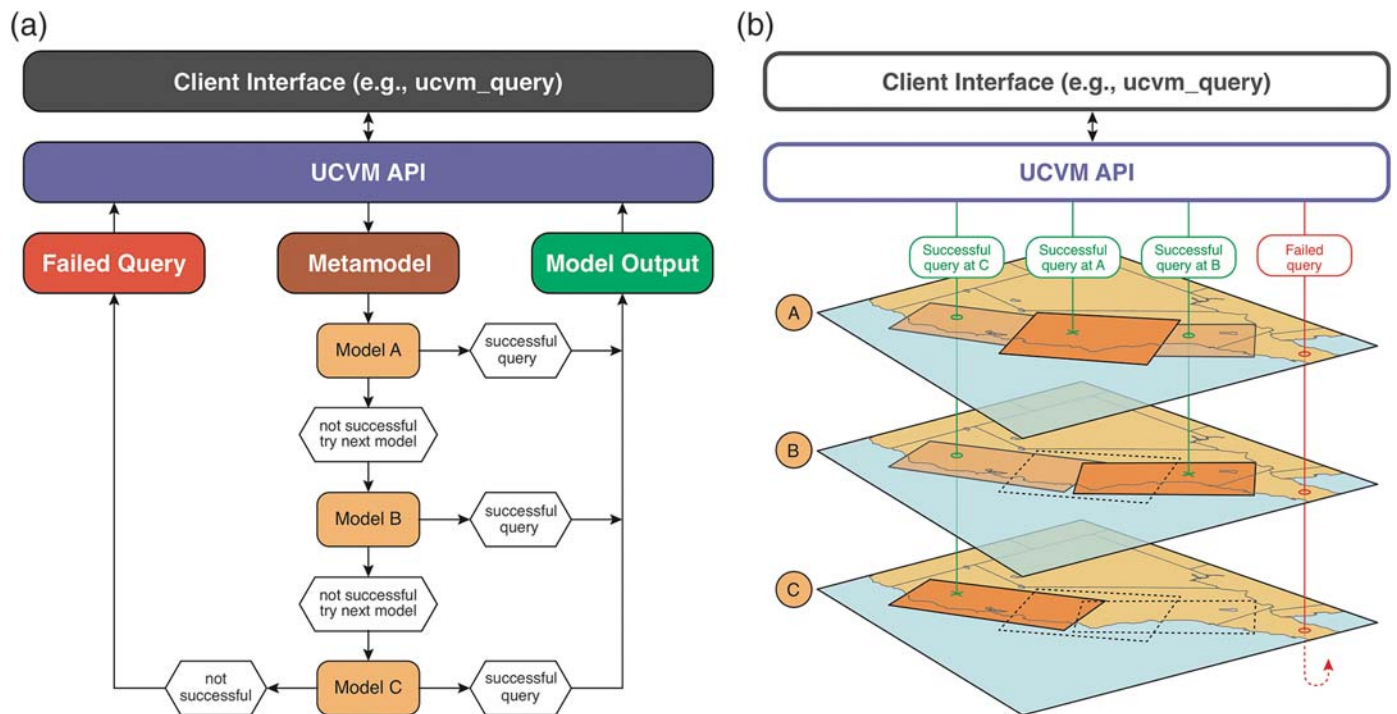
UCVM can also add a GTL model using an optional algorithm to modify the near-surface material properties. The goal here is to emulate the presence of weathered rocks and deposits in the upper few hundred meters and provide a smoother transition to the stiffer bedrock basement. This algorithm implements a V_{S30} -based interpolation from the free surface down to a given depth following Ely *et al.* (2010), in which V_{S30} represents the slowness-based average shear-wave velocity in the upper 30 m. V_{S30} is the most commonly used metric to characterize site-response effects in engineering seismology. To support this model, UCVM includes a built-in surface topography map for California at 1 arcsec resolution (following the USGS National Elevation Dataset standards) with precomputed values of V_{S30} for the region indicated in Figure 1 and Table 1, which covers all of California and parts of neighboring states. The precomputed values of V_{S30} are determined based on the work of Wills and Clahan (2006) within California and Wald and Allen (2007) outside of it. This map is stored as an etree database (Tu *et al.*, 2003).

The background option can be set by the user in a configuration file generated by UCVM upon its successful installation, whereas the GTL option is passed on with the querying

arguments when defining the models to be used by `ucvm_query`.

REGISTERING NEW MODELS INTO UCVM

Before velocity models can be queried by UCVM, the models must be registered into the framework. Registration of a new model involves writing software that converts between the parameters and projection defined in the UCVM standard query interface, and the native interface and projection of the new model. Although, in principle, the open-source distribution of UCVM makes it possible for users to register their own models, in practice, the registration process is usually performed by the SCEC software team. This is mainly due to the unique characteristics of the contributed models, which make a standardized, user-accessible model registration process difficult to define. In general terms, our SCEC software group begins the process of registering a new model by collecting the necessary information about the model, which is then used to develop the software interface and support documentation for distribution with UCVM releases. Such information includes the following: (1) the model's name, description, and abbreviation; (2) the responsible scientist or organization; (3) a publication reference (if available); (4) the geographical volume for which the model is defined; (5) the native geographical



▲ **Figure 3.** (a) UCVM querying scheme and (b) geographical illustration of the querying process. Information at a given geographic location is retrieved from the models registered in UCVM through a hierarchical querying scheme in which the user defines a preferred sequence of models, which are assembled internally in a metamodel. Queries to the models beneath the metamodel are carried out in the order specified by the user. Successful queried values (or failed-query results) are handled by the UCVM API, which is accessible to the user/client through a predefined interface such as the program `ucvm_query`.

projection used by the model; (6) the material properties provided by the model; (7) the description of the formats used by the model to manipulate data (text files, binary files, Hierarchical Data Format [HDF-5] files, or others); (8) the existing query interface (if any); (9) the preferred interpolation method (if needed); (10) whether the model uses elevation or depth as its vertical coordinate system; (11) how the model identifies error conditions, including points outside the defined region, points in the air (above the free surface), and points in water; and (12) any dependencies on external models including digital elevation, bathymetry, or hydrology models, or software libraries.

TILING MODELS

Although the 1D background and GTL options provide simple mechanisms to extend the range of individual velocity models, the need may arise for users to query regions that span multiple models, including regions that overlap between models. To meet this demand, UCVM implements the capability to query multiple velocity models through a process called tiling. This process allows users to define a priority list of models (CVMs, GTLs, and background), which specifies the querying order to be followed. UCVM will query each model in the given order, checking for material properties at the specified point until it either finds a model with material properties, or it determines none of the models are defined for the given point, in which case it returns a failed query indicator. UCVM returns the material

properties as defined in the first model, from its ordered list of models, which can provide properties for the given point. When UCVM returns the material properties for the given point, it also returns a code abbreviation indicating which model provided the results (see Table 1 for a list of abbreviations).

UCVM implements this tiling capability by aggregating multiple models into a composite model, called the metamodel. This composition is accomplished by organizing two or more models in 3D according to the user-specified priority order. UCVM implements this capability for both CVMs and GTLs as illustrated in Figure 3. These metamodels work in the same way as individual CVM and GTL models would work. In fact, when working with a single model (i.e., with no tiling), UCVM still encapsulates the given model into a metamodel.

STANDARD QUERY EXAMPLE

To illustrate several of the features described above, the following example shows how users can call the `ucvm_query` program provided by UCVM. In this and subsequent examples, we assume the user is working with UCVM (v.17.1) on a Linux computer from a terminal running the bash shell and from the UCVM installation directory.

The `ucvm_query` program reads arguments from the system's standard input (*stdin*) and writes results back to the system's standard output (*stdout*). The basic syntax to run the program is

```
./bin/ucvm_query -f[config]
                  -m[model][long][lat][z]
```

in which the `-f` flag is followed by the UCVm configuration file, the `-m` flag is followed by the queried model's abbreviation, followed by query point coordinates (longitude, latitude, and z for depth or elevation). If invoked as shown, the program will return the results to the screen (i.e., the default `stdout`). However, for multiple queries, it becomes convenient to redirect `stdin` and `stdout` to simple ASCII text files. The following example demonstrates how users query for material properties using tiling and a GTL. We define a set of three points at two different locations and three different depths written in an input file called `input.txt` with content:

```
-118.286 34.033 100.0
-118.286 34.033 1000.0
-122.000 34.033 10000.0
and then invoke ucvm_query as follows:
./bin/ucvm_query \
-f ./conf/ucvm.conf \
-m cvmh, cvms, elygtl:ely \
-z 0, 350 \
< input.txt \
> output.txt
```

Here, the `-m` flag defines a prioritized list of models (CVMs and GTLs) from which properties are to be retrieved, indicating the CVM-H model first, the CVM-S4 model second, and adding Ely GTL model and interpolation method (Ely *et al.*, 2010) last. The `-z` flag indicates the depth range that will be modified by the GTL algorithm. The UCVm input points are read in from the file `input.txt`, which contains a list of the query points, with longitude and latitude in decimal degrees and depth in meters. This query does not specify the `-z` flag, so `ucvm_query` will use the default “query by depth” setting and will interpret the third column of the input data as depth.

`ucvm_query` returns an output row with 17 values for each input query point. In this example, `stdout` is directed to the file `output.txt`. The returned values for this particular example are shown in Table 2, where an explanation is given for each output value (for simplicity, the rows that would be in `output.txt` are shown here as columns, and the number in the leftmost column in the table indicates the position in the output rows).

In this example, the output fields corresponding to the GTL information (columns 10–13 in Table 2) are shown only for the query at depth 100 m, whereas the deeper points at depths 1,000 and 10,000 m show the word “none” followed by zeros. This is because the `-z` flag specified the GTL model to take effect only from 0 to 350 m. Therefore, the GTL algorithm did not take effect when querying material properties at the deeper points. So, in those cases, the combined material properties are the same as those provided by the CVM meta-model. We also intentionally defined the third data point in this query to be outside the region covered by CVM-H (first in the `-m` prioritized model list) to show UCVm then satisfies the

material properties request from CVM-S, which was listed as second in priority in the input model list.

Additional details about `ucvm_query` and other UCVm programs can be found in the UCVm documentation as described below.

SPECIALIZED QUERY PROGRAMS

UCVM provides two additional query programs, `basin_query` and `vs30_query`, which implement specialized velocity model queries of interest to users. `basin_query` provides the user with a simple way to extract the depth to a given value of V_S at any location in a velocity model, whereas `vs30_query` returns the V_{S30} value. These programs require input parameters in a slightly different format from `ucvm_query`. The `basin_query` utility accepts a shear-wave velocity threshold (in meters per second), a step interval along the vertical axis (in meters), a prioritized list of velocity models (one or more, to compose the metamodel), and a list of (longitude, latitude) geographic coordinates. For each input coordinate, the metamodel is queried at a sequence of depths, starting at the surface and proceeding downward at the given step interval until either the configured velocity threshold is exceeded or a maximum search depth is reached. The shallowest depth at which this transition occurs, or the maximum depth if no transition is found, is reported to the user. Similar to the example shown for `ucvm_query`, the user can create an input file with multiple points defined as longitude and latitude, but without a depth value. For an input file called `vs30site_input.txt` with three query points

```
-117.286 33.033
-118.286 34.033
-119.286 35.033
the syntax to run the basin_query utility would be
./bin/basin_query \
-f ./conf/ucvm.conf \
-m cvmh \
-v 2500 \
< vs30site_input.txt \
> vs30site_output.txt
```

Here, the additional `-v` flag indicates the V_S threshold value the user considers to be representative of the transition between sedimentary and basement rock materials. The output file duplicates the input coordinates and contains two additional columns: the first column indicates the first occurrence of the specified V_S threshold value, and the second column the last occurrence of that value before reaching the maximum depth. These might differ if the model velocity structure contains velocity reversals. The step size used to advance downward and the maximum depth are set by default to 20 and 350 m, respectively. They can be specified by the user on the command line with `-i` (interval) and `-d` (depth) flags. A parallel version of this command, called `basin_query_mpi`, which uses the message passing interface (MPI) library for parallel computer systems, is also available to quickly generate V_S isosurface maps for large sets of coordinates.

Column Number	ucvm_query		Input/Output		
	Parameter	Description	Input 1	Input 2	Input 3
	Longitude	Coordinates provided as input parameters to the ucvm_query program	−118.286	118.286	−122.000
	Latitude	Coordinates provided as input parameters to the ucvm_query program	34.033	34.033	34.033
	Depth	Coordinates provided as input parameters to the ucvm_query program	100.0	1000.0	10,000.0
			Output 1	Output 2	Output 3
1	Longitude	Input longitude in decimal degrees	−118.2860	−118.2860	122.0000
2	Latitude	Input latitude in decimal degree	34.0330	34.0330	34.0330
3	Z	Input depth (or elevation) in meters	100.000	1000.000	10,000.000
4	Elevation	Returned elevation (m) from UCVM digital elevation model	60.386	60.386	−3,791.911
5	V_{S30}	Returned V_{S30} (m/s) from a Wills–Wald site type to V_{S30} relationship (for points in California), or from Wald topography relation (outside of California)	280.000	280.000	180.000
6	CVM Abbreviation	CVM abbreviation identifying which CVM from the input CVM list provided the returned material properties	cvmh	cvmh	cvms
7	V_P	Returned P -wave velocity (m/s) from the CVM in column 6	1935.708	2533.139	6,300.000
8	V_S	Returned S -wave (m/s) from the CVM in column 6	496.300	1011.327	3,637.307
9	ρ	Returned density (kg/m ³) from the CVM in column 6	1875.844	2103.371	2,859.770
10	GTL name	GTL algorithm used to modify V_P , V_S , and ρ values in the GTL depth range	elygtl	None	None
11	GTL V_P	Returned GTL V_P (m/s) at input point	0.000	0.000	0.000
12	GTL V_S	Returned GTL V_S (m/s) at input point	280.000	0.000	0.000
13	GTL ρ	Returned GTL ρ (kg/m ³) at input point	0.000	0.000	0.000
14	Combining algorithm	Name of algorithm used to combine base model V_P , V_S , and ρ with returned GTL values for V_P , V_S , and ρ	ely	crust	crust
15	Combined V_P	Returned V_P based on combined base model and GTL values	1987.721	2533.139	6,300.000
16	Combined V_S	Returned V_S based on combined base model and GTL values	432.608	1011.327	3,637.307
17	Combined ρ	Returned ρ based on combined base model and GTL values	1899.856	2103.371	2,859.770

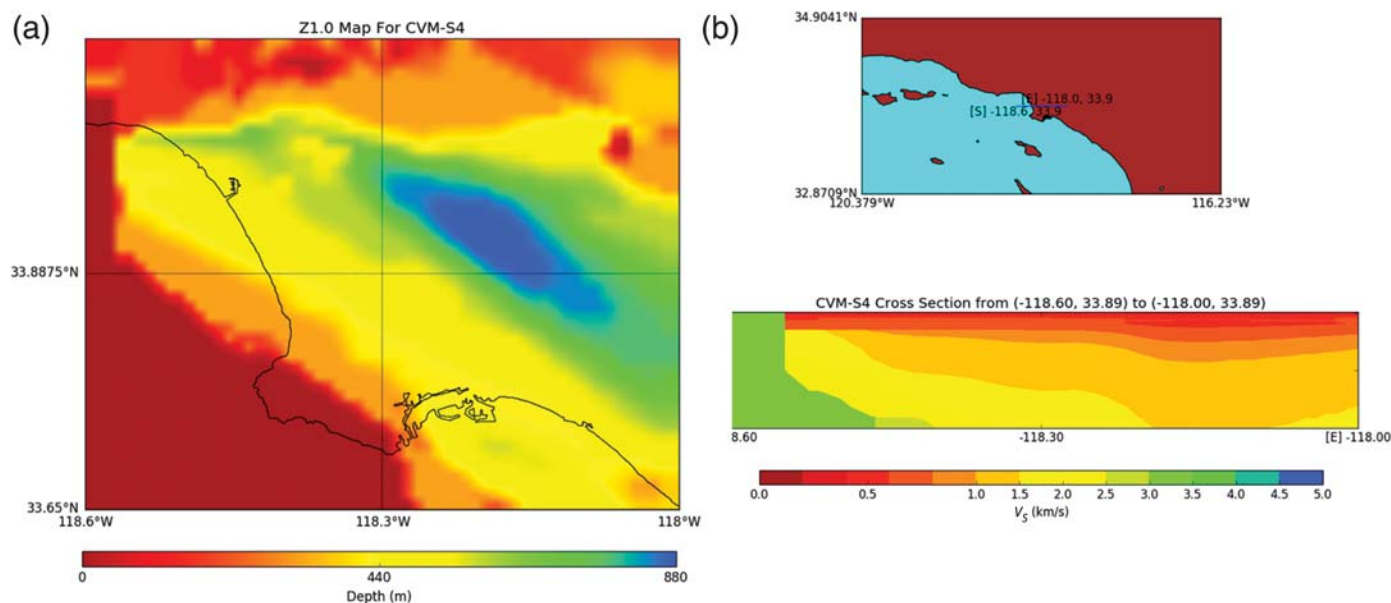
The program `vs30_query` operates in a similar fashion but returns the value of V_{S30} at any given point. It is important to highlight that the V_{S30} values returned by this query correspond to the result computed for the specified metamodel and are not expected to be the same as those provided by the built-in V_{S30} map. That is, they are different to those provided by `ucvm_query` (see Table 2).

VISUALIZATION SCRIPTS

UCVM also provides additional utility programs for visualizing slices and isosurfaces from any configured model. These are provided in the form of Python scripts that function as wrappers to the basic `ucvm_query`, `basin_query`, and `vs30_query` programs.

Plots of horizontal slices and vertical cross sections can be generated with the `plot_horizontal_slice.py` and `plot_cross_section.py` scripts, respectively. Horizontal slices are oriented by a simple bounding box specified in geographic coordinates at a given depth. Cross sections are oriented by two geographic endpoints and a depth range. In both cases, the sliced region is discretized into a regular grid of points queried from the metamodel. The value plotted may be any one of the three material properties (V_P , V_S , or ρ).

Basin maps at predefined threshold values of $V_S = 1000$ and 2500 m/s may be plotted using the `plot_z10_map.py` and `plot_z25_map.py` scripts, respectively. These operate analogously to `basin_query`. Similarly, UCVM can generate V_{S30} maps with the `plot_vs30_map.py` script. In all cases, the output images are saved to disk in Portable



▲ **Figure 4.** (a) Example of a plot generated using UCVM's `plot_z10_map.py` visualization tool, corresponding to the isosurface at shear-wave velocity $V_S = 1000$ m/s in the greater Los Angeles basin. (b) (Bottom) Example of a vertical cross section plot along the (top) east-west line in the map, generated with the plotting utility `plot_cross_section.py`.

Network Graphics (PNG) format. Figure 4a shows a plot example of the isosurface at $V_S = 1000$ m/s for the model CVM-H in the greater Los Angeles basin generated using `plot_z10_map.py`, and Figure 4b shows a vertical cross section of the basin generated using `plot_cross_section.py`.

LARGE-SCALE GRIDDING AND ETREE BUILDING CAPABILITIES

One of the main goals of UCVM is to support numerical ground-motion simulations. To this end, UCVM provides utility programs to discretize a volume from a metamodel and materialize it into uniformly structured grids or unstructured databases.

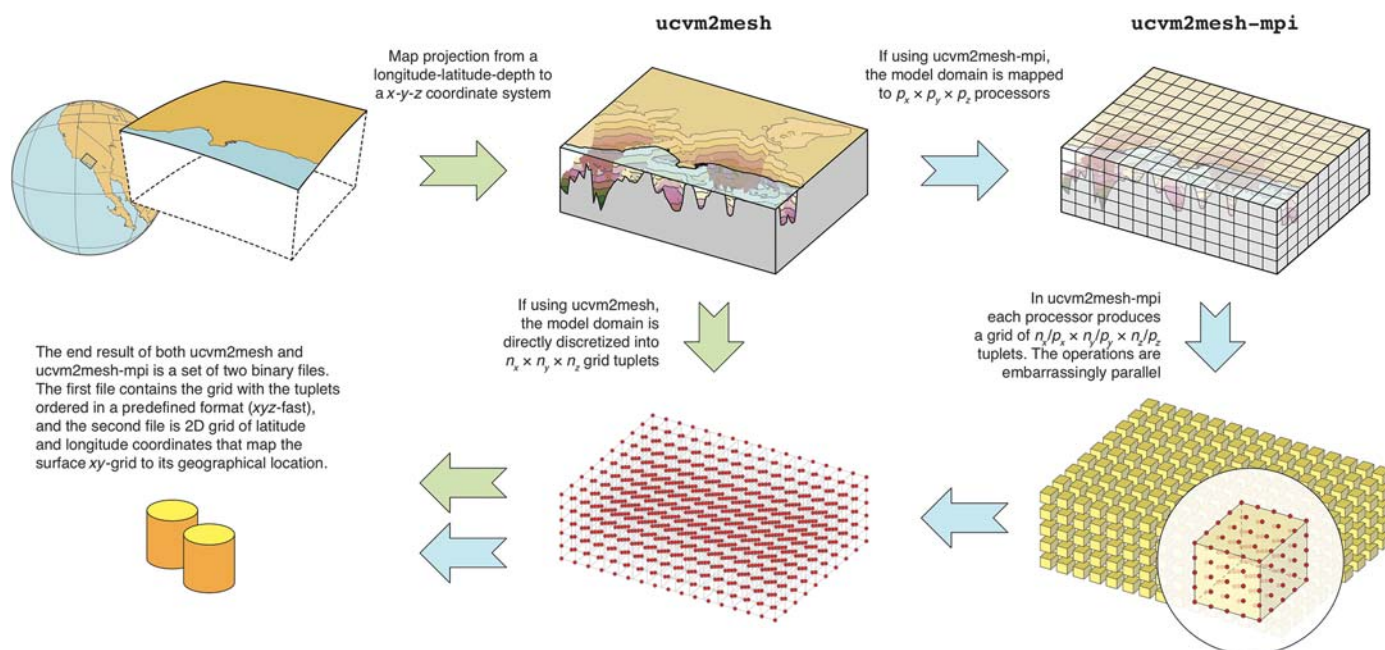
The gridding utilities are particularly tailored to be used by the staggered-grid finite-difference Anelastic Wave Propagation - Olsen, Day, Cui (AWP-ODC) simulation technique (Olsen *et al.*, 2009; Cui *et al.*, 2010; Olsen and Mayhew, 2010). This technique is the core engine of multiple SCEC forward and inverse simulation and physics-based seismic-hazard analysis efforts (e.g., Graves *et al.*, 2011; Lee, Chen, Jordan, *et al.*, 2014). In particular, UCVM offers two alternative gridding programs, a single-core program for local small-scale models called `ucvm2mesh`, and a parallel program called `ucvm2mesh-mpi`, for large-scale regional models used in high-performance computing simulations.

The `etree` building utilities are tailored to materialize velocity models in the form of unstructured `etree` databases (Tu *et al.*, 2003) following Taborda *et al.* (2007) and Schlosser *et al.* (2008). `Etrees` are efficient unstructured on-disk octree-based files used in finite-element simulations carried out with

Hercules, a SCEC-supported forward-wave propagation simulator (Tu *et al.*, 2006; Taborda *et al.*, 2010). In this case, UCVM also offers a single-core program called `ucvm2etree` and a bundle of three parallel applications to extract and optimize `etrees`.

A complete description of the internal details of these capabilities is out of the scope of this article. Therefore, we concentrate our attention on the schematic procedure followed by UCVM for creating structured grids, which is illustrated in Figure 5. In essence, the program `ucvm2mesh` projects the geographic coordinates of the Earth's surface into a Cartesian map projection using an anchor point at the origin of the rectangular area and discretizing the volume according to its dimensions and a given step size. For each point in the projected volume, `ucvm2mesh` determines the analogous geographic point in terms of latitude, longitude, and depth, and queries the underlying user-defined metamodel. The properties are then assigned to the corresponding grid points in the model's volume. In the process, a minimum V_S floor can also be set to bound lower velocities. `ucvm2mesh` outputs two binary files: one containing the geographic coordinates of the grid points (for reference and visualization purposes), and another one with a list of single precision floating point values representing V_P , V_S , and ρ . The Cartesian coordinates of each grid point are given implicitly by the position of the values in the list, which are arranged in x, y, z order.

The parallel version `ucvm2mesh-mpi` works in a similar way but decomposes the volume spatially by mapping sub-blocks of the region to individual processors for extraction. The mapping is specified by providing a processor partition to the program, which specifies the number of processors along each dimension, as illustrated on the right side of Figure 5.



▲ **Figure 5.** Schematic representation of the process implemented by UCV to construct a 3D uniform structure grid using the single-core **ucvm2mesh** or the parallel **ucvm2mesh-mpi** applications.

The only constraint in the mapping is that the processor count along a particular dimension must divide evenly into the number of grid points along that dimension. Using the parallel tools implemented in UCV has greatly sped up the process of building finite-difference grids and etree databases for finite element simulations. This is a process that in the past could take days to do sequentially, and now, thanks to UCV and parallel computers, takes only a few hours.

ADDING SMALL-SCALE HETEROGENEITIES TO MODELS

An advanced capability of UCV is to integrate small-scale heterogeneities (i.e., variations in the material properties) with materialized models (i.e., 3D grids) extracted using the programs described in the [Large-Scale Gridding and ETREE Building Capabilities](#) section. This capability was added to support the demand of numerical regional-scale earthquake ground-motion simulations—currently reaching frequencies of engineering interest (e.g., above 1–2 Hz)—to emulate the spatial variability of material properties observed in nature but not presently included in velocity models.

To this end, UCV implements an algorithm based on statistical descriptions via 3D von Karman autocovariance functions into the program `ssh_generate`, following the procedures introduced by [Olsen and Jacobsen \(2011\)](#) and later used by [Withers \(2016\)](#). This implementation allows the user to choose the parameters of the von Karman distributions (correlation length, Hurst number, and standard deviation from background model), for example, based on analyses of sonic logs ([Savran and Olsen, 2016](#)). The program begins by constructing a 3D grid of the same size and resolu-

tion of one previously created with **ucvm2mesh** or **ucvm2mesh-mpi**, but whose grid-point payloads are perturbations with zero mean and a standard deviation of 1. These perturbations are then applied to the original grid's material properties using the command `ssh_merge`, which iterates through the cells of the two grids and adds the heterogeneities multiplied by a user-defined scaling factor (standard deviation of perturbations relative to the background field) as described in [Withers et al. \(2013\)](#).

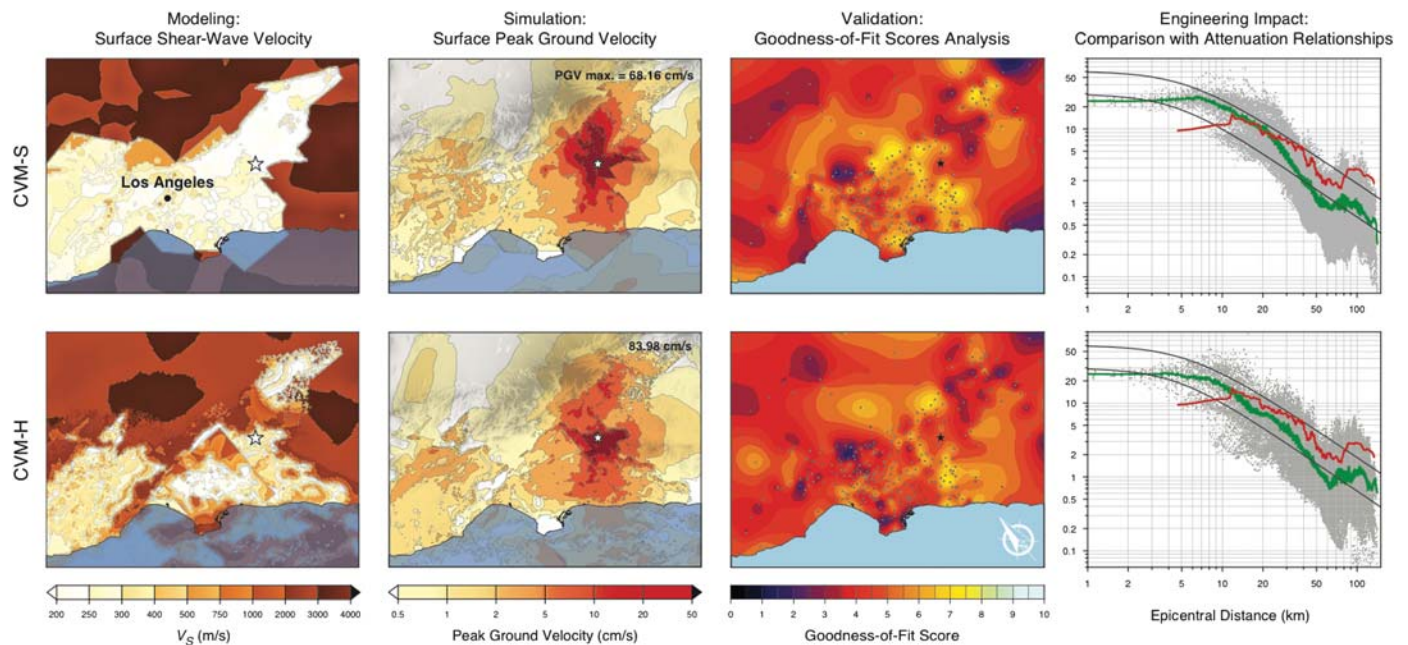
APPLICATION EXAMPLES

The following sections present two application examples of UCV in research efforts relating to seismic-velocity models: (1) the simulation of realistic high-frequency ground motions, and (2) the estimation of seismic hazard in southern California.

Evaluation of CVMs through Validation

One of the major advantages offered by UCV is that it facilitates comparing various velocity models and how their differences influence the results from 3D ground-motion simulations. This helps modelers and simulators build confidence in their work, especially when results can be compared with observations.

In recent years, the 29 July 2008 M 5.4 Chino Hills, California, earthquake has provided an excellent opportunity for earthquake research because it was recorded by over 500 seismic monitoring stations. Various simulations of the Chino Hills earthquake have been done with models created using UCV utilities (e.g., [Olsen and Mayhew, 2010](#); [Taborda and Bielak, 2013, 2014](#)). These simulations helped test current



▲ **Figure 6.** Evaluation of validation results obtained for simulations of the 2008 Chino Hills earthquake using two different velocity models: (top) CVM-S4 and (bottom) CVM-H v.11.9.1. The evaluation includes comparisons of the two input models' V_S surface values (far left), the simulation results for the surface peak ground velocity (center left), the outcome of a goodness-of-fit validation analysis with respect to data where lighter colors indicate more accurate results (center right), and the comparison of results with respect to empirical attenuation relationships used in engineering applications (far right) (modified after [Taborda and Bielak, 2014](#)).

modeling approaches to predict ground motions, but more importantly, in the context of UCVm, they helped evaluate the various velocity models available for the southern California region, and how their differences impact simulation results used for engineering applications.

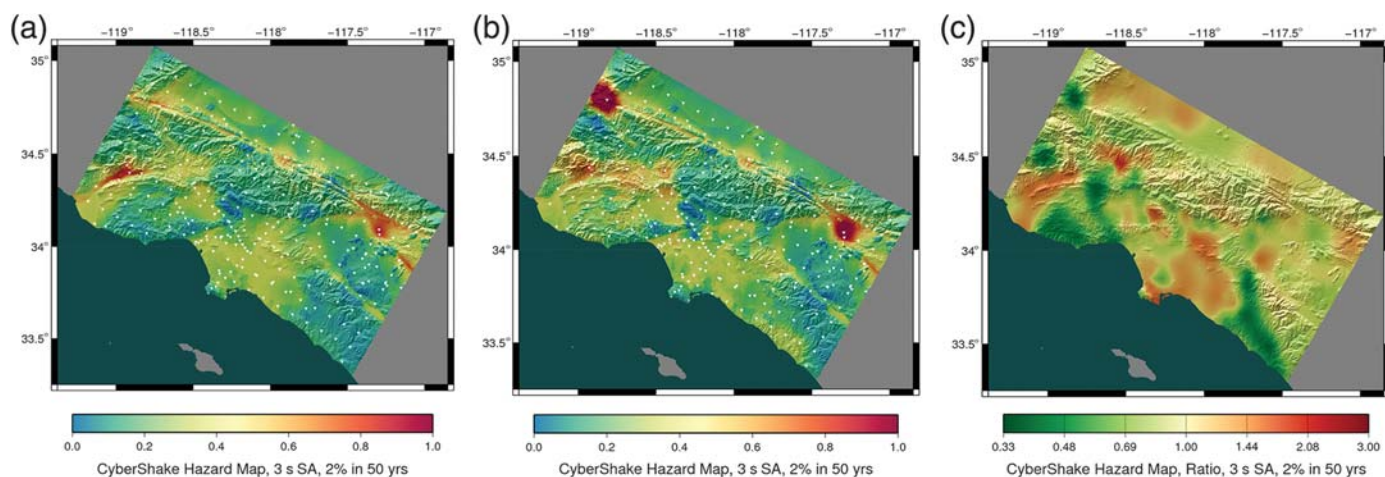
Figure 6 shows comparisons done by [Taborda and Bielak \(2014\)](#) between the surface V_S from two models (CVM-S4 and CVM-H), the surface peak ground horizontal velocity from simulations at a maximum frequency of 4 Hz, the results of goodness-of-fit validations with data in the 0.1–4 Hz range, and the differences of peak velocity attenuation with distance derived from the simulations with respect to data and an upper and lower bound of empirical relationships typically used in engineering. In this study, the discrete representations of the velocity models used as input were created using UCVm programs, which helped the authors evaluate the differences between the models and their effect on validation results over a region with dimensions 180 km by 135 km in the greater Los Angeles area. In particular, the materialized models used in these simulations were created utilizing the etree MPI utilities and were built on Blue Waters, a supercomputer system at the National Center for Supercomputing Applications. The etree databases ranged between 110 and 260 GB, and the resulting finite-element meshes ranged between 5 and 15 billion elements. More recently, [Taborda et al. \(2016\)](#) performed a similar comparative evaluation considering four UCVm-supported models (CVM-S4, CVM-S4.26.M01, CVM-H, and CVM-H with GTL) and observations from 30 different

earthquakes. In all cases, the models used in the simulations were built using UCVm.

Physics-Based Seismic-Hazard Analysis

UCVM also facilitates the construction of numerical models of the crustal structure for carrying out physics-based simulations used in probabilistic seismic-hazard analysis (PSHA). The SCEC CyberShake project provides computational means to perform physics-based PSHA using 3D deterministic wave propagation simulations ([Graves et al., 2011](#)). CyberShake performs PSHA by simulating a tensor-valued wavefield of strain Green tensors and uses seismic reciprocity to calculate synthetic seismograms for over 400,000 events per site of interest ([Zhao et al., 2006](#)). These seismograms are processed to obtain intensity measures, which are then combined with rates of events from the earthquake rupture forecast to produce a PSHA hazard curve. Hazard curves for hundreds of sites are combined into hazard maps, representing the seismic hazard across a region.

As in previous cases, one of the key inputs to CyberShake is the velocity model, which serves as the underlying material structure used to compute the strain Green tensors that are needed to compute site-specific seismograms. Initially, CyberShake was developed using custom interfaces for CVM-S and CVM-H, which added complexity to the CyberShake code base. To simplify this process, CyberShake was transitioned to use UCVm. Subsequently, CyberShake studies were completed in 2013 ([Callaghan et al., 2013](#)), 2014, and 2015. These studies produced a total of 10 hazard models for the Los Angeles area, requiring the generation of 2612 velocity model



▲ **Figure 7.** CyberShake physics-based seismic-hazard models show the different hazard estimates produced when different velocity models and GTLs are used in the simulations. (a) Results using CVM-S4.26.M01, which has a built-in GTL. (b) Results for the same calculation, except using CVM-H with no optional GTL. (c) Ratio of the two maps, emphasizing the areas with differences produced by different velocity models. The underlying hazard maps correspond to a 3 s spectral acceleration (SA) with a 2% chance of exceedance in 50 yrs. The discrete regular grids used as input for these computations were built using UCVm meshing utilities.

grids, each with up to 10 billion mesh points. More importantly, CyberShake has been able to calculate hazard maps for five different velocity models in the Los Angeles area, all while using UCVm utilities. This enabled detailed analysis of the impact of velocity models on seismic hazards. A comparison of CyberShake hazard maps calculated with CVM-S4.26.M01 and CVM-H is shown in Figure 7 as representative of the analysis facilitated by UCVm.

UCVM SOFTWARE DEVELOPMENT AND DOCUMENTATION

UCVM development began in 2011, and SCEC has released several versions of the software since then as open-source scientific software under an Apache 2 license. The starting point for UCVm software and documentation is the SCEC UCVm website (see [Data and Resources](#)) that directs users to the most recent UCVm release and documentation and provides links to earlier versions of the software. This article describes UCVm v.17.1, which was released in January 2017 and is distributed through a public GitHub repository. UCVm v.17.1 is written using the C programming language, but a future UCVm version under development will be based on the Python programming language.

The UCVm software can be built and run on a properly configured Linux computer that contains a software development environment, including Python 2.7, GNU Automake tools, and GNU open-source compilers. The UCVm documentation available on the SCEC website describes the complete software stack required to build UCVm.

The UCVm software distribution provides an installation script called `ucvm_setup.py`. When run, this script prompts the user to identify a UCVm installation directory and the CVMs

to be installed. After collecting this information, the script builds the UCVm programs and installs the selected velocity models, UCVm binaries, and scripts into the UCVm installation directory. The installation script detects whether MPI compilers and libraries are available, and if so, it builds the UCVm MPI programs as well. UCVm releases are also distributed in virtual box image format, with all currently supported models included. A UCVm virtual box image runs more slowly than a direct installation but facilitates the use of UCVm on many types of computers including Mac and Windows. Additional examples and documentation are available on the UCVm website.

CONCLUSIONS

The SCEC community is developing an ecosystem of open-source scientific software designed to support earthquake system science research. Codes in this ecosystem are designed to support specialized applications and research needs and to facilitate exchange of information. Developing independent, but interoperable, computer applications helps accelerate time to solution and science, and helps avoid the need to develop comprehensive monolithic software that can perform all aspects of a research calculation.

In this article, we described UCVm, a software platform for accessing, visualizing, and manipulating seismic-velocity models. UCVm has an important place in the SCEC earthquake system science software ecosystem. It provides CVM developers with the tools needed to review and distribute their models and ground-motion simulation researchers with the software tools needed to use and compare alternative velocity models in high-resolution ground-motion simulations.

DATA AND RESOURCES

The UCVM software and documentation can be found at <https://www.scec.org/research/ucvm> (last accessed August 2017). The CVM-S4 software and data files used in this work were obtained from the SCEC website <http://scec.usc.edu/scecpedia/CVM-S4> (last accessed April 2017). The CVM-H used in this work was obtained from the SCEC website <http://scec.usc.edu/scecpedia/CVM-H> (last accessed April 2017). The Bay Area Seismic-Velocity model used in this work was obtained from a USGS website <https://earthquake.usgs.gov/data/3dgeologic/download.php> (last accessed April 2017). Tomography-based perturbations used in CVM-S4.26 were obtained from researchers including E. Lee and T. Jordan. Other data described in this article including California Site-Conditions Map, the National Elevation Dataset, and the Global Relief Model were obtained from the sources listed in the references. The information on PROJ.4 are from Cartographic Projections Library available at <http://proj4.org> (last accessed July 2017). 

ACKNOWLEDGMENTS

This article is posthumously dedicated to Patrick Small, colleague and friend, who wrote the initial version of the UCVM software and this article, and left us before its publication.

We want to acknowledge the contributions to the work described in the article from the SCEC community velocity modelers including Harold Magistrale, John Shaw, Andreas Plesch, Peter Süß, Carl Tape, Guoqing Lin, En-Jui Lee, Po Chen, Robert W. Graves, and other model contributors. We also thank the Carnegie Mellon etree library development group led by David O'Hallaron for their code contributions to UCVM. The small-scale heterogeneities module was modified based on software provided by Bo Holm Jacobsen from Aarhus University.

This work was supported by National Science Foundation (NSF) Awards: "SI2-SSI: A Sustainable Community Software Framework for Petascale Earthquake Modeling" (ACI-1148493), "SI2-SSI: Community Software for Extreme-Scale Computing in Earthquake System Science" (ACI-1450451), and "Community Computational Platforms for Developing Three-Dimensional Models of Earth Structure, Phase II" (EAR-1349180). This research is also part of the Blue Waters sustained-petascale computing project, which is supported by NSF (Awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications (NCSA). Computational support for this work was provided by National Science Foundation (NSF) Petascale Computing Resource Allocations (PRAC) Awards: "Petascale Research in Earthquake System Science on Blue Waters (PressOnBlueWaters)" (ACI-0832698), "Extending the Spatiotemporal Scales of Physics-Based Seismic Hazard Analysis" (ACI-1440085), and "Improving Earthquake Forecasting and Seismic Hazard Analysis through Extreme-Scale Simulations" (OAC-1713792). Computational support was

also provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program. This research used resources of the Argonne Leadership Computing Facility, which is a Department of Energy (DOE) Office of Science User Facility supported under Contract DE-AC02-06CH11357; and the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. Additional computational support was provided by the Extreme Science and Engineering Discovery Environment (XSEDE) program, which is supported by NSF (ACI-1053575). Some of the computations described here were performed at the Center for High Performance Computing of the University of Southern California. SCEC is funded by NSF Cooperative Agreement EAR-1033462 and U.S. Geological Survey Cooperative Agreement G12AC20038. This is SCEC Publication Number 2067.

REFERENCES

- Aagaard, B. T., T. M. Brocher, D. Dolenc, D. Dreger, R. W. Graves, S. Harmsen, S. Hartzell, S. Larsen, K. McCandless, S. Nilsson, *et al.* (2008). Ground-motion modeling of the 1906 San Francisco earthquake, part II: Ground-motion estimates for the 1906 earthquake and scenario events, *Bull. Seismol. Soc. Am.* **98**, 1012–1046, doi: [10.1785/0120060410](https://doi.org/10.1785/0120060410).
- Amante, C., and B. W. Eakins (2009). ETOPO1 1 Arc-Minute Global Relief Model: Procedures, data sources and analysis, *NOAA Technical Memorandum NESDIS NGDC-24*, National Geophysical Data Center, NOAA, doi: [10.7289/V5C8276M](https://doi.org/10.7289/V5C8276M).
- Bielak, J., R. W. Graves, K. B. Olsen, R. Taborda, L. Ramirez-Guzman, S. M. Day, G. P. Ely, D. Roten, T. H. Jordan, P. J. Maechling, *et al.* (2010). The ShakeOut earthquake scenario: Verification of three simulation sets, *Geophys. J. Int.* **180**, 375–404, doi: [10.1111/j.1365-246X.2009.04417.x](https://doi.org/10.1111/j.1365-246X.2009.04417.x).
- Brocher, T. M., B. T. Aagaard, R. W. Simpson, and R. C. Jachens (2006). The USGS 3D seismic velocity model for northern California, presented at the *2006 Fall Meeting, AGU*, San Francisco, California, 11–15 December, Abstract S51B-1266.
- Callaghan, S., P. Maechling, K. Vahi, G. Juve, E. Deelman, Y. Cui, E. Poyraz, and T. H. Jordan (2013). Running a seismic workflow application on distributed resources, presented at the *SC '13: Proc. of the 2013 ACM/IEEE Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, Denver, Colorado, 17–22 November.
- Cui, Y., K. Olsen, T. Jordan, K. Lee, J. Zhou, P. Small, D. Roten, G. Ely, D. Panda, A. Chourasia, *et al.* (2010). Scalable earthquake simulation on petascale supercomputers, *SC '10: Proc. of the 2010 ACM/IEEE Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, New Orleans, Louisiana, 15–18 November, 1–20, doi: [10.1109/SC.2010.45](https://doi.org/10.1109/SC.2010.45).
- Ely, G. P., T. H. Jordan, P. Small, and P. J. Maechling (2010). A V_{S30} -derived nearsurface seismic velocity model, *AGU Fall Meet.*, San Francisco, California, 13–17 December, Abstract.
- Gesch, D., M. Oimoen, S. Greenlee, C. Nelson, M. Steuck, and D. Tyler (2002). The national elevation dataset, *Photogramm. Eng. Remote Sens.* **68**, 5–11.
- Gesch, D. B. (2007). The national elevation dataset, in *Digital Elevation Model Technologies and Applications: The DEM Users Manual*, D. Maune (Editor), Second Ed., American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland, 99–118.
- Graves, R., T. Jordan, S. Callaghan, E. Deelman, E. Field, G. Juve, C. Kesselman, P. Maechling, G. Mehta, K. Milner, *et al.* (2011). CyberShake: A physics-based seismic hazard model for southern California, *Pure Appl. Geophys.* **168**, 367–381, doi: [10.1007/s00024-010-0161-6](https://doi.org/10.1007/s00024-010-0161-6).

- Graves, R. W., and A. Pitarka (2010). Broadband ground-motion simulation using a hybrid approach, *Bull. Seismol. Soc. Am.* **100**, no. 5A, 2095–2123, doi: [10.1785/0120100057](https://doi.org/10.1785/0120100057).
- Hadley, D., and H. Kanamori (1977). Seismic structure of the Transverse Ranges, California, *Geol. Soc. Am. Bull.* **88**, 1469–1478, doi: [10.1130/0016-7606\(1977\)88<1469:SSOTTR>2.0.CO;2](https://doi.org/10.1130/0016-7606(1977)88<1469:SSOTTR>2.0.CO;2).
- Hauksson, E. (2010). Crustal structure and seismic distribution adjacent to the Pacific and North America plate boundary in southern California, *J. Geophys. Res.* **105**, no. B6, 13,875–13,903.
- Jordan, T. H., and P. Maechling (2003). The SCEC community modeling environment: An information infrastructure for system-level earthquake science, *Seismol. Res. Lett.* **74**, no. 3, 324–328, doi: [10.1785/gssrl.74.3.324](https://doi.org/10.1785/gssrl.74.3.324).
- Kohler, M. D., H. Magistrale, and R. W. Clayton (2003). Mantle heterogeneities and the SCEC reference three-dimensional seismic velocity model version 3, *Bull. Seismol. Soc. Am.* **93**, 757–774, doi: [10.1785/0120020017](https://doi.org/10.1785/0120020017).
- Lee, E. J., P. Chen, T. H. Jordan, P. J. Maechling, M. Denolle, and G. C. Beroza (2014). Full-3-D tomography for crustal structure in southern California based on the scattering-integral and the adjoint-wavefield methods, *J. Geophys. Res.* **119**, 6421–6451, doi: [10.1002/2014JB011346](https://doi.org/10.1002/2014JB011346).
- Lin, G., C. H. Thurber, H. Zhang, E. Hauksson, P. M. Shearer, F. Waldhauser, T. M. Brocher, and J. Hardebeck (2010). A California state-wide three-dimensional seismic velocity model from both absolute and differential times, *Bull. Seismol. Soc. Am.* **100**, 225–240, doi: [10.1785/0120090028](https://doi.org/10.1785/0120090028).
- Magistrale, H., S. Day, R. W. Clayton, and R. Graves (2000). The SCEC southern California reference three-dimensional seismic velocity model version 2, *Bull. Seismol. Soc. Am.* **90**, S65–S76, doi: [10.1785/0120000510](https://doi.org/10.1785/0120000510).
- Magistrale, H., K. B. Olsen, and J. C. Pechmann (2006). Construction and verification of a Wasatch Front Community Velocity Model, *Technical Report 06HQGR0012*, U.S. Geological Survey.
- Olsen, K. B., and B. H. Jacobsen (2011). Spatial variability of ground motion amplification from low-velocity sediments including fractal inhomogeneities with special references to the southern California basins, *4th LASPEI/LAEE International Symposium on the Effects of Surface Geology on Seismic Motion*, University of Santa Barbara, California, 23–26 August.
- Olsen, K. B., and J. E. Mayhew (2010). Goodness-of-fit criteria for broadband synthetic seismograms, with application to the 2008 M_w 5.4 Chino Hills, California, earthquake, *Seismol. Res. Lett.* **81**, 715–723, doi: [10.1785/gssrl.81.5.715](https://doi.org/10.1785/gssrl.81.5.715).
- Olsen, K. B., S. M. Day, L. A. Dalguer, J. Mayhew, Y. Cui, J. Zhu, V. M. Cruz-Atienza, D. Roten, P. Maechling, T. H. Jordan, et al. (2009). ShakeOut-D: Ground motion estimates using an ensemble of large earthquakes on the southern San Andreas fault with spontaneous rupture propagation, *Geophys. Res. Lett.* **36**, L04303, doi: [10.1029/2008GL036832](https://doi.org/10.1029/2008GL036832).
- Ramírez-Guzmán, L., O. S. Boyd, S. Hartzell, and R. A. Williams (2012). Seismic velocity model of the central United States (version 1): Description and simulation of the 18 April 2008 Mt. Carmel, Illinois, earthquake, *Bull. Seismol. Soc. Am.* **102**, 2622–2645, doi: [10.1785/0120110303](https://doi.org/10.1785/0120110303).
- Savran, W. H., and K. B. Olsen (2016). Model for small-scale crustal heterogeneity in Los Angeles basin based on inversion of sonic log data, *Geophys. J. Int.* **205**, 856–863.
- Schlosser, S. W., M. P. Ryan, R. Taborda, J. López, D. O'Hallaron, and J. Bielak (2008). Materialized community ground models for large-scale earthquake simulation, *SC'08: Proc. ACM/IEEE Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, IEEE Computer Society, Austin, Texas, 15–21 November, doi: [10.1109/SC.2008.5215657](https://doi.org/10.1109/SC.2008.5215657).
- Shaw, J. H., A. Plesch, C. Tape, M. P. Suess, T. H. Jordan, G. Ely, E. Hauksson, J. Tromp, T. Tanimoto, R. Graves, et al. (2015). Unified structural representation of the southern California crust and upper mantle, *Earth Planet. Sci. Lett.* **415**, 1, doi: [10.1016/j.epsl.2015.01.016](https://doi.org/10.1016/j.epsl.2015.01.016).
- Süss, M. P., and J. H. Shaw (2003). *P* wave seismic velocity structure derived from sonic logs and industry reflection data in the Los Angeles basin, *Calif. J. Geophys. Res.* **108**, 2170, doi: [10.1029/2001JB001628](https://doi.org/10.1029/2001JB001628).
- Taborda, R., and J. Bielak (2013). Ground-motion simulation and validation of the 2008 Chino Hills, California, earthquake, *Bull. Seismol. Soc. Am.* **103**, 131–156, doi: [10.1785/0120110325](https://doi.org/10.1785/0120110325).
- Taborda, R., and J. Bielak (2014). Ground-motion simulation and validation of the 2008 Chino Hills, California, earthquake using different velocity models, *Bull. Seismol. Soc. Am.* **104**, 1876–1898, doi: [10.1785/0120130266](https://doi.org/10.1785/0120130266).
- Taborda, R., S. Azizzadeh-Roodpish, N. Khoshnevis, and K. Cheng (2016). Evaluation of the southern California seismic velocity models through simulation of recorded events, *Geophys. J. Int.* **205**, no. 3, 1342–1364, doi: [10.1093/gji/ggw085](https://doi.org/10.1093/gji/ggw085).
- Taborda, R., J. López, H. Karaoglu, J. Urbanic, and J. Bielak (2010). Speeding up finite element wave propagation for large-scale earthquake simulations, *Tech. Report No. CMU-PDL-10-109*, Carnegie Mellon University, Parallel Data Lab.
- Taborda, R., J. López, D. O'Hallaron, T. Tu, and J. Bielak (2007). A review of the current approach to CVM-Etrees, presented at the *SCEC Annual Meet.*, Palm Springs, California, 8–12 September.
- Tape, C., Q. Liu, A. Maggi, and J. Tromp (2010). Seismic tomography of the southern California crust based on spectral-element and adjoint methods, *Geophys. J. Int.* **180**, 433–462, doi: [10.1111/j.1365-246X.2009.04429.x](https://doi.org/10.1111/j.1365-246X.2009.04429.x).
- Tu, T., J. López, and D. O'Hallaron (2003). The etree library: A system for manipulating large octrees on disk, *Technical Report CMU-CS-03-174*, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania.
- Tu, T., H. Yu, L. Ramírez-Guzmán, J. Bielak, O. Ghattas, K.-L. Ma, and D. R. O'Hallaron (2006). From mesh generation to scientific visualization: An end-to-end approach to parallel supercomputing, *SC'06: Proc. ACM/IEEE Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, IEEE Computer Society, Tampa, Florida, 11–17 November, doi: [10.1145/1188455.1188551](https://doi.org/10.1145/1188455.1188551).
- Wald, D. J., and T. I. Allen (2007). Topographic slope as a proxy for seismic site conditions and amplification, *Bull. Seismol. Soc. Am.* **97**, 1379–1395, doi: [10.1785/0120060267](https://doi.org/10.1785/0120060267).
- Wills, C. J., and K. B. Clahan (2006). Developing a map of geologically defined site condition categories for California, *Bull. Seismol. Soc. Am.* **96**, 1483–1501, doi: [10.1785/0120050179](https://doi.org/10.1785/0120050179).
- Withers, K. B. (2016). Ground motion and variability from 3-D deterministic broadband simulations, *Ph.D. Dissertation*, SDSU/UCSD Joint Ph.D. Program in Geophysics, 177 pp.
- Withers, K. B., K. B. Olsen, Z. Shi, S. M. Day, and R. Takedatsu (2013). Deterministic high-frequency ground motions from simulations of dynamic rupture along rough faults, *Seismol. Res. Lett.* **84**, no. 2, 335, doi: [10.1785/0220130011](https://doi.org/10.1785/0220130011).
- Zhao, L., P. Chen, and T. Jordan (2006). Strain Green's tensors, reciprocity, and their applications to seismic source and structure studies, *Bull. Seismol. Soc. Am.* **96**, 1753–1765, doi: [10.1785/0120050253](https://doi.org/10.1785/0120050253).

Patrick Small
Department of Computer Science
University of Southern California
3651 Trousdale Parkway, Suite 169
Los Angeles, California 90089-0742 U.S.A.

David Gill
Philip J. Maechling¹
Scott Callaghan
Thomas H. Jordan
Geoffrey P. Ely
Christine Goulet

*Southern California Earthquake Center
University of Southern California
3651 Trousdale Parkway, Suite 169
Los Angeles, California 90089-0742 U.S.A.
maechlin@usc.edu*

*Ricardo Taborda
Center for Earthquake Research and Information, and De-
partment of Civil Engineering
The University of Memphis
3890 Central Avenue
Memphis, Tennessee 38152 U.S.A.*

*Kim B. Olsen
Department of Geological Sciences
San Diego State University
5500 Campanile Drive, MC-1020
San Diego, California 92182-1020 U.S.A.*

Published Online 6 September 2017

¹ Also at Southern California Earthquake Center, University of Southern California, Building ZHS Room 156, Los Angeles, California 90089 U.S.A.