The Unified Community Velocity Model Software Framework

Patrick Small[a,1,*], David Gil[b,**], Phil J. Maechling[b,**], Ricardo Taborda[c], Thomas H. Jordan[b,d], Geoffrey P. Ely[e], Other Authors TBD

[a]*Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA*
[b]*Southern California Earthquake Center, 3651 Trousdale Parkway, Suite 169, Los Angeles, CA 90089, USA*
[c]*Center for Earthquake Research and Information, and Department of Civil Engineering, University of Memphis, Memphis, TN 38152, USA*
[d]*Department of Earth Sciences, University of Southern California, Los Angeles, CA 90089, USA*
[e]*Leadership Computing Facility, Argonne National Laboratory, Argonne, IL 60439, USA*

Abstract

This paper presents the Unified Community Velocity Model (UCVM) software framework developed by the Southern California Earthquake Center. UCVM is a collection of software tools and application programming interfaces designed to provide standard access to multiple seismic velocity models used in seismology and geophysics research. Seismic velocity models are key components of current research efforts dedicated to advancing our knowledge of the Earth's crustal structure and its influence on the ground response during earthquakes, including both the regional deep geology and local effects due to the geometry, spatial distribution, and material composition of sediments in basins and valleys. In general, the UCVM software framework is designed to facilitate all research activities involving the use of seismic velocity models, but its development has been particularly driven by applications for deterministic physics-based earthquake ground-motion simulation and seismic hazard analysis. The UCVM has been extensively used in modeling and simulation activities in southern California. Here we describe the background that led to the development of the UCVM and its various software components, including advanced high-performance computer tools available within UCVM, and its capabilities in terms of product deliverables and performance. We also present examples of recent applications that use UCVM tools or output datasets.

The second to last sentence mentions UCVM performance. Perhaps let us not include a performance analysis (strong/weak scaling, timing, etc) unless we find that we are short on length.

Keywords:
seismic velocity models, earthquake simulation, high-performance computing, meshing

1. Introduction

The quantitative understanding of the physical world is an essential goal of geoscience research. We use mathematical abstractions to represent the behavior of systems under static and dynamic conditions; and properties such as density and elastic moduli to characterize the capacity of materials to absorbe or transmit forces in stationary and transient processes. In seismology and geophysics, our understanding of physical phenomena associated to earthquakes, their genesis, and effects, depends in a good measure on our knowledge and accurate representation of the geometry and material properties of the Earth's structure, as well as on our capacity to represent the mechanical characteristics of the rupture process that takes place when a seismic fault breaks and the subsequent seismic wave propagation problem. For the former case, we use stress conditions and dynamic rupture models to describe the faulting process. On the other hand, for the latter case, we use seismic velocity and attenuation models to describe the geologic structure of the mantle and crust, and the mechanical properties of the materials in the different geologic units that compose these structures to describe the propagation of seismic waves through basins and sedimentary deposits, and thus determine the characteristics of the ground motion.

Source models and seismic velocity models are therefore the basic input to earthquake simulation. We are interested in how seismic velocity models are built and made available to geoscientists, and in particular, on how these models can help advance physics-based earthquake simulation. We utilize physics-based earthquake simulation, the modeling approaches that use deterministic numerical techniques—such as the finite element, finite difference, or spectral element methods—to simulate the ground motion in ways that incorporate the physics of earthquake processes explicitly. That is, methods that explicitly solve the associated wave propagation problem. The use of physics-based earthquake simulation has increased considerably over the last two decades thanks to the growth—in capacity and availability—of high-performance computing (HPC) facilities and applications (e.g., Aagaard et al., 2008; Olsen et al., 2009; Bielak et al., 2010; Cui et al., 2010). These simulations

*Principal corresponding author
**Corresponding author
Email addresses: patrices@usc.edu (Patrick Small), davidgil@usc.edu (David Gil), maechlin@usc.edu (Phil J. Maechling)
[1]This is only a tentative author line-up. Final line-up to be agreed upon by the whole group

have specific applications of great impact in seismology and earthquake engineering in aspects such as the assessment of regional seismic hazard (e.g., Graves et al., 2011).

Recent simulations have highlighted the importance of velocity models in the accuracy of simulation results (e.g., Taborda and Bielak, 2014). Numerous seismic velocity models have been built for specific regional or local structures and used in particular simulations over the years (e.g., Frankel and Vidale, 1992; Brocher, 2008; Graves, 2008). The need for these models in simulation gave way to the conception of community velocity models (CVMs). CVMs are seismic velocity models that have been developed, maintained, advanced and used by a community of interested investigators. Some examples of CVMs for the regions of southern and northern California, Utah, and the central United States are those models developed by Kohler et al. (2003); Brocher et al. (2006); Magistrale et al. (2006) Ramírez-Guzmán et al. (2012), and the CVM-H Harvard model (citation needed).

CVMs have been typically distributed in the form of datasets or collections of files, or in the form of computer programs that can dynamically operate on these datasets and files to provide information about the geometry and material properties of the crust in a particular region. However, these datasets and computer programs have not always been designed carefully from a computational perspective. In addition, recent advances in earthquake simulations, powered by the increasing capability of supercomputers, have increased significantly the computational demand placed on CVMs as input to these simulations.

This paper presents the Unified Community Velocity Model (UCVM), a software framework designed to provide standardized and computationally efficient access to seismic velocity models, developed and maintained by the Southern California Earthquake Center (SCEC). UCVM is a collection of software tools and application programming interfaces (APIs) that facilitate the access to the material properties stored in CVMs. Although UCVM was conceived as a tool to aid physics-based earthquake ground-motion simulation and regional seismic hazard assessment, it can be used in other geosciences and engineering applications. Here, we describe the development of UCVM and its various software components, including features for use in high-performance parallel computers, and present examples of recent applications of UCVM tools in earthquake research.

2. The UCVM Software Framework

The primary functionality provided by UCVM is the ability to query a wide array of CVMs for material properties in standardized formats, independently of the particularities of each dataset or CVM. UCVM achieves this by registering datasets and velocity models into the framework. Registration of a velocity model or dataset consists of creating the appropriate programming application interface (API) to facilitate the communication between the framework utilities and tools, and the velocity models and datasets. Once a velocity model or dataset has been registered with UCVM, a client can use the framework utilities to retrieve information from the models at any geographic point within the coverage region of the model. A client can be either a user (via the command-line) or another software. The primary data-point typically retrieved by a client consists of a float triplet with the seismic velocities (V_P and V_S), and the material's density (ρ). At times we refer to this triple as the payload. The UCVM can then be used to produce standardized output in the form of three-dimensional (3D) volumetric datasets, two-dimensional (2D) vertical cross-sections and horizontal slices, and individual data-points. A client can also use other UCVM utilities for plotting and transforming models and datasets.

In order to facilitate access to the models, UCVM conceals each model's local coordinate system behind a generic querying interface. Data points are queried through this interface by geographic latitude and longitude, and a vertical z-coordinate. The framework allows defining the z-axis as either depth below the free surface (in meters, positive downward) or elevation relative to mean sea level (where zero is at sea level, positive upward and negative downward). The framework further extends the standardized interface by allowing multiple velocity models to be aggregated into a single composite model. Composition is accomplished by tiling two or more velocity models in three dimensions according to a user-specified priority ordering. To support this flexible query mechanism consistently across all models, UCVM includes a high-resolution digital elevation model (DEM). The DEM is synthesized from the USGS National Elevation Dataset (citation needed) and the ETOPO1 Global Relief Model (citation needed). An additional advantage to providing the built-in DEM is that the client can retrieve the surface elevation at any query point in addition to the default data-point payload (V_P, V_S), ρ).

The abstraction of native CVM interfaces under a uniform API is illustrated by Figure 1. This API is written in the C language, and may be invoked directly by a user program to query any registered model as described previously. Alternatively, the framework provides a small set of pre-defined command-line tools to perform common operations, such as querying points, creating meshes, and producing simple graphs. As these tools are themselves defined in terms of the API, they may be leveraged as templates for creating new utilities. This layered approach to the framework design allows for future extensibility both in terms of new model support, and querying functionality.

With the exception of the Wasatch Front (Utah) CVM, currently the primary focus of UCVM has been on models available for the State of California (and portions of neighboring States). However, the framework has been designed to be easily modified to cover any arbitrary region of the Earth's surface, provided adequate resolution velocity and elevation models exist. Additional details about the models available through UCVM are given in the following section on Community Velocity Models. Subsequent sections provide further information on the main UCVM utilities and APIs. However, due to space limitations, not all UCVM utilities and options can described here. General and advanced users should refer to on-line manuals and documentation. Table 1 provides URL addresses linking to supporting material. The last section of the paper is dedicated to additional aspects on the computational performance of the

Table 1: Electronic addresses to UCVM on-line documentation.

Description	URL Address
General Documentation	http://scec.usc.edu/scecpedia/UCVM
General User Guide	http://scec.usc.edu/scecpedia/UCVM_User_Guide
Advanced User Guide	http://scec.usc.edu/scecpedia/UCVM_Advanced_User_Guide
Tutorial	http://scec.usc.edu/scecpedia/UCVM_Tutorial
Model Integration	http://scec.usc.edu/scecpedia/UCVM_Model_Integration_Guide

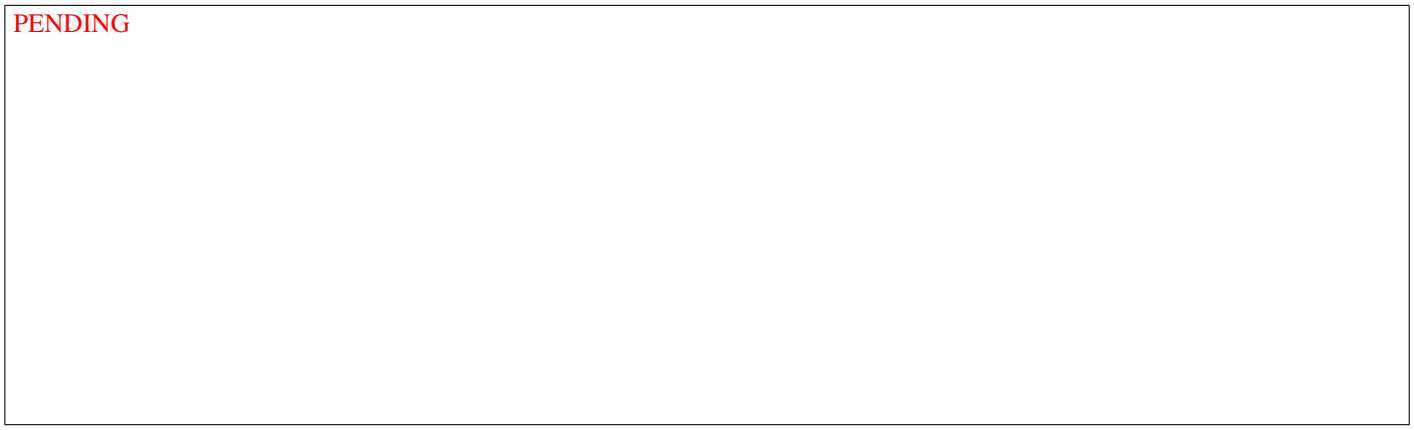

Figure 1: The UCVM software architecture.

UCVM framework and two recent case applications.

3. Supported Velocity Models and Datasets

Velocity models vary considerably in terms of area of coverage, depth extent, composition, and resolution. The UCVM framework is flexible in its support for such variability and has been designed to integrate different velocity models, as well as to interact with their individual features seamlessly. UCVM has built-in support for a number of standard CVMs, which the platform utilities identify through a series of corresponding string labels. Table 2 lists the models currently supported in the UCVM framework and provides additional details and references, along with their string labels. While all these models are supported by UCVM, only a fraction of them are included in the automated installation package, the remaining ones need to be installed manually before they can be accessed through the platform. Additional details are given in Section **??**. We indicate in the table which of the models are included in the automated installation and which require manual installation. Figure 2 shows a map with the coverage areas of various CVMs in Table 2.

Velocity models need to be enabled at installation time. UCVM is distributed with an easy installation method which runs a Python script (`ucvm_setup.py`) that prompts the user about which of the automated models are to be installed (see Section **??**. If the user wants to enable other velocity models at a later time, the installation process needs to be repeated to enable the desired additional models. Advanced users can also customize the installation to include other models, including user-defined velocity models, which can be added in the form of an *etree* (Tu et al., 2003) database or as a *patch* (rasterized) model. These advanced features are described in detail in the UCVM Advanced User guide (see Table 1).

In addition to the standard velocity models, UCVM includes two geotechnical layer (GTL) models, also listed in Table 2. These models are intended to supplement (replace) the near-surface information in the original velocity models for a smoother transition from the softer near-surface soil deposits to the stiffer bedrock basement. The first of the two GTL models implements a V_{S30}-based interpolation from the free surface down to a given depth z, following Ely et al. (2010). The second GTL model is a generic one-dimensional (1D) model identical to the 1D crustal model. Two interpolation schemes can be used to smooth GTL material properties with the underlying crustal model material properties: a linear interpolation, or the interpolation relationship used by Ely et al. (2010). As in the case of the CVMs, the GTLs and the interpolation schemes are identified with string labels, `elygtl` and `1dgtl` for the V_{S30}-based and the 1D models, respectively. Similarly, the interpolation schemes are identified with the labels `ely` and `linear`. Note that the Ely GTL and interpolation models are used in the CVM-H model by default, with a reference depth of $z = 350$ m. This option can be turned on or off for CVM-H by setting the model flag `USE_GTL = true/false` in the appropriate section of the UCVM configuration file, as shown in Figure **??**.

To support the V_{S30}-based GTL model, UCVM has two built-in standard maps for California at 1 arcsec resolution (following USGS NED standards). These maps contain elevation data and V_{S30} data for the region and V_{S30} values are assigned following one of two possible options. Option one implements the procedure by Wills and Clahan (2006) and Wald and Allen (2007).

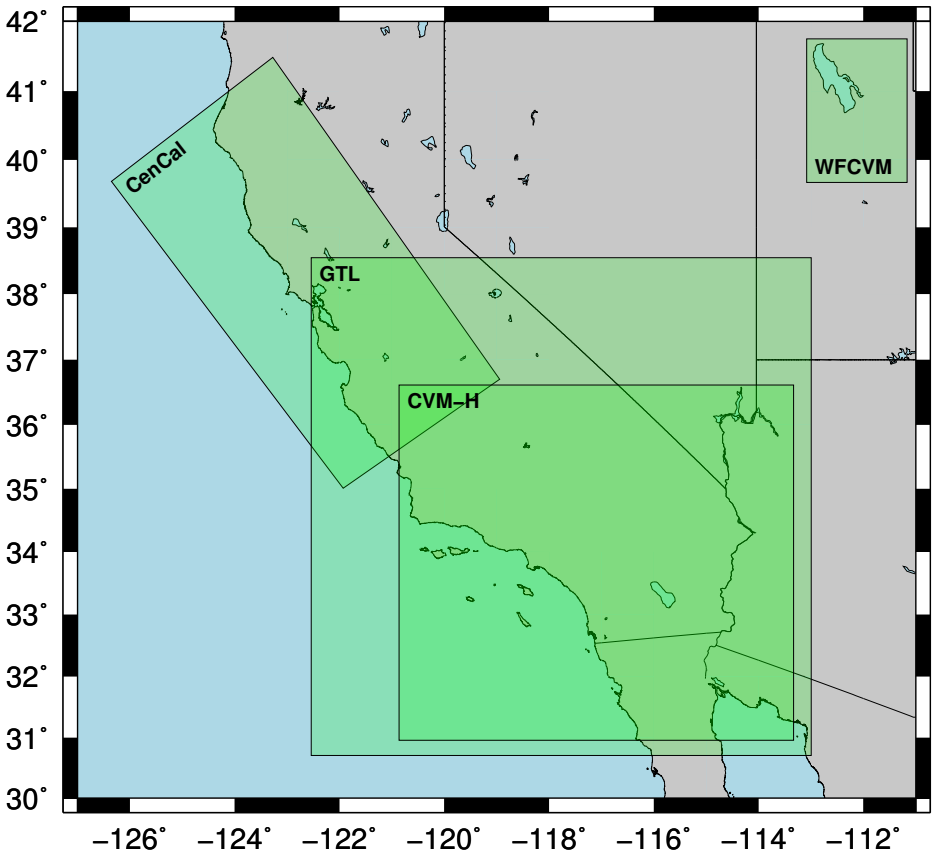

Figure 2: Temporary mock-up figure done in GMT to be improved later once we have all the boxes in Table 2. Surface horizontal projection of the areas covered by the various standard velocity models supported by the UCVM platform.

Option two follows Yong et al. (2012). These options are identified by the string labels `ucvm`, which is the default option, and `yong`. These maps are stored as etree databases, and their details are included in Table 2 and Figure 2.

The UCVM framework also provides an optional background 1D model to support queries at points outside and below the domains covered by the standard CVMs. This option is inactive by default and can be controlled manually by setting the model flag `USE_1D_BKG = true/false`.

The USE_1D_BKG option applies only to CVM-H. If the user wants to use the default 1D background with UCVM, they need to provide it in the list of query models, following the main models that they are interested in.

4. Features

The UCVM platform offers an API and a set of programs, many of which can be used either in a single processor context or in parallel. This section details the most relevant of these programs, categorized by the broad feature they are intended to support. The discussion will focus largely on how the single processor programs operate. Those features that have additional support for parallel computing will be noted, and any operational differences between the single core and parallel implementations shall be described.

The single-core commands can be run in any system where the platform has been successfully installed. The parallel commands, however, require a system where the standard Message Passing Interface (MPI) library and compilers are available. Single-core commands are useful to most users interested in exploring the properties of the regions covered by the models supported by the platform. On the other hand, advanced users needing to build large-scale (regional) materialized velocity models for earthquake modeling and simulation are the more likely to use the MPI commands.

4.1. Querying Material Properties

UCVM provides two methods for querying models. The first method is programmatically, directly through the provided C language API. The second is via the command-line program `ucvm_query`. Both methods query the underying models in the same manner; the `ucvm_query` program is merely a simplified front-end layered upon the API.

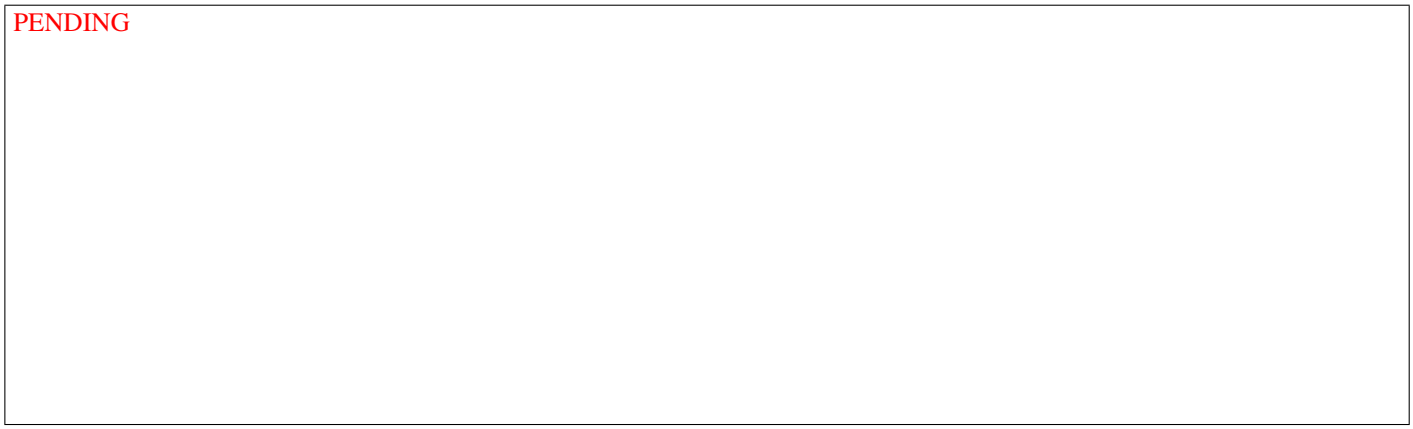

Figure 3: Querying a geographic point for material properties.

The query process begins with the identification of one or more CVMs as the source of material properties. In this respect, the framework distinguishes between geo-technical layers and standard crustal models, allowing the user to make selections for both. As illustrated in Figure 3, the set of standard crustal models is tiled in three dimensions to form a meta-model. The same operation is performed on the GTLs to define a meta-GTL. The interface between the meta-GTL and meta-model is then smoothed using an interpolation function (linear interpolation in the simplest case) along a user-defined interpolation zone parallel to the z-axis. Note that when two or more models overlap in three dimensional space, the model listed first within the tiling order will satisfy requests within that overlapping zone.

Once the models have been tiled in this manner, the API or program accepts one or more input query points from the user. For every point, the framework queries each component model of the two meta models, until either a valid set of velocities and density are returned, or all component models have been queried and the request was unsatisfied. Thus, for points that fall within the interpolation zone, two sets of material properties may be generated - one each for each meta model. These two sets of material properties are then combined using the interpolation function. As the native query interfaces of available CVMs accept query points in a wide array of formats (geographic coordinates versus UTM-11 map coordinates, or depth versus elevation for the vertical component, for example), UCVM may perform a coordinate transformation to convert its input format of decimal (latitude, longitude, depth/elevation) tuples to the local coordinate system of the component model being queried. This is accomplished transparently by utilizing the standard projection library Proj.4 (NOTE: cite Proj.4).

In the case of the API, the result of a successful query is a data-structure containing the velocities V_p and Vs, and the density ρ at the point of interest, along with the elevation in meters and V_{S30} of the corresponding point on the free-surface, and an indication of which velocity model within the meta-model ultimately satisfied the request. For those points which fall within the interpolation zone between the meta-GTL and meta-model, the framework additionally identifies the material properties reported by the meta-GTL and meta-model, as well as the component models from which they were extracted. For the `ucvm_query` program, this same information is formulated in tabular format and printed to the screen.

This tiling mechanism can be a powerful feature for combining multiple regional velocity models, but a problem arises when velocity models overlap in three-dimensional space. Simply tiling two overlapping models will create a three-dimensional interface that may contain sharp contrasts in either velocities or density. This artifact is undesirable for earthquake simulation applications as it can cause reflections in wavefronts. There are two approaches to remedy this problem. The simplest approach is to define a UCVM patch model to trilinearly interpolate the material properties within arbitrary cuboid geographic region. This patch model may be tiled along with the overlapping traditional velocity models to produce a smoothed meta-model. However, this numerical smoothing approach does not reflect the physical structures of the Earth's surface. The second approach is to utilize UCVM to query all models within the overlap zone individually, and then manually combine the results with a user-defined interpolation function.

4.2. Creating Structured 3D Meshes

The framework may be utilized to generate a structured uniform mesh in a format consistent with those used by the AWP-ODC and SORD (NOTE: need to confirm if SORD still uses this format) simulation codes with the program `ucvm2mesh`.

Construction of mesh proceeds as shown in Figure 4. The user specifies a two-dimensional map projection (such as UTM-11), a latitude and longitude geographic anchor point, mesh cell dimensions (nx, ny, nz) along the x,y and z axis (where x-y is in the plane of the projection and z is vertical), step size dx in meters, and rotation angle within the map projection. Additionally, the user provides a list of CVMs to query, and these models are tiled in the same manner as was described in the previous section.

The program `ucvm2mesh` projects the geographic coordinates of the Earth's surface into the map projection, placing the origin of the mesh as the anchor point and discretizing the volume according to the provided dimensions and step size.

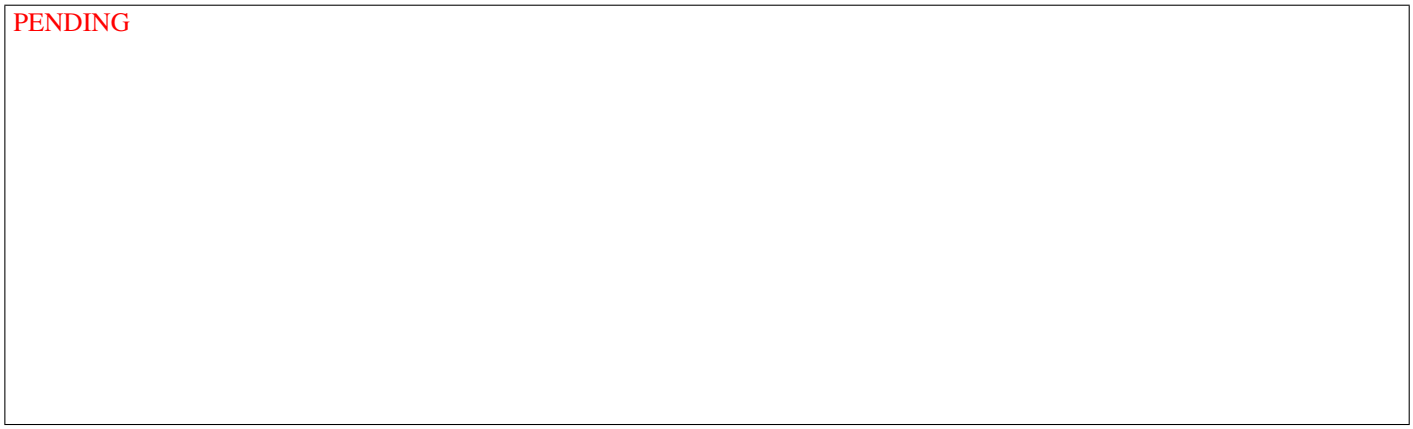

Figure 4: Construction of a mesh with the program ucvm2mesh.

For each point in the projected volume, `ucvm2mesh` determines the analous geographic point in terms of (latitude, longitude, depth) and queries the underlying CVMs for material properties. These material properties are then assigned to the mesh point.

Four binary mesh formats are supported: IJK-12, IJK-20, IJK-32, and SORD. The first three formats are mesh variants used by AWP-ODC (NOTE: cite) and the last format is used by SORD (NOTE: cite and check if still true). For IJK-12 formatted meshes, the output is a binary file consisting of a list of nx*ny*nz tuples. Each tuple contains three single precision floating point values (V_p.V_s,ρ), representing the material properties for a point in the mesh, with the mesh coordinates given implicitly by the position of the tuple in the list. The tuples are arranged in x-y-z order.

Similarly, the IJK-20 format adds the seismological parameters μ and quality factor Q so that each tuple contains (V_p.V_s,ρ,mu, Q) (NOTE: confirm and explain mu and Q), while the IJK-32 format extends IJK-20 further by adding the three four-byte integers (i, j, k) to each tuple, representing the index coordinate of the grid cell within the mesh. Thus, in this latter case, each tuple contains (i, j, k, V_p.V_s,ρ,mu, Q).

(NOTE: Explain SORD format)

(NOTE: Note ucvm2mesh-mpi exists, I don't think there are any operational differences)

4.3. Creating Etree Databases

TBD, reference Figure 5

4.4. Miscellaneous Features

TBD

4.4.1. Querying Iso-surfaces

TBD

4.4.2. Small-scale Heterogeneities

The last decade has seem a tremendous growth on high performance computing and applications dedicated to earthquake ground motion simulations which is leading the charge toward performing deterministic simulations at high frequencies of engineering interest ($f_{\max}$ 0–10 Hz). With increasing simulation frequencies, seismic velocity models must not only be accurate representations of a region's crustal structure at the geologic scale and represent the geotechnical characteristics of basins and deposits, but should also provide for the adequate representation of the scattering characteristics of the typical heterogeneities observed in geomaterials. To this end, UCVM implements an algorithm generates models that incorporate small-scale heterogeneities to any underlying velocity model. The command `ssh_generate` generates a binary float file of heterogeneities on a regular grid space with a normalized amplitude. Then, the command `ssh_merge` adds the heterogeneities multiplied by some scaling factor to the model. The introduction of the heterogeneities is actually done by applying the scaled perturbation to the slowness of the underlying mesh velocities, and then converted back to the seismic velocity. The following example generates and merges a set of small scale heterogeneities to ... describe example.

4.4.3. Etree Optimization

TBD `ecoalasce`, etc

4.4.4. Visualization

This section briefly describes a collection of utilities provided with the UCVM platform. These utilities are written in Python scripts that operate as wrappers around the basic single-core commands explained above, and work to produce specific output data commonly used for plotting. Because of brevity, we cannot fully describe here all the options and parameters that each of these commands can take as input and will only illustrate their operability. Additional details can be found at the UCVM User Guide (see Table 1).

5. Examples

TBD

PENDING

Figure 5: Construction of an Etree database with ucvm2etree.

5.0.5. *The* `ucvm_query` *command*

Here is an example of how to run this command.

```
> ./ucvm_query -f $UCVM_DIR/conf/ucvm.conf -m cvms
```

Note that the location of the package configuration file needs to be passed to the command using the flag "`-f`" and that the model to be used is set with the "`-m`" flag.

5.0.6. *The* `ucvm2mesh` *command*

This command generates a mesh or grid in either IJK-12, IJK-20, IJK-32, or SORD formats. The formats IJK-12, IJK-20, IJK-32, for instance, are used in discrete finite difference models for wave propagation simulations by the code AWP-ODC (?) (Cui et al., 2010). SORD, on the other hand, is a finite element code used to simulate dynamic rupture processes. The `ucvm2mesh` takes as input a configuration file that is passed on as an argument using the flag "`-f`". This file specifies the model to be used, the region size, and the resolution of the mesh or grid. The following example shows how to execute this command.

```
> ./ucvm2mesh -f ./ucvm2mesh_example.conf
```

5.0.7. *The* `ucvm2etree` *command*

The command `ucvm2etree` builds a materialized model in the form of an etree database, which uses an octree unstructured mesh-like format that adjust octant sizes to a specific resolution based on the material properties in the CVM being used at every point in a given volume. Resulting etrees built using this command are stand-alone discrete representations of a given velocity mode, and can be used as input models for other operations using UCVM. They can also be queried separately using APIs distributed with the etree library. In wave propagation problems, etrees are used by the code Hercules (**??**) in earthquake ground motion simulations. As in the previous case, this command takes an argument input configuration file. The following example shows how to execute this command.

```
> ./ucvm2etree -f ./ucvm2etree_example.conf
```

Note that `ucvm2etree` is the basic serial version of other parallel (MPI) commands used to build large materialized models in high performance computer systems. Parallel tools for building etrees are useful because regional size simulations require models that can be of the order of hundreds of gigabytes to terabytes and can take significant time and resources to build. Thus, for large etrees, we strongly recommend using the `ucvm2etree-extract-MPI`, `ucvm2etree-sort-MPI`, and `ucvm2etree-merge-MPI` commands explained below.

5.0.8. *The* `horizontal_slice.py` *command*

TBD

6. Recent UCVM Applications

This section showcases research efforts where UCVM has been used to facilitate the creation of materialized velocity models, that is, numerical discrete models based on velocity models that can be used in simulations and research in geosciences.

6.1. *The 2008 Chino Hills Case Study*

On 29 July 2008 the region of southern California and in particular, the greater Los Angeles metropolitan area, was struck by a magnitude M_w 5.4 earthquake at 11:42 a.m. The earthquake was the strongest felt in the city since the 1994 Northridge earthquake, but caused no significant damages or fatalities. It provided, nonetheless, an excellent opportunity for earthquake studies because its shaking was recorded on a numerous set of monitoring stations. Since then, UCVM and other SCEC initiatives have used the data during this earthquake as a case study for testing different simulation methods and models. Two particular efforts involving the use of the UCVM platform are noteworthy and serve here as application examples of how UCVM is being used to advance research in geoscience.

6.1.1. *Validation studies using different velocity models*

Various recent studies done have carefully conducted validations of simulations of the 2008 Chino Hills earthquake with models created using UCVM (e.g., Olsen and Mayhew, 2010;

Taborda and Bielak, 2013, 2014). They have helped test the simulation capabilities of current modeling approaches to predict the ground motion of moderate magnitude earthquakes at low (< 1 Hz) and moderately (1–4 Hz) and high (> 4 Hz) frequencies using deterministic and hybrid methods, as well as to propose and test different validation algorithms (goodness-of-fit criteria).

The latest of these studies (Taborda and Bielak, 2014), in particular, focuses on the validation of simulations of the Chino Hills earthquake using different velocity models, namely CVM-S and CVM-H. This in turn have helped evaluate the differences between the models and their accuracy. The validation performed in this study is based on comparisons with recorded seismograms in over 300 stations scattered throughout the region. The simulation and the models used created with UCVM cover an area of 180 km × 135 km and go as deep as 62 km. Figure 6 shows comparisons between the crustal structures of the two models, results of the surface peak ground velocity obtained from the simulations and contour maps of the validation results. The models created for these simulations were built using the etree MPI utilities on NICS's Kraken and NCSA's Blue Waters supercomputer systems employing between XX and YY thousand cores for up to ZZ hours. The sizes of the output (etree) files range between XXX and XXX GB and they comprised between 5 billion and 15 billion octants. (This paragraph needs to be improved.)

Do the logfiles still exist? Can provide runtimes, resources used (core hours, wallclock), and where it was run.

6.1.2. *Effects of small-scale heterogeneities in simulation*

I know the San Diego group did a series of simulations for Chino Hills in which they used SSH. The caveat, though, is that they did those prior to SSH being implemented in UCVM. Is there an alternative? If we have results using SSH as implemented in UCVM that are not Chino Hills, then we can simply move this up one level and present the Chino Hills results only from Hercules.

6.2. *CyberShake*

Desirable. I know there are results—preliminary, perhaps—if CyberShake using the different velocity models, and I presume the models used in the simulation where done using UCVM. Who would be the right person to help us prepare a text and a figure for this sub-section?

Scott can describe exactly how UCVM is used in the Cybershake mesher. If he fills in this section, we need to add him as a co-author.

6.3. *Something involving inversion and CVM-S4.26?*

Tentative. I ignore the level of importance UCVM had on preparing the models for Po and En-Jui's inversions in preparation to obtain CVM-S4.26. I suppose that in any case, the fact that CVM-S-4.26 was produced and included in UCVM is something worth showcasing here. i will need, however, to think harder about how to put it in in a nice way.

6.4. *Something involving the BBP?*

Tentative. Is UCVM used at all in the BBP?

At the time I developed UCVM, it wasn't used in BPP. Has that changed?

7. Final Remarks

Pending.

References

Aagaard, B.T., Brocher, T.M., Dolenc, D., Dreger, D., Graves, R.W., Harmsen, S., Hartzell, S., Larsen, S., McCandless, K., Nilsson, S., Petersson, N.A., Rodgers, A., Sjogreen, B., Zoback, M.L., 2008. Ground-motion modeling of the 1906 San Francisco earthquake, Part II: Ground-motion estimates for the 1906 earthquake and scenario events 98, 1012–1046. URL: http://www.bssaonline.org/cgi/content/abstract/98/2/1012.

Bielak, J., Graves, R.W., Olsen, K.B., Taborda, R., Ramírez-Guzmán, L., Day, S.M., Ely, G.P., Roten, D., Jordan, T.H., Maechling, P.J., Urbanic, J., Cui, Y., Juve, G., 2010. The ShakeOut earthquake scenario: Verification of three simulation sets 180, 375–404. URL: http://dx.doi.org/10.1111/j.1365-246X.2009.04417.x, doi:10.1111/j.1365-246X.2009.04417.x.

Brocher, T.M., 2005. Compressional and shear wave velocity versus depth in the San Francisco Bay Area, California: Rules for USGS Bay Area Velocity Model 05.0.0. Technical Report OFR-2005-1317. U.S. Geological Survey. URL: http://pubs.usgs.gov/of/2005/1317/.

Brocher, T.M., 2008. Compressional and shear-wave velocity versus depth relations for common rock types in northern California 98, 950–968. URL: http://www.bssaonline.org/cgi/content/abstract/98/2/950, doi:10.1785/0120060403.

Brocher, T.M., Aagaard, B.T., Simpson, R.W., Jachens, R.C., 2006. The USGS 3D seismic velocity model for northern California 87, Fall Meet. Suppl., Abstr. S51B–1266.

Cui, Y., Olsen, K., Jordan, T., Lee, K., Zhou, J., Small, P., Roten, D., Ely, G., Panda, D., Chourasia, A., Levesque, J., Day, S., Maechling, P., 2010. Scalable earthquake simulation on petascale supercomputers, in: SC '10: Proc. of the 2010 ACM/IEEE Int. Conf. for High Performance Computing, Networking, Storage and Analysis, pp. 1–20. doi:10.1109/SC.2010.45.

Ely, G.P., Jordan, T.H., Small, P., Maechling, P.J., 2010. A Vs30-derived near-surface seismic velocity model, in: Abstr. AGU Fall Meet., San Francisco, California, December 13–17. URL: http://web.alcf.anl.gov/~gely/pub/Ely2010-AGU-Vs30-GTL.pdf.

Frankel, A., Vidale, J., 1992. A three-dimensional simulation of seismic waves in the Santa Clara Valley, California, from a Loma Prieta aftershock 82, 2045–2074. URL: http://www.bssaonline.org/cgi/content/abstract/82/5/2045.

Graves, R., Jordan, T., Callaghan, S., Deelman, E., Field, E., Juve, G., Kesselman, C., Maechling, P., Mehta, G., Milner, K., Okaya, D., Small, P., Vahi, K., 2011. CyberShake: A physics-based seismic hazard model for Southern California 168, 367–381. doi:10.1007/s00024-010-0161-6.

Graves, R.W., 2008. The seismic response of the San Bernardino basin region during the 2001 Big Bear lake earthquake 98, 241–252. URL: http://www.bssaonline.org/cgi/content/abstract/98/1/241, doi:10.1785/0120070013.

Kohler, M.D., Magistrale, H., Clayton, R.W., 2003. Mantle heterogeneities and the SCEC reference three-dimensional seismic velocity model version 3 93, 757–774. URL: http://www.bssaonline.org/cgi/content/abstract/93/2/757, doi:10.1785/0120020017.

Magistrale, H., Day, S., Clayton, R.W., Graves, R., 2000. The SCEC southern California reference three-dimensional seismic velocity model version 2 90, S65–S76. URL: http://www.bssaonline.org/cgi/content/abstract/90/6B/S65, doi:10.1785/0120000510.

Magistrale, H., McLaughlin, K., Day, S., 1996. A geology-based 3D velocity model of the Los Angeles basin sediments 86, 1161–1166. URL: http://www.bssaonline.org/cgi/content/abstract/86/4/1161.

Magistrale, H., Olsen, K.B., Pechmann, J.C., 2006. Construction and Verification of a Wasatch Front Community Velocity Model. Technical Report 06HQGR0012. U.S. Geological Survey. URL: http://earthquake.usgs.gov/research/external/reports/06HQGR0012.pdf.

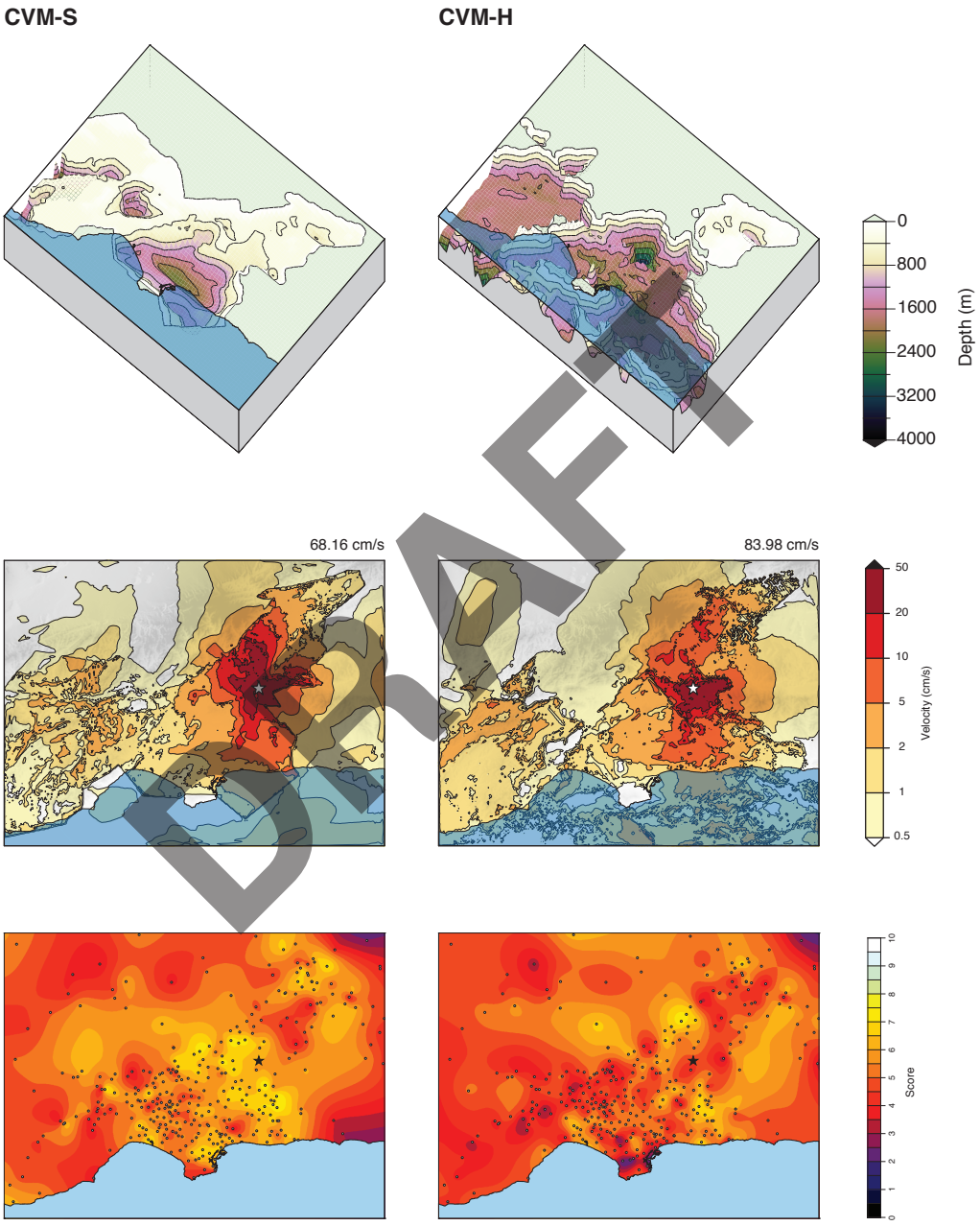

Figure 6: Temporary mock-up figure showing results from Chino Hills validation.

Olsen, K.B., Day, S.M., Dalaguer, L.A., Mayhew, J., Cui, Y., Zhu, J., Cruz-Atienza, V.M., Roten, D., Maechling, P., Jordan, T.H., Okaya, D., Chourasia, A., 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.

Olsen, K.B., Mayhew, J.E., 2010. Goodness-of-fit criteria for broadband synthetic seismograms, with application to the 2008 M_w 5.4 Chino Hills, California, earthquake 81, 715–723. doi:10.1785/gssrl.81.5.715.

Plesch, A., Tape, C., Graves, R., Shaw, J., Small, P., Ely, G., 2011a. Updates for the CVM-H including new representations of the offshore Santa Maria and San Bernardino basin and a new Moho surface, in: Proc. SCEC Annu. Meet.

Plesch, A., Tape, C., Shaw, J., Small, P., Ely, G., Jordan, T., 2011b. User Guide for the Southern California Earthquake Center Community Velocity Model: SCEC CVM-H 11.9.0. Harvard University and University of Southern California. URL: http://scec.usc.edu/scecwiki/images/5/51/Cvmh_manual.pdf.

Ramírez-Guzmán, L., Boyd, O.S., Hartzell, S., Williams, R.A., 2012. Seismic velocity model of the Central United States (Version 1): Description and simulation of the 18 April 2008 Mt. Carmel, Illinois, earthquake 102, 2622–2645. doi:10.1785/0120110303.

Taborda, R., Bielak, J., 2013. Ground-motion simulation and validation of the 2008 Chino Hills, California, earthquake 103, 131–156. doi:10.1785/0120110325.

Taborda, R., Bielak, J., 2014. Ground-motion simulation and validation of the 2008 Chino Hills, California, earthquake using different velocity models 104, in press.

Tu, T., López, J., O'Hallaron, D., 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. URL: http://www.cs.cmu.edu/~euclid/.

Wald, D.J., Allen, T.I., 2007. Topographic slope as a proxy for seismic site conditions and amplification 97, 1379–1395. doi:10.1785/0120060267.

Wills, C.J., Clahan, K.B., 2006. Developing a map of geologically defined site-condition categories for California 96, 1483–1501. doi:10.1785/0120050179.

Yong, A., Hough, S., Iwahashi, J., Braverman, A., 2012. (2012), a terrain-based site conditions map of california with implications for the contiguous united states 102, 114–128. doi:10.1785/0120100262.

Table 2: Question marks indicate fields that need to be completed. This list needs to be checked carefully. List of velocity models currently supported by the UCVM platform.

Model Name	Region	String Label	Installation	Coverage Coordinates	References
SCEC CVM-H	Southern California	cvmh	Automated	-120.8620, 30.9565 -113.3329, 30.9565 -113.3329, 36.6129 -120.8620, 36.6129	Plesch et al. (2011a) Plesch et al. (2011b) ?
SCEC CVM-S	Southern California	cvms	Automated	?	Magistrale et al. (1996) Magistrale et al. (2000) Kohler et al. (2003)
SCEC CVM-S4.26	Southern California	cvmsi	Automated	-116.0000, 30.4499 -122.3000, 34.7835 -118.9475, 38.3035 -112.5182, 33.7819	?
SCEC CVM-S5	Southern California	cvms5	Automated	-116.0000, 30.4499 -122.3000, 34.7835 -118.9475, 38.3035 -112.5182, 33.7819	?
Hadley-Kanamori 1D	Global	1d	Automated	Global	?
Carl Tape SoCal	Southern California	tape	Manual	-121.8549, 36.7301 -121.5939, 32.1919 -114.8286, 32.2616 -114.7137, 36.8027	?
Broadband 1D	Whittier Narrows	bbp1d	Automated	Global	?
Graves	Cape Mendocino	cmrg	Manual	?	?
USGS CenCalVM	Central California	cencal	Automated	-126.3532, 39.6806 -123.2732, 41.4849 -118.9445, 36.7022 -121.9309, 35.0090	Brocher (2005) Brocher et al. (2006)
SCEC CVM-NCI	?	cvmnci	Manual	?	?
Lin-Thurber	California Statewide	lt	Manual	-126.9210, 39.8816 -121.4117, 43.0597 -112.8281, 33.4362 -118.1781, 30.2581	?
USGS WFCVM	Wasatch Front, Utah	wfcvm	Manual	?	Magistrale et al. (2006)
Ely GTL	Southern California	elygtl	Automated	-122.5410, 30.7046 -112.9958, 30.7046 -112.9958, 38.5460 -122.5410, 38.5460	Ely et al. (2010)
1D GTL	Southern California	1dgtl	Automated	Global	?
V_{S30} Maps	Southern California	ucvm, yong		?	Wills and Clahan (2006) Wald and Allen (2007) Yong et al. (2012)