Searching for 3D wave-propagation effects in southern California

Geoffrey Ely, gely@usc.edu,
Thomas Jordan
University of Southern California

Abstract

Ground motions in southern California can be significantly influenced by three-dimensional basin wave-propagation effects. Simulations of large events on the southern San Andreas within the SCEC Community Velocity Model (CVM) find large amplifications at Whittier Narrows due to basin-guided waves. We have begun a study to search for similar basin effects in southern California, using a series 3D wave-propagation simulations over a suite of potential source scenarios. Selection of scenarios is being guided by the Uniform California Earthquake Rupture Forecast, as well as results from the CyberShake project. CyberShake attempts to calculate an exhaustive set of seismograms at specified sites using a 3D model, which provides a database that can be interrogated to identify sources with high basin-wave excitations at those sites. Simulations will be performed up to 0.5 Hz using the Support Operator Rupture Dynamics (SORD) code and kinematic finite-fault sources, with fault geometry (often non-planar) from the SCEC Community Fault Model. Waves will be propagated through the SCEC-CVM as well as the SCEC-CVMH (Harvard version) with comparisons helping to address uncertainty in the basin effect results. Selected scenarios will be recomputed with dynamic rupture sources.

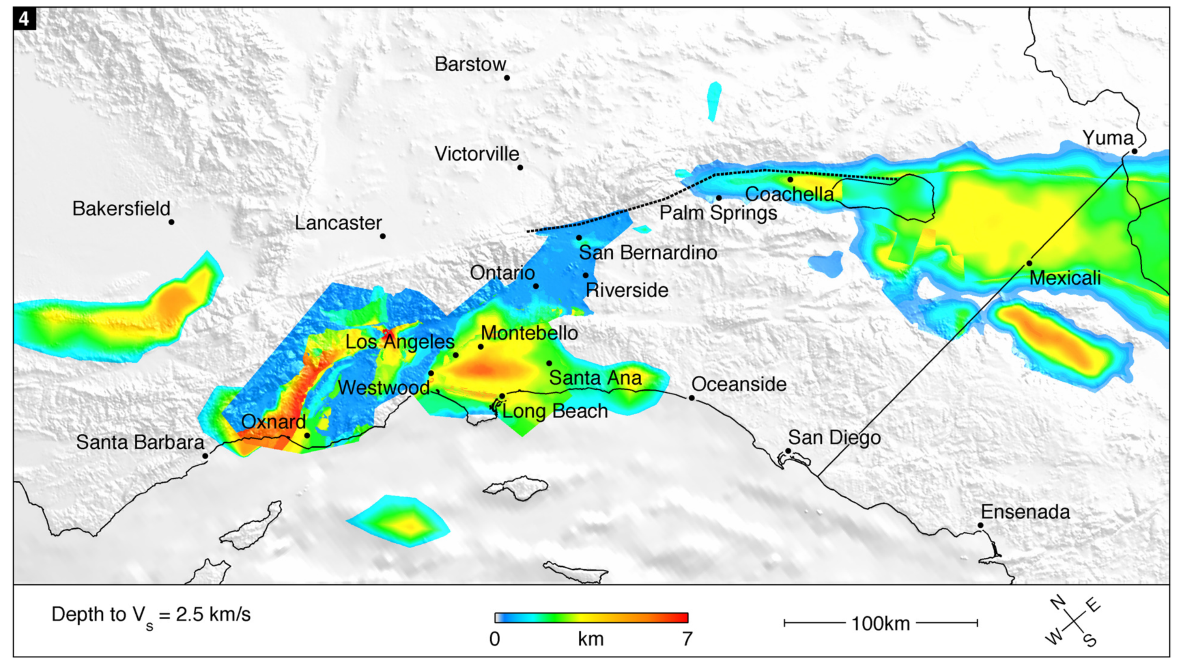


Figure 1. Sedimentary basin depth from the SCEC-CVM version 4.0 as defined by the depth to the 2.5 km/s S-wave velocity horizon.

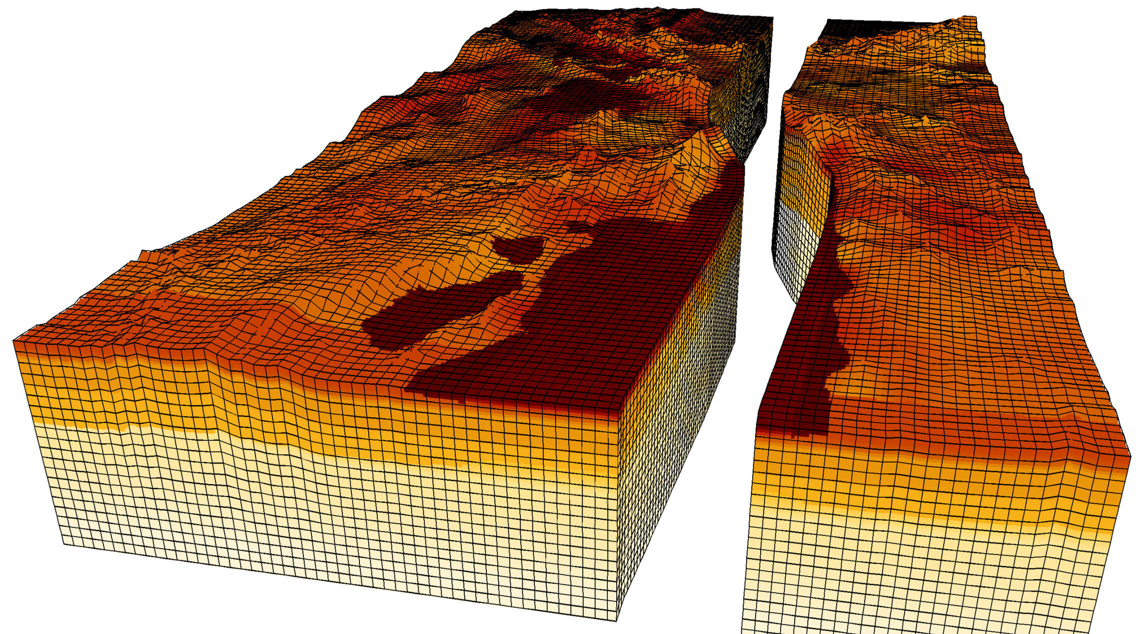


Figure 2. Computational mesh (1.8 billion nodes) used for simulations on the southern San Andreas fault. Fault geometry is vertical planar segments. Color scale indicates S-wave velocity from the SCEC-CVM version 4.0 (red=slow, yellow=fast).

Project Status

Dynamic rupture simulations have been completed for the southern San Andreas fault, up to 0.25 Hz using a simplified vertical planar segmented geometry. Current efforts are focused on two fronts:

1. Improved fault surface meshing using non-planar, and dipping fault surfaces from the SCEC Community Fault Model using CUBIT software from Sandia National Laboratory.
2. Improved I/O performance of the SORD code via buffering and computation overlap. Though SORD has demonstrated excellent computational scaling (Fig. 3), I/O bottlenecks necessitate further optimization.

Following the technical improvements, our initial focus is on 0.5 Hz simulations for the San Jacinto and Elsinore faults. For computation we are targeting 12,000 cores on the Ranger system at the Texas Advanced Computing Center.

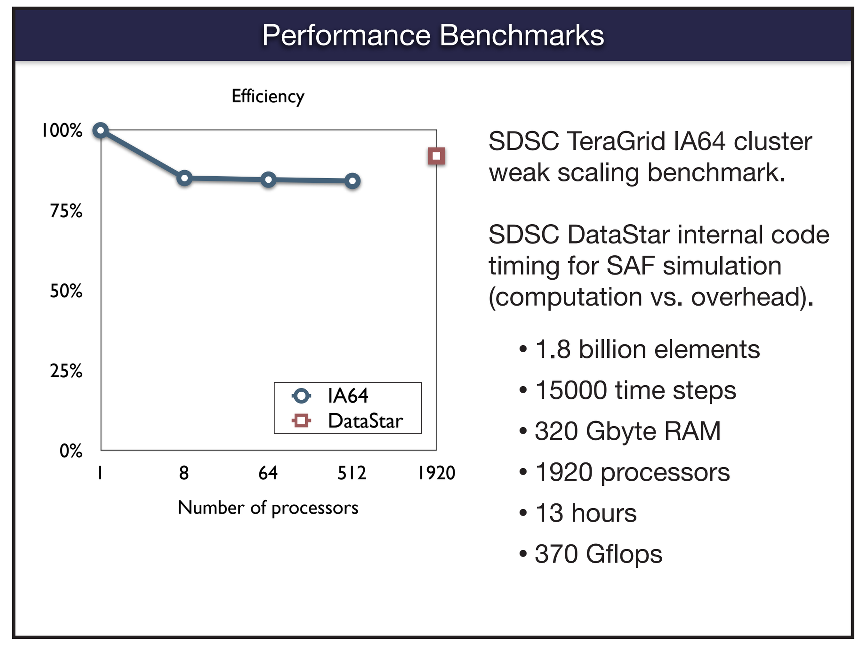


Figure 3. SORD benchmarks.

WebSims: A Python-based web application for storing, exploring and disseminating 4D earthquake simulation data

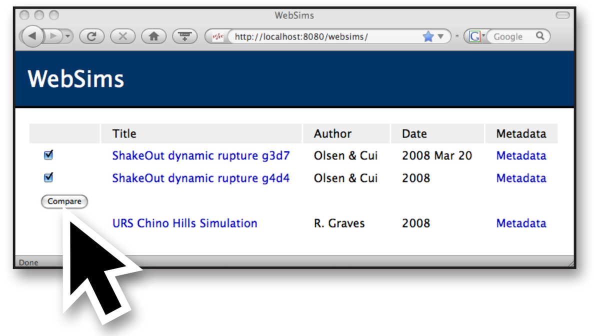
Geoffrey Ely, gely@usc.edu,
University of Southern California
Kim Olsen
San Diego State University

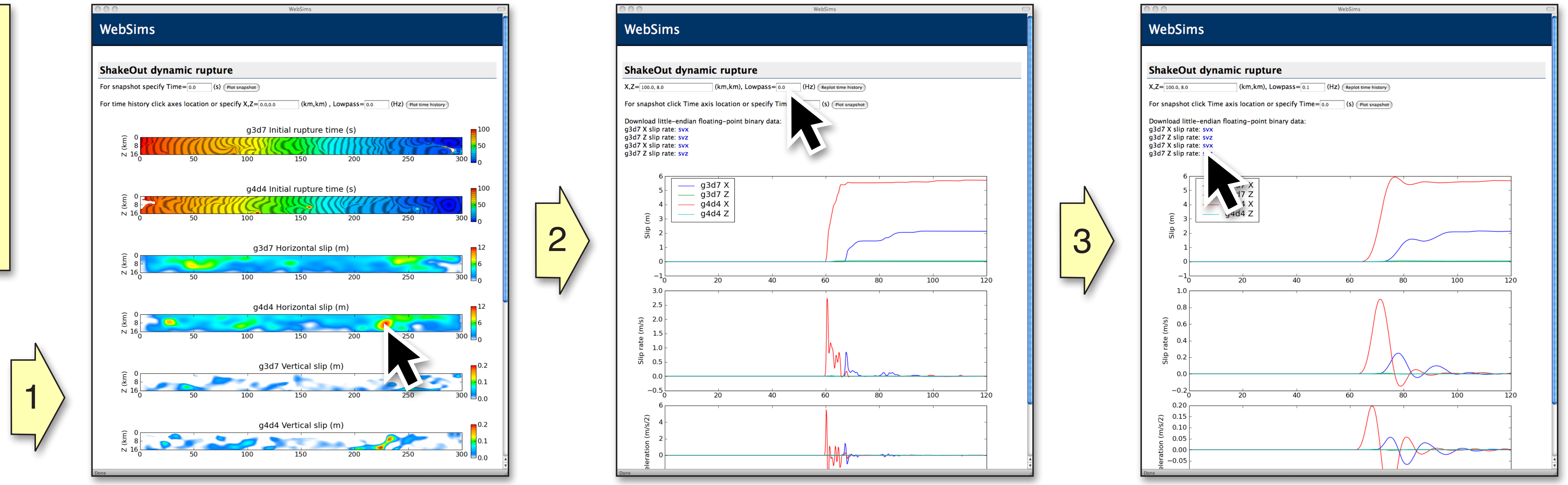
Abstract

WebSims aims to provide a tool for cataloging, exploring, comparing and sharing four-dimensional results of large numerical earthquake simulations. Users may extract time histories or two-dimensional slices via a clickable interface or by specifying precise coordinates. Extractions are plotted to the screen and optionally downloaded to local disk. Time histories may be low-pass filtered, and multiple simulations may be overlayed for comparison. Metadata is stored with each simulation in the form of a Pylab module. A well defined URL scheme for specifying extractions allows the web interface to be bypassed, thus allowing for batch scripting of both plotting and download tasks. This version of WebSims replaces a previous PHP implementation. It is written in Python using the NumPy, SciPy, and Matplotlib modules, which provide a MATLAB-like processing and visualization environment. The web pages are served by a web.py, a simple web application framework similar to Google App Engine. WebSims is open source and easy to customize, though not supported.

Example 1: Olsen ShakeOut dynamic rupture comparison.

1. Choose simulations to compare.
2. From fault surface display, click axes location for time history.
3. From time history display, specify filter.
4. Download time history data.





Example 2: Graves Chino Hills simulation.

1. Choose simulation.
2. From surface display, click axes location for time history.
3. From time history display, click time axis location for surface snapshot.
4. Repeat steps 2 and 3 indefinitely...