Virtual California Earthquake Simulator

by Michael K. Sachs, Eric M. Heien, Donald L. Turcotte, M. Burak Yikilmaz, John B. Rundle, and Louise H. Kellogg

INTRODUCTION

The model that would eventually become Virtual California started as a limited simulation model developed by Rundle (1988) for the distributed seismicity on the San Andreas and adjacent faults in southern California. This model included stress accumulation and release as well as stress interactions between faults. An updated version of this model was developed (Rundle *et al.* 2001, 2002, 2004), including the major strike-slip faults in California, and was called Virtual California. Details of the model were given in Rundle *et al.* (2006a,b). Yakovlev *et al.* (2006) utilized Virtual California simulations to examine the recurrence time statistics on faults in California. They concluded that the distribution of return times on a fault is well approximated by a Weibull distribution. Yikilmaz *et al.* (2010) used Virtual California to simulate earthquakes on the Nankai trough, Japan, and found an excellent agreement with the historical sequence of 13 great earthquakes. Yikilmaz *et al.* (2011) gave a composite simulation of seismicity in northern California utilizing Virtual California for earthquakes on mapped faults, a random background of smaller earthquakes, and a branching aftershock sequence (BASS) model simulation of aftershocks. In its current incarnation, Virtual California is a sophisticated tool for simulating earthquakes on a wide variety of fault geometries in a high-performance computing environment and is part of a larger effort by the Southern California Earthquake Center to unify the results from several different earthquake simulators. A description of this effort and a comparison of the results can be found in Tullis *et al.* (2012a) and Tullis *et al.* (2012b), respectively.

COMPONENTS OF VIRTUAL CALIFORNIA

There are three major components that make up Virtual California: a fault model, a set of quasi-static interactions (Green's functions), and an event model. In spite of the name, the only component of Virtual California that is specific to California is the fault model. This model can be changed to any physically realistic model and still correctly work with the simulation physics and event model.

Fault Model

As already mentioned, the fault model is the only component of Virtual California that is specific to California. The model that is currently in use is based on the Uniform California Earthquake Rupture Forecast, version 2, fault model (UCERF2; Field *et al.* , 2007) developed by the Working Group on California Earthquake Probabilities. The model includes 181 fault sections roughly corresponding to known faults in California, with some faults modeled by multiple sections. Each fault section is meshed into square elements that are roughly 3 km × 3 km, for a total of 14,474 elements. In the present version of our model, the creeping section of the San Andreas fault is removed (Fig. 1). This section produces many events, slowing the simulation down considerably. Each element in the model is given a constant back-slip velocity along a fixed rake vector and a failure stress. The rake vector always lies in the plane of the element; and, the angle of the vector relative to the surface of the Earth, along with the rate of slip, are quantities that are taken from the UCERF2 model. The failure stresses, which are also required for the model, are derived from paleoseismic event recurrence times.

Element Stress Interactions

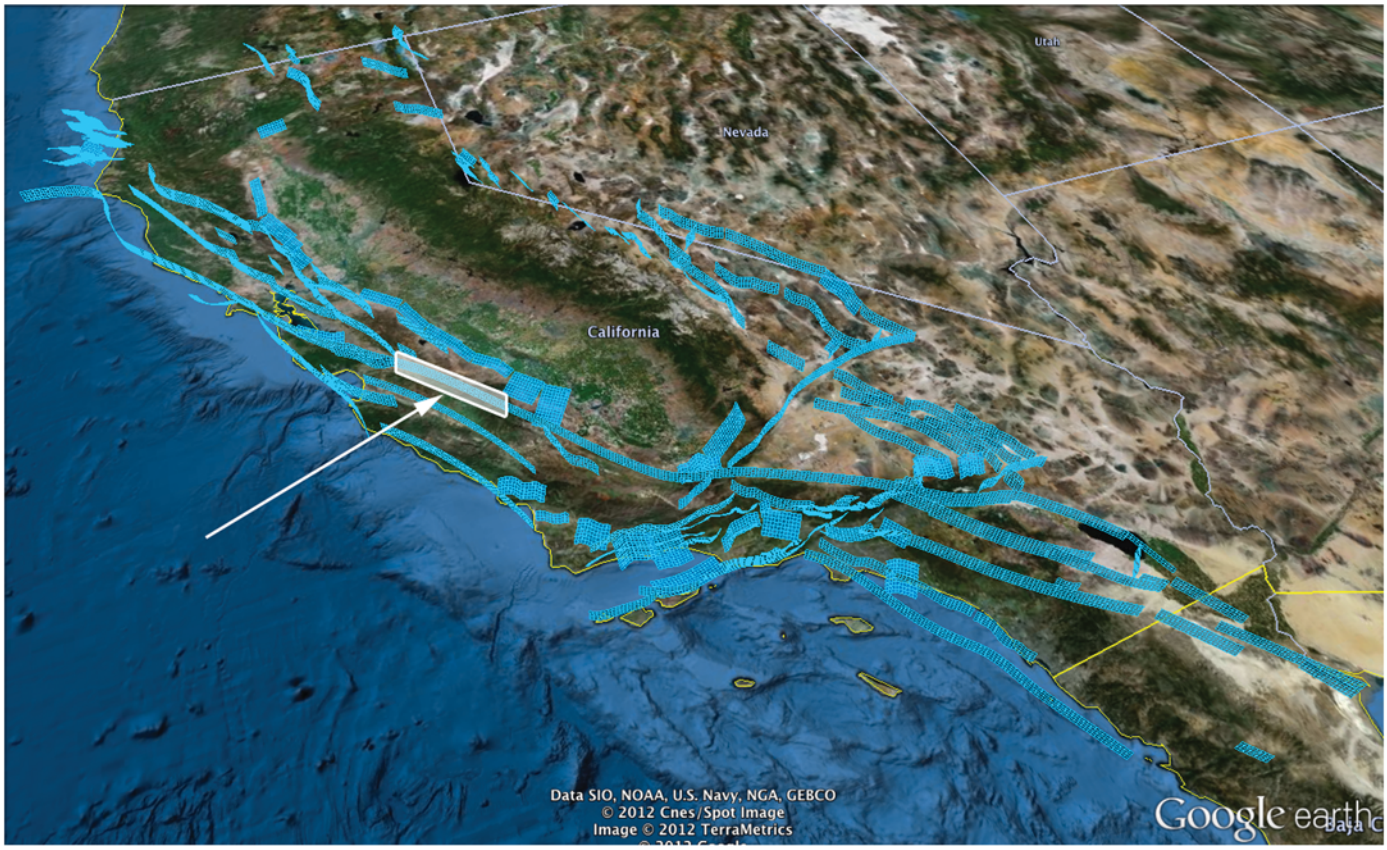
Unlike actual fault systems where the fault geometry is dynamic, the fault geometry used by Virtual California is static. This static geometry allows Virtual California to avoid the considerable complexities involved in modeling evolving fault systems. In this sense Virtual California is designed to explore seismicity in fault systems as they exist now. Because the faults don't actually move, back slip is used to model the effects of stress buildup and release along the fault plane. In a back-slip model the equilibrium position for a failed element is the same as its initial position, so failed elements do not actually change their position.

The static, back-slip approach greatly simplifies how interactions between fault elements are calculated. In general, how back slip on one element effects stress on another element depends on the position and orientation of both elements. Because the fault geometry is static, these interactions, or stress Green's functions, only need to be calculated once. In order to calculate the stress Green's functions, every element is back-slipped by a unit distance along its slip velocity vector, and the changes of stress are calculated for every other element. The stress changes at any location $\mathbf{x}$ in the simulation due to changes on all other elements is given by (Rundle *et al.*, 2006a,b):

$$\sigma_{ij}(\mathbf{x}, t) = \int dx'_k T^{kl}_{ij}(\mathbf{x} - \mathbf{x}') s_l(\mathbf{x}', t), \quad (1)$$

where $s_l(\mathbf{x}', t)$ is the three-dimensional slip density of element l and $T^{kl}_{ij}(\mathbf{x} - \mathbf{x}')$ is the Green's function tensor. The Einstein summation convention is assumed. The indexes i, j, k, and l

doi: 10.1785/0220120052



▲ **Figure 1.** An example of a Virtual California fault model. This particular model is based on the Field *et al.* (2007) UCERF2 fault model. The section indicated in the image is the creeping section of the San Andreas fault. This section tends to generate a large number of events and was removed from the model in order to speed up the simulation.

run over the Cartesian coordinate axes, x, y, and z. In the case of Virtual California, the field is only evaluated at the centers of elements, and slip is uniform across the surface of an element and is allowed only along the element's rake angle, which is defined by the model. Under these conditions equation 1 simplifies to

$$\sigma_{ij}^{A}(t) = T_{ij}^{AB} s_B(t), \tag{2}$$

where A and B run over all elements. Lastly, because we are only interested in the shear stress along the rake vector and the normal stress perpendicular to the plane of the element, the six elements of the tensor T_{ij} reduce to T_s for the shear stresses and T_n for the normal stresses. The final stresses are determined by

$$\sigma_s^A(t) = T_s^{AB} s_B(t) \quad \text{and} \quad \sigma_n^A(t) = T_n^{AB} s_B(t). \tag{3}$$

Therefore, if there are N elements in a model, Virtual California needs two $N \times N$ matrices to govern all interactions.

The actual values of the Green's functions are calculated using an implementation of Okada's half-space deformations (Okada, 1992).

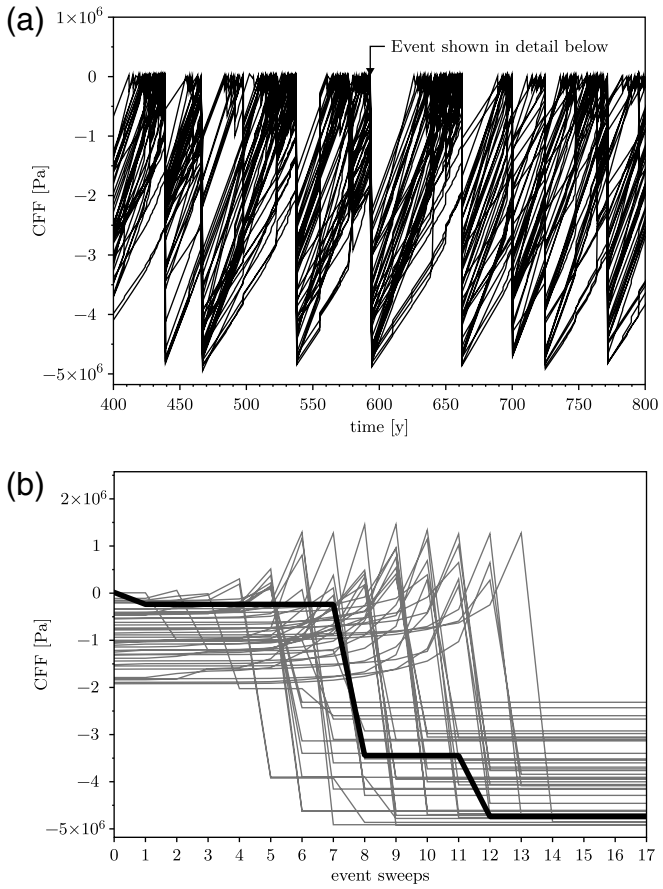
Rupture Event Model

Virtual California uses a static–dynamic friction law to determine when an element fails. This law is implemented by a Coulomb failure function (CFF):

$$\mathrm{CFF}^A(t) = \sigma_s^A(t) - \mu_s^A \sigma_n^A(t), \tag{4}$$

where μ_s^A is the static coefficient of friction calculated from the model element strengths. When $\mathrm{CFF}^A(t_f) = 0$, the element A fails. To simplify the notation, we will remove the element label A from the CFF function and just assume that it refers to a single element.

Elements in Virtual California gain and release stress through an event model that consists of two phases: a long-term slip phase and a rupture-propagation phase. The long-term slip phase models the time between earthquakes when stress builds up on the faults due to long-term plate movement. This involves applying back slip to all elements at their model-defined slip velocities. The long-term slip phase ends at time $t = t_f$, when one or more element's CFF becomes 0 ($\mathrm{CFF}(t_f) = 0$). Because the interactions in the Element Stress Interactions section are elastic, the relationship between slip and stress is known (equation 3), and it is not necessary to evolve the system step by step



▲ **Figure 2.** (a) The CFF for each of the 48 elements that make up the Parkfield section of the San Andreas fault. Drops in the CFF correspond to events. Large events are characterized by many elements undergoing large CFF drops. (b) The sweeps that make up the event at $t = 593.187$. The element that triggers the event is in bold. The initial failure triggers a cascade that results in all elements failing.

during this phase. Rather, the simulation time is directly advanced to the point at which the next element fails, and then the rupture propagation phase begins.

During the rupture propagation phase, the system releases accumulated stress through a cascading series of fault-element failures. Virtual California uses a cellular automata approach to modeling rupture propagation. Because of this, ruptures are not a dynamic process in the sense that there are no time-domain solutions to differential equations governing them. Instead, given an initial failure and stress state, the system iteratively slips elements to find the final stress state. Equations 5 and 6 (described below) are meant to approximate dynamical considerations and generate better output statistics.

When a rupture begins, the first element to fail is allowed to slip back toward its equilibrium position. The amount the element slips, Δs, is related to the stress drop defined for the element in the model, $\Delta\sigma$, by Rundle *et al.* (2006a,b):

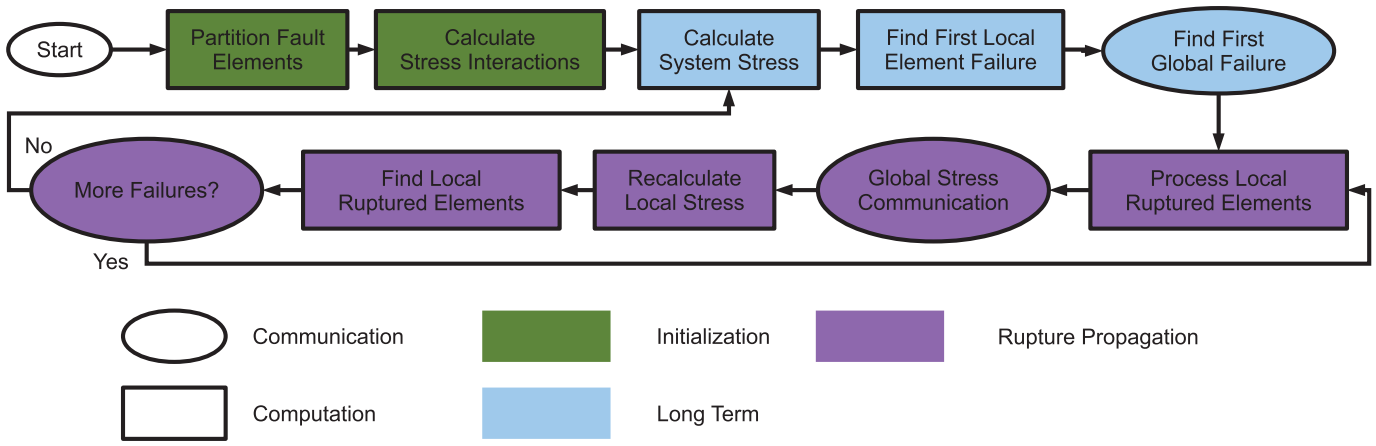
$$\Delta s = \begin{cases} \frac{1}{K_L}\frac{N_{ef}}{S_t}(\Delta\sigma - \mathrm{CFF}), & \text{if } N_{ef} \le S_t \\ \frac{1}{K_L}(\Delta\sigma - \mathrm{CFF}), & \text{otherwise} \end{cases}. \tag{5}$$

K_L is the element's stiffness or self-stress, defined (for element A) as $K_L = T_s^{AA} - \mu_s^A T_n^{AA}$. The factor N_{ef}/S_t is related to the current size of the rupture: N_{ef} is the number of failed elements on a particular fault, and S_t is the slip-scaling threshold. The slip-scaling threshold is set as an external parameter. This factor is used to prevent small ruptures from slipping too much.

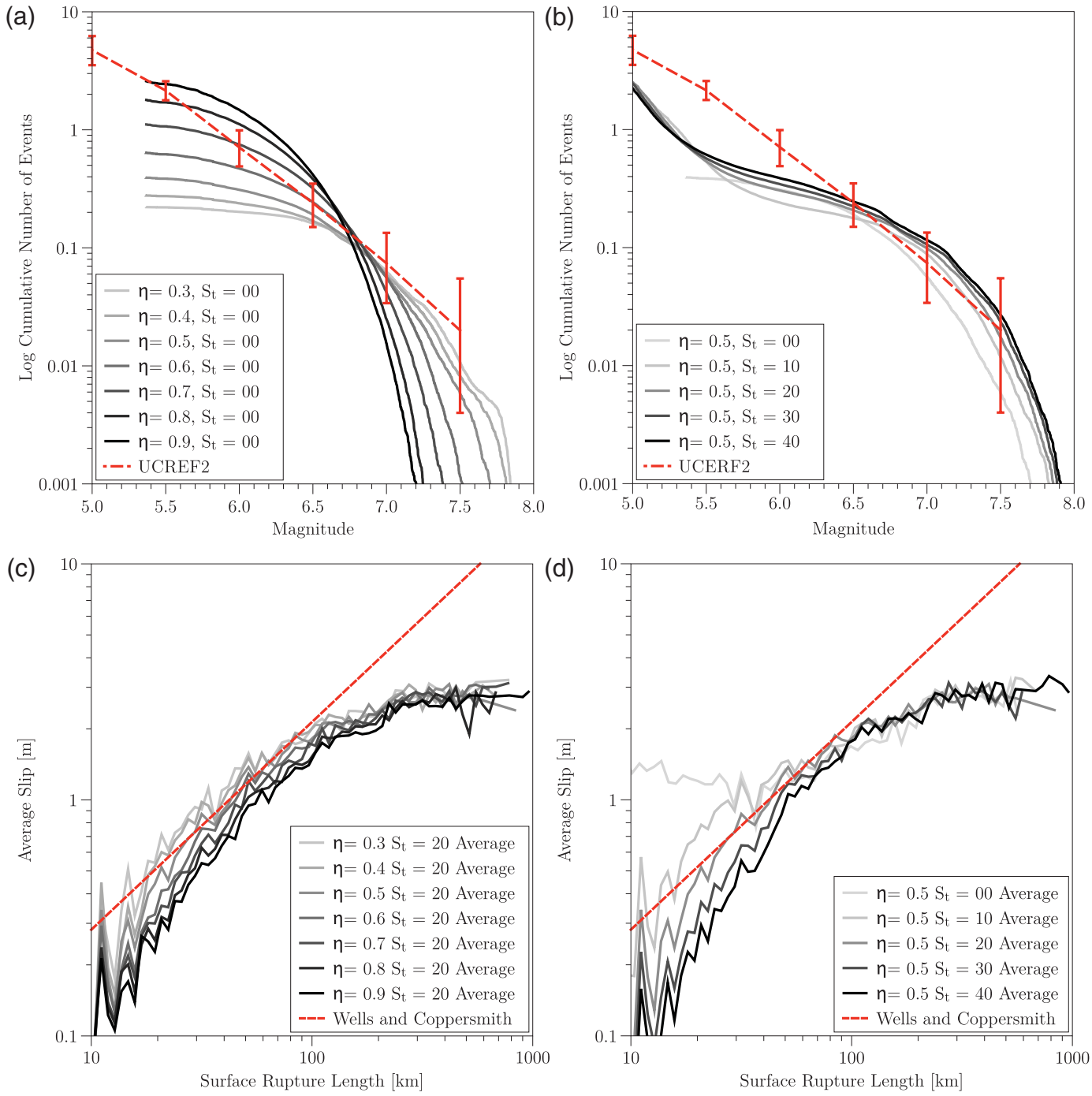
After the initial element slips, a new stress state is calculated for the entire system using equation 3. Additional elements will fail if their CFF = 0. In order to encourage rupture propagation, a dynamic triggering mechanism is used. Elements on the same fault and physically close to a failed element are allowed to fail at a lower stress than the defined failure stress, provided the amount of stress accumulated during the rupture is greater than a predefined dynamic triggering factor η:

$$\frac{\mathrm{CFF}_{\mathrm{init}} - \mathrm{CFF}_{\mathrm{final}}}{\mathrm{CFF}_{\mathrm{init}}} > \eta. \tag{6}$$

The dynamic triggering factor approximates the stress intensity factor at the tip of a propagating rupture. If failed elements have not slipped back to their equilibrium points due to their initial failures, they are allowed to fail again and release more stress



▲ **Figure 3.** Execution flow of Virtual California on a parallel system.



▲ **Figure 4.** The output from several Virtual California simulations with different values of the dynamic triggering ⍰ and the slip threshold S_t: (a) frequency–magnitude varying ⍰; (b) frequency–magnitude varying S_t; (c) average slip-surface rupture length varying ⍰; and (d) average slip-surface rupture length varying S_t. The dashed red line and error bars in (a) and (b) are the observed seismicity in California and the 95% confidence levels as reported by UCERF2 (Field *et al.* 2007). The dashed red lines in (c) and (d) are observed relationships reported by Wells and Coppersmith (1994).

into the system. They are not, however, allowed to slip away from their equilibrium point. This means that they will not absorb any stress released from newly failed elements. This behavior reflects the fact that during a rupture, failed elements are not allowed to heal or to release their accumulated stress all at once. This process continues until there are no more failures, at which point the event is over. It is important to note that elements may not release all of their accumulated stress during a rupture. This is partially due to the slip-scaling threshold (equation 5) and dynamic triggering (equation 6) but also partially

due to the overall stress state of the system when the rupture begins. If the system is in a high-stress state, then very small initial changes in stress can cause a cascading series of failures that result in the release of large amounts of stress. (This effect can be seen in Fig. 2b.) However, if the system is in a low-stress state, these small initial stress changes will not trigger additional failures.

Figure 2a shows an example of how stress is released and accumulated over multiple event cycles. Figure 2b shows how a single element failure can lead to a cascading series of failures in a single event; it also shows that elements can fail multiple times during an event but will not accumulate additional stress after they fail.

Simulation Computational Overview

Here we outline the computational flow of Virtual California. Figure 3 shows the path of execution in a parallel simulation running on multiple processors, either on a cluster or multicore machine. The execution is divided into three distinct phases: initialization, long-term stress interaction, and rupture propagation. A straightforward implementation of this simulation would be heavily bound by computation and communication, thus Virtual California uses techniques described in Heien and Sachs (2012) to make this more tractable.

The initialization phase begins by parsing the specified model and simulation parameters. If the simulation is running on multiple processors, it partitions the fault elements. This partitioning is intended to ensure that each processor is responsible for roughly an equal number of elements and that elements on the same processor are on the same fault or geographically close to each other. Next, each processor calculates stress influences by all model elements upon the local elements as described earlier in this article.

The core of the simulation involves cycling between two phases: the first determines long term stress buildup in the system, and the second propagates a rupture through the system, as indicated by blue and purple in Figure 3. The simulation begins in the long-term stage by calculating the rate of long-term stress buildup for each element. In a parallel simulation, each processor determines when each of the local elements will rupture, which is then globally reduced to find when and where the first rupture will occur.

Once this is determined, the rupture is propagated through the system using the mechanisms described earlier. First, the ruptured elements are processed and their new stresses are communicated through the system. Each processor recalculates the effects of this change on the stresses of their local elements and determines which (if any) ruptured. If any processors experience further ruptures, the rupture propagation phase continues. This phase generally involves multiple propagation steps until the earthquake is finished. Once there are no more ruptures, the simulation returns to the long-term stress calculation. After the specified number of simulation years have elapsed, the simulation is ended.

VIRTUAL CALIFORNIA SIMULATIONS

In order to illustrate the effect of the dynamic triggering η and the slip threshold S_t on the output of Virtual California, we explored the parameter space by varying $\eta \in [0.3, 0.9]$ in increments of 0.1 and $S_t \in [0, 40]$ in increments of 10. The result was 36 simulations, each run for 50,000 years using the model described in the Fault Model section. Results from these simulations are shown in Figure 4. As can be seen in the figure, η has the effect of encouraging larger ruptures at the expense of smaller ones (Fig. 4a) but has a relatively small effect on the amount of slip in events (Fig. 4c). The effect of S_t is more complex (Fig. 4b). Without S_t there is a hard lower limit to the magnitude of events that Virtual California can produce. This is because both the size of the elements and the amount by which they slip during an event are fixed by the model. Once S_t is enabled, however, the amount of slip during an event varies. This allows smaller events to be produced. Also, because smaller events will release less stress, larger values of S_t produce more large events when the extra accumulated stress is released. Last, by reducing the amount of slip for small ruptures, S_t prevents overslipping (Fig. 4d). A detailed description of the output of Virtual California and a comparison to the output of other simulations is included in Tullis *et al.* (2012b).

CONCLUSION

Understanding earthquake behavior is crucial to disaster prevention and management. The Virtual California program helps perform ensemble domain simulations, which are statistically similar to actual earthquakes and will aid in prediction and understanding of earthquakes in the future.

ACKNOWLEDGMENTS

Special thanks to Hiroyuki Noda for helpful reviews. This research was supported by the Southern California Earthquake Center. SCEC is funded by National Science Foundation Cooperative Agreement EAR-0529922 and U.S. Geological Survey Cooperative Agreement 07HQAG0008. The SCEC contribution number for this paper is 1630. This work was also supported by National Aeronautics and Space Administration (NASA) grant number NNX08AF69G, JPL subcontract number 1291967, and NASA Earth and Space Science fellowship number NNX11AL92H.

REFERENCE

Field, E. H., T. E. Dawson, K. R. Felzer, A. D. Frankel, V. Gupta, T. H. Jordan, T. Parsons, M. D. Petersen, R. S. Stein, R. J. Weldon II, and C. J. Wills (2007). The Uniform California Earthquake Rupture Forecast, version 2 (UCERF 2), *U.S. Geol. Surv. Open-File Rept. 2007–1437*, http://pubs.usgs.gov/of/2007/1437/ (last accessed September 2012).

Heien, E. M., and M. K. Sachs (2012). Understanding long term earthquake behavior through simulation. *Comput. Sci. Eng.* **14,** no. 5, 10.

Okada, Y. (1992). Internal deformation due to shear and tensile faults in a half-space, *Bull. Seismol. Soc. Am.* **82,** 1018–1040.
Rundle, J. B. (1988). A physical model for earthquakes. 2. Application to southern California, *J. Geophys. Res.* **93,** 6255–6274.
Rundle, J. B., P. B. Rundle, A. Donnellan, and G. Fox (2004). Gutenberg-Richter statistics in topologically realistic system-level earthquake stress-evolution simulations, *Earth Planets Space* **55** no. 8, 761–771.
Rundle, P. B., J. B. Rundle, K. F. Tiampo, J. S. S. Martins, S. McGinnis, and W. Klein (2001). Nonlinear network dynamics on earthquake fault systems, *Phys. Rev. Lett.*, **87,** no. 14, 148501-1–148501-3, doi: 10.1103/PhysRevLett.87.148501.
Rundle, J. B., P. B. Rundle, W. Klein, J. S. S. Martins, K. F. Tiampo, A. Donnellan, and L. H. Kellogg (2002). Gem plate boundary simulations for the Plate Boundary Observatory: A program for understanding the physics of earthquakes on complex fault networks via observations, theory, and numerical simulation, *Pure Appl. Geophys.* **159,** 2357–2381.
Rundle, J. B., P. B. Rundle, A. Donnellan, P. Li, W. Klein, G. Morein, D. L. Turcotte, and L. Grant (2006a). Stress transfer in earthquakes, hazard estimation and ensemble forecasting: Inferences from numerical simulations, *Tectonophysics*, **413,** 109–125.
Rundle, P. B., J. B. Rundle, K. F. Tiampo, A. Donnellan, and D. L. Turcotte (2006b). Virtual California: Fault model, frictional parameters, applications, *Pure Appl. Geophys.*, **163,** 1819–1846.
Tullis, T. E., K. Richards-Dinger, M. Barall, J. H. Dieterich, E. H. Field, E. M. Heien, L. H. Kellogg, F. F. Pollitz, J. B. Rundle, M. K. Sachs, D. L. Turcotte, S. N. Ward, and M. B. Yikilmaz (2012a). Generic earthquake simulator, *Seismol. Res. Lett.* **83,** no. 6, 959–963.
Tullis, T. E., K. Richards-Dinger, M. Barall, J. H. Dieterich, E. H. Field, E. M. Heien, L. H. Kellogg, F. F. Pollitz, J. B. Rundle, M. K. Sachs, D. L. Turcotte, S. N. Ward, and M. B. Yikilmaz (2012b). Comparison among observations and earthquake simulator results for allcal2 California fault model, *Seismol. Res. Lett.* **83,** no. 6, 994–1006.
Wells, D. L., and K. J. Coppersmith (1994). New empirical relationships among magnitude, rupture length, rupture width, rupture area, and surface displacement, *Bull. Seismol. Soc. Am.* **84,** no. 4, 974–1002.
Yakovlev, G., D. L. Turcotte, J. B. Rundle, and P. B. Rundle (2006). Simulation-based distributions of earthquake recurrence times on the San Andreas fault system, *Bull. Seismol. Soc. Am.* **96,** 1995–2007.
Yikilmaz, M., D. Turcotte, G. Yakovlev, J. Rundle, and L. Kellogg (2010). Virtual California earthquake simulations: Simple models and their application to an observed sequence of earthquakes, *Geophys. J. Int.*, **180,** 734–742.
Yikilmaz, M. B., E. M. Heien, D. L. Turcotte, J. B. Rundle, and L. H. Kellogg (2011). A fault and seismicity based composite simulation in northern California, *Nonlinear Process. Geophys.* **18,** no. 6, 955–966; also available at http://www.nonlin-processes-geophys.net/18/955/2011/ (last accessed September 2012).

Michael K. Sachs
John B. Rundle[1]
Department of Physics
University of California, Davis
Davis, California 95616 U.S.A.
mksachs@ucdavis.edu

Eric M. Heien
Donald L. Turcotte
M. Burak Yikilmaz
Louise H. Kellogg
Department of Geology
University of California, Davis
Davis, California 95616 U.S.A.

[1] Also at Geology Department, University of California, Davis, California 95616 U.S.A.; and Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501 U.S.A.