Searching for patterns in Virtual California

Michael Sachs
Department of Physics
mksachs@ucdavis.edu

Earthquakes are devastating and fascinating events. Understanding patterns in earthquakes would be useful not just in understanding the underlying mechanisms involved in producing these events, but also in forecasting and preparing for them. However, dynamics operating at many different scales and the difficulty of collecting data make earthquakes difficult to study. Because of this, simulations play an important role. I discuss Virtual California, a specific earthquake simulator that I have been involved in developing. I then apply tools from information theory to the output from this simulator and discuss the results.

Introduction

Earthquakes are among the most devastating natural events faced by society. In 2011 the damage resulting from earthquakes accounted for 59% of world wide economic losses due to natural disasters [1]. The magnitude 9.0 Tohoku earthquake off the coast of Japan on March 11, 2011 caused approximately $210 billion in damages and killed approximately 16,000 people [2]. Over the past decade events in the Indian Ocean on December 26, 2004 and Haiti on January 12, 2010 caused enormous devastation, totaling roughly 543,898 deaths [3] and $21.8 billion [4, 5] in economic losses. Overall the deaths caused by earthquakes over the last decade is estimated at 791,721 [6]. A repeat of the magnitude 7.9 1906 San Francisco earthquake could cause as much as $84 billion in damages [7] which would give this event the dubious distinction of being the third most costly natural disaster ever recorded, beating out Hurricane Katrina.

Aside from the destruction caused by earthquakes, they are also interesting phenomena in their own right. The largest earthquake ever recorded, the 1960, magnitude 9.5 Chilean earthquake released 2.24×10^{23} joules of energy or 5.35×10^{7} megatons of TNT. This is $\sim$ 10,000 times the destructive power of the world nuclear arsenal and $\sim$ 56% the energy released in the Chicxulub impact which is thought to be responsible for the Cretaceous-Paleogene extinction [8]. The Tohoku earthquake released enough energy to slow down the earth's rotation, shortening the length of the day by 1.8 microseconds [9].

Clearly, studying earthquakes is both useful – better understanding can lead to better disaster preparedness and prevention, and interesting – they are among the most energetic events in nature that we can study up close. However, the fact that they mostly occur underground, and are the result of forces that can take many of thousands of years to accumulate and operate at scales ranging from microscopic to global, makes studying them difficult. Up until recently the best data that had been collected was a list of recorded events – the seismic catalog – that consists of times, locations and magnitudes. This was supplemented by a much smaller catalog of historical events that were not directly recorded but were inferred by geologic evidence (the paleoseismic catalog). Starting a few decades ago this catalog data began to be supplemented by ground deformation data collected by GPS arrays and radar interferograms. However, the quality and coverage of this newer data is inconsistant and it may be several more decades before the gaps in it are filled.

The difficulty of collecting earthquake data directly make the use of computer simulations in studying them important. Simulations can be developed that match existing observations, but are capable of producing data over a much wider range of scales, potentially filling gaps in the observational data. Once this data is produced the question becomes how do we use the simulated data to study the actual fault systems. One answer to this question is to use the tools developed in information theory to understand the patterns within the simulations that are the result of the complex underlying dynamics of the systems.

The purpose of this project is to begin applying information theoretic tools to earthquake simulation data. I will describe one particular earthquake simulator that I have been involved in developing: Virtual California [10, 11, 12]. Using output from Virtual California I have developed an initial set of tools, based on information theoretic quantities, to analyze earthquake simulation data. I will discuss the initial results from these tools and outline future directions for this work.

Ensemble-Domain Verses Time-Domain Simulations

Earthquake fault simulations generally fall into one of two categories: ensemble-domain and time-domain. Before I describe Virtual California in detail it is important to draw a distinction between these two types of simulations. A time-domain fault simulation attempts to solve a set of differential equations that govern the evolution of the system. These simulations usually employ an approximation scheme like finite-element analysis. The result of these solutions is a function which has time as an independent variable, so in principle the state of the system at any given time is encoded in the solution. Aside from the considerable computational difficulty of applying the time-domain approach to fault systems, problems arise from the sensitivity of these solutions to initial conditions. As mentioned above, collecting information about the current state of a fault is difficult which makes precise definitions of initial conditions difficult.

An ensemble-domain simulation sidesteps the problem of initial conditions by looking for the most likely states of the system given some set of external parameters. Instead of generating a single "history" of the system, many histories are created – a so-called ensemble. This ensemble is then the basis of a statistical analysis. Because the ensemble is a combination of many different paths the system can take, it is less sensitive to where those paths begin. A familiar example of this approach is the Metropolis Monte Carlo algorithm [13]. Virtual California as well as several other simulators [14, 15, 16] are examples of an ensemble-domain simulation.

Virtual California

There are three major components that make up Virtual California: a fault model, a set of quasi-static interactions or "Green's functions" and an event model. In spite of the name, the only component of Virtual California that is California-specific 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. All of these components are designed around the concept of a "back-slip" model. Back-slip models have the advantage of a static fault geometry. In actual earthquakes, plate motions far away from faults cause stress buildups at the faults. When the faults break, the cracking of the earth causes discontinuous displacements across faults (Figure 1 top). These displacements cause the fault system to evolve over time. This evolution is very difficult to simulate due to the many complexities involved in crack propagation in the heterogenous earth. A back-slip model on the other hand, applies displacements *at the faults in the opposite direction of the distant plate motions* (Figure 1 bottom), much like pulling apart leaf springs. When stress builds up to some threshold, the displacements snap back to their equilibrium positions. In this way the accumulation and release of stress can be modeled without changing the actual geometry of the fault system. The fault model is designed to implement this approach by modeling the minimal amount of geometry needed to describe the faults, and then treating the faults as collections of disconnected elements that interact only through the Green's functions. The Greens functions encode all of the physics of stress and strain in the earth. Both of these components are described in detail below.

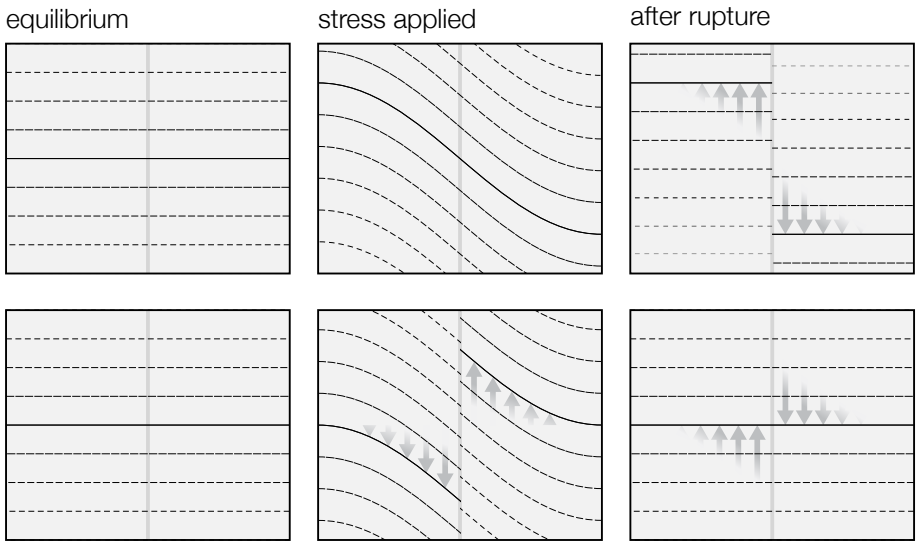


Figure 1: Actual fault ruptures compared to back-slip ruptures. Each scenario is shown (from left to right) before stresses are applied, as stresses are applied, and after the rupture. The horizontal lines show the deformation of the earth around the fault which is illustrated by the grey vertical line. **Top** an actual rupture. **Bottom** a back-slip rupture.

Fault Model

As mentioned above, the fault model is the only component of Virtual California that is specific to California. In general, a fault model for Virtual California is a model of the *faces* of the faults only, not the surrounding earth in which the faults are embedded. This is an important simplification that allows for large fault systems to be simulated. The face of the fault is meshed with square elements. Due to the fact that Virtual California is a back-slip model, it is not important that this meshing be continuous or that the elements not intersect. Elements will only interact with each other through the Green's functions described below. An example of a Virtual California fault model is shown in Figure 2. Each element is assigned a back-slip velocity and a breaking stress – values that are taken from observations for models of actual fault systems. The back-slip velocity is a vector that always lies in the plane of the element, but can point in any direction in the plane. In this way, thrust faults and strike slip faults can be modeled.

Green's Functions

Interactions between fault elements are governed by quasi-static Green's functions. The effect that one element has on another depends on their relative positions and orientation and on the direction of their slip displacements. Because the fault geometry is static, these interactions only need to be calculated once. In order to calculate the stress Green's functions, every element is back-slipped 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, x, in the simulation due to changes on all other elements is given by [11, 10]:

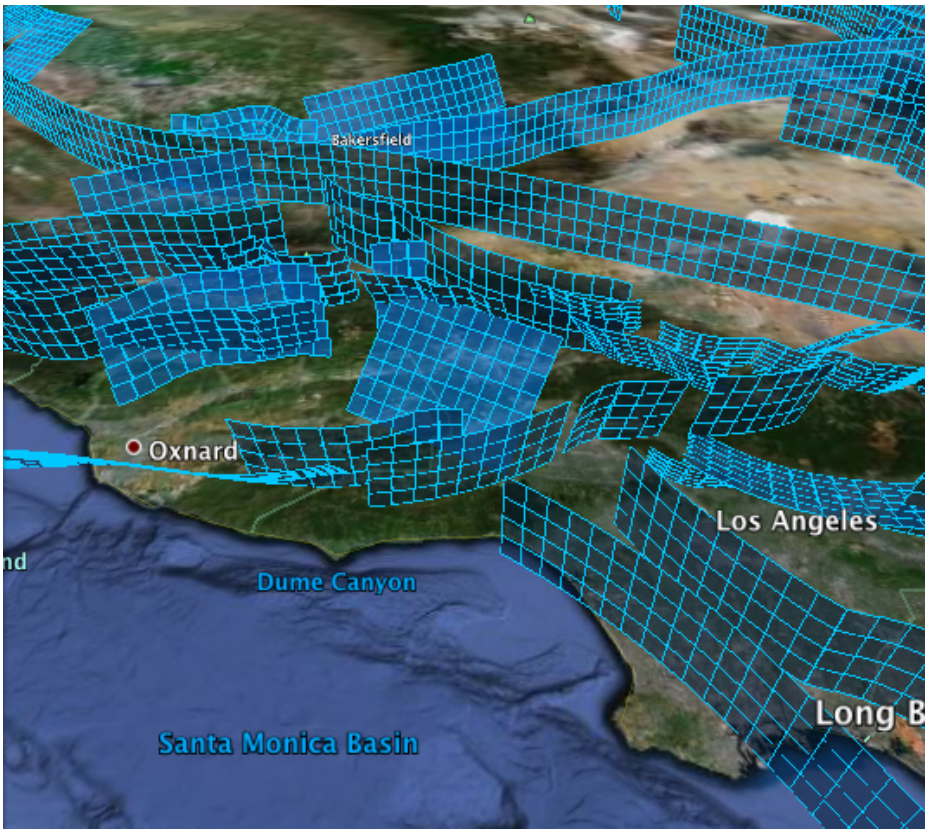


Figure 2: Detail of a Virtual California fault model.

$$\sigma_{ij}(x,t) = \int dx_k^{'} T_{ij}^{kl}(x - x^{'}) s_l(x^{'},t), \tag{1}$$

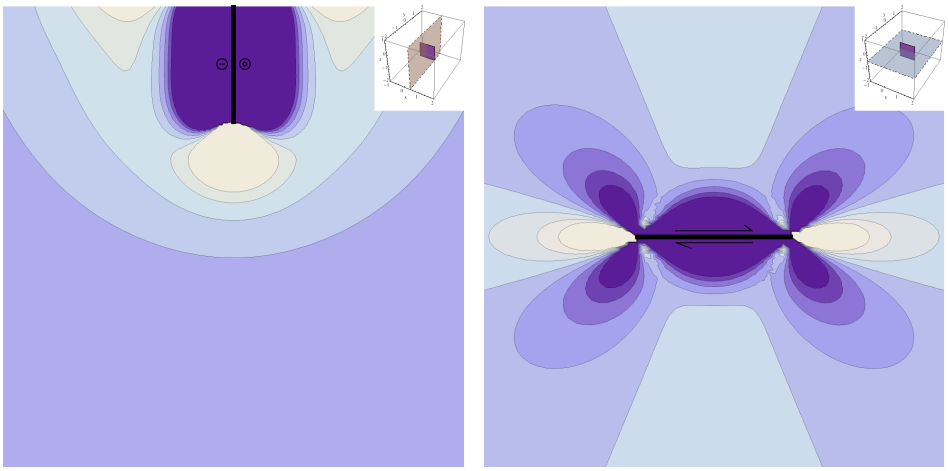
where $s_l(x^{'},t)$ is the slip in direction l and $T_{ij}^{kl}(x - x^{'})$ is the Green's function tensor. The Einstein summation convention is assumed. The indicies i, j, k and l run over the cartesian coordinate axes, x, y, and z. In the case of Virtual California, the field is only evaluated at 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 6 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:

$$\begin{array}{rcl} \sigma_s^A(t) & = & T_s^{AB} s_B(t) \\ \sigma_n^A(t) & = & T_n^{AB} s_B(t) \end{array} . \tag{3}$$

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



(a) Front view (b) Top view

Figure 3: The shear stress field, σ_{xy}, created by horizontal back-slip as viewed from the front and top of an element. The direction of the back-slip is indicated by the arrows. Tan color indicates $\sigma_{xy} > 0$ and blue $\sigma_{xy} < 0$.

The actual values of the Green's functions are calculated using an implementation of Okada's half-space deformations [17]. Example's of the output generated by this implementation of Okada's interactions are shown in Figure 3.

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):

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

where μ_s^A is the static coefficient of friction calculated from the model element strengths. When $\boldsymbol{CFF}^{A}(t_f) = 0$ the element $\boldsymbol{A}$ fails. To simplify the notation we will remove the element label $\boldsymbol{A}$ 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: $\boldsymbol{CFF}(t_f) = 0$. Because the interactions in the previous 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 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. The first element to fail is allowed to slip back toward its equilibrium position. The amount the element slips, Δs, is related to stress-drop defined for the element in the model, $\Delta\sigma$, by [11, 10]:

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

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 $\frac{N_{ef}}{S_t}$ 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 Eq. 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 that the amount of stress accumulated during the rupture is greater than a pre-defined dynamic triggering factor η:

$$\frac{CFF_{init} - CFF_{final}}{CFF_{init}} > \eta. \quad (6)$$

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 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, but also may not release their accumulated stress all at once. This process continues until there are no more failures, at which point the event is over.

Figure 4(a) shows an example of how stress is released and accumulated over multiple event cycles. Figure 4(b) shows how a single element failure can lead to a cascading series of failures in a single event. Figure 4(b) also shows that elements can fail multiple times during an event but will not accumulate additional stress after they fail.

A Virtual California Simulation

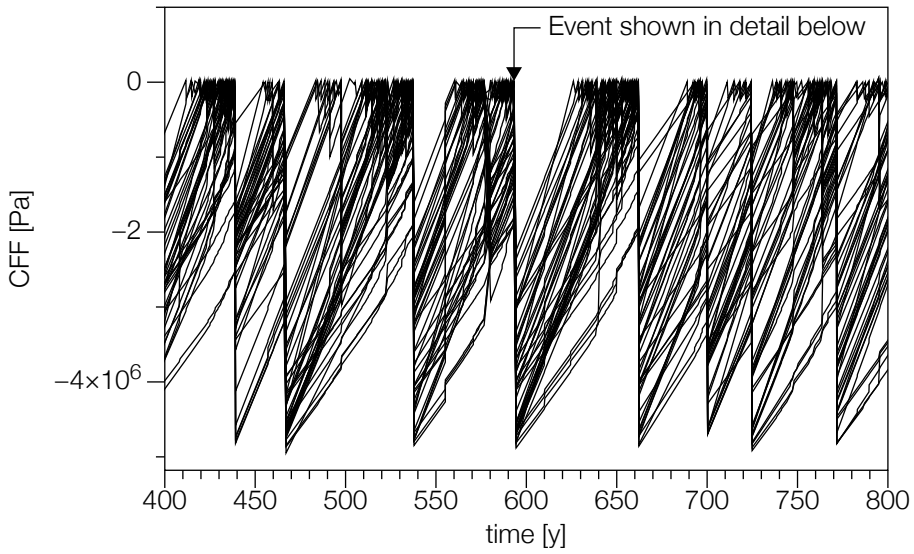
In order to illustrate the types of results that Virtual California produces, seven simulations were run on the model shown in Figure 2. Each simulation uses a different value of the dynamic triggering factor η (Eq. 6) and all of them were run for 50,000 simulated years.

The results are shown in Figs. 5 and 6. These plots are used because they allow a direct comparison with observed quantities. Thus we can evaluate how well Virtual California represents seismicity in California.

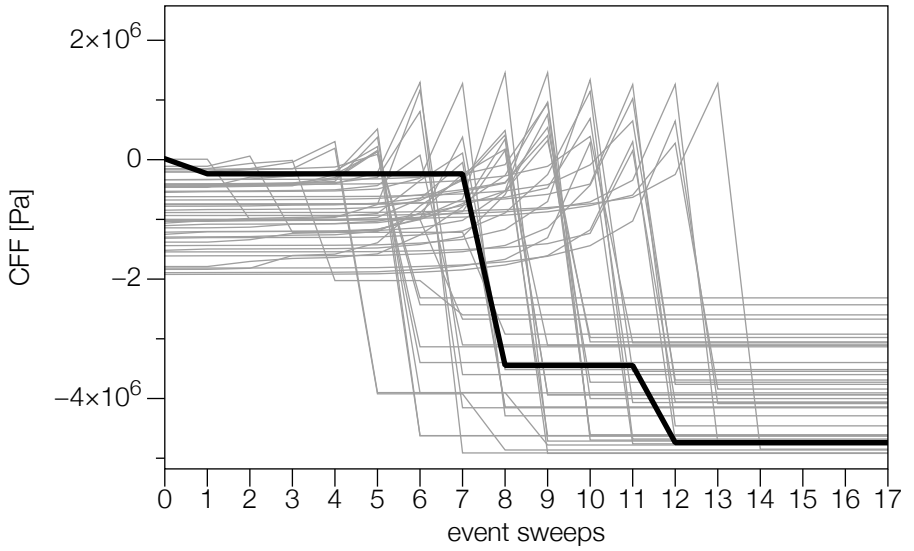
A particularly important scaling relation in seismology is the Gutenberg-Richter frequency-magnitude relation:

$$\log N_c = a - bM \quad (7)$$

where N_c is the total number of earthquakes with magnitude greater than M, b is a near universal constant in the range $0.8 < b < 1.1$, and a is a measure of the



(a) CFF vs time of a series of events



(b) CFF vs sweeps for the event at $t = 593.187$

Figure 4: **Top** 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. **Bottom** 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.

level of seismicity. What Eq. 8 tells us is that for every ~ 10 $M = 5.0$ earthquakes, for example, one can expect ~ 1 $M = 6.0$ earthquake. The Gutenberg-Richter relation for the results of the Virtual California simulations are shown in Figure 5. Results are given for seven values of the dynamic triggering factor η. For large η rupture propagation is inhibited and there are fewer large earthquakes. For small η ruptures propagate freely and large earthquakes dominate. This figure shows that

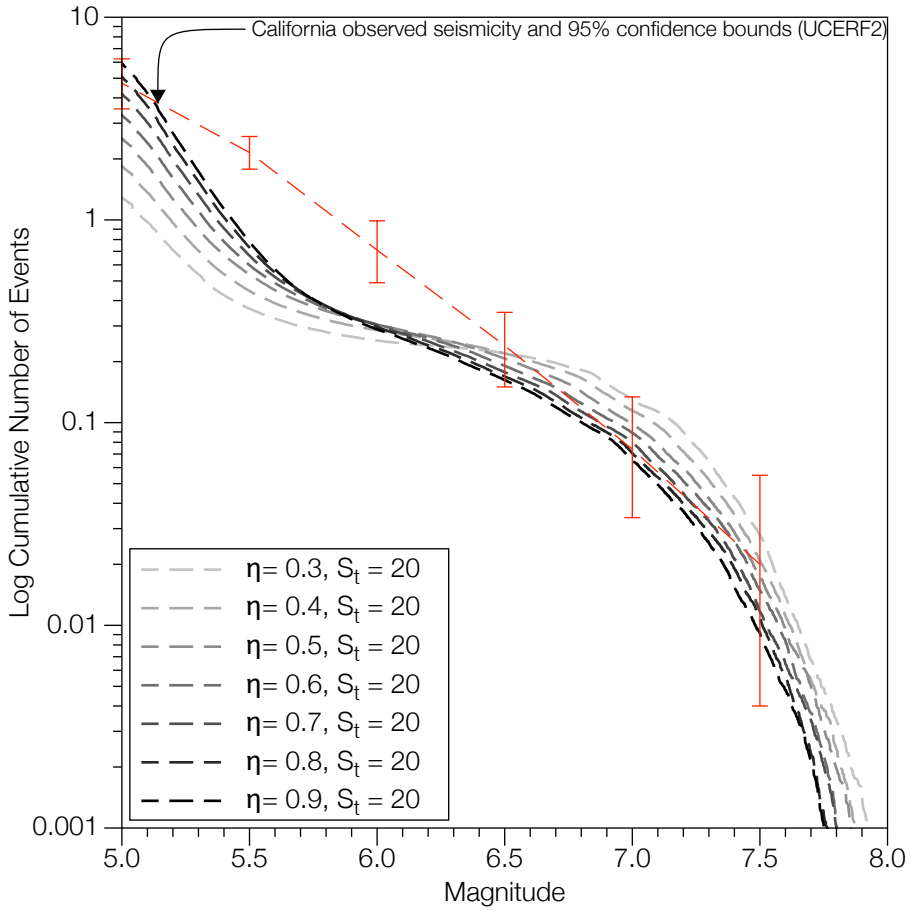


Figure 5: Frequency-magnitude statistics for seven Virtual California simulations, each with a different values of the dynamic triggering factor η. The red dashed line and error bars the 95% confidence bounds on the observed seismicity in California as reported by UCERF2 [18].

a M~7.6 or larger earthquake is expected about every 100 years. Also shown in this figure is the observed seismicity in California as reported by UCERF2 [18]. The drop-offs for events below M~6.3 and above M~7.3 are related to limiting aspects of the fault model's geometry. The magnitude of an earthquake is related to the surface are of ruptured faults by [19]:

$$M \sim \log(SA) \tag{8}$$

where A is the event rupture area and S is the event slip. Hence the fall-off of events at large magnitude is related to the maximum surface area of faults in the model. The fall off at low magnitudes is related to the element size. Smaller elements would allow smaller magnitude events to occur.

Several other empirical relationships between earthquake observations were reported by Wells and Coppersmith [20]. As an example of how the output of Virtual California compares to these relations, Figure 6 shows the relationship of event rupture area to event magnitude. There is some scatter in the Virtual California data, however, the relationship reported in [20] is based on 148 events while there

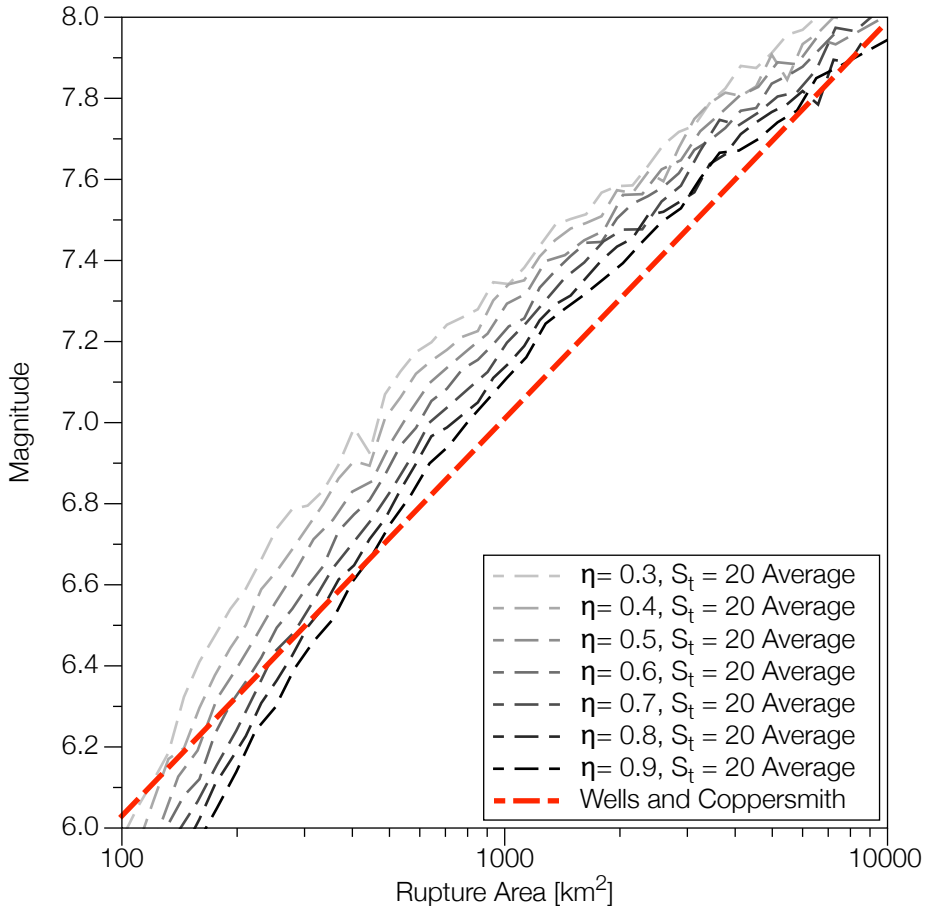


Figure 6: Magnitude versus rupture area for all events in two Virtual California simulations, each with a different value of the dynamic triggering factor η. The red dashed line is the empirical relationship as reported in [20].

are over 100,000 events in the Virtual California data. It is encouraging that the simulation does so well in reproducing the empirical relation over such a large number of events.

Information Theoretic Measures for Virtual California

In order to explore how tools from information theory can be used to analyze output from Virtual California, I begin with the simple fault model shown in Figure 7. This model is a vertical square, three elements wide and three elements deep. Each element is two kilometers square with a slip rate of 1×10^{-10} ms^{-1} and a breaking stress of 1×10^{7} pascals. The direction of the slip is right lateral strike slip, or in other words, if one were to stand on the ground looking at the fault the ground on the opposite side of the fault would move to the right. In order to remove asymetrical effects caused by the surface of the space in which the fault is embedded, the top of the fault is 50 kilometers below the surface. An example of the $\boldsymbol{CFF}$ (Equation 4) as a function of time for each of the nine elements is shown in Figure 8. There is clearly some periodic structure in Figure 8, but the frequency of events is not

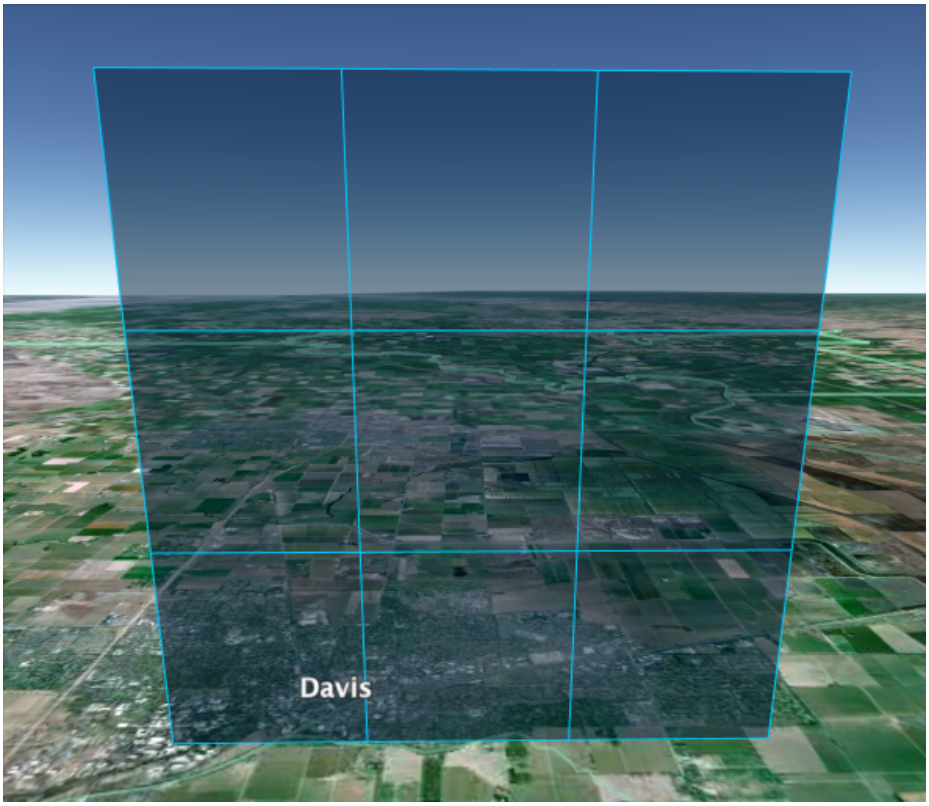


Figure 7: The three cell by three cell fault used in my initial studies. Each element is a two kilometer square.

obvious and the clustering of events appears chaotic.

To better understand the time series presented in Figure 8 I plot the map for each element's CFF in Figure 9. Before plotting I normalize the CFF so the lowest values (ie. less stress applied to the element) are mapped to one. $CFF = 0$ still corresponds to the point at which the element breaks. Each element is normalized to its own lowest value not the lowest value of CFF on the fault, so the magnitudes of CFF plotted here relative to each element. Because the plot is $CFF(t)$ vs $CFF(t+1)$, these maps give a sense of the types of CFF changes that the elements undergo. In general for each element the changes in CFF fall into two regions: **a)** points that are in a wedge below the line at $CFF(t) = CFF(t+1)$; and **b)** points that are in a vertical strip where $CFF(t) < CFF(t+1)$. Region **a** represents stress accumulation. $CFF(t) >= CFF(t+1)$ which means that the element is getting closer to its breaking point at $CFF = 0$. Region **b** represents stress release. Focusing first on region **a**, elements on the edge of the fault show more scatter in this area than the element in the center. This indicates that stress accumulates in smaller increments in the center element than in the edge elements. This is expected because stress is generally higher at the edges of cracks as illustrated in Figure 3. In region **b** however, there is more scatter in the center element. This seems to indicate that the center element generally enters the rupture propagation phase described above at a low stress but then accumulates enough stress during the rupture to fail. When it does fail however, it releases less stress in general than the elements at the edges.

Overall it seems in Figure 9 that the position of an element in the fault has some

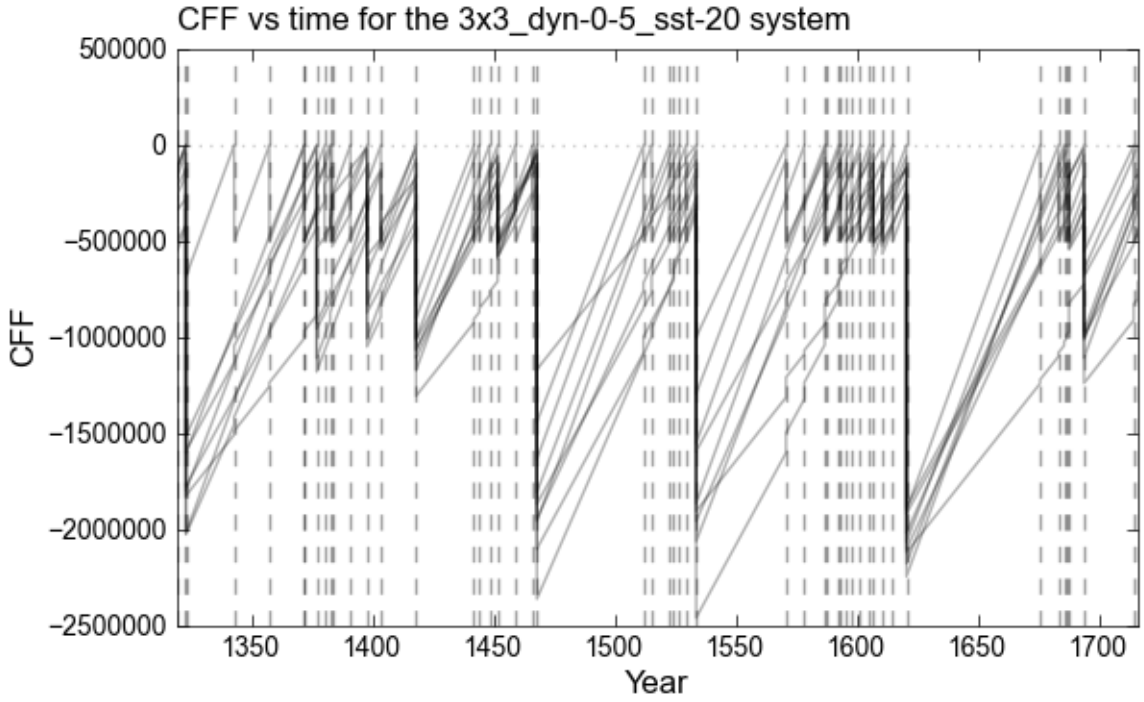


Figure 8: The CFF as a function of time for the nine elements in the fault in Figure 7. Vertical dashed lines mark the years where events occur.

effect on its behavior. What this effect is however is not clear. To explore this further is is desirable to calculate the entropy rate h_μ and the excess entropy E for each element. This presents a challenge: excess entropy and entropy rate are usually calculated from binary data. What Virtual California gives us however is a continuous real number between some minimum value and zero. It is not clear from the maps in Figure 9 how this data can be easily digitized. The solution to this problem lies in a quantity called the permutation entropy [21]. The idea behind permutation entropy is to count the sort permutations at each word length and use this quantity to calculate the entropy. For example, take the following time series:

$$x = (2, 5, 10, 1, 5, 8, 20, 3). \tag{9}$$

There are seven words of length two in x: (2,5), (5,10), (10,1), (1,5), (5,8), (8,20), and (20,3). The two possible sort permutations here are first number bigger than last number (1,0) and last number bigger than first number (0,1). There are five (0,1) permutations and two (1,0) permutations. The permutation entropy for this word length is then:

$$H(2) = -\frac{5}{7}\log\frac{5}{7} - \frac{2}{7}\log\frac{2}{7} \simeq 0.863, \tag{10}$$

where the logs are base 2. For word length three, there are six words in x: (2, 5, 10), (5, 10, 1), (10, 1, 5), (1, 5, 8), (5, 8, 20), and (8, 20, 3) with permutations (0,1,2), (1,2,0), (2,0,1), (0,1,2), (0,1,2), and (1,2,0). The permutation entropy for words of length three is:

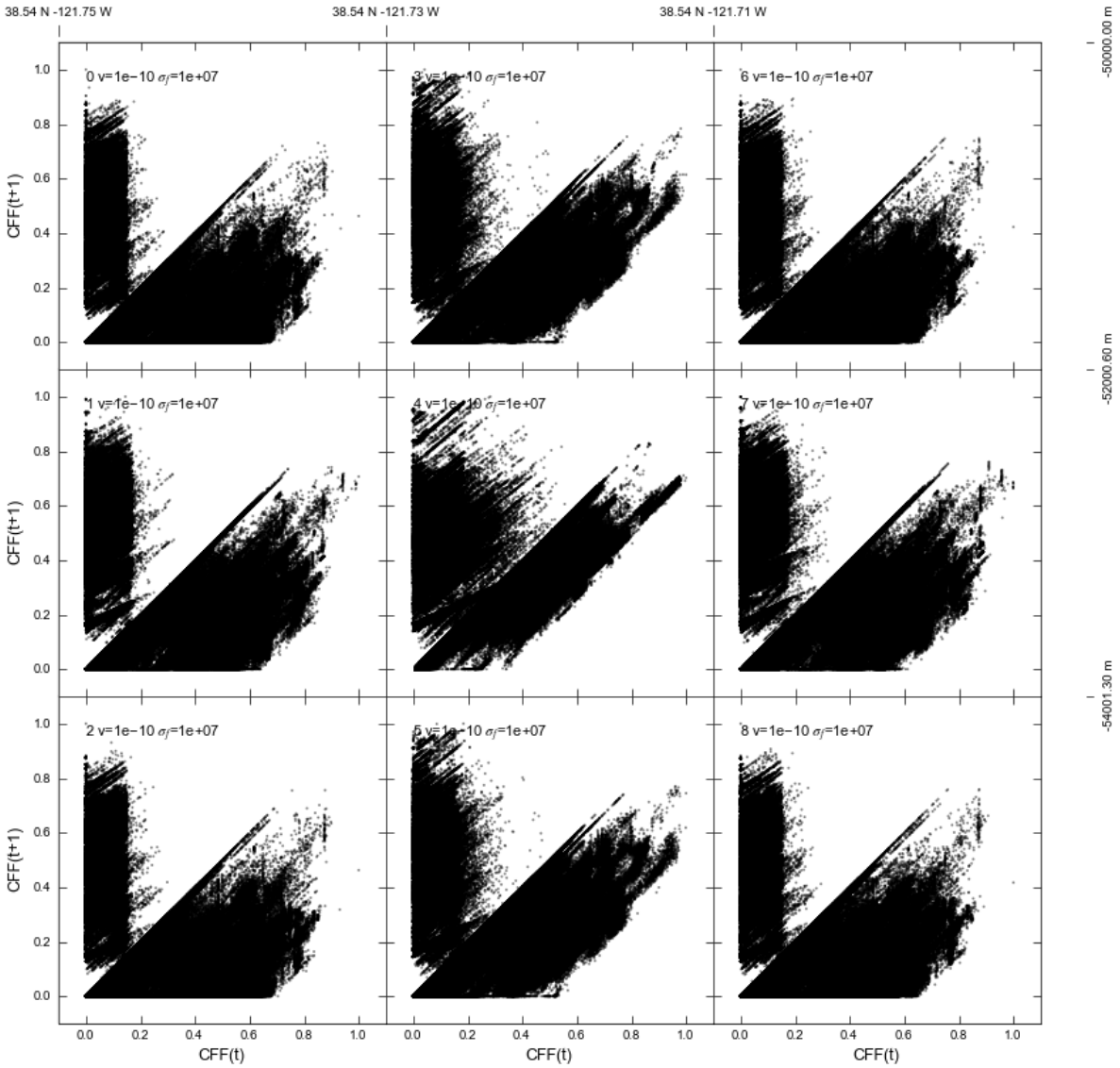


Figure 9: Return map for the time series in Figure 8. The three by three grid represents the fault in Figure 7 and the map for each element is plotted at that elements position in the fault.

$$H(3) = -\frac{3}{6}\log\frac{3}{6} - \frac{2}{6}\log\frac{2}{6} - \frac{1}{6}\log\frac{1}{6} \simeq 1.459. \tag{11}$$

Using this technique the permutation entropy can theoretically be calculated for any time series. To test this, I applied the algorithm to several known cases: the fair coin, the period eight process, and a uniformly distributed random variable. The results are shown in Figure 10. As can be seen in the figure, the permutation entropy reproduces the results of the block entropy calculation of the corresponding binary process.

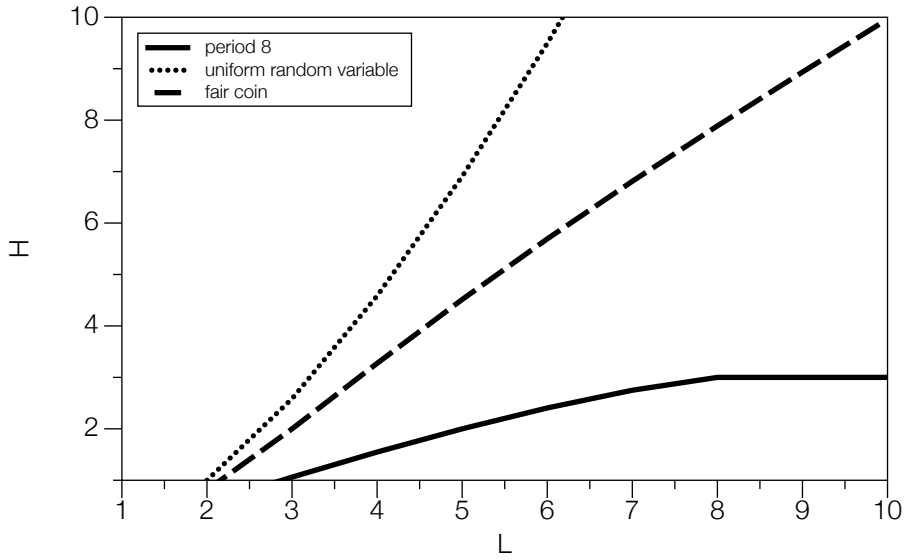


Figure 10: Tests of the permutation entropy calculation on several known systems.

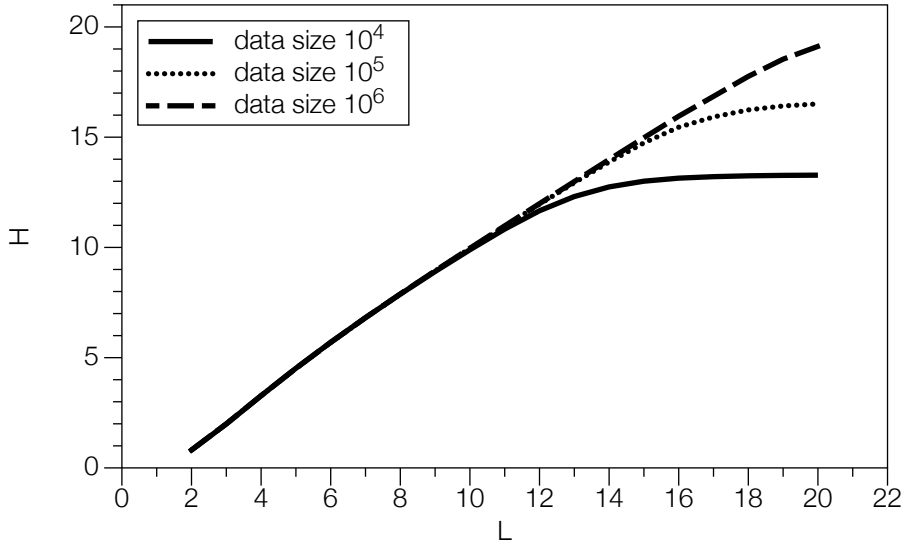


Figure 11: The effect of data length on the calculation of permutation entropy. The data used here was generated by a fair coin. At large word length H rolls off for smaller data sets.

A major issue with using permutation entropy is illustrated in Figure 11. In order to get an accurate calculation of $H(L)$ enough data is needed to sample the number of permutations completely. For words of length L there are $L!$ permutations, so for example, words of length 10 have 3,628,800 possible permutations. This means that data that contains 10^6 length 10 words is needed for a good sample. In practice the data being analyzed may not contain a complete set of permutations so the actual amount of data needed will depend on the system being analyzed. Figure 11 illustrates this effect, the value of H rolls off at higher word length for small datasets.

Run	Run length in years	Number of events produced
1	1,00,000	133,110
2	3,000,000	399,275
3	6,000,000	796,465
4	9,000,000	1,195,220
5	10,000,000	1,327,770

Table 1: The length of the various runs used to determine the the systems sensitivity to the under sampling effect.

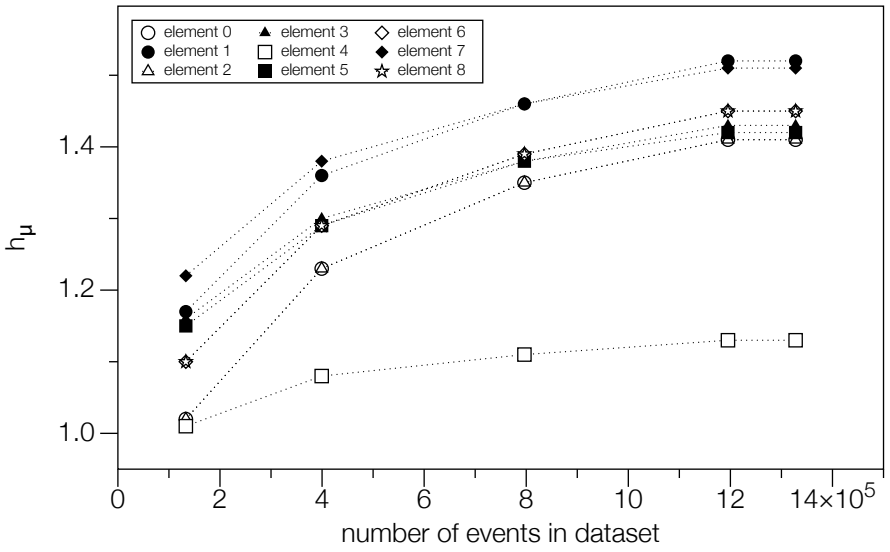


Figure 12: The change in h_μ for each element in the fault in Figure 7 calculated using different size datasets. For large datasets h_μ changes very little indicating a complete set of permutation samples.

To test how data from Virtual California is effected by the issue of undersampling I ran the simulation for several different time spans and calculated the permutation entropy up to word length 15. The time spans and the resulting number of events are listed in Table 1. I then estimated the entropy rate h_μ and the excess entropy E using:

$$\begin{aligned} h_\mu &\simeq H(15) - H(14) \\ E &\simeq \sum_{L=2}^{15} H(L) - h_\mu. \end{aligned} \tag{12}$$

The results are shown in Figure 12. Because the quantities in equation 12 use the high word length values of H they are sensitive to the undersampling roll off. As we increase the amount of data sampled in Figure 12, h_μ and E initially change a good deal. However as we approach 10^6 events the values settle down, indicating that we are getting good samples with this number of events.

The results of calculating the permutation entropy on the fault in Figure 7 are shown in Figure 13. As expected from the maps in Figure 9, the center figure behaves differently than the other elements. Its entropy rate h_μ is lower indicating that

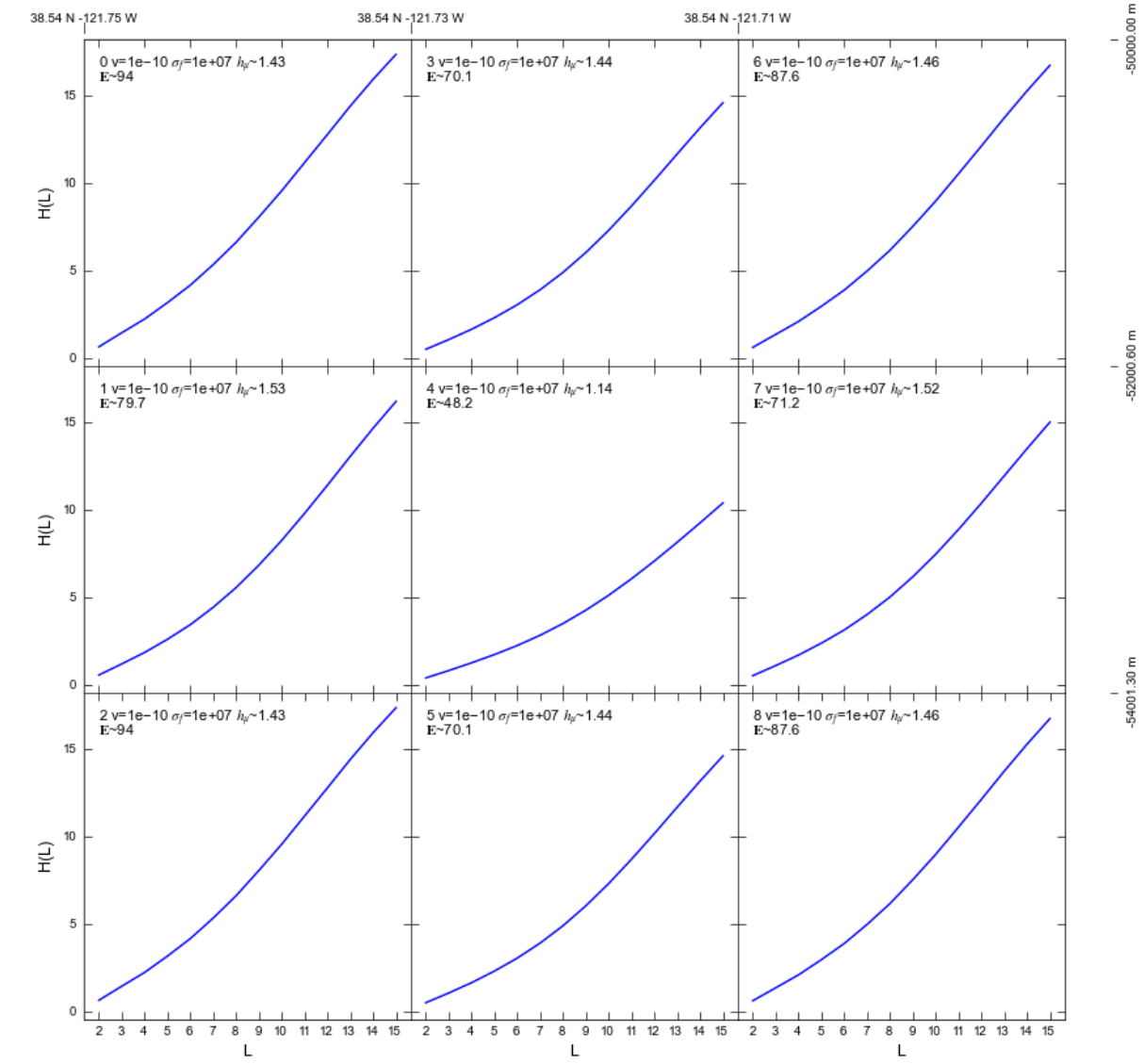


Figure 13: Permutation entropy as a function of word length for the elements in the fault shown in Figure 7.

it behaves less randomly than the elements at the edges. The corner elements have the highest excess entropy $\boldsymbol{E}$ but do not have the highest values of $\boldsymbol{h_\mu}$. The values of $\boldsymbol{h_\mu}$ and $\boldsymbol{E}$ as a function of position in the fault are shown in Figure 14.

Could the results in Figures 13 and 14 be specific to the 3×3 lattice? To answer this question I also analyzed square lattices from 1×1 element to 7×7 elements, all with the same size elements (2km), slip rates (1×10^{-10} ms^{-1}) and breaking stresses (1×10^{7} pascals). Results from the 5×5 case and the 7×7 case are shown in Figures 15 and 16. It is clear that the further from the edge an element is the less complex its behavior. How the distance from the edge effects the center element is shown

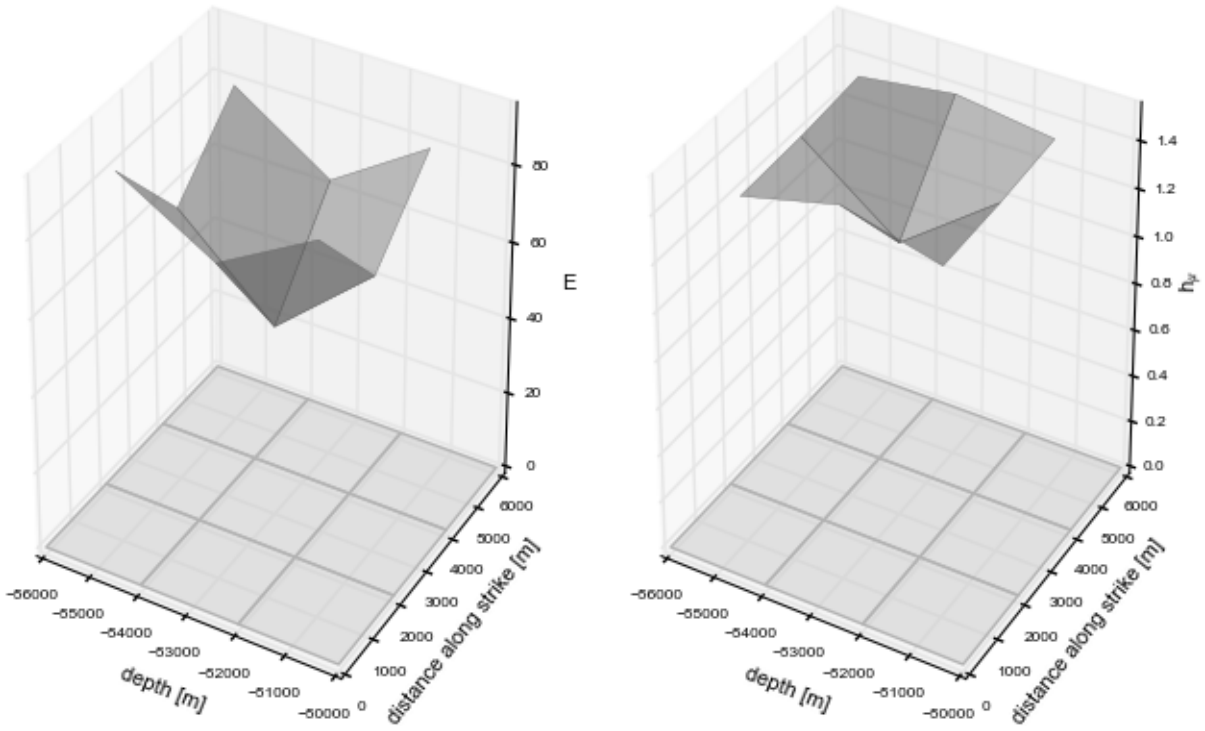


Figure 14: E and h_μ as a function of position in the fault in Figure 7.

in Figure 17. Here the values of h_μ and E are plotted versus lattice size n. For lattices with even lattice sizes, where there is no center element, the middle four elements were averaged. For $n = 1$ the CFF is periodic with period one so $h_\mu = 0$ and $E = 1$ as expected. The values at $n = 2$ are skewed because the average is taken over the four edge elements. For $n > 2$ however, h_μ and E steadily decrease.

Conclusion and Next Steps

The complexity of the output from Virtual California, even when dealing with very simple fault systems, make tools from information theory useful when analyzing the data. From my initial analysis it is clear that elements far from the edges of faults will exhibit less complex behavior than elements on the edges. It general these elements experience slower stress accumulation than the elements at the edges. It seems therefore, that the rate of stress accumulation on elements embedded in a fault is directly proportional to the randomness of their behavior. This is only a tentative conclusion however, more work needs to be done to verify this. It would be useful to vary other aspects of the model and see how h_μ and E vary. The slip rate and breaking stress could be changed to slow stress accumulation across the entire fault. Also, the size of the fault could be fixed and meshed with smaller and smaller elements to see if the effect in Figure 17 is due to euclidean distance from the edges of the fault or the number of elements only.

With this work I have only begun to scratch the surface of what is possible with the information theoretic tools available. Calculating the mutual information between

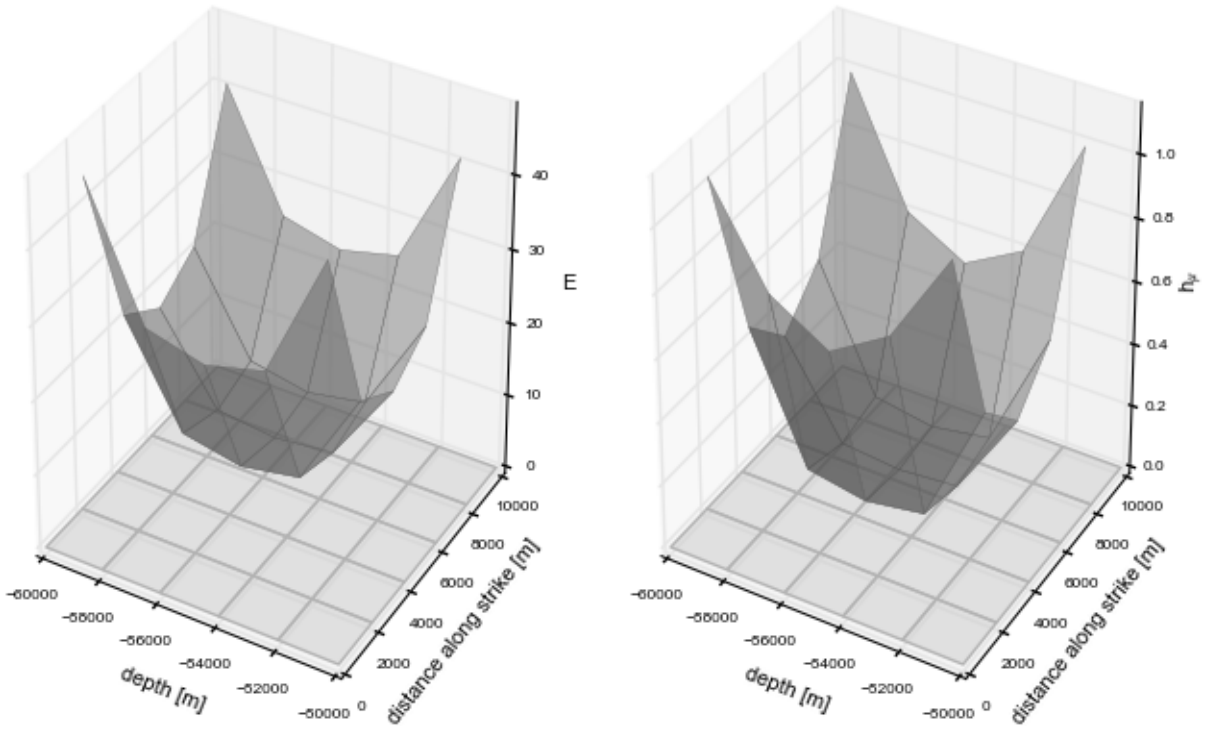


Figure 15: E and h_μ as a function of position for a 5×5 fault.

elements would be useful. Using ideas developed in [22] spacial patterns as well as temporal patterns can be analyzed. Looking at the behavior of models of actual fault systems is the ultimate goal. Improving our forecasting capabilities is an important possible outcome of these studies.

*

References

[1] Annual global climate and catastrophe report. http://thoughtleadership.aonbenfield.com/Documents/20120110_if_annual_global_climate_cat_report.pdf, January 2012.

[2] Review of natural catastrophes in 2011: Earthquakes result in record loss. http://www.munichre.com/en/media_relations/press_releases/2012/2012_01_04_press_release.aspx, January 2012.

[3] Earthquakes with 1,000 or more deaths since 1900. http://earthquake.usgs.gov/earthquakes/world/world_deaths.php, June 2012.

[4] Counting the cost. http://www.economist.com/blogs/dailychart/2011/03/natural_disasters, March 2011.

[5] Economic impact of the earthquake. http://haitiearthquake.web.unc.edu/economic-impact-of-the-earthquake/, June 2012.

[6] Earthquake facts and statistics. http://earthquake.usgs.gov/earthquakes/

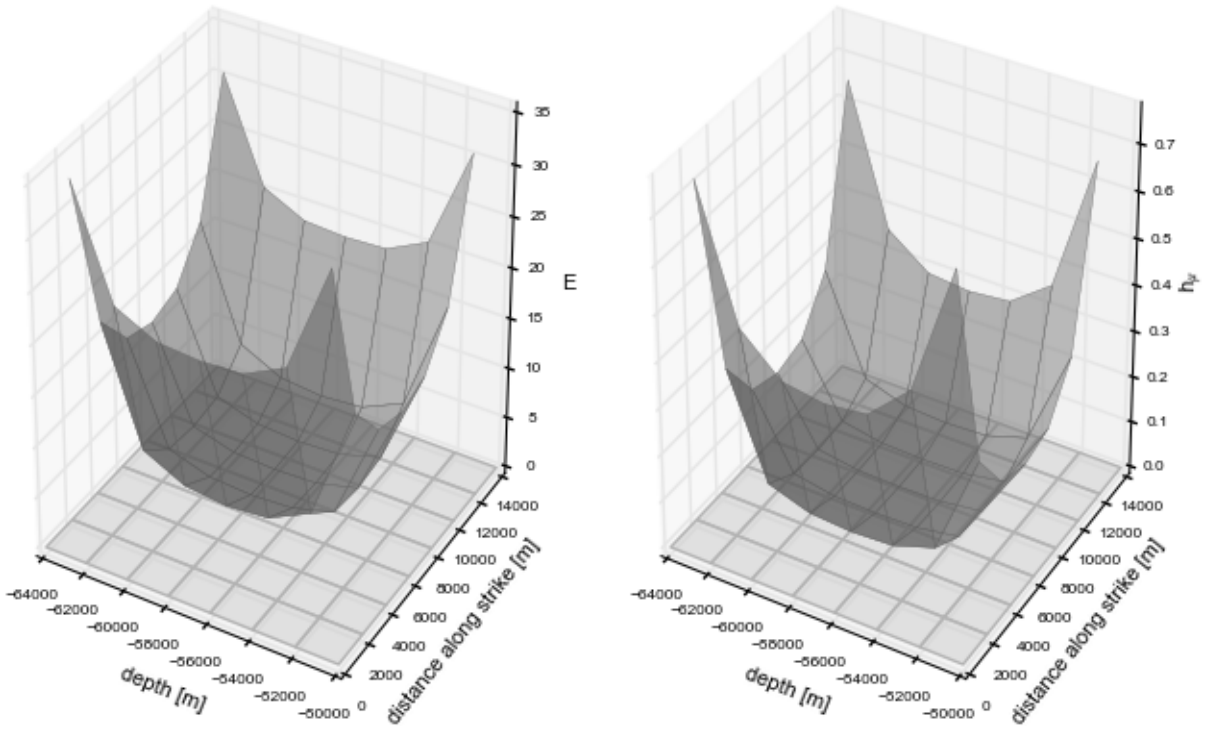


Figure 16: E and h_μ as a function of position for a 7×7 fault.

eqarchives/year/eqstats.php, June 2012.

[7] Rui Chen, David Branum, and Chris J. Wills. California geological survey - 2009 earthquake loss estimation. http://www.conservation.ca.gov/cgs/rghm/loss/Pages/2009_analysis.aspx, June 2009.

[8] Curt Covey, Starley L. Thompson, Paul R. Weissman, and Michael C. MacCracken. Global climatic effects of atmospheric dust from an asteroid or comet impact on earth. *Global and Planetary Change*, 9(3â€“4):263 – 273, 1994.

[9] Earth's day length shortened by japan earthquake. http://www.cbsnews.com/stories/2011/03/13/scitech/main20042590.shtml, March 2011.

[10] P. B. Rundle, J. B. Rundle, K. F. Tiampo, A. Donnellan, and D. L. Turcotte. Virtual California: fault model, frictional parameters, applications. *Pure Ap. Geophys.*, 163:1819–1846, 2006.

[11] J. B. Rundle, P. B. Rundle, A. Donnellan, P. Li, W. Klein, G. Morein, D. L. Turcotte, and L. Grant. Stress transfer in earthquakes, hazard estimation and ensemble forecasting: Inferences from numerical simulations. *Tectonophys.*, 413:109–125, 2006.

[12] M. K. Sachs, E. M. Heien, D. L. Turcotte, M. B. Yikilmaz, J. B. Rundle, and L. H. Kellogg. Virtual california earthquake simulator. *Seis. Rev. Lett.*, 2012 (forthcoming).

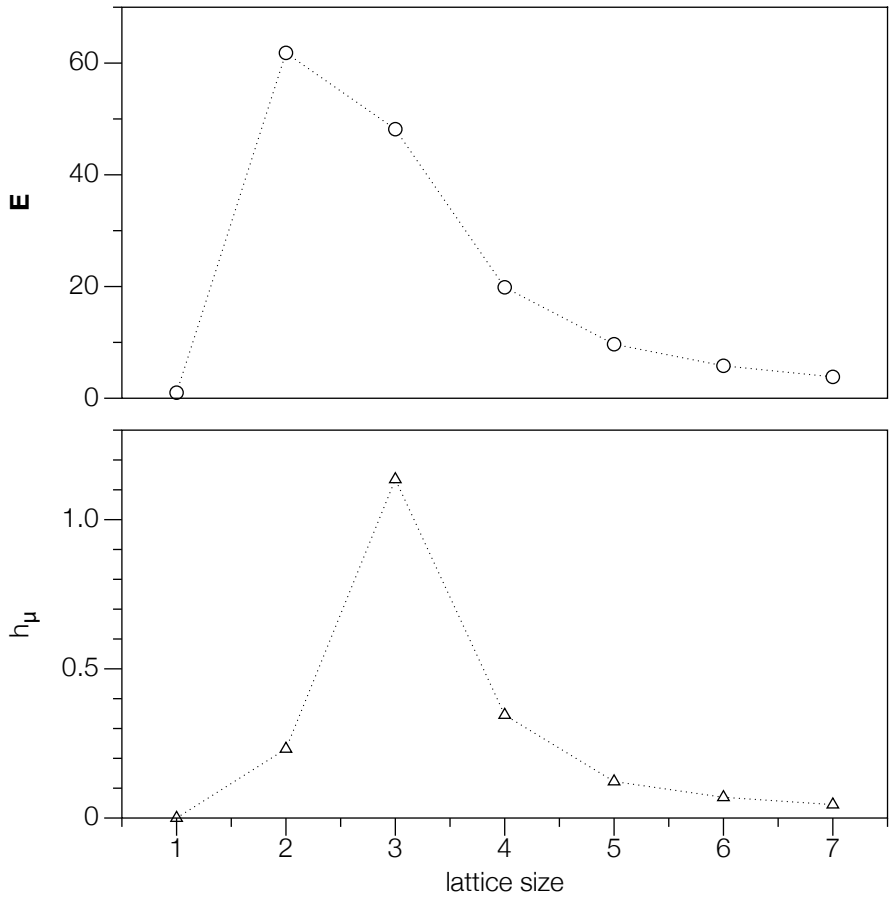


Figure 17: The change in E and h_μ for the center element (or the average over the four center elements for even lattice sizes) as the lattice size is increased.

[13] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087, 1953.

[14] S. N. Ward. A synthetic seismicity model for southern California: cycles, probabilities, and hazard. *J. Geophys. Res.*, 101(B10):22393–22418, 1996.

[15] James Dieterich and Keith Richards-Dinger. Earthquake recurrence in simulated fault systems. *Pure and Applied Geophysics*, 167:1087–1104, 2010. 10.1007/s00024-010-0094-0.

[16] Fred F. Pollitz. Epistemic uncertainty in california-wide synthetic seismicity simulations. *Bulletin of the Seismological Society of America*, 101(5):2481–2498, 2011.

[17] Y. Okada. Internal deformation due to shear and tensile faults in a half-space. *Bull. Seism. Soc. Am.*, 82:1018–1040, 1992.

[18] E. H. Field, 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. The uniform california earthquake rupture forecast, version 2 (ucerf 2). Technical report, Southern California Earthquake Center,

http://pubs.usgs.gov/of/2007/1437/, 2007.

[19] H. Kanamori and D. L. Anderson. Theoretical basis of some emperical relations in seismology. *Bull. Seis. Soc. Am.*, 65(5):1073–1095, October 1975.

[20] D. L. Wells and K. J. Coppersmith. New empirical relationships among magnitude, rupture length, rupture width, rupture area, and surface displacement. *Bull. Seis. Soc. Am.*, 84(4):974–1002„ 1994.

[21] Christoph Bandt and Bernd Pompe. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.*, 88:174102, Apr 2002.

[22] David P. Feldman and James P. Crutchfield. Structural information in two-dimensional patterns: Entropy convergence and excess entropy. *Phys. Rev. E*, 67:051104, 2003.