

Informational Analysis of Invasion Percolation Models of Hydraulic Fracturing

J. Quinn Norris

Physics Department University of California, Davis

jqnorris@ucdavis.edu

ABSTRACT: Recent developments in hydraulic fracturing have enabled the recovery of large reserves of natural gas and oil. These developments have raised concerns over the safety of the process, in particular its potential to generate large earthquakes and its potential to contaminate drinking water. The lack of publicly available research led us to the development of a simple model for fracking based on invasion percolation. Several information theoretic quantities for this model were calculated using Bayesian inference and permutation entropy. We find that the information quantities calculated for both the strength and spatial sequences of invaded bonds are consistent with fairly random processes.

Introduction

Hydraulic fracturing, commonly known as fracking, has recently become one of the most controversial environment issues in the United States. Hydraulic fracturing is just one of a number of techniques, including gas injection and waterflooding, used to enhance oil and gas recovery from petroleum reservoirs [1,2]. It is a fairly old process being developed and commercially deployed in the late 1940's, and has been regularly employed by oil companies since its development. The recent controversy has arisen over new developments which have greatly changed the process to allow oil and gas extraction from previously unrecoverable reservoirs. We will refer to the older, conventional method of fracking as traditional fracking, and the modern controversial method as superfracking.

While there has been a great deal of research on fracking [3], the vast majority of this research is proprietary and unavailable to the public. The lack of publicly available research makes it impossible for concerned citizens and politicians to make informed decisions regarding the benefits and potential risks of fracking. In light of this need for more publicly available research, we developed a simple cluster growth model based upon percolation theory that can be used to understand the basic statistical properties of superfracking. This model has been previously analyzed using the traditional methods of percolation theory and branching networks [4].

Because the traditional methods of percolation theory and branching networks were developed for static systems, they only yield an understanding of the structural properties of the model. These methods fail to capture the evolution of the cluster over time. In particular, they completely neglect the roles information and information processing play in the growth of the cluster. In this project, we use the framework of computational mechanics as outlined by Shalizi and Crutchfield [5] to determine the involvement of information and information processing in the growth of the cluster. For simplicity, we reduce the system to a one-dimensional time series. Using two different encodings, we produce binary strings representing the growth of the cluster. From these strings, we calculate several information theoretic values using the python module CMPy developed by the Computational Mechanics Group at UC Davis. Additionally, we compared these results to results obtained using permutation entropy.

Background

Hydraulic Fracturing

In general, the process of fracking is as follows. The pressure in a small region of a bore hole is increased until cracks form in the surrounding rock. These cracks enhance the natural permeability of the reservoir allowing for greater oil and gas extraction. To retain the enhanced permeability after the pressure is reduced, large volumes of "proppant" (generally sand) are added to keep the fractures "open". In order to understand the controversy behind superfracking, it is important to understand how it differs from traditional fracking. Traditional fracking typically uses 75-300 m³ of water. Guar gum or hydroxyethyl cellulose is added to *increase* the fluid viscosity. Traditional fracking generates a single fracture from the point of injection. The single fracture provides a pathway for oil and/or gas to migrate to the well. Traditional fracking is applied to granular reservoirs (say sandstones) with natural permeabilities in the range $k = 10^{-14}$ to 10^{-12} m². In these reservoirs, the natural permeability of the rock allows oil and/or gas to migrate to the single large fracture and then into the well. It is estimated that 90% of the producing wells in the United States have experienced traditional fracking.

In contrast, a typical superfrack uses 7.5×10^3 m³ to 11×10^3 m³ of water. Superfracking is used to extract oil and gas from older shales with natural permeabilities in the range $k = 10^{-19}$ to 10^{-21} m². Before superfracking, production of oil and/or gas from these rocks was not economically feasible. Extraction of large quantities of oil and gas from shale has been made possible by two primary technological developments. The first is horizontal drilling. This allows a single well to extract oil and gas from a much greater area, especially when the producing zone is flat lying and is relatively thin. The second development is the use of low viscosity fluid to produce distributed damage (arrays of cracks) to generate high permeability in the reservoir rock. Chemical additives such as potassium chloride and mineral oil are added to *decrease* the viscosity of the injection fluid. Fracking fluids with decreased viscosities are often called "slickwater".

In just the past 10 years or so super fracking techniques have been used to successfully extract large quantities of gas and oil from shale. The Barnett shale in Texas has 6600 producing gas wells, more than 90% drilled since 2000. Oil production from the Bakken shale in North Dakota increased from less than 1 million barrels per year in 2005 to 12 million barrels per year in 2011. The only state that produces more oil than North Dakota today is Texas. It is estimated that by 2040 more than half of all the natural gas produced in the United States will come from shale reservoirs [6].

With this increase in oil and gas production have come concerns over the safety and sustainability of fracking. Opponents of fracking typically cite two main concerns: induced earthquakes and contaminated drinking water. It has long been known that injecting fluids underground can produce earthquakes [7]. Also, it is well accepted that the increased pressures and rock damage associated with fracking has a negligible effect on the earthquake rate and intensity in the vicinity of the well. However, it has been shown that large earthquakes can result from reinjection of large volumes of fluid [8]. The maximum magnitude of these induced earthquakes has been directly related to volumes of the fluid injected [9]. Additionally, there have been elevated levels of methane detected in drinking water near hydraulic fracturing wells [10]. These findings are the main source of the controversy surrounding fracking.

Information Theoretic Measures

For completeness, we have included both a formal definition and a general interpretation for each of the information theoretic measures calculated for our model. We will use the notation and definitions given by Cover and Thomas [11], and Crutchfield and Feldman [12]. Except where noted, the measures are assumed to be over a discrete alphabet. It is important to note that for the expressions given, stationarity and ergodicity of the process is assumed.

$$\text{Entropy Rate: } h_\mu = H(\chi) = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) = \lim_{n \rightarrow \infty} H(X_n | X_{n-1}, X_{n-2}, \dots, X_1)$$

From these expressions two natural interpretations arise. The first expression naturally leads to the interpretation of

the average uncertainty per symbol of a given infinite symbol sequence. The second expression indicates that the entropy rate of a process is the uncertainty in the next symbol, given an infinite past. For stationary processes the two definitions and interpretations are equivalent.

$$\text{Excess Entropy: } \mathbf{E} = \sum_{L=1}^{\infty} [h_{\mu}(L) - h_{\mu}] = \lim_{L \rightarrow \infty} I[S_0, S_1 \dots S_{L-1}; S_L, S_{L+1}, \dots S_{2L+1}]$$

The excess entropy can be interpreted as the total sum of the difference between the L length estimate of the entropy rate and the true entropy rate of the symbol sequence. This difference is simply the redundancy and the excess entropy can be thought of as the total redundancy of the symbol sequence. The second interpretation is that of the mutual information between an infinite past and an infinite future. If viewed as a communication channel, the excess entropy is the amount of information from the past that is communicated to the future.

Following Abdullah and Plumbley [13], the entropy rate can be decomposed into two components ($h_{\mu} = b_{\mu} + r_{\mu}$) depending on whether the uncertainty in the present symbol (X_0) has an impact on future symbols ($X_{n>0}$) as shown in Figure 1.

$$\text{Predictive/Bound Information Rate: } b_{\mu} = \lim_{n \rightarrow \infty} \frac{1}{n} \left\{ H[X_{0:n}] - \sum_{A \in P(n); |A|=1} H[X_A | X_{\bar{A}}] \right\}$$

From Figure 1 we see that the predictive/bound information rate is the randomness in the present that does impact the future. It is the current information that did not come from the past but is communicated to the future.

$$\text{Residual/Ephemeral Entropy Rate: } r_{\mu} = \lim_{n \rightarrow \infty} \frac{1}{n} \left\{ \sum_{A \in P(n); |A|=1} H[X_A | X_{\bar{A}}] \right\}$$

From Figure 1 we can see that the residual/ephemeral entropy rate is the randomness in the present that does not impact the future or come from the past. It is current information that is not received from the past and is not communicated to the future.

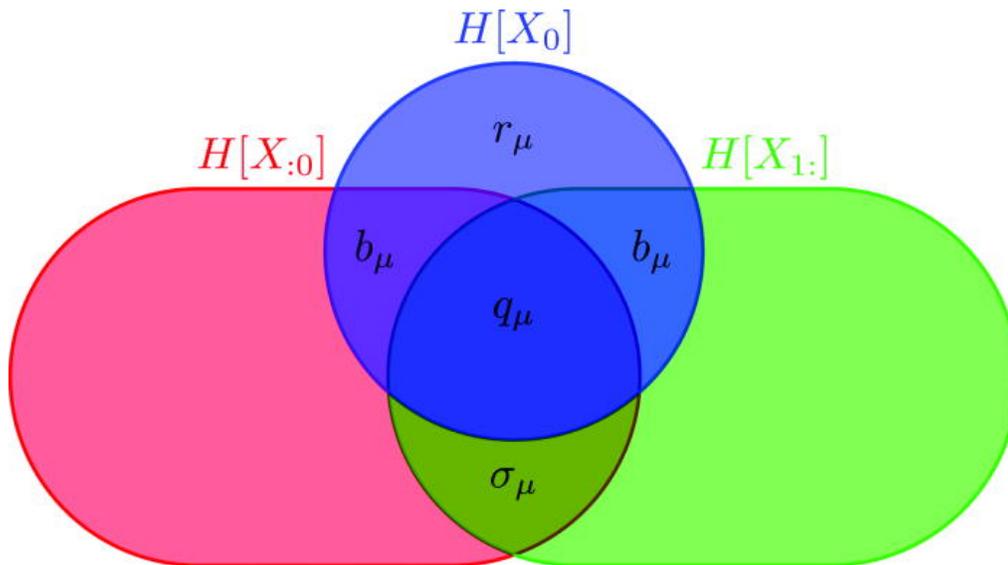


FIG. 1: Decomposition of entropy rate ($h_\mu = b_\mu + r_\mu$) into predictive/bound information rate (b_μ) and residual/ephemeral entropy rate (r_μ) using the equivalence between information and set theory. Figure taken from James et. al [14]

Within the framework of computational mechanics [5] we can compute measures of the computational complexity of the process by using causal states.

$$\textbf{Statistical Complexity: } C_\mu(\mathbf{S}) = - \sum_{S \in \mathbf{S}} \Pr(S) \log_2 \Pr(S)$$

From the expression we see that the statistical complexity is simply the Shannon entropy of the causal states. A more practical interpretation is the amount of stored information or the memory of a process. It is the amount of information required to optimally predict. It is important to note that $\mathbf{E} \leq C_\mu$, so there can be structure that is not apparent in the symbol sequences. This means that the statistical complexity cannot be calculated directly from a symbol sequences and requires a model. Statistical complexity is the amount of information needed to optimally predict the information measured by the excess entropy.

$$\textbf{Crypticity: } \chi = H[S_0 | X_{1:}] = C_\mu - E$$

Where the sum is over the causal states and $\Pr(S)$ is the asymptotic probability of being in causal state S . Crypticity is simply the difference between the statistical complexity and the excess entropy. It is a measure of how hidden the process is.

Up to now, we have only introduced measures over discrete alphabets. Because our fracking model uses a has a continuous range of strengths on the interval $[0, 1)$, these measures would require a digitization of the data, potentially introducing artifacts. Permutation entropy as introduced by Bandt and Pompe [15] however, allows for the computation of information quantities over continuous alphabets. The method involves considering sequence orderings rather than actual symbols. For example, the length three permutation entropy is calculated by finding the probability of having a specific ordering say (large, medium, small) or (small, large, medium). The number of orderings grows as the factorial of length meaning that much longer sequences series are typically needed for calculation.

$$\textbf{Permutation Entropy: } H^*(L) = - \sum \Pr(\pi) \log_2 \Pr(\pi)$$

Where the sum is over orderings of length L and $\Pr(\pi)$ is the probability of a particular ordering π . Permutation entropy is the information contained in comparing L consecutive values at a time. It is useful to look at the permutation entropy per symbol.

$$\textbf{Permutation Entropy Rate: } h_L^* = \lim_{L \rightarrow \infty} \frac{H(L)}{L - 1}$$

This is simply the permutation entropy per symbol which mathematically is identical to the formula for entropy rate except L is replaced by $L - 1$ because there is only a single ordering of length one. It has been shown that for stationary, ergodic, finite-alphabet processes the entropy rate and permutation entropy rate are equal [16]. Under similar assumptions, it was also shown that permutation excess entropy is equal to the excess entropy [17].

Dynamical System

The original basis for our fracking model is percolation theory. Percolation theory involves the study of the connectivity between sites in a lattice. Lattices in two and higher dimensions have been studied using a variety of lattice geometries (square, triangular, etc.). The probability p that a site is occupied is specified. If p is sufficiently high, a cluster of connected sites spans the lattice. As the lattice size goes to infinity the minimal value of p associated with a spanning cluster is a critical point. In the vicinity of this critical point ($p = p_c$), a variety of power-law scalings have been found. One example is the power-law scaling of the number of clusters as a function of cluster area. An alternative formulation utilized a lattice of bonds rather than sites. The probability p is the probability that a bond is present. Criticality and scaling laws for bond percolation are similar to those for site percolation. Details of percolation theory have been given by Stauffer and Aharony [18].

A standard procedure in oil production is to inject a fluid (typically water) in injection wells in order to drive the oil to production wells. In order to model this process the invasion percolation model was introduced [19]. In this model all sites were initially occupied by a defender fluid, for example oil. An invader fluid, for example water, is injected along one side of the region under consideration; typically, this region is a layer of width L . In the bond version of invasion percolation the pathways are assigned random strengths. A random number p is drawn from a uniform distribution in the range $[0, 1)$. This random number p determines the relative strength of each pathway. At each time step, the invading fluid flows through the weakest pathway (smallest p) occupying the adjacent pore. Assuming the fluid to be incompressible, the defending fluid in these pores is lost through the upper boundary (i.e. oil to a producing well).

Our invasion percolation model is an extension of the model given by Wilkinson and Barsony [20]. These authors considered invasion percolation of sites from a point source in two and three dimensions. The result was a single growing cluster of invaded sites. Each site is assigned a random number p , $0 < p < 1$. At each time step, the adjacent site with the smallest p is selected.

To illustrate our model we consider a two-dimensional square grid of sites as illustrated in Figure 1. The sites are connected by bonds. The sites are considered to be porosity which the injecting fluid will fill. The bonds are the pathways for fluid, which are initially closed (undamaged). If fluid injection is relatively slow, the injection is resisted primarily by capillary forces rather than viscous forces. The invasion percolation model neglects pressure drops associated with viscous flow. We will consider other geometries and our model is easily extended to three dimensions.

Our two-dimensional model is illustrated in Figure 2. Fluid is introduced at the central site $(0,0)$. The central site is linked to the four neighboring sites $(0,1)$, $(1,0)$, $(0,-1)$ and $(-1,0)$. These bonds are given random numbers in the range $[0, 1]$. The weakest bond (smallest r) opens and fluid flows through it into the adjacent site. In the example illustrated in Figure 1 the weakest bond is $(0,0)$ to $(1,0)$ and fluid fills site $(1,0)$. This site is now linked to three neighboring sites $(1, 1)$, $(2, 0)$, $(1, \hat{a}'1)$. These links are given random numbers r . On the next time step fluid is allowed to flow through the weakest (smallest r) of the six available bonds, bond $(1, 0)$ to $(1, 1)$ for the example illustrated. Site $(1, 1)$ is now linked to two neighboring sites $(0, 0)$ and $(1, 1)$, the new neighboring sites are given random numbers making a total of eight available bonds. On the final time step in our example the weakest bond is $(0, 0)$ to $(0, 1)$ and site $(0, 1)$ is filled. Two new nearest neighbor bonds are added but the bond $(0, 1)$ to $(1, 1)$ is removed because both sites are filled. This process is continued to form a single connected cluster. At each time step, an occupied site is added to the growing cluster.

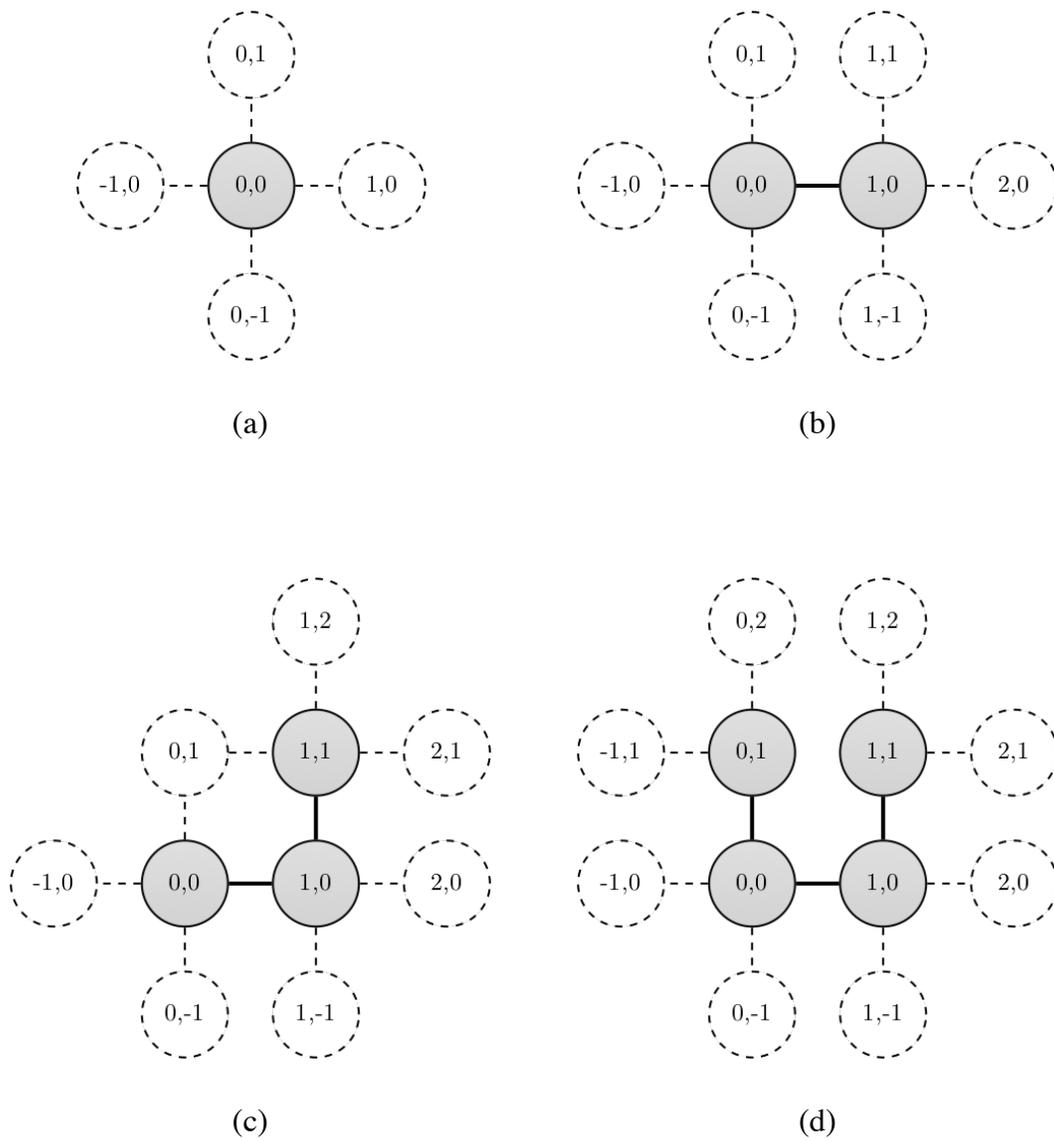


FIG. 2: Illustration of our invasion percolation model. Occupied sites and damaged bonds are shown in solid, available sites and undamaged bonds are dashed. The weakest bond (lowest p) between an occupied and an adjacent unoccupied site is damaged and the neighboring site is occupied. This process repeats. If at anytime there is an undamaged bond between two occupied sites, the bond is removed. A sequence of three time steps is shown in $I(a)$ to $I(d)$.

Methods

In the spirit of return maps, we reduced the dimensionality of the system by considering just the time series of p values. While there is not a single function in the return map there does appear to be some structure as shown in Figure 3. We note that because the pseudorandom number generator used has a period much larger than the size of the lattice, the bond strength uniquely determines the bond's location within the lattice. This would allow us to reconstruct the spatial lattice once the properties of the single variable time series are known to look for spatial patterns.

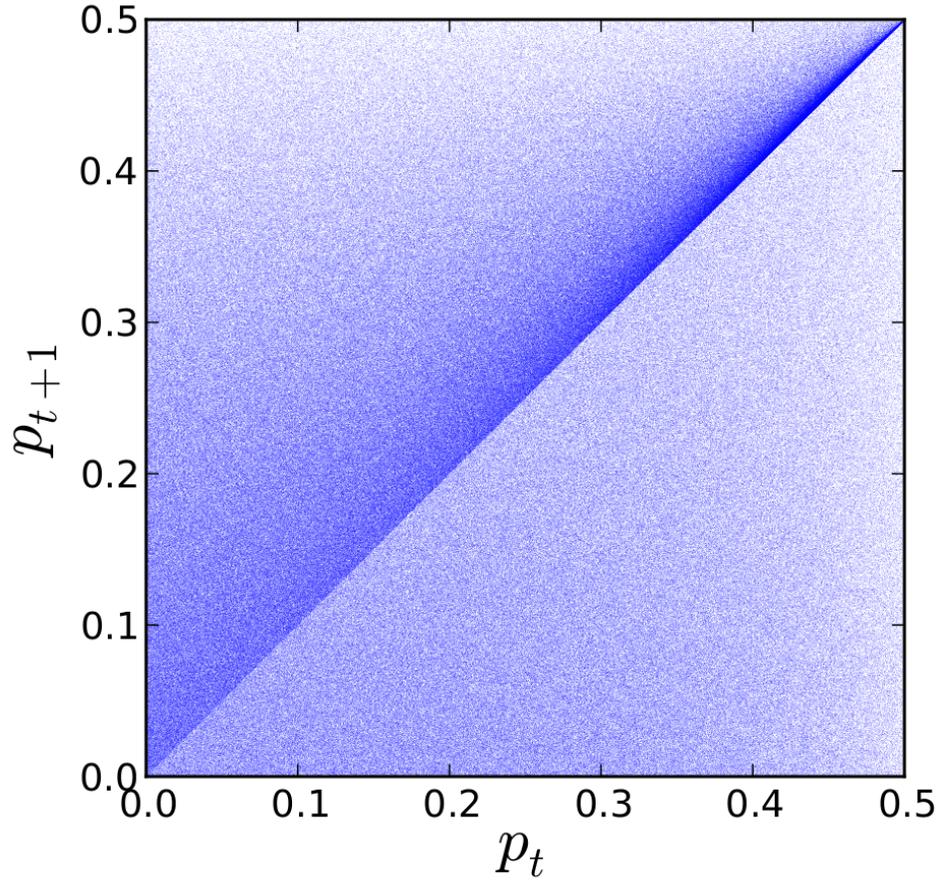


FIG. 3: Return map of the time series of broken bond strengths (p) for our modified invasion percolation model.

Because cascading failures, avalanches, earthquakes, bursts, etc. are common to many growth models, we wanted to calculate to what extent the next failed site could be predicted from previous failed sites. For simplicity, the times series of p values were converted into a binary string using the following encoding:

$$\text{StrengthEncoding} : x_t = \begin{cases} 1 & : p_{t+1} = p_t \\ 0 & : p_{t+1} < p_t \end{cases}$$

This sequence encodes whether the strength of the next failed site will be higher or lower than the previous failed site, giving a very coarse encoding which can be used for prediction.

To partially capture the spatial growth patterns, we looked at the coordinates of failed bonds and encoded instances where the next failed site was a nearest neighbor of the previous failed bond as a one and zero otherwise.

$$\text{SpatialEncoding} : y_t = \begin{cases} 1 & b_{t+1} \text{ is n.n. of } b_t \\ 0 & b_{t+1} \text{ is not n.n. of } b_t \end{cases}$$

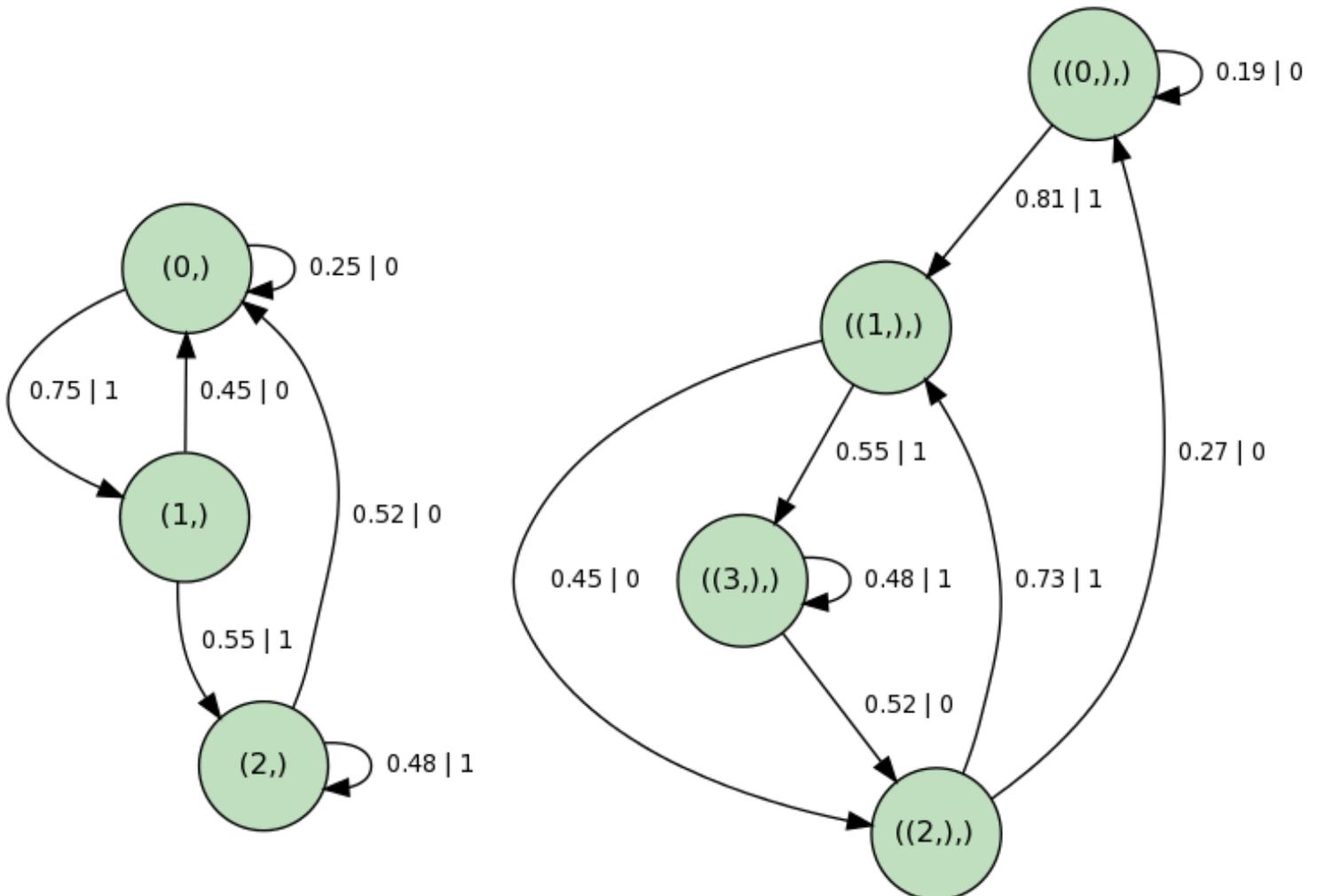
This encoding can be used to discriminate between fingering and uniform growth.

A single run of one million times steps was converted into two binary sequences using these encodings. These

sequences were converted into epsilon machines using Bayesian inference. We used the methods developed by Strelieff and Crutchfield [21] which have been incorporated into the Python module CMPy. The Bayesian inference algorithm takes a prior over a set of epsilon machines, a parameter β which is used to compute a cost function for larger epsilon machines, and the symbol sequence. The algorithm returns a posterior distribution over the give set of epsilon machines. In come cases the epsilon machine does not allow the given symbol sequence and the posterior probability of that machine is identically zero. Using the suggested value of $\beta = 4$, we used two different priors for both the strength and spatial encoding. The two priors used used are 1. All binary epsilon machines with 3 or fewer causal states and 2. All binary epsilon machines with 4 or fewer causal states. For the prior of all binary epsilon machines with 4 or fewer causal states, a single run of 100 thousand was used to obtain a posterior distribution over epsilon machines. From this posterior distribution, we selected only machines with significant probability (> 0.01) to use as a prior for the one million symbol sequence. From the final posterior distribution over epsilon machines, we calculated information quantities for both sequences using built-in CMPy routines. Additionally, permutation entropies were calculated for a different single run of 15 million time steps. The additional time steps were necessary to obtain accurate permutation entropies.

Results

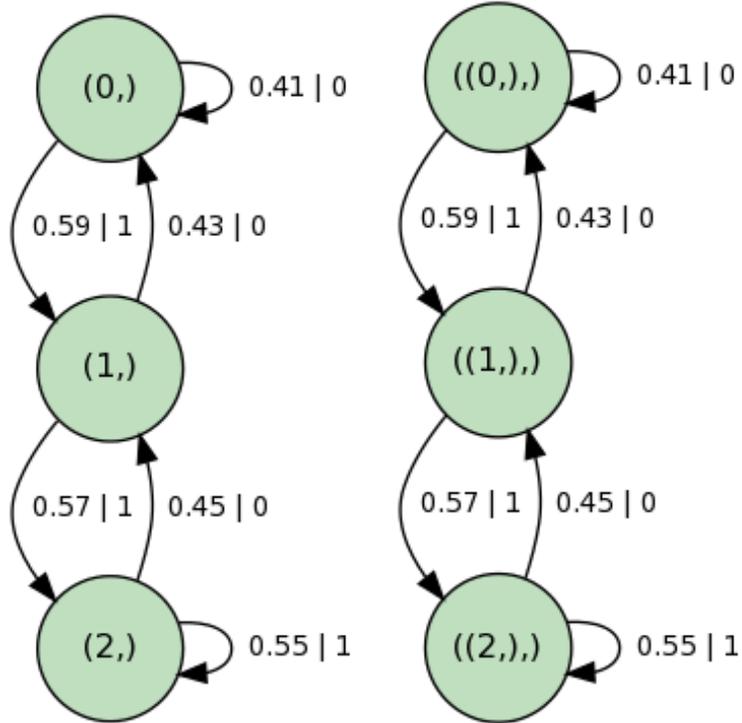
For the two different priors, the Bayseian algorithm returned a single epsilon machine with probability of 1 to machine precision for the strength encoding and .98 for the spatial encoding with the next highest probability being .0079. With such high confidence, we chose to use just the most probable epsilon machine as a posterior to calculate the information quantities of the sequences. The inferred epsilon machines are shown in Figures 4 and 5.



(a)

(b)

FIG. 4: Inferred epsilon machines for the strength encoding using a prior over (a) all binary epsilon machines with 3 or fewer causal states, and (b) all binary epsilon machines with 4 or fewer causal states.



(a)

(b)

FIG. 5: Inferred epsilon machines for the spatial encoding using a prior over (a) all binary epsilon machines with 3 or fewer causal states, and (b) all binary epsilon machines with 4 or fewer causal states.

Using the epsilon machines shown in Figures 4 and 5 we calculated the information quantities shown in Figures 6 and 7. Note that while there are only two decimal places shown for the transition probabilities in Figures 4 and 5, higher precision values are used for the quantities in Figures 6 and 7.

	3-State	4-State
Entropy Rate	0.924860193657	0.923061529804
Excess Entropy	0.044375314673	0.047973526459
Predictive/Bound Information Rate	0.043980220838	0.047228342944
Residual/Ephemeral Entropy Rate	0.880879972819	0.87583318686
Statistical Complexity	1.5731924104	1.89409658607
Crypticity	1.52881709573	1.84612305961

FIG. 6: Information theoretic quantities for the strength encoding for the 3 and 4 state machines shown in Figure 4.

	3(4)-State
Entropy Rate	0.98748207450
Excess Entropy	0.77611849505
Predictive/Bound Information Rate	0.00083479806
Residual/Ephemeral Entropy Rate	0.98664727644
Statistical Complexity	1.5522369901
Crypticity	0.7761184950

FIG. 7: Information theoretic quantities for the spatial encoding for the 3 state machines shown in Figure 5. The same machine was also returned when using all epsilon machines with 4 states or fewer as a prior in the Bayesian inference algorithm.

We have calculated both the block permutation entropy and block permutation entropy rate up to blocks of length 9. Beyond that there are not enough data points to accurately sample the permutations—fewer than 10 samples per permutation. For comparison we have plotted both the block permutation entropy and the block permutation entropy for the strength values of our model along with the same quantities for a standard pseudorandom number generator in Figures 8 and 9.

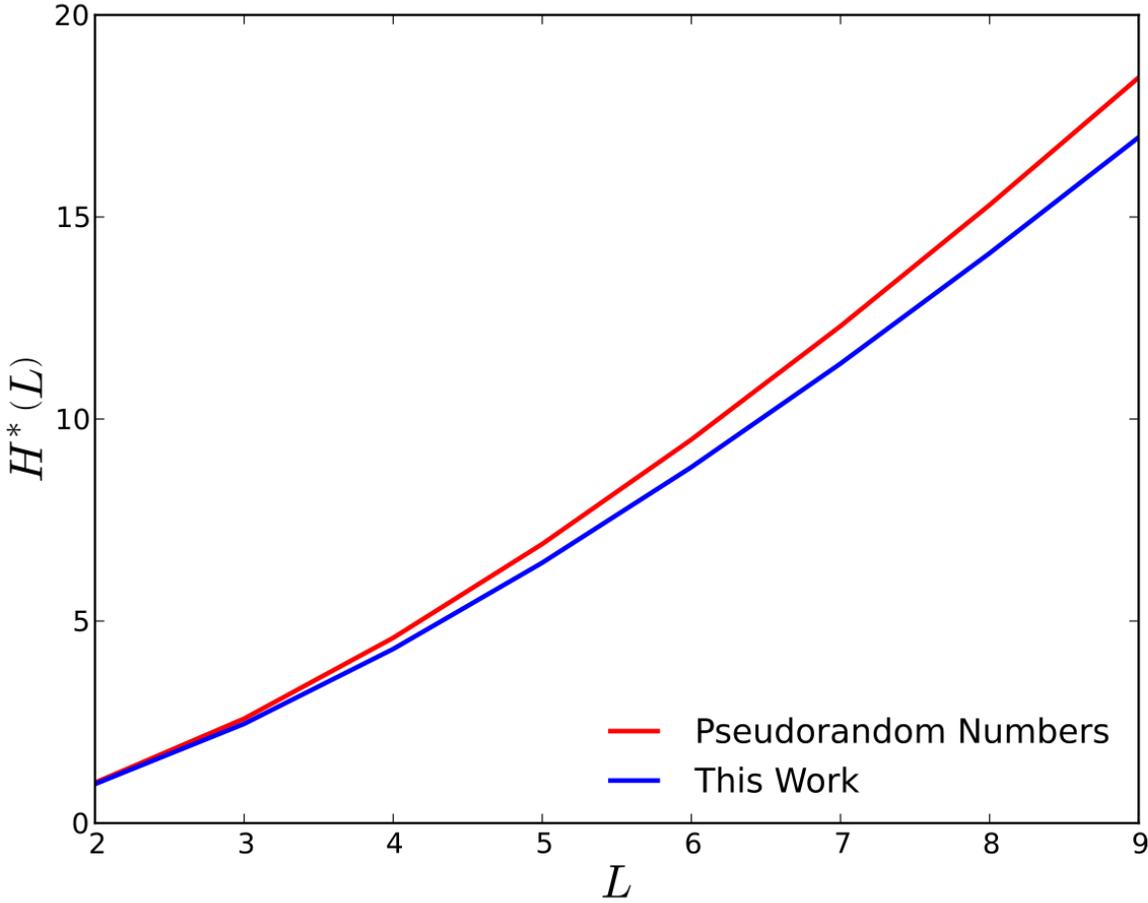


FIG. 8: Return map of the time series of broken bond strengths (p) for our modified invasion percolation model.

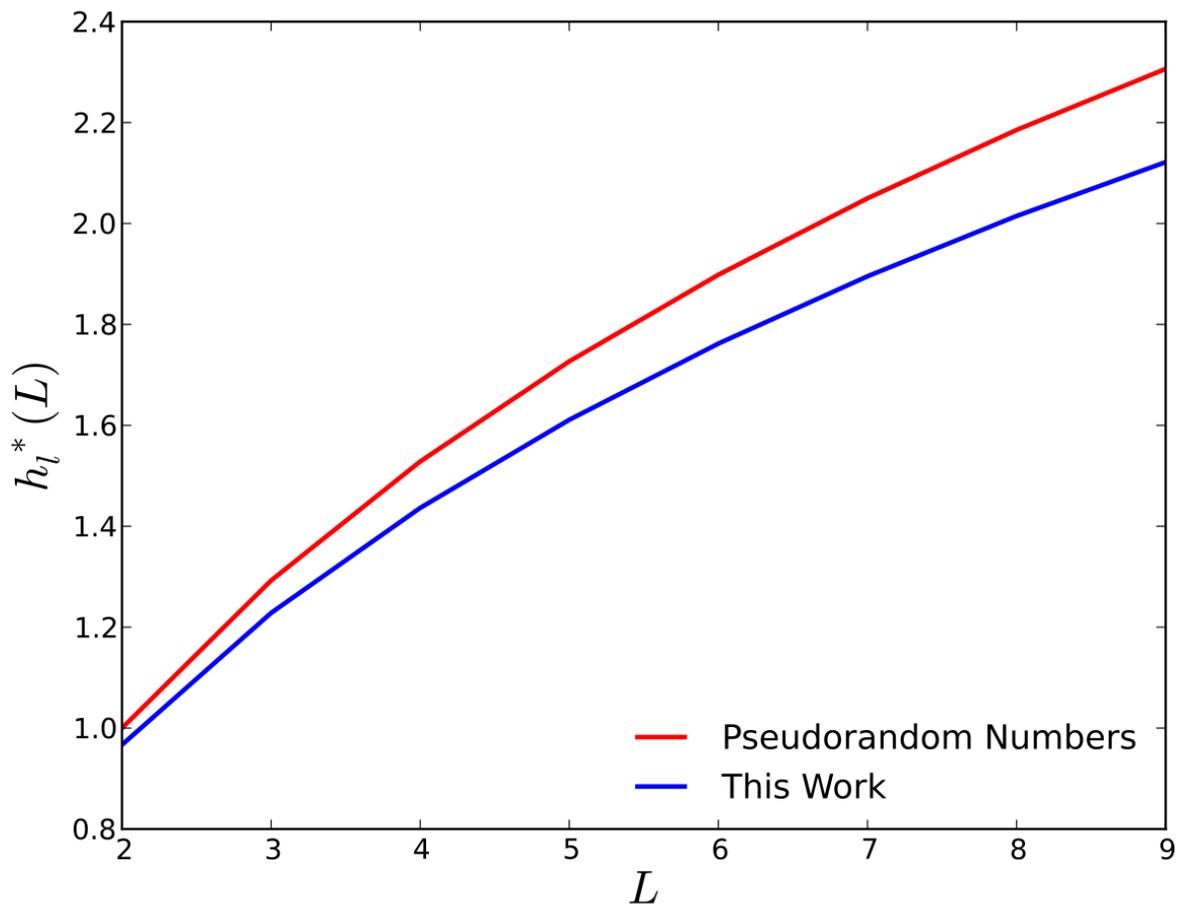


FIG. 9: Return map of the time series of broken bond strengths (p) for our modified invasion percolation model.

Conclusion

Overall, both Bayesian inference and permutation entropy produced information theoretic values consistent with a highly random process. For the strength encoding, the Bayesian inference algorithm returned similar epsilon

machines for both the 3 or fewer state and 4 or fewer state priors. The 4 state epsilon machine is just a refinement of the 3 state machine. States 1 and 2 in the 3 state machine are equivalent (have exiting transitions with the same probabilities) to states 1 and 3 respectively in the 4 state machine. State 0 in the 3 state machine has been refined to states 0 and 2 in the four state machine. The information measures for each machine are very similar except for the statistical complexity and the crypticity which depend directly on the model size. The entropy rate near maximal for a binary machine indicating that the process is highly random. The low excess entropy indicates that there is little correlation between the past and the future which is also reflected in a low predictive/bound information rate.

This highly random nature of the strength encoding agrees well with the permutation entropy results. Both the block permutation entropy and block permutation entropy rates have a shape similar to the pseudo-random number generator and show only small differences at large block lengths. This suggests that the entropy of the process is close to random. The strength encoding retains this randomness as shown in the information measures.

The spatial encoding returned identical machines for both the 3 and 4 state priors. This suggests that this epsilon machine is a good representation of the process, or that there is very little additional information gained by looking at length 4 words. The entropy rate for this encoding was very high, approaching a fair coin. It is interesting however that the excess entropy is higher for the spatial encoding than for the strength encoding indicating that there is structure in the randomness. The almost vanishing predictive/bound information rate indicates that a single symbol has very little impact on the future of this process. It is interesting to note that the crypticity is identical to the excess entropy for this process. In some sense this means the process displays exactly as much structure as it hides which could have interesting interpretations.

The epsilon machine gives a fairly straight forward description of the process. At any given time the next symbol is determined by a biased coin. The degree of the bias depends on the previous symbols. If the process produces two or more 1's in a row the bias is minimal. If the process produces two or more 0's in a row the bias is maximal. If the process produces a 0 the bias increases. If the process produces a 1 the bias decreases. In a limited way, this can be viewed as a biased coin that reduces the probability of long sequence of repeated symbols. If you see one symbol, the bias shifts so you are less likely to see that symbol again.

It is uncertain to what extent this encoding actually captures the spatial growth of the model. One way to check this encoding would be to perform an Eden growth simulation subject to this epsilon machine. When a new bond is added, the epsilon machine produces a random number. If the result is a 1, the next bond added must be a nearest neighbor. If the result is a 0, the next bond is selected randomly from the available bonds. The fractal dimension and branching structure of the resulting cluster could be compared to the original model to check for agreement.

One possible explanation for the apparent randomness might be the simple result of assuming stationarity for a non-stationary process. Future work could examine, to what extent the process is stationary. If the process is found to be non-stationary, the information measures would need to be extended to handle cases of non-stationarity. Another possibility is that this process contains very long range correlations that cannot be captured by small (3 and 4 state) epsilon machines. However, the permutation entropy also seems to indicate a very random process.

It is interesting to note that while this process appears very random from an information theoretic standpoint, it was shown to produce a power-law distribution of bursts [4], similar to earthquakes, avalanches and wildfires. This indicates that power-law distributions can arise from very random looking processes. Additionally, the strength encoding had a very low excess entropy meaning that the future is fairly independent of the past. This could be very significant in the area of earthquake forecasting where historical records and power-law distributions form the core of most forecasting models and might explain why these forecasting are unable to predict earthquakes. It also indicates that the methods used in this project might be able to detect the existence of power-law distributions.

References

[1] E. Fjaer, R. M. Holt, P. Horsrud, A. M. Raaen, R. Risnes, Petroleum Related Rock Mechanics, 2nd Ed., Elsevier, Amsterdam, 2008.

- [2] C. H. Yew, *Mechanics of Hydraulic Fracturing*, Gulf Pub. Co., Houston, Tex., 1997.
- [3] S. Maxwell, Microseismic hydraulic fracture imaging: The path toward optimizing shale gas production, *The Leading Edge* 30 (3) (2011) 340-346. doi:10.1190/1.3567266.
- [4] J. Q. Norris, D. L. Turcott, M. R. Yoder, J. B. Rundle, (In progress).
- [5] C. R. Shalizi and J. P. Crutchfield, Computational Mechanics: Pattern and Prediction, Structure and Simplicity, *Journal of Statistical Physics* 104 (2001) 817-879. Santa Fe Institute Working Paper 99-07-044.
- [6] Annual Energy Outlook 2013. Washington, DC. 2013 Retrieved from www.eia.gov/forecasts/aeo
- [7] C. B. Raleigh, J. H. Healy, J. D. Bredehoeft, An Experiment in Earthquake Control at Rangely, Colorado. *Science*, 191(4233), 1230-1237. doi:10.1126/science.191.4233.1230
- [8] R. A. Kerr, Learning How to NOT Make Your Own Earthquakes. *Science*, 335(6075), 1436-1437. doi:10.1126/science.335.6075.1436
- [9] E. Balcerak, Linking earthquakes and hydraulic fracturing operations. *Eos, Transactions American Geophysical Union*, 94(1), 3. doi:10.1002/2013EO010005
- [10] S. G. Osborn, A. Vengosh, N. R. Warner, R. B. Jackson, Methane contamination of drinking water accompanying gas-well drilling and hydraulic fracturing, *Proceedings of the National Academy of Sciences of the United States of America*, 108(20), 8172-6. doi:10.1073/pnas.1100682108
- [11] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, 2nd Ed, John Wiley & Sons, Inc., Hoboken, NJ, 2006.
- [12] J. P. Crutchfield, D. P. Feldman, Regularities unseen, randomness observed: Levels of entropy convergence, *Chaos*, 13(1), 25-54. doi:10.1063/1.1530990
- [13] S. A. Abdallah, M. D. Plumbley, A measure of statistical complexity based on predictive information, 2010, arXiv:1012.1890.
- [14] R. G. James, C. J. Ellison, J. P. Crutchfield, Anatomy of a bit: Information in a time series observation. *Chaos*, 21(3), 37109. doi:10.1063/1.3637494
- [15] C. Bandt, B. Pompe, Permutation Entropy: A Natural Complexity Measure for Time Series. *Physical Review Letters*, 88(17), 174102. doi:10.1103/PhysRevLett.88.174102
- [16] J. M. Amigó, M. B. Kennel, L. Kocarev, The permutation entropy rate equals the metric entropy rate for ergodic information sources and ergodic dynamical systems. *Physica D*, 210(1-2), 77-95. doi:10.1016/j.physd.2005.07.006
- [17] T. Haruna, K. Nakajima, Permutation complexity via duality between values and orderings. *Physica D*, 240(17), 1370-1377. doi:10.1016/j.physd.2011.05.019
- [18] D. Stauffer, A. Aharony, *Introduction to percolation theory*, Rev. 2nd ed. Taylor & Francis, New York, 1994.
- [19] D. Wilkinson, J. F. Willemsen, Invasion percolation: a new form of percolation theory, *Journal of Physics A*, 16(14), 3365-3376. doi:10.1088/0305-4470/16/14/028
- [20] D. Wilkinson, M. Barsony, Monte Carlo study of invasion percolation clusters in two and three dimensions, *Journal of Physics A*, 17(3), L129. doi:10.1088/0305-4470/17/3/007

[21] C. C. Strelhoff, J. P. Crutchfield, Optimal instruments and models for noisy chaos. *Chaos*, 17(4), 43127.
doi:10.1063/1.2818152