Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

A CELL-DEVS MODEL FOR FRACTURE PROPAGATION IN ROCK

Gabriel A. Wainer
Scott Stewart

Carleton University – Department of Systems and Computer Engineering
1125 Colonel By Drive
Ottawa, ON, K1S 5B6, CANADA

ABSTRACT

We present a cellular model to study the propagation of cracks in a given sample of rock. The model uses the fractal geometry found in real rocks and the stress and strain on an element of rock to update the element's strength. We study the propagation speed of the cracks with different initial conditions. The model shows that the more inhomogeneous a rock sample is the quicker a fracture can propagate through it. Because fractures release a large amount of energy when they grow, earthquakes are more likely to occur in these situations.

1 INTRODUCTION

Earthquakes are one of the most unpredictable and destructive natural disasters known. Although years of studying earthquakes through empirical observation have yielded several well-known laws (such as the Gutenberg-Richter law (Gutenberg and Richter, 1954), it is very difficult to predict individual earthquakes. Most of the early work in studying earthquakes involved using seismographs to measure the intensity pattern throughout a given region (Ammon, 1997), and would later move to experimenting on samples of rock from a region to better understand the mechanisms of an earthquake.

Because performing experiments on actual rock samples is both time consuming and costly, it became advantageous to use computer simulation of earthquakes – as these simulations are flexible and cost effective because a "sample" can be changed by providing a new set of values and the experimental conditions can be easily modified. Although several earthquake libraries have been built (an extensive list ca be found here: https://nisee.berkeley.edu/elibrary/software.html), due to the complex nature of the calculations used in such simulations, these programs are very time consuming for even simple simulations and are expensive for the more up-to-date models.

In recent years, Cellular models started to being used to deal with these issues. This is a good approach, because earthquakes tend to follow regions and paths of weak rock (referred to as *geological faults*), and, as a small element of rock responds to stress in a well-defined manner (Brune 1970), a set of rules can be derived for use in constructing a cellular model dependent on the strength and/or stress of the rock sample. Here, we proose a cellular model built using Cell-DEVS (Cellular Discrete Event Systems Specifications), a well-defined framework for modeling and simulating discrete event cellular models (Wainer 2009). We discuss the use of these models, and different tools to properly capture the behaviour of the crack propagation through a sample of rock, respecting such aspects as the stress and strain that the rock experiences. The models considers that rocks have different compressive strength depending on what they are made of.

2 BACKGROUND

2.1 Earthquake propagation

A *fault* is defined as a planar fracture or discontinuity found in rock, which is generally manifested as some form of displacement of the material as a result of rock mass movement (Park, 1997). On a large scale, faults can be generated within the Earth's crust through plate tectonic forces, where two tectonic plates would interact with each other and cause cracks along the boundaries between the plates. In general, though, faults are generated by the friction between the rocks in a given volume. Because rocks are rigid and there is friction between them, it is very difficult for them to flow past each other, which causes movement to stop. The forces acting upon the rocks cause a buildup of stress, and when the level of stress exceeds the strain threshold for the rock, it breaks up. Then, the stored potential energy is dissipated quickly along the plane that will allow the most stress to be relieved (Brune 1970). This release of energy results in the widening of a fault within the rock. Geologists describe this behaviour as stick-slip friction. A volume of rock will continuously build up stress and release the energy along the fault plane, damaging the rock over time. As long as the binding force of the rock along a fault is sufficiently strong, the elastic strain continues to build up in the rock adjacent to the fracture, which stores up even more energy. If enough energy is stored up over time, an earthquake can occur, and it propagates along the path of least resistance: the fault in the rock.

As previously mentioned, there are several laws that describe the behaviour of earthquakes. One such law is the Gutenberg-Richter law, which expresses the relationship between the magnitude and total number of earthquakes in a given region and period of time as (Gutenberg and Richter, 1954):

$$\log_{10} N = a - bM \qquad (2.1)$$

where a and b are constants, and N is the number of events with a magnitude greater than or equal to M. The constants a and b vary between regions depending on how seismically active the area is. As we can see, the law states that there are a more events with low magnitude. For instance, a value of b close to 1.0 indicates that a region is seismically active. As the law is equally valid for small or big earthquakes, it can be deduced that small earthquakes behave using the same mechanisms as larger earthquakes. Using this assumption, the mechanics behind large-scale fractures and earthquakes can be captured by using knowledge on how small rock samples respond to different stimuli.

An element of rock in a large volume responds to local stress in a predictable manner: when subjected to a strong differential stress, the rock becomes weaker and is susceptible to deformations (Brune, 1970). For example, if the top of an element of rock has a large amount of stress acting upon it but the bottom of the rock experiences no stress, the element of rock will crumble and deform. When enough of these elements of rock weaken, the rock can deform and create a geological fault. If we know the mechanisms that determine when a small element of rock becomes weak and fracture, we could model a sample of rock by discretizing its behaviour into small elements in order to model how an initial crack can slowly grow into a large fracture. This, however, requires knowledge as to how a sample of rock is structured.

2.2 Fractals

Fragments, faults, mineral deposits, oil fields, and even earthquakes have been shown (Wang 1997) to take on shapes that can be modeled using fractals, a mathematical distribution with repeating patterns that can be viewed at any scale (Mandelbrot 1983). The patterns of the cracks inside of rock samples also appear to follow this behaviour (Wang and Li 2001). If this type of distribution is understood thoroughly enough, a suitable model of a real rock formation could be used for analysis. If a pattern is the same at every scale, the fractal is considered a "self-similar pattern"; rock formations are not self-similar.

Fractals are different from other geometric figures by how they scale. For instance, if you were to double the side-length of a square, the area would be multiplied by four (two dimensions). Fractals do not follow this. If a fractal's 1D length is doubled, the area contained within the fractal is not necessarily multiplied 2n. Instead, it uses a parameter called the fractal dimension, which is a measure of how the complexity of a fractal changes when you change the scale that it is being measured at (Falconer 2003). A real-world application for this is the coastline of a landmass. The more detail you attempt to capture by using a smaller length for measurement, the larger the coastline that is recorded. This is why there tends to be a large discrepancy between different recordings. As an aside, to measure the fractal dimension for a 2D fractal, we would use squares (cubes in 3D) instead of lines. To quantify this scaling, we use the following equation

$$N \, \alpha \, \epsilon^{-D} \tag{2.2}$$

where N is the number of sticks used to measure the region, ϵ is the scaling factor (how much the measuring stick changes), and D is the fractal dimension. If we measure the fractal dimension of an area, we use squares instead of sticks. Equation (2.2) can be rearranged to get the fractal dimension:

$$-D = \frac{\log N}{\log \epsilon} \tag{2.3}$$

It is possible to generate a random discrete fractal distribution in a simulation (Turcotte 1988), however, the theory behind these formulas are outside of the scope of this paper. The steps to produce a fractal distribution are outlined in Section 4. Studies whose purpose was to observe the fractals in the surface of rocks found that the value of D is 1.89 for quartz, 2.5 for coal, and 2.82 for gravel (Wang 1997).

2.3 Crack growth mechanisms

In an ideal situation, when a piece of rock is compressed using external stress, the local stress is evenly distributed across the entire rock sample. Because there are elastic deformations and boundary effects, and most rock is not completely homogeneous, this ideal situation never occurs. The local stress intensity factor, *K* captures the interactions between the geometry of existing cracks and the mechanical properties of the rock when a remote external stress is applied (Wang 1997). When *K* is greater than the fracture toughness of the rock *Kc*, there will be a crack in the rock sample. When a crack is present in a rock sample, the local stress in the region around the crack changes. As the crack grows larger, the energy stored is released and the local stress is relieved in the rock, which slows down the rate that the crack grows. This negative feedback is defined as follows (Wang 1997):

$$K \, \alpha \, \frac{d-a}{d} (\pi a)^{\frac{1}{2}} \tag{2.4}$$

where *a* is the length of the crack and *d* is the size of the domain of stress relaxation. When two cracks are located close to each other, the local stress between the cracks increases significantly. This causes the cracks to feed off each other, increasing the rate of failure in the region and causing the rate of crack growth to increase. This positive feedback can be described by the following formula (Wang 1997):

$$K \, \alpha \, \left(c \tan\left(\frac{\pi a}{2c}\right)\right)^{\frac{1}{2}} \tag{2.5}$$

where c is the center-to-center distance between the cracks. As the local stress intensity factor is proportional to the effects of both of these feedbacks, the crack propagation will depend on how they interact in a given sample of rock.

2.4 The Cell-DEVS formalism

Cell-DEVS (Wainer 2009) is a formal modeling and simulation methodology that allows defining discrete-event cellular models with explicit timing delays. Each Cell-DEVS atomic cell holds state variables and a computing function to update the cell's state. This is done by using the present cell state and those of a finite set of nearby cells (called neighborhood). The efficient computation of cell-state variations allows one for developing complex models, and it provides straightforward integration of the models with other modeling formalisms.

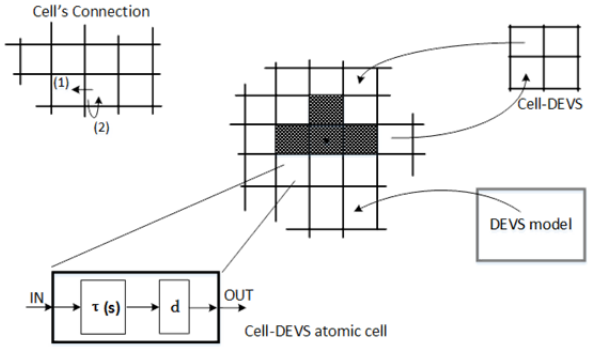

Figure 1: Definition of Cell-DEVS (Wainer 2009).

Figure 1 describes the basic concept of Cell-DEVS. A Cell-DEVS atomic cell is defined formally as:

$$TDC = <X, Y, N, delay, d, \delta_{int}, \delta_{ext}, \tau, \lambda, D>$$

where X is a set of external input events, Y is a set of external output events, N is a set of inputs, *delay* is a type of delay, d is the delay value for the cell, δ_{int}, δ_{ext} are the internal and external transition functions, τ is a local computing function, λ is the output function, and D is the lifetime function of the state. A Cell-DEVS atomic model uses a set of inputs to compute its future state from state set using the local computing function $\tau(N)$. These results are transmitted after a delay d (with different semantics for the delay function). After the behavior of a cell is defined, a coupled model is defined to integrate the atomic models representing a cell space, which is formally defined as:

$$GCC = <Xlist, Ylist, X, Y, n, \{t1 \ldots tn\}, N, C, B, Z>$$

where *Xlist* and *Ylist* are the input and output coupling lists, X and Y are the sets of external input and output events, n is the dimension of the cell space and $\{t1 \ldots tn\}$ is the number of cells in each of the dimensions. N is the neighborhood set, C is the cell space, B is the border cells set, and Z is the translation function. A Cell-DEVS coupled model is an array of atomic cells, each connected to a set of neighboring cells and potentially to other external DEVS or Cell-DEVS models.

CD++ (Wainer 2009) is an open source M&S tool that provides a development environment for implementing Cell-DEVS models using a built-in specification language based on the formal specifications of Cell-DEVS, including the size and dimension of the cell space, borders and the shape of the neighborhood. The cell's local computing function is defined using a set of rules with the form POSTCONDITION ASSIGNMENTS DELAY {PRECONDITION}. When the PRECONDITION is satisfied, the state of the cell will change to the designated POSTCONDITION, whose values will be transmitted to other components after the DELAY. If the precondition is false, the next rule in the list is evaluated until a rule is satisfied or there are no more rules. If model's state variables need to be modified, the ASSIGNMENTS section can be used. CD++ interprets this specification language and executes a simulation of the model.

3 CRACK PROPAGATION MODEL DEFINITION

The model has two separate phases: definition of the initial rock samples and the crack propagation phase, which is discussed in this section. We consider the eight nearest neighbours for calculations. Once the cells have been initialized with strength values, several iterations are performed updating the strength value of the cell based on the states of its nearest neighbours. Because Cell-DEVS is used to model the cells, the updates for all of the cells are calculated in parallel across the array; therefore, there is no bias in any of the directions. When a cell's strength becomes very small, the cell becomes broken. When there are many broken cells in an area, a fracture has grown. Once broken, the cell remains broken.

As discussed, both a positive and negative feedback come into effect for the propagation of cracks. As we only adjust the individual cell's strength, and adjacent cells only knows if the first cell apart is broken or not, these feedbacks are defined as a function of the number of broken neighbours. If a cell has a large number of broken neighbours, its strength is reduced, simulating the large difference in the stress on each side of the cell through positive feedback.

In (Wang 1997), a formula was presented for updating the strength of each cell based on how many of its neighbours are broken. If n represents the number of broken neighbours of a cell, its strength is updated using the following function:

$$f(n) = \begin{pmatrix} \min\left(1, \left(1 + \rho e^{-\frac{(n-4)^2}{20}}\right) e^{-\frac{n^2}{16}}\right) & for\ n = 1\ to\ 8 \\ 1 & for\ n = 0 \end{pmatrix} \quad (3.1)$$

where *f(n)* (called the *feedback function*) represents the effect of positive and negative feedbacks. The cell's strength is:

$$strength_{new} = f(n) * strength_{old} \quad (3.2)$$

ρ is a parameter that determines how strong the effect of the negative feedback is compared to the positive feedback for that rock. In order to see the effects of this parameter, **Figure 2**.a) shows the feedback function versus the number of broken neighbours for ρ = 1 from n = 0 to 8. **Figure 2**.b) shows the same range, instead for ρ = 5.

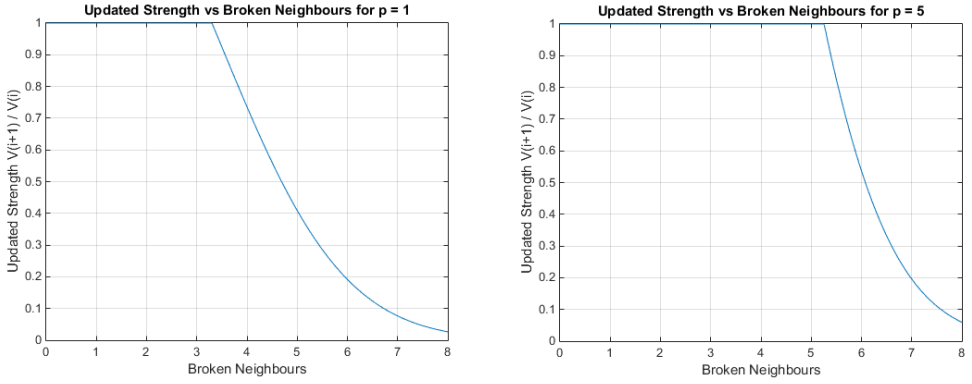

Figure 2: Feedback function/broken neighbours (a)ρ=1; (b)ρ=5 (curve fitted to number of neighbors).

These two figures show how ρ will affect the propagation behaviour of the fracture. From **Figure 2** (a), it can be seen that for ρ = 1, a given cell will not start to break down until there are four or more neighbours that are broken, because $f(n \leq 3) = 1$. **Figure 2** (b), however, shows that higher values of ρ will increase how many neighbours need to be broken in order to start decreasing the cell's strength, where ρ = 5 needs a minimum of six broken neighbours to start fracturing. This behaviour is similar for other values of ρ, and in general it means that the higher the value of ρ the more neighbours need to be broken for the crack to propagate. In essence, higher values of ρ simulate rock that is harder for fractures to propagate. By giving each cell a different ρ value and proper initialization, the model can simulate different sections of rock with different degrees of toughness.

3.1 Model implementation

The model was implemented using the CD++ toolkit and a Cell-DEVS implementation of the rules above. **Figure 3** shows a code snippet that defines the structure of the coupled model for this example, including the number of cells in the space, the type of border (non-wrapped, because the model is only looking at a single section of rock), the neighborhood shape relative to the cell, and the state variables that are used: "value" represents the strength of the cell and "rho" is the value of ρ for the cell.

```
[fracture]
type : cell     width : 256   height : 256
delay : transport    border : nowrapped
neighbors : (-1,-1) (-1,0) (-1,1)(0,-1) (0,0) (0,1) (1,-1) (1,0) (1,1)
statevariables : value rho
localtransition : fractureBehaviour

[fractureBehaviour]
rule : { $value } { $value := 0 } 100 { $value < 0.125 and $value != 0 }
rule : { $value } { $value := $value * 0.9394; } 100
                          { $rho=0 and $value>0 and statecount(0)=1 }
rule : { $value } { $value := $value * 0.7788; } 100
                          { $rho=0 and $value>0 and statecount(0)=2 }
…
rule : { $value } { $value := $value * 0.7358; } 100
                          { $rho=1 and $value>0 and statecount(0)=4 }     …
```

Figure 3: Fracture model specification

We used the values of ρ and to defined different rules for each value of n. The figure shows a snippet of the rules, where, if the strength of the cell is less than 0.125 and the cell is not broken (represented by a zero), the cell is set to be broken. For the actual crack propagation rules, if a cell is not broken, it will go down the list of rules until it finds a rule that matches its stored value of ρ and the number of broken neighbours, calculated by using *statecount(0)*. If one is found, the strength of the cell is multiplied by a pre-computed value from the feedback function, lowering the cell's strength. The cell passivates unless one neighbour changes and generates a new input. This prevents wasted CPU cycles on needless computations.

4 MODEL INITIALIZATION

One of the most important parts of simulating this earthquake model is the initialization phase. When a rock formation is divided into small elements, there are two variables to consider: the local stress that the rock experiences from all sides and the strength of the given rock element (Brune 1970). Because the stress calculations are handled by the model, the initialization phase needs to generate the initial strength distribution across the model. These cells are size-invariant, meaning that if properly defined they can represent any size of rock element – in this research, the size of the cell is unitless. In (Wang 1997, Turcotte 1988) the authors outline how to generate a fractal distribution. These steps are as follows:

- For each of the cells in an N x N array, two numbers are randomly generated: b, a real value between 0 and 1; and c, a real value between $-\pi$ and π.
- A third number a is generated for each cell using the following formula, ensuring the number follows a Gaussian distribution centered about zero.

$$a = \sqrt{-2logb} * \sin c \qquad (4.1)$$

- As fractal distributions are dependent on the spatial frequency of the array, we use a spatial frequency based on a 2D Fourier transform on the array. The complex Fourier coefficients A_{st} are computed using as follows:

$$A_{st} = \left(\frac{1}{N}\right)^2 \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} a_{nm} \exp\left[-\frac{2\pi i}{N}(sn + tm)\right] \quad (4.2)$$

where a_{nm} is the value of a at coordinate (n, m) in the old array, and A_{st} is the value at coordinate (s, t) in the new array.

- To modify the spatial frequencies represented by the coefficients of the 2D Fourier transform, the absolute frequency K_{st} is calculated for each cell (s, t) as follows:

$$K_{st} = (s^2 + t^2)^{1/2} \quad (4.3)$$

The fractal dimension D for the fractal distribution is then taken into account to calculate the fractal dimension of the spectral energy density (Wang 1997):

$$\beta = 7 - 2D \quad (4.4)$$

The value of A in each cell is then recalculated to generate a new value A' using the following formula:

$$A'_{st} = \frac{A_{st}}{K_{st}^{\beta/2}} \quad (4.5)$$

By performing this calculation, the higher spatial frequencies are dampened in a well-defined manner based on the value of D.

- An inverse 2D Fourier transform is then applied on the array, generating a new N x N array of cells. The values for these cells follow a fractal distribution.
- An absolute value of the elements of the array was calculated and the array was normalized so that the final values were between 0 and 1, to keep consistency between models of different sizes.

The method generates an array of size N and a fractal dimension D. **Figure 4** shows a contour figure of the fractal distribution when the fractal dimension is set to 2.1.

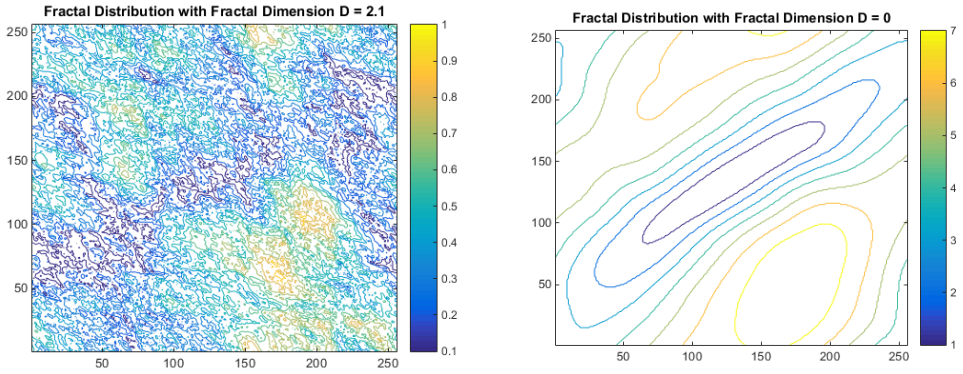

Figure 4: Contour of a fractal distribution (a) D=2.1. N = 256 (b) D=0; N = 256.

It can be seen that there are regions of the rock with higher strength (yellow/green regions); the blue and dark purple contours show regions of weaker rock; the darker colours are rock that is very likely to break. Qualitatively, this gives a good rock sample to use in the model. In Section 3, we showed that ρ is used to indicate how the elements of rock respond to local stress: the higher the value of ρ, the stronger the rock is. To integrate this parameter into the model, the distribution of ρ has to follow a general rule

where it is higher where the rock has the most strength. We generate a new fractal distribution using a lower value ($D = 0$) and the same array of random Gaussian values. By using the same array, the two distribution's lower spatial frequencies are similar to each other. Figure 4 (b) shows the results obtained. We generated a region with a low value of ρ in the same area where the strength of the rock is also low.

From these two contour figures, we can see that the areas with low ρ overlap with the areas of low strength, while still providing enough variance in the strength fractal distribution to create more interesting simulations.

5 SIMULATION RESULTS

The simulations were executed using the ARSLab RISE simulation server, which can run the CD++ simulations remotely (Wang and Wainer 2015). The Initialization model was modified so that it could have two separate stages: the random Gaussian matrix generation stage and the fractal distribution stage. When properly executed, many different fractal distributions can be generated from the exact same random Gaussian matrix. This means that because all of the fractal distributions stem from the same matrix, the results are comparable. A default simulation was obtained for analysis. For the defaults, we used the fractal dimensions in Section 4. **Figure 5** (a) shows the contour generated for the default strength distribution of the rock (fractal dimension $D = 2.1$). **Figure 5** (b) then shows the contour figure for the default ρ distribution, with a fractal dimension of $D = 0$. It can be seen that the low values of ρ properly line up with the regions of low strength from the previous figure.

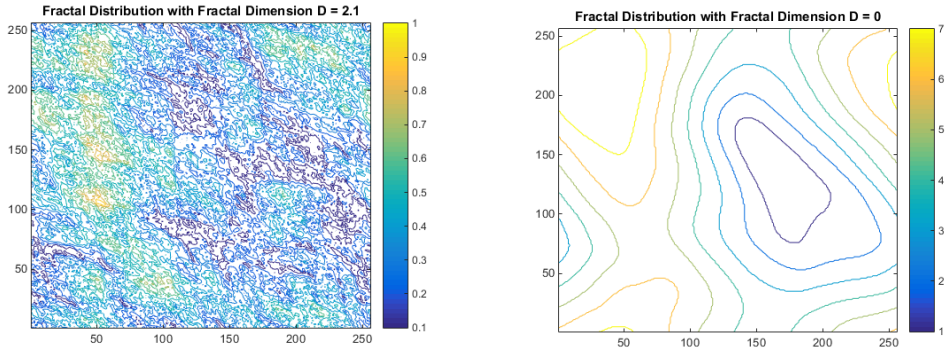

Figure 5: Contour. Default strength (a) D=2.1; N = 256. (b) D=0; N = 256.

Using the steps outlined at the beginning of this section, the simulation was run and visualized. A visualization of the results can be found found here: https://goo.gl/J3nu8t. **Figure 6** shows the simulation results.

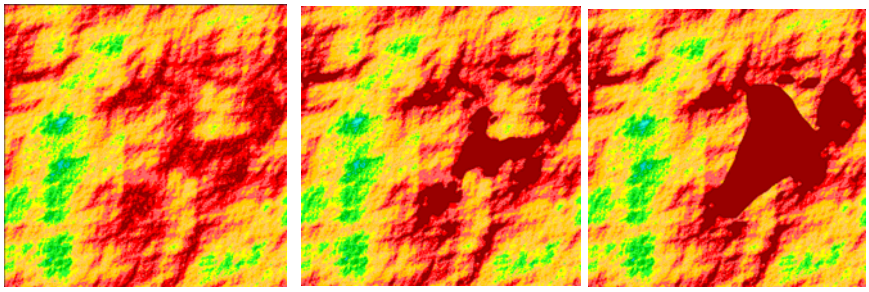

Figure 6: Simulation results. Baseline parameters. t=0; t=10; t=100.

At the beginning, we can see that we have the strength values from the initialization. After after 10 iterations, we have an initial crack that has grown, increasing the areas where the strength of the cell has fallen to zero. Although the cracks have grown very quickly near the middle-right of the figure, the left

side of the figure has not. The top-left corner of the figure corresponds to an area with a high ρ value. After 100 iterations, we see fractures combining into a larger crack. When two cracks get close enough, the rock between the cracks start to fail faster, eventually failing and linking the two independent cracks together. This happens several times throughout the simulation, and each time it happens it appears to accelerate the rate at which the crack as a whole propagates in that direction, albeit temporarily. The behaviour of two cracks merging together into a larger crack is an example of the emergent behaviour in the model. This graph also shows that the different regions in the model grow at different paces: even though similar structures are present in different areas, the areas with higher value of ρ have not grown significantly after 90 iterations. Because of this, we can say that the model depicts the effects of having different strengths of rock by altering the value of ρ.

One interesting behaviour observed in this simulation is how the speed of the propagation changes from the beginning to the end. At the beginning, the crack grows quickly, increasing the size of the cracks while also merging them together. However, after approximately 100 iterations, the pacing of the growth slowed down – once the major faults in the simulation merged together, the growth inched along at a very insignificant pace. This properly models the negative feedback effect of crack growth discussed in Section 2.3. The set of rules for the model capture the effects of the local feedback of the crack growth.

5.1 Varying strength fractal dimension

One of the parameters for the simulation whose influence we can study is the fractal dimension for both the strength distribution as well as the ρ distribution. We show only one of these parameters changed on each simulation. The first parameter we altered was the fractal dimension for the strength fractal distribution. **Figure 7** shows the figure for the strength distribution when the fractal dimension for the strength is lowered from $D_{strength}$ = 2.1 to 1.7.

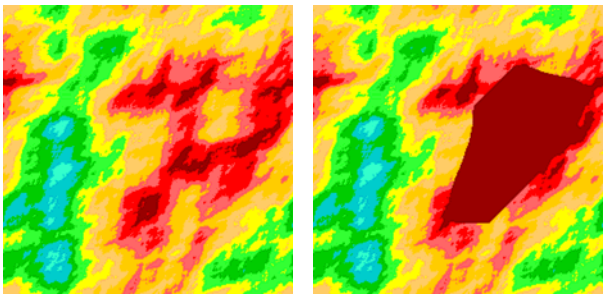

Figure 7: $D_{strength}$=1.7, Dρ=0; (a) t = 0; (b) end (556 iterations).

We can see that, although the general shape is the same as the baseline simulation, the ratio between the smaller shapes and the larger shapes has been reduced. We can see that by decreasing the value of the fractal dimension for the strength, the number of iterations required for the crack to stop propagating increases (556 iterations compared to 448 in the baseline). This is a side-effect of the fractal distribution: in the baseline simulation, the cracks spread out more evenly across the entire region, so that all of the cracks would grow independently first before eventually merging into a single large fracture. However, because there are less regions of small cracks, a single crack formed. This took longer/more iterations than the baseline simulation just because only one crack was growing instead of multiple at the same time.

The model states that cracks grow at a faster rate if there are smaller cracks than if it is just one big crack propagating. To test out this observation, the fractal dimension for the strength was raised to 2.3 and simulated with the same ρ fractal distribution, shown in **Figure 8**.

Where the simulation for the lower $D_{strength}$ yielded a distribution with less overall detail, the distribution for this simulation has more area that either is weak or is close to a weak region. First, the simulation only required 256 iterations compared to 418 iterations for the baseline and 556 iterations for $D_{strength}$ =

1.7. This continues the trend discussed previously where lower values of $D_{strength}$ increases the number of iterations required.

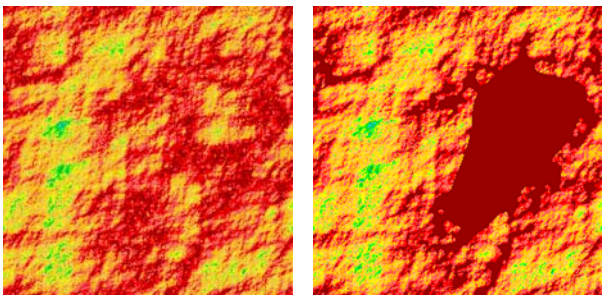

Figure 8: $D_{strength}$=2.3, Dρ=0; (a) t = 0; (b) end (t=256).

Across all the simulations completed, the general formation of the large crack in each figure is the same, barring the obvious variances that occur due to the changed fractal dimension. This means that even if $D_{strength}$ is different, the crack is confined to the rock types with low ρ.

5.2 5.2 Varying ρ fractal dimension

The next parameter of interest that is studied is $D\rho$, which is the fractal dimension for the ρ distribution. For these simulations, the fractal dimension for the strength is unaltered from the baseline and as such the starts of the simulations are unchanged. **Figure 9** shows the contour of the ρ distribution for when $D\rho$ = -1.

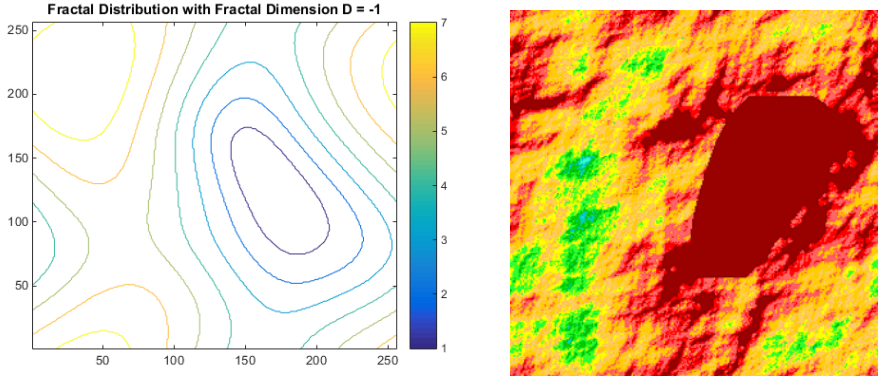

Figure 9: (a) Contour figure of Dρ = -1 fractal distribution; $D_{strength}$=2.1, (b) end of the simulation.

Even lower values of the fractal dimension do not cause a noticeable effect on the contour figure, so it is assumed that there is some limit at which a change in this parameter will cause no effect. Figure 9 (b) shows the figure of the simulation after 539 iterations, when the simulation ended. At the end of this simulation, it can be seen that the main crack in the rock has a more rounded look to it compared to the baseline simulation. As the regions with low ρ are closer together in this simulation, the initial cracks grow to fill that area more completely than the baseline simulation. The rounded shape of the ρ distribution also limits the growth of other cracks since apart from this one area with very weak rock formations the cracks are not able to grow as ably. Because the effects of decreasing the fractal dimension further are not very pronounced, custom ρ distributions would have to be constructed to further analyze this effect.

Although it is not possible to decrease $D\rho$ any further, the effects of increasing it can be analyzed. **Figure 10** shows the contour figure of the ρ distribution for when $D\rho$ = 1.

By increasing the fractal dimension, the regions of low-strength rock are further apart from each other. The effect of increasing $D\rho$ from the baseline value on the contour figure is more drastic than when it was decreased. The effect is more prominent on the simulation itself, however. Figure 10 shows the figure of

the simulation after 378 iterations. This simulation provides more insight as to how the ρ distribution affects the crack formation. When the rock is more varied, the areas where the cracks propagate quickly (by lower ρ values) are more spread out. Because of this, the rock quickly propagates through the regions with very weak rock the same as previous simulations, but they are not able to combine together into a larger crack.

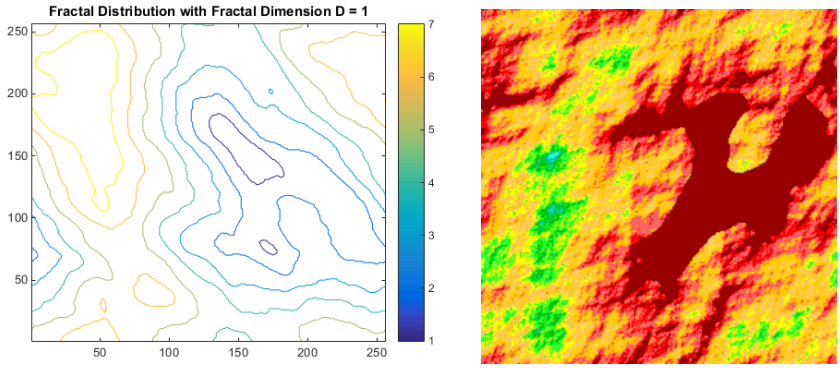

Figure 10: (a) Contour figure of the Dρ = 1 fractal distribution $D_{strength}$=2.1, (b) end of simulation.

Another way to explain it is because the ρ distribution has more variance, it is harder in general to propagate the crack. The region between the two prongs of the "y" in the figure is an area of higher ρ, but the value is approximately the same in the other simulations. However, because it was harder for the smaller cracks to combine together to encircle the region, the region remains untouched when in previous simulations it was completely filled in – previous simulations joined all of the smaller cracks around the opening of the "y" together.

When the fractal dimension for ρ was increased, the number of iterations required before no cell needed updating decreased. Similarly, when the fractal dimension was decreased the number of iterations required increased. In the previous section, it was seen that changing $D_{strength}$ had a similar effect on the iterations for the simulation. From this it is evident that the more complex a fracture structure is, the less number of iterations it takes to complete a simulation. It does not have a noticeable effect on how long the simulation overall takes, which indicates that each iteration handles more cell updates overall when the setup is more complex.

6 CONCLUSIONS

A fracture propagation model was implemented using the Cell-DEVS formalism to simulate how a crack would develop through a sample of rock. The model builds initialization files that set the strength and feedback functions of each of the cells in the model in accordance to a fractal distribution, which is considered the best model for the crack distribution in rocks. The model was then simulated and showed the propagation of cracks through a sample of rock. The model successfully captures several empirical effects, such as crack propagation slowing down as the crack's size increases and the acceleration of crack growth as two or more cracks get closer together. The model can also handle a distribution of rock with different rock types and crack growth functions, being versatile enough simulate these situations easily.

REFERENCES

Ammon, C. 1997. "Early Observations of Earthquakes - Shaking Intensity" SLU Department of Geosciences. [Online]. Available: http://eqseis.geosc.psu.edu/~cammon/HTML/Classes/IntroQuakes/Notes/intensity.html.

Brune, J. 1970. “Tectonic Stress and the Spectra of Seismic Shear Waves from Earthquakes”. *Journal of Geophysical Research* 75:4997-5009.
Falconer, K. 2003. “*Fractal Geometry*”. Wiley.
Gutenberg, B., and C. F. Richter. 1954. “*Seismicity of the Earth and Associated Phenomena*”. Princeton University Press, Princeton, N.J.
Hoek, E., and E. T. Brown. 1988. “The Hoek-Brown Failure Criterion—A 1988 Update”. *Proceedings of the 15th Canadian Rock Mechanics Symposium*, Toronto, ON, Canada.
Mandelbrot, B. 1983. “*The Fractal Geometry of Nature*”. Macmillan.
Park, R. 1997. “*Foundations of Structural Geology*”. Routeledge.
Turcotte, J. H. 1988. "Fractal Distributions of Stress and Strength and Variations of b-value". *Earth and Planetary Science Letter*, vol. 91, pp. 223-230.
Wainer, G. 2009. “*Discrete-Event Modeling and Simulation: A Practitioner's Approach*”. CRC Press.
Wang, Z. 1997."A Parallel Implementation of a Cellular Automata based Earthquake Model". M. Sc. Thesis. Carleton University. Ottawa, ON.
Wang, S., and G. Wainer. 2015. "A Simulation As A Service Methodology with Application for Crowd Modeling, Simulation and Visualization". *SIMULATION: Transactions of the Society for Modeling and Simulation International*. Vol. 91(1) 71–95.
Wang, Y., and T. Li. 2001. "The Fractal Characteristics of Micro-Cracks in Rock Under Uniaxial Compression". *Frontiers of Rock Mechanics and Sustainable Development in the 21st Century*. W.Sijing, F. Bingjun, L. Zhonkui Eds. CRC Press.

AUTHOR BIOGRAPHIES

GABRIEL A. WAINER (FSCS, SMIEEE), received the M.Sc. (1993) at the University of Buenos Aires, Argentina, and the Ph.D. (1998, with highest honors) at the Université d’Aix-Marseille III, France. In 2000 he joined the Department of Systems and Computer Engineering at Carleton University (Ottawa, ON, Canada), where he is now Full Professor and Associate Chair for Graduate Studies. He is the author of three books and over 350 research articles; he edited four other books, and helped organizing numerous conferences, including being one of the founders of the Symposium on Theory of Modeling and Simulation, SIMUTools and SimAUD. Prof. Wainer was Vice-President Conferences and Vice-President Publications, and is a member of the Board of Directors of the SCS. Prof. Wainer is the Special Issues Editor of SIMULATION, member of the Editorial Board of IEEE Computing in Science and Engineering, Wireless Networks (Elsevier), Journal of Defense Modeling and Simulation (SCS). He is the head of the Advanced Real-Time Simulation lab, located at Carleton University's Centre for advanced Simulation and Visualization (V-Sim). He has been the recipient of various awards, including the IBM Eclipse Innovation Award, SCS Leadership Award, and various Best Paper awards. He has been awarded Carleton University's Research Achievement Award (2005, 2014), the First Bernard P. Zeigler DEVS Modeling and Simulation Award, the SCS Outstanding Professional Award (2011), Carleton University’s Mentorship Award (2013), the SCS Distinguished Professional Award (2013), and the SCS Distinguished Service Award (2015). He is a Fellow of SCS. He is the Program Chair of WinterSim 2017. His e-mail address is gwainer@sce.carleton.ca.

SCOTT STEWART received a Bachelor of Engineering (B. Eng.) in Engineering Physics at Carleton University (2015) as well as his Master of Science in Electrical and Computer Engineering at the same university (2017). Currently he is working on his Ph. D. in Electrical and Computer Engineering at Carleton University. His current graduate work concerns the modelling of space-time modulated optical materials in the time-domain, which includes the simulation of two-dimensional space-time modulated Huygens' metasurfaces. He is also the recipient of 2017 Ontario Graduate Scholarship. His e-mail address is scott.stewart@carleton.ca.