CONTINUOUS AND DISCRETE MODELS FOR SIMULATING STONE EROSION

JONATHAN ALLEN, JOHN WANG

Abstract. In this paper we explore various techniques for modeling erosion processes on a stone. We develop and analyze an algorithm for modeling chipping. We also develop a method for modeling gradual stone erosion. Lastly, we demonstrate an algorithm for redistributing spectral points.

1. Introduction

Stones in riverbeds and on the ocean floor have very distinct shapes and contours. These stones tend to be smooth and flat, but the reason for this is not obvious. This paper examines how erosion affects the shape of stones. In particular, we examine two different mechanisms for stone erosion, a discrete one and a continuous one, and produce models for each one.

The first mechanism we study is the chipping of stones. One can imagine stones in a riverbed being tossed around and colliding with other stones. We create a discrete, polygonal model of a stone and a means of chipping the stone through a shear force. This model predicts that stones will tend to be flat and elongated.

The second mechanism we study is the wearing down of stones by gradual erosion. This mechanism appears when stones are gradually worn away by sand, water, or some other agent. We start of with a simple smooth model of the stone using spectral methods with the Fast Fourier Transform. This leads to problems, since the size of the stone decreases over time and changes shape, so we need to somehow resample. Most of our work involves designing a method for this resampling, which we are succesful at doing. We are then able to demonstrate our continuous model on a few simple problems, namely the processes of smoothing a stone and the process of erroding a stone from the bottom.

Date: October 28, 2013.

2. Discrete Model

In this section, we shall describe a model for simulating the erosion of a stone. In particular, we think of the erosion of a stone as coming from shear forces. This discrete model for eroding the stone allows us to simulate random processes and view the shape of a stone after some number of time steps. We provide computer simulations for the erosion process and analyze the model through these simulations.

2.1. **Shearing Stones.** The discrete model for stone erosion is based on the idea that stones are eroded via chipping. The process of chipping is analogous to dropping a stone from a specified height and applying a shear force to a particular point on the stone. Intuitively, one can think of a stone in a stream being eroded by being tossed and turned by the force of the river. This tossing and turning causes the stone to collide with other stones on the bottom of the riverbed, causing chipping to occur.

We model the interaction between two stones as a shear force - a force that causes two parts of a stone to move in opposite directions. The point where a stone collides with another stone is the point where the shear force is concentrated. This causes the stone to break into two pieces. In mechanics, the shear stress τ applied to the stone is given by the following equation:

$$\tau = \frac{F}{A} \tag{1}$$

Where F is the force applied and A is the cross-sectional area of material with area parallel to the applied force vector. If the shear stress τ is above the stress that the stone's material can withstand, then the stone will crack.

Notice that the stress is proportional to the cross-sectional area of meterial which is parallel to the force vector. Thus, we can see that if the amount of force applied is constant, the total area of the material that will be sheared if shearing occurs will be constant as well.

2.2. **Definitions.** This section will define how we represent a stone in the discrete model.

We will represent a stone as a polygon with n vertices. Formally, a polygon s can be defined as the set of vertices $s = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, where a vertex $\mathbf{v}_i = (x_i, y_i)$ is a tuple in R^2. In a polygon s, vertices $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ are connected by a line segment for $i \in \{1, 2, \ldots, n-1\}$. In addition, vertices $\mathbf{v}_n$ and $\mathbf{v}_1$ are also connected by a line segment.

The centroid $\mathbf{c}_s$ of a stone s is the center of mass of the stone when the stone has uniform density. In other words, the centroid is the point $\mathbf{c}_s \in \mathrm{R}^2$ where the following holds:

$$\int_A (\mathbf{r} - \mathbf{c}_s) dA = 0 \tag{2}$$

Where A is the area of the stone.

2.3. **The Chipping Process Model.** With this in mind, we shall now develop a model for the erosion of a stone based on shearing. The main idea of the model is that we will represent a stone as a two-dimensional polygon, and chip off a constant amount of area of the stone at every time step. In this way, we will attempt to capture the process of a stone colliding with another stone on a riverbed. Randomness will be introduced into the model by randomly selecting somewhere on the stone to start the shearing.

We shall now give a formal definition of the chipping process. A chip for a probability distribution P, stone s, and area A is denoted $Chip(\mathrm{P}, A, s)$ and is represented as follows:

(1) Select a vertex $\mathbf{v}_j$ from s. To select this vertex, we find the centroid $\mathbf{c}_s$ of stone s and choose an angle $\gamma \in [0, 2\pi)$ uniformly at random. Now, define the ray $\mathbf{l}$ as the ray with an initial point of $\mathbf{c}_s$ which extends outwards infinitely at an angle of γ from the horizontal. The vertex with the minimum perpendicular distance to $\mathbf{l}$ will be $\mathbf{v}_j$.
(2) Given vertex $\mathbf{v}_j$, define $\mathbf{l}_1$ as the line segment which connects $\mathbf{v}_j$ and $\mathbf{v}_{j-1 \pmod n}$. Similarly, define $\mathbf{l}_2$ as the line segment connecting $\mathbf{v}_j$ and $\mathbf{v}_{j+1 \pmod n}$.
(3) Select a point $\mathbf{p}_1$ at random from $\mathbf{l}_1$. To do this, we select a random $t \in [0, 1]$ such that $\mathbf{p}_1 = \mathbf{v}_j t + (1-t)\mathbf{v}_{j-1 \pmod n}$. The distribution of t is defined by the probability distribution P so that $t \sim \mathrm{P}$.
(4) Select the point $\mathbf{p}_2$ which lies on $\mathbf{l}_2$ for which the polygon defined by $\{\mathbf{p}_1, \mathbf{v}_j, \mathbf{p}_2\}$ has an area of A. If no such polygon exists, then choose $\mathbf{p}_2 = \mathbf{v}_{j+1 \pmod n}$.
(5) Create a new stone s' whose vertices are given by

$$s' = \{\mathbf{v}_1, \ldots, \mathbf{v}_{j-1}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{v}_{j+1}, \ldots, \mathbf{v}_n\}$$

Here, the vertex $\mathbf{v}_j$ has been replaced by $\mathbf{p}_1$ and $\mathbf{p}_2$ in the list of vertices for s. The new stone s' now has one more line segment and vertex than the old stone s.

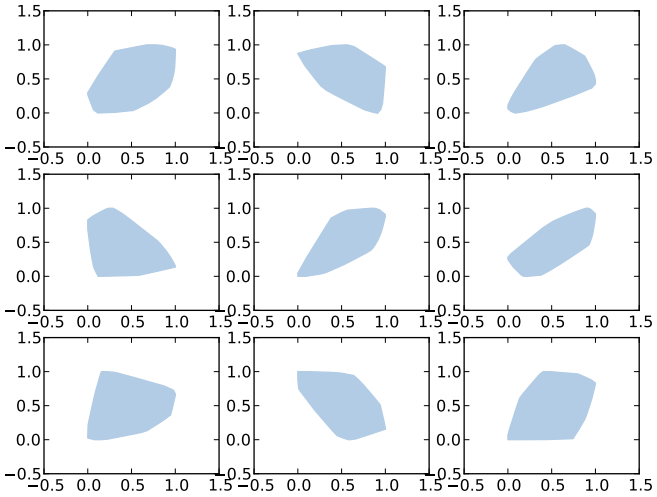

FIGURE 1. Chipping Process for a Unit Square

A chipping process for a probability distribution P, stone s, area A, and iterations k is denoted $ChipProcess(\mathrm{P}, A, s, k)$. A chipping process returns a stone s' which has iteratively been through k chips. Thus, a chipping process takes a stone s and creates $s_1 = Chip(\mathrm{P}, A, s)$, $s_2 = Chip(\mathrm{P}, A, s_1)$, $\ldots$, $s_k = Chip(\mathrm{P}, A, s_{k-1})$, and returns $s' = s_k$.

2.4. **Simulations.** The chipping process model allows us to simulate changes to a stone. A stone which undergoes a chipping process should have a shape which is close to the stones which undergo erosion and chipping in the real world (if our chipping process model is an accurate description of erosion).

To examine the accuracy of our model, we performed computer simulations using the Python programming language and the Shapely library. With computer simulations, we were able to create multiple instances of a chipping process and analyze the shapes of the resulting stones. We were also able to use multiple probability distributions P and areas A to generate different chipping processes.

Figure 1 shows the results for nine chipping process runs. The parameters used in these runs were $\mathrm{P} = \mathrm{N}(0.5, 0.15)$, $A = 0.05$, $k = 50$, and $s = \{(0,0),(1,0),(1,1),(0,1)\}$. The chipping processes each started out with a stone which was a unit square. The stones were chipped 50 times by the method described by the model. The probability distribution P is a normal distribution so that $t = 0.5$ is the most likely t to

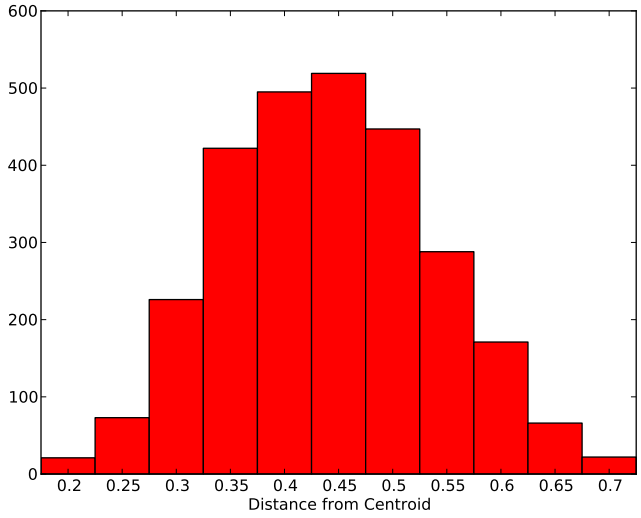

FIGURE 2. Histogram of Vertex Distance From Centroid

choose. In other words, it is most likely that $\mathbf{p}_1$ will be chosen as the midpoint of the line segment $\mathbf{l}_1$.

This distribution was chosen because it is more likely that shear forces will cause a symmetric split about the vertex $\mathbf{v}_j$. However, it is still possible for non-symmetric splits to occur, i.e. when $t \neq 0.5$. The normal distribution captures this nicely (all realizations of t which are not in the interval $[0, 1]$ were discarded).

Figure 1 displays the typical results of the chipping process model. None of the stones resemble a square and it would be hard for anyone to have guessed that these shapes came from a common origin. In fact, the randomization from the distribution P makes all of the shapes slightly different.

To examine the chipping process more closely, we can look at how circular or elliptical the resulting shapes are. One would expect rocks in a stream or riverbed to be elliptical in shape. We can therefore examine the distance of each vertex from the centroid of the stone. In a circle, all of the distances would be the same. In an elipse, one would expect to have an even number of points which are close and far away.

Figure 2 shows the distribution of vertex distances which were simulated with the same parameters as used for figure 1. One can see that the distribution looks fairly normal, with a mean of a distance of 0.472. The distribution of vertex distances from the centroid show that the shapes that result from this set of chipping process conditions

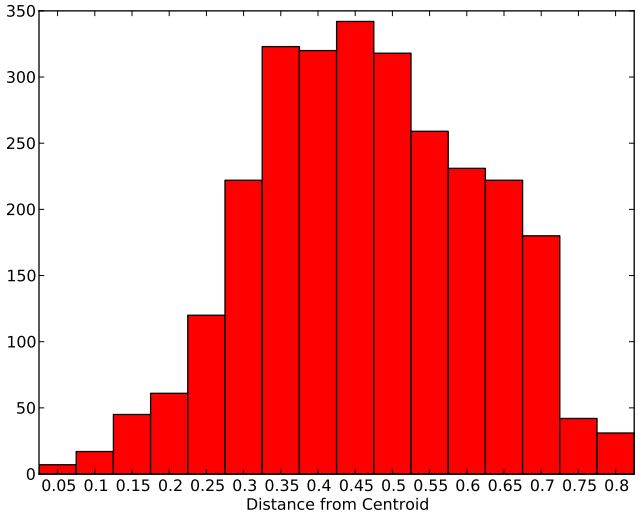

FIGURE 3. Histogram of Vertex Distance From Centroid for $P = N(0.05, 0.15)$

are neither elliptical nor spherical. However, a probability distribution $P = N(0.5, 0.15)$ tends to create a distribution of vertex distances from the centroid which are normal.

Choosing a different distribution for P yields dramatically different results. For example, if we choose $P = N(0.05, 0.15)$, we obtain the distribution shown in figure 3.

One can see that the distribution of vertex distances from the centroid are skewed so that there are far more vertices which are further away from the centroid. The resulting stones are much flatter than the stones shown in figure 1. The intuitive explanation for this is that there is a much higher probability that the shear will not be symmetric, so that the area that is removed will be much thinner. Because of this, it is more likely that longer, thinner stones will result.

3. Continuous Model

We next moved on to develop a continuous surface erosion model. The surface is still a shape in $\mathbb{R}^2$, but now it has bounded higher derivatives, rather than being piecewise linear. Physically this model would represents a surface being worn down, such as a bar of soap in water or ice melting.

Continuous Erosion Processes. There are a wide range of different erosion processes, and most of them are far too complex for the scope of this paper. To create simpler models, all of our processes only erode (move) each point on the surface in a direction normal to the surface at the point. This very rarely happens in real life, because materials are non-homogeneous, processes are nonlinear and chaotic, etc., but it is a reasonable assumption to make for a simplified model.

In contrast to modeling erosion as a discrete process, as was done in the previous section, we will now model it with the following differential equation

$$\frac{d\mathbf{x}}{dt} = g(\mathbf{x})\ \hat{\mathbf{n}}(\mathbf{x}) \tag{3}$$

for some function $g(\hat{\mathbf{x}})$ that depends on the erosion process. $\hat{\mathbf{n}}(\mathbf{x})$ is the unit vector normal to the surface at $\mathbf{x}$ and pointing inwards.

The method used to represent the continuous surface, $\mathbf{x}$ will be explained momentarily, but first we will take a moment to explain the various erosion processes that we will model in this section.

Smoothing Process. One of the processes that we will look at is a soap bar being worn down by a person holding it in their hand, which we will call the *Smoothing Process*. In this process, the corners of the surface will get worn down quickest, because they are the first parts of the surface to come into contact with the person's hand. Technically the furthest protruding bumps will be worn down quickest, but to simplify the problem, we consider every point of the surface to wear down at a rate proportional to the curvature at the point. We will use a signed curvature κ, so that a bump will have positive curvature, and an indentation will have negative curvature. $g(\mathbf{x})$ for this model is the following

$$g(\kappa(\mathbf{x})) = \tan^{-1}(\beta\,(\kappa(\mathbf{x}) - \alpha)) + \frac{\pi}{2} \tag{4}$$

For our model we chose parameters, $\alpha = 1,\ \beta = 5$.

Bottom Recession Process. Our second process is an object sitting in a bath of acid, which we will call our *Bottom Recession Process*. In this process, the lowest points (smallest values of $\mathbf{x}_2$) on the surface will erode quickest. $g(\mathbf{x})$ for this model is the following

$$f(\mathbf{x}) = \frac{1}{\alpha\left(\mathbf{x}_2 - x_{2,min}\right) + \beta} - \gamma \tag{5}$$

$$g(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{for} \quad f(\mathbf{x}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

For our model we chose parameters, $\alpha = 10,\ \beta = \gamma = 0.1$.

Example Surface. The example shape that we will use for modeling our continuous processes is shown in Figure 4.

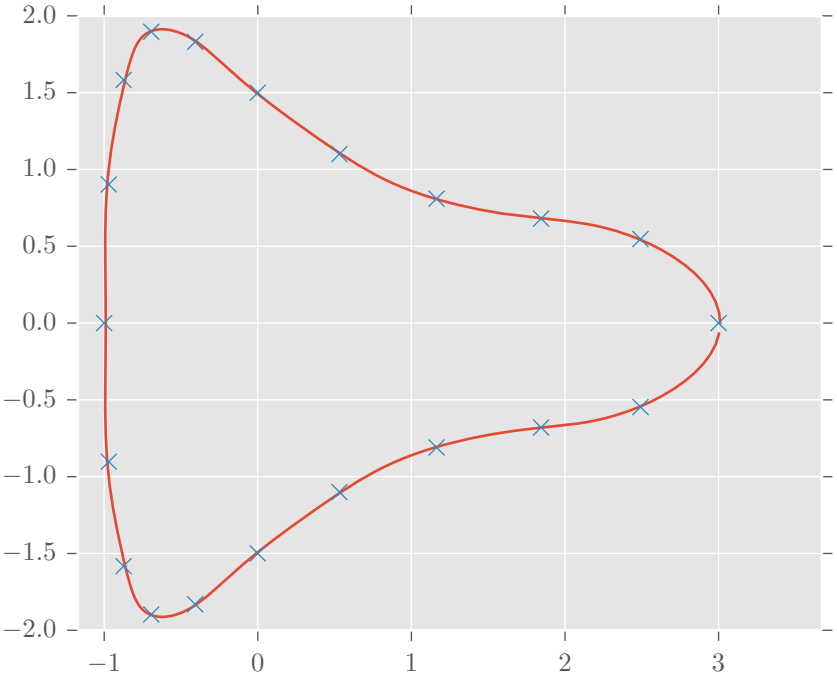

FIGURE 4. Example shape.

The equations for the shape are the following (T_i is the i$^{\text{th}}$ Chebyshev polynomial)

$$\mathbf{x}_1(s) = \cos(s)\left(2T_0(\left(\frac{s-\pi}{\pi}\right)) + T_2\left(\frac{s-\pi}{\pi}\right)\right)$$
$$\mathbf{x}_2(s) = \sin(s)\left(2T_0(\left(\frac{s-\pi}{\pi}\right)) + T_4(\left(\frac{s-\pi}{\pi}\right))\right)$$

Surface Equations. Our surface is modeled with two vectors, $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^N$ for some resolution N. $\mathbf{x} \coloneqq \left[\mathbf{x}_1^T, \mathbf{x}_2^T\right]$ is the vector of points measured counterclockwise around the surface, which we will call the evaluation points, since these are the points at which we will evaluate the rhs of Equation 3.

$$\text{(Velocity)} \qquad \dot{\mathbf{x}} \coloneqq \frac{d\mathbf{x}}{dt}$$

$$\text{(Acceleration)} \qquad \ddot{\mathbf{x}} \coloneqq \frac{d^2\mathbf{x}}{dt^2}$$

$$\text{(Curvature)} \qquad \kappa = \frac{\dot{\mathbf{x}} \times \ddot{\mathbf{x}}}{\dot{x}^3}$$

Numerical Modeling. We chose to model the surface parametrically with a Discrete Fourier series and use spectral methods for implementing Equation 3. Our processes will keep the surface reasonably smooth as time progresses, so spectral methods will maintain the smoothness of the surface much better than a local approximation of the surface using finite differences which only acts locally at points. When we refer to smoothness, we are really referring to how many derivatives of the function are bounded, or in other words, how quickly the Fourier coefficients go to 0. Since, the surface is in $\mathbb{R}^2$, we will use the Real Fast Fourier Transform ($\mathcal{RFFT}$), since it is slightly faster than the standard Fast Fourier Transform (equations for the $\mathcal{RFFT}$ can be found in the appendix).

To model the surface parametrically, we define a parameter vector $\mathbf{s} = 2\,\pi\,n/N, \qquad \text{for} \quad n = 0, \ldots, N-1$. Now we let $\mathbf{s}$ be a counterclockwise sampling of the surface at locations $\mathbf{s}$ ($\mathbf{s}$ can be scaled by a constant factor if the surface is not of length 2π).

The $\mathcal{RFFT}$ allows us to calculate derivatives in $n \log n$ time. The following equations are derived in most introductory spectral method texts, so they will be stated without proof here

$$\dot{\mathbf{x}} = \mathcal{RFFT}^{-1}(diag(i\,[0 : N/2, 0])\ \mathcal{RFFT}(\mathbf{x})) \tag{7}$$

$$\ddot{\mathbf{x}} = \mathcal{RFFT}^{-1}\big(diag(i^2[0 : N/2 + 1]^2)\ \mathcal{RFFT}(\mathbf{x})\big) \tag{8}$$

Note: $diag(\mathbf{a})$ is the diagonal matrix with the elements of $\mathbf{a}$ along the diagonal. Also, $\mathbf{x}_1$ and $\mathbf{x}_2$ are treated separately, e.g. $\mathcal{RFFT}(\mathbf{x}) \coloneqq [\mathcal{RFFT}(\mathbf{x}_1), \mathcal{RFFT}(\mathbf{x}_2)]$.

Using the above equations, the surface normal and surface curvature can now be quickly computed

$$\hat{\mathbf{n}} = [-\dot{\mathbf{x}}_2, \dot{\mathbf{x}}_1] \tag{9}$$

$$k = \frac{\dot{\mathbf{x}}_1\ddot{\mathbf{x}}_2 - \dot{\mathbf{x}}_2\ddot{\mathbf{x}}_1}{(\mathbf{x}_1^2 + \mathbf{x}_2^2)^{3/2}} \tag{10}$$

Forward time integration of Equation 3 can now be implemented with a standard ODE time stepping integration methods. In this paper, we used SciPy's `odeint` method for integrating Equation 3.

Point Collision Problems. Unfortunately, the simplicity gained from modeling the surface with a Fourier series is soon lost when we start time stepping. As can be seen in Figure 5, the process works well for a short time period, but then the surface folds over itself (a physical impossibility).

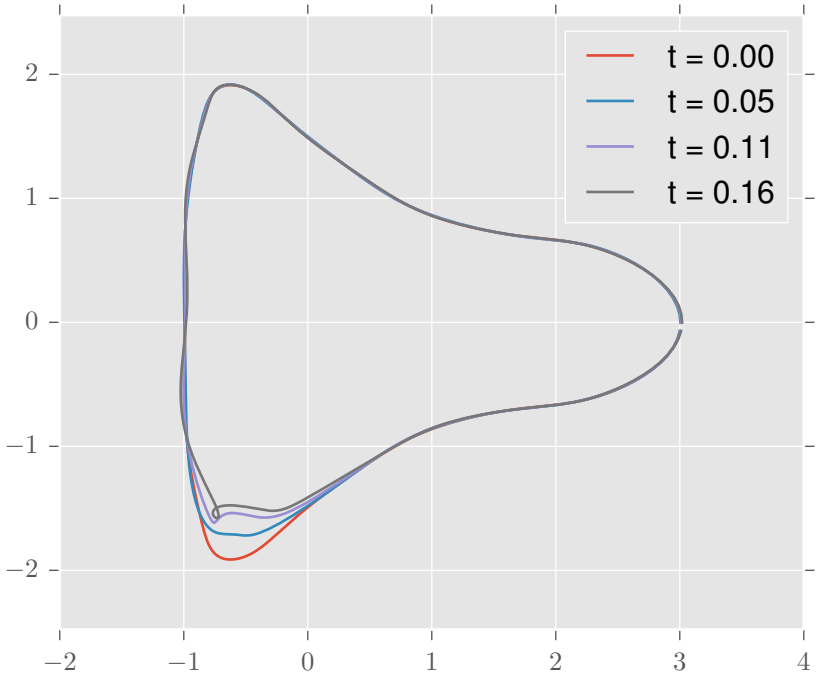

FIGURE 5

To investigate why this occurs, we will show a zoomed in portion of the same figure, with the evaluation points shown.

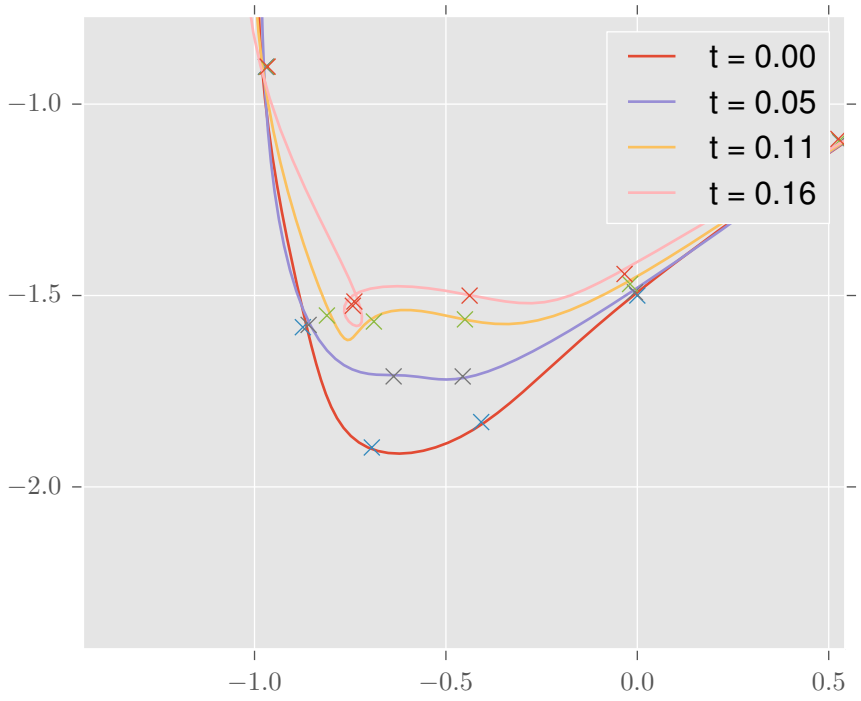

FIGURE 6

Looking closely, it can be seen that areas of high curvature form when evaluation points get close to each other, and this eventually leads to surface folding. Our guess is that this occurs because we are using discrete time stepping, so there are errors in the approximation of the shape's change with time. This is fine when the evaluation points are far from each other, but when they get close to each other, the errors become significant and affect the direction of the surface unit normal vectors. The folding causes unit normal vectors of nearby points to point at each other and which subsequently causes the points to cross over each other.

Redistributing Points. The most obvious way to fix this problem would be to redistribute the points evenly on the surface at every step. This wouldn't be too difficult of a task, since we can easily integrate and interpolate with the FFT and find a new set of evaluation points, $\mathbf{x}$ that are evenly spaced on the surface.

Unfortunately, if we arbitrarily move points around, we lose spectral accuracy, because our new points will be sampled at different parameter values $\tilde{s} \neq s$, but the Discrete Fourier transform assumes that the

points are sampled at s. This parameter transformation is okay if there exists some smooth (ideally analytic) function, $h(s)$, such that $h(s) = \tilde{s}$. This technique of parameter transformation is utilized extensively in spectral methods involving Chebyshev polynomials which use $h(s) = \cos(s)$ to cluster evaluation points near both endpoints of the domain.

Redistributing points in such a way to ensure that $h(s)$ exists is difficult though, so we will first describe some different methods we attempted at eliminating the surface folding.

High Frequency Damping. Our first approach was to use fewer Fourier terms than the number of evaluation points, in an attempt to better approximate the shape while ignoring the higher frequencies. This unfortunately performed even worse than our original algorithm.

Partial High Frequency Damping. In the *Bottom Recession Process*, $\mathbf{x}_1$ should be changing slowly and smoothly with time, since the recession model primarily affects $\mathbf{x}_2$. In Figure 7, $\mathbf{x}_1$ is plotted, and we can see that this is not the case in our simple algorithm and that large high frequency oscillations appear as we move forward in time.

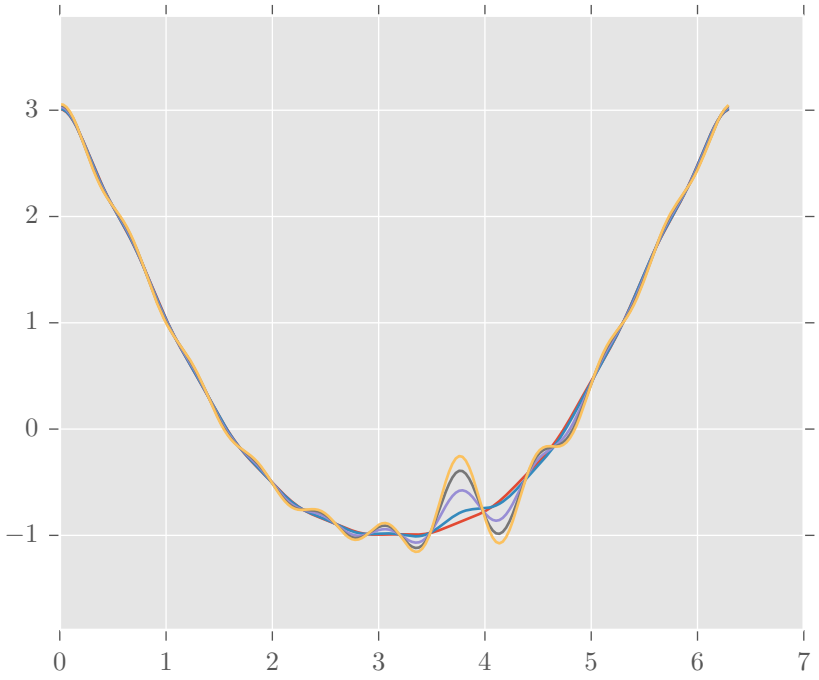

FIGURE 7

This problem led us to attempt to artificially dampen the high frequencies in $\mathbf{x}_1$, by manually zeroing out the high frequency Fourier coefficients. This worked reasonably well, but was not a very robust method. Our shape is nice, because $\mathbf{x}_1$ can be represented accurately with very few Fourier terms, but this would not be the case for more complicated shapes and other models, such as the *Smoothing Process* that affects $\mathbf{x}_1$ and $\mathbf{x}_2$ equally.

Continuous Redistribution. To redistribute points, such that there exists some function $h(s)$ as described above we decided to model our redistribution by adding a term to Equation 3, which is shown below. This works out nicely, because now the redistribution is smooth, so $h(s)$ must exist, as long as Equation 11 has a solution.

$$\frac{d\mathbf{x}}{dt} = g(x)\,\hat{\mathbf{n}} + \alpha\,\mathbf{a}_t \tag{11}$$

$$\mathbf{a}_t = \frac{\ddot{\mathbf{x}} \cdot \dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}\,\frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|} \tag{12}$$

Physically, $\mathbf{a}_t$ is the component of $\ddot{\mathbf{x}}$ tangent to the surface. The reasoning behind adding this term is that $\dot{\mathbf{x}}$ is inversely proportional to the density of points on the surface; we want to shift points away from denser areas, which means moving them in the direction of larger $\dot{\mathbf{x}}$ which is $\mathbf{a}_t$. For the moment, we set $\alpha = 1$, since $\mathbf{a}_t$ is of the same order as $g(x)$ for our surface. More analysis should be done in the future to best determine what α should be (it seems likely that the optimal would depend on $g(x)$).

As can be seen in Figure 8, this dramatically improves our algorithm.

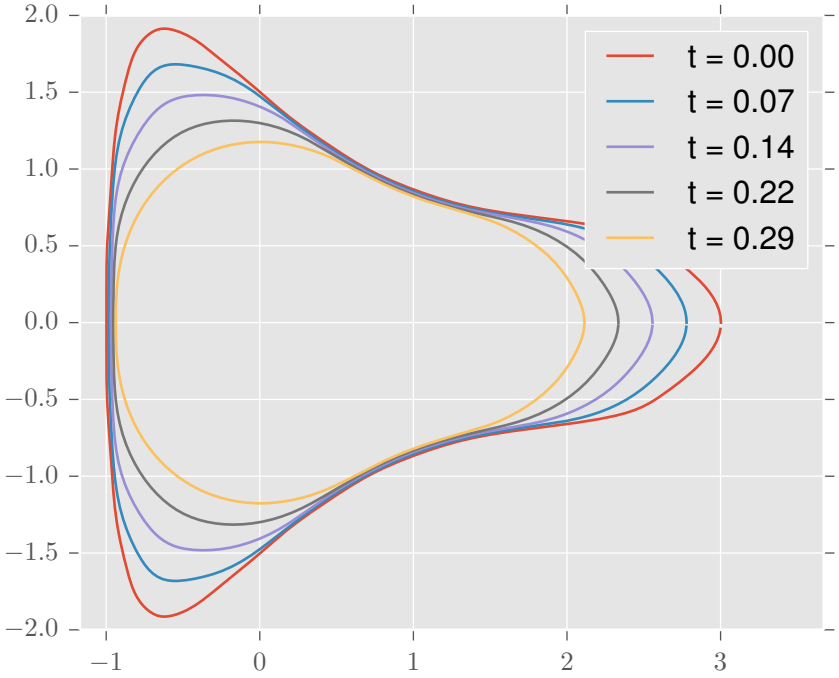

FIGURE 8

If we plot the evaluation points (Figure 9), it can be seen that the points redistribute themselves over time to stay evenly spaced on the surface.

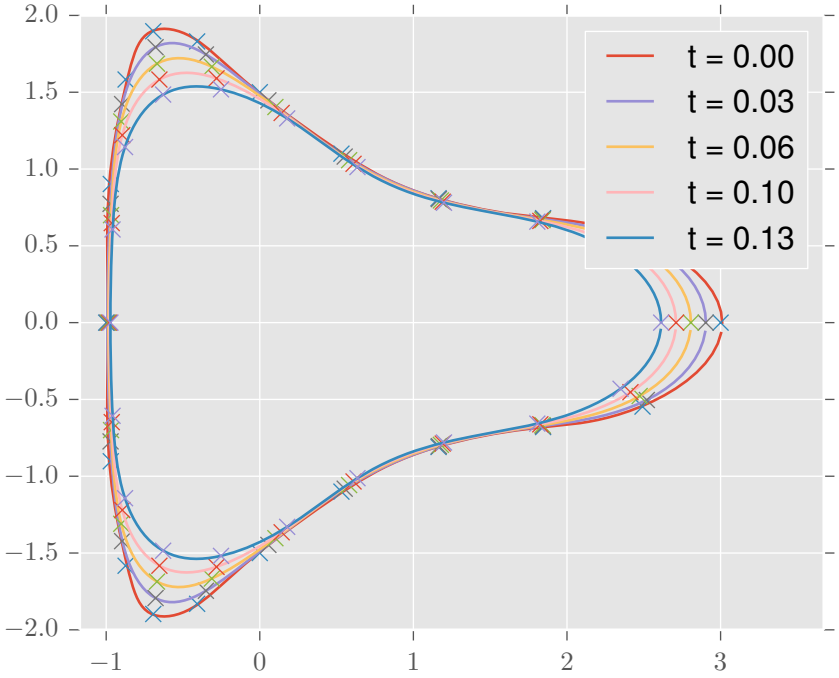

FIGURE 9

4. Conclusion

In this paper, we presented two models which could determine how stones are shaped by erosion. The first model was a discrete model based off of a mechanism for chipping a stone via shear forces. The second model was a continuous model that used Fourier transforms to model a stone being worn down. Each model targets a different aspect of erosion, and taken together, one can better understand the actual mechanism for the erosion of stones.

Appendix

Real Fourier Transform. The Real Fourier transform is defined as

$$\begin{aligned}\hat{\mathbf{x}} &:= \mathcal{RFFT}(x)_k \\ &:= \sum_{n=0}^{N-1} e^{-2\pi i k(n/N)} x_n \qquad \text{for} \quad k = 0, \ldots, N/2+1\end{aligned}$$

Note: the terms FFT and Fourier transform are used interchangeably in the paper. Technically they are not synonyms, but it is common practice to use the terms interchangeably.

And the inverse transform is defined as

$$x := \mathcal{RFFT}^{-1}(\hat{\mathbf{x}})$$
$$:= \frac{1}{N}\left(\sum_{k=0}^{N/2+1} e^{2\pi i k(n/N)} X_k + \sum_{k=N/2+2}^{N-1} e^{2\pi i k(n/N)} \bar{X}_k\right) \qquad \text{for} \quad n = 0, \ldots, N-1$$

References

[1] Fast Fourier Transform. (2013, 10 8). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Fft

[2] Trefethen, L. N. (2000). Spectral Methods in MATLAB. Philadelphia: SIAM.