

1. CONTINUOUS SURFACE

Method. We next moved on to develop a continuous surface erosion model. Physically this model represents a surface being worn down, such as a bar of soap in water or ice melting.

There are a wide range of different erosion processes, and most of them are very complex. To simplify our models, all of our processes erode (move) each point along the surface in a direction normal to the surface at the point. This rarely happens in real life, because of non-homogeneous materials, nonlinear behavior, etc., but it is a reasonable assumption to make for a simplified model.

Rather than modeling erosion as a discrete process, as was done in the previous section, we can model it with the following differential equation

$$(1) \quad \frac{d\mathbf{x}}{dt} = g(\mathbf{x}) \hat{\mathbf{n}}(\mathbf{x})$$

for some function $g(\hat{\mathbf{x}})$ to be defined shortly. $\hat{\mathbf{n}}(\mathbf{x})$ is the unit vector normal to the surface at \mathbf{x} .

The method used to represent the continuous surface will be explained momentarily, but first we will take a moment to explain the various erosion processes that we modeled.

The first process is a soap bar being worn down by a person holding it in their hand, which we will call our *smoothing model*. In this process, the corners of the soap bar 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, so that positive curvature represents an outward facing bump, and negative curvature represents an indentation in the surface. $g(\mathbf{x})$ for this model is the following (κ is the curvature)

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

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

Our second process is an object sitting in a bath of acid, which we will call our *bottom recession model*. In this process, the bottom of the object will recede quickest. $g(\mathbf{x})$ for this model is the following ($x_{2,min}$ is the lowest point on the object)

$$(3) \quad f(\mathbf{x}) = \frac{1}{\alpha(x_2 - x_{2,min}) + \beta} - \gamma$$

$$(4) \quad g(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{for } f(\mathbf{x}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

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

Surface Equations. Our surface is modeled with two vectors, $x_1, x_2 \in \mathbb{R}^N$ for some resolution N . Then $\mathbf{x} := [x_1^T, x_2^T]$, with x_1 and x_2 being the two coordinates of points sampled counterclockwise along the surface.

$$\begin{aligned} \text{(Velocity)} \quad \dot{\mathbf{x}} &:= \frac{d\mathbf{x}}{dt} \\ \text{(Acceleration)} \quad \ddot{\mathbf{x}} &:= \frac{d^2\mathbf{x}}{dt^2} \\ \text{(Curvature)} \quad \kappa &= \frac{\dot{\mathbf{x}} \times \ddot{\mathbf{x}}}{\dot{x}^3} \end{aligned}$$

Numerical Modeling. We chose to model the surface parametrically with a Discrete Fourier series using spectral methods. Our processes tend to smooth the surfaces out over time, so spectral methods would capture this much better than a local approximation of the surface using finite differences which only act locally at points. All of our surface are in \mathbb{R}^2 , so we will use the Real Fast Fourier Transform (equations in the appendix).

$$(5) \quad \dot{\mathbf{x}} = RFFT^{-1}(diag(ik) RFFT(\mathbf{x}))$$

$$(6) \quad \ddot{\mathbf{x}} = RFFT^{-1}(diag(-k^2) RFFT(\mathbf{x}))$$

$$(7)$$

$$(8) \quad \hat{\mathbf{n}} = [-\dot{x}_2, \dot{x}_1]$$

$$(9) \quad k = \frac{\dot{a}\ddot{b} - \dot{b}\ddot{a}}{(a^2 + b^2)^{3/2}}$$

Example Surface. The example shape that we will use throughout this section is shown in Figure 1.

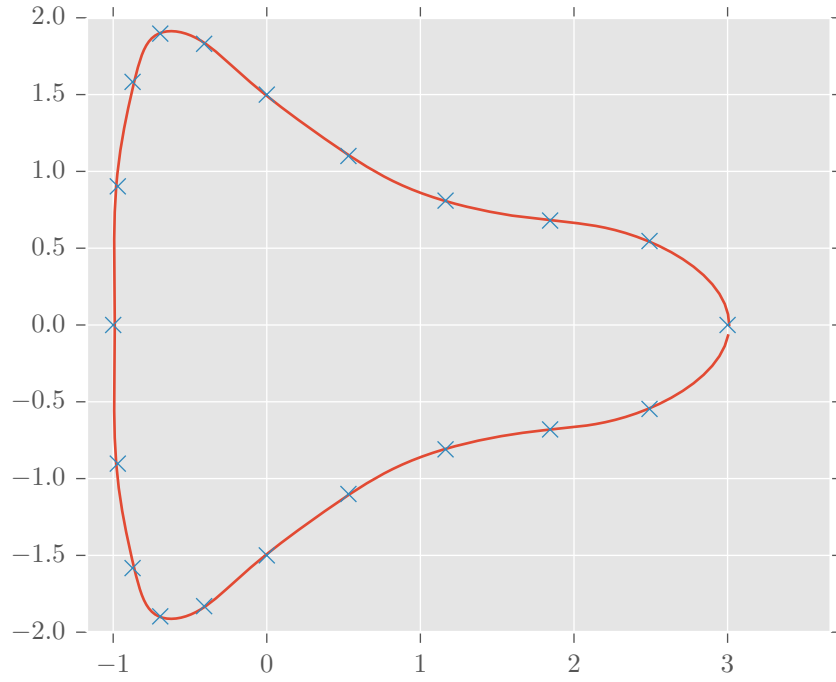


FIGURE 1. Example shape.

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

$$x_1(s) = \cos(s) \left(2T_0\left(\frac{s-\pi}{\pi}\right) + T_2\left(\frac{s-\pi}{\pi}\right) \right)$$

$$x_2(s) = \sin(s) \left(2T_0\left(\frac{s-\pi}{\pi}\right) + T_4\left(\frac{s-\pi}{\pi}\right) \right)$$

Point Collision Problems. Unfortunately, the simplicity of modeling the surface with Fourier series breaks down when the surface starts to shrink. As can be seen in Figure 2, the process works well for a short time period, but sharp points start to form as time increases.

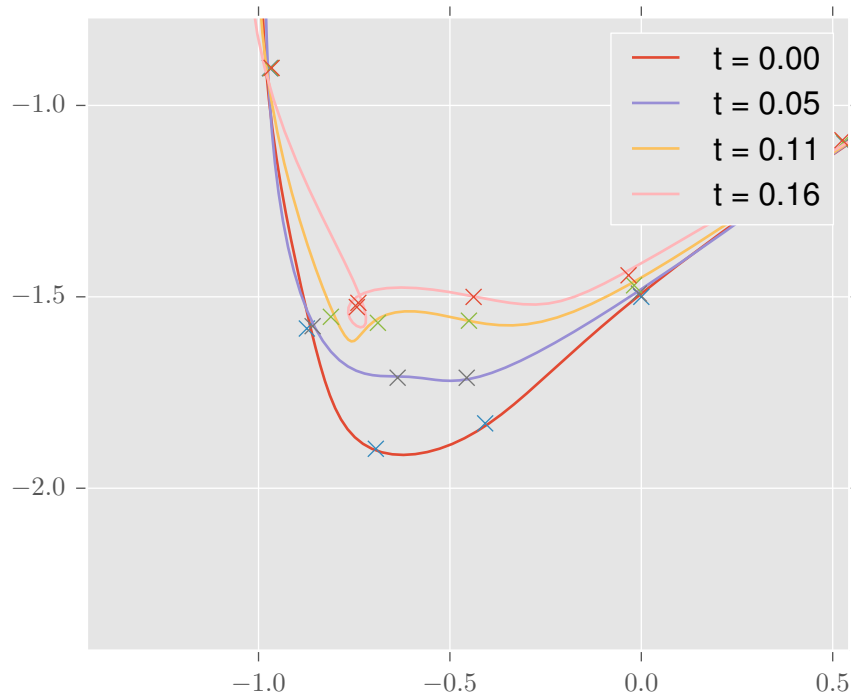


FIGURE 2

To see why this occurs, we will show the same figure, but with the evaluation points shown. Looking closely, it can be seen that the sharp corners form when evaluation points get too close to each other. Our guess is that this occurs because we are using discrete time stepping, so the shape's change with time is only an approximation. 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. Rather

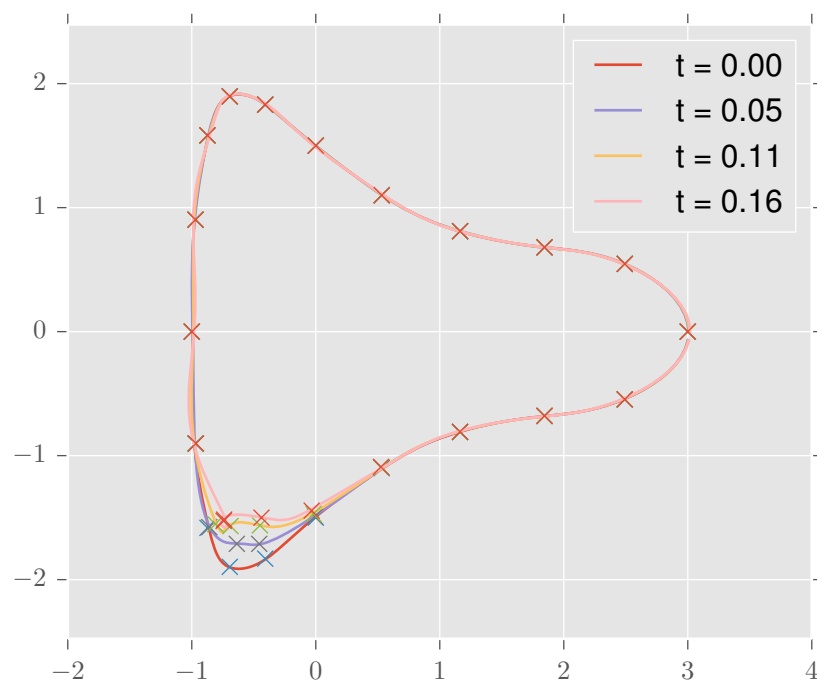


FIGURE 3

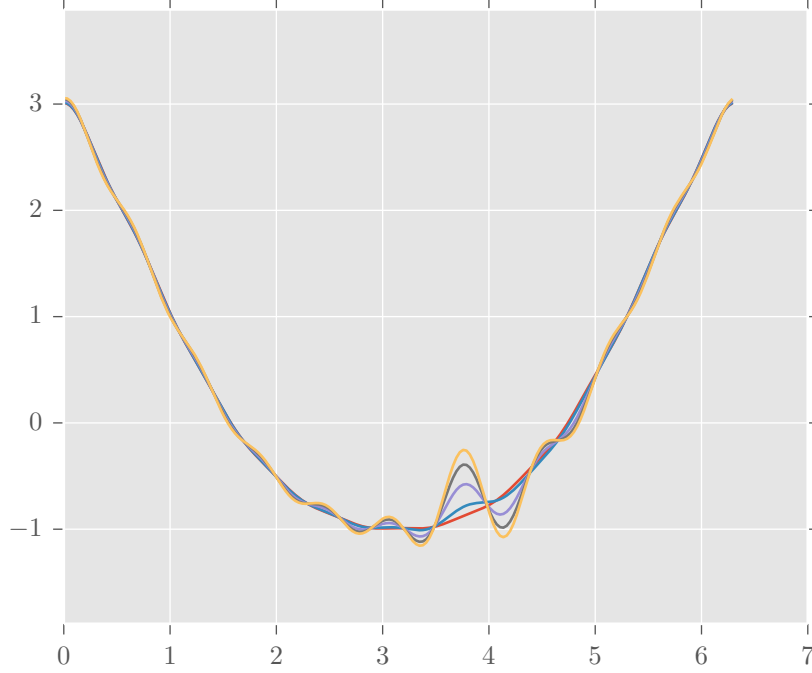


FIGURE 4

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 with the FFT.

Unfortunately, if we arbitrarily move points around, we lose spectral accuracy. Spectral methods . This method turns out to work, but I will hold off explaining it

High Frequency Damping.

Partial High Frequency Damping. In the *erode lowest model*,

Continuous Redistribution. To redistribute points, we need to find some smooth function $f(s)$ that will map our evenly distributed $theta$. This can be achieved by

$$(10) \quad \frac{d\mathbf{x}}{dt} = g(x) \hat{\mathbf{n}} + f(\mathbf{a}_t)$$

$$(11) \quad \mathbf{a}_t = \frac{\ddot{\mathbf{x}} \cdot \dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|} \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}$$

\mathbf{a}_t is the component of $\ddot{\mathbf{x}}$ tangent to the surface. $\dot{\mathbf{x}}$ is inversely proportional to the density of points on the surface. We want to shift points away from denser areas, which means moving in the direction of larger $\dot{\mathbf{x}}$.

As can be seen in Figure 5, this dramatically improves our algorithm.

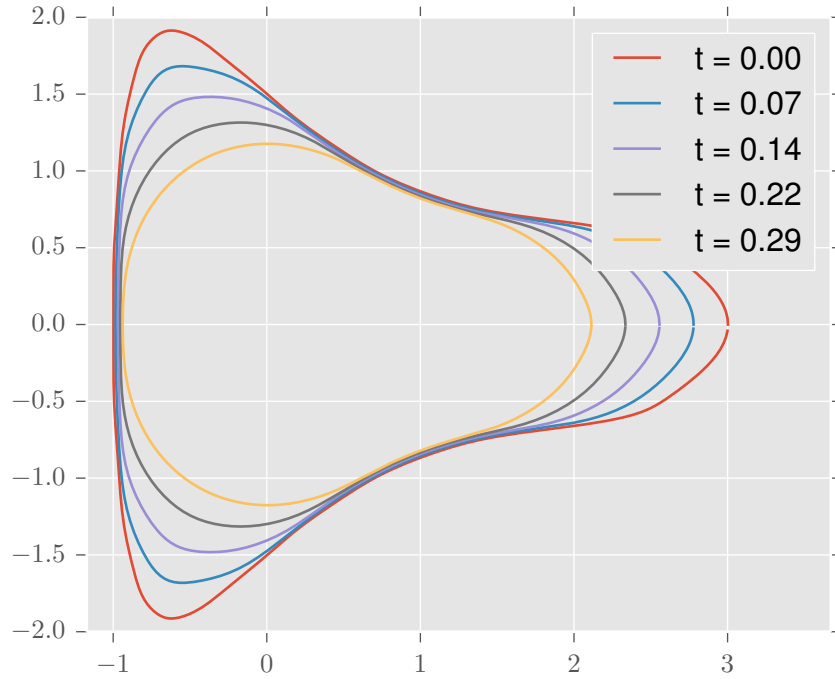


FIGURE 5

If we plot the evaluation points (Figure 6, it can be seen that the points redistribute themselves over time to stay evenly spaced on the surface.

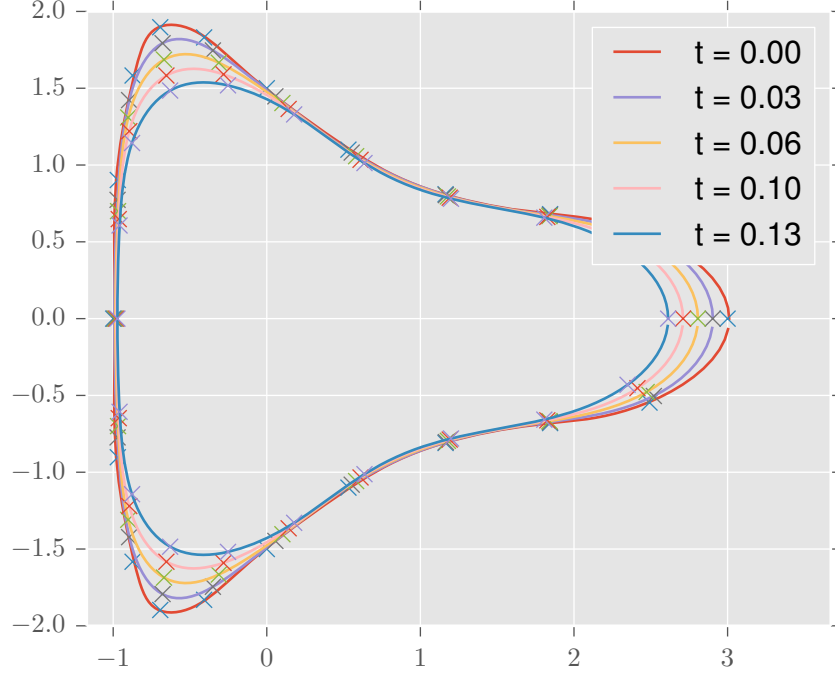


FIGURE 6

APPENDIX

Real Fourier Transform. The Real Fourier transform is defined as

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

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

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