Hyperbolicity of Surfaces with Positive Curvature

Katherine Wong and Vaughn Climenhaga

Department of Mathematics, University of Houston, Houston, TX, USA

UNIVERSITY OF HOUSTON

Introduction

Consider a Riemannian surface composed of a cylinder and two hemispheres at each end. This surface has net positive **Gaussian curvature**, which is defined as the product of principal curvatures. We'll call this surface $\mathcal{S}$. If $\mathcal{S}$ is flattened in the one direction, and geodesic flow is applied, we want to show that this system exhibits characteristics of chaotic behavior. Think of a **geodesic** as a generalization of a straight line or the shortest possible path. More precisely, we will show that the **Lyapunov exponent**, λ, is positive. This implies sensitivity to initial conditions. It is already well known that surfaces of negative Gaussian curvature exhibit hyperbolicity.

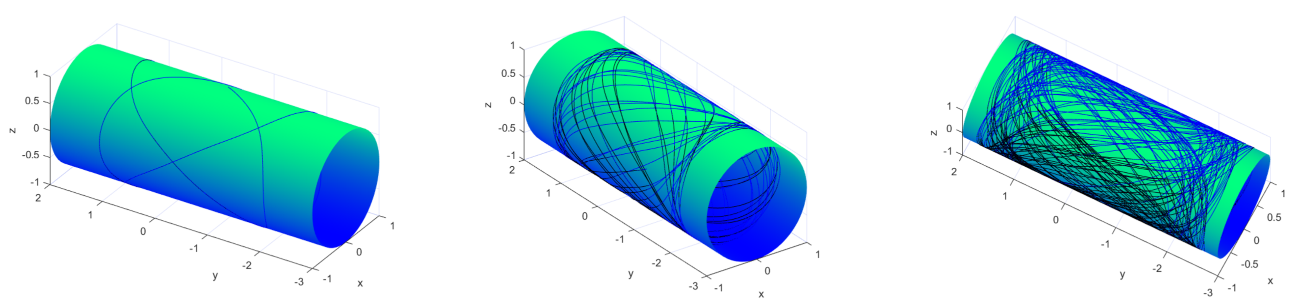

FIG. 1: Geodesics on a cylinder of radius 1 with N = 5000, 50000, and 200000 steps.

Mathematical Problem Formulation

We first draw $\mathcal{S}$ using the parameterization definitions. This mimics spherical and cylindrical coordinates where θ and ϕ are the parameters for the former and θ' and z are parameters for the latter. Now we flatten $\mathcal{S}$ in the z-direction. We do this by defining a global variable $s \in (0, 1)$ that scales all z-components of our parameterizations. Now that $\mathcal{S}$ is drawn, we need to discretize this geodesics. They can be approximated using a method analogous to **Euler's fixed-step method.** Since geodesics can be represented by differential equations, we can use a method that approximates solutions to differential equations numerically. In order to use Euler's method, we must specify a step size, ε, and number of steps, N. These will both be global variables that will be determined by the user. We first create a function that inputs a point sufficiently close to the surface but not on it and projects that point back onto $\mathcal{S}$. We want to ensure the projection is normal to $\mathcal{S}$ in order to minimize the distance. This whole idea of minimizing distance leads us to using **Lagrange multipliers** in order to calculate our projected point in terms of our nearby point. We will use the level set definition of $\mathcal{S}$ for this step. If we let (x_0', y_0', z_0') be a point nearby $\mathcal{S}$ and (x_1, y_1, z_1) be a point on $\mathcal{S}$, then we get the vector $\vec{\mathbf{v}} = (x_0'-x_1, y_0'-y_1, z_0'-z_1)$. We then let $f(x, y, z)$ be our level set definition of one part of $\mathcal{S}$, in order to solve for some $t \in \mathbb{R}$ such that $t\nabla f(x, y, z) = \vec{\mathbf{v}}$. After this, we must Taylor expand both sides of the equation about $t = 0$ and we ignore all orders higher than 2. For the cylindrical part, our projected point is $\left(\frac{x_0'}{1+2t}, y_0', \frac{z_0'}{1+\frac{2t}{s^2}}\right)$ with $t = \frac{s^2-x_0'^2s^2-z_0'^2}{4x_0'^2+4z_0'^2-4s^2-4}$ For the hemisphere centered at $y = 0$, our projected point is $\left(\frac{x_0'}{t+1}, \frac{y_0'}{t+1}, \frac{z_0'}{\frac{t}{b^2}+1}\right)$ with $t = \frac{s^4(1-x_0'^2-y_0'^2)}{2(s^2x_0'^2+s^2y_0'^2+z_0'^2-s^4-s^2)}$ In order to get the projected point for the hemisphere centered at $y = -1$, we would just need to do a variable change $y \to y - 1$ for our projected point on the hemisphere centered at $y = 0$. Now that we have a function that projects a point back onto $\mathcal{S}$, we need a function to project a vector back onto the tangent space of the surface. Equivalently, we need to take a vector that's not tangent to the surface and make it tangent to the surface. We do this to ensure that we get the straightest possible line, since that's a characteristic of a geodesic. We first take the gradient of our level set definition of our surface in order to get a normal vector. We then create a function utilizing **Gram-Schmidt Orthogonalization** that inputs a vector not tangent to $\mathcal{S}$, $\vec{\mathbf{u}}_0'$, and a point on the surface, $\vec{\mathbf{x}}_n$. We then get a tangent vector $\vec{\mathbf{j}} = \vec{\mathbf{u}}_0' - \frac{\vec{\mathbf{n}}_{x_n}\cdot\vec{\mathbf{u}}_0'}{\vec{\mathbf{n}}_{x_n}\cdot\vec{\mathbf{n}}_{x_n}}\vec{\mathbf{n}}_{x_n}$ After that, we create a function that takes a point on $\mathcal{S}$ and a tangent vector and updates the direction and location once. This will be done by taking our step size, ε, and iterating our geodesic in the direction of our initial tangent. That is, we get $x_0' = x_0 + \varepsilon \cdot \vec{\mathbf{u}}_0$. where x_0 is a point on $\mathcal{S}$ and $\vec{\mathbf{u}}_0$ is a vector tangent to the surface. Since $\mathcal{S}$ has steep curvature near the poles, we'll have to take smaller steps in order to prevent the erratic behavior of the geodesics. We'll let $\varepsilon = s/150$ when we're around the poles, and we'll take ε size steps otherwise. Now we create a function that approximates the entire geodesic. We use a for loop that iterates from 1 to N, where N is our step size. We'll create a matrix that stores all positions and directions of our geodesic. So for every iterate, the matrix will increase size. In order to produce a unique geodesic, we need a point and tangent vector. Computing a point is straightforward. To produce a tangent vector, we take the **Jacobian**, J, of the parameterization definitions of our surface. Since $\mathcal{S}$ has nonzero curvature, we convert the Jacobian of our surface into polar coordinates. This is done by taking the matrix representation of the polar coordinate map and multiplying that by the Jacobian in order to get a tangent vector. Observe the following surfaces with geodesic flow.

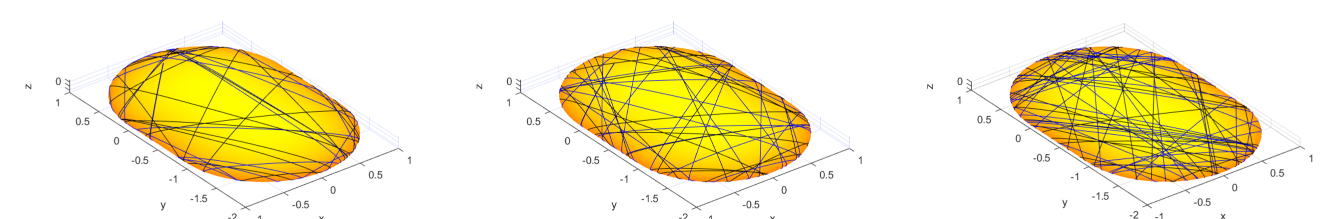

FIG. 2: 2 geodesics on our biscuit surface with N = 50k, 75k, and 100k steps.

Now that we are able to produce geodesic flow on our surface, we can determine λ which is defined as the exponential growth rate of how quickly nearby trajectories separate. Thus, we must produce two geodesics that start off fairly close to each other, and compute the difference of their tangent vectors. Since the geodesics will eventually diverge and possibly be on different parts of the surface, they may have different lengths due to a slower step size near the poles. Thus, we must truncate points from the larger geodesic. Now we use methods of **linear regression** to determine λ. We'll find an association between the logarithm of the distance between nearby trajectories vs. the number of iterates. Under certain conditions, we get the following plots.

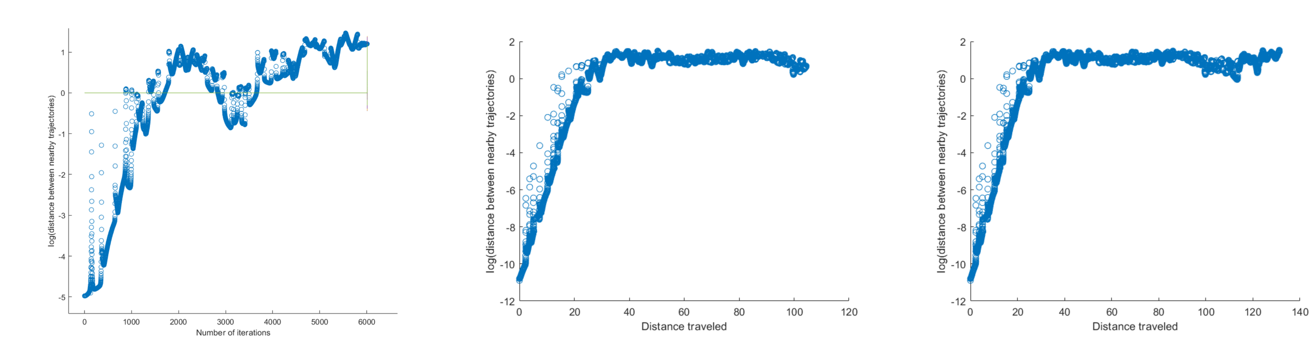

FIG. 3: The corresponding linear regression plots of the systems in Figure 2.

Results

Using methods of linear regression and an ensemble of initial conditions, we find that $\lambda > 0$. We wanted to observe if λ was invariant under changes in initial conditions, so we experimented with different initial displacements, as well as different step sizes and flattening factors. We only considered the initial positive slope of the plot when determining λ. It seems to stabilize between 0.45 to 0.65.

Conclusions and Further Study

By observing a few figures for sufficiently high s, we can observe that the geodesics seem to converge to the **Bunimovich stadium billiard**. Observe the following plots.

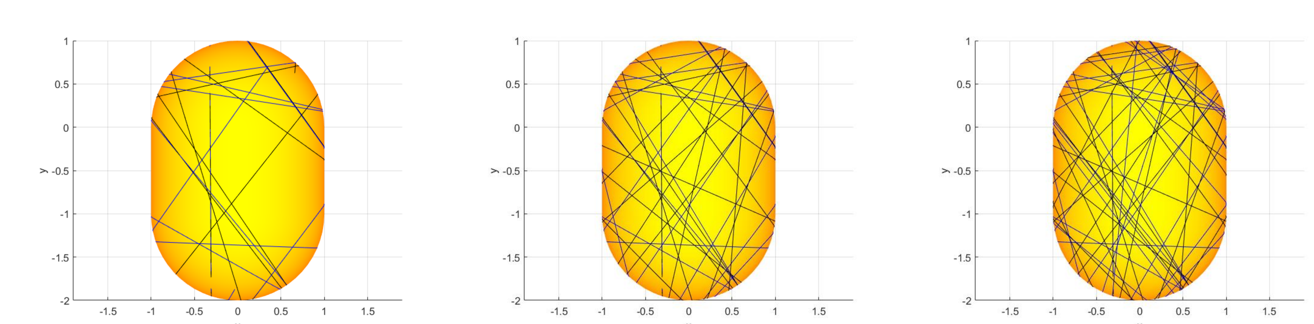

FIG. 4: 2D Projection of our system with N = 25k, 50k, and 75k steps.

Since λ tends to exhibit chaotic behavior, we can conclude that the numerical approximation of this system with non-negative curvature has characteristics of chaotic behavior. After observing the numerical counterpart of this system, it is natural to consider how would one prove this rigorously. We could also determine the topological entropy and measure-theoretic entropy next. We can then try to generalize these results for manifolds for higher dimensions, as well as observe how the curvature changes when a surface is being flattened in one direction.

References

[1] Crane, Livesu, Puppo, Qin, July 2020, *A Survey of Algorithms for Geodesic Paths and Distances*

[2] Mildenhall, April 2017, *Chaotic Systems of Geodesics on Surfaces of Revolution*

[3] Kennedy, Narayan, Saniee, Nov. 2021 *On the Hyperbolicity of Large-Scale Networks*