Analyzing the Tent Map and Its Chaotic Behavior

Brandon Poblette
Gonzaga University
Washington, United States
bpoblette@zagmail.gonzaga.edu

Arjuna Herbst
Gonzaga University
Washington, United States
aherbst@zagmail.gonzaga.edu

The purpose of this research paper is to analyze the chaotic behavior of the tent map. In this paper, we are going to examine the fixed points, the stability of the fixed points, analysis of several orbit diagrams, analysis of the time series plots, the Schwarzian derivative, Lyapunov exponent, bifurcation diagram, and histograms of the tent map.

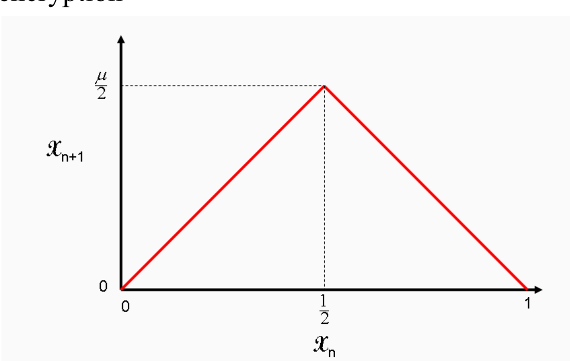
I. Introduction

The tent map is a piecewise function which can be defined as follows:

$$f(x) = \begin{cases} x & \text{if } 0 \leq x \leq \frac{1}{2} \\ 1 - x & \text{if } \frac{1}{2} \leq x \leq 1 \end{cases}$$

In our analysis, we put a coefficient of *a* where *a* is a value greater than 0. Any value less than 0 would just result in a reflection of the tent map thus we decided it wouldn't be necessary for our analysis.

The tent map is a piece wise function where it is ax from the interval 0 to ½ and a(1-x) from the interval ½ to 1. Chaos in economics, Image encryption



As we can see on the image above, the shape of the map looks like a tent which is where the map get's its name from. It's important to note that in this diagram mu is the same as the *a* value which we manipulated in our research. The value a is responsible for the height of the graph and determines how the graph is stretched.

II. Methods

In this section, we will go over the methodologies used in our research of the tent map. To begin, let's go over the specs of the machine used to iterate over our map. Tent map iterations were performed in Python version 3.10.11 on a Dell XPS 15 9500 with an Intel(R) Core(TM) i7- 10875H CPU @ 2.30 GHz, 32GB RAM, x64-based processor running Windows 11 Home. Many of our plots and numerical values did not require a large amount of iterations to calculate, but having a faster CPU certainly helped in our research.

The first algorithm we will explain is the algorithm we used to calculate our Lyapunov Exponents.

Tangent Vector Update:
Within the iterative process, the code line:

```
v = a * (1 - 2 * abs(x - 0.5)) * v
```

updates the tangent vector v in accordance with the dynamic evolution of chaotic systems, as expressed by the formula:

$v = a(1-2\,|x - 0.5|) * v$

Lyapunov Exponent Accumulation:
During each iteration, the code accumulates the logarithm of the norm of the tangent vector:

```
sum_lyapunov += np.log(np.linalg.norm(v))
```

This corresponds to the summation of Lyapunov exponents, reflecting the system's sensitivity to initial conditions.

Average Lyapunov Exponent Calculation:
Following the iterative process, the average Lyapunov exponent λ is computed as:

```
return sum_lyapunov / n_iter
```

This calculation provides a measure of the system's overall sensitivity to initial conditions, essential for understanding its chaotic properties.

Now, let's talk about the method used to generate frequency histograms for our map.

This function iteratively generates the trajectory of the tent map starting from the initial condition x0.

```
tent_map_iterations(x0, a, n)
```

It computes n iterations of the tent map, storing each value of *x* in a list:

```
x_values = tent_map_iterations(x0, a, n_iter)
```

Frequency Histogram Generation:
This function utilizes the generated trajectory of the tent map to construct a frequency histogram:

```
frequency_histogram(x0, a, n_iter, num_bins)
```

This histogram visualizes the distribution of *x* values obtained from the tent map iterations, providing insights into its dynamic behavior.

Bifurcation Diagram Generation:
This function generates a bifurcation diagram for the tent map:

```
tent_bifurcation(a_values, x0, n_skip, n_iter)
```

The bifurcation diagram illustrates the behavior of the tent map across different parameter values a, providing insights into its dynamic transitions.

Parameter Setup:
The parameter a values are defined as a range of values within a specified interval:

```
a_values = [a/100 for a in range(100, 200)]
```

This range defines the span of *a* values over which the bifurcation diagram will be generated. Our code then iterates over each parameter a value in the specified range via a for loop. Within each iteration, the tent map is iteratively computed to generate the bifurcation diagram.

Cobweb Diagram Generation:
This function generates a cobweb diagram for the tent map:

```
cobweb(x0, a, n)
```

The cobweb diagram visualizes the iterated dynamics of the tent map starting from an initial condition *x0*, providing insights into its behavior.

The function iterates over a specified number of steps:

```
for _ in range(n):
```

Within each iteration, it computes the next value *y* of the tent map using the current value *x*.

III. Results

To find the fixed points:

ax for 0 <= x <= 1/2

ax = x

Fixed point at x = 0 and a = 1

a(1-x) for ½ <= x <= 1

a(1-x) = x

For x:

Fixed point at a/a+1 where a ≠ 0

If *F'(x)* where *x* is the fixed point is less than 1 then the point is attracting.

F(x) = ax

F'(0) = a

This means that the stability of the fixed point is reliant on the a value.

If 0 < a < 1,

Then the fixed point is attracting towards 0.

If a > 1 then the fixed point is repelling

For the fixed point at a/a+1

F(x) = a(1-x)

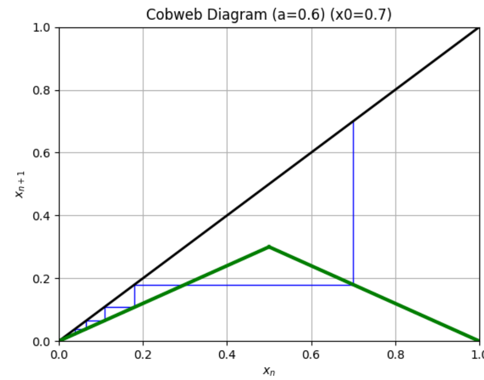
F'(a/a+1) = -a

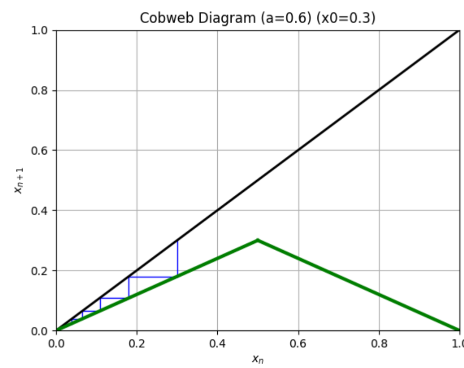
The fixed point at a/a+1 is attracting if a is greater than -1 is repelling when a < -1

Cobweb Diagrams:

Cobweb diagrams serve as indispensable tools for comprehensively analyzing the dynamics of the tent map. We tested a number of different *a* and *x0* values in order to get the best results.

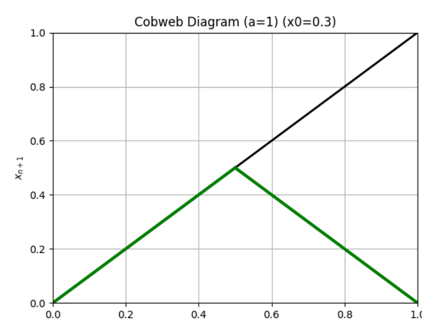
Cobweb Diagrams — a < 1:



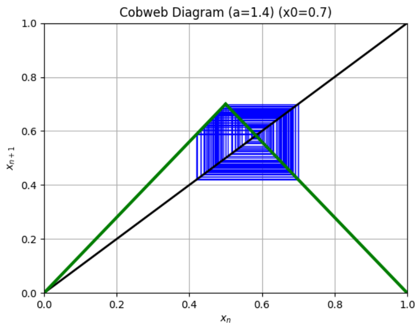


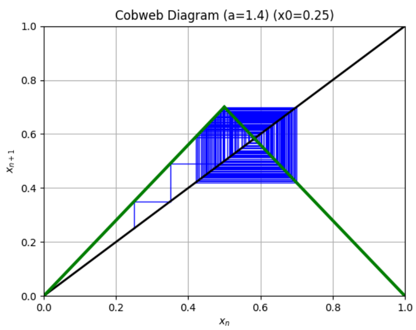
Here we see convergence to the fixed point at *x* = 0 occur.

Cobweb Diagrams — a = 1:



Cobweb Diagrams — a > 1:



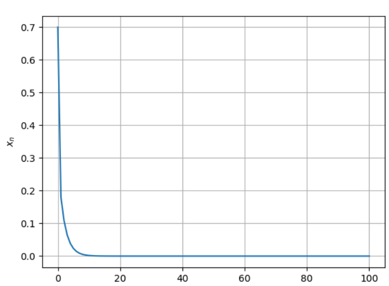


Here we see the tent map begin to display chaotic behavior.

Time Series plots:

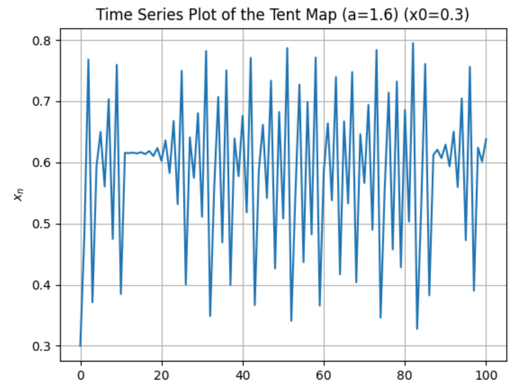
Time series plots play a pivotal role in understanding the dynamics of chaotic systems, including the tent map. By visualizing the evolution of the system over time, we were able to gain valuable insights into its behavior and characteristics.

Time Series plot when a < 1:



This time series plot again reinforces the idea that when $a < 1$, convergence to the fixed point at $x = 0$ occurs.

Time Series plot when a > 1:



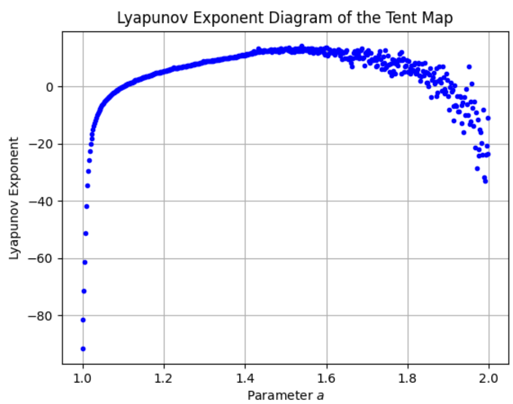
Similar to the cobweb diagrams, we see the tent map begin to showcase chaotic behavior when $a > 1$.

Schwarzian Derivative:

$$(Sf)(z) = \left(\frac{f''(z)}{f'(z)}\right)' - \frac{1}{2}\left(\frac{f''(z)}{f'(z)}\right)^2 = \frac{f'''(z)}{f'(z)} - \frac{3}{2}\left(\frac{f''(z)}{f'(z)}\right)^2.$$

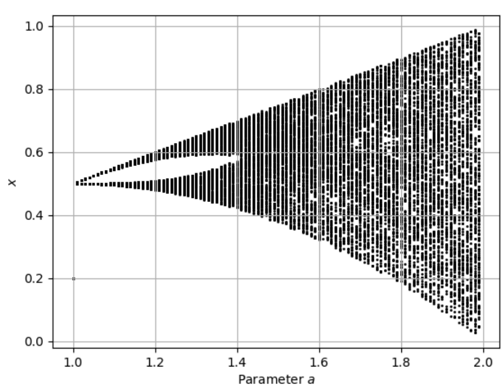
ax for 0 <= x <= ½	f'(x) = -a
f'(x) = a	f''(x) = 0
f''(x) = 0	f'''(x) = 0
f'''(x) = 0	
a(1-x) for ½ <= x <= 1	

Lyapunov Exponent Diagram:



Since negative Lyapunov exponents indicate stable behavior, and positive Lyapunov exponents indicate chaotic behavior, we again see the idea that when our *a* parameter is greater than 1, the tent map showcases chaotic behavior.

Bifurcation Diagram:

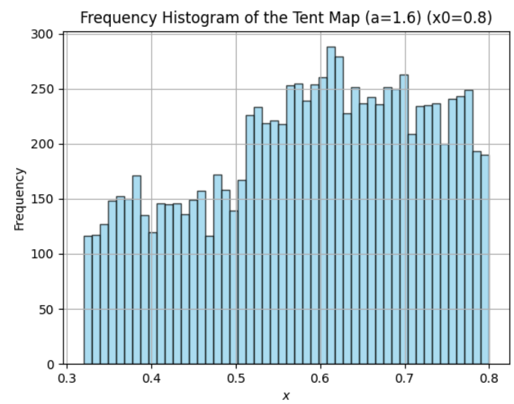


Here we see some interesting things happening, period doubling bifurcations at 1.4 > *a* > 1.0, and then chaotic behavior at *a* > 1.4.

Frequency Histograms:

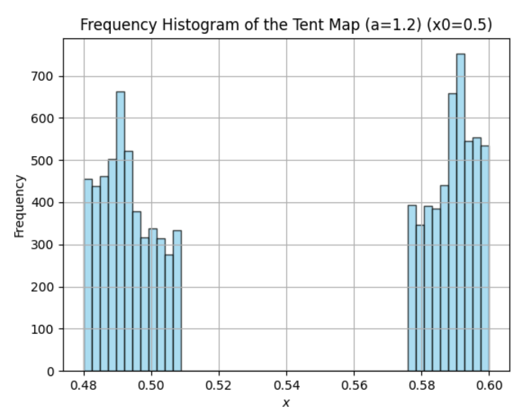
Frequency histograms provide a quantitative analysis of the distribution of values generated by the tent map, offering valuable insights into its underlying dynamics and properties.

Frequency Histogram when a > 1:



This histogram further reinforces the idea that when *a* > 1, the tent map enters it's chaotic regime.

Frequency Histogram when a = 1:



Here we see two sets of stable fixed points, which matches up with what we see in the bifurcation diagram when *a* = 1.

IV. Discussion

Fixed Points: To find the fixed points of the map we have to find the derivative of both functions. The fixed points for the interval 0 to ½ is x = 0 and a = 1. For the function a times (1-x), we found a fixed point at 1/a+1 where a does not equal 0.

Fixed Points stability: What we found for our research was that the stability of the fixed points is dependent on our *a* value. For the fixed point at x = 0, where a is in the interval 0 to 1, the fixed point is attracting to 0. For any value greater than 1, the fixed point is repelling. For the fixed point at a/a+1, the fixed point if attracting if a is greater than -1 and repelling when a < -1.

Schwarzian Derivative: We concluded that the Schwarzian derivative of our function was in fact 0. So while that might seem like a boring answer, this actually tells us a couple things about the map. For example, one thing it tells us is that the map has conformal geometry. What this means is the value of the angle is preserved after a transformation. In other words, no matter what value we use for a, the angle of the tent is the same. A Schwarzian derivative of 0 also means that there is an absence of non-linear distortions. This means that there lacks a deformation that significantly alters the geometry of the map.

Bifurcation Diagram: In this region there are still have stable fixed points and our map hasn't entered chaos. It is at around 1.4 where the tent map enters it's chaotic regime, and there is no longer periodic orbits. Although we didn't show it on this graph, the bifurcation diagram from 0 to 1 is a straight line at 0. This makes sense considering any a value less than 1 should be attracting towards 0.

V. Conclusion and Future Work

In conclusion, our comprehensive analysis of the tent map has provided profound insights into its dynamic behavior across a spectrum of parameter values. Beginning with the stability analysis, we established that the map remains generally stable when a is less than 1.4, with a notable convergence towards 0 evident in our graphical representations. Within the range of a values from 1 to approximately 1.4, we observed a fascinating double period bifurcation, characterized by the emergence of multiple stable fixed points.

Further delving into the nature of fixed points, we meticulously examined their existence and stability by scrutinizing the derivatives of the map functions. Our findings revealed intriguing patterns: for instance, the fixed point at

$x = 0$ exhibits attracting behavior for a values within the interval [0, 1], transitioning to repelling behavior for $a > 1$. Similarly, the fixed point at $a + 1/a$ demonstrates attracting behavior for $a > -1$ and repelling behavior for $a < -1$. This nuanced understanding underscores the intricate relationship between parameter values and the dynamics of the tent map.

Moreover, our investigation into the Schwarzian derivative yielded a remarkable result of 0, indicating the map's conformal geometry and the absence of significant nonlinear distortions. This insight not only elucidates the preservation of angles across transformations but also underscores the map's inherent structural stability, devoid of drastic geometric deformations.

Additionally, the linear behavior of the bifurcation diagram for $a < 1$ aligns with theoretical expectations, affirming the attractor towards 0 for a values below 1.

Overall, our comprehensive exploration of the tent map has not only elucidated its dynamic behavior across varying parameter values but also shed light on fundamental principles governing nonlinear dynamical systems. These findings contribute significantly to the broader understanding of chaotic systems and pave the way for further exploration.

Sources:

Cook, John D. “The Tent Map and Its Lyapunov Exponent.” *John D. Cook | Applied Mathematics Consulting*, 30 Sept. 2022, www.johndcook.com/blog/2022/02/09/tent-map/.

McClure, Mark. “Chaos and Fractals.” *Tent Maps and the Cantor Set*, www.marksmath.org/classes/Fall2023ChaosAndFractals/chaos_and_fractals_2023/section-tent_maps.html. Accessed 9 May 2024.

“Tent Map.” *Wikipedia*, Wikimedia Foundation, 19 Apr. 2024, en.wikipedia.org/wiki/Tent_map.

Thornton, Stephen T. “Step by Step Solution.” *Vaia*, www.vaia.com/en-us/textbooks/physics/classical-dynamics-of-particles-and-systems-5-edition/chapter-4/problem-17-the-tent-map-is-represented-by-the-following-iter/. Accessed 9 May 2024.