Problem Set #6 —— Poincare Sections

University of Colorado - Boulder
CSCI4446 — Chaotic Dynamics

JACK MATHEWS
john.mathewsiii@colorado.edu

February 27, 2026

1 Temporal Poincare Section

My Poincare section program runs a full RK4 simulation and then goes through the list of generated states to find those whose $t-$value is approximately equal to any natural number multiple of a user-defined T value, *i.e.*

$$t = nT, \qquad \forall n \in \mathbb{N}$$

If T is very specific, there will be few instances where a point satisfies the above equation exactly, and if T is irrational, there will be zero instances. To handle this, my program takes the instances where it is *nearest* to satisfying the equation. Any time a point $t_{k-1} < nT \leq t_k$, I accept t_k as a point and then update nT as $(n+1)T$.

See my Github Repository if interested in the specifics of the implementation. The logic for the Poincare section is in `ode_solver/src/section.cpp`.

1.1 Poincare Section at Natural Frequency

When T is equal to the natural period of the unforced, undamped pendulum, we expect the results of a Poincare section to be a small cluster of points at the initial condition, provided that the initial velocity $\omega_0 = 0$—a different velocity would result in a different point being reached. After all, the natural period is defined to be the time it takes for the pendulum to return to the same point after a full oscillation. For smaller and smaller initial position θ_0, this cluster would look more like a single point, as the natural frequency is only accurate for small enough swings.

Instead of calculating the natural period by the formula $T_{natural} = 2\pi\sqrt{\frac{l}{g}}$, I ran a full simulation with $\theta_0 = 0.1$ and $\omega_0 = 0$ and printed the time values whenever $\theta > 0.0999$, *i.e.* when $\theta \approx 0.1$. The resulting time was 0.635 seconds. Then, I let $T = 0.635$ and ran a Poincare section simulation, producing the plot in Figure 1.

As we can see, the results were close to the expectation. In this simulation, the pendulum completed 31 full oscillations, and the program found 20 instances where $t \pmod{T} < 10^{-12}$—this condition

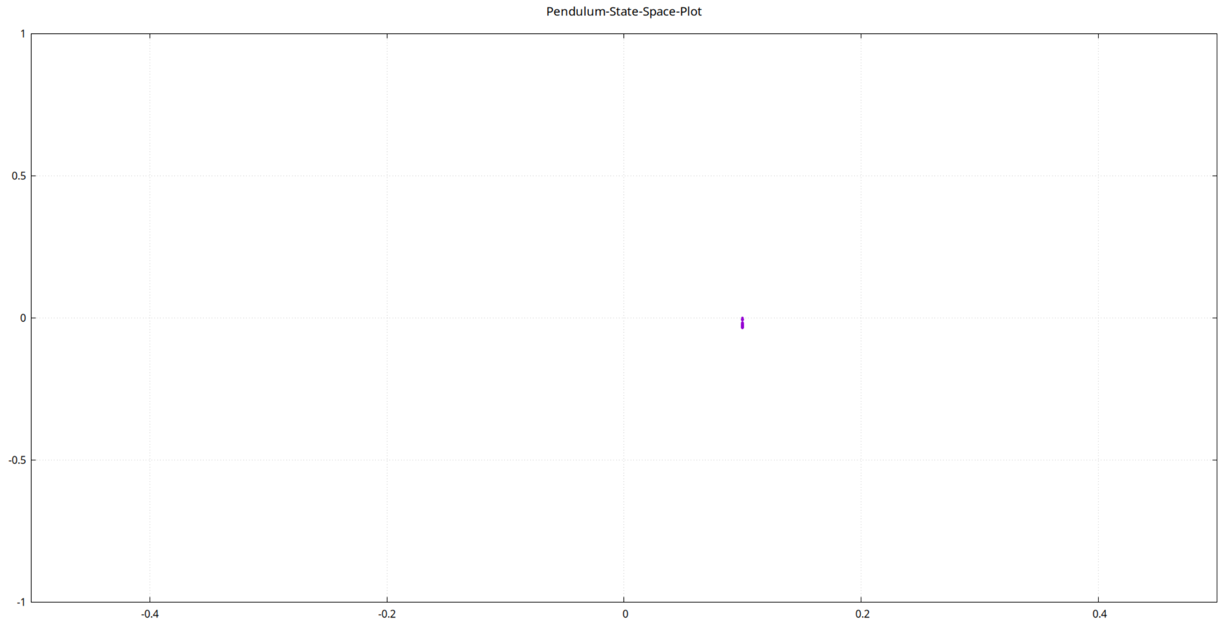

Figure 1: Poincare Section at Natural Period of the Pendulum

was introduced for this simple version of the problem to increase accuracy. All such instances are plotted in 1, and all are in the cluster around $(0.1, 0)$.

1.2 Poincare Section at Arbitrary Period

Figure 2 shows a Poincare section at $T = 0.111n$, a completely arbitrary number. The simulation ran for exactly the same number of iterations as 1, but the results were completely different (as we would expect). Since T had no relation to the physics, the pendulum was sampled at relatively-evenly-spaced points in the state space, creating a nearly-full state space trajectory.

1.3 Poincare Section of Chaotic Trajectory

To get chaotic result, I set the damping coefficient to $\beta = 0.25$, the drive amplitude to $A = 1.2$, and the drive frequency to $\alpha = 5$. I then set the initial position of the pendulum to $\theta_0 = 1.1$ and the initial velocity to zero. With $\alpha = 5$, the period of the pendulum is

$$\begin{aligned} T_{forcing} &= \frac{2\pi}{5} \\ &\approx 1.256637061 \end{aligned}$$

I let this be the value for the temporal hyperplane and generated the result in Figure 3. The simulation ran for 600 000 iterations, and I discarded half of them to remove the transient. This gave me 1 193 points.

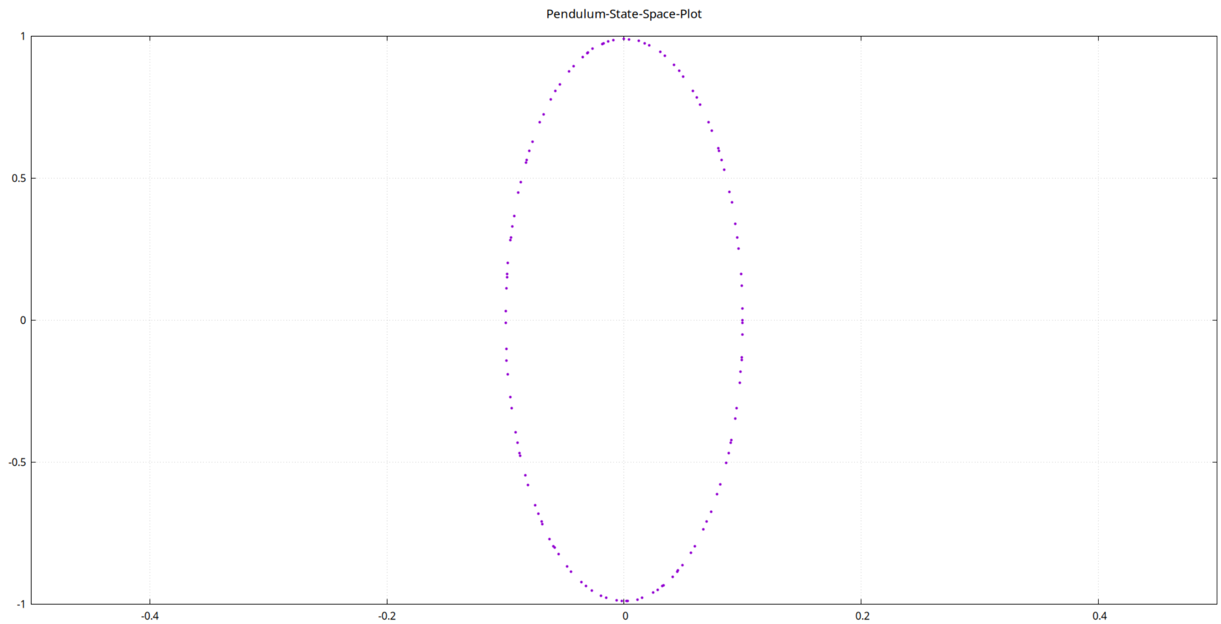

Figure 2: Poincare Section at Period Unrelated to the Natural Frequency

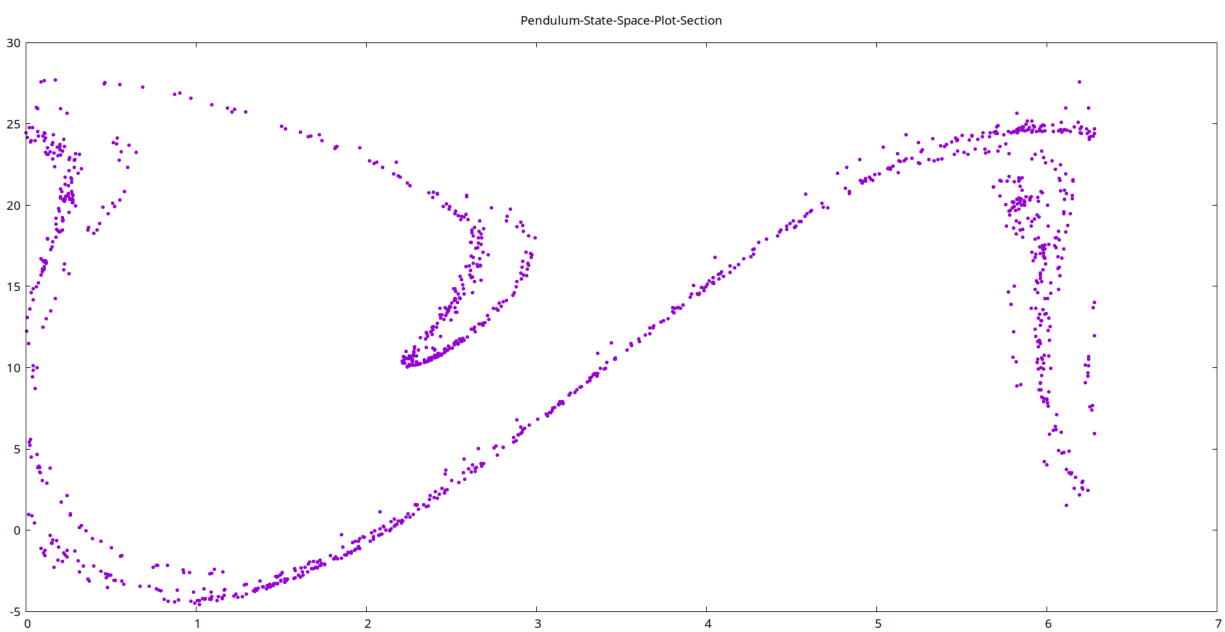

Figure 3: Poincare Section of Chaotic Trajectory Sampled at Forcing Function's Period

1.4 Time Step Analysis

For 3, I used a step size of $dt = 0.005$. When I doubled the step size, the results were "fuzzier". The structure was the same, but the points were further apart, *i.e.* the "lines" produced by the points were less coherent. The reason for this is straightforward. With my current program, the points in the graph are not sampled at exactly nT, they are located at the times nearest to nT that the step size can accomodate. For example, with $dt = 0.01$, the first point would not be sampled at exactly $t = 1.256637061$, but at $t = 1.26$. For smaller time steps, this error becomes smaller, leading tighter clusters of points in the plot.

2 Interpolation

To improve the accuracy of the points being sampled, I applied linear interpolation between the two points in state space on either side of nT. For example, instead of taking the point at $t = 1.26$, I approximated where the pendulum might be at $t = 1.256637061$ via linear interpolation. The method is as follows:

Given two points in time t_{k-1} and t_k such that $t_{k-1} < nT < t_k$ for some positive integer n, calculate the distance between t_{k-1} and T divided by the distance between t_{k-1} and t_k (dt):

$$L = \frac{T - t_{k-1}}{t_k - t_{k-1}}$$

Then, determine the difference vector between the points associated with t_k and t_{k-1}

$$\vec{x}_d = \vec{x}_k - \vec{x}_{k-1}$$

The approximate location of the pendulum's trajectory at $t = T$, $\vec{x}_T$, can be found by starting at $\vec{x}_{k-1}$ and traveling along a portion the difference vector $\vec{x}_d$. That portion is L. Thus

$$\vec{x}_T = \vec{x}_{k-1} + L\vec{x}_d$$

and $\vec{x}_T$ is stored for the Poincare section. Once again, the implementation is in the Github repo.

2.1 Chaotic Trajectory Revisited

Using the interpolation approach, I was able to get a more accurate representation of the Poincare section from 3. As we can see in Figure 4, the points are much closer together.

2.2 Time Step Analysis Revisited

Once again, doubling the time step decreased accuracy, but in this case, the differences were not as obvious. The points are still quite close together, and the lines they create are distinct. In fact, the picture is more similar to 4 than 3.

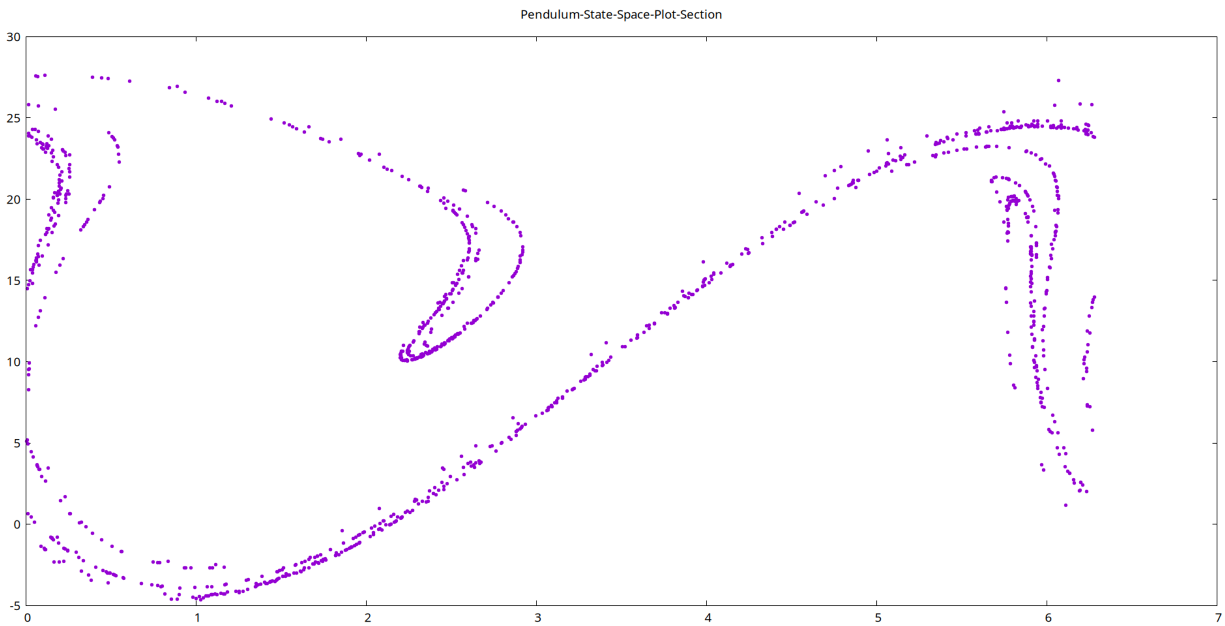

Figure 4: Poincare Section of Chaotic Trajectory with Interpolation