Ryan Riley
CSCI4446
Problem Set 6
27 February 2017

Ryan Riley
CSCI4446 - PS6

Problem 1

In this problem, a simple Poincare section algorithm was written in Matlab. The code looks for the first point that occurs after a given time mulitple.

(a)

The following is a plot in figure 1 of the temporal Poincare section with $[\theta,\ \omega] = [0.01,\ 0]$ with $\Delta t = 0.005,\ m = 0.1kg,\ l = 0.1m,\ \beta = 0,\ A = 0,\ \alpha = 0.$

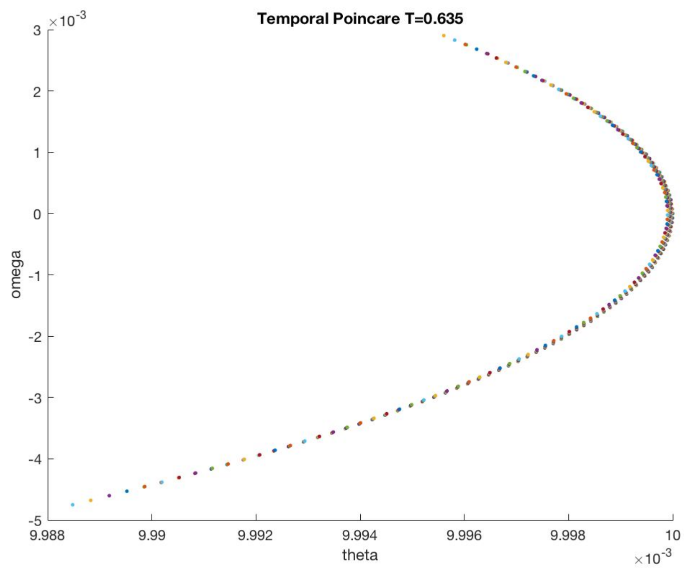

Figure 1

This is to be expected, the system is being sampled at the natural frequency, determined by $2\pi/\sqrt{g/l}$ and so we are only seeing a small segment of the trajectory, roughly once per cycle.

Ryan Riley
CSCI4446 - PS6

(b)

Next, the same system is plotted with a T that is not rationally related to the natural frequency. I picked T = 0.7. The following plot (figure 2 below) displays the results.

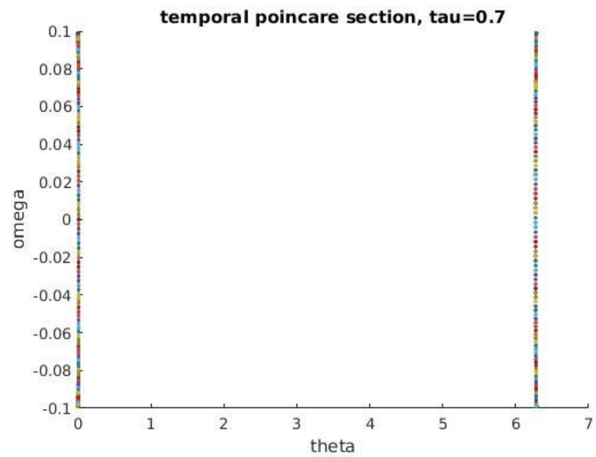

Figure 2

There are noticeably more values of ω shown. Since the sampling rate is not related to the natural frequency, more of the state space is being captured with more samples. We are seeing more of the complete trajectory.

(c)

Next, a chaotic trajectory that was used in problem set 6 is plotted, the results are seen below in figure 3.

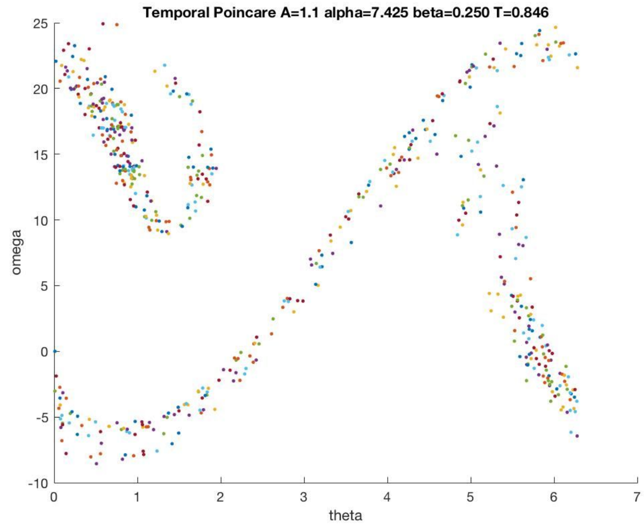

Figure 3

(d)

If the timestep is increased while keeping all other parameters constant including the total time, the plot becomes “sloppier.” Points fall off the actual trajectory and fill a less confined space. This is because the naive Poincare algorithm is simply looking for the first point after the given point in time. If the timestep is bigger there is a greater chance that there is a large difference between the actual point at the given time and the point that is returned by the naive Poincare algorithm.

Ryan Riley
CSCI4446 - PS6

Problem 2

In this problem, a slightly more intelligent Poincare algorithm was written in Matlab that interpolates the precise point on the trajectory at the exact point in time.

Figure 4, below, is a repeat of problem 1c in which a chaotic trajectory was plotted. This time the smarter, interpolated Poincare section was used.

Figure 5, below, is a repeat of 1d in which the same trajectory is plotted but the timestep increased

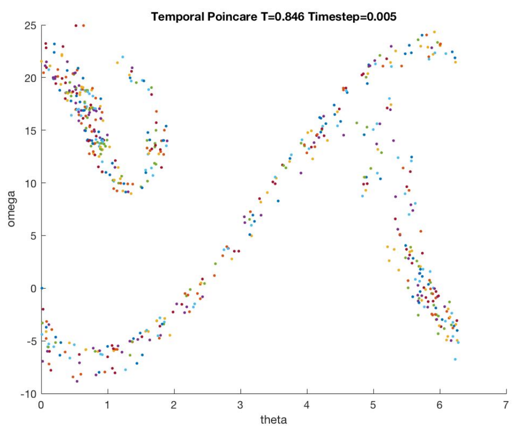

Figure 4

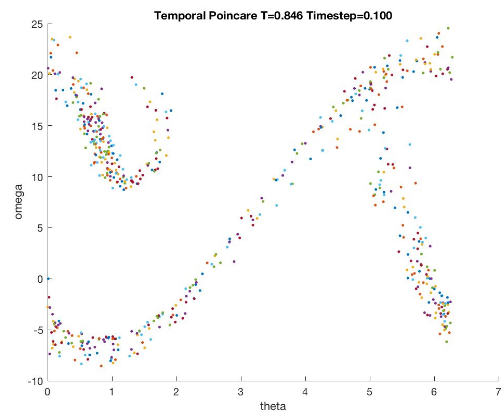

Figure 5

It is apparent that these plots are very similar. This similarity was not shared between the more naive Poincare section. Essentially, the linear interpolation is making up for the error induced by the larger timestep. When the naive Poincare algorithm selects the first point after the point in time, regardless of the distance from the true time, the interpolated version estimates its way back to the true point in time.