Python DeCal Final Project:
Quantizing the Kapitza Pendulum

SuryaNeil Ahuja

December 3, 2023

1 Introduction and Background

The Kapitza Pendulum (also called inverted pendulum) is a type of driven harmonic oscillator where the support point is oscillated up and down rapidly relative to the frequency of the pendulum oscillations. Throughout the project the frequency of oscillation will be called ω, and the driving frequency will be called γ. Here, we are in the regime where $\gamma \gg \omega$.

In the Kapitza Pendulum, this high frequency driving causes the point where the pendulum is upside-down to become a stable point (previously an unstable point). As the frequency is increased, this stable regime at the top gets wider. You can then move the pendulum from the stable regime at the bottom into the top, and it will oscillate upside-down (hence inverted pendulum).

This scheme provides a very natural way to quantize the Kapitza pendulum, where the top and bottom states become two *independent Quantum Harmonic Oscillator potentials* that are separated far enough apart such that they don't interact.

As mentioned in the proposal, the origins for this project started much before, and is part of a larger reading/research project with a postdoc in the Physics department. Coding a simulation of the Kaptiza Pendulum was the natural step and aligned very well with the skills I learned in the Python DeCal throughout the semester, making it a fitting final project. For more background on the original premise, see the project proposal.

Now, to the project itself. To summarize how I fulfilled the requirements:

1. I generated test data from a simulation of two independent Quantum Harmonic Oscillators (this is the approximation suggested by the theory outlined above and in the project). I then compared them to the analytically calculated solutions, which I obtained from the Lagrangian outlined in [4] §30, combined with my own calculation for the quantization.

2. The result of the generated data was many different spectra for all the different values of the many constants (e.g. frequency, amplitude, mass, etc.). I filtered the generated data by tuning the various parameters of

the system, by examining the spectra and eigenfunctions to see when the spectra were most linear (will explain why in Section 3.

3. The one parameter I did not tune initially was γ, because the stability of the classical Kaptiza Pendulum depends on $\gamma \gg \omega$, and this project revolves around the quantum stability of the quantized system. I then fit curves to the generated data (for many values of γ) and obtained an r^2 correlation coefficient for each linear fit, which I plotted against γ to find where the stability breaks down (nonlinearity of the spectrum). There was some difficulty (and intrigue) when calculating the variance/error (two-point function) from the analytically calculated data, which I will discuss later.

4. See the proposal, [1], [2], [3], and [4] for more background.

In the end, I used a total of three packages: numpy, matplotlib, and scipy (constants, optimize, and stats from scipy). I apologize that the rest of this report ended up so long.

2 Generating Data

2.1 Numerical Data

Generating the numerical data was fairly straightforward. All I needed to do was calculate the Kinetic Energy and Potential Energy for two indepdendent Quantun Harmonic Oscillators, then sum the two together, then calculate the eigenvalues. The equations for both are known, with

$$K = \frac{-\hbar^2}{2m}\frac{n^2}{l^2}\partial_{\hat{x}}^2$$

and

$$U_1 = \frac{1}{2}m\omega_1^2\hat{x}_1^2$$

and

$$U_2 = \frac{1}{2}m\omega_2^2\hat{x}_1^2$$

Here U_1 and U_2 are identical, but are shifted from the origin $\pm\frac{l}{2}$, where l is the total length of the system. ω_1 and ω_2 are given by the analytic calculations in the next subsection.

2.2 Analytic Data

There are quite a few steps and calculations involved here, which for brevity I will skip and cut to the chase, where we see that we can use the ladder operator method (as outlined in [1]). The ladder operators, a and $a^\dagger$, are known, so we can simply write

$$H_1 = \frac{\hbar\omega_1}{2}\left(a^\dagger a + \frac{1}{2}\right)$$

and

$$H_2 = \frac{\hbar\omega_1}{2}\left(a^\dagger a + \frac{1}{2}\right)$$

where

$$\omega_1 = \sqrt{2\omega^2 + \frac{4A^2\gamma^2}{ml^2}}$$

and

$$\omega_2 = \sqrt{-2\omega^2 + \frac{4A^2\gamma^2}{ml^2}}$$

For A being the amplitude of the driving, m being the mass, and l as the length.

With H_1 and H_2, the only thing to ensure is that there is no overlap between the Hamiltonians (since the Kapitza Pendulum has two separate stable regimes, but our potential is quadratic so it will eventually cross, which is bad because it does not reflect the system we want to simulate—ideally the two potentials would be placed so far such that the distance is infinite, but I don't have infinite computational time). The last step is to sum the two Hamiltonians and calculate the eigenvalues.

3 Filtering Data

Filtering the data was perhaps the longest and most frustrating part of this project. There are many different constants to tune, the full list is $\hbar, m, \omega, l, n, A$, and γ. The only new one I have mentioned here is n, which is the number of position sites, and comes as a necessity of the discretization. Initially I had a lot of trouble tuning the parameters properly, the spectra just would not behave as expected, and with so many values to tune it took far too long. Part of the problem was that $\hbar$ is so small relative to everything else, so this threw off a lot of the approximations necessary to simulate the project. After lots of trial and error, I arrived at the simplest solution of setting $m, \omega, \hbar, A = 1$.

Now I only had two parameters to tune n and l (remember γ is left alone since it is the variable we want to focus on to test the approximation). I knew, after generating the spectrum from the analytic calculations, that the spectrum for the numerical data should be linear. I then filtered the data by tuning l (keeping n as high as reasonably possible, since it was the limiting factor on computational time) until the spectrum became most linear. It turns out this happened with best with an $n \sim 1000$ and $l \sim 43$.

4 Curve Fitting

With the filtering from Section 3 in mind, I then set to compute the spectra using the numerical calcution for many different value of γ. We know that the stability of the classical system breaks down when *gamma* $\sim \omega$, so I varied γ in a wide range from 1 to 2000, and performed a linear fit. Figure 1 is one such example of that:

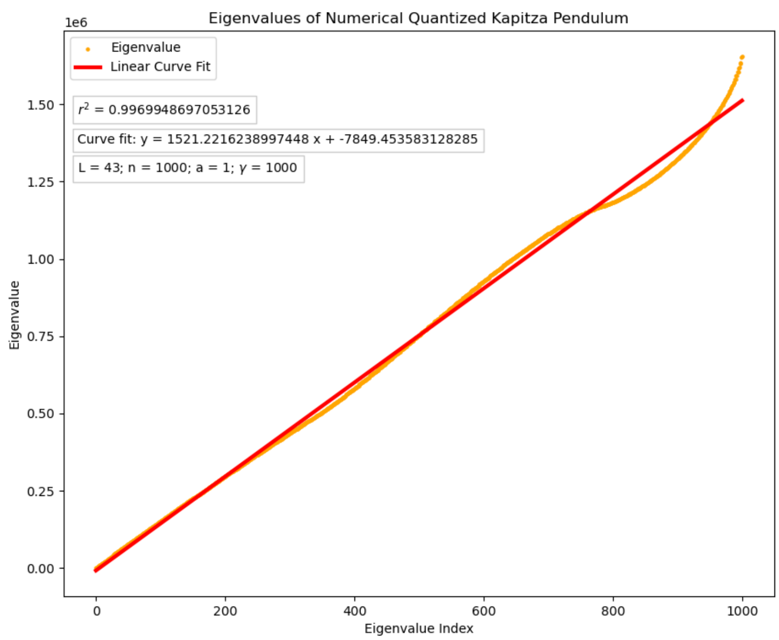

Figure 1: Numerically Computed Spectrum

Using the r^2 value from the linear fit, I plotted this for many values γ, as shown below in Figure 2.

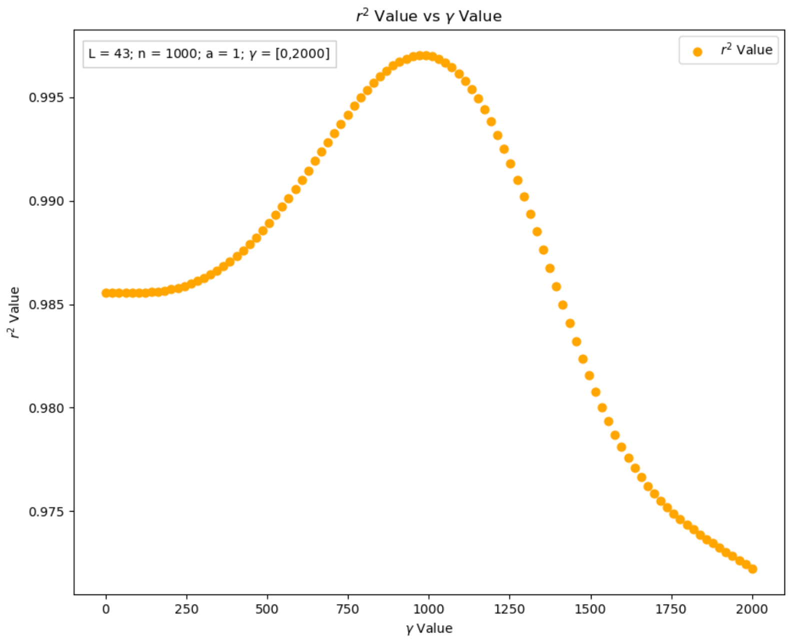

Figure 2: r^2 values versus γ

I attempted to perform a gaussian fit on the r^2 values to obtain an estimate of whichever γ maximizes the r^2, but this turned out to overfit or not work well with the data, so I eventually abandoned the approach.

The results in Figure 2 are quite interesting, as I expected the linearity to get better as γ increased, but this was not the case—after some thinking it is likely because with a higher γ, there are too many frequencies compared to the discrete and finite amount of n position sites. As evidence for this, I set $n = 500$ and the peak moved towards lower γ values.

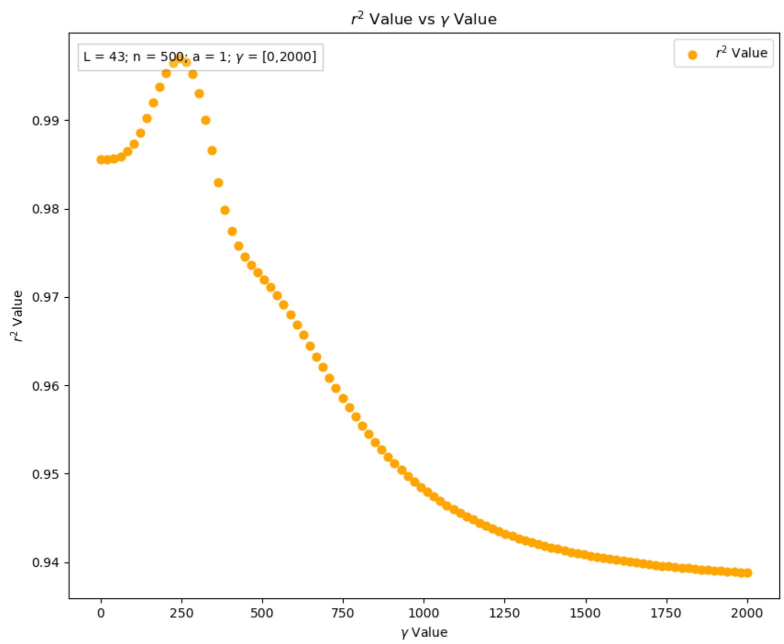

Figure 3: r^2 values versus γ with lower n

As a reference, Figure 4 are the analytically-computed solutions (rescaled so that the y-scale matches), which we can see is completely linear.

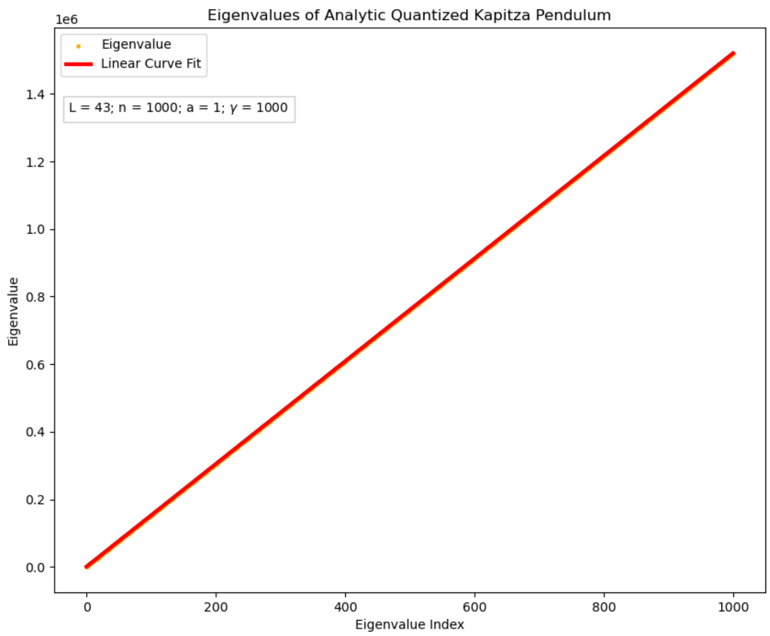

Figure 4: Analytically Computed Spectrum

In both cases I pushed the system to the breaking point in γ, but it actually turned out to be rather difficult (since ω_2 can become complex) once the parameters were tuned properly, which suggests that the quantum stability is highly dependant on the classical stability.

Finally, I calculated the variance between the analytic and numeric data to analyze the error, but the results were very odd. There was some very strange oscillation with the variance, with it jumping up and down from point to point, creating some interesting looking graphs. I'm not sure why this is the case, but the graphs are provided nonetheless

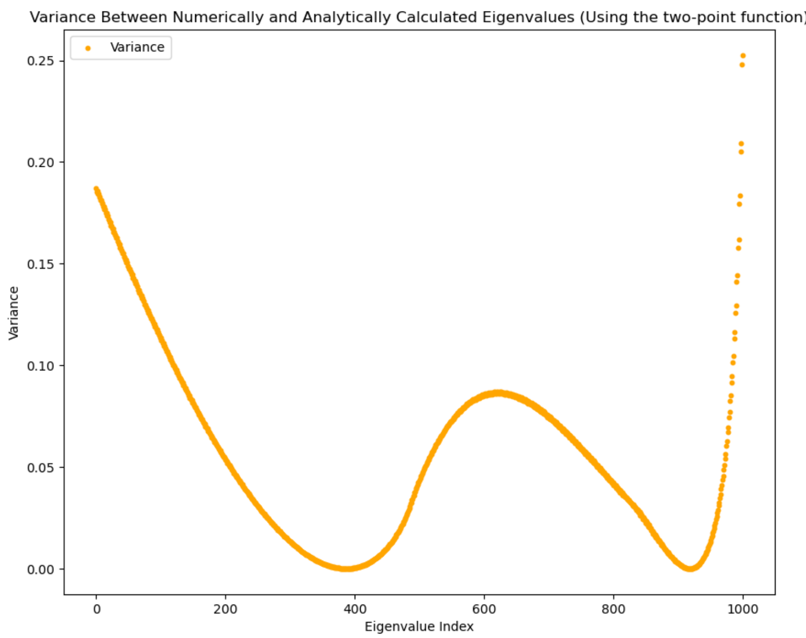

Figure 5: Variance for $n = 1000$ and $\gamma = 1000$

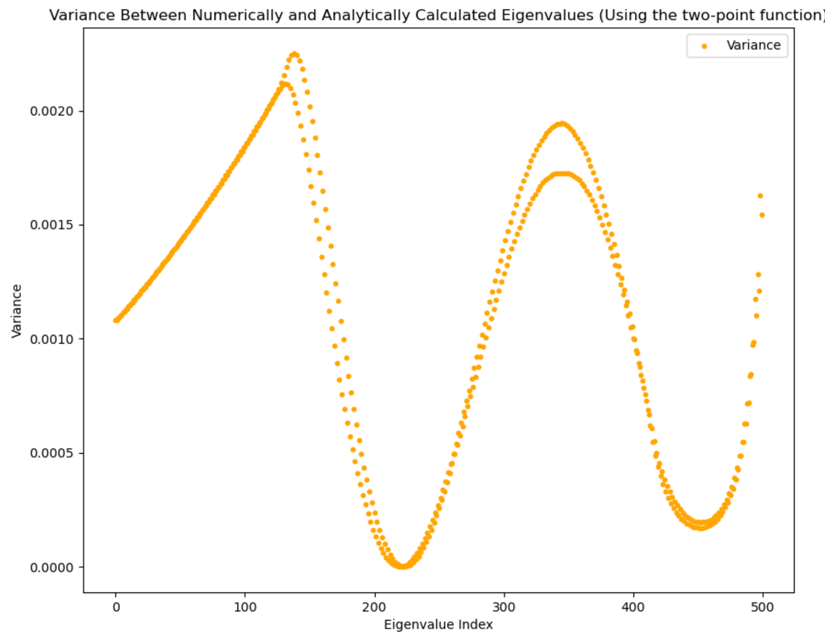

Figure 6: Variance for $n = 500$ and $\gamma = 200$

5 Conclusion

In conclusion, the major result that this project suggests is that the stability of the quantum system is highly dependant on the stability of the classical system as long as the spectrum of our quantum operator is discrete, though there is a lot more that I could talk about, however this 2-3 page report is already triple that length, so I will wrap it up here.

Again, I'm so sorry this report has ended up so long, I learned quite a bit about discretizing the variables to work with the computer, and how to handle lots of messy calculations and graphics cleanly in Python. The next logical step (which I was actually working on first) is to simulate the full quantum problem, but that was far more complex and time-consuming than initially expected, so I focused on the simpler quantizations instead.

References

[1] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.

[2] James S. Howland. "Floquet operators with singular spectrum. I". en. In: *Annales de l'I.H.P. Physique théorique* 50.3 (1989), pp. 309–323. URL: http://www.numdam.org/item/AIHPA_1989__50_3_309_0/.

[3] James S. Howland. "Stability of quantum oscillators". In: *Journal of Physics A: Mathematical and General* 25.19 (Oct. 1992), p. 5177. DOI: 10.1088/0305-4470/25/19/025. URL: https://dx.doi.org/10.1088/0305-4470/25/19/025.

[4] Evgeny Mikhailovich Lifshitz Lev Davidovich Landau. *Mechanics Third Edition*. Butterworth-Heinemann, 1976.