Increasing data quality and quantity in an optical data acquisition system to better model superfluid glitches

Sebastian Serrato

Advisor: Professor Rena J. Zieve

Department of Physics and Astronomy, University of California, Davis

March 21, 2025

Abstract

In this paper, the methods to improve both data quantity and quality in an experimental setup were investigated. Particularly, a numerical algorithm was developed and implemented to determine important values from a previously compressed data set, leading to increased data quality. An annotated version of the code is available at: https://github.com/SupefluidGlitch/Iterative-Algorithm

1 Introduction

At the end of a star's life, if the conditions are right, it will collapse into a neutron star. In these celestial objects, conditions are pushed to the extremes: their densities are second only to black holes, rotational rates increase, and their magnetic fields become significantly stronger. Due to this last factor, neutron stars emit immensely powerful beams of radiation from their poles. This, alongside their rotational behavior, causes beams of radiation to occasionally sweep by Earth, allowing astronomers to observe both the presence of neutron stars as well as their rotational rates. From this, astronomers noted an interesting behavior: amidst the rotational decay of these neutron stars, there would occasionally be sudden increases in rotational velocity, dubbed "glitches", as seen in Figure 1 [1]. While the reason behind this exotic rotational behavior is not fully understood, it is presently believed to be due to the superfluid component of a neutron star's interior.

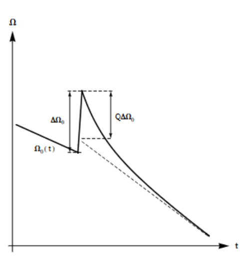


Figure 1: An observed glitch in a neutron star's rotational decay [1].

2 Superfluids

Put simply, superfluids are liquids that have zero viscosity; they exhibit frictionless flow in which kinetic energy is conserved [2]. This causes exotic behavior to occur, such as indefinite rotation, film flow, and superleaks [3]. Of these properties, the one of particular interest is that of indefinite rotation. While it is true that, under ideal circumstances, a superfluid, once spun, will rotate forever, this behavior breaks down under certain conditions. Particularly, when a superfluid is placed in a container and spun fast enough, its rotational rate will indeed decay. Additionally, as the superfluid slows down, it will occasionally exhibit an exotic "glitch" behavior, where the container's measured angular momentum suddenly spikes [4]. To fully grasp this property, it is first necessary to understand quantum vortices: macroscopic manifestations of the quantum properties inherent to superfluids. These quantum vortices—dubbed "quantum" as their angular momentum can only take on certain discrete values—can be thought of as individual pillars sprawled throughout the liquid that the superfluid circulates about [5], the sum of which leads to the appearance of classical fluid rotation. Additionally, these vortices can "pin" to surfaces, locking the vortices in place while still allowing for the superfluid to circulate about them [5]. Moreover, if there is ever a large enough discrepancy between the rotational rate of the superfluid and the surface it is pinned to, then a depinning event occurs. In this, vortices unlatch

and translate freely throughout the fluid. If these vortices happen to crash into the wall of the container holding the superfluid, then an exchange of angular momentum occurs. In this, the container wall accelerates (due to the imparted angular momentum) and the superfluid, in turn, decelerates (due to the loss of vortices) This exact mechanism is believed to be the underlying reason for the rotational "glitches" observed in neutron stars [6].
Specifically, quantum vortices throughout the neutron star's superfluid interior first pin to the non-superfluid crust (or other normal parts of the star). Then, as time passes, the crust slows down, whilst the superfluid interior maintains its angular momentum. Eventually, the difference in the rotational rate between the crust and the interior reaches a critical point, leading to a mass depinning event. As vortices crash into the crust, angular momentum is transferred and the neutron star's exterior speeds up. The superfluid interior, on the other hand, is speculated to slow down due to the loss of vortices.

3 Description of the Apparatus

While it is currently unfeasible to replicate the interior conditions of a neutron star within a lab, it is certainly possible to recreate the superfluid behavior of its interior.
To do so, an experimental setup based off the work of Tsakadze and Tsakadze [4] was constructed. The apparatus, housed primarily within a dewar, works as follows. The dewar is first cycled through a helium atmosphere to reduce the probability of gaseous impurities from the air condensing within the dewar. Afterwards, liquid helium is transferred into the cavity, which is subsequently pumped on to reduce its temperature below helium-4's lambda point (2.17K), leading to superfluidity. Once a sufficient amount of time has passed, an aluminum bucket, affixed to a stainless steel rod, is submerged in the pool of superfluid helium, filling the container through a small opening in the bucket's cap. Once this process is complete, the bucket is electromagnetically levitated, and a contactless AC motor rotates the rod, and by extension, the superfluid container. At the operator's discretion, the motor is turned off and the bucket is permitted to freely spin down. In our case, we often rotate the container up to 0.1 - 4Hz, then allow it to spin down for hours. Both values are arbitrary, and as a result, really any rotational frequency may be chosen.
Throughout its rotation, an optical system (which is further discussed in the following section) relays the rotational rate back to the computer for analysis and storage. The behavior of this data acquisition system, particularly the methods used to improve data quality and quantity, are the primary focus of this paper.

4 Data Acquisition

Measuring the rotational rate of the superfluid is done through an optical system, which relies on a laser that shines on the outer circumference of the superfluid container lid (Figure 2). Simply put, if the laser lands on a reflective tooth, the optical sensor returns a high intensity reading to the computer; if the laser lands on a gap, the sensor will return a low intensity reading (Figure 3). From this, the rotational period is computed, which is then used to determine the container's rotational frequency.
The greatest advantage of our current system is that, compared to prior experiments, the sampling rate is much higher; rather than a single measurement per revolution, our system receives hundreds of readings per second. Not only does this lead to thousands of data points being gathered per revolution, but it vastly increases the fidelity of our measurements, allowing us to spot more subtle variations in rotational frequency (glitches.)
As the experimental setup was being developed, it was deemed that storing all incoming data would lead to significant system slowdowns, impeding the accuracy of measurements. For this reason, the data is preemptively analyzed (through a microcomputer) prior to storage. In this analysis script, a threshold value (dubbed the midpoint) is determined prior to startup.

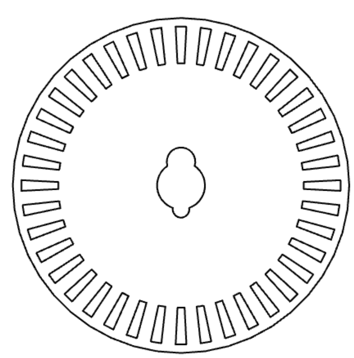


Figure 2: The container lid, with 40 teeth and 40 gaps on the outer diameter.

Once the incoming readings cross this threshold (for example, as intensity values begin to increase), the computer begins to calculate a running average of all incoming data. Then, once the threshold is crossed again (this time as intensity values begin to decrease), the computer saves the final average intensity alongside a count of how many datapoints were averaged (a measure directly proportional to time). From this, the rotational period (and by extension, frequency) of the container may be determined. Importantly, as the container lid has 40 teeth and 40 gaps, there are 80 potential readings per revolution. As a result, the ensuing data resolution is much higher than that of the Tsakadze's setup, where only a single reading was made per revolution [4]. Unfortunately, this system is not without flaw. As the temperature necessary for superfluid helium to exist is approached, an array of mechanical and technical issues begin to arise. Most notably, the intensity values start to vary throughout the experiment: rather than receiving perfect maximum intensity values when the laser is above a reflective tooth, the sensor may read back a value half as strong. This behavior is presumed to occur due to the condensation of ice on the container lid, which would lower the intensity of the reflected light.

As the analysis script relies on each waveform having a fixed, identical midpoint (and by extension, an identical peak), when intensity begins to decay, errors start to appear in the saved data. For example, if the midpoint is set too high (Fig 4c), then when intensity decay occurs, entire readings may be missed, leading to longer intervals that actually present. On the other hand, if the midpoint is set too low, then an oscillatory artifact arises in the data (Fig 4a). The reason behind this behavior is due to the time it takes for light to traverse fully from gap to tooth, or vice-versa. As the laser shines on the edge of a tooth, only a partial intensity reading is returned, which causes the data to take the form of a trapezoidal wave (as opposed to a square wave) (Figure 3). If, then, the midpoint is set too low, a large portion of the base of the trapezoid is included the measurement, which leads to a larger value for the time between the rise and fall of the intensity readings. Subsequently, the following measurement will be significantly smaller, as the space between two waveforms is now foreshortened. So, with this midpoint issue at hand, the question becomes not only a matter of what the perfect midpoint is prior to compression, but if a varying (dynamic) midpoint can be determined from the available data and applied retroactively to enhance data quality.

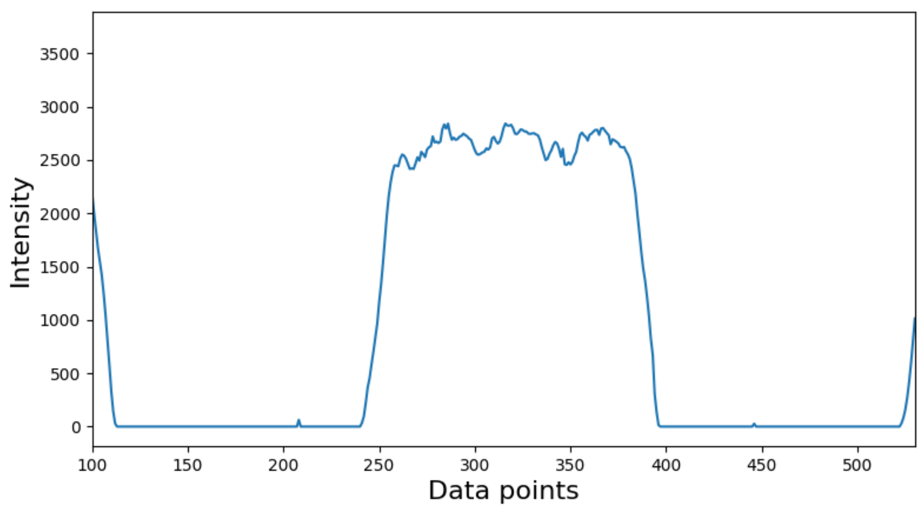


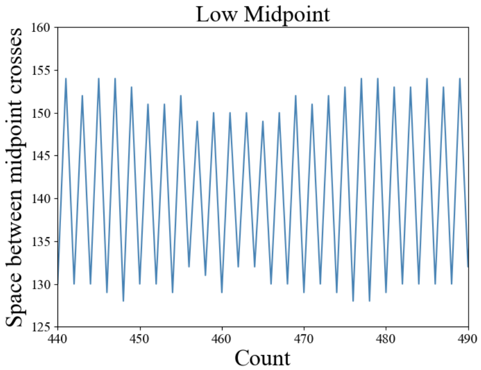
Figure 3: The raw data which the model is based off of. The slope between the lows and highs arises from the time it takes the laser to travel across the edge of a tooth.

5 The Model

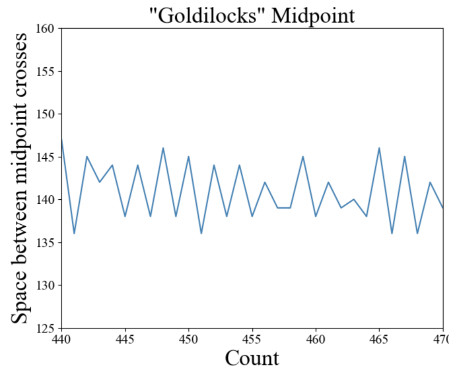
The first step in determining a varying midpoint is constructing a simplified model of the raw data, which relies on the assumptions listed below. The visual representation of this model may be seen in Figure 5, alongside the numeric definition of the variables used.

1. The averaged intensity values for the bottom and top (a_nB/a_nT) are equivalent to the true intensity values (L_nB/L_nT)

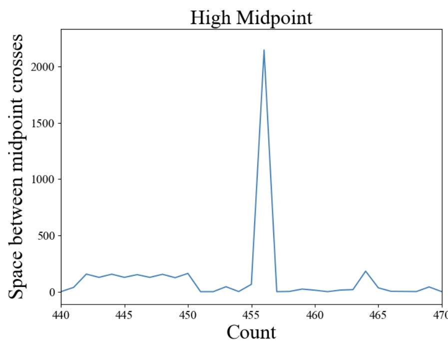
2. The upwards slope fraction, which refers to the fraction of the wave period which is spent



a) A low midpoint resulting in oscillatory behavior.



b) A "Goldilocks" (ideal) midpoint.



c) A high midpoint leading to missed readings and false threshold crossings.

Figure 4: The effects of different midpoints.

going from low to high, is a constant (S_{up}).

3. The downwards slope fraction, which refers to the fraction of the wave period which is spent going from high to low, is a constant (S_{down}).

4. The constants from (2) and (3) may be empirically determined from raw data. In our case, S_{up} = 0.06 and S_{down} = -0.06

5. The slope throughout both rise and fall is consistent throughout.

.

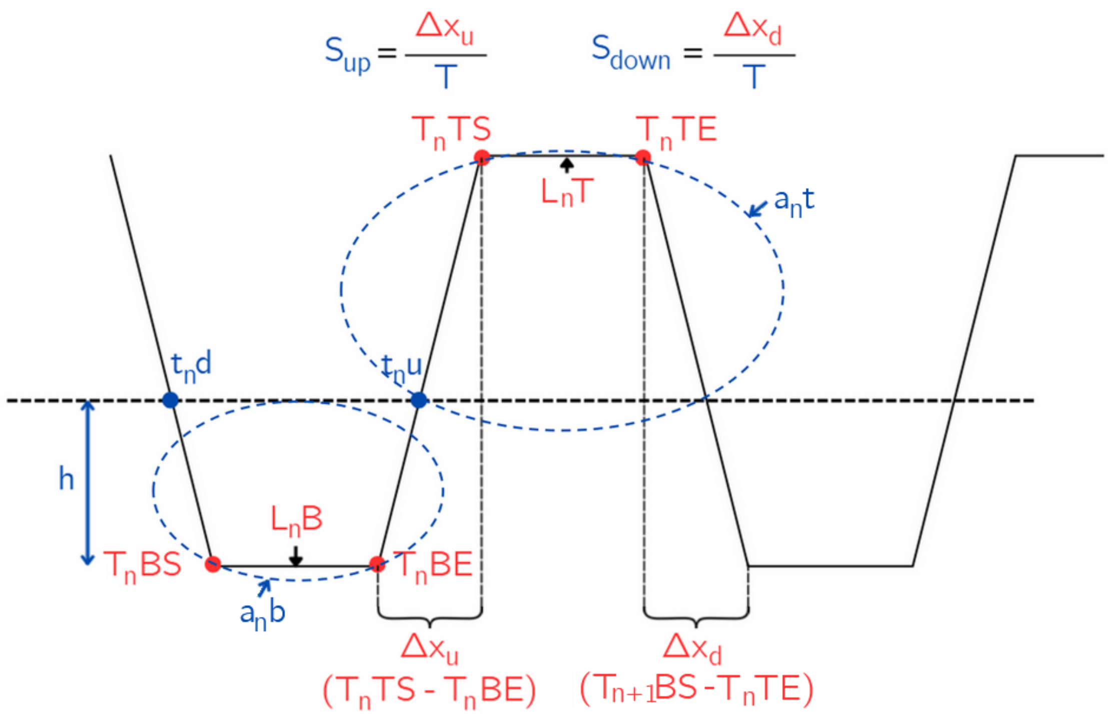


Figure 5: A visual representation of the variables used in the model. Variables in red are unknowns, while variables in blue are known or may be determined empirically.

- $t_n u$ (x) - The point where intensity crosses the threshold on its way up; where the compression algorithm begins to calculate a running average.
- $t_n d$ (x) - The point where intensity crosses the threshold on its way down; where the compression algorithm stops calculating a running average.
- $T_n BS$ (x)– The point where intensity stops decreasing and levels out to a low value; this represents when the laser is completely off a tooth.
- $T_n BE$ (x) – The point where intensity starts to increase. This represents where the laser begins to travel over the edge of a tooth.
- $T_n TS$ (x) – The point where intensity stops increasing and levels out to a high value. This represents when the laser is completely over a tooth.
- $T_n TE$ (x) – The point where intensity begins to decrease. This represents where the laser begins to transition from tooth to gap.
- $L_n B$ (y) – The actual low intensity reading.
- $L_n T$ (y) – The actual high intensity reading.
- $a_n B$ (y) – The intensity value (obtained from the analysis script's running average) when the laser is over a gap (non-reflective).
- $a_n T$ (y) – The intensity value (obtained from the analysis script's running average) when the laser is over a reflective tooth.
- h (y) – The previously selected midpoint for the analysis script.

- S_{up} (constant) – The fraction of the period the line formed by the points (T_nBE, L_nB) and (T_nTS, L_nT) occupies.

- S_{down} (constant) – The fraction of the period the line formed by the points (T_nTE, L_nT) and ($T_{n+1}BS$, $L_{n+1}b$) occupies.

6 The Algorithm

From this model, the following systems of equations arise:

a) $T_nBS = t_nd + \frac{S_{down}(h-L_nB)(t_nu-t_{n-1}u)}{L_{n-1}T-L_nB}$

b) $T_nBE = t_nu - \frac{S_{up}(h-L_nB)(t_{n+1}d-t_nd)}{L_nT-L_nB}$

c) $L_nB = \frac{h(T_nBS-T_nBE-t_nd+t_nu)+2a_nB(t_nd-t_nu)}{T_nBS-T_nBE+t_nd-t_nu}$

d) $T_nTS = t_nu + \frac{S_{up}(L_nT-h)(t_{n+1}d-t_nd)}{L_nT-L_nB}$

e) $T_nTE = t_{n+1}d - \frac{S_{down}(h-L_nT)(t_{n+1}u-t_nu)}{L_{n+1}B-L_nT}$

f) $L_nT = \frac{h(T_nTS-T_nTE+t_{n+1}d-t_nu)-2a_nT(t_{n+1}d-t_nu)}{T_nTS-T_nTE-t_{n+1}d+t_nu}$

(a), (b), (d), and (e) are derived through a system of equations which relates the slope period (S_{down} or S_{up}) to the equality assumed from (5). Formulas (c) and (f) are obtained by rewriting the equations which give the average intensity values, solving instead for L_nB and L_nT. Using these equations, alongside assumption (1) and the constant obtained from (4), estimates of T_nBS and T_nBF are produced. These two values, in turn, are used to compute a more refined guess at L_nB. This process then proceeds for T_nTS, T_nTF, and L_nT. Once these six variables are determined, then the first iteration of the algorithm is complete. As these values are all, in part, based on assumption (1), they are not exact representations of the data structure. But, as the first cycle of calculations produced more refined approximations of L_nB and L_nT, an additional round of computation may take place, superseding assumption (1) with the new values for L_nB and L_nT. This process is then repeated, with each iteration–hopefully–approaching the true values of T_nBS, T_nBE, T_nTS, $TnTE$, L_nB, and L_nT, which in turn give rise to better approximations of the midpoint for a particular waveform. If the values produced display convergent behavior, then there is good reason to believe the true values are being approached. On the other hand, if the values begin to diverge, then another approach must be taken.

6.1 Determination of a Dynamic Midpoint

As subsequent iterations of the algorithm were computed, convergent behavior indeed arose, potentially indicating that the true midpoint was being approached. With this, it is now possible to compare the effects of using a fixed versus dynamic midpoint. In using a low midpoint, it is effectively guaranteed that every waveform is logged. The downside, though, is that the period between the rise and fall of the waveform would vary between readings, due to the trapezoidal (imperfect) nature of the waveforms. In using a medium midpoint (the ideal), not only are all critical readings registered, but the "zigzag" behavior produced by a low midpoint simmers down. Unfortunately, due to the previously mentioned intensity decay present in most data, it is not always possible for this idealized midpoint to be determined. For small snippets of raw data though, this is not an issue. In using a high midpoint, certain readings may be completely missed, while also falsely flagging noise in the waveform as a threshold crossing, leading to either large spikes in the data or periods where the space between crossings is near zero (Figure 4c). Lastly, in using a variable midpoint, an interesting result arises. While we believed that the use of a tailored midpoint would outperform even the median midpoint (as it was intended to be computationally "perfect"), this was not the case (Figure 6). Rather, some of the oscillatory behavior representative of a low midpoint was apparent, although not as intensely (Figure 7). We speculate that this imperfect behavior could be due to an incomplete understanding of the structure of the data, lack of deeper iteration, or an incorrect value used for h.

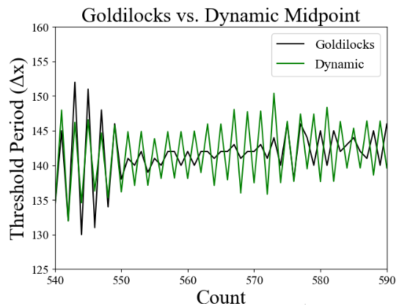


Figure 6: The dynamic midpoint not yet performing as well as the best choice midpoint.

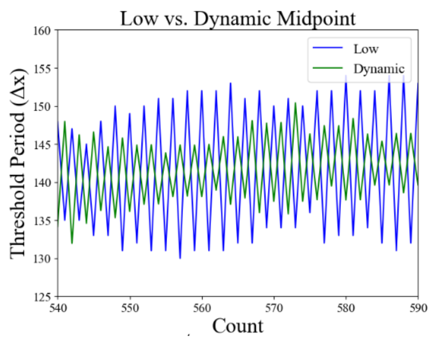


Figure 7: The dynamic midpoint still outperforms the low midpoint in terms of reduced variance between points.

7 Raw Data

During the initial development of the experimental setup, it was decided that storing all raw data would prove inefficient, as system slowdowns may occur, leading to lagged and inaccurate data. While this problem was addressed by preprocessing the data on a microcomputer, it still holds true that storing all raw data would prove useful in the case of post-analysis and troubleshooting. For this reason, the feasibility of storing all raw data was to be reevaluated, during which two criteria were investigated:

1. Is it feasible to store the quantity of data produced?
2. How fast is data produced, and will the computer be able to keep up with it?

The first criteria was promptly investigated by running the experimental setup and noting how much data was recorded in a given time frame. From this, it was determined that approximately 4.1 megabytes of data were produced per minute of operation (equivalent to 246 megabytes per hour). While certainly not insignificant, it is by no means unfeasible. For reference, a conventional one terabyte solid-state drive ($100 as of 2024) would provide enough space to store approximately 4000 hours of continuous data. The question of just how well the computer could keep up with this, though, would require further testing.

7.1 Rate

While it is certain that most modern day computers are capable of keeping up with the previously determined data acquisition rate, it was still necessary to determine if any small skips or anomalies would occur throughout the experiment.
The first step in this process was to modify the output of the sensor readings. In its default state, the data streamed would prove to be too noisy to realistically detect any encoding errors on the scale of single values amongst millions. For this reason, the microcomputer's code was altered to create a simplified stream of data. In this, a loop is initiated at index zero, which operates as follows:

1. Take a pin reading
2. Discard the pin reading
3. Send the value of the current index to the computer
4. If the index is a value between 0-98, increase it by 1. If it is 99, reset it to 0.

The reason for steps one and two is to ensure that the ensuing mock data would be produced at the same rate as the true sampled data (as the read pin command was the bottleneck in the optical system's sampling rate). In implementing this system, the resulting data took upon a typical sawtooth pattern, which would not only permit for easy visual determination of encoding errors, but when fully stored, a simple algorithm could run through the entire list of numbers and detect any skipped values. With a diagnostic dataset ready to stream to the computer, it was now time to determine which–if any–data storage mechanism would be ideal for complete data storage.
The first option is a direct-to-disk system, in which incoming data would be immediately saved to a traditional hard disk file. This has the benefit of being the most reliable, as even in the event of total power loss, most data is still retained. Unfortunately, traditional mechanical storage is typically one of the slowest components in a computer, and as a result, may not be able to keep up with the data. This weakness may be alleviated through the use of solid-state storage in place of a mechanical hard drive.

The second option is storing directly to the computer's random access memory (RAM). While this method is by far the fastest (only second to the CPU's internal cache), it proves to be significantly more expensive, as well being prone to complete data loss in case of system failure. Additionally, as more RAM is taken up by the data, less is available for the computer's operation, leading to an overall system slowdown.
The third option is a hybrid of the previous two. In this, the computer's RAM is used as a cache. After a certain buffer limit is reached, the cache is dumped to hard storage. Not only does this mechanism provide the speed inherent to RAM, but it also provides the reliability inherent to storing directly to disk. The crux of this method, though, is that data acquisition may slow down or halt during dumping periods.

7.2 Results

Once the three systems were tested, it was determined that storing directly to a solid-state drive proved to be the ideal method, as there were effectively no encoding errors present (in contrast, a typical mechanical hard drive would have occasional errors crop up). Storing to RAM proved too expensive and vulnerable to data loss, whilst the hybrid system, as predicted, experienced encoding errors during dumping periods.

8 Concluding Remarks and Future Work

The use of an iterative algorithm showed strong signs of being an effective tool not only to increase the data quality of prior datasets, but to ameliorate the issues that occur from the intensity decay present throughout most experiments. Unfortunately, while the dynamic midpoint served better than a low value, it still did not outperform an "ideal" midpoint. There could have been many reasons for this. Primarily, it was determined that uneven icing occurred on one of the lid's teeth, which was not accounted for the the model. Additionally, the value used for h was a partial assumption, which would again interfere with the algorithm (in the case that h is not known, another algorithm may be developed to test different values of h, eventually settling on the one which shows least variance between adjacent midpoint values). Lastly, it may have simply been an issue of insufficient iteration (in my testing, I went as far as 11 cycles).

In regard to pure data storage, it was determined that storing all raw data (to a solid-state drive, ideally) is indeed feasible and may serve to either replace the analysis algorithm or supplement it as a backup for the sake of troubleshooting.

9 Acknowledgments

I would like to thank Professor Rena Zieve not only for all her efforts in guiding me, but for making the REU program what it is today. Additionally, I would like to thank Professor Zieve's graduate students, Haoyang Zhou and Michael Spiegel, for all the time and help they provided me with throughout the summer. Lastly, I would like to thank the National Science Foundation, the University of California, Davis, and Dr. Nicholas Curro for making this entire experience possible.

10 References

[1] C. A. van Eysden, "Introduction," in Superfluid spin up and pulsar glitch recovery, Melbourne: University of Melbourne, 2011, pp. 21–21

[2] A. J. Leggett, "Superfluidity," Encyclopædia Britannica, https://www.britannica.com/science/superfluidity (accessed Jan. 2, 2025). itannica, https://www.britannica.com/science/superfluidity (accessed Jan. 2, 2025).

[3] J. Liu, "Superfluids Are Not Fluids," Superfluidity: Particles without attractions, https://cs.stanford.edu/people/zjl/superfluidity.html (accessed Jan. 2, 2025).

[4] J. S. is the casedze and S. J. Tsakadze, "Properties of slowly rotating helium II and the superfluidity of pulsars," Journal of Low Temperature Physics, vol. 39, no. 5–6, pp. 649–688, Jun. 1980. doi:10.1007/bf00114899

[5] L. Warszawski, A. Melatos, and N. G. Berloff, "Unpinning triggers for superfluid vortex avalanches," Physical Review B, vol. 85, no. 10, Mar. 2012. doi:10.1103/physrevb.85.104503

[6] V. L. Ginzburg, "Superfluidity and superconductivity in the universe," Journal of Statistical Physics, vol. 1, no. 1, pp. 3–24, Mar. 1969. doi:10.1007/bf01007238