

When is a noisy signal oscillating and when is it just noisy?

Ritesh Balayan

August 19, 2023



THE UNIVERSITY
of EDINBURGH

MSc in Theoretical Physics

Supervised by :

Dr. Peter Swain

Abstract

This thesis tackles the challenge of detecting periodic signals in complex datasets. Traditional methods, including the widely used Fast Fourier Transform (FFT), often fall short with sparse or noisy data. This thesis emphasises the shift from traditional Fourier analysis to Bayesian techniques and focusses on the Gregory-Lorendo methods and the Gaussian process. These techniques not only enhance the detection of rhythmic patterns but also improve FFT's efficiency. In addition, a versatile software package is presented, suitable for a range of research disciplines. Research culminates in a comparative analysis of different methods and recommendations for specific use cases.

Declaration

I hereby declare that this dissertation was composed entirely by me. Chapter 2 provides a literature review of the topics and is not my work; I have some interpretations of results and plots which are mine, but apart from that it is the work already in literature. Chapter 3 describes the work that was done entirely by me. The results of this chapter have been improved/modified or thoroughly studied to reach new conclusions about the methods. Chapter 4 is about methodologies and computational work; this is also my own work. For the Gregory model, every piece of code was written from scratch in Python by myself. For the Gaussian process, I am using the skikit-learn prebuilt library and adding my modifications on top of it. Chapter 5 results were achieved by myself and all the results are computed entirely by the programme which I built from scratch (apart from basic libraries like Numpy, Pandas, Scipy, and scikit learn). Note that for the FFT spectrum and Gaussian process optimisation, I am using a pre-built software package.

Personal Statement

I began working on this project in early 2023, when Dr. Peter Swain explained it to me. I was particularly drawn to the open nature of the project. I am also grateful to Dr. Frank Herzog for informing me about this. I was excited from the first presentation on 15 February, for which I had only 10 days to prepare. Those 10 days were the most productive of my Master programme's, as I learnt about Gaussian processes from the book 'Rasmussen' recommended by my supervisor, which I still possess. By the end of March, my understanding of Gaussian processes had become both theoretical and computational. I started working full-time on my thesis in early June, but before that I was in constant communication with my supervisor every week. During June, I studied the Gregory Lorendo method and replicated the results from their original paper. For a month, I worked on modifying the algorithm to make it faster and more efficient. I also discussed with Arin (now Dr Arin, a PhD student in our lab) about trying to find oscillations in his experimental observations of budding yeast. I had many failed or inconclusive results on my hard drive from my attempts to find oscillations reliably in that data set. I am thankful to my supervisor for guiding me every day to get out of the rat trap in which I would otherwise have spent a lot of time. Later, I succeeded in significantly increasing the algorithm calculation time and then wrapping up the complex flow of the algorithm to make it user-friendly. This was the most pleasant part of the project. I apologise for not writing good comments and making many spelling mistakes in internal discussions with my supervisors, who were kind enough to correct me on these silly mistakes again and again. The authors also thank the members of the Swain lab who gave me valuable feedback throughout the year on their work. The most productive time for my thesis was actually the last two weeks, when I had enough time to execute my beloved Gaussian process and make my number of algorithm analyses to three, which turned out to be the most effective method I developed in this project.

Contents

1	Introduction and Problem	1
2	Literature Review: From Fourier to Bayesian	3
2.1	Fourier Transform: Fundamentals and Applications	3
2.2	An Overview of the Bayesian Framework	4
2.2.1	Example of Bayesian Inference: A linear model	6
2.3	Gregory-Loredo Method	9
2.3.1	Model	10
2.3.2	Priors	11
2.3.3	Likelihood	11
2.3.4	Calculation	12
2.3.5	Estimation of frequency	15
2.3.6	Visualising on Gregory-Lorendo model	16
2.4	Gaussian Processes: Concepts and Applications	18
2.4.1	Switching from weight space view to functional space view	18
2.4.2	Gaussian process for oscillation detection	19
3	Theoretical Enhancements to Existing Methods	21
3.1	Refining the Gregory-Loredo Method	21
3.1.1	Log-Likelihood vs Likelihood	22
3.1.2	Behaviour of GL likelihood with phase	23
3.1.3	Discontinuous Likelihood of GL	23
3.2	Advancements in Gaussian Process Techniques	24
3.2.1	Relation to GL	25
4	Methodology and Computational Implementation	27
4.1	Initial Challenges	28

4.2	Class Structure	28
4.3	Generator Detector Interplay	29
4.4	Crash Course in Algorithm	29
4.5	Resources	30
5	Results	31
5.1	With Sinosodial Signal	31
5.2	With saw-tooth Signal	32
5.3	With Impulsive Signal	32
5.4	Noise Analysis	33
5.4.1	Pure Noise	33
5.4.2	Signal with embedded noise	33
6	Discussion	35
6.1	Strong peak at half frequency	35
6.2	Better Resolution and range than FFT	35
6.3	Combination of frequency's	36
6.4	Possible improvements on algorithms	36
6.5	Possible improvements on analysis	36
6.6	Final Words	37

List of Figures

2.1	Linear model with Gaussian noise	7
2.2	Normalised likelihood of linear model	8
2.3	Inferred linear model obtained from Bayes theorem	9
2.4	A sample from GL model with arbitrary values of parameters	10
2.5	GL model sample with data points overlaid after every period	11
2.6	Flowchart of Calculation in GL Method	15
2.7	GL model sample from Posterior	16
2.9	Polar plot of GL model	17
2.10	Gaussian process priors	20
3.1	GL method calculation flow chart from original paper	21
3.2	GL method calculation flow chart after our modifications	22
3.3	GL log-likelihood vs ϕ	23
3.4	Calculation flow chart of GP Method	24
3.5	Gaussian process hyperparameter optimisation	25
4.1	Replication of Gregory 1999 paper	27
4.2	Class structure of Oscillation Detector	28
5.1	Signal used for analysis	31
5.2	Results plotted in $\log_2 \frac{true}{Inferred}$ scale of frequency	32
5.3	Algorithm result with pure noise	33
5.4	Algorithm behaviour with increase in noise	33
6.2	Resolution power of inference algorithms	36

Chapter 1

Introduction and Problem

Problem

Periodic patterns, from star vibrations to regular behaviours of atoms and molecules, are found everywhere in the physical world. These rhythms help us to understand the systems from which they come from. However, detecting and analysing these patterns is not always straightforward. There are challenges. Sometimes, patterns are buried in noise or are very complex. Therefore, better methods are needed to better understand these rhythms with more precision.

Literature review

The Fourier Transform algorithm is a highly efficient method that is widely used in many applications, making it one of the most popular numerical methods in modern research. However, it has been shown to be ineffective in sparse and noisy time series with few data points. Therefore, more sophisticated tools are needed. This thesis will focus on Bayesian methods. Firstly, a basic introduction to the Bayesian framework will be provided. We will then analyse Gregory Lorendo's methods, which were developed in a series of papers from 1992[4] to 1999[3] in the field of astrophysics. Following this, a brief overview of the Gaussian process will be given, highlighting its status as a reliable tool, especially useful in cases with limited data[7] by using it as an oscillation detection tool.

Improvements

We conducted an examination of the (Gregory Lorendo) GL and (Gaussian process) GP techniques to identify their unique characteristics and assess their effectiveness and usability. We then modified them to get efficient results and then developed a reliable method to compare the results of Bayesian methods with the Fourier power spectrum. Additionally, under certain conditions, we discovered an alternative approach to enhance the Fourier transform based solely on the principles of probability theory.

Goals

Our main goal in this thesis was to create a user-friendly tool that researchers could use in the laboratory. We were successful in creating a universally applicable programme with an interface that is flexible enough to be applied to research in various disciplines such as physics, engineering, chemistry, astronomy, and biology. Additionally, we have provided a comprehensive user guide to ensure that it is easy to use in any research setting.

Results

We compare the efficiency of (fast Fourier transform) FFT, (Gregory Lorendo method) GL, and (Gaussian process) GP in various generated signals to provide a clear overview of the advantages and disadvantages of these techniques. Additionally, we provide essential considerations and warnings for their implementation.

Conclusion

In summary, this thesis presents and evaluates the results compared to existing methods and suggests potential areas for improvement in signal detection when there is limited or no knowledge about the existence or type of oscillation.

Chapter 2

Literature Review: From Fourier to Bayesian

2.1 Fourier Transform: Fundamentals and Applications

Named after Jean-Baptiste Joseph Fourier, Fourier analysis is a mathematical technique that breaks down functions into their oscillatory components, such as sine and cosine waves. At the heart of this analysis is the notion of the *Fourier transform*. This transform enables a function, which is usually time-dependent, to be expressed in terms of its frequency components.

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i f t} dt \quad (2.1)$$

The DFT is a linear transformation that can be used to decompose a signal into its constituent frequencies. The Discrete Fourier Transform (DFT) is a key tool when working with digital signals or discrete data samples.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i(2\pi/N)kn} \quad (2.2)$$

The Fast Fourier Transform (FFT) is an efficient algorithm for computing the discrete Fourier Transform (DFT) which can be computationally intensive for large data sets. This technique is essential for time series analysis, as it allows for the detection of oscillatory patterns and periodicities in the data. When converting a time series into its frequency components, one can identify the underlying cycles and oscillations. The Fourier methods are advantageous as they can represent complex signals in terms of simple oscillatory components, providing insights into the dominant frequencies present in a data set. However, the resolution and range of detectable frequencies depend on the length of the data and its sampling rate.

When analysing time-series data using the Fast Fourier Transform (FFT), two important

parameters are the resolution and the maximum detectable frequency. Resolution, or frequency resolution, is the smallest distinguishable frequency difference in the spectrum and is related to the total duration of the time series (T) and f_s is the sampling interval.. The formula to calculate the resolution is:

$$\text{Resolution} = \frac{f_s}{T} \quad (2.3)$$

This means that a longer time series provides a finer frequency resolution. The minimum detectable frequency (apart from DC) is the smallest nonzero frequency that can be identified and is equal to the resolution, which is the inverse of the total duration:

$$\text{Minimum Frequency} = \text{Resolution} = \frac{f_s}{T} \quad (2.4)$$

However, to fully understand the frequency range, one should also consider the Nyquist frequency, defined as half the sampling rate:

$$f_{\text{Nyquist}} = \frac{f_s}{2} \quad (2.5)$$

The Nyquist frequency is the highest frequency that can be sampled in a Fast Fourier Transform (FFT). The power spectrum of the Fourier transform at these frequencies is plotted by taking the square of the absolute value of the complex Fourier transform at these discrete points, resulting in the usual Fourier power spectrum.

Problem with Fast Fourier Transform (FFT)

Consider a scenario in which 200 data points are extracted over a time span of 100 seconds, yielding a sampling interval of 0.5 seconds and sampling frequency of 2 Hz. With this configuration, the lowest discernible non-zero frequency is 0.02 Hz. Subsequent observable frequencies increase in increments of 0.02 Hz, culminating in an upper limit of 1 Hz. This imposes a constraint where frequencies exceeding 1 Hz remain undetectable, and the resolution between identifiable frequencies is restricted to no finer than 0.01 Hz.

Additionally, the Fast Fourier Transform (FFT) algorithm presents complications when faced with incomplete datasets. While certain techniques, such as zero padding, offer a workaround, they are not universally applicable to all scientific enquiries. The inherent limitations of FFT, particularly in contexts with sparse observations, often render its utility inadequate for a comprehensive analysis. To address these challenges, Bayesian methodologies present a robust alternative. Consequently, this study will pivot its emphasis towards the exploration and application of Bayesian techniques.

2.2 An Overview of the Bayesian Framework

The Bayesian approach is a method of combining data with prior beliefs to generate new hypotheses. The Bayes Theorem is derived from the fundamental equation of the

multiplication of probabilities. In contrast, the frequentist view emphasises the frequency or proportion of data occurrences without incorporating prior beliefs.

$$\mathcal{P}(\mathcal{D})\mathcal{P}(\mathcal{H}|\mathcal{D}) = \mathcal{P}(\mathcal{D}|\mathcal{H})\mathcal{P}(\mathcal{H}) \quad (2.6)$$

By rearranging these, we get the Bayes theorem.

$$\mathcal{P}(\mathcal{H}|\mathcal{D}) = \frac{\mathcal{P}(\mathcal{D}|\mathcal{H})\mathcal{P}(\mathcal{H})}{\mathcal{P}(\mathcal{D})} \quad (2.7)$$

Different terms are often referred to in literature, such as:

$$Posterior = \frac{Likelihood \times Prior}{Marginal Likelihood} \quad (2.8)$$

The sum of all the Priors must be equal to one, and the likelihood is usually calculated to determine how well the hypothesis is compatible with the data. This is usually done by calculating the residual of curve fitting, which is interpreted as the probability of the hypothesis given the data. The posterior, or the probability of a hypothesis given the data, is then used to quantify the updated belief in the hypothesis after the evidence has been taken into account.

The bottom part of the right side serves as a normalisation factor and can be determined by the marginalisation principle of probability.

$$\mathcal{P}(\mathcal{D}) = \sum_h \mathcal{P}(\mathcal{D}, \mathcal{H} = h) = \sum_h \mathcal{P}(\mathcal{D}|\mathcal{H} = h)\mathcal{P}(\langle) \quad (2.9)$$

All hypotheses \mathcal{H} have discrete values in the specific hypotheses h , and for continuous variables, the summation is substituted with an integral.

$$\mathcal{P}(\mathcal{D}) = \int dh \mathcal{P}(\mathcal{D}, \mathbf{H}) = \int dh \mathcal{P}(\mathcal{D}|\mathbf{H})\mathcal{P}(\langle) \quad (2.10)$$

Throughout the text, the following notation will be used $\mathcal{P}(D|a, b, c, I)$ where a , b and c are parameters of the model and character "I" is reserved for not explicitly mentioned parameters.

Marginalization

Integrating out a parameter in a probability formula involves summing over all possible values of that parameter. This process is known as marginalisation and is expressed by a generalised formula:

$$\mathcal{P}(D|M, I) = \int da \int db \int dc \int \dots \mathcal{P}(a|M, I)\mathcal{P}(b|M, I)\mathcal{P}(c|M, I)\mathcal{P}(D|a, b, c, \dots M, I) \quad (2.11)$$

Where M is the Model with Parameters a,b,c...

Nuisance parameters

The Bayesian framework enables us to infer one parameter without having to explicitly determine the other parameters, as they can be marginalised. For example, in Equation (2.11), the probability distribution of D can be determined by taking the integration over of a, b and c. This may seem straightforward, but it is a feature that is generally absent in frequentist statistics. This simplifies the calculations, and the integrated variables are called nuisance parameters.

Bayesian Inference

Bayesian inference employs Bayes' theorem to update the probability estimate for a hypothesis as additional evidence becomes available. Instead of producing a single best-fit estimate for model parameters, Bayesian inference yields a posterior probability distribution over the parameter space.

Consider the task of fitting a Gaussian (or normal) curve to a given set of data. The Bayesian approach to this problem can be summarised as follows.

- Specify a Gaussian model characterised by parameters, typically the mean (μ) and the standard deviation (σ).
- Assign prior probability distributions to these parameters based on existing beliefs or information. This can be done using informative priors (based on previous knowledge) or non-informative priors if no prior knowledge exists.
- Construct a likelihood function that expresses the probability of observing the given data as a function of the model parameters.
- Use Bayes' theorem to compute the posterior probability distribution of the parameters given the data. This involves combining the likelihood with the priors.

The result is not a single set of best-fit parameters, but a distribution that captures our uncertainty about the possible values of these parameters in light of the data.

2.2.1 Example of Bayesian Inference: A linear model

In this example, we will consider a simple straight line with slope and intercept.

$$y = mx + c \quad (2.12)$$

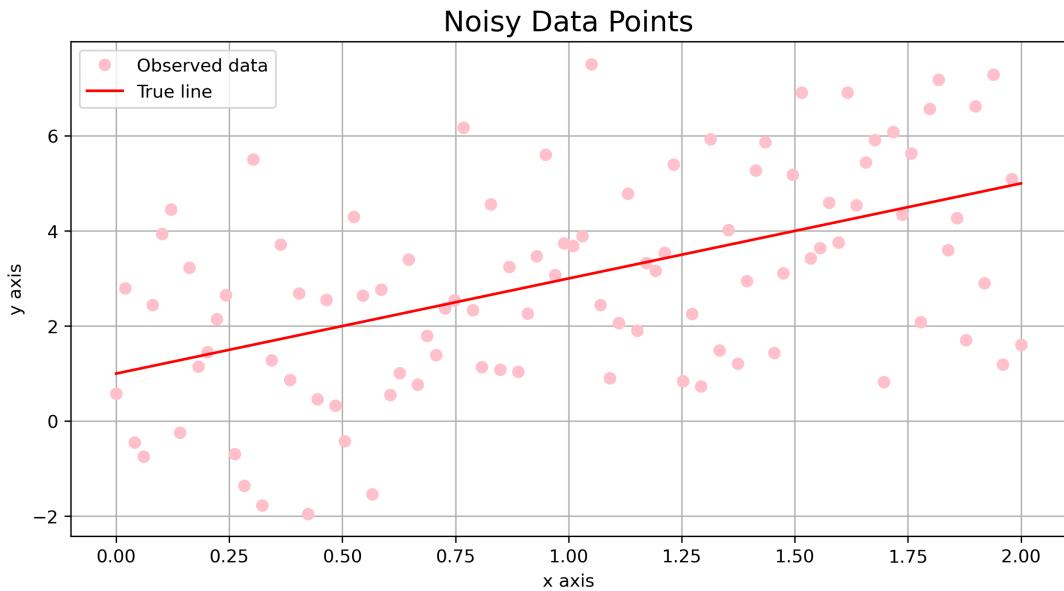


Figure 2.1: Linear model with Gaussian noise

We begin our Bayesian inference by assuming that all values of m and c are equally probable before any data is observed. This is known as a flat prior.

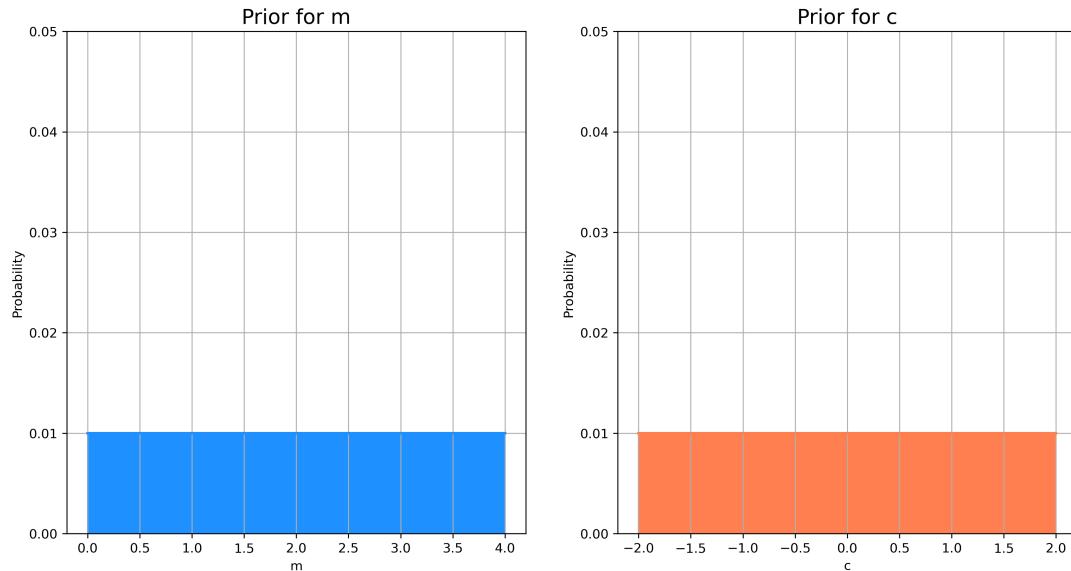


Figure : Flat priors for slope and intercept of linear model

Next, we must select a likelihood function. This function explains how probable the data are for different values of the parameters. In this case, we can use the Gaussian likelihood function, since we know that the data are distributed around the true line with Gaussian noise. The likelihood function for a set of data points (x_i, y_i) is expressed as follows:

$$\mathcal{L}(m, c) = \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - (mx_i + c))^2}{2\sigma^2}\right) \quad (2.13)$$

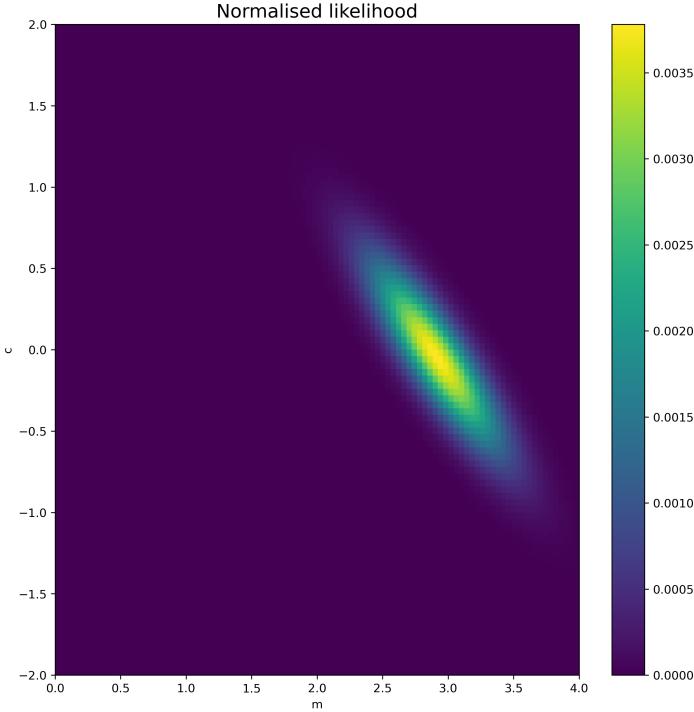


Figure 2.2: Normalised likelihood in color gradient, slope on x axis and intercept on y axis

We can use Bayes' theorem to adjust our initial beliefs in the light of the data.

We can calculate the posterior distribution by multiplying the likelihood by the priors and then normalising. The marginal posterior for each parameter is given by:

$$\mathcal{P}(m|data) = \int \mathcal{P}(m, c|data) dc \quad (2.14)$$

$$\mathcal{P}(c|data) = \int \mathcal{P}(m, c|data) dm \quad (2.15)$$

Finally, we can sample from the posterior distribution to see the range of plausible lines that are consistent with the data. We will take 1000 random samples from the credibly credible posterior region 68.3%.

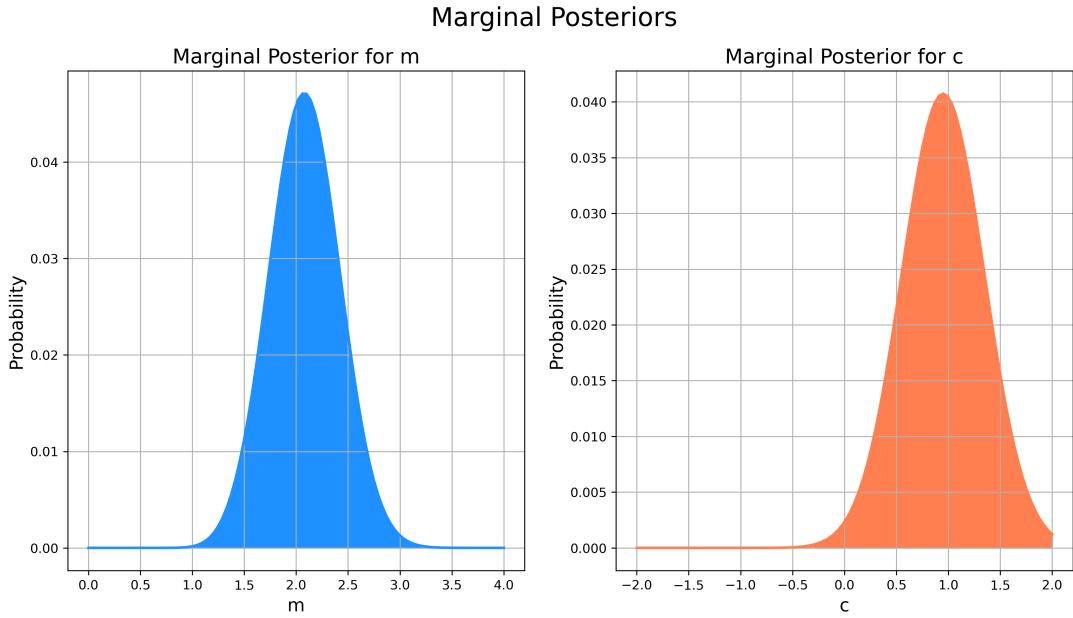


Figure : Posterior distribution of slope and intercept after applying the Bayes theorem

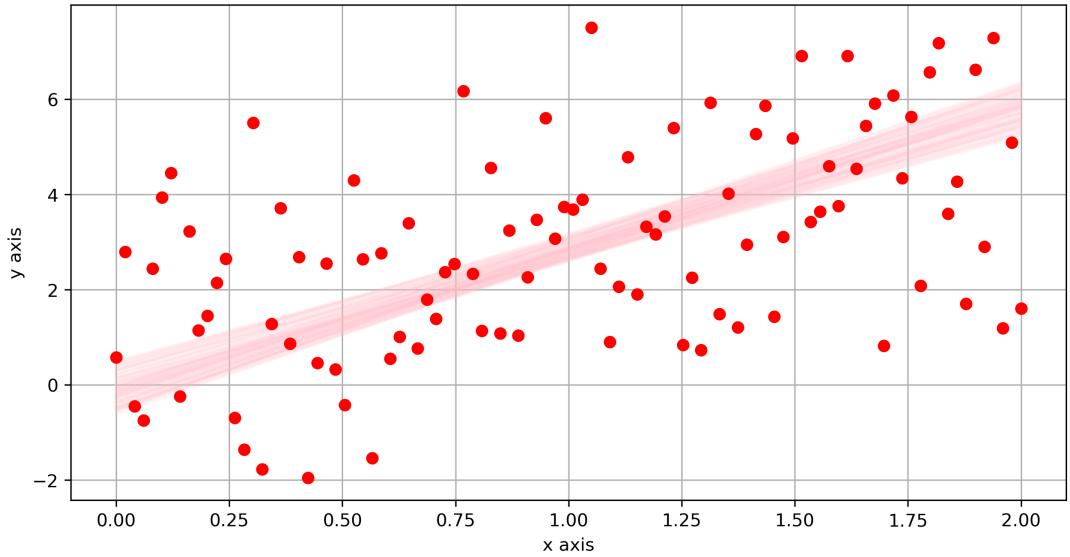


Figure 2.3: Inferred Model from 68.3 % credible region of posterior

2.3 Gregory-Loredo Method

The Gregory-Loredo technique is a Bayesian approach to detect periodic components in time series. It was initially designed for use in astrophysics. The GL method is applicable when the noise sampling distribution is independent Gaussian and the data is non-uniformly sampled. As per the general rule, the GL method consists of four steps, which were discussed in the preceding section.

2.3.1 Model

The model in the GL method is a step wise model. Defined by followings parameters:

- m : Number of bins in model
- w : Frequency of periodic model
- ϕ : Phase of period model

GL model is defined as:

$$j(t) = \text{int} \left[1 + \frac{m((wt + \phi) \bmod 2\pi)}{2\pi} \right] \quad (2.16)$$

Another important parameter to introduce is:

- \vec{r} : Vector of m dimensions
- r_j : j_{th} component of vector \vec{r}

with j varying from 1 to m . r_j demonstrating the height of the j_{th} bin in the periodic model. Giving the model $m + 2$ parameters to work with.

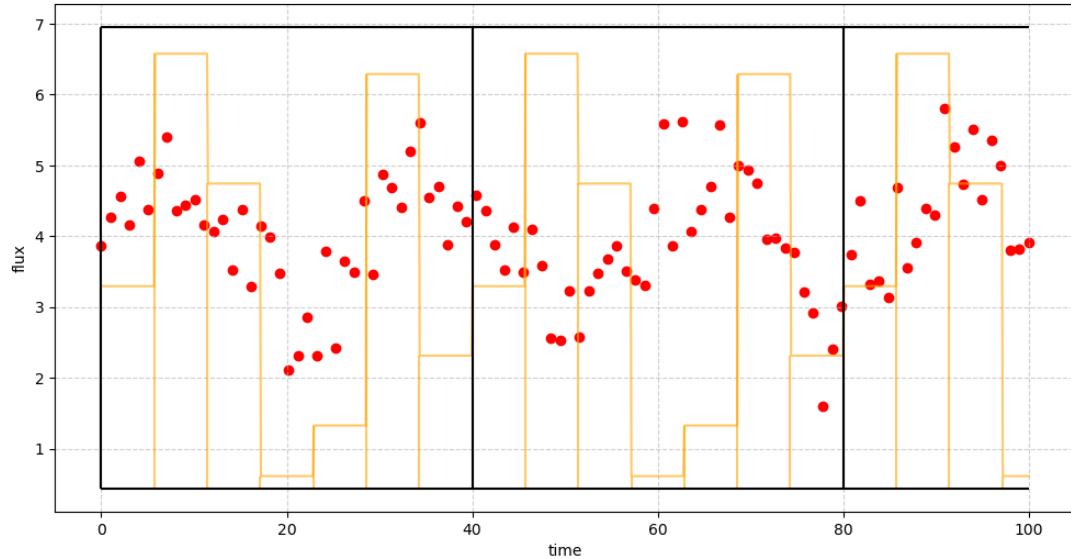


Figure 2.4: A sample from GL model with arbitrary values of parameters

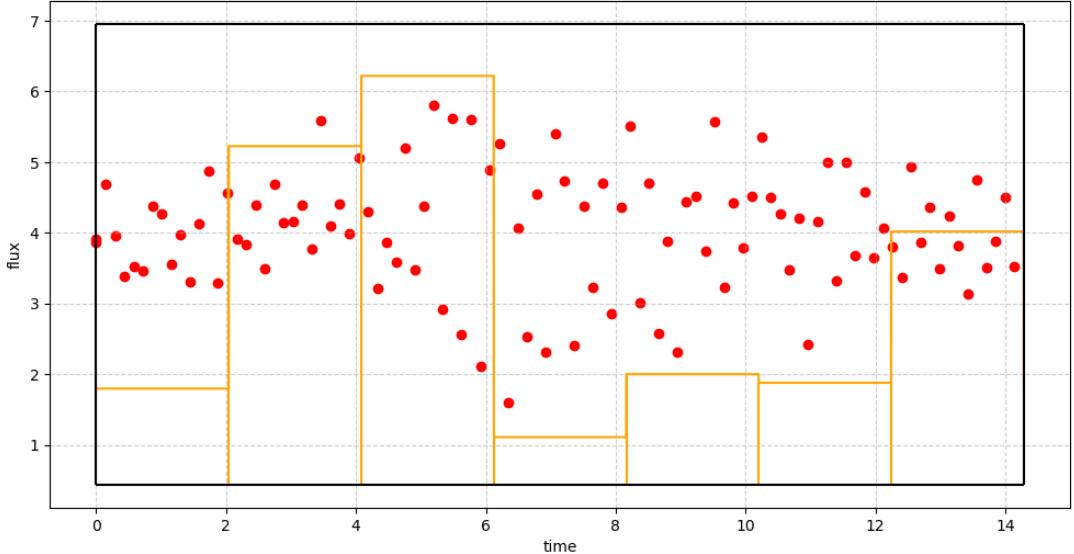


Figure 2.5: GL model sample with data points overlaid after the every period

when $m = 1$ means that there is no periodic motion and that the data are independent noise around the constant. The higher m means the periodic model.

2.3.2 Priors

Once a model with m steps has been selected, there are $m + 2$ parameters that must be assigned prior values. The original Gregory-Lorendo approach utilises the following priors.

$$\mathcal{P}(\phi|m, I) = \frac{1}{2\pi} \quad (2.17)$$

$$\mathcal{P}(w|m, I) = \frac{1}{w \ln \frac{w_{hi}}{w_{lo}}} \quad (2.18)$$

$$\mathcal{P}(r_j|m, I) = \frac{1}{\Delta r} = \frac{1}{r_{max} - r_{min}} \quad (2.19)$$

These are Jeffrey's Priors, They are employed in Bayesian analysis due to their transformation-invariant property, offering a neutral perspective irrespective of the parametrisation of the problem[5].

2.3.3 Likelihood

The data points d_i are represented by i ranging from 1 to N (the total number of data points). The difference between a particular data point d_i and the height of the bin r_j at which it falls is assumed to be a Gaussian distribution.

Given the frequency w , the phase ϕ , the number of bins m and \vec{r} . We can find the Likelihood as:

$$p(D|w, \phi, \vec{r}, m, I) = \prod_{j=1}^m \left[(2\pi)^{-\frac{n_j}{2}} \left(\prod_{i=1}^{n_j} (s_i)^{-1} \right) \exp\left(\frac{-\alpha}{2}\right) \right] \quad (2.20)$$

where n_j is number of samples in j_{th} bin. and s_i is expected noise in i_{th} data point.

$$\alpha = \sum_{i=1}^{n_j} \frac{(d_i - r_j)^2}{s_i^2} \quad (2.21)$$

2.3.4 Calculation

Although we get all components of the Bayes theorem to infer the best-fit parameters, calculations are not practical at this point as we have to calculate convoluted integrals of variables $m + 2$ (2.11). We will integrate across all $m+2$ parameters and attempt to use analytical integrals whenever feasible.

Isolating \vec{r} term

To make the calculation more straightforward, first isolate r_j from the likelihood. Recall that the subscript j is for the bin j_{th} .

$$\alpha = \sum_{i=i}^{n_j} \frac{d_i^2}{s_i^2} - 2r_j \sum_{i=i}^{n_j} \frac{d_i^2}{s_i^2} + r_j^2 \sum_{i=i}^{n_j} \frac{1}{s_i^2}$$

Let's call

$$\sum_{i=1}^{n_j} \frac{1}{s_i^2} = W_j \quad : \quad \frac{\sum_{i=1}^{n_j} \frac{d_i^2}{s_i^2}}{W_j} = d_{W_j} \quad : \quad \frac{\sum_{i=1}^{n_j} \frac{d_i^2}{s_i^2}}{W_j} = d_{W_j}^2$$

α can be simplified to

$$\alpha = W_j(d_{W_j}^2 - 2r_j d_{W_j} + r_j^2)$$

To separate out r_j we can add and subtract $d_{W_j}^2$ giving

$$\alpha = W_j \left[(r_j^2 - 2r_j d_{W_j} + d_{W_j}^2) + d_{W_j}^2 - d_{W_j}^2 \right] \quad (2.22)$$

Where we can separate out $\chi^2 = d_{W_j}^2 - d_{W_j}^2$ and complete the square of the extra term.

$$\alpha = W_j \left[(r_j - d_{W_j})^2 + \chi^2 \right] \quad (2.23)$$

Here we have separated out r_j as a single variable to exponential where every other part can be treated as constant and we can perform analytical integral. In short, our likelihood is in the form of a standard formula for a Gaussian integral.

$$\mathcal{L} \approx e^{-r^2} \quad (2.24)$$

Marginalizing over \vec{r}

Using generalised tricks to marginalise:

$$\mathcal{P}(D|M_m, I) = \int dw \int d\phi \int d\vec{r} \mathcal{P}(\phi|M_m, I) \mathcal{P}(w|M_m, I) \mathcal{P}(\vec{r}|M_m, I) \mathcal{P}(D|w, \phi, \vec{r}, M_m, I)$$

Marginalizing \vec{r} first.

$$\mathcal{P}(D|w, \phi, M_m, I) = \int d\vec{r} \mathcal{P}(\vec{r}|M_m, I) \mathcal{P}(D|\vec{r}, w, \phi, M_m, I)$$

The first term on the right-hand side is Prior and the second term we have already simplified.

$$\mathcal{P}(D|w, \phi, M_m, I) = \prod_{j=1}^m \left[(2\pi)^{-\frac{n_j}{2}} \left(\prod_{i=1}^{n_j} (s_i)^{-1} \right) \exp\left(-\frac{\chi^2 W_j}{2}\right) R \right]$$

where R is

$$R = \int_{r_{min}}^{r_{max}} dr_j \mathcal{P}(r_j|M_m, I) \exp\left(-\frac{W_j(r_j - d_{W_j})^2}{2}\right)$$

see R is inside \prod that's why \vec{r} turns to r_j . Now expanding along the product.

$$= (2\pi)^{-N/2} (\Delta r)^{-m} \left(\prod_{i=1}^N (s_i)^{-1} \right) \exp\left(-\sum_{j=1}^m \frac{\chi^2 W_j}{2}\right) \times \prod_{j=1}^m \left[\int_{r_{min}}^{r_{max}} dr_j \exp\left(-\frac{W_j(r_j - d_{W_j})^2}{2}\right) \right]$$

Here terms that are not depending on j turn from the product (zero to n_j) to (zero to N). The prior of r used is $1/\Delta r$ and the product turns into the sum in exponential.

The integrand is sufficiently isolated, and we can perform integral analytically.

*Complementary error function

This integral behaves well in the following limits and depends only on the limit of the integral.

$$\text{erfc}(y) = \frac{2}{\sqrt{\pi}} \int_y^\infty du \exp(-u^2) \quad (2.25)$$

In above equation If we take $u^2 = W_j(r_j - d_{W_j})^2/2$ we can Simplify:

$$\int_{r_{min}}^{r_{max}} dr_j \exp\left(-\frac{W_j(r_j - d_{W_j})^2}{2}\right) = \sqrt{\frac{\pi}{w}} W_j^{-\frac{1}{2}} [\text{erfc}(y_{jmin}) - \text{erfc}(y_{jmax})] \quad (2.26)$$

where

$$y_{jmin} = \sqrt{\frac{W_j}{2}}(r_{min} - d_{w_j}) ; \quad y_{jmax} = \sqrt{\frac{W_j}{2}}(r_{max} - d_{w_j})$$

After performing the integral we are left with

$$\begin{aligned} \mathcal{P}(D|w, \phi, M_m, I) &= \\ (2\pi)^{-N/2}(\Delta r)^m \left(\prod_{i=1}^N (s_i)^{-1} \right) \exp \left(-\sum_{j=1}^m \frac{\chi^2 W_j}{2} \right) (\pi/2)^{m/2} \times \prod_{j=1}^m &(W_j^{1/2} [\text{erfc}(y_{jmin}) - \text{erfc}(y_{jmax})]) \end{aligned}$$

An advantage of the GL method is its ability to analytically marginalise r .

Marginalisation over m and ϕ

Again using the marginalisation trick

$$\mathcal{P}(D|M_m, I) = \int dw \int d\phi \mathcal{P}(D|w, \phi, M_m, I) \mathcal{P}(w|M_m, I) \mathcal{P}(\phi|M_m, I)$$

Substituting the priors specified earlier in the equation we get:

$$\begin{aligned} \mathcal{P}(D|M_m, I) &= \frac{(2\pi)^{-N/2}(\Delta r)^m \left(\prod_{i=1}^N (s_i)^{-1} \right) (\pi/2)^{m/2}}{2\pi \ln \frac{w_{hi}}{w_{lo}}} \\ &\times \int \frac{dw}{w} \int d\phi \exp \left(-\sum_{j=1}^m \frac{\chi^2 W_j}{2} \right) \\ &\times \prod_{j=1}^m (W_j^{1/2} [\text{erfc}(y_{jmin}) - \text{erfc}(y_{jmax})]) \end{aligned} \quad (2.27)$$

We will keep track of this integral as it is key in GL methods and will be used multiple times.

A Note on Calculation of $\mathcal{P}(D|M_m, I)$ Probability of Data given Model

A word about formula (2.27). It is called the probability of data given model. It is quite a powerful formula, as it can quantify the likelihood of the model itself. Note that it gives the likelihood of a model with m bins and that there can be different models with different numbers of bins. This formula can quantify the relative probability that which model is more likely to fit the given data/evidence.

The integral in this formula is quite challenging because wherever there is a dependency of j , it means the dependency of w and ϕ and m . revising model (2.16). So, we have done m integrals analytically, but the remaining integrals are challenging and computationally

expensive, and one of the limitations of GL methods. Conceptually, we can find all the posterior distribution of parameters once we do these integrals.

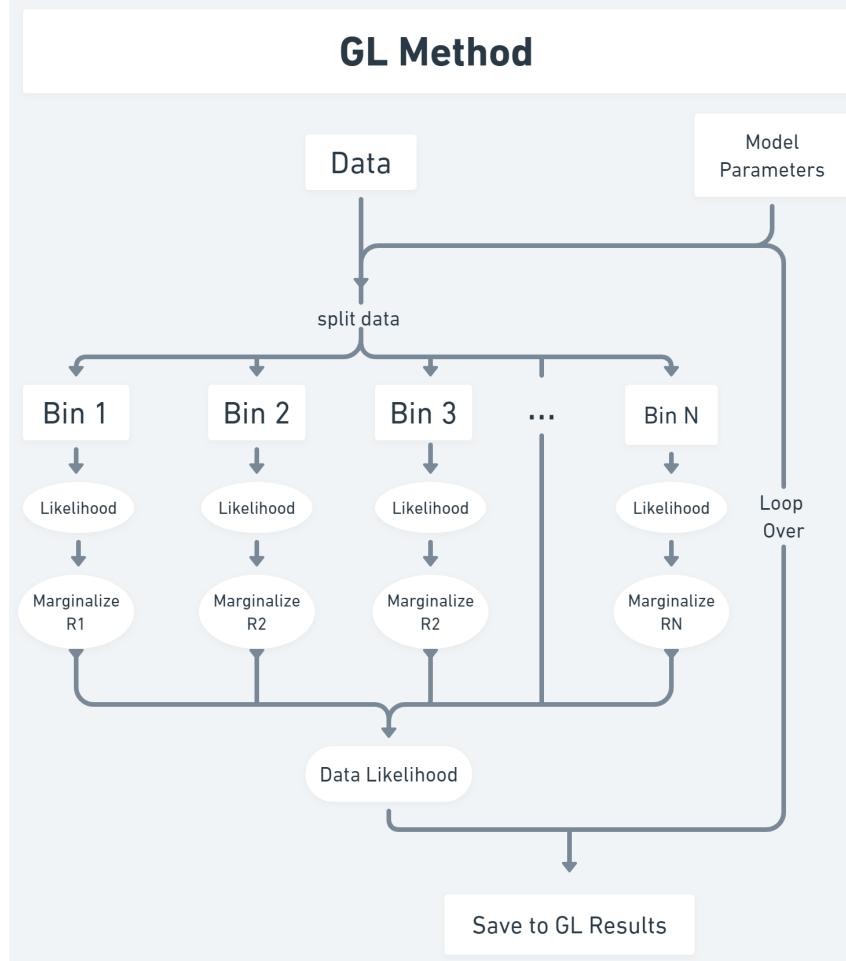


Figure 2.6: Flowchart of Calculation in GL Method

2.3.5 Estimation of frequency

The posterior of the frequency probability distribution can be calculated from:

$$\mathcal{P}(w|D, M_m, I) = \mathcal{P}(w|M_m, I) \frac{\mathcal{P}(D|w, M_m, I)}{\mathcal{P}(D|M_m, I)}$$

As denominator is just a normalisation constant. We will call it C. The first term is prior. And the second term is (2.27) with integral not over w.

$$\mathcal{P}(w|D, M_m, I) = \frac{C}{w} \int_0^{2\pi} d\phi \exp \left(- \sum_{j=1}^m \frac{\chi^2 W_j}{2} \right) \prod_{j=1}^m (W_j^{1/2} [erfc(y_{jmin}) - erfc(y_{jmax})])$$

And w in the denominator comes out from prior, all are constants subsumed in C. This formula tells the probability of frequency w . We can perform this integral at different values of w to get a probability distribution of w for a given GL Model.

Averaging over different GL-models

We can determine the optimal GL model by looking for the one that best satisfies the data. However, in a Bayesian framework, we can improve the posterior distribution of parameters by averaging over all the models.

$$\mathcal{P}(w|m > 1, D, I) = \sum_{m=2}^{m_{max}} \mathcal{P}(M_m|D, I) \mathcal{P}(w|D, M_m, I)$$

The posterior distribution on the right side of the equation is derived from a single model, while the probability of the model on the left side can be determined using Bayes' theorem.

$$\mathcal{P}(M_m|D, I) = \frac{\mathcal{P}(D|M_m, I)}{\sum_{m=2}^{m_{max}} \mathcal{P}(D|M_m, I)}$$

We now possess all the components necessary to complete the calculation. The posterior distribution of ϕ does not provide useful information, so we do not need to calculate it. It is remarkable that we can bypass this calculation in Bayesian inference.

2.3.6 Visualising on Gregory-Lorendo model

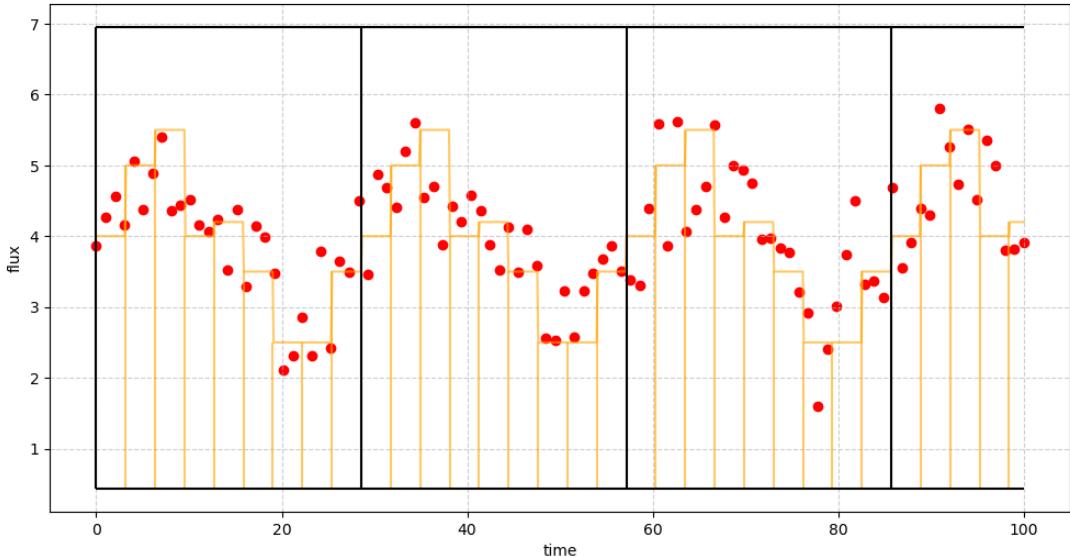


Figure 2.7: GL model sample from Posterior

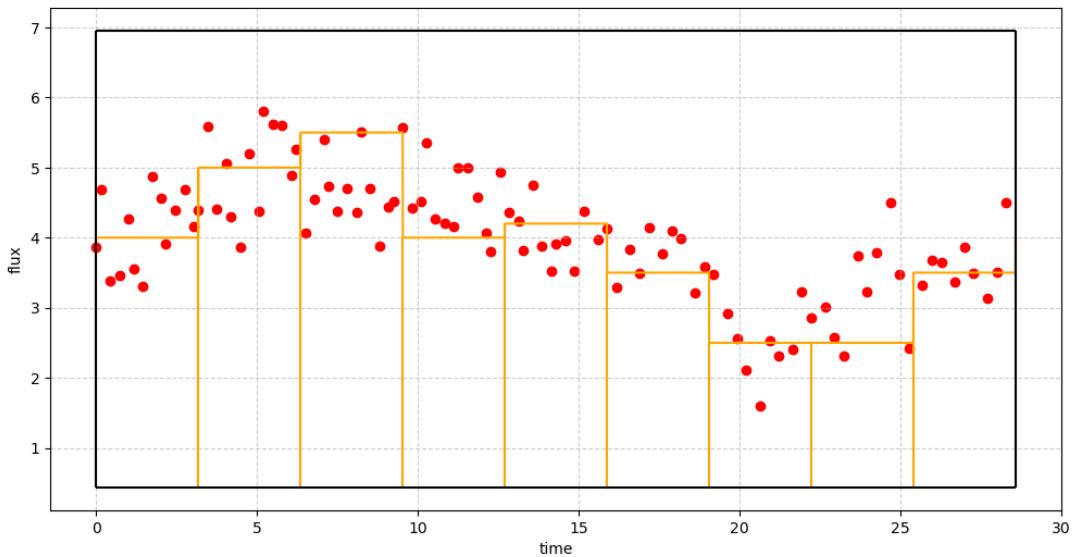


Figure : GL model sample from posterior with data points overlaid after the every period

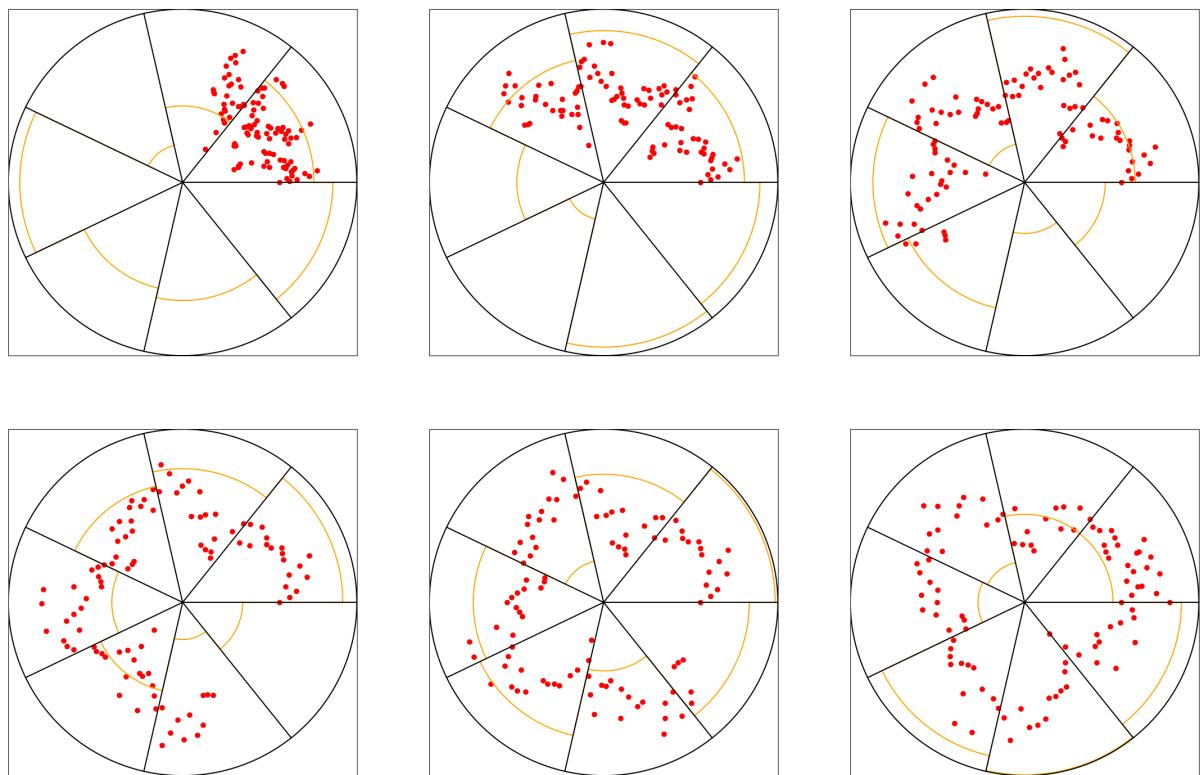


Figure 2.9: A polar plot of a GL model with the period gradually increasing in each successive sample is used to comprehend the wrapping of data on the model.

2.4 Gaussian Processes: Concepts and Applications

A Gaussian process is a set of random variables, any finite number of which have a joint Gaussian distribution. This process can be thought of as a smooth curve fitting technique that does not require explicit models or parameters.

2.4.1 Switching from weight space view to functional space view

Up to this point, we have explicitly defined the model and are trying to optimise the parameters of the model to fit the data. But the Gaussian process is non-parametric. The fitted function by the Gaussian process is of the form

$$f(x) \sim \mathcal{N}(m(x), k(x, x'))$$

It is important to emphasize that the normal distribution applies to the functional space rather than the weight space.

Covariance matrix

The mean value vector $m(x)$ and the covariance matrix are used to determine the Gaussian distribution in the functional space, where each random sample of the Gaussian distribution is a function itself.

$$K(x, x') = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

Kernels

The kernel, often referred to as the covariance function, quantitatively establishes the relationship between any pair of input points. This relationship encodes prior beliefs about the smoothness, periodicity, and other properties of the function being modelled. Within this framework, all possible functions, constrained by the kernel's characteristics, constitute the prior distribution over functions. By leveraging Bayes' theorem, one can infer a posterior distribution over these functions. A pivotal step in this inference is the computation of the marginal likelihood, which requires integration over the infinite-dimensional function space. Fortunately, due to the properties of Gaussian distributions, this integration can be performed analytically[7]. Thus, once data is observed, the posterior distribution over the function space can be determined directly.

2.4.2 Gaussian process for oscillation detection

Our method uses a combination of two kernels appropriate for oscillation detection. The first exponential sine square gives a periodic correlation to priors.

$$k_{\text{ExpSineSquared}}(x, x') = \exp\left(-\frac{2 \sin^2\left(\pi \frac{|x-x'|}{p}\right)}{l^2}\right)$$

And second is white noise kernel.

$$k_{\text{WhiteNoise}}(x, x') = \sigma^2 \delta(x, x')$$

Hyperparameters

Although it is a nonparametric regression model, we still have hyperparameters in the model (parameters of kernels). Once these hyperparameters are known, the posterior distribution can be determined analytically by performing Gaussian integrals in the functional space.

Optimising Hyperparameters

We have three hyperparameters in the model that must be adjusted to identify the most suitable posterior distribution. Each of these hyper-parameters has a different interpretation.

- Length Scale l : Length is used to show the relationship between points that are distant from each other. Points that are close together have a stronger correlation than those that are further apart, which helps to keep the function smooth. The magnitude of the length scale determines how quickly the correlation changes with the distance between the points.
- Period scale p : The magnitude of the period provides an indication of the periodicity of the points, i.e. the interval at which the values repeat.
- Noise scale σ : It gives a quantitative estimate of noise required to fit the data to the model.

Optimising hyperparameters involves maximising the log-likelihood as a function of three parameters. The Scikit-learn gradient descent algorithm is used to optimise these parameters[6].

Priors

The GP method is shown to be highly customised in the sample priors. An infinite variety of functions can be used to fit the data in GP methods. The following are four samples from the Gaussian process Priors, which tells that the Gaussian process can fit almost to any shape of periodic signals.

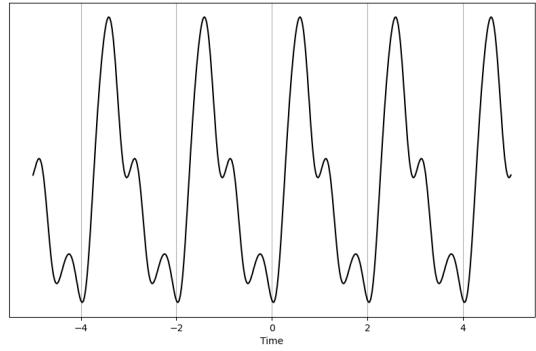
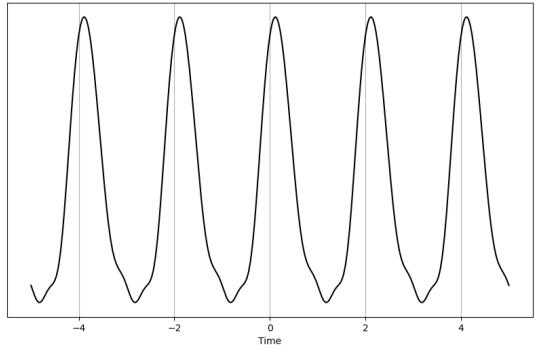
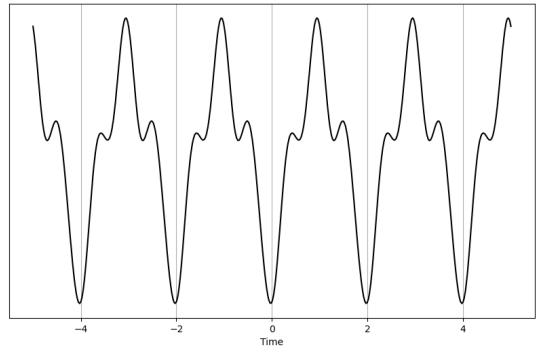
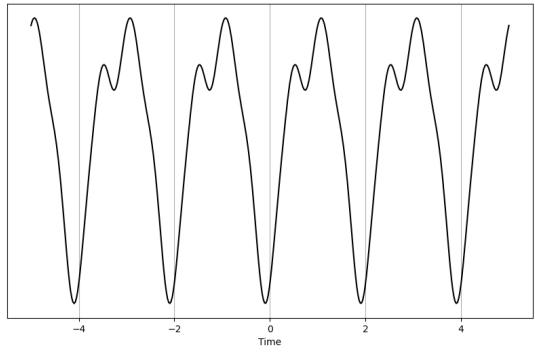


Figure 2.10: Four random samples from Prior used in the GP method

Chapter 3

Theoretical Enhancements to Existing Methods

3.1 Refining the Gregory-Loredo Method

Following is the details calculation flowchart of original GL paper[4]

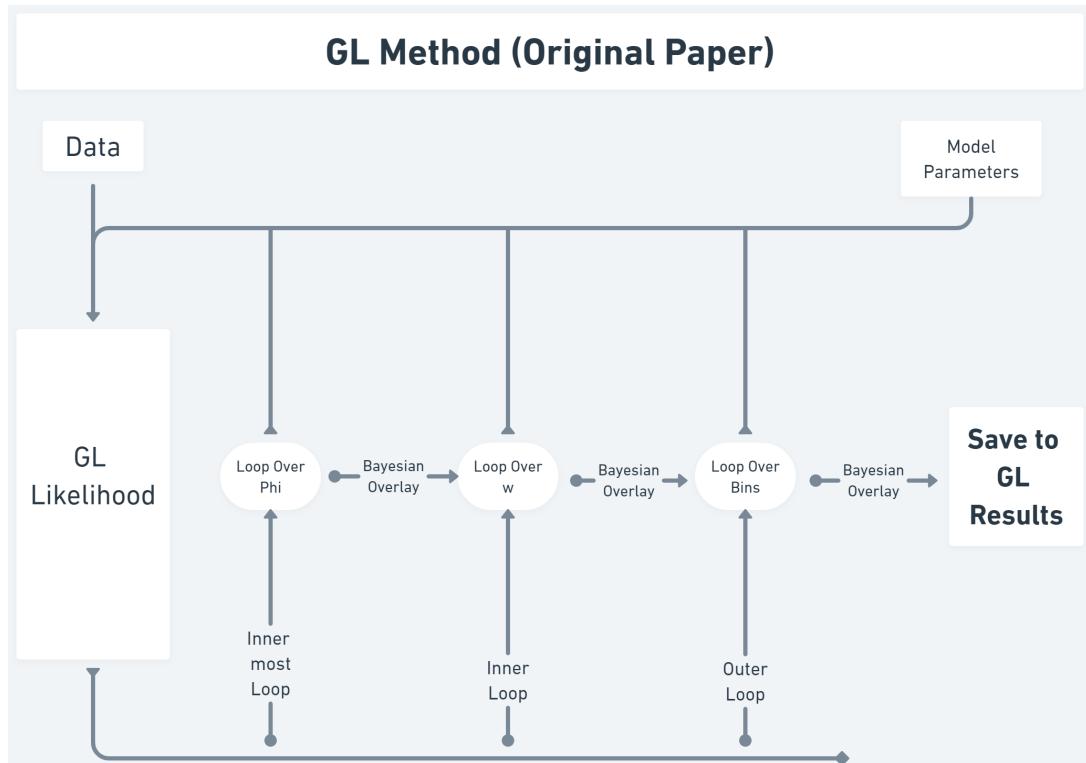


Figure 3.1: GL method calculation flow chart from original paper

We made three Major modification to to algorithm.

- Used the optimised value ϕ instead of marginalising over it.

- Discretise w prior to calculation.
- Linear overlay of different GL models instead of Bayesian overlay

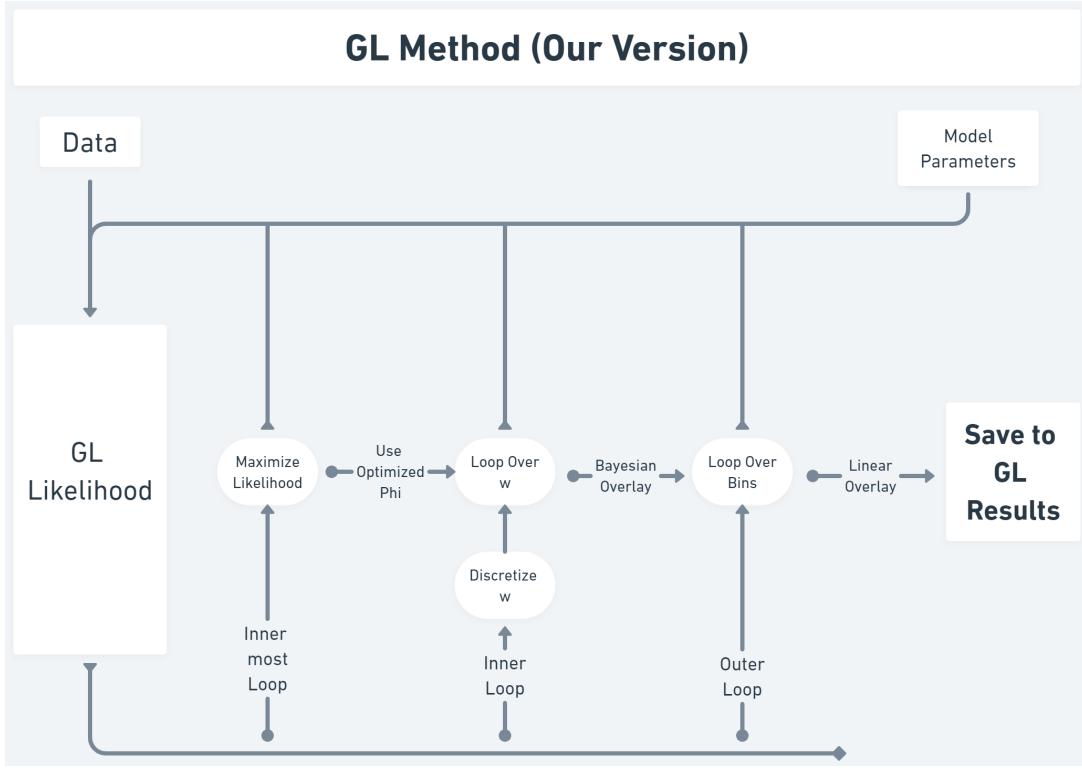


Figure 3.2: GL method calculation flow chart after our modifications

These modifications makes results less reliable but trade off was paid off due to another 20 times faster calculation as compared to original method.

3.1.1 Log-Likelihood vs Likelihood

The log likelihood is selected for analysis instead of the normal likelihood, which was used in Gregory's original work, for the following reasons.

- In most cases, the normal likelihood is astronomically low and prone to numerical errors. Furthermore, for signals that are well behaved, the likelihood peak is so sharp that our discrete step calculation may miss the peak and may result in a flat posterior. Additionally, this sharp peak makes it difficult to perform numerical integration, such as the trapezoid rule or Simpson rule, as the peak can be computationally expensive to determine.
- Our findings demonstrate that the log-likelihood behaves similarly to a power spectrum. This is because when the log of the square exponential is taken, the likelihood is proportional to the square of the difference between the model and the observation, which is the same as the power spectrum, which is the square of the absolute

of the Fourier transform. Therefore, both the Fourier power spectrum and the log-likelihood scale with the square of the deviation, which explains why the power spectrum and log-likelihood yield similar results.

3.1.2 Behaviour of GL likelihood with phase

The original paper on the GL, phase marginalised integration over the phase from zero to 2π . However, it is more efficient to integrate it from zero to $1/noofbins$. (I am using the convention of frequency as the inverse of period, removing the factor of 2π used in astronomical papers.) Due to the nature of the model, every bin is equally likely to be the first bin in the period.

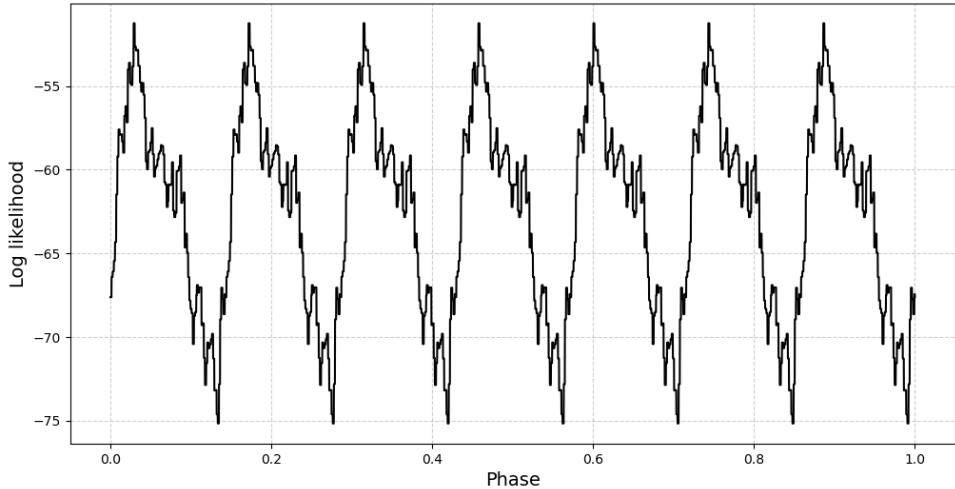


Figure 3.3: The log-likelihood of the GL Model is plotted against the phase within a range of one period.

3.1.3 Discontinuous Likelihood of GL

After zooming in on the likelihood as a function of phase and frequency, I discovered that it was discontinuous and jumped between values. This was counter-intuitive at first, but it made sense.

The fact that when phase or frequency is slightly altered causes a certain range of behaviour to remain constant. This is because when the model is compressed or stretched on the time axis, the data points remain the same. The likelihood will also remain the same if the data points stay in the same bin, as the likelihood only takes into account the residue, and the model stays the same in the particular bin. Discrete jumps occur when the data points jump from one bin to another, resulting in a discrete change in likelihood, which is visible in the observations.

This explanation may make the observation appear straightforward; however, it is more complex than that. The likelihood is plotted after marginalising over r , the height of the bins, and the discreteness should be considered as a mere observation at this stage.

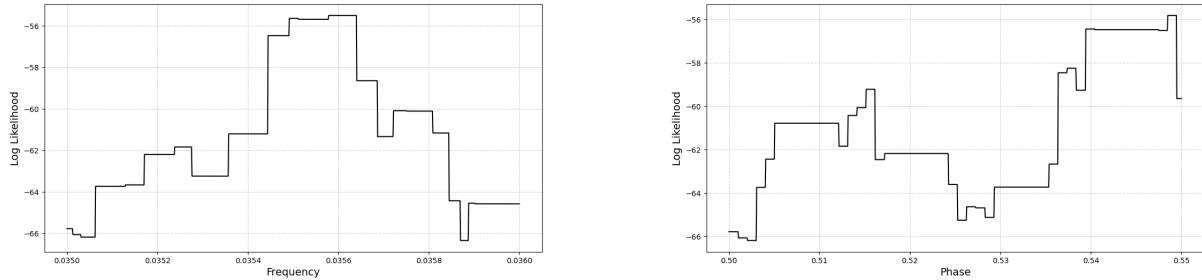


Figure : Log-likelihood of GL (zoomed in) plotted against frequency (top) and plotted against phase (bottom).

The discreteness of the original GL calculation makes it computationally expensive, as gradient-based numerical calculations cannot accommodate this nature of the likelihood.

3.2 Advancements in Gaussian Process Techniques

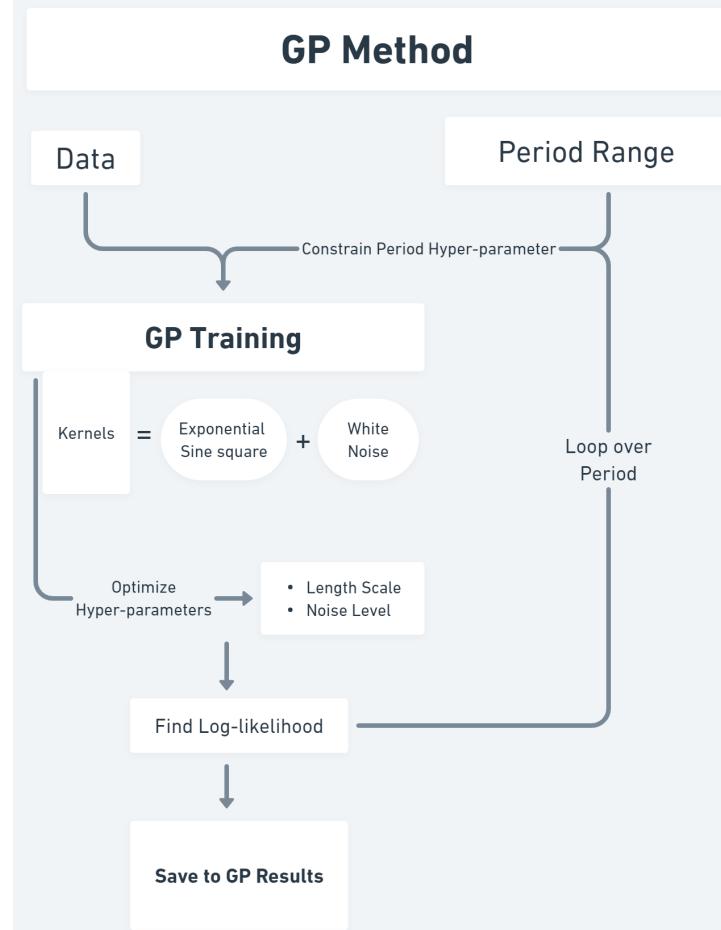


Figure 3.4: Calculation flow chart of GP Method

I am constraining the hyperparameters of the period to obtain a spectrum of log-likelihoods vs frequency because it can be compared to those of the GL and FFT methods.

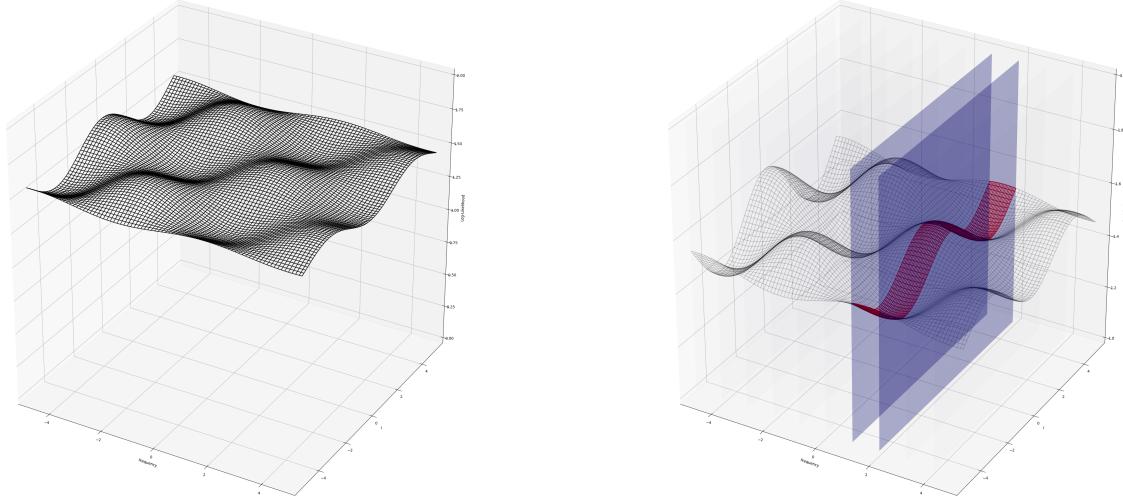


Figure : Log-Likelihood Abstract Fig. (left) and Constraint Function in Shallow Boundaries of Frequency (right)

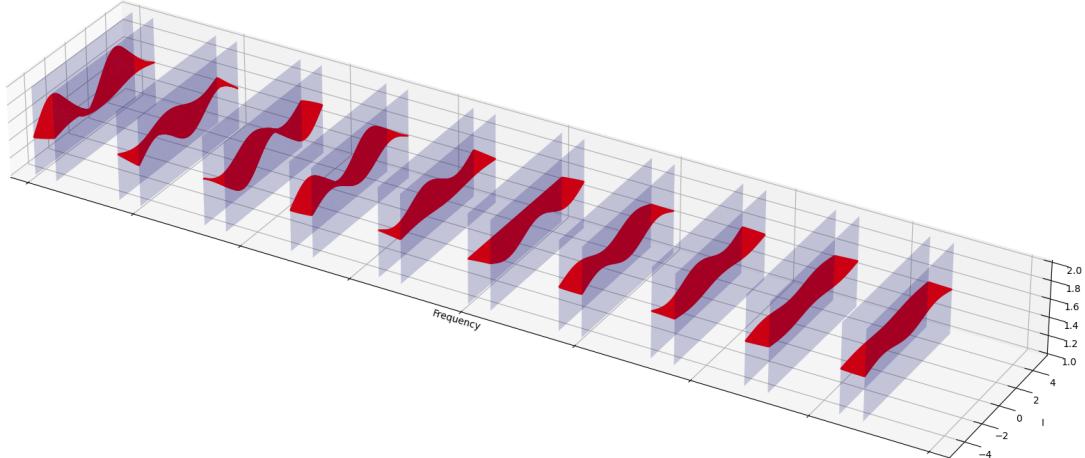


Figure 3.5: . The Log-Likelihood is maximised for each bin to get GP results

In each discrete bin of log likelihood in the above plot, the maximum of each bin is plotted against the median frequency of the bins to get the log-likelihood vs. frequency graph of Gaussian process. Thus our method makes the result of GP easy to compare with the FFT and GL methods.

3.2.1 Relation to GL

Gaussian Process (GP) provides a valuable insight, as it predicts a Gaussian distribution for each point in time. It is the limit of the Gregory-Lorenz model, which also gives a

Gaussian distribution, but fixed over fixed bins. GP improves the results of the GL model by making it smoother and can be thought of as a Gregory-Lorenz model with an infinite number of bins.

Chapter 4

Methodology and Computational Implementation

Gregory's work was initially replicated. The entire Gregory model was re-created from the ground up in Python. Results were obtained that were reasonably close to Gregory's paper[3], though not exactly close as all the parameters used were not taken into account[1]. The Python code for this work is included in the resources. In 1999 paper periodic oscillation in start brightness was determined.

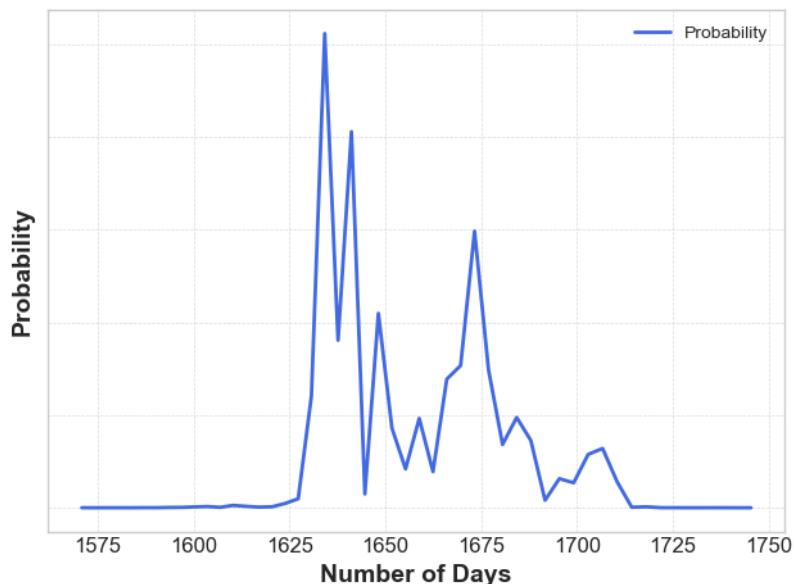


Figure 4.1: Replication of Gregory 1999 paper

4.1 Initial Challenges

Data structure

Initial research was conducted using pandas data frames, as they are quite efficient and optimised for data analysis in Python. However, the nature of GL methods necessitates that the time series be divided into multiple parts, so Numpy arrays with indexing were used later to split the data into bins, which increased the calculation time by at least 50 times.

Arguments

Initially, codes with many arguments were used to instantiate the classes created, which made it quite difficult to work with. Over time, these arguments were automated or default values were given to simplify the code flow. In addition, a single class was used for all methods, resulting in a very user-friendly final algorithm.

4.2 Class Structure

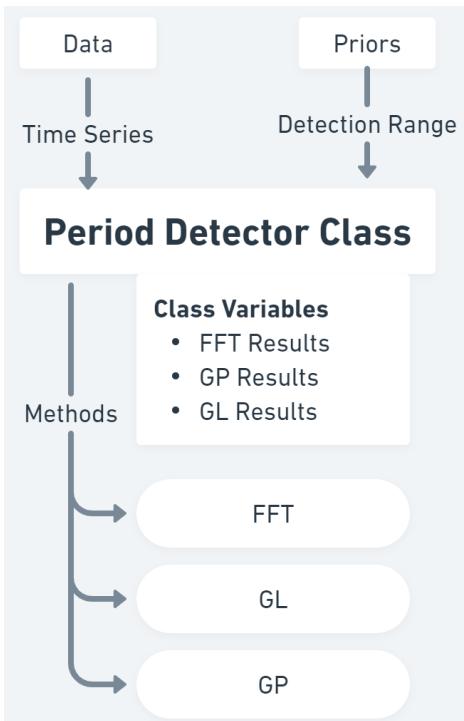


Figure 4.2: Class structure of Oscillation Detector

Once the method is executed within the class, all the results can be obtained as each significant calculation is stored as class variables. This makes the analysis with this code

efficient. Additionally, auxiliary methods to plot data, priors and interactive tools are available.

4.3 Generator Detector Interplay

The final code includes two classes: Generator and detector. The generator can generate periodic time series of three types with eleven choices of parameters with just one or two lines of code. The detector can execute all the methods discussed in this thesis with a single line of code. Both Generator and Detector have auxiliary tools for visualisation.

4.4 Crash Course in Algorithm

Essential Code

```
1 generator = SignalGenerator()
2 pc = Oscillation_Detector(generator.data, w_min=00.02, w_max=0.05)
3 pc.Compute_FFT()
4 pc.Compute_GL()
5 pc.Compute_GP()
```

The first line creates a time series, with a sinusoidal wave as the default. The second command instantiates an oscillation detector class, with the data generated in the previous line and the frequency range to be given as arguments. The last three lines of code simply execute the mentioned algorithm.

Signal Generator

```
1 generator = SignalGenerator(no_data_points=500, time_range= 100,
sample_rate= 5, frequency= 0.5, amplitude= 2, dc= 3, noise=2,
random_seed=42, asymmetry=0.5)
2 generator.generate_triangular_signal()
3 generator.generate_impulsive_signal()
4 generator.plot_pure_data()
5 generator.plot_noise_data()
```

It is possible to generate a signal with the control of eleven parameters, and three distinct types of signals can be produced. Additionally, the signal can be plotted with or without noise.

Mixing Signal

```
1 generator1 = SignalGenerator(frequency=0.050, noise=0)
2 generator2 = SignalGenerator(frequency=0.060)
3 generator2.data.T[1] = generator2.data.T[1] + generator1.data.T[1]
```

Two signals can be mixed using this command.

In-depth Calculation

```
1 pc.Compute_GL(resolution=1000, bins=9)
2 pc.Compute_GP(resolution=500)
3 pc.fft_result
4 pc.GP_result
5 pc.GL_result
```

Resolution used in calculation can be controlled, and best fit frequency of method can be extracted.

Plotting tools

```
1 pc.plot_data()
2 pc.Plot_FFT()
3 pc.Plot_GL()
4 pc.Plot_GP()
5 pc.Sample()
```

There are numerous plotting tools available for creating visualizations of data, models, and outcomes.

These basic tools were used recursively or in loops to get in-depth analysis of these algorithms to get results, which are described in the next chapter. All figures and results in this thesis were produced using this programme. A Jupiter notebook containing all the results is accessible in the resources.

4.5 Resources

Base Code

(750 + lines Base code) [2]:

- Oscillation Detector

Thesis Results

(Jupiter notebooks that can be read online)[1]

- Thesis work

Chapter 5

Results

The results of the study were based on three different types of signals: sinusoidal, triangular, and action potential-like impulses.

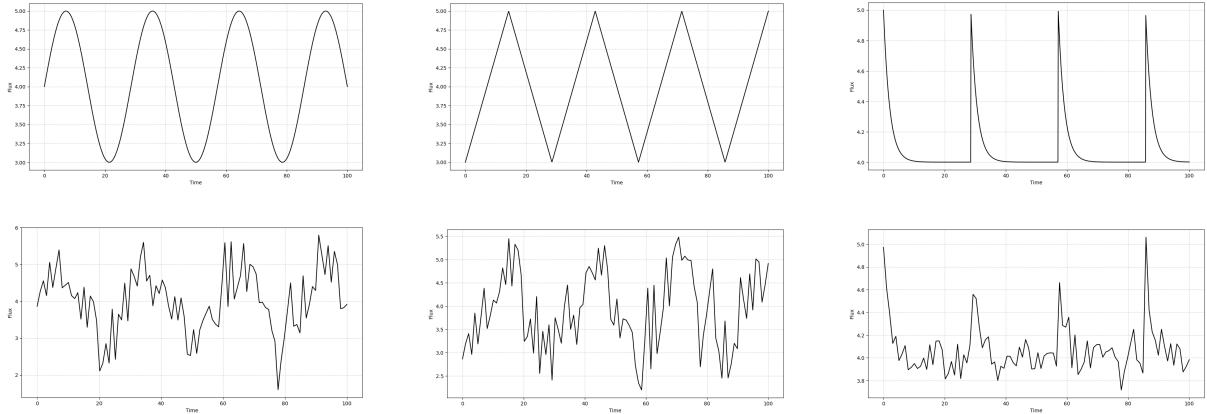


Figure 5.1: Three types of signal used in the analysis in the top row. Signal with added noise in the bottom row.

5.1 With Sinosodial Signal

The experiments were carried out on signals with different frequencies, levels of noise, and amounts of data points.

As frequency increases, all methods improve, but GP yields significantly better results than the others. It is important to note that the oscillations in the results are due to the discrete resolution of the methods, as there is a magnetic effect near the methods' frequency, and the results become better. All the methods produce better results with increasing data points. Moreover, the results become more deviated as the noise increases, with the deviation being comparatively greater in GL than in GP.

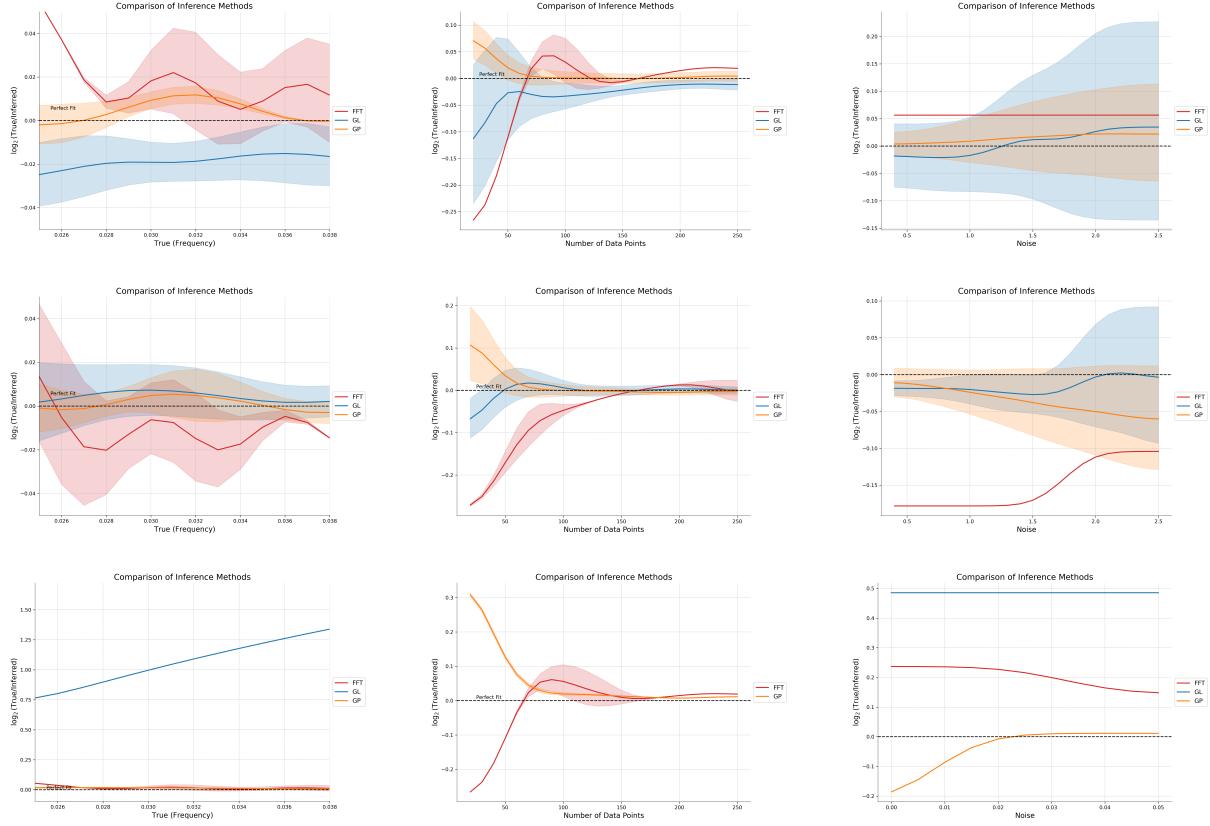


Figure 5.2: In all figures, the y-axis depicts the frequency $\log_2 \frac{\text{true}}{\text{Inferred}}$. FFT is represented in red, GL in blue, and GP in yellow. The standard deviation is indicated in most of the figures. The first row illustrates sinusoidal signals, followed by triangular signals in the second, and action potential-like signals in the third. For the columns, the first uses the frequency on the x-axis, the second displays the number of data points, and the third represents the noise scale on the x-axis.

5.2 With saw-tooth Signal

It was noticed that the same behaviour occurred with different frequencies and data points. However, a peculiar observation was made when the noise was increased; GL and FFT actually improved with the increased noise. This could be because the triangular waves become more sinusoidal with the noise and thus are more easily detected.

5.3 With Impulsive Signal

It was found that GL was ineffective when dealing with impulsive signals, while GP was able to provide reliable results even when there was a lot of noise and few data points.

5.4 Noise Analysis

5.4.1 Pure Noise

In order to accurately distinguish between noise and signal, it is essential to have a pure noise sample. The following graph displays the spectrum of all methods.

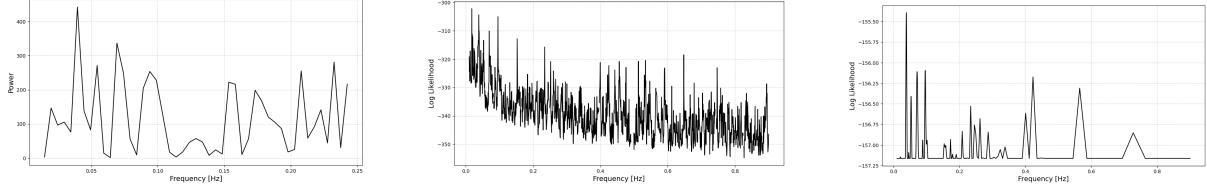


Figure 5.3: Power Spectrum vs frequency of FFT on left with pure Gaussian noise in time series, Log-Likelihood vs frequency of GL in middle and GP on right

5.4.2 Signal with embedded noise

The evolution of the spectrum with noise from different methods was studied, and the following are the results.

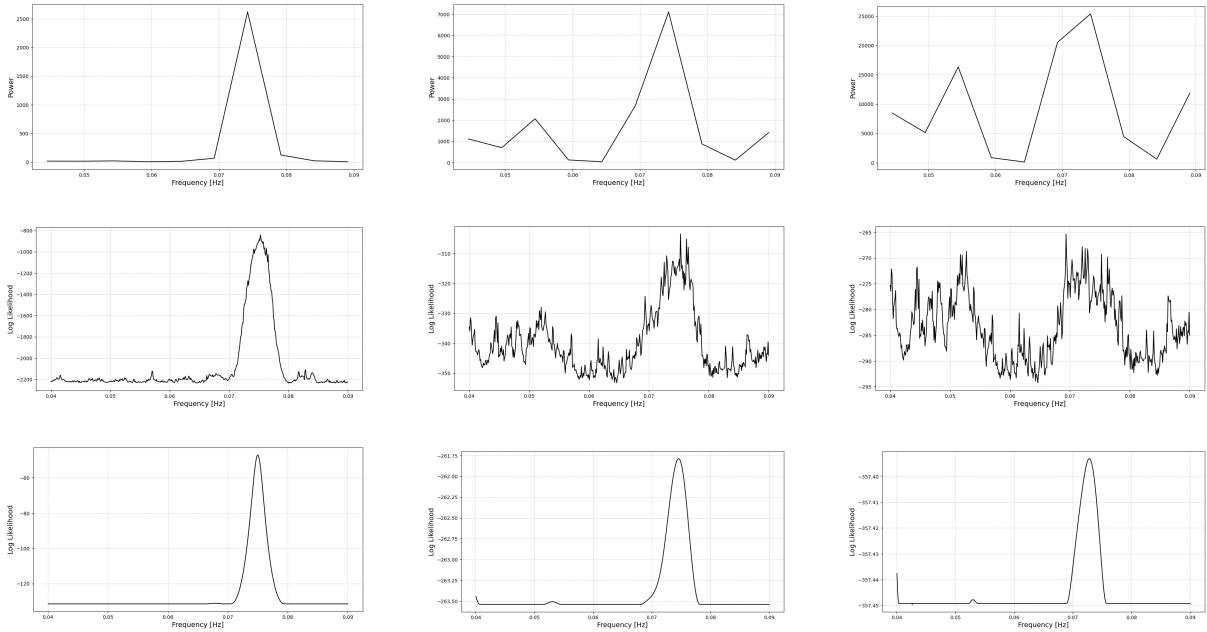


Figure 5.4: The first row displays the FFT power spectrum against frequency, with the noise increasing from left to right. The second row illustrates the GL log-likelihood against frequency, with a similar increase in noise from left to right. The third row presents the GP log-likelihood against frequency, again with noise intensifying from left to right.

The Fast Fourier Transform (FFT) spectrum is highly dependable when noise is amplified in data, as it begins to display multiple peaks, and the intensity of the peaks increases steadily until the technique is rendered ineffective at a certain noise.

GL methods are quite unreliable, as frequency spectrum is very noisy even in lower noise, but these anomalies have chances of being fixed as explored in previous chapter.

GP is best with detecting oscillation in high noisy environment, It can be seen in results as GP results are practically same as noise increases when on similar data set FFT was rendered ineffective.

Chapter 6

Discussion

6.1 Strong peak at half frequency

A strong peak is seen at half the frequency or double the period, and likewise at a quarter of the frequency and quadruple the period, and so on. This is applicable to periodic models such as GL and GP.

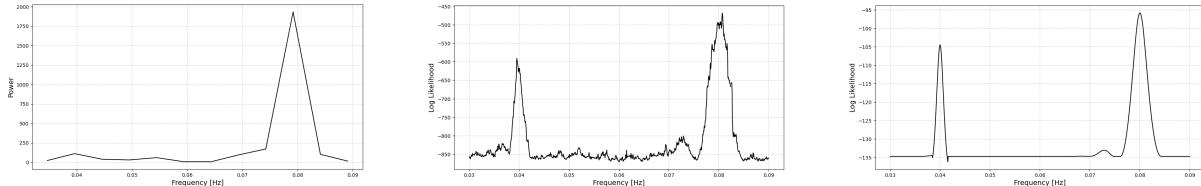


Figure : The first plot (left) shows the FFT power spectrum against a frequency in the range of 0.03 to 0.09 Hz. The middle plot shows the logarithmic likelihood versus frequency for the GL method within the same frequency range. The last plot (right) represents the log-likelihood of the GP. All analyses use a signal with a frequency of 0.08 Hz.

The Fast Fourier Transform (FFT) does not provide the peaks that are expected, as it measures the components of sine and cosine waves, which have a predetermined shape. It is worth noting that these peaks at half the frequency are usually weaker, likely because of the fewer periods that are included in this part of the model.

6.2 Better Resolution and range than FFT

Bayesian methods have a much higher resolution than the fast Fourier transform (FFT). FFT has a fixed sampling rate discussed in the second chapter. However, the resolution of Bayesian methods can be adjusted based on the requirements and available computational resources.

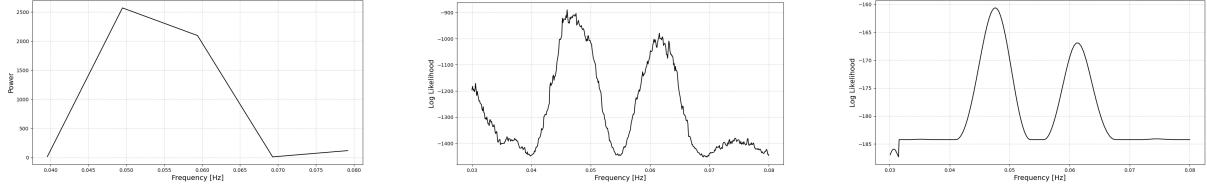


Figure 6.2: The left plot illustrates the FFT Power Spectrum against frequency, the middle plot presents the log-likelihood vs frequency for the GL method, and the right plot displays the log-likelihood vs frequency for the GP method. The analysed signal oscillates at two frequencies: 0.050 and 0.055 Hz.

6.3 Combination of frequency's

Anomalous frequencies can be observed in the spectrum of Bayesian methods; however, most of them can be explained by frequency mixing. As illustrated in the figure, the combination of frequencies at 50 and 75 in arbitrary units produces a strong peak at 25.

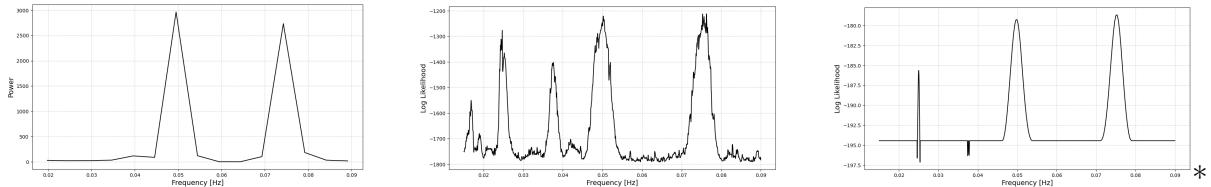


Figure : On the left, the plot displays the FFT Power Spectrum versus frequency. The middle plot illustrates the logarithmic likelihood against frequency for the GL method, while the right plot displays the logarithmic likelihood against frequency for the GP method. The time series analysed contains two oscillating frequencies: 0.050 Hz and 0.075 Hz.

6.4 Possible improvements on algorithms

GL methods can be significantly enhanced, as I mentioned in the GL chapter, by having a trade-off with calculation time. To further improve GP, it is possible to experiment with different kernels based on the signal being analysed.

6.5 Possible improvements on analysis

I have only used the power spectrum and log-likelihood to obtain my results, but my code provides all the necessary calculations that can be used for a more in-depth analysis of the data, such as comparing models or classifying data.

6.6 Final Words

The purpose of this project was to provide an alternative to the Fast Fourier Transform (FFT) for the analysis of oscillating signals. Bayesian methods, such as Gregory-Lorendo and Gaussian processes, are capable of working with sparse and noisy data points and have various applications. The best thing this project achieved is that all these algorithms being used in this thesis can be used by my grandchildren of all backgrounds with simple lines of codes.

Bibliography

- [1] Ritesh Balayan. MSC_Thesis_work GitHub Repository. https://github.com/RiteshBalayan/MSC_Thesis_work, 2023. Accessed: yyyy-mm-dd.
- [2] Ritesh Balayan. Oscilation_Detector GitHub Repository. https://github.com/RiteshBalayan/Oscilation_Detector, 2023. Accessed: yyyy-mm-dd.
- [3] P. Gregory and Thomas Loredo. A new method for the detection of a periodic signal of unknown shape and period. *The Astrophysical Journal*, 398, 05 1999.
- [4] P. C. Gregory and Thomas J. Loredo. A new method for the detection of a periodic signal of unknown shape and period. *Astrophysical Journal*, 398:146, October 1992.
- [5] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.
- [6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [7] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.