STONY BROOK UNIVERSITY
DEPARTMENT OF PHYSICS AND ASTRONOMY

AST 390: SPECIAL TOPICS IN ASTROPHYSICS AND COSMOLOGY:
COMPUTATIONAL ASTROPHYSICS

Some Aspects of the Spectral Analysis of Non-Uniformly Sampled Data

Sameer Singh
Instructor: Prof. Michael Zingale

Contents

1 Motivation **1**

2 Resources and Data Availability **2**

3 Theoretical Overview **2**
3.1 The Fourier Transform, the Convolution Theorem, and Frequency Spacing . . . 2
3.2 Non-Uniform Spacing . . . 3
3.3 The Classical and Lomb-Scargle Periodograms . . . 4
3.4 Variations of the Lomb-Scargle Periodogram . . . 4
3.4.1 Noise Handling . . . 4
3.4.2 Fitting Form Modifications . . . 5
3.4.3 The Bayesian Periodogram . . . 5

4 Codes Exploring Basic Aspects of the Lomb-Scargle Periodogram **5**
4.1 Helper Subroutines . . . 5
4.2 Basic Lomb-Scargle Implementation, Frequency Grid Considerations, and Scaling . . . 5
4.3 Application to an Eclipsing Binary . . . 6

5 Codes Exploring Subtler Aspects of the Lomb Scargle Periodogram **7**
5.1 Floating Mean Adjustment . . . 7
5.2 Effects due to Window Functions . . . 8
5.3 Failure Modes due to Aliasing and Pseudo-Aliasing . . . 8

6 Conclusion **10**

Term Project
May 8, 2025

1 Motivation

In class, we studied the Fourier transform (and in particular the discrete and fast variants) as a means for the spectral analysis of data. There was, however, a caveat: the Fourier transform requires uniformly spaced data. In practice, we may have to deal with non-uniformly sampled data: perhaps we originally sampled data with uniform spacing, but lost some non-consecutive measurements due to instrumental dropout (the "missing data problem"); or perhaps, due to not having full control over the instrumentation (as may happen when requesting telescope time), we were simply dictated a set of non-uniformly spaced data [1]. To salvage our ability to use the FFT in such cases, we could in principle go from unevenly spaced t_i's to evenly spaced ones by, say, superimposing an evenly-spaced grid on the uneven one and interpolating. In the missing data problem in particular, we need only interpolate for the missing data points (for the original data *would* have been evenly spaced): it is standard to set these values to zero or to set them to the last accessible value. Alas, "the experience of practitioners of such interpolation techniques is not reassuring" [1]. We wrote a program to show the FFT's failure at spectral analysis in this context; note that although in our discussion thereof we will quote specific values for the parameters, these are settable; our code is here.

We begin by constructing a sinusoid $y = \sin t$ at $N(= 1000)$ evenly spaced points between $t_{\text{min}}(= 0)$ and $t_{\text{max}}(= 20)$. We perturb this sinusoid with Gaussian-normalized random error, scaled by a factor $\sigma(= 0.5)$; this yields our experimental data (without dropout) (Figure 1a). We then take the FFT of this data using `numpy.fft.rfft`, convert the resultant frequencies into physical ones, and plot the power spectrum, finding excellent agreement with the expected frequency of $\sin t$, $1/2\pi$ (Figure 1b). We then remove non-consecutive intervals of varying length from the experimental data, simulating instrumental dropout: we do this by specifying the number $n(= 5)$ of intervals to be removed (5), generating a random integer $\text{ID}_{i,\text{start}}$ from 0 to N for the i^{th} interval ($i = 1, \ldots, n$), corresponding to the beginning of a dropout incident; and then by generating a random integer L_i from 1 to a specified $L_{\text{max}}(= N/5 = 200)$ corresponding to the duration of the i^{th} dropout interval. Thus we have that the i^{th} dropout interval consists of the points at steps $[\text{ID}_{i,\text{start}}, \text{ID}_{i,\text{start}} + 1, \ldots, \text{ID}_{i,\text{start}} + (L_i - 2), \text{ID}_{i,\text{start}} + (L_i - 1)]$ (with, to be clear, random variable start and random variable length). We then remove all n such intervals from the experimental data, yielding Figure 1d from Figure 1a; Figure 1c is just a cleaner visualization of the functionality of this algorithm, unobscured by noise. We then apply a modified version of the interpolation scheme alluded to in [1]: for those intervals whose extent is at least than $\Delta(= 0.05)$ as a fraction of N, we clamp the data at the last measured value, for these intervals are so large that it does not offer any additional benefit to use a more complicated interpolation scheme (we may well just be missing periods, etc.); for those intervals with lesser extent, we take the average of the last measured value and the next measured value, amounting to a linear interpolation. This scheme results in Figure 1e from Figure 1d. Finally, we take the FFT of the interpolated data and plot the power spectrum (Figure 1f), finding egregious disagreement with the underlying $1/2\pi$ frequency. It is particularly interesting to note the spurious excess of power at low frequencies, which results from the fact that the scale of the gaps in the data are comparable to the wavelength of the signal. The upshot is that in order to deal with the spectral analysis of non-uniform data, we need to refine our thinking.

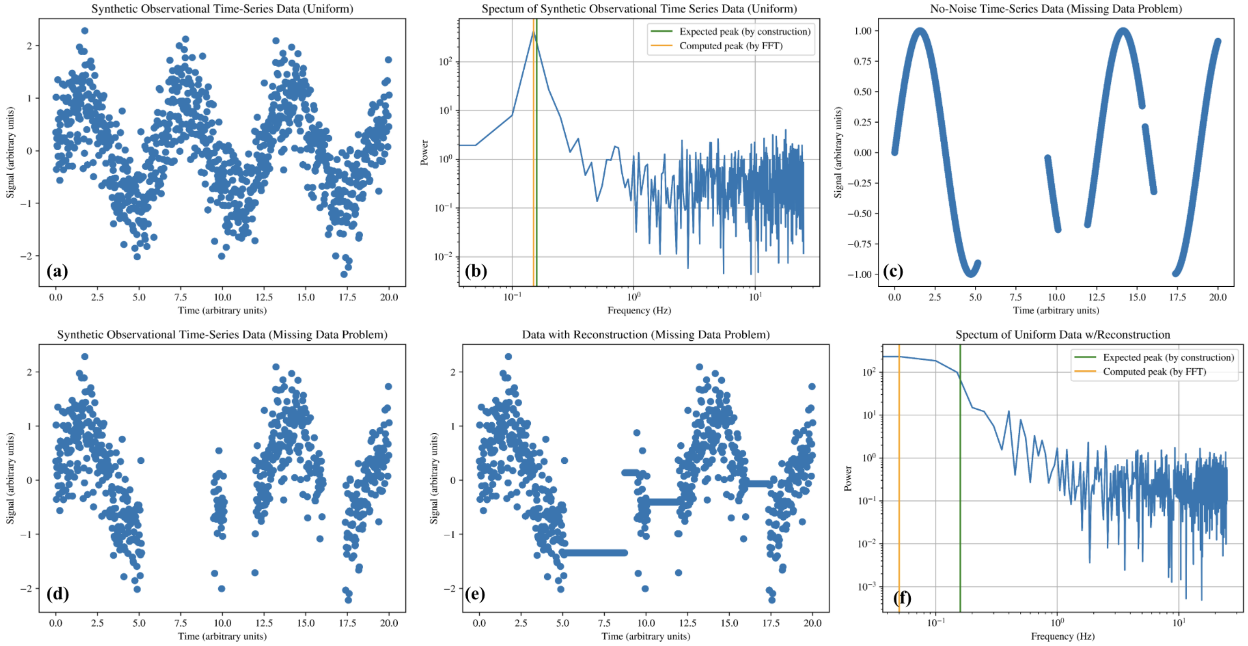

Figure 1: Failure of interpolation + FFT in simulated missing data problem

2 Resources and Data Availability

The Large Synoptic Survey Telescope Corporation Data Science Fellowship Program has made available under an MIT license a repository of lecture slides, Jupyter notebooks, and other material from their special schools. Among these are two notebooks on the Lomb-Scargle periodogram from A.A. Miller: here [2] and here [3]. The code and results described in §4 (except §4.4, inspired by `LightKurve`) come from our own working of the exercises in the first notebook, and the code and results described in §5 come from our own working of the exercises in the second notebook. We modified the notebooks as we found necessary. Note also that in the second notebook, we also often used the `astropy` implementation of the fast Lomb-Scargle algorithm [4]. Our inspiration for the analysis of KIC 10264202 in §4.4 comes from this tutorial [5] from the developers of the `LightKurve` package (although we do not use `LightKurve` analysis code); our code is here. The theoretical discussion in §3 is based on the now classic 2018 paper by J.T. VanderPlas [6]; beyond this, we draw inspiration from the paper in understanding time-complexity results in §4.3 and failure modes in §5.3. Of course, in various parts of our code, we used the standard `NumPy`, `SciPy`, and `matplotlib` suite.

All of the code we wrote for this project is available in one place at this GitHub respository.

3 Theoretical Overview

In this section, we discuss classical spectral analysis with the Fourier transform, and generalize it to the non-uniform case of the Lomb-Scargle Periodogram.

3.1 The Fourier Transform, the Convolution Theorem, and Frequency Spacing

Recall that the Fourier transform of a continous signal $g(t)$ takes the form

$$\hat{g}(f) \equiv \int_{\infty}^{\infty} g(t)e^{-2\pi i f t}dt \tag{1}$$

Recall also that the squared amplitude of the transform is the *power spectrum*:

$$\mathscr{P}_g = |F\{g\}|^2 \tag{2}$$

which is a positive, real-valued function that quantifies "how much" of frequency f is present in the data.

Now, the sampling of a continuous function $g(t)$ at regular intervals spaced by Δt is equivalent to the pointwise multiplication of $g(t)$ by a so-called *Dirac comb* (or "impulse train"), $\text{III}_{\Delta t}(t)$, which represents a series of δ-functions at the measurement points: $g_{\text{obs}} = g(t)\text{III}_{\Delta t}(t)$. More generally, an observed signal is the pointwise product of the underlying signal with a window function $W(t)$, $g_{\text{obs}}(t) = g(t)W(t)$ and the Fourier transform $\mathscr{F}\{g_{\text{obs}}\} = \mathscr{F}\{g\} * \mathscr{F}\{W\}$, with $*$ representing convolution, according to the *convolution theorem*. Now, the Fourier transform according to (1) in the Dirac comb case is

$$\hat{g}_{\text{obs}}(f) = \sum_{n=-\infty}^{\infty} g(n\Delta t)e^{-2\pi i f n\Delta T} \tag{3}$$

with the integral becoming a sum due to the discrete sampling introduced by the Dirac comb. In practice, we only have a finite of number of samples N, and only frequencies from 0 to $1/\Delta T$ represent unique content—this is because the Fourier transform of a Dirac comb is another Dirac comb, so such a window creates a series of *aliases* of the underlying transform spaced by $1/T$, making everything outside of $0 \leq f < 1/T$ simply repeated information. If we define $g_n \equiv g(n\Delta t)$, we have

$$\hat{g}_{\text{obs}}(f) = \sum_{n=0}^{N} g_n e^{-2\pi i f n\Delta t} \tag{4}$$

and with N frequencies uniformly spaced with $\Delta f = 1/(N\Delta t)$ in the relevant range and the notation $\hat{g}_k \equiv \hat{g}_{\text{obs}}(k\Delta f)$, this becomes the familiar *discrete Fourier transform*:

$$\hat{g}_k = \sum_{n=0}^{N} g_n e^{-2\pi i k n/N} \tag{5}$$

A subtle point to note is that in going from (3) to (4) we applied a rectangular window function of width $N\Delta t$; the Fourier transform of such a function is a sinc function of width $1/(N\Delta t)$, so by our earlier point about the convolution theorem, the resultant Fourier transform in (4) is actually "smeared" due to its convolution with the sinc function, leading to the effective blurring of frequencies within $1/(N\Delta t)$ of one another—meaning that it only makes sense to sample with $\Delta f \geq 1/(N\Delta t)$.

On the subject of smart sampling frequencies, we must certainly mention the *Nyquist frequency* f_{Ny}. The *Nyquist-Shannon theorem* says that in order to faithfully represent the spectral content of a signal band-limited between $\pm B$ (i.e. having zero Fourier transform outside $\pm B$), we must sample the signal at a rate at least $f_{\text{Ny}} = 2B$. Another view is that if we have a function regularly sampled at $f_0 = 1/T$, we can only fully recover the spectral content if the signal is band-limited between $\pm f_0/2$. In fact this latter view follows naturally from the Dirac comb perspective. The Fourier transform of a function not band-limited as such will not "fit" in between the teeth of the Fourier transformed Dirac comb window with width $1/T$, leading to overlapping of different portions of the signal and making the true Fourier transform irrecoverable.

3.2 Non-Uniform Spacing

Formally, the evenly spaced Dirac comb is $\text{III}_{\Delta t}(t) \equiv \sum_{n=-\infty}^{\infty} \delta(t - nT)$. The analogous non-uniform window, with measurements at N times $\{t_n\}$, looks similar: $W_{\{t_n\}}(t) = \sum_{n=1}^{N} \delta(t - t_n)$. In the non-uniform case, then, we have $g_{\text{obs}}(t) = g(t)W_{\{t_n\}}(t) = \sum_{n=1}^{N} g(t_n)\delta(t - t_n)$ and $\mathscr{F}\{g_{\text{obs}}\} = \mathscr{F}\{g\} * \mathscr{F}\{W_{\{t_n\}}\}$. A key point is that the nonstructured spacing of the t_n leads to nonstructured peaks in $\mathscr{F}\{W_{\{t_n\}}\}$, which in turn leads to random peaks in $\mathscr{F}\{g_{\text{obs}}\}$ by the convolution theorem. Importantly, this destroys in the non-uniform case the symmetry that explained the origin of the Nyquist limit in the last section. The upshot is that the Nyquist limit does *not* apply to non-uniform data, and various "psuedo-Nyquist" limits based on one of the mean, harmonic mean, median, or minimum of the sample spacing all fail in that nonuniform data can probe frequencies larger than any of them. One sound extension, however, is in the case where each Δt_i can be written as an integer multiple of some factor; with p the largest such factor, the "Nyquist frequency" is $1/(2p)$. This fails if any two Δt_i have an irrational ratio, but since this would require infinitely-precise measurements of t_i, it does not happen in practice, and the precision of the time measurements sets "f_{Ny}."

3.3 The Classical and Lomb-Scargle Periodograms

The *classical periodogram* is an estimator of the power spectrum and is easily obtained by combining (2) and (5) (apart from normalization):

$$P_S(f) = \frac{1}{N}\left|\sum_{n=1}^{N} g_n e^{-2\pi i f t_n}\right|^2 = \frac{1}{N}\left[\left(\sum_n g_n \cos(2\pi f t_n)\right)^2 + \left(\sum_n g_n \sin(2\pi f t_n)\right)^2\right] \tag{6}$$

Applying (6) to uniformly sampled Gaussian white noise results in χ^2-distributed periodogram values, but this useful statistical property does not hold for uneven data under (6). To redeem the statistical utility, Scargle considered a generalization of (6):

$$P(f) = \frac{A^2}{2}\left(\sum_n g_n \cos(2\pi f[t_n-\tau]\right)^2 + \frac{B^2}{2}\left(\sum_n g_n \sin(2\pi f[t_n-\tau]\right)^2 \tag{7}$$

The A, B, and τ are arbitrary functions of f and the observing times $\{t_i\}$. A particularly useful choice of these parameters gives

$$P_{\rm LS}(f) = \frac{1}{2}\left\{\frac{\left(\sum_n g_n \cos(2\pi f[t_n-\tau]\right)^2}{\sum_n \cos^2(2\pi f[t_n-\tau])} + \frac{\left(\sum_n g_n \sin(2\pi f[t_n-\tau]\right)^2}{\sum_n \sin^2(2\pi f[t_n-\tau])}\right\} \text{ with } \tau = \frac{1}{4\pi f}\tan^{-1}\left(\frac{\sum_n \sin(4\pi f t_n)}{\sum_n \cos(4\pi f t_n)}\right) \tag{8}$$

These are useful in that they make it so that (8) is time-shift invariant, has an analytically computable distribution, and reduces to (6) in the uniform spacing case.

A beautiful result based on the work of Lomb is that the approach (8), which emerges from the perspective of Fourier analysis, is equivalent to a relatively simple approach from the perspective of least squares analysis. In particular, if a sinusoidal model

$$y(t;f) = A_f \sin(2\pi f(t-\phi_f)) \tag{9}$$

is fit at each candidate frequency f (with the amplitude and phase being functions of f) by minimizing the standard χ^2 statistic

$$\chi^2(f) = \sum_n (y_n - y(t_n;f))^2 \tag{10}$$

at each frequency with respect to A_f and ϕ_f, yielding the optimal model $\hat{y}(t;f)$ with minimum $\chi^2 \equiv \hat{\chi}^2(f)$, then (8) is equivalent to

$$P(f) = \frac{1}{2}\left[\hat{\chi}_0^2 - \chi^2(f)\right] \tag{11}$$

with $\hat{\chi}_0^2$ being that resulting from the "fit" simply of the mean of the data, making (11) something of a quantification of how much better sinusoidal variation describes the data than no variation.

3.4 Variations of the Lomb-Scargle Periodogram

3.4.1 Noise Handling

To enable the Lomb-Scargle periodogram to handle Gaussian uncertainties σ_n in the y_n, we need only replace (10) by the familiar version of χ^2 weighted by the σ_n,

$$\chi^2(f) = \sum_n \left(\frac{y_n - y_{\rm model}(t_n;f)}{\sigma_n}\right)^2 \tag{12}$$

Even if the noise is not Gaussian but correlated, we need only replace (10) by the appropriate version of χ^2, which turns out to be $\chi^2(f) = (\mathbf{y}-\mathbf{y}_{\rm model})^T \Sigma^{-1}(\mathbf{y}-\mathbf{y}_{\rm model})$, adopting row vector notation for $\mathbf{y}$ and $\mathbf{y}_{\rm model}$ and with Σ the covariance matrix.

3.4.2 Fitting Form Modifications

The standard Lomb-Scargle periodogram assumes that the data have zero mean; this is not an issue in and of itself, since it can be realized simply by subtracting the sample mean off the data. However, if there is incomplete phase coverage, the sample mean may be a poor estimator of the underlying mean. This problem is alleviated by the "floating-mean" adjustment, which involves adding a constant offset to the model in (9),

$$y_{\text{model}}(t;f) = y_0(f) + A_f \sin(2\pi f(t-\phi_f)) \tag{13}$$

We might also include higher-order Fourier terms in our fit, adding, say, $K-1$ additional sinusoidal components at harmonics of the fundamental frequency: $y_{\text{model}}(t;f) = A_f^0 + \sum_{k=1}^{K} A_f^{(k)} \sin\left(2\pi k f(t-\phi_f^{(k)}\right)$. Such an adjustment allows us to capture more complex variations, but has the effect that the periodogram will be higher at all frequencies and not just those of interest, increasing the incidence of spurious peaks able to wash out true ones.

3.4.3 The Bayesian Periodogram

As we have seen, the Lomb-Scargle periodogram admits natural interpretations from both the Fourier and least squares perspectives. It also turns out to admit a natural interpretation from the Bayesian perspective. In particular, exponentiation of the Lomb-Scargle periodogram gives the posterior probability of frequency f given the data and sinusoidal model. Note, however, that such Bayesian periodograms do not give the probability that the data are *generically* periodic with frequency f, since they are conditioned on the assumption of specifically sinusoidal variation.

4 Codes Exploring Basic Aspects of the Lomb-Scargle Periodogram

In this section, we describe the code we wrote to solve the exercises in the first notebook, and results therefrom.

4.1 Helper Subroutines

We began by writing two helper subroutines that we would use often. The first was used to synthesize data of the form $y = A\sin\left(\frac{2\pi x}{P} - \phi\right) + \sigma_y$ (i.e. Gaussian-perturbed arbitrary sinusoid) on a user-supplied array x. The second was used to *phase fold* data: phase folding refers to the stacking of cycles in data atop one another to generate a signal vs. phase plot, increasing the signal-to-noise. Given x, y data and a period P, our code calculates phases simply as $(x \bmod P)/P \times 2\pi$ (i.e. how many units into a cycle, what fraction of a cycle that number of units is, and a scaling by 2π, since the phase evolution in a cycle is in general 2π).

4.2 Basic Lomb-Scargle Implementation, Frequency Grid Considerations, and Scaling

We implemented the form of Lomb-Scargle given by (11), with χ^2 as in (12). Note that we obtained $\hat{\chi}^2$ using `minimize` from `scipy.optimize`. In Figure 2, we show an example of the use of our code: (a) shows a periodic signal with 100 observations taken at random intervals over a time period of 10 days, $y = 7.4\sin(2\pi t/5.25)$, plus Gaussian noise with variance 0.8; (b) shows the periodogram generated from the data on a frequency grid with minimum frequency $1/10$, maximum frequency 10, and 50 frequencies overall, resulting in undercoverage and thus a poor result; (c) shows the periodogram generated from the data on a frequency grid with minimum frequency $1/10$, maximum frequency 10, and an adequate 1195 frequencies, finding the peak power at ≈ 5.29 days, very close to the expected 5.25; (d) shows the data in (a) phase folded on the period from (c), resulting in a clean plot showing the form of the oscillation. We determined that we needed 1195 frequencies for adequate coverage by the following general recipe

- $f_{\min}$ should correspond to the slowest conceivable signal, i.e. one which only experiences one cycle over the whole span of the data (in practice, putting $f_{\min} = 0$ is also fine and not computationally much different)
- $f_{\max}$ should just be the Nyquist frequency in the case of uniform data, but since as we discussed this concept is ill-defined for nonuniform data, a practical recommendation is to put $f_{\max}$ to the most rapid variation expected from the data (which usually requires some reference to the underlying astronomy/phenomenon, so in this case of synthetic data our choice of $1/10$ was a little arbitrary)

- Δf should respect the fact that the grid spacings should be smaller than the expected widths of the periodogram peaks. Earlier we noted that because of the convolution theorem and the fact that the Fourier transform of a rectangular function is a sinc function, data observed with a rectangular window of a width T will have peaks "smeared" to width $\sim 1/T$, suggesting $\Delta f \sim 1/T$. To make sure that peaks are adequately sampled, though, an oversampling factor n_0 should be applied, $\Delta f = 1/(n_o T)$. We used the typical $n_0 = 5$.

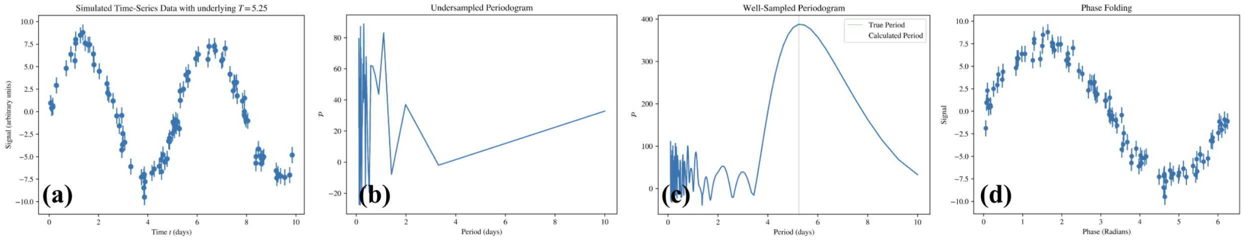

Figure 2: Basic Lomb-Scargle on simulated data

Incidentally, by this method, we calculated that for LSST light curves, with a survey duration of 10 years and a smallest recoverable period of one hour (an astronomically-informed choice), the optimal frequency grid would have 437996 grid points. This number is prohibitively large, because Lomb-Scargle scales like $\mathcal{O}(N^2)$ with the number of points: in (8), we see that the computation requires sums over N sinusoids at each of the N_f frequencies, and the number of frequencies required is proportional to the number of points. Moreover, these operations are not just adds or multiplies; they are expensive calls to trigonometric functions. If the frequencies examined are linearly spaced (as we have been implicitly discussing), then a factor of four speed-up can be obtained by replacing the trigonometric calls by recurrences (relying on trigonometric addition formulas), as suggested by [1]. Still, though, this is not much of an improvement in the face of N^2. Fortunately, $\mathcal{O}(N \log N)$ approximate methods to arbitrary precision exist, including a popular one due to Press & Rybicki which relies on inverse interpolation (or "extirpation") and an FFT to compute the trigonometric terms in (8) (note that this does not mean that the periodogram itself is an FFT periodogram; the FFT is just used to speed up (8) [1]). Various implementations with various scalings are available in the `LombScargle` class of `astropy.timeseries`: putting `method` = "`slow`" gives the standard algorithm, `method` = "`fast`" gives the Press & Rybicki algorithm, etc. The default is `method` = "`auto`", which heuristically selects among the possible options.

4.3 Application to an Eclipsing Binary

We downloaded and cleaned a quarter's worth of data for the eclipsing binary KIC 10264202 from the original Kepler mission using the `search_lightcurve` function in the `LightKurve` package; Figure 3a shows the light curve. We used `astropy` to compute the Lomb-Scargle periodogram, shown in Figures 3b and 3c (note that power is normalized by default); significant structure due to aliasing is visible. We thus identified the period as ≈ 0.52 days. However, on folding on this period, we obtained an incorrect looking plot, Figure 3d. This was not surprising, because for an eclipsing binary system, we would expect both a primary and secondary transit, and a one-term sinusoid would not be able to fit both. To account for this, we phase folded on half of 0.52 days (in case 0.52 was the second harmonic), obtaining further nonsense, Figure 3e, and on twice 0.52 days (the second harmonic), obtaining a clean plot showing both transits, Figure 3f. We thus concluded that KIC 10264202 has a period of ≈ 1 day, agreeing with the `LightKurve` analysis of the same object.

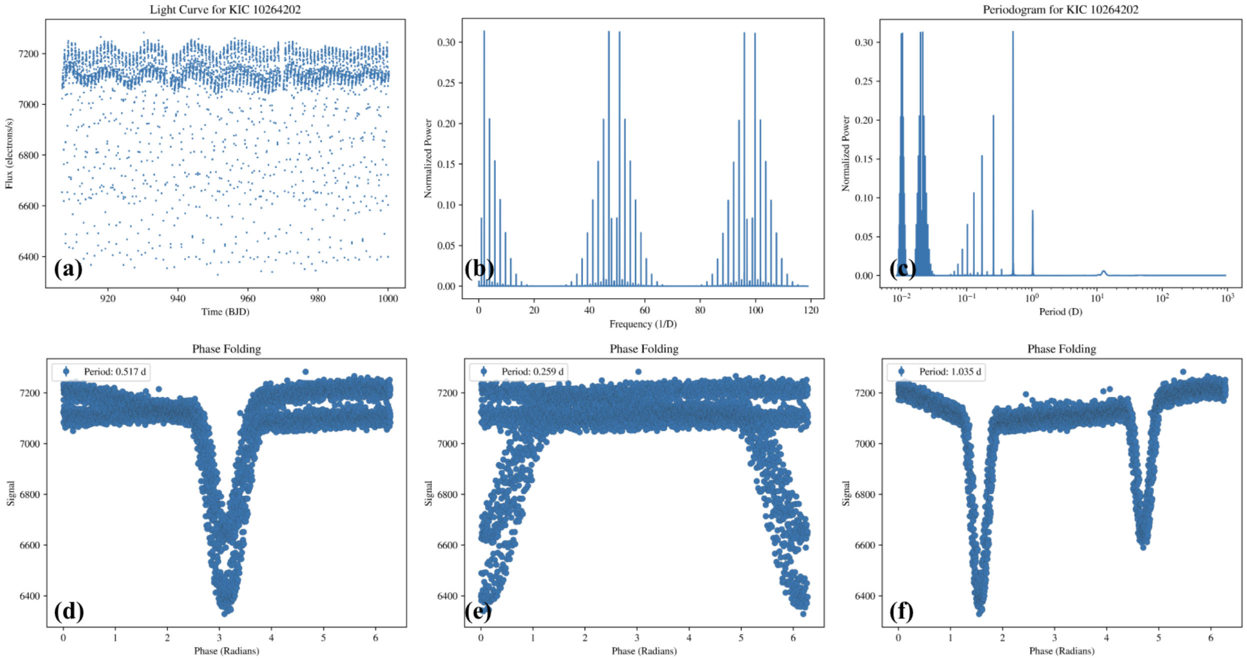

Figure 3: Analysis of KIC 10264202

5 Codes Exploring Subtler Aspects of the Lomb Scargle Periodogram

In this section, we describe the code we wrote to solve the exercises in the second notebook, and results therefrom.

5.1 Floating Mean Adjustment

We removed from the simulated data described in §4.3 (Figure 2) those t and y observations where $y \leq 2$ (using an index mask constructed from `np.where`), leading to a situation of incomplete phase coverage. We calculated the periodogram for the data with and without allowing for a floating mean, as described in §3.4.2 (by setting the `fit_mean` argument in the `astropy` implementation). As expected from the discussion in that section, we saw that the vanilla code was unable to find the peak power at the true period of 5.25 days (Figure 4a), whereas the adjusted code was able to do so (Figure 4b). Moreover, we saw explicitly that without the adjustment, the Lomb-Scargle best fit does not actually match all the data (Figure 4c).

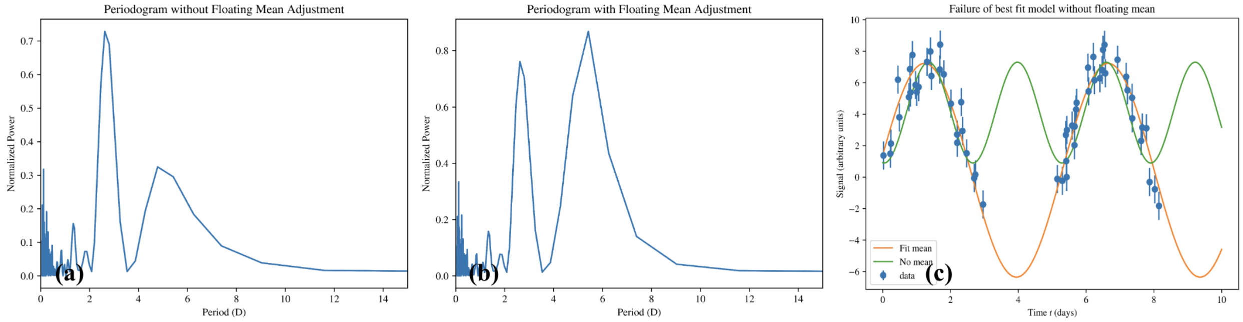

Figure 4: Comparison of periodograms with and without floating mean adjustment

5.2 Effects due to Window Functions

We simulated data according to $y = 12.4 \sin(2\pi t/220)$ plus Gaussian noise with variance 1. We turned this into realistic astronomical data by sampling it every three days (the LSST cadence) over a survey length of ten years, adding jitter by allowing the observation times to vary randomly within ±4 hours on each night of observation, dropping a random 30% of the observations to mimic occlusions due to bad weather, and removing the first 40% of data from each year to account for the source being behind the Sun (Figure 5a). To achieve the jitter, we simply added a random number drawn uniformly from $-1/6$ to $1/6$ (working in days) to each observation time. To drop the random 30% of the observations, we drew an array of random numbers uniformly distributed between 0 and 1 having the same length as our data, and constructed a mask of it by using `np.where` on the condition ≤ 0.7 (with the idea being that about 70 percent of uniform draws between 0 and 1 will be less than 0.7, so such a mask would be about 70% true and 30% false). To drop the first 40% of each year, we kept only those t_i for which $(t_i \bmod 365)/365 > 0.4$ (days into the year as a fraction of the year). We then calculated the periodogram solely due to the observing window by putting $y = 1$ in the `LombScargle` object of the observations, obtaining the periodogram Figure 5b. The key observation is that there are strong power peaks at 3 (and aliases thereof) and 365 days, in the first case due to the cadence and in the second due to annual-scale variability (for example, our dropping the first 40% of data at the beginning of every year due to solar occlusion). In the top panel of Figure 5c, we show the periodogram due to the overall signal (on a limited horizontal scale) atop the periodogram due solely to the window (from Figure 5b). The key point, as we expected from the discussion of the convolution theorem in §3.1 and §3.2, is that the overall periodogram retains power from both the actual signal and the window function—notice the vertical coincidences.

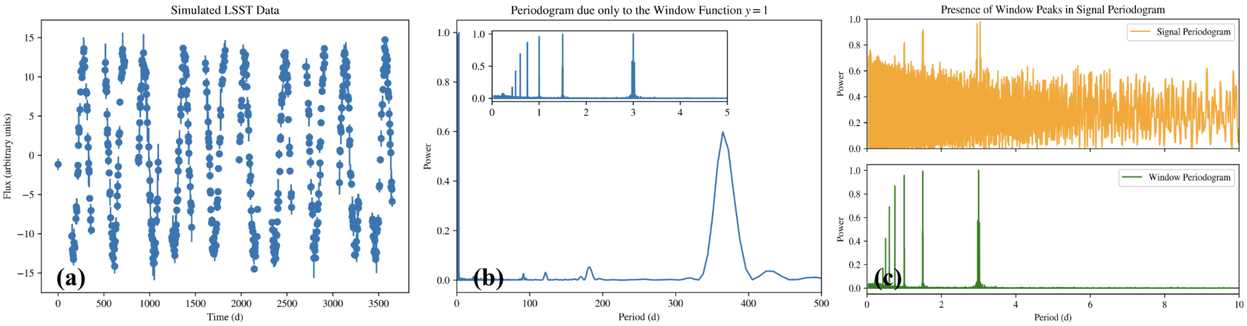

Figure 5: Window effects on realistic LSST data

5.3 Failure Modes due to Aliasing and Pseudo-Aliasing

To enhance the realism of our synthetic data further still, we modified our data-generating function by expanding it beyond a simple perturbed sinusoid to include the first four harmonics of a Fourier series; in particular, we made it so that the $n > 1$ harmonics had random phase offsets drawn randomly from a uniform distribution between 0 and 2π and amplitude drawn randomly from a uniform distribution between 0 and the amplitude of the fundamental component. We then simulated 1000 light curves over two years with the astronomical conditions described in §5.2, drawing the period randomly from $[0.2, 10]$, the amplitude randomly from $[1, 5]$, and the variance of the noise randomly from $[1, 2]$. We ran Lomb-Scargle on each of these curves and stored the underlying period and that according to Lomb-Scargle; these are plotted against each other in Figure 6a. In fact, over repeated simulations, we found that Lomb-Scargle only obtained the correct period $\approx 50\%$ of the time (in other words, only $\approx 50\%$ of the points fell on the main diagonal LS peak = True period in Figure 6a). However, we were able to explain most of the rest of the structure in the plot by straightforward yet elegant considerations of the interplay between the window function and the underlying spectral power.

To understand the first such interaction, we refer back to the bottom panel of Figure 5c. Notice in particular that we have significant power at 3 days due to the cadence, and also at $3/2 = 1.5, 3/3 = 1, 3/4 = 0.75\ldots$ days due to aliasing at $3/n$ days for integer n. In other words, each frequency signature f_0 is partially aliased at $f_0 + n\delta f$ for integers n and

$\delta f = 1/3$ cycle day^{-1}; or equivalently

$$P_{\text{LS}} = \left(\frac{1}{P_{\text{true}}} + \frac{n}{3}\right)^{-1} \tag{14}$$

We overplotted (14) on Figure 6a in Figure 6b for $n = -2, -1, 1, 2$ (taking only those values where $P_{\text{LS}} > 0$, since in this case we were not looking for "negative periods"), capturing much of the off-diagonal structure.

To understand the second such interaction, we refer back to the fact that we expanded our data generator to include higher-order harmonics. In the same way as we had aliasing in (14), we can also have aliasing for the order m harmonic due to three-day cadence, giving

$$P_{\text{LS}} = \left(\frac{m}{P_{\text{true}}} + \frac{n}{3}\right)^{-1} \tag{15}$$

We overplotted (15) on Figure 6a in Figure 6c for $n = -2, -1, 1, 2$ for the $m = 2$ harmonic, again taking only $P_{\text{LS}} > 0$, capturing much of the reminaing off-diagonal structure.

To understand a final such interaction, we refer to the symmetry of Lomb-Scargle: the periodogram is even, so although we have not shown the power at negative frequencies (because this would be unphysical), it is mathematically true that every periodogram peak at f_0 has a mirror image at $-f_0$, and analogously for aliases thereof as well. If the alias of a peak spills over into negative frequencies, which would happen for $n < 0$ in the equations above, the its power is reflected back into the positive frequency range, which we can account for by taking the absolute value of (15):

$$P_{\text{LS}} = \left|\frac{m}{P_{\text{true}}} + \frac{n}{3}\right|^{-1} \tag{16}$$

We overplotted (16) on Figure 6a in Figure 6d for $n = -3, -2, -1$ for the "reflected" $m = 1$ harmonic; importantly, in this case, unlike the preceding ones, we took only $P_{\text{LS}} < 0$. Most of the remaining structure is captured by this final interaction.

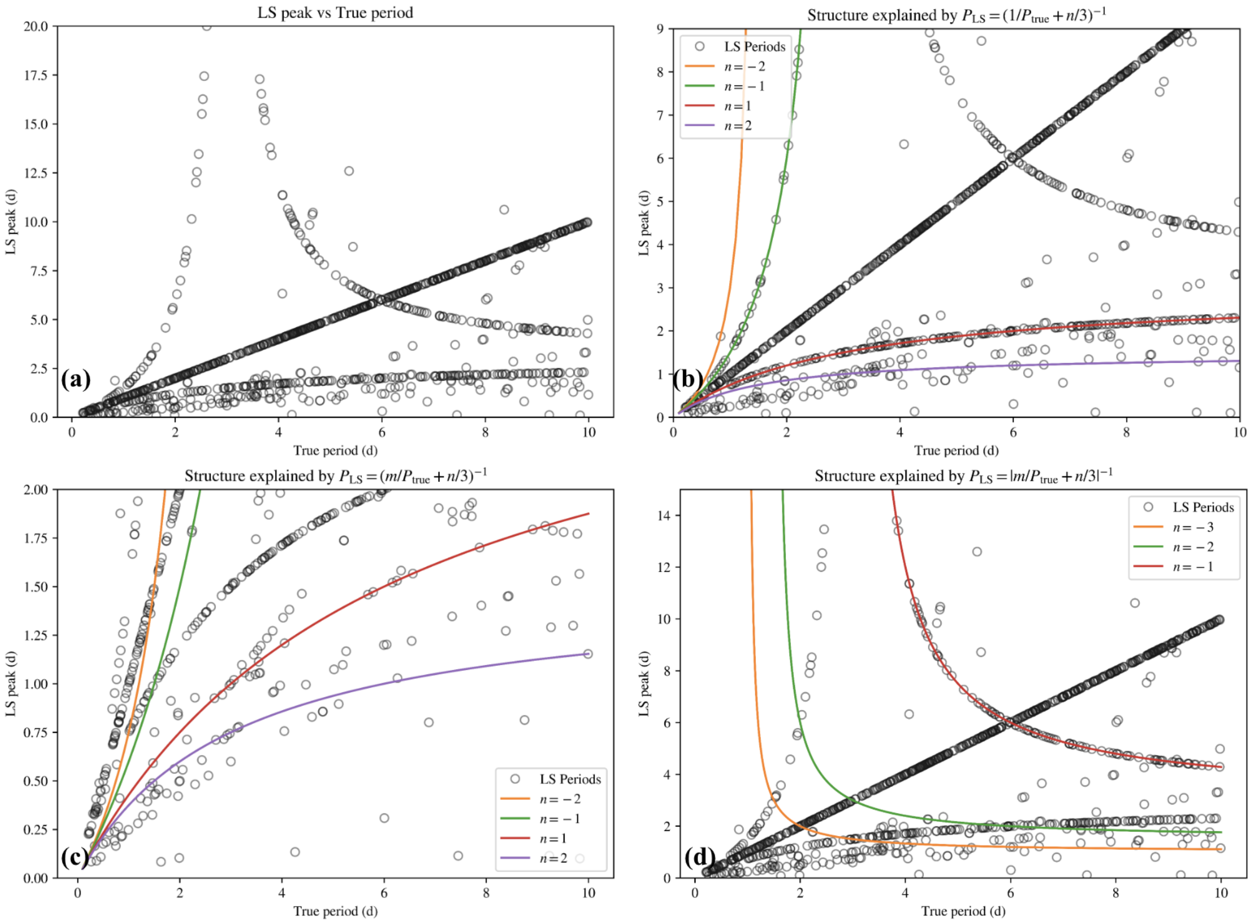

Figure 6: Failure modes due to window-signal-aliasing interaction

6 Conclusion

We began this work by showing that in the case of imposing an evenly-spaced grid of times on an unevenly-sampled dataset, methods involving the FFT on interpolated data have the potential to fail massively, motivating a more flexible approach to spectral analysis. We studied in depth the classical Fourier transform and its reduction to the discrete Fourier transform via multiplication by the Dirac comb, and offered much discussion concerning the convolution theorem and windowing. Moreover, we showed how familiar concepts concerning frequency spacing and the Nyquist limit emerge from the Dirac-comb/convolution perspective. We proceeded to consider these ideas in the case non-uniform spacing, with our most important point being the non-existence of a Nyquist limit for non-uniform data. We then motivated the Lomb-Scargle periodogram as a generalization of the classical periodogram, and showed the elegant connection between the Fourier and least squares perspectives on it. We subsequently explored a number of Lomb-Scargle extensions. We then detailed our approach to generating synthetic periodic data, detailed our `Python` implementation of the Lomb-Scargle algorithm, and showed some results. We further offered discussion on a sensible recipe for choosing a frequency grid when running Lomb-Scargle, and offered a discussion of time-complexity in that context. We then applied our code to the analysis of real astronomical data, studying the transits of an eclipsing binary system documented by the Kepler mission. In our final section, we studied a number of subtleties associated with Lomb-Scargle, including the necessity of the floating mean adjustment, imprinting of window periodicity in the overall periodgram due to the convolution theorem, and failure modes due to aliasing and pesudo-aliasing effects from windowing and harmonicity.

References

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. USA: Cambridge University Press, 2007, ISBN: 0521880688.

[2] A. A. Miller. "Introduction to the Lomb-Scargle Periodogram". (2021), [Online]. Available: https://github.com/LSSTC-DSFP/LSSTC-DSFP-Sessions/blob/main/Sessions/Session13/Day3/IntroductionToTheLombScarglePe ipynb.

[3] A. A. Miller. "Real World Considerations for the Lomb-Scargle Periodogram". (2021), [Online]. Available: https://github.com/LSSTC-DSFP/LSSTC-DSFP-Sessions/blob/main/Sessions/Session13/Day3/RealWorldLombScargle.ipynb.

[4] The Astropy Developers. "Lomb-Scargle Periodograms". (2025), [Online]. Available: https://docs.astropy.org/en/stable/timeseries/lombscargle.html.

[5] O. Hall and G. Barentsen. "Creating periodograms and identifying significant peaks". (2020), [Online]. Available: https://lightkurve.github.io/lightkurve/tutorials/3-science-examples/periodograms-creating-periodograms.html.

[6] J. T. VanderPlas, "Understanding the Lomb-Scargle Periodogram", *The Astrophysical Journal Supplement Series*, vol. 236, no. 1, p. 16, 2018.

Appendix: Nyquist Considerations

As part of our solutions to the first notebook, we wrote some code to explore concepts related to the Nyquist theorem, but since it was outside the primary vein of discussion in §4, we have relegated it to this appendix.

It is worth being explicit about the distinction between the *Nyquist frequency* and the *Nyquist rate*. The Nyquist frequency is a function of *our sampling frequency* f_s, and is $f_s/2$, the maximum frequency that we would be able to detect. The Nyquist rate, on the other hand, is a function of the highest frequency *present in the signal*: if that highest frequency is f_{hi}, we would need to sample above the Nyquist rate $2f_{\text{hi}}$.

In Figure 7a, we sample at a rate of 1 Hz, and so our associated Nyquist frequency is 0.5 Hz. Thus we are faithfully able to reconstruct the underlying signal $y = (2\pi x/2)$, although since this is *right* at the Nyquist frequency, our data could equally well be explained by the trivial signal $y = 0$. In Figure 7b, we again sample at 1 Hz with associated $f_{\text{Ny}} = 0.5$ Hz, but from a signal with frequency $f = 0.7 > f_{\text{Ny}}$. At first glance, this does not appear to be an issue, as it seems that our observations are matched with the signal. If we look at Figure 7c, however, we see that our observations could equally well be explained by $f = 2.7$; this is because for all $f > f_{\text{Ny}}$, f will be aliased with $f \pm 2nf_{\text{Ny}}$ signals for integers n, and $2.7 = 0.7 + 2 \times 1 \times 0.5$; this of course means that it does not make sense to search for frequencies higher than f_{Ny}. (In fact, all such frequencies "wrap around" back into the range $(0, f_{\text{Ny}})$; 0.7 and 2.7 in this case "wrap" to 0.3).

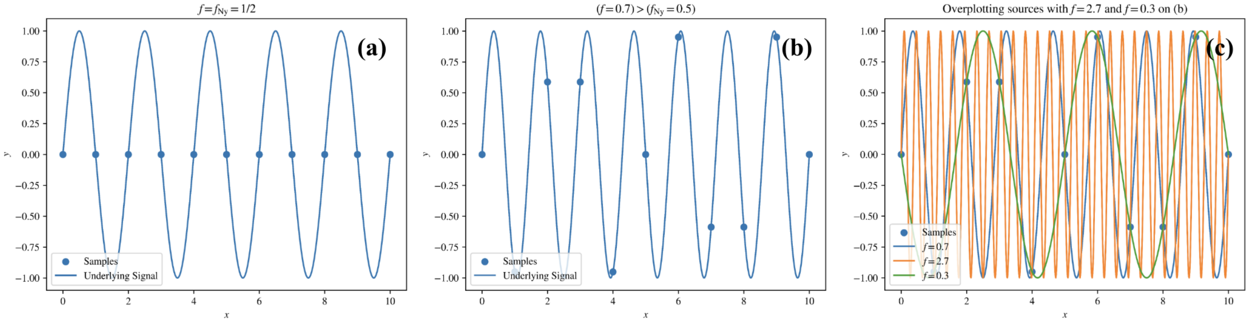

Figure 7: Nyquist considerations