Draft version February 24, 2017
Preprint typeset using LaTeX style emulateapj v. 01/23/15

A FAST TEMPLATE PERIODOGRAM FOR DETECTING NON-SINUSOIDAL FIXED-SHAPE SIGNALS IN IRREGULARLY SAMPLED TIME SERIES
DRAFT VERSION 1

J. Hoffman[1], J. Vanderplas[2], J. Hartman[1], G. Bakos[1]

Draft version February 24, 2017

ABSTRACT

Astrophysical time series often contain periodic signals. The sheer volume of time series data from photometric surveys demands computationally efficient methods for detecting and characterizing such signals. The most efficient algorithms available for this purpose are those that exploit the $\mathcal{O}(N \log N)$ scaling of the Fast Fourier Transform (FFT). However, these methods are not optimal for non-sinusoidal signal shapes. Template fits (or periodic matched filters) optimize sensitivity for *a priori* known signal shapes but at enormous computational cost. Current implementations of template periodograms scale as $\mathcal{O}(N_f N_{\rm obs})$, where N_f is the number of trial frequencies and $N_{\rm obs}$ is the number of lightcurve observations, and they do not gaurantee the best fit at each trial frequency. In this work, we present a non-linear extension of the Lomb-Scargle periodogram which provides a template-fitting algorithm that is both accurate (the exact optimal solutions are obtained except in rare cases) and computationally efficient (scaling as $\mathcal{O}(N_f \log N_f)$). The non-linear optimization of the template fit at each frequency is recast as a polynomial zero-finding problem, where the coefficients of the polynomial can be computed efficiently with FFTs. We show that our method, which uses truncated Fourier series to approximate templates, is twice as fast as existing algorithms for small problems ($N \lesssim 10$ observations) and up to 3 orders of magnitude faster for long base-line time series with $\mathcal{O}(10^4)$ observations. An open-source implementation of the fast template periodogram is available at github.com/PrincetonUniversity/FastTemplatePeriodogram.

1. INTRODUCTION

Astrophysical time series are challenging to analyze. Unlike time series in other domains like economics and finance, astrophysical observations are often irregularly sampled in time with heteroskedastic, non-Gaussian, and time-correlated measurement uncertainties.

Irregular sampling thwarts the straightforward application of many well-known time series tools like the discrete Fourier transform (DFT) and the auto-regressive moving average (ARMA) models. The DFT is a particularly unfortunate loss, since the Fast Fourier Transform (Cooley & Tukey 1965) reduces the $\mathcal{O}(N^2)$ DFT to $\mathcal{O}(N \log N)$, and is a powerful tool for finding periodic signals.

Fortunately, the DFT can be extended to irregularly sampled data via what is sometimes referred to as the classical periodogram (Stoica et al. 2009)

$$P_x(\omega) = \frac{1}{N^2}\left|\sum_{n=0}^{N-1} y_n e^{-i\omega t_n}\right|^2. \tag{1}$$

However, as Stoica et al. (2009) point out, this is not an optimal measure of periodicity. A more robust estimate of the power spectrum is given by the Lomb-Scargle periodogram (Lomb 1976; Scargle 1982; Barning 1963; Vaníček 1971).

ARMA models can also be extended to unevenly sampled data with the CARMA model (Kelly et al. 2014; Zinn et al. 2016), but for the purposes of this paper, we focus solely on tools applicable to the detection of periodic signals in astrophysical data.

The Lomb-Scargle periodogram and its extensions can be expressed in terms of least-squares minimization between the data $\{y_n\}_{n=1}^N$ and a model $\hat{y}$. In the original formulation of the Lomb-Scargle periodogram,

$$\hat{y}_{\rm LS}(t|\theta,\omega) = \theta_0 \cos\omega t + \theta_1 \sin\omega t. \tag{2}$$

This is equivalent to performing a DFT if the data is regularly sampled. The Lomb-Scargle periodogram can be obtained from solving the linear system of equations that arise from the condition that the summed squares of residuals between the data and the optimal model,

$$\chi^2(\theta, S) \equiv \sum_i (y_i - \hat{y}(t_i|\theta))^2, \tag{3}$$

must be a local minimum. This means that

$$\left.\frac{\partial \chi^2}{\partial \theta_i}\right|_{\theta=\theta_{\rm best}} = 0 \quad \forall \theta_i \in \theta. \tag{4}$$

The resulting periodogram can be expressed as

$$P_{\rm LS} = \frac{1}{2\sigma^2}\left(\frac{\left[\sum_{n=1}^N (y_n - \bar{y})\cos\omega t_n\right]^2}{\sum_{n=1}^N \cos^2\omega t_i} + \frac{\left[\sum_{n=1}^N (y_n - \bar{y})\sin\omega t_n\right]^2}{\sum_{n=1}^N \sin^2\omega t_i}\right), \tag{5}$$

jah5@princeton.edu
[1] Department of Astrophysical Sciences, Princeton University, Princeton NJ 08540
[2] eScience Institute, University of Washington, Seattle, WA 98195

where $\bar{y} = \mathbb{E}[y_n]$, the mean of the data, and $\sigma = \mathrm{Var}(y_n)$, the variance of the data.

Heteroskedasticity can be handled by using weighted least squares,

$$\chi^2(\theta, S) \equiv \sum_i \frac{(y_i - \hat{y}(t_i|\theta))^2}{\sigma_i^2}, \tag{6}$$

with weights $w_i = \frac{W}{\sigma_i^2}$, $W \equiv \sum \sigma_i^{-2}$ being a normalization factor to ensure $\sum w_i = 1$, and correlated uncertainties can be accounted for by using the full covariance matrix, $\Sigma_{ij} = \mathrm{Cov}\left((y_i - \bar{y})(y_j - \bar{y})\right)$.

$$\chi^2(\theta, S) \equiv (y_i - \hat{y}(t_i|\theta))^{\mathrm{T}} \Sigma (y_i - \hat{y}(t_i|\theta)). \tag{7}$$

If we assume the covariance matrix is diagonal, the Lomb-Scargle periodogram can be evaluated quickly in one of two popular ways. The first, by Press & Rybicki (1989) involves"extirpolating" irregularly sampled data onto a regularly sampled mesh, and then performing FFTs to evaluate the necessary sums. The second, as pointed out in Leroy (2012), is to use the non-equispaced FFT (NFFT) Keiner et al. (2009) to evaluate the sums; this provides an order of magnitude speedup over the Press & Rybicki (1989) algorithm, and both algorithms scale as $\mathcal{O}(N \log N)$.

There is a growing population of alternative methods for detecting periodic signals in astrophysical data. Some of these methods can reliably outperform the Lomb-Scargle periodogram, especially for non-sinusoidal signal shapes (see Graham et al. (2013) for a recent empirical review). However, a key advantage that the LS periodogram and its extensions have over many alternatives is speed. Virtually all other methods scale as $N \times N_f \sim N^2 \sim N_f^2$, where N is the number of observations and N_f is the number of trial frequencies, while the Lomb-Scargle periodogram scales as $N_f \log N_f \sim N \log N$.

Algorithmic efficiency will become increasingly important as the volume of data produced by astronomical observatories continues to grow larger. The HATNet survey (Bakos et al. 2004), for example, has already made $\mathcal{O}(10^4)$ observations of $\mathcal{O}(10^6 - 10^7)$ stars. The Gaia telescope (Gaia Collaboration et al. 2016) is set to produce $\mathcal{O}(10-100)$ observations of $\mathcal{O}(10^9)$ stars. The Large Synoptic Survey Telescope (LSST; LSST Science Collaboration et al. (2009)) will make $\mathcal{O}(10^2-10^3)$ observations of $\mathcal{O}(10^{10})$ stars during its operation starting in 2023.

This paper develops new extensions to least-squares spectral analysis for arbitrary signal shapes. For non-periodic signals this method is known as matched filter analysis, and can be extended to search for periodic signals by phase folding the data at different trial periods. We refer to the latter technique, the subject of this paper, as template fitting.

Recently, Sesar et al. (2016) found that template fitting significantly improved period and amplitude estimation for RR Lyrae in Pan-STARRS DR1 (Chambers et al. 2016). Since the signal shapes for RR Lyrae in various bandpasses are known *a priori* (see Sesar et al. (2010)), template fitting provides an optimal estimate of amplitude and period, given that the object is indeed an RR Lyrae star well modeled by at least one of the templates. Templates were especially crucial for Pan-STARRS data, since there are typically only 35 observations per source over 5 bands (Hernitschek et al. 2016), not enough to obtain accurate amplitudes empirically by phase-folding. By including domain knowledge (i.e. knowledge of what RR Lyrae lightcurves look like), template fitting allows for accurate inferences of amplitude even for undersampled lightcurves.

However, the improved accuracy comes at substantial computational cost: the template fitting procedure took 30 minutes per CPU per object, and Sesar et al. (2016) were forced to limit the number of fitted lightcurves ($\lesssim 1000$) in order to keep the computational costs to a reasonable level. Several cuts were made before the template fitting step to reduce the more than 1 million Pan-STARRS DR1 objects to a small enough number, and each of these steps removes a small portion of RR Lyrae from the sample. Though this number was reported by Sesar et al. (2016) to be small ($\lesssim 10\%$), it may be possible to further improve the completeness of the final sample by applying template fits to a larger number of objects, which would require either more computational resources, more time, or, ideally, a more efficient template fitting procedure.

The paper is organized as follows. Section 2 poses the problem of template fitting in the language of least squares spectral analysis and derives the fast template periodogram. Section 3 describes a freely available implementation of the new template periodogram. Section 4 summarizes our results, addresses caveats, and discusses possible avenues for improving the efficiency of the current algorithm.

2. DERIVATIONS

We define a template $\mathbf{M}$

$$\mathbf{M} : [0, 1) \to \mathbb{R}, \tag{8}$$

as a mapping between the unit interval and the set of real numbers. We restrict our discussion to sufficiently smooth templates such that $\mathbf{M}$ can be adequately described by a truncated Fourier series

$$\hat{\mathbf{M}}(\omega t|H) = \sum_{n=1}^{H} \left[c_n \cos n\omega t + s_n \sin n\omega t\right] \tag{9}$$

for some $H > 0$. Specifically, we require that

$$\forall t \in (0, 1] : \lim_{H\to\infty} \left|\mathbf{M}(t) - \hat{\mathbf{M}}(t|H)\right| = 0 \tag{10}$$

That the c_n and s_n values are *fixed* (i.e., they define the template) is the crucial difference between the template periodogram and the Multiharmonic Lomb Scargle (or FastChi) (Palmer 2009), where c_n and s_n are *free parameters*.

We now construct a periodogram for this template. The periodogram assumes that an observed time series $S = \{(t_i, y_i, \sigma_i)\}_{i=1}^N$ can be modeled by a scaled, transposed template that repeats with period $2\pi/\omega$, i.e.

$$y_i \approx \hat{y}(\omega t_i|\theta, \mathbf{M}) = \theta_1 \mathbf{M}(\omega t_i - \theta_2) + \theta_3, \tag{11}$$

where $\theta \in \mathbb{R}^3$ is a set of model parameters.

The optimal parameters are the location of a local minimum of the (weighted) sum of squared residuals,

$$\chi^2(\theta, S) \equiv \sum_i w_i (y_i - \hat{y}(\omega t_i|\theta))^2, \tag{12}$$

and thus the following condition must hold for all three model parameters at the optimal solution $\theta = \theta_{\rm opt}$:

$$\left.\frac{\partial \chi^2}{\partial \theta_j}\right|_{\theta=\theta_{\rm opt}} = 0 \quad \forall \theta_j \in \theta. \tag{13}$$

Note that we have implicitly assumed $\chi^2(\theta, S)$ is a C^1 differentiable function of θ, which requires that $\mathbf{M}$ is a C^1 differentiable function. Though this assumption could be violated if we considered a more complete set of templates, (e.g. a box function), our restriction to truncated Fourier series ensures that $\mathbf{M}$ is C^1 differentiable and thus that $\chi^2(\theta, S)$ is C^1 differentiable.

We can derive a system of equations for $\theta_{\rm opt}$ from the condition given in Equation 13. The explicit condition that must be met for each parameter θ_j is simplified below, using

$$\hat{y}_i = \hat{y}(\omega t_i|\theta) \tag{14}$$

and

$$\partial_j \hat{y}_i = \left.\frac{\partial \hat{y}(\omega t|\theta)}{\partial \theta_j}\right|_{t=t_i} \tag{15}$$

for brevity:

$$\begin{aligned} 0 &= \left.\frac{\partial \chi^2}{\partial \theta_j}\right|_{\theta=\theta_{\rm opt}} \\ &= -2 \sum_i w_i \left(y_i - \hat{y}_i\right) \left(\partial_j \hat{y}\right)_i \\ \sum_i w_i y_i (\partial_j \hat{y})_i &= \sum_i w_i \hat{y}_i (\partial_j \hat{y})_i. \end{aligned} \tag{16}$$

The above is a general result that extends to all least squares periodograms. To simplify derivations, we adopt the following notation:

$$\langle X \rangle \equiv \sum_i w_i X_i \tag{17}$$

$$\langle XY \rangle \equiv \sum_i w_i X_i Y_i \tag{18}$$

$$\mathrm{Var}(X) \equiv \left\langle X^2 \right\rangle - \langle X \rangle^2 \tag{19}$$

$$\mathrm{Cov}(X, Y) \equiv \langle XY \rangle - \langle X \rangle \langle Y \rangle \tag{20}$$

In addition, we denote the shifted template $\mathbf{M}(x - \theta_2)$ by $\mathbf{M}_{\theta_2}(x)$.

For the amplitude and offset model parameters (θ_1 and θ_3, respectively), we obtain the following relations from Equation 16

$$\langle y \mathbf{M}_{\theta_2}(\omega t) \rangle = \theta_1 \left\langle \mathbf{M}^2_{\theta_2}(\omega t) \right\rangle + \theta_3 \left\langle \mathbf{M}_{\theta_2}(\omega t) \right\rangle \tag{21}$$

$$\theta_3 = \bar{y} - \theta_1 \left\langle \mathbf{M}_{\theta_2}(\omega t) \right\rangle, \tag{22}$$

where $\bar{y} \equiv \langle y \rangle$. Combining these expressions yields

$$\theta_1 = \frac{\langle (y - \bar{y}) \mathbf{M}_{\theta_2}(\omega t) \rangle}{\mathrm{Var}(\mathbf{M}_{\theta_2}(\omega t))}. \tag{23}$$

For the offset parameter θ_2,

$$\begin{aligned} \frac{\partial \hat{y}}{\partial \theta_2} &= \theta_1 \frac{\partial \mathbf{M}_{\theta_2}}{\partial \theta_2} \\ &= -\theta_1 \partial \mathbf{M}_{\theta_2}, \end{aligned} \tag{24}$$

where

$$\partial \mathbf{M}_{\theta_2}(x) = \sum_n \left[s_n \cos n(x - \theta_2) - c_n \sin n(x - \theta_2) \right]. \tag{25}$$

From Equations 16, 22, and 24 we obtain

$$\theta_1 = \frac{\langle (y - \bar{y}) \partial \mathbf{M}_{\theta_2}(\omega t) \rangle}{\mathrm{Cov}\left(\mathbf{M}_{\theta_2}(\omega t), \partial \mathbf{M}_{\theta_2}(\omega t)\right)}, \tag{26}$$

which, combined with Equation 23, provides a non-linear expression for θ_2:

$$\begin{aligned} &\langle (y - \bar{y}) \partial \mathbf{M}_{\theta_2}(\omega t) \rangle \, \mathrm{Var}(\mathbf{M}_{\theta_2}(\omega t)) \\ &= \langle (y - \bar{y}) \mathbf{M}_{\theta_2}(\omega t) \rangle \, \mathrm{Cov}(\mathbf{M}_{\theta_2}(\omega t), \partial \mathbf{M}_{\theta_2}(\omega t)). \end{aligned} \tag{27}$$

To obtain an explicit expression for Equation 27, we define several quantities,

$$CC_{nm} \equiv \mathrm{Cov}(\cos n\omega t, \cos m\omega t) \tag{28}$$

$$CS_{nm} \equiv \mathrm{Cov}(\cos n\omega t, \sin m\omega t) \tag{29}$$

$$SS_{nm} \equiv \mathrm{Cov}(\sin n\omega t, \sin m\omega t) \tag{30}$$

$$YC_n \equiv \langle (y - \bar{y}) \cos n\omega t \rangle \tag{31}$$

$$YS_n \equiv \langle (y - \bar{y}) \sin n\omega t \rangle, \tag{32}$$

all of which can be evaluated efficiently with NFFTs. We also obtain an expression for the shifted template:

$$\begin{aligned} \mathbf{M}_{\theta_2}(x) &= \sum_n \left[c_n \cos n(x - \theta_2) + s_n \sin n(x - \theta_2) \right] \\ \mathbf{M}_{\theta_2}(x) &= \sum_n \left[(c_n \cos n\theta_2 - s_n \sin n\theta_2) \cos nx \right. \\ &\qquad \left. + (s_n \cos n\theta_2 + c_n \sin n\theta_2) \sin nx \right] \\ \mathbf{M}_{\theta_2}(x) &= \sum_n \left[\left(c_n T_n(u) \mp s_n \sqrt{1 - u^2} U_{n-1}(u) \right) \cos nx \right. \\ &\qquad \left. + \left(s_n T_n(u) \pm c_n \sqrt{1 - u^2} U_{n-1}(u) \right) \sin nx \right] \\ \mathbf{M}_{\theta_2}(x) &= \sum_n \left[A_n(u) \cos nx + B_n(u) \sin nx \right] \end{aligned} \tag{33}$$

where $u \equiv \cos \theta_2$, T_n and U_n are the Chebyshev polynomials of the first and second kind, respectively, and the $\pm$ ambiguity arises out of the two possible signs for $\sin \theta_2$.

The derivatives of first and second order Chebyshev polynomials are known to be

$$\frac{dT_n}{dx} = nU_{n-1}(x) \tag{34}$$
$$\frac{dU_n}{dx} = \frac{(n+1)T_{n+1}(x) - xU_n(x)}{x^2-1}, \tag{35}$$

and this implies that the first derivative of the shifted template is

$$\begin{aligned}\partial\mathbf{M}_{\theta_2}(x) = \sum_n \Bigg[& n\left(c_n U_{n-1}(u) \pm s_n \frac{T_n(u)}{\sqrt{1-u^2}}\right)\cos nx \\ & + n\left(s_n U_{n-1}(u) \mp c_n \frac{T_n(u)}{\sqrt{1-u^2}}\right)\sin nx\Bigg] \\ \partial\mathbf{M}_{\theta_2}(x) = \sum_n & \left[\partial A_n(u)\cos nx + \partial B_n(u)\sin nx\right]\end{aligned} \tag{36}$$

Using the sums provided in Equations 28 – 32, writing A_n and B_n as as shorthand for $A_n(u)$ and $B_n(u)$, and employing Einstein summation notation, we have that

$$\langle (y-\bar{y})\mathbf{M}_{\theta_2}\rangle = A_n Y C^n + B_n Y S^n \tag{37}$$
$$\langle (y-\bar{y})\partial\mathbf{M}_{\theta_2}\rangle = \partial A_n Y C^n + \partial B_n Y S^n \tag{38}$$
$$\begin{aligned}\mathrm{Var}(\mathbf{M}_{\theta_2}^2) = & A_n A_m C C^{nm} \\ & + 2A_n B_m C S^{nm} + B_n B_m S S^{nm}\end{aligned} \tag{39}$$
$$\begin{aligned}\mathrm{Cov}(\mathbf{M}_{\theta_2}, \partial\mathbf{M}_{\theta_2}) = & A_n \partial A_m C C^{nm} \\ & + (A_n \partial B_m + B_n \partial A_m) C S^{nm} \\ & + B_n \partial B_m S S^{nm}\end{aligned} \tag{40}$$

Equation 27 now becomes

$$\begin{aligned}0 = \quad & A_n A_m \partial A_k \left(Y C^n C C^{mk} - Y C^k C C^{mn}\right) \\ & + A_n A_m \partial B_k \left(Y C^n C S^{mk} - Y S^k C C^{mn}\right) \\ & + A_n B_m \partial A_k \left(Y C^n C S^{km} + Y S^m C C^{kn}\right) \\ & + A_n B_m \partial B_k \left(Y C^n S S^{mk} + Y S^m C S^{nk}\right) \\ & + B_n B_m \partial A_k \left(Y S^n C S^{km} - Y C^k S S^{nm}\right) \\ & + B_n B_m \partial B_k \left(Y S^n S S^{mk} - Y S^k S S^{mn}\right).\end{aligned} \tag{41}$$

Each $AA\partial A$, $AA\partial B$, ..., $BB\partial B$ can be expressed as

$$AA\partial A,\ AA\partial B,\ \ldots = p(u) \pm (1-u^2)^{-1/2} q(u), \tag{42}$$

where both $p(u)$ and $q(u)$ are polynomials in u. Therefore, Equation 41 can be expressed as

$$0 = (1-u^2)\hat{p}^2(u) - \hat{q}^2(u) = \hat{P}(u) \tag{43}$$

for some polynomials $\hat{p}, \hat{q}$, and $\hat{P}$.

We have derived an explicit, non-linear system of equations to solve for the parameters θ_1, θ_2, and θ_3. Solving this system of equations requires finding the zeros of a polynomial $\hat{P}(u)$ at the given trial frequency.

2.1. *Negative amplitude solutions*

The model $\hat{y} \propto \theta_1 \mathbf{M}_{\theta_2}$ allows for $\theta_1 < 0$ solutions. In the original formulation of Lomb-Scargle and in linear extensions involving multlple harmonics, negative amplitudes translate to phase differences, since $-\cos x = \cos(x-\pi)$ and $-\sin x = \sin(x-\pi)$.

However, for non-sinusoidal templates, $\mathbf{M}$, negative amplitudes do not generally correspond to a phase difference. For example, a detached eclipsing binary template $\mathbf{M}_{\mathrm{EB}}(x)$ cannot be expressed in terms of a phase-shifted negative eclipsing binary template; i.e. $\mathbf{M}_{\mathrm{EB}} \neq -\mathbf{M}_{\mathrm{EB}}(x-\phi)$ for any $\phi \in [0, 2\pi)$.

Negative amplitude solutions found by the fast template periodogram are usually undesirable, as they may produce false positives for lightcurves that resemble flipped versions of the desired template, and allowing for $\theta_1 < 0$ solutions increases the number of effective free parameters of the model, which lowers the signal to noise, especially for weak signals.

One possible remedy for this problem is to set $P_{\mathrm{FTP}}(\omega) = 0$ if the optimal solution for θ_1 is negative, but this complicates the interpretation of P_{FTP}. Another possible remedy is, for frequencies that have a $\theta_1 < 0$ solution, to search for the optimal parameters while enforcing that $\theta_1 > 0$, e.g. via non-linear optimization, but this likely will eliminate the computational advantage of FTP over existing methods.

Thus, we allow for negative amplitude solutions in the model fit and caution the user to check that the best fit θ_1 is positive. Negative amplitude solutions may only be a problem for

- Weak signals
- Signals that, when flipped, are similar to other astrophysical signals
- Signals dominated by the first harmonic ($c_1^2 + s_1^2 \gg c_n^2 + s_n^2$ for $n > 1$).

2.2. *Extending to multi-band observations*

As shown in VanderPlas & Ivezić (2015), the multiphase periodogram (their $(N_{\mathrm{base}}, N_{\mathrm{band}}) = (0,1)$ periodogram), for any model can be expressed as a linear combination of single-phase periodograms:

$$P^{(0,1)}(\omega) = \frac{\sum_{k=1}^K \chi^2_{0,k} P_k(\omega)}{\sum_{k=1}^K \chi^2_{0,k}} \tag{44}$$

where K denotes the number of bands, $\chi^2_{0,k}$ is the weighted sum of squared residuals between the data in the k-th band and its weighted mean $\langle y\rangle$, and $P_k(\omega)$ is the periodogram value of the k-th band at the trial frequency ω.

With Equation 44, the template periodogram is readily applicable to multi-band time series, which is crucial for experiments like LSST, SDSS, Pan-STARRS, and other current and future photometric surveys.

2.3. *Computational requirements*

For a given number of harmonics H, the task of deriving $\hat{P}$ requires a triple sum over H terms, with each sum

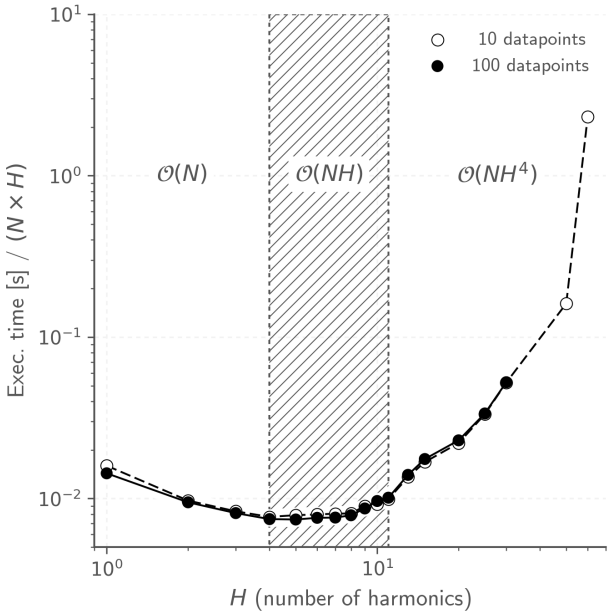

FIG. 1.— Computation time of FTP scaled by NH for different numbers of harmonics. For $H \lesssim 3$, FTP scales sublinearly in H (possibly due to a constant overhead per trial frequency, independent of H). When $3 \lesssim H \lesssim 11$, FTP scales approximately linearly in H, and when $H \gtrsim 11$ FTP approaches the $\mathcal{O}(H^4)$ scaling limit.

requiring $\mathcal{O}(n_{\hat{P}})$ operations, where $n_{\hat{P}}$ is the order of $\hat{P}$. The order of $\hat{P}$ can be shown to be

$$6H - \mathrm{gcf}\left((1-u^2) \star \hat{p}^2, \hat{q}^2\right) \propto H, \tag{45}$$

where $\mathrm{gcf}(p_1, p_2)$ denotes the greatest common polynomial factor between polynomials p_1 and p_2. Computing the coefficients of $\hat{P}$ therefore scales as $\mathcal{O}(H^4)$ at each trial frequency.

The computational complexity of polynomial root finding is algorithm dependent. If we choose to perform singular value decomposition of the polynomial companion matrix[3], the root finding step scales as $\mathcal{O}(n_{\hat{P}}^3) = \mathcal{O}(H^3)$. The polynomial root-finding step should be asymptotically faster (for large H) than the computation of the polynomial coefficients.

When considering N_f trial frequencies, the polynomial computation and root-finding step scales as $\mathcal{O}(H^4 N_f)$. The computation of the sums (Equations 28 – 32) scales as $\mathcal{O}(HN_f \log HN_f)$. Therefore, the entire template periodogram scales as

$$\mathcal{O}(HN_f \log HN_f + H^4 N_f). \tag{46}$$

For a fixed number of harmonics H, the template periodogram scales as $\mathcal{O}(N_f \log N_f)$. However, for a constant number of trial frequencies N_f, the template algorithm scales as $\mathcal{O}(H^4)$, and computational resources alone limit H to reasonably small numbers $H \lesssim 15$ (see Figure 1).

3. IMPLEMENTATION

An open-source implementation of the template periodogram in Python is available.[4] Computing $\hat{P}(u)$ is done using the `numpy.polynomial` module (Jones et al. 2001–). The `pynfft` Python module, [5] which provides a Python wrapper for the NFFT library (Keiner et al. 2009), is used to compute the necessary sums for a particular time series.

No explicit parallelism is used anywhere in the current implementation, however the NFFT library optionally exploits OpenMP if compiled to do so (requires specifying the `--enable-openmp` flag when running `configure`) and certain linear algebra operations in `Scipy` may use OpenMP via calls to BLAS libraries that have OpenMP enabled.

All timing tests were run on a quad-core 2.6 GHz Intel Core i7 MacBook Pro laptop (mid-2012 model) with 8GB of 1600 MHz DDR3 memory. The NFFT library (version 3.2.4) was compiled with `--enable-openmp`, and the `Scipy` stack (version 0.18.1) was compiled with multi-threaded MKL libraries. However, the slowest portion of the algorithm, computing the polynomial coefficients, uses the `numpy.einsum` function which is not multi-threaded.

3.1. *Comparison with `gatspy`*

The `gatspy` (General tools for Astronomical Time Series in Python; VanderPlas (2016); VanderPlas & Ivezić (2015)) library provides template fitting routines, which rely on non-linear optimization at each trial frequency to pick the optimal parameters θ. We compare the accuracy and speed of the fast template periodogram with the template fitting procedure provided by the `gatspy` library.

Periodograms computed in Figures 2, 3, and 4 used simulated data. The simulated data has uniformly random observation times, with gaussian-random, homoskedastic, uncorrelated uncertainties. An eclipsing binary template, generated by fitting an eclipsing binary in the HATNet dataset (**put in HATID for template**) with a 10-harmonic truncated Fourier series.

3.1.1. *Accuracy*

Gatspy uses non-linear optimization at each trial frequency to solve for the best-fit amplitude, phase, and offset of the template. For weak signals or signals folded at the incorrect trial period, there may be a large number of local χ^2 minima in parameter space, and thus the optimization algorithm may have trouble finding the global minimum. The FTP, on the other hand, solves for the optimal parameters directly, and thus is able to recover optimal solutions even when the signal is weak or not present.

Figure 3 illustrates the accuracy improvement with FTP. For large $P_{\mathrm{FTP}}(\omega)$, Gatspy and FTP agree well, while for $P_{\mathrm{FTP}} \lesssim 0.25$, FTP consistently finds better template fits than Gatspy. For many trial frequencies, Gatspy returns a periodogram value of $P_{\mathrm{gatspy}} = 0$, indicating no improvement over a constant fit, while FTP is able to find superior solutions at many of these frequencies.

[3] The `numpy.polynomial.polyroots` function uses this method.

[4] https://github.com/PrincetonUniversity/FastTemplatePeriodogram

[5] https://pypi.python.org/pypi/pyNFFT

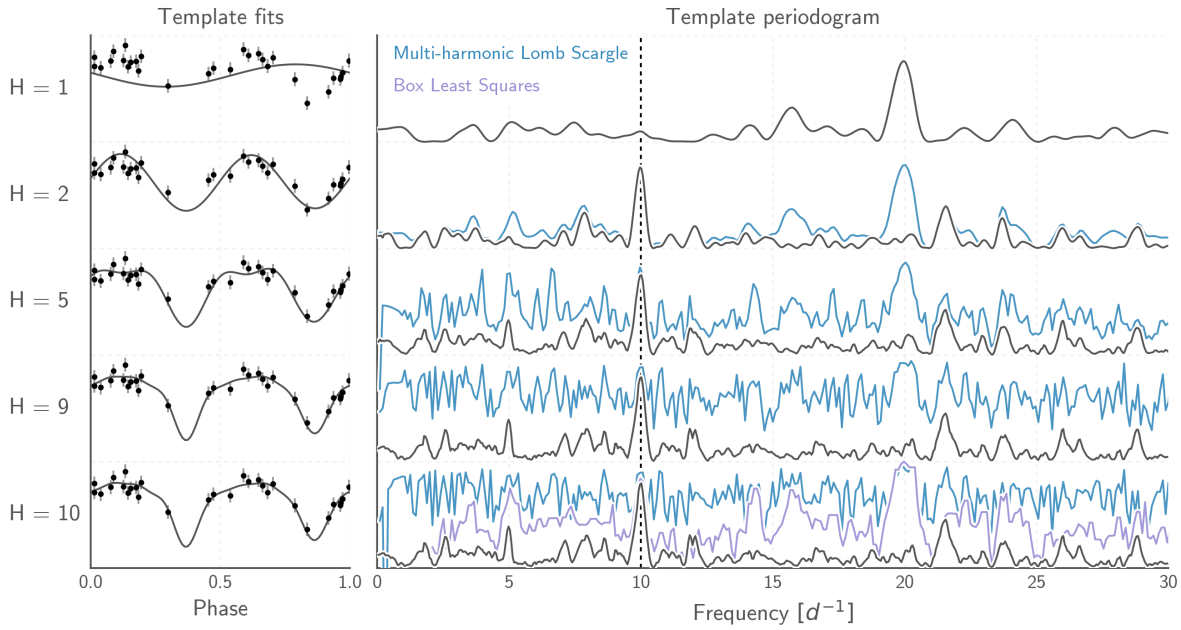

FIG. 2.— Template periodograms performed on a simulated eclipsing binary lightcurve (shown phase-folded in the left-hand plots). The top-most plot uses only one harmonic, equivalent to a Lomb-Scargle periodogram. Subsequent plots use an increasing number of harmonics, which produces a narrower and higher peak height around the correct frequency. For comparison, the multi-harmonic extension to Lomb-Scargle is plotted in blue, using the same number of harmonics as the FTP. The Box Least-Squares (Kovács et al. 2002) periodogram is shown in the final plot.

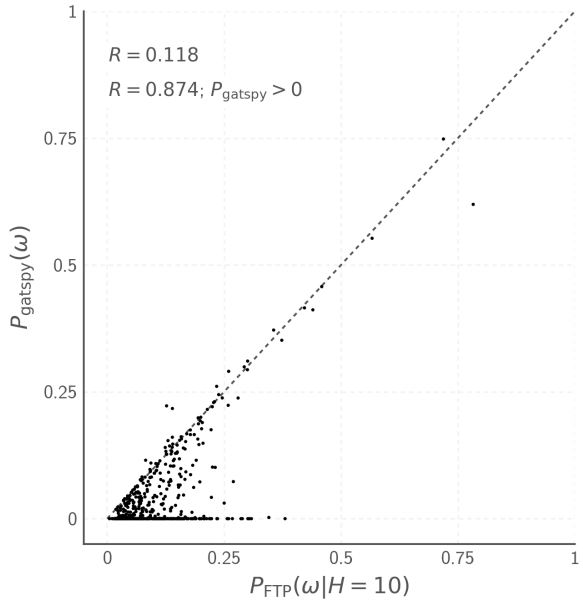

FIG. 3.— Comparing the gatspy template periodogram with the FTP using the same simulated data as shown in Figure 2. FTP consistently finds more optimal template fits than Gatspy. Gatspy numerically minimizes the χ^2 at each frequency, and sometimes gets caught in a local minimum. The FTP solves for the optimal fit parameters directly, and therefore is able to achieve greater accuracy than Gatspy.

Figure 4 compares FTP results obtained using the full template ($H = 10$) with those obtained using smaller numbers of harmonics. The left-most plot compares the $H = 1$ case (weighted Lomb-Scargle), which, as also demonstrated in Figure 2, illustrates the advantage of the template periodogram for known, non-sinusoidal signal shapes.

3.1.2. *Computation time*

FTP scales asymptotically as $\mathcal{O}(N_f H \log N_f H)$, however the computational time is usually dominated by computing polynomial coefficients and zero-finding, which scales as $\mathcal{O}(N_f H^4)$, even for very large numbers of frequencies ($N_f > 10^7$).

Figure 5 shows that FTP achieves a factor of 3 speedup for even the smallest test case (15 datapoints), while for larger cases ($N \sim 10^4$) FTP offers 2-3 orders of magnitude speed improvement over Gatspy.

4. DISCUSSION

Template fitting is a powerful technique for accurately recovering the period and amplitude of objects with *a priori* known lightcurve shapes. It has been used in the literature by, e.g. Sesar et al. (2016, 2010), to analyze RR Lyrae in the SDSS and PS1 datasets, where it has been shown to produce purer samples of RR Lyrae at a given completeness. The computational cost of current template fitting algorithms, however, limits their application to larger datasets or with a larger number of templates.

We have presented a novel template fitting algorithm that extends the Lomb-Scargle periodogram (Lomb 1976; Scargle 1982; Barning 1963; Vaníček 1971) to handle non-sinusoidal signals that can be expressed in terms of a truncated Fourier series with a reasonably small number of harmonics ($H \lesssim 10$).

The fast template periodogram (FTP) asymptotically scales as $\mathcal{O}(N_f H \log N_f H)$, while previous template fit-

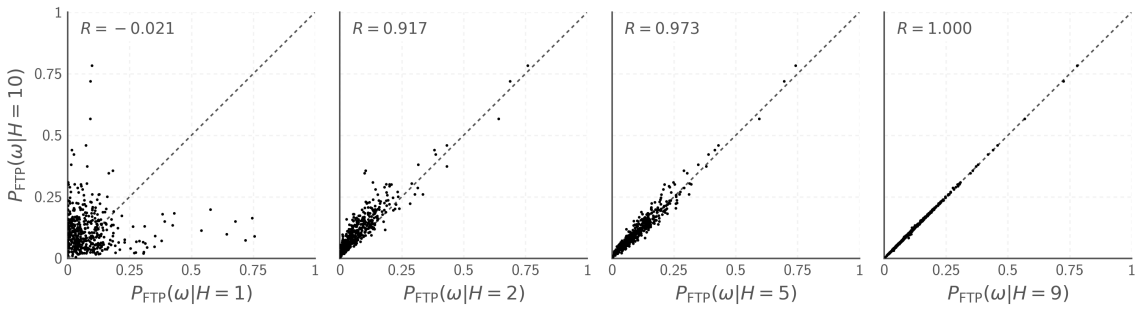

FIG. 4.— Comparing the template periodogram calculated with $H = 10$ harmonics to the template periodogram using a smaller number of harmonics $H < 10$. The template and data used to perform the periodogram calculations are the same as those shown in Figure 2.

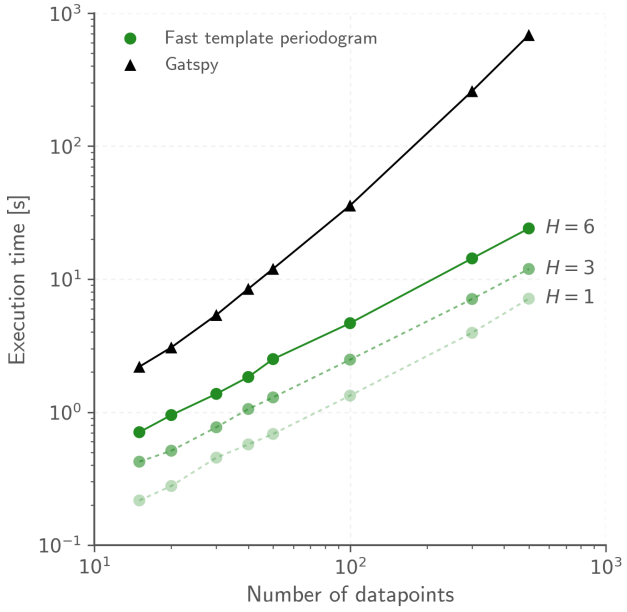

FIG. 5.— Computation time of FTP compared with Gatspy. For a 6-harmonic template and 15 observations, the FTP is three times faster than Gatspy. FTP scales roughly as $\mathcal{O}(N)$ (or, for large enough values of N, $\mathcal{O}(N \log N)$), while Gatspy scales as N^2, thus the speedup of FTP over Gatspy scales linearly in N. For large time series of $N \approx 10,000$, FTP offers over three orders of magnitude speedup over Gatspy.

Survey	N_{LC}	N_{obs}	Refs.
CoRoT	1.5×10^5	53,000	
Kepler	1.7×10^5	65,000	
HATNet	5.6×10^6	10,000	
Gaia	10^9	70	
SuperWASP	3.2×10^7	13,870	
OGLE-IV	10^9	5000	
LSST	3.7×10^{10}	825	

TABLE 1
SURVEY PARAMETERS USED FOR FIGURE 6. **Add references (if we decide to keep this figure)**

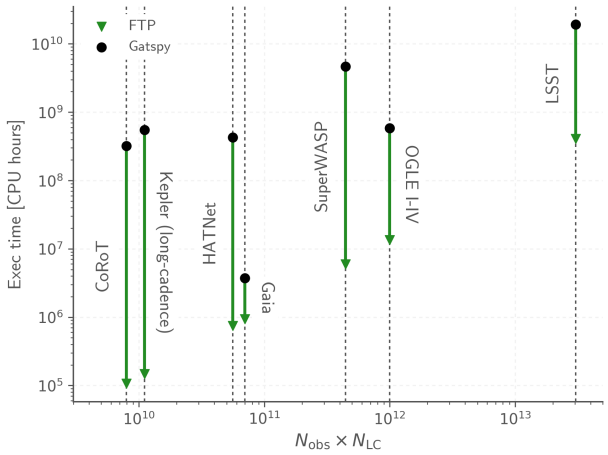

FIG. 6.— Computational resources needed for performing a single template periodogram ($H = 6$) on an entire survey dataset. Our python implementation of FTP improves computational efficiency of template searches by orders of magnitude in most cases, and in one case by over three orders of magnitude (CoRoT). Parameters for N_{obs} and N_{LC} were estimated from publicly available information (see text).

ting algorithms such as the one used in the `gatspy` library (VanderPlas 2016), scale as $\mathcal{O}(N_{obs}N_f \sim N_f^2)$. However, the FTP effectively scales as $\mathcal{O}(N_f H^4)$, since the time needed to compute polynomial coefficients and perform zero-finding dominates the computational time for most practical cases ($N_{obs} \sim N_f \lesssim 10^7$). This effectively restricts the space of templates to those that are sufficiently smooth to be explained by a small number of Fourier terms.

FTP also improves the accuracy of previous template fitting algorithms, which rely on non-linear optimization at each trial frequency to minimize the χ^2 of the template fit. The FTP routinely finds superior fits, especially when the signal is weak.

An open-source Python implementation of the FTP is available at GitHub.[6] The current implementation could likely be improved by:

1. Improving the speed of the polynomial coefficient calculations and the zero-finding steps.

2. Porting FTP to C/C++ and using CUDA to exploit GPU parallelism.

[6] https://github.com/PrincetonUniversity/FastTemplatePeriodogram

As pointed out in VanderPlas & Ivezić (2015), current template fitting procedures are too slow to be practical for LSST-sized time-domain surveys. We attempt to quantify the improvement in computational efficiency for several important time-domain surveys, using estimated survey values for $N_{\rm obs}$, the number of observations per object, and $N_{\rm LC}$, the number of objects with lightcurves in the survey.

Figure 6 shows estimated computation time for a single template periodogram performed on the entirety of a given survey. For all surveys, the FTP improves computational efficiency in one case over three orders of magnitude, but typically between 2-3 orders of magnitude. Improving the existing implementation, and porting to C/C++ and CUDA, should further improve these numbers.

Template fitting remains prohibitvely slow for practical applications to large time-domain surveys, but this work presents a mathematical shortcut that could eventually make template fitting a fast and valuable tool.

(acknowledge GRANTS.)

REFERENCES

Bakos, G., Noyes, R. W., Kovács, G., et al. 2004, PASP, 116, 266
Barning, F. J. M. 1963, Bull. Astron. Inst. Netherlands, 17, 22
Chambers, K. C., Magnier, E. A., Metcalfe, N., et al. 2016, ArXiv e-prints, arXiv:1612.05560
Cooley, J. W., & Tukey, J. W. 1965, Math. Comput., 19, 297
Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., et al. 2016, A&A, 595, A1
Graham, M. J., Drake, A. J., Djorgovski, S. G., et al. 2013, MNRAS, 434, 3423
Hernitschek, N., Schlafly, E. F., Sesar, B., et al. 2016, ApJ, 817, 73
Jones, E., Oliphant, T., Peterson, P., et al. 2001–, SciPy: Open source scientific tools for Python, [Online; accessed 2017-01-19]
Keiner, J., Kunis, S., & Potts, D. 2009, ACM Trans. Math. Softw., 36, 19:1
Kelly, B. C., Becker, A. C., Sobolewska, M., Siemiginowska, A., & Uttley, P. 2014, ApJ, 788, 33
Kovács, G., Zucker, S., & Mazeh, T. 2002, A&A, 391, 369
Leroy, B. 2012, A&A, 545, A50
Lomb, N. R. 1976, Ap&SS, 39, 447
LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, ArXiv e-prints, arXiv:0912.0201
Palmer, D. M. 2009, ApJ, 695, 496
Press, W. H., & Rybicki, G. B. 1989, ApJ, 338, 277
Scargle, J. D. 1982, ApJ, 263, 835
Sesar, B., Ivezić, Ž., Grammer, S. H., et al. 2010, ApJ, 708, 717
Sesar, B., Hernitschek, N., Mitrović, S., et al. 2016, ArXiv e-prints, arXiv:1611.08596
Stoica, P., Li, J., & He, H. 2009, IEEE Transactions on Signal Processing, 57, 843
VanderPlas, J. 2016, gatspy: General tools for Astronomical Time Series in Python, Astrophysics Source Code Library, ascl:1610.007
VanderPlas, J. T., & Ivezić, Ž. 2015, ApJ, 812, 18
Vaníček, P. 1971, Ap&SS, 12, 10
Zinn, J. C., Kochanek, C. S., Kozłowski, S., et al. 2016, ArXiv e-prints, arXiv:1612.04834