

FAST AND SCALABLE GAUSSIAN PROCESS MODELING WITH APPLICATIONS TO ASTRONOMICAL TIME SERIES

DANIEL FOREMAN-MACKEY^{1,2}, ERIC AGOL², AND RUTH ANGUS^{3,4}

¹Sagan Fellow

²Astronomy Department, University of Washington, Seattle, WA

³Simons Fellow

⁴Department of Astronomy, Columbia University, New York, NY

ABSTRACT

We present a scalable method for Gaussian Process regression in one dimension with a specific emphasis on large astronomical time-series data sets. This method can be applied to any Gaussian Process model where the spectral density can be expressed as any general mixture of damped sinusoid functions.

1. INTRODUCTION

Gaussian Processes (GPs; Rasmussen & Williams 2006) are popular stochastic models for time-series analysis. For GP modeling, a functional form is chosen to describe the autocovariance of the data and the parameters of this function are fit for or marginalized. In the astrophysical literature, GPs have been used to model stochastic variability in light curves of stars (CITE), active galactic nuclei (CITE), and X-ray binaries (CITE). They have also been used as models for the cosmic microwave background (CITE), correlated instrumental noise (CITE), spectroscopic calibration (CITE) and residuals caused by model inconsistencies (CITE + better words). While these models are widely applicable, their use has been limited, in practice, by the computational cost and scaling. In general, the cost of computing a GP likelihood scales as the third power of the number of data points $\mathcal{O}(N^3)$ and in the current era of large time-domain surveys – with $\sim 10^{4-9}$ targets with $\sim 10^{3-5}$ observations each — this cost is prohibitive.

In this paper, we present a class of GP models that enable likelihood calculations that scale linearly with the number of data points $\mathcal{O}(N)$ for one dimensional data sets. This method is a generalization of a method developed by Ambikasaran (2015) that was, in turn, built on intuition from a twenty year old paper (Rybicki & Press 1995). For this method to be applicable, the data must be one-dimensional and the covariance function must have a specific form. However, there is no further constraint on the data or the model. In particular, the measurements don't need to be evenly spaced and the uncertainties can be heteroscedastic. This method is especially appealing

compared to other similar methods – we will return to these below – because it is exact, flexible, robust, simple, and fast.

In the following pages, we will motivate the general problem of GP regression, describe the previously published scalable method (Rybicki & Press 1995; Ambikasaran 2015) and our generalization, and demonstrate the model’s application on various real and simulated data sets. Alongside this paper, we have released efficient and well-tested implementations of this method written in C++, Python, and Julia. These implementations are available online at GitHub <https://github.com/dfm/celerite> and Zenodo [DFM: add zenodo archive](#).

2. GAUSSIAN PROCESSES

Gaussian Processes (GPs; Rasmussen & Williams 2006) are a class of stochastic models consisting of a mean function $\mu_{\boldsymbol{\theta}}(\mathbf{x})$ and a covariance or “kernel” function $k_{\boldsymbol{\alpha}}(\mathbf{x}_i, \mathbf{x}_j)$ parameterized by the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ respectively. Under this model, the log-likelihood of observing a dataset

$$\mathbf{y} = \begin{pmatrix} y_1 & \cdots & y_N \end{pmatrix}^T \quad (1)$$

at coordinates

$$X = \begin{pmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_N \end{pmatrix}^T \quad (2)$$

is

$$\ln p(\mathbf{y} | X, \boldsymbol{\theta}, \boldsymbol{\alpha}) = -\frac{1}{2} \mathbf{r}_{\boldsymbol{\theta}}^T K_{\boldsymbol{\alpha}}^{-1} \mathbf{r}_{\boldsymbol{\theta}} - \frac{1}{2} \ln \det K_{\boldsymbol{\alpha}} - \frac{N}{2} \ln(2\pi) \quad (3)$$

where

$$\mathbf{r}_{\boldsymbol{\theta}} = \begin{pmatrix} y_1 - \mu_{\boldsymbol{\theta}}(\mathbf{x}_1) & \cdots & y_N - \mu_{\boldsymbol{\theta}}(\mathbf{x}_N) \end{pmatrix}^T \quad (4)$$

is the vector of residuals and the elements of the covariance matrix K are given by $[K_{\boldsymbol{\alpha}}]_{nm} = k_{\boldsymbol{\alpha}}(\mathbf{x}_n, \mathbf{x}_m)$. The maximum likelihood values for the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ for a given dataset (\mathbf{y}, X) can be found by maximizing Equation (3) with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ using a non-linear optimization routine [DFM: examples and CITE](#). Similarly, probabilistic constraints on $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ can be obtained by multiplying the likelihood by a prior $p(\boldsymbol{\theta}, \boldsymbol{\alpha})$ and using a Markov Chain Monte Carlo (MCMC; [DFM: CITE](#)) algorithm to sample from the posterior probability density.

The application of GP models is generally limited to small datasets because the computational cost of computing the inverse and determinant of the matrix $K_{\boldsymbol{\alpha}}$ scales as the cube of the number of data points N , $\mathcal{O}(N^3)$. This means that for large datasets, every evaluation of the likelihood will quickly become computationally intractable. In this case, standard non-linear optimization or MCMC will no longer be practical inference methods.

In the following Section, we present a method of substantially improving this scaling in many circumstances. We call our method and its implementations **celerite**.¹ The **celerite** method requires using a specific model for the covariance $k_{\alpha}(\mathbf{x}_n, \mathbf{x}_m)$ and it has several limitations but, in subsequent sections, we demonstrate that it can be used to increase the computational efficiency of many astronomical data analysis problems. The main limitation of this method is that it can only be applied to one-dimensional datasets. When we say “one-dimensional” here, it means that the *input coordinates* \mathbf{x}_n are scalar, $\mathbf{x}_n \equiv t_n$.² Furthermore, the covariance function for the **celerite** method is “stationary”. This means that the function $k_{\alpha}(t_n, t_m)$ is only a function of $\tau_{nm} \equiv |t_n - t_m|$.

3. THE CELERITE MODEL

To scale GP models to larger datasets, Rybicki & Press (1995) presented a method of computing the first term in Equation (3) in $\mathcal{O}(N)$ operations when the covariance function is given by

$$k_{\alpha}(\tau_{nm}) = \sigma_n^2 \delta_{nm} + a \exp(-c \tau_{nm}) \quad (5)$$

where $\{\sigma_n^2\}_{n=1}^N$ are the measurement uncertainties, δ_{nm} is the Kronecker delta, and $\alpha = (a, c)$. The intuition behind this method is that, for this choice of k_{α} , the inverse of K_{α} is tridiagonal and can be computed with a small number of operations for each data point. Subsequently, Ambikasaran (2015) generalized this method to arbitrary mixtures of exponentials

$$k_{\alpha}(\tau_{nm}) = \sigma_n^2 \delta_{nm} + \sum_{j=1}^J a_j \exp(-c_j \tau_{nm}) \quad (6)$$

In this case, the inverse will be dense but Equation (3) can still be evaluated in $\mathcal{O}(N)$ operations where J is the number of components in the mixture and N is still the number of data points.

It turns out that this kernel function can be made even more general by introducing complex parameters $a_j \rightarrow a_j \pm i b_j$ and $c_j \rightarrow c_j \pm i d_j$. In this case, the covariance function becomes

$$k_{\alpha}(\tau_{nm}) = \sigma_n^2 \delta_{nm} + \sum_{j=1}^J \left[\frac{1}{2} (a_j + i b_j) \exp(-(c_j + i d_j) \tau_{nm}) + \frac{1}{2} (a_j - i b_j) \exp(-(c_j - i d_j) \tau_{nm}) \right] \quad (7)$$

¹ The name **celerite** comes from the French word *célérité* meaning the speed of light in a vacuum.

² We are using t as the input coordinate because one-dimensional GPs are often applied to time series data but this isn’t a real restriction and the **celerite** method can be applied to *any* one-dimensional dataset.

and, for this function, Equation (3) can still be solved with $\mathcal{O}(N)$ operations. The details of this method and a few implementation considerations are discussed in the following Section but we will first discuss some properties of this covariance function.

By rewriting the exponentials in Equation (7) as sums of sine and cosine functions, we can see the autocorrelation structure is defined by a mixture of quasiperiodic oscillators

$$k_{\alpha}(\tau_{nm}) = \sigma_n^2 \delta_{nm} + \sum_{j=1}^J [a_j \exp(-c_j \tau_{nm}) \cos(d_j \tau_{nm}) + b_j \exp(-c_j \tau_{nm}) \sin(d_j \tau_{nm})] \quad . \quad (8)$$

For clarity, we will refer to the argument within the sum as a “celerite term” for the remainder of this paper. The Fourier transform³ of this covariance function is the power spectral density (PSD) of the process and it is given by

$$S(\omega) = \sum_{j=1}^J \sqrt{\frac{2}{\pi}} \frac{(a_j c_j + b_j d_j) (c_j^2 + d_j^2) + (a_j c_j - b_j d_j) \omega^2}{\omega^4 + 2(c_j^2 - d_j^2) \omega^2 + (c_j^2 + d_j^2)^2} \quad . \quad (9)$$

The physical interpretation of this model isn’t immediately obvious and we will return to a more general discussion of the physical intuition in a moment but we can start with a discussion of some useful special cases of this model.

If we set the imaginary amplitude b_j for some component j to zero, that term of Equation (8) becomes

$$k_j(\tau_{nm}) = a_j \exp(-c_j \tau_{nm}) \cos(d_j \tau_{nm}) \quad (10)$$

and the PSD for the this component is

$$S_j(\omega) = \frac{1}{\sqrt{2\pi}} \frac{a_j}{c_j} \left[\frac{1}{1 + \left(\frac{\omega - d_j}{c_j}\right)^2} + \frac{1}{1 + \left(\frac{\omega + d_j}{c_j}\right)^2} \right] \quad . \quad (11)$$

This PSD is the sum of two Lorentzian or Cauchy distributions with width c_j centered on $\omega = \pm d_j$. This model can be interpreted intuitively as a quasiperiodic oscillator with amplitude $A_j = a_j$, quality factor $Q_j = d_j (2 c_j)^{-1}$, and period $P_j = 2 \pi d_j^{-1}$.

Similarly, setting both b_j and d_j to zero, we get a Ornstein–Uhlenbeck process

$$k_j(\tau_{nm}) = a_j \exp(-c_j \tau_{nm}) \quad (12)$$

with the PSD

$$S_j(\omega) = \sqrt{\frac{2}{\pi}} \frac{a_j}{c_j} \frac{1}{1 + \left(\frac{\omega}{c_j}\right)^2} \quad . \quad (13)$$

³ Here and throughout we have defined the Fourier transform of the function $f(t)$ as $F(\omega) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt$.

It's worth noting that the product of two terms of the form found inside the sum in Equation (8) can also be re-written as a sum with updated parameters

$$k_j(\tau) k_k(\tau) = e^{-\tilde{c}\tau} [a_+ \cos(d_+ \tau) + b_+ \sin(d_+ \tau) + a_- \cos(d_- \tau) + b_- \sin(d_- \tau)] \quad (14)$$

where

$$\tilde{a}_\pm = \frac{1}{2} (a_j a_k \pm b_j b_k) \quad (15)$$

$$\tilde{b}_\pm = \frac{1}{2} (b_j a_k \mp a_j b_k) \quad (16)$$

$$\tilde{c} = c_j + c_k \quad (17)$$

$$\tilde{d}_\pm = d_j \mp d_k \quad (18)$$

4. IMPLEMENTATION & PERFORMANCE

Rybicki & Press (1995) demonstrated that the inverse of a matrix K where the elements are given by

$$K_{nm} = k(\tau_{nm}) \quad (19)$$

for $k(\tau_{nm})$ given by equation Equation (5) could be computed efficiently by taking advantage of the structure of this covariance function and Ambikasaran (2015) generalized this computation to apply to the full mixture of J terms in Equation (7) and derived an equally efficient method for computing the determinant of K .

4.1. An example

To provide some insight for this method, we will start by working through a simple example. In this case, we'll assume that we have three data points $\{y_1, y_2, y_3\}$ observed at times $\{t_1, t_2, t_3\}$ with measurement variances $\{\sigma_1^2, \sigma_2^2, \sigma_3^2\}$ and we would like to compute the likelihood of these data under a GP model with the covariance function

$$k(\tau_{nm}) = \sigma_n^2 \delta_{nm} + a \exp(-c \tau_{nm}) \quad (20)$$

This function is identical to Equation (7) with $J = 1$, $b = 0$, and $d = 0$. To demonstrate this method, we will write out the full system of equations that we must solve to apply the inverse of K in order to compute the first term of Equation (3). In matrix notation, this can be written as

$$K \mathbf{z} = \mathbf{y}, \quad (21)$$

$$\begin{pmatrix} a + \sigma_1^2 & a e^{-c \tau_{2,1}} & a e^{-c \tau_{3,1}} \\ a e^{-c \tau_{2,1}} & a + \sigma_2^2 & a e^{-c \tau_{3,2}} \\ a e^{-c \tau_{3,1}} & a e^{-c \tau_{3,2}} & a + \sigma_3^2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \quad (22)$$

where our goal is to solve for the unknown vector \mathbf{z} for a given matrix K and vector \mathbf{y} . In this equation, we have assumed that the mean function is zero but a non-zero mean

could be included by replacing \mathbf{y} by \mathbf{r}_θ as defined in Section 2. Now, if we introduce the variables

$$u_n = e^{-c\tau_{n+2,n+1}} u_{n+1} + a z_{n+1} \quad (23)$$

where $u_N = 0$, and

$$g_n = e^{-c\tau_{n+1,n}} g_{n-1} + e^{-c\tau_{n+1,n}} z_n \quad (24)$$

where $g_0 = 0$, the system of equations can be rewritten as

$$(a + \sigma_1^2) z_1 + e^{-c\tau_{2,1}} u_1 = y_1 \quad (25)$$

$$a g_1 + (a + \sigma_2^2) z_2 + e^{-c\tau_{3,2}} u_2 = y_2 \quad (26)$$

$$a g_2 + (a + \sigma_3^2) z_3 = y_3 \quad (27)$$

The system defined by Equation (23) through Equation (27) can be rewritten as a matrix equation to show the benefit that this seemingly trivial reformulation provides:

$$\begin{pmatrix} a + \sigma_1^2 & e^{-c\tau_{2,1}} & 0 & 0 & 0 & 0 & 0 \\ e^{-c\tau_{2,1}} & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & a & e^{-c\tau_{3,2}} & 0 & 0 \\ 0 & 0 & a & a + \sigma_2^2 & e^{-c\tau_{3,2}} & 0 & 0 \\ 0 & 0 & e^{-c\tau_{3,2}} & e^{-c\tau_{3,2}} & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & a \\ 0 & 0 & 0 & 0 & 0 & a & a + \sigma_3^2 \end{pmatrix} \begin{pmatrix} z_1 \\ u_1 \\ g_1 \\ z_2 \\ u_2 \\ g_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ 0 \\ 0 \\ y_2 \\ 0 \\ 0 \\ y_3 \end{pmatrix}$$

where we will follow Ambikasaran (2015) and call this the “extended” system and rewrite Equation (21) as

$$K_{\text{ext}} \mathbf{z}_{\text{ext}} = \mathbf{y}_{\text{ext}} \quad (28)$$

Even though K_{ext} is a larger matrix than the original K that we started with, it is now has a sparse banded structure that can be exploited to solve the system efficiently. In particular, band matrix solvers are available that can perform an LU-decomposition of banded matrices like this in $\mathcal{O}(N)$ operations – instead of the $\mathcal{O}(N^3)$ that would be required in general – and we can take advantage of these algorithms to solve our system exactly because the target vector \mathbf{z} is a subset of the elements of \mathbf{z}_{ext} .

In the following section, we will discuss this method more generally but it’s worth noting a few important facts that can already be seen in this example. First, the fundamental reason why this matrix K can be solved efficiently is the following property of exponentials

$$e^{-c(t_3-t_2)} e^{-c(t_2-t_1)} = e^{-c(t_3-t_2+t_2-t_1)} = e^{-c(t_3-t_1)} \quad (29)$$

and it is important to note that this property does not extend to other common covariance functions like the “exponential-squared” function

$$k(\tau) \propto e^{-c\tau^2} \quad . \quad (30)$$

Furthermore, our derivation of the extended matrix requires that the data points be monotonically sorted in time. Neither of these properties will be satisfied in general for multidimensional inputs and all of our following discussion will assume a sorted one-dimensional dataset.

Ambikasaran (2015) demonstrated two key facts that allow us to use this extended matrix formalism in practice. First, even if the covariance function is a mixture of exponentials, the extended matrix will still be banded with a bandwidth that scales linearly with the number of components J . Second, Ambikasaran (2015) proved that the determinant of K_{ext} is equal to the determinant of K up to a sign. This means that we can use this extended matrix formalism to compute the marginalized likelihood in $\mathcal{O}(N)$ operations.

4.2. The algorithm

In this section, we generalize the method from the previous section to the covariance function given by Equation (8). This derivation follows Ambikasaran (2015) but it includes explicit treatment of complex parameters, and their complex conjugates.

In the case of the full **celerite** covariance function (Equation 8), we introduce the following auxiliary variables in analogy to the u_n and g_n that we introduced in the previous section

$$\phi_{n,j} = e^{-c_j \tau_{n+1,n}} \cos(d_j \tau_{n+1,n}) \quad (31)$$

$$\psi_{n,j} = -e^{-c_j \tau_{n+1,n}} \sin(d_j \tau_{n+1,n}) \quad (32)$$

$$g_{n,j} = \phi_{n,j} g_{n-1,j} + \phi_{n,j} z_n + \psi_{n,j} h_{n-1,j} \quad (33)$$

$$h_{n,j} = \phi_{n,j} h_{n-1,j} - \psi_{n,j} z_n - \psi_{n,j} g_{n-1,j} \quad (34)$$

$$u_{n,j} = \phi_{n+1,j} u_{n+1,j} + a_j z_{n+1} + \psi_{n+1,j} v_{n+1,j} \quad (35)$$

$$v_{n,j} = \phi_{n+1,j} v_{n+1,j} - b_j z_{n+1} - \psi_{n+1,j} u_{n+1,j} \quad (36)$$

with the boundary conditions

$$g_{0,j} = 0 \quad , \quad h_{0,j} = 0 \quad , \quad u_{N,j} = 0 \quad , \quad \text{and} \quad v_{N,j} = 0 \quad (37)$$

for all j . Using these variables and some algebra, we find that the following expression

$$\sum_{j=1}^J [a_j g_{n,j} + b_j h_{n,j}] + \left[\sigma_n^2 + \sum_{j=1}^J a_j \right] + \sum_{j=1}^J [\phi_{n,j} u_{n,j} + \psi_{n,j} v_{n,j}] = r_{\theta,n} \quad (38)$$

is equivalent to the target matrix equation

$$K \mathbf{z} = \mathbf{r}_{\theta} \quad (39)$$

if $r_{\boldsymbol{\theta},n}$ is the n -th element of the residual vector $\mathbf{r}_{\boldsymbol{\theta}}$ defined in Section 2. Equation (31) through Equation (38) define a banded matrix equation in the “extended” space and, as before, this can be used to solve for $K^{-1} \mathbf{r}_{\boldsymbol{\theta}}$ and $\det K$ in $\mathcal{O}(N)$ operations. Figure 1 shows a pictorial representation of the sparsity pattern of the extended matrix K_{ext} . Given that definition of K_{ext} , the corresponding extended vectors \mathbf{z}_{ext} and \mathbf{r}_{ext} are defined schematically as

$$\mathbf{z}_{\text{ext}}^{\text{T}} = \begin{pmatrix} z_1 & u_{1,j} & v_{1,j} & g_{1,j} & h_{1,j} & z_2 & u_{2,j} & \cdots & h_{N-1,j} & z_N \end{pmatrix} \quad (40)$$

and

$$\mathbf{r}_{\text{ext}}^{\text{T}} = \begin{pmatrix} r_{\boldsymbol{\theta},1} & 0 & 0 & 0 & 0 & r_{\boldsymbol{\theta},2} & 0 & \cdots & 0 & r_{\boldsymbol{\theta},2} \end{pmatrix} . \quad (41)$$

After constructing the extended matrix (using a compact storage format), the extended matrix can be factorized using a LU-decomposition⁴ routine optimized for band or sparse matrices. This decomposition can then be used to compute the determinant of K , solve $K^{-1} \mathbf{r}_{\boldsymbol{\theta}}$, and subsequently calculate the marginalized likelihood in Equation (3).

⁴ Even though K_{ext} is symmetric, it is not positive definite so a Cholesky solver cannot be used for increased efficiency.

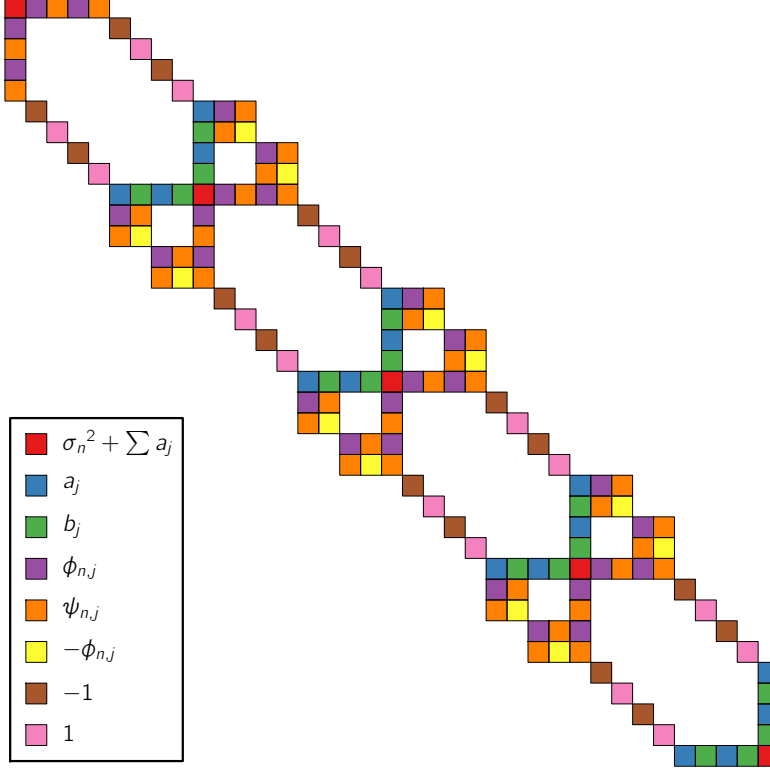


Figure 1. A pictorial representation of the sparse extended matrix K_{ext} with $N = 5$ and $J = 2$. Each colored block corresponds to a non-zero entry in the matrix as described in the legend.

4.3. Implementation considerations \mathcal{E} scaling

The extended system defined in the previous section is sparse with fewer than a few percent non-zero entries and band structure. In this Section, we empirically investigate the performance and scaling of three different algorithms for solving this extended systems:

1. A general sparse LU solver that exploits the sparsity but not the band structure (Guennebaud et al. 2010).
2. A simple algorithm for computing the LU decomposition for banded matrices using Gaussian elimination (Press et al. 1992; Press et al. 2007),
3. The general banded LU decomposition implementation from LAPACK⁵ (Anderson et al. 1999) using optimized BLAS routines, and

Ambikasaran (2015) found an empirical scaling of $\mathcal{O}(N J^2)$ with their method that used the sparse LU decomposition implemented in the **SuperLU** package (Demmel et al.

⁵ We use the `dgbtrf` and `dgbtrs` methods from LAPACK.

1999). The theoretical scaling for a band LU decomposition is $\mathcal{O}(N J^3)$ because the dimension of the extended matrix scales as $N J$ and the bandwidth scales with J (Press et al. 1992; Press et al. 2007). We find that, while the vanilla banded solver scales as expected, the LAPACK algorithm with tuned BLAS routines scales empirically as $\mathcal{O}(N J^2)$ empirically and offers the fastest solves for $J \gtrsim 8$ on all platforms that we tested.

$$\text{NNZ: } (10 J + 1) N - 4 J + 1$$

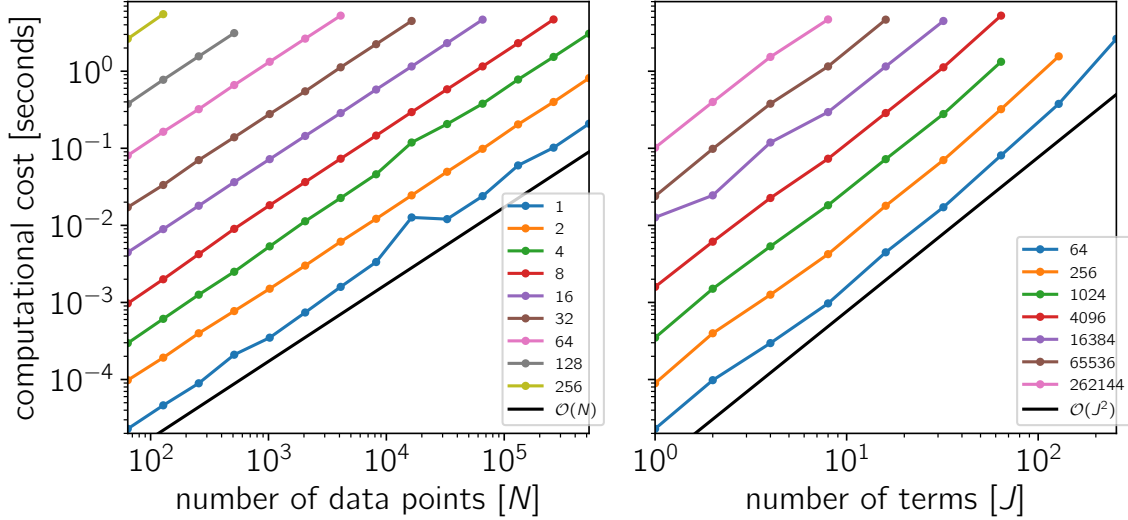


Figure 2. Benchmark

5. CELERITE AS A MODEL OF STELLAR VARIATIONS

DFM: Add more intro to this section.

A special case of the `celerite` model of great physical interest is a stochastically-driven simple harmonic oscillator. The differential equation for this system is

$$\left[\frac{d^2}{dt^2} + \frac{\omega_0}{Q} \frac{d}{dt} + \omega_0^2 \right] y(t) = \epsilon(t) \quad (42)$$

where ω_0 is the frequency of the undamped oscillator, Q is the quality factor of the oscillator, and $\epsilon(t)$ is a stochastic driving force. In the limit of an infinite time series and white random forcing, the PSD of this equation is given by (Anderson et al. 1990)

$$S(\omega) = \sqrt{\frac{2}{\pi}} \frac{S_0 \omega_0^4}{(\omega^2 - \omega_0^2)^2 + \omega_0^2 \omega^2 / Q^2} \quad (43)$$

where S_0 is proportional to the power at $\omega = \omega_0$, $S(\omega_0) = \sqrt{2/\pi} S_0 Q^2$. The power spectrum in Equation (43) matches Equation (9) if

$$a_j = S_0 \omega_0 Q \quad (44)$$

$$b_j = \frac{S_0 \omega_0 Q}{\sqrt{4Q^2 - 1}} \quad (45)$$

$$c_j = \frac{\omega_0}{2Q} \quad (46)$$

$$d_j = \frac{\omega_0}{2Q} \sqrt{4Q^2 - 1} \quad , \quad (47)$$

for $Q \geq \frac{1}{2}$. For $0 < Q \leq \frac{1}{2}$, Equation (43) can be captured by a pair of **celerite** terms with parameters

$$\begin{aligned} a_{j\pm} &= \frac{1}{2} S_0 \omega_0 Q \left[1 \pm \frac{1}{\sqrt{1 - 4Q^2}} \right] \\ b_{j\pm} &= 0 \\ c_{j\pm} &= \frac{\omega_0}{2Q} \left[1 \mp \sqrt{1 - 4Q^2} \right] \\ d_{j\pm} &= 0 \quad . \end{aligned} \quad (48)$$

These identities yield a kernel of the form

$$k(\tau) = S_0 \omega_0 Q e^{-\frac{\omega_0 \tau}{2Q}} \begin{cases} \cosh(\eta \omega_0 \tau) + \frac{1}{2\eta Q} \sinh(\eta \omega_0 \tau), & 0 < Q < 1/2 \\ 2(1 + \omega_0 \tau), & Q = 1/2 \\ \cos(\eta \omega_0 \tau) + \frac{1}{2\eta Q} \sin(\eta \omega_0 \tau), & 1/2 < Q \end{cases} \quad (49)$$

where $\eta = |1 - (4Q^2)^{-1}|^{1/2}$. It is interesting to note that, because of the damping, the characteristic oscillation frequency in this model d_j , for any finite quality factor $Q > 1/2$, is not equal to the frequency of the undamped oscillator ω_0 .

The power spectrum in Equation (43) has several limits of physical interest:

- For $Q = 1/\sqrt{2}$, Equation (43) simplifies to

$$S(\omega) = \sqrt{\frac{2}{\pi}} \frac{S_0}{(\omega/\omega_0)^4 + 1} \quad . \quad (50)$$

This, in turn, is the most commonly used model for the background granulation noise in astero-seismic (Kallinger et al. 2014) and helioseismic (Harvey 1985; Michel et al. 2009) analyses. The Fourier transform of this PSD corresponds to the kernel

$$k(\tau) = S_0 \omega_0 e^{-\frac{1}{\sqrt{2}} \omega_0 \tau} \cos\left(\frac{\omega_0 \tau}{\sqrt{2}} - \frac{\pi}{4}\right). \quad (51)$$

- Substituting $Q = 1/2$, Equation (43) becomes

$$S(\omega) = \sqrt{\frac{2}{\pi}} \frac{S_0}{[(\omega/\omega_0)^2 + 1]^2} \quad (52)$$

with the corresponding covariance function (using Equation 8 and Equation 48)

$$k(\tau) = \lim_{f \rightarrow 0} \frac{1}{2} S_0 \omega_0 \left[(1 + 1/f) e^{-\omega_0 (1-f) \tau} + (1 - 1/f) e^{-\omega_0 (1+f) \tau} \right] \quad (53)$$

$$= S_0 \omega_0 e^{-\omega_0 \tau} [1 + \omega_0 \tau] \quad (54)$$

or equivalently (using Equation 8 and Equation 44)

$$k(\tau) = \lim_{f \rightarrow 0} S_0 \omega_0 e^{-\omega_0 \tau} \left[\cos(f \tau) + \frac{\omega_0}{f} \sin(f \tau) \right] \quad (55)$$

$$= S_0 \omega_0 e^{-\omega_0 \tau} [1 + \omega_0 \tau] \quad (56)$$

This covariance function is also known as the Matérn-3/2 function (Rasmussen & Williams 2006). This suggests that the Matérn-3/2 covariance can be well approximated using the `celerite` framework with a small value of f in Equation (55) but we caution that this could also lead to numerical issues with the solver.

- Finally, in the limit of large Q , the model approaches a high quality oscillation with frequency ω_0 and covariance function

$$k(\tau) \approx S_0 \omega_0 Q \exp\left(-\frac{\omega_0 \tau}{2Q}\right) \cos(\omega_0 \tau) \quad (57)$$

Figure 3 shows a plot of the PSD for these limits and several other values of Q . This figure demonstrates that for $Q \leq 1/2$, the model has no oscillatory behavior and that for large Q , the shape of the PSD near the peak frequency approaches a Lorentzian.

These special cases demonstrate that the stochastically-driven simple harmonic oscillator provides a physically motivated model that is flexible enough to describe a wide range of stellar variations. Low $Q \approx 1$ can capture granulation noise and high $Q \gg 1$ is a good model for asteroseismic oscillations. In practice, we will take a sum over oscillators with different values of Q , S_0 , and ω_0 to give a sufficient accounting of the power spectrum stellar time series. Since this kernel is exactly described by the exponential kernel, the likelihood (Equation 3) can be evaluated for a time series with N measurements in $\mathcal{O}(N)$ operations using the `celerite` method described in the previous section.

6. EXAMPLES WITH SIMULATED DATA

6.1. Recovery of a *celerite* process

In this first example, we will simulate a dataset using a known `celerite` process and fit it with `celerite` to demonstrate that valid inferences can be made in this idealized case. The simulated dataset is shown in the left panel of Figure 4 and it was generated using a SHO kernel (Equation 49) with parameters $S_0 = 1$, $\omega_0 = e^2$, and $Q = e^2$. The true PSD is shown as a dashed line in the right panel of Figure 4. We applied

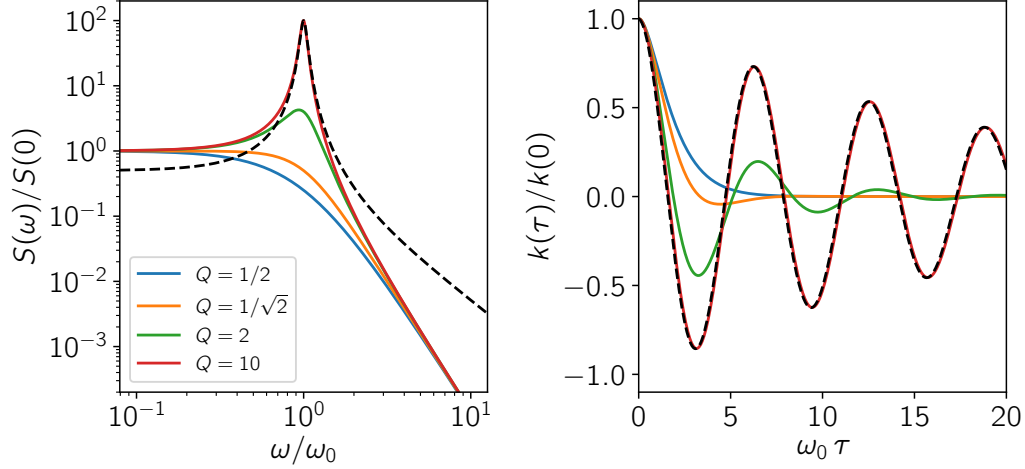


Figure 3. (left) The power spectrum of a stochastically-driven simple harmonic oscillator (Equation 43) plotted for several values of the quality factor Q . For comparison, the dashed line shows the Lorentzian function from Equation (11) with $c_j = \omega_0/2Q = 1/20$ and normalized so that $S(d_j)/S(0) = 100$. (right) The corresponding autocorrelation functions with the same colors.

log-uniform priors to all of the parameters and used `emcee` (Foreman-Mackey et al. 2013) to sample the joint posterior density and computed the marginalized posterior inference of the PSD. This inference is shown in the right panel of Figure 4 as a blue contour indicating 68% of the posterior mass. It is clear from this figure that, as expected, the inference correctly reproduces the true PSD.

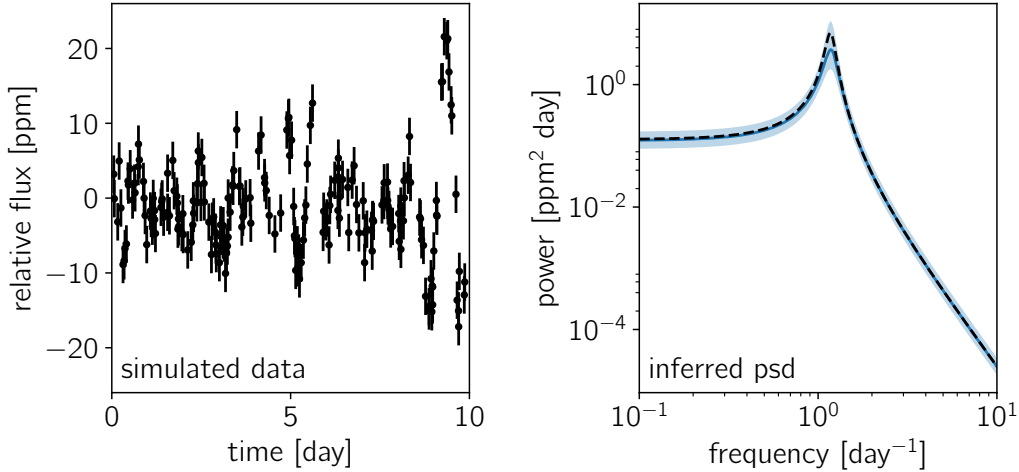


Figure 4. (left) A simulated dataset. (right) The inferred PSD – the blue contours encompass 68% of the posterior mass – compared to the true PSD (dashed black line).

6.2. Inferences with the “wrong” model

For this example, we simulate a dataset using a known GP model with a kernel outside of the support of a **celerite** process. This means that the true autocorrelation of the process can never be correctly represented by the model that we’re using to fit but we will use this example to demonstrate that, at least in this case, valid inferences can still be made about the physical parameters of the model.

For this example, the data are simulated from a quasiperiodic GP with the kernel

$$k_{\text{true}}(\tau) = \alpha \exp\left(-\frac{\tau^2}{2\lambda^2}\right) \cos\left(\frac{2\pi\tau}{P_{\text{true}}}\right) \quad (58)$$

where P_{true} is the fundamental period of the process. This autocorrelation structure yields the power spectrum

$$S_{\text{true}}(\omega) = \frac{\lambda\alpha}{2} \left[\exp\left(-\frac{\lambda^2}{2} \left(\omega - \frac{2\pi}{P_{\text{true}}}\right)^2\right) + \exp\left(-\frac{\lambda^2}{2} \left(\omega + \frac{2\pi}{P_{\text{true}}}\right)^2\right) \right] \quad (59)$$

that for large ω falls off exponentially. When compared to Equation (9) – that for large ω goes as ω^{-4} , at most – it is clear that a **celerite** model can never perfectly reproduce the structure of this process. That being said, we will demonstrate that rigorous inferences can be made about P_{true} even with an effective model. The left panel of Figure 5 shows the simulated dataset. We then fit this simulated data using the product of two SHO terms (Equation 49) where one of the terms has $S_0 = 1$ and $Q = 1/\sqrt{2}$ and the other has $\omega_0 = 2\pi/P$. We note that using Equation (14), the product of two **celerite** terms can also be expressed as a **celerite** term. We apply log-uniform priors on all the parameters and use **emcee** (Foreman-Mackey et al. 2013) to sample the posterior probability for all of the parameters. The inferred distribution for the parameter P is shown in the right panel of Figure 5 and compared to the true period P_{true} and the inferences made using the correct model (Equation 58). The inference made using this effective **celerite** model are indistinguishable from the inferences made using the correct model but substantially less computation time is required for the **celerite** inference.

7. EXAMPLES WITH REAL DATA

In this section we will demonstrate several use cases of **celerite** when applied to real datasets. Each of these examples touches on an active area of research so we will limit our examples to be qualitative in nature and make no claim of optimality but we hope that these examples will encourage interested readers to investigate the applicability of **celerite** to their research.

All of the following examples show time domain datasets with a clear bias in favor of large homogeneous photometric surveys but these methods can similarly be applied to spectroscopy, where wavelength – instead of time – is the independent coordinate and other one-dimensional domains. It is also possible to cast some two-dimensional problems in the **celerite** framework (*DFM: CITE Ian C*).

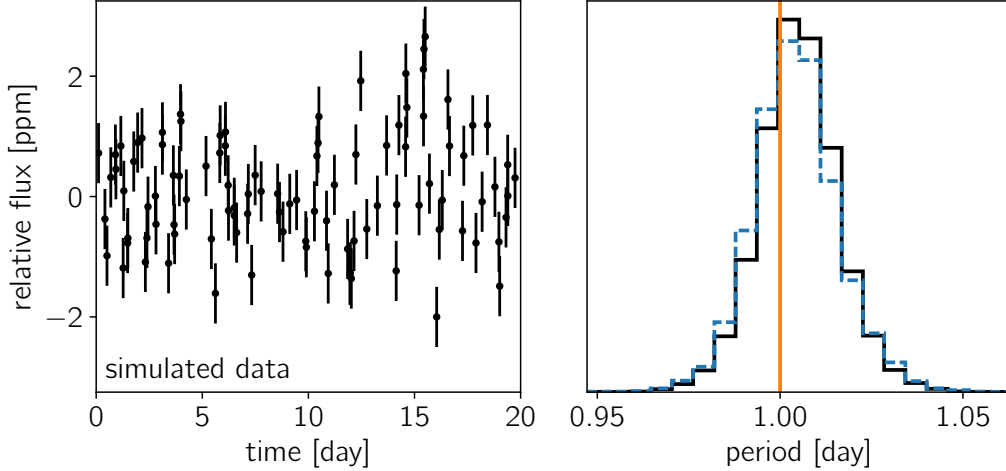


Figure 5. (left) A simulated dataset. (right) The inferred period of the process. The true period is indicated by the vertical orange line, the posterior inference using the correct model is shown as the blue dashed histogram, and the inference made using the “wrong” effective model is shown as the black histogram.

7.1. Asteroseismic oscillations

The asteroseismic oscillations of thousands of stars were measured using light curves from the **Kepler** mission (*DFM: cite*) and asteroseismology is a key science driver for many of the upcoming large scale photometric surveys (*DFM: cite*). Most asteroseismic analyses have been limited to relatively high signal-to-noise oscillations because the standard methods based on statistics of the empirical periodogram of the data cannot be used to formally propagate the measurement uncertainties to the constraints on physical parameters (*DFM: CITE*) and more sophisticated methods are computationally expensive and they scale poorly to the current state-of-the-art datasets (*DFM: CITE Brewer etc.*).

celerite alleviates these problems by providing a physically motivated probabilistic model that can be evaluated efficiently even for large datasets. In practice, we will model the star as a mixture of stochastically driven simple harmonic oscillators where the amplitudes and frequencies of the oscillations are computed using a physical model and evaluate the probability of the observed dataset using a Gaussian Process with a PSD given by a sum of terms given by Equation (43). This gives us a method of computing the likelihood function for the parameters of the physical model in $\mathcal{O}(N)$ operations and this can be combined with standard non-linear optimization or posterior sampling methods to infer constraints on the parameters for the given dataset.

To demonstrate this method, we will use a very simple heuristic model based on *DFM: CITE* where the model PSD is given by a mixture of 8 components with amplitudes and frequencies specified by ν_{\max} , $\Delta\nu$, and several nuisance parameters. The first term is used to capture the granulation “background” using Equation (50) with two free parameters S_g and ω_g . The remaining 7 terms are given by Equation (43)

where Q is a nuisance parameter shared between terms and the frequencies are given by

$$\omega_{0,j} = 2 \pi (\nu_{\max} + j \Delta\nu + \epsilon) \quad (60)$$

and the amplitudes are given by

$$S_{0,j} = \frac{A}{Q^2} \exp \left(-\frac{[j \Delta\nu + \epsilon]^2}{2 W^2} \right) \quad (61)$$

where j is an integer running from -3 to 3 and ϵ , A , and W are shared nuisance parameters. This model could be easily extended to include small frequency splitting and ν_{\max} and $\Delta\nu$ could be replaced by physical parameters like $\log g$.

To demonstrate the applicability of this model, we apply it to infer the asteroseismic parameters of the giant star KIC 11615890 observed by the **Kepler** Mission. The goal of this example is to show that, even for a low signal-to-noise dataset with a short baseline, it is possible to infer asteroseismic parameters with formal uncertainties that are consistent with the parameters inferred with a much larger dataset. Looking forward to **TESS**, we will infer ν_{\max} and $\Delta\nu$ using only one month of **Kepler** data and compare our results to the results inferred from the full 4 year baseline of the mission. For KIC 11615890, the published asteroseismic parameters based on the full dataset are (Pinsonneault et al. 2014)

$$\nu_{\max} = 171.94 \pm 3.62 \mu\text{Hz} \quad \text{and} \quad \Delta\nu = 13.28 \pm 0.29 \mu\text{Hz} \quad . \quad (62)$$

We randomly select a month-long segment of **Kepler** data, initialize our **celerite** model using a grid search in the parameter space, and then use **emcee** (Foreman-Mackey et al. 2013) to sample the joint posterior density for the full set of parameters. Figure 6 shows the marginalized density for ν_{\max} and $\Delta\nu$ compared to the results from the literature. *DFM: Add some more discussion here.*

7.2. Stellar rotation

Another source of variability that can be measured from time series measurements of stars is rotation. The inhomogeneous surface of the star (spots, plage, *etc.*) imprints itself as quasiperiodic variations in photometric or spectroscopic observations (*DFM: CITE*). It has been demonstrated that for light curves with nearly uniform sampling, the empirical autocorrelation function provides a reliable estimate of the rotation period of a star (*DFM: CITE*) and that a GP model with a quasiperiodic covariance function can be used to make probabilistic measurements even with sparsely sampled data (R. Angus, *et al.* in prep.). The covariance function used for this type of analysis has the form

$$k(\tau) = A \exp \left(-\frac{\tau^2}{2 \ell^2} - \Gamma \sin^2 \left(\frac{\pi \tau}{P} \right) \right) \quad (63)$$

where P is the period of the oscillation. The key difference between this function and other quasiperiodic kernels is that it is everywhere positive. We can construct a

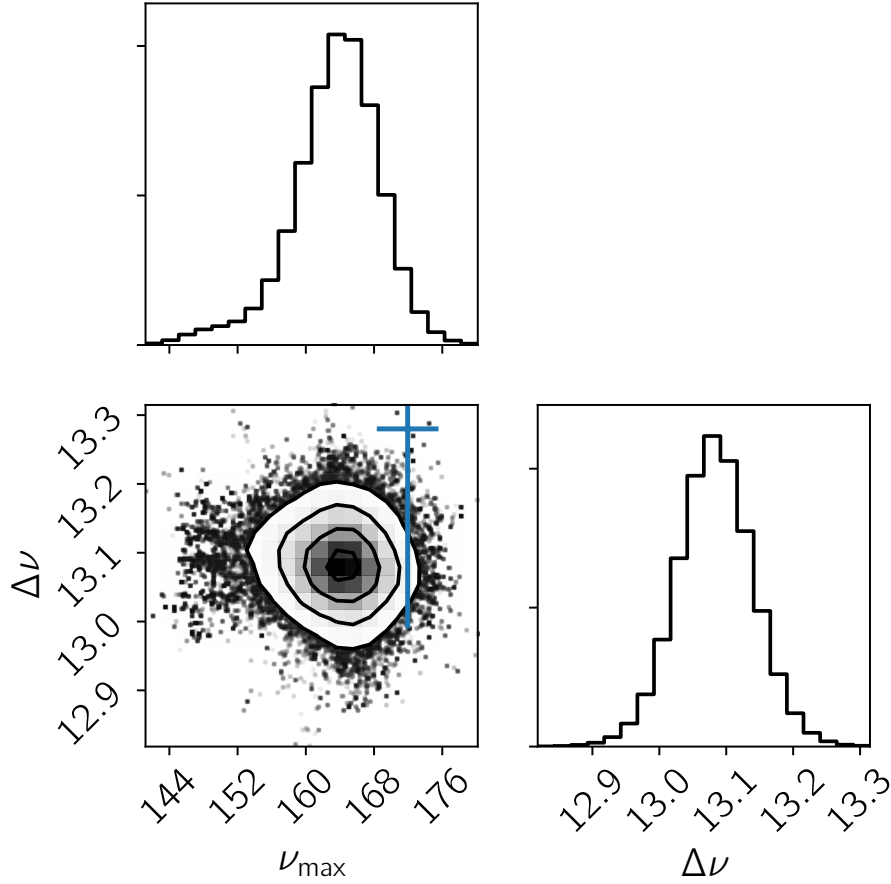


Figure 6. The probabilistic constraints on ν_{\max} and $\Delta\nu$ from the inference shown in Figure 7 compared to the published value (error bar) based on the full Kepler dataset *DFM: CITE*.

simple *celerite* covariance function with similar properties as follows

$$k(\tau) = \frac{a}{2+b} e^{-c\tau} \left[\cos\left(\frac{2\pi\tau}{P}\right) + (1+b) \right] \quad (64)$$

for $a > 0$, $b > 0$, and $c > 0$. The covariance function in Equation (64) cannot exactly reproduce Equation (63) but, since Equation (63) is only an effective model, Equation (64) can be used as a drop-in replacement for a substantial gain in computational efficiency.

7.3. Exoplanet transit fitting

In this example, we inject the signal of a simulated exoplanet transit into a real Kepler light curve and then demonstrate that we can recover the true physical parameters of the exoplanet while modeling the stellar variability using *celerite*. This example is different from all the previous examples because in this case, we are uninterested in the inferred parameters of the covariance model. Instead, we're interested in inferring constraints on the parameters of the mean model. In Equation (3) these parameters are called θ and in this example, the mean function $\mu_\theta(t)$ is a limb-darkened transit

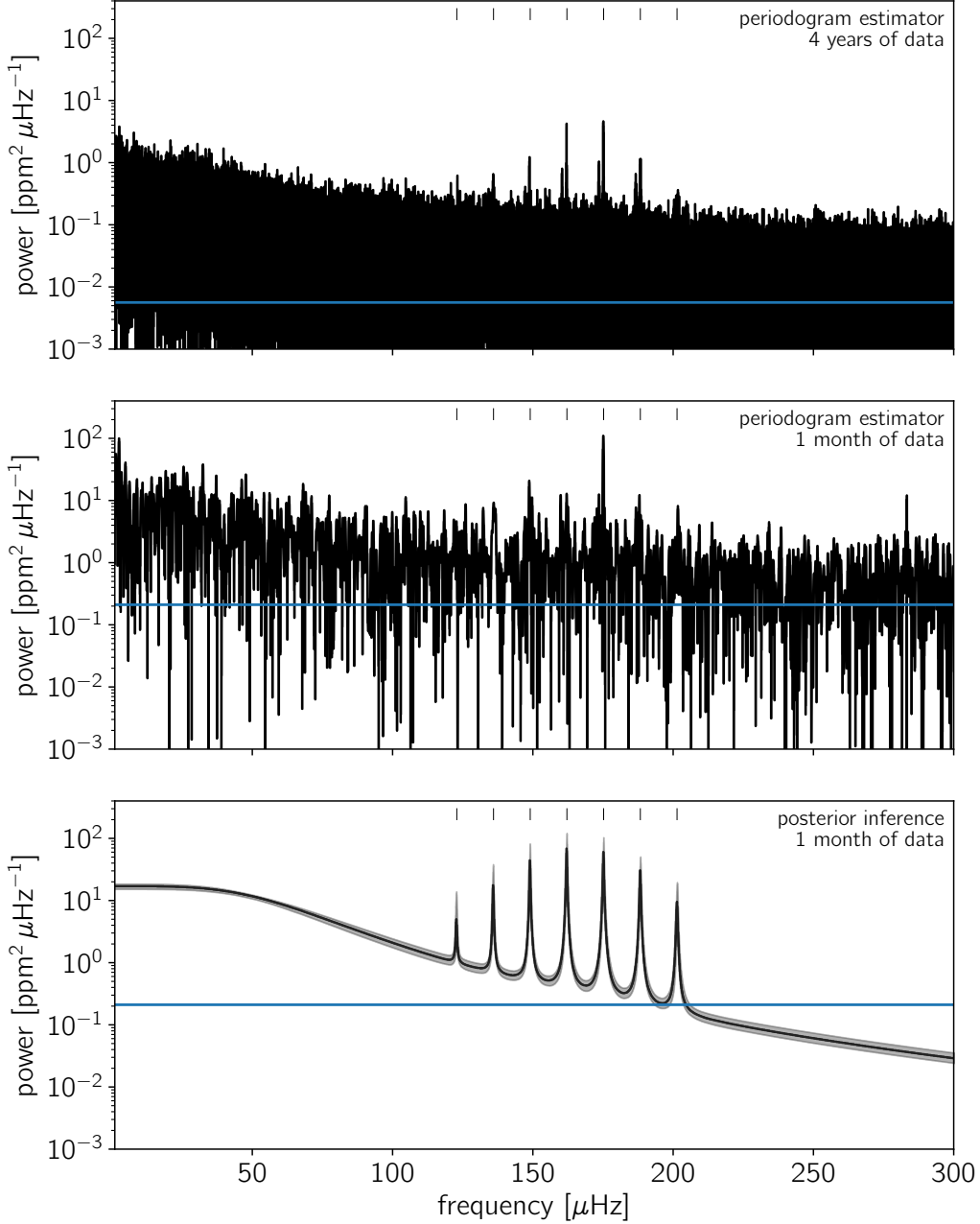


Figure 7. A comparison between the empirical PSD and the posterior inference of the PSD as a mixture of stochastically driven simple harmonic oscillators. (top) The periodogram of the Kepler light curve for KIC 11615890 computed on the full four year baseline of the mission. The light gray curve shows the raw periodogram and the black curve has been smoothed with a Gaussian filter with *DFM: SOME WIDTH*. (middle) The same periodogram computed using about a month of data. (bottom) The power spectrum inferred using the mixture of SHOs model described in the text and only one month of Kepler data. The black line shows the median of posterior PSD and the gray contours show the 68% credible region.

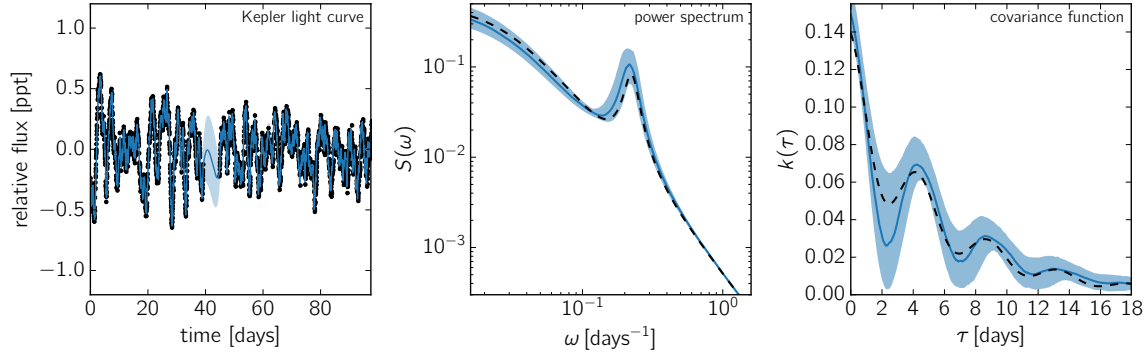


Figure 8. Inferred constraints on a quasiperiodic GP model using the covariance function in Equation (64) and one quarter of Kepler data. (left) The Kepler data (black points) and the maximum likelihood model prediction (blue curve). The solid blue line shows the predictive mean and the blue contours show the predictive standard deviation. (center) Inferred constraints on the model PSD. The dashed line shows the maximum likelihood PSD, the blue solid line shows the median of posterior PSD, and the blue contours show the 68% credible region. (right) Inferred constraints on the model covariance function from Equation (64). The dashed line shows the maximum likelihood model, the blue solid line shows the median of posterior, and the blue contours show the 68% credible region.

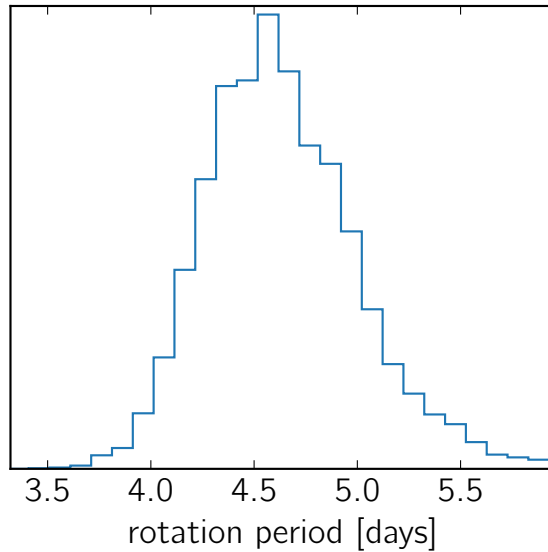


Figure 9. The posterior constraint on the rotation period of KIC 1430163 using the dataset and model from Figure 8. The period is the parameter P in Equation (64) and this figure shows the posterior distribution marginalized over all other nuisance parameters in Equation (64). This is consistent with the published rotation period made using the autocorrelation function and the full Kepler data (*DFM: CITE*).

light curve *DFM: CITE MA* parameterized by a period P , a transit duration T , a phase or epoch t_0 , an impact parameter b , the radius of the planet in units of the stellar radius R_P/R_\star , and several parameters describing the limb-darkening profile of the star *DFM: CITE*.

KIC 1430163 with simulated transit. Figure 10 and Figure 11.

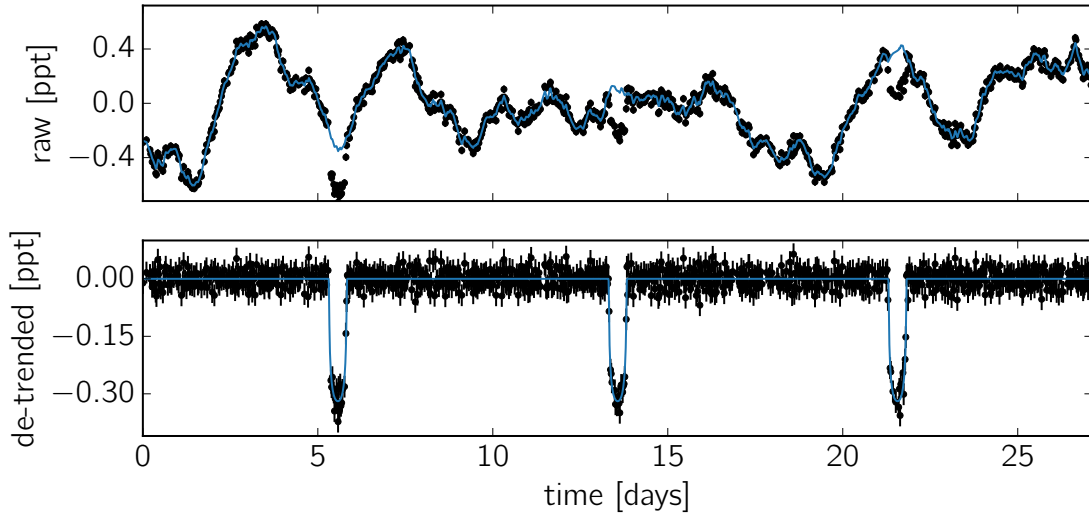


Figure 10.

8. COMPARISONS TO OTHER METHODS

Toeplitz, KISS-GP, CARMA, HODLR.

Limitations of celerite: one-dimension, stationary, etc.

9. SUMMARY

10. SUMMARY

Although we have in mind application of this fast method to stellar variability, the method is general for one-dimensional GP problems, and may be applied to other problems. Within astrophysics, correlated noise (due to the environment, detector, or modeling uncertainty) may be present in gravitational wave time series, and so Gaussian processes may be a way to address this problem (Moore et al. 2016). Accreting black holes show time series which may be modeled by correlated noise (Kelly et al. 2014); indeed, this was the motivation for the original technique developed by Rybicki & Press (Rybicki & Press 1992, 1995). This approach may be broadly used for characterizing quasar variability (MacLeod et al. 2010), measuring time lags with reverberation mapping (Zu et al. 2011), and modeling time delays in multiply-imaged gravitationally-lensed systems (Press & Rybicki 1998).

Outside of astronomy, this technique may have application to seismology (Robinson 1967),

There are three primary applications we envision for the *celerite* formalism: 1). modeling the variability of an astrophysical system to infer its properties; 2). accounting for astrophysical variability as a source of noise when trying to detect additional phenomena; 3). interpolating or extrapolating variability to future or missing times. Some examples of the first are determining the characteristic variability timescale

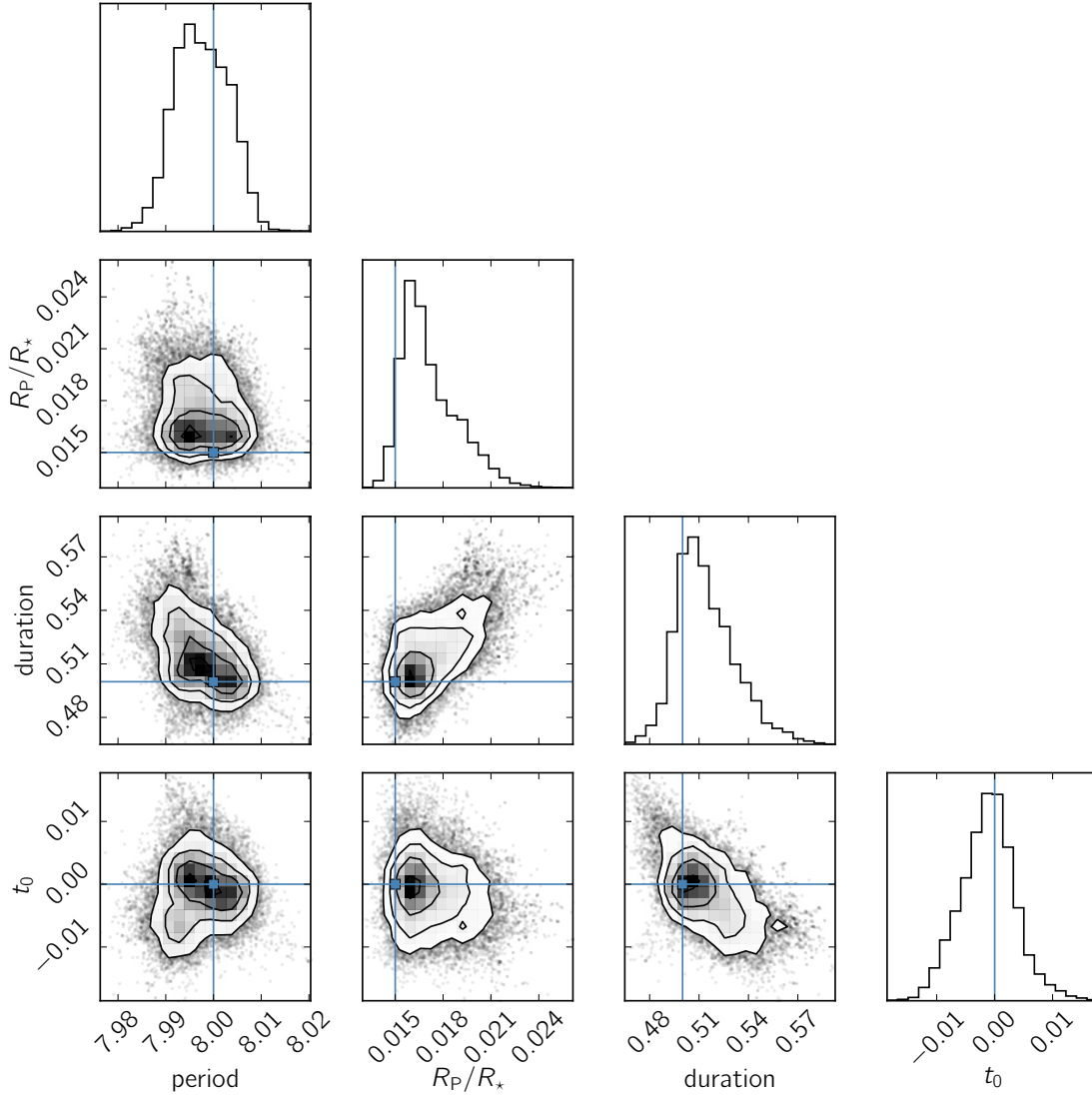


Figure 11.

of a quasar, measuring the asteroseismic variation of a star, or detection of quasi-periodic variability in a high-energy source. Examples of the second are detecting transiting exoplanets and correcting for stellar activity in radial velocity measurements. Examples of the third are measuring time delays in multiply-imaged gravitationally lensed sources and reverberation mapping of AGN.

Our background is in studying transiting exoplanets, a field which has only recently begun to adopt full covariance matrices in analyzing the noise in transiting planet lightcurves. One promising early attempt at accounting for correlated noise was the wavelet approach of Carter & Winn (2010). Their technique allowed for a power-law spectrum for the noise scaling as $\omega^{-\alpha}$, which unfortunately is not a normalizable, nor physically-accurate, description of stellar variability. Their approach runs in $\mathcal{O}(N)$

time for $\alpha = 1$, which is ‘flicker’ or pink noise; however, this particular type of noise overpredicts the power spectrum at low frequency compared to the granulation power spectrum. It also cannot describe quasi-periodic noise.

Further progress was made by parameterizing the covariance matrix with simple, analytic functions that describe the autocorrelation function. The parameters of these functions can then be optimized to best match the observed variability pattern of the residuals (after subtracting a transit model). The disadvantage of this approach is that the functions used are chosen somewhat arbitrarily (e.g. the exponential-squared function), and the Cholesky decomposition of the covariance matrix for computing the likelihood takes $\mathcal{O}(N^3)$ operations which prohibits application of this technique to large datasets.

Given the drawbacks of these approaches, the *celerite* formalism allows both a fast computation of the likelihood in $\mathcal{O}(N)$ time, as well as a functional form that accurately describes stellar variability due to the relation to the simple harmonic oscillator power spectrum, which is an analog of stellar asteroseismic oscillations. As higher signal-to-noise observations of transiting exoplanet systems are obtained, the effects of stellar variability will more dramatically impact the correct inference of planetary transit parameters, and so we expect that *celerite* will be important for transit timing, transit spectroscopy, Doppler beaming, phase functions, and more.

APPENDIX

A. ENSURING POSITIVE DEFINITENESS: STURM’S THEOREM

The power spectrum is computed from taking the square of the Fourier transform of the data time series; hence, the power-spectrum must be non-negative. In constructing a kernel, if any of the parameters a_j or b_j are negative, then it is possible for the power spectrum to go negative, which violates the definition of the power spectrum. Consequently, if any of these coefficients is negative, it is necessary to check that the power spectrum is still non-negative. Note that a non-negative power-spectrum does not require the auto-correlation function to be positive for all τ . The positive power spectrum is related to the positive eigenvalues of the covariance matrix which are necessary for a positive-definite matrix in limiting cases (Messerschmitt 2006). We find empirically that requiring an everywhere positive power spectrum results in positive eigenvalues for the covariance matrix, and so we describe here how to ensure a positive power spectrum using Sturm’s theorem.

In the case of J *celerite* terms, we can check for negative values of the PSD by solving for the roots of the power spectrum, abbreviating with $z = \omega^2$:

$$P(\omega) = \sum_{j=1}^J \frac{q_j z + r_j}{z^2 + s_j z + t_j} = 0 \quad (\text{A1})$$

where

$$q_j = a_j c_j - b_j d_j \quad (\text{A2})$$

$$r_j = (d_j^2 + c_j^2)(b_j d_j + a_j c_j) \quad (\text{A3})$$

$$s_j = 2(c_j^2 - d_j^2) \quad (\text{A4})$$

$$t_j = (c_j^2 + d_j^2)^2. \quad (\text{A5})$$

The denominators of each term are positive, so we can multiply through by $\Pi_j (z^2 + s_j z + t_j)$ yielding:

$$P_0(z) = \sum_{j=1}^J (q_j z + r_j) \Pi_{k \neq j} (z^2 + s_k z + t_k) = 0, \quad (\text{A6})$$

which is a polynomial with order $p_{ord} = 2(J - 1) + 1$. With $J = 2$, this yields a cubic equation which may be solved exactly for the roots.

For arbitrary J , a procedure based upon Sturm's theorem (Dörrie 1965) allows one to determine whether there are any real roots within the range $(0, \infty]$. We first construct $P_0(z)$ and its derivative $P_1(z) = P'(z)$, and then loop from $k = 2$ to $k = p_{ord}$, computing $P_k(z) = -\text{rem}(P_{k-2}, P_{k-1})$. The function $\text{rem}(p, q)$ is the remainder polynomial after dividing $p(z)$ by $q(z)$.

We evaluate the z^0 coefficients of each of the polynomial in the series by evaluating $f_0 = \{P_0(0), \dots, P_{p_{ord}}(0)\}$ to give us the signs of these polynomials evaluated at $z = 0$. Likewise, we evaluate the coefficients of the largest order term in each polynomial which gives the sign of the polynomial as $z \rightarrow \infty$. This gives $f_\infty = \{C(P_0, p_{ord}), C(P_1, p_{ord} - 1), \dots, C(P_{p_{ord}}, 1)\}$ where $C(p(z), m)$ returns the coefficient of z^m in polynomial $p(z)$.

With the series of coefficients f_0 and f_∞ , we then determine how many times the sign changes in each of these, where $\sigma(0)$ is the number of sign changes at $z = 0$, and $\sigma(\infty)$ is the number of sign changes at $z \rightarrow \infty$. The total number of real roots in the range $(0, \infty]$ is given by $N_+ = \sigma(0) - \sigma(\infty)$.

We have checked that this procedure works for a wide range of parameters, and we find that it robustly matches the number of positive real roots which we evaluated numerically.

The advantage of this procedure is that it does not require computing the roots, but only carrying out algebraic manipulation of polynomials to determine the number of positive real roots. If a non-zero real root is found, then likelihood may be set to zero.

We have implemented this algorithm in subroutines that may be called to check for positive-definiteness of a kernel with some values of a_j or b_j negative.

All of the code used in this project is available from <https://github.com/dfm/celerite> under the MIT open-source software license. This code (plus some depen-

dencies) can be run to re-generate all of the figures and results in this paper; this version of the paper was generated with git commit 271e8ac (2017-03-14).

It is a pleasure to thank Sivaram Ambikasaran, Megan Bedell, Will Farr, Sam Grunblatt, David W. Hogg, and Dan Huber for helpful contributions to the ideas and code presented here.

EA acknowledges support from NASA grants NNX13AF20G, NNX13AF62G, and NASA Astrobiology Institute’s Virtual Planetary Laboratory, supported by NASA under cooperative agreement NNH05ZDA001C.

This research made use of the NASA **Astrophysics Data System** and the NASA Exoplanet Archive. The Exoplanet Archive is operated by the California Institute of Technology, under contract with NASA under the Exoplanet Exploration Program.

This paper includes data collected by the **Kepler** mission. Funding for the **Kepler** mission is provided by the NASA Science Mission directorate. We are grateful to the entire **Kepler** team, past and present.

These data were obtained from the Mikulski Archive for Space Telescopes (MAST). STScI is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS5-26555. Support for MAST is provided by the NASA Office of Space Science via grant NNX13AC07G and by other grants and contracts.

Facility: Kepler

Software: `corner.py` (Foreman-Mackey 2016), `Eigen` (Guennebaud et al. 2010), `emcee` (Foreman-Mackey et al. 2013), `george` (Ambikasaran et al. 2016), `LAPACK` (Anderson et al. 1999), `matplotlib` (Hunter et al. 2007), `numpy` (Van Der Walt et al. 2011), `transit` (Foreman-Mackey & Morton 2016), `scipy` (Jones et al. 2001).

REFERENCES

- | | |
|--|--|
| <p>Ambikasaran, S. 2015, Numer. Linear Algebra Appl., 22, 1102</p> <p>Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D. W., & O’Neil, M. 2016, IEEE Transactions on Pattern Analysis and Machine Intelligence, 38, 252</p> <p>Anderson, E., Bai, Z., Bischof, C., et al. 1999, LAPACK Users’ Guide, 3rd edn. (Philadelphia, PA: Society for Industrial and Applied Mathematics)</p> <p>Anderson, E. R., Duvall, Jr., T. L., & Jefferies, S. M. 1990, ApJ, 364, 699</p> <p>Demmel, J. W., Eisenstat, S. C., Gilbert, J. R., Li, X. S., & Liu, J. W. H. 1999, SIAM J. Matrix Analysis and Applications, 20, 720</p> <p>Dörrie, H. 1965, 100 Great Problems of Elementary Mathematics: Their History and Solution, Dover Books on Mathematics Series, §24, (Dover Publications), 112–116</p> | <p>Foreman-Mackey, D. 2016, The Journal of Open Source Software, 24, doi:10.21105/joss.00024</p> <p>Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306</p> <p>Foreman-Mackey, D., & Morton, T. 2016, dfm/transit: v0.3.0, , , doi:10.5281/zenodo.159478</p> <p>Guennebaud, G., Jacob, B., et al. 2010, Eigen v3, http://eigen.tuxfamily.org, ,</p> <p>Harvey, J. 1985, in ESA Special Publication, Vol. 235, Future Missions in Solar, Heliospheric & Space Plasma Physics, ed. E. Rolfe & B. Battrock</p> <p>Hunter, J. D., et al. 2007, Computing in science and engineering, 9, 90</p> <p>Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open source scientific tools for Python, ,</p> |
|--|--|

- Kallinger, T., De Ridder, J., Hekker, S., et al. 2014, *A&A*, 570, A41
- Kelly, B. C., Becker, A. C., Sobolewska, M., Siemiginowska, A., & Uttley, P. 2014, *ApJ*, 788, 33
- MacLeod, C. L., Ivezić, Ž., Kochanek, C. S., et al. 2010, *ApJ*, 721, 1014
- Messerschmitt, D. 2006, EECS Dept., Univ. of California, Berkeley, Tech. Rep. No. UCB/EECS-2006-90
- Michel, E., Samadi, R., Baudin, F., et al. 2009, *A&A*, 495, 979
- Moore, C. J., Berry, C. P. L., Chua, A. J. K., & Gair, J. R. 2016, *Physical Review D*, 93, doi:10.1103/physrevd.93.064001
- Pinsonneault, M. H., Elsworth, Y., Epstein, C., et al. 2014, *ApJS*, 215, 19
- Press, W. H., & Rybicki, G. B. 1998, *ApJ*, 507, 108
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical recipes in FORTRAN. The art of scientific computing*
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2007
- Rasmussen, C. E., & Williams, K. I. 2006, *Gaussian Processes for Machine Learning*
- Robinson, E. A. 1967, *GEOPHYSICS*, 32, 418
- Rybicki, G. B., & Press, W. H. 1992, *ApJ*, 398, 169
- . 1995, *Physical Review Letters*, 74, 1060
- Van Der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, *Computing in Science & Engineering*, 13, 22
- Zu, Y., Kochanek, C. S., & Peterson, B. M. 2011, *ApJ*, 735, 80