Draft version December 28, 2016
Preprint typeset using LATEX style AASTeX6 with modifications

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

Daniel Foreman-Mackey[1,2] and Eric Agol[1]

[1]Astronomy Department, University of Washington, Seattle, WA, 98195, USA

[2]Sagan Fellow

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 be written as a mixture of damped sinusoid functions. 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/GenRP` and Zenodo *DFM*: add zenodo archive.

2. GAUSSIAN PROCESSES

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

$$\boldsymbol{y} = \begin{pmatrix} y_1 \cdots y_N \end{pmatrix}^{\mathrm{T}} \tag{1}$$

at coordinates

$$X = \begin{pmatrix} \boldsymbol{x}_1 \cdots \boldsymbol{x}_N \end{pmatrix}^{\mathrm{T}} \tag{2}$$

is

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

where

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

is the vector of residuals and the elements of the covariance matrix K are given by $[K_{\boldsymbol{\alpha}}]_{nm} = k_{\boldsymbol{\alpha}}(\boldsymbol{x}_n,\,\boldsymbol{x}_m)$. The maximum likelihood values for the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ for a given dataset $(\boldsymbol{y},\,X)$ 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 improving this scaling that we call the genrp method. The genrp method requires using a specific model for the covariance $k_{\boldsymbol{\alpha}}(\boldsymbol{x}_n,\,\boldsymbol{x}_m)$ and it has several limitations. The method can only be applied to one-dimensional datasets. When we say "one-dimensional" here, it means that the *input coordinates* $\boldsymbol{x}_n$ are scalar, $\boldsymbol{x}_n \equiv t_n$.[1] Furthermore, the covariance function for the genrp method is "stationary". This means that the function $k_{\boldsymbol{\alpha}}(t_n,\, t_m)$ is only a function of $\tau_{nm} \equiv |t_n - t_m|$.

3. THE GENRP MODEL

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

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

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

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

In this case, the inverse becomes more complicated but Equation (3) can still be evaluated in $\mathcal{O}(J^2\,N)$ 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 $a_j \to a_j + i\,b_j$ and $c_j \to c_j + i\,d_j$. In this case, the covariance function becomes

$$\begin{aligned} k_{\boldsymbol{\alpha}}(\tau_{nm}) = \sigma_n^2\,\delta_{nm} + \sum_{j=1}^{N} \bigg[&\frac{1}{2}(a_j + i\,b_j)\,\exp\left(-(c_j + i\,d_j)\,\tau_{nm}\right) \\ &+ \frac{1}{2}(a_j - i\,b_j)\,\exp\left(-(c_j - i\,d_j)\,\tau_{nm}\right)\bigg] \end{aligned} \tag{7}$$

and, for this function, Equation (3) can still be solved with $\mathcal{O}(J^2\,N)$ operations. The details of this method and a few implementation considerations are discussed in Appendix A.

Equation (7) can also be expressed as

$$\begin{aligned} k_{\boldsymbol{\alpha}}(\tau_{nm}) = \sigma_n^2\,\delta_{nm} + \sum_{j=1}^{J} &\left[a_j\,\exp\left(-c_j\,\tau_{nm}\right)\,\cos\left(d_j\,\tau_{nm}\right)\right. \\ &\left. + b_j\,\exp\left(-c_j\,\tau_{nm}\right)\,\sin\left(d_j\,\tau_{nm}\right)\right] \quad . \end{aligned} \tag{8}$$

[1] 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 genrp method can be applied to *any* one-dimensional dataset.

The Fourier transform of this covariance function is the power spectral density (PSD) of the model and it is given by

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

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

$$k_j(\tau_{nm}) = a_j\ \exp\left(-c_j\,\tau_{nm}\right)\ \cos\left(d_j\,\tau_{nm}\right) \tag{10}$$

and the PSD for the this component is

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

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 $Q_j = c_j$, and period $P_j = 2\,\pi/d_j$.

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

$$k_j(\tau_{nm}) = a_j\ \exp\left(-c_j\,\tau_{nm}\right) \tag{12}$$

with the PSD

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

4. IMPLEMENTATION & PERFORMANCE

4.1. *Implementation considerations*

Solvers. Banded vs. sparse.

4.2. *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 p damped sinusoids we can check for negative values of the PSD by solving for the roots of the power spectrum, abbreviating with $z = \omega^2$:

$$P(\omega) = \sum_{j=1}^{p} \frac{q_j z + r_j}{z^2 + s_j z + t_j} = 0 \tag{14}$$

where

$$\begin{aligned} q_j &= a_j c_j - b_j d_j & (15)\\ r_j &= (d_j^2 + c_j^2)(b_j d_j + a_j c_j) & (16)\\ s_j &= 2(c_j^2 - d_j^2) & (17)\\ t_j &= (c_j^2 + d_j^2)^2. & (18) \end{aligned}$$

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

$$P_0(z) = \sum_{j=1}^{p} (q_j z + r_j) \Pi_{k \neq j}\left(z^2 + s_k z + t_k\right) = 0, \tag{19}$$

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

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 it's derivative $P_1(z) = P'(z)$, and then loop from $k = 2$ to $k = p_{ord}$, computing $P_k(z) = -\mathrm{rem}(P_{k-2}, P_{k-1})$. The function $\mathrm{rem}(p, q)$ 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 $f_0 = \{P_0(0), ..., P_{p_{ord}}(0)\}$ 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 $f_\infty = \{C(P_0, p_{ord}), C(P_1, p_{ord} - 1), ..., C(P_{p_{ord}}, 1)\}$ 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.

4.3. *Benchmarks & scaling*

4.4. *Parameterization & API*

5. GENRP AS A MODEL OF STELLAR VARIATIONS

Another special case of the genrp model of great physical interest is a stochastically driven simple harmonic oscillator. The differential equation for this system is

$$\left[\frac{\mathrm{d}^2}{\mathrm{d}t^2} + \frac{\omega_0}{Q}\,\frac{\mathrm{d}}{\mathrm{d}t} + \omega_0^2\right]\,y(t) = \epsilon(t) \tag{20}$$

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)

$$S(\omega) = \sqrt{\frac{2}{\pi}}\,\frac{S_0\,\omega_0^4}{(\omega^2-\omega_0^2)^2+\omega_0^2\omega^2/Q^2} \tag{21}$$

where S_0 is a normalization constant. The power spectrum in Equation (21) matches Equation (9) if

$$\begin{aligned}
a_j &= S_0\,\omega_0\,Q & (22)\\
b_j &= \frac{S_0\,\omega_0\,Q}{\sqrt{4\,Q^2-1}} & (23)\\
c_j &= \frac{\omega_0}{2\,Q} & (24)\\
d_j &= \frac{\omega_0}{2\,Q}\sqrt{4\,Q^2-1} \quad , & (25)
\end{aligned}$$

for $Q \geq \frac{1}{2}$. For $0 < Q \leq \frac{1}{2}$, Equation (21) can be captured by a pair of genrp terms with parameters

$$\begin{aligned}
a_{j\pm} &= \frac{1}{2}\,S_0\,\omega_0\,Q\,\left[1 \pm \frac{1}{\sqrt{1-4\,Q^2}}\right] & (26)\\
b_{j\pm} &= 0 & (27)\\
c_{j\pm} &= \frac{\omega_0}{2\,Q}\,\left[1 \mp \sqrt{1-4\,Q^2}\right] & (28)\\
d_{j\pm} &= 0 \quad . & (29)
\end{aligned}$$

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, is not equal to the frequency of the undamped oscillator ω_0.

The power spectrum in Equation (21) has several limits of physical interest:

- For $Q = 1/\sqrt{2}$, Equation (21) simplifies to

$$S(\omega) = \sqrt{\frac{2}{\pi}}\,\frac{S_0}{(\omega/\omega_0)^4 + 1} \quad . \tag{30}$$

  A model like this has been used to model the granulation is Solar data (Michel et al. 2009)

- Substituting $Q = 1/2$, Equation (21) becomes

$$S(\omega) = \sqrt{\frac{2}{\pi}}\,\frac{S_0}{\left[(\omega/\omega_0)^2 + 1\right]^2} \tag{31}$$

  with the corresponding covariance function

$$k(\tau) = \lim_{f\to 0} \frac{1}{2}\,S_0\,\omega_0\,e^{-\omega_0\,\tau}\left[\cos(f\,\tau) + \frac{\omega_0}{f}\,\sin(f\,\tau)\right] \tag{32}$$

$$= \frac{1}{2}\,S_0\,\omega_0\,e^{-\omega_0\,\tau}\,[1 - \omega_0\,\tau] \quad . \tag{33}$$

  This covariance function is also known as the Matérn-3/2 function (Rasmussen & Williams 2006).

- Finally, in the limit $Q \to \infty$, Equation (21) becomes

$$S(\omega) = \sqrt{\frac{2}{\pi}}\,\frac{S_0}{\left[(\omega/\omega_0)^2 - 1\right]^2} \quad . \tag{34}$$

The stochastically-driven, damped simple-harmonic oscillator can then describe a wide range of intrinsic stellar variability, with low $Q \approx 1$ representing granulation noise, and high $Q \gg 1$ representing asteroseismic variability. Taking the sum over oscillators with various Q, S_0, and ω_0 can give an accurate accounting of the power spectrum of stellar variability with $Q \geq \frac{1}{2}$ and $S_0 > 0$. As this kernel is exactly described by the exponential kernel, this functional form is tractable to solving in $\mathcal{O}(N)$ operations, as described next.

6. EXAMPLES

6.1. *Simulated data*

1. an OU process
2. a full GenRP process

3. Some standard kernels: Matern, Exp-squared, etc.
4. A KISS-GP process

6.2. *Real data*

1. AGN
2. RR Lyrae
3. Kepler with simulated transit
4. Asteroseismic target
5. Stellar rotational variability

7. COMPARISONS TO OTHER METHODS

Toeplitz, KISS-GP, CARMA, HODLR.
Limitations of GenRP: one-dimension, stationary, etc.

8. SUMMARY

9. GENERALIZED PRESS-RYBICKI

The general problem to be solved for Gaussian Process analysis of stellar time series is to evaluate the likelihood function,

$$\mathcal{L} = \ln p(\mathbf{y}|\mathbf{t},\boldsymbol{\theta},\boldsymbol{\alpha}) = -\frac{1}{2}\ln|\mathbf{K}(\boldsymbol{\alpha})| - \frac{N_{time}}{2}\ln(2\pi) - \frac{1}{2}\mathbf{r}(\boldsymbol{\theta})^T\mathbf{K}(\boldsymbol{\alpha})^{-1}\mathbf{r}(\boldsymbol{\theta}), \tag{35}$$

where $\mathbf{y}$ are the measured time series (for example fluxes or radial velocities) at times $\mathbf{t}$, while the data are modeled with a function $m(\mathbf{t};\boldsymbol{\theta})$ (for example, a transit model or Keplerian orbital model) which leaves residuals $\mathbf{r}(\boldsymbol{\theta}) = \mathbf{y} - m(\mathbf{t};\boldsymbol{\theta})$. The vector $\boldsymbol{\alpha}$ specifies the parameters of the kernel, while the vector $\boldsymbol{\theta}$ specifies the parameters of the physical model. In addition to this likelihood, a prior may be placed on the values of these model parameters (either the physical or kernel); the log of this prior may be added to this log likelihood function. Note that the kernel parameters may also have a physical meaning, but they describe a noisy process, and thus affect the correlated variability of stellar noise rather than a deterministic model as encapsulated in $m(\mathbf{t};\boldsymbol{\theta})$.

Although the complex kernel is a straightforward extension to the `GenRP` formalism, there is a disadvantage in computational speed: complex arithmetic requires ≈ 3 times as many operations as real arithmetic. In addition, it turns out that the complex conjugate component, $(c_j + id_j)^* = c_j - id_j$, in the extended matrix gives a solution which, not surprisingly, is the complex conjugate of the component $c_j + id_j$; thus the computation is redundant. Finally, complex arithmetic has the disadvantage that it is not included in automatic-differentiation packages, which is what we would like to use to compute the derivative of the likelihood (as it is very difficult to compute this explicitly!).

Consequently, we use the fact that $\exp(id_j\tau) = \cos(d_j\tau) + i\sin(d_j\tau)$ to rewrite the extended matrix in terms of real arithmetic using the fact that the imaginary number i may be replaced by a 2×2 anti-symmetric matrix,

$$\begin{pmatrix} 0 & 1 \\ \text{-1} & 0 \end{pmatrix}, \tag{36}$$

while real numbers use the 2×2 identity matrix, $\mathbf{I}$, so that $\cos(d_j\tau) + i\sin(d_j\tau)$ may be rewritten as the rotation matrix

$$\begin{pmatrix} \cos(d_j\tau) & \sin(d_j\tau) \\ -\sin(d_j\tau) & \cos(d_j\tau) \end{pmatrix}. \tag{37}$$

The algebraic properties of these matrices are identical to the complex numbers, but only involve real components. We accomplish this transformation in Appendix **??** by taking sums and differences of the complex equations, resulting in a real extended matrix that is the same size as the complex matrix. This transformation leads to about a factor of two in computational savings, but has the additional advantage of allowing the automatic differentiation due to the use of real arithmetic.

10. KERNEL COMPONENTS

The functional form of k_j may be used to approximate some commonly used Kernel components. The squared exponential (Gaussian) kernel and the Matérn kernels are commonly used radial kernels, while the cosine kernel and the exponential-sine-squared kernels are commonly used periodic kernels. Linear combinations of our basis of damped cosine kernels may be used to approximate each of these kernels to better than 1% with various numbers of damped cosine kernel components:

- **Constant kernel.** A constant kernel component can be used to capture long-timescale correlations in data, and can be represented exactly with $c_j = d_j = 0$.

- **Exponential kernel.** This kernel, $k(\tau) = e^{-|\tau/\tau_0|}$, is exactly represented with $d_j = 0$, $c_j = \tau_0^{-1}$.

- **Cosine kernel.** The cosine kernel, $k(t) = \cos\left(\frac{2\pi}{P}|\tau|\right)$, may be represented exactly with $c_j = 0$ and $d_j = \frac{2\pi}{P}$.

- **Squared exponential.** The squared exponential or Gaussian kernel, $G(z) = e^{-z^2/2}$, has the behavior of declining steeply at large time separation, is an even function, and has $c_1 = 0$. Likewise, the Fourier transform of the squared exponential kernel is $\hat{G}(\omega) = e^{-\omega^2/2}$, which also declines steeply at large ω; this doesn't match the more gradual decline of the power spectrum of a damped sinuosoid. Nevertheless, the squared exponential kernel may be well approximated by the sum of four exponential kernels:

$$\begin{aligned} G(z) \approx\, & 63.119512e^{-1.479579z} - 32.173615e^{-1.593691z} - 29.425156e^{-1.388929z} && (38) \\ & + e^{-1.383928z}\left[-0.519196\cos\left(1.883396z\right) + 0.285478\sin\left(1.883396z\right)\right] && (39) \\ & && (40) \end{aligned}$$

which has an accuracy of better than $< 3\times 10^{-3}$ for all z, and the power spectrum has been constrained to be positive.

- **Matérn $p + 1/2$ kernel.** The Matérn kernel is given by:

$$C_{p+1/2}(x) = \sigma^2 e^{-\sqrt{(2p+1)}x}\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}(\sqrt{(8p+4)}x)^{p-i}, \tag{41}$$

where p is a positive integer. In the limit $p \rightarrow \infty$, this becomes the squared exponential kernel: $C_\infty(x) = \sigma^2 e^{-x^2/2}$. But, for small values of p, the Matérn kernel may be approximated well by the sum of $2p$ exponential kernels. The exponential kernel is the Matérn 1/2 kernel, $C_{1/2}(x) = \sigma^2 e^{-x}$ ($p = 0$), while $p = 1$ is relevant to stellar variability, which we describe next.

- **Matérn 3/2 kernel.** The Matérn 3/2 ($p = 1$) kernel is given by:

$$k_{3/2}(\tau) = C_{3/2}(\tau/\tau_0) = (1+v)e^{-v}, \tag{42}$$

where $v = \sqrt{3}|\tau/\tau_0|$. This kernel has a Fourier transform[2] of

$$\hat{k}_{3/2}(\omega) = \frac{\tau_0 6^{3/2}}{\pi^{1/2}}(3+\tau_0^2\omega^2)^{-2}, \tag{43}$$

where $\omega = 2\pi f$. This kernel may be well approximated by the sum of two exponential kernels:

$$k_{3/2}(\tau) \approx (1-b)e^{-v} + be^{-v\frac{b-1}{b}}, \tag{44}$$

where $b \gg 1$ is a dimensionless parameter; this approximation is exact in the limit $b \rightarrow \infty$. For large values of b, the maximum error of the approximation scales as $2/(e^2 b) \approx 0.27/b$; for $b = 100$, the error in this approximation is $< 0.3\%$. Note that the two exponential terms have opposite signs, which may lead to numerical instability for values of b that are large.

One advantage of the Matérn 3/2 kernel is that it is smooth across $\tau = 0$ since its derivative is zero at $\tau = 0$. The exponential kernel *does not* have this property, which leads to qualitatively different noise properties on small timescales due to

[2] Using the normalization convention: $\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t)e^{i\omega t}dt$

the sharpness of the exponential kernel at $\tau = 0$. The Matérn 3/2 kernel turns out to be similar to the observed properties of stellar activity and granulation, and thus the approximate formula is simultaneously physically relevant and computationally convenient. The power spectrum of a single exponential scales as f^{-2} at high frequency, while the Matérn kernel scales as f^{-4} which more closely matches the power spectrum due to activity and granulation in the Sun (Aigrain et al. 2004; **?**).

- **Matérn 5/2 kernel.** The Matérn 5/2 kernel is given by

$$k_{5/2}(t) = C_{5/2}(y) = \sigma^2(1 + y + y^2/3)e^{-y}, \tag{45}$$

where $y = \sqrt{5}|\tau/\tau_0|$. The Fourier transform is:

$$\hat{k}_{5/2}(\omega) = \frac{2\tau_0 10^{5/2}}{3\pi^{1/2}}(5 + \tau_0^2\omega^2)^{-3}. \tag{46}$$

This kernel may be approximated well by the sum of four exponential kernels

$$\begin{align} k_{5/2}(\tau) \approx & -147.6965573e^{-1.0097610y} + 85.8875034e^{-1.0621734y} \tag{47} \\ & -4.6540017e^{-0.8353745y} + 67.4630744e^{-0.9160394y}, \tag{48} \end{align}$$

which is accurate to $< 4 \times 10^{-6}$ for $y \leq 10$. The Fourier transform has a steeper dependence at large frequencies, f^{-6}, and as we are not aware of a stellar power spectrum that requires a steeper spectrum than this, we forego approximating further Matérn kernels.

- **Exponential-sine-square kernel.** This kernel is a periodic, positive valued kernel which has been used to model quasi-periodic stellar variability due to star spots. It is given by $E(w, \Gamma) = e^{-\Gamma \sin^2 w}$ where $w = \frac{\pi}{P}|\tau|$, which may be expanded as a series in $\cos{(2kw)} = \cos\frac{2\pi}{P}k|\tau|)$, with $k \in \mathbb{N}$. We have expanded this to $k = 4$, and fit for the coefficients, giving:

$$E(w, \Gamma) \approx \sum_{k=0}^{4}\left(c_{k,1} + c_{k,2}e^{-c_{k,4}\Gamma^{c_{k,3}}}\right)\cos{(2kw)}, \tag{49}$$

with $c_{0,1} = 0.24999, c_{0,2} = 0.752803, c_{0,3} = 0.960477, c_{0,4} = 0.643393$ for $k = 0$, $c_{1,1} = 0.450851, c_{1,2} = -0.445338, c_{1,3} = 0.960477, c_{1,4} = 1.18102$ for $k = 1$, $c_{2,1} = 0.206593, c_{2,2} = -0.207361, c_{2,3} = 1.65902, c_{2,4} = 0.210641$ for $k = 2$, $c_{3,1} = 0.0728449, c_{3,2} = -0.0729381, c_{3,3} = 2.43149, c_{3,4} = 0.0471224$ for $k = 3$, and $c_{4,1} = 0.019204, c_{4,2} = -0.0192129, c_{4,3} = 3.27977, c_{4,4} = 0.0114198$ for $k = 4$. This approximation is accurate to better than 0.6% for $\Gamma < 3$, with a standard deviation of 0.1%.

Examples of the foregoing kernels and their approximations are shown in Figure 1. It is clear from this figure that these standard kernels are well approximated by the sums of between two and five damped cosine kernels, demonstrating that these form an adequate set for modeling a wide range of kernel behaviors.

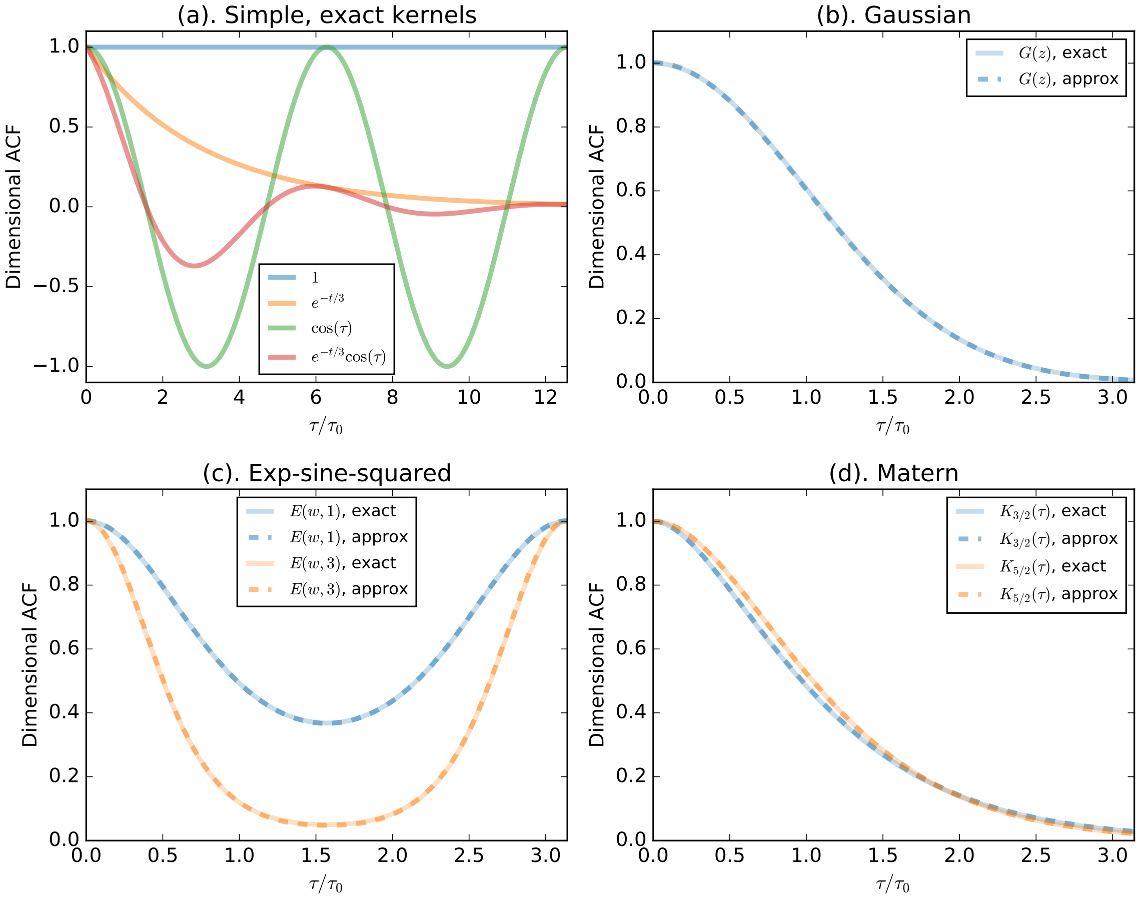

Figure 1. Approximation of various kernels with the sums of damped cosines. (a). Kernels that may be expressed exactly: constant (dark grey), exponential (cyan), cosine (olive), and damped cosine (salmon). (b). Approximation of the squared-exponential (Gaussian) kernel (pink) with three damped cosine kernels (magenta). (c). The approximation of the exponential-sine-squared kernel with $\Gamma = 1$ (blue) and $\Gamma = 3$ (green). (d). Two Matérn kernels, $k_{3/2}$ (yellow/brown) and $k_{5/2}$ (violet). The Matérn 3/2 kernel is approximated with $b = 100$. The exact kernels are solid, while approximate are dashed.

A standard practice in constructing kernels for solving problems with Gaussian processes is to add or multiply the preceding standard kernel components. In the case of damped cosine kernels, the product of two kernels is:

$$k_j(\tau)k_k(\tau) = \frac{1}{2}e^{-(c_j+c_k)\tau}\left((a_ja_k + b_jb_k)\cos\left[(d_j - d_k)\tau\right] + (a_jb_k - a_kb_j)\sin\left[(d_j - d_k)\tau\right]\right. \tag{50}$$

$$\left. + (a_ja_k - b_jb_k)\cos\left[(d_j + d_k)\tau\right] - (a_jb_k - a_kb_j)\sin\left[(d_j + d_k)\tau\right]\right), \tag{51}$$

which is the sum of two damped sinusoidal kernels. Thus, only addition is necessary to construct new kernels from the product of damped sinuosidal kernels.

The upshot is that the damped sinusoid is an expressive basis kernel that may be used to represent a wide range of autocorrelation functions relevant for stellar variability, and more. We recommend using a series of sums of damped sinusoids (in which some parameters may be fixed to specified values to represent particular kernels) to model different amplitudes and timescales of variability. If an initial estimate of the power spectrum or autocorrelation function is available, then these may be fit with this series to initialize the coefficients of the damped sinusoid kernel components. Since a Gaussian Process must be positive definite, we recommend modeling the logarithm of a_j, b_j, c_j, and d_j to enforce this to be the case. One caveat is that to produce certain kernels, a negative value of a_j or b_j is needed (such as the Matérn 3/2 kernel). In this case, the value of the sum of two coefficients may be sampled from a logarithm, while the individual values are allowed to be negative; this will enforce the positive definite constraint, but may allow for other constraints, such as an autocorrelation function which has a derivative of zero at $\tau = 0$.

11. 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),

All of the code used in this project is available from `https://github.com/dfm/GenRP` under the MIT open-source software license. This code (plus some dependencies) 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 `3361743` (2016-12-28).

12. SOLVING FOR LOG LIKELIHOOD WITH EXTENDED MATRIX

Here we summarize Ambikasaran (2015), and then point to appendix for how to do this with oscillating kernels.

13. TRAINING

May use derivative of likelihood + BFGS-B to optimize kernel parameters.

Q: How do we determine how many kernels to use?

It is a pleasure to thank ... 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:

APPENDIX

A. IMPLEMENTATION DETAILS

Here we adapt the generalized Rybicki-Press (GRP) algorithm to handle periodic kernels. We follow closely the notation and equations given in Ambikasaran (2015). We start with the observation that a damped sinusoid correlation function may be decomposed into the sum of two exponential functions:

$$k_j(\tau) = a_j e^{-c_j \tau} \cos(-d_j \tau) + b_j e^{-c_j \tau} \sin(-d_j \tau) \tag{A1}$$

$$= \frac{1}{2}(a_j + i b_j) \left[e^{-(c_j + i d_j)\tau} + e^{-(c_j - i d_j)\tau} \right], \tag{A2}$$

where $c_j + i d_j$ is the complex kernel parameter and $d_j = 2\pi/P$ is the frequency of oscillation of the kernel with period P.

Now, the damped sinusoid kernal may be included in the GRP algorithm using the above two exponential kernels. This requires the algorithm to use complex arithmetic, which has several disadvantages: 1). complex arithmetic is more expensive by a factor of 2-3; 2). the complex conjugate equation is redundant as it turns out that $l_2 = l_1^*$, $r_2 = r_1^*$, and $x_i = x_i^*$ (i.e. x_i is real), meaning that the same equations are being solved twice for the real and complex components; and 3). if the derivative of the likelihood is computed with automatic differentiation (AD), most AD algorithms require real variables.

Given these drawbacks, we instead modify the GRP algorithm to use a combination of two real components rather than two complex components; this addresses all three of these drawbacks simultaneously. The one cost is that the banded extended matrix has a bandwidth which is larger by one, both above and below the diagonal.

Rather than using $c_j + id_j$ and its complex conjugate as components of the kernel, we instead find the real and imaginary components of the complex equation by taking the sum and difference of the $c_j + id_j$ and $c_j - id_j$ equations (divided by 2 and $2i$, respectively), and then discard the $c_j - id_j$ equation, which is redundant. This keeps the same number of additional equations for each (c_j, d_j) with non-zero imaginary component (four additional equations in the extended matrix), but converts the complex equations to real, thus avoiding complex arithmetic, while the kernel components with $d_j = 0$ still only have two additional components in the extended matrix. The extended matrix adds an extra band above and below the diagonal when the $c_j + id_j$ and $c_j - id_j$ equations are combined since this mixes the components of the original equations which are shifted by one column from one another. We will enumerate the number of $d_j = 0$ cases as p_0, while the total number of c_j's is still p. This gives a total number of equations to be solved with the extended matrix of $N_{ex} = (4(p - p_0) + 2p_0 + 1) \times (N - 1) + 1$, where N is the number of data points.

For the complex values with $d_j \neq 0$, the equations are modified as follows. We introduce complex auxiliary vectors $l_k = l_k^R - il_k^I$, with $k \in \{1, 2, ..., N\}$ and $l_1 = 0$; $r_k = r_k^R - ir_k^I$, with $k \in \{2, ..., N\}$ and $r_{N+1} = 0$. Note that we have defined the imaginary component as being negative for aesthetic reasons: this yields a symmetric extended matrix (unfortunately the extended matrix has negative eigenvalues, making it non positive-definite, so that Cholesky decomposition cannot be applied; nevertheless we prefer to use a symmetric set of equations, and since the values of these variables are discarded in the end, their sign is irrelevant). The length of the vectors l_k and r_k is the number of kernel components, p; we denote the individual elements of these vectors as $l_{k,j}$ and $r_{k,j}$, $j \in \{1, ..., p\}$. Note that the first p_0 elements we take to be the kernels with $d_j = 0$, and for these we do not include the imaginary component equations, which are unnecessary and only add extra computation time. We allow each vector γ_k to have real and imaginary components as well: $\gamma_k = \left[\exp\left(-(c_1 + id_1)t_{k,k+1}\right) ... \exp\left(-(c_p + id_p)t_{k,k+1}\right)\right]^T = \gamma_k^R + i\gamma_k^I$. These elements are given by $\gamma_{k,j}^R = e^{-c_j t_{k,k+1}} \cos\left(d_j t_{k,k+1}\right)$ and $\gamma_{k,j}^I = -e^{-c_j t_{k,k+1}} \sin\left(d_j t_{k,k+1}\right)$. Equation (61) becomes:

$$\sum_{j=1}^{p} \left(\frac{a_j}{2} l_{k,j}^R + \frac{b_j}{2} l_{k,j}^I \right) + \frac{d}{2} x_k + \sum_{j=1}^{p} \left[\gamma_{k,j}^R r_{k+1,j}^R + \gamma_{k,j}^I r_{k+1,j}^I \right] = \frac{h_k}{2}, \tag{A3}$$

for $j \in \{1, ..., p\}$ and $k \in \{1, ..., N\}$, where h_j is the jth data point. There is no imaginary component to equation (61) in Ambikasaran (2015).

Equation (60) in Ambikasaran (2015) has real and imaginary components:

$$\gamma_{k,j}^R l_{k,j}^R + \gamma_{k,j}^I l_{k,j}^I + \gamma_{k,j}^R x_k - l_{k+1,j}^R = 0, \tag{A4}$$

$$\gamma_{k,j}^I l_{k,j}^R - \gamma_{k,j}^R l_{k,j}^I + \gamma_{k,j}^I x_k + l_{k+1,j}^I = 0, \tag{A5}$$

with $j \in \{1, ..., p\}$ and $k \in \{1, ..., N-1\}$ (note that we have shifted the indices of this equation by one from the indexing using in Ambikasaran 2015).

Finally, equation (59) in Ambikasaran (2015) has real and imaginary components:

$$-r^R_{k,j} + \frac{a_j}{2} x_k + \gamma^R_{k,j} r^R_{k+1,j} + \gamma^I_{k,j} r^I_{k+1,j} = 0, \tag{A6}$$

$$r^I_{k,j} + \frac{b_j}{2} x_k + \gamma^I_{k,j} r^R_{k+1,j} - \gamma^R_{k,j} r^I_{k+1,j} = 0, \tag{A7}$$

with $j \in \{1, ..., p\}$ and $k \in \{2, ..., N\}$. Note that for the p_0 real components we can drop the variables $r^I_{k,j}$ and $l^I_{k,j}$ as these are zero, and we can drop the imaginary components of equations (59) and (60) in Ambikasaran (2015) as well for these, which reduces the total number of equations to solve.

With these additional equations, we can solve for the likelihood of sums of damped sinusoid in $O(N)$ operations, but with slightly more computational expense due to having 4 extra variables for each damped sinusoid with non-zero ω, and due to having a slightly larger bandwidth due to the cross terms between the real and imaginary variables. The off-diagonal components are $p_0 + 2(p - p_0) + 2$ above and below the diagonal, for a total bandwidth of $2p_0 + 4(p - p_0) + 5$. In the case in which $p_0 = p$, so that $d_j = 0$, the off-diagonal components revert to $p + 1$ above and below the diagonal, giving a bandwidth of $2p + 3$ (the same as the GRP algorithm).

As with the GRP method, we can build an extended matrix which is real and symmetric, and which may be solved with a banded solver, as in Press et al. (1992). The determinant of this extended matrix may be computed from the LU decomposition, and it is related to the determinant of the kernel by a factor of 2^N (this is due to dividing each equation by 2 to yield a symmetric matrix). We arrange the auxiliary variables with the $d_j = 0$ terms first, followed by the complex variables, yielding the vector $x_{ex} = [x_1, \{r^R_{2,j}, l^R_{2,j}; j = 1, ..., p_0\}, \{r^R_{2,j}, r^I_{2,j}, l^R_{2,j}, l^I_{2,j}; j = p_0 + 1, ..., p\}, x_2, ..., x_N]$. The right hand side is given by: $y_{ex} = [h_1/2, \{0; j = 1, ..., p_0 + 2(p - p_0)\}, h_2/2, ..., h_N/2]$. The extended matrix, A_{ex}, is constructed by inserting the coefficients of equations A3, A4, and A6 into a matrix of dimensions $(2p_0 + 4(p - p_0) + 5) \times N_{ex}$. The solution is found by solving $A_{ex} x_{ex} = y_{ex}$ with LU decomposition and back-substitution with a banded algorithm.

The matrix has a similar structure as the real case constructed by (Ambikasaran 2015), but the imaginary and real components mix (this is due to the representation of imaginary numbers as 2×2 matrices). Figure A1 shows an example of the structure of the extended matrix for two damped sinusoids with non-zero ω values. This is analagous to the real case with $p = 2$ which also has four additional equations for each row of the original kernel. This matrix shows that there is an extra band above and below the diagonal (compare with Figure 2 of Ambikasaran 2015).

Should I give some examples of this? Some pseudocode?

REFERENCES

Aigrain, S., Favata, F., & Gilmore, G. 2004, A&A, 414, 1139

Ambikasaran, S. 2015, Numer. Linear Algebra Appl., 22, 1102

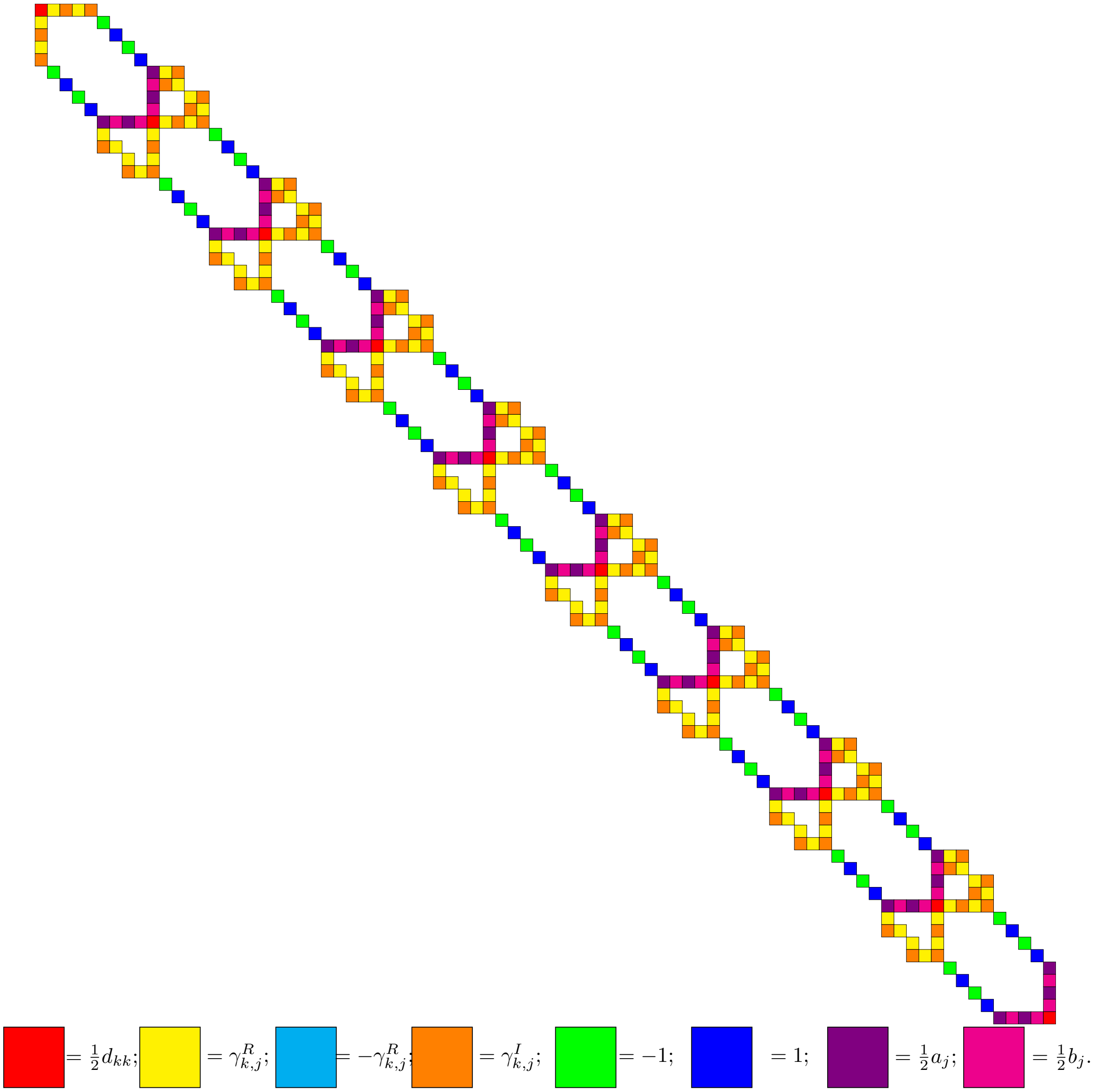

Figure A1. Pictorial description of the extended sparse matrix where $N = 10$, $p_0 = 0$, and $p = 2$, following the example shown in Ambikasaran (2015).

Anderson, E. R., Duvall, Jr., T. L., & Jefferies, S. M. 1990, ApJ, 364, 699

Dörrie, H. 1965, 100 Great Problems of Elementary Mathematics: Their History and Solution, Dover Books on Mathematics Series, §24, (Dover Publications), 112–116

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

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

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

Zu, Y., Kochanek, C. S., & Peterson, B. M. 2011, ApJ, 735, 80