A&A 638, A95 (2020)
https://doi.org/10.1051/0004-6361/201936906

Astronomy & Astrophysics

Efficient modeling of correlated noise

II. A flexible noise model with fast and scalable methods*

J.-B. Delisle, N. Hara**, and D. Ségransan

Département d'astronomie, Université de Genève, 51 chemin des Maillettes, 1290 Versoix, Geneva, Switzerland
e-mail: jean-baptiste.delisle@unige.ch

Received 11 October 2019 / Accepted 10 April 2020

ABSTRACT

Correlated noise affects most astronomical datasets and to neglect accounting for it can lead to spurious signal detections, especially in low signal-to-noise conditions, which is often the context in which new discoveries are pursued. For instance, in the realm of exoplanet detection with radial velocity time series, stellar variability can induce false detections. However, a white noise approximation is often used because accounting for correlated noise when analyzing data implies a more complex analysis. Moreover, the computational cost can be prohibitive as it typically scales as the cube of the dataset size. For some restricted classes of correlated noise models, there are specific algorithms that can be used to help bring down the computational cost. This improvement in speed is particularly useful in the context of Gaussian process regression, however, it comes at the expense of the generality of the noise model. In this article, we present the S + LEAF noise model, which allows us to account for a large class of correlated noises with a linear scaling of the computational cost with respect to the size of the dataset. The S + LEAF model includes, in particular, mixtures of quasiperiodic kernels and calibration noise. This efficient modeling is made possible by a sparse representation of the covariance matrix of the noise and the use of dedicated algorithms for matrix inversion, solving, determinant computation, etc. We applied the S + LEAF model to reanalyze the HARPS radial velocity time series of the recently published planetary system HD 136352. We illustrate the flexibility of the S + LEAF model in handling various sources of noise. We demonstrate the importance of taking correlated noise into account, and especially calibration noise, to correctly assess the significance of detected signals.

Key words. methods: data analysis – methods: statistical – methods: analytical – planets and satellites: general

1. Introduction

Astronomical datasets, like most datasets, are contaminated by various sources of noise, such as photon noise, the intrinsic variability of the object of interest, contamination by the Earth's atmosphere, instrumental noise, etc. While the photon noise is purely white (i.e., uncorrelated), most of the other sources of noise have temporal or spatial correlations. When neglected, these correlations can lead to spurious signal detections.

In the context of exoplanet detection with radial velocity time series, stellar variability could induce signals that mimic planetary signatures (e.g., Queloz et al. 2001). The mitigation of stellar variability has become a major subject in planet search studies and is now routinely achieved by modeling it as correlated Gaussian noise. Adopting such models significantly improves the robustness of planet detection (e.g., Haywood et al. 2014; Rajpaul et al. 2015; Faria et al. 2016). Correlated noise also affects the determination of a planet's parameters and can induce, in particular, spurious eccentricities when it is not properly accounted for (e.g., Hara et al. 2019).

In many cases, the physical processes inducing correlated noise cannot be modeled precisely but qualitative properties, typical timescales, and amplitudes can be estimated. Thus, a common approach is to use simple parametric noise models. For a time series of size n with observations taken at times t_i ($i < n$), the covariance matrix of the noise is typically modeled as:

$$C_{i,j} = \delta_{i,j}\sigma_i^2 + K(t_i, t_j), \tag{1}$$

where σ_i are individual errorbars (e.g., photon noise) and K is the kernel of the correlated noise. The noise is often assumed to be stationary, such that $K(t_i, t_j)$ only depends on $|t_i - t_j|$,

$$K(t_i, t_j) = k(|t_i - t_j|). \tag{2}$$

A simple, widespread model assumes the correlation to decrease exponentially with time, with a timescale of τ,

$$k(\Delta t) = \sigma_{\text{corr.}}^2 \mathbf{e}^{-\frac{\Delta t}{\tau}}, \tag{3}$$

but it is sometimes chosen to decrease as a squared exponential (Schwarzenberg-Czerny 1991),

$$k(\Delta t) = \sigma_{\text{corr.}}^2 \mathbf{e}^{-\frac{\Delta t^2}{2\tau^2}}, \tag{4}$$

or other similar functions. Slightly more complex models have also been proposed, for instance, quasiperiodic kernels, such as that of Haywood et al. (2014)

$$k(\Delta t) = \sigma_{\text{corr.}}^2 \exp\left(-\frac{\Delta t^2}{2\tau^2} - \frac{2}{\eta}\sin^2\left(\frac{\pi\Delta t}{P_{\text{rot.}}}\right)\right), \tag{5}$$

* We provide an open-source reference implementation of the S + LEAF model, the spleaf package (C library with python wrappers), available at https://gitlab.unige.ch/jean-baptiste.delisle/spleaf

** NCCR CHEOPS fellow.

which allow for a more flexible modeling of the underlying physical processes.

In the case of a poorly understood noise source, the choice of a kernel is somewhat arbitrary but nonetheless, it should be governed by the qualitative properties that the noise is expected to present (typical timescales, periodicities, etc.). For instance, quasiperiodic kernels are well-suited to model the radial velocity signal induced by stellar spots coming in and out of view due to the rotation of the star (see Haywood et al. 2014). Even if the connection to the exact physics of the process is loose, the qualitative properties of quasiperiodic kernels are sufficient to bring a significant improvement in detection reliability.

While correlated noise models improve detection robustness, they might be prohibitive in terms of computational cost and memory footprint. Indeed, for a dataset of size n, the covariance matrix of the noise is of size $n \times n$. In the general case, the memory footprint of storing C is thus $O\left(n^2\right)$. Then some operations must be performed with this matrix to compute useful quantities (such as the χ^2 or the likelihood of a model). The computational cost of these operations (e.g., inversion, dot product, determinant) typically scales as $O\left(n^2\right)$ to $O\left(n^3\right)$ in the general case. These scalings make a correct modeling of the noise intractable for large datasets. To address this issue, Ambikasaran (2015) and Foreman-Mackey et al. (2017) proposed a flexible parametric noise model, which allow a linear scaling of the memory footprint and computational cost of the correlated noise. This so-called *celerite* model is capable of handling a mixture of quasiperiodic covariance kernels of the form:

$$k(\Delta t) = \sum_{s<n_\mathrm{c}} \left(a_s \cos(\nu_s \Delta t) + b_s \sin(\nu_s \Delta t)\right) \mathbf{e}^{-\lambda_s \Delta t}, \tag{6}$$

where n_c is an arbitrarily high number of components in the model. This model has the property to be semiseparable, which allows a scaling of the computational cost as $O\left(nn_\mathrm{c}^2\right)$ (Ambikasaran 2015). It is similar to the quasiperiodic kernel of Haywood et al. (2014) which is detailed in Eq. (5). The *celerite* model is well-suited to represent stellar signals modulated by the rotation period of the star (e.g., Foreman-Mackey et al. 2017). It has been used, in particular, for the analysis of radial velocity and photometric time series.

The star is not the only source of noise in the data. Instruments also introduce a correlated signature. For instance, for precise radial velocity time series (and in other fields), the instrument must be calibrated periodically, typically once per night. Several scientific measurements might use the same calibration and, therefore, share the same calibration noise. The covariance matrix of the calibration noise is then block-diagonal with the blocks corresponding to each calibration (each night). This calibration noise is not stationary and, thus, it is not well represented by the *celerite* model (see Eq. (6)). More generally, when considering various sources of noise together, the complete covariance matrix might present quasiperiodic components and sparse (block diagonal, banded, etc.) components. While efficient dedicated algorithms exist for both quasiperiodic (semiseparable) and sparse covariance matrices, they cannot be applied in a straightforward way for a mixture of both.

In this article, we extend the method described by Foreman-Mackey et al. (2017) to correlated noise with a semiseparable component plus a sparse component. We introduce the notion of LEAF matrices, a general class of sparse, "close to diagonal" symmetric matrices encompassing banded, block-diagonal, staircase matrices, etc. Our complete model, which we call the S + LEAF model, is the sum of a semiseparable component and a LEAF component.

In Sect. 2, we present the S + LEAF correlated noise model and dedicated algorithms. In Sect. 3, we illustrate our methods using the HARPS radial velocities of HD 136352. We discuss our results in Sect. 4. We provide an open-source reference implementation of S + LEAF matrices and related algorithms as a C library with python wrappers[1].

2. The S + LEAF noise model

The likelihood (i.e., the probability of the data assuming a given model is correct) is a common tool for assessing the agreement of a given model to a dataset. In a Bayesian approach, the quantity of interest is the posterior probability (probability of a model given the data), but the computation of the likelihood is still required as an intermediate step. In this section, we describe the S + LEAF noise model and dedicated algorithms which allow, in particular, for the efficient computation of the likelihood and its derivatives.

In Sect. 2.1, we introduce notations and describe the computation of the likelihood in the general case. We define S + LEAF matrices in Sect. 2.2, and we present dedicated algorithms for S + LEAF matrices in Sect. 2.3.

2.1. Likelihood computation, general case

Let us assume that a given dataset y_i $(0 \leq i < n)$ can be modeled with a deterministic component (the model) m with parameters θ, and a correlated Gaussian noise component ϵ with parameters α:

$$\begin{aligned} &y_i = m_i(\theta) + \epsilon_i, \\ &\epsilon \sim \mathcal{G}(0, C(\alpha)). \end{aligned} \tag{7}$$

The log-likelihood of a given set of parameters (θ, α) is read as:

$$\begin{aligned} \ln \mathcal{L}(\theta, \alpha) &= \ln p(y|\theta, \alpha) \\ &= -\frac{1}{2}\Big(y - m(\theta)\Big)^\mathrm{T} C^{-1}(\alpha)\Big(y - m(\theta)\Big) \\ &\quad - \frac{1}{2} \ln \det\Big(2\pi C(\alpha)\Big), \end{aligned} \tag{8}$$

where $C(\alpha)$ is the $n \times n$ covariance matrix of the correlated noise ϵ.

The computational cost of evaluating the log-likelihood obviously depends on the cost of evaluating the model $m(\theta)$. However, once the model is obtained, we still have to compute the $\chi^2 = r^\mathrm{T} C^{-1} r$ (where r represents the residuals, $r = y - m$) and the determinant of C.

An efficient and robust way to compute the log-likelihood in the general case is to compute the Cholesky decomposition of C as an intermediate step. By definition, the covariance matrix C is symmetric, positive, and definite. It could, in principle, be singular (only semi-definite) but this would mean that some almost-certain affine relation exists in the noise component. This almost-certain relation could thus be included in the deterministic part of the model. Assuming C to be invertible (non-singular), its Cholesky decomposition can be read as:

$$C = LDL^\mathrm{T}, \tag{9}$$

[1] Available at https://gitlab.unige.ch/jean-baptiste.delisle/spleaf

where D is diagonal and L is lower triangular with ones on the diagonal. The classical Cholesky decomposition is actually $C = \Lambda\Lambda^{\mathrm{T}}$, where $\Lambda = L\sqrt{D}$ is also lower triangular. However, we use the alternative form of Eq. (9) throughout the article since this notation is more convenient in our case. The computational cost of the Cholesky decomposition is $O\left(n^3\right)$ in the general case. Once the Cholesky decomposition is obtained, the determinant of C is easily computed (in $O(n)$) since $\ln\det C = \ln\det D = \sum_i \ln D_i$. The computation of the χ^2 is performed in $O\left(n^2\right)$ in the general case by first solving $u = L^{-1}r$ and then computing $u^{\mathrm{T}}Du$.

2.2. Symmetric S + LEAF covariance matrix

A common method for improving the computational cost and memory footprint of correlated noise models is to obtain a sparse representation of the covariance matrix and to then use dedicated algorithms for solving, computing the determinant, and other functions that make use of this sparsity. For instance, dedicated representations and algorithms for banded matrices, block-diagonal matrices, etc., exist, allowing for the linear scaling in n of the computational cost and footprint of the model ($O(\alpha n)$, where α depends on the bandwidth, block size, etc.). In Delisle et al. (2018), the covariance matrix was truncated and approximated by a banded matrix. This representation improved the computational speed of the Markov chain Monte Carlo (MCMC) algorithm used to compute the posterior densities of the orbital elements and noise parameters.

2.2.1. LEAF matrix

Here we introduce a general class of sparse, "close to diagonal" matrices, called LEAF matrices, that encompasses banded, block-diagonal, staircase matrices, etc. A symmetric LEAF matrix F must verify:

$$F_{i,j} = F_{j,i} = 0 \quad \text{for } j < i - b_i, \tag{10}$$

where b_i is the number of non-zero entries left to the diagonal at line i. A sketch of a symmetric LEAF matrix is shown in Fig. 1.

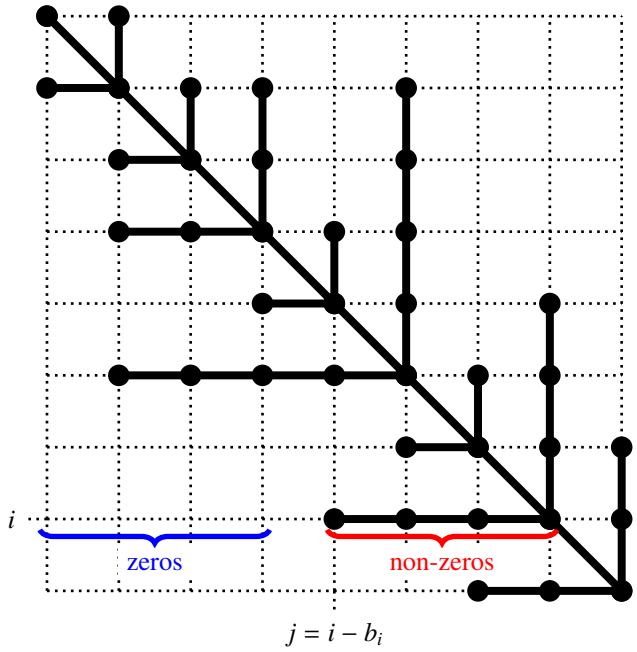

Fig. 1. Sketch of a symmetric LEAF matrix.

2.2.2. Semiseparable matrix

For efficient computations (typically linear in n), the covariance matrix does not need to be sparse itself, but it should be expressed as a function of sparse matrices (sum, product, inverse, etc.). For instance, Rybicki & Press (1995) showed that exponential matrices, defined as:

$$C_{i,j} = \mathbf{e}^{-\lambda\Delta t_{i,j}}, \tag{11}$$

with $\Delta t_{i,j} = |t_i - t_j|$, possess a tridiagonal inverse T which can be computed directly, without requiring to compute C first (see Rybicki & Press 1995). While the covariance matrix C is not sparse, using the property $C = T^{-1}$ allows for a very efficient (i.e., in $O(n)$) representation and computation. These exponential matrices (as defined in Eq. (11)) are also a particular example of semiseparable matrices which makes it possible to obtain another sparse representation. Indeed, assuming t to be ordered increasingly, and defining $u = \mathbf{e}^{-\lambda t}$ and $v = \mathbf{e}^{\lambda t}$ (vectors of size n), C can be decomposed as:

$$C = \mathbb{1} + \mathrm{tril}\left(uw^{\mathrm{T}}\right) + \mathrm{triu}\left(vu^{\mathrm{T}}\right), \tag{12}$$

where tril (respectively triu) stands for the strictly lower (respectively upper) triangular part. The two sparse representations of the exponential matrix of Eq. (11) (i.e., tridiagonal inverse and semiseparable form) are actually linked one to the other, since the inverse of invertible tridiagonal matrices are rank one semiseparable matrices and vice versa (e.g., Vandebril et al. 2005, and references therein).

More generally, a symmetric semiseparable matrix is defined as:

$$C = \mathrm{diag}(A) + \mathrm{tril}\left(UV^{\mathrm{T}}\right) + \mathrm{triu}\left(VU^{\mathrm{T}}\right), \tag{13}$$

where $\mathrm{diag}(A)$ is the diagonal matrix built from the vector A (size n), U, and V are $(n \times r)$ matrices, and r is the rank of the semiseparable matrix C. Semiseparable matrices can represent a large class of correlated noise models. For instance, the *celerite* model (see Eq. (6)) proposed by Foreman-Mackey et al. (2017) can be represented as a semiseparable matrix of rank $r = 2n_{\mathrm{c}}$, with:

$$\begin{aligned}
A_i &= \sigma_i^2 + \sum_{s<n_{\mathrm{c}}} a_s,\\
U_{i,s} &= \mathbf{e}^{-\lambda_s t_i}\left(a_s\cos(\nu_s t_i) + b_s\sin(\nu_s t_i)\right),\\
U_{i,n_{\mathrm{c}}+s} &= \mathbf{e}^{-\lambda_s t_i}\left(a_s\sin(\nu_s t_i) - b_s\cos(\nu_s t_i)\right),\\
V_{i,s} &= \mathbf{e}^{\lambda_s t_i}\cos(\nu_s t_i),\\
V_{i,n_{\mathrm{c}}+s} &= \mathbf{e}^{\lambda_s t_i}\sin(\nu_s t_i).
\end{aligned} \tag{14}$$

The computational cost and memory footprint of a semiseparable noise model are linear in n (footprint in $O(rn)$ and cost in $O\left(r^2 n\right)$, see Ambikasaran 2015; Foreman-Mackey et al. 2017).

2.2.3. S + LEAF matrix

We define a S + LEAF matrix simply as the sum of a semiseparable and a LEAF matrix. A symmetric S + LEAF matrix takes, thus, the form of:

$$C = \mathrm{diag}(A) + \mathrm{tril}\left(UV^{\mathrm{T}}\right) + \mathrm{triu}\left(VU^{\mathrm{T}}\right) + F, \tag{15}$$

where A is a vector of size n representing the diagonal part of C; U and V are $n \times r$ matrices representing the symmetric semiseparable part of C; and F is the symmetric LEAF part of C, as defined in Eq. (10). Since the diagonal part of C is represented by the vector A, we assume the diagonal of F to be filled with zeros. As in Eq. (10), we denote by b_i the number of non-zero entries left to the diagonal, at line i of F (see also Fig. 1). The sparse matrix F can thus be stored in a compact way (i.e., storing only non-zero entries, and using its symmetry) with $\overline{b}n$ values. The memory footprint of the S + LEAF model scales as $O\left(\left(r+\overline{b}\right)n\right)$ and the computational cost as $O\left(\left(r^2+r\overline{b}+\overline{b^2}\right)n\right)$, where r is the number of components in the semiseparable part and for any vector x, $\overline{x}$ stands for the mean of x.

2.3. Likelihood computation with S + LEAF matrices

2.3.1. Cholesky decomposition

We then look for a sparse representation and an efficient computation of the matrices D and L involved in the Cholesky decomposition (see Eq. (9)) of C as defined by Eq. (15). In the case $F = 0$, Foreman-Mackey et al. (2017) showed that L can be written as:

$$L = \mathbb{1} + \mathrm{tril}\left(UW^{\mathrm{T}}\right), \tag{16}$$

where W is a new $n \times r$ matrix which need to be determined. In the case $F \neq 0$, this decomposition does not hold but we can prove that there exist a $n \times r$ matrix W and a strictly lower triangular LEAF matrix G with the same shape as F (i.e., same values of b_i), such that:

$$L = \mathbb{1} + \mathrm{tril}\left(UW^{\mathrm{T}}\right) + G. \tag{17}$$

Let us first simply assume that G is strictly lower triangular (not necessarily LEAF). In this case, the decomposition is degenerated but always exists. Replacing L by the expression of Eq. (17) in the Cholesky decomposition of C (Eq. (9)) and equating it to Eq. (15), we obtain (for $j < i$):

$$\begin{aligned}
C_{i,i} = A_i = D_i &+ \sum_{k<i}\left(\sum_s U_{i,s}W_{k,s} + G_{i,k}\right)^2 D_k \\
&= D_i + \sum_s U_{i,s}\left(\sum_t S_{i,s,t}U_{i,t} + 2Z_{i,i,s}\right) + \sum_{k<i} G_{i,k}^2 D_k, \qquad (18)\\
C_{i,j} &= \sum_s U_{i,s}V_{j,s} + F_{i,j} \\
&= \left(\sum_s U_{i,s}W_{j,s} + G_{i,j}\right)D_j \\
&\quad + \sum_{k<j}\left(\sum_s U_{i,s}W_{k,s} + G_{i,k}\right)\left(\sum_t U_{j,t}W_{k,t} + G_{j,k}\right)D_k \\
&= \sum_s U_{i,s}\left(W_{j,s}D_j + \sum_t U_{j,t}S_{j,s,t} + Z_{j,j,s}\right) \\
&\quad + G_{i,j}D_j + \sum_{k<j} G_{i,k}G_{j,k}D_k + \sum_s U_{j,s}Z_{i,j,s}, \qquad (19)
\end{aligned}$$

where S is defined following Foreman-Mackey et al. (2017),

$$S_{i,s,t} = \sum_{k<i} W_{k,s}D_k W_{k,t}, \tag{20}$$

and Z is defined as:

$$Z_{i,j,s} = \sum_{k<j} G_{i,k}D_k W_{k,s}. \tag{21}$$

We then break the degeneracy in the expression of L by identifying the terms in front of $U_{i,s}$ in Eq. (19). Thus we obtain:

$$\begin{aligned}
V_{j,s} &= W_{j,s}D_j + \sum_t U_{j,t}S_{j,s,t} + Z_{j,j,s}, \\
F_{i,j} &= G_{i,j}D_j + \sum_{k<j} G_{i,k}G_{j,k}D_k + \sum_s U_{j,s}Z_{i,j,s}. \qquad (22)
\end{aligned}$$

We deduce the following expressions for D, W, and G (for $j < i$):

$$D_i = A_i - \sum_s U_{i,s}\left(\sum_t S_{i,s,t}U_{i,t} + 2Z_{i,i,s}\right) - \sum_{k<i} G_{i,k}^2 D_k, \tag{23}$$

$$W_{i,s} = \frac{1}{D_i}\left(V_{i,s} - \sum_t S_{i,s,t}U_{i,t} - Z_{i,i,s}\right), \tag{24}$$

$$G_{i,j} = \frac{1}{D_j}\left(F_{i,j} - \sum_{k<j} G_{i,k}G_{j,k}D_k - \sum_s U_{j,s}Z_{i,j,s}\right). \tag{25}$$

From Eqs. (21) and (25), we can check by induction that $G_{i,j} = Z_{i,j,s} = 0$ for $j < i - b_i$. Therefore, G and Z have the same LEAF shape as F, which proves that the decomposition of Eq. (17) always exists.

Using this property, we are able to compute compact recursion formulas for the expression of S, Z, G, D, and W. We find that for increasing values of i and increasing values of j at i fixed (with $i - b_i \leq j < i$):

$$\begin{aligned}
&S_{0,s,t} = 0, \\
&S_{i,s,t} = S_{i-1,s,t} + W_{i-1,s}D_{i-1}W_{i-1,t} \quad (i > 0), \qquad (26)\\
&Z_{i,i-b_i,s} = 0, \\
&Z_{i,j,s} = Z_{i,j-1,s} + G_{i,j-1}D_{j-1}W_{j-1,s} \quad (j > i - b_i), \qquad (27)
\end{aligned}$$

$$G_{i,j} = \frac{1}{D_j}\left(F_{i,j} - \sum_{k=\max(i-b_i,j-b_j)}^{j-1} G_{i,k}G_{j,k}D_k - \sum_s U_{j,s}Z_{i,j,s}\right), \tag{28}$$

$$D_i = A_i - \sum_s U_{i,s}\left(\sum_t S_{i,s,t}U_{i,t} + 2Z_{i,i,s}\right) - \sum_{k=i-b_i}^{i-1} G_{i,k}^2 D_k, \tag{29}$$

$$W_{i,s} = \frac{1}{D_i}\left(V_{i,s} - \sum_t S_{i,s,t}U_{i,t} - Z_{i,i,s}\right). \tag{30}$$

While S is a $n \times r \times r$ tensor, it is not necessary to keep all its values in memory, and S can be stored as a $r \times r$ matrix which is updated in place for increasing values of i. The same reasoning holds for Z, which can be stored as a vector of size r, and updated for increasing values of i and j. However, if the backpropagation of the gradient is required, all the values of S and Z should be stored for reasons of stability and performance (see Sect. 2.3.4). In this case, the memory footprint of the S + LEAF model increases but remains linear in n (i.e., $O\left(\left(r+\overline{b}\right)rn\right)$ instead of $O\left(\left(r+\overline{b}\right)n\right)$).

2.3.2. Computing the determinant and solving

As explained in Sect. 2.1, once the Cholesky decomposition of the covariance matrix C is known, we need to compute its determinant and solve for $x = L^{-1}y$ to compute the likelihood of a

set of parameters. The determinant is trivially obtained in $O(n)$ operations,

$$\ln\det(C) = \ln\det(D) = \sum_i \ln D_i. \tag{31}$$

We can then describe how to solve for $x = L^{-1}y$ (with L defined as in Eq. (17)). Since $y = Lx$, we have:

$$\begin{aligned} y_i &= x_i + \sum_{j<i}\left(\sum_s U_{i,s}W_{j,s} + G_{i,j}\right)x_j \\ &= x_i + \sum_s U_{i,s}f_{i,s} + \sum_{j=i-b_i}^{i-1} G_{i,j}x_j, \end{aligned} \tag{32}$$

with f defined as in Foreman-Mackey et al. (2017),

$$f_{i,s} = \sum_{j<i} W_{j,s}x_j. \tag{33}$$

We thus obtain the following recursion formulas for increasing values of i:

$$\begin{aligned} &f_{0,s} = 0, \\ &f_{i,s} = f_{i-1,s} + W_{i-1,s}x_{i-1} \quad (i > 0), \end{aligned} \tag{34}$$

$$x_i = y_i - \sum_s U_{i,s}f_{i,s} - \sum_{j=i-b_i}^{i-1} G_{i,j}x_j. \tag{35}$$

As for the Cholesky factorization, the values of f can be stored in a vector of size r and updated in place for increasing values of i, except in the case where the backpropagation of the gradient is required (see Sect. 2.3.4). The computational cost of this solving is in $O\left(\left(r+\overline{b}\right)n\right)$.

While it is not needed in the calculation of the likelihood, the computation of the dot product $y = Lx$ is very similar to the solving problem ($x = L^{-1}y$). For increasing values of i, we compute:

$$\begin{aligned} &f_{0,s} = 0, \\ &f_{i,s} = f_{i-1,s} + W_{i-1,s}x_{i-1} \quad (i > 0), \end{aligned} \tag{36}$$

$$y_i = x_i + \sum_s U_{i,s}f_{i,s} + \sum_{j=i-b_i}^{i-1} G_{i,j}x_j. \tag{37}$$

Similar recursion formulas for the dot product $y = L^{\mathrm{T}}x$ and the solving of $x = L^{-\mathrm{T}}y$ are easily obtained.

2.3.3. Overflows and preconditioning

As noted by Ambikasaran (2015); Foreman-Mackey et al. (2017), a naive computer implementation of exponential semiseparable matrices can lead to numerical underflows and overflows. Indeed, the separation of the exponential $\mathbf{e}^{-\lambda_s|t_i-t_j|}$ in $U_{i,s} = \mathbf{e}^{-\lambda_s t_i}$ and $V_{j,s} = \mathbf{e}^{\lambda_s t_j}$ exhibits very interesting theoretical properties (semiseparable matrix) but in practical applications, $\lambda_s t_i$ and $\lambda_s t_j$ can reach values that are much larger than $\lambda_s|t_i - t_j|$, which causes underflows for U and overflows for V.

To circumvent this numerical issue, we follow Foreman-Mackey et al. (2017) and introduce the $(n-1)\times r$ preconditioning matrix ϕ, and the preconditioned matrices $\tilde{U}$ and $\tilde{V}$, such that

$$U_{i,s}V_{j,s} = \tilde{U}_{i,s}\tilde{V}_{j,s}\prod_{k=j}^{i-1}\phi_{k,s}. \tag{38}$$

For instance, in the case of the *celerite* model – Eqs. (6) and (14) – Foreman-Mackey et al. (2017) proposed the following preconditioning:

$$\begin{aligned} \tilde{U}_{i,s} &= a_s\cos(\nu_s t_i) + b_s\sin(\nu_s t_i), \\ \tilde{U}_{i,n_c+s} &= a_s\sin(\nu_s t_i) - b_s\cos(\nu_s t_i), \\ \tilde{V}_{i,s} &= \cos(\nu_s t_i), \\ \tilde{V}_{i,n_c+s} &= \sin(\nu_s t_i), \\ \phi_{i,s} &= \phi_{i,n_c+s} = \mathbf{e}^{-\lambda_s(t_{i+1}-t_i)}, \end{aligned} \tag{39}$$

which avoids the computation of exponentials with large exponents. All the algorithms presented above (Cholesky decomposition, dot product and solving) can be adapted to take into account this preconditioning. We refer the reader to Appendix A for more details.

2.3.4. Efficient computation of the likelihood derivatives

Once the model is chosen, we typically need to determine a point estimate or the posterior distribution of the parameters. In order to use efficient optimization or exploration algorithms, it might be useful to compute the gradient of the log-likelihood (Eq. (8)) with respect to the model parameters (θ) and the noise parameters (α). Foreman-Mackey (2018) provided gradient backpropagation algorithms for the Cholesky decomposition, dot product, and solving problem in the case of semiseparable matrices ($F = 0$ in our notations). These algorithms allow to very efficiently compute (in $O\left(r^2n\right)$, see Foreman-Mackey 2018) the gradient of the log-likelihood using analytical formulas. The generalization of this method to S + LEAF matrices is straightforward and we provide more details in Appendix B.

3. Application to the analysis of radial velocities

In this section, we illustrate the use of the S + LEAF noise model by reanalyzing the HARPS radial velocity time series of HD 136352 (see Udry et al. 2019). The star HD 136352 is a quiet G4V star known to host three super-Earth planets, at periods of 11.5824 d, 27.5821 d, and 107.6 d, and with minimum masses of 4.8, 10.8, and 8.6 $M_\oplus$ respectively (see Udry et al. 2019). These results were obtained by binning the data and only searching for planets with periods above 1 d. This is a common practice that allows to damp many instrumental and stellar short-term variations (Dumusque et al. 2011). However, it does not allow us to characterize these short-term variations and to fully correct for them. Moreover, binning the data could significantly damp the amplitude of short period planets. Here we reanalyze the raw radial velocities and do not restrict our study to periods above 1 d.

The radial velocities of HD 136352 taken with HARPS consist of 648 points, taken over almost 11 yr (2004–2015), and spread over 238 distinct nights. The number of points per night varies between one and ten, with an average of 2.7 points per night.

We describe the different noise models we use for our study in Sect. 3.1 and present our reanalysis of the HD 136352 system in Sect. 3.2.

3.1. Noise models

To illustrate the role of each component in our S + LEAF noise model, we analyze the data using five different noise models:

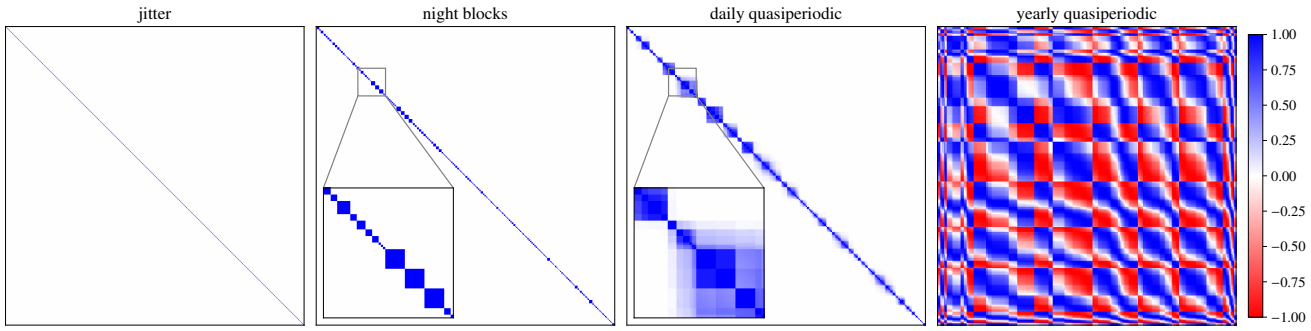

Fig. 2. Shapes of the four components of the noise models used for the analysis of the HD 136352 system (Sect. 3).

1. diag.: a diagonal matrix, with the observational errorbars σ_i plus a jitter term ($\sigma_{\rm jit.}$) added in quadrature (same value for all data points)

$$C_{i,j} = (\sigma_i^2 + \sigma_{\rm jit.}^2)\delta_{i,j}; \tag{40}$$

2. bin.: same as diag. but using nightly binned radial velocity data;
3. *celerite*: same as diag. plus quasiperiodic terms at 1 d and 1 yr,

$$C_{i,j} = (\sigma_i^2 + \sigma_{\rm jit.}^2)\delta_{i,j} + \sigma_{\rm d}^2 \mathbf{e}^{-0.1|t_i - t_j|} \cos\left(2\pi \frac{t_i - t_j}{1\ \rm d}\right) + \sigma_{\rm yr}^2 \cos\left(2\pi \frac{t_i - t_j}{1\ \rm yr}\right); \tag{41}$$

4. LEAF: same as diag. but the estimated calibration error σ_{b_i} (which is part of the observational error σ_i) is shared by night blocks (identified by b_i), and an additional calibration error term ($\sigma_{\rm cal.}$) is added in quadrature to these blocks (same value for all blocks),

$$C_{i,j} = (\sigma_i^2 - \sigma_{b_i}^2 + \sigma_{\rm jit.}^2)\delta_{i,j} + (\sigma_{b_i}^2 + \sigma_{\rm cal.}^2)\delta_{b_i,b_j}; \tag{42}$$

5. S + LEAF: same as LEAF plus the two quasiperiodic terms at 1 d and 1 yr as in the *celerite* model,

$$C_{i,j} = (\sigma_i^2 - \sigma_{b_i}^2 + \sigma_{\rm jit.}^2)\delta_{i,j} + (\sigma_{b_i}^2 + \sigma_{\rm cal.}^2)\delta_{b_i,b_j} + \sigma_{\rm d}^2 \mathbf{e}^{-0.1|t_i - t_j|} \cos\left(2\pi \frac{t_i - t_j}{1\ \rm d}\right) + \sigma_{\rm yr}^2 \cos\left(2\pi \frac{t_i - t_j}{1\ \rm yr}\right). \tag{43}$$

The quasiperiodic terms of the *celerite* and S + LEAF models are modeled according to Eq. (6) and could represent instrumental systematics (CCD stitching, wavelength solution instabilities, incorrect BERV correction, incorrect airmass corrections, etc.; see Dumusque et al. 2015). The HARPS radial velocities of HD 136352 are already corrected from the CCD stitching issue using the method of Dumusque et al. (2015), but remaining systematics could still be present. The amplitudes of the cosines (a_s in Eq. (6)) are noted $\sigma_{\rm d}$ and $\sigma_{\rm yr}$. For the sake of simplicity, we fix the amplitudes of the sines to zero ($b_s = 0$), such that the correlation is always maximum for $\Delta t = 0$ (see Eq. (6)). The exponential decay timescale is fixed to 10 d for the daily term ($\lambda_{\rm d} = 1/10$) and is infinite for the yearly term ($\lambda_{\rm yr} = 0$).

The noise parameters that remain to be determined are, thus, $\alpha = (\sigma_{\rm jit.}^2, \sigma_{\rm cal.}^2, \sigma_{\rm d}^2, \sigma_{\rm yr}^2)$, or a subset of it depending on the chosen noise model. The components of the covariance matrices corresponding to each of these four parameters are illustrated in Fig. 2. For these illustrations, the matrices are expanded as full $n \times n$ matrices, but we use their sparse representation (as described in Sect. 2) in the following computations.

3.2. Reanalysis of the HD 136352 system

We analyze the HARPS radial velocity time series of HD 136352 using each of the five noise models of Sect. 3.1. The deterministic part of the model is read as:

$$y_i = \gamma + \sum_{p<n_p} K_p \Big(\cos(v_p(t) + \omega_p) + e_p \cos\omega_p\Big), \tag{44}$$

where γ is the velocity offset, n_p is the number of planets, and, for each planet p, K_p is its semi-amplitude, v_p its true anomaly, e_p its eccentricity, and ω_p its argument of periastron. We start our study by considering a model without any planet and add them gradually, one after the other, by computing a periodogram of the residuals. At each step of this process, we adjust all the free parameters (deterministic and noise parameters). The deterministic parameters (vector θ) are the offset γ and the orbital parmeters P, K, M_0 (mean anomaly at a reference epoch), e, and ω for each planet included in the model. The noise parameters α are a subset of $(\sigma_{\rm jit.}^2, \sigma_{\rm cal.}^2, \sigma_{\rm d}^2, \sigma_{\rm yr}^2)$ depending on the chosen noise model. We use the L-BFGS-B algorithm (Byrd et al. 1995) to maximize the likelihood (Eq. (8)) and we make use of the backpropagation algorithms described in Sect. 2.3.4 (see also Appendix B) to compute the derivatives of the log-likelihood with respect to the free parameters. We also use classical analytical expressions for the derivatives of the Keplerian model (Eq. (44)) with respect to the orbital parameters of the planets. Then we compute a periodogram of the residuals of this maximum likelihood solution. The offset γ is readjusted for each frequency explored in the periodogram, but the previous planets and noise parameters are fixed (at the values obtained with the last fit).

We compute the periodograms and associated false alarm probability (FAP) using the analytical method of Delisle et al. (2020), based on the previous work by Baluev (2008). For a frequency ν, we define the normalized power as:

$$\text{Normalized Power}\ (\nu) = \frac{\chi^2_{\mathcal{H}} - \chi^2_{\mathcal{K}}(\nu)}{\chi^2_{\mathcal{H}}}, \tag{45}$$

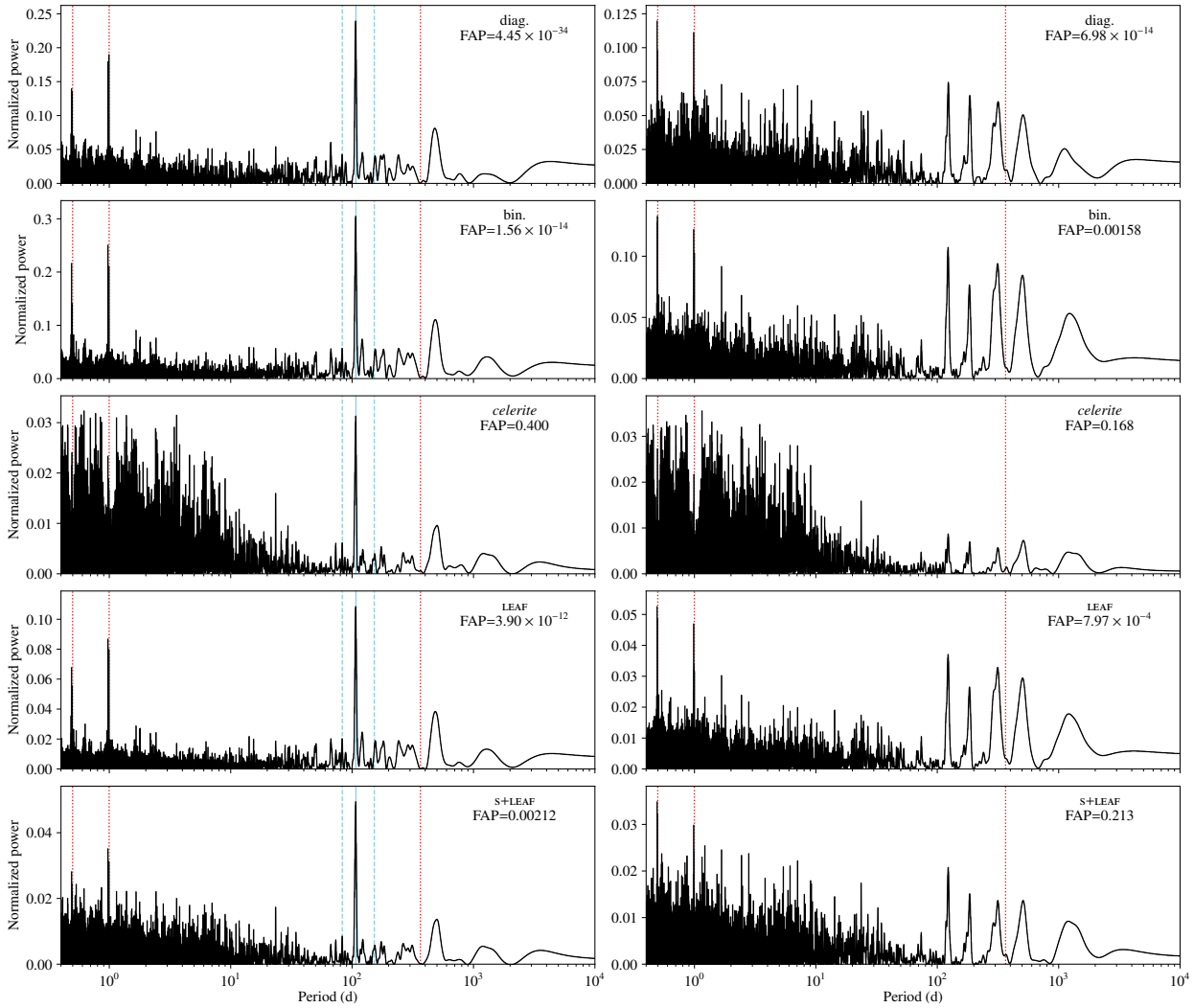

Fig. 3. Periodograms of the radial velocity residuals of HD 136352 after subtracting the two first planets (at 11.5824 d and 27.5821 d, *left*), and after subtracting the three known planets (11.5824 d, 27.5821 d, and 107.6 d *right*), for the five noise models defined in Sect. 3.1. The noise parameters are set to the values provided in Table 1. The vertical blue line highlights the period of the third planet (107.6 d), and the dashed blue lines highlight its aliases at 1 yr. The dotted vertical red lines highlight 0.5 sd, 1 sd, and 1 yr. For the sake of readability, we do not show here the two first periodograms (raw time series and after subtracting the first planet), since the two first planets (11.5824 d and 27.5821 d) are unambiguously detected (highest peaks and FAP $< 10^{-10}$) independently of the noise model. Assuming that the 107.6 d signal is due to a planet while the signals at 0.5 sd, 1 sd, and around 1 yr are due to correlated noise, we expect the correct noise model to show a low FAP in the *left column* and a high FAP in the *right column*.

which corresponds to the definition of the Generalized Lomb-Scargle periodogram (GLS, see Ferraz-Mello 1981; Zechmeister & Kürster 2009), and to $(2/n_{\mathcal{H}})z_1(\nu)$ in the notations of Baluev (2008) and Delisle et al. (2020). In this definition, $\mathcal{H}$ stands for the base model (only the offset γ is adjusted) and $\mathcal{K}$ stands for the model with frequency ν (γ plus the amplitudes of the sine and cosine at frequency ν are adjusted). The χ^2 of a model $m(\theta)$ is defined as:

$$\chi^2 = r^T C^{-1} r, \tag{46}$$

where r is the vector of the model residuals ($r = y - m(\theta)$).

The resulting periodograms are shown in Fig. 3. For the sake of readability, we do not show the first two periodograms since the first two planets (at 11.5824 d and 27.5821 d) are unambiguously detected (highest peaks and FAP $< 10^{-10}$) independently of the noise model. We additionally provide in Table 1 the values of the noise parameters used to compute each of the periodograms of Fig. 3.

Table 1. Noise parameters adjusted for HD 136352 and used to compute the periodograms of Fig. 3.

	Diag.	Bin.	*Celerite*	LEAF	S + LEAF
$\sigma^2_{\text{jit.}}$	2.73, 1.94	2.46, 1.63	0.37, 0.39	0.39, 0.39	0.39, 0.39
σ^2_{d}	–	–	3.23, 2.67	–	1.69, 0.70
σ^2_{yr}	–	–	0.00, 0.00	–	0.00, 0.00
$\sigma^2_{\text{cal.}}$	–	–	–	2.28, 1.45	0.56, 0.78

Notes. For each parameter, the first value corresponds to the left column and the second value to the right column of Fig. 3.

We observe in Fig. 3 (left column) that the last planet (HD 136352 d) is well revovered (highest peak and low FAP) by all models except the *celerite* model. With the *celerite* model, the peak corresponding to the planet is not the highest peak

and the FAP is high (0.4). We see in Table 1, that the amplitude of the daily quasiperiodic term of the *celerite* model is adjusted to a high value (3.23 $m^2\,s^{-2}$). On the contrary, for the S + LEAF model, the amplitude of the noise is shared between the daily quasiperiodic term (1.69 $m^2\,s^{-2}$) and the calibration noise (0.56 $m^2\,s^{-2}$). It thus seems that the daily quasiperiodic term of the *celerite* model is overestimated due to the presence of the unmodeled calibration noise. This shows that the way the calibration noise is accounted for in the S + LEAF model is well suited and does correspond to the behavior of the HARPS instrument.

The periodograms of the residuals of HD 136352 after subtracting all known planets (Fig. 3, right) do not show any significant peak for the *celerite* (FAP = 0.168) and S + LEAF (FAP = 0.213) models. On the contrary, the diag., bin., and LEAF models show significant peaks (with a low FAP) around 0.5 sd and 1 sd, as well as around 1 yr (see Fig. 3, right). These signals could be of planetary origin but are more probably due to instrumental systematics (CCD stitching, wavelength solution instabilities, incorrect BERV correction, incorrect airmass corrections, etc.; see Dumusque et al. 2015). They could also originate from a combination of stellar correlated noise and aliasing. These potential systematics are taken into account in the *celerite* and S + LEAF models with the daily and yearly quasiperiodic terms. While the amplitude of the daily quasiperiodic term is adjusted to significant values in the *celerite* and S + LEAF models, the amplitude of the yearly quasiperiodic term is completely negligible in both cases (see Table 1). We performed a similar analysis on the HARPS radial velocities of HD 136352 without the stitching correction and obtained higher values for the yearly term ($\sigma^2_{\rm yr} \approx 0.25\ m^2\,s^{-2}$). This highlights the improvements in the radial velocities obtained with this correction. In the S + LEAF model, the final levels (after substracting all known planets) of the daily quasiperiodic term and the calibration term are of the same order of magnitude (respectively, 0.7 and 0.78 $m^2\,s^{-2}$, see Table 1). This provides a good illustration of the importance of taking into account both components in the noise model.

The modeling of the systematics using daily and yearly quasiperiodic terms is a rough approximation, and a further investigation is necessary to confirm that these signals are instrumental systematics, to better characterize the systematics for several systems, to understand the mechanisms that might introduce them, and to correct for them, ideally directly in the HARPS data reduction software (DRS). However, this is beyond the scope of this study, and we simply highlight the ability of the S + LEAF model to roughly account for these systematics.

4. Conclusion

In this article, we present the S + LEAF correlated noise model. While in the general case, accounting for correlated noise in a dataset of size n has a cost of $O\left(n^3\right)$ and a footprint of $O\left(n^2\right)$, the S + LEAF noise model scales linearly (i.e., in $O(n)$). This linear scaling is made possible by the sparse properties of the S + LEAF covariance matrices (see Sect. 2). The S + LEAF model incorporates a mixture of quasiperiodic components (see Eq. (6)) as the *celerite* model (Foreman-Mackey et al. 2017) but it additionally takes into account a LEAF component. We call LEAF matrix a general class of "close to diagonal" matrices which encompasses banded, block-diagonal, and staircase matrices (see Eq. (10) and Fig. 1). For instance, the LEAF component of our model is well suited to account for calibration noise in radial velocity time series.

We illustrate the use of the S + LEAF model in the context of radial velocity time series but the model is more general and could be adapted to other fields. We reanalyze the HARPS radial velocity time series of HD 136352 using different noise models (see Sect. 3.2) and observe that the periodograms and FAP levels strongly depend on the chosen noise model. We find that neglecting the short term correlated noise (short period quasiperiodic noise or calibration noise) can lead to spurious detections of signals (underestimation of the FAP), or to a poor detection power (over estimation of the FAP). We thus show that the calibration noise, which can be included in the S + LEAF model, has a substantial effect on detections.

Acknowledgements. We thank the anonymous referee for their useful comments. We thank X. Dumusque and C. Lovis for fruitful discussions, and V. Bourrier for finding the name LEAF while advocating against the use of S + LEAF. We acknowledge financial support from the Swiss National Science Foundation (SNSF). This work has, in part, been carried out within the framework of the National Centre for Competence in Research PlanetS supported by SNSF.

References

Ambikasaran, S. 2015, Numer. Linear Algebra Appl., 22, 1102
Baluev, R. V. 2008, MNRAS, 385, 1279
Byrd, R., Lu, P., Nocedal, J., & Zhu, C. 1995, SIAM J. Sci. Comput., 16, 1190
Delisle, J.-B., Ségransan, D., Dumusque, X., et al. 2018, A&A, 614, A133
Delisle, J. B., Hara, N., & Ségransan, D. 2020, A&A, 635, A83
Dumusque, X., Udry, S., Lovis, C., Santos, N. C., & Monteiro, M. J. P. F. G. 2011, A&A, 525, A140
Dumusque, X., Pepe, F., Lovis, C., & Latham, D. W. 2015, ApJ, 808, 171
Faria, J. P., Haywood, R. D., Brewer, B. J., et al. 2016, A&A, 588, A31
Ferraz-Mello, S. 1981, AJ, 86, 619
Foreman-Mackey, D. 2018, Res. Notes Am. Astron. Soc., 2, 31
Foreman-Mackey, D., Agol, E., Ambikasaran, S., & Angus, R. 2017, AJ, 154, 220
Hara, N. C., Boué, G., Laskar, J., Delisle, J. B., & Unger, N. 2019, MNRAS, 489, 738
Haywood, R. D., Collier Cameron, A., Queloz, D., et al. 2014, MNRAS, 443, 2517
Queloz, D., Henry, G. W., Sivan, J. P., et al. 2001, A&A, 379, 279
Rajpaul, V., Aigrain, S., Osborne, M. A., Reece, S., & Roberts, S. 2015, MNRAS, 452, 2269
Rybicki, G. B., & Press, W. H. 1995, Phys. Rev. Lett., 74, 1060
Schwarzenberg-Czerny, A. 1991, MNRAS, 253, 198
Udry, S., Dumusque, X., Lovis, C., et al. 2019, A&A, 622, A37
Vandebril, R., Barel, M. V., Golub, G., & Mastronardi, N. 2005, CALCOLO, 42, 249
Zechmeister, M., & Kürster, M. 2009, A&A, 496, 577

Appendix A: Cholesky decomposition and solving in the preconditioned case

In this appendix, we show how to adapt the algorithms of the Cholesky decomposition (Sect. 2.3.1) and solving (Sect. 2.3.2) to the preconditioned case. As explained in Sect. 2.3.3 (and following Foreman-Mackey et al. 2017), we introduce the $(n-1)\times r$ preconditioning matrix ϕ, and the preconditioned matrices $\tilde{U}$ and $\tilde{V}$, such that:

$$U_{i,s}V_{j,s} = \tilde{U}_{i,s}\tilde{V}_{j,s}\prod_{k=j}^{i-1}\phi_{k,s}. \tag{A.1}$$

To stay consistent with this preconditioning, we additionally define $\tilde{W}$, $\tilde{S}$, and $\tilde{Z}$ such that:

$$\begin{aligned}
U_{i,s}W_{j,s} &= \tilde{U}_{i,s}\tilde{W}_{j,s}\prod_{k=j}^{i-1}\phi_{k,s},\\
U_{i,s}S_{i,s,t}U_{i,t} &= \tilde{U}_{i,s}\tilde{S}_{i,s,t}\tilde{U}_{i,t},\\
U_{j,s}Z_{i,j,s} &= \tilde{U}_{j,s}\tilde{Z}_{i,j,s}.
\end{aligned} \tag{A.2}$$

The recursion formulas for the Cholesky decomposition in the preconditioned case (see Eqs. (26)–(30)) are:

$$\tilde{S}_{0,s,t} = 0,$$

$$\tilde{S}_{i,s,t} = \phi_{i-1,s}\phi_{i-1,t}\left(\tilde{S}_{i-1,s,t} + \tilde{W}_{i-1,s}D_{i-1}\tilde{W}_{i-1,t}\right) \quad (i>0), \tag{A.3}$$

$$\tilde{Z}_{i,i-b_i,s} = 0,$$

$$\tilde{Z}_{i,j,s} = \phi_{j-1,s}\left(\tilde{Z}_{i,j-1,s} + G_{i,j-1}D_{j-1}\tilde{W}_{j-1,s}\right) \quad (j > i-b_i), \tag{A.4}$$

$$G_{i,j} = \frac{1}{D_j}\left(F_{i,j} - \sum_{k=\max(i-b_i,j-b_j)}^{j-1} G_{i,k}G_{j,k}D_k - \sum_s \tilde{U}_{j,s}\tilde{Z}_{i,j,s}\right), \tag{A.5}$$

$$D_i = A_i - \sum_s \tilde{U}_{i,s}\left(\sum_t \tilde{S}_{i,s,t}\tilde{U}_{i,t} + 2\tilde{Z}_{i,i,s}\right) - \sum_{k=i-b_i}^{i-1} G_{i,k}^2 D_k, \tag{A.6}$$

$$\tilde{W}_{i,s} = \frac{1}{D_i}\left(\tilde{V}_{i,s} - \sum_t \tilde{S}_{i,s,t}\tilde{U}_{i,t} - \tilde{Z}_{i,i,s}\right). \tag{A.7}$$

The recursion formulas for the solving ($x = L^{-1}y$) in the preconditioned case (see Eqs. (34) and (35)) are:

$$\tilde{f}_{0,s} = 0,$$

$$\tilde{f}_{i,s} = \phi_{i-1,s}\left(\tilde{f}_{i-1,s} + \tilde{W}_{i-1,s}x_{i-1}\right) \quad (i>0), \tag{A.8}$$

$$x_i = y_i - \sum_s \tilde{U}_{i,s}\tilde{f}_{i,s} - \sum_{j=i-b_i}^{i-1} G_{i,j}x_j, \tag{A.9}$$

where $\tilde{f}$ is defined such that

$$U_{i,s}f_{i,s} = \tilde{U}_{i,s}\tilde{f}_{i,s}. \tag{A.10}$$

The case of the dot product is very similar to the above (see Eqs. (36) and (37)) as well as the dot product and solving with L^{T}.

Appendix B: Backpropagation of the gradient for the S + LEAF model

In this section, we explain how to obtain gradient backpropagation algorithms for the S + LEAF model. Foreman-Mackey (2018) provided backpropagation algorithms for the Cholesky decomposition and solving in the case of semiseparable matrices ($F = 0$ in our notations). We generalize this method to S + LEAF matrices. We do not detail here the full algorithms but we rather describe the method used to obtain them and refer the reader to the reference implementation[2] for further details.

Let us first recall the steps required to evaluate the log-likelihood (see Sect. 2.1):

- Compute the deterministic part of the model $m(\theta)$, and the residuals $r = y - m(\theta)$;
- Compute the S + LEAF representation of the covariance matrix $A(\alpha)$, $\tilde{U}(\alpha)$, $\tilde{V}(\alpha)$, $\phi(\alpha)$, $F(\alpha)$;
- Compute the Cholesky decomposition of the covariance matrix D, $\tilde{W}$, G;
- Compute the log-determinant $\ln\det(C) = \sum_i \ln D_i$;
- Solve for $u = L^{-1}r$;
- Compute $\chi^2 = u^{\mathrm{T}}D^{-1}u = \sum_i \frac{u_i^2}{D_i}$;
- Compute $\ln\mathcal{L} = -\frac{1}{2}\left(\chi^2 + \ln\det(C) + n\ln\det(2\pi)\right)$.

We then need to compute the derivatives $\frac{\partial\ln\mathcal{L}}{\partial\theta}$ and $\frac{\partial\ln\mathcal{L}}{\partial\alpha}$. There are typically two ways to achieve this, the forward and backward propagation of the gradient. In the forward approach, computing the gradient (or the Jacobian matrix) of $y_n(x) = f_n\circ\cdots\circ f_2\circ f_1(x)$ is performed by first computing $\nabla f_1(x)$ and propagating it using the relation,

$$\nabla y_{k+1}(x) = \nabla f_{k+1}(y_k(x))\nabla y_k(x), \tag{B.1}$$

for $k = 1\ldots n-1$. In the backward approach, we first compute $\nabla f_n(y_{n-1}(x))$, and propagate it using the relation:

$$\nabla g_k(y_k(x)) = \nabla g_{k+1}(y_{k+1}(x))\nabla f_{k+1}(y_k(x)), \tag{B.2}$$

for $k = n-1\ldots 1$, with $g_k = f_n\circ\cdots\circ f_{k+1}$. In both methods, we need to compute the gradient of each function appearing in the composition (each f_i). The relative efficiency of both methods depends on the number of dimension of the parameter space and of the output space. Let us note p the number of parameters, and m_k the number of dimension of $y_k(x) = f_k\circ\cdots\circ f_2\circ f_1(x)$. In the forward approach, each step consists in the computation of a $m_k\times p$ matrix as the dot product of a $m_k\times m_{k-1}$ and a $m_{k-1}\times p$ matrices. In the backward approach, each step consists of computing a $m_n\times m_k$ matrix as the dot product of a $m_n\times m_{k+1}$ and a $m_{k+1}\times m_k$ matrices (with $m_0 = p$). Therefore, in the case $p < m_n$, the forward method should be more efficient, while in the case $m_n < p$, the backward method should be faster.

In the case of the log-likelihood, we have $m_n = 1$ (the log-likelihood is a scalar function), and the backward propagation should be preferred. The backpropagation method to compute the gradient of the log-likelihood can be decomposed in the following steps:

- Compute $\frac{\partial\ln\mathcal{L}}{\partial u_i} = -\frac{u_i}{D_i}$;

[2] https://gitlab.unige.ch/jean-baptiste.delisle/spleaf

- Compute $\frac{\partial \ln \mathcal{L}}{\partial D_i} = \frac{1}{2}\left(\left(\frac{u_i}{D_i}\right)^2 - \frac{1}{D_i}\right)$;
- Compute the gradient of $\ln \mathcal{L}$ with respect to $\tilde{U}$, $\tilde{W}$, ϕ, G, and r by using a backpropagation algorithm for the solving ($u = L^{-1}r$), and the values of $\frac{\partial \ln \mathcal{L}}{\partial u_i}$;
- Use a backpropagation algorithm for the Cholesky decomposition to compute the gradient of $\ln \mathcal{L}$ with respect to A, $\tilde{U}$, $\tilde{V}$, ϕ, and F;
- Backpropagate the gradient of $\ln \mathcal{L}$ with respect to the residuals to compute $\frac{\partial \ln \mathcal{L}}{\partial \theta} = \frac{\partial \ln \mathcal{L}}{\partial r}\frac{\partial r}{\partial \theta}$;
- Backpropagate the gradient of $\ln \mathcal{L}$ with respect to the S + LEAF decomposition of the covariance to compute $\frac{\partial \ln \mathcal{L}}{\partial \alpha}$.

The kth line of code appearing in the implementation of an algorithm (Cholesky decomposition, dot product $y = Lx$, solving, etc.) can be seen as a function f_k, while the full code is the composition $y_n(x) = f_n \circ \cdots \circ f_2 \circ f_1(x)$. In the case of the Cholesky decomposition, the vector x represents all the entries of A, $\tilde{U}$, $\tilde{V}$, ϕ, and F, while the output $y_n(x)$ represents D, $\tilde{U}$, $\tilde{W}$, ϕ, and G. The backpropagation of the gradient for the Cholesky decomposition consists in computing the derivatives $\frac{\partial h}{\partial A_i}$, etc., from the values of $\frac{\partial h}{\partial D_i}$, etc., for some function h. Applying the backpropagation method described above (Eq. (B.2)) is equivalent to reading the code of the algorithm in the reverse order (starting from the last line, and reversing the order of each loop) and backpropagating the gradient for each line.

Special care should be taken to ensure the stability of the method. For instance, divisions by zero (or small numbers) should be avoided. The only divisions that appear in the computation of the log-likelihood are the divisions by D_i (in the Cholesky decomposition and in the computation of the χ^2) which are unavoidable but not problematic for a well conditioned matrix. In the backpropagation algorithms, we also avoid any division other than divisions by D_i. Let us illustrate why this is preferable with the update formula for the tensor $\tilde{S}$ involved in the Cholesky decomposition algorithm (see Eq. (A.3)). As mentioned in Sect. 2.3.1, when computing the Cholesky decomposition of a S + LEAF matrix, the $n \times r \times r$ tensor $\tilde{S}$ could be stored in memory as a much smaller $r \times r$ matrix and updated in place using Eq. (A.3). Then the final value of this $r \times r$ matrix $\tilde{S}_{n-1}$ could be used as an initial value in the backpropagation algorithm and updated in place by computing $\tilde{S}_{i-1}$ from $\tilde{S}_i$ (see Eq. (A.3)),

$$\tilde{S}_{i-1,s,t} = \frac{\tilde{S}_{i,s,t}}{\phi_{i-1,s}\phi_{i-1,t}} - \tilde{W}_{i-1,s} D_{i-1} \tilde{W}_{i-1,t}. \tag{B.3}$$

This is done in the *celerite* code (Foreman-Mackey 2018) as it has a smaller memory footprint. However, looking at the update formula (B.3), we can see that when $\phi_{i-1,s}\phi_{i-1,t} \approx 0$, this turns out to be unstable numerically. This issue could thus induce a wrong determination of the gradient, which could slow down or prevent the convergence of minimization algorithms. We thus store the full $\tilde{S}$ tensor in the Cholesky decomposition algorithm, which increases the memory footprint of the algorithm but improves the efficiency and stability of the backpropagation method.