Allesfitter: Flexible Star and Exoplanet Inference from Photometry and Radial Velocity

Maximilian N. Günther1 © and Tansu Daylan2 ©

Department of Physics and Kavli Institute for Astrophysics and Space Research, Massachusetts Institute of Technology, Cambridge, MA 02139, USA;
maxgue@mit.edu

Received 2020 May 19; revised 2021 February 10; accepted 2021 February 15; published 2021 April 27

Abstract

We present allesfitter, a public and open-source Python software for flexible and robust inference of stars and exoplanets given photometric and radial velocity data. Allesfitter offers a rich selection of orbital and transit/eclipse models, accommodating multiple exoplanets, multistar systems, transit-timing variations, phase curves, stellar variability, starspots, stellar flares, and various systematic noise models, including Gaussian processes. It features both parameter estimation and Bayesian model selection, allowing either a Markov Chain Monte Carlo or Nested Sampling fit to be easily run. For novice users, a graphical user interface allows all input and perform analyses to be specified; for Python users, all modules can be readily imported into any existing script. Allesfitter also produces publication-ready tables, LaTeX commands, and figures. The software is publicly available (https://github.com/MNGuenther/allesfitter), pip-installable (pip install allesfitter), and well documented (www.allesfitter.com). Finally, we demonstrate the software’s capabilities in several examples and provide updates to the literature where possible for Pi Mensae, TOI-216, WASP-18, KOI-1003, and GJ 1243.

Unified Astronomy Thesaurus concepts: Exoplanets (498); Binary stars (154); Stellar flares (1603); Bayesian statistics (1900); Astronomy software (1855); Starspots (1572); Astronomy data modeling (1859)

1. Introduction

With the wealth of available photometric and radial velocity (RV) observations from ground- and space-based exoplanet missions, data analysis and modeling can become a limiting factor. Hence, the automation of this inference process in a reliable, scalable, and reproducible way is crucial. The exoplanet community can especially profit from a user-friendly, all-in-one package that allows for fast and robust model comparison.

Various packages have been developed for forward-modeling of exoplanets and binary star systems, including jktebop (Southworth et al. 2004a, 2004b), pytransit (Parviainen 2015), batman (Kreidberg 2015), elic (Maxted 2016), and starry (Luger et al. 2019). Using their underlying generative model of exoplanets and stars, these packages predict photometric and RV signals. General-purpose sampling algorithms can then be used to explore the parameter space of these forward-models consistent with observed data and subject to certain user-defined priors. The exoplanet community commonly uses Markov Chain Monte Carlo (MCMC) samplers (see Section 2.1), with popular implementations including emcee (Foreman-Mackey et al. 2013) and pymc3 (Salvatier et al. 2016). In this spirit, many researchers connect their forward-models with a sampler to analyze exoplanet-related data using private software (e.g., mcmc by Gillon et al. 2012, gp-ebob by Gillen et al. 2017, and amelie by Hodzić et al. 2018). Only recently, the exoplanet community started to develop standardized public software, such as exofast (Eastman et al. 2013, 2019), allesfitter (this work), juliet (Espinoza et al. 2019), and exoplanet (Foreman-Mackey 2019). Despite their recency, all of these packages have already been successfully and widely used in the literature.

Many existing packages focus on a specific task. For example, many apply to exoplanet transits and RV signals, but often not to exoplanet phase curves, Rossiter–McLaughlin effects, brown dwarfs, low-mass binaries, starspots, or stellar flares. Additionally, many packages rely only on MCMC samplers. While MCMCs can be robust and fast, they generally do not provide a statistically robust model comparison, given the absence of a low-variance estimate of the Bayesian evidence (e.g., Skilling 2006).

To overcome these caveats and provide general functionality and robustness, we developed the allesfitter package, which enables simultaneous (i.e., joint) inference of models for

1. exoplanet transits, occultations, and RV signals,
2. binary star eclipses and RV signals,
3. transit-timing variations,
4. phase curves,
5. stellar variability,
6. starspots,
7. stellar flares,
8. systematic noise, and
9. injection-recovery tests,

and was already used in various publications (e.g., Huang et al. 2018, hereafter H18; Daylan et al. 2021a, 2021b; Dragomir et al. 2019; Günther et al. 2019, 2020; Zhan et al. 2019; Shporer et al. 2019, hereafter S19; Badenas-Agustí et al. 2020). In a global analysis of both photometric and RV data, allesfitter also offers several ways to model red (systematic) noise, including polynomials, splines, or Gaussian processes (GPs). Additionally, the software allows the user to choose between MCMC and various Nested Sampling algorithms. Both take fair samples from the posterior of the selected model, while the latter...
also provides a low-variance estimation of the Bayesian evidence for statistical model comparison and is more robust for high dimensionality (see Section 2.2 for details).

Toward this purpose, allesfitter provides one framework uniting the versatile, publicly available packages ellc (light-curve and RV models; Maxted 2016), afiare (flare model; Davenport et al. 2014), dynesty (static and dynamic Nested Sampling; Speagle 2020), emcee (MCMC sampling; Foreman-Mackey et al. 2013), and celerite (GP models; Foreman-Mackey et al. 2017).

A graphical user interface allows novices to define all input parameters and settings without needing coding experience, making it well suited for undergraduate research programs, high school internships, or outreach events. However, users with Python experience can import the package into their scripts and use its application programming interface (API). The outputs of allesfitter are publication-ready plots, ASCII and LaTeX tables, and LaTeX commands. The software is public and open source,3 easily installable using PyPi,4 and well documented.5 Feedback and contributions are very welcome. We note that this manuscript reflects allesfitter version 1.2. Some of the functionality described here might not be available in older versions. For future additions and the most up-to-date documentation, see www.allesfitter.com.

This paper is structured as follows. Section 2 introduces Bayesian statistics and the inference framework, while Section 3 discusses the forward-models, including GPs. Section 4 explains the user interface, underlying routine, parameters, and settings. Section 5 showcases the application and performance for several test cases. In Section 6, we discuss our results and conclude.

2. Bayesian Statistics and Sampling

In Bayesian statistics, we compare models and infer their parameters using the “degree of belief” definition of probability (see, e.g., MacKay 2003). In this context, we are interested in the posterior probability \( \mathcal{P}(\theta|D) = P(\theta|M, D) \), i.e., the degree of belief about a set of parameters \( \theta \) given a selected model \( M \) and observed data \( D \). The foundation for this inference problem is Bayes’ theorem, which states that the posterior probability is given by

\[
P(\theta|M, D) = \frac{\mathcal{L} \mathcal{P}(\theta|M)}{P(D|M)}.
\]

Here, the likelihood \( \mathcal{L}(\theta) = P(D|M, \theta) \) is the probability of observing the data \( D \) under the given model \( M \) with parameters \( \theta \). The prior probability \( \mathcal{P}(\theta|M) \) of the parameters \( \theta \) given model \( M \) encapsulates our knowledge of the model before the arrival of the data, \( D \). Finally, \( Z := P(D|M) \) is the marginal likelihood, also known as the Bayesian evidence. It is calculated as the integral over the entire parameter space \( \Omega_\theta \):

\[
Z := \int_{\Omega_\theta} P(D|M, \theta) \mathcal{L} \mathcal{P}(\theta|M) d\theta.
\]

It quantifies the degree of belief one should have about the model \( M \) given the observed data \( D \). Estimating the Bayesian evidence allows a comparison of different physical models. However, the integral is computationally expensive to solve. While MCMC sampling completely bypasses its computation and leaves the Bayesian evidence unknown, Nested Sampling is specifically designed to estimate it (see below).

In the context of exoplanet science, the set of parameters \( \theta \) may contain, for example, the orbital period, planet radius, and stellar radius. The observed data \( D \) may be time series such as the normalized light curve and radial velocity of a target. The choice of priors can be motivated by other data sets or scaling arguments; for example, the period might be unknown but the stellar radius might be constrained by stellar models. Often, one would then assign a uniform prior to the period and a Gaussian prior on the stellar radius, with its mean and standard deviation reflecting the inference based on the characterization of the star using broadband or high-resolution spectra.

The data-informed part of the posterior is the likelihood function. For \( N \) data points, \( y_k \in (y_1, y_2, y_N) \) with uncertainties \( \sigma_k \) collected at times \( t_k \in (t_1, t_2, t_N) \), and a model evaluated on the same temporal grid \( M(t) \), the logarithm of the likelihood is given as

\[
\log \mathcal{L} = -\frac{1}{2} \sum_{k=1}^{N} \left( \frac{y_k - M(t_k)}{\sigma_k} \right)^2 + \log 2\pi\sigma_k^2.
\]

where we assume that uncertainties in data (light curves and RVs) have a Gaussian distribution.

2.1. MCMC

MCMC methods are a class of tools for taking fair samples from a given probability distribution (see, e.g., MacKay 2003) by constructing a Markov chain, i.e., a memoryless sequence of elements \( \theta_0, \theta_1, \ldots, \theta_N \). The stationary distribution of this chain approximates the relevant probability distribution \( P(\theta_{i+1}|\theta_i) \) in the limit \( N \to \infty \). The memorylessness property requires that each new state \( \theta_{i+1} \) depends only on the current state, \( \theta_i \). Depending on the initial state of a Markov chain, the mixing of the chain will require a certain number of state transitions. But even after that, consecutive samples will have a nonvanishing autocorrelation. Therefore, after a Markov chain is constructed, the samples are split into a burn-in and an evaluation part, where the burn-in samples are discarded because they are not drawn from the posterior. The evaluation samples are used to estimate the posterior probability distribution. These can be further thinned by a given factor, which can have practical advantages such as being more storage and memory efficient, but care must be taken to preserve the sampling efficiency (e.g., Geyer 1992; MacEachern & Mark Berliner 1994; Owen 2017). The resulting chain yields the desired posterior that is used for parameter estimation. There are multiple ways to implement an MCMC algorithm. Examples include the Metropolis–Hastings algorithm (Metropolis et al. 1953; Hastings 1970), Gibbs sampling (Geman & Geman 1984), and affine-invariant sampling (Goodman & Weare 2010). These algorithms’ common property is the concept of a random walk implemented via a proposal distribution to transitioning between such states. The proposal gets rejected with a probability set by how much it lowers the posterior and is accepted otherwise.

For allesfitter, we adopt the emcee package, which uses the affine-invariant sampling by default (for details see GitHub: https://github.com/MNGuenther/allesfitter. Installation: pip install allesfitter.

 Documentation and tutorials: www.allesfitter.com.
Goodman & Weare 2010; Foreman-Mackey et al. 2013). This enables efficient sampling from potentially skewed posterior probability distributions with correlated parameters and precludes the necessity to specify a proposal scale for each parameter. To do so, it employs multiple walkers (i.e., chains) with leap-frog proposals to explore the parameter space. The default settings for allesfitter’s MCMC implementation can be found in Table A1.

The user can also choose between different “moves” settings for emcee, determining how the walkers explore the parameter space. Additionally, emcee allows mixing different move algorithms. For a detailed description, refer to the reader the emcee documentation.6 In allesfitter versions prior to 1.2, we used the StretchMove described by Goodman & Weare (2010); Foreman-Mackey et al. (2013). From allesfitter version 1.2 onward, the default is to use the differential evolution move, DEMove (Ter Braak 2006). Both the DEMove and the related DESnookerMove are generally superior in computational speed and convergence (Ter Braak 2006; Ter Braak & Vrugt 2008; Nelson et al. 2014). In allesfitter, the user can also pass the setting mcmc_moves and choose one or multiple moves associated with their mixture-weight (see Table A1). For example, selecting “StretchMove 0.2 DEMove 0.8” would use the StretchMove for 20% of all steps and the DEMove for 80% of all steps.

Future updates for allesfitter could include an implementation of the PyMC3 sampler (Salvatier et al. 2016) as another option. PyMC3 can lead to further computational speed-up and can be more robust in complex, multimodal likelihoods.

### 2.1.1. Assessing Convergence

Despite discarding the initial samples and thinning the remaining chain, the resulting chain may still not be fully mixed (see Goodman & Weare 2010; Foreman-Mackey et al. 2013). Therefore, confirming their independence and thus the convergence of the MCMC sampling is important, yet often not strictly mathematically possible. To assess convergence nevertheless, two commonly used approximate criteria are (i) requiring a maximum autocorrelation or (ii) the Gelman–Rubin test statistic. In allesfitter, we implement the integrated autocorrelation time as the convergence criterion. Using it, we estimate the effective number of independent samples in the chain. It is recommended that the user runs the MCMC chains until all parameters have a chain length of at least 30 times the autocorrelation time (see Foreman-Mackey et al. 2013).

### 2.1.2. Limitations for Model Selection

MCMC is an efficient tool to take samples from the posterior of a model given some data. In addition to parameter estimation, one might also want to compare two models, e.g., a model of RV data with one and two planets, respectively. However, MCMC is limited when estimating the overall degree of belief in the associated model when using the harmonic mean because the contribution of rare samples from the posterior make increasingly large contributions to the Bayesian evidence (see, e.g., Weinberg 2012). This makes it hard to compare different models by estimating the Bayes factor, i.e., the ratio of the Bayesian evidence of the models.

### 2.2. Nested Sampling

Nested Sampling is an inference algorithm to directly estimate the Bayesian evidence by sampling from the prior subject to evolving constraints on the likelihood (Skilling et al. 2004; Skilling 2006; Feroz et al. 2009, 2019; Handley et al. 2015; Higson et al. 2018, 2019). Its low-variance estimate of the Bayesian evidence allows it to be used for robust model comparison. In the exoplanet context, this then enables model tests such as comparing models with different numbers of exoplanets (see Hall et al. 2018), a circular orbit against an eccentric orbit, or transit times with transit-timing variations (TTVs) to those without TTVs.

Nested Sampling achieves this by avoiding sampling directly from the posterior. Instead, it divides the problem into a series of simpler sampling problems. First, it draws a number of live points from the prior. Next, the live point with the lowest likelihood is removed. A new live point is created by sampling from the prior while requiring that it has a higher likelihood than before. The algorithm iterates over this process until the change in the resulting Bayesian evidence is below a certain threshold. The resulting samples from nested slices are sorted according to their likelihoods and used to compute the evidence integral by rewriting the multidimensional marginalization integral as a one-dimensional integral over the prior mass X of the hypervolume defined by points with a likelihood larger than the likelihood threshold of each slice,

\[
Z = \int_{\Omega_0} L(\theta) \pi(\theta) d\theta = \int_0^1 L(X) dX. \tag{4}
\]

Here, the prior volume X is defined as the fraction of the prior where the likelihood \(L(\theta)\) is greater or equal to a certain threshold \(\lambda\), i.e.,

\[
X(\lambda) = \int_{\Omega_0: L(\theta) > \lambda} \pi(\theta) d\theta, \tag{5}
\]

where \(L(X)\) is the isolikelihood. The bounds of the integral are defined by the normalization of the prior, leading to \(X(\lambda = 0) = 1\) and \(X(\lambda \to \infty) = 0\). Nested Sampling then uses a statistical approach to generate samples from the prior \(P(\theta)\). With this, it can approximate the prior volume \(X(\theta)\) and its differential, and thus compute the evidence integral.

allesfitter implements the dynesty package, which offers both static and dynamic Nested Sampling, as well as multiple options such as slicing, multinest, or polynest algorithms (for details see Speagle 2020). Dynamic Nested Sampling, in particular, is recommended as a generalization of Nested Sampling, in which samples can be drawn more efficiently by varying the number of live points. The default settings for allesfitter’s Nested Sampling implementation can be found in Table A1.

#### 2.2.1. Assessing Convergence

In Nested Sampling, the algorithm is considered converged once the gain in logarithmic Bayesian evidence, \(\Delta \ln Z\), is below a certain threshold. For allesfitter, we recommend the default threshold of \(\Delta \ln Z \leq 0.01\) (following dynesty; Speagle 2020).
Table 1
Interpretation of Bayes’ Factors from Jeffreys (1998) and Kass & Raftery (1995)

| ln R   | Strength of Evidence |
|--------|----------------------|
| 0 to 1.2 | Barely worth mentioning |
| 1.2 to 2.3 | Substantial |
| 2.3 to 4.6 | Strong to very strong |
| >4.6    | Decisive |

2.2.2. Model Selection

Because Bayesian evidence \( Z := P(D|M) \) is marginalized over the parameters of a given model, it allows us to compare models given the same data. This can be done by calculating the Bayes factor \( R \), which is defined as

\[
R := \frac{Z_{\text{Model 1}}}{Z_{\text{Model 2}}},
\]

where \( Z_{\text{Model 1}} \) and \( Z_{\text{Model 2}} \) are the Bayesian evidence for each model (e.g., a one-planet versus a two-planet model), and \( P_{\text{Model 1}} \) and \( P_{\text{Model 2}} \) are the prior beliefs in each model (not to be confused with the prior density for a set of parameters of a model).

Jeffreys (1998) and Kass & Raftery (1995) suggest that, given a null model \( M_1 \), the alternative (more complex) model \( M_2 \) should only be selected if there is sufficient relative Bayesian evidence for it as quantified by \( \ln R \gtrsim 5 \). In detail, they suggest the interpretation given in Table 1.

2.3. Residual Diagnostics

Even if a fit converged following the criteria of the MCMC or Nested Sampler (see above), there could be structure left in the residuals, for example, due to an insufficient baseline model. To investigate this for the user, allesfitter’s output functions also create text files with summary statistics for all residuals. These include the following: (i) the theorem-based Anderson–Darling test (Anderson & Darling 1952), which (in our case) tests the null hypothesis that the residuals are normally distributed at various significance levels. (ii) The augmented Dickey–Fuller test (Dickey & Fuller 1979), which tests the null hypothesis that the residuals show nonstationarity (trends) at various significance levels. (iii) The Durbin–Watson test (Durbin & Watson 1951), which tests the null hypothesis that there is no correlation among the residuals. (iv) The Ljung–Box test (Ljung & Box 1978), which also tests the null hypothesis that there is no correlation among the residuals at various significance levels. The success of these tests indicates that the residuals are consistent with being normally distributed, stationary, and uncorrelated. Their failure justifies modification of the fitting model or extension or refinement of the sampling.

3. Generative Models

3.1. Orbits, Eclipses/Transits/Occultations, and Stellar Brightness Features

Our forward-model for the photometry and radial velocity observed in stellar and planetary systems is largely implemented via the public, open-source software ellc (Maxted 2016). At its core, ellc is a fast, flexible, and accurate binary star model, which is also readily applicable to exoplanets. The software incorporates an eclipse model and the effects of starspots, Doppler boosting, light-travel times, the flux-weighted radial velocity during an eclipse (Rossiter–McLaughlin effect), and light from a blended source. The ellc generative model was substantially tested, compared with other existing models, and is already widely used in the literature. We summarize the core principles of ellc in the following and refer the reader to Maxted (2016) for all details.

**Shapes:** ellc models the stars as triaxial ellipsoids and calculates their flux using Gauss–Legendre integration over the visible surface. The shape of the objects can be calculated in three ways: (i) in the spherical limit, (ii) via the Roche potential and including nonsynchronous rotation, and (iii) via a polytropic equation of state. This flexibility allows the user to adjust the settings to their desired models and readily model, for example, the ellipsoidal modulation in the phase curve of a binary star or hot Jupiter system.

**Orbits:** To compute the objects’ positions, ellc applies Keplerian orbits with fixed orbital eccentricity \( e \) and apsidal motion (if chosen). It updates all positions using Kepler’s equation, \( E = E - e \sin E \), to solve for the eccentric anomaly \( E \) from the mean anomaly \( M = 2\pi(t - t_0)/P \), for a fixed anomalistic period \( P \), and to compute the true anomaly \( \nu \). This approach incorporates the calculation of, and a correction for, the light-travel times.

**Surface brightness:** The surface brightness distribution \( I(s, t) \) at any surface point \( s \) at time \( t \) incorporates established limb-darkening laws \( U(\mu) \), gravity darkening \( G(s, t) \), and the irradiation of the body by its companion \( H(s, t)U_H(\mu) \) following the relation

\[
I(s, t) = I_0U(\mu)G(s, t) + H(s, t)U_H(\mu).
\]

The limb-darkening laws include all standard choices, from constant to four-parameter laws, depending on the normalized distance \( \mu \) from the center (see Table 2). The gravity-darkening calculation assumes that the specific intensity relates to the local gravity by a wavelength-dependent power law. For this, ellc calculates the local gravity via the gradient of the Roche potential. The user decides whether the local gravity should be

Table 2
Common Limb-darkening Laws

| Name | Equation for \( U(\mu) \) |
|------|--------------------------|
| constant | 1 |
| linear | \( 1 - c_1(1 - \mu) \) |
| square-root | \( 1 - c_1(1 - \mu) - c_2/(1 - \sqrt{\mu}) \) |
| exponential | \( 1 - c_1(1 - \mu) - c_2/(1 - e^\mu) \) |
| logarithmic | \( 1 - c_1(1 - \mu) - c_2\ln \mu \) |
| quadratic | \( 1 - c_1(1 - \mu) - c_2(1 - \mu)^2 \) |
| three-parameter | \( 1 - c_1(1 - \mu) - c_2(1 - \mu)^{3/2} - c_3(1 - \mu)^2 \) |
| four-parameter | \( 1 - c_1(1 - \mu)^{3/2} - c_2(1 - \mu)^2 - c_3(1 - \mu)^3 \) |

Notes.

\( ^a \) Schwarzschild & Villiger (1906).
\( ^b \) Diaz-Cordoves & Gimenez (1992).
\( ^c \) Claret & Hauschildt (2003).
\( ^d \) Klinglesmith & Sobieski (1970).
\( ^e \) Kopal (1950).
\( ^f \) Sing (2010).
\( ^g \) Claret (2000).
and survival of life on exoplanets posing disadvantages as well as opportunities for the genesis core and convective envelope. Flares on M-dwarf stars are dynamo, driven by the shearing rotation of the star.

**Other effects:** eLLC directly computes the weighted flux from each grid point on the objects’ visible surfaces, taking into account light-travel times. It thus can compute Doppler boosting (relativistic beaming) and Doppler shift, which are relativistic effects that increase and decrease, respectively, the observed flux depending on the bodies’ radial velocity, spectra, and observation wavelengths. It also models starspots, which are approximated as circular regions with different brightness, set at a given longitude and latitude on the star. Similar to the bodies’ shapes, the spot brightness is integrated over triaxial ellipsoids taking limb darkening into account.

**Light curves**: The final light curves are generated by integrating over the bodies’ surface brightness distributions using a mix of Gaussian–Legendre and analytical integration methods. This includes all effects from different object shapes, starspots, eclipses, and dilution by a third light term (originating from a blended object).

**RVs**: Likewise, the bodies’ RV signals are computed from the weighted flux from every visible grid point. These computations automatically incorporate terms for Keplerian orbit velocities, projected rotational velocity (both equatorial and asynchronous), different shapes, starspots, and eclipses. By construction, the latter includes the Rossiter–McLaughlin effect when modeling RV data, even for distorted bodies.

### 3.2. Stellar Flares

Stellar flares are explosive magnetic reconnection events that emit large amounts of radiation, predominately in the UV to X-ray spectrum. They are the product of the stellar magnetic dynamo, driven by the shearing rotation of the star’s radiative core and convective envelope. Flares on M-dwarf stars are especially much more frequent and energetic than on our Sun, posing disadvantages as well as opportunities for the genesis and survival of life on exoplanets (e.g., Pettersen 1989; Rimmer et al. 2018; Günther et al. 2020).

To model stellar flares, we adopt the aflare module from the appaloosa package (Davenport et al. 2014; Davenport 2016). This model only depends on three parameters: the flare’s peak time \(t_{\text{peak}}\), amplitude \(A\), and FWHM \(t_{1/2}\). This empirical template was created from a sample of 885 flares on the M dwarf GJ 1243, assuming that stellar flares share a common formation mechanism across stellar types. The authors selected “classical flares” in the Kepler light curve with a duration between 20 and 75 minutes. They then detrended the data to remove modulation by starspots and normalized all flares to scale from 0 to 1 in amplitude and by a single timescale factor in width to fulfill the normalization \(t_{1/2} = 1\). Next, they fitted a third-order polynomial to describe the rise time as

\[
F_{\text{rise}} = 1 + c_1 t_{1/2} + c_2 t_{1/2}^2 + c_3 t_{1/2}^3 + c_4 t_{1/2}^4, \quad (8)
\]

and a double-exponential parameter to describe the decay time as

\[
F_{\text{decay}} = 1 + c_5 e^{-c_6 t_{1/2}} + c_6 e^{-c_7 t_{1/2}}. \quad (9)
\]

The rapid rise toward the peak flux was motivated by the morphology seen in ground-based white-light photometry (e.g., Kowalski et al. 2013). We note that this does not hold true for all flares as, for example, pointed out by Jackman et al. (2018, 2019), who used high-cadence photometry and found the need for additional terms to describe a “rollover” rather than a sharp peak. The double-exponential decay represents two physically distinct regions with independent exponential cooling profiles. Davenport et al. (2014) argue that the initial decay might be dominated by a hotter region that cools rapidly and that the gradual decay stems from a cooler region that cools slower.

#### 3.3. Red Noise, Stellar Variability, and Gaussian Processes

Generative models used to fit observations are never perfect descriptions of the data, even up to white (uncorrelated) noise, because observed data are always affected by physical processes not available in the fitted model. The total effect of these unaccounted processes in the data is usually referred to as red (correlated) noise. The apparent correlation of this noise is a consequence of the time-variability of the unaccounted physical processes. Examples include instrumental noise, atmospheric scintillation effects for ground-based observatories, light from blended objects, or scattered light from Earth and the moon for space-based observatories. Stellar variability may also be counted as correlated noise if one is only interested in an exoplanet’s properties but not the star’s behavior. However, at other times, one may wish to model stellar variability explicitly to characterize the stellar rotation or activity.

**Allesfit** includes various options to model red noise and stellar variability, including constant offsets, polynomial trends, cubic splines, and various GP models. An overview of all these models is given in Table 3. In the software interface, these models are implemented in two complementary ways:

1. a baseline model, which is instrument dependent (see Section 4.6) and

| Name | Equation for \(M(t)\) |
|------|----------------------|
| none | 0 \(c_1\) |
| offset | \(c_1 + c_2 t\) |
| linear | \(c_1 + c_2 t + c_3 t^2\) |
| quadratic | \(c_1 + c_2 t + c_3 t^2 + c_4 t^3\) |
| third-order poly. | \(c_1 + c_2 t + c_3 t^2 + c_4 t^3 + c_5 t^4\) |
| fourth-order poly. | \(c_1 + c_2 t + c_3 t^2 + c_4 t^3 + c_5 t^4 + c_6 t^5\) |
| cubic spline | \(c_1(t) + c_2(t) + c_3(t) + c_4(t)\) |
| GP real | \(a e^{-\alpha t}\) |
| GP complex | \(\frac{1}{2}((a + b) e^{i(\omega t + \phi)} + (a - b) e^{i(\omega t - \phi)}\) |
| GP Matérn 3/2 | \((1 - 1/\nu) e^{-\|\mathbf{r}\|_2^\nu/\nu}\) |
| GP SHO | \(\frac{\sigma^2}{\nu} \sum_{n=0}^{\nu} \frac{(\nu^n-n!)^2}{n!} \mathbf{r}\) |

**Note.** Other variables are the fit parameters explained in the text.
2. a stellar variability model, which fits a common trend across all instruments (see Section 4.7).

Some of the most versatile baseline or stellar variability models are GPs (Rasmussen & Williams 2005; Bishop 2006; Gibson et al. 2012; Roberts et al. 2012). Instead of fitting for the parameters of a chosen model (e.g., coefficients of a polynomial), GP regression fits for a family of functions to marginalize over the choice of the basis in a so-called nonparametric approach.

In Bayesian inference, a GP can be interpreted as a prior on the space of functions that describe the data (see, e.g., Murphy 2012). When updated based on observed data, the GP model’s posterior characterizes the distribution of baselines needed to fit the data along with the desired physical model. The autocorrelation of the GP is described by the specified distance metric and kernel, i.e., a covariance matrix that sets the flexibility of the GP. Kernels with a large autocorrelation scale produce smoother baselines, whereas those with a small autocorrelation scale produce turbulent baselines. The GP kernel is fitted to the data by sampling from the posterior of the hyperparameters. This posterior can then be linked to physical processes such as stellar variability, atmospheric scintillation, or instrumental noise.

**Allesfitter** implements the celerite package, which provides a series of expressions of typical GP kernels, achieving a significant improvement in execution time (see Foreman-Mackey et al. 2017, for details). Out of the available kernel functions (see Table 3 for equations), the real kernel is the simplest, with exponential decay and two hyperparameters, the amplitude $a$ and timescale $c$. The complex kernel is a relatively more complex model with amplitudes $a$ and $b$, and timescales $c$ and $d$. The Matérn $3/2$ kernel is one of the most versatile and frequently used kernels in astronomy, as it can describe smooth long-term trends as well as stochastic short-term variations. It features two hyperparameters, i.e., the amplitude scale $\sigma$ and length scale $\rho$. Lastly, the simple harmonic oscillating (SHO) kernel is appropriate for (semi) periodic signals, as it represents a stochastically driven, damped harmonic oscillator. Its hyperparameters are the amplitude scale $S_0$, the damping $Q$, and the frequency $\omega$.

### 4. User Interface, Settings, and Parameters

#### 4.1. Overview

Every **allesfitter** run is designed to operate in a user-designated working directory. The input configuration to **allesfitter** is provided via comma-separated value (CSV) files for the settings and parameters in this working directory, named settings.csv and params.csv, respectively. All possible inputs in these files are explained in Tables A1 and A2, and any special implementations are laid out in the sections below.

All data must be stored as CSV files in this working directory. The data file names must match those provided in the settings and parameters files. For example, if the user names the instruments TESS and ESPRESSO in those files, the data file names must be TESS.csv or ESPRESSO.csv. Light-curve files need three columns: the time (in days), relative flux (i.e., normalized to 1), and the relative flux’s uncertainty. The errors are only needed for their relative values across time because the errors are scaled (mean of all errors) using a model parameter (see Section 4.5 and Table A2). Therefore, if the errors are unknown, the final column can be filled with values of 1. RV files also need three columns: the time (in days), radial velocity (in km s$^{-1}$), and instrumental error of the radial velocity. For radial velocity instruments, both the error weights and their scaling will affect the fits. If the instrumental noise is unknown, the final column can be filled with values of 0, as a stellar jitter term will still be added in quadrature during the fit (see Section 4.5 and Table A2).

There are two ways to start an **allesfitter** run. First, the graphical user interface (GUI) can guide the user through the entire setup, from assigning a working directory to generating the necessary settings and parameters files to running the analyses. Second, the user can manually create the settings and parameter files in the working directory, either from scratch or by using any of the template files. Then the user can use the API to import the **allesfitter** module and call all respective functions to start the run (see below). When a run is started, **allesfitter** creates a “results” folder in the base folder. A log file is created for each run, uniquely named with the ISO 8601 compliant date and time. All output will also be saved into this folder (see below).

#### 4.2. API and Output

**Allesfitter** is built in a modular way, and as such, many functions can be called directly from the Python interface. In this section, we briefly lay out the most important parts of the API and refer the user to www.allesfitter.com for details, future updates, and the most up-to-date documentation.

As a first step after setting up the working directory, the users can investigate how well their initial guess matches the data by calling

```
allesfitter.show_initial_guess(datadir),
```

where `datadir` is the working directory path. This creates initial guess plots and a log file in the working directory. Once the user verifies that the data and initial guess look as intended, the inference (MCMC or Nested Sampling) can be initiated.

An MCMC fit can be performed by calling

```
allesfitter.mcmc_fit(datadir).
```

This creates the file `mcmc_save.h5` and a log file in the given directory path, and the state of the sampling can be monitored with a waitbar in the Python console. Any time during the execution, output files can be created by calling

```
allesfitter.mcmc_output(datadir).
```

including the samples up to the last stored state. This allows an efficient way to diagnose whether the run is configured and behaves as intended, e.g., by inspecting the evolution of the chains. Once the sampling is completed, this should be re-executed to generate the final results.

A Nested Sampling fit can be performed by calling

```
allesfitter.ns_fit(datadir).
```

This creates a log file in the given directory path, and the state of the sampling can be monitored through the sampling output in the Python console. Once the sampler has converged, the samples will be stored in the file. Note that, due to Nested Sampling’s iterative algorithm conditional on convergence, the progress cannot be monitored in a waitbar. However, the progress can be gauged by monitoring how the value $d \log Z$ decreases to the chosen tolerance limit (default: 0.01). As a rule
of thumb, the time needed for completion scales logarithmically, such that, for example, decreasing from 100 to 10 takes the same time as from 10 to 1. Once the algorithm converges, all output files can then be created by calling

\texttt{allesfitter\_ns\_output(datadir)}.

A helpful feature for fine-tuning figures and retrieving results from converged runs is the \texttt{allesclass}, which can be called as

\texttt{allesfitter\_allesclass(datadir)}.

This allows the user to easily retrieve all data, parameters, settings, initial guess, and posterior samples, as well as the baseline, stellar variability, transit, and phase-curve forward-model samples. It also offers various plotting utilities to easily customize figures for publications.

Other major modules of \texttt{allesfitter} are the easy-to-use interfaces for transit injection with \texttt{ellc} and recovery with \texttt{tls}. We refer the user to \url{www.allesfitter.com} for detailed documentation on these modules. Additionally, \texttt{allesfitter} contains various modules to process light-curve and RV data and perform tasks such as transit-masking and phase-folding.

4.3. Transit/Eclipse Epoch

In a linear ephemeris model, the transit times are described by an epoch and a period (we use the word “epoch” as a synonym for eclipse/transit midtime). We follow the definition of \texttt{ellc} and define the epoch as the reference time when the projected separation of the two bodies’ centers-of-mass is at a minimum. In the literature or archives, the epoch is often reported as the time of the first observed transit, which introduces an unnecessary degeneracy between the orbital period and epoch. Shifting the transit epoch into the middle of the temporal baseline, \( t \), reduces this degeneracy. \texttt{Allesfitter} automatically handles this if the user sets \texttt{shift\_epoch} to \texttt{True} in the settings file (Table A1). The shifted epoch is then the one closest to \( \text{min}(t) + \text{max}(t)/2 \). The user can further select the data used for this operation (see Table A1), as not all data are equally informative for this operation. For example, if presented with three data sets, one of archival low-precision RVs, one of current high-precision photometry, and one of current high-precision RVs, the user might wish to select only the latter two data sets to shift the epoch.

4.4. Limb-darkening Parameterizations

\texttt{Allesfitter} allows users to chose between limbdarkening parameterizations in either (i) the physical \( \mu \)-space (Table 2) or (ii) the transformed \( q \)-space from Kipping (2013) and Kipping et al. (2017). The user can choose any of the common limb-darkening laws listed in Table 2 when sampling the parameters in the physical \( \mu \)-space. Alternatively, they can choose the quadratic and three-parameter laws when sampling in the transformed \( q \)-space. For the corresponding software settings and parameters, see Tables A1 and A2.

We generally recommended the users to sample in the transformed \( q \)-space with uniform priors between [0, 1], and let the data inform the limbdarkening parameters. Especially for high signal-to-noise-ratio (S/N) cases, this can mitigate biases (Kipping 2013). However, in low-S/N cases, free sampling for limb darkening could reduce the precision of the posterior with negligible gain in accuracy. In such cases, tabulated values for limb-darkening parameters in the physical \( \mu \)-space (e.g., Claret et al. 2013) can provide prior knowledge.

The user can switch between the \( \mu \) - and \( q \) -space using the settings \texttt{host\_ld\_space\_inst} and \texttt{[companion]\_ld\_space\_inst}, and choose the limbdarkening law via the settings \texttt{host\_ld\_law\_inst} and \texttt{[companion]\_ld\_law\_inst} (Tables A1 and A2). The parameters have to correspond to the chosen law and sampling space; for example, the parameters for the host star with a quadratic law in \( q \)-space are \texttt{host\_ld\_q1\_inst} and \texttt{host\_ld\_q2\_inst}.

For a constant and linear law, \( \mu \) - and \( q \) -space are the same. For a quadratic law, the transformation is given as (Kipping 2013)

\begin{align}
    u_1 &= 2\sqrt{q_1} q_2 \\
    u_2 &= \sqrt{q_1(1 - 2q_2)} \\
    q_1 &= (u_1 + u_2)^2 \\
    q_2 &= 0.5u_1(u_1 + u_2)^{-1}.
\end{align}

For the three-parameter law, transformation algorithms are provided by Kipping et al. (2017). To translate between \( \mu \) - and \( q \) -space for quadratic and three-parameter laws, we also provide two convenience functions. Their usage is explained in the following examples:

\begin{verbatim}
import allesfitter
allesfitter.u_to_q([u1,u2],law='quad')
allesfitter.q_to_u([q1,q2,q3],law='sing')
\end{verbatim}

In the software, “quad” is the quadratic law from Kopal (1950) and “sing” is the three-parameter law from Sing (2010) (see also Table A1).

After convergence, \texttt{Allesfitter} always recomputes the physical parameters in the physical \( \mu \)-space for comparison and interpretability (see Section 4.11).

4.5. White Noise and Jitter Terms

The photometric uncertainties of the user inputs are normalized to 1, such that only their weights toward another are important. The mean of the uncertainties are fitted as a model parameter,

\begin{equation}
    \sigma_{\text{white, total}}^\text{result} = \sigma_{\text{white, weights}}^\text{user input} \cdot \sigma_{\text{white, scaling}}^\text{fit param}.
\end{equation}

In contrast, for RV data, a jitter term is fitted. Therefore, the input values are not normalized. Instead, the total uncertainty on each RV data point is calculated as

\begin{equation}
    \sigma_{\text{white, total}}^\text{result} = \sqrt{\sigma_{\text{white, inst}}^2 + \sigma_{\text{jitter}}^2}^\text{fit param}.
\end{equation}

4.6. Baselines (Red Noise)

Various baseline models are available to handle red noise caused by instrumental systematics and stellar variability. While these models are described in detail in Section 3.3, we here explain how they can be called via the API. In the settings file, the user can choose between options from two major groups (see Table A1):

1. sampling from the posterior of the parameters that describe the baseline (called \texttt{sample*}),
2. profiling the likelihood by maximizing it for each proposal (called \texttt{hybrid*}).

For all \texttt{sample*} options, the user must also provide the respective parameters in the parameter file (see Table A2). All available baseline options are:

\begin{itemize}
    \item \texttt{white,inst}
    \item \texttt{white,scaling}
    \item \texttt{white,weights}
    \item \texttt{white,scaling,weights}
    \item \texttt{white,scaling,inst}
    \item \texttt{white,scaling,inst,weights}
    \item \texttt{white,scaling,inst,weights,inst}
    \item \texttt{white,scaling,inst,weights,inst,weights}
\end{itemize}
1. No baseline fitting. The respective setting is *none*, and the baseline is fixed at 1 for light-curve data and at 0 for RV data.

2. Sampling a constant offset. The respective setting is *sample_offset* and the corresponding parameter is `baseline_offset_|key|_[inst]`.

3. Sampling a linear trend. The respective setting is *sample_linear* and the two corresponding parameters are `baseline_offset_|key|_[inst]` and `baseline_slope_|key|_[inst]`.

4. Sampling a GP with a real kernel. The respective setting is `sample_GP_real` and the two corresponding parameters are `baseline_gp_real_Lna_|key|_[inst]` and `baseline_gp_real_Lnc_|key|_[inst]`.

5. Sampling a GP with a complex kernel. The respective setting is `sample_GP_complex` and the four corresponding parameters are `baseline_gp_complex_Lna_|key|_[inst]`, `baseline_gp_complex_Lnb_|key|_[inst]`, `baseline_gp_complex_Lnc_|key|_[inst]`, and `baseline_gp_complex_Lnc_|key|_[inst]`.

6. Sampling a GP with a Matérn-3/2 kernel. The respective setting is `sample_GP_Matern32` and the two corresponding parameters are `baseline_gp_matern32_Lnsigma_|key|_[inst]` and `baseline_gp_matern32_Ln rhoo_|key|_[inst]`.

7. Sampling a GP with an SHO kernel. The respective setting is `sample_GP_SHO` and the two corresponding parameters are `baseline_gp_sho_Lns0_|key|_[inst]`, `baseline_gp_sho_LnQ_|key|_[inst]`, and `baseline_gp_sho_Lnomega0_|key|_[inst]`.

8. Hybrid offset. The respective setting is `hybrid_offset`. At each step, the median of the residuals will be taken as the baseline.

9. Hybrid polynomials. The respective setting is `hybrid_poly_+`, followed by a number from 1 to 4, which sets the polynomial degree. At each step, a least-squares minimization will determine the polynomial parameters to set the baseline.

10. Hybrid cubic spline. The respective setting is `hybrid_spline`. At each step, a least-squares minimization will determine the cubic spline parameters to set the baseline.

### 4.7. Stellar Variability

Stellar variability can generate a signal or red noise shared between different instruments, especially for those in similar bands. Hence, it is implemented as a separate component in addition to the baselines for individual instruments. For example, the user may wish to fit two data sets from different instruments that have distinct instrumental red noise but a common stellar variability trend.

The functionality is the same as for baselines (see Section 4.6 and Tables A1 and A2). The user only needs to replace the keyword `baseline` with `stellar_var` and drop the part `_[inst]`. For example, for a GP with a Matérn 3/2 kernel, the setting is `sample_GP_Matern32` and the two corresponding parameters are `stellar_var_gp_matern32_Lnsigma_|key|` and `stellar_var_gp_matern32_Ln rhoo_|key|`.

### 4.8. External Priors: Stellar Host Density

If enabled by the user, an external normal prior on the host’s bulk density is calculated from the input stellar radius and mass (by setting `use_stellar_density_prior` to `True` and passing a stellar parameters file; see Table A1). At each sampling proposal and for each companion, this is compared to the host’s bulk density $\rho_*$ as (e.g., Seager & Mallén-Ornelas 2003)

$$\rho_* = \frac{3\pi}{GP^2} \left( \frac{a}{R_*} \right)^3 + \left( \frac{R_{\text{comp}}}{R_*} \right)^3 \rho_{\text{comp}}$$

Here, $R_*$ is the host’s radius, $R_{\text{comp}}$ is the companion’s radius, $\rho_{\text{comp}}$ is the companion’s density, $a$ is the companion’s semimajor axis, and $P$ is the companion’s orbital period.

If the user provides transit and RV data, `allelesfitter` uses the full equation. It computes the mass by solving the binary mass function while accounting for eccentricity, thus it is applicable to exoplanets and binaries on circular and eccentric orbits alike. If the user only provides transit data, `allelesfitter` only applies this external prior if $(R_{\text{comp}}/R_*)^3 < 0.01$ and ignores the companion’s density term. We chose this cutoff empirically to minimize bias while still allowing free sampling for exoplanet applications.

This external prior is particularly beneficial for small exoplanets in low-S/N data. The stellar density is usually better constrained from Gaia and spectral measurements, evolutionary models, or empirical models than it is from the transit light curve. Additionally, the external prior can provide consistency for multiplanet systems.

#### 4.9. Phase Curves

`allelesfitter` offers three options for modeling exoplanet and binary star phase curves. First, a parametric method can be used to fit a linear combination of sine and cosine waveforms. The semi-amplitudes of these terms can then be interpreted as physical quantities, which is a common approach in exoplanet phase-curve analyses (Section 4.9.1). Second, a similar but transformed sinusoidal parameterization can be chosen to ensure that the user input directly relates to physical quantities (Section 4.9.2). Third, a physical model can be employed by utilizing the forward-model of `ellc` (Section 4.9.3).

#### 4.9.1. Phase Curves Using Sines

One can approximate a phase curve as a linear combination of sine and cosine waveforms, which models the out-of-eclipse variation of the system’s flux, $F$, as a third-order harmonic series dependent on the orbital phase $\phi(t)$ (e.g., Carter et al. 2011; S19; Wong et al. 2020):

$$F \propto \sum_{k=1}^{3} A_k \sin k \phi(t) + \sum_{k=1}^{3} B_k \cos k \phi(t).$$

These terms can be related to the following three physical effects:

1. Doppler-boosting (beaming) modulation, which is caused by the periodic redward and blueward color shifts of the emission from the host star due to its orbital motion (e.g., Shakura & Postnov 1987). The effect can be approximated by the sinusoidal term $A_1 \sin(\phi(t))$. Only positive values of $A_1$ allow a physical interpretation as the semi-amplitude of the host star’s beaming effect, $A_{\text{semi}} = A_1$.

2. Atmospheric modulation, which includes the thermal and reflected emission from the companion (e.g., Snellen et al. 2009). It can be approximated by the fundamental cosine term $B_1 \cos(\phi(t))$, where $B_1$ is a semi-amplitude. Only negative values of $B_1$ allow a physical interpretation as the full (peak-to-peak) amplitude of the companion’s atmospheric component, $A_{\text{atmos}} = -2B_1$.

3. Ellipsoidal modulation, which appears when the host star is tidally distorted due to the gravity of the companion (e.g.,
Morris 1985). It can be approximated by the sum of harmonic cosine terms, with the leading-order term being $B_2 \cos(2\phi(t))$ and the next-order term being $B_3 \cos(3\phi(t))$. Note that the leading-order term is sufficient for exoplanet phase curves, but the next-order term can become detectable for binary phase curves. Only negative values of $B_2$ and $B_3$ allow physical interpretation as the system’s ellipsoidal components, $A_{\text{full},1st}^\text{ellc} = -2B_2$, $A_{\text{full},2nd}^\text{ellc} = -2B_3$.

In allesfitter, this phase-curve model can be selected by setting `phase_curve_style` to `sine_series` (see Table A1). The above terms are parameterized with

1. `companion/phase_curve_A1_inst` for beaming,
2. `companion/phase_curve_B1_inst` for atmospheric,
3. `companion/phase_curve_B2_inst` for 1st ellipsoidal,
4. `companion/phase_curve_B3_inst` for 2nd ellipsoidal.

(see Table A2). We do not include the terms $A_2$ and $A_3$, which have no physical interpretation.

The atmospheric component can further be separated into thermal and reflected components, both of which can receive a phase shift, using the expanded set of parameters explained in Table A2.

### 4.9.2. Phase Curves Using Transformed Sines

A drawback with the option above is that the pure harmonic series of waveforms requires that some semiamplitudes be negative to admit a physical interpretation. For example, a user might instead desire to fit for a “physical” full (peak-to-peak) amplitude of the atmospheric component.

By selecting the setting `phase_curve_style` to be `sine_physical` (see Table A1), the user can therefore model the phase curve with a linear combination of sinusoids while defining all quantities as physical quantities. The respective set of parameters is:

1. `companion/phase_curve_beaming_inst`: positive semiamplitude of the beaming effect, representing the term $A_1 \sin \phi(t)$, i.e., a modulation around the median flux level of the star.
2. `companion/phase_curve_atmospheric_inst`: positive full (peak-to-peak) amplitude of the atmospheric contribution, representing the term $-2B_1(1 - \cos(\phi(t)))$, i.e., an additive component to the companion’s night-side flux.
3. `companion/phase_curve_ellipsoidal_inst`: positive full (peak-to-peak) amplitude of the leading-order term of the ellipsoidal modulation, representing the term $-2B_2(1 - \cos(2\phi(t)))$, i.e., an additive component to the system’s flux from spherical (nondistorted) bodies.
4. `companion/phase_curve_ellipsoidal_2nd_inst`: positive full (peak-to-peak) amplitude of the next-order term of the ellipsoidal modulation, representing the term $-2B_3(1 - \cos(3\phi(t)))$, i.e., an additive component to the system’s flux from spherical (nondistorted) bodies.

As above, the atmospheric component can incorporate phase shifts and allows us to distinguish between thermal and reflected contributions, using the expanded set of parameters explained in Table A2.

### 4.9.3. Phase Curves with ellc’s Physical Model

An alternative way to model these effects with allesfitter is utilizing ellc’s relevant physical forward-model directly by using the setting `phase_curve_style` as `ellc`. The physical model is driven by parameters that affect the computation of the heated day side of the companion, `companion/heat_inst`, the gravity-darkening coefficients, `companion/gdc_inst`, and the Doppler-boosting factor `companion/bfac_inst`, as well as the desired stellar shape approximation (see Section 4.10 and Tables A1 and A2). As this approach requires a thorough understanding of the chosen settings and parameters, we only recommend it to users who are proficient with ellc. For a detailed description of all effects and caveats, we thus refer the reader to Maxted (2016).

#### 4.10. Stellar/Planetary Grid and Shape

The ellc implementation constructs all objects in the system as three-dimensional bodies and computes the light curve and RV forward-models by integrating over the visible surfaces. This allows a physically accurate representation of starspots and heat redistribution on the surface. The user can set the density of this interpolation grid using one of five options from `very_sparse` to `very_fine` (see Table A1). The available grid options have a strong impact on the computational speed but usually do not noticeably impact the results (see Maxted 2016). Thus, we recommend the user run all test runs with `very_sparse` and only run the publication-ready model with a finer spacing.

Additionally, the user can efficiently compute deviations of the stellar/planetary shape (see Table A1). The `sphere` option is the default and appropriate for any model that does not incorporate interaction between the objects. The `roche` shape calculates the object’s shape using the Roche equation (Wilson 1979). The `roche_v` shape is suited for synchronous rotation, where the volume of the star can be modeled via Kopal (1978). With `poly1p5` or `poly3p0`, the object is modeled as a polytrope with index $n = 1.5$ or $n = 3.0$, respectively (Chandrasekhar 1933; James 1964). Finally, the `love` option computes the objects’ shape via Correia (2014).

#### 4.11. Derived Parameters

In addition to the fitted parameters, allesfitter also uses the samples drawn from the posterior distribution of parameters to derive an extensive list of additional quantities. The full list is shown in Table A3, along with explanations on how these values are derived from the posterior samples.

### 5. Examples and Case Studies

#### 5.1. The Two-planet System Pi Mensae

In this section, we demonstrate how allesfitter can be used to infer the parameters of a multiplanet system from photometric and RV data from different telescopes. We also show how the Bayesian evidence can be used to compare different models for limb-darkening laws, eccentric versus circular orbits, and systematic noise.

For this, we first reanalyze TESS’ first exoplanet discovery, Pi Mensae c (H18), using TESS Sector 1 data only and compare our results with those from H18 (Section 5.1.1). Afterward, we include all newly available data from TESS Year 1 to update the literature values (Section 5.1.2).7

---

7 All data, code, and results available at https://github.com/MNGuenther/allesfitter/tree/master/paper/Pi_Mensae.
The Pi Mensae system hosts two known planets. The 10 $M_{\text{jup}}$ planet Pi Mensae b was originally discovered using RV surveys (Jones et al. 2002; Wittenmyer et al. 2012) on a highly eccentric 5.7 yr orbit. In 2018, shortly after its launch, TESS unveiled photometric transits of an inner companion, Pi Mensae c, only twice the size of Earth and on a 6.27 day orbit. We have already used allsefitter for independent analysis in the TESS discovery paper, and here we showcase our results in more depth, emphasizing additional aspects. We perform five different reanalyses of the Sector 1 data from H18 in global fits of all available photometric and RV data, i.e., their TESS Sector 1 spline-detrended light curve along with all archival RV data from HARPS and AAT. We define the following as our “standard” settings to reproduce the original study by H18: we use constant baselines for the TESS and RV data, assume a circular orbit for c, and apply a quadratic limb-darkening law. The only difference from H18 is that we uniformly sample in the transformed parameter space from Kipping (2013), while the original study set the quadratic limb darkening to tabulated values from Claret (2017).

5.1.1. Reanalysis of TESS Sector 1 and RV Data

The Pi Mensae system hosts two known planets. The 10 $M_{\text{jup}}$ planet Pi Mensae b was originally discovered using RV surveys (Jones et al. 2002; Wittenmyer et al. 2012) on a highly eccentric 5.7 yr orbit. In 2018, shortly after its launch, TESS unveiled photometric transits of an inner companion, Pi Mensae c, only twice the size of Earth and on a 6.27 day orbit. We have already used allsefitter for independent analysis in the TESS discovery paper, and here we showcase our results in more depth, emphasizing additional aspects. We perform five different reanalyses of the Sector 1 data from H18 in global fits of all available photometric and RV data, i.e., their TESS Sector 1 spline-detrended light curve along with all archival RV data from HARPS and AAT. We define the following as our “standard” settings to reproduce the original study by H18: we use constant baselines for the TESS and RV data, assume a circular orbit for c, and apply a quadratic limb-darkening law. The only difference from H18 is that we uniformly sample in the transformed parameter space from Kipping (2013), while the original study set the quadratic limb darkening to tabulated values from Claret (2017).
Table 4
Updated parameters from the all-sky fit of Pi Mensae, Using All Available data from TESS Year 1 (Sectors 1, 4, 8, 11, 12, and 13) as Well as All RV Data Used in Huang et al. (2018)

| Parameter                                                                 | Value                          | Source |
|---------------------------------------------------------------------------|-------------------------------|--------|
| Transformed limb darkness, $q_{1, TESS}$                                  | $0.46^{+0.18}_{-0.14}$        | fit    |
| Transformed limb darkness, $q_{2, TESS}$                                  | $0.21^{+0.25}_{-0.15}$        | fit    |
| Epoch, $T_{0, b}$ (BJD TDB)                                              | $2456552.4 \pm 2.5$           | fit    |
| Period, $P_b$ (days)                                                     | $2093.1 \pm 1.8$              | fit    |
| RV semi-amplitude, $b$, $K_b$ (km s$^{-1}$)                               | $0.1926 \pm 0.0013$           | fit    |
| Eccentricity term, $b$, $\sqrt{e} \cos \omega_b$                        | $0.6956 \pm 0.0043$           | fit    |
| Eccentricity term, $b$, $\sqrt{e} \sin \omega_b$                        | $-0.3919 \pm 0.0060$          | fit    |
| Sum of radii over semimajor axis, $c$, ($R_c + R_s$)/a                    | $0.0761^{+0.0019}_{-0.0016}$  | fit    |
| Radius ratio, $c/R_e$                                                     | $0.01696 \pm 0.000023$        | fit    |
| Cosine of inclination term, $\cos i$                                    | $0.0427^{+0.0033}_{-0.0031}$  | fit    |
| Epoch, $T_{0, e}$ (BJD TDB)                                              | $2458501.03034^{+0.00035}_{-0.00039}$ | fit |
| Period, $P_e$ (days)                                                     | $6.267850 \pm 0.000018$       | fit    |
| RV semi-amplitude, $c$, $K_c$ (km s$^{-1}$)                               | $0.00153 \pm 0.000028$        | fit    |
| Eccentricity term, $c$, $\sqrt{e} \cos \omega_c$                        | $0.0$                         | fixed  |
| Eccentricity term, $c$, $\sqrt{e} \sin \omega_c$                        | $0.0$                         | fixed  |
| GP, ln $\sigma_{TESS}$ (In rel. flux)                                    | $-10.471 \pm 0.037$           | fit    |
| GP, ln $\rho_{TESS}$ (In days)                                           | $-1.98 \pm 0.11$              | fit    |
| RV offset, $\Delta RV_{AAT}$ (km s$^{-1}$)                               | $0.03198 \pm 0.000085$        | fit    |
| RV offset, $\Delta RV_{HARPS}$ (km s$^{-1}$)                             | $10.7084 \pm 0.00038$         | fit    |
| RV offset, $\Delta RV_{HARPS}$ (km s$^{-1}$)                             | $10.73058 \pm 0.00069$        | fit    |
| Nat. log. error scaling, ln $\sigma_{TESS}$ (In rel. flux)                | $-8.6313 \pm 0.0024$          | fit    |
| Nat. log. jitter term, ln $\sigma_{AAT}$ (In km s$^{-1}$)                 | $-5.013^{+0.094}_{-0.087}$    | fit    |
| Nat. log. jitter term, ln $\sigma_{HARPS}$ (In km s$^{-1}$)               | $-6.041 \pm 0.078$            | fit    |
| Nat. log. jitter term, ln $\sigma_{HARPS}$ (In km s$^{-1}$)               | $-6.40^{+0.321}_{-0.18}$      | fit    |

Derived parameters

| Parameter                                                                 | Value                          | Source |
|---------------------------------------------------------------------------|-------------------------------|--------|
| Eccentricity $b$, $e_b$                                                   | $0.6375 \pm 0.0024$           | derived|
| Arg. of periastron $b$, $\omega_b$ (deg)                                  | $330.60 \pm 0.53$             | derived|
| Period ratio, $P_h/P_e$                                                   | $333.95 \pm 0.29$             | derived|
| Host radius over semimajor axis, $R_s/a_c$                                | $0.0749^{+0.0013}_{-0.0016}$  | derived|
| Semimajor axis $c$ over host radius, $a_c/R_s$                            | $13.36^{+0.029}_{-0.031}$     | derived|
| Planet radius $c$ over semimajor axis $c$, $R_c/a_c$                      | $0.00127^{+0.00007}_{-0.000037}$ | derived |
| Planet radius $c$, $R_c$                                                   | $2.035 \pm 0.052$             | derived|
| Semimajor axis of $c$, $a_c (R_c)$                                       | $14.67 \pm 0.46$              | derived|
| Semimajor axis of $c$, $a_c$ (au)                                        | $0.0682 \pm 0.0021$           | derived|
| Inclination $c$, $i_c$ (deg)                                              | $87.5^{+0.18}_{-0.19}$        | derived|
| Planet mass $c$, $M_c (M_\odot)$                                        | $4.71^{+0.00}_{-0.05}$        | derived|
| Impact parameter $c$, $b_{impact}$                                       | $0.571 \pm 0.031$             | derived|
| Total transit duration, $T_{tot, c}$ (h)                                  | $3.029^{+0.032}_{-0.033}$     | derived|
| Full-transit duration, $T_{full, c}$ (h)                                  | $2.870^{+0.031}_{-0.032}$     | derived|
| Stellar density from orbit $c$, $\rho_c (\mathrm{cgs})$                   | $1.148^{+0.072}_{-0.042}$     | derived|
| Planet density $c$, $\rho_c (\mathrm{cgs})$                              | $3.06^{+0.07}_{-0.06}$        | derived|
| Planet surface gravity $c$, $g_\ast (\mathrm{cgs})$                      | $1100^{+200}_{-200}$          | derived|
| Equilibrium temperature $c$, $T_{eq, c}$ (K)                              | $1069^{+15}_{-11}$            | derived|
| Transit depth $c$, $\delta u_c, TESS$ (ppt)                             | $0.3204^{+0.0080}_{-0.0069}$  | derived|
| Limb darkening $u_1, TESS$                                                | $0.28^{+0.20}_{-0.20}$        | derived|
| Limb darkening $u_2, TESS$                                                | $0.40^{+0.28}_{-0.33}$        | derived|

We find a good fit to the data (Figure 1), and all results from our different model variations agree well with one another and with those published by H18; MCMC and Nested Sampling give consistent results for the standard settings. Furthermore, comparing the Bayesian evidence of all Nested Sampling model fits, we find that the model with a GP Matérn 3/2 baseline is strongly favored (Figure 2). This is likely because the TESS Sector 1 data from H18 were affected by remnant systematics on timescales shorter than those removed by the original spline detrending. These short-term systematics are possibly caused by the satellite’s pointing jitter, which is now well characterized and understood. Moreover, we find that the data justify the circular orbit assumption for planet c and the choice of a quadratic limb-darkening model.
5.1.2. New Analysis of All TESS Year 1 and RV Data

Finally, we go beyond mere comparison with the discovery paper and update the literature values for Pi Mensae by analyzing all available TESS data from the first year of operations, i.e., observations from Sectors 1, 4, 8, 11, 12, and 13, along with all RV data used in H18. Due to the bright host star, we use custom-aperture light curves, which are detrended against the quaternions and the first seven components of the cotrending basis vectors (custom light curves courtesy of Chelsea X. Huang). Our *allesfit* approach is equivalent to variation 5 in Section 5.1.1 (i.e., the circular orbit of planet c, quadratic limb darkening, GP baseline, Nested Sampling). The resulting fit is shown in Figure 1, and all results are summarized in Table 4 and Figure A1.

We again find a good agreement with the discovery paper, along with a significant improvement in the median and precision of planet c’s orbital period, as expected from the extended observing baseline (see Figure 3). We also find that the updated detrending of the full TESS Year 1 light curve, now incorporating all state-of-the-art understanding of systematics, removed the remnant short-term noise that was picked up by the GP in Section 5.1.1. Hence, for the Year 1 analysis, the GP baseline turns out flat and is comparable to a constant offset. This also marginally updates our posteriors of the radius ratio and limb darkening.

5.2. TTVs in the TOI-216 System

*Allesfit* allows fitting a global light-curve model with individual transit/eclipse midtime offsets for each transit event, even if those occur for multiple companions and were observed by different telescopes. We highlight these abilities on the example of the two-planet system TOI-216 (TIC 55652896), the first discovery by TESS that shows clear TTVs (Dawson et al. 2019; Kipping et al. 2019). From only the first few months of TESS data, the system has been characterized to contain a pair of warm, large exoplanets. These planets orbit at mean periods near 17.1 and 34.5 days, close to a 2:1 mean-motion resonance. Dawson et al. (2019), in particular, analyze the TESS Sectors 1–6 TTVs and find two families of solutions for the masses of planet b and c, respectively: either like a sub-Saturn and Neptune, or like a Jupiter and sub-Saturn.

| Parameter | Value | Source |
|-----------|-------|--------|
| Radius ratio b, $R_b/R_c$ | $0.0846^{+0.0030}_{-0.0029}$ | fit |
| Sum of radii over semimajor axis b, $(R_b + R_a)/a_b$ | $0.0349^{+0.0041}_{-0.0028}$ | fit |
| Cosine of inclination b, $\cos i_b$ | $0.0304^{+0.0044}_{-0.0029}$ | fit |
| Linear-ephemerides epoch b, $T_{0,b}$ (BJD$_{TDB}$) | 2458496.1366 | fixed |
| Linear-ephemerides period b, $P_s$ (days) | 17.0714 | fixed |
| Sum of radii over semimajor axis c, $R_c/R_s$ | $0.12332 \pm 0.00077$ | fit |
| Radius ratio c, $(R_c + R_a)/a_c$ | $0.02091^{+0.0026}_{-0.0018}$ | fit |
| Cosine of inclination c, $\cos i_c$ | $0.0020^{+0.0015}_{-0.0011}$ | fit |
| Linear-ephemerides epoch c, $T_{0,c}$ (BJD$_{TDB}$) | 2458504.0408 | fixed |
| Linear-ephemerides period c, $P_c$ (days) | 34.5555 | fixed |
| Transformed limb darkening, $q_{1;TESS}$ | $0.351^{+0.13}_{-0.10}$ | fit |
| Transformed limb darkening, $q_{2;TESS}$ | $0.39^{+0.14}_{-0.11}$ | fit |
| GP: $\ln \sigma_{TESS}$ (ln rel. flux) | $-7.507 \pm 0.037$ | fit |
| GP: $\ln \rho_{TESS}$ (ln days) | $-0.262 \pm 0.063$ | fit |
| Nat. log. error scaling, $\log \sigma_{TESS}$ | $-6.0019 \pm 0.0014$ | fit |

### Derived parameters

| Parameter | Value | Source |
|-----------|-------|--------|
| Host radius over semimajor axis b, $R_b/a_b$ | $0.0320^{+0.0028}_{-0.0023}$ | derived |
| Semimajor axis b over host radius, $a_b/R_c$ | $3.13 \pm 2.5$ | derived |
| Planet radius b over semimajor axis b, $R_b/a_b$ | $0.0027^{+0.0013}_{-0.00047}$ | derived |
| Planet radius b, $R_b$ (R$_J$) | $3.53^{+0.04}_{-0.05}$ | derived |
| Semimajor axis b, $a_b$ (R$_J$) | $11.9^{+0.23}_{-0.25}$ | derived |
| Semimajor axis b, $a_b$ (au) | $0.055^{+0.0054}_{-0.0050}$ | derived |
| Inclination b, $i_b$ (deg) | $88.26^{+11.12}_{-14.37}$ | derived |
| Impact parameter b, $b_{mb}$ | $0.95^{+0.031}_{-0.026}$ | derived |
| Total transit duration b, $T_{trans}$ (h) | $2.163 \pm 0.068$ | derived |
| Stellar density from orbit b, $\rho_{b,\star}$ (cgs) | $1.99^{+0.51}_{-0.44}$ | derived |
| Equilibrium temperature b, $T_{eq,b}$ (K) | $392 \pm 22$ | derived |
| Transit depth b, $c_{b,\star}$ TESS (ppt) | $0.0044^{+0.00014}_{-0.00017}$ | derived |
| Host radius over semimajor axis c, $R_c/a_c$ | $0.0186^{+0.00023}_{-0.00016}$ | derived |
| Semimajor axis c over host radius, $a_c/R_s$ | $53.72^{+0.47}_{-0.66}$ | derived |
| Planet radius c over semimajor axis c, $R_c/a_c$ | $0.00229^{+0.00036}_{-0.00018}$ | derived |
| Planet radius c, $R_c$ (R$_J$) | $5.11 \pm 0.27$ | derived |
| Semimajor axis c, $a_c$ (au) | $20.4 \pm 1.1$ | derived |
| Semimajor axis c, $a_c$ (R$_J$) | $0.0948 \pm 0.0052$ | derived |
| Inclination c, $i_c$ (deg) | $89.88^{+0.066}_{-0.089}$ | derived |
Here, we analyze TOI-216 with `allesfitter` while freely fitting for the transit midtimes, with the goal of deriving all planetary and orbital parameters including a TTV diagram (i.e., observed minus calculated).\(^9\) We include a total of 12 sectors of TESS data, which have been collected for this target by now (Sectors 1–9 and 11–13), doubling the original baselines of the discovery papers.

For TTV fitting with `allesfitter`, we want to keep the epoch and period of a linear transit model fixed at the median value. In this case, we simply estimate them by linearly interpolating between the first and last recorded transit of each planet. It is not important to have an exact value here, as all individual transit midtimes will be freely sampled in the fit. With these transit midtimes, we could later model the exact

---

\(^9\) All data, code, and results available at https://github.com/MNGuenther/allesfitter/tree/master/paper/TOI-216.

---

### Table 5 (Continued)

| Parameter | Fitted parameters | Value | Source |
|-----------|-------------------|-------|--------|
| Impact parameter $c$, $b_{\text{rise}}$ | | $0.110^{+0.081}_{-0.062}$ | derived |
| Total transit duration $c$, $T_{\text{tot,c}}$ (h) | | $5.487^{+0.036}_{-0.032}$ | derived |
| Full-transit duration $c$, $T_{\text{full,c}}$ (h) | | $4.262^{+0.036}_{-0.033}$ | derived |
| Stellar density from orbit $c$, $\rho_{c,c}$ (cgs) | | $2.456^{+0.060}_{-0.089}$ | derived |
| Equilibrium temperature $c$, $T_{\text{eq,c}}$ (K) | | $299 \pm 12$ | derived |
| Transit depth $c$, $d_{c,c, \text{TESS}}$ (ppt) | | $0.01829^{+0.00014}_{-0.00013}$ | derived |
| Limb darkening $u_1, \text{TESS}$ | | $0.466 \pm 0.075$ | derived |
| Limb darkening $u_2, \text{TESS}$ | | $0.12^{+0.17}_{-0.16}$ | derived |
| Median stellar density from orbits, $\rho_c$ (cgs) | | $2.39^{+0.13}_{-0.03}$ | derived |

### Table 6

Updated Transit Midtimes and TTV $O - C$ Values for TOI-216 from All TESS Year 1 Data (Sectors 1–9 and 11–13)

| Transit Midtime (BJDTDB) | TOI-216 b | $O - C$ (min.) |
|-------------------------|-----------|----------------|
| 2458325.3202 ± 0.0023  | −0.0857 ± 0.0023 |
| 2458342.4307 ± 0.0022  | −0.0476 ± 0.0022 |
| 2458359.5391 ± 0.0019  | −0.0115 ± 0.0019 |
| 2458376.6313 ± 0.0019  | 0.0084 ± 0.0019 |
| 2458393.7232 ± 0.0022  | 0.0280 ± 0.0022 |
| 2458427.8792 ± 0.0021  | 0.0392 ± 0.0021 |
| 2458444.9571 ± 0.0026  | 0.0448 ± 0.0026 |
| 2458462.0308 ± 0.0025  | 0.0462 ± 0.0025 |
| 2458479.0941 ± 0.0026  | 0.0372 ± 0.0026 |
| 2458496.1550 ± 0.0026  | 0.0257 ± 0.0026 |
| 2458513.2250 ± 0.0033  | 0.0234 ± 0.0033 |
| 2458547.3777 ± 0.0030  | −0.0086 ± 0.0028 |
| 2458564.4029 ± 0.0028  | −0.0157 ± 0.0028 |
| 2458615.6037 ± 0.0030  | −0.0319 ± 0.0031 |
| 2458632.6794 ± 0.0029  | −0.0286 ± 0.0029 |
| 2458649.7588 ± 0.0030  | −0.0215 ± 0.0030 |
| 2458666.8508 ± 0.0026  | −0.0018 ± 0.0026 |

| TOI-216 c |
|-----------|----------------|
| 2458331.28513 ± 0.00058 | 0.01593 ± 0.00058 |
| 2458365.82452 ± 0.00060 | 0.00125 ± 0.00060 |
| 2458400.36849 ± 0.00056 | −0.0086 ± 0.00056 |
| 2458434.92246 ± 0.00054 | −0.0089 ± 0.00054 |
| 2458469.47727 ± 0.00078 | −0.0022 ± 0.00078 |
| 2458538.59217 ± 0.00069 | −0.00148 ± 0.00069 |
| 2458607.70834 ± 0.00071 | 0.00654 ± 0.00071 |
| 2458642.26111 ± 0.00095 | 0.00523 ± 0.00095 |
| 2458676.80853 ± 0.00070 | −0.00142 ± 0.00070 |
Figure 4. Global fit with free TTVs to the TOI-216 system from all TESS Year 1 data (Sectors 1–9 and 11–13). The system hosts two warm, large exoplanets near a mean-motion resonance of 2:1. All shown data are TESS 2 minute cadence observations. The left and middle columns show transit windows for TOI-216 b, the right column those for TOI-216 c. Red curves show 20 fair samples drawn from the posterior of the global model, including free TTVs for each transit. Orange curves show 20 fair samples drawn from the posterior of the GP Matérn 3/2 baseline model.
linear signal and TTV modulation using dedicated software, such as $N$-body simulations for TTV signals.

We uniformly sample from the posterior of the radius ratios $R_b/R_*$ and $R_c/R_*$, sums of radii over semimajor axis $(R_b + R_*)/a$ and $(R_c + R_*)/a$, cosines of the inclination $\cos i$, and quadratic limb darkening in the Kipping (2013) transformation $q_1$ and $q_2$, a GP Matérn 3/2 baseline with parameters $\ln \rho_{GP}$ and $\ln \sigma_{GP}$, and the white noise error scaling $\ln \sigma_r$. We first fit the GP Matérn 3/2 model to the out-of-transit data and then apply normal priors on it for the fit to the in-transit data (see e.g., Günther et al. 2019).

Notably, the grazing transit of planet b leads to a degeneracy between the radius ratio and orbital inclination, which can lead to a “runaway” solution if using wide uniform priors and no external constraints. A possible way to overcome this is by imposing an external planet density prior (e.g., Bayliss et al. 2018). For this example, however, we chose to follow the approach by Dawson et al. (2019) and constrain the radius ratio to a uniform prior between 0 and 0.17, because the “runaway” solution starts around 0.2.

We find a good fit to the data, and our results agree well with those from Dawson et al. (2019) and Kipping et al. (2019). The per-transit light curves and posterior models are shown in Figure 4. By including all available TESS Year 1 data and hence doubling the baseline from Dawson et al. (2019) and Kipping et al. (2019), we can also update the TTV $O-C$ diagrams, as shown in Figure 5. All posteriors are summarized in Table 5 for updated physical and orbital parameters, Table 6 for updated transit midtimes and TTV $O-C$ values, and Figure A2 for posterior corner plots.

5.3. The Phase Curve of WASP-18b

Allesfiter can also model phase curves of exoplanets and binary stars, which we demonstrate here on the example of the hot Jupiter WASP-18 b (TIC 100100827; Hellier et al. 2009; Southworth et al. 2009). The system harbors a 10 $M_{Jup}$ companion on a short orbital period of 0.94 days. This extreme combination leads to interactions between the star and planet that cause a phase-curve signature at visible wavelengths. In turn, studying this phase curve gives insight into the atmosphere of this hot Jupiter. As for Pi Mensae (see above), allesfiter was already used to perform an independent analysis for the original TESS study by S19, and we here showcase how such an analysis can be performed.

For this example, we speed up our analysis by phase-folding the TESS light curve on an epoch of 2458361.048072 BJD$_{TDB}$ and a period of 0.9414576 days, which are posterior medians from our preliminary analysis. We then bin the phase curve over a grid of 1000 points in phase, which corresponds to a bin width of 1.4 minutes. We perform two fits, one with the “sine_series” phase-curve model (as in S19) and the other with the “sine_physical” phase-curve model. We uniformly sample from the posterior of the radius ratio $R_b/R_*$, the sum of radii over semimajor axis $(R_b + R_*)/a$, cosine of the inclination $\cos i$, surface brightness ratio $J$ of the planet’s day side and star, the Doppler-boosting (beaming) effect $A_1$ in sine_series, $A_{\text{beaming}}$ in sine_physical, the atmospheric modulation from thermal emission and reflected light $B_1$ in sine_series, $A_{\text{atmospheric}}$ in sine_physical, the ellipsoidal modulation $B_2$ in sine_series, $A_{\text{ellipsoidal}}$ in sine_physical), a constant baseline offset $\Delta F$, and the white noise error scaling $\ln \sigma_r$. As we here only fit photometric TESS data, we also apply prior information from RV observations. For simplicity in this example, we fix the eccentricity to $e = 0.0091$ and the argument of periastron to $\omega = 269^\circ$ (Knutson et al. 2014; Stassun et al. 2017). We run an MCMC analysis starting from the values found by previous studies, with 500 walkers, a thinning of 50 steps, 1000 burn-in steps, and 5000 total steps. We consider the fits to be converged as all chains are $\gg 42 \times$ their autocorrelation lengths.

We find a good fit to the TESS light curve of WASP-18 (Figure 6) and a good agreement with the results from S19 (Figure A3) with both phase-curve settings. In particular, we can individually interpret the components of the phase-curve forward-model. The ellipsoidal modulation in our sine_series model has a semi-major axis of $192.2 \pm 5.9$ ppm ($190.5^{+10}_{-5.7}$ ppm in S19). We also find evidence for Doppler boosting, with a semi-major axis of $22.1 \pm 4.5$ ppm ($21.0 \pm 4.5$ ppm in S19). There is a slight difference in our semi-major axis of the atmospheric phase modulation $-144.3 \pm 5.6$ ppm and radius ratio of $0.09757 \pm 0.00014$ compared to S19 $(-174.4^{+0.6}_{-0.5}$ ppm and $0.09716^{+0.00014}_{-0.00014}$, respectively). This is likely caused by our simplified example (phase-folded and binned data, fixed parameters, constant offset baseline) and the fact that S19 also fit a polynomial background model and additional higher-order sinusoidal harmonics. Fitting for the surface brightness ratio of the planet’s day side and star, we find $J = 0.0056 \pm 0.0016$, an occultation depth of $342 \pm 15$ ppm ($341^{+1}_{-17}$ ppm in S19).
5.4. The Spotted Binary Star System KOI-1003

In this example, we show how \texttt{allesfitter} can be used to infer parameters for binary star models in the presence of stellar variability and for long-cadence data, using the example of KOI-1003 (TIC 122374527).\footnote{All data, code, and results available at https://github.com/MNGuenther/allesfitter/tree/master/paper/KOI-1003.} KOI-1003 is an active, spotted binary star system (Roettenbacher et al. 2016, hereafter R16) and is classified as an RS Canum Venaticorum (RS CVn) binary. Such systems are close binaries, where the primary is an evolved giant or subgiant that partially fills its Roche potential and the secondary is a fainter main sequence. The star was observed in Kepler Quarters 2–17 nearly continuously with 29.4 minutes cadence. R16 found that the binary’s orbital and stellar rotation periods are nearly synchronized at 8.36 and 8.23 days, respectively. To create a fast-running example for the user, we here only utilize data from the first 28 days of Quarter 2, covering three primary eclipses and three stellar rotation periods.

We use this example to illustrate an approach to tackling similar systems in multiple steps. In KOI-1003, the stellar variability is the dominant component of the observed light curve. To model it, we use \texttt{allesfitter} to mask out the eclipse regions and to fit an SHO GP along with the white noise scaling (see Section 4.6). As initial guesses for the white noise scaling and SHO frequency, we use Kepler’s median flux error and $2\pi/(8.23/2)$ days, respectively. We use half the rotation period, as two large opposite spots are apparent in the light curve. The initial guesses for the SHO amplitude and damping factor are set to small values, enforcing a smooth GP as the starting point for the MCMC. Our \texttt{allesfitter} run uses 500 walkers and performs one preliminary run with only 1000 steps to obtain relatively high-likelihood initial guesses for the nominal run. It then runs 1000 steps of burn-in and 5000 total steps, all thinned by a factor of 10, leading to 40,000 samples after convergence ($>47 \times$ autocorrelation length).

Second, we utilize the trained GP to remove the stellar variability from the light curve. We investigate if the shallow secondary eclipse can be detected in the detrended light curve despite the short range of data (expected depth of 1.8 ppt from R16). To this end, we use \texttt{allesfitter}’s interface to call the \texttt{transit least squares} algorithm (Hippke & Helmer 2019). We detect the primary eclipse with a period of 8.36 days and depth of $\sim$28 ppt at an $S/N = 38.2$, and a second signal with a period of 8.7 days and depth of $\sim$2 ppt at $S/N = 5.3$. We consider this to be likely related to the secondary eclipse, which is only a weak signal given the short range of data.

Third, we use the information gained above and perform a full model fit of the system with MCMC. We uniformly sample from the posterior of the radius ratio $R_B/R_A$, the sum of radii over semimajor axis $(R_B + R_A)/a$, cosine of the inclination cos$i$, quadratic limb darkening in the Kipping (2013) transformation $q_1$ and $q_2$, surface brightness ratio $J$, the GP hyperparameters, and the white noise error scaling $\ln \sigma_F$. We fix the eccentricity and argument of periastron as $\sqrt{e} \cos \omega$ and $\sqrt{e} \sin \omega$ to the values provided by R16, as our short data range does not reliably constrain the secondary eclipse. We set the initial guesses for the physical values close to those by R16, and those for the GP and white noise scaling to the posterior medians obtained in the first step. Because we analyze long-cadence data (29.4 minutes), we also use a 10 times finer evaluation grid to interpolate each point. The MCMC is run with 500 walkers, 1000 steps of pre-run, 2000 steps of burn-in, and 10,000 total steps, all thinned by a factor of 100, leading to 40,000 samples after convergence ($>33 \times$ autocorrelation length).

We find a good fit to the data which, despite the short data range, agrees well with the results from R16 (Figures 7 and A4). In particular, we find a period of $8.35992 \pm 0.00094$ days ($8.360613 \pm 0.000003$ days in R16), an inclination of $85.75 \pm 0.31$ (86.0 $\pm$ 0.5), and the ratio of the semimajor axis to the primary radius of $8.23 \pm 0.19$ (8.2 $\pm$ 0.5 in R16). We do find a significantly lower radius ratio of $0.1634_{-0.0021}^{+0.0017}$ ($0.177 \pm 0.003$ in R16) in this particular region of data, which is likely caused by spot crossings, i.e., the alignment of the planet with the stellar spots during the transit. This agrees with the fact that R16 found individual transit depths to vary between 2.73% and 4.59% due to spot crossings. Going forward, our modeling of these spot crossings could be refined by using a physical spot model (demonstrated in Section 5.5) or including an additional short-term GP, e.g., using a Matérn 3/2 kernel.
Most importantly, by directly fitting for the eclipsing binary’s surface brightness ratio using a physical forward-model, we find $J = 0.053 \pm 0.012$, which could be used to constrain the spectral type of the secondary. The derived secondary eclipse depth of $1.40 \pm 0.31$ ppt agrees well with R16 ($1.76 \pm 0.12$ ppt in R16) and confirms the detection of the secondary eclipse.

5.5. Starspots and Flares on GJ 1243

In addition to modeling eclipses of stars and exoplanets, allesfitter also models starspots and stellar flares. While starspots can cause semisinusoidal variations in the light curve as fainter regions rotate in and out of the visible disk, stellar flares cause an abrupt rise and subsequent exponential decay in the stellar brightness. Joint modeling of these effects can be relevant when stars exhibit both processes simultaneously, as often is the case for active M dwarfs.

Here, we demonstrate allesfitter’s abilities on the example of GJ 1243. A M4 dwarf star is one of the most frequently flaring stars known and was extensively studied with Kepler data (Davenport et al. 2014, 2015; Hawley et al. 2014; Silverberg et al. 2016). These studies found evidence for a 0.59 day rotation period, differential rotation, and starspot evolution in four years of Kepler data, along with a high flare frequency. TESS recently reobserved the system during its Sector 14 and 15.

We analyze a 1.8 day (three rotation periods) snapshot of TESS observations and simultaneously fit for starspots and stellar flares in this part of the light curve. We again use different approaches and compute the Bayesian evidence to compare the models:

1. two starspots and three flares,
2. one starspot and three flares,
3. two starspots and two flares.

Using Nested Sampling, we uniformly sample from the posterior of the rotation period; the starspots’ longitudes, latitudes, relative brightness, and sizes; flares’ peak times, amplitudes and FWHMs; white noise scaling; and a constant baseline. Silverberg et al. (2016) reported a spectroscopic $v \sin i \approx 25 \text{ km s}^{-1}$ and stellar radius of $R_s \approx 0.36 R_\odot$. Using also the photometric rotation period $P_{\text{rot}} \approx 0.59$ days, we can compute

$$i = \sin^{-1} \left( \frac{v \sin i P_{\text{rot}}}{2 \pi R_s} \right) \approx 54^\circ,$$

where we freeze the inclination in our fit12.

We find that the model with two starspots and three flares describes the data best, according to the Bayes factors (Figures 8 and 9). In this model, the primary spot lies close to the pole (longitude $345.3^\circ \pm 2.7^\circ$, latitude $79.5^\circ \pm 1.6^\circ$) with an angular radius of $3.51_{-1.1}^{+0.72} \circ$ and a relative brightness of $0.561_{-0.077}^{+0.060}$ compared to the stars surface brightness in the TESS band (Figure 10). This puts it at an effective temperature of about 2900 K. In comparison, the host star’s temperature is about 3300 K (Stassun et al. 2017). The second spot lies slightly closer to the equator (longitude $215.1^\circ \pm 1.6^\circ$, latitude $28.8_{-6.8}^{+7.6}^\circ$), and is smaller and darker (angular radius $5.44_{-0.39}^{+0.52}$, relative brightness $0.22_{-0.13}^{+0.15}$), corresponding to a spot temperature of about 2500 K. The three flares we identify have amplitudes of 10%, 4%, and 3%, respectively, with the first two flares appearing in sequence and overlapping each other. These “outbursts” of multiple, subsequent flares are common on active M dwarfs and can, to some extent, be disentangled into individual flares using Bayesian evidence (Günther et al. 2020), as also demonstrated here (Figures 8 and A5).

12 Note that the $31^\circ$ reported in Silverberg et al. (2016) and Davenport et al. (2015) are apparently erroneous and should have been $54^\circ$.

---

Figure 7. KOI-1003 model fit with stellar variability, overplotted with a 28 day snapshot of the Kepler long-cadence light curve of this spotted binary system (blue points). Red and orange lines show forward-model light curves generated using 20 fair draws from the posterior of the eclipse and GP models.
6. Summary and Conclusion

In this work, we introduced the allesfitter package to perform a global inference based on photometric and RV data. allesfitter unites various robust and well-tested generative models of exoplanets and stars to perform parameter inference and model testing. It provides a flexible graphical user interface as well as a Python API. We illustrated a range of analyses to exemplify use cases, including multiplanet systems on eccentric orbits, TTVs, phase curves, eclipsing binaries, starspots, and stellar flares. In all cases, we found a good agreement with the original studies.

We acknowledge helpful discussions with Edward Gillen, Néstor Espinoza, Josh Speagle, Pierre Maxted, Daniel Foreman-Mackey, Ismael Mireles, Mariona Badenas-Agusti, and the TESS team during the development of allesfitter and its various predecessors. M.N.G. acknowledges support from MIT’s Kavli Institute as a Torres postdoctoral fellow. T.D. acknowledges support from MIT’s Kavli Institute as a Kavli postdoctoral fellow. Allesfitter is written in Python 3 (van Rossum 1995). In addition to the above-mentioned model-specific packages, it also makes use of the open-source software corner (Foreman-Mackey 2016), transitleastsquares (Hippeke & Heller 2019), wotan (Hippeke et al. 2019), numpy (van der Walt et al. 2011), scipy (Jones et al. 2001), matplotlib (Hunter 2007), tqdm (da Costa-Luis et al. 2021), and seaborn (https://seaborn.pydata.org/index.html).

Funding for the TESS mission is provided by NASA’s Science Mission Directorate. We acknowledge the use of public TESS data from pipelines at the TESS Science Office and at the TESS Science Processing Operations Center. This research has made use of the Exoplanet Follow-up Observation Program website, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program. Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center for the production of the SPOC data products. This paper includes data collected by the TESS mission that are publicly available from the Mikulski Archive for Space Telescopes (MAST).

Appendix

Figures A1–A5 are corner plots of the analyses presented in Sections 5.1–5.5, respectively. Tables A1 and A2 describe the input parameters and settings, respectively, which a user can provide in allesfitter.
Table A1
All Possible Settings for allesfitter, Which Can Be Given in the settings.csv File

| Setting                  | Explanation                                                                 | Default       |
|--------------------------|-----------------------------------------------------------------------------|---------------|
|                          |                                                                             |               |
| **General settings**     |                                                                             |               |
| companions_phot          | The companion(s) in the photometric data, space separated                    |               |
|                          | Example: companions_phot,b c e                                              |               |
| companions_rv            | The companion(s) in the radial velocity data, space separated                |               |
|                          | Example: companions_rv,B                                                    |               |
| inst_phot                | The photometric instrument(s), space separated                              |               |
|                          | Example: inst_phot,TESS                                                    |               |
| inst_rv                  | The radial velocity instrument(s), space separated                          |               |
|                          | Example: inst_rv,HARPS ESPRESSO                                             |               |
| multiprocess             | Use multiprocessing (True/False)                                           | False         |
| multiprocess_cores       | Number of cores for multiprocessing (1,2,3,...,all)                         | 1             |
| fast_fit                 | Mask out the out-of-transit data (True/False)                               | False         |
| fast_fit_width           | If using fast fit, select the window size around the transit (in days)      | 0.33333       |
| secondary_eclipse        | If using fast fit, also keep a window around phase 0.5 (True/False)        | False         |
| phase_curve              | Generate output and figures for phase curves (True/False)                  | False         |
| phase_curve_style        | Which phase-curve model to use (see Section 4.9; None/sine_physical/sine_series/ellc_physical) | None |
| shift_epoch              | Shift the epoch into the middle of the data set (True/False)                | False         |
| inst_for_[comp]_epoch    | If using shift epoch, which data files should be used                       |               |
|                          |                                                                             |               |
| **MCMC settings**        |                                                                             |               |
| mcmc_nwalkers            | Number of MCMC walkers                                                      | 100           |
| mcmc_total_steps         | Total steps in the MCMC chain, including burn-in steps                      | 2000          |
| mcmc_burn_steps          | Burn-in steps in the MCMC chain                                            | 1000          |
| mcmc_thin_by             | Only save every nth step in the MCMC chain                                  | 1             |
| mcmc_pre_run_steps       | Run n steps of pre-burn-in to refine the initial guess                      | 0             |
| mcmc_pre_run_loops       | Run m loops with the above n steps of pre-burn-ins                          | 0             |
| mcmc_moves               | Algorithm to update the positions of walkers (RedBlueMove/StretchMove/ WalkMove/KDEMove/DEMove/DESnoo- kerMove/MHMover/GaussianMove) | DEMove       |
|                          |                                                                             |               |
| **Nested Sampling settings** |                                                                             |               |
| ns_modus                 | Nested Sampling algorithm (static/dynamic)                                 | dynamic       |
| ns_alive                 | Number of live points                                                      | 500           |
| ns_bound                 | Method to bound the prior (None/single/multiballs/cubes)                    | single        |
| ns_sample                | Method to update live points (auto/unif/rwalk/rstagger/slice/rslice/hslice) | rwalk         |
| ns_tol                   | Tolerance of the convergence criterion                                      | 0.01          |
|                          |                                                                             |               |
| **External priors: stellar host density** | Use an external normal prior on the host density (see Section 4.8; True/False) | True         |
|                          |                                                                             |               |
| **Limb darkening per object and instrument** | Limb-darkening law for the host (None/logn/sqrt/expholfs/logquad/sing/claret for the constant/linear/square-root/exponential/logarithmic/quadratic/three-parameter/four-parameter laws from Table 2) | None         |
| host_ld_law_.[inst]      |                                                                             |               |
| [comp].ld_law_.[inst]    | Limb-darkening law for a companion (as above)                               | None          |
| host_ld_space_.[inst]    | Limb-darkening sampling space for the host (q/a)                            | q             |
| [comp].ld_space_.[inst]  | Limb-darkening sampling space for a companion (q/a)                         | q             |
|                          |                                                                             |               |
| **Baseline settings per instrument** | The baseline method used per instrument (see Section 4.6, none/sample_offset/sample_linear/sample_GP_real/ sample_GP_complex/sample_GP_Matern32/sample_GP_SHO/sample_offset/hybrid_offset/hybrid_poly_1/hybrid_poly_2/hybrid_poly_3/hybrid_poly_4/hybrid_spline) | none         |
| baseline_[key]_[inst]   |                                                                             |               |
|                          |                                                                             |               |
| **Error settings per instrument** |                                                                             |               |

19
| Setting                     | Explanation                                                                                                                                                                                                 | Default |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| error_[key]_[inst]          | The white noise method per instrument, which either scales the noise for photometry or adds a jitter term in quadrature for RV data (see Section 4.5; sample/hybrid)                                                 | sample  |
| t_exp_[inst]                | Exposure time of the instrument (in days); crucial for long exposures or binned data, to sample a high-cadence light-curve model and bin it up to match the data binning Example for 30 minute cadence: t_exp_[inst].0.0208333 | None    |
| t_exp_n_int_[inst]         | Number of fine sampling points for the exposure interpolation Example for 30 minute cadence: t_exp_n_int_[inst].10                                                                                           | None    |

### Stellar spots per object and instrument

| host_N_spots_[inst]        | Number of starspots on the host to include in the model; this will unlock the respective rows in the parameters file 0                                                                                          | 0       |
| [comp]_N_spots_[inst]      | Number of starspots on the companion to include in the model; this will unlock the respective rows in the parameters file 0                                                                               | 0       |

### Stellar flares

| N_flares                   | Number of stellar flares to include in the model; this will unlock the respective rows in the parameters file                                                                                             | 0       |

### Transit-timing variations

| fit_ttvs                   | Address transit/eclipse timing variations by freely fitting each transit/eclipse midtime; this will unlock the respective rows in the parameters file                                                                 | False   |

### Stellar grid per object and instrument

| host_grid_[inst]           | How finely to integrate over the surface of the host star (see Section 4.10; very_sparse/sparse/default/fine/very_fine)                                                                                       | default |
| [comp]_grid_[inst]         | How finely to integrate over the surface of the companion (see Section 4.10; very_sparse/sparse/default/fine/very_fine)                                                                                     | default |

### Stellar shape per object and instrument

| host_shape_[inst]          | How to compute the shape of the host star (see Section 4.10; sphere/roche/roche Vy/poly1p5/poly3p0/love)                                                                                                      | default |
| [comp]_shape_[inst]        | How to compute the shape of the companion (see Section 4.10; sphere/roche/roche Vy/poly1p5/poly3p0/love)                                                                                                     | default |

### Flux-weighted RVs per object and instrument

| host_flux_weighted_[inst]  | Compute the flux-weighted RV over the host’s entire surface (for the Rossiter–McLaughlin effect) or the RV of their center of mass (True/False)                                                              | False   |
| [comp]_flux_weighted_[inst]| Compute the flux-weighted RV over the companion’s entire surface (for the Rossiter–McLaughlin effect) or the RV of their center of mass (True/False)                                                     | False   |

**Note.** [comp]; placeholder for the actual name given to the companion [inst]; placeholder for the actual name given to the instrument [key]; placeholder for either flux or rv.
Figure A1. Posteriors for the global fit to the two-planet system Pi Mensae, using all available data from TESS Year 1 (Sectors 1, 4, 8, 11, 12, and 13) as well as all RV data used in Huang et al. (2018). Red lines are the published median values from Huang et al. (2018), which used TESS Sector 1 data. The `allesfitter` posteriors agree well with the published values.
Figure A2. Posteriors for the global fit with free TTVs to the two-planet system TOI-216, using all available data from TESS Year 1 (Sectors 1–9 and 11–13). Red lines are the published median values from Dawson et al. (2019), which used TESS Sector 1–6 data. The allefitter posteriors agree well with the published values.
Figure A3. Posteriors for the global fit to the TESS optical phase curve of WASP-18 using \texttt{allesfitter}'s \texttt{sine_series} model. Red lines are the published values from Shporer et al. (2019). The \texttt{allesfitter} posteriors agree well with the published values overall. The deviations for the radius ratio and amplitude of the atmospheric modulation are likely due to our simplified example, which is run on a phase-folded and binned light curve with fixed parameters and a constant baseline.
Figure A4. Posteriors for the global fit of the spotted binary star system KOI-1003, using the first 28 days of the Kepler Quarter 2 long-cadence light curve. Red lines are the published values from Roettenbacher et al. (2016). The allefitter posteriors agree well with the published values. Spot crossings likely cause the deviation for the radius ratio in this section of the light curve, as also discussed by Roettenbacher et al. (2016).
Figure A5. Posteriors for the global fit of GJ 1243, using a model of two spots and three flares for a 1.8 day section of the TESS short-cadence light curve.
### Table A2

A List of All Possible Parameters for *allesfitter*, Which can be Given in the *params.csv* File

| Parameter                  | Explanation                                                                 | Default |
|----------------------------|-----------------------------------------------------------------------------|---------|
| [comp]_rr                  | The radius ratio of the companion to host, $R_{\text{comp}}/R_*$.            | 0       |
| [comp]_rsrma               | The sum of the stellar and companion radii divided by the semimajor axis, $(R_{\text{comp}} + R_*)/a$. | 0       |
| [comp]_cosi                | The cosine of the orbit of this companion, $\cos i$.                         | 0       |
| [comp]_epoch               | The epoch/transit midtime in days, $T_0$.                                    | 0       |
| [comp]_period              | The orbital period of the companion in days, $P$.                           | 0       |
| [comp]_K                   | The host’s RV semiamplitude caused by the companion in km s$^{-1}$, $K$.     | 0       |
| [comp]_f_c                 | Transformation of eccentricity and argument of periastron, $\sqrt[e]{e} \cos \omega$. | 0       |
| [comp]_f_s                 | Transformation of eccentricity and argument of periastron, $\sqrt[e]{e} \sin \omega$. | 0       |
| [comp]_sbratio_[inst]      | Surface brightness ratio between the companion and host star, $J$.            | 0       |
| dil_[inst]                | Dilution of the signal in the given instruments bandpass, $D_{\text{dil}} = 1 - (F_{\text{source}}/(F_{\text{source}} + F_{\text{blend}}))$. | 0       |

#### Limb-darkening coefficients—physical (see Section 4.4)

| host ldc u1_[inst]        | Physical coefficient $u_1$ for host (for all laws).                          | None    |
| host ldc u2_[inst]        | Physical coeff. $u_2$ for host (for laws with $\geq 2$ params.)            | None    |
| host ldc u3_[inst]        | Physical coeff. $u_3$ for host (for laws with $\geq 3$ params.)            | None    |
| host ldc u4_[inst]        | Physical coeff. $u_4$ for host (for laws with $\geq 4$ params.)            | None    |
| [comp]_ldc u1_[inst]      | Physical coeff. $u_1$ for companion (for all laws).                         | None    |
| [comp]_ldc u2_[inst]      | Physical coeff. $u_2$ for companion (for laws with $\geq 2$ params.)       | None    |
| [comp]_ldc u3_[inst]      | Physical coeff. $u_3$ for companion (for laws with $\geq 3$ params.)       | None    |
| [comp]_ldc u4_[inst]      | Physical coeff. $u_4$ for companion (for laws with $\geq 4$ params.)       | None    |

#### Limb-darkening coefficients—transformed (see Section 4.4)

| host ldc q1_[inst]        | Transformed coefficient $q_1$ for host (for all laws).                      | None    |
| host ldc q2_[inst]        | Transformed coeff. $q_2$ for host (for laws with $\geq 2$ params.)         | None    |
| host ldc q3_[inst]        | Transformed coeff. $q_3$ for host (for laws with $\geq 3$ params.)         | None    |
| host ldc q4_[inst]        | Transformed coeff. $q_4$ for host (for laws with $\geq 4$ params.)         | None    |
| [comp]_ldc q1_[inst]      | Transformed coeff. $q_1$ for companion (for all laws).                      | None    |
| [comp]_ldc q2_[inst]      | Transformed coeff. $q_2$ for companion (for laws with $\geq 2$ params.)    | None    |
| [comp]_ldc q3_[inst]      | Transformed coeff. $q_3$ for companion (for laws with $\geq 3$ params.)    | None    |
| [comp]_ldc q4_[inst]      | Transformed coeff. $q_4$ for companion (for laws with $\geq 4$ params.)    | None    |

#### Errors (white noise; see Section 4.5)

| ln_err_flux_[inst]        | Natural logarithm of the error scaling for photometry, gets multiplied with the weights for the user-given errors | 0       |
| ln_jitter_rv_[inst]       | Natural logarithm of the jitter term for the RV, gets added in quadrature to the user-given errors | 0       |

#### Baselines (red noise)—constant offset (see Section 4.6)

| baseline_offset_[key]_[inst] | Constant offset | 0       |

#### Baselines (red noise)—linear trend (see Section 4.6)

| baseline_offset_[key]_[inst] | Constant offset | 0       |
| baseline_slope_[key]_[inst]  | Linear slope    | 0       |

#### Baselines (red noise)—GP with real kernel (see Section 4.6)

| baseline_gp_offset_[key]_[inst] | Constant offset (optional; default 1 for flux, 0 for RV) | 0       |
| baseline_gp_real_lna_[key]_[inst] | Natural logarithm of $a$ | 0       |
| baseline_gp_real_lnc_[key]_[inst] | Natural logarithm of $c$ | 0       |

#### Baselines (red noise)—GP with complex kernel (see Section 4.6)

| baseline_gp_offset_[key]_[inst] | Constant offset (optional; default 1 for flux, 0 for RV) | 0       |
| baseline_gp_complex_lna_[key]_[inst] | Natural logarithm of $a$ | 0       |
| baseline_gp_complex_lnb_[key]_[inst] | Natural logarithm of $b$ | 0       |
| baseline_gp_complex_lnc_[key]_[inst] | Natural logarithm of $c$ | 0       |
| baseline_gp_complex_lnd_[key]_[inst] | Natural logarithm of $d$ | 0       |
| Parameter                                | Explanation                                                                 | Default |
|------------------------------------------|-----------------------------------------------------------------------------|---------|
| **Baselines (red noise)—GP with Matérn 3/2 kernel (see Section 4.6)**                     |                                               |         |
| baseline_gp_offset_[key]_[inst]          | Constant offset (optional; default 1 for flux, 0 for RV)                    | 0       |
| baseline_gp_matern32_Insigna_[key]_[inst]| Natural logarithm of \( \sigma \)                                           | 0       |
| baseline_gp_matern32_Inrh0_[key]_[inst]  | Natural logarithm of \( \rho \)                                            | 0       |
| **Baselines (red noise)—GP with SHO kernel (see Section 4.6)**                           |                                               |         |
| baseline_gp_offset_[key]_[inst]          | Constant offset (optional; default 1 for flux, 0 for RV)                    | 0       |
| baseline_gp_sho_lnS0_[key]_[inst]        | Natural logarithm of \( S_0 \)                                             | 0       |
| baseline_gp_sho_lnQ_[key]_[inst]         | Natural logarithm of \( Q \)                                               | 0       |
| baseline_gp_sho_lnomega0_[key]_[inst]    | Natural logarithm of \( \omega_0 \)                                        | 0       |
| **Stellar variability—linear trend (see Section 4.7)**                                  |                                               |         |
| stellar_var_gp_offset_[key]_[inst]       | Constant offset (optional; default 1 for flux, 0 for RV)                    | 0       |
| stellar_var_offset_[key]_[inst]          | Constant offset                                                             | 0       |
| stellar_var_slope_[key]_[inst]           | Linear slope                                                                | 0       |
| **Stellar variability—GP with real kernel (see Section 4.7)**                           |                                               |         |
| stellar_var_gp_offset_[key]_[inst]       | Constant offset (optional; default 1 for flux, 0 for RV)                    | 0       |
| stellar_var_gp_real_lna_[key]_[inst]     | Natural logarithm of \( a \)                                               | 0       |
| stellar_var_gp_real_lnc_[key]_[inst]     | Natural logarithm of \( c \)                                               | 0       |
| **Stellar variability—GP with complex kernel (see Section 4.7)**                         |                                               |         |
| stellar_var_gp_offset_[key]_[inst]       | Constant offset (optional; default 1 for flux, 0 for RV)                    | 0       |
| stellar_var_gp_complex_lna_[key]_[inst]  | Natural logarithm of \( a \)                                               | 0       |
| stellar_var_gp_complex_lnb_[key]_[inst]  | Natural logarithm of \( b \)                                               | 0       |
| stellar_var_gp_complex_lnc_[key]_[inst]  | Natural logarithm of \( c \)                                               | 0       |
| stellar_var_gp_complex_lnd_[key]_[inst]  | Natural logarithm of \( d \)                                               | 0       |
| **Stellar variability—GP with Matérn 3/2 kernel (see Section 4.7)**                     |                                               |         |
| stellar_var_gp_offset_[key]_[inst]       | Constant offset (optional; default 1 for flux, 0 for RV)                    | 0       |
| stellar_var_gp_matern32_Insigna_[key]_[inst]| Natural logarithm of \( \sigma \)                                           | 0       |
| stellar_var_gp_matern32_Inrh0_[key]_[inst]| Natural logarithm of \( \rho \)                                            | 0       |
| **Stellar variability—GP with SHO kernel (see Section 4.7)**                             |                                               |         |
| stellar_var_gp_sho_lnS0_[key]_[inst]     | Natural logarithm of \( S_0 \)                                             | 0       |
| stellar_var_gp_sho_lnQ_[key]_[inst]      | Natural logarithm of \( Q \)                                               | 0       |
| stellar_var_gp_sho_lnomega0_[key]_[inst] | Natural logarithm of \( \omega_0 \)                                        | 0       |
| **Phase-curve parameters—sine_series model (see Section 4.9.1)**                      |                                               |         |
| [comp]_phase_curve_A1_[inst]             | Semiamplitude of the sine term \( A_0 \sin \Phi(t) \) approximating the Doppler-boosting (beamng) modulation (in parts per thousand, ppt) | None    |
| [comp]_phase_curve_B1_[inst]             | Semiamplitude of the cosine term \( B_0 \cos \Phi(t) \) approximating the atmospheric (thermal and reflected light) mod. (in ppt) | None    |
| [comp]_phase_curve_B1_shift_[inst]       | Time shift \( s \) of the cosine term \( B_0 \cos \Phi(t + s) \) (in days) | 0       |
| [comp]_phase_curve_B2_[inst]             | Semiamplitude of the cosine term \( B_0 \cos 2\Phi(t) \) approximating the leading-order ellipsoidal (tidal distortion) mod. (in ppt) | None    |
| [comp]_phase_curve_B3_[inst]             | Semiamplitude of the cosine term \( B_0 \cos 3\Phi(t) \) approximating the next-order ellipsoidal (tidal distortion) mod. (in ppt); this is usually negligible for exoplanets but can become measurable for binary stars | None    |
| [comp]_phase_curve_B1t_[inst]            | Semiamplitude of the cosine term \( B_t \cos \Phi(t) \) approximating the thermal emission part of the atmospheric mod. (in ppt) | None    |
| [comp]_phase_curve_B1t_shift_[inst]      | Time shift \( s \) of the cosine term \( B_t \cos \Phi(t + s) \) (in days) | 0       |

To differentiate thermal emission and reflected light, one can use:
### Table A2 (Continued)

| Parameter | Explanation | Default |
|-----------|-------------|---------|
| [comp]_phase_curve_B1r_shift_[inst] | Time shift \( s \) of the cosine term \( B_1 \cos(\phi(t + s)) \) (in days) | 0 |

Phase-curve parameters—sine_physical model (see Section 4.9.2)

#### Standard set:

| Parameter | Explanation | Default |
|-----------|-------------|---------|
| [comp]_phase_curve_beaming_[inst] | Positive semi-amplitude of the beaming effect, representing the term \( A_1 \sin(\phi(t)) \), i.e., a mod. around the median flux level of the star (in ppt) | None |
| [comp]_phase_curve_atmospheric_[inst] | Positive full (peak-to-peak) amplitude of the atmospheric contribution, representing the term \( -2B_1(1 - \cos(\phi(t))) \), i.e., an additive component to the companion’s night-side flux (in ppt) | None |
| [comp]_phase_curve_atmospheric_shift_[inst] | Time shift of the atmospheric contribution term (in days) | None |
| [comp]_phase_curve_ellipsoidal_[inst] | Positive full (peak-to-peak) amplitude of the leading-order term of the ellipsoidal mod., representing the term \( -2B_1(1 - \cos(2\phi(t))) \), i.e., an additive component to the system’s flux from spherical (nondistorted) bodies (in ppt) | None |
| [comp]_phase_curve_ellipsoidal_2nd_[inst] | Positive full (peak-to-peak) amplitude of the next-order term of the ellipsoidal mod., representing the term \( -2B_1(1 - \cos(3\phi(t))) \), i.e., an additive component to the system’s flux from spherical (nondistorted) bodies (in ppt) | None |

To differentiate thermal emission and reflected light, one can use:

| Parameter | Explanation | Default |
|-----------|-------------|---------|
| [comp]_phase_curve_atmospheric_thermal_[inst] | Positive full (peak-to-peak) amplitude of the atmospheric thermal emission (in ppt) | None |
| [comp]_phase_curve_atmospheric_thermal_shift_[inst] | Time shift of the atmospheric thermal emission (in days) | 0 |
| [comp]_phase_curve_atmospheric_reflected_[inst] | Positive full (peak-to-peak) amplitude of the atmospheric reflected light (in ppt) | None |
| [comp]_phase_curve_atmospheric_reflected_shift_[inst] | Time shift of the atmospheric reflected light (in days) | 0 |

#### Star spots \( (i = 1, 2, 3, \ldots, N_{\text{spots}}) \)

| Parameter | Explanation | Default |
|-----------|-------------|---------|
| host_spot_[i]_long_[inst] | Longitude of the star spot number \( i \) on the host (in degrees from 0° to 360°) | 0 |
| host_spot_[i]_lat_[inst] | Latitude of the star spot number \( i \) on the host (in degrees from −90° to 90°) | 0 |
| host_spot_[i]_size_[inst] | The angular radius of the star spot number \( i \) on the host (in degrees) | 0 |
| host_spot_[i]_brightness_[inst] | The brightness ratio between the star spot number \( i \) and the surface of the host | 0 |
| [comp]_spot_[i]_long_[inst] | Longitude of the star spot number \( i \) on the companion (in degrees from 0° to 360°) | 0 |
| [comp]_spot_[i]_lat_[inst] | Latitude of the star spot number \( i \) on the companion (in degrees from −90° to 90°) | 0 |
| [comp]_spot_[i]_size_[inst] | The angular radius of the star spot number \( i \) on the companion (in degrees) | 0 |
| [comp]_spot_[i]_brightness_[inst] | The brightness ratio between the star spot number \( i \) and the surface of the companion | 0 |

#### Stellar flares \( (i = 1, 2, 3, \ldots, N_{\text{flares}}) \)

| Parameter | Explanation | Default |
|-----------|-------------|---------|
| flare_tpeak_[i] | Peak time of flare number \( i \) | 0 |
| flare_fwhm_[i] | FWHM of flare number \( i \) | 0 |
| flare_ampl_[i] | Amplitude of flare number \( i \) | 0 |

#### Advanced parameters (for proficient users of ellc)

| Parameter | Explanation | Default |
|-----------|-------------|---------|
| [comp]_q | Mass ratio between the companion and host | 1 |
| host_gdc | Gravity-darkening coefficient for the host | None |
| [comp]_gdc | Gravity-darkening coefficient for the companion | None |
| host_atmo_[inst] | Coefficient of a simplified reflection and emission model on the host’s side facing the companion | None |
| [comp]_atmo_[inst] | Coefficient of a simplified reflection and emission model on the companion’s side facing the host | None |
| host_bfac_[inst] | Doppler-boosting factor of the host | None |
| [comp]_bfac_[inst] | Doppler-boosting factor of the companion | None |
| didt_[inst] | Rate of change of inclination (in degrees per anomalous period) | None |
| domdt_[inst] | Rate of apsidal motion (in degrees per anomalous period) | None |
| host_rotfac | Asynchronous rotation factor for the host | None |
| [comp]_rotfac | Asynchronous rotation factor for the companion | None |
| host_hf_[inst] | Fluid second Love number for radial displacement, for the host; only used if host_shape_[inst] is love | 1.5 |
| [comp]_hf_[inst] | Fluid second Love number for radial displacement, for the companion; only used if [comp]_shape_[inst] is love | 1.5 |
| host_lambda | Sky-projected angle between orbital and rotation axes for the host (in degrees) | None |
| [comp]_lambda | Sky-projected angle between orbital and rotation axes for the companion (in degrees) | None |
Table A2
(Continued)

| Parameter          | Explanation                                                                 | Default       |
|--------------------|-----------------------------------------------------------------------------|---------------|
| host_vsinic        | Rotational $\nu \sin i$ for calculation of the Rossiter–McLaughlin effect for the host (in km s$^{-1}$) | None          |
| [comp]_vsinic      | Rotational $\nu \sin i$ for calculation of the Rossiter–McLaughlin effect for the companion (in km s$^{-1}$) | None          |

Notes. This list reflects allesfitter version 1.2. For future additions and the most up-to-date documentation, see www.allesfitter.com. Note that not all of these can be selected at the same time, as some combinations depend on which models are chosen (for example, either a linear or a quadratic limb-darkening model). [comp]: placeholder for the actual name given to the companion. [inst]: placeholder for the actual name given to the instrument. [key]: placeholder for either flux or $rv$.

Table A3
A List of All Values that Will be Derived from the allesfitter Posterior

| Derived parameter | Equation |
|-------------------|----------|
| Host radius over semimajor axis; $R_{\text{host}}/a_{\text{comp}}$ | $r/(1 + k)$ |
| Semimajor axis over host radius; $a_{\text{comp}}/R_{\text{host}}$ | $(1 + k)/r$ |
| Companion radius over semimajor axis; $R_{\text{comp}}/a_{\text{comp}}$ | $r \cdot k/(1 + k)$ |
| Companion radius; $R_{\text{comp}}$ | $R_0 \cdot k$ |
| Companion radius; $R_{\text{comp}}$ | $R_0 \cdot k$ |
| Semimajor axis; $a_{\text{comp}}$ | $R_0 / R_0/a$ |
| Semimajor axis; $a_{\text{comp}}$ | $R_0 / R_0/a$ |
| Inclination; $i_{\text{comp}}$ (deg) | $\arccos(\cos i)$ |
| Eccentricity; $e_{\text{comp}}$ | $\sqrt{f_x^2 + f_y^2}$ |
| Argument of periastron; $\omega_{\text{comp}}$ (deg) | $\arctan(2(f_x, f_y) \% 360^\circ$ |
| Mass ratio; $q_{\text{comp}}$ | $a_1/a_2$ with $a_i = R_0 \cdot P_i \cdot \sqrt{(1-e_i^2)/\sin^2 \omega}$ |
| Companion mass; $M_{\text{comp}}$ | $q \cdot M_0$ |
| Companion mass; $M_{\text{comp}}$ | $q \cdot M_0$ |
| Impact parameter; $b_{\text{imp,comp}}$ | $a \cos i / \sqrt{1 + e \sin \omega}$ |
| Total transit duration (I to IV); $T_{\text{tot,comp}}$ (h) | $P \sin^{-1} \left( \frac{R_0}{a} \sqrt{\frac{(1+k)^2 - p^2}{\sin^2 \omega}} \right)$ |
| Full-transit duration (II to III); $T_{\text{full,comp}}$ (h) | $P \sin^{-1} \left( \frac{R_0}{a} \sqrt{\frac{(1+k)^2 - k^2}{\sin^2 \omega}} \right)$ |
| Epoch of occultation; $T_{\text{occ,comp}}$ (days) | $\approx T_0 + \frac{a}{R_0} \left( 1 + \frac{e \cos \omega}{\sqrt{1 - e^2}} \right)$ |
| Impact parameter of occultation; $b_{\text{occ,comp}}$ | $\approx \frac{3P}{4\pi} \left( \frac{a}{R_0} \right)^{2/3}$ if $k^2 < 0.01$ |
| Host density from orbit; $\rho_{\text{host,comp}}$ (g/cm$^3$) | $2\pi P^2 (R_{\text{comp}}/a_{\text{comp}})^2 \sin^2 \omega = \frac{8R_0}{2\pi}$ |
| Companion surface gravity from orbit; $g_{\text{comp}}$ (g/cm$^3$) | $\sqrt{(1-A)^{1/4} \frac{R_0}{2\pi}}$ |
| Equilibrium temperature; $T_{\text{eq,comp}}$ (K) | $\approx \frac{3P^2}{4\pi} \left( \frac{a}{R_0} \right)^{2/3}$ |

Notes. Note that not all of these can be derived every time. This list reflects allesfitter version 1.1.1. For future additions and the most up-to-date documentation, see www.allesfitter.com. [comp]: placeholder for the actual name given to the companion. [inst]: placeholder for the actual name given to the instrument. [key]: placeholder for either flux or $rv$. For readability, we define $k = [\text{comp}]_n$, $r = [\text{comp}]_\text{r_rama}$, $T_0 = [\text{comp}]_\text{epoch}$, $P = [\text{comp}]_\text{epoch}$, $\cos \iota = [\text{comp}]_\text{cos}$, $f_x = [\text{comp}]_fx$, $f_y = [\text{comp}]_fy$, and $D_0 = \text{dim}_\text{inst}$. Additionally, the [comp] suffixes in the equations were omitted (aside from $R_{\text{comp}}$) and the host suffixes were replaced with h. For explanation of these parameters, see, e.g., Winn (2011) and references therein.

ORCID iDs
Maximilian N. Günther @ https://orcid.org/0000-0002-3164-9086
Tansu Daylan @ https://orcid.org/0000-0002-6939-9211

References
Anderson, T. W., & Darling, D. A. 1952, Ann. Math. Stat., 23, 193
Badenas-Agustí, M., Günther, M. N., Daylan, T., et al. 2020, AJ, 160, 113
Bayliss, D., Gillen, E., Eiglmüller, P., et al. 2018, MNRAS, 475, 4467
