Stable estimates of autocorrelation length from short GSP-Phot Aeneas MCMC chains

prepared by: R. Andrae
reference: GAIA-C8-TN-MPIA-RAN-033
issue: 1
revision: 5
date: 2015-12-08
status: Issued

Abstract

The GSP-Phot core algorithm Aeneas employs an MCMC to estimate parameters. Apart from the MCMC acceptance rate, another important quality control of MCMC chains is their autocorrelation length. We investigate several different estimators for MCMC autocorrelation length, in order to find one which provides the most stable results given the relatively short MCMC chains produced by Aeneas. We find that we can estimate τ reliably from the short Aeneas MCMC chain by modelling the MCMC chain itself as an Ornstein-Uhlenbeck process. Early test results suggest that $8 < \tau < 25$ may be a good quality control, ruling out most cases of multimodality and failed convergence.

Document History

Issue	Revision	Date	Author	Comment
1	5	2015-12-08	RAN	First issue.
D	5	2015-12-08	RAN	Including comments from Coryn.
D	4	2015-11-17	RAN	Including comments from Morgan.
D	3	2015-11-12	RAN	Including Appendix A.
D	2	2015-11-12	RAN	Clarification exponential vs. integrated autocorrelation length.
D	1	2015-11-11	RAN	Extended draft, including more test results.
D	0	2015-11-09	RAN	First draft.

1 Introduction

In this TN, we discuss the estimation and interpretation of autocorrelation length from a given MCMC chain. We are interested in the autocorrelation length in order to obtain a parameter for testing the quality of MCMC chains produced by GSP-Phot's core algorithm Aeneas. This should enable us to easily and automatically identify cases where the MCMC chain has not converged or is multimodal, thus enabling us to flag potentially problematic results. One particular problem we are facing is that Aeneas' MCMC chains are unusually short, due to technical limitations on computational cost. Typically, Aeneas' MCMC chains have only 100-140 iterations. This will cause substantial problems for standard methods estimating the autocorrelation length.

Before we embark on different methods for estimating the autocorrelation length of an MCMC chain, we first want to briefly discuss the basics of autocorrelation lengths and what we can learn from them about MCMC chains. This discussion will largely follow Sokal (1997).

By definition, an MCMC ("Markov chain Monte Carlo") is a Markov chain. That means, if $X = \{x_1, x_2, \ldots, x_N\}$ is an MCMC chain of N iterations, then the joint probability $P(X)$ factorises as

$$P(X) = P(x_1, x_2, \ldots, x_N) \tag{1}$$

$$= P(x_N|x_1, \ldots, x_{N-1})P(x_{N-1}|x_1, \ldots, x_{N-2}) \ldots P(x_3|x_1, x_2)P(x_2|x_1)P(x_1) \tag{2}$$

$$= P(x_N|x_{N-1})P(x_{N-1}|x_{N-2}) \ldots P(x_3|x_2)P(x_2|x_1)P(x_1)\,. \tag{3}$$

In simple words, any points x_n of a Markov chain (e.g. an MCMC) only depends on its direct predecessor, x_{n-1}, but not the entire history of the chain. Such a stochastic process is called

an AR(1) process.[1] Moreover, if the MCMC chain has converged, then it is also stationary, meaning that it does oscillate around the best-fit values but does not run off to infinity anymore.

For any Markov chain $X = \{x_1, x_2, \ldots, x_N\}$, such as an MCMC, we can define its autocovariance function,

$$C(t) = \langle (x_n - \mu_X)(x_{n+t} - \mu_X) \rangle \,, \tag{4}$$

where μ_X is the *true* mean of X (not the estimated mean). Obviously, $C(0)$ is the *true* variance of X. From this autocovariance function, we can obtain the normalised autocorrelation function,

$$\rho(t) = \frac{C(t)}{C(0)} \,. \tag{5}$$

In practice, this autocorrelation function can usually be approximated by an exponential (Sokal, 1997),

$$\rho(t) = \frac{C(t)}{C(0)} \propto e^{-|t|/\tau_{\rm exp}} \,, \tag{6}$$

where $\tau_{\rm exp}$ is the *exponential* autocorrelation length. This naming convention is to distinguish $\tau_{\rm exp}$ from the *integrated* autocorrelation length, which is defined as

$$\tau_{\rm int} = \frac{1}{2} + \sum_{t=1}^{\infty} \rho(t) = \frac{1}{2} \sum_{t=-\infty}^{\infty} \rho(t) \,. \tag{7}$$

In general, $\tau_{\rm exp}$ and $\tau_{\rm int}$ can have values of completely different orders of magnitudes. These two different concepts of autocorrelation lengths have two different meanings (Sokal, 1997):

- The exponential autocorrelation length, $\tau_{\rm exp}$, tells us how many iterations the system requires to attain equilibrium after it has been disturbed. Such a disturbance could be the initial guess or a clipping applied to the MCMC chain. This is generally called "burn-in" or "convergence" and Sokal (1997) recommends to discard $20\tau_{\rm exp}$ samples after any such disturbance.
- The integrated autocorrelation length, $\tau_{\rm int}$, controls the "redundancy" of the MCMC chain when it is in equilibrium. As we show in Appendix A, the autocorrelations in the MCMC chain do not compromise mean estimates but concerning parameter uncertainties, the MCMC sample variance is too large by a factor of $\approx 2\tau_{\rm int}$. The reason is that the sample variance estimate divides by N instead of the number of "effectively independent" samples, which is only $\frac{N}{2\tau_{\rm int}} < N$. That is important to know when estimating the parameter uncertainties from the MCMC chain or to define the thin-out factor of the MCMC chain.

[1] AR(1) means that the stochastic process is *a*uto*r*egressive of first order, i.e., x_n only depends on x_{n-1}.

We can now see that the autocorrelation length is a very important quantity in order to draw reliable conclusions from an MCMC chain. We need to have an idea of τ to set an a priori length of the MCMC chain before running it, to know how many samples we need to discard for burn-in, and to know how much to thin out the MCMC chain before estimating parameter uncertainties.

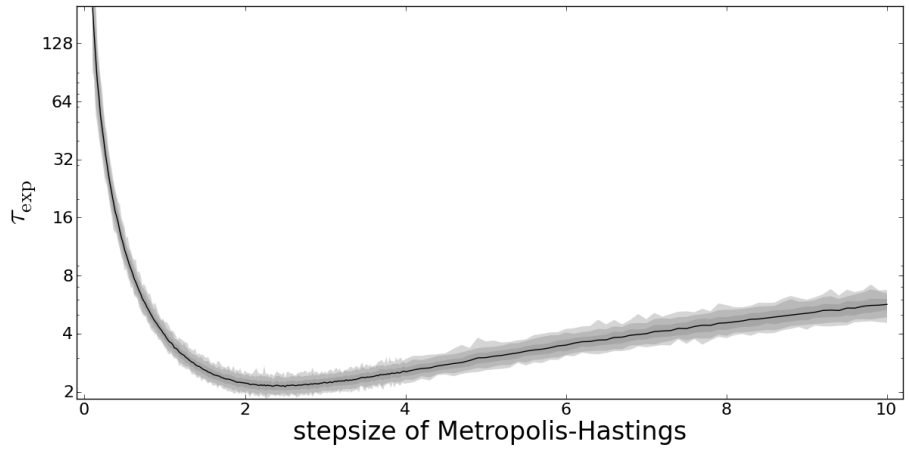

Figure 1: Autocorrelation length τ vs. stepsize of proposal distribution for a Metropolis-Hastings algorithm. The underlying posterior was designed as $\log P(x) = -x^2/2$ for the sake of simplicity. At each stepsize, 200 MCMC chains were simulated, each having 5 000 iterations.

Let us further discuss a toy example in order to get a more intuitive feeling for the autocorrelation length of an MCMC. Let us consider a Metropolis-Hastings algorithm for some variable x whose posterior probability we simply set as $\log P(x) = -x^2/2$, i.e., a unit Gaussian. For a Metropolis-Hastings algorithm, the choice of the stepsize of its proposal distribution is very delicate. If we choose the stepsize too small, the MCMC can only make tiny steps and takes ages to go anywhere. In such a case, we expect a large autocorrelation length. If this was the only effect at work, then starting from a very small stepsize and increasing it should therefore decrease the autocorrelation length. However, if the stepsize becomes too large, the MCMC can overshoot and thus run away from the posterior maximum, or take ages to return to it. In this case, we expect the autocorrelation length to eventually start to increase again, if the stepsize becomes too large. Figure 1 shows simulation results that indeed confirm this behaviour. There is a sweet spot, an optimal choice of the stepsize, which minimises the autocorrelation length. Choosing that stepsize will also minimise our computational cost because it minimises the number of samples we need to discard in burn-in and that we need to thin out before estimating parameters.[2]

[2]This "optimal" stepsize may minimise the autocorrelation length and thus the computational effort, but it is not obvious how good the parameter estimation will be.

2 The `acor` method

In their `emcee` paper, Foreman-Mackey et al. (2013) recommend using the `acor` method[3] to estimate $\tau_{\rm int}$. Given a time series $X = \{x_n\}_{n=1}^N$, e.g., a univariate MCMC chain of length N, the autocovariance function $C(t)$ is estimated by

$$\hat{C}(t) = \frac{1}{N-t} \sum_{n=1}^{N-t} (x_{m+t} - \hat{\mu})\,(x_m - \hat{\mu}) \;, \tag{8}$$

where $\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n$ is the *estimated* mean of X (*not* the true mean). The latter makes the difference to the definition of the theoretical autocovariance function in Eq. (4), where we operate with true instead of estimated values. Furthermore, the chain is no of finite length N. Both effects will be a huge source of numerical noise in practice. Obviously, $\hat{C}(0)$ will just be the total *estimated* variance of the time series X. The integrated autocorrelation length, τ, is then estimated similarly to Eq. (7)

$$\tau_{\rm int} = 1 + 2 \sum_{t=1}^{N} \frac{\hat{C}(t)}{\hat{C}(0)} \,. \tag{9}$$

While Eq. (9) is reasonably straightforward to implement, it is numerically highly unstable. Even if $\tau_{\rm int}$ itself may have a small value (say $\tau_{\rm int} = 10$) we may still need many more MCMC samples (say millions) to get a reliable estimate of $\tau_{\rm int}$. As already mentioned, the reason for this numerical instability is the noise in the estimates of $\hat{\mu}$ and $\hat{C}(t)$, which propagate nonlinearly into Eq. (9).

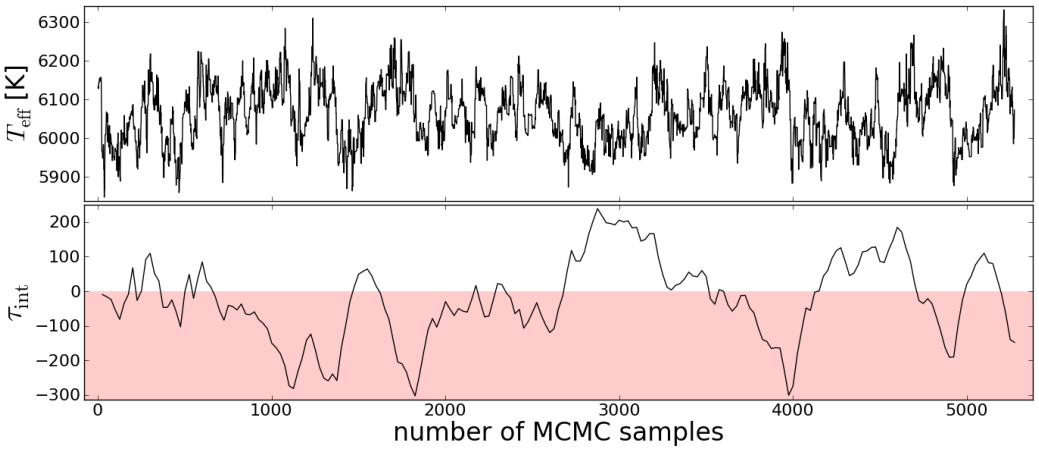

Figure 2: Numerical instability of Eq. (9). Top panel: Univariate MCMC chain as $T_{\rm eff}$ vs. iteration number. Bottom panel: Estimate of τ from Eq. (9) vs. number of MCMC samples used. The red area marks $\tau_{\rm int} < 0$.

[3] `http://www.math.nyu.edu/faculty/goodman/software/acor`

This numerical instability is demonstrated in Fig. 2, using an actual MCMC chain from GSP-Phot Aeneas. While we can see from the top panel that the MCMC has clearly converged and thus exhibits no trends, the bottom panel shows us that the estimate of $\tau_{\rm int}$ remains unstable even if we use thousands of MCMC samples. The value given by Eq. (9) varies between -300 and 200, without any evidence of convergence. More importantly, while $\tau_{\rm int} < 0$ is in principle possible for a time series with strong anticorrelations, in practice it usually indicates numerical instability due to a lack of MCMC samples. This is even more dramatic if we consider the limitation on computational cost imposed on GSP-Phot by DPCC, which implies that we cannot run the MCMC for more than 200-300 iterations.

3 Autocorrelation length of an OU process

Figure 2 clearly demonstrates that the `acor` method is not stable enough to provide useful estimates of MCMC autocorrelation length for Aeneas. Instead, we now want to investigate another method for estimating τ, which should be more stable (but possibly more biased) because it makes the additional (possibly incorrect) assumption that the MCMC chain, $X = \{x_n\}_{n=1}^N$, is a stationary Gaussian AR(1) process. In simple words, we assume that the MCMC chain itself originates from some kind of model, such that we can obtain a more robust estimate of the autocorrelation length by first fitting the model and then deriving τ from the model's properties.

3.1 Definition of a stationary Gaussian AR(1) process

A Gaussian AR(1) process, $X = \{x_n\}_{n=1}^N$, is defined by the recursive relation

$$x_{n+1} = \phi x_n + z_{n+1}\,, \tag{10}$$

where $Z = \{z_n\}_{n=1}^N$ is Gaussian white noise with mean μ and variance σ^2 that drives the process. In general, the constant ϕ can have any real value. For instance, if $\phi = 1$, this is a random walk. However, if the constant ϕ satisfies $|\phi| < 1$, the Gaussian AR(1) process is called stationary. In this case, the time series will converge to the mean value $\frac{\mu}{1-\phi}$ with variance $\frac{\sigma^2}{1-\phi^2}$. Moreover, if the process is not only stationary ($|\phi| < 1$) but also $0 < \phi < 1$ holds true, then this is called an Ornstein-Uhlenbeck (OU) process (Uhlenbeck & Ornstein, 1930; Gillespie, 1996).

In our particular case, we want to use the OU process to model an MCMC chain and to infer the autocorrelation length. For that, we already assume that the MCMC chain has converged and any MCMC samples from the burn-in phase have been discarded. This can be accommodated into the AR(1) model by setting $\mu = 0$, such that there are no trends anymore. Note that the formal definition of a converged MCMC chain is that of a stationary Markov process, which is a stationary AR(1) process. There are only two additional assumptions here, namely Gaussianity and $0 < \phi < 1$.[4] Any or both of these additional assumptions can be violated in practice.

[4]For $\phi = 0$, the MCMC chain would be completely uncorrelated, i.e., x_n would be independent of x_{-1}. Any $\phi > 0$ introduces autocorrelations.

For instances, "banana-shaped" MCMC contours that can appear in the $T_{\rm eff}$-A_0 parameter space would be a definite signature of non-Gaussianity. Likewise, the MCMC chain could exhibit anticorrelations, which would violate $0 < \phi < 1$ (see below).

3.2 Autocovariance function of an OU process

Let the Gaussian AR(1) process be stationary, i.e., $|\phi| < 1$. We then obtain for the first element of the autocovariance function that

$$C(0) = \langle (x_n - \langle x_n \rangle)^2 \rangle = \langle x_n^2 \rangle = \frac{1}{N} \sum_{n=1}^{N} x_n^2 = \sigma^2 \,, \tag{11}$$

where we have used $\langle x_n \rangle = \mu = 0$, which reflects our working assumption that the MCMC chain has converged and that there are no trends in X. Note that we are working with the *true* mean here. This is the major advantage of coming up with an explicit model for the MCMC chain. Let us calculate the next element of the autocovariance function,

$$C(1) = \langle (x_n - \langle x_n \rangle) \, (x_{n+1} - \langle x_{n+1} \rangle) \rangle = \langle x_n x_{n+1} \rangle \,. \tag{12}$$

We now use Eq. (10) to replace x_{n+1} by $\phi x_n + z_{n+1}$ and obtain

$$C(1) = \langle x_n \, (\phi x_n + z_{n+1}) \rangle = \phi \langle x_n^2 \rangle + \langle x_n z_{n+1} \rangle = \phi \sigma^2 \,, \tag{13}$$

where we have used the previous result $\langle x_n^2 \rangle = C(0) = \sigma^2$ and $\langle x_n z_{n+1} \rangle = 0$ because Z is Gaussian white noise and does not correlate with x_n. Continuing these calculations, we would find that the autocovariance function is given by

$$C(t) = \phi^t \sigma^2 \,. \tag{14}$$

If we now further restrict the stationarity condition $|\phi| < 1$ to strictly positive values, $0 < \phi < 1$, i.e., we make this an OU process, we can rewrite the autocovariance function as an exponential,

$$C(t) = \sigma^2 e^{t \ln \phi} = \sigma^2 e^{-t/\tau_{\rm exp}} \,, \tag{15}$$

where the exponential autocorrelation length is given by

$$\tau_{\rm exp} = -\frac{1}{\ln \phi} \,. \tag{16}$$

Similarly, inserting Eq. (14) into Eq. (7), we obtain for the integrated autocorrelation length of an AR(1) process that

$$\tau_{\rm int} = \frac{1}{2} + \sum_{t=1}^{\infty} \phi^t = -\frac{1}{2} + \sum_{t=0}^{\infty} \phi^t = \frac{1}{1-\phi} - \frac{1}{2} \,. \tag{17}$$

Obviously, for an OU process, there is a clearly defined relation between τ_{int} and τ_{exp}, as is shown in Fig. 3. For $\tau_{\text{exp}} > 2$, we have $\tau_{\text{int}} \approx \tau_{\text{exp}}$, but as τ_{exp} approaches zero, τ_{int} will approach $\frac{1}{2}$. In practice, we will always have $\tau_{\text{exp}} > 2$ (or else the MCMC chain should be flagged as bad anyway) such that $\tau_{\text{int}} \approx \tau_{\text{exp}}$ makes our life easier and we only need to talk about a single autocorrelation length $\tau \approx \tau_{\text{int}} \approx \tau_{\text{exp}}$. Therefore, "all we need to do" is to find a way to efficiently fit an OU model to our given MCMC chain. This fit will provide an estimate of ϕ and thus an estimate of τ.

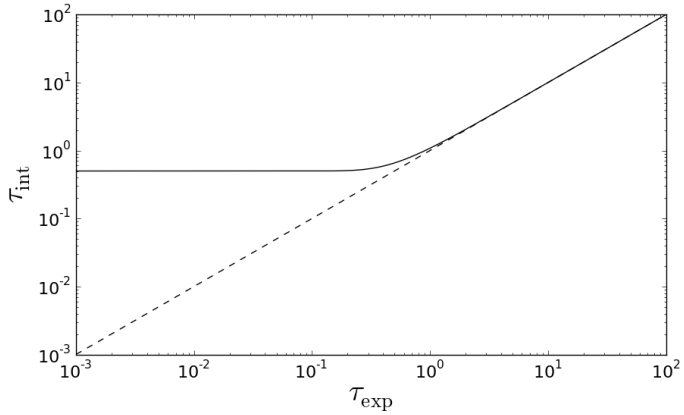

Figure 3: Mathematical relation between τ_{int} and τ_{exp} for an OU process.

3.3 Fitting the OU process without measurement errors

We now discuss how to fit an OU process to a given univariate MCMC chain, $X = \{x_n\}_{n=1}^{N}$. Let us emphasise that the MCMC chain itself has *no measurement errors*. The values of the MCMC samples are known perfectly, without any uncertainty.

For a general AR(1) process with $\mu \neq 0$, the likelihood function is given by

$$\mathcal{L} = p(x_1, x_2, \ldots, x_N | \mu, \sigma^2)\,, \tag{18}$$

which we can factorise using the conditional probabilities such that

$$\mathcal{L} = p(x_N | x_{N-1}, \ldots, x_1, \mu, \sigma^2) p(x_{N-1} | x_{N-2}, \ldots, x_1, \mu, \sigma^2) \ldots p(x_2 | x_1, \mu, \sigma^2) p(x_1 | \mu, \sigma^2)\,. \tag{19}$$

Given that this is an AR(1) process, i.e., a Markov chain, any x_n only depends explicitly on its direct predecessor x_{n-1} such that we can drop all others and obtain

$$\mathcal{L} = p(x_N | x_{N-1}, \mu, \sigma^2) p(x_{N-1} | x_{N-2}, \mu, \sigma^2) \ldots p(x_2 | x_1, \mu, \sigma^2) p(x_1 | \mu, \sigma^2)\,. \tag{20}$$

$$= p(x_1 | \mu, \sigma^2) \prod_{n=2}^{N} p(x_n | x_{n-1}, \mu, \sigma^2)\,. \tag{21}$$

Note that the likelihood does not completely factorise, since the observations x_n are *not* statistically independent of each other by definition of Eq. (10). In the case of an OU process, we thus obtain

$$-2\log\mathcal{L} = (N-2)\log\sigma^2 + \sum_{n=2}^{N}\frac{(x_n - \mu - \phi x_{n-1})^2}{\sigma^2}\,. \tag{22}$$

The ordinates for fitting observation x_n are provided by its own previous observations. In other words, the time series is fitting itself, it is "autoregressive". If we further use $\mu = 0$, reflecting our assumption of a converged MCMC chain, we finally obtain the objective function

$$\chi^2 = -2\log\mathcal{L} = (N-2)\log\sigma^2 + \frac{\sum_{n=2}^{N}(x_n - \phi x_{n-1})^2}{\sigma^2}\,. \tag{23}$$

Such an OU process without measurement errors is trivial to fit and the result is fully analytical. Since σ^2 is constant for all x_n and thus cancels out, we obtain the estimates

$$\frac{\partial\chi^2}{\partial\phi} = -2\frac{\sum_{n=2}^{N}(x_n - \phi x_{n-1})x_{n-1}}{\sigma^2} = 0 \quad\Leftrightarrow\quad \hat{\phi} = \frac{\sum_{n=2}^{N} x_n x_{n-1}}{\sum_{n=2}^{N} x_{n-1}^2} \tag{24}$$

$$\frac{\partial\chi^2}{\partial\sigma^2} = \frac{N-2}{\sigma^2} - \frac{\sum_{n=2}^{N}(x_n - \phi x_{n-1})^2}{\sigma^4} = 0 \quad\Leftrightarrow\quad \hat{\sigma^2} = \frac{\sum_{n=2}^{N}(x_n - \hat{\phi} x_{n-1})^2}{N-2} \tag{25}$$

from which we are actually only interested in $\hat{\phi}$. Therefore, we obtain a fully analytic estimate of the autocorrelation length,

$$\hat{\tau} = -\frac{1}{\ln\hat{\phi}} \qquad \text{or} \qquad \hat{\tau} = \frac{1}{1-\hat{\phi}} - \frac{1}{2}\,. \tag{26}$$

Figure 4 shows that this OU estimator is drastically more stable than the `acor` method. We only need around 1 000 MCMC samples to obtain a stable estimate of the autocorrelation length ($\hat{\tau} \approx 21.4$ in this case). However, even for only a few hundred MCMC samples – as are given by Aeneas – the estimate of τ is drastically less volatile and we may even take it as a useful result.

3.4 Simulations for short MCMC chains

The previous tests in Figs. 2 and 4 used real Aeneas MCMC chains from fitting synthetic BP/RP spectra (cycle 11 simulations). However, these chains were artificially elongated to have 5 000 iterations, whereas in practice Aeneas MCMC chains will only have 100 iterations.

In order to get a more realistic impression of the OU method, we now simulate MCMC chains of 100 iterations. These chains now do *not* come from Aeneas but they are simulations drawn

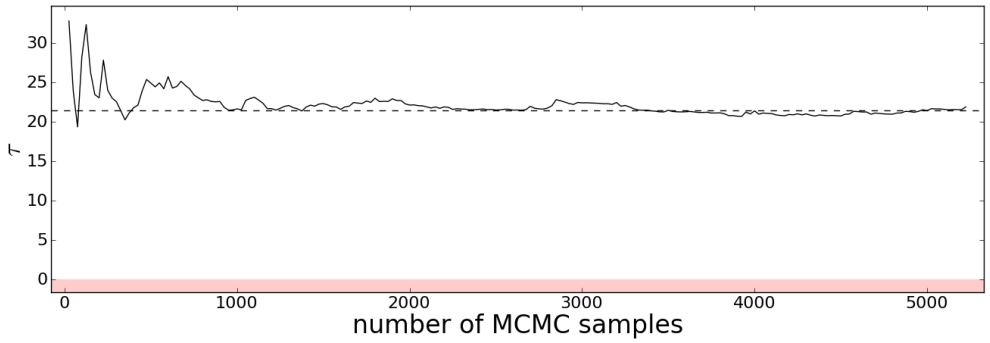

Figure 4: Estimates of τ from Eq. (26) vs. number of MCMC samples used. The red area marks $\tau < 0$.

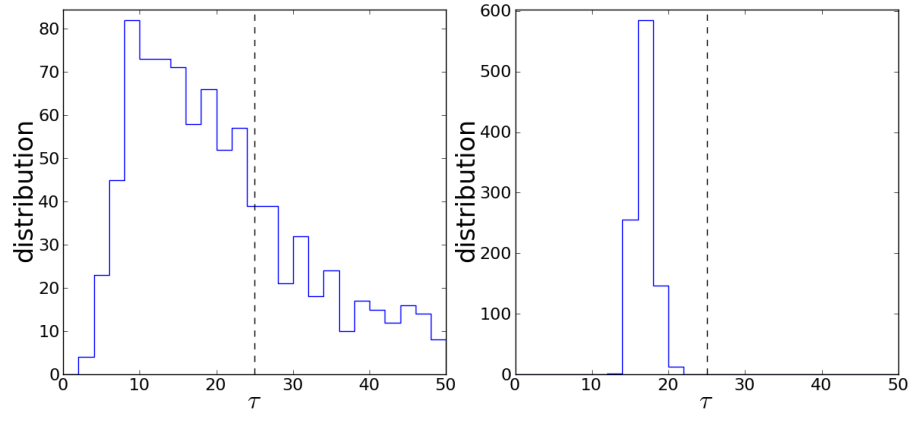

Figure 5: Estimates of τ from Eq. (26) based on MCMC chains of 100 iterations. 1 000 such MCMC chains were simulated. Left panel: Estimates of τ from a single MCMC chain. Right panel: Estimates of τ from ensemble of 100 walkers. The vertical dashed lines indicate the true value of $\tau = 25$.

from an OU process with a known true autocorrelation length of $\tau = 25$. This value of τ is inspired by Fig. 4. Can we use the OU method to reliably estimate τ?

Figure 5 (left panel) shows that estimation of τ is rather noisy from a single MCMC chain of 100 iterations. However, it is still much better than the `acor` method. Fortunately, in Aeneas, we are not only given a single MCMC chain, but rather an ensemble of 100 `emcee` walkers. We can therefore obtain an ensemble estimate of τ. We have two options: First, we could get a direct ensemble average of $\hat{\tau}$ itself, i.e., we estimate a separate $\hat{\tau}_w$ for every walker and then obtain $\hat{\tau}_{\text{ensemble}} = \frac{1}{W}\sum_{w=1}^{W}\hat{\tau}_w$. Second, we could obtain an ensemble average of $\hat{\phi}$ and calculate $\hat{\tau}$ from that. Obviously, the second approach is more robust because it can compensate for some walkers w getting $\hat{\phi}_w < 0$ as long as the overall ensemble average,

$$\hat{\phi}_{\text{ensemble}} = \frac{1}{W}\sum_{w=1}^{W}\hat{\phi}_w\,, \tag{27}$$

remains strictly positive such that $\hat{\tau} = -\frac{1}{\ln\hat{\phi}_{\text{ensemble}}}$ is still defined. Conversely, the first approach will fail as soon as a single walker has $\hat{\phi}_w \leq 0$. As Fig. 5 (right panel) shows, the ensemble estimate of Eq. (27) is much better than the single-chain estimate. Therefore, the ensemble estimate of τ is to be favoured.

Unfortunately, the right panel of Fig. 5 also clearly shows that the ensemble estimate of τ is biased low.[5] The reason is very simple: The MCMC chains have only 100 iterations and thus they are too short to estimate an autocorrelation length of $\tau = 25$. Figure 6 shows that if we increase the length of the MCMC chains from 100 to 5 000 iterations, we indeed obtain unbiased estimates of τ.[6]

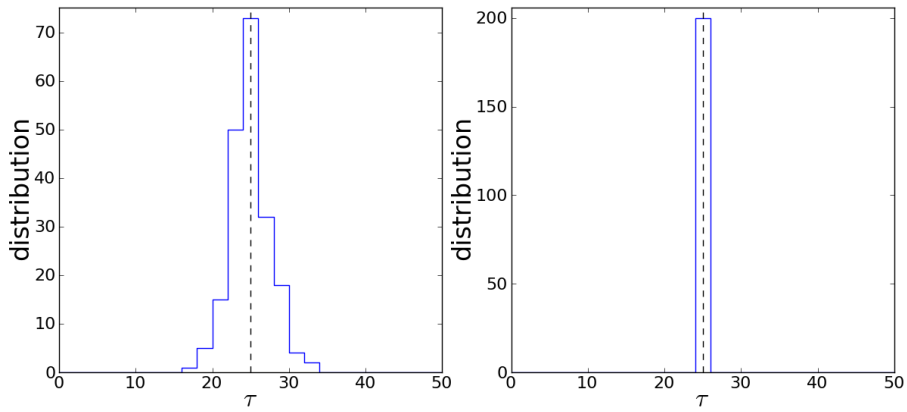

Figure 6: Same as Fig. 5 but now for MCMC chains of 5 000 iterations instead of only 100.

It is totally unrealistic to run Aeneas for 5 000 iterations because that would exceed the DPCC

[5] Most likely, also the single-chain estimate is biased low, but we cannot see it due to the noise level.
[6] Remember from Fig. 2 that the `acor` method still gives horribly volatile estimates for 5 000 MCMC samples.

limitations on computational cost by more than an order of magnitude. In Fig. 7, we investigate the bias in τ as a function of number of MCMC iterations. So far, Aeneas was run for 100 fit iterations. As of the cycle 19 software delivery (RAN-031, RAN-032), this number has been mildly increased to 140 fit iterations. In both cases, the MCMC chains are clearly too short to obtain an unbiased estimate of τ. Even in the most optimistic scenario, Aeneas will never run longer than 1 000 fit iterations, which would still give biased estimates of τ according to Fig. 7.

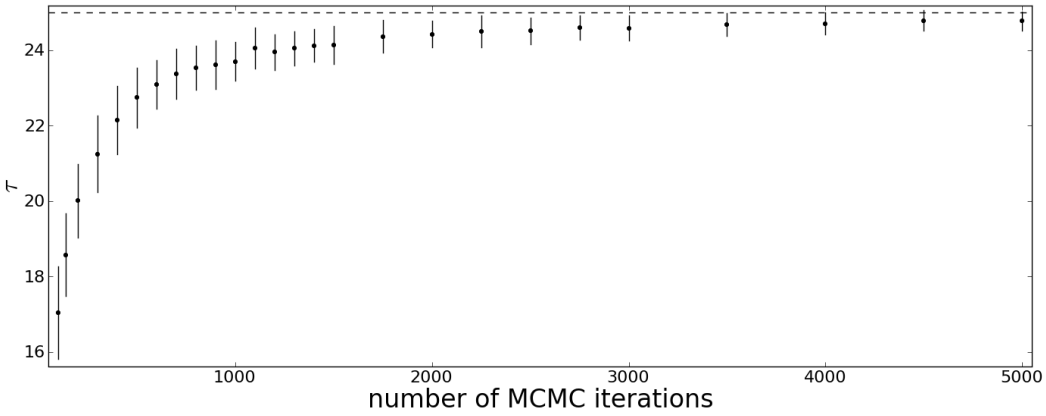

Figure 7: Bias in ensemble estimate of τ as a function of number of MCMC iterations. Note that this bias is *not* a problem of the OU method but it originates from the MCMC chains being too short in comparison to the true underlying autocorrelation length of $\tau = 25$ used in these simulations (horizontal dashed line).

Figure 8 also shows that, for fixed length of MCMC chains, the bias gets stronger if the true value of τ increases. This conforms with our understanding of this bias. Moreover, Fig. 8 shows that the bias is a monotonic one-to-one mapping between true and estimated values, which makes it possible to apply a bias correction. Such a bias correction would make no additional assumptions – apart from that of an OU process which we have already invoked anyway. As shown in Fig. 8 (red lines), the debiasing functions can be well approximated as

$$\tau_{100}^{\text{debiased}} = 0.73626441 \cdot \hat{\tau}_{100} + 0.04498744 \cdot \hat{\tau}_{100}^2 \tag{28}$$

$$\tau_{140}^{\text{debiased}} = 0.83312381 \cdot \hat{\tau}_{140} + 0.02810098 \cdot \hat{\tau}_{140}^2 \tag{29}$$

for MCMC chains of lengths 100 and 140 iterations, respectively. Note again, that these bias correction functions make no additional assumptions other than the OU process and are only valid for MCMC chains of lengths $N = 100$ and $N = 140$, respectively.

However, there may be a better way to estimate τ, which may completely avoid this "finite-chain-length bias". To see this, we need to go back to Eq. (17). Instead of summing over an infinitely long chain, we set the MCMC chain length N as the upper limit of the summation, such that Eq. (17) now reads

$$\hat{\tau}_{\text{int}}(N) = \frac{1}{2} + \sum_{t=1}^{N} \hat{\phi}^t = -\frac{1}{2} + \sum_{t=0}^{N} \hat{\phi}^t = \frac{1 - \hat{\phi}^{N+1}}{1 - \hat{\phi}} - \frac{1}{2}\,. \tag{30}$$

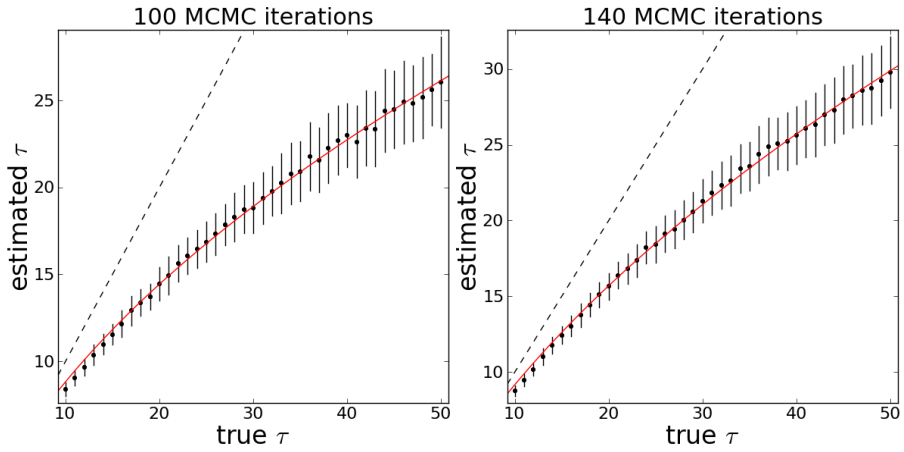

Figure 8: Debiasing τ for MCMC chains of 100 iterations (left panel) and 140 iterations (right panel). Error bars were estimated by simulating 100 MCMC chains in every configuration. The diagonal dashed line indicates unbiased estimates. Red lines show the debiasing functions defined in Eqs. (28) and (29).

By design, this estimator explicitly takes into account the finite length of the MCMC chain, exploiting the nice mathematical properties of the geometric series. Is this sufficient to avoid the "finite-chain-length bias"? Figure 9 shows that the "finite-chain-length bias" is indeed reduced slightly but unfortunately it is still very prominent. Therefore, the estimator of Eq. (30) offers no real benefit.

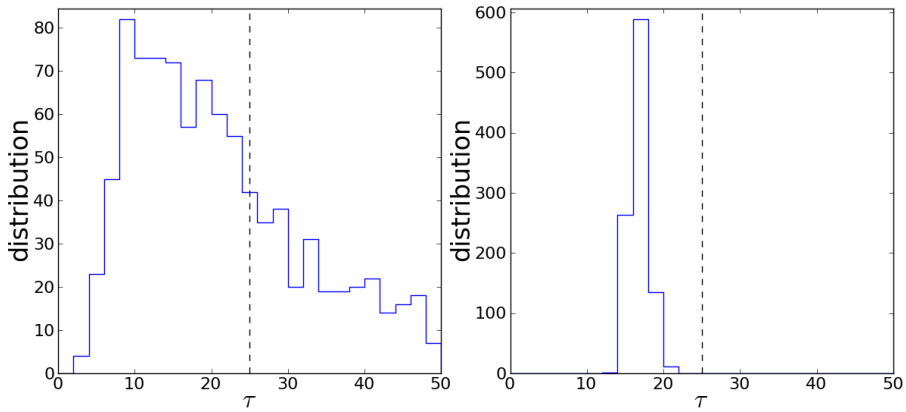

Figure 9: Same as Fig. 5 but using Eq. (30) to estimate τ.

4 Autocorrelation length as MCMC quality control

How can we use the autocorrelation length as a quality control? Figure 10 shows that for most of the stars in that test data set, the estimated autocorrelation lengths range between 10 and 15,

corresponding to debiased values of ca. 12 to 20 (c.f. right panel of Fig. 8). However, not all stars fall into that range. There are also a few examples (14 out of 10 000) where τ takes very large or even negative values.[7] All these examples are shown in Fig. 11 and we can clearly see that almost all of these MCMC chains exhibit multimodalities or prominent tails, indicating a lack of convergence.

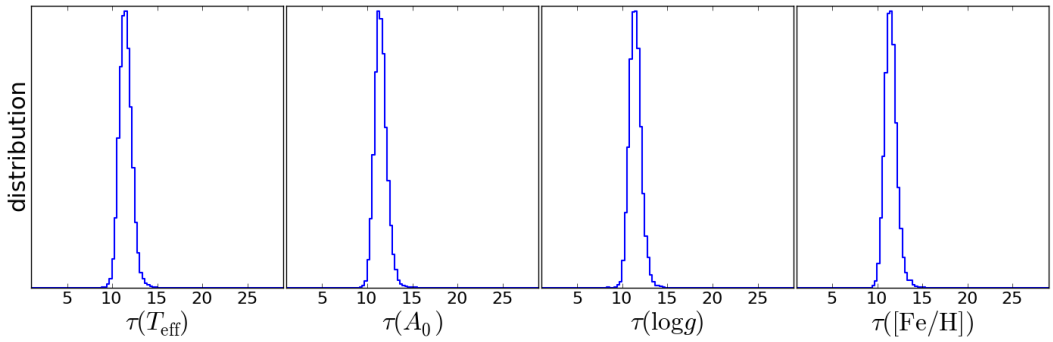

Figure 10: Distributions of (biased) autocorrelations lengths for $T_{\rm eff}$, A_0, $\log g$, [Fe/H] based on 10 000 stars from Main Stellar Library PHOENIX at $G = 15$.

For comparison, Fig. 12 shows 20 randomly selected examples where $8 < \tau < 25$ for all autocorrelation lengths of all four parameters. These clearly look much more well behaved than those examples shown in Fig. 11. In fact, while Fig. 12 only displays 20 examples, we visually inspected a total of 100 random examples and found that only $\approx 2\%$ exhibit minor signs of failed convergence and another $\approx 4\%$ exhibit multimodalities. These findings suggest that $8 < \tau < 25$ may be a good way to rule out the most ill-behaved MCMC chains.

5 Conclusions

The `acor` method advocated in Foreman-Mackey et al. (2013) is clearly not useful for GSP-Phot Aeneas, given the short lengths of these MCMC chains. The estimated autocorrelation lengths are so volatile that they are totally useless. Conversely, the OU method provides much more stable estimates of MCMC autocorrelation length. The reason is that the OU method makes rather severe assumptions by modelling the converged MCMC chain as a stationary Gaussian AR(1) process (Ornstein-Uhlenbeck process) without any trends. This model leads to an exponentially suppressed autocovariance function, whose autocorrelation length can be calculated fully analytically from the given MCMC samples. Let us emphasise that the OU method may lead to biased estimates of the MCMC autocorrelation length, namely if the MCMC chain is no OU process. This can obviously happen, if the MCMC chain itself is non-Gaussian. As an example, the strong degeneracy between $T_{\rm eff}$ and A_0 can take a "banana-shaped" contour, that is clearly non-Gaussian, thus violating the OU model assumptions.

[7] A negative value of $\hat{\tau}$ can occur if $\hat{\phi}_{\rm ensemble} > 1$, i.e., if the MCMC chain appears to be non-stationary.

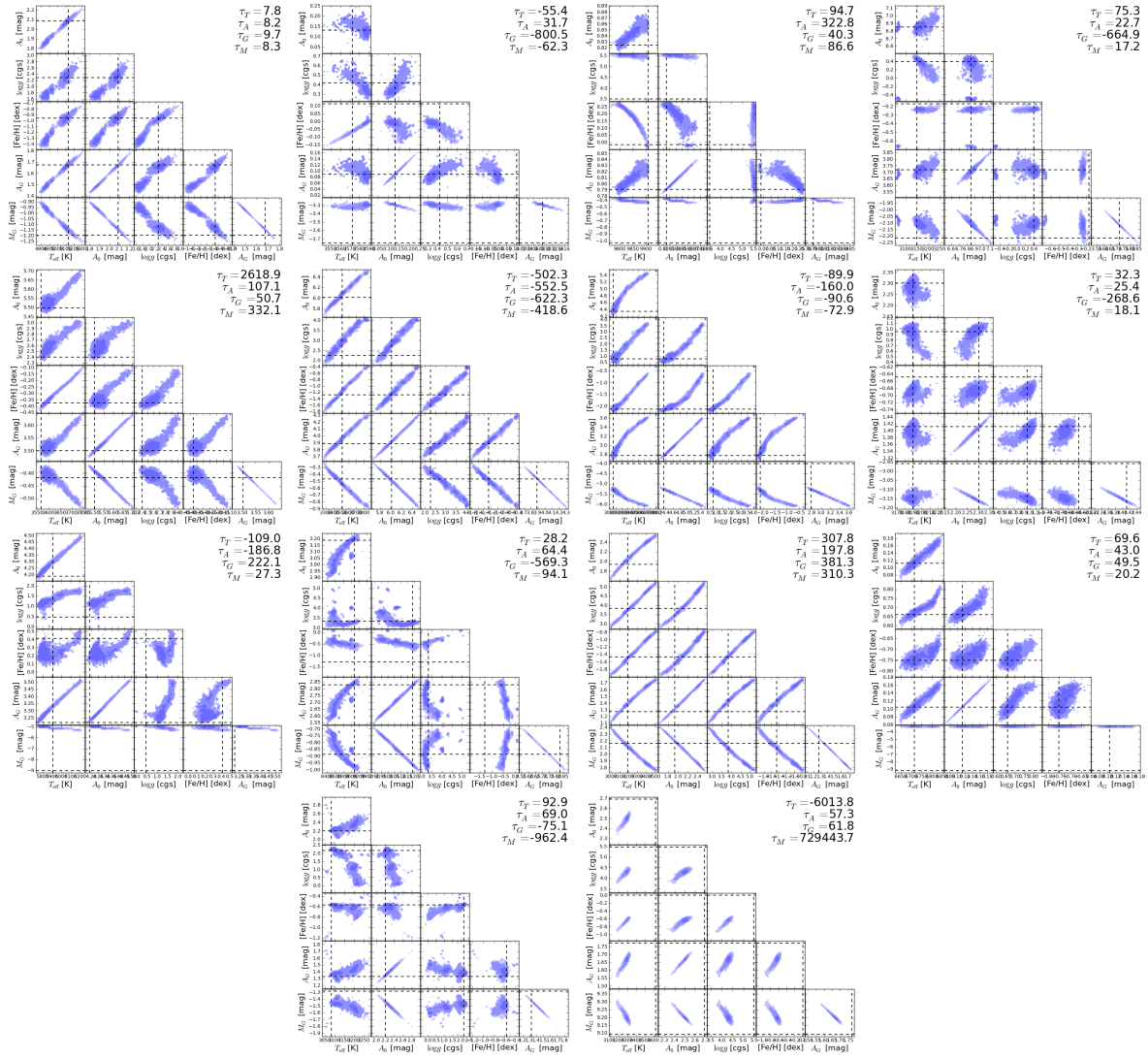

Figure 11: Examples of stars with negative or very large values of τ. For each corner plot, the autocorrelation lengths for all four APs are quoted in the top right corner.

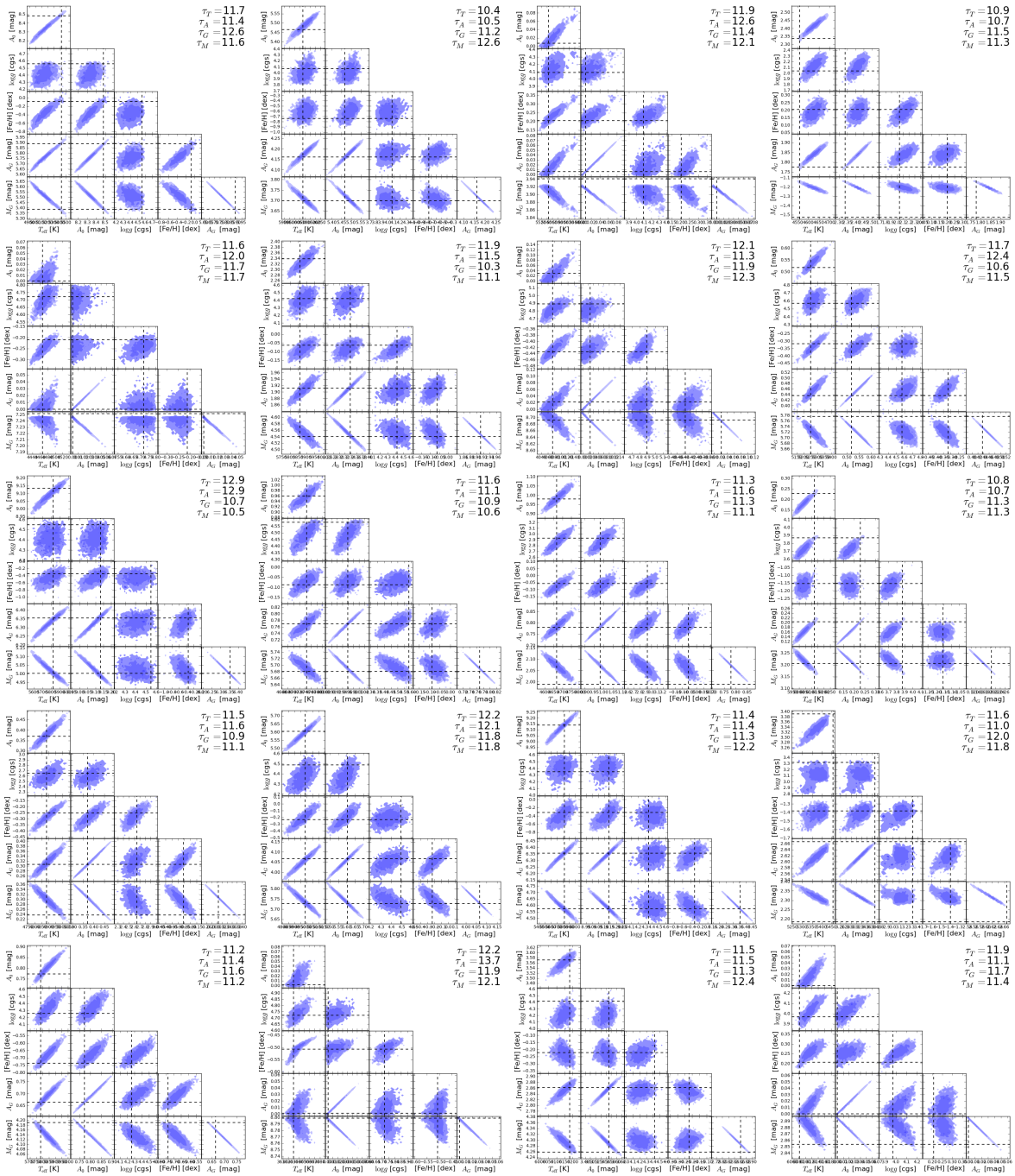

Figure 12: Randomly chosen examples of stars with $8 < \tau < 25$. For each corner plot, the autocorrelation lengths for all four APs are quoted in the top right corner.

For the cycle 19 software delivery (cycle 2 operations, RAN-031, RAN-032) it was decided to implement the OU model, using an `emcee` ensemble estimate of ϕ via Eq. (27), from which $\hat{\tau} = -\frac{1}{\ln \hat{\phi}_{\text{ensemble}}}$ is computed. Furthermore, it was decided *not* to implement the debiasing of $\hat{\tau}$ via Eq. (29) because this can be done offline later. Preliminary tests showed that most stars have $10 < \tau < 15$ and that $8 < \tau < 25$ may be a good quality control to rule out most of bad MCMC chains which have not converged or exhibit strong multimodalities.

References

Foreman-Mackey, D., Hogg, D.W., Lang, D., Goodman, J., 2013, , 125, 306, ADS Link

Gillespie, D.T., 1996, Phys. Rev. E, 54, 2084,
URL `http://link.aps.org/doi/10.1103/PhysRevE.54.2084`

Sokal, A., 1997, In: DeWitt-Morette, C., Cartier, P., Folacci, A. (eds.) Functional Integration, vol. 361 of NATO ASI Series, 131–192, Springer US,
URL `http://dx.doi.org/10.1007/978-1-4899-0319-8_6`

Uhlenbeck, G.E., Ornstein, L.S., 1930, Phys. Rev., 36, 823,
URL `http://link.aps.org/doi/10.1103/PhysRev.36.823`

A Uncertainty estimates and integrated autocorrelation length

In this appendix, we want to discuss the impact of autocorrelations onto MCMC parameter estimation. Our discussion follows Sokal (1997) with some additional explanations.

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a given MCMC chain, say values of T_{eff} from GSP-Phot Aeneas. How do we obtain an estimate of the value of T_{eff} and its uncertainty? The parameter estimate itself is simply given by the estimated mean of all MCMC samples,

$$\hat{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \,. \tag{31}$$

This mean estimator is *not* compromised by the autocorrelations in the MCMC chain. It is unharmed and if enough samples have been discarded after initialisation, it will provide unbiased estimates. However, the autocorrelations in the MCMC chain have a profound impact onto the uncertainty estimate of x.[8] Since the covariance between any two terms x_i and x_j in the

[8]Note that Sokal (1997) consider the uncertainty in $\hat{x}$ instead of x, which gives a factor $\sqrt{N}$ difference between the two estimates. For parameter estimation from MCMC, we are interested in the uncertainty of x but not in the uncertainty of its mean value. The latter would decrease towards zero as the number of MCMC samples is increased, regardless of the data quality.

summation of Eq. (31) is given by Eq. (14), we have

$$C(i,j) = C(|i-j|) = \sigma^2 \phi^{|i-j|} = \sigma^2 e^{-|i-j|/\tau_{\rm exp}} \,. \tag{32}$$

As we know, if X had no correlations whatsoever, the variance of the sum of Eq. (31) would simply be the sum of the variances, such that we would obtain $\hat{\sigma}^2 = \frac{1}{N} \cdot N \cdot \sigma^2 = \sigma^2$. In the presence of covariances, described by a covariance matrix Σ, the variance of the sum is rather $\boldsymbol{e}^T \cdot \Sigma \cdot \boldsymbol{e}$, where $\boldsymbol{e} = (1, 1, \ldots, 1)^T$. We therefore obtain the variance of the sum of Eq. (31),

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i,j=1}^{N} C(i,j) = \frac{1}{N} \sum_{i,j=1}^{N} C(|i-j|) \,. \tag{33}$$

The term $C(|i-j|)$ clearly only depends on $t = i - j$, which ranges from $t = -(N-1)$ to $t = N-1$. For fixed $t = i - j$, the term $C(|i-j|)$ is constant and does not depend on which values of i and j actually produce $t = i - j$. Now, if the value of $t = i - j$ is fixed, then there are $N - |t|$ different combinations of i and j that can produce this value of t. We can therefore write

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{t=-(N-1)}^{N-1} (N - |t|)\, C(|t|) \tag{34}$$

$$= \sum_{t=-(N-1)}^{N-1} \left(1 - \frac{|t|}{N}\right) C(|t|) \,. \tag{35}$$

Given that $C(|t|) \propto e^{-|t|/\tau_{\rm exp}}$ and assuming that $\tau_{\rm exp} \ll N$, we can approximate $\frac{|t|}{N} C(|t|) \ll C(|t|)$ such that we obtain

$$\hat{\sigma}^2 \approx \sum_{t=-(N-1)}^{N-1} C(|t|) = C(0) \sum_{t=-(N-1)}^{N-1} \frac{C(|t|)}{C(0)} \,. \tag{36}$$

Using Eq. (7), we can finally approximate

$$\hat{\sigma}^2 \approx 2\tau_{\rm int} C(0) = 2\tau_{\rm int} \sigma^2 \,, \tag{37}$$

i.e., the sample variance estimated from the MCMC chain is too large by a factor of $\approx 2\tau_{\rm int}$. The reason is that the sample variance estimate normalises by the factor N, whereas the number of "effectively independent" samples is only $\frac{N}{2\tau_{\rm int}} < N$. In other words, the autocorrelations in the MCMC chain lead the sample variance to overestimate the actual variance.

If we want to take the OU model literally, we can return to Eq. (35) and compute a precise result for the factor by which we need to correct the sample variance. Splitting the two terms and using $C(|t|) = \sigma^2 \phi^{|t|}$, we obtain

$$\hat{\sigma}^2 = \sigma^2 \left(\sum_{t=-(N-1)}^{N-1} \phi^{|t|} - \frac{1}{N} \sum_{t=-(N-1)}^{N-1} |t| \phi^{|t|} \right) \tag{38}$$

$$= \sigma^2 \left(-1 + 2 \sum_{t=0}^{N-1} \phi^t - \frac{2}{N} \sum_{t=0}^{N-1} t\phi^t \right) \tag{39}$$

$$= \sigma^2 \left(-1 + 2 \frac{1-\phi^N}{1-\phi} - \frac{2}{N} \frac{(N-1)\phi^{N+1} - N\phi^N + \phi}{(1-\phi)^2} \right) \tag{40}$$

$$= \sigma^2 \left(-1 + 2 \frac{1-\phi^N}{1-\phi} - \frac{2}{N} \frac{\phi - \phi^{N+1} + N\phi^{N+1} - N\phi^N}{(1-\phi)^2} \right) \tag{41}$$

$$= \sigma^2 \left(-1 + 2 \frac{1-\phi^N}{1-\phi} - \frac{2}{N} \frac{\phi(1-\phi^N) - N\phi^N(1-\phi)}{(1-\phi)^2} \right) \tag{42}$$

$$= \sigma^2 \left(\frac{\phi - 1}{1-\phi} + \frac{2 - 2\phi^N}{1-\phi} + \frac{2\phi^N}{1-\phi} - \frac{2\phi(1-\phi^N)}{N(1-\phi)^2} \right) \tag{43}$$

$$= \sigma^2 \left(\frac{1+\phi}{1-\phi} - \frac{2\phi(1-\phi^N)}{N(1-\phi)^2} \right) . \tag{44}$$

We can use this factor to correct the sample variance $\hat{\sigma}^2$ in order to obtain the actual variance σ^2.