DRAFT: JULY 5, 2014
Preprint typeset using LaTeX style emulateapj v. 12/16/11

A METHOD FOR THE SPECTROSCOPIC INFERENCE OF STELLAR PARAMETERS

IAN CZEKALA, SEAN M. ANDREWS, ET AL.
Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138

1. INTRODUCTION

A single stellar spectrum contains a vast amount of information about the physical properties of the star. It is of supreme interest to extract a maximal amount of this information from the spectrum in order to study the fundamental stellar parameters such as effective temperature, surface gravity, and metallicity. Accurate and unbiased estimates of stellar parameters are crucial to the study of numerous fields of astrophysics. For example, the mass, radius, and temperature of every newly discovered exoplanet (save for directly imaged planets) are measured relative to the properties of its host star. Large uncertainties or biases in the stellar parameters will translate into incorrect estimates of the habitable zone and the occurrence rate of earth-like planets (Kane 2014). High quality stellar parameters are also necessary for the study of stellar evolution. Modelers of stellar evolution rely upon accurate samples of stellar parameters to chart the evolutionary tracks of stars through the Hertsprung-Russell diagram.

However, extracting physical parameters of stars from a stellar spectrum is often difficult, due to several factors. First, generating synthetic stellar spectra to use as a benchmark for comparison requires a sophisticated model of the stellar atmosphere and radiative transfer, along with an extensive, high-quality database of opacity sources. Accurately calibrating the atomic constants in this database is a laborious task, and requires high resolution, high signal-to noise spectra for verification, which limits targets mainly to extremely bright stars like the Sun and Vega. This makes it difficult to achieve the same precision when extending synthetic stellar models to intrinsically fainter stars, like M dwarfs, which have additional sources of molecular opacity. Second, synthesizing a spectrum with such a sophisticated model requires a large amount of computational power, such that a full-optical spectrum at high resolution will take a day or more to process on a supercomputer.

Several, well-tested techniques exist that use spectra to infer stellar parameters. The most widely used technique relies upon measuring ratios of lines, or line indexes, which are matched to a set of indices measured from a synthetic spectrum. More sophisticated codes exist that synthesize spectra for direct comparison with certain well-studied lines, (e.g., Fe , Mg , and Ca) such as MOOG (Sneden 1973) and SME (Valenti & Piskunov 1996). Other codes such as SPC (Buchhave et al. 2012) cross-correlate the observed spectrum against a grid of pre-computed synthetic spectra and then fit a surface to the normalized cross-correlation coefficients to determine parameter uncertainties.

Owing to the high quality of libraries of pre-computed synthetic stellar spectra, it is now possible to generate an accurate forward model of a stellar spectrum. We seek to place this model in a Bayesian framework and construct a likelihood function suitable for comparing synthetic spectral models to real spectral data. We use a non-trivial covariance matrix, parameterized by a Gaussian process covariance kernel, to account for the residual pixel correlations that arise from a spectroscopic fit, which is a more complex problem than can be addressed with a simple χ^2 best-fit metric. While designed for fitting stellar spectra, our method is general for all comparisons of spectra, and could be used to fit any type of spectra with a synthetic model, (e.g., galaxies or unresolved stellar clusters). In §2, we describe the methodology, including a treatment of the covariance introduced by a spectroscopic fit. In §3, we present applications of our method to two examples, using optical spectra of an F type star and infrared spectra of an M star. In §4 we discuss the implications of our method and potential applications to develop data-driven models of stellar spectra.

2. METHODOLOGY

All astronomers recognize that spectroscopy offers a wealth of information that can help characterize the fundamental properties of the observing target. However, as noted in Section 1, the reliable and statistically robust inference of those physical (or empirical) parameters from an observed spectrum can be extraordinarily challenging. Here we describe a generative Bayesian modeling framework that confronts some of the key obstacles in that process. The goal of this approach is to conservatively extract the maximal amount of information about a prescribed (and degenerate) parameter set by forward-modeling an observed spectrum, while also recognizing and explicitly accounting for the covariances and biases introduced by pathologically imperfect models or calibrations. The method is modular, and therefore can easily incorporate additional physical or nuisance parameters as desired without sacrificing an accurate reflection of the limitations in the data. Moreover, with a well-crafted observational sample, this data-driven approach should ultimately enable us to systematically learn how synthetic spectral models can be improved. The specific applications discussed here are related to the spectra of individual stars, but the methodology is generic (and could be used for composite spectra of unresolved stellar clusters, galaxies, etc.).

The remainder of this section describes the mechanics of this modeling framework. First, a model spectrum is generated for a given set of physical parameters (Section 2.1), and then post-processed to mimic reality using a set of observational and practical nuisance parameters (Section 2.2). Next, a direct, pixel-by-pixel comparison between the data and model spectra is made with a prescribed likelihood function and a parametric treatment of the covariances between pixel residuals (Section 2.3).

That process is iterated under the guise of hierarchical Monte Carlo Markov Chain (MCMC) simulations to numerically explore the posterior probability density of the model conditioned on the data, and thereby to determine constraints on the parameters of interest (Section 2.4). Along the way, these procedures are illustrated with real observations of the high resolution optical spectrum from a nearby F star. That specific application, along with some alternative demonstrations of the method, are discussed in more detail in Section 3.

2.1. *Generating a Model Spectrum*

There are various approaches to synthesizing a spectrum, f_λ, for a specific set of model parameters, $\boldsymbol{\theta}_*$. In an ideal case, a model stellar atmosphere is constructed and then subsequently processed through a radiative transfer code (e.g., Kurucz 1993; Hauschildt et al. 1999). However, in general this approach is still computationally prohibitive for any iterative method of probabilistic inference. One partial compromise is to interpolate over a library of atmosphere structures that were pre-computed for a discrete grid of parameter values, $\{\boldsymbol{\theta}_*\}^{\rm grid}$, for some arbitrary $\boldsymbol{\theta}_*$, and then perform a radiative transfer calculation with that interpolated atmosphere to synthesize f_λ (e.g., as for SME; Valenti & Piskunov 1996). A more common variant is to instead rely on interpolation over a library of pre-synthesized model spectra, $f_\lambda(\{\boldsymbol{\theta}_*\}^{\rm grid})$ (e.g., Castelli & Kurucz 2004; Allard et al. 2012; Husser et al. 2013). While technically the former approach is most similar to the ideal case, the computational cost of repeated spectral synthesis is sufficiently high to make a detailed exploration of parameter space (particularly for data with a large spectral range) considerably less appealing. A related, but different approach is to eschew forward modeling entirely (and therefore repeated spectral syntheses and/or library interpolations), and instead evaluate the models only at the discrete grid points of the library. Then, these discretized samples of the posterior probability density can be interpolated to an arbitrary $\boldsymbol{\theta}_*$ to construct appropriate confidence intervals (e.g., the method of SPC; Buchhave et al. 2012). The difficulty with this latter approach is that the parameter uncertainties can be smaller than the grid spacing; in that case, there is valid concern that this interpolation might not accurately recover intrinsic parameter degeneracies.

Here we opt to take the computationally expedient approach that employs a library of model spectra, $f_\lambda(\{\boldsymbol{\theta}_*\}^{\rm grid})$, where $\boldsymbol{\theta}_* = [T_{\rm eff},\ \log g,\ Z]$ (in practice, the metallicity Z is often parameterized by the iron abundance, [Fe/H]). However, it is worth noting that the techniques we will develop are applicable to *any* approach for generating a model spectrum. In our adopted approach, the model spectrum for an arbitrary $\boldsymbol{\theta}_*$ must be interpolated from among the spectral library,

$$f_\lambda(\{\boldsymbol{\theta}_*\}^{\rm grid}) \rightsquigarrow f_\lambda(\boldsymbol{\theta}_*), \tag{1}$$

where we assign the symbol $\rightsquigarrow$ as an interpolation operator. The multi-dimensional interpolation in Eq. 1 needs to be performed many times, so computational efficiency is critical. In practice, a simple tri-linear interpolation is suitably fast, but introduces an undesireable level of inaccuracy (particularly in the Z dimension). The interpolation quality can be empirically estimated by performing the operation in Eq. 1 across a calculated location in $\{\boldsymbol{\theta}_*\}^{\rm grid}$, and then comparing the interpolated spectrum with the corresponding library spectrum. After an extensive exploration of such calculations (see also Husser 2012), we concluded that the best combination of speed and accuracy can be achieved by pre-computing a *refined* spectral library using a cubic spline interpolation with a $\{\boldsymbol{\theta}_*\}^{\rm grid}$ spacing of [20 K, 0.1 dex, 0.1 dex], and then performing tri-linear interpolation over that refined grid. Overall, this interpolation technique is found to be accurate within a few percent per high resolution model pixel. Ideally, this pre-interpolation could be avoided if the spectral library was computed over a refined grid (with a substantial up-front computational investment); but for the time being, we can empirically propagate these interpolation uncertainties into the likelihood calculations (as will be described in Section 2.3).

2.2. *Post-Processing*

Generally, the "raw" model spectrum $f_\lambda(\boldsymbol{\theta}_*)$ will be highly over-sampled compared to a typical observed spectrum, and does not account for several additional observational and instrumental effects that become important in comparisons with real data. Therefore, a certain amount of post-processing is required before assessing the model quality. We treat that post-processing in two stages: the first deals with an additional set of "observational" parameters, $\boldsymbol{\theta}_{\rm obs}$, that incorporate dynamical effects, geometry, and the relative location of the target, while the second employs a suite of nuisance (hyper-)parameters, $\boldsymbol{\Theta}_{\rm n}$, designed to mitigate an imperfect data calibration.

We can further divide $\boldsymbol{\theta}_{\rm obs}$ into those parameters that impact the model primarily in the spectral or flux dimensions; $\boldsymbol{\theta}_{\rm obs} = [\boldsymbol{\theta}_{\rm obs,v},\ \boldsymbol{\theta}_{\rm obs,f}]$. For the former, we consider three kernels that contribute to the observed line-of-sight velocity distribution function, φ_v. The first, $\mathcal{F}_v^{\rm inst}$, treats the instrumental spectral broadening. For illustrative purposes we assume $\mathcal{F}_v^{\rm inst}$ is a Gaussian with a mean at $v = 0$ and a constant width σ_v at all λ, although more sophisticated forms could be adopted. The second, $\mathcal{F}_v^{\rm rot}$, characterizes the broadening induced by (projected) stellar rotation, parameterized by $v \sin i$ as described by Gray (2008, his Eq. 18.14). And the third, $\mathcal{F}_v^{\rm dop} = \delta(v - v_z)$, incorporates the radial velocity through a Doppler shift. The model spectrum is modified by the parameters $\boldsymbol{\theta}_{{\rm obs},v} = [\sigma_v,\ v \sin i,\ v_z]$ through these kernels, using a convolution in velocity-space,

$$f_\lambda(\boldsymbol{\theta}_*, \boldsymbol{\theta}_{{\rm obs},v}) = f_\lambda(\boldsymbol{\theta}_*) \otimes \varphi_v = f_\lambda(\boldsymbol{\theta}_*) \otimes \mathcal{F}_v^{\rm inst} \otimes \mathcal{F}_v^{\rm rot} \otimes \mathcal{F}_v^{\rm dop}, \tag{2}$$

and then re-sampled onto the discrete wavelengths corresponding to each data pixel,

$$f_\lambda(\boldsymbol{\theta}_*, \boldsymbol{\theta}_{{\rm obs},v}) \mapsto \mathsf{M}(\boldsymbol{\theta}_*, \boldsymbol{\theta}_{{\rm obs},v}), \tag{3}$$

where the $\mapsto$ symbol denotes a re-sampling operator that maps the model spectrum onto the $N_{\rm pix}$-dimensional array M (and $N_{\rm pix}$ is the number of pixels in the spectrum). For reference, Figure 1 shows a graphical representation of these post-processing steps.

At this stage, the model is further modified in the flux dimension. A typical synthetic spectrum is computed as the flux that would be measured *at the stellar surface*, and so needs to be diluted by the subtended solid

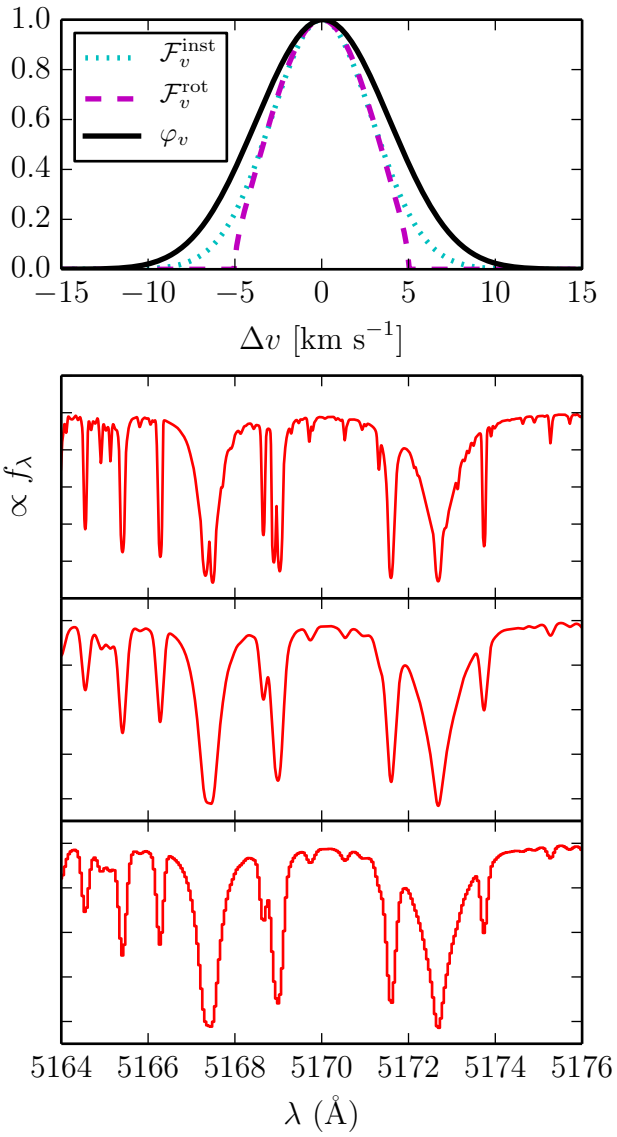

Figure 1. Panel 1: the instrumental (FWHM = 6.8 km s^{-1}) and rotational ($v \sin i$ = 5 km s^{-1}) broadening kernels, and their composite. Panel 2: A section of the raw, high resolution synthetic model at R =500,000. Panel 3: The spectrum after applying the composite broadening kernel, φ_v. Panel 4: The broadened spectrum downsampled to the pixels of the detector.

angle, $\Omega = (R_*/d)^2$, where R_* is the stellar radius and d is the distance. An additional wavelength-dependent scaling factor is applied to account for interstellar extinction, assuming a derived extinction law A_λ (e.g., Cardelli et al. 1989) that is parameterized by A_V. The parameters $\boldsymbol{\theta}_{{\rm obs},f} = [\Omega,\ A_V]$ are applied as

$$\mathsf{M}(\boldsymbol{\theta}) \equiv \mathsf{M}(\boldsymbol{\theta}_*, \boldsymbol{\theta}_{\rm obs}) = \mathsf{M}(\boldsymbol{\theta}_*, \boldsymbol{\theta}_{{\rm obs},v}) \times \Omega \times 10^{-0.4\,A_\lambda}, \quad (4)$$

where we have simplified the notation by writing $\boldsymbol{\theta} \equiv [\boldsymbol{\theta}_*,\ \boldsymbol{\theta}_{\rm obs}]$.

So far, the procedure summarized in Eq. 1-4 is composed of relatively straightforward operations demanded by practical astronomical and computing issues. If the observed spectrum was *perfectly* calibrated, we could proceed to a likelihood calculation (Section 2.3) that makes a direct comparison with $\mathsf{M}(\boldsymbol{\theta})$ at this point. However, that is unlikely to be the case. The primary concern is that an imperfect calibration produces (presumably low-level) mismatches in the underlying shape of the observed spectrum on relatively broad wavelength scales. When compared with the model, these mismatches might represent a non-trivial contribution to the likelihood, and thereby bias our estimates of the desired physical parameters. Often, this concern is treated externally to any modeling procedure, usually by dividing the observed spectrum (and model spectrum) by an appropriate polynomial function. But that "normalization" procedure implicitly assumes that there is no relevant information content on those mismatched scales; if $\boldsymbol{\theta}$ also contributes to the broad spectral shape, then adopting this approach will corrupt the inferences of these parameters. Moreover, in practice this approach is limited, since defining an appropriate polynomial becomes difficult in cases where the spectral line density is high (e.g., molecular bands for cool stars).

We adopt a somewhat analogous approach to deal with this issue, but cast it internal to the modeling framework to appropriately propagate the uncertainty introduced by additional degrees of freedom in the model. The residual calibration errors are treated as an explicit contribution to the model spectrum, enabling us to explore the distribution of possible calibrations (see Section 2.4) and then eventually marginalize out the associated nuisance parameters. In essence, the inferences for the relevant stellar parameters will properly account for the underlying uncertainty in the calibration process. This is achieved in practice by distorting the model spectrum with a (low-order) Chebyshev polynomial (e.g., Eisenstein et al. 2006; Koleva et al. 2009),

$$\mathsf{M}(\boldsymbol{\theta}, \boldsymbol{\Theta}_{\rm cheb}) = \mathsf{M}(\boldsymbol{\theta}) \times \sum_n c_n\, T_n, \quad (5)$$

where T_n are the standard Chebyshev functions of order n and the coefficients are treated as a set of nuisance (hyper-)parameters, $\boldsymbol{\Theta}_{\rm cheb} = \{c_n\}^{\rm order}$, for each available spectral order in the dataset. With judicious priors on $\boldsymbol{\Theta}_{\rm cheb}$, we can ensure that the unintended treatment of real spectral features (e.g., broad, deep molecular bands) as calibration artifacts is negligible (see Section 3 for examples). The lowest-order (scaling) coefficient, c_0, is by its nature degenerate with the solid angle parameter, Ω. Therefore, we enforce an additional constraint by requiring that the mean of the polynomial is unity. For data with a single spectral order, this means setting $c_0 = 1$. If the goal is to model multiple spectral orders, we assign $c_0 = 1$ in an arbitrary order as an anchor, but permit the c_0 values in other orders to be different (as necessary). It is worth noting that this formalism can, in principle, be extended to develop models for completely uncalibrated spectra, rather than the residual calibration mismatches as described here (with a suitable relaxation of the priors on $\boldsymbol{\Theta}_{\rm cheb}$). Figure 2 offers a practical demonstration of how these nuisance parameters are applied.

2.3. *Model Evaluation*

The quality of the model spectrum is assessed by comparing to the data with a pixel-by-pixel likelihood calculation. If we denote the data spectrum as D, then

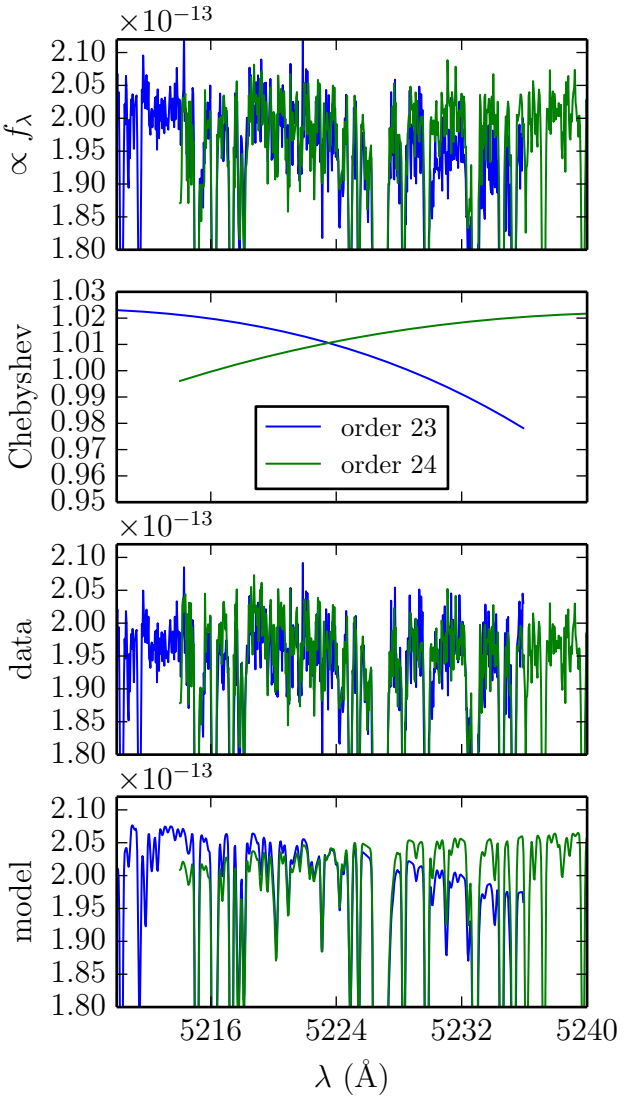

Figure 2. The spectrum at the overlap of two echelle orders (23 & 24). **Panel 1**: The flux-calibrated dataset shows a slight discrepancy of $\lesssim 3\%$ between orders. **Panel 2**: To account for this residual error in the flux calibration, we multiply the model spectrum by a Chebyshev polynomial, whose coefficients are parameters that are part of the model. **Panels 3 & 4**: In principle, we could divide the data by these polynomials to recover what the true flux-calibrated data should be (panel 3), but instead we multiply the model by the polynomials to recover the original discrepancy between orders (panel 4). This has the advantage of preserving the original dataset as a fixed quantity and makes the model (Equation 5) linear in the Chebyshev polynomial coefficients.

a corresponding $N_{\rm pix}$-dimensional residual spectrum can be defined for any input parameter set,

$$\mathsf{R} \equiv \mathsf{R}(\boldsymbol{\theta}, \boldsymbol{\Theta}_{\rm cheb}) \equiv \mathsf{D} - \mathsf{M}(\boldsymbol{\theta}, \boldsymbol{\Theta}_{\rm cheb}). \quad (6)$$

To quantify the probability of the data conditioned on the model, we adopt a standard multi-dimensional Gaussian likelihood function,

$$p(\mathsf{D}|\mathsf{M}) = \frac{1}{[(2\pi)^{N_{\rm pix}} \det(\mathsf{C})]^{1/2}} \exp\left(-\frac{1}{2}\mathsf{R}^{\mathsf{T}}\mathsf{C}^{-1}\mathsf{R}\right), \quad (7)$$

which penalizes models that yield larger residuals and explicitly allows for covariances in the residual spectrum through the $N_{\rm pix} \times N_{\rm pix}$ matrix, C. For practical numerical reasons, we generally consider the log-likelihood as the fit quality metric, which in this case is

$$\ln p(\mathsf{D}|\mathsf{M}) = -\frac{1}{2}\left(\mathsf{R}^{\mathsf{T}}\mathsf{C}^{-1}\mathsf{R} + \ln\det\mathsf{C} + N_{\rm pix}\ln 2\pi\right). \quad (8)$$

The covariance matrix C characterizes both the measurement uncertainty (σ; "noise") in each pixel and the intrinsic covariance between pixels. The special case where each pixel represents an independent measurement results in a diagonal covariance matrix, $\mathsf{C}_{ij} = \delta_{ij}\,\sigma_i$ where σ_i is the uncertainty in pixel i and δ_{ij} is the Kronecker delta function, and Eq. 8 reduces to the familiar

$$\ln p(\mathsf{D}|\mathsf{M}) = -\frac{1}{2}\sum_{i}^{N_{\rm pix}} \frac{\mathsf{R}_i^2}{\sigma_i^2} \equiv -\frac{\chi^2}{2}, \quad (9)$$

the sum of the square of the residuals weighted by the inverse variances (squared uncertainties). However, the problem being addressed here necessitates the use of a more complex covariance matrix; additional off-diagonal terms that can explicitly characterize (1) pixel-to-pixel covariances imposed by the discrete over-sampling of the line-spread function, and (2) highly correlated residuals as manifestations of the still-imperfect model library are required to avoid biasing our inferences of the physically interesting parameters ($\boldsymbol{\theta}$). The following subsections describe how these issues are addressed in the practical implementation of C.

2.3.1. *Global Covariance Structure*

Astronomical spectrographs are designed so that the detector over-samples the instrumental line broadening function with at least a few pixels. Therefore, adjacent pixels are never completely independent samples of the observed spectrum. In that case, a difference between an observed and modeled spectral feature will create a residual that spans multiple pixels. This can be demonstrated in practice by examining the autocorrelation of the residual spectrum: a slight model mismatch will produce correlated residuals over a characteristic scale similar to the observational line-broadening kernel width. Figure 3 highlights a specific example of these correlated residuals in real data, where an imperfect model generates residuals which exhibit a significant autocorrelation signal on the scale of $\sim$4 pixels, which corresponds to the typical line width in this spectrum.

It seems important to distinguish here between "noise" and the fit residuals. Noise introduced to the spectrograph by astrophysical or instrumental effects is generally uncorrelated with wavelength. The arrival of each photon to the detector is an independent event; while these photons are scattered by the instrumental line-spread function, the magnitude and direction of that scatter is independent for each such event. In essence, the noise itself is uncorrelated, but the fit residuals likely are correlated. However, from a mathematical perspective the correlated residuals can be treated in the same way as correlated noise, by constructing a non-trivial covariance matrix with off-diagonal terms. In practice, this is achieved by parameterizing C with a kernel that describes

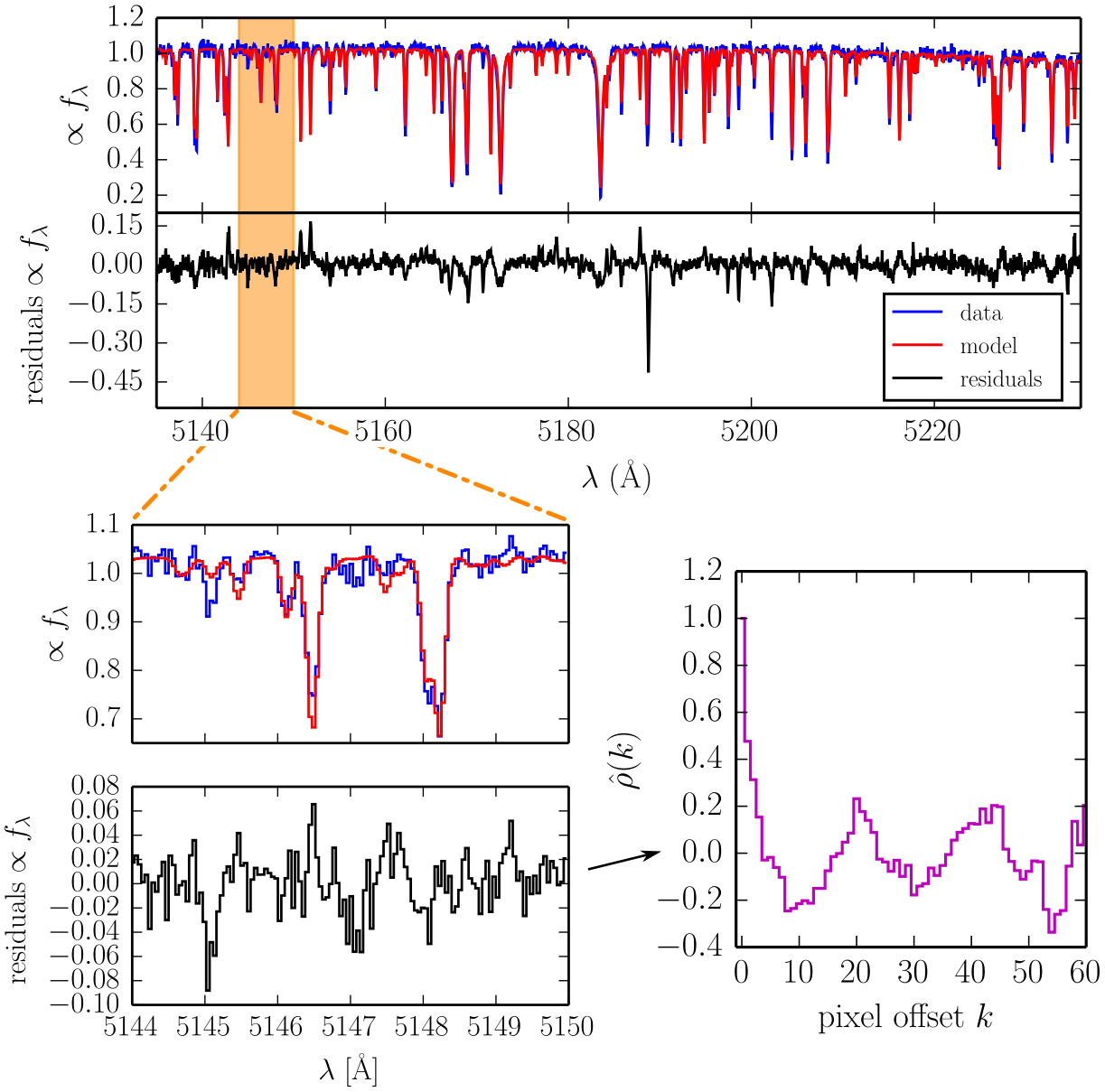

Figure 3. **Top**: the big spectrum. **Left** The same low-amplitude, mildly covariant residuals from Figure 9, panel 0, but enlarged to show the mildly covariant structure produced by slight mismatch between the data and model spectra. **Right** The autocorrelation (Equation **??**) of the residual sequence shown at left. Notice that there is significant correlation for offsets of $\lesssim 4$ pixels.

the covariance between any pair of pixels, representing wavelengths λ_i and λ_j (in many ways analogous to the standard two-point spatial correlation function used in cosmology).

For a well-designed spectrograph and sufficiently accurate model spectrum, this *global* (i.e., present throughout the spectrum) covariant structure should have a relatively low amplitude and small correlation length. To describe that structure, we assume a stationary covariance kernel (also called a radial basis function) with an amplitude that depends only on the velocity distance between two pixels,

$$r_{ij} = r(\lambda_i, \lambda_j) = \Delta v = \frac{c}{2}\left|\frac{\lambda_i - \lambda_j}{\lambda_i + \lambda_j}\right|, \tag{10}$$

where c is the speed of light. The kernel is chosen to parametrically describe the covariance between pixel residuals, such that

$$\mathcal{K}_{\rm g}(\lambda_i, \lambda_j) = \langle \mathsf{R}_i\, \mathsf{R}_j \rangle. \tag{11}$$

A variety of kernels have been used in the field of Gaussian processes to parameterize such a mildly covariant structure (e.g., Rasmussen & Williams 2005). Here we adopt the Matérn kernel with $\nu = 3/2$,

$$\mathcal{K}_{\rm g}(\lambda_i, \lambda_j \,|\, a_{\rm g}, \ell) = a_{\rm g}\left(1 + \frac{\sqrt{3}\, r_{ij}}{\ell}\right)\exp\left(-\frac{\sqrt{3}\, r_{ij}}{\ell}\right), \tag{12}$$

which is parameterized by an amplitude a_g and scale ℓ and makes for a smooth transition to negligible covariance at large r. To ensure that C remains a relatively sparse matrix that enables computational expendiency,

we employ a Hann window function

$$w\left(\lambda_i, \lambda_j | \, r_0\right) = \begin{cases} \frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi r_{ij}}{r_0}\right) & r_{ij} \leq r_0 \\ 0 & r_{ij} > r_0 \end{cases} \tag{13}$$

to taper the kernel (so the effective kernel is $w\,\mathcal{K}_g$). The truncation distance r_0 can be fixed to some reasonable multiple of the scale parameter (here we set $r_0 = 4\ell$). The examples in Figure 4 demonstrate that such a kernel readily produces correlated structure in the residual spectrum that is similar to those seen in a typical data–model comparison.

2.3.2. *Local Covariance Structure*

Aside from the global covariance structure described above, there are likely also local regions of strong, highly correlated residuals that need to be treated in the modeling framework. These large amplitude residual regions are usually produced by imperfect spectral lines in the models (e.g., missing opacity sources, uncertain oscillator strengths, etc.); some representative examples are highlighted in Figure 9. To parameterize such regions in the covariance matrix, we introduce a sequence of nonstationary kernels that explicitly depend on the actual wavelength values of a pair of pixels (on λ_i and λ_j), and not simply the distance between them (r_{ij}).

Assuming that these local residuals are produced primarily by pathological differences in the spectral line strength (rather than shape or center), a simple Gaussian is a reasonable residual model. In that case, the $k^{\rm th}$ such local residual can be described as

$$\mathsf{R}_\lambda(a_k, \mu_k, \sigma_k) = \frac{a_k}{\sqrt{2\pi}\sigma_k} \exp\left[-\frac{r^2(\lambda, \mu_k)}{2\sigma_k^2}\right], \tag{14}$$

with an amplitude a_k, a mean wavelength μ_k, and a width σ_k. Following Eq. 11, the kernel that describes the covariance of any two pixels related to the $k^{\rm th}$ residual region is

$$\mathcal{K}_k(\lambda_i, \lambda_j | a_k, \mu_k, \sigma_k) = \frac{1}{2\pi}\left(\frac{a_k}{\sigma_k}\right)^2 \exp\left[-\frac{r^2(\lambda_i, \mu_k) + r^2(\lambda_j, \mu_k)}{2\sigma_k^2}\right]. \tag{15}$$

We again taper this kernel with a Hann window (Eq. 13) to ensure computational efficiency with a sparse covariance matrix; in this case, the truncation distance r_0 can be set to some multiple of the Gaussian width (we set $r_0 = 4\sigma_k$). Figure 5 demonstrates how this nonstationary Gaussian kernel generates a localized region of enhanced variance that successfully mimics the kind of residuals produced by an inaccurate spectral line model. In effect, these kernels down-weight the influence of such strong residuals in the likelihood calculation, mitigating any potential bias they might induce on inferences of the interesting parameters ($\boldsymbol{\theta}$). In essence, this is a robust, flexible, and unbiased method for (correlated) outlier rejection that preserves the integrity of the probabilistic framework being developed (as opposed to manual or threshold-based clipping/masking).

These local kernels can be further modified to account for more complex residual structures. For example, late-type stars with imperfectly modeled molecular bandheads may produce a complicated pattern of positive and negative residuals or a pronounced mismatch over a relatively large spectral scale. This phenomenologically different local covariance behavior can still be treated in this framework, if we permit the kernel in Eq. 15 to be modified by another (stationary) function. Here we assume that function is a squared exponential,

$$\mathcal{S}_k(\lambda_i, \lambda_j | \, h_k) = \exp\left(-\frac{r_{ij}^2}{2\,h_k^2}\right), \tag{16}$$

where h_k is a bandwidth parameter; small h_k generates high-frequency structure, and vice versa. The functionality of this kernel modification (where the appropriate kernel is now $w_k\,\mathcal{S}_k\,\mathcal{K}_k$) is demonstrated for the worked example presented in Section **??**. SA: I have to admit that I do not really understand this very well. I think some appropriate figures in Sect 3.2 will really help, though, so I'll wait for that before decided whether or not we need to change this part of the text.

2.3.3. *Composite Covariance Matrix*

We can now compute the covariance matrix employed in the likelihood calculation (Eq. 8) as the linear combination of these kernels and the trivial pixel-by-pixel noise matrix,

$$\begin{aligned} \mathsf{C}_{ij}(\boldsymbol{\Theta}_{\rm cov}) = {} & b\,\delta_{ij}\sigma_i + \\ & w(r_{ij}|r_0 = 4\ell)\ \mathcal{K}_g(\lambda_i, \lambda_j | a_g, \ell) + \\ \sum_k & w(r_{ij}|r_0 = 4\sigma_k)\ \mathcal{S}_k(\lambda_i, \lambda_j | h_k)\ \mathcal{K}_k(\lambda_i, \lambda_j | a_k, \mu_k, \sigma_k), \end{aligned} \tag{17}$$

where the covariance hyperparameters $\boldsymbol{\Theta}_{\rm cov} = [a_g, \ell, \{a_k, \mu_k, \sigma_k, h_k\}^{N_{\rm reg}}]$, $N_{\rm reg}$ is the number of local residual regions (see below for details on how this is determined), and b is assumed to be a fixed parameter that scales up the pixel noise values to account for read noise, noise added during the 2D to 1D spectral extraction procedure, and interpolation uncertainties (see Section 2.1; reasonable values are $b \approx 1.02$–1.10 for well-calibrated optical spectra; see Section 3 for examples).

2.4. *Exploring the Posterior*

When iteratively fitting a spectrum (see §2.4), we continue to add line covariance kernels until we have covered all of the high amplitude residuals. Typically, we will add line kernels until all residuals greater than three times the amplitude of the global covariance kernel are covered. In a single order of an echelle spectrum, there may be N regions of high covariance that are parameterized by several line kernels (Equation **??**), which we group into an aggregate parameter $\boldsymbol{\theta}_{\rm lines} = \{\boldsymbol{\theta}_{\rm line_1}, \boldsymbol{\theta}_{\rm line_2}, \ldots, \boldsymbol{\theta}_{\rm line_N}\}$. Along with the global covariance parameters and Chebyshev parameters for this order, we call the collection of nuisance hyperparameters for a specific order (e.g., order 1) $\boldsymbol{\theta}_{\rm order_1} = \{\boldsymbol{\theta}_{\rm Cheb}, \boldsymbol{\theta}_{\rm global}, \boldsymbol{\theta}_{\rm lines}\}$. Taken together, the aggregated nuisance parameters for all of the N orders are stored in $\boldsymbol{\theta}_{\rm orders} = \{\boldsymbol{\theta}_{\rm order_1}, \boldsymbol{\theta}_{\rm order_2}, \ldots, \boldsymbol{\theta}_{\rm order_N}\}$.

We explicitly fit for the hyperparameters of the covariance kernels at the same time we fit for the stellar parameters. While including these extra parameters does increase the dimensionality and complexity of our

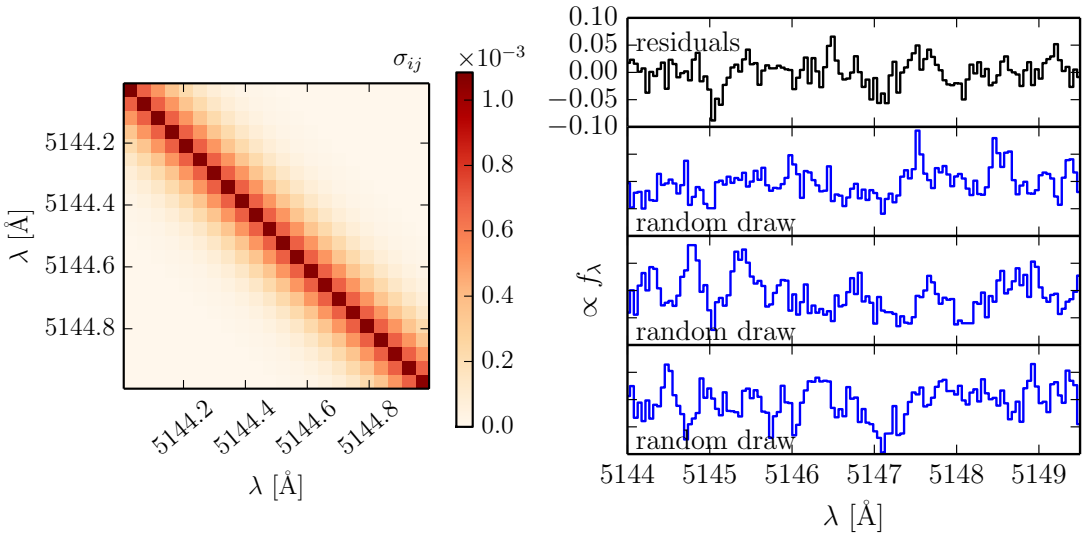

Figure 4. Left Inset zoomed to show a small region of a typical covariance matrix, generated using the kernel in Equation **??** and common values for the hyperparameters. There is a small degree of covariance in elements within a few pixels off the diagonal, which quickly tapers off so that the majority of the matrix remains sparse ($\sigma_{ij} = 0$). This matrix is used to model a spectrum which is correlated on a ∼few pixel scale. **Right** To demonstrate that this matrix properly models the correlated structure of the residuals, we compare the residuals to random residuals generated from a multivariate normal distribution with this covariance matrix. The top panel shows the same residuals shown in Figure 3, and below are plotted three sets of simulated residuals. The amplitude and correlation length of the simulated residuals closely approximates the structure of the actual pixel residuals.

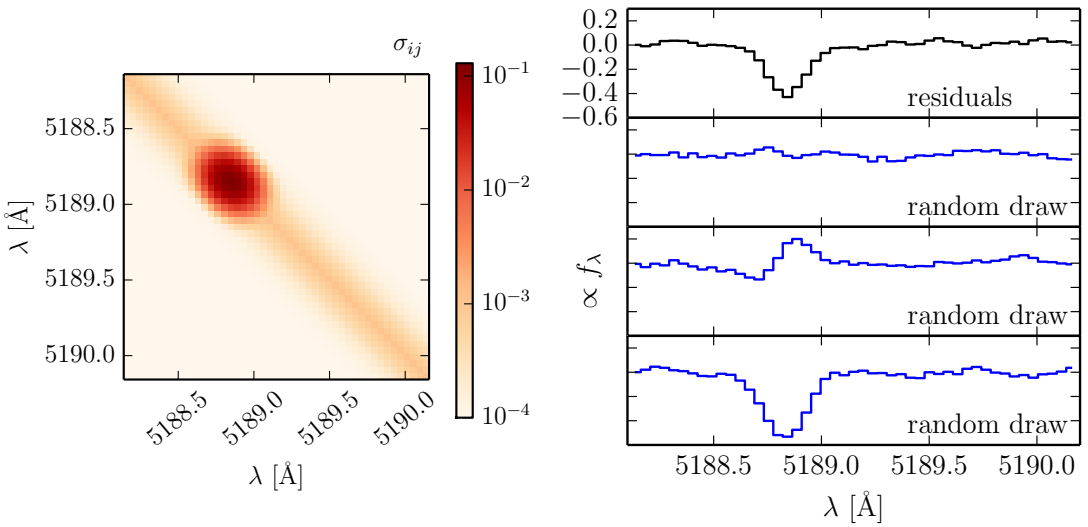

Figure 5. Left A typical covariance matrix including the Gaussian line kernel (Equation **??**). The same global covariance shown in Figure 4 is still present with the same hyperparameters, however now there is an additional patch of high covariance corresponding to the large, Gaussian-shaped residuals. These larger elements in the covariance matrix effectively down-weight the contribution of a poorly modeled synthetic spectral line. **Right** The same spectroscopic residuals shown in Figure 9, class I, shown with three random draws from the covariance matrix. The random draws can take on a range of amplitudes–positive, negative, or even flat–because they are simply random draws that are described by the covariance matrix. The wide range of possible residual amplitudes match the structure and amplitude of the pixel residuals.

model, they do provide several advantages. *A priori* we do not know which regions of the spectrum are improperly modeled. The covariance hyperparameters provide a framework to identify these regions iteratively and in a self-consistent manner.

The traditional practice method of dealing with spectral mismatch is to simply mask out the regions of the spectrum which do not agree to within a certain tolerance. Rather than arbitrarily excluding regions of the spectrum from the fit, these regions should instead be incorporated into in the fit with the appropriate weight. A model that includes covariance is far more likely than forcing the synthetic spectrum to fit perfectly, and far more flexible than arbitrarily masking regions which do not fit. The fitting procedure allows these weights to be determined self-consistently, such that lines which are slightly wrong can still bring information to bear on the stellar parameters. In fact, it may be the case that the

lines that are off by a small amount (class 0 and III lines in Figure 9) actually provide the *most information* about the stellar parameters, precisely because these lines are the most sensitive to stellar structure and consequentially are the most difficult to model correctly.

Another powerful benefit of including the covariance hyperparameters is the result of their ability to quantify and account for model-data mismatch. When a data spectrum is fit with a high-quality spectral library, the pixel residuals are less likely to be correlated, and the covariance structure will have a smaller amplitude and correlation length. This means that the inference on the stellar parameters will be more precise, in the same way that high quality data allows a more precise result. However, if the systematic mismatch between the data and model is large, then the hyperparameters will be larger and the precision of the stellar parameters will naturally inflate to respect the quality of model-data fit.

The benefits of using covariance hyperparameters extend beyond the use case of fitting a single stellar spectrum. If we fit many stars with the same set of synthetic models, we can use the structure of the covariance matrix to improve the models themselves (see §2.5).

We use a Markov chain Monte Carlo (MCMC) algorithm coupled to a Gibbs sampler to explore the posterior distribution (Equation 8). The MCMC routine efficiently explores a high dimensional probability space using a stochastic, iterative approach. The Gibbs sampler provides a way to easily sample a large number of parameters in a simple, organized manner by sampling only a subset of parameters at any given time, but then rotating among all of the subsets of parameters. For more on MCMC and Gibbs samplers see Gelman et al. (2013, ch 11) and the references therein.

In addition to the stellar parameters $\boldsymbol{\theta}_\star$, we have introduced several nuisance parameters for calibration and residual modeling. These nuisance parameters have a logical hierarchical structure. At the lowest level of the hierarchy, a single order of an echelle spectrum has N regions of high covariance $\boldsymbol{\theta}_{\rm lines} = \{\boldsymbol{\theta}_{\rm line_1}, \boldsymbol{\theta}_{\rm line_2}, \ldots, \boldsymbol{\theta}_{\rm line_N}\}$. If we wish to evaluate the probability of the parameters of a specific line (e.g., line 1) conditional on the current values of all other lines, then we denote this by $p(\boldsymbol{\theta}_{\rm line=1}|\boldsymbol{\theta}_{\rm lines\neq 1})$. The collection of nuisance parameters for a specific order (e.g., order 1) is the aggregate parameter $\boldsymbol{\theta}_{\rm order_1} = \{\boldsymbol{\theta}_{\rm Cheb}, \boldsymbol{\theta}_{\rm global}, \boldsymbol{\theta}_{\rm lines}\}$. Taken together, the aggregated nuisance parameters for all of the N orders are stored in $\boldsymbol{\theta}_{\rm orders} = \{\boldsymbol{\theta}_{\rm order_1}, \boldsymbol{\theta}_{\rm order_2}, \ldots, \boldsymbol{\theta}_{\rm order_N}\}$. Because the nuisance parameters for a single echelle order are independent from the nuisance parameters for any other echelle order, we have $p(\boldsymbol{\theta}_{\rm order=1}|\boldsymbol{\theta}_{\rm orders\neq 1}) = p(\boldsymbol{\theta}_{\rm order=1})$.

The Gibbs sampler iteratively explores this hierarchy of parameters. At each level of the hierarchy, corresponding to a different subset of nuisance parameters, we use the Metropolis-Hastings algorithm to propose a new subset of parameters and then either accept or reject them. This process of proposal and acceptance/rejection is called "sampling." The Gibbs sampler rotates between sampling in different subsets of parameters, eventually sampling all of the nuisance parameters at a certain cadence. To initialize the MCMC algorithm, we make a reasonable guess for the starting parameters. We use existing spectral types in the literature for the stellar parameters, flat Chebyshev polynomials (no flux-calibration correction), and no covariance structure in the residuals (the global covariance parameters are set to zero and no line kernels are instantiated). We use superscripts to denote iterations of the MCMC algorithm. i denotes the parameters from the current iteration and $i-1$ denotes the previous iteration. The Gibbs sampler rotates among subsets of the parameters as follows

1. Sample in the stellar parameters. For each proposal of the Metropolis-Hastings algorithm, generate a model spectrum following the steps in §2.2. When deciding whether or not to accept the parameters, the Gibbs sampler evaluates $p(\boldsymbol{\theta}^i_\star|\boldsymbol{\theta}^{i-1}_{\rm orders})$.
2. For each order of the echelle spectrum
   (a) Sample in the Chebyshev polynomial parameters. Adjust the spectrum as detailed in §2.2.Gibbs sampler evaluates $p(\boldsymbol{\theta}^i_{\rm Cheb}|\boldsymbol{\theta}^i_\star, \boldsymbol{\theta}^{i-1}_{\rm global}, \boldsymbol{\theta}^{i-1}_{\rm lines})$.
   (b) Sample in the global covariance parameters. Adjust the covariance matrix C as described in §**??**. Gibbs sampler evaluates $p(\boldsymbol{\theta}^i_{\rm global}|\boldsymbol{\theta}^i_\star, \boldsymbol{\theta}^i_{\rm Cheb}, \boldsymbol{\theta}^{i-1}_{\rm lines})$.
   (c) Check to see whether the algorithm should instantiate/delete new/old line kernels
   (d) For each line kernel k
      i. Sample in the parameters for each line, conditional on the parameters for all the other lines. Adjust the covariance matrix C as described in §**??**. Gibbs sampler evaluates $p(\boldsymbol{\theta}^i_{\rm line_k}|\boldsymbol{\theta}^{i-1}_{\rm lines\neq k})$.

At each stage of the Gibbs sampler, the likelihood function (Equation 8) is evaluated for a proposal of a particular subset of parameters, conditional on the current values of all the other parameters. This hierarchical parameter structure and the kernel parameterization of the covariance matrix influences how the algorithm may be efficiently implemented in code. For a typical optical spectrum with $\gtrsim$1,000 pixels, the most computationally intensive step of the likelihood evaluation is usually the matrix product $\boldsymbol{R}^T C^{-1} \boldsymbol{R}$. Since we have designed the covariance matrix to be sparse, we can use optimized sparse matrix algorithms which are much faster and memory efficient than dense matrix operations. Because we are not interested in the matrix inverse C^{-1} by itself, but rather the product with the residual vectors, we can use efficient routines for solving linear systems to bypass the computationally difficult step of matrix inversion. Additionally, because the covariance matrix is positive semi-definite, we can use the Cholesky factorization of the matrix to optimize the evaluation of the matrix product. Once the covariance matrix is factorized, any subsequent evaluation of the matrix product for different residual vectors $\boldsymbol{R}$ is extremely rapid. This makes the $\boldsymbol{\theta}_\star$ and $\boldsymbol{\theta}_{\rm Cheb}$ steps of the Gibbs sampler extremely fast. When we sample in the nuisance parameters which affect the covariance matrix, we must redo the Cholesky factorization of C for each update. However, because we designed the kernels to deliver a sparse matrix these operations are efficient

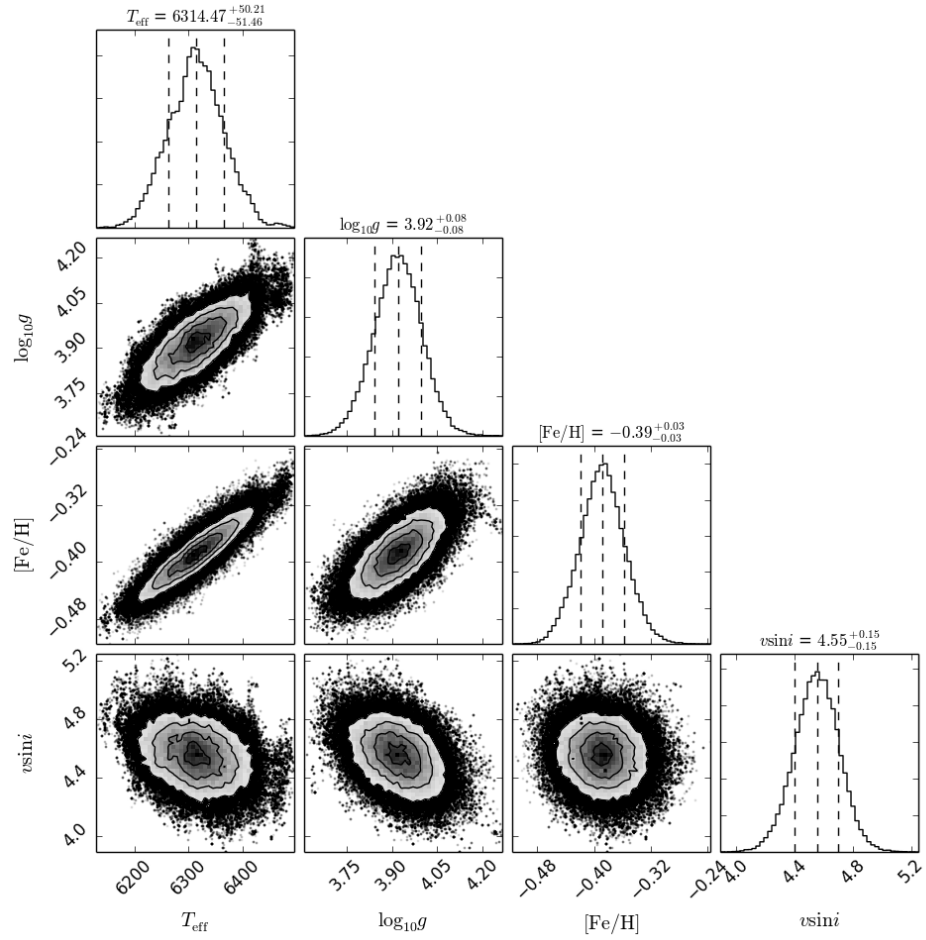

Figure 6. The posterior probability function of the stellar parameters for WASP-14, an F star, as explored by the MCMC Gibbs sampler. These stellar parameters are marginalized over the Chebyshev and noise parameters. Figure generated with `triangle.py` (Foreman-Mackey et al. 2014).

enough to be used with an MCMC algorithm. We use the high-performance `SuiteSparse/CHOLMOD`[1] library to implement the sparse matrix and Cholesky factorization operations (Chen et al. 2008; Davis & Hager 2009) and the Metropolis-Hastings sampler included in the `Python` MCMC package `emcee` (Foreman-Mackey et al. 2012).

We run the MCMC Gibbs sampler for many iterations until the estimate of the posterior distribution has converged. To check that the chain is not stuck in a local maximum of the posterior, we redo the MCMC run many times with different starting parameters, to ensure that the algorithm converges to the same global maximum. A major advantage of using the MCMC algorithm to explore the multidimensional probability space is that it provides numerical samples in each dimension. Therefore, marginalizing out a parameter (i.e., numerically integrating over a dimension in probability space) is as simple as combining all of the samples in this dimension. This enables us to present a posterior of the stellar parameters $\boldsymbol{\theta}_\star$ (Figure 6) which has been marginalized over all of the nuisance hyperparameters. This posterior is the final estimate of the stellar parameters which *incorporates* any inherent uncertainty due to model mismatch (via the covariance hyperparameters) and flux-calibration (via the Chebyshev polynomials).

The benefit of including nuisance parameters and then marginalizing over them is that we can self-consistently model the uncertainty inherent to any spectrum while naturally capturing any degeneracy between the model parameters. If stellar parameters are estimated using a method which ignores these nuisance parameters (e.g., by-eye fitting) and then the astronomer arbitrarily inflates the parameter uncertainties to reflect intuition, the degeneracy between parameters is artificially destroyed. This leads to estimates of stellar parameters with more uncertainty than necessary.

TODO: Since this point is better carried with an example or test using the global covariance structure and not using it (see Test 1), we might want to defer this discussion to the testing section.

2.5. *Applications*

By cataloguing the covariance structure of the residuals, especially those generated from strong spectral line mismatch, we collect valuable information about the quality of the synthetic spectra. After fitting many stars, the accumulated knowledge of the data-model mismatch can be used when fitting a new star. The previous struc-

[1] http://www.cise.ufl.edu/research/sparse/cholmod/

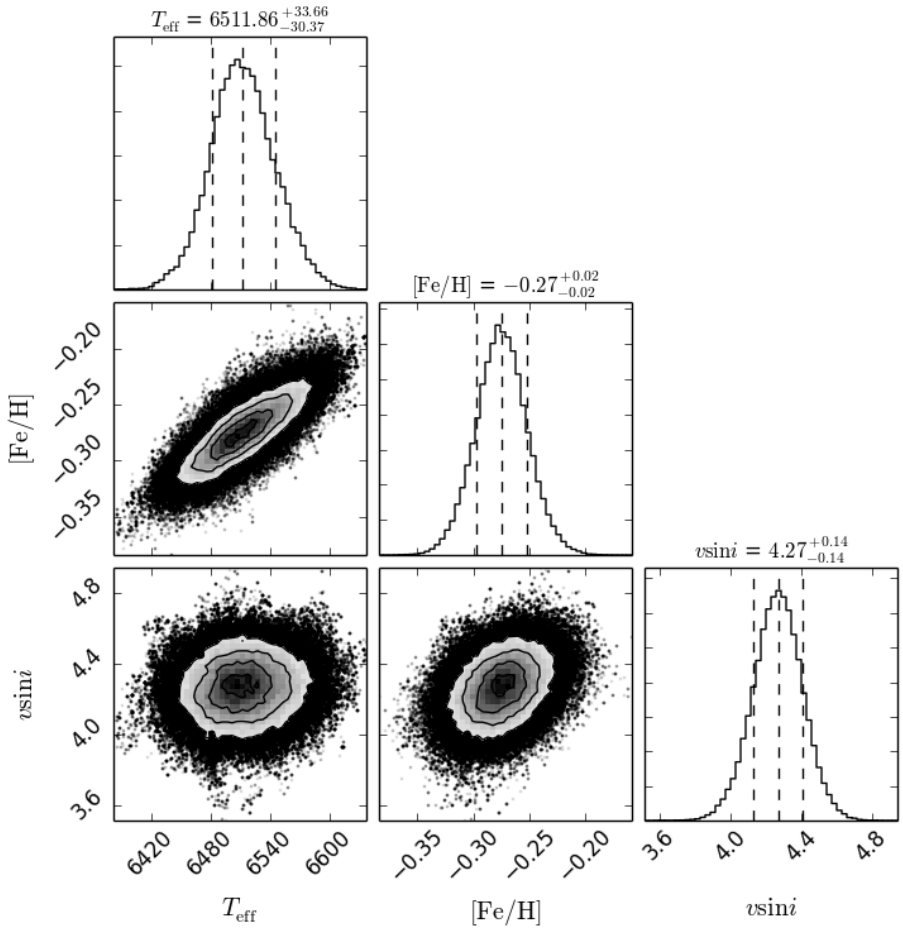

Figure 7. The posterior probability function of the stellar parameters for WASP-14, logg fixed to match (Torres et al. 2012). Using three orders, Kurucz grid. Same assumptions.

figs/global_posteriors.pdf

figs/line_posteriors.pdf

Figure 8. **Left**: The posterior probability function of the global covariance parameters, as explored by the MCMC Gibbs sampler, marginalized over the stellar and line noise parameters. **Right**: The posterior probability function of the line parameters, marginalized over the stellar and global noise parameters.

ture of the covariance matrix allows us to set priors on what the covariance in certain regions of the spectrum

should be, which will speed convergence for this new star. Additionally, after fitting several stars, the average value of the covariance matrix will inform us about the quality of specific spectral lines in the synthetic models.

Linking the covariance matrices of stars could be done serially, where the aggregate covariance matrix of all previous fits is used as a prior for the current iteration. Or, the covariance matrices could instead be linked hierarchically, in that the parameters describing the depth and width of a line residual for a specific star a_1 and σ_1, are modeled as coming from of a population of possible depths and widths for a given spectral type. Each stellar spectrum will have a slightly different realization of a spectral line, which will have some scatter about the average residual height. Linking the covariance matrix between spectra of similar stars allows us to grow more confident in our assessment that certain synthetic spectral lines are indeed outliers and should be appropriately down-weighted. In turn, as we become more certain of the weights, the stellar posteriors will become narrower and make our estimates of the stellar parameters more precise. This ability of the model to mutually inform sets of parameters is one of the major advantages of hierarchical Bayesian analysis (Kruschke 2010).

Once determined, this average covariance matrix could be delivered to the communities that created the synthetic libraries, which would enable them to rapidly pinpoint and correct defects in the synthetic models.
TODO: more delicate phrasing? Alternatively, we could correct the models ourselves by using the chain of logic and mathematical post-processing that we used to created the synthetic model spectrum to reverse-engineer what the behavior of the raw synthetic spectrum *should* be, at the raw $R \gtrsim 100{,}000$ resolution. This fundamental application of machine learning would enable us to create our own library of semi-empirical stellar models. Rather than simply assembling an empirical spectral library using only real stellar spectra, this combined approach is more powerful because the stellar atmospheres provide an actual anchor point of fundamental stellar parameters tied down by the laws of stellar physics.

3. WORKED EXAMPLES

Here are some ideas of which tests we might want to show.

3.1. *Fitting data*

[This may need to be eventually moved to §2, but here it is for now] Cite papers that acquired the data. Describe the instruments. Describe when it was taken. Describe the noise properties, etc. Maybe a table might be worthwhile to describe the spectra. Here also describe the WASP-14 fit using the Kurucz spectra, using a fixed log g, and state that they are the same as Willie's paper. Talk about putting things in a Bayesian framework, using a sensitivity to a prior, to a δ-function prior. Point out that this Bayesian approach allows you to do this.

3.2. *Generic Tests*

Designed to show off what the model framework can actually do.

Talk about giving more reasonable errors using regions, in the biased case.

Test 1— : Fitting with and without the global covariance kernel to show how the width of the posteriors nicely inflates to reasonable uncertainties.

Test 2— : I think we should fit an optical spectrum of an F star (WASP-14) and a near-IR spectrum of an M star (Gl51). The WASP will be fit with both the Kurucz and PHOENIX spectra, while the M star will be fit with just the PHOENIX. This will show a few things

- Spectral parameters can vary by a large margin depending on which spectral library you use (200 K or more).
- Both spectral libraries have stars that they perform better and worse on.
- This will be reflected in the increased level of global noise, and number of "bad" regions that have been instantiated.

Summary end of this section is that we are
1) validating other techniques (WASP14, Kurucz, etc)
2) and two different synthetic libraries give different answers
this can feed into a discussion about tweaking models.

4. DISCUSSION

We can (should?) add an additional level to the hierarchy of hyperparameters and add parameters to describe the *population characteristics* of poorly modeled spectral lines (mostly the typical width, amplitude of these lines). This will tell us about the frequency and distribution of spectral modelling errors.

5. CONCLUSION

REFERENCES

Allard, F., Homeier, D., & Freytag, B. 2012, Royal Society of London Philosophical Transactions Series A, 370, 2765
Buchhave, L. A., Latham, D. W., Johansen, A., et al. 2012, Nature, 486, 375
Cardelli, J. A., Clayton, G. C., & Mathis, J. S. 1989, ApJ, 345, 245
Castelli, F., & Kurucz, R. L. 2004, ArXiv Astrophysics e-prints, astro-ph/0405087
Chen, Y., Davis, T. A., Hager, W. W., & Rajamanickam, S. 2008, ACM Trans. Math. Softw., 35, 22:1
Davis, T. A., & Hager, W. W. 2009, ACM Trans. Math. Softw., 35, 27:1
Eisenstein, D. J., Liebert, J., Harris, H. C., et al. 2006, ApJS, 167, 40
Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2012, ArXiv e-prints, arXiv:1202.3665 [astro-ph.IM]
Foreman-Mackey, D., Price-Whelan, A., Ryan, G., et al. 2014
Gelman, A., Carlin, J., Stern, H., et al. 2013, Bayesian Data Analysis, Third Edition, Chapman & Hall/CRC Texts in Statistical Science (Taylor & Francis)
Gray, D. F. 2008, The Observation and Analysis of Stellar Photospheres
Hauschildt, P. H., Allard, F., & Baron, E. 1999, ApJ, 512, 377
Husser, T.-O. 2012, 3D-Spectroscopy of Dense Stellar Populations (Universitätsverlag Göttingen)
Husser, T.-O., Wende-von Berg, S., Dreizler, S., et al. 2013, A&A, 553, A6
Kane, S. R. 2014, ApJ, 782, 111
Koleva, M., Prugniel, P., Bouchard, A., & Wu, Y. 2009, A&A, 501, 1269
Kruschke, J. 2010, Doing Bayesian Data Analysis: A Tutorial Introduction with R (Academic Press)

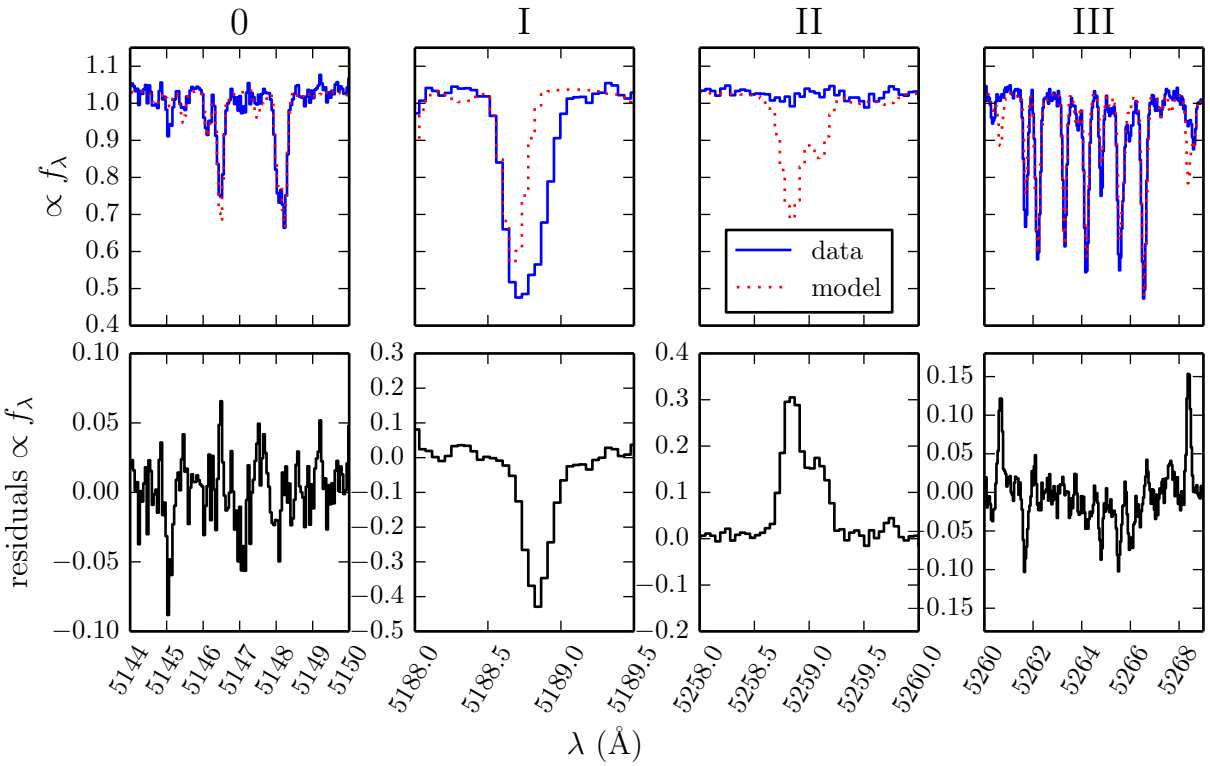

Figure 9. A collection of spectral lines which have imperfect model fits. From left to right: **Class 0** The majority of spectral lines ($\gtrsim 60\%$) will have minor differences in strength between data and model spectrum, which produce low-amplitude correlations in the residuals on the length scale of the width of a typical spectral line. **Class I**: Sometimes ($\lesssim 5\%$ of all lines), a missing opacity source in the model (in this case a line-blended Ca II) leaves a large, highly correlated patch of negative residuals. **Class II**: Sometimes ($\lesssim 5\%$ of all lines), an extraneous line in the model leaves a large, highly correlated patch of positive residuals. **Class III**: If the line strengths are substantially discrepant ($\lesssim 10\%$ of all lines), there will be many correlated residuals of moderate amplitude. The difficulty with class III lines is that for any specific line, there might exist a $\boldsymbol{\theta}_\star$ that will fit the line, but there does not exist a $\boldsymbol{\theta}_\star$ that will properly fit *all* the lines.

Kurucz, R. L. 1993, SYNTHE spectrum synthesis programs and line data

Rasmussen, C. E., & Williams, C. K. I. 2005, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning) (The MIT Press)

Sneden, C. A. 1973, PhD thesis, The University of Texas at Austin.

Torres, G., Fischer, D. A., Sozzetti, A., et al. 2012, ApJ, 757, 161

Valenti, J. A., & Piskunov, N. 1996, A&AS, 118, 595

6. TABLES

APPENDIX

This version of the paper was generated from a git repository available at http://github.com/iancze/StellarSpectra/ with git hash `4ffa13b (2014-07-04)`.

Table 1
Nomenclature used in this document

Symbol	Description
i	index specifying a pixel
λ_i	wavelength corresponding to a given pixel i
$\boldsymbol{\theta}_{\star,\mathrm{grid}}$	fundamental stellar parameters, $T_{\mathrm{eff}}, \log(g), [\mathrm{Fe/H}], [\alpha/\mathrm{Fe}]$ that parameterize a synthetic spectrum from the grid
$\boldsymbol{\theta}_{\star,\mathrm{post}}$	stellar parameters $v \sin i$, v_z, A_V, and R^2/d^2 that are applied during "post processing" of the synthetic spectrum
$\boldsymbol{\theta}_\star$	$\{\boldsymbol{\theta}_{\star,\mathrm{grid}}, \boldsymbol{\theta}_{\star,\mathrm{post}}\}$
$f_{\lambda,\mathrm{inst}}(\lambda)$	data spectrum
$f_{\lambda,\mathrm{synth}}(\lambda)$	synthetic spectrum
$\boldsymbol{\theta}_{\mathrm{Cheb}}$	the set of Chebyshev polynomial coefficients $\{c_0, c_1, \ldots, c_N\}$
$\boldsymbol{\theta}_{\mathrm{line}_1}$	
$\boldsymbol{\theta}_{\mathrm{lines}}$	
$\boldsymbol{\theta}_{\mathrm{order}_1}$	
$\boldsymbol{\theta}_{\mathrm{orders}}$	
$\boldsymbol{\theta}$	the parameters $\{\boldsymbol{\theta}_{\star,\mathrm{grid}}, \boldsymbol{\theta}_{\star,\mathrm{post}}, \boldsymbol{\theta}_{\mathrm{N}}\}$ that completely describe a model spectrum
D_i	data flux for a given pixel, $D(\lambda_i)$
$\vec{\boldsymbol{D}}$	data vector comprised of all D_i, $i = \{1, \ldots, N\}$
M_i	model flux for a given pixel, $M(\lambda_i \vert \boldsymbol{\theta})$
$\vec{\boldsymbol{M}}$	model vector comprised of all M_i, $i = \{1, \ldots, N\}$
σ_i	Poisson noise for a given pixel i
R_i	residuals $D_i - M_i$
$\boldsymbol{R}$	residual vector $\vec{\boldsymbol{D}} - \vec{\boldsymbol{M}}$
C	covariance matrix
σ_{ij}	element in the covariance matrix
$r(\lambda_i, \lambda_j)$	radial distance in wavelength space corresponding to Δv
k_{global}	global covariance kernel
k_{line}	regional covariance kernel

Table 2
Tests

Object	Orders	λ range [Å]	library	$T_{\mathrm{eff}} \pm \sigma$	$\log g \pm \sigma$	$[\mathrm{Fe/H}] \pm \sigma$	comments
WASP-14	22-24	5060-5315	Kurucz	6310±52	3.91±0.08	-0.38±0.03	Poisson only
WASP-14	22-24	5060-5315	Kurucz	6314±50	3.92±0.08	-0.39±0.03	Matern only
WASP-14	22-24	5060-5315	Kurucz	6512±32	···	-0.27±0.02	Matern + fixed $\log g = 4.29$
WASP-14	23		PHOENIX	6021±16	3.83±0.03	-0.50±0.01	Poisson only
WASP-14	23		PHOENIX	5965±70	3.78±0.09	-0.68±0.07	Matern only, no regions
WASP-14	23		PHOENIX	±	±	±	Matern and regions
WASP-14	22-24	5060-5315	PHOENIX	6117±30	3.73±0.06	-0.52±0.02	Poisson only
WASP-14	22-24	5060-5315	PHOENIX	5865±43	3.2±0.08	-0.85±0.03	Matern only. Because of 24.
WASP-14	22-24	5060-5315	PHOENIX	±	···	±	Matern + fixed $\log g = 4.29$
WASP-14	22-24	5060-5315	PHOENIX	±	±	±	Regions included