

# A Method for the spectroscopic inference of fundamental stellar parameters

May 13, 2014

Ian Czekala<sup>1</sup> et al.

iczekala@cfa.harvard.edu

## 1. Introduction

- Where our technique fits in the ecosystem of stellar methods.
- Applicable to any kind of spectrum (long-slit, echelle, infrared, flux-calibrated or not)
- Examples of fields where accurate and unbiased stellar parameters are crucial. Exoplanets. T Tauri stars.

## 2. Method

A single spectrum provides a wealth of information that can be used to infer the fundamental properties of a star: effective temperature  $T_{\text{eff}}$ , surface gravity  $\log g$ , and metallicity  $[\text{Fe}/\text{H}]$ ; together  $\vec{\theta}_{\star, \text{grid}}$ . We wish to extract the maximal amount information from a given spectrum while also recognizing the limitations imposed by our stellar model and exposing any degeneracies between the parameters. For example, if we model a large range of the spectrum at once (e.g., the full optical range at high resolution), then we must explicitly account for any covariances and biases introduced by improper calibration or inaccuracies in the model spectrum. We present a generative model which uses a synthetic stellar library as a backend and apply post-processing techniques to mimic the observed spectrum as a function of several stellar parameters and calibration parameters in order to determine the best fit. By forward modelling the data spectrum in a Bayesian framework, our modular approach can easily incorporate additional calibration artifacts and stellar complexity (e.g., veiling or Zeeman splitting) while still being true to the limitations of the dataset. This data driven approach will enable us to learn how the underlying synthetic models may be improved and refined in an iterative manner.

The roadmap for our method is as follows. First, for a given set of fundamental stellar parameters  $\vec{\theta}_{\star, \text{grid}}$ , we generate a synthetic spectrum by interpolating from a grid of synthetic spectra (§2.1). Then, we post-process the interpolated spectrum for realism by using additional parameters to describe Doppler shift, projected rotational velocity, interstellar extinction, and residual flux calibration errors (§2.2). We perform a pixel-by-pixel comparison of the generated spectrum with the data spectrum using a prescribed likelihood function (§2.3) and a parametric treatment of the covariance between pixel residuals (§2.4). We

---

<sup>1</sup>Harvard-Smithsonian Center for Astrophysics, 60 Garden Street MS 10, Cambridge, MA 02138

use a Markov chain Monte Carlo (MCMC) Gibbs sampler to numerically explore the posterior probability density and determine constraints on the stellar and nuisance parameters (§2.5). The benefit of our method is that it provides robust estimates of the stellar and nuisance parameters with well-quantified uncertainties and parameter degeneracies. Our parameterization of the covariance between pixel residuals also yields a means to identify and quantify the key discrepancies between the data and model, and provides a path forward to improve the stellar models in an iterative fashion (§2.6).

TODO: I really want to overload the word “covariance” and use it to describe both the covariance between  $T_{\text{eff}}$  and  $\log g$  as well as the covariance structure of the pixel residuals themselves. However I think this might be confusing so I am using “degeneracies” for the former and reserving “covariance” for the later.

## 2.1. Synthetic Model Spectra

Many methods exist to generate a synthetic model of a stellar spectrum. Generally, when a high quality spectrum is needed for a specific set of stellar parameters, one runs a stellar atmosphere and radiative transfer code to synthesize the spectrum. Unfortunately, it is computationally expensive to accurately generate a spectrum over a wide range of wavelengths, and so these codes are generally employed to generate libraries of stellar spectra, which exist over a range of  $\vec{\theta}_{\star, \text{grid}}$  at specified points on a regular grid, typically covering the main sequence of stellar types. Several high-quality libraries of synthetic spectra now exist (Husser et al. 2013, Kurucz, and more are being developed for GAIA).

In a forward-modelling approach, which may make many calls to generate a spectra of varying  $\vec{\theta}_{\star, \text{grid}}$ , it is currently computationally impractical to iterate the structure of the stellar atmosphere and then run the stellar synthesis for each call of  $\vec{\theta}_{\star, \text{grid}}$ . Some methods compromise by taking a halfway approach, whereby a precomputed library of stellar atmospheres is interpolated for a given  $\vec{\theta}_{\star, \text{grid}}$ , and then a simple radiative transfer code is run to synthesize a spectrum (e.g. SME; Valenti & Piskunov 1996). This technique has the advantage that interpolating atmospheres and then running the spectral synthesis for each  $\vec{\theta}_{\star, \text{grid}}$  might be more accurate than directly interpolating synthetic spectra. However, if a large region of wavelength is required at high resolution, then the synthesis step will be slow. A different approach might be instead to eschew forward modelling (and thus repeated synthesis or grid interpolation) of the spectra entirely and evaluate the posterior probability density only at the grid points of the spectral library. Then, these discrete samples of the posterior density can be interpolated to find the posterior surface and draw confidence intervals on the parameters (e.g. SPC; Buchhave et al. 2012). However, the difficulty with this approach is that the parameter uncertainty might be smaller than the grid spacing, and then interpolation will not properly capture degeneracies between parameters.

Because we desire a generative model of the spectra without the added complexity or computational overhead of running a stellar synthesis, we opt to use a

TODO: Implement a more accurate but still fast interpolator for the grid. Likely this will be a combination of using splines to pre-interpolate a finer grid ( 20 K, 0.1 dex spacing in gravity and metallicity and then doing linear interpolation off of this new grid during the actual MCMC run. It will be described here. However,

the techniques we develop in this paper to account for and parameterize the correlated, systematic pixel residuals are generally applicable to spectra fit by any of these methods.

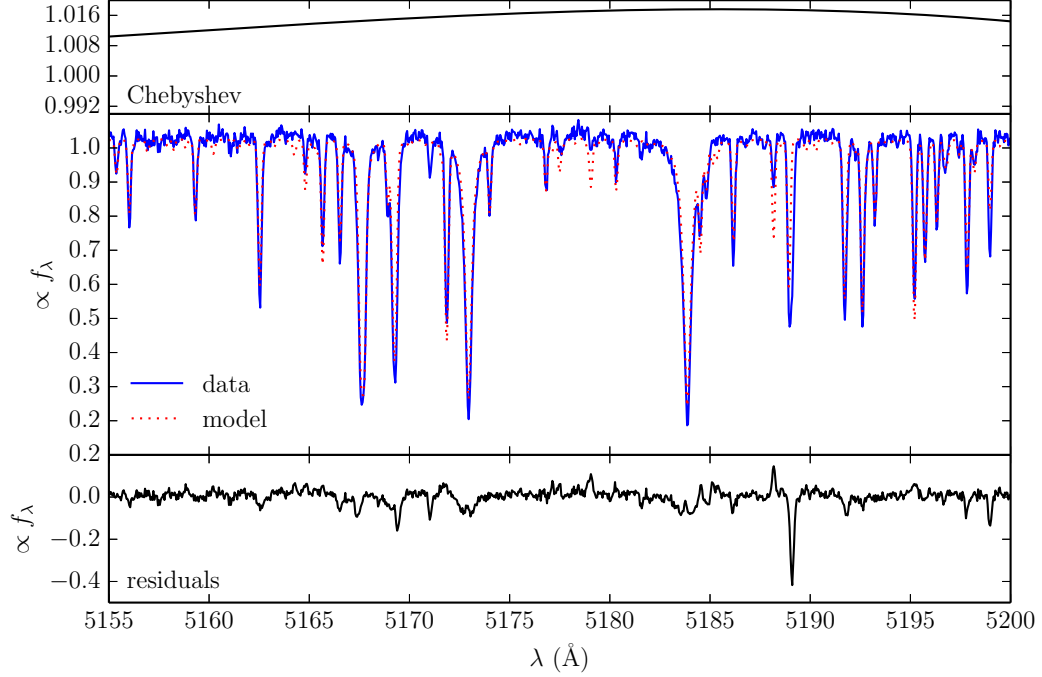


Fig. 1.— **Top:** Chebyshev polynomial which multiplies the model spectrum to account for inaccuracies in the flux-calibration. **Middle:** The data spectrum and model spectrum, after it has been interpolated, post-processed, and multiplied by the Chebyshev polynomial. **Bottom:** Residuals from the model fit. Note the large residual at  $\lambda 5189\text{\AA}$  due to a missing opacity source from Ca I.

## 2.2. Post-Processing

Libraries of synthetic spectra are generally computed at high resolution ( $R \geq 100,000$ ), sampled with many pixels per resolution element, and do not include any rotational broadening or account for any instrumental effects. In order to transform a raw synthetic spectrum interpolated from the grid into one that matches the spectrum of a real star, we must post-process the spectrum to account for these secondary effects. In addition to these fundamental parameters, a star also has several observed properties that are a function of its kinematics, geometry, and location in our galaxy: projected rotational velocity  $v \sin i$ , line of sight velocity  $v_z$ , total extinction  $A_V$ , and solid angle  $\Omega$ . We call these secondary parameters  $\vec{\theta}_{\star, \text{post}}$ , because we can model their effects on the synthetic spectra in a “post-processing” step by convolution with an appropriate kernel or multiplication with a smooth function. Together, we call  $\vec{\theta}_\star = \{\vec{\theta}_{\star, \text{grid}}, \vec{\theta}_{\star, \text{post}}\}$ .

The projected rotational velocity of the star, parameterized by  $v \sin i$ , broadens spectral line profiles and is mathematically described by convolution with a specific kernel (Gray 2008, Equation 18.14). UV, optical, and infrared spectra are acquired using a spectrograph which disperses light onto a CCD, defining a specific resolution and sampling rate for the spectrum. Because the resolution of the spectrograph and pixel sampling are different from the synthetic spectral library, we must convolve the raw spectrum with the line spread function (LSF) of the instrument and resample to the exact pixels of the CCD. In wavelength space, the  $v \sin i$  and LSF operations are represented by the convolution of the synthetic spectrum with these two kernels. Using the convolution theorem, we can rewrite these convolutions as multiplications in Fourier space. In order to use the fast Fourier transform (FFT) with discrete samples of the synthetic spectrum, we must first resample it onto a uniform grid. In the case of a spectrum, the natural uniform grid is one that is equally sampled in velocity space, such there is an equal velocity shift  $\Delta v$  between pixels. This results in a wavelength grid that is linearly sampled in the logarithm of wavelength. Next, we multiply the Fourier transform of the synthetic spectrum by the Fourier transforms of the rotational velocity and line spread function kernels. Lastly, we do an inverse FFT to transform the modified spectrum back to wavelength space, where it is sampled at the wavelength locations of the pixels in the detector. When resampling the spectrum, we are careful to ensure band-limited interpolation using splines in order to prevent introducing non-physical high frequency structure into the spectrum.

We scale the raw synthetic flux (often reported at the stellar surface) to an observed flux by multiplying by the solid angle of the star,  $\Omega = R_\star^2/d^2$ , where  $R_\star$  is the stellar radius and  $d$  is the distance to the star from earth. Then, we correct for a radial velocity difference  $v_z$  between the star and the earth by Doppler shifting the model spectrum. Finally, we correct for interstellar extinction by dividing by a dust extinction model  $A_\lambda$ , which is parameterized by extinction in the  $V$  photometric band,  $A_V$ .

TODO: We can add in convolution with photometric filter profiles to use photometry and provide a better constraint on the luminosity of the star

TODO: We might try implementing the convolution in wavelength space, rather than the FFT. This would allow the LSF to change with wavelength.

**Calibration polynomials** In contrast to a collection of photometric points in a spectral energy distribution, or any ensemble of data for which each data point is acquired by an individual measurement, adjacent wavelength points in a spectrum are measured simultaneously, with the same detector and—more importantly—the same calibration. Typically, spectra are flux calibrated by dividing by a “sensitivity function,” which is determined by comparing an observation of a spectrophotometric standard to a previously calibrated model spectrum. Therefore, any error in the sensitivity function, which is typically a smooth function with wavelength, will create a region of the data spectrum which is systematically offset by a small amount from what the true flux should be. The flux calibration error is then the residual difference in the empirically determined sensitivity function from the what the “true” sensitivity function would be if we had a perfectly calibrated instrument and spectrophotometric standard. Therefore, it is necessary to introduce additional degrees of freedom into our model in order to account for the calibration error and allow the model spec-

trum to properly match the data. By adding the residual calibration as a parameter in our model, we can explore the distribution of possible calibrations (§2.5) and then marginalize the calibration parameters out of the final posterior distribution of the stellar parameters. This means that we can derive estimates of stellar parameters that properly account for the intrinsic uncertainty in the flux calibration.

A traditional way to deal with errors in the flux calibration is to avoid flux-calibration entirely and simply normalize both the data and the spectrum to the continuum level before comparison. While this procedure may work well for hotter stars with well-defined continua, normalizing spectra of cooler stars with large molecular features may introduce artificial features into the spectrum when the pseudo-continuum is incorrectly placed. Instead, we choose to parameterize the residual flux calibration error by a low-order Chebyshev polynomial which is normalized to unity. The coefficients of the polynomial are parameters in the model that will be determined by the fitting algorithm. If we have multiple observations of spectrophotometric standards, then we can characterize the typical flux-calibration error of the spectrograph and put priors on the coefficients that will prevent overfitting of the polynomials. For an example of the Chebyshev polynomial determined for a typical spectrum, see Figure 1. Using priors on the Chebyshev coefficients ensures that real spectral features, such as deep molecular bands in later type stars, are not accidentally treated as calibration artifacts. The overall normalization of the polynomial  $c_0$  is degenerate with the solid angle of the star  $\Omega$ . Therefore we apply an additional constraint and enforce that the mean of the Chebyshev polynomial be equal to 1. If there is only one order in the spectrum, such as with a long-slit spectrum or a single order of an echelle spectrum, then this means setting  $c_0 = 1$ . If we are fitting more than one order of an echelle spectrum then we adopt an independent Chebyshev polynomial for each order. This means that we chose the  $c_0$  of a specific order to anchor to 1, while the  $c_0$  for other orders are allowed to take on perturbations about 1.

This polynomial formalism can in principle be used to model raw (not flux calibrated) spectra by using the Chebyshev polynomial to solve directly for the sensitivity function itself, rather than a residual calibration error as is currently done. When used in this manner, there will be greater uncertainty about the properties of the spectrograph, which will translate into wider priors on the Chebyshev coefficients. In this case, there is more of a chance that the polynomial might destroy broad-scale information present in the spectrum by fitting it out, something that would be avoided with more accurate priors provided by a flux calibration.

### 2.3. Likelihood Calculation

To evaluate which parameters of the model spectrum fit the data spectrum best, we use a likelihood function to compare individual pixels between the two spectra. A likelihood function provides the probability of the data spectrum  $\vec{D}$ , given the model spectrum  $\vec{M}(\vec{\theta})$ , which is parameterized by the stellar and calibration parameters  $\vec{\theta}$ . The data spectrum and model spectrum are both composed of  $N$  pixels, and so  $\vec{D}$  and  $\vec{M}$  represent the collection of these pixels in vector form,  $\vec{D} = \{D_i\}_{i=1}^N$  and  $\vec{M} = \{M_i\}_{i=1}^N$ , respectively.

The vector of residuals is the difference between the data spectrum and the model spectrum

$$R(\vec{\theta}) = \vec{D} - \vec{M}(\vec{\theta}) \quad (1)$$

In the language of probability calculus, the likelihood function is written as  $p(\vec{D}|\vec{M}(\vec{\theta}))$ , or  $p(\vec{D}|\vec{\theta})$  for short. We choose a standard multidimensional Gaussian likelihood function which allows for covariance between the pixel residuals

$$p(\vec{D}|\vec{\theta}) = \frac{1}{\sqrt{(2\pi)^N \det(C)}} \exp\left(-\frac{1}{2} R^T C^{-1} R\right) \quad (2)$$

which penalizes models that yield larger residuals.  $C$  is the covariance matrix which describes the covariance between the pixel residuals. The structure of the covariance matrix is discussed in detail in the next section (§2.4). Because the likelihood function generally exhibits a large dynamic range, it is numerically preferred to use the log-likelihood function

$$\ln[p(\vec{D}|\vec{\theta})] = -\frac{1}{2} R^T C^{-1} R - \frac{1}{2} \ln \det C - \frac{N}{2} \ln 2\pi \quad (3)$$

## 2.4. Parameterizing the covariance structure

The covariance matrix  $C$  describes the measurement noise in each pixel of the spectrum as well as the covariance between pixel residuals. It is fundamental to the likelihood calculation described in §2.3. For most problems, such as fitting a spectral energy distribution to photometric measurements, each data point is acquired in an independent measurement from the other data points, which means that the noise for each data point,  $\sigma_i$ , is also independent from the other data points. This results in a diagonal covariance matrix

$$C = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ 0 & \sigma_1^2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_N^2 \end{bmatrix} \quad (4)$$

with  $\sigma_{ij} = \delta_{ij} \sigma_i$ , where  $\delta_{ij}$  is the Kronecker delta function. When the covariance matrix is diagonal, the likelihood function (Equation 3) reduces to the familiar  $\chi^2$  form of a sum over the square of the residuals, weighted by the inverse variance of each data point

$$\ln[p(\vec{D}|\vec{\theta})] \propto -\frac{1}{2} \chi^2 = -\frac{1}{2} \sum_i \frac{R_i^2}{\sigma_i^2} \quad (5)$$

However, the pixel residuals from a spectroscopic fit will often be correlated and will invalidate Equation 5. These correlations arise because the CCD pixels of a spectrograph do not sample independent wavelengths of the spectrum. Instead, spectrographs are designed so that the LSF is properly sampled by at least a few pixels, meaning that adjacent pixels do not sample completely independent parts of the spectrum. Because of this, when a spectral feature differs between the model and the data it creates a residual which spans

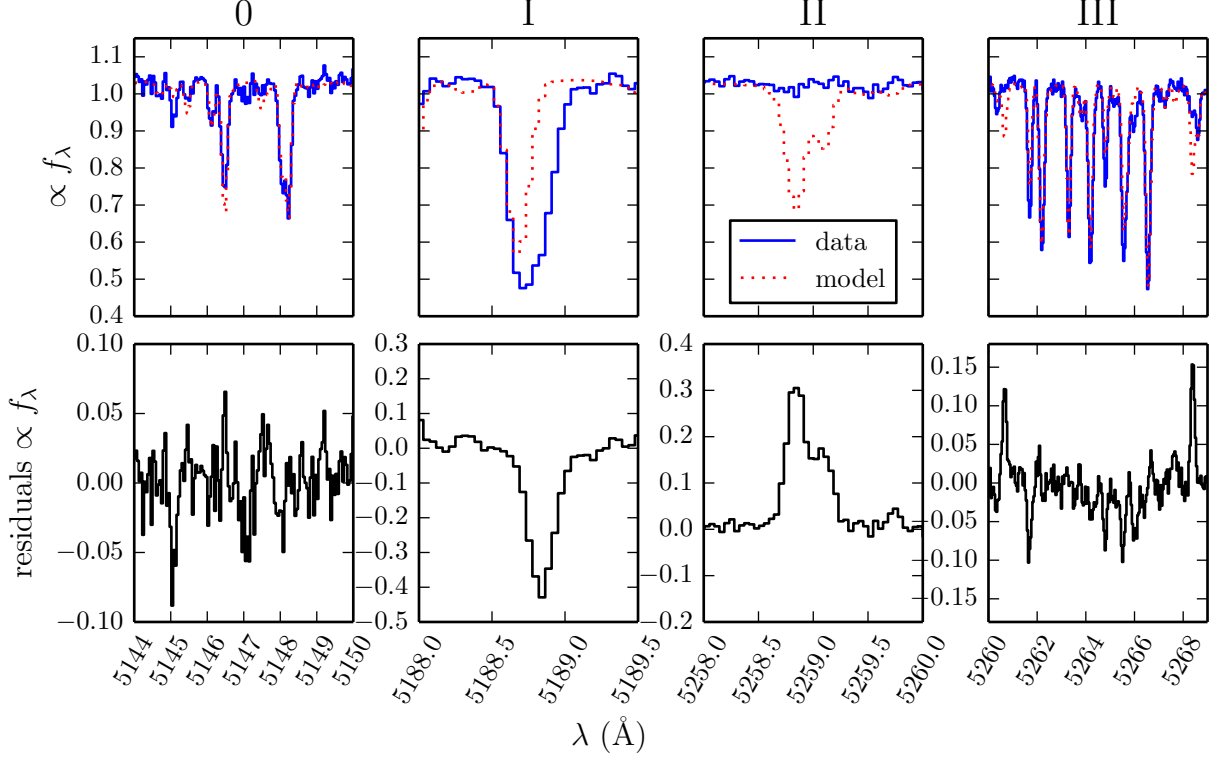


Fig. 2.— A collection of spectral lines induced by imperfect model fits. From left to right: **Class 0** The majority of spectral lines ( $\gtrsim 60\%$ ) will have minor differences in strength between data and model spectrum, which produce low-amplitude correlations in the residuals on the length scale 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 I) 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 residual. **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  $\vec{\theta}_\star$  that will fit the line, but there does not exist a  $\vec{\theta}_\star$  that will properly fit *all* the lines.

several adjacent data points. For a collection of correlated residuals produced by a typical spectroscopic fit, see Figure 2.

There is an important distinction between astrophysical or instrumental “noise” and the amplitude and correlation length of the residuals from a spectroscopic fit. Noise introduced to a spectrum by astrophysical or instrumental effects is generally uncorrelated with wavelength. This is because the arrival of each photon to the detector is a discrete event, and while photons are scattered by the LSF, the particular scattering direction of each photon is independent from any other photon. We can verify this by measuring the auto-correlation of residuals in a region of well-fit stellar continuum, and show that it is flat. However, when we compare two spectra that have many absorption lines, even small differences between the data and model

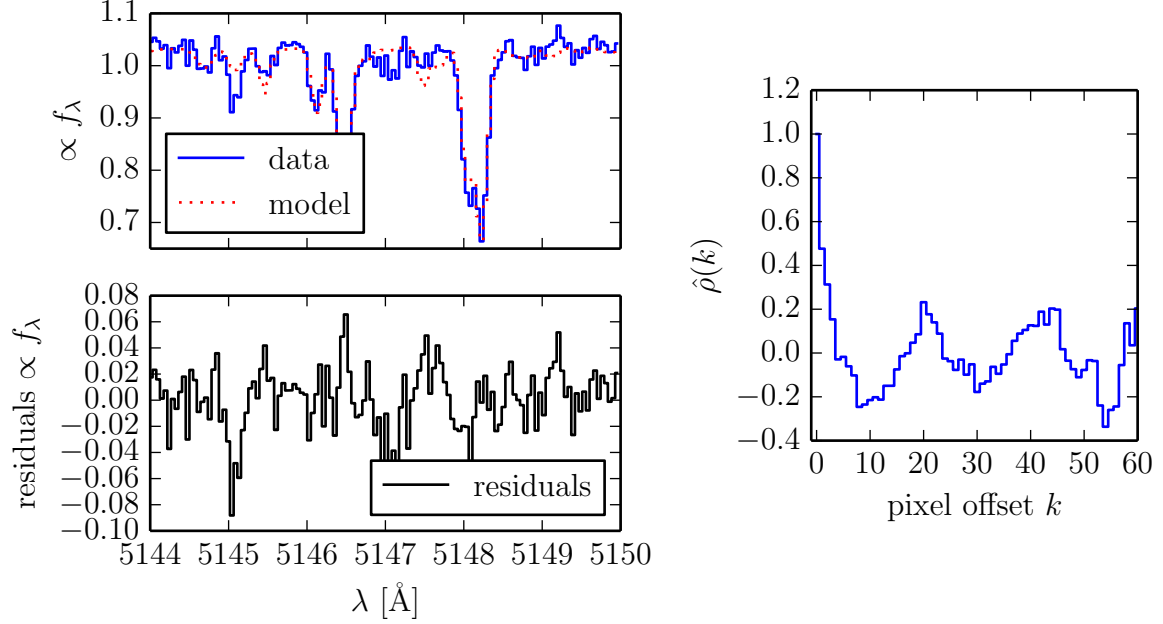


Fig. 3.— **Left** The same residuals from Figure 2, panel 0, but shown at greater zoom to show the mildly covariant structure produced by slight mismatch between the data and model spectra. **Right** The autocorrelation (Equation 6) of the residual sequence shown at left. Notice that there is significant correlation for offsets of  $\sim 3-4$  pixels.

spectra will produce correlated *residuals*. We can verify this correlation by estimating the autocorrelation of the residuals

$$\hat{\rho}(k) = \frac{1}{(n-k)\sigma^2} \sum_{i=1}^{n-k} (R_i - \mu_R)(R_{i+k} - \mu_R) \quad (6)$$

over a region of spectrum of length  $n$ . Figure 3 shows a general residual structure induced by slight model mismatch and the autocorrelation of this series. There is significant correlation on length scales of  $\sim 3-4$  pixels, which is also the typical width of a spectral line for this spectrum.

Although the correlation in the residuals is artificially induced by an imperfect model fit rather than originating from a physical source of correlated noise, we can treat these two effects as mathematically similar. By using a non-trivial covariance matrix with off-diagonal terms we can describe the small-scale covariance between pixel residuals shown in Figures 2 and 3. A non-diagonal covariance also requires that we use a likelihood function (Equation 3) which uses a matrix product instead of a sum over independent pixels.

We seek to understand and parameterize the structure of covariance in the spectroscopic residuals, because ignoring this complexity will bias the estimates of the model parameters  $\vec{\theta}$ . As a simple analogy, consider the fit of a straight line to a dataset. If the noise in the dataset is correlated, then adjacent data points



might be offset from the linear trend in the same direction by a similar amount. If the covariant structure of the noise is ignored, a simple  $\chi^2$  will treat these correlated offsets as part of the linear trend, which will result in a biased determination of the slope and intercept of the line, typically with uncertainties that are too small. We face a mathematically similar problem with correlated pixel residuals from a spectroscopic fit.

We parameterize the covariance structure using covariance kernels, taken from the canon of Gaussian processes. Covariance kernels describe the covariance between two pixels  $\lambda_i$  and  $\lambda_j$ . These kernels are analogous to the cosmological two-point correlation function, where the distance between two galaxies is used instead of wavelength of light.

**Global and mildly covariant structure** To account for the mild covariance structure shown in Figure 3, which is generally low amplitude and has a correlation length of only a few pixels but is present throughout the majority of the spectrum, we use a *stationary* covariance kernel. Stationary covariance kernels, also called *radial basis functions*, satisfy the property that the amount of correlation only depends on the distance between the two pixels  $r$ . For spectra, we map  $r$  to the velocity difference between two pixels

$$r(\lambda_i, \lambda_j) = \Delta v = \frac{c}{2} \left| \frac{\lambda_i - \lambda_j}{\lambda_i + \lambda_j} \right| \quad (7)$$

Then the covariance kernel describes the expected covariance between pixel residuals

$$k(\lambda_i, \lambda_j) = \langle R_i R_j \rangle \quad (8)$$

In theory, we could chose from a variety of covariance kernels to parameterize this mildly covariant structure, as long as they satisfy the property that the degree of covariance smoothly declines with increasing radial distance. In practice, there are a few covariance kernels which are well-respected within the machine-learning community and frequently employed to parameterize covariance of this nature (Rasmussen & Williams 2005). We choose to use the Matérn  $\nu = 3/2$  kernel because it is one of the most commonly used kernels for this purpose and has an extensive history of use (cite?).

$$k_{\nu=3/2}(r, a, l) = a \left( 1 + \frac{\sqrt{3}r}{l} \right) \exp \left( -\frac{\sqrt{3}r}{l} \right) \quad (9)$$

This kernel is an exponential tapered by a polynomial, which makes for a smooth transition from small scale covariance to negligible covariance at longer scales. In order to ensure sparsity of the covariance matrix for computational efficiency, we taper the Matérn kernel for compact support using a Hann window

$$w(r, r_0) = \begin{cases} 0.5 + 0.5 \cos(\pi r / r_0) & r \leq r_0 \\ 0 & r > r_0 \end{cases} \quad (10)$$

for a final covariance kernel of

$$k_{\text{global}}(r, a, l, r_0) = w(r, r_0) k_{\nu=3/2}(r, a, l) \quad (11)$$

We add this covariance kernel to the independent pixel noise  $\delta_{ij}\sigma_{ij}$  already present along the matrix diagonal, and introduce an additional parameter to allow a constant rescaling of the independent noise. Thus, each element of the covariance matrix is completely described by

$$k(\lambda_i, \lambda_j | r, a, l, r_0, b) = k_{\text{global}}(r, a, l, r_0) + b \delta_{ij} \sigma_{ij} \quad (12)$$

Therefore, the *hyperparameters* of this covariance kernel directly dictate the structure of the covariance matrix. By varying the hyperparameters, we can create a covariance matrix which accurately mimics the correlated structure of the pixel residuals. Most importantly, when the hyperparameters are correctly chosen to properly model this correlated structure, the *likelihood function will be maximized* (Equation 2) while encapsulating the *true uncertainty in the model-data fit*.

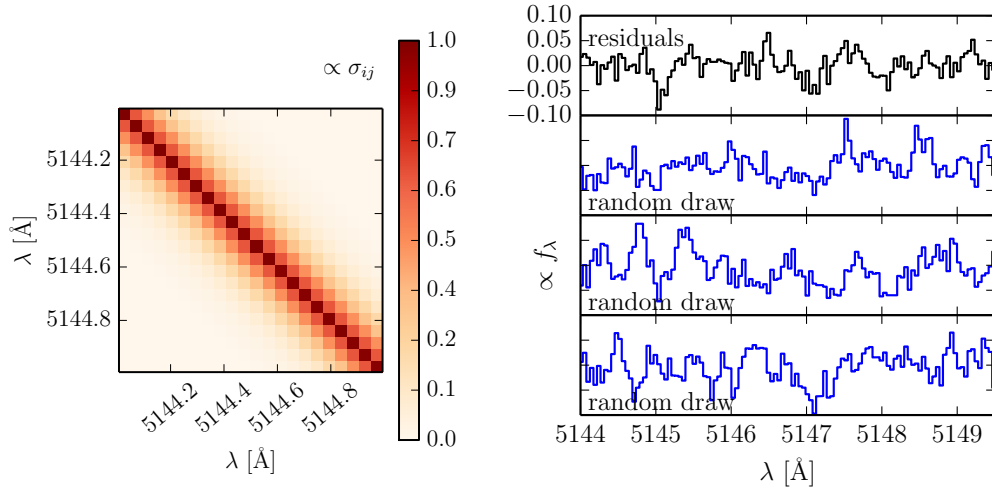


Fig. 4.— **Left** Inset zoomed to show a small region of a typical covariance matrix, generated using the kernel in Equation 11. There is a small degree of covariance in elements within a few pixels off the diagonal, which quickly tapers off such that the majority of the matrix remains sparse ( $\sigma_{ij} = 0$ ). This matrix can be used to model spectrum which is correlated on a  $\sim$ 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. Top: the same residuals shown in Figure 3, plotted against three sets of simulated residuals. The amplitude and correlation length of the simulated residuals closely approximates the structure of the actual pixel residuals.

The left panel of Figure 4 shows what the covariance matrix might look like for reasonable choices of the kernel hyperparameters. To verify that this covariance matrix indeed mimics the correlated structure seen in the residuals, we can draw a random vector of residuals  $R_{\text{rand}}$  from our noise model and compare it to the actual pixel residuals. Because we are using a multidimensional Gaussian as our likelihood function (Equation 2), we have assumed that the underlying residual structure in a spectroscopic fit is described by a multidimensional Gaussian. This is directly analogous to the implicit assumption of independent Gaussian noise when using a standard  $\chi^2$  likelihood (Equation 5). If one wished to compare the residuals that resulted

from a  $\chi^2$  fit to the noise in the data set, then one could simulate  $R_{\text{rand}}$  by drawing  $N$  random samples from a univariate normal distribution with zero mean and variance  $\sigma_i^2$ . We can make the same test with our multidimensional Gaussian likelihood function, however in this case the mean is a vector  $\vec{\mu} = 0$  with length  $N$  and the variance is described by the covariance matrix  $C$ . The residual vector is then a random draw from this multivariate normal distribution

$$R_{\text{rand}} \sim \mathcal{N}(\vec{\mu} = 0, C) \quad (13)$$

Functions to generate random samples from a multidimensional Gaussian are implemented in most numerical programming languages.

**Specific line covariance** Even after the mildly covariant structure has been accounted for by the global covariance kernel (Equation 11), there still exist a significant number large amplitude pixel residuals that are *highly correlated*. These patches of large amplitude residuals are caused by incorrect lines like those in the class I, II, and III examples in Figure 2. To parameterize the covariance in a given patch of these residuals, we introduce an additional *non-stationary* covariance kernel. In this context, non-stationary means that the degree of covariance explicitly depends on the values of  $\lambda_i$  and  $\lambda_j$ , and not simply the distance between them, therefore we can now individually target the large residuals induced by the mismatch of a *specific* spectral line.

Because the shape of a spectral line can be generally approximated by a Gaussian function, and the difference of two Gaussians with the same mean is also another Gaussian—the same shape of residuals that are shown in Figure 2—we use Equation 8 to design a new covariance kernel about the Gaussian function. If we expect the shape of the residuals to be a Gaussian function about the line center  $\lambda_\mu$

$$R(\lambda|a, \lambda_\mu, \sigma) = \frac{a}{\sqrt{2\pi}\sigma} \exp\left(-\frac{r^2(\lambda, \lambda_\mu)}{2\sigma^2}\right) \quad (14)$$

then the Gaussian covariance kernel is

$$k_G(\lambda_i, \lambda_j|a, \lambda_\mu, \sigma) = \langle R(\lambda_i) R(\lambda_j) \rangle \quad (15)$$

$$= \frac{a^2}{2\pi\sigma} \exp\left(-\frac{r^2(\lambda_i, \lambda_\mu) + r^2(\lambda_j, \lambda_\mu)}{2\sigma^2}\right) \quad (16)$$

We again enforce compact support by tapering the kernel with a Hann window (Equation 10), and so the covariance kernel for the highly correlated residuals induced by a poor line fit becomes

$$k_{\text{line}}(\lambda_i, \lambda_j|a, \lambda_\mu, \sigma, r, r_0) = w(r, r_0) k_G(\lambda_i, \lambda_j|a, \lambda_\mu, \sigma) \quad (17)$$

**TODO:** Similar paragraph about visualizing the covariance matrix, drawing samples. References to Figure 5.

non-stationary kernel localizes increased variance to a specific spectral line

better than just masking out the region, since these contain info

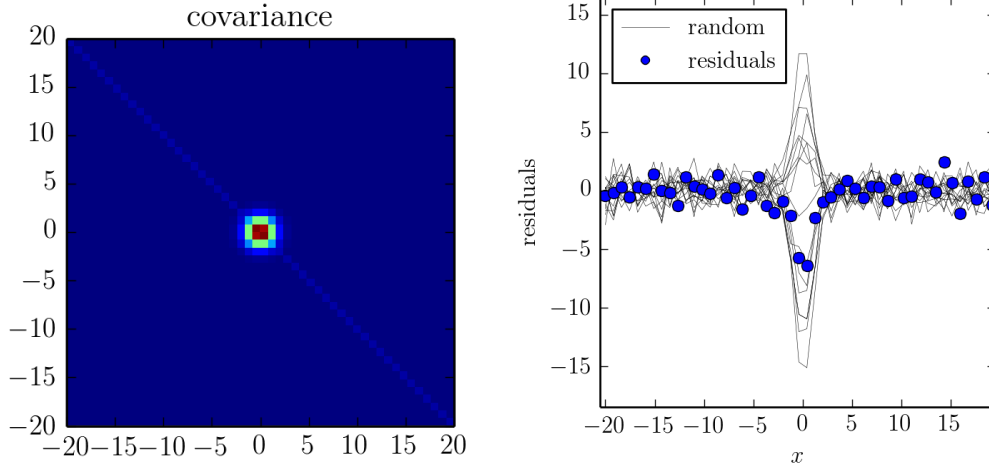


Fig. 5.— **Left** a covariance matrix generated with the region kernel. **Right** general spectroscopic residuals near a mismatched stellar line overlaid with a sample draw using the covariance matrix, showing that the two look similar in structure and amplitude.

**TODO:** Similar improvements to Figure 4.

here will be many sets of regions ( $\sim 1\text{-}4\%$  of all lines might be covered by a region)

Covariance kernel parameters are parameters in our model that we explicitly sample for. There can be hyperparameters describing the *population* of regions in a hierarchical model (mostly width, amplitude).

The benefit mainly comes from simply modelling these systematic residuals to begin with, since a model with covariance is far more likely than forcing the fit, and far more justified than masking regions which do not fit.

Where to draw the line between specific region and global covariance? When we reach some threshold, say three times the global amplitude?

**M dwarf residual** The squared exponential kernel is

**TODO:** Why do we use the squared exponential and not Matérn?

$$k_{\text{exp}}(r, h) = \exp\left(\frac{-r^2}{2h^2}\right) \quad (18)$$

$h$  is a bandwidth. If  $h$  is small, then there will be high-frequency structure. If  $h$  is large, then only low-frequency structure will remain. We also use the Hann window to taper the kernel for compact support

$$k(\lambda_i, \lambda_j | h, a, \mu, \sigma, r_0) = w(r, r_0) k_{\text{exp}}(r, h) k_G(\lambda_i, \lambda_j | a, \mu, \sigma) \quad (19)$$

TODO: Show how the kernel can accounts for complicated M dwarf structure where there is large, messy mismatch.

## 2.5. Exploring the posterior

Markov Chain Monte Carlo.

Gibbs sampler. Sparse matrix. Implemented using SuiteSparse and the Cholesky factorization, means that the RCR step can be computed quickly and reused.

Show figure using dfm/triangle.py that shows recovered stellar parameters and degeneracies. Separately for the regions.

## 2.6. Applications

Learnt covariance structure can be used to correct the models.

## 3. Tests

TODO: Which tests to show?

## REFERENCES

- Buchhave, L. A., Latham, D. W., Johansen, A., et al. 2012, *Nature*, 486, 375
- Gray, D. F. 2008, *The Observation and Analysis of Stellar Photospheres*
- Husser, T.-O., Wende-von Berg, S., Dreizler, S., et al. 2013, *A&A*, 553, A6
- Rasmussen, C. E., & Williams, C. K. I. 2005, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (The MIT Press)
- Valenti, J. A., & Piskunov, N. 1996, *A&AS*, 118, 595

This version of the paper was generated from a git repository available at <http://github.com/iancze/StellarSpectra/> with git hash b958931 (2014-05-07).

Symbol	Description
$i$	index specifying a pixel
$\lambda_i$	wavelength corresponding to a given pixel $i$
$\vec{\theta}_{\star,\text{grid}}$	fundamental stellar parameters, $T_{\text{eff}}, \log(g), [\text{Fe}/\text{H}], [\alpha/\text{Fe}]$ that parameterize a synthetic spectrum from the grid
$\vec{\theta}_{\star,\text{post}}$	stellar parameters $v \sin i, v_z, A_V$ , and $R^2/d^2$ that are applied during “post processing” of the synthetic spectrum
$\vec{\theta}_{\star}$	$\{\vec{\theta}_{\star,\text{grid}}, \vec{\theta}_{\star,\text{post}}\}$
$f_{\lambda,\text{inst}}(\lambda)$	data spectrum
$f_{\lambda,\text{synth}}(\lambda)$	synthetic spectrum
$c_0, c_1, \dots$	Chebyshev polynomial coefficients for residual flux calibration
$\vec{\theta}_{\text{N}}$	the set of nuisance parameters composed of $\{c_0, c_1, \dots, c_N\}$
$\vec{\theta}$	the parameters $\{\vec{\theta}_{\star,\text{grid}}, \vec{\theta}_{\star,\text{post}}, \vec{\theta}_{\text{N}}\}$ that completely describe a model spectrum
$D_i$	data flux for a given pixel, $D(\lambda_i)$
$\vec{D}$	data vector comprised of all $D_i, i = \{0, \dots, N\}$
$M_i$	model flux for a given pixel, $M(\lambda_i \vec{\theta})$
$\vec{M}$	model vector comprised of all $M_i, i = \{0, \dots, N\}$
$\sigma_i$	Poisson noise for a given pixel $i$
$R_i$	residuals $D_i - M_i$
$R$	residual vector $\vec{D} - \vec{M}$
$C$	covariance matrix
$k_{\text{global}}$	global covariance kernel
$k_{\text{region}}$	regional covariance kernel

Table 1: Nomenclature used in this document.