A method for inference of fundamental stellar parameters using high resolution spectra

January 19, 2014

Ian Czekala[1] et al.

`iczekala@cfa.harvard.edu`

ABSTRACT

Matching observed spectra of stars against a grid synthetic spectra is a popular approach to determine fundamental stellar parameters such as effective temperature, surface gravity, and metallicity. However, a fundamental problem with this approach is that the synthetic spectra contain systematic errors such as wrong or missing spectral lines. In the presence of such errors, a simple χ^2 fit between spectra will likely yield biased parameters and incorrect uncertainty estimates. This behaviour is analogous to the way a few strong outliers might severely bias the fit to a straight line. These spectral outliers are difficult to identify *a priori*; procedures such as sigma clipping might be prone to their own systematic effects and discard useful information. We outline an approach to probabilistically identify and account for spectral features that deviate from reality. Such an approach is most needed when operating at high spectral resolution ($R > 10,000$), since this is when systematic deviations begin to appear most striking, however our technique is still valid for lower resolution spectra as well. We also outline some approaches for spectral fits to brown dwarfs and planets, where instead of spectral lines there may be systematic deviations of a larger swath of the spectrum.

1.   Introduction and review of existing methods

The fundamental parameters of stars—mass, luminosity, effective photospheric temperature, surface gravity, and metallicity–are of supreme interest to astrophysicists for a wide variety of reasons. A salient example is that most newly discovered exoplanets must

---

[1]Harvard-Smithsonian Center for Astrophysics, 60 Garden Street MS 10, Cambridge, MA 02138

necessarily their mass and radius measured relative to the properties of their host stars (e.g. Torres et al. 2012; Buchhave et al. 2012; Santos et al. 2013). The properties of stars are of course also important in their own right because they inform us about our ideas of stellar evolution (D'Antona & Mazzitelli 1997; Baraffe et al. 2002). Because stars are the fundamental units of the cosmos, techniques to infer the properties of stars date back to the dawn of astronomy. We briefly review some of the modern and popular techniques in the literature. Approaches can be broadly classified into techniques that compare stars by some broadly defined indices and techniques that attempt to forward-model an observed stellar spectrum by doing a pixel-to-pixel comparison.

1.1. Classification techniques

Many of existing techniques were written to analyze low to medium resolution ($R <$ 10,000) spectra. I am calling them similarity techniques because they focus on determining and recognizing patterns of spectra useful in classification.

Line indexes By cataloguing the equivalent widths of chosen spectral lines into indexes, one can correlate the index with stellar properties and identify features that are sensitive temperature, surface gravity, or metallicity. Then, one matches the indecies of a new star against these relationships and derives a set of stellar parameters. A recent example using M dwarfs is Newton et al. (2014). This technique works, but there are many things that could be improved. First, the selection of which lines to measure is arduous and based upon expert knowledge of previously catalogued stars. This is both a disadvantage and an advantage. On one hand, by choosing a subset of lines to fit, one may be unwittingly throwing away useful information. On the other hand, it allows the modeller to discard regions of the spectrum that do not follow a simple trend with stellar parameters. Second, while measuring equivalent widths is easy to do, this discards any useful information contained in the shape of the line profile.

Principal component analysis PCA reduces the dimensionality of the data space and searches for the most salient features of the spectra (Paletou et al. 2014). How does one properly estimate parameter uncertainties in such an approach? It seems like this approach exists because it is an easy thing to do.

Matrix inversions: MATISSE (Recio-Blanco et al. 2006). This method is similar to PCA in that it derives basis vectors that describe the fundamental parameters of the

star. In the PCA analysis, the principal components do not necessarily correspond to true physical quantities. The MATISSE algorithm determines weights that ensure the components are similar to the stellar parameters. They only test the algorithm against input synthetic stellar spectra to detect biases in the parameters, and against the Sun and Arcturus. They do acknowledge that there will be systematic deviations between the synthetic spectra and real spectra, but they do not account for the possible inflation of error estimates.

Given the inherent systematic uncertainties involved in each method, some groups elect to combine the inference from different algorithms, such as (SLOAN SSPP) (Lee et al. 2008). This approach includes machine learning along with many of the previously mentioned techniques.

Machine learning Many machine learning techniques exist and classify spectra by comparing patterns between observed spectra and a training set of spectra (which could be real spectra of stars classified by an expert or a synthetic library that has known $\vec{\theta}_{\star,\mathrm{grid}}$). However, my first impression is that these tools are designed to handle large collections of spectra from stellar surveys such as SDSS (Sánchez Almeida et al. 2010). In our immediate project involving T Tauri stars, we only have a few ($\lesssim 20$) stars that we wish to study in exquisite detail, incorporating specific parameters such as $v \sin i$, extinction, and veiling. However, I am a novice when it comes to machine learning so I may be missing something here.

1.2. Pixel to pixel techniques using synthetic spectral libraries

Many high quality synthetic spectral grids are now available. Nearly all synthetic stellar libraries are parameterized by the "fundamental" stellar parameters of effective temperature, surface gravity, metallicity, and alpha enhancement. We define this combination as

$$\vec{\theta}_{\star,\mathrm{grid}} = \{T_{\mathrm{eff}}, \log(g), [\mathrm{Fe/H}], [\alpha/\mathrm{Fe}]\} \tag{1}$$

Naively, it makes sense to compare an observed spectrum against spectra from the synthetic grid. Using some goodness of fit statistic one can identify the $\vec{\theta}_{\star,\mathrm{grid}}$ that best reproduce the observed spectrum.

The pixel-to-pixel likelihood function will evaluate the residuals in some form of

$$\mathrm{residual} = \frac{\mathrm{data} - \mathrm{model}}{\mathrm{noise}} \tag{2}$$

where the best model is defined as one that minimizes the residuals in some sense. We will discuss other goodness of fit functions soon. Each of these quantities are functions of wavelength λ, so moving forward we will refer to them as the data spectrum $f_D(\lambda)$, the model spectrum $f_M(\lambda|\vec{\theta})$, the noise spectrum $\sigma(\lambda)$, and the residual spectrum $R(\lambda|\vec{\theta})$.

Most pixel-to-pixel techniques use a χ^2 likelihood function and seek to minimize the square of the residuals

$$\chi^2 = \sum_\lambda \left[\frac{f_D(\lambda) - f_M(\lambda|\vec{\theta}_\star)}{\sigma(\lambda)} \right]^2 \tag{3}$$

It appears that for many techniques, $\sigma(\lambda)$ is the Poisson photon counting errors. For some data reduction pipelines, a per-pixel "sigma spectrum" that accounts for spectral extraction errors due to night sky line contamination or low signal to noise is available, and should be used instead. For more discussion of the noise sources beyond statistical Poisson errors, see §4.

ULySS ULySS (Koleva et al. 2009) is a mature package to make comparisons for individual spectra as well as unresolved stellar populations. Any spectral library can be used as a backend, including empirical spectral libraries such as the one from ELODIE (Prugniel et al. 2007). They explicitly take into account some of the necessary procedures when fitting spectra, such as continuum-multiplication by a polynomial to account for flux calibration errors and rejection of bad data or model points. They use an iterative kappa-sigma clipping method to discard "bad" points for the fit. They test that their clipping technique converges to the same answer by starting guesses located at different parameters $\vec{\theta}$ and seeing if similar lines are masked and similar best fit parameters $\vec{\theta}$ are obtained.

Mann et al. (2013) fit BT-Settl PHOENIX models to M dwarfs. Because high quality, high resolution M dwarf spectra are difficult to obtain owing to their intrinsic faintness, the development of stellar libraries for later spectral types has typically lagged behind. They use a χ^2-like likelihood function and iterate to mask bad pixels. Unlike ULySS, however, they iterate the method against a sample of ~ 20 M dwarfs and track which pixels consistently have the high residuals (median residual amount $> 10\%$, or ten times their measurement error) across all stars. Presumably, such a consistent residual would be due to an incorrect physical constant for a certain line or molecular feature. After masking, they repeat the fitting and claim that after two iterations the masks do not change. They note that in the future, rather than selecting good and bad pixels with a binary mask, "a more nuanced weighting scheme would be to weight each interval (pixel)

according to how consistent it is with the models, or test different weighting schemes to see which gives the best agreement with the bolometric temperatures. **However regions with modest agreement between the real and synthetic spectra may contain more temperature information than regions with slightly better matches, and we have no a priori information about what spectral regions are the most temperature sensitive.**" Thus, masking "bad regions" could very well destroy stellar information in the pursuit of a better "fit." This is a nuanced point that masking schemes generally neglect. We aim to address later in this document.

"Bayesian" methods do exist. Shkedy et al. (2007) developed a hierarchical Bayesian approach to compare infrared spectra at low resolution ($R \sim 1,200$) to a synthetic grid in order to determine stellar parameters $\vec{\theta}_{\star,\mathrm{grid}}$. Their model is hierarchical in that it has parameters describing the statistical and systematic uncertainties of the observed spectrum. The systematic errors are due to the imperfect performance of the instrument, for example a bad flux calibration. They use an "empirical" Bayesian model for the noise, meaning that they use data to estimate the prior parameters for the noise (Gelman et al. 2004). They use a χ^2 likelihood function and allow for "shrinkage" of their hierarchical parameters if there is good agreement between observed and synthetic spectra. Disappointingly, they do explicitly account for systematic errors in the synthesis of actual lines, for example due to wrong atomic constants. One could imagine having a systematic error parameter which describes the scatter in the relationship between $f_{\mathrm{M}}(\lambda|\vec{\theta}_{\star,\mathrm{grid}})$ and the "true" physical spectrum in the absence of noise f_{D}. Nor do they account for any systematic effect that likely exists between $\vec{\theta}_{\star,\mathrm{grid}}$ and the true physical stellar parameters (as far as T_{eff} has any physical significance) and is a function of which synthetic grid they use[1]. However, they do account for interpolation errors by testing interpolating spectra in the pixel space and atmospheric interpolations. Then, they select the best model using squared error loss function and a penalty measure for model complexity.

Schoenrich & Bergemann (2013) present a comprehensive Bayesian method that can fit low or high resolution spectra in addition to photometry and parallax data. They too, mention that they use a χ^2 likelihood. They mention using special masks for the spectra that block noisy and uninformative regions of the spectrum, however they do not go into further detail about how these masks were created or applied. Their unpublished stellar library focuses on parts of the optical spectrum which contain the most stellar information,

---

[1]For example, in our tests with Kurucz vs. PHOENIX spectra we often find a systematic offset in temperature of $\gtrsim 200$ K.

as determined by experts. As noted by Schoenrich & Bergemann (2013), the grids "sample the wavelength windows around the spectral features important for diagnostic of FGKM stars: 3850 - 4050Å (Ca I lines), 4350 - 4450Å (G-band, CN sensitive), 4600 - 4900Å (Hβ), 5100 - 5300Å (Mg I triplet, main gravity diagnostics), 6400 - 6640Å (Hα), 8400 - 8800Å (Ca II triplet, also used in *Gaia* and in RAVE stellar survey)."

SPC   A workhorse method behind the Kepler follow-up program is Stellar Parameter Classification (SPC) (Buchhave et al. 2012)). This compares high resolution optical echelle spectra ($R \sim$ 45,000) to high resolution synthetic grids computed with Kurucz models by a cross-correlation method. The best fit parameters and uncertainties are calculated by the relative values of the cross-correlation coefficient. It is uncertain how statistically justified this uncertainty estimate is. Currently SPC uses only a small region of the optical spectrum ($5000 - 5300$Å) where the spectral library has been carefully tweaked[2] to match high resolution models of the Sun and Vega.

Other methods   Doppmann & Jaffe (2003) do not use a Bayesian model, but they do χ^2 fitting to line profiles in the infrared. Bailer-Jones (2010) use a forward modelling approach that interpolates a synthetic grid to estimate stellar parameters from low resolution Gaia spectra.

Comparison of synthetic stellar libraries   Gaia team members are in a mild panic about the upcoming flood of data. Martins & Coelho (2007) compare many spectral grids to determine their weaknesses. (Sordo et al. 2011) discuss the range of stellar grids available for Gaia. (Coelho et al. 2005) is the Coehlo grid. Munari et al 2005, (Husser et al. 2013).

General trends   Successful pixel-to-pixel methods either mask bad spectral lines or limit themselves to a narrow region of the spectrum, where they can be relatively certain that the spectral library does not contain gross systematical errors (and effectively masking the rest of the spectrum). Cherry picking regions of the spectrum like this evokes similar concerns to those of Mann et al. (2013): how do you know that you are fitting the region(s) of the spectrum that deliver the right balance of information about the stellar properties?

---

[2]changes to line centers and $\log(gf)$ values

1.3. Line fitting using spectral synthesis

Direct synthesis methods take a step closer towards stellar parameter estimation in the correct domain. For as large a region of the spectrum as is computationally tractable, these codes synthesize a spectrum for direct comparison with the data. Because this is an expensive operation, one must necessarily choose only a few good lines to fit.

MOOG[3] (Sneden 1973) and **SME**[4] (Valenti & Piskunov 1996) are two popular synthesis codes. Given a set of stellar parameters, elemental abundances, and oscillator strengths, the program will interpolate a model atmosphere and then synthesize a spectrum (over a small region of wavelength). Then, they use a Levenberg-Marquartd algorithm to converge to the best-fit parameters. Their approach has the benefit of enabling individual tweaks to elemental abundances and oscillator strengths, something that is impossible in the aforementioned approaches.

In order to make the direct synthesis approach computationally tractable, one must use the simplest atmospheric models and radiative transfer. Powerful codes like PHOENIX (Husser et al. 2013) *should* have more accurate atmospheres and physics, but they take a long time to run ($\sim$ hours) per model, and the code is not publicly available.

1.4. Stellar parameter estimation in a utopia

The difficulties encountered by the similarity techniques and synthetic grid comparisons are a result of fitting in the wrong parameter space. Yes, the combination of $\vec{\theta}_{\star,\mathrm{grid}}$ is an excellent proxy to describe a star, but in truth (and at high resolution) each star is slightly different. The "model" that is actually producing the synthetic spectrum has many, many parameters that have been collapsed down into $\vec{\theta}_{\star,\mathrm{grid}}$. These parameters include the atmospheric structure (relations of temperature and pressure as a function of stellar depth), individual elemental abundances (all models quote a value relative to solar ([Fe/H]) and some synthetic grids include alpha enhancements ([α/Fe])), and atomic constants describing the opacity contribution of a line. *All* of these parameters have uncertainties in them, some of them quite dramatic (opacities and line lists are the frequently cited), yet for a synthetic grid these parameters are *fixed*. There may still be some missing physics that would be needed to produce a perfect spectrum within the noise (say for

---

[3]http://www.as.utexas.edu/~chris/moog.html

[4]http://www.stsci.edu/~valenti/sme.html

example the cores of Ca lines, or dust settling at the latest spectral types), but my naive impression is that the field of stellar spectral synthesis is mature enough that nearly all lines could be properly reproduced given sufficient scrutiny to the atmosphere and atomic constants.

If we had infinite computational power, a tantalizing approach would be to use a model with stellar parameters $\vec{\theta}_{\star,\text{grid}}$, atmospheric parameters (such as pressure and temperature profiles), and uncertain atomic constants such as oscillator strengths. One could fit for individual elemental abundances and marginalize over the atomic constants. This would extract a tremendous amount of information from a high resolution spectrum. Unfortunately, the relationship between $\vec{\theta}$ and the spectrum $f_{\text{D}}(\lambda|\vec{\theta})$ is non-linear, meaning that a new spectrum would need to be synthesized for each new $\vec{\theta}$. It may be possible to make an approximate a linear relationship between say an oscillator strength and the line depth as long as the tweak is minor. If no reasonable approximation can be found, then the radiative transfer becomes an insurmountable bottleneck.

If the approximations could reduce the computation of synthetic spectrum, efficient MCMC sampling parallelized on a cluster might put this utopian vision within reach in the near future. However, without a collaborator with expert knowledge of spectral synthesis, I do not have the expertise to take such an approach. Many high quality synthetic spectral grids do exist, and so our current efforts are focused on addressing the shortfalls of the previously mentioned pixel-to-pixel methods.

2. General problems with the spectral library approach

Forward modelling approaches (pixel-to-pixel) have the potential to be more powerful than other stellar parameter estimation techniques. When we model directly to the data space, we have the ability to account for instrumental and systematic effects, allowing us to derive a robust estimate of our uncertainty by marginalizing over the nuisance parameters. We can also derive more complicated models to fit unresolved binary or multiple star systems. In fact, many of the most interesting systems for which we might want to determine stellar parameters are in multiple systems since we can dynamically determine masses for these systems and thus compare to evolutionary models.

As discussed in §1.2, many forward modelling approaches exist, yet nearly all use a χ^2 likelihood function and most employ some sort of masking scheme. The more successful pixel-to-pixel methods use low or moderate resolution spectra ($R \sim 1{,}000$ to $10{,}000$) or a grid over a narrow range of wavelength space with finely tuned atomic constants, eschewing

difficult decisions about what to mask. **The true fundamental problem with a pixel-to-pixel comparison is that the stellar spectra are not a perfect representation of reality**. This may be an obvious statement, but systematic errors in the synthetic spectra means that the previously valid assumption of Poisson errors and a χ^2 likelihood function breaks down. In order to derive unbiased parameter estimates, systematic errors must be identified and properly handled.
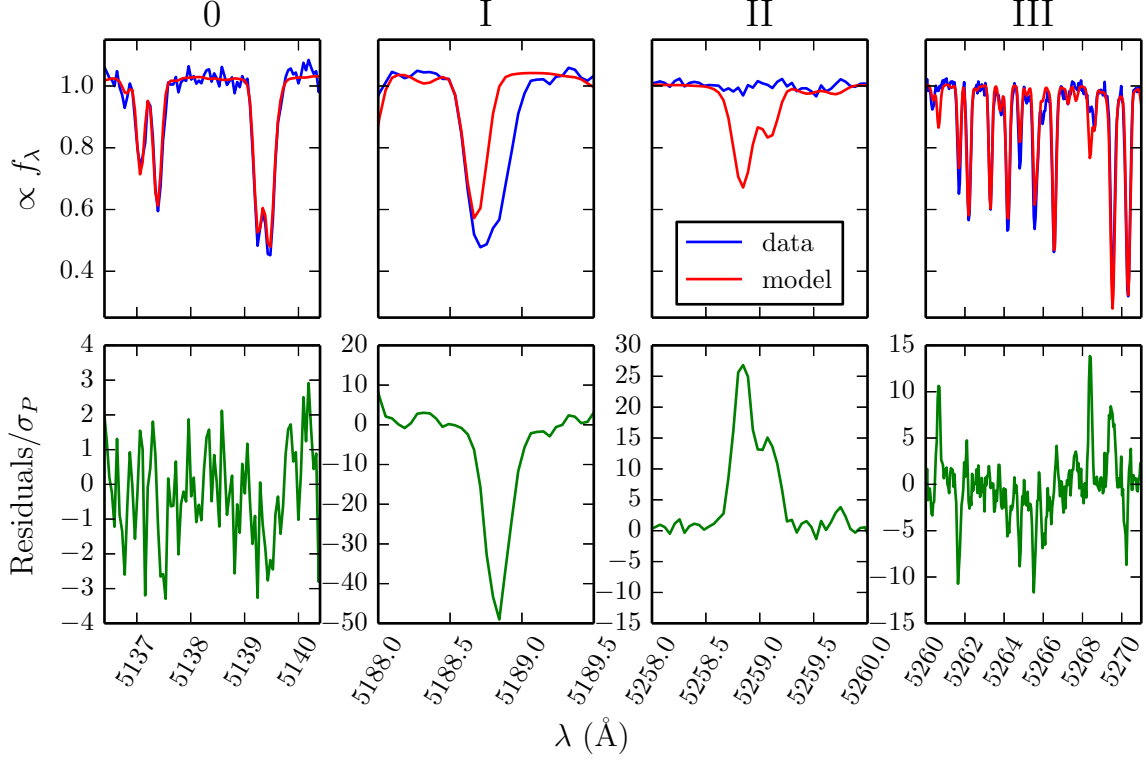

Fig. 1.— The four classes of line behavior, from left to right: **Class 0**: the model behaves as expected. **Class I**: a spectral line is present in the data but missing from the model (in this case, the line is also blended). **Class II**: a spectral line exists in the model where there is only continuum in the data. **Class III**: lines exist in both the data and model, however they are of the wrong strength. Some lines are too strong while others are too weak. The lower panels show the residual spectra in units of the observational uncertainty (Poisson uncertainty only).

2.1. Classes of synthetic spectral line behavior

From our forward modelling experiments using the modified Kurucz grid used in Buchhave et al. (2012) and the recent PHOENIX grid (Husser et al. 2013), we have identified ways that synthetic spectra diverge from the our observed data. To start, we will describe these classes of behaviour as being tied to specific spectral lines present against a well-defined continuum. Later on, we will use the classes to describe features more generally to cover the behavior of molecular features in later type spectra. See Figure 1 for visual examples of each class of behavior.

Class 0 These lines actually produce a good fit with a residual nearly consistent with the Poisson counting uncertainty (see §3 for how the data is processed). If every region of the spectrum behaved like this, we might be justified in using a χ^2 likelihood function and calling it a day.

Class I A spectral line appears in the data but not in the model spectrum over a wide range of probable stellar parameters (1000 K in $T_{\rm eff}$, and 1 dex in $\log(g)$, [Fe/H], and [α/Fe]. Such a type of error is perhaps the easiest to understand and to forgive. For example, such a line could easily appear in the data due to interstellar absorption or telluric absorption, something which a stellar model could never produce. The model could also fail to reproduce a true stellar line due to a missing opacity source.

Class II A spectral line appears in the model but not in the data over a wide range of probable stellar parameters (1000 K in $T_{\rm eff}$, and 1 dex in $\log(g)$, [Fe/H], and [α/Fe]. This is a more puzzling error, because presumably atomic constants would be much more likely to be *missing* than added in by accident. Presumably this might be a combination of incorrect atomic data coupled with incorrect atmospheric structure that incorrectly excites a line.

The traditional way to account for Class I and Class II errors is to sigma clip the bad lines in the spectrum (Koleva et al. 2009; Mann et al. 2013). A region of the spectrum with Class I or II errors is useless for determining spectral parameters. While sigma-clipping routines are to be avoided, if they can correctly mask these regions, then at least they are moving in the right direction.

There is additional information available to help determine which regions of the spectrum suffer from Class I or II errors.For a normal star with absorption lines against a

continuum, Class I residuals will always be negative and not change significantly with $\vec{\theta}_{\star,\mathrm{grid}}$, since we the model can only produce continuum in that region. Class II residuals will always be positive and their magnitude should decrease with decreasing metallicity, since lower metallicity will decrease the strength of all lines. We could easily store the partial derivatives of an individual synthetic spectrum with respect to $\vec{\theta}_{\star,\mathrm{grid}}$

$$\left\{\frac{\partial f_{\mathrm{M}}(\lambda|\vec{\theta}_{\star,\mathrm{grid}})}{\partial T_{\mathrm{eff}}}, \frac{\partial f_{\mathrm{M}}(\lambda|\vec{\theta}_{\star,\mathrm{grid}})}{\partial \log(g)}, \frac{\partial f_{\mathrm{M}}(\lambda|\vec{\theta}_{\star,\mathrm{grid}})}{\partial [\mathrm{Fe/H}]}, \frac{\partial f_{\mathrm{M}}(\lambda|\vec{\theta}_{\star,\mathrm{grid}})}{\partial [\alpha/\mathrm{Fe}]}\right\} \tag{4}$$

and coupled with a prior on the range of parameters over which the residuals will be correlated, we could shore up the classification of Class I and II errors.

Class III errors occur when the line strengths between the data and model do not match at a $\vec{\theta}_{\star,\mathrm{grid}}$ that fits a majority of the other lines. What makes Class III errors truly pernicious is that for a specific line, there does exist a set of $\vec{\theta}_{\star,\mathrm{grid}}$ that could properly fit the line, though this parameter combination would result in bad fits for the majority of the other lines. Class I and II errors are easy to identify since the lines will always be bad across a wide range of $\vec{\theta}_{\star,\mathrm{grid}}$ and have predictable behaviors with change in $\vec{\theta}_{\star,\mathrm{grid}}$. Class III errors, however, come in both directions: the line can be too strong or too weak. Identifying Class III errors requires making a choice about which lines to trust and how much.

Most of the Class III errors can probably be identified by finding the regions of the spectrum that have the highest residuals after dealing with Class I and II errors, although some mildly incorrect Class III lines will be difficult to distinguish from correct (Class 0) lines that are simply fit with a less probable $\vec{\theta}_{\star,\mathrm{grid}}$. Once identified, properly handling Class III errors requires careful thought because these regions of the spectrum *still contain information about* $\vec{\theta}_{\star,\mathrm{grid}}$. If we take an masking approach as with Class I and II errors, we will be throwing it away. It might even be the case that Class III regions contain *the most* information about $\vec{\theta}_{\star,\mathrm{grid}}$, since these "delicate regions" are the most sensitive to changes in $\vec{\theta}_{\star,\mathrm{grid}}$ at that particular $\vec{\theta}_{\star,\mathrm{grid}}$ and thus are tricky to tune correctly. In other words, Class III regions might also be regions where the spectrum derivative (Eqn 4) is largest.

To summarize, the fundamental problem with comparing high resolution stellar spectra to grids of synthetic stellar spectra is that no one model $\vec{\theta}_{\star,\mathrm{grid}}$ will fit all of the lines properly. One can either limit their fits to regions of the spectrum that they know their spectral libraries to be accurate (implicitly masking the rest and potentially discarding a large amount of information) or suck it up and forward model the regions where the models perform poorly.

3. Instrumental effects and "nuisance" parameters in the limit of a perfect stellar model

While this document aims to be general in scope, any forward-modelling approach must also discuss the characteristics of the instrument that performed the observations. We are currently modelling data obtained with the *TRES*[5] echelle spectrograph, and so we will describe the characteristics of the instrument and the calibration approaches we have taken, although we note that our approach is easily generalized for data obtained with other spectrographs.

TRES is a 51-order echelle spectrograph on the 1.5-m Tillinghast reflector at Whipple Observatory. It spans the full optical range $(3800 - 9000\text{Å})$ at a resolution of $R = 48,000$ (6.8 km/s). Echelle spectrographs impart a strong blaze function upon each order, where transmission is largest in the center of the order and declines rapidly as one moves towards the edge of the order. The *TRES* pipeline extracts orders such that they span a wavelength region until the transmission reaches $\lesssim 25\%$ of the central transmission in that order. Each order spans $\sim 100\text{Å}$ and the blue orders overlap while there are gaps between the wavelength coverage of the red orders. This spectra are blaze-corrected by dividing by a blaze function determined from fitting nightly exposures of flat lamps. Then, the spectrum is flux calibrated using sensitivity functions determined from nightly observations of spectrophotometric standards (O stars). The flux-calibrated spectrum is in units of ergs/s/cm^2/Å.

Processing TRES Data Because the *TRES* reduction pipeline does not produce a "sigma spectrum" like that determined by a standard IRAF reduction, we instead calculate the Poisson uncertainty on each pixel. We can find the signal-to-noise ratio of each pixel using the raw (still-blazed) spectrum, which is in units of "counts."

$$[S/N](\lambda) = \sqrt{\text{cts}(\lambda)} \tag{5}$$

the Noise-to-Signal ratio is the inverse

$$[N/S](\lambda) = \frac{1}{\sqrt{\text{cts}(\lambda)}} \tag{6}$$

To determine the uncertainty of the flux-calibrated spectrum, we multiply by N/S. The $[N/S](\lambda)$ "spectrum" is larger at the edges of each order due to the reduced transmission.

---

[5]http://tdc-www.harvard.edu/instruments/tres/

Please note that statistic pixel uncertainties derived in this manner neglect regions of the spectrum that might have a higher variance due to sitting atop a night sky line, but short of implementing our own reduction pipeline for *TRES*, there isn't much more we can do at the moment.

Generating a model spectrum   We parameterize the fit to a star by $\vec{\theta} = \{\vec{\theta}_\star, \vec{\theta}_N\}$, where stellar parameters include $\vec{\theta}_\star = \{T_{\mathrm{eff}}, \log(g), [\mathrm{Fe/H}], [\alpha/\mathrm{Fe}], v\sin i, v_z, A_v, R_\star^2/d^2\}$. The data vector $\vec{D} = f_D(\lambda)$ represents the *TRES* spectrum. To generate a model spectrum given a set of parameters $f_M(\vec{\theta})$, a model spectrum is created by first linearly interpolating in $T_{\mathrm{eff}}, \log(g), [\mathrm{Fe/H}], [\alpha/\mathrm{Fe}]$ space between grid points of instrumentally broadened and over-sampled synthetic stellar spectra. Then, the spectrum is rotationally broadened for $v\sin i$ using Fourier techniques, Doppler shifted by v_z, extinction corrected for A_V, multiplied by $R_\star^2/d^2$ for a solid angle correction, and finally downsampled to the exact pixels of the TRES spectrum using spline interpolation[6].

FIGURE: showing the high resolution, raw $R = 500{,}000$ spectrum compared to the one convolved with a kernel down to $R = 50{,}000$. This shows that lines are blended, but it is important to keep in mind that we do have access to the high resolution models when talking about bad pixels or bad line-species. For example, we could inject bad lines in this high resolution space.

3.1.   Applied to a Gaussian likelihood function

For the rest of §3, we proceed as if there were no Class I, II, or III errors present in our model so that we may illustrate our technique for dealing with uncertainties in the flux-calibration. Although per-pixel photon counting errors are Poisson, we can safely assume that we have enough counts that we are in the Gaussian limit, although see the discussion in §4. Then our likelihood function becomes a pixel-by-pixel χ^2 comparison between the data spectrum f_D, and the model spectrum f_M, summed over the wavelength axis.

$$p(\vec{D}|\vec{\theta}) = \mathcal{L} \propto \exp\left(-\frac{\chi^2}{2}\right) \tag{7}$$

---

[6]Because we are dealing with flux-calibrated data and synthetic spectra, we *resample* to lower resolution. We do not *rebin* because we are dealing with flux-density, not photon counts

$$\chi^2 = \sum_\lambda \left[\frac{f_{\mathrm{D}}(\lambda) - k(\lambda|\vec{\theta}_{\mathrm{N}}) f_{\mathrm{M}}(\lambda|\vec{\theta}_\star)}{\sigma(\lambda)} \right]^2 \tag{8}$$

With flat priors on $\vec{\theta}_\star$ and $\vec{\theta}_{\mathrm{N}}$ (an assumption we will later relax) the logarithm of our posterior probability function is then

$$\ln\left[p(\vec{\theta}|\vec{D})\right] \propto -\frac{1}{2} \sum_\lambda \left[\frac{f_{\mathrm{D}}(\lambda) - k(\lambda|\vec{\theta}_{\mathrm{N}}) f_{\mathrm{M}}(\lambda|\vec{\theta}_\star)}{\sigma(\lambda)} \right]^2 \tag{9}$$

The prefactor $k(\lambda|\vec{\theta}_{\mathrm{N}})$ is a calibration function that aims to account for errors in the blaze-correction or flux-calibration by multiplying the synthetic spectrum by a polynomial. The parameters describing this function are the coefficients of the polynomial. This approach of accounting for large-scale calibration errors in the spectrum is more justified than the traditional approach of normalizing to the stellar continuum. From a flux-calibration experiment where we observed spectrophotometric standards BD+28 and BD+25 multiple times, we calibrated the dispersion in (FIGURE showing the sensitivity function for a few representative orders) flux-calibration function, or IRAF's "sensitivity function." These tests revealed that the largest source of error in flux-calibration was actually in the overall magnitude of individual order, this could be due to slit losses, poor correction due to parallactic angle, or scattered light within the instrument. These corrections are smaller than 10%. But more importantly, the corrections in the higher order terms (ie, the shape of the bandpass calibration for each order) are small ($\lesssim 3\%$). As described in §3.3, we can set priors on these coefficients. This prior information on the calibration parameters motivates our choice of functional parameterization for the nuisance parameters in §3.2. The polynomial approach is also better than the normalization approach if the order only spans a small region of the spectrum. If a M star absorption band spanned the whole order, one might accidentally normalize to the feature rather than a pseudo-continuum.

3.2. Nuisance parameters

In our case, $\vec{\theta}_{\mathrm{N}} = \{c_0, c_1, \ldots, c_N\}$ are a set of 4 Chebyshev polynomial coefficients for each order. This amounts to a "re-fluxing" of the model following the techniques of Eisenstein et al. (2006). We apply the re-fluxing to the model rather than attempt to re-flux the data since this operation would also need to re-scale the noise, making χ^2 a complicated function of k.

In order to understand the effects of the $k(\lambda|\vec{\theta}_N)$ term, we now consider its effects on just one order of the 51-order echelle spectrum. The first four Chebyshev polynomials are

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_2(x) = 2x^2 - 1$$
$$T_3(x) = 4x^3 - 3x$$

In our implementation, we map the full pixel range of a specific order $\lambda \in [\lambda_{min}, \lambda_{max}]$ to $x \in [-1, 1]$. By mapping pixel number in a given order, rather than wavelength, we are insensitive to wavelength shifts of the model. Then, for a given set of Chebyshev coefficients, $\vec{\theta}_N = \{c_0, c_1, c_2, \ldots, c_N\}$, we have

$$k(\lambda|\vec{\theta}_N) = c_0 T_0(\lambda)\left[1 + c_1 T_1(\lambda) + \ldots + c_N T_N(\lambda)\right] \tag{10}$$

$$k(\lambda|\vec{\theta}_N) = c_0 \left[1 + \sum_{i=1}^{N} c_i T_i(\lambda)\right] \tag{11}$$

When applied to scale the model flux f_M, this becomes

$$k(\lambda|\vec{\theta}_N)f_M = \left[1 + \sum_{i=1}^{N} c_i T_i(\lambda)\right](c_0 f_M) \tag{12}$$

$$k(\lambda|\vec{\theta}_N)f_M = c_0 f_M + c_1 T_1(c_0 f_M) + \ldots + c_N T_N(c_0 f_M) \tag{13}$$

This functional form of k allows c_0 to scale by large amounts while the higher order terms (denoted by c_n) are perturbations on top of the *scaled* model ($c_0 f_M$). However, see §A for an alternative model when perturbations are small. We can rewrite Equation 9 as

$$\ln\left[p(\vec{\theta}|\vec{D})\right] \propto -\frac{1}{2}\sum_\lambda \left[\frac{f_D(\lambda) - \left[1 + \sum_{i=1}^{N} c_i T_i(\lambda)\right]c_0 f_M(\lambda|\vec{\theta}_\star)}{\sigma(\lambda)}\right]^2 \tag{14}$$

To simplify the following discussion, we consider a single wavelength λ_i and then generalize this to a sum over λ later. If we expand out the square in Equation 14, then for a given λ_i we have

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2\sigma^2}\left[f_D^2 - 2f_D c_0 f_M\left[1 + \sum_{i=1}^{N} c_i T_i\right] + c_0^2 f_M^2 \left[1 + \sum_{i=1}^{N} c_i T_i\right]^2\right] \tag{15}$$

where σ, f_{D}, f_{M}, and $T_{n,m}$ are all evaluated at λ_i. We can further expand this to

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2\sigma^2}\left[f_{\mathrm{D}}^2 - 2f_{\mathrm{D}}c_0 f_{\mathrm{M}}\left[1 + \sum_{i=1}^{N} c_i T_i\right] + c_0^2 f_{\mathrm{M}}^2\left[1 + 2\sum_{i=1}^{N} c_i T_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_i T_i c_j T_j\right]\right] \tag{16}$$

To simplify this math, we can rewrite the Chebyshev coefficients and polynomials as column vectors $\vec{c}$ and $\vec{T}(\lambda)$, respectively

$$\vec{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} \qquad\qquad \vec{T}(\lambda) = \begin{bmatrix} T_1(\lambda) \\ T_2(\lambda) \\ \vdots \\ T_N(\lambda) \end{bmatrix} \tag{17}$$

then

$$k(\lambda|\vec{\theta}_{\mathrm{N}}) = c_0\left[1 + \sum_{i=0}^{N} c_i T_i\right] = c_0\left[1 + \vec{T}^{\mathsf{T}}\cdot\vec{c}\right] \tag{18}$$

If we let

$$\boldsymbol{W}(\lambda) = \vec{T}\cdot\vec{T}^{\mathsf{T}} = \begin{bmatrix} T_1 T_1 & T_1 T_2 & \ldots & T_1 T_N \\ T_2 T_1 & T_2 T_2 & \ldots & T_2 T_N \\ \vdots & \vdots & \ddots & \vdots \\ T_N T_1 & T_N T_2 & \ldots & T_N T_N \end{bmatrix} \tag{19}$$

then we have

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i T_i c_j T_j = \vec{c}^{\mathsf{T}}\cdot\boldsymbol{W}\cdot\vec{c} \tag{20}$$

We can rewrite Equation 16 quadratically in $\vec{c}$ as

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2\sigma^2}\left[f_{\mathrm{D}}^2 - 2f_{\mathrm{D}}c_0 f_{\mathrm{M}}\left[1 + \sum_{i=1}^{N} c_i T_i\right] + c_0^2 f_{\mathrm{M}}^2\left[1 + 2\sum_{i=1}^{N} c_i T_i + \sum_{i=1}^{N}\sum_{j=1}^{N} c_i T_i c_j T_j\right]\right] \tag{21}$$

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2}\frac{c_0^2 f_{\mathrm{M}}^2}{\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{N} c_i T_i c_j T_j + \left(\frac{-c_0^2 f_{\mathrm{M}}^2 + c_0 f_{\mathrm{D}} f_{\mathrm{M}}}{\sigma^2}\right)\sum_{i=1}^{N} c_i T_i - \left(\frac{c_0^2 f_{\mathrm{M}}^2 - 2c_0 f_{\mathrm{D}} f_{\mathrm{M}} + f_{\mathrm{D}}^2}{2\sigma^2}\right) \tag{22}$$

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2}\frac{c_0^2 f_{\mathrm{M}}^2}{\sigma^2}\vec{c}^{\mathsf{T}}\boldsymbol{W}\vec{c} + \left(\frac{-c_0^2 f_{\mathrm{M}}^2 + c_0 f_{\mathrm{D}} f_{\mathrm{M}}}{\sigma^2}\right)\vec{T}^{\mathsf{T}}\vec{c} - \left(\frac{c_0^2 f_{\mathrm{M}}^2 - 2c_0 f_{\mathrm{D}} f_{\mathrm{M}} + f_{\mathrm{D}}^2}{2\sigma^2}\right) \tag{23}$$

Because matrix multiplication is associative, we can sum $\boldsymbol{W}(\lambda)$ and $\vec{T}^{\mathsf{T}}(\lambda)$ across all λ, and define

$$\boldsymbol{A} = c_0^2 \sum_\lambda \frac{f_{\mathrm{M}}^2(\lambda)}{\sigma^2(\lambda)} \boldsymbol{W}(\lambda) \tag{24}$$

$$\vec{B} = \sum_\lambda \frac{-f_{\mathrm{M}}^2(\lambda)c_0^2 + f_{\mathrm{D}}(\lambda)f_{\mathrm{M}}(\lambda)c_0}{\sigma^2(\lambda)} \vec{T}(\lambda) \tag{25}$$

$$g = -\frac{1}{2} \sum_\lambda \frac{c_0^2 f_{\mathrm{M}}^2(\lambda) - 2c_0 f_{\mathrm{D}}(\lambda)f_{\mathrm{M}}(\lambda) + f_{\mathrm{D}}^2(\lambda)}{\sigma^2(\lambda)} \tag{26}$$

Where $\boldsymbol{A}$, $\vec{B}$, and g are each a function of $\vec{\theta}_\star$, $\vec{\theta}_{\mathrm{N}}$, and $\vec{D}$. Now we can rewrite Equation 14 as

$$\ln\left[p(\vec{\theta}_\star, \vec{\theta}_{\mathrm{N}}|\vec{D})\right] \propto -\frac{1}{2}\vec{c}^{\mathsf{T}}\boldsymbol{A}\vec{c} + \vec{B}^{\mathsf{T}}\vec{c} + g \tag{27}$$

$$p(\vec{\theta}|\vec{D}) = p(\vec{\theta}_\star, \vec{\theta}_{\mathrm{N}}|\vec{D}) \propto \exp\left(-\frac{1}{2}\vec{c}^{\mathsf{T}}\boldsymbol{A}\vec{c} + \vec{B}^{\mathsf{T}}\vec{c} + g\right) \tag{28}$$

Here we have the full posterior probability distribution assuming flat priors. This posterior is a function of both the stellar parameters $\vec{\theta}_\star$ and the nuisance parameters $\vec{\theta}_{\mathrm{N}}$. Generally, we are most interested in the stellar parameters after they have been marginalized over the nuisance parameters. This marginalization encapsulates the uncertainty due to the inference on the nuisance parameters. Because this probability function is a multi-dimensional Gaussian in the higher order Chebyshev coefficients $\vec{c} = \{c_1, c_2, \ldots, c_N\}$, we can do this marginalization analytically, and then sample from the marginalized distribution.

$$p(\vec{\theta}_\star, c_0|\vec{D}) = \int p(\vec{\theta}_\star, c_0, \vec{c}\,|\vec{D})\, \mathrm{d}\vec{c} \tag{29}$$

The analytic multi-dimensional Gaussian integral with linear term (Stoof et al. 2009) yields

$$p(\vec{\theta}_\star, c_0|\vec{D}) \propto \int \exp\left(-\frac{1}{2}\vec{c}^{\mathsf{T}}\boldsymbol{A}\vec{c} + \vec{B}^{\mathsf{T}}\vec{c} + g\right) \mathrm{d}\vec{c} \tag{30}$$

$$p(\vec{\theta}_\star, c_0|\vec{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\boldsymbol{A}|}} \exp\left(\frac{1}{2}\vec{B}^{\mathsf{T}}\boldsymbol{A}^{-1}\vec{B} + g\right) \tag{31}$$

Where N is the number of Chebyshev coefficients marginalized over, in our case $N = 3$. By doing this marginalization we have reduced the dimensionality of our posterior space from 200+ parameters (~ 7 stellar parameters plus 3 Chebyshev coefficients for each of 51 echelle orders) to ~ 60 (~ 7 stellar parameters and one c_0 for each of the 51 echelle

orders). This enables a dramatic speedup in the MCMC sampling. For flux-calibrated spectra where the variation in c_0 is expected to be small (less than how much?), we can approximate k in a different manner which allows marginalization over the c_0 coefficients as well, reducing the dimensionality of the space to only the ~ 7 stellar parameters. This approach is detailed in §A.

3.3. Including Gaussian priors on nuisance parameters

Using this polynomial formalism, it is possible to incorporate prior knowledge about the degree of necessary "re-fluxing" through priors on the nuisance parameters. From our flux-calibration experiment, we determined that the overall scaling could range by 10%, while the higher order corrections were less than 3%. Because c_0 is a scale factor, it should be equally probable to scale up by a factor of two as it is to scale down by a factor of two. This means we should be using a log-normal prior on c_0

$$p(c_0) = \frac{1}{\sqrt{2\pi}\sigma_{c_0}c_0} \exp\left(-\frac{(\ln c_0)^2}{2\sigma_{c_0}^2}\right) \tag{32}$$

$$\ln p(c_0) = \ln\left(\frac{1}{\sqrt{2\pi}\sigma_{c_0}c_0}\right) - \frac{(\ln c_0)^2}{2\sigma_{c_0}^2} \tag{33}$$

centered on $c_0 = 1$. Because the higher order polynomial terms are perturbations on the scaled model $(c_0 f)$, we assume Gaussian priors centered about 0

$$p(c_n) = \frac{1}{\sqrt{2\pi}\sigma_{c_n}} \exp\left(-\frac{c_n^2}{2\sigma_{c_n}^2}\right) \tag{34}$$

which we can express in vector form as

$$p(\vec{c}) \propto \exp\left(-\frac{1}{2}(\vec{c} - \vec{\mu})^\mathsf{T} \boldsymbol{D}(\vec{c} - \vec{\mu})\right) \tag{35}$$

where

$$\boldsymbol{D} = \begin{bmatrix} \sigma_1^{-2} & 0 & \dots & 0 \\ 0 & \sigma_2^{-2} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \sigma_N^{-2} \end{bmatrix} \tag{36}$$

and σ_i represents the width of the Gaussian prior on the i-th Chebyshev coefficient, in our case $\sigma_i = \sigma_{c_n} \approx 0.03$. $\vec{\mu}$ is the mean of the Gaussian prior, in nearly all cases we expect

to set this to $\vec{\mu} = \{0, 0, \ldots, 0\}$, unless we have prior knowledge that the flux-calibration is systematically warped in one direction. Now, we can write the posterior with priors as

$$p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D}) = p(\vec{\theta}_\star, c_0, \vec{c} \,| \vec{D}) \propto p(\vec{D} | \vec{\theta}_\star, c_0, \vec{c}\,) p(\vec{c}\,) p(\vec{\theta}_\star) p(c_0) \tag{37}$$

$$p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D}) \propto \exp\left(-\frac{1}{2}\vec{c}^\mathsf{T} \boldsymbol{A} \vec{c} + \vec{B}^\mathsf{T}(\vec{\theta}_\star) \vec{c} + g(\vec{\theta}_\star)\right) \exp\left(-\frac{1}{2}(\vec{c} - \vec{\mu})^\mathsf{T} \boldsymbol{D}(\vec{c} - \vec{\mu})\right) p(\vec{\theta}_\star) p(c_0) \tag{38}$$

we can expand and rearrange the argument of Equation 35 to a similar form that is quadratic in $\vec{c}$

$$-\frac{1}{2}(\vec{c} - \vec{\mu})^\mathsf{T} \boldsymbol{D}(\vec{c} - \vec{\mu}) = \frac{1}{2}\left(-\vec{c}^\mathsf{T} \boldsymbol{D} \vec{c} + \vec{c}^\mathsf{T} \boldsymbol{D} \vec{\mu} + \vec{\mu}^\mathsf{T} \boldsymbol{D} \vec{c} - \vec{\mu}^\mathsf{T} \boldsymbol{D} \vec{\mu}\right) \tag{39}$$

$$= -\frac{1}{2}\vec{c}^\mathsf{T} \boldsymbol{D} \vec{c} + (\boldsymbol{D} \vec{\mu})^\mathsf{T} \vec{c} - \frac{1}{2}\vec{\mu}^\mathsf{T} \boldsymbol{D} \vec{\mu} \tag{40}$$

then we can rewrite

$$\boldsymbol{A}' = \boldsymbol{A} + \boldsymbol{D} \tag{41}$$

$$\vec{B}' = \vec{B} + (\boldsymbol{D} \vec{\mu})^\mathsf{T} \tag{42}$$

$$g' = g - \frac{1}{2}\vec{\mu}^\mathsf{T} \boldsymbol{D} \vec{\mu} \tag{43}$$

in the case that $\vec{\mu} = \{0, 0, \ldots, 0\}$, there is only a non-zero correction to $\boldsymbol{A}$. The full posterior probability distribution is then

$$\boxed{p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D}) = p(\vec{\theta}_\star, c_0, \vec{c}\,) \propto \exp\left(-\frac{1}{2}\vec{c}^\mathsf{T} \boldsymbol{A}' \vec{c} + \vec{B}'^\mathsf{T} \vec{c} + g'\right) p(\vec{\theta}_\star) p(c_0) p(\vec{c}\,)} \tag{44}$$

This probability function appears in the code as `lnprob_lognormal` and has ~ 7 stellar dimensions and ~ 200 nuisance dimensions.

Using the previous multi-dimensional integral, we can marginalize over the $\vec{c}$ including the Gaussian priors

$$\boxed{p(\vec{\theta}_\star, c_0 | \vec{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\boldsymbol{A}'|}} \exp\left(\frac{1}{2}\vec{B}'^\mathsf{T} \boldsymbol{A}'^{-1} \vec{B}' + g'\right) p(\vec{\theta}_\star) p(c_0)} \tag{45}$$

where N is the number of Chebyshev coefficients marginalized over, in our case $N = 3$. This probability function appears in the code as `lnprob_lognormal_marg` and has ~ 7 stellar dimensions and ~ 50 nuisance dimensions. If we increase the permissible degree of re-fluxing by increasing σ_i, this will have the effect of smearing out or widening the posterior on a given $\vec{\theta}_\star$.

3.4. Sampling a un-flux calibrated model

There also exists a wealth of archival *TRES* data that is not flux calibrated, since its primary purpose was for radial velocity determination of FGK stars. To fit this data we can use outdated sensitivity functions and soften the priors on the coefficients.

This invalidates the previous section on normalization. However, for corrections of less than XX%, this approximation is better than XX% accurate.

For the flux-calibration exercise, the overall spread in c_0 across all orders should not be more than $\sigma_{c_0} \leq 2$, or the one-sigma deviation of c_0 is between $0.125 \leq c_0 \leq 8.0$.

The plot in plots/priortests shows that the log-normal prior is the correct choice. Since we care most about the inference on F, we want to know that the choice on prior of c_0 is not biasing the final outcome, while a Gaussian prior might if σ_{c_0} was large. How to quantify this error and create a dividing line? For 5% error? In peak? Do these plots need to be in the paper?

Compare distributions by doing a K-S test on the cumulative distributions between Gaussian and Log-normal?

3.5. Hierarchical modelling

Can have a c_1, etc, that are marginalized over but we sample c_0 in a hierarchical manner which informs the overall flux-level.

3.6. Sampling with emcee

Citation to DFM (Foreman-Mackey et al. 2012), GW (Goodman & Weare 2010), and CosmoHAMMER (Akeret et al. 2013).

4. Moving beyond a perfect stellar model

4.1. Additional sources of noise beyond Poisson

Presumably we shouldn't have to add an additional source of "noise" to the fit, because our nuisance parameters will find an exact flux calibration.

Interpolation errors. Quantified via Husser et al. Ph.D. thesis as well as custom tests.

4.2. Noise, Masking, Residuals, and Alternative likelihood functions

In the future, we could modify our likelihood function to use a Student-t statistic instead of a Gaussian, such that it would be more robust in the presence of outlier pixels, however this does not seem necessary at the moment.

4.3. Masking

We must necessarily mask certain wavelength regions of the T Tauri spectra that are contaminated (one might also say "made more interesting") by the astrophysical realities of accretion or stellar spots (such as the Balmer lines, which are in emission). We also accept the assumption that rotation or accretion onto the star does not fundamentally alter the structure of the model atmosphere that generates the model spectrum and that a spectral comparison of the "clean" regions of the spectrum is valid.

Section on how to choose masked regions, testing the PHOENIX grid to see where the derivative is highest and thus gives the best leverage on stellar parameters.

Iterative masking to deal with Class I and II errors a la ULySS. They use a $\kappa - \sigma$ routine, where they iteratively sigma clip (Website link).

5. Gallery of likelihood functions

If you do a straight up χ^2 fit to the data, you can get posteriors that look like ± 1 K and ± 0.01 dex in gravity and metallicity. This would be better than that obtained by astroseismology! Clearly there is something going wrong here.

Lorentzian, linear combination of Gaussians, Sivia w/ prior on sigma. χ_Q^2 model.

These likelihood functions seem to be an improvement over the straight χ^2 function, in the sense that they give parameter estimates that are

5.1. Bad data model

The bad data model (Press 1997; Hogg et al. 2010) is a powerful framework for forward-modelling of data, allowing the modeler to account for the probability of outliers.

Now, we have a q_i for each data point, such that for the entire data set we have an ensemble of $\{q_i\}_{i=1}^{N}$. If this ensemble of $\{q_i\}_{i=1}^{N}$ is chosen appropriately, and the good data points are flagged as good and the bad data points are flagged as bad, the likelihood will be maximized. Written out explicitly, this is

$$\mathcal{L} = p(\boldsymbol{D}|m, b, \{q_i\}_{i=1}^{N}, Y_b, V_b) = \prod_{i=1}^{N} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y_i - mx_i - b)^2}{2\sigma^2} \right] \right]^{q_i} \times$$

$$\left[\frac{1}{\sqrt{2\pi(V_b + \sigma_{y_i}^2)}} \exp\left[-\frac{(y_i - Y_b)^2}{2(V_b + \sigma_{y_i}^2)} \right] \right]^{1-q_i} \quad (46)$$

using a Bernoulli prior

$$p(q_i|P_b) = [1 - P_b]^{q_i} P_b^{[1-q_i]} \quad (47)$$

$$p(\{q_i\}_{i=1}^{N}|P_b) = \prod_{i=1}^{N} [1 - P_b]^{q_i} P_b^{[1-q_i]} \quad (48)$$

For this data point, we can integrate (in this case, discrete sum) over the two states of q_i: $q_i = 1$ and $q_i = 0$.

$$p(m, b, P_b, Y_b, V_b|D_i) = \int p(m, b, q_i, P_b, Y_b, V_b|D_i)\mathrm{d}q_i$$

$$= p(m, b, q_i = 1, P_b, Y_b, V_b|D_i) + p(m, b, q_i = 0, P_b, Y_b, V_b|D_i)$$

$$= \left[p_{\text{good}}(D_i|m, b)\, p(q_i = 1|P_b) + p_{\text{bad}}(D_i|Y_b, V_b)\, p(q_i = 0|P_b) \right] \times p(m, b, P_b, Y_b, V_b) \quad (49)$$

$$p(m, b, P_b, Y_b, V_b \mid \boldsymbol{D}) \propto \prod_{i=1}^{N} \Big[[1 - P_b] p_{\text{good}}(D_i \mid m, b) + P_b p_{\text{bad}}(D_i \mid Y_b, V_b) \Big] \times p(m, b, P_b, Y_b, V_b)$$

$$= \prod_{i=1}^{N} \left\{ \frac{1 - P_b}{\sqrt{2\pi\sigma_{yi}^2}} \exp\left[-\frac{(y_i - mx_i - b)^2}{2\sigma_{yi}^2} \right] + \frac{P_b}{\sqrt{2\pi(V_b + \sigma_{yi}^2)}} \exp\left[-\frac{(y_i - Y_b)^2}{2(V_b + \sigma_{yi}^2)} \right] \right\} \times p(m, b, P_b, Y_b, V_b)$$

$$(50)$$

In it's simplest form (Hogg 2010), the bad data model results in a linear combination of Gaussians.

A benefit of the bad data model over a chosen likelihood function is that it is motivated. It may still result in the linear combination of Gaussians (nice), but preserves real parameter correlations. You are not forcing the same mean.

5.2. Spatial correlations of pixels

We can do more. The bad model of the previous section assumes that each data point is independent. For example, the spectral lines are actually bad, and this means that if a single line is wrong, not just the core of the line (which may be a 10 - 20 sigma residual) will be wrong, but also the wings of the line, which may be only 5 or 3 sigma wrong.

Now, let's come up with a model which includes spatial (wavelength) correlations in the bad data model.

How could we accomplish this? If we identify one point as bad, then the flags around it will have a correlation length, ie, the prior on q_i will be bad. This means that for however many nearby pixels that will be bad (most likely a constant amount for uncrowded lines), then this means that we could still integrate over the qis, it's just that we would now have to take a contribution from the nearest 5 pixels, say. That central pixel would then mark the "line" that is bad.

That is one way to mark the "badness of the line," that the qi are correlated. But we also know that the y_i will be correlated too.

This correlation length will probably depend on the type of star, and the contribution of the band. For example, if they're simple FGK stars, then the bad lines might be

singular. But if they are M stars, then there is a chance that there might be a large "pseudo-continuum" region that might be incorrect.

Once we have identified the center of the bad line, then there will be a decrease in the nearby pixels, similar to a Gaussian shape. Presumably we could come up with a correlation in this space too. (or, it could simply be that the correlation between y_i is flat in the case of an M star.

This might actually save the marginalization. We could still do the backwards "what is the probability a given line is bad."

For Class I and II errors, we could have a hyper parameter describing the width of each line, and another hyper parameter describing the (distribution of) depth. These could also be sampled in.

For Class III errors, describing how line strengths are off makes sense for stars with continuum, but as we move to later type stars (K stars) that do not have a nice continuum, we may run into trouble. For the later stars (M dwarfs), there might be a whole region that is incorrect.

There are different ways to describe a correlation over pixels. Gaussian random fields. Autocorrelation function.

5.3.   Simple correlation between pixels

For a simple correlation, say that if a given q_i is bad, then there is also a high probability that the neighboring pixels are also bad. Say we know from inspection that it seems as though bad lines knock out 5 pixel chunks.

Caveat: I think this assumes that bad lines will not occur within the same spread of each other. Maybe we should include a P_B prior on the good counts too?

$$p(q_i = 0 | q_{i\pm5} = 0) = P_B \tag{51}$$

$$p(q_i = 0|q_{i\pm5} = 0) = 1/5 \tag{52}$$
$$q_{i\pm4} = 2/5 \tag{53}$$
$$q_{i\pm3} = 3/5 \tag{54}$$
$$q_{i\pm2} = 4/5 \tag{55}$$
$$\tag{56}$$

This might need more work with P_B.

Then the probability functions would be

$$p(D_{i+j}|\vec{\theta}, q_i = 0) = \prod_{j=-4,\neq0}^{4} \big[p(D_{i+j}|\vec{\theta}, q_{i+j} = 0) p(q_{i+j} = 0|q_i = 0) +$$
$$p(D_{i+j}|\vec{\theta}, q_{i+j} = 1) p(q_{i+j} = 1|q_i = 0) \big] \tag{57}$$

$$p(D_{i+j}|\vec{\theta}, q_i = 1) = \prod_{j=-4,\neq0}^{4} \big[p(D_{i+j}|\vec{\theta}, q_{i+j} = 0) p(q_{i+j} = 0|q_i = 1) +$$
$$p(D_{i+j}|\vec{\theta}, q_{i+j} = 1) p(q_{i+j} = 1|q_i = 1) \big] \tag{58}$$

The marginalization could still be done analytically, but now the likelihood function involves a correlation between adjacent pixels. Could we use a sparse array for this? Or, we could evaluate all of the pixels, and then have the final likelihood function involve some correlation terms between the lnprobs evaluated at each pixel. We could do this using the logarithm so that everything is additive.

Or, what if we evaluated the FFT of the $\log(p(D_i))$ and just the autocorrelation in Fourier space?

5.4. Correlation between y_i

Not only will the flags be spatially correlated (if a point is bad, the nearby ~ 5 points are also likely to be bad) but for OBAFGK stars, the lines are likely to be Gaussian shape. That is if q_i specifies the center of the line, then the neighbooring y_{i+1} or y_{i-1} will be of

lower residual height. This means that the inside of the likelihood function should also have spatial correlations.

Could this correlation be described by a Gaussian random process? (Rasmussen & Williams 2005).

5.5. Different flags and weights

Instead of a good or bad flag, what if we had a flag that would identify which class of error the line became? For example we could have the option of $q_i = \{0, 1, 2, 3\}$, where $q_i = 0$.

How do we account for the fact that some parts of the spectrum carry much more information about $\vec{\theta}_{\star,\mathrm{grid}}$ than other parts of the spectrum? Is it important to weight this parts beforehand? Or should this contribution to the posterior naturally weight itself, the way certain points might heavily contribute to the slope and intercept of a linear fit?

If so, we could imagine having a Gibbs sampler on the weights of each pixel.

Proper Class III modeling is very important, following the discussion of (Mann et al. 2013) Mann et al, and see recent example from Rajpurohit 2103. Some features will not fit perfectly, especially at the lower temperature scale. However, it's important to still take these features int account with the modelling. They also use a χ^2 likelihood function, but perhaps even less appropriate is that they draw error bars by saying that the χ^2 value changes by 5%. In reality, they are checking the fits by eye and saying whether or not a spectrum matches. This approach is no more justified than a by-eye spectral classification.

5.6. Hyperparameters

Difficulty of combining different data sets with different errors (Ma & Berndsen 2013). Or a hyperparameter like (Shkedy et al. 2007). Shrinkage similar to Kruschke (2010).

Shkedy et al. (2007) use ISO SWS infrared spectra. Hierarchical approach. Able to synthesize their own spectra. Use systematic error flags from the instrument, as well as statistical flags. Hierarchical model that uses shrinkage between the observed spectrum and the synthetic spectrum. They do a fit to the variance with a hierarchical linear mixed model (sec 5.2.2, and their figure 7) using design matricies and regression coefficients. This is probably to incorporate some sort of correlation length on the systematic errors, and

"smooth" the function, as they say. I was having some difficulty understanding it, but it seems as though they were keeping the noise components fixed, and they did not have a contribution to the error budget from regions where the spectral synthesis *itself* contained systematic errors. Perhaps these were masked by the systematic error from the instrument.

5.7. Sampling such a model

If our bad model is sufficiently complex that we are not able to analytically marginalize over the q_i, it is still possible to sample in this huge parameter space. A stellar model with flags for each pixel would have $\vec{\theta}_\star + N$ parameters, where for a typical optical spectrum $N > 100,000$. Even using an ensemble sampler, sampling in this space would take a tremendous amount of time to converge.

An alternative way to approach this problem is to do Gibbs sampling in the flags. It may be possible to do do Gibbs sampling where we draw from the conditional probability distribution, or at the very worst we could still do a Gibbs sampler with a Metropolis-Hastings backend, where we propose a jump in a single q_i at a time. Such an sample requires none of the previous heavy lifting (spectral grid interpolation, FFT, or pixel interpolation), and so even though there are $\sim 100,000$ flags to sample, this is likely to be computationally feasible.

6. Injection of fake lines

An alternative approach is instead of modelling the systematically wrong regions of the spectrum, actually add in lines to the spectrum. This can be thought of as correcting the models. We could add in a set number of lines.

Presumably, each line would be a Gaussian parameterized by λ_0, strength, and FWHM. Strength could be described by a hyper prior, and FWHM described by a hyper parameter, since all lines should have pretty much the same FWHM (in velocity space).

In later type spectra, this might actually need to be a larger continuum feature that is added.

We could penalize the model fits by having a hyperprior on the number of bad lines that can be created.

In theory, it is possible to fit the profiles of only certain elements to determine the

abundances of certain elements (see citation in conference proceedings of Husser).

It may be possible to use a linelist from the group that produces the PHOENIX spectra, which describes the element that gave the highest contribution to the opacity in that bin. Now, if the vanadium lines are bad, then we could have a prior that the other vanadium lines might also be bad. We could assemble a piecewise spectrum, where certain pixels are drawn from a different combination of $\vec{\theta}_{\star,\text{grid}}$, and these pixels had an associated normalization constant so that the piecewise transition was smooth (if, say the bolometric temperature was different).

Class III errors might be difficult to fit in this approach. However, if the line width does not change dramatically, then we should be able to account for things this way.

7. Validation tests

Comparison to main sequence stars of HAT-P-9, and WASP-14, from the TRES archive (Torres et al. 2012).

Comparisons using HIRES data?

A. Simplification for small c_0 perturbations

If the correction in c_0 is small, we can use an alternate formulation of k and put Gaussian priors on all nuisance parameters. This allows analytic marginalization over all nuisance parameters (including c_0), and reduces the dimensionality of the posterior to only the ~ 7 stellar parameters, $\vec{\theta}_\star$.

$$k(\lambda|\vec{\theta}_{\text{N}}) = c_0 T_0(\lambda) + c_1 T_1(\lambda) + \ldots + c_N T_N(\lambda) \tag{A1}$$

$$k(\lambda|\vec{\theta}_{\text{N}}) = \sum_{i=0}^{N} c_i T_i(\lambda) \tag{A2}$$

Following §3.2

$$\ln\left[p(\vec{\theta}|\vec{D})\right] \propto -\frac{1}{2}\sum_{\lambda}\left[\frac{f_{\mathrm{D}}(\lambda) - \left[\sum_{i=0}^{N} c_i T_i(\lambda)\right] f_{\mathrm{M}}(\lambda|\vec{\theta}_\star)}{\sigma(\lambda)}\right]^2 \tag{A3}$$

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2\sigma^2}\left[f_{\mathrm{D}}^2 - 2f_{\mathrm{D}}f_{\mathrm{M}}\sum_n c_n T_n + f_{\mathrm{M}}^2 \sum_n \sum_m c_n T_n c_m T_m\right] \tag{A4}$$

Now the column vectors $\vec{c}$ and $\vec{T}(\lambda)$ include c_0 as well

$$\vec{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \qquad\qquad \vec{T}(\lambda) = \begin{bmatrix} T_0(\lambda) \\ T_1(\lambda) \\ \vdots \\ T_N(\lambda) \end{bmatrix} \tag{A5}$$

$$k(\lambda|\vec{\theta}_{\mathrm{N}}) = \sum_{i=0}^{N} c_i T_i = \vec{T}^{\mathsf{T}} \cdot \vec{c} \tag{A6}$$

$$\boldsymbol{W}(\lambda) = \vec{T} \cdot \vec{T}^{\mathsf{T}} = \begin{bmatrix} T_0 T_0 & T_0 T_1 & \ldots & T_0 T_N \\ T_1 T_0 & T_1 T_1 & \ldots & T_1 T_N \\ \vdots & \vdots & \ddots & \vdots \\ T_N T_0 & T_N T_1 & \ldots & T_N T_N \end{bmatrix} \tag{A7}$$

$$\sum_n \sum_m c_n T_n c_m T_m = \vec{c}^{\mathsf{T}} \cdot \boldsymbol{W} \cdot \vec{c} \tag{A8}$$

$$\ln\left[p(\vec{\theta}|\lambda_i)\right] \propto -\frac{1}{2\sigma^2}\left[f_{\mathrm{D}}^2 - 2f_{\mathrm{D}}f_{\mathrm{M}}\vec{T}^{\mathsf{T}}\vec{c} + f_{\mathrm{M}}^2\,\vec{c}^{\mathsf{T}}\boldsymbol{W}\vec{c}\right] \tag{A9}$$

$$\boldsymbol{A} = \sum_\lambda \frac{f_{\mathrm{M}}^2(\lambda)}{\sigma^2(\lambda)}\boldsymbol{W}(\lambda) \tag{A10}$$

$$\vec{B}^{\mathsf{T}} = \sum_\lambda \frac{f_{\mathrm{D}}(\lambda)f_{\mathrm{M}}(\lambda)}{\sigma^2(\lambda)}\vec{T}^{\mathsf{T}}(\lambda) \tag{A11}$$

$$g = -\frac{1}{2}\sum_\lambda \frac{f_{\mathrm{D}}(\lambda)}{\sigma^2(\lambda)} \tag{A12}$$

$$\tag{A13}$$

Where $\boldsymbol{A}$, $\vec{B}$, and g are each a function of $\vec{\theta}_\star$, $\vec{\theta}_N$, and $\vec{D}$. If we include Gaussian priors, then we have

$$\boldsymbol{A}' = \boldsymbol{A} + \boldsymbol{D} \tag{A14}$$

$$\vec{B}' = \vec{B} + (\boldsymbol{D}\vec{\mu})^\mathsf{T} \tag{A15}$$

$$g' = g - \frac{1}{2}\vec{\mu}^\mathsf{T} \boldsymbol{D} \vec{\mu} \tag{A16}$$

where now $\boldsymbol{D}$ and $\vec{\mu}$ include c_0 as well. The main difference from before is that now the prior on c_0 is now Gaussian instead of log-normal. This is acceptable as long as σ_{c_0} is small $(< XX\%)$, otherwise this will bias the correction factor. This also means that we will usually have $\vec{\mu} = \{1, 0, \ldots, 0\}$. The full posterior probability function becomes

$$p(\vec{\theta}_\star, \vec{\theta}_N | \vec{D}) = p(\vec{\theta}_\star, \vec{c}\,) \propto \exp\left(-\frac{1}{2}\vec{c}^\mathsf{T} \boldsymbol{A}' \vec{c} + \vec{B}'^\mathsf{T} \vec{c} + g'\right) p(\vec{\theta}_\star) p(\vec{c}) \tag{A17}$$

Using the previous multi-dimensional integral, we can marginalize over all $\vec{c}$ including the Gaussian priors

$$p(\vec{\theta}_\star | \vec{D}) \propto \sqrt{\frac{(2\pi)^N}{\det|\boldsymbol{A}'|}} \exp\left(\frac{1}{2}\vec{B}'^\mathsf{T} \boldsymbol{A}'^{-1} \vec{B}' + g'\right) p(\vec{\theta}_\star) \tag{A18}$$

To gain insight, we can re-express this new version of k in the original form as

$$k(\lambda|\vec{\theta}_N) = c_0 T_0 + c_1 T_1 + \ldots + c_N T_N \tag{A19}$$

where $c_1 = c_0 c_1' T_0$ and $c_N = c_0 c_N' T_0$. Thus, if there needs to be a large perturbation in c_0, then there must also be a large perturbation in c_n. The prior on c_1 is now a product of a log-normal prior on c_0 and a Gaussian prior on c_1', which is not analytically tractable. Therefore, in the case of large perturbations, it is better to use Equation and use a log-normal prior on c_0. When we use the original framework, we are saying that perturbations in c_0 are small enough that we can approximate the log-normal prior on c_0 as Gaussian and that the priors on c_n can also be correctly approximated by a Gaussian. From tests, the original framework gives acceptable accuracy (using $\sigma_{c_n} = 0.05$) if the corrections in c_0 are $\sigma_{c_0} \leq 0.35$, or the one-sigma deviation of c_0 is between $0.7 \leq c_0 \leq 1.42$. Beyond this, one should switch to this framework, for example to correct for un-flux calibrated data.

REFERENCES

Akeret, J., Seehars, S., Amara, A., Refregier, A., & Csillaghy, A. 2013, CosmoHammer: Cosmological parameter estimation with the MCMC Hammer, 1212.1721, astrophysics Source Code Library

Astropy Collaboration et al. 2013, A&A, 558, A33

Bailer-Jones, C. A. L. 2010, MNRAS, 403, 96, 0911.5242

Baraffe, I., Chabrier, G., Allard, F., & Hauschildt, P. H. 2002, A&A, 382, 563, arXiv:astro-ph/0111385

Buchhave, L. A. et al. 2012, Nature, 486, 375

Coelho, P., Barbuy, B., Meléndez, J., Schiavon, R. P., & Castilho, B. V. 2005, A&A, 443, 735, astro-ph/0505511

D'Antona, F., & Mazzitelli, I. 1997, Mem. Soc. Astron. Italiana, 68, 807

Doppmann, G. W., & Jaffe, D. T. 2003, AJ, 126, 3030, astro-ph/0308521

Eisenstein, D. J. et al. 2006, ApJS, 167, 40, arXiv:astro-ph/0606700

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2012, ArXiv e-prints, 1202.3665

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. 2004, Bayesian data analysis (Chapman & Hall/CRC)

Goodman, J., & Weare, J. 2010, Communications in Applied Mathematics and Computational Science, 5, 65

Hogg, D. W., Bovy, J., & Lang, D. 2010, ArXiv e-prints, 1008.4686

Husser, T.-O., Wende-von Berg, S., Dreizler, S., Homeier, D., Reiners, A., Barman, T., & Hauschildt, P. H. 2013, A&A, 553, A6, 1303.5632

Koleva, M., Prugniel, P., Bouchard, A., & Wu, Y. 2009, A&A, 501, 1269, 0903.2979

Kruschke, J. 2010, Doing Bayesian Data Analysis: A Tutorial Introduction with R (Academic Press)

Lee, Y. S. et al. 2008, AJ, 136, 2022, 0710.5645

Ma, Y.-Z., & Berndsen, A. 2013, ArXiv e-prints, 1309.3271

Mann, A. W., Gaidos, E., & Ansdell, M. 2013, ApJ, 779, 188, 1311.0003

Martins, L. P., & Coelho, P. 2007, MNRAS, 381, 1329, 0705.1994

Newton, E. R., Charbonneau, D., Irwin, J., Berta-Thompson, Z. K., Rojas-Ayala, B., Covey, K., & Lloyd, J. P. 2014, AJ, 147, 20, 1310.1087

Paletou, F., Trouilhet, J.-F., & Boehm, T. 2014, ArXiv e-prints, 1401.1083

Press, W. H. 1997, in Unsolved Problems in Astrophysics, ed. J. N. Bahcall & J. P. Ostriker, 49–60, arXiv:astro-ph/9604126

Prugniel, P., Soubiran, C., Koleva, M., & Le Borgne, D. 2007, VizieR Online Data Catalog, 3251, 0

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

Recio-Blanco, A., Bijaoui, A., & de Laverny, P. 2006, MNRAS, 370, 141, astro-ph/0604385

Sánchez Almeida, J., Aguerri, J. A. L., Muñoz-Tuñón, C., & de Vicente, A. 2010, ApJ, 714, 487, 1003.3186

Santos, N. C. et al. 2013, A&A, 556, A150, 1307.0354

Schoenrich, R., & Bergemann, M. 2013, ArXiv e-prints, 1311.5558

Shkedy, Z., Decin, L., Molenberghs, G., & Aerts, C. 2007, MNRAS, 377, 120, arXiv:astro-ph/0701449

Sneden, C. A. 1973, PhD thesis, THE UNIVERSITY OF TEXAS AT AUSTIN.

Sordo, R. et al. 2011, Journal of Physics Conference Series, 328, 012006

Stoof, H. T., Gubbels, K. B., & Dickerscheid, D. B. 2009, Ultracold quantum fields (Springer)

Torres, G., Fischer, D. A., Sozzetti, A., Buchhave, L. A., Winn, J. N., Holman, M. J., & Carter, J. A. 2012, ApJ, 757, 161, 1208.1268

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

_____