Infrared dwarfs: Surface gravity sensitivity in H-band spectra

Santiago Rocha Novella

Division of Astrophysics
Department of Physics

LUND UNIVERSITY

2024–EXA237

Degree project of 15 higher education credits
August 2024

Supervisor: Thomas Bensby

Division of Astrophysics
Department of Physics
Box 118
SE-221 00 Lund
Sweden

Abstract

The age of the center of our galaxy is poorly known. Determining the ages of individual stars in the galactic center usually requires isochrone fitting on a Hertzsprung-Russel diagram, which in turn requires knowledge of the star's fundamental parameters, such as surface gravity, which is often determined with spectroscopic methods; and metallicity, which affects the shape of the isochrone. This can only be done for dwarf and subgiant stars, as in their respective regions of the diagram the isochrones are well separated and distinct. Such analysis has been performed for dwarf stars in the outer regions of the galactic bulge, but not for dwarf stars in the inner bulge due to the high amounts of extinction. However, infrared wavelengths are less extincted. To this end, this project investigates infrared wavelengths of synthetic stellar spectra for spectral lines sensitive to changes of 0.25, 0.5, 0.75, and 1.0 dex in $\log g$. Synthetic spectra are produced in PySME. Changes in $\log g$ are investigated by dividing two synthetic spectra varying by a certain dex in $\log g$ while sharing all other input parameters to create the so-called response curve, where regions that vary between both spectra have values greater or less than 1. Peaks in this response curve thus indicate spectral lines sensitive to changes in surface gravity. Response peaks with a strength greater than 2% are analyzed qualitatively as a large sample, as well as individually in representative cases. A small analysis with synthetic Gaussian noise is performed to qualitatively determine at what signal-to-noise ratios (S/R) the synthetic spectrum or the response curve become unrecognizable to inform future observations of these spectral lines in bulge stars. It is concluded that large numbers of sensitive peaks exist in the chosen wavelength range, most are considered weak (of response $\approx 2\%$). Of the identified peaks, C I, Si I, Mg I, and H I transitions account for the majority of sensitive spectral lines. These are suggested as avenues for future theoretical work in synthetic models of dwarf stars.

Aknowledgements

This work has made use of the VALD database, operated at Uppsala University, the Institute of Astronomy RAS in Moscow, and the University of Vienna.

In Plain Sight: The Unknown Stars of our Galaxy

Astronomy has often been called humanity's oldest science: ever since we could recognize ourselves as such, humans have been gazing at the stars and wondering. Our access to technology has only expanded our horizons of observation, which uncover new questions about the universe and our place within it. Even though we can now observe galaxies billions of light years away, some of the most enigmatic stars are a little closer to home.

Aside from the stars themselves, our galaxy is composed in large part of free-floating gas and dust, molded by gravity, radiation pressure, and supernova shockwaves into all sorts of beautiful shapes. However, this dust also absorbs and scatters starlight, causing stars behind concentrations of this dust to become significantly dimmed in a phenomenon known as extinction. Depending on where a star is located in the galaxy relative to us, it could be subject to varying amounts of extinction, and those lying in the galactic plane (where most of this dust is located) will be the most heavily extincted.

The stars of the galactic center, known as the bulge, have long remained mysterious due to the intense extinction their light suffers before it reaches our telescopes. Some of these stars can be made up to 100 times dimmer—for small dwarf stars, this can make them virtually unobservable. Thus, if we are to understand this population of stars, novel observation methods must be employed. To this end, this project focuses on infrared light, which is much less subject to extinction than shorter wavelengths.

Scientists can learn a great deal about a star by analyzing the imprint certain elements leave on the star's light. Such a graph is known as a spectrum. It is a fingerprint of sorts, showing the composition of the star (by which lines appear at what wavelength), as well as many of the star's properties (by the shape and strength of specific spectral lines). Two of these properties, the effective temperature and surface gravity of the star, have well-known effects on the spectral lines of the visible spectrum.

However, in the case of surface gravity, knowing exactly which spectral lines are affected is necessary, and this is not well known for the infrared region. In order to make observation of the bulge dwarfs with infrared telescopes at all feasible, these spectral lines must first be identified, and adequate candidates for study selected from this sample. This, in short, is the aim of this project.

Learning more about the dwarf stars of the galactic center has numerous implications for our understanding of both our galaxy itself and how galaxies form generally. Perhaps most of these stars are incredibly similar in age and composition, suggesting a common origin in space and a rapid phase of star formation. Perhaps they have a great range of ages and all sorts of masses and temperatures, which would point to a continuous seeding of the galactic center with stars from diverse locations. Perhaps we will uncover a situation we have yet to conceive, and unearth even more questions about the history of our galaxy. Whatever the case, these stars present a tremendous opportunity to expand our knowledge horizon, and with it enrich both science and humanity.

Contents

Chapter 1

Introduction

1.1 Motivation: Our Little Niche

Every project forms a piece of a larger whole. Such a thing is obvious, and yet it bears reiterating. Where a project lies in the great web of its field informs its scope, its focus, what past knowledge it draws from and what new knowledge it hopes to create. So let me begin by placing this project within its proper context.

The centers of spiral galaxies, and the center of our Milky Way in particular, have been an active focus of modern astronomy. They are known as "bulges" in the case of spiral galaxies, in reference to how the density of stars and dust make these features appear to burst at their seams with light. However, their relative brightness compared to the rest of the galactic disk belies the difficulty of observing the bulge stars themselves. The aforementioned dust is but one part of the highly varied interstellar medium, which tends to absorb and scatter starlight with surprising effectiveness. Such is its density in the line of sight of the bulge that stars in this region are extincted by between 2 and 3.5 magnitudes, according to Figure 6 of Gonzalez, O. A. et al. (2012). Thus, already dim (and numerous) dwarf stars become too faint to observe in visible wavelengths. This poses a particular problem for our understanding of the galactic bulge, as we have thus far only characterized a minority of its stellar population.

One of the most important properties of a star is its age, which is usually determined via a technique known as isochrone fitting. Isochrones are curves on a Hertzprung-Russell (HR) diagram that represents a population of equally-old stars of different masses, as shown in Figure 1.1. In order to adequately determine the age of a star, one must know its surface gravity and its temperature to a high degree of accuracy, and where exactly the star lies on the HR diagram will determine exactly how much accuracy is required for each parameter. Consider Figure 1.2; for a "horizontally flat" section of an isochrone, significant uncertainties in temperature will scarcely affect a determination of which isochrone a star belongs to. Significant uncertainty in luminosity, however, may leave this rather ambiguous. Thus, if we are to adequately characterize this enigmatic region of our galaxy, we must turn our telescopes to its much more numerous dwarf stars, and accurately determine their fundamental parameters. In particular, we must focus them on their infrared spectra, as infrared radiation is less subject to extinction from the interstellar medium. This poses new challenges, however, as the known spectral lines used to determine surface gravity are not present at infrared wavelengths.

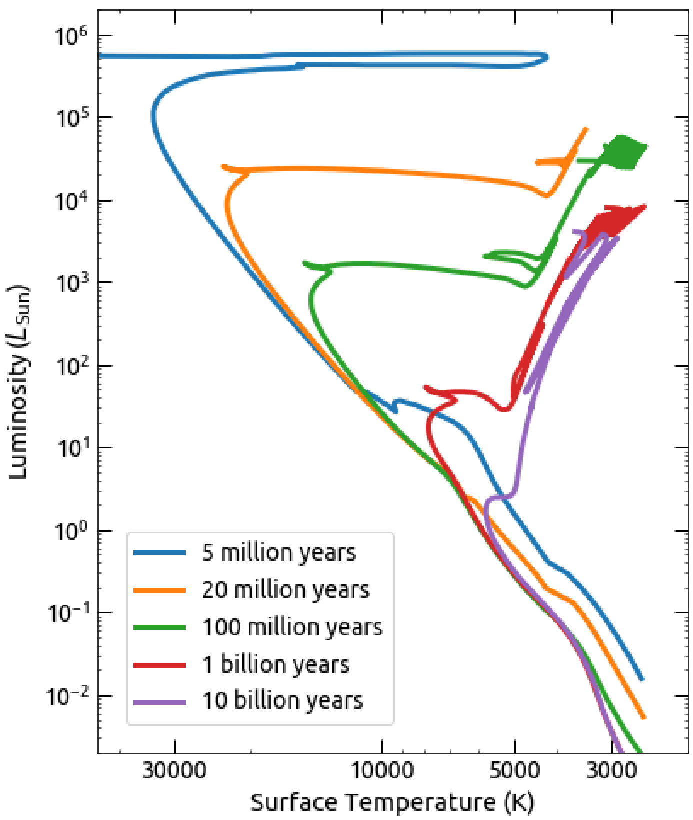

Figure 1.1: Theoretical isochrones for near-solar metallicities. Produced by Ivan Ramirez, Astronomy/Physics Professor at Tacoma Community College. `https://commons.wikimedia.org/wiki/File:Isochrones_of_several_ages.png`.

Herein lies our little niche. Without access to the regular methods to determine surface gravity, new ones must be investigated. This project is a proof-of-concept of sorts, aiming to evaluate the viability of determining surface gravity from infrared spectra of dwarf stars in the galactic center. This is done by the creation of synthetic spectra with PySME, covering a range of surface gravity and metallicity values typical of galactic-center dwarf stars in the sample from Bensby et al. (2017). Gravity-sensitive peaks are found in this wavelength range. Gaussian noise is then added to a representative sample of these peaks to model the effects of real observations, and to judge whether these peaks will be visible even in noisy data. Versions of this methodology have been employed in past, such as in Nandakumar et al. (2023), where stellar parameters are determined for giant stars using near-infrared spectroscopy and an iterative method with SME, with an eye to future applications in

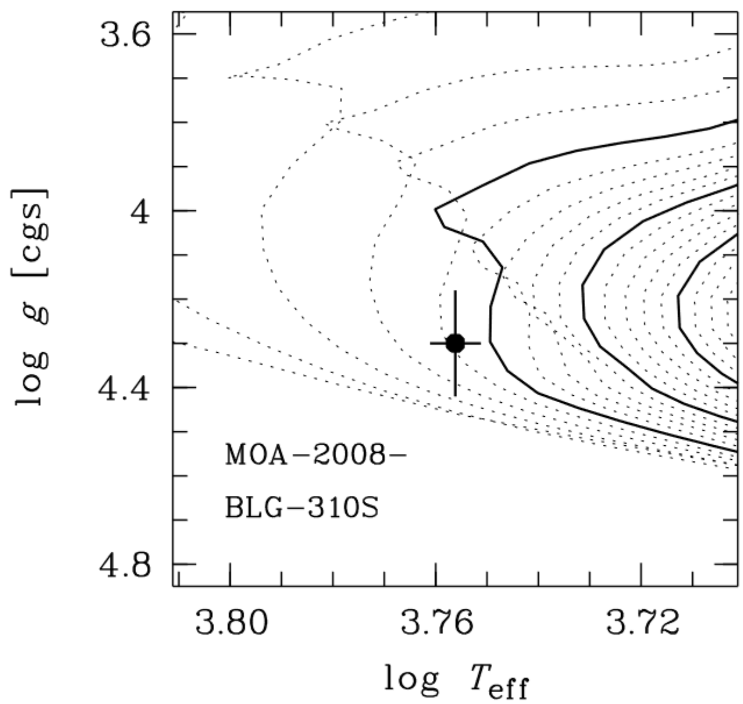

Figure 1.2: A star fitted onto isochrones as determined by Demarque et al. (2004). Plot is taken from Figure 4 of Bensby et al. (2010). Solid lines represent isochrones for 5, 10, and 15 Gyr, from left to right. Dotted lines represent isochrones in steps of 1 Gyr from 0.1 to 20 Gyr.

Galactic center astronomy. These methods are employed again in Nandakumar et al. (2024) in a successful study of Bulge giants, proving the reliability of this approach. Indeed, the work of this thesis is to lay the groundwork for a parallel method for dwarf stars.

1.2 The Galactic Center

Our galaxy and others like it are characterized by their structure. It is self-evident in our nomenclature: they are called "spiral galaxies," in reference to the great arms that curve outward from a central dense region, itself seen to assume all sorts of shapes across our sample of spiral galaxies. Predictably enough, these central regions are known as "bulges."

Barbuy et al. (2018) provides an extensive and recent overview of the state of the literature on galactic bulges, as well as the many open questions presently in the field. The one that primarily concerns this thesis is the question of the Bulge's age, as determining the precise ages of Bulge stars will greatly constrain scenarios of the galactic center's formation. Previous literature reviews, such as the equally extensive one by Wyse et al. (1997) already revealed that our assumptions about the Bulge's age—that it was the oldest part of the galaxy—was being undermined by new discoveries, and Barbuy et al. (2018) provides salient examples. It mentions the finding of "two or three dozen" metal-rich dwarf stars, which implies at least a second generation of star formation seeded by nucleosynthesis from ancient supernovae. What proportion do these young stars form of the Bulge population? When did they form? Were they formed in the Bulge, or do they come from other populations that migrated inwards? These are some of the questions that such findings bring to mind, and are among the questions that this thesis aims to build towards answering.

Many projects have already moved in this direction. In Bensby et al. (2017), for instance, the authors use spectroscopic methods to determine the ages of 90 microlensed dwarf stars in the outer Bulge, determining that a significant portion of these stars are much younger than earlier models predict. However, this sample only covered between 2-8 degrees of galactic latitude-longitude coordinates; in other words, the outer reaches of the galactic bulge. It is within this inner 2 degrees that our interest lies, as the stars within this region are the most heavily extincted. The following sections will thus detail the specific challenges involved with observing these stars and how the fundamental parameters of a star, particularly surface gravity, affect spectral lines.

1.3 Observing the Bulge

The interstellar medium (henceforth ISM) is composed of a great variety of gaseous and particulate matter, in multiple stages of ionization, with a structured distribution resulting in regions of higher relative density. Its dust components are agreed to be mostly silicates and carbon-based molecules, some of which are coated in frozen volatiles such as water or methane (Stelter & Eikenberry 2020). The gas, meanwhile, is primarily hydrogen, alongside helium and trace amounts of heavier elements (Herbst 1995). In our galaxy, notable overdensities of the ISM occur in star-forming regions, in planetary nebulae, in hydrogen clouds, and in the spiral arm structures. The reason behind the ISM's infamous hampering of astronomical observations lies in the interactions between light and matter, as well as the sheer distances involved in said observations. Even though the ISM may be less dense in some regions than the best laboratory vacuums on Earth, the distances that photons emitted from stars travel are so vast that extinction becomes significant. Extinction is typically measured in units of magnitudes dimmed per kiloparsec. Therefore, the difficulty of observing the galactic center may now be apparent: in order for light from these stars to reach Earth, it must travel through several kiloparsecs of the galactic plane, which contains the highest densities of the ISM, before reaching our instruments.

As the effects of extinction decrease rapidly with increasing wavelength, our choice of observing in infrared bands minimizes the effect of extinction, while still leaving identifiable spectral features in our observation range. Even by limiting our observations to infrared, dwarf stars are incredibly faint, a problem which is only magnified by the tremendous distances involved. Bensby et al. (2017) makes use a technique known as microlensing to magnify the brightness of these Bulge dwarfs. According to Einstein's theory of general relativity, potent enough gravitational fields are able to bend the pathway of light. When a mass (such as another star) lies between our telescope and our target, the middle mass can act as a lens of sorts, magnifying the object behind it and making it appear brighter in the process. The best known examples of this gravitational lensing occur with incredibly heavy masses the size of galaxies or supermassive black holes, though this effect is still visible with stellar mass objects—hence the term 'microlensing.' The stars of the galactic center exist in both much greater in density and orbit with much greater velocities than stars of the galactic disk, which makes these microlensing events occur with sufficient enough frequency to be readily observable. When such an event occurs, the apparent magnitude of the target star increases by a factor of up to several hundred times, thus making it observable to our instruments.

1.4 The Stellar Spectrum

When photons are generated in the fusion processes of a star, they are absorbed and re-emitted constantly until they eventually reach a star's more transparent upper layers: what we would consider its "surface," known as the photosphere. These are the photons that reach our telescopes, and thus our only way to obtain information about the star. Due to this, the spectral information provided by this light is paramount to characterizing a star in detail, and thus much research has been done in understanding the behavior of spectral lines with respect to a variety of parameters. Three of these—surface gravity, effective temperature, and metallicity—are known as the fundamental parameters, and surface gravity and metallicity will be what primarily concerns this project.

1.4.1 The Fundamental Parameters

Pressure Dependence

Earth's gravity plays a crucial role in determining the pressure gradient of its atmosphere. So too does the gravity of a star determine the pressure gradient of its photosphere, with various effects on the spectral lines. It is for this reason that pressure and gravity dependence can be considered to be synonymous.

Determining a dwarf star's surface gravity with spectroscopic methods poses numerous challenges. Firstly, pressure effects in spectral lines are weaker than temperature effects. Usually, this is counteracted by the fact that, across the universal stellar population, gravity can vary over 4-5 orders of magnitude, while temperature varies across only a single order (Gray (2022)). Restricting our sample to dwarf stars removes this statistical advantage. Secondly, the species used for gravity determination at lower wavelengths are not guaranteed to exhibit peaks in the H-band. New candidates will thus have to be discovered. Additionally, only a minority of spectral lines are sensitive to pressure, unlike the temperature dependence, which affects all spectral lines.

Surface gravity in spectroscopic astrophysics is measured in CGS units as the logarithm of the surface gravity, denoted henceforth as $\log g$.

Metallicity Dependence

Metallicity is a measure of how much of a star is composed of elements heavier than hydrogen, and is defined in many different ways to suit different kinds of observations. In this case, the iron abundance ratio definition will be used, commonly denoted as [Fe/H], and defined as such:

$$[\mathrm{Fe/H}] = \log\left(\frac{N_{Fe}}{N_H}\right)_* - log\left(\frac{N_{Fe}}{N_H}\right)_\odot . \tag{1.1}$$

Thus, metallicity values with this definition are reported relative to the solar metallicity.

Metallicity directly relates to the abundance of particular elements in a star. As the strength of a spectral line largely depends on the sheer number of absorbants in the photosphere, higher metallicities will uniformly increase the strength of spectral lines belonging to elements other than hydrogen. Conversely, any of the star's hydrogen peaks will be overlaid by metallic peaks at high metallicities; as will be seen in the Results section, hydrogen features are much more visible at low metallicities.

Temperature Dependence

Though we do not consider temperature as an independent variable, it is important to understand how temperature affects the spectral line. As the excitation and ionization processes that produce spectral lines to begin with are highly temperature dependent, temperature has the greatest effect on the strength of a spectral line (Gray 2022). Thus, any variation in the strength of a spectral line between two different stars is likely to be mostly caused by a difference in temperature, especially so in a sample of dwarf stars of similar mass as aforementioned.

Parameter Determination from Spectral Lines

SME employs an iterative method to determine stellar parameters by comparing an observed spectrum to a synthetic spectrum, generated with an initial guess of unknown variables that were not previously determined by other means. The program then changes the desired value by a small amount, and compares the two spectra again—this process repeats until the two spectra match as closely as possible. Using this method, specific parameters can be fixed, and others left as free parameters to be solved for by the iteration. This is the aforementioned method used to determine fundamental parameters in Nandakumar et al. (2023). Their determination of effective temperature relies in the excitation balance of Fe I lines, while their determination of surface gravity relies on the ionization balance of Fe I and Fe II lines. Thus, the clear presence of a sufficient number of these spectral lines is a prerequisite for a spectroscopic determination of these variables. They are able to perform this analysis with giant stars in the near-infrared as these lines are visible and numerous. However, these lines are much weaker and much fewer in the spectra of dwarf stars. This prevents the application of this exact method.

Chapter 2

Methodology

As mentioned in the previous chapter, this thesis makes use of the Python version of Spectroscopy Made Easy (SME), originally developed in 1996 to fit an observed stellar spectrum onto a synthetic spectrum. The creation of such a synthetic spectrum is a rather remarkable achievement: the shape of a spectrum and the lines within it are the result of dozens of simultaneous processes. Even something as fundamental as *which* lines appear in the spectrum requires highly detailed and accurate atomic and molecular transition data from innumerable lab experiments across the world. In order to generate a synthetic spectrum, SME thus requires a list of all electron transitions that might occur in the stellar photosphere within a desired wavelength range, and all their associated data, compiled into a file known as a linelist.

For this project, such a linelist was obtained from the VALD database, spanning a wavelength range of 14300Å to 18000Å. This was chosen to correspond with the approximate coverage of H-band detectors on the European Southern Observatory's CRIRES instrument. In choosing this wavelength range, we center our objective to show the viability of this method—were a viable set of spectral lines identified in this interval, observation could begin immediately with existing instruments.

Further centering this project on previous work, we refer to the sample of Bulge stars found in Figure 4 in Bensby et al. (2017), and choose our fundamental parameters accordingly. The majority of the sample stars have a $\log g$ between 3.5 and 4.5, centered around an effective temperature T_{eff} of 5500K, with metallicities hovering between [Fe/H] = 0.3 and [Fe/H] = -1.0. Thus, these will be the ranges encompassing our set of synthetic spectra. A total of 15 synthetic spectra are created, using all combinations of the following parameters. Furthermore, synthetic Gaussian noise with three different signal-to-noise (S/N) ratios will be added to these spectra to investigate at which signal qualities the spectrum can still be recognized.

- $\log g$: 3.5, 3.75, 4.0, 4.25, 4.5
- T_{eff}: 5500K
- [Fe/H]: -1.0, 0.0, 0.3

- S/N: 30, 50, 100

These ranges cover the typical sorts of fundamental parameters observed for F and G-type dwarfs and subgiants, which are the potential population of interest in future H-band studies that concern this thesis. For the purposes of this investigation, we assume that effective temperature is fixed by other means, such as excitation balance, and could thus be input as a fixed parameter in SME.

2.1 Code

Aside from PySME itself, two additional Python scripts were made. The first one creates a synthetic spectrum and saves it as a file by specifying input parameters for SME. The second reads and divides pairs of these spectrum files element-wise, such that only one parameter changes from one spectrum to another. We call the resulting curve a response curve, which represents the sensitivity of a particular region of the spectrum to changes in particular parameters. Response R is measured as a ratio of relative fluxes $R = \frac{I}{I_0}$. The full code blocks are included in the appendix.

2.1.1 Spectrum Generator

Due to the aforementioned complexity of simulating a stellar spectrum, SME includes a large number of input parameters. These include:

- **teff:** Effective temperature of the star, in Kelvin.
- **logg:** Surface gravity of the star, in log base 10 of CGS units.
- **monh:** Overall metallicity of the star, in log base 10 relative to the Sun's abundances.
- **vmic:** Microturbulence velocity in km/s.
- **vmac:** Macroturbulence velocity in km/s. Includes the star's rotational velocity.

SME also requires a provided set of abundances. Below are constant values set for some of the above parameters across all synthetic spectra. Figure 2.1 provides a sample range of one such synthetic spectrum.

Table 2.1: Table of fixed input variables.

Parameter	**Value**	**Unit**
teff	5500	K
vmic	1.0	km/s
vmac	2.0	km/s
Spectral Resolution $\Delta\lambda$	0.006	Å

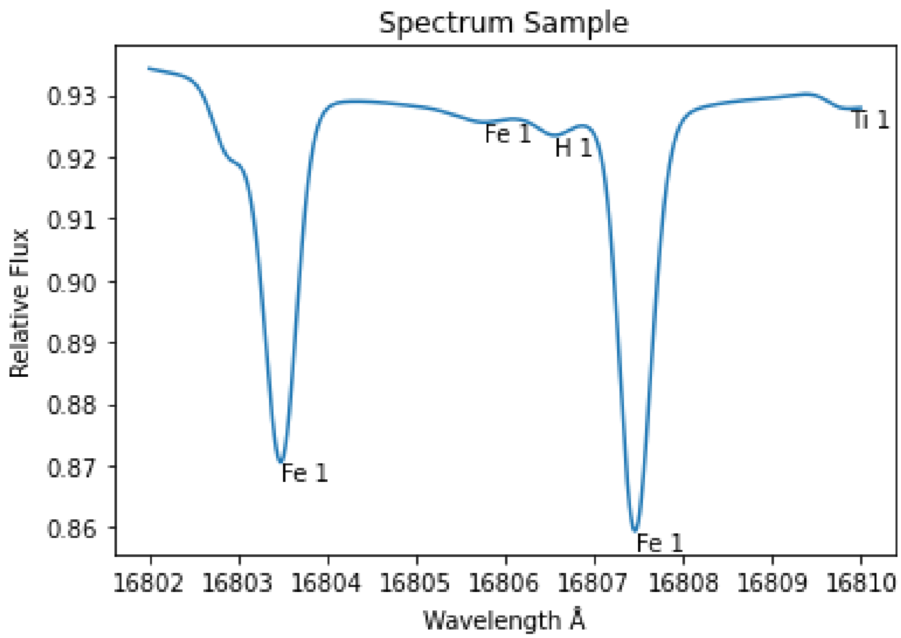

Figure 2.1: A sample of two prominent Fe I absorption lines as well as nearby features. The spectrum was created with the Spectrum Generator code file in Appendix B1. Peaks are annotated with **scipy.findpeaks**.

Microturbulence values are chosen as typical values for dwarf stars in reference to Bensby et al. (2013). Macroturbulence serves simply as a peak broadening parameter, with negligible changes to a line's equivalent width, and so is set arbitrarily.

SME also models the shape of the stellar atmosphere, and is programmed to accommodate for several different models. This thesis makes use of a MARCS model atmosphere as detailed in Gustafsson et al. (2008), with plane-parallel geometry.

2.1.2 Spectrum Reader

With a variety of different spectra created, the Spectrum Reader program selects a pair of spectrum files that vary by a particular dex in $\log g$ and divides them by each other, creating the so-called response curve. Once generated, it runs **scipy.find_peaks** to locate the regions in the spectrum most sensitive to the independent variable based on a chosen input threshold. Using the information from the linelist it then associates a peak in the response curve to a particular atomic or molecular species.

Depending on the threshold, this can produce hundreds, or even thousands, of response peaks—many of which are separate instances of the same changing peak in the original spectrum. The task then becomes to represent this wealth of information in a clear and concise manner, with a particular focus on peaks that show a strong response to $\log g$.

Additionally, this portion of the code is the one responsible for producing the plots in

Section 3, as well as adding Gaussian noise to a spectrum after loading for Section 3.3. When generating the response curves in such a way, unique noise arrays are generated for each spectrum to adequately simulate separate instances of observation.

Chapter 3

Results and Discussion

3.1 An Overview

At a constant T_{eff} of 5500K, the code found a total of 78 peaks sensitive to a change of 0.25 dex in $\log g$, and displaying a response R of at least 2% within the wavelength range. This minimum is chosen for two reasons: one, to select for peaks with a noticeably prominent response to changes in $\log g$; two, to exclude smaller peaks that would likely be undetectable in noisy data. The properties of sensitive peaks, including response strength, wavelength, and species are shown in Figures 3.1 to 3.3, for varying minimum R. Refer to Appendix A for histograms of response peaks for 0.5 and 1.0 dex in $\log g$.

Figure 3.1 arranges all peaks in the response curve that meet the 2% threshold by their response strength. Different metallicities are shown as different colors. Notable here are the large number of weak responses, as well as the significant number of low-metallicity responses. As perhaps expected, most sensitive peaks exhibited only a minor, yet detectable, change in $\log g$, with the strongest responses between 3% and 4%. In this case, the difference between a "strong" and a "weak" response is quite minor.

Figure 3.2 arranges these peaks by species. Notable here is the significant presence of Si I peaks at low metallicity, which is unexpected. There are also more Fe I peaks at [Fe/H] $= -1.0$ than at higher metallicities, which is equally unexpected. At [Fe/H] $= 0.0$ and [Fe/H] $= 0.3$ Mg I accounts for a significant plurality of peaks sensitive to $\log g$. There is also a notable number of H I peaks, with similar numbers present for each metallicity.

It must be noted here that, according to spectroscopic theory, an Fe I peak should not be sensitive to changes in $\log g$. The fact that the code marked 7 peaks as such is evidently an error, likely a result of a linelist entry being too close to the one truly responsible for the absorption peak. This is thus an error in my results.

Figure 3.3 arranges these peaks by their wavelength. Notable here is how most of the peaks are clustered at longer wavelengths, as well as the large number of sensitive peaks at [Fe/H] $= -1.0$ at these longer wavelengths. Their number roughly appears to increase with increasing wavelength. This trend is also followed by sensitive peaks at [Fe/H] $= 0.0$. Though there are also a larger number of sensitive peaks at higher wavelengths for [Fe/H] = 0.3, the disparity is not as striking, and the prior trend cannot be established conclusively.

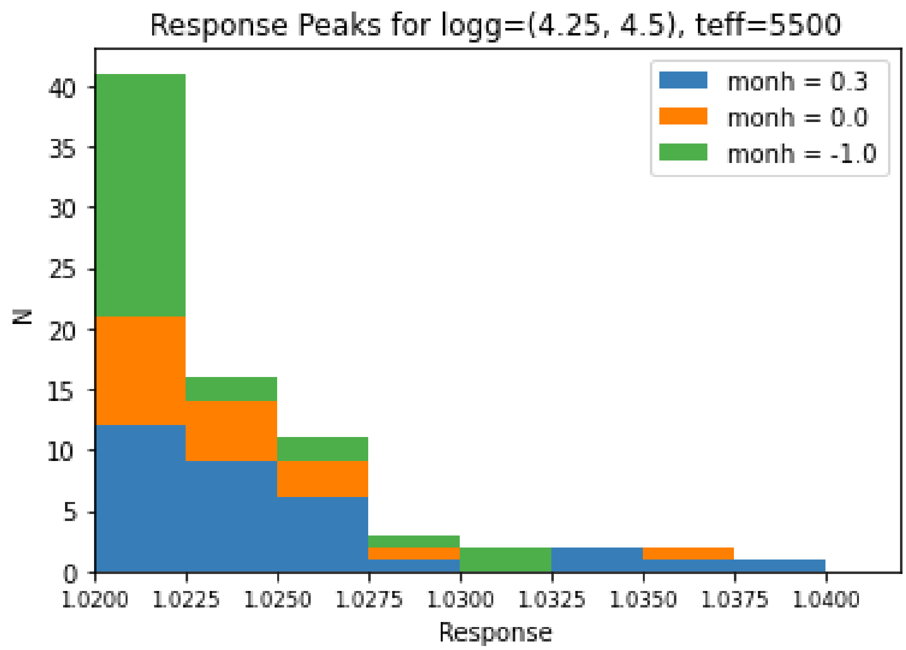

Figure 3.1: Histogram of all response peaks with $R > 2\%$ for 0.25 dex in $\log g$, sorted by R.

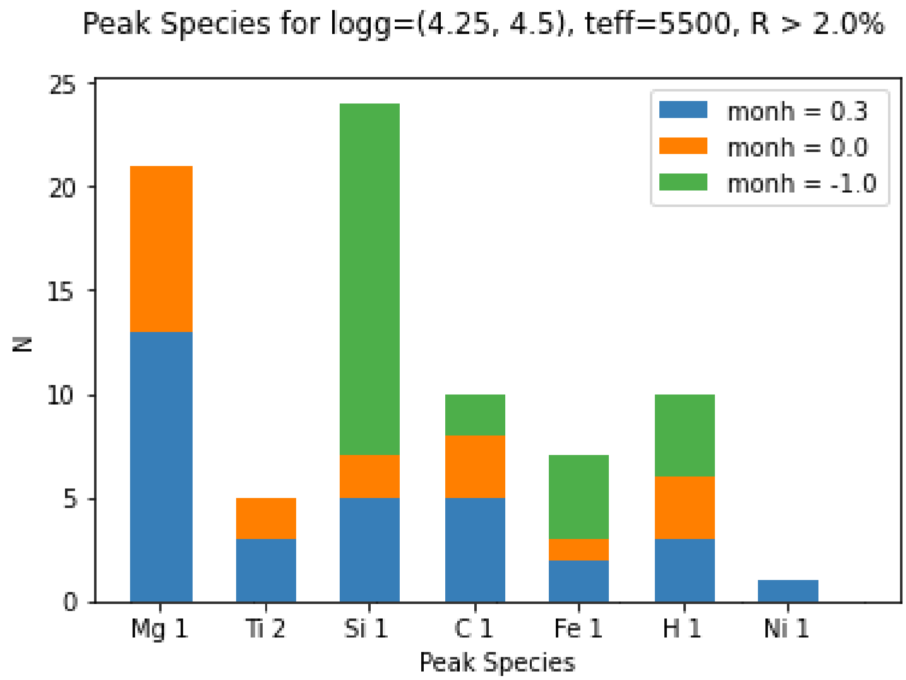

Figure 3.2: Histogram of all response peaks with $R > 2\%$ for 0.25 dex in $\log g$, sorted by species.

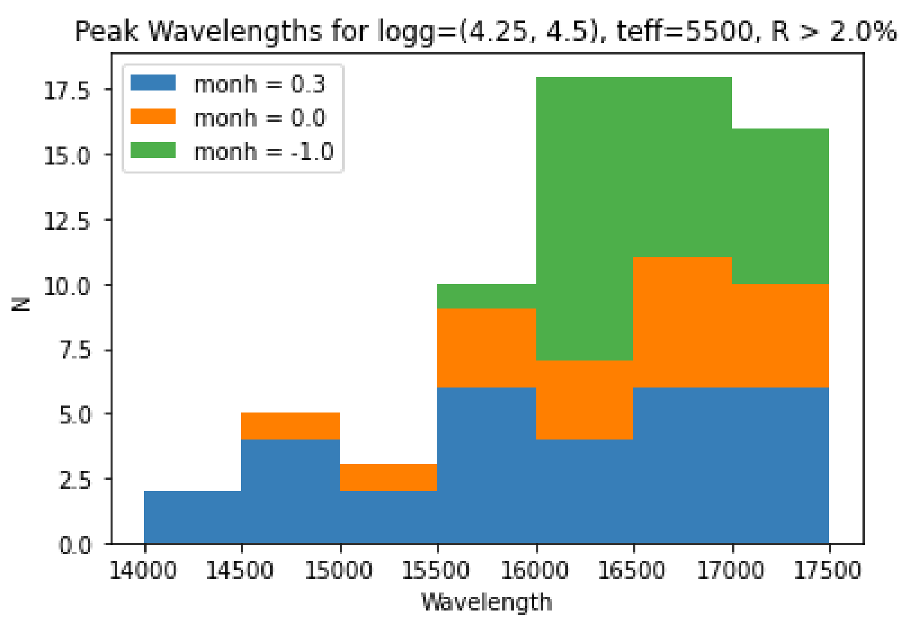

Figure 3.3: Histogram of all response peaks with $R > 2\%$ for 0.25 dex in $\log g$, sorted by wavelength.

We immediately notice some noteworthy trends. For 0.25 dex the majority of the sensitive spectral lines are concentrated at the latter half of the wavelength range. Furthermore, the majority of these peaks exhibit a weak sensitivity to $\log g$. A near-majority of the sensitive peaks result from C I transitions, with significant samples of other metals including Mg I, Fe I, and Ti II, among others, at [Fe/H] values of 0.0 and 0.3. We also notice an intriguing sample of hydrogen and magnesium peaks at [Fe/H] = −1.0. These hydrogen peaks also exhibit a particularly strong response at 1 dex.

A representative sample of these peaks is closely examined in the following sections.

3.2 Selected Peaks

Four peaks are selected for further study: Mg I, Fe I, Si I, and H I. The Si I peak will be used to examine Gaussian noise in Section 3.3. In the following plots, the left side shows response peaks for different dex in $\log g$, and the right side shows the spectrum itself for different values in $\log g$.

Figure 3.4 shows a Mg I peak at 14306.9Å, and its corresponding response in four different dex in $\log g$. While not important to our results, Figure 3.4 provides a perfect example of a peak that is *not* sensitive to $\log g$ next to one that *is*. The deep absorption peak is essentially unchanged in intensity for all four values of $\log g$. Figure 3.5 shows a supposed Fe I peak at 14631.7Å, and its corresponding response in four different dex in $\log g$. As mentioned in the previous section, Fe I is theoretically not sensitive to $\log g$; thus, the code marking this peak as such is an error. Figure 3.6 shows a Si I peak at 16129.0Å, and its corresponding response in four different dex in $\log g$. Figure 3.7 shows a H I peak at 16806.5Å, and its corresponding response in four different dex in $\log g$. It has a particularly extended wavelength range, as the peak wings are much more prominent than for the metallic peaks.

Table 3.1: Table of selected representative peaks, with references from VALD's list of linelist references.

Species	**Wavelength (Å)**	**Reference**
Mg 1	14306.9	Kurucz & Peytremann (1975)
Fe 1	14631.7	Kurucz (2014)
Si 1	16129.0	Kurucz (2007)
H 1	16806.5	Kramida (2010)

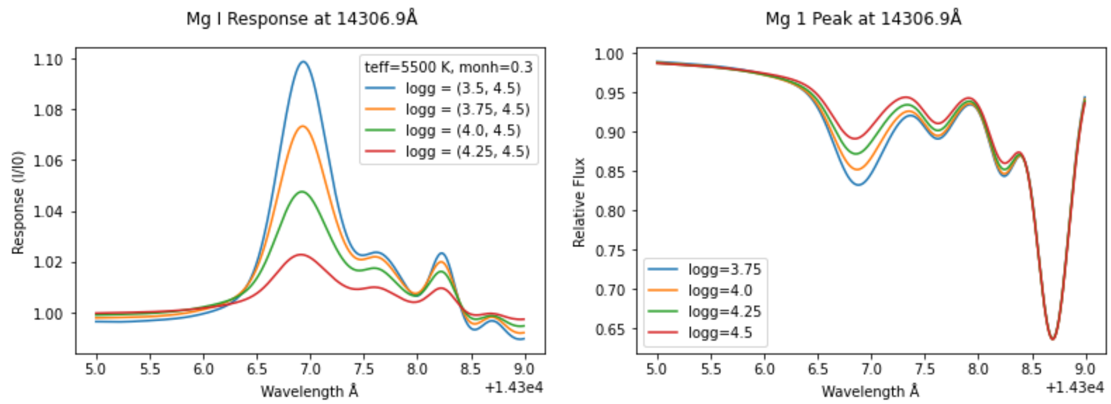

Figure 3.4: Mg I Response for 1, 0.75, 0.5, and 0.25 dex in $\log g$, with synthetic spectra visually representing various steps of change.

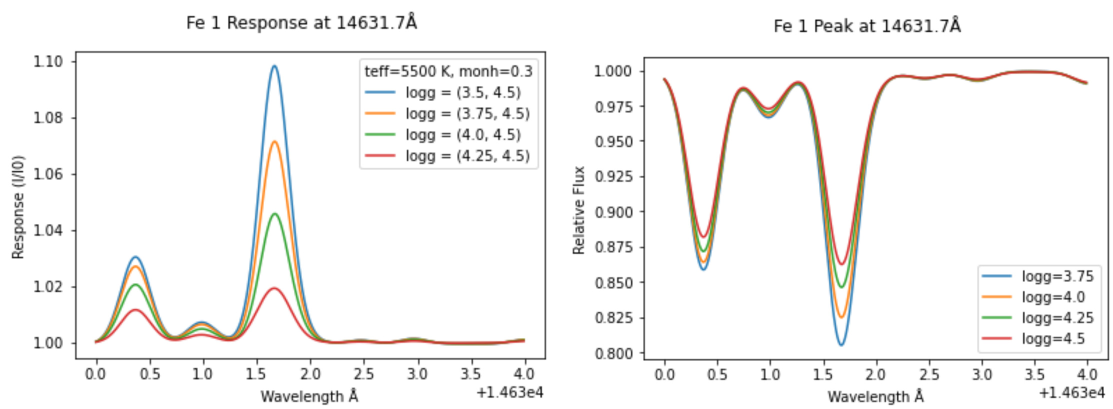

Figure 3.5: Fe I Response for 1, 0.75, 0.5, and 0.25 dex in $\log g$, with synthetic spectra visually representing various steps of change.

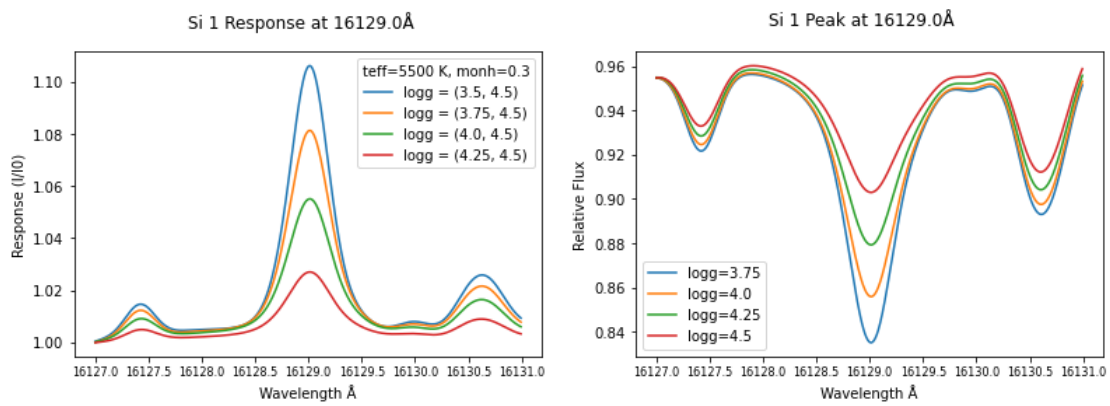

Figure 3.6: Si I Response for 1, 0.75, 0.5, and 0.25 dex in $\log g$, with synthetic spectra visually representing various steps of change.

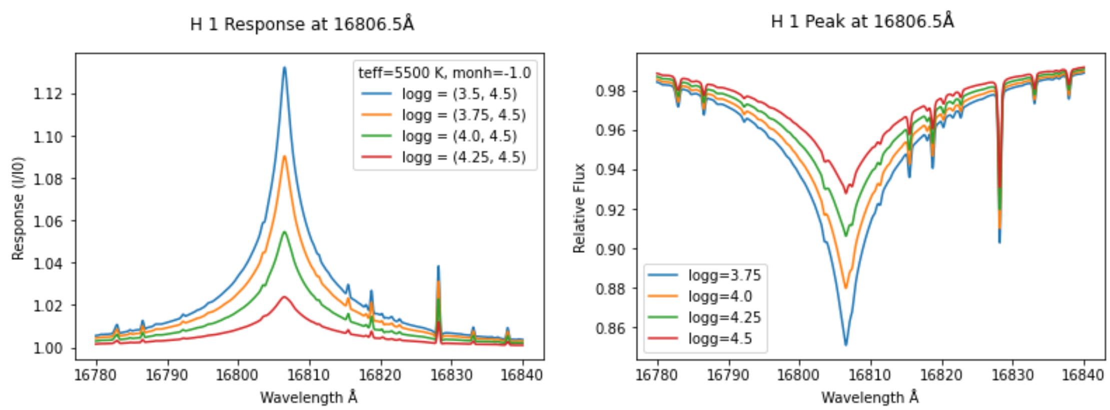

Figure 3.7: H I Response for 1, 0.75, 0.5, and 0.25 dex in $\log g$, with synthetic spectra visually representing various steps of change.

3.3 Artificial Noise

We now add varying intensities of Gaussian noise to the selected peaks of the previous section. Three S/N values, simulating high, medium, and low noise are chosen: 30, 50, and 100 per pixel respectively. The noise arrays are added to the base spectra before the response curve is created, resulting in significant magnification of random noise. In Figures 3.8 and 3.9, the peak structure is only vaguely recognizable when compared to Figure 3.6; without the unmodified spectrum as reference, it would be almost impossible to discern the response peak from the surrounding continuum. In Figure 3.10, meanwhile, the response peak can indeed be identified from the continuum, and its amplitude can be somewhat discerned.

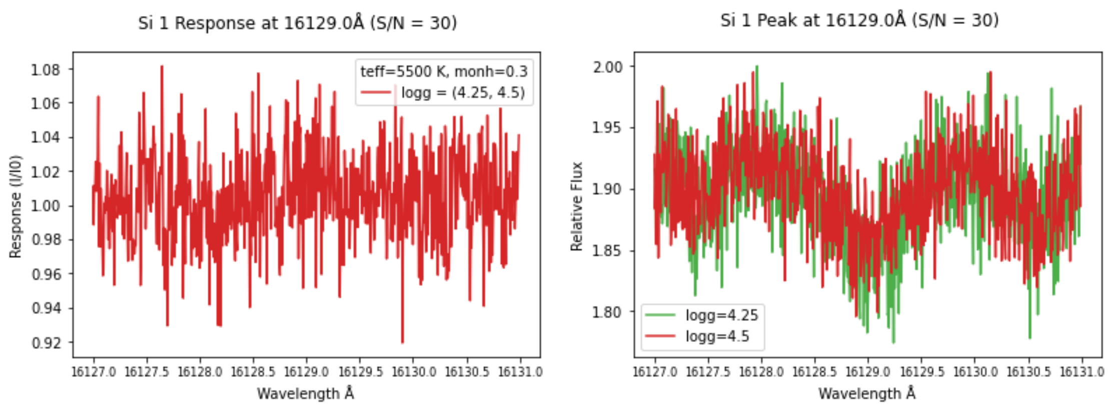

Figure 3.8: Si I response and spectra with S/N = 30. Notice how the previously clear response is rendered almost unrecognizable without prior signal processing.

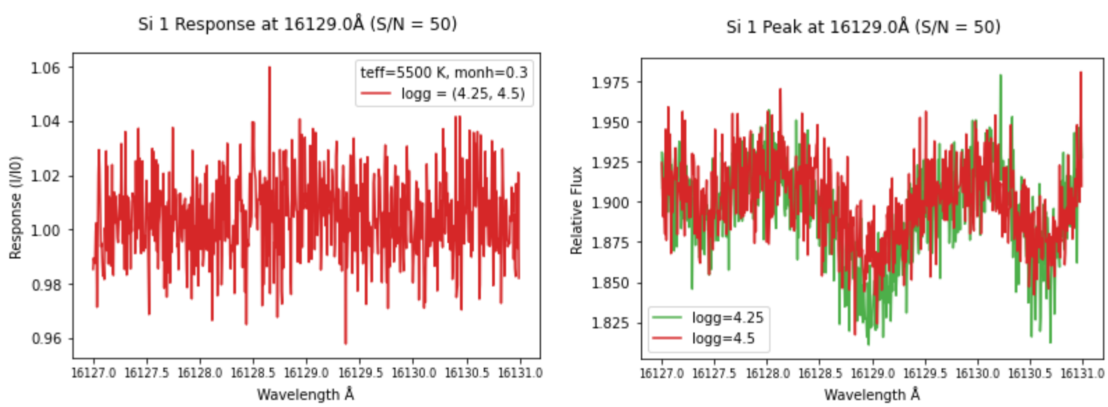

Figure 3.9: Si I response and spectra with S/N = 50. Though the noise in the response curves is noticeably reduced, the peak structure is still far from visible.

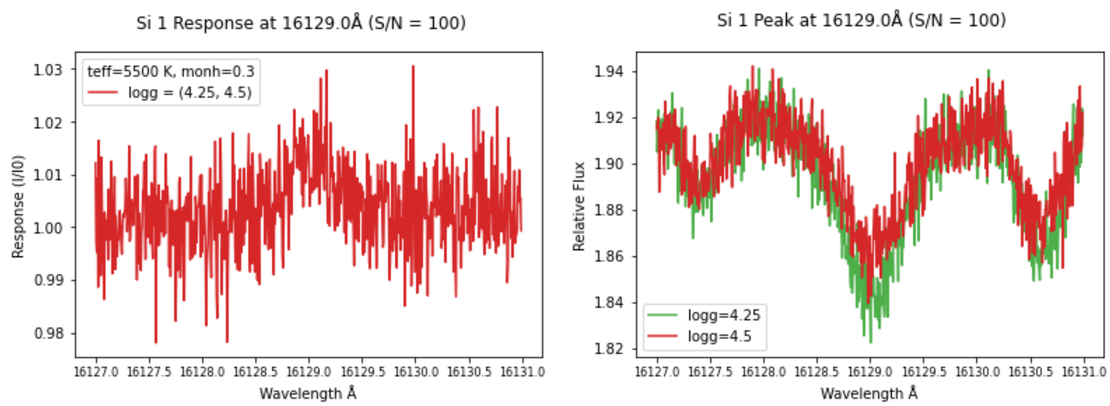

Figure 3.10: Si I response and spectra with S/N = 100. The location of the peak can finally be discerned, but its exact amplitude from the continuum is ambiguous at best.

3.4 Discussion

The first result we can glean is an encouraging one: there appears to be no shortage of peaks in the H-band that are sensitive to changes in surface gravity. Particularly, the abundance of sensitive Mg I and Si I peaks provide a wide sample for comparison and calibration. As seen in Figures 3.4 and reffig: Si I, a 0.25 dex change in $\log g$ produces a response of $\approx 2\%$ in both peaks. There are further options as shown in Figures 3.1, A.1, and A.4, particularly the notable H I peaks at the end of the wavelength range for low-metallicity stars. However, as 3.1 illustrates, only a small number of these peaks exhibit a response greater than 2%. Robust observational equipment will thus be necessary to distinguish the vast majority of these responses in real observations.

The primary obstacle will be the quality of the observed spectrum, which is typified quite plainly in Section 3.3. Though the shape of the original spectrum can still be identified in Figures 3.8, 3.9, and 3.10, their corresponding responses vary from entirely unrecognizable at low S/N to having an identifiable peak structure at high S/N. This results from how the response function is constructed as a ratio between two spectra. As variations on both spectra are random, regions that would have a small or negligible response in Figure A.8 could have an elevated response in any of the figures in Section 3.3. The opposite is also true: regions with response peaks could have their response reduced. These combined are particularly clear in Figure 3.10; the minor response peaks in Figure A.8 are effectively lost, and the primary peak at 15852.6Å can barely be recognized. This highlights the need for robust signal processing of the observed spectra, both for iteration with SME's synthetic spectra and for the potential creation of similar response curves.

There is also the matter of the Fe I false-positives. This presents a rather problematic prospect for the rest of the results, for two reasons. For one, if seven peaks in this case were misidentified, there is every reason to believe other peaks were misidentified, too. For another, it shows that, in the current iteration of the code, there is no adequate vetting mechanism to exclude false-positives from the results. Any future investigation along these lines would have to address both of these points to further refine their results, and minimize the instance of such false-positives.

Despite this, we say with some confidence that, for dwarf stars with $T_{eff} \approx 5500$ and $\log g \approx 4.0$, Mg I, Si I, C I and H I absorption peaks in the H-band present a viable observation target for existing infrared observation infrastructure. H I peaks and wings are of particular interest, as they exhibited the strongest responses of the sampled spectral lines. Furthermore, across all sampled dex in $\log g$, a handful of Ti II peaks at solar and higher metallicites were found, potentially allowing for ionization balance studies as is used in the determination of T_{eff}. Such studies would require a much higher sample size, however. The vast majority of sensitive peaks belong to ground-state species.

Indeed, the final result of this project is the uncovering of numerous lines of research. Naturally, it would be of interest to attempt to recover the synthetic spectrum and the response curve after the Gaussian noise is applied, in order to test observational and computational methods with sample data. The aforementioned peak species, and their precise behaviors, are also each worth dedicated research, as numerous of these sensitive peaks are bound to appear in the spectra of Bulge dwarfs. Naturally, this theoretical treatment will eventually need to be put to the test with real observations. Before these are approved, however, it pays to have a robust hypothesis and clear expected behaviors for a specific sample of spectral lines in mind, such that any gaps in the models can be quickly identified. For the moment, this uncovering shall suffice.

3.5 Conclusion

Every project forms a piece of a larger whole—and though this one inhabits a rather humble corner, it is still one with importance. There were numerous points during this project which necessitated a reduction of scope; originally, the project aimed to compare this theoretical approach with fresh observations of microlensed bulge dwarfs, and furthermore would have examined response curves in T_{eff} and metallicity. These reductions, however, allowed for a more focused investigation on surface gravity sensitivity than would have been otherwise possible. Furthermore, the absence of observational data allowed this thesis to fully embrace its theoretical and simulational character, rather than need to elaborate a hypothesis from scratch and test it within the same work. In this regard, the limitations encountered during the elaboration of this thesis proved beneficial.

In summary, this work has determined the presence of numerous pressure sensitive spectral lines in H-band wavelengths, using models of stellar atmospheres parameterized to model the known properties of dwarf stars in the galactic center. As more observational studies are performed on this enigmatic population, these models and parameter ranges can be further refined, thus improving the determination of stellar ages, with multiple implications in the study of our galaxy's history. With the above findings in mind, we recommend future observations with the CRIRES spectrograph to make use of wavelength setting H1582, as it is slanted the most towards longer wavelengths out of the four available H-band settings. This assessment is made in reference to the latest available version of the CRIRES User Manual at time of writing (Period 114, Phase 2). Please ensure to use the most recent version of the manual on the European Southern Observatory's website (`https://www.eso.org/sci/facilities/paranal/instruments/crires/doc.html`). CRIRES data is based on the prior papers by Kaeufl et al. (2004), Arsenault et al. (2014), and Dorn et al. (2023).

With this said, we conclude this thesis.

Bibliography

Arsenault, R., Paufique, J., Kolb, J., et al. 2014, The Messenger, 156, 2

Barbuy, B., Chiappini, C., & Gerhard, O. 2018, ARA&A, 56, 223

Bensby, T., Feltzing, S., Gould, A., et al. 2017, A&A, 605, A89

Bensby, T., Feltzing, S., Johnson, J. A., et al. 2010, A&A, 512, A41

Bensby, T., Yee, J. C., Feltzing, S., et al. 2013, A&A, 549, A147

Demarque, P., Woo, J.-H., Kim, Y.-C., & Yi, S. K. 2004, The Astrophysical Journal Supplement Series, 155, 667

Dorn, R. J., Bristow, P., Smoker, J. V., et al. 2023, Astronomy amp; Astrophysics, 671, A24

Gonzalez, O. A., Rejkuba, M., Zoccali, M., et al. 2012, AA, 543, A13

Gray, D. F. 2022, The Observation and Analysis of Stellar Photospheres (Cambridge University Press)

Gustafsson, B., Edvardsson, B., Eriksson, K., et al. 2008, A&A, 486, 951

Herbst, E. 1995, Annual Review of Physical Chemistry, 46, 27

Kaeufl, H.-U., Ballester, P., Biereichel, P., et al. 2004, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 5492, Ground-based Instrumentation for Astronomy, ed. A. F. M. Moorwood & M. Iye, 1218–1227

Kramida, A. E. 2010, Atomic Data and Nuclear Data Tables, 96, 586

Kurucz, R. L. 2007, Robert L. Kurucz on-line database of observed and predicted atomic transitions

Kurucz, R. L. 2014, Robert L. Kurucz on-line database of observed and predicted atomic transitions

Kurucz, R. L. & Peytremann, E. 1975, SAO Special Report, 362, 1, (KP)

Nandakumar, G., Ryde, N., Casagrande, L., & Mace, G. 2023, A&A, 675, A23

Nandakumar, G., Ryde, N., Mace, G., et al. 2024, ApJ, 964, 96

Ralchenko, Y., Kramida, A., Reader, J., & NIST ASD Team. 2010, NIST Atomic Spectra Database (ver. 4.0.0), [Online].

Stelter, R. D. & Eikenberry, S. S. 2020, Extinction at the Galactic Center Using Near- and Mid-infrared Broadband Photometry: A Twist on the Rayleigh-Jeans Color Excess Method

Wyse, R. F. G., Gilmore, G., & Franx, M. 1997, ARA&A, 35, 637

Appendix A

Additional Plots

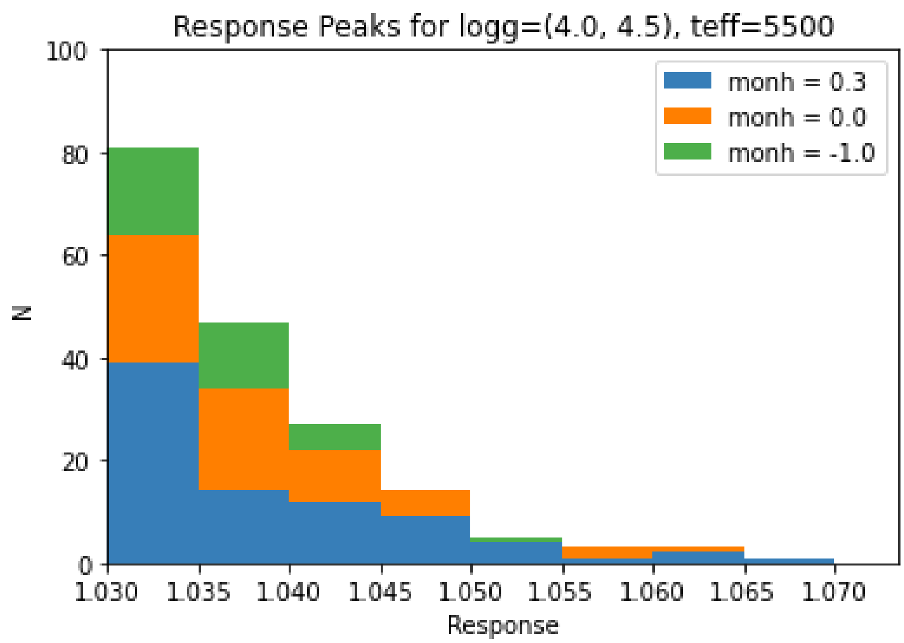

Figure A.1: Histogram of all response peaks with $R > 2.5\%$ for 0.5 dex in $\log g$, sorted by R. Notable here are the large number of weak responses.

Table A.1: Table of selected representative peaks, with references from VALD's list of linelist references.

Species	**Wavelength (Å)**	**Reference**
C 1	14420.1	Ralchenko et al. (2010)
C 1	15852.6	Ralchenko et al. (2010)

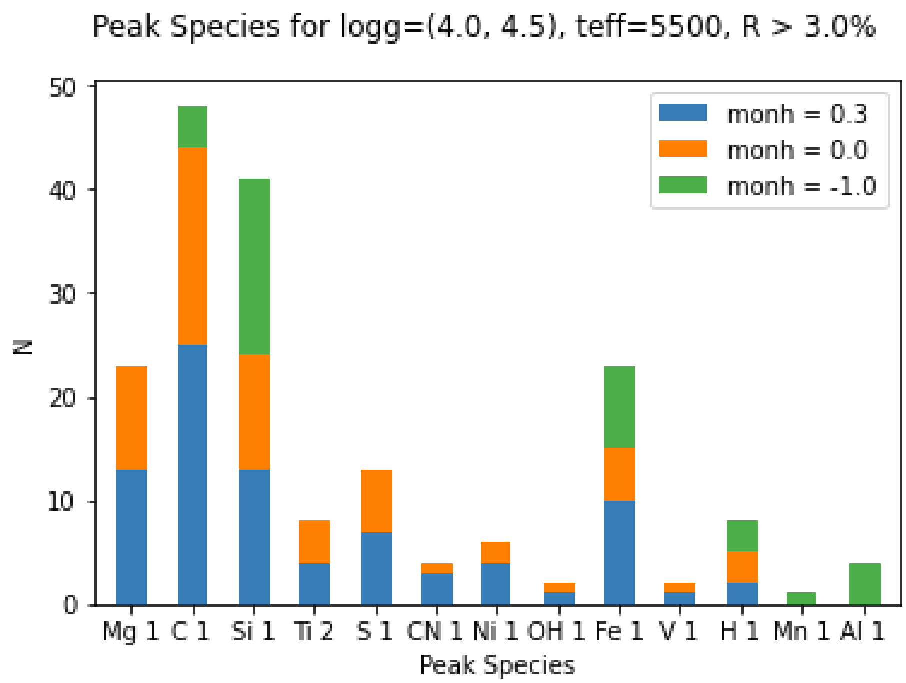

Figure A.2: Histogram of all response peaks with $R > 2.5\%$ for 0.5 dex in $\log g$, sorted by species. Notable here are the large number of C I, Si I, and Fe I absorption peaks that are sensitive to $\log g$.

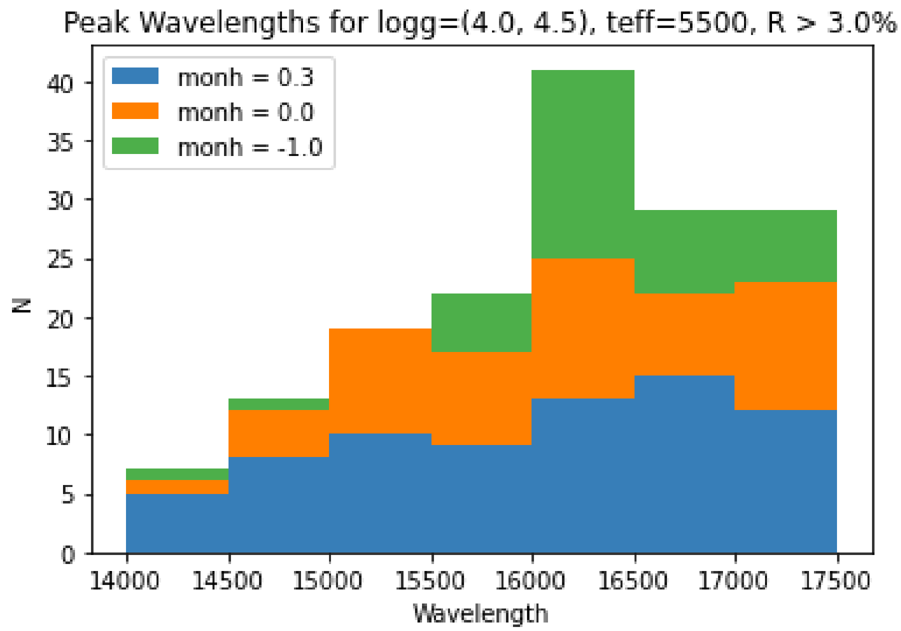

Figure A.3: Histogram of all response peaks with $R > 2\%$ for 0.5 dex in $\log g$, sorted by wavelength. Notable here is how the peaks are clustered near the middle of the wavelength range, with a roughly decreasing number of peaks with decreasing metallicity.

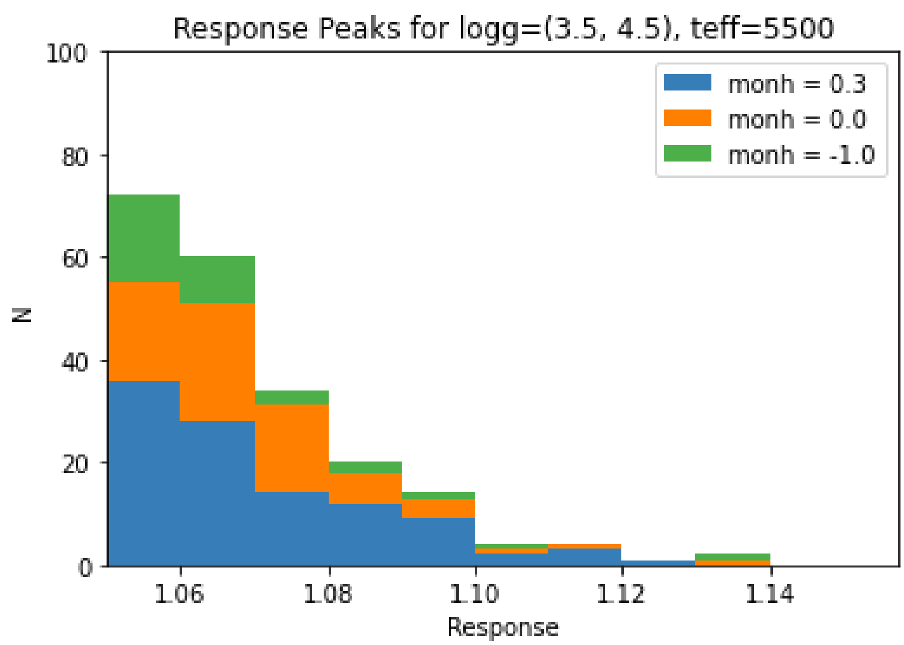

Figure A.4: Histogram of all response peaks with $R > 5\%$ for 1 dex in $\log g$, sorted by R. Notable again is the rapidly diminishing number of peaks as R increases, across all metallicities.

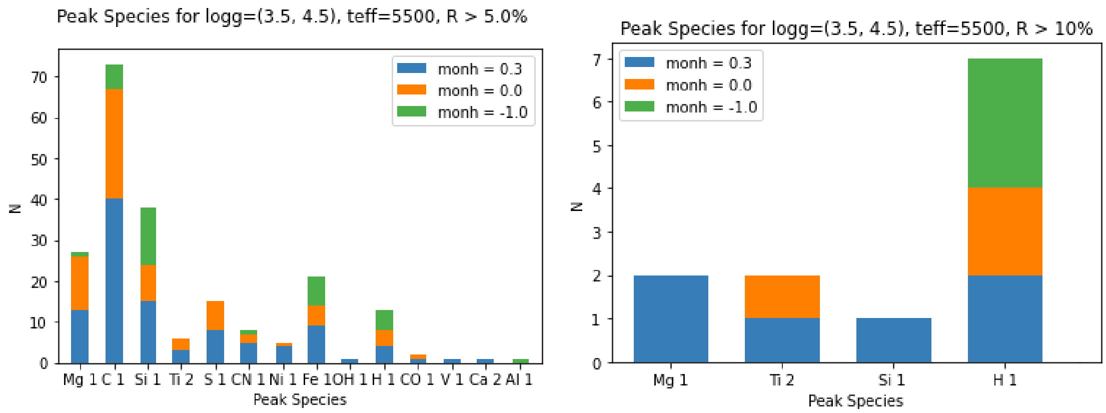

Figure A.5: Histograms of all response peaks for $R > 5\%$ on the left and $R > 10\%$ on the right for 1 dex in $\log g$, sorted by species. Notable here is the large number of C I peaks for R between 5% and 10%, whereas for $R > 10\%$, H I peaks are the majority. Notable also is how the number of H I peaks above the threshold *increases* with decreasing metallicity.

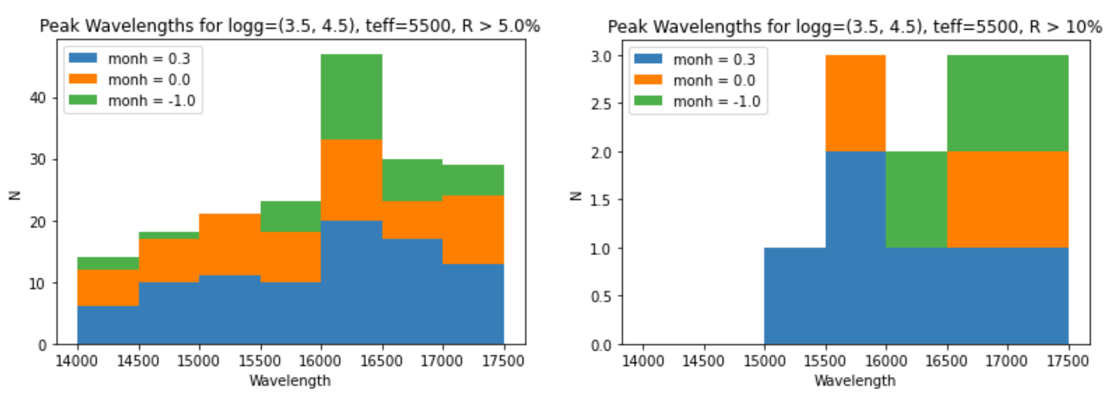

Figure A.6: Histograms of all response peaks for $R > 5\%$ on the left and $R > 10\%$ on the right for 1 dex in $\log g$, sorted by wavelength. Notable here is the concentration of peaks at the beginning, middle, and end of the wavelength range for weak responses, with responses of $R > 10\%$ tending towards higher wavelengths.

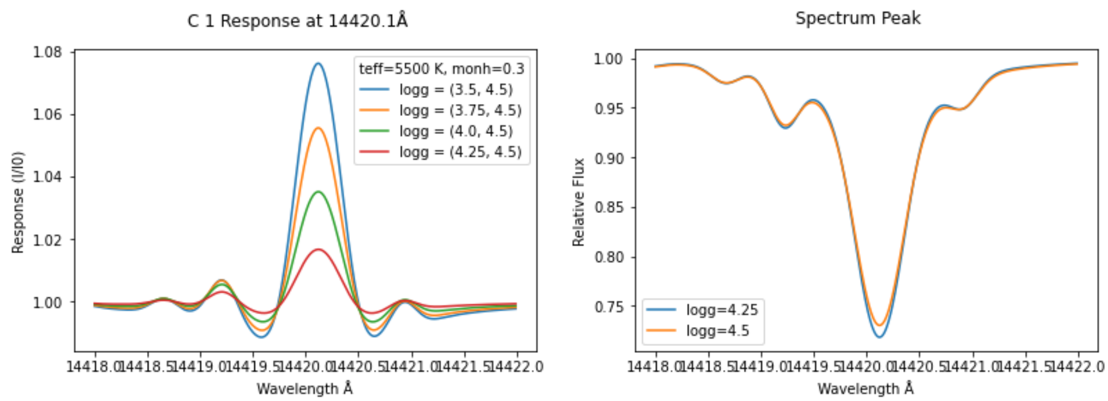

Figure A.7: C I response for 1, 0.75, 0.5, and 0.25 dex in $\log g$, with synthetic spectra visually representing 1 dex of change.

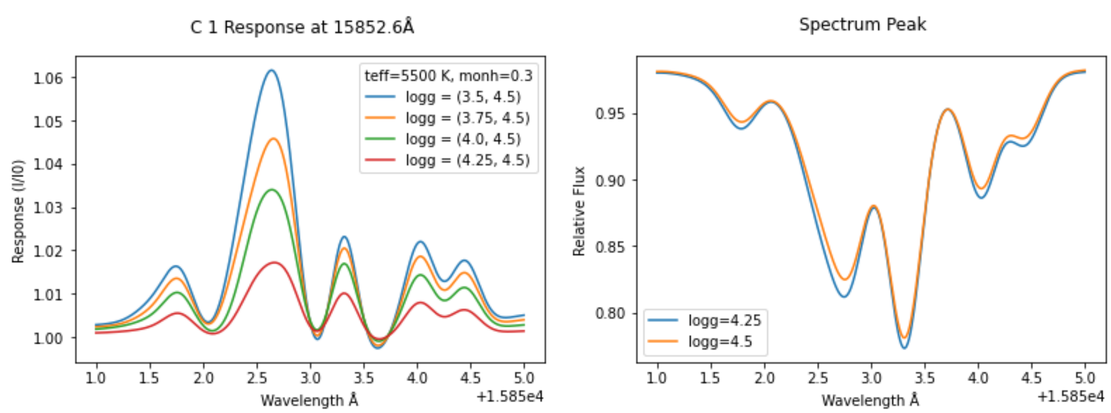

Figure A.8: C I response for 1, 0.75, 0.5, and 0,25 dex in $\log g$, with synthetic spectra visually representing 1 dex of change. Notable here is the presence of a peak adjacent to the C I peak of interest with a much lower sensitivity to changes in $\log g$.

Appendix B

Code

B.1 Spectrum Generator

```
import numpy as np
import matplotlib.pyplot as plt
from pysme.sme import SME_Structure as SME_Struct
from pysme.synthesize import synthesize_spectrum

from pysme.abund import Abund

# Loading spectrum files
def load_file(logg, teff, monh):
    # global file
    file = np.load(f"/home/santiago/Thesis Spectra/{logg}, {teff}, {monh}/spectrum_{teff}_{logg}_{monh}.npy")
    return file

# Find nearest index function
def find_nearest_idx(array, value):
    array = np.asarray(array)
    index = (np.abs(array - value)).argmin()
    return index

# Add noise to a spectrum, given a signal to noise ratio (SNR)
def add_noise(SNR, signal):
    signal_mean = np.mean(signal)
    standard_deviation = signal_mean/SNR
    noise = np.random.normal(signal_mean, standard_deviation, len(signal))
    signal = signal + noise
    return signal

# Linelist creation
from pysme.linelist.vald import ValdFile
vald = ValdFile("/home/santiago/Example/Infrared.lin") # Atomic data for wavelength range
```

```
lambda_array = vald.wlcent

# NOTE: n is the number of points for linspace. Use 600000.
def create_spectrum(n, logg, teff, monh):

    sme = SME_Struct()

    # Definition of core variables

    lambda_start = 14300
    lambda_end = 18001

    sme.wave = np.linspace(lambda_start, lambda_end, n)
    sme.linelist = vald

    sme.teff, sme.logg = teff, logg

    sme.abund = Abund.solar()
    sme.ipres, sme.iptype = 50000, "gauss" # Resolving power of
   instrument
    sme.monh = monh # metallicity; scales with given abundances.

    # Microturbulence, macroturbulence, rotational velocity
    sme.vmic, sme.vmac, sme.vsini = 1.0, 2.0, 0.1

    # SME comes with a few model atmospheres see Atmosphere section
    sme.atmo.source = "marcs2012p_t1.0.sav"
    sme.atmo.method = "grid"
    sme.atmo.geom = "PP"

    # Setting mu
    nmu = 7
    sme.mu = np.flipud(np.sqrt(0.5*(2*np.arange(nmu)+1)/nmu))

    # Create synthetic spectrum!
    sme = synthesize_spectrum(sme)

    # Save spectrum to file
    spectrum = np.array([sme.wave[0], sme.synth[0]])
    np.save(f"spectrum_{teff}_{logg}_{monh}", spectrum)
    file = np.load(f"spectrum_{teff}_{logg}_{monh}.npy")

    return file

def graph_spectrum(lambda_start, lambda_end, SNR):

    # Choosing parameters of spectrum to graph
    print('CHOOSE YOUR PARAMETERS:')

    logg = input("Set logg value:")
    teff = input("Set teff value:")
```

```
    monh = input("Set monh value:")

    # loading correct spectrum:
    spectrum = load_file(logg, teff, monh)

    # Add noise to spectrum
    if SNR != 0:
        spectrum[1] = add_noise(SNR, spectrum[1])

    # Obtain plotting indices
    index_start = find_nearest_idx(spectrum[0], lambda_start)
    index_end = find_nearest_idx(spectrum[0], lambda_end)

    # Plotting
    plt.plot(spectrum[0][index_start:index_end],
             spectrum[1][index_start:index_end])
    plt.xlabel('Wavelength   ')
    plt.ylabel('Relative Flux')
    plt.title(f'Teff = {teff}, logg = {logg}, monh = {monh}')

    return print('Graph Done!')
```

B.2 Spectrum Reader

Note: the written code produces misaligned x-axis labels for the species histograms (Figures 3.2, A.2, and A.5). These were edited manually to align the x-axis labels to their corresponding bar.

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
from matplotlib.ticker import ScalarFormatter

import scipy.signal as scipy
from itertools import combinations
from itertools import product
import sys
sys.path.append("/home/santiago/Example/")
from ResponseClass import *

# Linelist creation
from pysme.linelist.vald import ValdFile
vald = ValdFile("/home/santiago/Example/Infrared.lin") # Atomic data for wavelength range
lambda_array = vald.wlcent

# Find nearest index function
def find_nearest_idx(array, value):
    array = np.asarray(array)
    index = (np.abs(array - value)).argmin()
    return index
```

```

# Loading spectrum files
def load_file(logg, teff, monh):
    file = np.load(f"/home/santiago/Thesis Spectra/{logg}, {teff}, {monh
   }/spectrum_{teff}_{logg}_{monh}.npy")
    return file

# Check length of a spectrum
def check_file_len(logg, teff, monh):
    file = np.load(f"/home/santiago/Thesis Spectra/{logg}, {teff}, {monh
   }/spectrum_{teff}_{logg}_{monh}.npy")
    return len(file[0])

# Add noise to a spectrum, given a signal to noise ratio (SNR)
def add_noise(SNR, signal):
    signal_mean = np.mean(signal)
    standard_deviation = signal_mean/SNR
    noise = np.random.normal(signal_mean, standard_deviation, len(signal)
   )
    signal = signal + noise
    return signal

# Response object dictionary
curve = {}

# Variables
logg_values = np.array([3.0, 3.25, 3.5, 3.75, 4.0, 4.25, 4.5])
teff_values = 5500
monh_values = np.array([-1.0, 0.0, 0.3])


### CREATE RESPONSES ###

def create_responses(thresh, prominence, SNR):

    variables = ["logg"]

    count = 1

    for delta in variables:
        for i, k in product(range(21), range(3)):

            # Define variables
            if delta == "logg":
                delta_values = list(combinations(logg_values, 2))[i]
                control = "teff"
                control_value = teff_values

                logg = [delta_values[0], delta_values[1]]
                teff = teff_values
                monh = monh_values[k]
```

```

            file_1 = load_file(delta_values[0], teff, monh)
            file_2 = load_file(delta_values[1], teff, monh)

        # Add noise to file
        if SNR != 0:
            file_1[1] = add_noise(SNR, file_1[1])
            file_2[1] = add_noise(SNR, file_2[1])

        # Initialize response object
        curve[f"{delta}={delta_values}, {control}={control_value}, monh={monh}"] = Responses(delta, delta_values, control, control_value, monh)

        # Create response
        wavelengths = file_1[0]
        response = np.divide(file_2[1], file_1[1])
        peaks_indices = scipy.find_peaks(response, height=thresh) # an array of indices of peaks
        # peaks_indices = scipy.find_peaks(response, height=thresh, prominence=prominence) # an array of indices of peaks

        peaks = []
        species = []
        responses = []
        references = []

        for n in peaks_indices[0]:

            # Populating of peak and species lists
            peak_wavelength = np.round(wavelengths[n], 1)
            peaks.append(peak_wavelength)

            peak_index = find_nearest_idx(vald.wlcent, wavelengths[n])
            species.append(vald.species[peak_index])
            references.append(vald.reference[peak_index])

            peak_response = response[n]
            responses.append(peak_response)

        # Assign values to response object
        curve[f"{delta}={delta_values}, {control}={control_value}, monh={monh}"].wavelength = wavelengths
        curve[f"{delta}={delta_values}, {control}={control_value}, monh={monh}"].response = response
        curve[f"{delta}={delta_values}, {control}={control_value}, monh={monh}"].peak_species = species
        curve[f"{delta}={delta_values}, {control}={control_value}, monh={monh}"].peak_values = peaks
```

```
            curve[f"{delta}={delta_values}, {control}={control_value}, monh={monh}"].peak_responses = responses

            print(count)
            count += 1

        print(f'{delta} done!')

    return curve


### GRAPH RESPONSES ###

def graph_responses(lambda_start, lambda_end):

    peak_species = []
    peak_values = []
    peak_keys = []

    all_peak_species = []
    all_peak_values = []
    all_peak_keys = []

    # Flagging all unique peaks in all response curves
    for key in curve:

        # Check if the response curve has peaks that passed thresh
        if not curve[key].peak_species:
            print("Curve has no peaks that meet threshold.")
        else:
            for i in range(len(curve[key].peak_species)):

                # Add peak, if not already in list, and if within bounds
                if curve[key].peak_values[i] not in peak_values and curve[key].peak_values[i] > lambda_start and curve[key].peak_values[i] < lambda_end:

                    peak_species.append(curve[key].peak_species[i])
                    peak_values.append(curve[key].peak_values[i])
                    peak_keys.append(key)

                # Populate peaks_to_plot with all necessary info

                all_peak_species.append(curve[key].peak_species[i])
                all_peak_values.append(curve[key].peak_values[i])
                all_peak_keys.append(key)

    print(f"{len(peak_species)} unique peaks found!")

    # Sort lists by ascending wavelength
```

```
    peak_values, peak_species, peak_keys = (list(t) for t in zip(*sorted(zip(peak_values, peak_species, peak_keys))))
    all_peak_values, all_peak_species, all_peak_keys = (list(t) for t in zip(*sorted(zip(all_peak_values, all_peak_species, all_peak_keys))))

    # Place lists into output matrix
    peaks_matrix = [peak_species, peak_values, peak_keys]
    peaks_to_plot = [all_peak_species, all_peak_values, all_peak_keys]

    # Plotting of peaks
    for i in range(len(peaks_matrix[1])):
        figure_ID = f'{peaks_matrix[1][i]}, {peaks_matrix[0][i]}'

        fig, axs = plt.subplots(3, 1, num=figure_ID)
        fig.suptitle(f'{peaks_matrix[0][i]} peak response at {peaks_matrix[1][i]}    ')
        plot_offset = 500

        main_peak_wavelength = peaks_matrix[1][i]

        for j in range(len(peaks_to_plot[1])):

            peak_wavelength = peaks_to_plot[1][j]

            # Fishing out curve parameters from library
            monh = curve[peaks_to_plot[2][j]].monh
            if curve[peaks_to_plot[2][j]].delta == "teff":
                teff = curve[peaks_to_plot[2][j]].delta_values
                logg = curve[peaks_to_plot[2][j]].control_value
            if curve[peaks_to_plot[2][j]].delta == "logg":
                logg = curve[peaks_to_plot[2][j]].delta_values
                teff = curve[peaks_to_plot[2][j]].control_value

            wavelengths = curve[peaks_to_plot[2][j]].wavelength
            response = curve[peaks_to_plot[2][j]].response

            if np.isclose(main_peak_wavelength, peak_wavelength) == True:

                peak_index = find_nearest_idx(wavelengths, peaks_to_plot[1][j])

                if monh == -1.0:
                    axs[0].plot(wavelengths[peak_index-plot_offset:peak_index+plot_offset],
                                response[peak_index-plot_offset:peak_index+plot_offset],
                                label = f'logg = {logg}, teff = {teff}, monh = {monh}')
                    plt.xlabel('Wavelength    ')
                    plt.ylabel('Response (I/I0)')

```

```python
                    fig.legend(loc='center left', bbox_to_anchor=(1, 0.5))
                    axs[0].annotate(f'{peaks_matrix[0][i]}',
                                  (peaks_matrix[1][i]+0.5, response[peak_index]-0.02))

                if monh == 0.0:
                    axs[1].plot(wavelengths[peak_index-plot_offset:peak_index+plot_offset],
                              response[peak_index-plot_offset:peak_index+plot_offset],
                              label = f'logg = {logg}, teff = {teff}, monh = {monh}')
                    plt.xlabel('Wavelength   ')
                    plt.ylabel('Response (I/I0)')

                    fig.legend(loc='center left', bbox_to_anchor=(1, 0.5))
                    axs[1].annotate(f'{peaks_matrix[0][i]}',
                                  (peaks_matrix[1][i]+0.5, response[peak_index]-0.02))

                if monh == 0.3:
                    axs[2].plot(wavelengths[peak_index-plot_offset:peak_index+plot_offset],
                              response[peak_index-plot_offset:peak_index+plot_offset],
                              label = f'logg = {logg}, teff = {teff}, monh = {monh}')
                    plt.xlabel('Wavelength   ')
                    plt.ylabel('Response (I/I0)')

                    fig.legend(loc='center left', bbox_to_anchor=(1, 0.5))
                    axs[2].annotate(f'{peaks_matrix[0][i]}',
                                  (peaks_matrix[1][i]+0.5, response[peak_index]-0.02))

    return peaks_matrix

def precision_graph(lambda_start, lambda_end):

    response_keys = []
    cont = "Yes"
    colors = ['#377eb8', '#ff7f00', '#4daf4a', '#d62728']

    # Selecting response curves
    manual_input = input("Manual input? Yes or No: ")

    while cont == "Yes":

```

```
        if manual_input == "Yes":

            response_keys.append(input("Copy and paste string: "))

        else:

            print("Choose your response parameters, in metallicity order: ")

            delta = input("Choose delta variable: ")
            delta_values = input("Set delta values as (1, 2): ")
            control = input("Choose control variable: ")
            control_value = input("Set control value: ")
            monh = input("Set monh value: ")

            response_keys.append(f"{delta}={delta_values}, {control}={control_value}, monh={monh}")

        manual_input = input("Manual input? Yes or No: ")
        cont = input("Add another response? ")

    # Plot response curves

    plt.suptitle(input("Set plot title as string: "))

    for key in response_keys:

        if curve[key].delta == "logg":
            logg = curve[key].delta_values
            teff = curve[key].control_value

        if curve[key].delta == "teff":
            logg = curve[key].control_value
            teff = curve[key].delta_values

        monh = curve[key].monh

        # Obtain plotting indices
        index_start = find_nearest_idx(curve[key].wavelength, lambda_start)
        index_end = find_nearest_idx(curve[key].wavelength, lambda_end)

        plt.plot(curve[key].wavelength[index_start:index_end],
                 curve[key].response[index_start:index_end],
                 label = f'logg = {logg}')

    plt.legend(title=f'teff={teff} K, monh={monh}')

    plt.xlabel('Wavelength   ')
    plt.xticks(fontsize=8)
    plt.ylabel('Response (I/I0)')
```

```python

    return response_keys

def create_response_histograms(logg_start, logg_end, thresh):

    teff = 5500

    # Response Parameters
    delta = 'logg'
    delta_values = (logg_start, logg_end)
    control = 'teff'
    control_value = teff

    # Defining keys
    key_1 = f"{delta}={delta_values}, {control}={control_value}, monh=0.3"
    key_2 = f"{delta}={delta_values}, {control}={control_value}, monh=0.0"
    key_3 = f"{delta}={delta_values}, {control}={control_value}, monh=-1.0"

    # Load peak responses
    peak_responses_1 = curve[key_1].peak_responses
    peak_responses_2 = curve[key_2].peak_responses
    peak_responses_3 = curve[key_3].peak_responses

    # Load peak species for minimum% thresh
    peak_species_1_five = []
    peak_species_2_five = []
    peak_species_3_five = []

    for i in range(len(curve[key_1].peak_species)):
        if curve[key_1].peak_responses[i] > thresh:
            peak_species_1_five.append(curve[key_1].peak_species[i])

    for i in range(len(curve[key_2].peak_species)):
        if curve[key_2].peak_responses[i] > thresh:
            peak_species_2_five.append(curve[key_2].peak_species[i])

    for i in range(len(curve[key_3].peak_species)):
        if curve[key_3].peak_responses[i] > thresh:
            peak_species_3_five.append(curve[key_3].peak_species[i])

    # Load peak species for 10% thresh
    peak_species_1_ten = []
    peak_species_2_ten = []
    peak_species_3_ten = []

    for i in range(len(curve[key_1].peak_species)):
        if curve[key_1].peak_responses[i] > 1.10:
            peak_species_1_ten.append(curve[key_1].peak_species[i])
```

```

    for i in range(len(curve[key_2].peak_species)):
        if curve[key_2].peak_responses[i] > 1.10:
            peak_species_2_ten.append(curve[key_2].peak_species[i])

    for i in range(len(curve[key_3].peak_species)):
        if curve[key_3].peak_responses[i] > 1.10:
            peak_species_3_ten.append(curve[key_3].peak_species[i])

    # Load peak lambdas for 5% thresh
    peak_lambdas_1_five = []
    peak_lambdas_2_five = []
    peak_lambdas_3_five = []

    for i in range(len(curve[key_1].peak_values)):
        if curve[key_1].peak_responses[i] > thresh:
            peak_lambdas_1_five.append(curve[key_1].peak_values[i])

    for i in range(len(curve[key_2].peak_values)):
        if curve[key_2].peak_responses[i] > thresh:
            peak_lambdas_2_five.append(curve[key_2].peak_values[i])

    for i in range(len(curve[key_3].peak_values)):
        if curve[key_3].peak_responses[i] > thresh:
            peak_lambdas_3_five.append(curve[key_3].peak_values[i])

    # Load peak lambdas for 10% thresh
    peak_lambdas_1_ten = []
    peak_lambdas_2_ten = []
    peak_lambdas_3_ten = []

    for i in range(len(curve[key_1].peak_values)):
        if curve[key_1].peak_responses[i] > 1.10:
            peak_lambdas_1_ten.append(curve[key_1].peak_values[i])

    for i in range(len(curve[key_2].peak_values)):
        if curve[key_2].peak_responses[i] > 1.10:
            peak_lambdas_2_ten.append(curve[key_2].peak_values[i])

    for i in range(len(curve[key_3].peak_values)):
        if curve[key_3].peak_responses[i] > 1.10:
            peak_lambdas_3_ten.append(curve[key_3].peak_values[i])

    ### PLOT HISTOGRAMS ###

    colors = ['#377eb8', '#ff7f00', '#4daf4a']
    labels = ['monh = 0.3', 'monh = 0.0', 'monh = -1.0']

    # Response Histogram

    plt.figure("fig_response")
```

```
    peak_responses = [peak_responses_1, peak_responses_2, peak_responses_3]

    if logg_start == 4.25:
        bins = np.arange(1, 1.04, 0.0025)
    if logg_start == 4.0:
        bins = np.arange(1, 1.07, 0.005)
    if logg_start == 3.5:
        bins = np.arange(1, 1.16, 0.01)

    plt.hist(peak_responses, bins=bins, stacked=True, color=colors, label=labels)

    plt.xlim(left=thresh)
    if thresh > 1.045:
        plt.ylim(top=100)
    if logg_start == 4.0:
        plt.ylim(top=100)

    plt.xlabel('Response')
    plt.ylabel('N')
    plt.xticks(fontsize=8.5)
    plt.legend()
    plt.title(f'Response Peaks for logg={delta_values}, teff={control_value}')

    # Species Histogram, 5%

    plt.figure("spec_response_5%")

    peak_species_five = [peak_species_1_five, peak_species_2_five, peak_species_3_five]
    print(peak_species_five)

    bins = list(set(sum(peak_species_five, [])))
    print(bins)

    if len(bins) > 0:
        plt.hist(peak_species_five, bins=len(bins), stacked=True, color=colors,
                 label=labels, width=0.5)

        plt.xlabel('Peak Species')
        plt.ylabel('N')
        plt.legend()
        plt.suptitle(f'Peak Species for logg={delta_values}, teff={control_value}, R > {((thresh-1)*100):.1f}%')
    else:
        print("No peaks meet threshold.")
```

```
# Species Histogram, 10%

plt.figure("spec_response_10%")

peak_species_ten = [peak_species_1_ten, peak_species_2_ten, peak_species_3_ten]
print(peak_species_ten)

bins = list(set(sum(peak_species_ten, [])))
print(bins)

if len(bins) > 0:
    plt.hist(peak_species_ten, bins=len(bins), stacked=True, color=colors,
                 label=labels, width=0.5)

    plt.xlabel('Peak Species')
    plt.ylabel('N')
    plt.legend()
    plt.title(f'Peak Species for logg={delta_values}, teff={control_value}, R > 10%')
else:
    print("No peaks meet threshold.")


# Wavelength Histogram, 5%

plt.figure("lambda_response_5%")

peak_lambdas_five = [peak_lambdas_1_five, peak_lambdas_2_five, peak_lambdas_3_five]

plt.hist(peak_lambdas_five, bins=np.arange(14000, 18000, 500), stacked=True, color=colors, label=labels)

plt.xlabel('Wavelength')
plt.ylabel('N')
plt.legend()
plt.title(f'Peak Wavelengths for logg={delta_values}, teff={control_value}, R > {((thresh-1)*100):.1f}%')

# Wavelength Histogram, 10%

plt.figure("lambda_response_10%")

peak_lambdas_ten = [peak_lambdas_1_ten, peak_lambdas_2_ten, peak_lambdas_3_ten]

plt.hist(peak_lambdas_ten, bins=np.arange(14000, 18000, 500), stacked=True, color=colors, label=labels)

```

```
    plt.xlabel('Wavelength')
    plt.ylabel('N')
    plt.legend()
    plt.title(f'Peak Wavelengths for logg={delta_values}, teff={
   control_value}, R > 10%')

    overview_peaks = []

    return overview_peaks


def graph_spectrum(lambda_start, lambda_end, SNR):

    # Choosing parameters of spectrum to graph
    print('CHOOSE YOUR PARAMETERS:')

    logg = input("Set logg value:")
    teff = input("Set teff value:")
    monh = input("Set monh value:")

    print('CHOOSE YOUR OTHER PARAMETERS:')

    logg2 = input("Set logg value:")
    teff2 = input("Set teff value:")
    monh2 = input("Set monh value:")

    print('CHOOSE YOUR OTHER PARAMETERS:')

    logg3 = input("Set logg value:")
    teff3 = input("Set teff value:")
    monh3 = input("Set monh value:")

    print('CHOOSE YOUR OTHER PARAMETERS:')

    logg4 = input("Set logg value:")
    teff4 = input("Set teff value:")
    monh4 = input("Set monh value:")

    colors = ['#377eb8', '#ff7f00', '#4daf4a', '#d62728']

    # loading correct spectrum:
    spectrum = load_file(logg, teff, monh)
    spectrum2 = load_file(logg2, teff2, monh2)
    spectrum3 = load_file(logg3, teff3, monh3)
    spectrum4 = load_file(logg4, teff4, monh4)

    # Add noise to spectrum
    if SNR != 0:
        spectrum[1] = add_noise(SNR, spectrum[1])
        spectrum2[1] = add_noise(SNR, spectrum2[1])
        spectrum3[1] = add_noise(SNR, spectrum3[1])
```

```python
        spectrum4[1] = add_noise(SNR, spectrum4[1])

    # Obtain plotting indices
    index_start = find_nearest_idx(spectrum[0], lambda_start)
    index_end = find_nearest_idx(spectrum[0], lambda_end)

    # Plotting
    fig, axs = plt.subplots(1, 1)

    axs.plot(spectrum[0][index_start:index_end],
             spectrum[1][index_start:index_end],
             label=f'logg={logg}')

    axs.plot(spectrum2[0][index_start:index_end],
             spectrum2[1][index_start:index_end],
             label=f'logg={logg2}')

    axs.plot(spectrum3[0][index_start:index_end],
              spectrum3[1][index_start:index_end],
              label=f'logg={logg3}')

    axs.plot(spectrum4[0][index_start:index_end],
              spectrum4[1][index_start:index_end],
              label=f'logg={logg4}')

    plt.legend()
    plt.xlabel('Wavelength   ')
    plt.ylabel('Relative Flux')
    plt.xticks(fontsize=8)
    plt.suptitle(input("Set graph title: "))

    return print('Graph Done!')

def get_citation(peak_species, peak_wavelength):

    peak_index = find_nearest_idx(vald.wlcent, peak_wavelength)
    vald_species = vald.species[peak_index]
    vald_wlcent = vald.wlcent[peak_index]

    if peak_species == vald_species:
        print(vald.reference[peak_index])
        print(f'Wavelength offset: {peak_wavelength-vald_wlcent}')
        print(f'Peak Index: {peak_index}')
    else:
        print('Species do not match.')
```

B.3 Response Class

```python
import numpy as np
```

```python

class Responses:

    def __init__(self, delta, delta_values, control, control_value, monh):

        self.wavelength = np.array([])
        self.response = np.array([])
        self.delta = delta
        self.delta_values = delta_values
        self.control = control
        self.control_value = control_value
        self.monh = monh
        self.peak_species = []
        self.peak_values = np.array([])
        self.peak_responses = np.array([])
        self.references = []
```