Matching sMILES SSPs to observations of GAMA422436 obtained with SALT

Olliver Hughes
Jeremiah Horrocks Institute
University of Central Lancashire

May 18, 2022

Abstract

sMILES is a library of semi-empirical SSPs (Simple Stellar Populations) based on the MILES library, it has SSPs with $\left[\frac{\alpha}{Fe}\right]$ abundances varying from -0.2, +0.0, +0.2, +0.4 and +0.6. 636 spectra with varying age and metallicities in each of these $\left[\frac{\alpha}{Fe}\right]$ abundances have been provided, giving a total of 3180 spectra. This software aims to find the SSP that best fits provided observational data of the galaxy G422436 from the South African Large Telescope (SALT). This is achieved by blurring, rebinning and continuum normalizing the SSPs and then determining the reduced chi-squared for each SSP and then find the best fitting SSP for each $\left[\frac{\alpha}{Fe}\right]$ abundance. Overall the sMILES SSPs do not match the target galaxy well as all the reduced chi-squared of all SSPs are more than 1. The software has determined that the best fitting SSP is from the solar-like $\left[\frac{\alpha}{Fe}\right]$ which has an age of 5 Gyrs and metallicity of +0.4, the reduced chi-squared for this best fitting SSP was determined to be 7.13. The parameters that best fit the SSPs to the target galaxy are metallicities from -0.4 to +0.4 and at ages of more than 2 Gyrs, SSPs with solar-like $\left[\frac{\alpha}{Fe}\right]$ to $\left[\frac{\alpha}{Fe}\right] = +0.2$ are where the SSPs are best fitted.

Contents

1 Introduction

With improvements made on the MILES library, the sMILES library was created. sMILES improves on the MILES library by using predictions of abundance patterns to differentially correct the MILES library, creating the sMILES library. The galaxy GAMA422436 (G422436) was detected in the Herschel-ATLAS survey (Eales et al., 2010) and is in the GAMA survey (Moffett et al., 2016), it is a dusty early type galaxy positioned at right ascension 08h43m51.082s and declination +02d42m49.87s. Classified as an Sa type galaxy by NED, G422436 has a heliocentric redshift of 0.02586 ± 0.00001 (NASA/IPAC Extragalactic Database). Below Fig.1 can be seen which shows an image of G422436 from the SDSS9 survey.

Figure 1: Image of the target galaxy GAMA422436 from the SDSS9 survey (Aladin Lite)

Eales et al.(2010) describes the Herschel-ATLAS survey as the largest survey done at the Herschel Space Observatory which surveyed 510 square degrees of extragalactic sky which took 600 hours to survey.

The software detailed in this report aims to find the best fitting parameters for the SSPs from the sMILES library to the SALT data of G422436. This is done by blurring and rebinning the sMILES SSPs to the same binning as the SALT data and then continuum normalizing both the models and the data so then a Chi-square test can be performed to determine the best fit. A software called pPXF (Cappellari and Emsellem, 2004) can extract the stellar kinematic or gas kinematics and stellar population from spectra with absorption lines by full spectrum fitting. This software then fits the stellar and gas kinematics however this is not for SSPs and thus the creation of a new software is required.

1.1 Initial Mass Function (IMF)

The IMF is a distribution function describing the distribution of initial masses of stars in a population. Many variations of IMF exist, the first variant being the Salpeter IMF that made the approximation that the IMF has a constant negative gradient. This approximation is good for higher mass stars however does not hold for lower mass stars. The Salpeter IMF is given as:

$$\xi(m) = cM^{-\alpha}$$

Where for the Salpeter IMF,

$$\alpha = 2.35$$

As c is a normalization constant the Salpeter IMF can also be expressed as:

$$\xi(m) \propto M^{-\alpha}$$

Typically the number density is plotted on the y axis and the mass is on the x axis, both of which are given on a log scale. To determine the number of stars at a given mass from the IMF an equation can be derived.

$\xi(m)$ is the number density of the population

The area below the line in these plots that is within Δm contains a number of stars between the upper and lower mass. Hence the number of stars between these limits can be expressed as:

$$\text{Number of stars} = \int_{m_l}^{m_u} \xi(m) dm \tag{1}$$

And hence this can lead to the number fraction of stars by dividing Eqn.1 by the number of all stars in the population:

$$\text{Number fraction} = \frac{\int_{m_l}^{m_u} \xi(m)dm}{\int_{0.08M_\odot}^{100M_\odot} \xi(m)dm}$$

$100M_\odot$ and $0.08M_\odot$ are suitable lower and upper limits for the denominator as very few stars are found at masses higher than $100M_\odot$ and masses lower than $0.08M_\odot$ are where stars do not have enough mass to sustain nuclear fusion of Hydrogen into Helium in their cores which is where brown dwarfs start to appear.

The incremental mass around m_* is defined as:

$$M_* = \text{Number of stars} \times m_* \tag{2}$$

Hence from Eqn.1 and Eqn.2, the total mass can be determined:

$$M_{total} = \int_{m_l}^{m_u} \xi(m)mdm$$

From this the mass fraction can be determined:

$$\text{Mass fraction} = \frac{\int_{m_l}^{m_u} \xi(m)mdm}{\int_{0.08M_\odot}^{100M_\odot} \xi(m)mdm}$$

1.1.1 Isochrones

Isochrones are curves on HR diagrams which describe the mass of stars with the same age. The isochrones determine the relation between effective temperature, surface gravity and mass for a given time and metallicity. The way isochrones are created is through calculations of the stellar evolution for stars that range from the hydrogen burning limit to the maximum stellar mass (Conroy, 2013), the same limits described for the lower and upper mass limits for the IMF.

1.2 Simple Stellar Populations (SSPs)

SSPs are a population of stars that have the same age, same $\left[\frac{Fe}{H}\right]$ metallicity and abundance pattern, $\left[\frac{Fe}{H}\right]$ is the abundance ratio of iron-peak elements to hydrogen in the SSPs. A galaxy can become redder because their age increases, their metallicity increases or because their dust abundance increases or due to multiple of these. Because a galaxy can be red due to any of these indicators, it is difficult to differentiate between old metal poor galaxies and young metal rich galaxies for example. This is known as the age-metallicity degeneracy. An important parameter in the study of SSPs is the α capture elements to iron-peak elements abundance ratio, denoted as $\left[\frac{\alpha}{Fe}\right]$. Conroy (2013) describes how

SSPs are generated, where they give the 3 main ingredients for an SSP to be the IMF, isochrones and stellar spectra. A historical review of SSPs and stellar population studies are presented in Li, de Grijs and Deng (2016), the paper describes how information about the formation of star clusters and their evolution are learned from stellar populations. They explain that star clusters are believed to approximate SSPs because stellar feedback is strong enough to unbind most residual gas from the protocluster and any remaining gas will be cleared due to multiple supernovae from O-type stars. Li, de Grijs and Deng (2016) also describes that no evidence exists that star clusters younger than 100 Myrs have multiple stellar populations or ongoing star formation. The morphologies of the photometric features and chemical abundance variations in old globular clusters show that there are multiple stellar populations present. The study of SSPs is used to understand galaxies with similar abundance patterns by comparing their spectra. Worthey (1994) gives a detailed explanation of how SSPs are made, they explain that a groups of stars of known luminosity, temperature and gravity are represented by each point on the isochrone.

1.3 sMILES library

sMILES is a library of semi-empirical spectra of SSPs with 5 $\left[\frac{\alpha}{Fe}\right]$ abundances ranging from -0.2 to +0.6 dex (Knowles et al., 2021). The $\left[\frac{\alpha}{Fe}\right]$ abundance is a parameter which describes the ratio of alpha elements that come from the feedback from type II supernovae. The library has a wavelength coverage of 3540.5 to 7409.6 Å and has a Full width at half maximum (FWHM) of 2.5 Å. The pixel size of this library is 0.9 Å and the number of pixels is 4300. Because the sMILES SSPs are binned differently to the SALT data the software was made to process the sMILES SSPs to match the SALT data through blurring, rebinning and continuum normalizing the SSPs. The files from the sMILES library follow the MILES naming convention that details the spectral range of the SSP, the IMF type, the value of the IMF slope, the Z metallicity which is the total metallicity $\left[\frac{M}{H}\right]$, the age in Gyrs and the isochrone type (either Padova+00 or BaSTI) (MILES Library).

Knowles et al. (2021) presents the sMILES library as an improvement on the MILES library as the library is created from differential corrections based on the differential abundance predictions of the theoretical MILES library. The predictions of the differential abundance obtained by interpolating to create 2 sets spectra, one set matches the MILES stars in their atmospheric parameters and the other are the MILES stars with different abundance patterns. Then the differential correction is determined by dividing the set matching the MILES stars by the MILES stars with the different abundance patterns, the sMILES library is then made by multiplying the MILES wavelengths by the differential correction (Knowles et al., 2021).

1.4 South African Large Telescope

The observational data used in the development of this software was obtained with the SALT. SALT is equipped with the Robert Stobie Spectrograph (RSS). SALT has

3 HAWAII-2RG detectors (Sheinis et al., 2006) with gaps between their detection range, meaning that there are regions within the spectra obtained with the SALT where no data is present. Table 1 can be seen below which summarizes details of the SALT data and sMILES SSPs. The RSS has an approximate spectral coverage of 3200 Å to 9000 Å which extend up to 1.6 μm when the visible and near infra-red arms are used simultaneously (Sheinis et al., 2006) and it was this spectrograph that was used to obtain the data on G422436.

	Semi-Empirical SSP's	SALT Observations
Pixel Size	0.9 Å/Pixel	1.25 Å/Pixel
Resolution Size	2.5Å (FWHM)	5 Å (FWHM)
Number of Pixels	4300	2440
Wavelength Range	3540-7409.6 Å	3770 – 6818.75 Å

Table 1: FWHM, pixel size, wavelength coverage and number of pixels of the SALT data and sMILES SSPs

2 Method

In the development of this software, Spyder Python 3.9 (Spyder IDE) on a Windows 10 PC and Python 3.7 on the UCLan Starlink network were used to write the code of the software, the Starlink network was accessed from a Linux OS machine in LE204 at UCLan. The UCLan Starlink network uses CentOS, a distribution of the Linux OS. Both of these Python work spaces were used with the modules `specutils`, `SpectRes` (Carnall, 2017), `astropy`, `scipy`, `glob`, `numpy`, `matplotlib` and a custom local module `fit_tools.py`. The module `scipy` was used to blur the SSPs, `SpectRes` was used to rebin the SSPs to the same binning as the SALT data and `specutils` was used to continuum normalize the SSPs and the SALT data. Functions in the local module `fit_tools.py` were used to read in the SSPs and perform the reduced chi-squared test to determine the best fit. During the development of this software the Windows 10 PC with Spyder Python 3.9 was primarily used to write the code of the software due to the slow speed and long processing times on the Starlink network. Before beginning the loop for the $\left[\frac{\alpha}{Fe}\right]$ directories a list of 5 lists is created, one list is for each $\left[\frac{\alpha}{Fe}\right]$ abundance which is where the results of the data processing will be stored. Before beginning the second loop for the SSPs, 5 empty lists are created which will be used to store the reduced chi-squared, age, metallicity, processed flux and filename of the SSPs.

2.1 Reading in the data

Because the SALT data and sMILES SSPs are not in the same format, different methods to read in the respective data are used. The file structure of the sMILES SSP library contains 5 directories, each for a respective $\left[\frac{\alpha}{Fe}\right]$ abundance, the included $\left[\frac{\alpha}{Fe}\right]$ abundances are -0.2, +0.0, +0.2, +0.4, +0.6. Because of this file structure, it was thought best in development of the code to perform the required processes for a single SSP file in an arbitrary abundance directory then to loop these processes over a list of multiple SSP files which can then be looped for each directory. Each abundance directory contains spectral data files and variance files for 636 SSPs.

2.1.1 SALT data of GAMA422436

The SALT data has two FITS files, one is for the observed flux and the other is for the velocity dispersion which is used as an error. To read in data from FITS files the `astropy` module is used to read in the FITS files define the header of the FITS files and an array containing the fluxes. The FITS header is used to build an array of the SALT wavelengths from the starting wavelength, number of pixels and pixel size found in the fits header. The SALT data is then ready to be continuum normalized.

2.1.2 sMILES SSPs

A single SSP in the sMILES library has two file types associated with it, a spectral data file which contains the flux of the SSP across the wavelength range 3540.5 – 7409.6 Å and the var file which contains the variance in the flux of the SSP across the same wavelength range. The fluxes and the wavelengths in both these file types are in columns that are spaced by six whitespaces. The sMILES SSPs are split into 5 directories, one for each $\left[\frac{\alpha}{Fe}\right]$ abundance so the method for reading in the SSP files is to build a list of the found $\left[\frac{\alpha}{Fe}\right]$ directories using pattern matching with the `glob` module. The `glob` module finds all path names based on a given naming pattern. Once the list of found $\left[\frac{\alpha}{Fe}\right]$ abundance directories is defined, a loop is made to find the spectral data files of each SSP in a $\left[\frac{\alpha}{Fe}\right]$ abundance directories by using the pattern matching from `glob` again and build a list of the found spectral data files in the $\left[\frac{\alpha}{Fe}\right]$ abundance directory then another loop is made within the previous loop to read in each the spectral data file and it's respective variance file by adding the suffix that the variance files have in their naming convention.

To read in the files for an SSP a function in the local module `fit_tools.py` checks which lines split and converted into a floating point number. This ensures that the lines which include headers or comments are ignored and only those with numbers that are spaced by six spaces are split into two separate values which in this case is the flux and the wavelength. With the fluxes and wavelengths of the files separated they can be defined as two arrays so that the flux can then be blurred, rebinned and continuum normalized.

2.2 Data Processing

The sMILES SSPs were processed by blurring, rebinning and continuum normalizing the flux and variance in that order. The processing of the SALT data only consisted of continuum normalization.

2.2.1 Blurring

To blur the sMILES SSPs the `gaussian_filter1d` function from the `ndimage` is used. `ndimage` is a submodule from the module `scipy`. The `gaussian_filter1d` function blurs the input flux by an input sigma value, the sigma value used for this was calculated as the sigma difference in pixels prior beginning any of the loops mentioned previously. This method follows a similar method used in the Sauron kinematics example from the software `pPXF` (Cappellari, 2017).

A Diagnostic plot was made to check if the blurring was working correctly, this was tested by plotting the spectra before blurring and the spectra blurred with two different values of sigma to show the effect the value of sigma has on the result of the blurring. The described diagnostic plot is seen below in Fig.2.

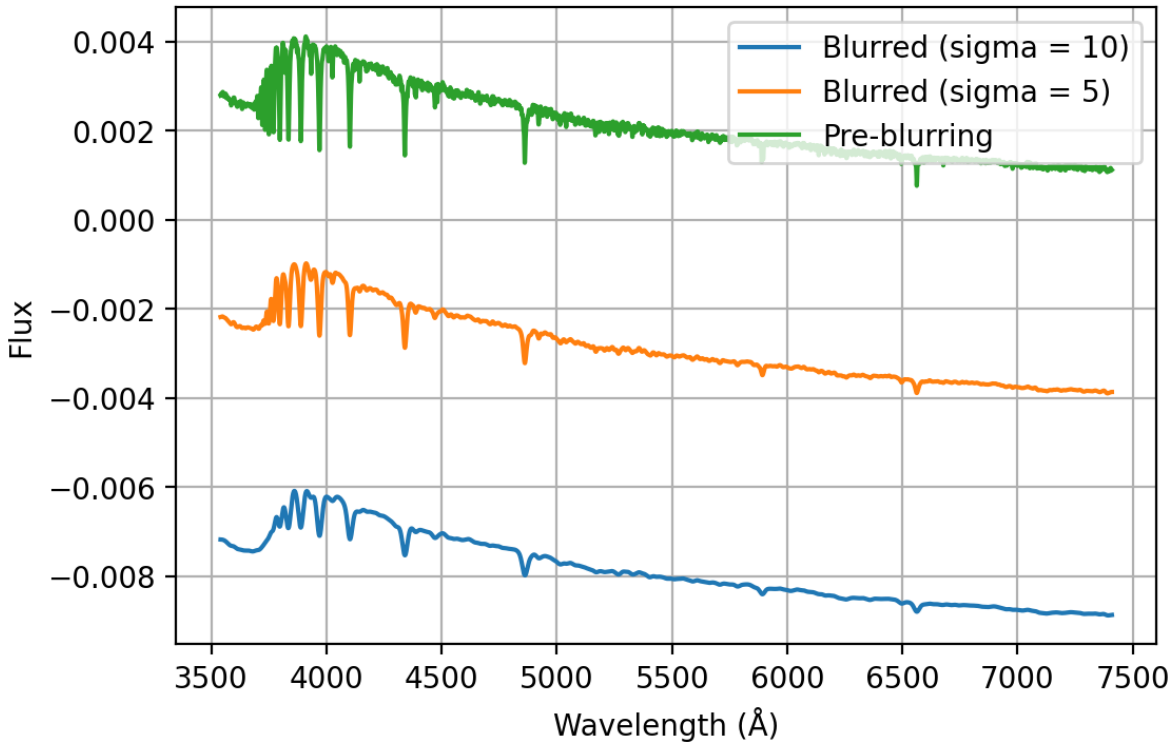

Figure 2: Blurring diagnostic plot. This shows 2 spectra of an SSP blurred by a gaussian with an input sigma value of 5 and the other blurred with a sigma value of 10 and the spectra before any blurring. The blurred spectra have been applied an offset to show them more clearly. The plot shows that the blurring of the models is working correctly.

2.2.2 Rebinning

The rebinning of the sMILES SSPs was done with the module `SpectRes` (Carnall, 2017). The function to rebin the models requires three input parameters, the desired wavelength to rebin the data to, the current binning of the data and the data to be rebinned. The blurred SSP was given as the input data for the function. A method to rebin the SSPs from the Scipy Cookbook (SciPy Cookbook) was attempted however didn't work for unknown reasons so it was decided that using the `SpectRes` module would be the best approach for this. The SciPy Cookbook is a collection of user submitted functions to perform various processes, one of which included rebinning. To check if the blurred SSP was correctly rebinned to the same binning as the SALT data a diagnostic plot was made showing zoomed in data points of the spectra before and after rebinning. Fig.3 shows this diagnostic plot.

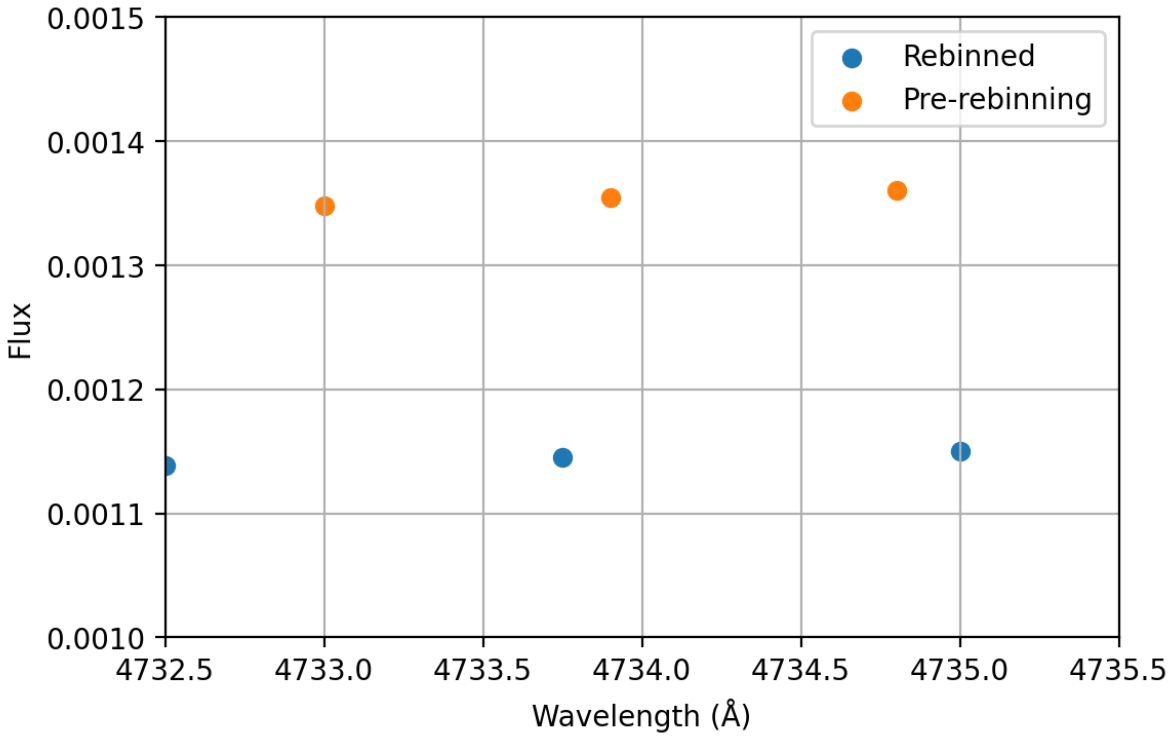

Figure 3: Diagnostic plot. This shows that the SSPs have been rebinned from their original binning (0.9 Å) to the same binning as the SALT data (1.5 Å).

2.2.3 Continuum Normalization

The sMILES SSPs and SALT data were continuum normalized using a Chebyshev polynomial. To continuum normalize the data, an example of continuum normalization written by Adam Knowles was followed, this code was provided by the project supervisor. This method utilizes the module `specutils`. The function from `specutils` for continuum normalizing the data from this module requires the data be in a specific container type also from the `specutils` module. The container type is called `Spectrum1D` and

requires the data be given as an `astropy.units.Quantity` which is done by using the submodule `astropy.units` to assign the flux and wavelength of the data units. The function `fit_continuum` from the `specutils` module obtains the continuum of the input `Spectrum1D`, another parameter for this function is the order of the Chebyshev polynomial which in order to change the order requires that `Chebyshev1D` be imported from `astropy.modeling.polynomial`. The continuum is then fitted by giving the wavelength range of the SALT data with the previously assigned units as an input. Upon plotting the continuum normalized spectra, it was clear that the function is working correctly however the order of Chebyshev polynomial to be used for the fit was yet to be decided. To determine which order of Chebyshev polynomial should be used, different orders of the polynomial were tested and plotted to be compared. Fig.4-6 can be seen below which shows the effect the order of the Chebyshev polynomial has on the fit.

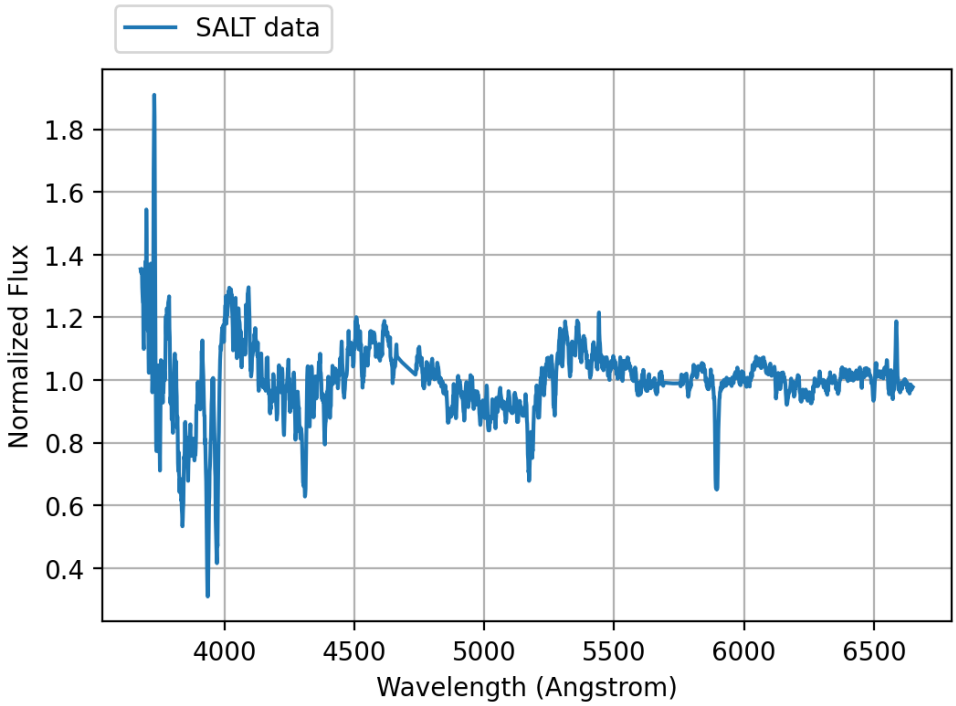

Figure 4: Continuum normalized SALT data using a Chebyshev polynomial order of 5

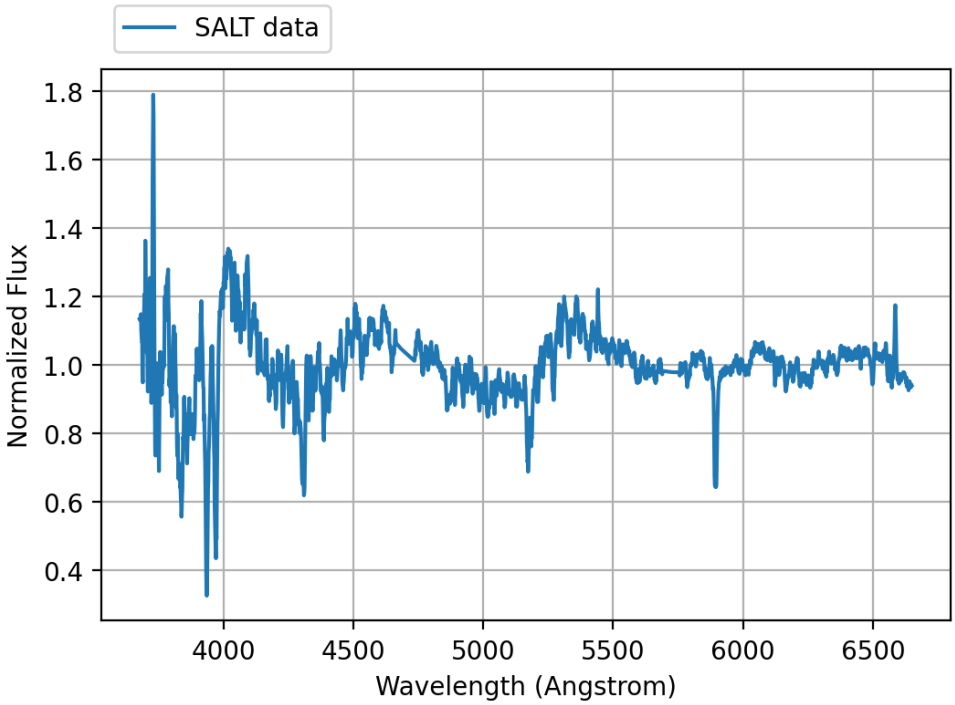

Figure 5: Same as Fig.4, but for order of 6

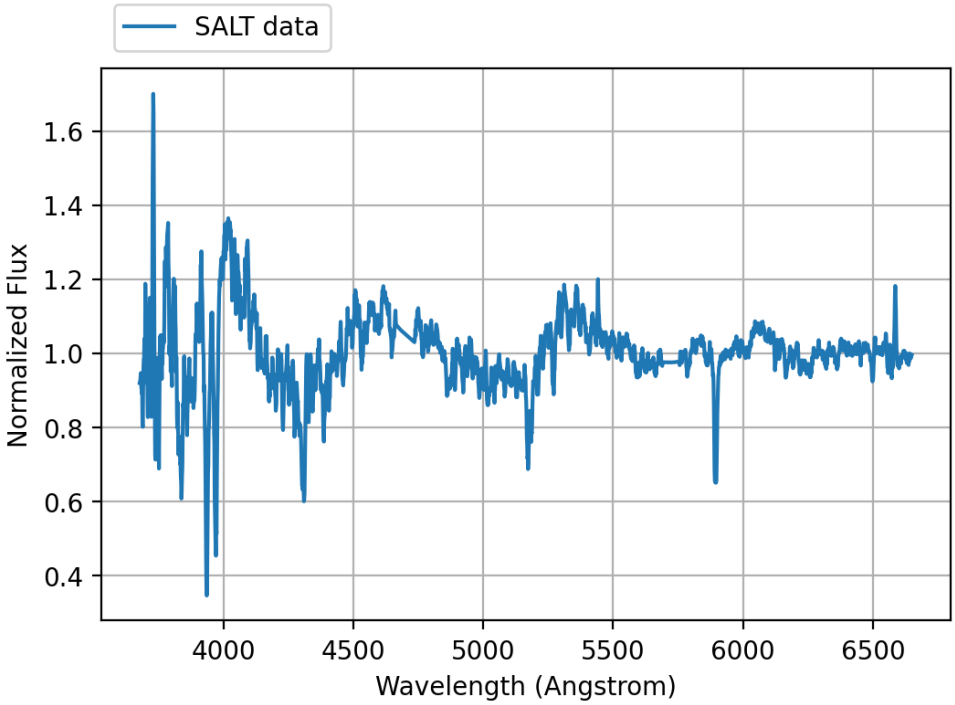

Figure 6: Same as Fig.4, but for order of 7

From Figs.4-6, going beyond a Chebyshev polynomial order of 6 has little effect on the resulting fit and a polynomial order of 5 does not fit well at wavelengths shorter than 4000 Åin particular. From this analysis it was determined that a good order to

use for the polynomial is 6 as going beyond this can also remove some features in the spectra. To reduce the edge effects of the fit when calculating chi-squared a mask was applied to the SSPs and SALT data. The mask excludes data outside the specified wavelength, the wavelength range for the mask was determined by observing where the fit of the continuum is not a good fit around the edges. Fig.7 below shows the continuum normalized SALT data before and after the mask was applied.

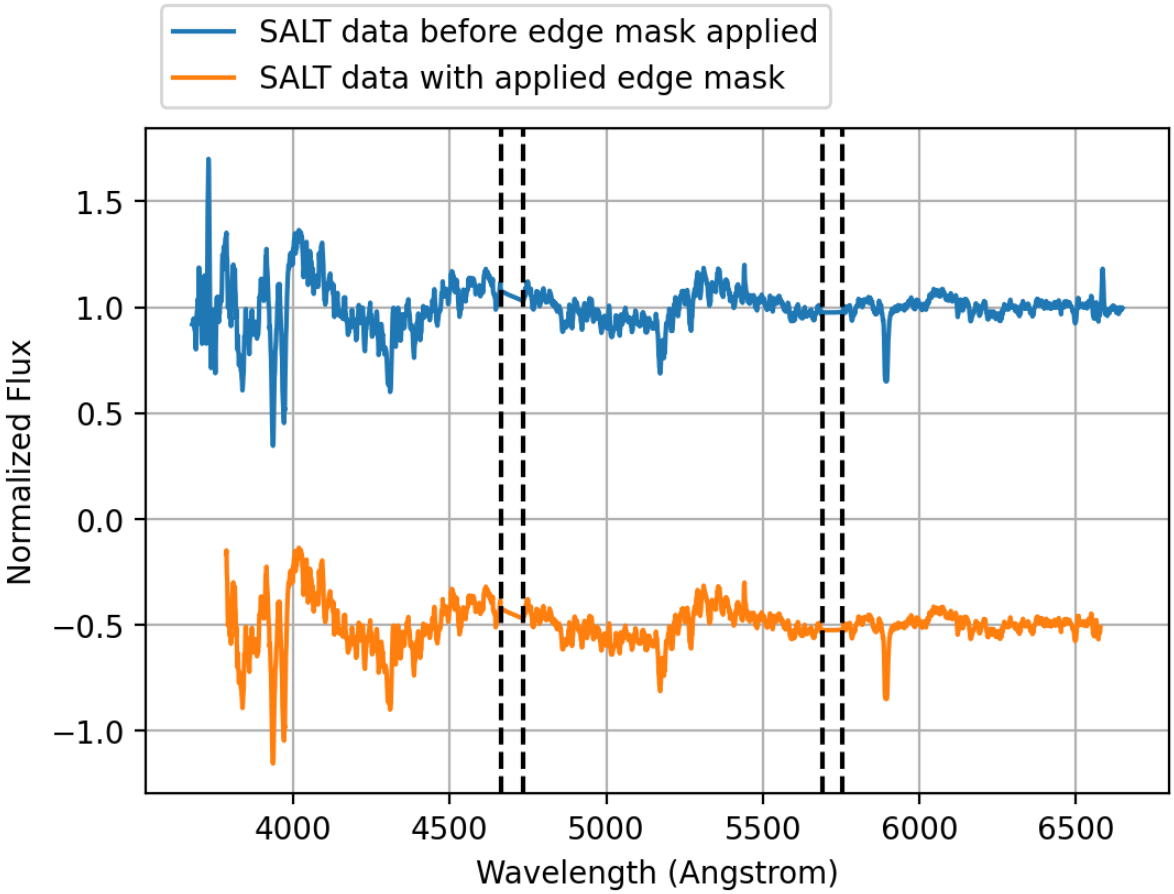

Figure 7: Continuum normalized SALT data before and after masking of the edges. The dashed black vertical lines indicate where there is not data due to the gaps in the detectors. These lines appear at 4663.9 Å, 4734.7 Å, 5690.3 Å and 5753.8 Å, the location where these lines should be placed were determined by zooming into the plot to see where there is no data and estimating where the data ends.

2.2.4 Extracting the age and metallicity from filenames

In the custom local module `fit_tools.py` a function was created to read the file name of an SSP that follows the MILES library naming convention (MILES Library) and extract the age and metallicity from the file name then convert it into a floating point number which is then used in plots. The function detects if the metallicity from the file name has an 'm' or 'p' before the value of the metallicity to determine if the number is positive or negative.

2.3 Calculating reduced Chi-squared

Calculating reduced chi-squared will be important as this will be how the best fitting SSPs are determined. To calculate reduced chi-squared the chi-squared is first calculated using a function in the local module `fit_tools.py`, this function follows Eqn.3 shown below.

$$\chi^2 = \sum \frac{(Flux_{SALT} - Flux_{SSP})^2}{(Error_{SALT})^2} \tag{3}$$

The fluxes in Eqn.3 are the respective fluxes after being processed. Then reduced chi-squared is determined from the chi-squared by dividing the sum of the chi-squared by degrees of freedom which is determined by subtracting the degrees of freedom from the number of data points in the FITS header of the SALT data. The degrees of freedom is 3 as the parameters being varied are the age, metallicity and $\left[\frac{\alpha}{Fe}\right]$. Eqn.4 shows the calculation for reduced chi-squared where DOF is the degrees of freedom.

$$\chi^2_{red} = \frac{\chi^2}{DOF} \tag{4}$$

2.4 Storage of the processed data

After the flux has been processed, reduced chi-squared determined and age and metallicity have been extracted for a single SSP, this data is then appended into their respective lists made prior to the data processing. Once this process has been repeated for all SSPs in a single $\left[\frac{\alpha}{Fe}\right]$ abundance directory, the 5 lists containing the processed data are then appended into the one of the 5 empty lists made before the first loop for the $\left[\frac{\alpha}{Fe}\right]$ abundances. This leaves the final list as a list of 5 lists, one for each of the $\left[\frac{\alpha}{Fe}\right]$ abundances and within each of these 5 lists there are another 5 lists which store their respective reduced chi-squared, ages, metallicities, processed flux and filenames of the SSPs. A block diagram of this loop can be seen in Fig.8 below.

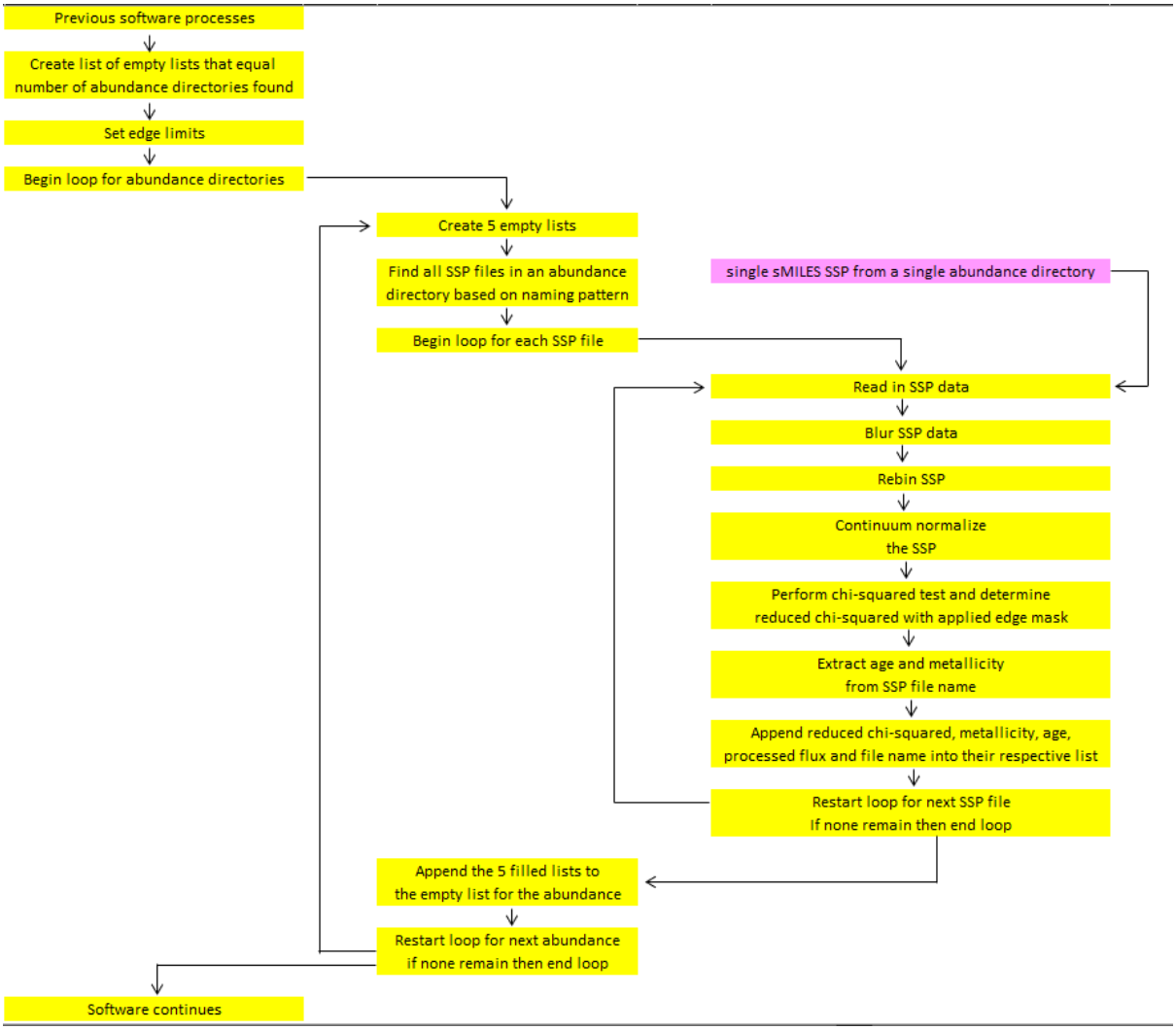

Figure 8: Block diagram of the loops and processes described throughout section 2.2

From Fig.8, where the software continues is where the plots are made and the best fitting SSP are determined simply by finding the index in the list at which the reduced chi-squared is lowest. Once the best fitting SSPs from each $\left[\frac{\alpha}{Fe}\right]$ abundance has been found, their $\left[\frac{\alpha}{Fe}\right]$ abundance, file name, age, metallicity and reduced chi-squared are then written to an output `.txt` file called `sMILES_fit_output.txt`.

3 Results

3.1 Reduced Chi-squared trends in the different $\left[\frac{\alpha}{Fe}\right]$ abundances

By plotting the age, metallicity and reduced chi-squared on a 3d plot the trends of how chi-squared changes as the age and metallicity are varied. The described 3d plots for the different $\left[\frac{\alpha}{Fe}\right]$ abundances can be seen below in Fig.9-13.

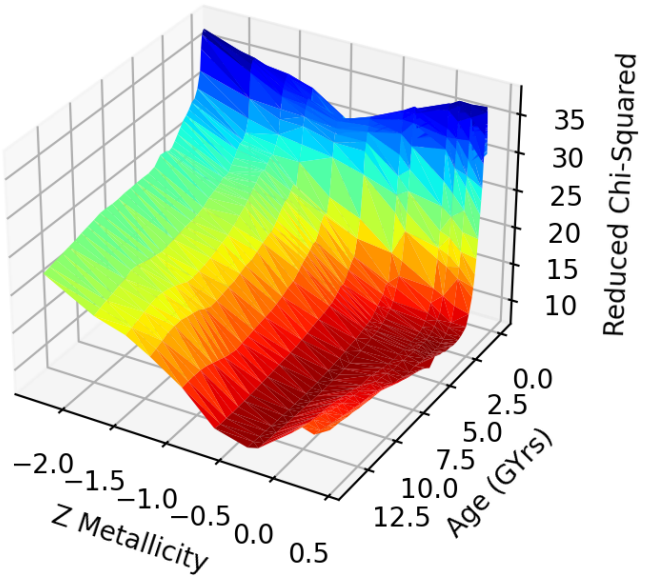

Figure 9: 3d plot showing the trends in reduced chi-squared as age and metallicity are varied for $\left[\frac{\alpha}{Fe}\right] = -0.2$, the red areas show the lowest reduced chi-squareds. The best fitting SSP from this $\left[\frac{\alpha}{Fe}\right]$ has an age and metallicity of 3.25 Gyrs and +0.26 respectively and it's reduced chi-squared is 8.01.

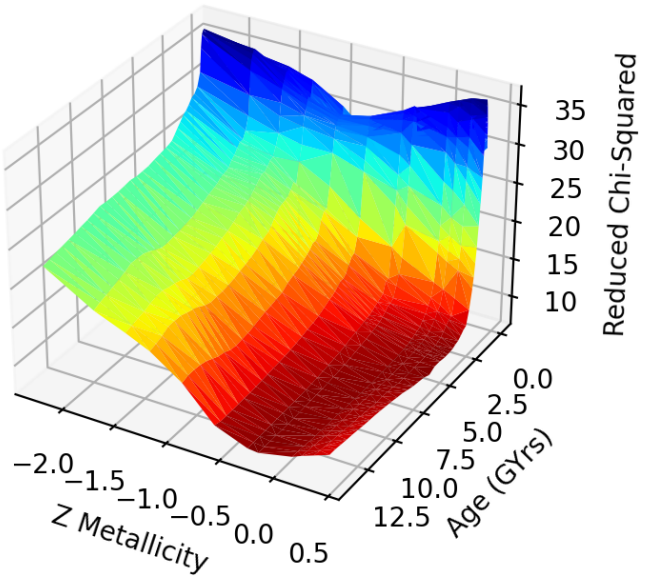

Figure 10: Same as Fig.9, but for $\left[\frac{\alpha}{Fe}\right] = +0.0$. The best fitting SSP from this $\left[\frac{\alpha}{Fe}\right]$ has an age and metallicity of 5.0 Gyrs and +0.4 respectively and it's reduced chi-squared is 7.13.

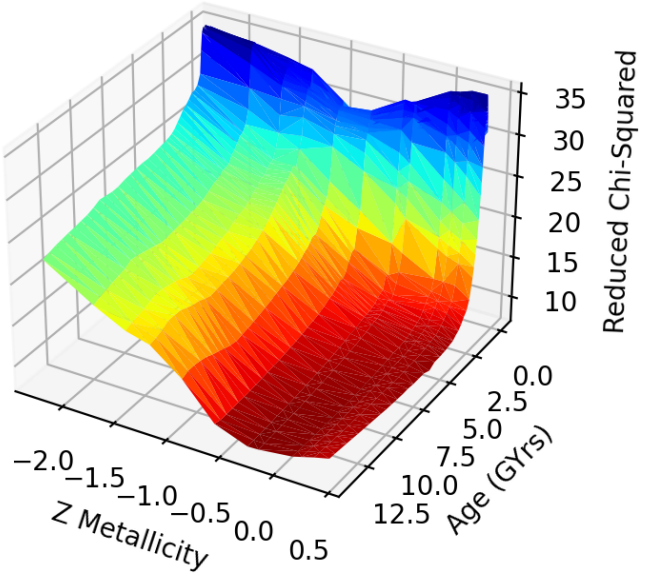

Figure 11: Same as Fig.9, but for $\left[\frac{\alpha}{Fe}\right] = +0.2$. The best fitting SSP from this $\left[\frac{\alpha}{Fe}\right]$ has an age and metallicity of 14.0 Gyrs and -0.25 respectively and it's reduced chi-squared is 7.69.

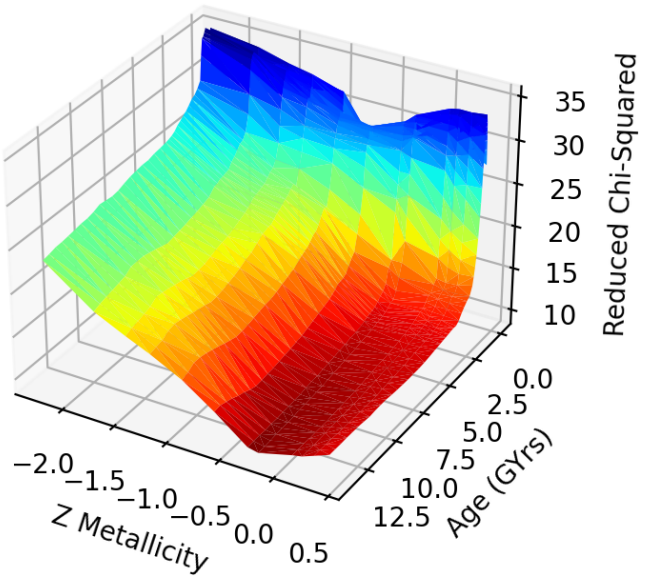

Figure 12: Same as Fig.9, but for $\left[\frac{\alpha}{Fe}\right] = +0.4$. The best fitting SSP from this $\left[\frac{\alpha}{Fe}\right]$ has an age and metallicity of 14.0 Gyrs and -0.25 respectively and it's reduced chi-squared is 8.98.

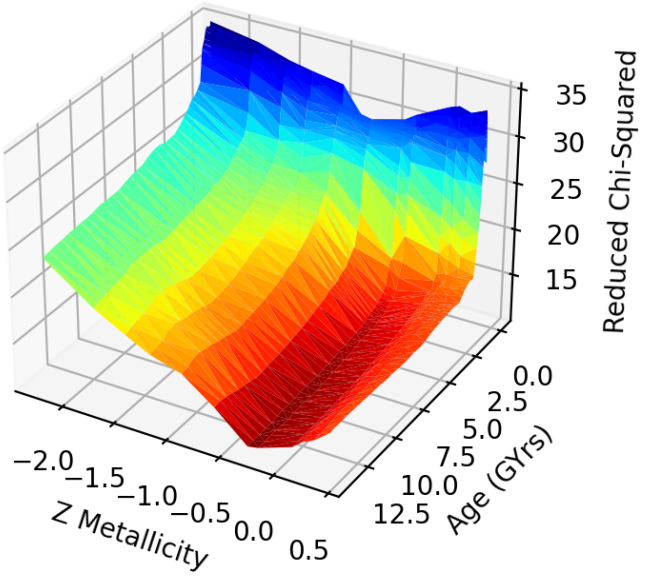

Figure 13: Same as Fig.9, but for $\left[\frac{\alpha}{Fe}\right] = +0.6$. The best fitting SSP from this $\left[\frac{\alpha}{Fe}\right]$ has an age and metallicity of 14.0 Gyrs and -0.35 respectively and it's reduced chi-squared is 10.66.

From Figs.x-y, the SSPs are generally best fitted at solar-like $\left[\frac{\alpha}{Fe}\right]$ and $\left[\frac{\alpha}{Fe}\right] = +0.2$. The lowest chi-squareds are found at the red regions in the plots and Fig.10 and Fig.11 are not so different in terms how wide their red regions are compared to Fig.12-13 which

show the red regions becoming more narrow as $\left[\frac{\alpha}{Fe}\right]$ increases. The best fitting SSP is found at solar-like with an age of 5 Gyrs and metallicity of +0.4. Fig.14 below shows what is happening at the individual data points $\left[\frac{\alpha}{Fe}\right] = +0.0$ more clearly.

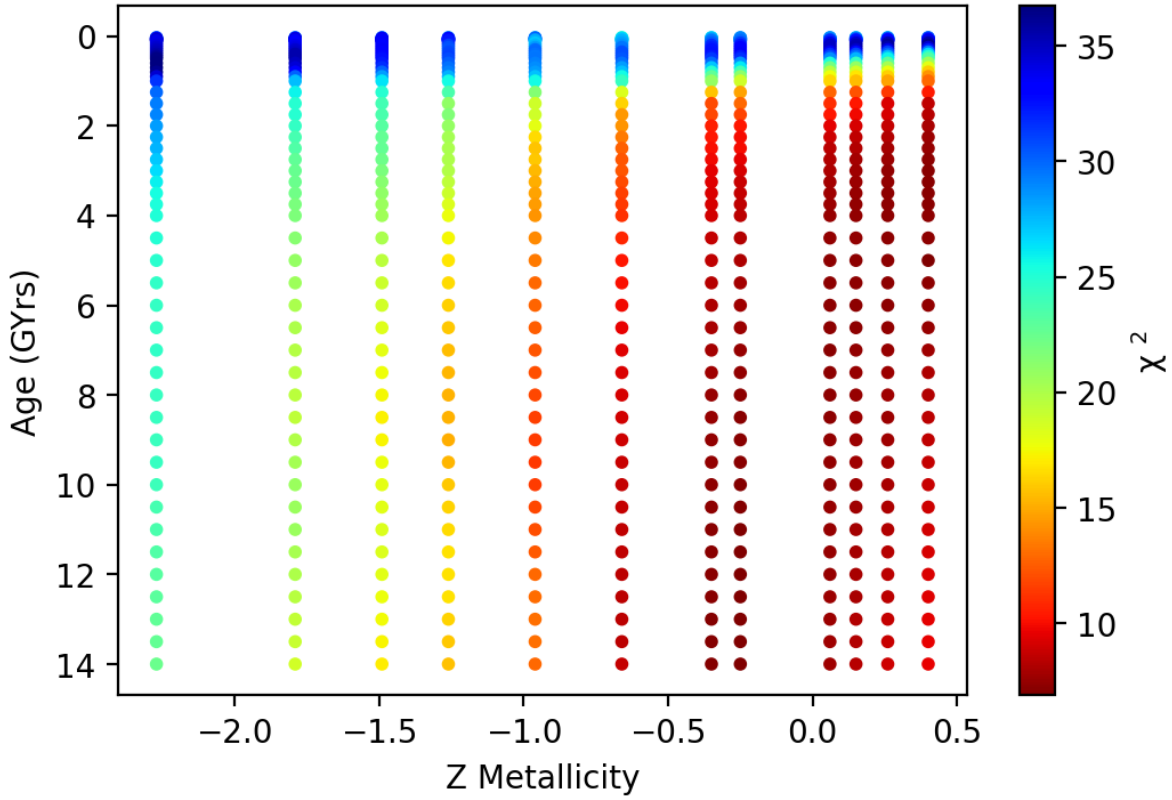

Figure 14: Scatter plot of age (Gyrs) vs metallicity with a reduced chi-squared as a colour map. This plot shows that the SSPs with the lowest reduced chi-squared are generally found at a metallicity range of -0.4 to +0.4.

Fig.14 shows that the best fits can be found in the metallicity range of -0.4 to +0.4 and can have any age more than 2 Gyrs. This demonstrates the age-metallicity degeneracy as the young metal rich SSPs cannot be differentiated by the old metal poor SSPs.

3.2 Best fitting SSP from $\left[\frac{\alpha}{Fe}\right]$ abundance

Fig.15-19 can be seen below which show the spectra of the best fitting SSP from each $\left[\frac{\alpha}{Fe}\right]$ abundance along with the with the spectra from the SALT data and the residuals from these two spectra. The residuals have been calculated as the difference between the spectra.

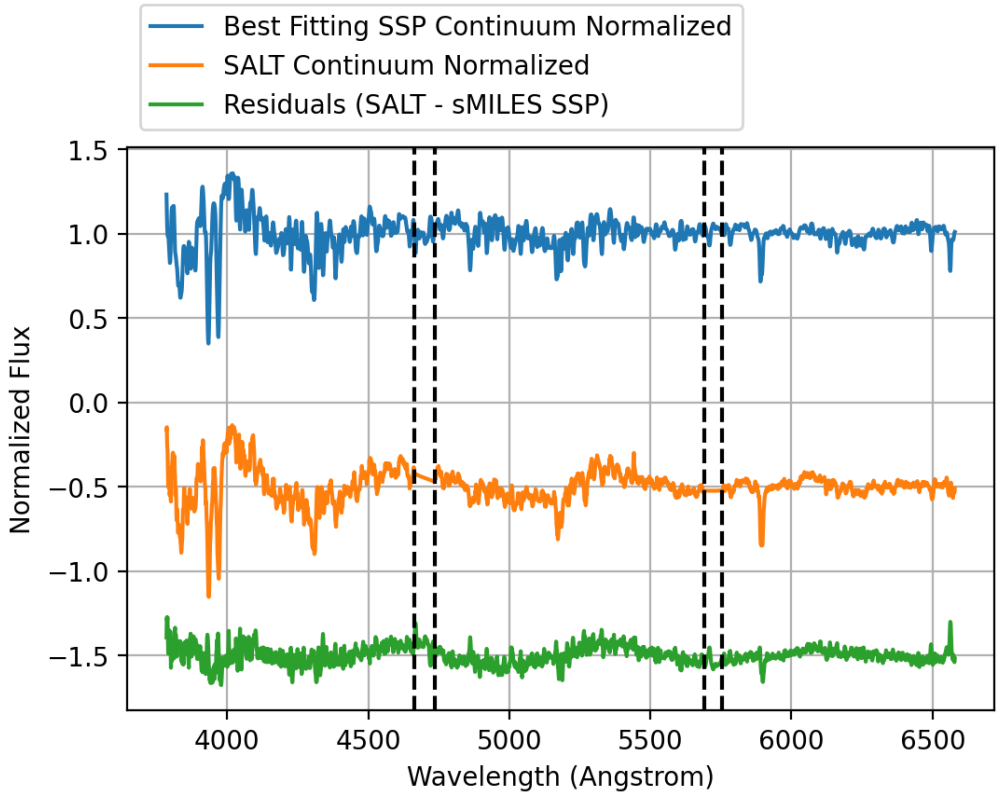

Figure 15: This plot shows the continuum normalized SALT data, continuum normalized best fitting SSP $\left[\frac{\alpha}{Fe}\right] = -0.2$ from and it's residuals. The flux of the SSP and residuals have been applied an offset by subtracting 1.5 from the normalized flux. The dashed black vertical lines at 4663.9 Å, 4734.7 Å, 5690.3 Å and 5753.8 Å are the same as in Fig.7 which show the gaps in the detectors hence there is no data. Prominent spectral features in the sMILES SSP spectra include the CaH, CaK, Na, Mg, Hδ, Hβ and Hα features. The SALT data has the same features except for the Hδ feature.

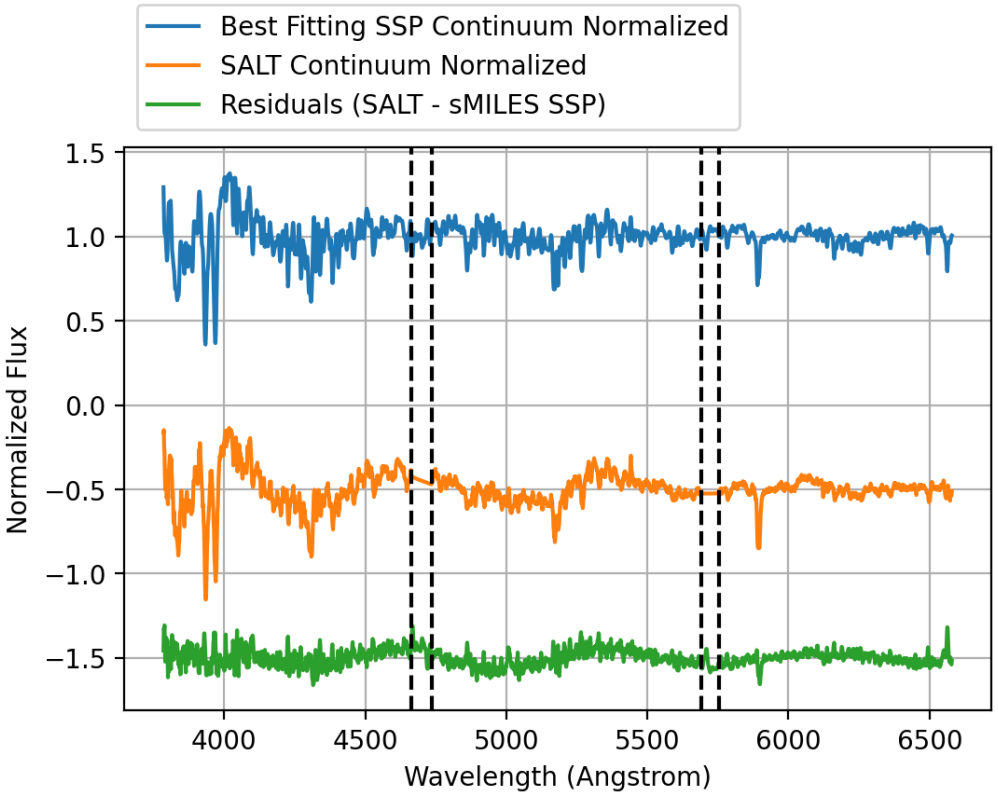

Figure 16: Same as Fig.15, but for $\left[\frac{\alpha}{Fe}\right] = +0.0$.

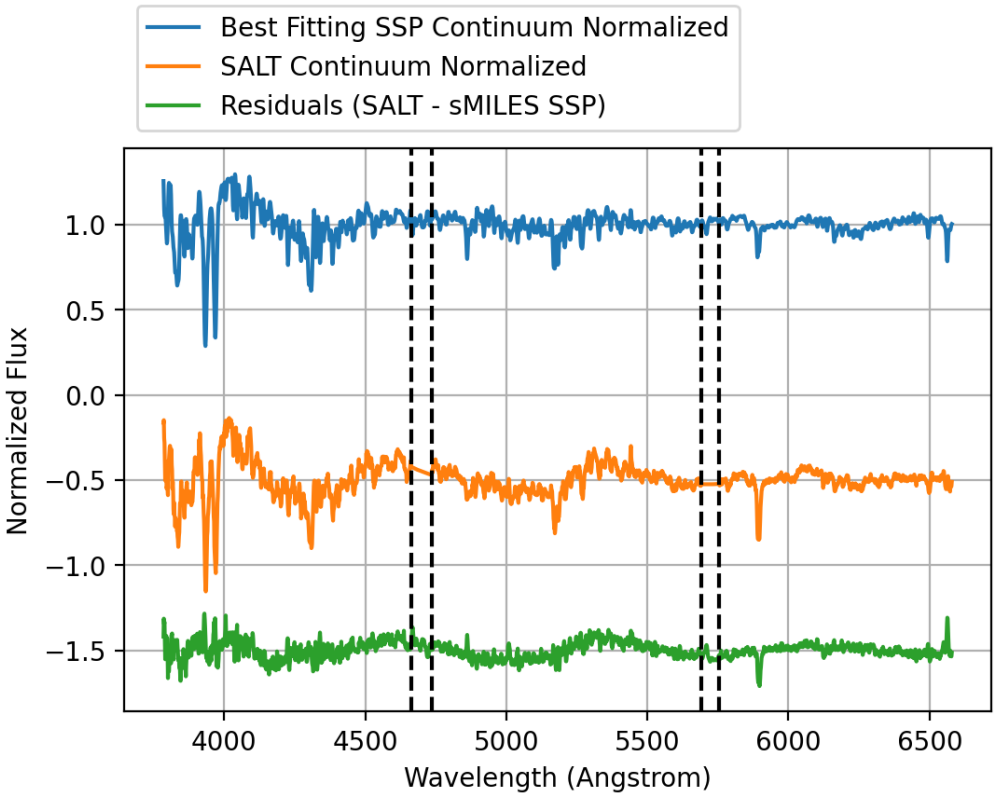

Figure 17: Same as Fig.15, but for $\left[\frac{\alpha}{Fe}\right] = +0.2$.

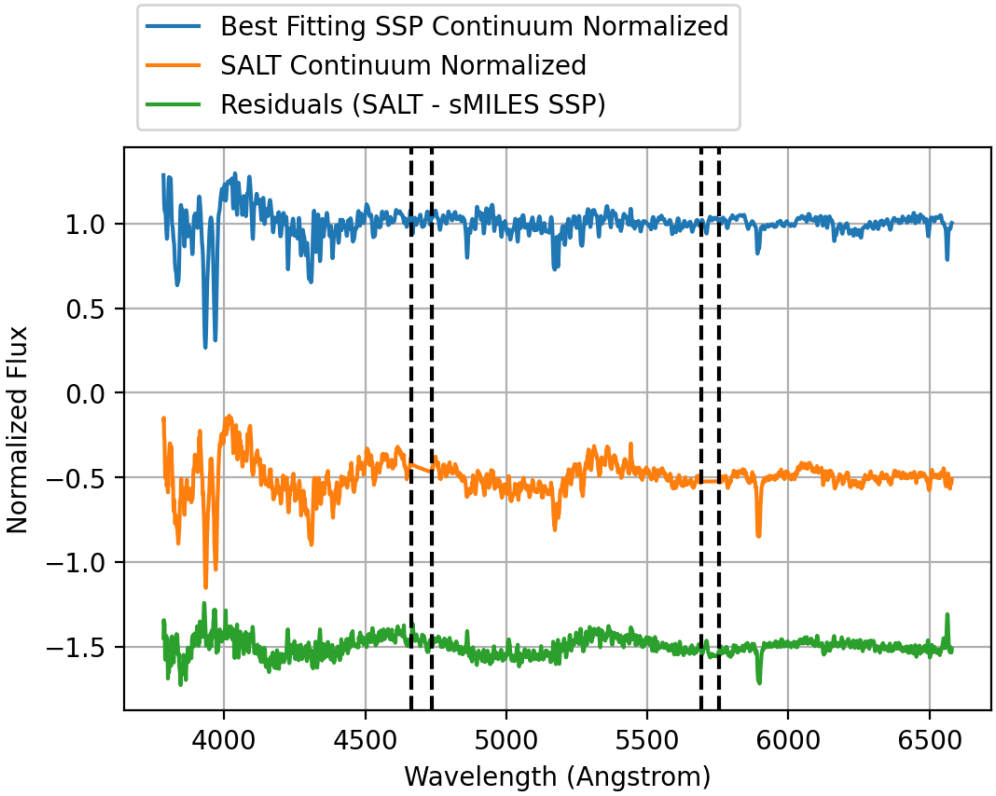

Figure 18: Same as Fig.15, but for $\left[\frac{\alpha}{Fe}\right] = +0.4$.

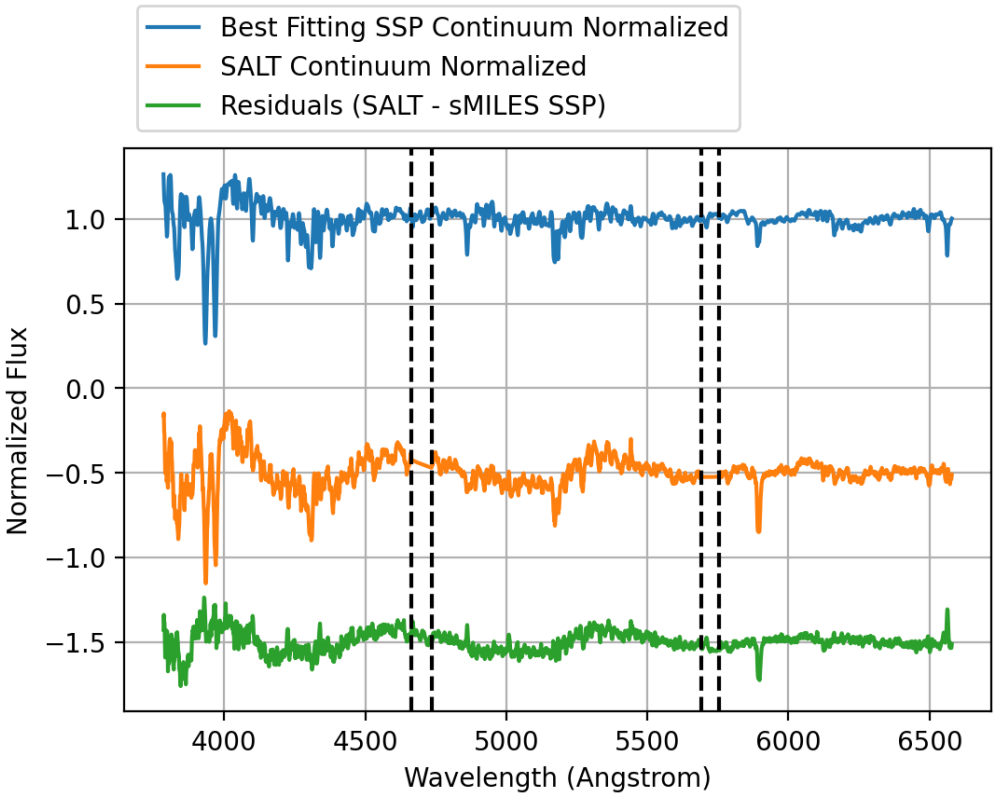

Figure 19: Same as Fig.15, but for $\left[\frac{\alpha}{Fe}\right] = +0.6$.

In Figs.x-y, spectral features can be identified with the CaK and CaH lines seen at the wavelengths 3934.8 Å and 3969.6 Å respectively. The Na, Hα, Mg, Hβ, Hδ absorption lines have been identified in the plots at 5895.6 Å, 6562.8 Å, Mg 5176.7 Å, 4861.4 Å and 4101.7 Å respectively (SDSS). The absorption line at 4305.6 Å is suspected to be the CaG absorption line.

4 Discussion

The plots shown in Fig.9-14 show that the reduced chi-squared has a stronger dependency on the metallicity as the once the age reaches 2 Gyrs going beyond this does not have a noticeable effect on the reduced chi-squared. These plots also show that the best fitting models can be found anywhere from -0.4 to +0.4 metallicity as this is where the chi-squareds are found to be lowest, this is also shown in Fig.9-14 where the redder regions are in this metallicity range.

The best fitting SSP has a reduced chi-squared of 7.13 with an age and metallicity of 5 Gyrs and +0.4 respectively and has $\left[\frac{\alpha}{Fe}\right] = +0.0$ abundance indicating that the models with this abundance fit the SALT data best. The abundance that best fits the SALT data is not clear as the $\left[\frac{\alpha}{Fe}\right] = +0.2$ abundance also shows a wide red region similar to $\left[\frac{\alpha}{Fe}\right] = +0.0$ that can indicate that generally the models with $\left[\frac{\alpha}{Fe}\right] = +0.2$ can also match the SALT data just as well. The best fitting SSP with $\left[\frac{\alpha}{Fe}\right] = +0.2$ has a reduced chi-squared of 7.69 and an age and metallicity of 14 Gyrs and -0.25 respectively.

The best fitting SSP from $\left[\frac{\alpha}{Fe}\right] = +0.2$ is not much of a worse fit compared to the $\left[\frac{\alpha}{Fe}\right] = +0.0$ best fitting SSP. These two best fitting SSPs demonstrate the age-metallicity degeneracy as the $\left[\frac{\alpha}{Fe}\right] = +0.0$ best fitting SSP is a young metal rich SSP while the $\left[\frac{\alpha}{Fe}\right] = +0.2$ best fitting SSP is an old metal poor SSP, showing that the SSP that can best match the target galaxy can be a young metal rich SSP or an old metal poor SSP.

While none of the fits are particularly well fitted, it is clear that $\left[\frac{\alpha}{Fe}\right] = +0.0$ and $\left[\frac{\alpha}{Fe}\right] = +0.2$ SSPs are where the best fits are found. Line broadening is present in the SALT data characterized by the velocity dispersion, This may have had some effect on the fits however the results clearly show that these SSPs do not fit particularly well. The median of the signal to noise ratio of the SALT data is 53.9. While a finer grid could be used, a finer grid will not drastically change the outcome of this project.

4.1 Edge effects from continuum normalization

The continuum fits of the SSPs and SALT data are not well fitted around the edges. To reduce the effects of the poor fitting during the calculation of reduced chi-squared a mask was applied to exclude these poorly fitted edges. The limits for this mask were decided by estimating where the edges are reasonably fitted, for the limits at the longer wavelengths, the Hα absorption line in the SSPs was attempted to be included while removing the Hα emission line in the SALT data.

4.2 Effect of gaps in SALT observations on reduced chi-squared

The gaps in the detectors in the SALT observations will have a small effect on the reduced chi-squared. The data between the gaps is interpolated and thus not real data. To reduce the effect this has on the reduced chi-squared another mask could be applied similarly for the edge effects to remove the data in the SSPs and SALT data where there is no real data to be analysed. Even the data between the gaps were masked to minimize the effects on the reduced chi-squared, the difference this would make on the reduced chi-squared would be minimal as even the best fitting SSPs are not well fitted.

5 Conclusion

In this report, the parameters to achieve a best fit has been investigated and the results show that the best fitting SSP to the SALT data is at solar-like $\left[\frac{\alpha}{Fe}\right]$ abundance with an age and metallicity of 5 Gyrs and +0.4 respectively with a reduced chi-squared of 7.13 which means that the best fitting SSP is not well fitted. From the plot in Fig.14 that the parameters for the best fit can have a large metallicity range of -0.4 to +0.4 and that the age can be upwards of 2 Gyrs and have solar-like to +0.2 $\left[\frac{\alpha}{Fe}\right]$ abundances. These are large ranges for the age and metallicity thus making it difficult to accurately determine the best fit for the age and metallicity, this is expected from the age-metallicity degeneracy.

5.1 Future work

Improvements could be made to the code of the software such as removing the data in the gaps of the detectors, this wouldn't drastically change the result of the fits however it would be more accurate. User quality of life changes could be made to the software such as user input prompts for which SSPs to fit, the order of the Chebyshev polynomial to use for the continuum normalization and which plots should be made.

6 References

Aladin Lite. Available at: `https://aladin.u-strasbg.fr/AladinLite/`
(Accessed: 14/05/2022).

NASA/IPAC Extragalactic Database. Available at: `https://ned.ipac.caltech.edu/`
(Accessed: 14/05/2022)

MILES Library. Available at: `http://research.iac.es/proyecto/miles/pages/ssp-models/name-convention.php`
(Accessed: 14/05/2022).

Spyder IDE. Available at https://www.spyder-ide.org/
(Accessed: 16/05/2022)

SciPY Cookbook. Available at: https://scipy-cookbook.readthedocs.io/items/Rebinning.html
(Accessed: 17/05/2022)

SDSS. Available at: http://classic.sdss.org/dr6/algorithms/linestable.html\(Accessed:14/05/2022) Knowles, A.T., Sansom, A.E., Allende Prieto, C. and Vazdekis, A. (2021) 'sMILES: a library of semi-empirical MILES stellar spectra with variable [α/Fe] abundances', Monthly Notices of the Royal Astronomical Society, 504, pp. 2286-2311. doi: 10.1093/mnras/stab1001; eprintid: arXiv:2104.04822.

Cappellari, M. (2017) 'Improving the full spectrum fitting method: accurate convolution with Gauss-Hermite functions', Monthly Notices of the Royal Astronomical Society, 466, pp. 798-811. doi: 10.1093/mnras/stw3020; eprintid: arXiv:1607.08538.

Cappellari, M. and Emsellem, E. (2004) 'Parametric Recovery of Line-of-Sight Velocity Distributions from Absorption-Line Spectra of Galaxies via Penalized Likelihood', Publications of the Astronomical Society of the Pacific, 116, pp. 138-147. doi: 10.1086/381875; eprintid: arXiv:astro-ph/0312201.

Carnall, A.C. (2017) SpectRes: A Fast Spectral Resampling Tool in Python.

Conroy, C. (2013) 'Modeling the Panchromatic Spectral Energy Distributions of Galaxies', Annual Review of Astronomy and Astrophysics, 51, pp. 393-455. doi: 10.1146/annurev-astro-082812-141017; eprintid: arXiv:1301.7095.

Eales, S., Dunne, L., Clements, D., Cooray, A., De Zotti, G., Dye, S., Ivison, R., Jarvis, M., Lagache, G., Maddox, S., Negrello, M., Serjeant, S., Thompson, M.A., Van Kampen, E., Amblard, A., Andreani, P., Baes, M., Beelen, A., Bendo, G.J., Benford, D., Bertoldi, F., Bock, J., Bonfield, D., Boselli, A., Bridge, C., Buat, V., Burgarella, D., Carlberg, R., Cava, A., Chanial, P., Charlot, S., Christopher, N., Coles, P., Cortese, L., Dariush, A., da Cunha, E., Dalton, G., Danese, L., Dannerbauer, H., Driver, S., Dunlop, J., Fan, L., Farrah, D., Frayer, D., Frenk, C., Geach, J., Gardner, J., Gomez, H., González-Nuevo, J., González-Solares, E., Griffin, M., Hardcastle, M., Hatziminaoglou, E., Herranz, D., Hughes, D., Ibar, E., Jeong, W., Lacey, C., Lapi, A., Lawrence, A., Lee, M., Leeuw, L., Liske, J., López-Caniego, M., Müller, T., Nandra, K., Panuzzo, P., Papageorgiou, A., Patanchon, G., Peacock, J., Pearson, C., Phillipps, S., Pohlen, M., Popescu, C., Rawlings, S., Rigby, E., Rigopoulou, M., Robotham, A., Rodighiero, G., Sansom, A., Schulz, B., Scott, D., Smith, D.J.B., Sibthorpe, B., Smail, I., Stevens, J., Sutherland, W., Takeuchi, T., Tedds, J., Temi, P., Tuffs, R., Trichas, M., Vaccari, M., Valtchanov, I., van der Werf, P., Verma, A., Vieria, J., Vlahakis, C. and White, G.J. (2010) 'The Herschel ATLAS', Publications of the Astronomical Society of the Pacific, 122, pp. 499. doi: 10.1086/653086; eprintid: arXiv:0910.4279.

Knowles, A.T., Sansom, A.E., Allende Prieto, C. and Vazdekis, A. (2021) 'sMILES: a library of semi-empirical MILES stellar spectra with variable [α/Fe] abundances', Monthly

Notices of the Royal Astronomical Society, 504, pp. 2286-2311. doi: 10.1093/mnras/stab1001; eprintid: arXiv:2104.04822.

Li, C., de Grijs, R. and Deng, L. (2016) 'Stellar populations in star clusters', Research in Astronomy and Astrophysics, 16, pp. 179. doi: 10.1088/1674-4527/16/12/179; eprintid: arXiv:1609.00894.

Moffett, A.J., Ingarfield, S.A., Driver, S.P., Robotham, A.S.G., Kelvin, L.S., Lange, R., Meštrić, U., Alpaslan, M., Baldry, I.K., Bland-Hawthorn, J., Brough, S., Cluver, M.E., Davies, L.J.M., Holwerda, B.W., Hopkins, A.M., Kafle, P.R., Kennedy, R., Norberg, P. and Taylor, E.N. (2016) 'Galaxy And Mass Assembly (GAMA): the stellar mass budget by galaxy type', Monthly Notices of the Royal Astronomical Society, 457, pp. 1308-1319. doi: 10.1093/mnras/stv2883; eprintid: arXiv:1512.02342.

Sheinis, A.I., Wolf, M.J., Bershady, M.A., Buckley, D.A.H., Nordsieck, K.H. and Williams, T.B. (2006) The NIR upgrade to the SALT Robert Stobie Spectrograph. . 2006/06/1.

Worthey, G. (1994) 'Comprehensive Stellar Population Models and the Disentanglement of Age and Metallicity Effects', The Astrophysical Journal Supplement Series, 95, pp. 107. doi: 10.1086/192096.

7 Acknowledgements

This project was supervised by Anne Sansom. The SALT observations for which this data was used from were also taken by Anne Sansom.

8 Appendices

8.1 Software code

```
# -*- coding: utf-8 -*-
"""
@author: Olliver Hughes


sMILES SSP Fitting software performs blurring, rebinning and continuum
↪  normalization of the sMILES SSPs and input observed data.
The best fitting SSPs are determined via a Chi-squared test.
This software is intended to be used with the module fit_tools.py,
↪  developed specifically for this software.
In development of this software SALT data of the galaxy G422436 was
↪  used for the input data.
```

```
Olliver Hughes 26 Jan 2022
Olliver Hughes 1 Feb 2022 - Blurring for a single SSP has been
↪  implemented
Olliver Hughes 14 Feb 2022 - Blurring for multiple SSPs implemented
Olliver Hughes 7 Mar 2022 - Rebinning of sMILES SSPs to SALT data
↪  binning using spectres implemented
Olliver Hughes 7 Mar 2022 - Reads files names based on naming pattern
Olliver Hughes 10 Mar 2022 - Continuum fitting of SSPs and SALT data
↪  using specutils implemented
Olliver Hughes 25 Mar 2022 - Chi-Squared test functionality added to
↪  fit_tools.py module
"""

import numpy as np
from astropy.io import fits
import matplotlib.pyplot as plt
from specutils.fitting import fit_continuum
from specutils.spectra import Spectrum1D, SpectralRegion
from astropy import units as u
import glob
import spectres as sp
from scipy import ndimage
from astropy.modeling.polynomial import Chebyshev1D
from matplotlib import cm
from mpl_toolkits import mplot3d
import fit_tools as ft #Local module



def sMILES_SSP_fitting():


    file = 'G422436_sci_apr_central_rest.fits' #SALT observational data
    ↪  to be read in
    err_file = 'G422436_sci_sig_central_rest.fits' #Error of read in
    ↪  SALT observational data

    hdu = fits.open(file) #Read in fits file
    err_hdu = fits.open(err_file) #Read in error fits file

    h1=hdu[0].header #Define fits header
    err_h = err_hdu[0].header #Define error fits header


    SALT_flux = hdu[0].data #Array containing the SALT fluxes from the
    ↪  fits file
    SALT_err = err_hdu[0].data #Array containing the errors of the SALT
    ↪  fluxes
```

```
SALT_median_norm = SALT_flux/np.median(SALT_flux) #SALT flux
↪  normalized by the median of the SALT flux

SALT_wvl = np.arange(h1['CRVAL1'], h1['CRVAL1'] + (h1['CDELT1'] *
↪  (h1['NAXIS1'])), h1['CDELT1']) #Create an array containing the
↪  wavelengths from the data in the fits file

Chebyshev_NCONT=6 #Chebyshev polynomial order


SALT_spectrum=Spectrum1D(flux=SALT_flux*(u.erg / u.angstrom / u.cm **
↪  2 / u.s),spectral_axis=SALT_wvl*u.angstrom) #Container for SALT
↪  spectral data

SALT_cont_model = fit_continuum(SALT_spectrum,
↪  model=Chebyshev1D(Chebyshev_NCONT)) #Shape of the continuum of
↪  the specified spectrum as an astropy model

SALT_cont_flux = SALT_cont_model(SALT_wvl*u.angstrom) #Continuum
↪  fitted

SALT_cont_norm = SALT_flux / SALT_cont_flux #Normalise the SALT flux
↪  by it's continuum
SALT_err_cont_norm = SALT_err / SALT_cont_flux #Normalise SALT
↪  errors by the continuum of the SALT flux to maintain
↪  Signal-Noise ratio

SALT_FWHM = 5 #SALT Telescope has an instumental resolution of 5
↪  Angstroms

wvlRange = h1['CRVAL1'] + np.array([0., h1['CDELT1']*(h1['NAXIS1'] -
↪  1)]) #Lower and upper wavelength of the SALT data

sMILES_FWHM = 2.5 #sMILES has a resolution of 2.5 Angstroms

sMILES_pix_size = 0.9000 #sMILES SSP's have a pixel size of 0.9000
↪  Angstroms

FWHM_diff = np.sqrt(SALT_FWHM**2 - sMILES_FWHM**2)
sigma = FWHM_diff/2.355/sMILES_pix_size #Sigma difference in pixels

alphaFe_path = 'sMILES_SSP/Mbi_iTp/OUT_aFe*/M/'

alphaFe_dir = glob.glob(alphaFe_path)
print(alphaFe_dir)
```

```
sMILES_list = [[] for i in range(len(alphaFe_dir))] #Create list of
↪  empty lists equal to same number of Alpha/Fe directories found
↪  in alphaFe_path

edge_array = np.array([3785.0, 6580]) #Wavelength range for edge
↪  masking
edge_limits = (SALT_wvl >= edge_array[0]) & (SALT_wvl <=
↪  edge_array[1]) #To reduce edge effects, data outside the
↪  wavelength range in edge_array are excluded

for alphaFe, aFe_list in zip(alphaFe_dir, sMILES_list):

    print('Processing files in directory: ' + alphaFe)

    #Create lists to store the parameters from the file names
    Reduced_chi_list = []
    Filenames_list = []
    Age_list = []
    Z_list = []
    cont_norm_SSP_list = []

    file_pattern = 'Mbi*[!_var]' #File path containing sMILES SSPs
    ↪  and defines pattern of the file names (Files with names
    ↪  beginning with "Mbi" and exlude those that end with "_var")
    SSP_path = alphaFe+file_pattern #Location of the sMILES SSPs
    ↪  found based on file naming pattern

    SSP_files = glob.glob(alphaFe+file_pattern) #Finds files based
    ↪  on the specified file path and naming pattern and create a
    ↪  list of these file names

    #Begin loop to process data from the files found in the
    ↪  SSP_files list
    for f in SSP_files:

        #Define filename of the current SSP file being processed
        original_filename = f[32:]

        #Prints name of the SSP file currently being processed for
        ↪  user feedback
        print('Processing file: ' + original_filename)

        #Define the var file of the respective SSP file
        var_f = f+'_var'


        #Read in SSP file and it's respective var file in read mode
        i = open(f, 'r')
```

```
j = open(var_f, 'r')

#Read lines of the SSP and var file ignoring those with
↪  strings     (i.e only lines containing numbers only are
↪  read in)
rows_ssp = [[float(number) for number in row.rstrip().split("
↪  ")] for row in i.readlines() if ft.can_convert(row)]
rows_var = [[float(number) for number in row.rstrip().split("
↪  ")] for row in j.readlines() if ft.can_convert(row)]

#Convert read in numbers to 2d numpy array containing the
↪  wavelengths and fluxes of the SSP file and var file
ssp = np.array([[row[0] for row in rows_ssp], [row[1] for row
↪  in rows_ssp]])
var = np.array([[row[0] for row in rows_var], [row[1] for row
↪  in rows_var]])

#Define the wavelengths and fluxes for the SSP file and var
↪  file
ssp_Wavelength = ssp[0]
ssp_Flux = ssp[1]
var_Wavelength = var[0]
var_Flux = var[1]

#Blur the flux of the SSP and var file using a Gaussian
↪  filter
blurred_ssp = ndimage.gaussian_filter1d(ssp_Flux, sigma)
blurred_var = ndimage.gaussian_filter1d(var_Flux, sigma)

#Rebin the flux of the SSP and var file to match the SALT
↪  observational data
rebinned_ssp = sp.spectres(SALT_wvl, ssp_Wavelength,
↪  blurred_ssp)
rebinned_var = sp.spectres(SALT_wvl, var_Wavelength,
↪  blurred_var)

#Define the final processed fluxes of the SSP and var files
processed_ssp = rebinned_ssp
processed_var = rebinned_var

#Containers for the SSP and var spectral data
ssp_spectrum = Spectrum1D(flux=processed_ssp*(u.erg /
↪  u.angstrom / u.cm ** 2 / u.s),
↪  spectral_axis=SALT_wvl*u.angstrom)
var_spectrum = Spectrum1D(flux=processed_var*(u.erg /
↪  u.angstrom / u.cm ** 2 / u.s),
↪  spectral_axis=SALT_wvl*u.angstrom)
```

```
        ssp_fit = fit_continuum(ssp_spectrum,
        ↪  model=Chebyshev1D(Chebyshev_NCONT))


        ssp_cont = ssp_fit(SALT_wvl*u.angstrom)

        #Continuum normalize processed flux of SSP and var file
        ↪  spectral data respectively
        ssp_cont_norm = processed_ssp/ssp_cont
        var_cont_norm = processed_var/ssp_cont


        #Calculate chi squared using function from local module
        chi = ft.chi_test(SALT_cont_norm[edge_limits],
        ↪  ssp_cont_norm[edge_limits],
        ↪  SALT_err_cont_norm[edge_limits])
        #print('Sum of Chi-squared: ')
        #print(chi)

        #Define degrees of freedom to be used in reduced chi squared
        ↪  calculations
        #Degrees of freedom = (Number of data points - number of
        ↪  varied parameters), varied parameters are Age, [/Fe] and
        ↪  Total Metallicity [M/H]
        dof = h1['NAXIS1'] - 3

        #Determine the reduced chi squared of the SSP file
        reduced_chi_squared = chi/dof

        #Extract the age and metallicity from the filename of the
        ↪  SSP
        file_Z, file_Age = ft.convert_Z_Age(original_filename)

        #Append the processed parameters to their respective list so
        ↪  they can be used in plots later
        Reduced_chi_list.append(reduced_chi_squared.value)
        Age_list.append(file_Age)
        Z_list.append(file_Z)
        cont_norm_SSP_list.append(ssp_cont_norm)
        Filenames_list.append(original_filename)

        #Close the SSP and var files
        i.close()
        j.close()

    #Append the list of parameters of all SSPs in the abundance
    ↪  direcory being looped to the sMILES_list
```

```
    aFe_list.append(Reduced_chi_list)
    aFe_list.append(Age_list)
    aFe_list.append(Z_list)
    aFe_list.append(cont_norm_SSP_list)
    aFe_list.append(Filenames_list)



output_f = open("sMILES_fit_output.txt", "a") #Creates the output
↪  file where the parameters of the best fitting SSPs from each
↪  Alpha/Fe abundance will be written to
output_f.write('[alpha/Fe] Directory'+'\t'+'Filename'+'\t'+'Age
↪  (Gyrs)'+'\t'+'Metallicity'+'\t'+'Reduced Chi-Squared'+'\n')
↪  #Writes the name of the parameter with tab spacing


#Begin loop to make plots for each Alpha/Fe abundance
for aFe, aFe_directory in zip(sMILES_list, alphaFe_dir):

    #Define the lists for each parameter in the Alpha/Fe abundance
    ↪  that is currently being worked on
    Red_chi_sqr = np.array(aFe[0])
    Age = np.array(aFe[1])
    Z = np.array(aFe[2])
    SSP_Flux = np.array(aFe[3])
    Filename = np.array(aFe[4])


    #Creates 3d plot of surfce with metallicity, age and reduced
    ↪  chi-squared
    plt.figure(dpi=200)
    ax=plt.axes(xlabel='Z Metallicity', ylabel='Age (GYrs)',
    ↪  zlabel='Reduced Chi-Squared', projection ='3d')#
    ↪  title=aFe_directory[23:29], projection ='3d')
    print(aFe_directory[23:29])
    ax.plot_trisurf(Z, Age, Red_chi_sqr, cmap=cm.jet_r,
    ↪  label=aFe_directory[23:29])
    ax.invert_zaxis()
    ax.view_init(210, 240)
    #plt.savefig(aFe_directory[23:29]+'_3d_plot.png',
    ↪  bbox_inches="tight") #Save the plot as a .png file with name
    ↪  based off the Alpha/Fe abundance that the plot is for
    plt.show()


    #Creates a scatter plot of age vs metallicity with colours of
    ↪  the points qualitatively indicating their reduced
    ↪  chi-squared
```

```
plt.figure(dpi=200)
plt.axes(xlabel='Z Metallicity', ylabel='Age (GYrs)')#,
↪  title=aFe_directory[23:29])
print(aFe_directory[23:29])
plt.scatter(Z, Age, c=Red_chi_sqr, s=10, cmap=cm.jet_r,
↪  label=aFe_directory[23:29])
plt.colorbar(label='\u03C7 $^2$')
plt.gca().invert_yaxis()
#ax.view_init(210, 240)
#plt.savefig(aFe_directory[23:29]+'_scatter_plot.png',
↪  bbox_inches="tight")
plt.show()


min_chi_index =np.argmin(Red_chi_sqr) #Defines the index of the
↪  lowest reduced chi-squared in the list of reduced
↪  chi-squareds


#Plots processed SALT data, best fitting SSP and the residuals
↪  all on the same plot with an offset in each to show them
↪  more clearly
plt.figure(dpi=200)
plt.axes(xlabel='Wavelength (Angstrom)', ylabel='Normalized
↪  Flux')
plt.grid()
plt.plot(SALT_wvl[edge_limits],
↪  SSP_Flux[min_chi_index][edge_limits], label='Best Fitting SSP
↪  Continuum Normalized')
plt.plot(SALT_wvl[edge_limits], SALT_cont_norm.value[edge_limits]
↪  - 1.5, label='SALT Continuum Normalized')
plt.plot(SALT_wvl[edge_limits],
↪  (SALT_cont_norm.value[edge_limits]-SSP_Flux[min_chi_index][edge_limits]
↪  - 1.5, label='Residuals (SALT - sMILES SSP)')

#4 black vertical dashed lines are placed on the plots to
↪  indicate where there are gaps in the detectors
plt.axvline(x=4663.9, linestyle= '--', color = 'black')
plt.axvline(x=4734.7, linestyle= '--', color = 'black')
plt.axvline(x=5690.3, linestyle= '--', color = 'black')
plt.axvline(x=5753.8, linestyle= '--', color = 'black')

plt.legend(bbox_to_anchor=(0,1.01), loc='lower left')
#plt.savefig(aFe_directory[23:29]+'_best_fit.png',
↪  bbox_inches="tight")

#Used to zoom in on a specific region, useful for when trying to
↪  indentify spectral lines
```

```
    #plt.xlim([4220, 4230])
    #plt.ylim([0.1, 1.5])
    plt.show()


    #Plots the best fitting SSP before and after the edge limits
    ↪ have been applied
    plt.figure(dpi=200)
    plt.axes(xlabel='Wavelength (Angstrom)', ylabel='Normalized
    ↪ Flux')
    plt.grid()
    plt.plot(SALT_wvl, SSP_Flux[min_chi_index], label='SSP pre-mask')
    plt.plot(SALT_wvl[edge_limits],
    ↪ SSP_Flux[min_chi_index][edge_limits] - 1.5, label='SSP post
    ↪ edge masking')

    plt.axvline(x=4663.9, linestyle= '--', color = 'black')
    plt.axvline(x=4734.7, linestyle= '--', color = 'black')
    plt.axvline(x=5690.3, linestyle= '--', color = 'black')
    plt.axvline(x=5753.8, linestyle= '--', color = 'black')

    plt.legend(bbox_to_anchor=(0,1.01), loc='lower left')
    #plt.savefig(aFe_directory[23:29]+'_SSP'+'_masking.png',
    ↪ bbox_inches="tight")

    #plt.xlim([4200, 4400])
    #plt.ylim([0.1, 1.5])
    plt.show()



    ↪ output_f.write(aFe_directory[23:29]+'\t'+Filename[min_chi_index]+'\t'+s
    ↪ decimals=2))+'\n') #Writes the parameters of the best
    ↪ fitting SSP to the output file with tab spacing
output_f.close()


#Plots the processed SALT data before and after the edge limits
↪ have been applied
plt.figure(dpi=200)
plt.axes(xlabel='Wavelength (Angstrom)', ylabel='Normalized Flux')

plt.grid()
plt.plot(SALT_wvl, SALT_cont_norm.value, label='SALT data before edge
↪ mask applied')
plt.plot(SALT_wvl[edge_limits], SALT_cont_norm.value[edge_limits] -
↪ 1.5, label='SALT data with applied edge mask')
```

```
    plt.axvline(x=4663.9, linestyle= '--', color = 'black')
    plt.axvline(x=4734.7, linestyle= '--', color = 'black')
    plt.axvline(x=5690.3, linestyle= '--', color = 'black')
    plt.axvline(x=5753.8, linestyle= '--', color = 'black')

    plt.legend(bbox_to_anchor=(0,1.01), loc='lower left')
    #plt.savefig('SALT'+'_masking.png', bbox_inches="tight")

    #plt.xlim([4200, 4400])
    #plt.ylim([0.1, 1.5])
    plt.show()

    SNR= (SALT_cont_norm.value)/SALT_err_cont_norm.value

    #Plots the Signal-Noise ratio of the SALT data
    plt.figure(dpi=200)
    plt.axes(xlabel='Wavelength (Angstrom)', ylabel='Signal-Noise')
    #plt.title('SALT Signal-Noise ratio', loc='right')
    plt.grid()
    plt.plot(SALT_wvl, SNR, label='SALT Signal-Noise')
    plt.axvline(x=4663.9, linestyle= '--', color = 'black')
    plt.axvline(x=4734.7, linestyle= '--', color = 'black')
    plt.axvline(x=5690.3, linestyle= '--', color = 'black')
    plt.axvline(x=5753.8, linestyle= '--', color = 'black')

    plt.legend(bbox_to_anchor=(0,1.01), loc='lower left')
    #plt.savefig('SALT_SNR'+'.png', bbox_inches="tight")

    #plt.xlim([4220, 4230])
    #plt.ylim([0.1, 1.5])
    plt.show()

    #Prints the median of the Signal-Noise ratio
    print('Median of the SNR:', np.round(np.median(SNR), decimals=2))




sMILES_SSP_fitting()
```

8.2 Local module

```
# -*- coding: utf-8 -*-
"""
@author: Olliver
```

```
This module was developed and intended to be used with the software
↪  sMILES SSP fitting.

Olliver Hughes 22 Jan 2022 - Function to convert SSP files added
Olliver Hughes 25 Mar 2022 - Function for Chi-squared test added
Olliver Hughes 16 Apr 2022 - Function to extract the Age and
↪  Metallicity from filenames following sMILES naming convention added

"""

import numpy as np

#Checks if the line can be converted and ignores lines with text (i.e.
↪  Headers, comments etc)
def can_convert(row_text):

    for entry in row_text.rstrip().split('        '):

        try:

            float(entry)
        except:

            return False


    return True


#Performs Chi-squared test and returns the sum of all chi-squareds to
↪  be used in calculating reduced chi-squared
def chi_test(obs_flux, model_flux, error):

    chi_array = ((obs_flux-model_flux))**2/(error)**2

    chi_squared_sum = np.sum(chi_array)

    return chi_squared_sum


#Extracts metallicity and age from input filename then converts and
↪  returns them as floating point numbers
def convert_Z_Age(filename):
    Z = filename[8:13]
    Age = filename[14:21]
```

```
    #Reads the letter before the metallicity number and decides if the
    ↪  number if positive or negative based on if it is p or m
    if Z[0] == 'm':
        Z_converted = np.negative(float(Z[1:]))
    elif Z[0] == 'p':
        Z_converted = np.positive(float(Z[1:]))
    else:
        print('Filename does not follow sMILES naming convention')

    Age_float = float(Age)

    return Z_converted, Age_float
```

8.3 Project proposal

The fitting of G422436 to the sMILES semi-empirical SSPs

Olliver Hughes

February 9, 2022

1 Project Description

Spectroscopy is the analysis of stellar spectra, how these spectra correlate to the chemical composition, temperature and other properties of stars. By measuring the light and wavelengths that a star gives off it's spectra is produced. To measure the light and the wavelengths of the light an instrument called a spectrograph typically with a telescope. The spectrograph measures the light by first making the rays of the light parallel through the use of a slit and then the light is dispersed, typically a plane diffraction grating is used for this as it's luminous efficiency is higher than other dispersion devices. Once the light is dispersed it is then recorded by detector.

The development of software written in python for the fitting of observations of the G422436 galaxy from the SALT telescope to SSPs (simple stellar populations) from the sMILES library, this software will also deal with the binning and blurring of the data. Simple stellar populations are a population of stars that have the same age, $\left[\frac{Fe}{H}\right]$ metallicity (iron to hydrogen ratio) and abundance patterns. The goodness of the fits will be determined via a χ^2 test, performed at each grid point, however other methods to determine this will be researched and discussed.

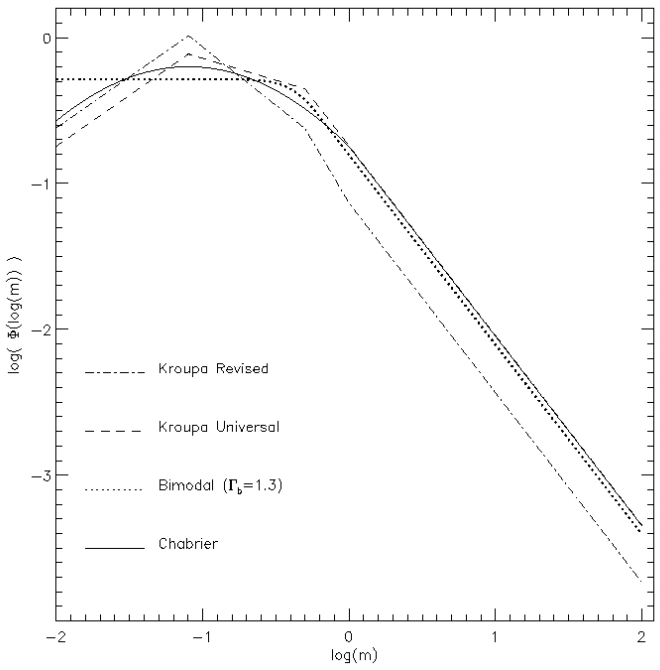

Figure 1: Example of IMF Types (From MILES Library website)

The sMILES SSPs have been computed with a wavelength range of 3540.5-7409.6 Å with a bimodal IMF (Initial Mass Function) type. The IMF describes the distribution of initial masses of a stellar population. The IMF is typically plotted with the number density on the y axis and mass on the x axis and from these plots the number of stars at a particular mass range can be determined as well as the number fraction and mass fraction of stars within a set range of masses (for example, the number fraction of stars with masses in the $5M_{\odot}$ to $10M_{\odot}$ range). Fig.1 shows some of the different types of IMF which are used for the analysis of stellar populations.

The sMILES library contains spectra categorized into 5 different abundances of $\left[\frac{\alpha}{Fe}\right]$, the α elements are the elements created from the nuclear fusion of helium to create heavier elements, these heavier elements are the α elements. They are called this as their most abundant isotope is made up of an integer number of alpha particles (He nuclei). This process is called the triple alpha process and occurs in red giant stars. The Fe peak elements are the elements with high abundance with atomic numbers around the same as Fe (elements from Cr to Ni) hence their name, and they are produced from stellar nucleosynthesis

In order to fit the observational data to these semi-empirical spectra the software will make use of many scientific python modules such as scipy and astropy, with similar software such as pPXF (Cappellari, 2017) being used as a guide to help with the development of this software.

2 Reasons For Research

Galaxies can appear redder as their age, metallicity or their dust abundance increases. Any of these can may be indicators as to why a galaxy appears red making difficult to determine the difference between young galaxies with high metallicities and old galaxies with low metallicities for example. The fitting of the observed G422436 galaxy with the SALT telescope to the sMILES SSPs will help further the understanding of abundance patterns of other galaxies with SSPs.

The development of this software will provide a stepping stone towards a useful tool for astronomers to fit observed data to theoretical or semi-empirical SSPs. The $\left[\frac{\alpha}{Fe}\right]$ abundance can give indications on the stellar formation history of galaxies which can then be compared to other galaxies with $\left[\frac{Fe}{H}\right]$ abundances.

3 Literature Survey

Previous works include the development of the software pPXF (Penalized Pixel-Fitting) (Cappellari, 2017). The pPXF software uses an improved method for full spectrum fitting to extract the stellar and gas kinematics, and the stellar population of galaxies. The method used is an improvement over the previous method used (Cappellari & Emsellem, 2004) as it directly computes a Fourier transform to be used with the convolution theorem.

Convolution theorem:

$$O(\lambda) = S(\lambda) * F(v(\lambda)), \text{where } * \text{ means that they are convolved}$$

$$\tilde{O}(\lambda) = \tilde{S}(\lambda) \times \tilde{F}(\lambda)$$

Where $\tilde{O}$, $\tilde{S}$ and $\tilde{F}$ are the Fourier transforms of O, S and F respectively.

The aforementioned paper (Cappellari, 2017) also details that the velocities can be accurately

determined regardless of the velocity dispersion σ which is a key improvement from the previous method used in pPXF.

The work of Conroy (Conroy, 2013) provides an overview of the SPS (Stellar Population Synthesis) technique and which parameters can be accurately measured from galaxy SEDs (Spectral Energy Distributions). The purpose of the SPS method is to determine many properties from observed SEDs, these properties can include the star-formation history, metallicity, abundance pattern, dust mass and more. This technique is unsuitable for this project as it requires resolved SEDs, whereas unresolved observations are provided for this project hence why the development of software for this is required.

sMILES is a library of semi-empirical stellar spectra based on the MILES library created specifically for the study of stellar populations (Knowles et al., 2021). It contains 801 spectra in 5 categories of $\left[\frac{\alpha}{Fe}\right]$ abundances ranging from -0.2 to 0.6 dex with wavelengths ranging from 3540.5 to 7409.6 Å. This library is called semi-empirical as it created based on predictions of the differential abundance of the MILES library.

The work of Worthey (Worthey, 1994) details the construction of various stellar population models. It describes populations for main-sequence and red giant branch isochrones as well as horizontal branch and asymptotic giant branch populations.

4 Project Details

4.1 Work Location

LE204 Preston, University of Central Lancashire. (Monday morning and afternoon, and Thursday mornings)

4.2 Equipment

- Computers provided in LE204
- Python 2.7 via Starlink SSH
- Personal Computer
- Spyder Python 3.9

4.3 Plan of Action

Table.1 shows the planned deadlines for the Project.

Date	Goal
09/02/2022	Submit Project Proposal
17/02/2022	Achieve binning and blurring of a single data file
27/02/2022	Research methods for reading in and binning multiple template spectra
07/03/2022	Achieve read in of multiple template spectra for binning and blurring
16/03/2022	Research literature
20/03/2022	Research methods for fitting spectra and decide which approach would be best for this project
02/04/2022	Apply the chosen fitting method in the software and make final changes to the software
10/04/2022	Perform a $\chi 2$ test to determine the goodness of the fits
16/04/2022	Plot results of the fits
20/04/2022	Submit draft Project Report
27/04/2022	Feedback of draft report is given
30/04/2022	Make changes to Project Report based on feedback from draft
04/05/2022	Submit Project Report
18/05/2022	MPhys Project Viva

Table 1: Planned Project deadlines

4.4 Arrangements

Meetings with the Project Supervisor (Anne Sansom) are set twice a week. One meeting every Monday at 10am and another every Thursday at 10am. All meetings in LE204, Leighton Building, University Of Central Lancashire.

5 References

Cappellari, M. (2017) 'Improving the full spectrum fitting method: accurate convolution with Gauss-Hermite functions', Monthly Notices of the Royal Astronomical Society, 466, pp. 798-811. doi: 10.1093/mnras/stw3020; eprintid: arXiv:1607.08538.

Cappellari, M. and Emsellem, E. (2004) 'Parametric Recovery of Line-of-Sight Velocity Distributions from Absorption-Line Spectra of Galaxies via Penalized Likelihood', Publications of the Astronomical Society of the Pacific, 116, pp. 138-147. doi: 10.1086/381875; eprintid: arXiv:astro-ph/0312201.

Conroy, C. (2013) 'Modeling the Panchromatic Spectral Energy Distributions of Galaxies', Annual Review of Astronomy and Astrophysics, 51(1), pp. 393-455. doi: 10.1146/annurev-astro-082812-141017.

MILES Library. Available at: `http://research.iac.es/proyecto/miles/` (Accessed: 06/02/2022).

Knowles, A.T., Sansom, A.E., Allende Prieto, C. and Vazdekis, A. (2021) 'sMILES: a library of semi-empirical MILES stellar spectra with variable [α/Fe] abundances', Monthly Notices of the Royal Astronomical Society, 504, pp. 2286-2311. doi: 10.1093/mnras/stab1001; eprintid: arXiv:2104.04822.

Worthey, G. (1994) 'Comprehensive Stellar Population Models and the Disentanglement of Age and Metallicity Effects', The Astrophysical Journal Supplement Series, 95, pp. 107. doi: 10.1086/192096.