Problem Set 6

Dario Loprete

October 15, 2024

GitHub link

`https://github.com/dariolop76/phys-ga2000`

1 Part a

We load the data using the package `astropy` and plot in Figure 1 the flux vs the wavelength for galaxies 1, 2, 3, 4 and 5.

By using the relation $\lambda(nm) = 10^{\lambda(\mathring{A})} \cdot 0.1$, we can obtain the value of the wavelength in nanometers. Hence, we can compare where the peaks of the various spectra are located to the hydrogen spectrum, namely the spectrum of emitted radiation due to transitions between different quantum levels.

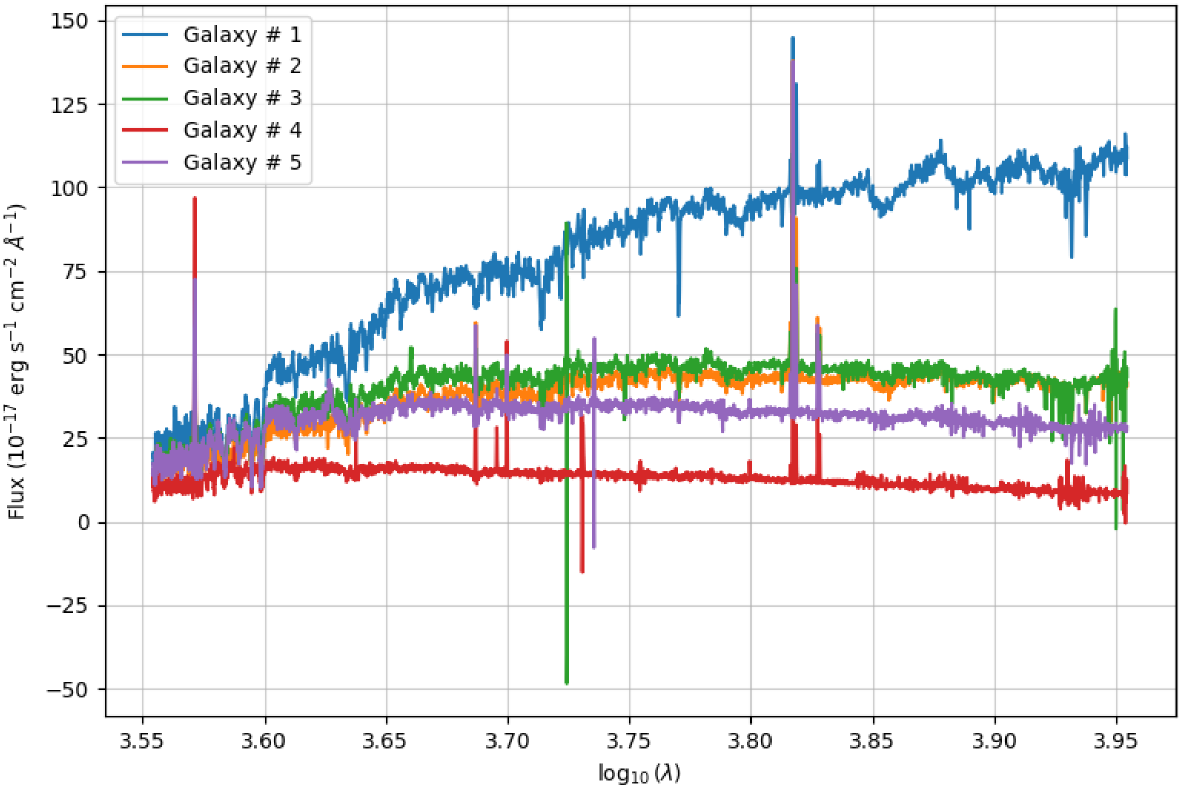

Figure 1: Flux plotted against the $log_{10}(\lambda)$, where λ is the wavelength in $\AA$, for galaxies 1, 2, 3, 4 and 5.

2 Part b

We perform the normalization of the data by computing the integral of the data over the wavelength interval, using the trapezoidal rule (`numpy.trapz`). We then divide our data by the found normalization constants. Figure 2 shows the normalized data plotted against the wavelength interval.

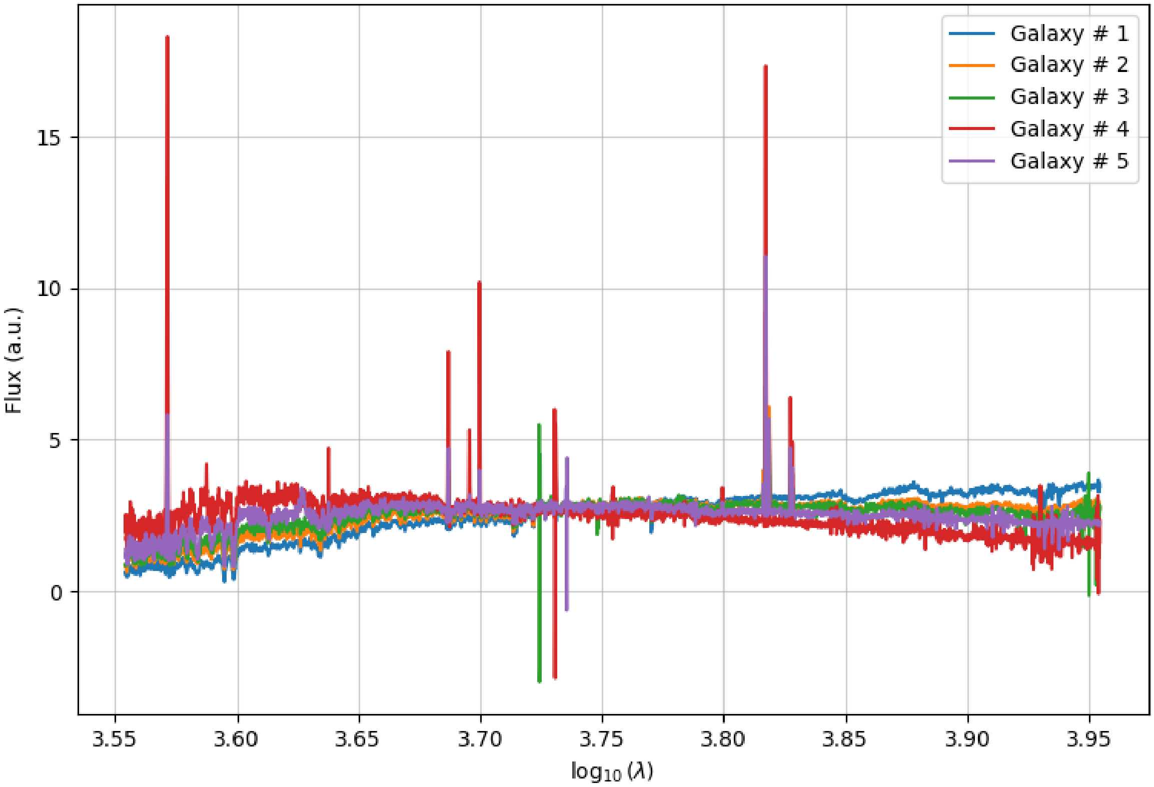

Figure 2: Normalized flux plotted against the $log_{10}(\lambda)$, where λ is the wavelength in $\mathring{A}$, for galaxies 1, 2, 3, 4 and 5.

3 Part c

We compute the mean value of the spectra for each value and subtract it to the normalized flux, to obtain the residuals, which are shown in Figure 3.

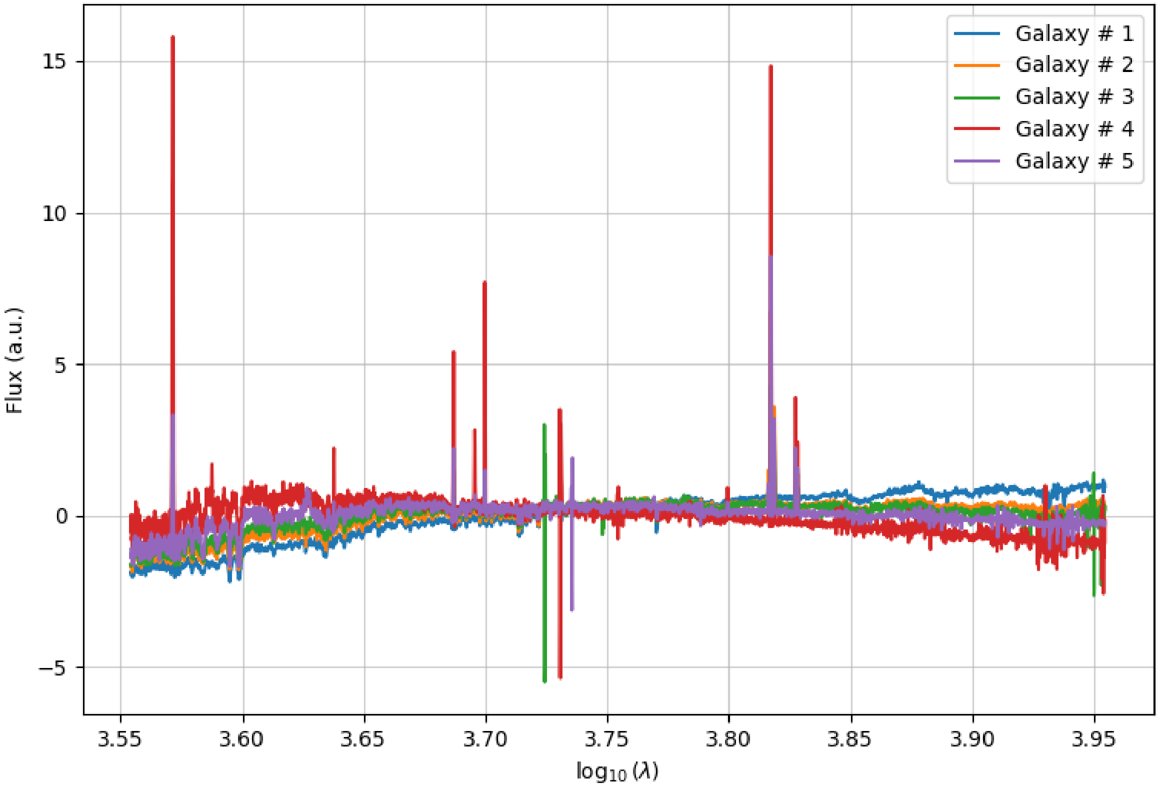

Figure 3: Residuals plotted against the $log_{10}(\lambda)$, where λ is the wavelength in $\mathring{A}$, for galaxies 1, 2, 3, 4 and 5.

4 Part d

We compute the covariant matrix $C = R^t R$, where R is the matrix containing the residuals, and find its eigenvectors. Figures 4 - 8 show the first 5 eigenspectra.

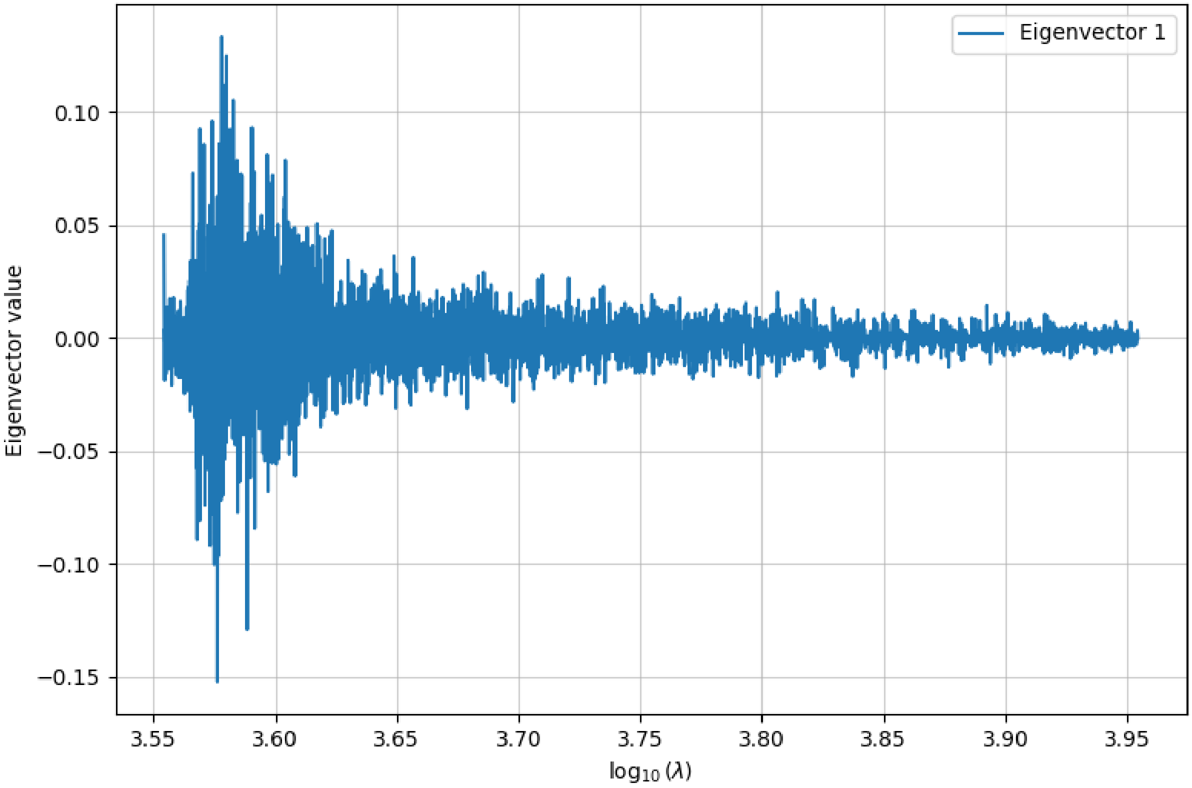

Figure 4: Eigenspectrum 1.

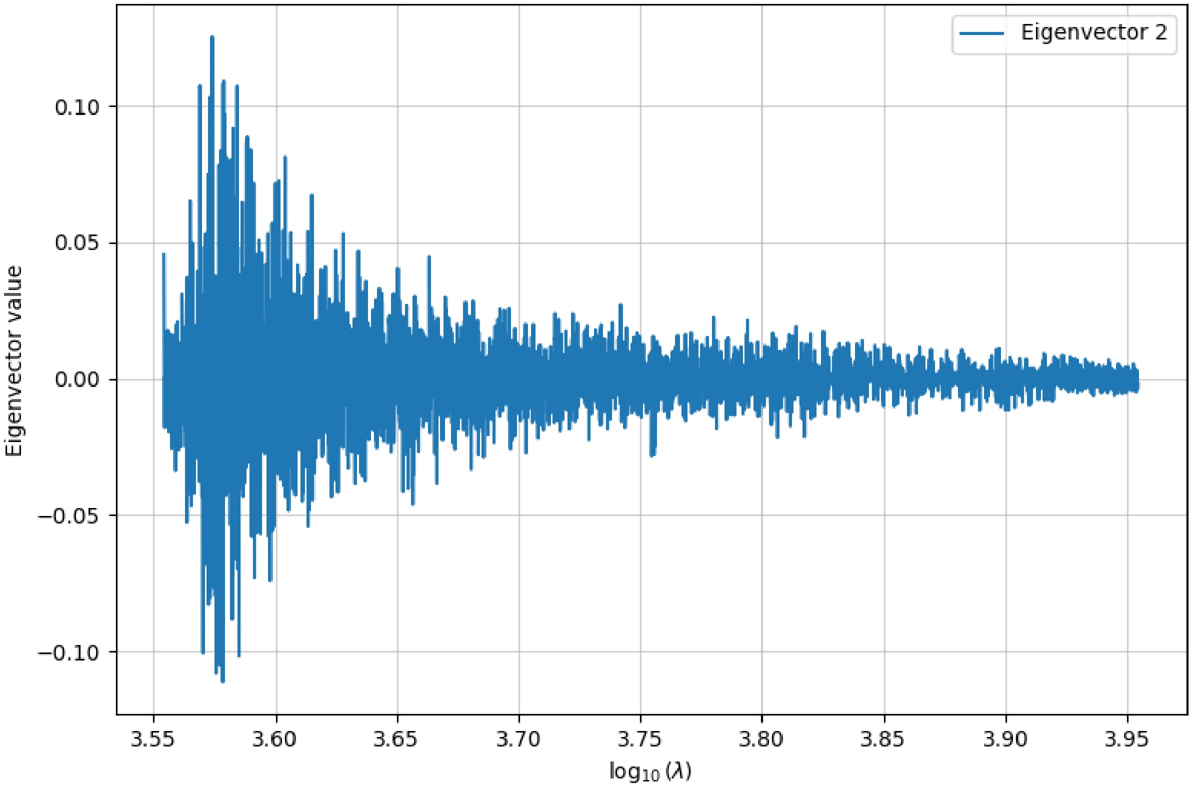

Figure 5: Eigenspectrum 2.

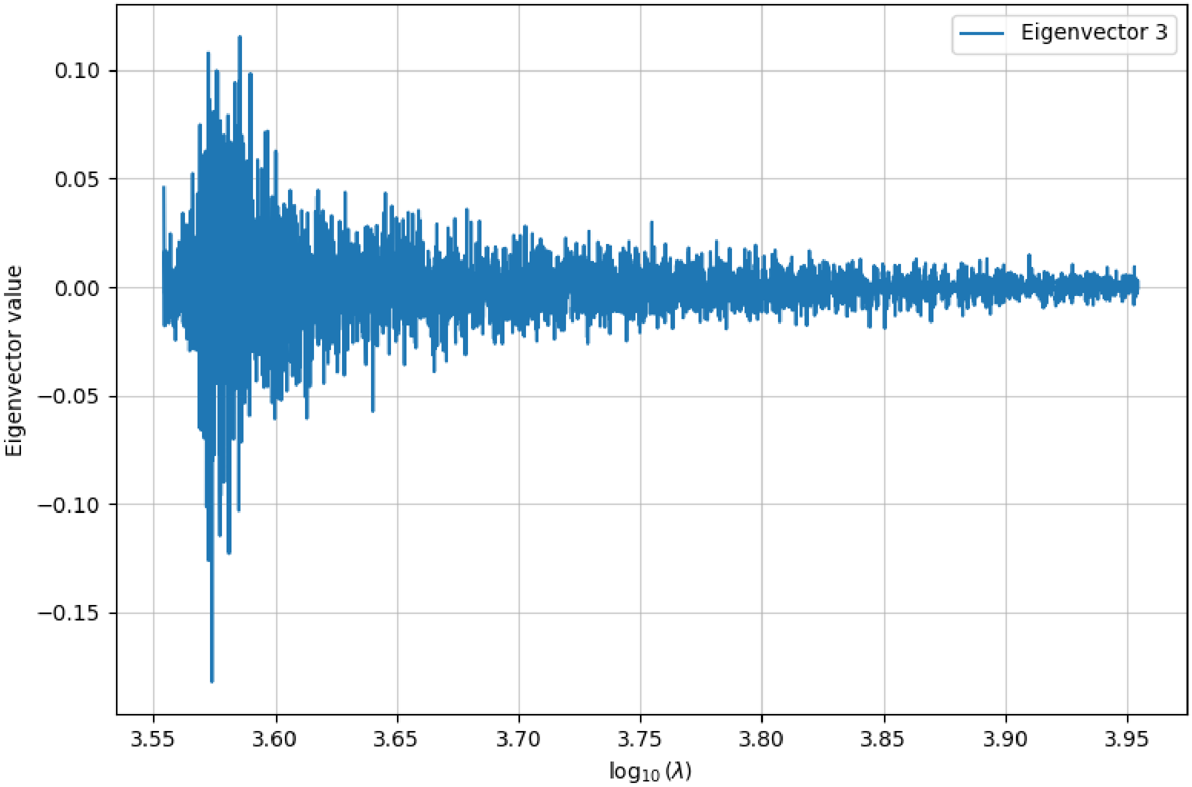

Figure 6: Eigenspectrum 3.

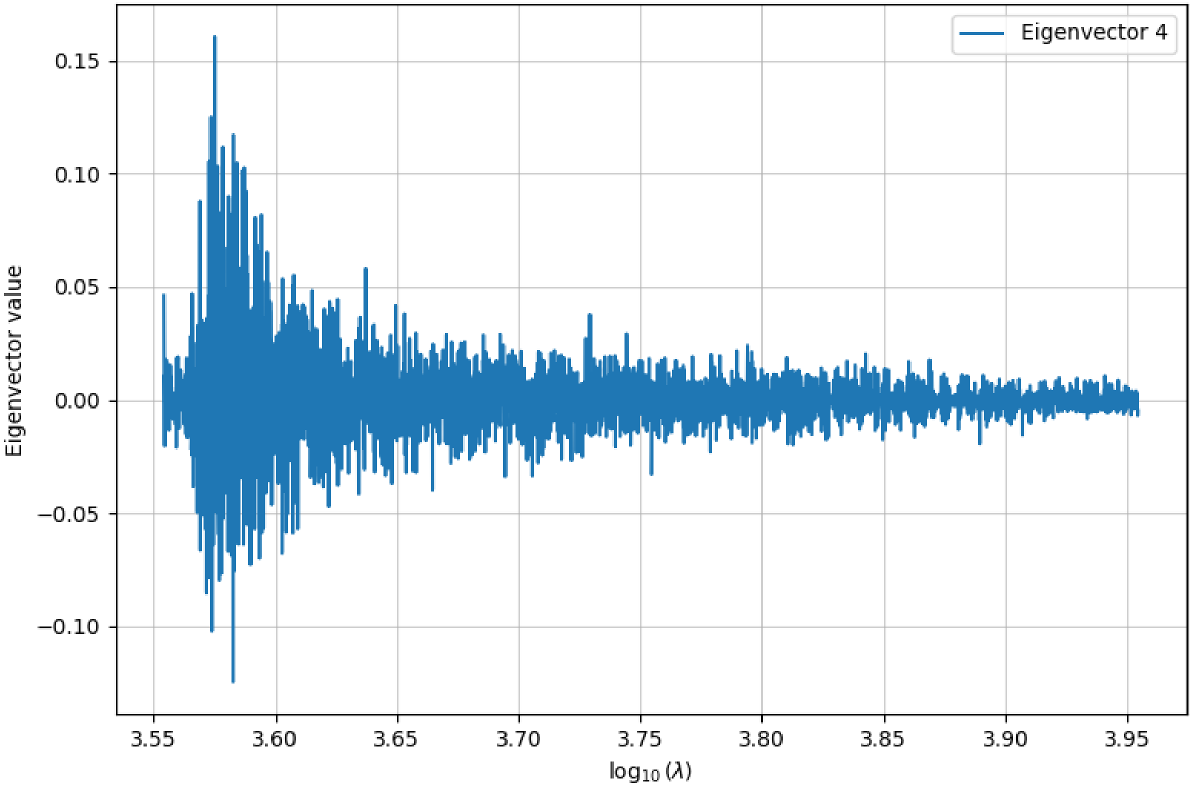

Figure 7: Eigenspectrum 4.

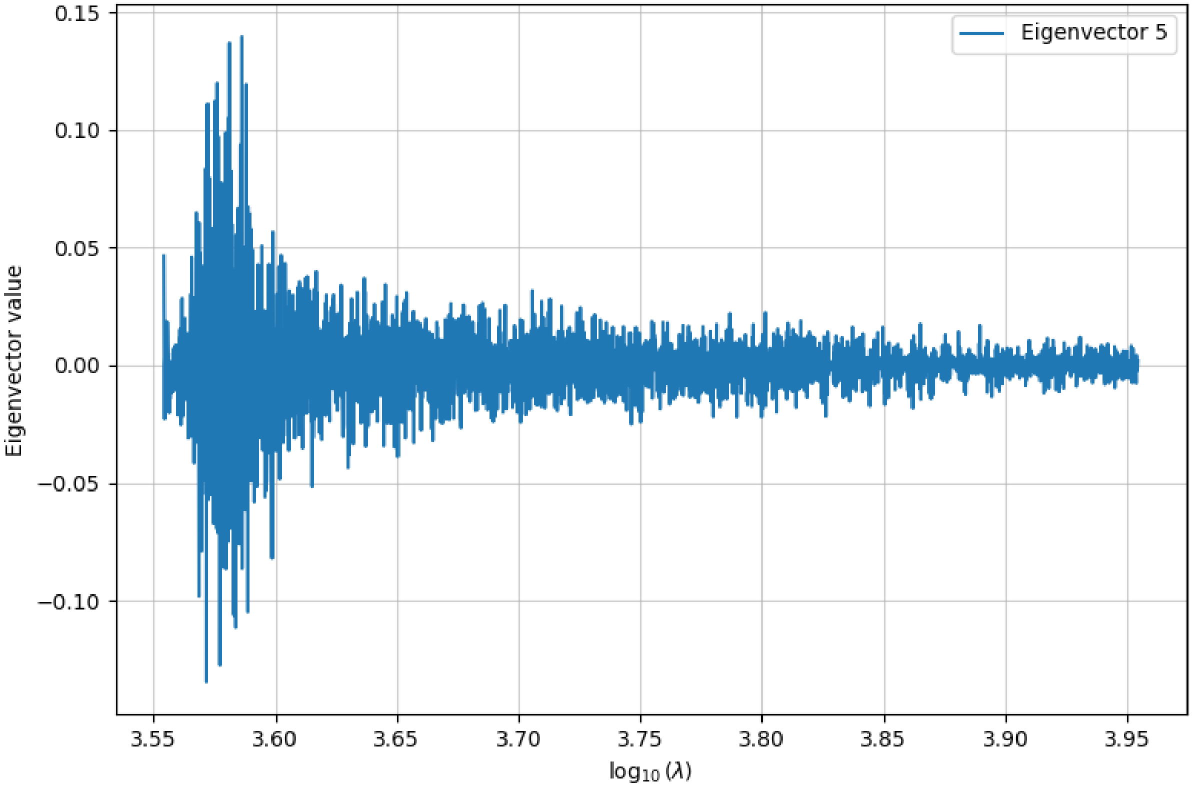

Figure 8: Eigenspectrum 5.

5 Part e

We apply the SVD decomposition to the matrix R containing the residuals: $R = UWV^t$. The matrix V contains the eigenvectors of our interest. We check that these are the same of those obtained from the covariant matrix approach. We do this by using the method `numpy.allclose`, which returns `True` if two arrays are element-wise equal within a tolerance. Additionally, we use the package `timeit` to compare the execution time between the two approaches. We obtain:
C matrix approach: $\Delta t = 71$ s
SVD approach: $\Delta t = 153$ s

6 Part f

Even though from the comparison between the execution times it seems that the C matrix approach allows for a quicker evaluation of the eigenspectra than the SVD decomposition, we should also look at the condition numbers to properly compare these two methods:
Condition number (C): $1.6 \cdot 10^{18}$
Condition number (R): $3 \cdot 10^{15}$
We notice that the condition number of C is bigger than that of R by three orders of magnitude. Therefore, using the covariant matrix can lead to numerical instability.

7 Part g

We can reconstruct the flux spectrum by using the following expansion:

$$f = f_N(f_m + \sum_i c_i e_i), \tag{1}$$

where f is the flux, f_N is the renormalization constant (computed in Part b), f_m is the mean (computed in Part c) and $\sum_i c_i e_i$ are the residuals written in terms of the eigenvectors e_i of the matrix R, with c_i being the corresponding coefficients. By truncating this sum, we can obtain an approximate spectrum. For this part, we keep only the first 5 terms of this sum. Figure 9 shows the flux spectrum for galaxy 1 and its approximation.

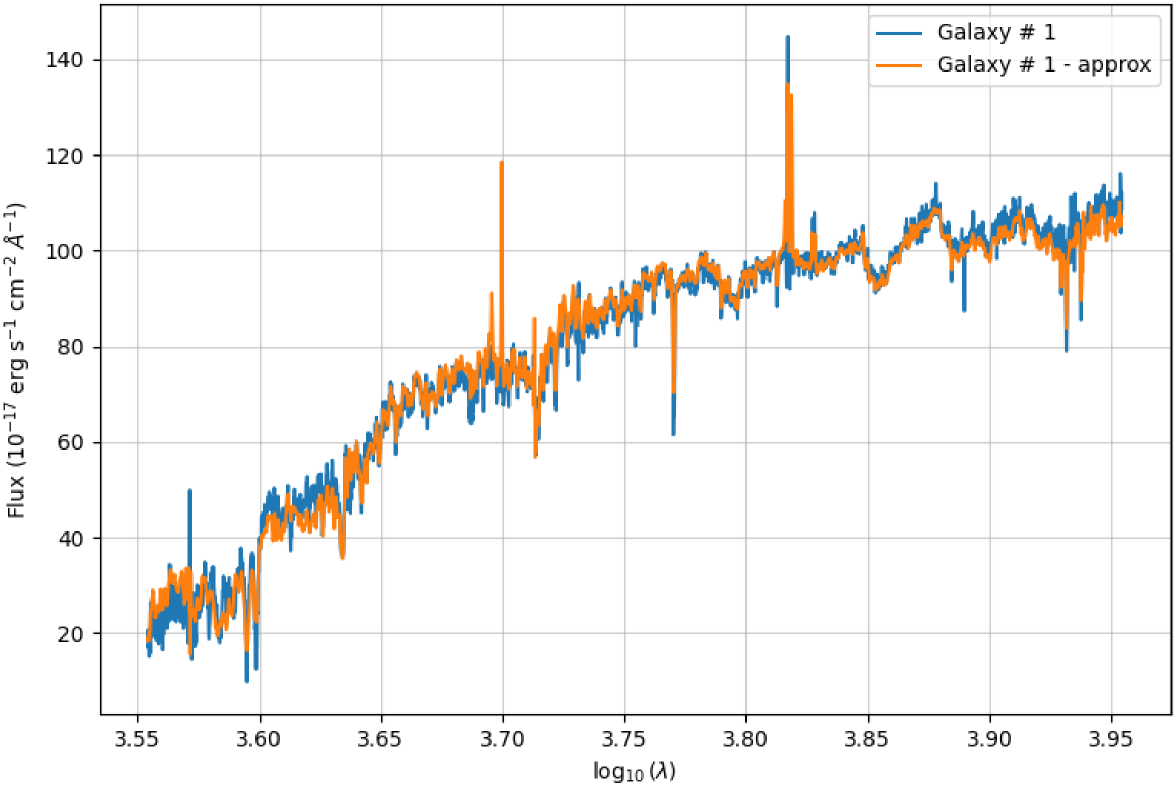

Figure 9: Comparison between the actual data and the approximate spectrum obtained by keeping only the first 5 eigenvectors, for galaxy 1.

8 Part h

Figure 10 shows the coefficient c_0 vs c_1, while Figure 11 shows c_0 vs c_2.

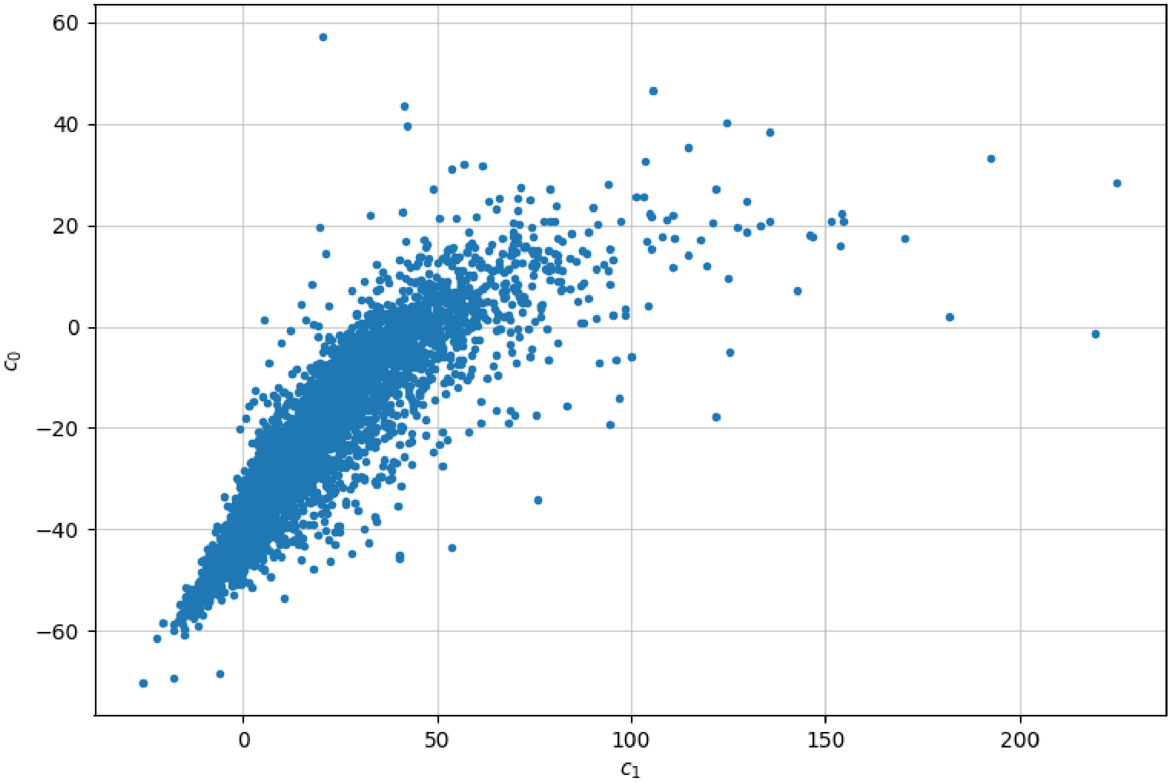

Figure 10: c_0 vs c_1

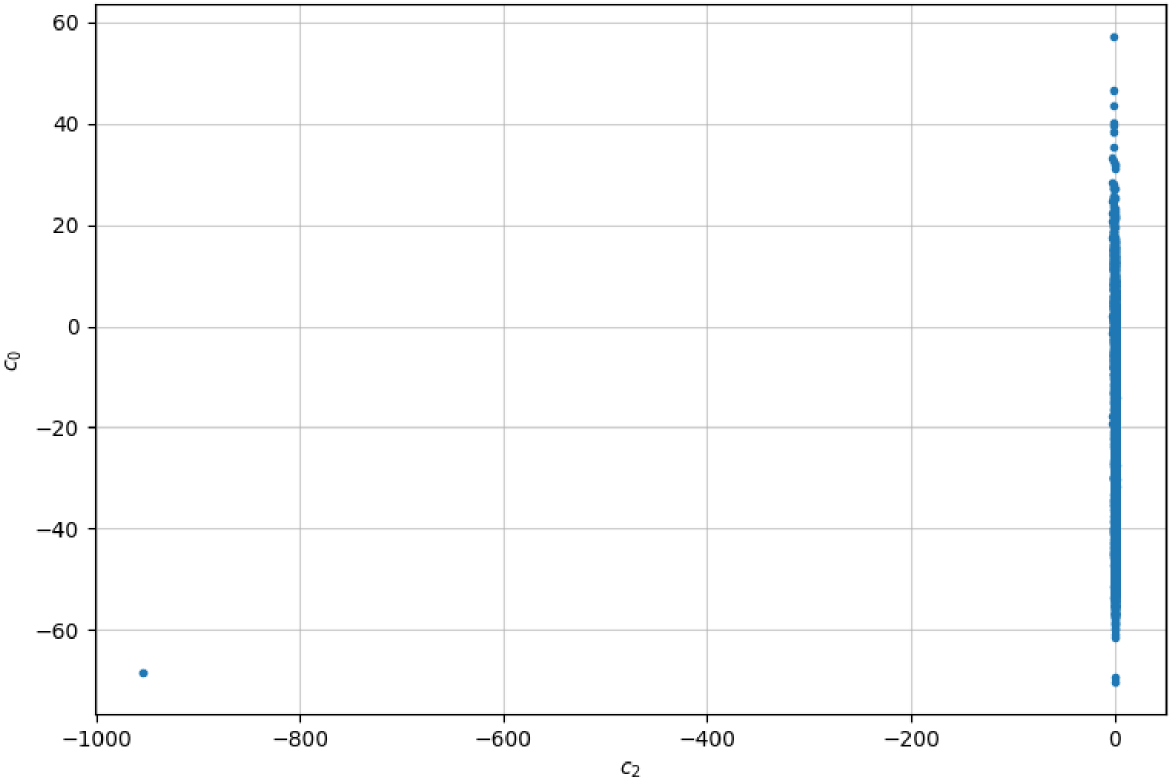

Figure 11: c_0 vs c_2

9 Part i

We vary the number of principal components in the interval $N_c \in \{1, 2, 3..., 20\}$ and compute the Root Mean Square (RMS) for each case, where the RMS is defined as

$$RMS = \sqrt{\frac{\sum_{j=1}^{N_{data}} (f_j - \tilde{f}_j)}{N_{data}}}, \tag{2}$$

where N_{data} is the number of data points, f_j are the values of the flux and $\tilde{f}_j$ are the values of the approximate spectrum. Figure 12 shows the obtained values for the RMS as a function of the number N_c of principal components, for galaxy 1.

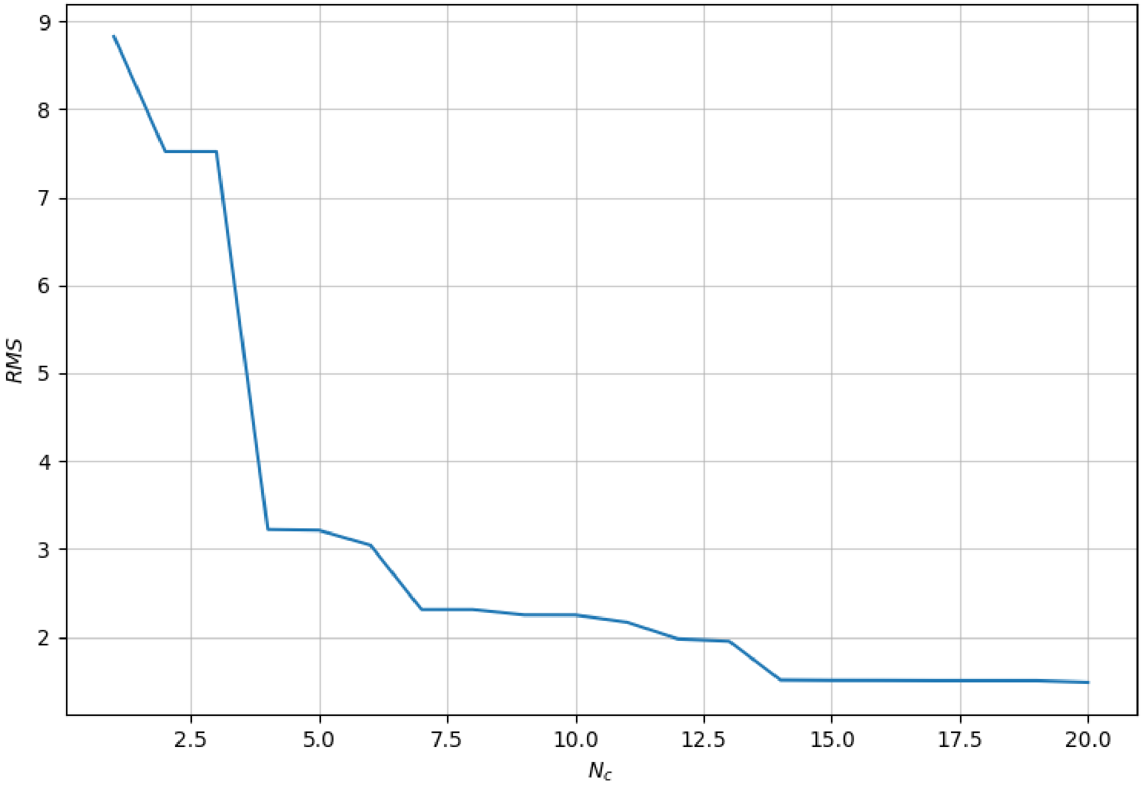

Figure 12: RMS as a function of N_c in the range $[1, 20]$, for galaxy 1.

As we can see, the RMS decreases as N_c increases. This reflects the fact that keeping more principal components allows for a better approximation of the data.

The value of the RMS for $N_c = 20$ is 1.5.