Version B1

Procedure for analyzing data cubes after the reduction process

The procedure here detailed describes the treatment that must be applied to data cubes, after the reduction process, in order to extract information from it, with great precision. This document comes with a programs pack, written in IDL language, which are used along the procedure here described. Each one of these programs has a Help file (in pdf format), which describes its functioning in details.

1) Differential atmospheric refraction correction

In this step, the effect of the differential atmospheric refraction, existent in data cubes after the reduction process, is corrected. To do that, first of all, it is necessary to obtain the functions that describe the variation of the coordinates x and y of the centroids of the data cube images as a function of the wavelength. From these functions, then, the effect of the differential atmospheric refraction can be corrected. This procedure can be done either by the experimental method or by the theoretical method, which are described below.

1a) Experimental method

This method consists in obtaining the coordinates of the centroids of the data cube images in different wavelengths and, after that, in fitting a third degree function to these experimental points. So, first of all, it's necessary to select images in different wavelengths of the data cube and determine the coordinates x and y of its centroids. These images must be selected in a way that its wavelengths does not match to any existent spectral line, besides, the wavelength interval between them don't need to be regular, but the images must be taken in a proper way to cover the entire spectral range of the data cube. It is important to mention that the coordinates to be taken can be of any punctual structure existent in the images (in other words, this structure doesn't need to be, necessarily, in the central region o the images). It's not recommendable to take just one image to each wavelength of the data cube, because the low photon statistic in each one of them can make difficult to determine the precise coordinates of the centroid. A better strategy to determine the centroids' coordinates is to choose short wavelength intervals (of about 10 Angstroms, for example) along the data cube and sum the images in these intervals. To each one of these images must be associated the mean wavelength of the interval used to make it. The construction o these images can be done using the program "make_images_plus", written for IDL, although this process can also be performed using basic tasks of the software IRAF.

Once the images, along the entire spectral range of the data cube, have been created, the coordinates of their centroids (X_{cent}, Y_{cent}) must be determined. This procedure can be performed using, for example, the task "imexamine" of the software IRAF, following the below steps:

- open the image with SaoImage
- apply the task "imexamine" in IRAF terminal
- position the cursor on the center of the structure whose centroid coordinates must be determined
- type "j" to obtain an estimation of the coordinate x of the centroid (X_{cent})
- type "k" to obtain an estimation o the coordinate y of the centroid (Y_{cent})

After that, two third degree functions of wavelength must be adjusted to X_{cent} and Y_{cent}, in the following way:

$$X_{cent} = A_x + B_x \cdot \lambda + C_x \cdot \lambda^2 + D_x \cdot \lambda^3 \qquad (1)$$

$$Y_{cent} = A_y + B_y \cdot \lambda + C_y \cdot \lambda^2 + D_y \cdot \lambda^3 \qquad (2)$$

This procedure must be applied, independently, to all data cubes that will be combined after the differential atmospheric refraction correction.

Finally, the program "refraction_corrector_plus" must be used to correct the effect of the differential atmospheric refraction in each one of the data cubes that will be combined at the end of the process. The data cubes generated by this program have shorter spatial dimensions than the initial ones, which is a natural consequence of the differential atmospheric refraction correction process. As the entire procedure of the differential atmospheric refraction correction is done by applying spatial shifts in each one of the images of the data cubes, at the end of the process, some borders of the images must be cut off. The program "refraction_corrector_plus", in each execution, can apply the differential atmospheric refraction correction in many data cubes, however, if the user's purpose is to combine these cubes after the correction, then, obviously, the coordinates X_{ref} and Y_{ref} (which correspond to the final coordinates of the centroid of the same structure used for determining the third degree equations (1) and (2)) must be the same to all data cubes. The spectral range of the data cubes generated by the program is $N_{min} \rightarrow N_{max}$ (where N_{min} and N_{max} are, respectively, the lowest and the highest spectral pixels considered for the atmospheric refraction correction) and, for this reason, its relation between spectral pixel and wavelength is different of the initial ones. If, originally, that relation was given by

$$\lambda = \lambda_{initial} + (n-1) \cdot \Delta_{lambda} \qquad (3)$$

where n = spectral pixel
Δ_{lambda} = wavelength interval between two consecutive spectral pixels
$\lambda_{initial}$ = wavelength corresponding to spectral pixel 1

then, after the differential atmospheric refraction correction, it becomes

$$\lambda = \lambda_{N\min} + (n-1)\cdot\Delta_{lambda} \qquad (4)$$

where λ_{Nmin} = wavelength corresponding to the spectral pixel N_{min}

IMPORTANT NOTE: It's highly recommended that the values of X_{ref} and Y_{ref} don't be taken inside the intervals of values covered by the third degree equations (1) and (2). In other words, the value assumed for X_{ref} must be either lower than the lowest of the values of X_{cent} (equation (1)) in the considered interval for λ or higher than the highest of the values of X_{cent} in the same wavelength interval. An analogous relation must be assumed for Y_{ref} and Y_{cent} (equation (2)). If this care is not taken, then discontinuities can appear in the final spectra of the data cubes generated by the program "refraction_corrector_plus".

1b) Theoretical method

In preparation.

2) Combining of data cubes

After the differential atmospheric refraction correction, the obtained data cubes can be combined in order to achieve the highest possible signal to noise. This process can be done using the task "scombine" of the software IRAF or using the program "cube_combine_plus", written in IDL language. This combination can be done in three different ways, according to the preference of the user: mean, sum or median.

3) Richardson-Lucy deconvolution

The process of Richardson-Lucy deconvolution consists in a correction of the "seeing" caused by the atmosphere, which is present in the images of the data cube resulting from the procedure previously described. This deconvolution can be performed using a PSF that varies with the wavelength, i.e., a PSF as a function of the spectral pixel of the data cube. This variable PSF can be obtained with three different methods, which are better described below. Although the user is encouraged to use one of these three methods, there is a possibility of making a Richardson-Lucy deconvolution with a single PSF along the data cube. This can be done using a gaussian synthetic PSF or a real PSF. Both procedures can be applied using the program "lucy_deconvolution_plus".

3a) Method 1 for determining the PSF for the Richardson-Lucy deconvolution

This method consists in adopting a pre-defined function for the PSF (that varies with the wavelength) of the data cube's images. This function was obtained using IFU data of an observation of the star Cal 83 and is given by

$$FWHM(\lambda) = \frac{(FWHM)_{ref} \cdot \lambda^{-0.484}}{(\lambda_{ref})^{-0.484}} \tag{5}$$

where λ = wavelength
λ_{ref} = reference wavelength
$(FWHM)_{ref}$ = FWHM at the reference wavelength

From the value of the FWHM in each wavelength, a synthetic gaussian PSF is constructed and the deconvolution can be done. In order to use this method, first of all, it's necessary to choose a reference wavelength and determine the FWHM of a punctual structure at this wavelength. After that, using the program "lucy_deconvolution_plus" the method can be applied in the entire data cube. It is worth to mention that the user has to decide what is the most appropriate number of iterations to be used. Six iterations have given good results.

3b) Method 2 for determining the PSF for the Richardson-Lucy deconvolution

In this procedure, instead of adopting a pre-defined function to describe the variation of the PSF of the images with the wavelength, the PSF at each wavelength is determined from a weighted mean between the values of the PSFs at the edges of the spectral range covered by the data cube. This weighted mean is given by

$$FWHM(\lambda) = (FWHM)_A \cdot \left(1 - \frac{(\lambda - \lambda_A)}{(\lambda_V - \lambda_A)}\right) + (FWHM)_V \cdot \left(\frac{\lambda - \lambda_A}{\lambda_V - \lambda_A}\right) \tag{6}$$

where $(FWHM)_A$ = FWHM at the blue edge of the spectral range of the data cube
$(FWHM)_V$ = FWHM at the red edge of the spectral range of the data cube
λ = wavelength
λ_A = wavelength corresponding to the blue edge of the spectral range of the data cube
λ_V = wavelength corresponding to the red edge of the spectral range of the data cube

From the value of the weighted mean for each wavelength, a synthetic gaussian PSF is created and the deconvolution can be done. To use this method, first of all, it's necessary to determine the FWHM of a punctual structure in an image at the blue and at the red edge of the spectral range covered by the data cube. This can be done, for example, making the images corresponding to the spectral edges of the data cube using the program "make_images_plus" (which makes a sum of the images in a spectral interval of the data cube in order to obtain an image with a better photon statistic) and measuring the FWHM of a punctual structure in these images with the task "imexamine" of the software IRAF, following the below steps:

- open the image with SaoImage
- apply the task "imexamine" in IRAF terminal
- position the cursor on the center of the structure whose FWHM must be measured
- type "r" to obtain as estimation of the FWHM

At this point, it's important to mention that, although the images representing the blue and red edges of the spectral range covered by the data cube must be made by summing the original images of the data cube in certain wavelength intervals (in order to obtain a better photon statistic), the representative wavelengths attributed to each of these two images must not be the mean wavelengths of the spectral intervals used to create them, otherwise the equation (6) can't be applied. Instead of that, one must attribute to each of these two images the wavelength corresponding to the edge of the spectrum, i.e., it must be attributed to the images representing the blue and red edges of the spectrum, respectively, the lowest and highest wavelength of the spectral range covered by the data cube.

Once this procedure has been done, this method for the Richardson-Lucy deconvolution can be applied to the entire data cube using the program "lucy_deconvolution_plus". Again, it must be mentioned that the user has to decide what is the most appropriate number of iterations to be used. Six iterations have given good results.

3c) Method 3 for determining the PSF for the Richardson-Lucy deconvolution

In this method, a weighted mean, analogous to that given by equation (6), is used to obtain the PSF in each wavelength of the data cube, however, in this case, this mean is done using real PSF images and not the values of the FWHM at the edges of the spectral range covered by the data cube. So, the weighted mean used in this method is given by

$$PSF(\lambda) = \left(PSF\right)_A \cdot \left(1 - \frac{\left(\lambda - \lambda_A\right)}{\left(\lambda_V - \lambda_A\right)}\right) + \left(PSF\right)_V \cdot \left(\frac{\lambda - \lambda_A}{\lambda_V - \lambda_A}\right) \tag{7}$$

where $(PSF)_B$ = image representing the PSF at the blue edge of the spectral range covered by the data cube

$(PSF)_R$ = image representing the PSF at the red edge of the spectral range covered by the data cube

With the image PSF(λ) obtained from the weighted mean of the images $(PSF)_B$ and $(PSF)_R$ for each wavelength, it's possible to apply the deconvolution in the entire data cube. To use this method, first of all, it's necessary to make the images $(PSF)_B$ and $(PSF)_R$ using, for example, the program "make_images_plus". These images must contain, only, one punctual object at the center (they can be made, for example, from a data cube of an observation of a standard star). Again, it's important to mention that, although the images $(PSF)_B$ and $(PSF)_R$ must be made by summing the original images of the data cube in certain wavelength intervals (in order to obtain a better photon statistic), the representative wavelengths attributed to each one of these two images must not be the mean wavelengths of the spectral intervals used to create them, otherwise the equation (7) can't be applied. Instead of that, one must attribute to each of these two images the wavelength corresponding to the edge of the spectrum, i.e., it must be attributed to the images $(PSF)_B$ and $(PSF)_R$, respectively, the lowest and highest wavelength of the spectral range covered by the data cube.

Once this procedure has been done, this method for the Richardson-Lucy deconvolution can be applied to the entire data cube using the program "lucy_deconvolution_plus". Again, it must be mentioned that the user has to decide what is the most appropriate number of iterations to be used. Six iterations have given good results.

An important fact to be mentioned is that, in many situations, the process of the Richardson-Lucy deconvolution (executed with any of the three mentioned methods) can result in border effects, i.e., the deconvolution dos not give good results at the spatial region of the border of the data cube's images, originating distortions, structures similar to bright "traces", etc. This problem can be avoided by enlarging the spatial dimensions of the data cube before the application of the deconvolution. The program "lucy_deconvolution_plus" offers this option to the user. Once the deconvolution is done, the spatial dimensions are reduced again to its original values by the program. It's worth to mention, however, that the enlargement of the spatial dimensions of the data cube doesn't need to be executed every time the deconvolution is applied. Only in some cases (in which it's possible to note the appearing of border effects) this process is necessary.

4) Applying the PCA Tomography

After all the procedure described above is applied to the data cube, it's possible to use the PCA Tomography method to extract information from it. It's important to mention that none of the procedures described until now is strictly necessary for the PCA Tomography be applied, however, they are very important to make the obtained result in this step as precise and detailed as possible, therefore, the user is encouraged to only apply the PCA Tomography after executing the previous procedures.

The complete process of the PCA Tomography consists in the following steps: mounting the data table, executing the PCA and constructing the tomograms and eigenspectra. Besides, the user will also be able to reconstruct the data cube from the obtained results, eliminating, if that is the case, eigenvectors that represent only noise an do not contribute with a significant fraction of the variance. It's also possible to reconstruct the data cube combining the eigenvectors in different ways in order to emphasize a certain characteristic or object. All the procedure of the PCA Tomography (except for the reconstruction of the data cube) can be applied using the program "pca_tomography_plus", whose execution can take several minutes (or even some hours), depending on the dimensions of the data cube or the power of the used computer. The user, however, can also opt to apply the steps of mounting the data table, executing the PCA and constructing the tomograms and eigenspectra separately, as explained below.

4a) Mounting the data table

This step consists in transforming the data cube into a numerical table. The table is constructed in such a way that each row corresponds to a spatial pixel and each column corresponds to a spectral pixel. The procedure of mounting this data table can be applied using the program "make_table_plus" (normally the file containing the table is in txt format), and the user can choose for mounting a table of just part of the spectral range or part of the field of view of the initial data cube.

IMPORTANT OBSERVATION: If the execution of the PCA is done using the software Matlab, then the user must be warned that, perhaps, it will not be possible to apply the process in data cubes with much more than 2000 spectral pixels. If that is the case, then the user can choose between applying the PCA in just a spectral part of the data cube or executing the re-binning process, in which the total number of spectral pixels is reduced. This last procedure must be executed with the program "make_binning_plus" (which reduces the spectral dimension of the data cube in such a way that, at the end, the number of spectral pixels is equal to the initial number divided by a constant value) before the construction of the data table (described in the previous section). If the user is performing the entire PCA Tomography with the program "pca_tomography_plus", then, the option of re-binning will be presented by the program before the beginning of the procedure.

4b) Executing the PCA

Once the previous step is done, the Principal Component Analysis (PCA) must be applied in the constructed table. This procedure can be done, in principle, using any software, but, at the end of the process, three tables must be obtained. The first of them shall contain the weight of initial variables on each one of the obtained eigenvectors, the second must have the projection of the initial data on each eigenvector and the third must have the eigenvalues corresponding to each one of the eigenvectors.

The procedure corresponding to the execution of the PCA can be done, for example, using the software Matlab, following the below steps:

- initiate the software Matlab
- go to *file* and then select the option *import data*
- select the name of the file containing the table originated from the data cube (as described in the previous section) and press OK

- in the followings menus, if everything is OK, keep selecting the option *next* and, at the end, select the option *finish*
- in the proper place, type the following command:

[PC,SCORE,latent] = princomp(name of the data table)

this command will execute the PCA on the data table and generate a matrix (PC) containing the weight of each variable on each one of the obtained eigenvectors, a matrix (SCORE) containing the projection of the data on each eigenvector and a matrix (latent) containing the eigenvalues corresponding to each one of the eigenvectors
- type the commands:

save *table1.dat* PC –ascii

save *table2.dat* SCORE –ascii

save *table3.dat* latent –ascii

where table1.dat, table2.dat and table3.dat correspond, respectively, to the names of the tables containing the data of the matrices PC, SCORE and latent. These tables can be named freely, according to the preferences of the user.

One relevant point to be mentioned is that, before the execution of the PCA, the mean calculated for each one of the initial variables must be subtracted from the data. This process is done automatically by Matlab, however, users that intend to execute the PCA using other software must be warned that this step is very important and must be always applied.

As mentioned before, the process of the PCA can be done, in principle, using any software, however, the tables containing the results must be generated in the following way: the generated eigenvectors must correspond to the columns of the table PC and the projections of the data on each eigenvector must correspond to each one of the columns of the table SCORE.

If the user is executing all the procedure of the PCA Tomography with the program “pca_tomography_plus”, then this entire step (including the subtraction of the mean values of the variables) will be applied automatically.

4c) Constructing the eigenspectra and tomograms

After the execution of the PCA, the eigenspectra and tomograms corresponding to each one of the eigenvectors must be constructed. This can be done using the program “make_eigen_spectra_and_tomograms_plus”. The eigenvalues corresponding to each one of the eigenspectra and tomograms can be found in the table constructed from the matrix latent, as explained in the previous section. Many times, however, it’s preferable to express the eigenvalues as the percentage of the variance explained by each eigenvector. In order to obtain that, the user must sum all eigenvalues, divide each one of them by this sum and multiply each result by 100.

If the user is executing all the procedure of the PCA Tomography with the program “pca_tomography_plus”, then the step of constructing the eigenspectra and tomograms will be done automatically.

4d) Reconstructing the data cube

Using the tables obtained from the matrices PC and SCORE, it’s possible to reconstruct the initial data cube. This can be done using the program “reconstruction_and_feature_enhancement_plus”. This program reconstructs the initial data cube, summing the initial mean of the data (if the user chooses to do that) that had been subtracted before the execution of the PCA, as described in section 4b. The program “reconstruction_and_feature_enhancement_plus” also allows the user to reconstruct the data cube eliminating the eigenvectors that do not contribute with an appreciable fraction of the variance and represent only noise. Mathematically, summing the initial mean of the data is only correct if the reconstruction is done using all the obtained eigenvectors, however, if the eigenvectors eliminated during the reconstruction represent only noise, then, the user can still make this sum, without making a great mistake.

4e) Feature suppression and enhancement

It’s also possible to reconstruct the data cube making a specific combination of eigenvectors, in order to emphasize or suppress some characteristic or structure in the data cube. This process, called “feature suppression and enhancement” can be done using the program “reconstruction_and_feature_enhancement_plus” in two different ways, called methods 1 and 2. In method 1 the user performs the reconstruction attributing the weights (called feature factors) 1, 0 or -1 to each one of the eigenvectors. In this process, what is being done is a sum of the intensities of the eigenvectors. However, there is a second way of combining eigenvectors, not summing its intensities, but summing its intensities normalized by the respective eigenvalues. The so called method 2 consists in this second strategy. In this case, the user not only multiplies the eigenvectors by its respective feature factors, but also multiplies the corresponding tomograms by the respective N_k factors, which are given by:

$$N_k = \frac{1}{(\Lambda_k)^{\frac{1}{2}} \cdot (n-1)^{\frac{1}{2}}} \tag{8}$$

where Λ_k = eigenvalue of the eigenvector
n = number of spatial pixels of the data cube

This normalization factor N_k is defined in such a way that the quadratic sum of the values of the spatial pixels of each tomogram that has been multiplied by it is equal to 1.