How to use the PynPoint module *HesseMatrix*

Gabriele Cugno

June 29, 2017

1 What is needed

Prior to the use of the *HesseMatrix* module, the data have to be reduced and prepared. This includes: dark subtraction, bad pixel cleaning, cut (the images must have the desired size), alignment and centering, background subtraction, stacking. Furthermore, the 'NEW_PARA' keyword must be inserted in the header. This consists in an array of de-rotation angles, used to align all the images to the same direction at the end of the PSF subtraction. The datacube has to be imported in the workspace directory through the *ReadFitsCubesDirectory* module. The *HesseMatrix* module will have access to the data only through this database. The *HesseMatrix* module should be used only in case a point source companion candidate is present in the images.

A single master frame of the stellar PSF must be prepared before running the module. This will be rescaled and introduced in the images as fake signals. It has to be a .fits file.

2 Parameters

working_place_in: (String) It points to the working directory, where the database can be found and where the module should work.
input_place_in: (String) It points to the input directory of the pipeline. This parameter is not important for the results, but is needed to define pipelines during the analysis.
output_place_in: (String) It points to the output directory of the pipeline. This parameter is needed to define pipelines during the analysis. Furthermore, the outputs of the module will be saved into the *output_place_in* directory.
psf_file: (String) It points to the image will be used for the creation of fake planets. The image can be passed as the path to a single .fits file (datacubes are not supported). The image is assumed to be centered on the central star, and can be cut to a different pixel size using the parameter *cutting_psf*.
image_in_tag: (String) Tag of the data, through which the module can access to the images in the database.
name_in: (String) It represents the name of the module. Default is "HesseMatrix".
inner_mask: (Float) Size of the inner mask to be placed on top of the central star during the PSF subtraction. It must be given as percentage of the whole image. Default is 0.1.
pc_number: (Integer) Amount of principal components of the stellar PSF one wants to subtract. Default is 20.
rough_pos: (Array or list) Rough position of the target in (integer) pixels, as a list or an array with shape: [x,y]. The origin (0, 0) is in the lower-left corner. Default is (0, 0).
rough_mag: (Float) Rough magnitude contrast of the target. This value will be used as a starting point for the photometry calculation.Default is 8.0.
subpix: (Integer) Subpixel precision with which the best position found during the astrometry step is rounded (e.g.: if the best position is [49.39,51.778] and the subpixel precision is 4, then the precision is $1/4 = 0.25$, so the best position is rounded to [49.5,51.75]). The same precision is then used to insert the fake negative planets during the photometry step. It is recommended to use a subpix value equal or greater than 2, to allow more precise results. Default is 2.
ROI_size: (Integer) Size of the side of the Range of Interest in pixels, whose curvature will be investigated. The ROI is a square with at its center the rough position given before. Default is 10.
cutting_psf: (Integer) The unsaturated image of the central star (psf_file) will be cut to an image of

size cutting_psf×cutting_psf pixels. Default is 30.
tolerance: (Float) When minimizing the curvature of the image within the ROI, the algorithm is considered 'converged' if the change in each variable between two consecutive iterations is less than the given tolerance. The optimization returns those values for contrast and position. Default is 0.1.
conv: (Float) Before the surface of the image is interpolated to calculate the hessian matrix, it is convolved with a gaussian filter to reduce the small pixel to pixel variations which should not be taken into account in order to probe the curvature of the ROI. The *conv* variable represent the standard deviation (FWHM/2) in pixels which describes the gaussian kernel for the convolution step. Default is 1.
num_pos: (Integer) Within the ROI, the curvature at $num\_pos \times num\_pos$ equally spaced positions is sampled. Default is 100.

3 Description

The HesseMatrix algorithm calculates the sum of the absolute values of the determinants of the hessian matrices within the signal region. The hessian matrix represents the second derivative of a more-dimensional function, and its determinant is a measure for the curvature of the surface described by the function. Considering the sum of the determinants in different positions allows to sample the curvature in an extended region, called Region Of Interest (ROI), of variable size and to compare the surfaces of different post-processed images. The routine then uses the Nelder Mead algorithm to minimize this value, inserting fake negative planets with the best combination of contrast and position, to make the signal disappear and to make the surface as flat as possible. Since the interpolation tries to reproduce also the small deviations from pixel to pixel, before the curvature analysis the images are convolved with a gaussian kernel with the desired FWHM, in order to flatten their surface and to have comparable results.

4 Output

The module prints out on the screen the final astrometric and photometric results. It saves them as an ASCII table in a .txt file in the output directory of the pipeline, if requested. The module also saves the .fits file of the image when the planet is best subtracted. Finally, it also creates a figure with the original ROI after the final signal subtraction and the one after the convolution, to check if this step had the desired effect.