Massimo Instructions

I managed to run Kathy’s code;

The key classes need to be fixed a bit for some update, e.g. [np.int](http://np.int/)(…) is no more in numpy so one has to use int(…);

I have also fixed the path using some generic definition that should allow you to run the basic

test without changing them.

Here is the guide to reproduce what I have.

1. copy all the stuff from Kathy in a fresh root directory, mine is named /Kate\_Oram\_Massimo
2. I added a sandbox directory under the root /Kate\_Oram\_Massimo, but at the moment is empty. It may be used

to save the results, but at the moment nothing is saved at the end of the execution.

1. Copy/overwrite in the root directory the attached files:
   1. Hadamard\_class\_v3.py
   2. data\_sim\_class.py

and under the /examples directory copy

* 1. photon\_flux\_variability\_long\_MR.py

1. The key file is this last one.

Starting at line 114 there is the definition of the parameters.  The original code of Kathy does

something rather basic: defines a small square flat area and runs on it the HTS machine with various levels of noise and different matrices and orders; there is a giant loop that  produces

tables that show how the SNR changes, basically what she used to create the plots for her thesis.

However, we want to run it on a science data cube. I have therefore commented the part that

creates the uniform square and substituted with a single line that creates the astronomical datacube using

data\_sim\_class (lines 136:141 -> line 145).

I have eliminated the loop over the various noise levels (line 63 -> 64) and changed the return instruction (line 110->111)

to get only the reconstructed datacube and not the noise analysis.

Line 155-163 display one slice/wavelength of the data cube, both original and the after HTS reconstruction.

Note that the reconstructed has noise added, but looks good.

Line 163 is a sys.exit(1) command that ends the execution. You may want to put a breakpoint before it to inspect the data cube and change e.g. the wavelength displayed.

You should be able to run on the other astronomical images (see line 145), the type of matrix (line 132) and the order (line 127).

The key part of the code is in the Hadamard\_class\_V3.py, e.g. lines 335, 340 and 376 for the S matrix. The calculations are done with simple matrix products. That is quite clever, and I wonder if Kathy came up with that or if there is an underlying code she has adapted written by a mathematician.

Unless I forgot something, this should be all what is needed to run the machine. At the moment it assumes square arrays, and I don’t see why one cannot have rectangular areas, but the hack may be not trivial since we are dealing with 4 dimentions i.e. x, y, lambda and order, that get convolved and stacked, so let’s leave it as an exercise for the future.