C++ Xtensor Instructions:

First we need to install Xtensor, and Xsimd

* **xtensor** is a library for C++ that allows us to manipulate arrays and matrices much like numpy does, in fact it is based off of numpy so is you know numpy xtensor will be easy to use
* **xsimd** is basically a wrapper for xtensor that makes reduction calcualtions faster (Such as sum)

**xtensor:**

* conda install -c conda-forge xtensor

**xsimd:**

* conda install -c conda-forge xsimd

To make use of the xtensor library we have to include the appropriate header files for whatever functions you are using. Check out the xtensor documents for more information. To do so we need to Include the path to the xtensor, xtl (a sub-package of xtensor), and xsimd header files when we compile the code. To actually enable xsimd to automatically “speed up” the code (read more on that here) we have to include the lines:

* -mavx2
* -ffast-math
* -DXTENSOR\_USE\_XSIMD -03

to the compiler too. To get the path to the folders head over to your pkgs folder in anaconda3 and search for “xtensor” and “xtl” and “xsimd” and add the paths to your compile command. For example my path to the three are:

* /home/loicbeus/anaconda3/pkgs/xtensor-0.21.10-h0efe328\_0/
* /home/loicbeus/anaconda3/pkgs/xtl-0.6.21-h0efe328\_1/include
* /home/loicbeus/anaconda3/pkgs/xsimd-7.4.9-h0efe328\_1/include/

Compiling the C++ files in the terminal:

* g++ -mavx2 -ffast-math -DXTENSOR\_USE\_XSIMD -O3 -I /home/loicbeus/anaconda3/pkgs/xtensor-0.21.10-h0efe328\_0/include -I /home/loicbeus/anaconda3/pkgs/xtl-0.6.21-h0efe328\_1/include -I /home/loicbeus/anaconda3/pkgs/xsimd-7.4.9-h0efe328\_1/include/ PSF\_prep.cpp -o PSF\_prep
* g++ -mavx2 -ffast-math -DXTENSOR\_USE\_XSIMD -O3 -I /home/loicbeus/anaconda3/pkgs/xtensor-0.21.10-h0efe328\_0/include -I /home/loicbeus/anaconda3/pkgs/xtl-0.6.21-h0efe328\_1/include -I /home/loicbeus/anaconda3/pkgs/xsimd-7.4.9-h0efe328\_1/include/ single\_fitting.cpp -o single\_fitting
* g++ -mavx2 -ffast-math -DXTENSOR\_USE\_XSIMD -O3 -I /home/loicbeus/anaconda3/pkgs/xtensor-0.21.10-h0efe328\_0/include -I /home/loicbeus/anaconda3/pkgs/xtl-0.6.21-h0efe328\_1/include -I /home/loicbeus/anaconda3/pkgs/xsimd-7.4.9-h0efe328\_1/include/ binary\_fitting.cpp -o binary\_fitting
* g++ -mavx2 -ffast-math -DXTENSOR\_USE\_XSIMD -O3 -fopenmp -I /home/loicbeus/anaconda3/pkgs/xtensor-0.21.10-h0efe328\_0/include -I /home/loicbeus/anaconda3/pkgs/xtl-0.6.21-h0efe328\_1/include -I /home/loicbeus/anaconda3/pkgs/xsimd-7.4.9-h0efe328\_1/include/ rn\_binary.cpp -o rn\_binary
* g++ -mavx2 -ffast-math -DXTENSOR\_USE\_XSIMD -O3 -fopenmp -I /home/loicbeus/anaconda3/pkgs/xtensor-0.21.10-h0efe328\_0/include -I /home/loicbeus/anaconda3/pkgs/xtl-0.6.21-h0efe328\_1/include -I /home/loicbeus/anaconda3/pkgs/xsimd-7.4.9-h0efe328\_1/include/ rn\_single.cpp -o rn\_single

Note that the last two have -fopenmp in the compile command as these two files make use of parallel processing and that command is needed for it to work. It will run without it but it will be a lot slower.

Before running the code make sure the image, corresponding model, “cameras.json”, and each of the code files are in the same directory.

Run the main python script like so (Make sure python3 is installed, some systems if they haven’t been updated are still on python 2.7)

python3 -W”ignore” PSFfitting.py tinytim\_filename rnl\_filename

This is to run the code with the monte carlo simulation. This will take considerably longer than running the normal code. It is currently set to create 100 extra images with random noise added. If desired this can be changed by going into the python file, rn\_binary, and rn\_single files and changing the variable “size” in all of them to whatever number you want them to.

The -W”ignore” will make sure the program ignores the warnings (astropy gives a warning on almost every fit that “the fit may not be successful” and since the monte carlo simulation does like 900 fits you get 900 warnings sometimes that drowns out your terminal).

I would recommend running each of the HST images in its own folder as the plots generated in the monte carlo simulation are not named in a way that would tie them to any particular image.