# Software instructions for the lectures of the "Sterile neutrinos in physics, astrophysics, cosmology" course

S. Gariazzo, C. Giunti

March 14, 2021

We list here the software requisites for performing the exercises that will be proposed during the tutorials associated with the "Sterile neutrinos in physics, astrophysics, cosmology" course at the 2021 edition of the "Theoretical Aspects of Astroparticle Physics, Cosmology and Gravitation" school at GGI.

In case of problems following these instructions, please contact S. Gariazzo (gariazzo@to.infn.it), or ask us during the Office Hours and Tutorials.

## 1 First tutorial – sterile neutrinos at reactor experiments

For the first set of exercises we will use the `GLoBESfit`[1] software, which was developed on top of the `GLoBES`[2] software. For more information on these codes, see `https://www.mpi-hd.mpg.de/personalhomes/globes/documentation.html` and `https://arxiv.org/abs/2005.01756`.

### 1.1 GLoBES

The only dependency of `GLoBESfit` is `GLoBES`. If you already have it in your system, either you already know how to install and use it, or someone prepared it for you. If you have an old version, you should upgrade to the most recent one (`GLoBES` 3.2.18) for `GLoBESfit` to work, otherwise you can just skip to section 1.2.

Before installing the main code, one has to prepare the mandatory dependences of `GLoBES`, see `https://www.mpi-hd.mpg.de/personalhomes/globes/download.html`. In linux systems, these dependencies are easily installed using the software manager. In other systems, you may need to install them manually.

Once they have been installed, in linux a typical system-wide `GLoBES` install is performed using the following commands:

```
wget https://www.mpi-hd.mpg.de/personalhomes/globes/download/globes-3.2.18.tar.gz
tar xzvf globes-3.2.18.tar.gz
cd globes-3.2.18/
./configure
make
sudo make install
sudo ldconfig
```

If you do not have administrator privileges, you should install `GLoBES` locally: you will find some useful instruction in the `INSTALL` file included in the `globes-3.2.18` folder. After this, one can optionally remove the downloaded file and the extracted folder and verify that the software has been properly installed (`globes-config --version`).

### 1.2 GLoBESfit

The second step is to install `GLoBESfit`. To obtain the code, you can clone the git repository from GitHub (`git clone https://github.com/JMBerryman/GLoBESfit.git`), so that it will be easier to update the code when new versions will be released. After cloning, you just need to open the newly created folder (`GLoBESfit/` within the current working directory). Another possibility is to download the code from the web page (`https://www.globesfit.org/globesfit-1.0.3.tar.gz`), extract the content of the .tar.gz

[1] `https://www.globesfit.org/`

[2] General Long Baseline Experiment Simulator, `https://www.mpi-hd.mpg.de/personalhomes/globes/index.html`.

file and go in the new folder. Once inside `GLoBESfit/`, go to `source/` and compile with `make`. If the `globes-config` executable is not present in the current `$PATH`, you need to go to `source/` and edit the `Makefile` first, to match the actual position of `globes-config`.

# 2 Second tutorial – sterile neutrinos in cosmology

To perform cosmological calculations, several tools are available. In the tutorial, we will consider `CLASS`[3], a code written in C by prof. J. Lesgourgues and collaborators, and the associated Markov Chain Monte Carlo (MCMC) code `MontePython` (`https://github.com/brinckmann/montepython_public`).

## 2.1 CLASS

As for `GLoBESfit`, there are two main ways to download `CLASS`: download the compressed file (see `https://github.com/lesgourg/class_public/releases`) or clone the repository (recommended for easier updates or development, use `git clone https://github.com/lesgourg/class_public.git`). Once you have the code, enter the class directory. Installing the `CLASS` executable in C should be even easier than installing `GLoBES`: you should have it ready after `make clean; make -j`. Check the compilation worked with `./class explanatory.ini`.

It is recommended, however, to install also the python wrapper, `classy`, which is automatically created by `make`. If the compilation worked, you should be able to run `python -c "from classy import Class"` without receiving error messages. If the previous command fails, you probably need to install some python packages such as `numpy` and `cython` before running `make`. You should be able to install them using the `pip` utility (for example with `pip install --user numpy cython`). If something goes wrong, check the official documentation: `https://github.com/lesgourg/class_public/wiki/Installation`.

## 2.2 MontePython

Once the `CLASS` python wrapper is installed, there is not much more to do to work with `MontePython`, unless the Planck likelihoods and the plotting features are required.

For a full list of instructions to install the code and the possible dependencies, check `https://github.com/brinckmann/montepython_public`. Here we will use only a basic version of the MCMC algorithm, with fake Planck data. In order to do that, we only need to install the main `MontePython` code. You can download it as a compressed archive from the web page `https://github.com/brinckmann/montepython_public`, or you can clone the git repository as discussed previously for the other codes (use the command `git clone https://github.com/brinckmann/montepython_public`). Once you go inside the folder, you should be able to run e.g. `python montepython/MontePython.py --help`. If it works without errors, everything should be fine and you should be able to run simple MCMC.

If you want to analyze the output chains and produce plots, `MontePython` requires two additional python libraries (`matplotlib` and `scipy`), which are very common and should be easy to install within each system. With `pip`, they are obtained using for example `pip install --user matplotlib scipy`.

[3] Cosmic Linear Anisotropy Solving System, `http://class-code.net/`.