**Preparing an ASCOT input file**

ASCOT5 input files use the hdf5 file format. I generate input files from Python scripts, e.g.

cd /project/projectdirs/m3195/ascot/ascot5/runs (or issue the ‘toruns’ alias)

python group\_go\_703.py init

where group\_go\_703.py is the Python script, and ‘init’ tells the Python script to generate an input file. Generally, I copy an existing Python script into a new filename, then make some changes to the input file, and then execute the command above to generate the input file, which in this case would be group\_go\_703.h5. So the full sequence is:

cp group\_go\_703.py group\_go\_725.py

<edit the file group\_go\_725.py>

python group\_go\_725.py init 🡪 generates group\_go\_725.h5

**IMPORTANT!** Toward the end of the .py file there is a line that defines the name of the output file, which in this case would be ‘group\_go\_703.h5’. So you **MUST** change this string to properly correspond to the new simulation number … otherwise you will overwrite the .h5 file of the parent simulation.

**RECOMMENDATION:** Before executing the new python script above to generate the new .h5 input file, I typically do a “diff” on the parent and daughter python scripts to make sure that I changed what I thought I changed, no more and no less. I do this because, more than once, I have made some error when editing the new Python script, leading to erroneous output, and it is tedious and time-consuming to determine why the output is wrong.

**IMPORTANT!**  My python codes that generate the .h5 ASCOT input file call ASCOT-provided routines that actually write the various parameters into the .h5 file. For reasons that I don’t fully understand, the ASCOT development team allows multiple sets of input data (and output data) to be contained in a single .h5 file. I don’t like this approach, so I just have a one-to-one relationship between input files and simulations. Unfortunately, the ASCOT utilities are written so that if you ask to write a set of input data to an **existing** .h5 file, they will **not** erase the existing input data, but simply create a second (or third, …) set of input data into the .h5 file.

This can obviously lead to confusion. So suppose you create an .h5 file, run it, and find that there was an error. You go back to the python script that generated the .h5 file, e.g. group\_go\_725.py and make some change. You **MUST** then delete the file group\_go\_725.h5 before re-generating it, otherwise you will get multiple sets of input data in your .h5 file.