ASCOT workflows

S. Scott

CFS

August 31, 2021

*This file: 20210831\_ascot\_workflows\_01*

**Related documents**

* computing\_kw\_losses.docx
* computing\_pnp\_10.docx
* scenario\_notes\_1\_2.docx (PRD: 16.8% losses 95% rho\_poloidal\_birth=0.7)
* ascot\_handover\_20.docx (newer versions, i.e. 21 and above, may be available)
* sample\_ascot\_input\_file.docx

The following templates implement a simplified ‘pnp’ workflow that do not use the lossmap algorithm.

Template files (python scripts that generate the ASCOT .h5 input file)

* Compute marker loss fractions to the wall: group\_go\_1283.py
* Compute spatial distribution of marker loss to the wall: group\_go\_1273.py

Will need to make adjustements:

* options that change the ensemble of markers from alphas to REs.
* fn\_bfield for ripple data
* number of markers Nmrk
* fn\_wall\_3d for 3D wall shape (or see other approaches as described below)

Bread crumbs:

* 1283 🡨 1253 🡨 1249 🡨 template\_shape\_2e.py.
* 1273🡨 1270 🡨 1269 🡨 … 1248 🡨 template\_step\_1e.py. Note that “step\_1” template files were originally constructed to generate ASCOT input files that would generate lossmap data and so the maximum simulation time was set very short, 1.e-6, in the template\_step\_1e.file. But this was changed to 5.e-4 before 1273.

**Development of ‘template’ files for generating ASCOT .h5 input files**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | marker | Marker |  |  |  | gp |  | N | m |
| template | Nmrk | simtime | generation | set |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| step\_1a | 5,360,000 | 1.e-6 | markers\_03 | 7 | - | - | no |  |  |  |  |
| step\_1b | 5,360,000 | 1.e-6 | markers\_03 | 7 | - | - | no |  |  |  |  |
| step\_1c | 2,680,000 | 1.e-6 | markers\_03 | 7 | - | - | no |  |  |  |  |
| step\_1d | 5,360,000 | 1.e-6 | markers\_03 | 7 | - | - | yes |  |  |  |  |
| step\_1e | 2,680,000 | 1.e-6 | markers\_03 | 7 | - | - | yes | 1248 | 44489720 | 10 | 2 |
| (step\_1e) | 3,000,000 | 5.e-4 | markers\_03 | 7 |  |  | yes | 1275 | 45812933 | 25 | 97 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| step\_2a | 10,720 | 0.08 | markers\_03 | 7 | - | - | no |  |  |  |  |
| step\_2b | 21,440 | 0.05 | markers\_03 | 7 | - | - | no |  |  |  |  |
| step\_2c | 10,720 | 0.05 | markers\_03 | 7 | - | - | no |  |  |  |  |
| step\_2d | 21,440 | 0.05 | markers\_03 | 7 | - | - | yes |  |  |  |  |
| step\_2e | 10,720 | 0.05 | markers\_03 | 7 | - | - | yes | 1249 | 44493303 | 5 | 166 |
|  | 10,720 | 0.05 | markers\_03 | 7 | - | - | yes | 1253 | 44504032 | 5 | 164 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| step\_3a | - | 5.e-4 | markers\_10 | 1 | 40560057 | 40877296 | no |  |  |  |  |
| step\_3b | - | 5.e-4 | markers\_10 | 1 | 40560057 | 40877296 | no |  |  |  |  |
| step\_3c | - | 5.e-4 | markers\_10 | 1 | 40560057 | 40877296 | no |  |  |  |  |
| step\_3d | - | 5.e-4 | markers\_10 | 1 | 40560057 | 40877296 | yes |  |  |  |  |
| step\_3e | - | 5.e-4 | markers\_10 | 1 | 40560057 | 40877296 | yes |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

*Table 1. comparison of ASCOT workflow templates.*

Differences, as of 8/30/2021

1d/1e: Nmrk = 5,360,000 🡪 2,680,000

2d/2e: Nmrk = 21,440 🡪 10,720

3e/3e: Nmrk = 21,440 🡪 10,720

So ‘e’ seems to be a reduced-resolution version of ‘d’

Historicall, I think that ‘step 1’ template files were written to generate ASCOT .h5 files that would compute lossmaps, ‘step 2’ generated .h5 files to compute alpha loss fractions, and ‘step 3’ genereated .h5 files to compute spatial patterns of the alpha loss at the wall.

then the historical development was ‘a’ … first attempt at standardized template files, ‘b’ = next generation, etc.

**Wall shape**

Generally, we provide ASCOT with a numerical representation of the 3D PFC shape using a flat-ascii file generated by triangulate\_torus.py (or its successors, e.g. triangulate\_torus\_8.py as of 8/31/2021). The PFC shape is represented by an ensemble of triangles – we supply the [x,y,z] coordinates of the three vertices of each triangle, one triangle per line (i.e. 9 ascii values). Typically in group\_go\_XXXX.py the name of the file containing the PFC wall shape is named fn\_wall\_3d but this is not guaranteed.

Typically, e.g. in group\_go\_1280.py, a 3-dimensional PFC wall shape is specified by:

# ----------------------------

# - 3D wall shape -

# ----------------------------

wall\_settings={}

aa\_wall = sparc\_proc.write\_3d\_wall(fn\_hdf5, fn\_wall\_3d, wall\_settings, desc=my\_description)

where sparc\_proc points to a home-grown Pythhon script, …/mypython/sparc\_processing.py.

Further hen down further in the grou\_go\_XXXX.py file:

fn\_wall\_3d = 'shape254\_triangles.txt' # possible\_change

It has been well documented that the magnitude and spatial pattern of the surface power density of the lost alphas is affected strongly by even small (~mm-scale) details of the wall shape. For many physics studies, these details confuse the bigger picture, i.e. “where do the alphas leave the plasma?”.

There are three ways to force ASCOT to compute alpha losses to the LCFS rather than to a complicated 3D wall shape.

1. Physics options: there are several input options in the ASCOT input file that control when ASCOT should terminate its calculation of a marker’s orbit. There is some description of these options in the file: sample\_ascot\_input\_file.docx.

Basically, if you set ENDCOND\_RHOLIM=1 then the orbit simulation will be terminated if the marker’s rho\_poloidal leaves the user-defined range [ENDCOND\_RHO\_MIN, ENDCOND\_RHO\_MAX]. If you set ENDCOND\_WALLHIT=1, then a marker will be considered lost if its orbit crosses through one of the triangles in the ensemble of PFC-triangles that comprise the user-defined PFC surface.

Note that ENDCOND\_RHOLIM and ENDCOND\_WALLHIT are completely independent, and I think for example that it is not prohibited to set both ENDCOND\_RHOLIM=1 and ENDCOND\_WALLHIT=1 (in which case the marker would be considered lost if either it leaves the LCFS at rho=1 or if it strikes a PFC-triangle). But generally I think it would be appropriate to exclusively set only ENDCOND\_RHOLIM or ENDCOND\_WALLHIT to unity, so that you are clear on what is the condition that caused a marker’s orbit to be terminated.

**So to force ASCOT to regard a marker as being ‘lost’ when it crosses the LCFS, you would set:**

* **ENDCOND\_RHOLIM=1**
* **END\_RHO\_MIN=0**
* **ENDCOND\_RHO\_MAX=1**
* **ENDCOND\_WALLHIT=0.**

To force ASCOT to regard a marker as being lost when it hits a PFC wall triangle, set ENDCOND\_RHOLIM=0, ENDCOND\_WALLHIT=1 and then make a call in the group\_go\_XXXX.py file to Python method write\_3d\_wall.

1. Regard markers as being lost only when they strike a wall, and construct a 2D wall that is conformal to the LCFS. For example in group\_go\_1280.py:

wall\_settings={}

wall\_settings["edge\_thickness"] = 0.

wall\_settings["eq\_index"] = 0

aa\_wall = sparc\_proc.write\_sparc\_conformal\_wall(fn\_hdf5,fn\_geqdsk, wall\_settings,

desc=my\_description)

Note: I think (need to check this) that the ‘wall’ will be constructed OUTSIDE the actual LCFS by the user-defined distance “edge\_thickness” (I think the distance is specified in meters, but this should be checked).

1. Regard markers as being lost only when they strike a wall, and pass along a 3D wall geometry whose coordinates are specified in a user-supplied flat-ascii file. But specify file sparc\_conformal\_3d\_edge\_00.txt as the shape of the PFC surfaces (note: there are other files, e.g. sparc\_conformal\_3d\_edge\_20.txt which has the wall moved back 2 cm from the LCFS). See e.g. group\_go\_654.py for an implementation of this approach.

My recollection is that ASCOT uses slightly different logic to implement the ENDCOND\_RHOLIM and ENDCOND\_WALLHIT conditions for terminating a marker’s orbit. Furposes of implementing the parent/daughter workflow, where we follow markers until they are lost from the plasma, and then in a separate ASCOT simulation follow the orbits from the loss-point to an actual intersection with a wall triangle, at some point I decided that it was more accurate to use the ENDCOND\_WALLHIT=1 end condition (typically with a call to write\_sparc\_conformal\_wall to define a wall that is coincident with the LCFS) in the parent simulation. But this conclusion should be checked.

Also, I need to review whether I have proven that the computed loss magnitudes and spatial patterns are the same for approaches #2 and #3 above.