Generating VTP output files for 3D rendering by ParaView

ParaView is a powerful, free 3D rendering program. The ASCOT software suite includes a postprocessor that exports the surface power density computed on the defined 3D PFC surface (of contiguous triangles) in the .vtp format required by ParaView. The Paraview 3D visualization of the surface heating has advantages and disadvantages over the calculations of surface power density as computed by my home-grown ASCOT postprocessor (process\_ascot.py):

* **(Big) advantage:** ParaView “knows” the full 3D shape of the PFC surface, including the “sides” of the poloidal and toroidal belt limiters. These “sides” are present due to the fact that the plasma-facing surfaces of the limiters are, by definition, proud of the wall surface by some distance, typically up to ~10 mm. So at the edges of the limiters, there are “gaps” between the wall surface and the front-facing surface of the limiters – these gaps are the “sides” of the limiters.

By design, these “sides” are recessed by some distance relative to the maximum height of the limiter surfaces and so they are effectively in the “shadow” of the limiters. So one might expect that these sides will receive little of the lost-alpha power. On the other hand, at least in my present design-family of candidate wall shapes, some of the sides are nearly perpendicular to magnetic field lines.

So we must always check the surface power density on these sides, and we must use Paraview to do that.

* **Disadvantage:** the surface power density is computed by ASCOT on a grid of triangles that is defined in the ASCOT input file. So if you want to evaluate a different PFC shape, you need to rerun ASCOT again. This takes about 2 hours of CPU time on NERSC using ~10 nodes, which is not terribly expensive but could add up if a large number of candidate wall shapes are being evaluated.
* **Disadvantage:** my Python script triangulate\_torus generates the ensemble of surface triangles that collectively represent the PFC surface; it generates a file that can be read by the ASCOT preprocessor which constructs the ASCOT input file. Triangular\_torus.py “knows” the [R,Z] geometry of individual structures such as limiters but the knowledge that say triangle #4532 resides in the upper corner of the 14th poloidal limiter is **lost** when the ensemble of triangles is created. So it is not possible, or at least not easy, to average the pattern of surface power density over the 18 poloidal limiters in order to improve statistics.

I have written a [Paraview tutorial](file:///C:\Users\sscott\Documents\ripple\ASCOT_handover\paraview_tutorial.pdf) which explains how to do simple stuff in Paraview, with an emphasis on those functions that are useful for rendering ASCOT5 output.

**Generating the .vtp file**

python $dir\_mypython/export\_vtp\_sds.py ascot\_12345678.h5 ploss\_kW

python $dir\_mypython/export\_vtp\_hits.py ascot\_12345678.h5

python $dir\_mypython/export\_vtp\_norm.py ascot\_12345678.h5 ploss\_MW

The first plots the surface power density on the defined 3D PFC surface; the second plots the number of marker ‘hits’ per triangle on the same surface, and the third plots the normalized surface power density on the same surface (need to remember whether the maximum normalization is 1.0 or 100.0)

Corresponding output files: ascot\_12345678.vtp, ascot\_12345678\_hits.vtp, and ascot\_12345678\_norm.vtp.

**Generating the vtp file: combining two runs**

When we compute the spatial distribution of the ‘prompt’ and ‘nonprompt’ alpha losses in separate ASCOT simulations, we need a utility to combine them into a single .vtp file for use by Paraview and by Matt Reinke’s software that computes the surface heating. To do this:

In the standard ‘runs’ directory (/project/projectdirs/m3195/ascot/ascot5/run):

python $dir\_mypython/export\_vtp\_combine.py

Enter name of hdf5 file with prompt losses: ( default=) ascot\_40890729.h5

Enter name of hdf5 file with nonprompt losses: ( default=) ascot\_40921464.h5

Enter total prompt power loss (kW) ( default=0.0) 476.

Enter total nonprompt power loss (W) ( default=0.0) 170.

This assumes that you already have copied the files e.g. ascot\_40890729.h5 from the output directory into the runs directory with the copy\_to\_runs 40890729 command.

Output: ascot\_40890729\_40921464.vtp

Note: the utility doesn’t do anything different when processing the ‘prompt’ versus ‘nonprompt’ data. It just combines the two simulations with the associated user-defined total alpha power losses. The prompts include ‘prompt’ and ‘nonprompt’ just to remind the user what the utility is intended to do.

**Combining prompt and non-prompt ASCOT output**

python $dir\_mypython/process\_ascot\_pnp.py

enter comma-separated list (w/o spaces) of ascot output .h5 files for PROMPT losses( default=) ascot\_41267512.h5

enter comma-separated list (w/o spaces) of ascot output .h5 files for NONPROMPT losses( default=) ascot\_41268353.h5

enter kW of lost alphas (prompt)( default=0.0) 476.

enter kW of lost alphas (non-prompt)( default=0.0) 170.

enter profiles filename( default=v1e\_profiles\_3.txt) <cr>

enter name of geq file( default=v1e.geq) <cr>

enter stub for outout filename (none for outout to screen)( default=) ascot\_pnp\_41267512\_41268353

enter filename for triangles-parameters( default=) shape81\_parameters.txt

enter True to suppress toroidal belt limiters ( default=False) True

enter maximum marker (0 for all of them) ( default=0) 40000

**Combining ASCOT output files**

The home-grown ASCOT postprocessor process\_ascot.py allows you to combine multiple ASCOT output files and thus the instructions given below are not needed (and in fact would cause a problem for process\_ascot.py).

But the export\_vtp\_sds.py works with just a single ASCOT output file, and so if you want to plot the surface power density from the combination of multiple ASCOT output files, you must first combine them into a single ASCOT output file.

ASCOT has an official utility to do this. I recommend that if you have files ascot\_1.h5, ascot\_2.h5, and ascot\_3.h5 that you want to combine, do the following:

cp ascot\_1.h5 ascot\_123\_combined.h5

a5combine add ascot\_123\_combined.h5 ascot\_2.h5 ascot\_3.h5

In response to the prompt, type ‘c’. The reason for copying ascot\_1.h5 into ascot\_123.h5 is that the a5combine utility copies the contents of the second and third files into the first. So this approach leaves the files ascot\_1.h5, ascot\_2.h5 and ascot\_3.h5 unperturbed.

a5combine seems to be a Python script that currently lives in

/global/homes/s/sscott/.local/cori/3.6-anaconda-4.4/bin/a5combine

**3D rendering .vtp files with ParaView**

Obtain the Paraview download file from paraview.org and install it on your PC

Invoke the ParaView application

Click on File (upper-left corner, as usual) 🡪 Open (or Recent Files)

Migrate to the folder that has your .vtp file and click on it.

For both better and worse, ParaView is a fully-featured application. There can be a considerable learning curve, and I am not very far up that curve.