Computing lost alpha power and surface power density with prompt/non-prompt losses

S. Scott

CFS

April 20, 2021

*This file: computing\_pnp10.docx*

**Executive summary: Temporal separability of alpha particle losses**

The spatial pattern of alphas whose orbits cross the LCFS falls into two distinct patterns:

* Prompt losses (t < 0.02 msec): alphas are lost along a broad region in the upper-outer wall.
* Nonprompt losses (0.02 < t < thermalization, which for alphas born outside rho > 0.7 is about 100 msec for the SPARC PRD).

***Significantly, the spatial pattern of the non-prompt losses changes only slightly over the entire slowing down period. We can exploit this situation to greatly (factor >50) increase the number of alpha-orbits that we follow, thereby improving the statistics of the computed surface power density of the lost-alpha population.***

1. Follow e.g. 30,000 markers for the full thermalization time to compute the magnitude of the prompt and non-prompt losses.
2. Follow e.g. 5-10 million markers for 0.5 ms to compute the spatial distribution of the losses.

This report describes a workflow for the ASCOT simulations that implements this approach. Two important points must be mentioned at the outset:

* The proposed approach is based on the observation that the spatial pattern of nonprompt lost alpha particles at SPARC’s LCFS is invariant with respect to simulation time. We have not explored whether this fortunate behavior will be realized in other scenarios, such as losses of RF tail ions in SPARC, beam ions in ITER, or even alpha particles in SPARC in the presence of sawteeth and MHD. Each case must be justified by evaluating the spatial pattern of the nonprompt losses as a function of simulation time.
* This workflow is appropriate ONLY for the calculation of the spatial distribution of the surface power density of lost alphas. It is completely useless for computing e.g. the alpha distribution function of the confined alphas.

**ASCOT simulation**

The ASCOT simulations discussed in this report are for the SPARC V1E ‘primary reference discharge’ assuming a fusion power of 100 MW. A relatively poor alignment of the TF coil set is assumed: the coils are randomly offset in the in/out direction by a random distance selected from a normal distribution having a sigma of 6 mm, with an imposed maximum of 9 mm. The average absolute value of the in/out displacements is 4.45 mm. The assumption of such poor alignment increases the ripple-induced alpha loss; as we will see below, although classical first-poloidal-orbit losses dominate the total alpha energy loss, the maximum surface power density of the lost alphas is dominated by the nonprompt losses due to greater spatial concentration of the nonprompt losses.

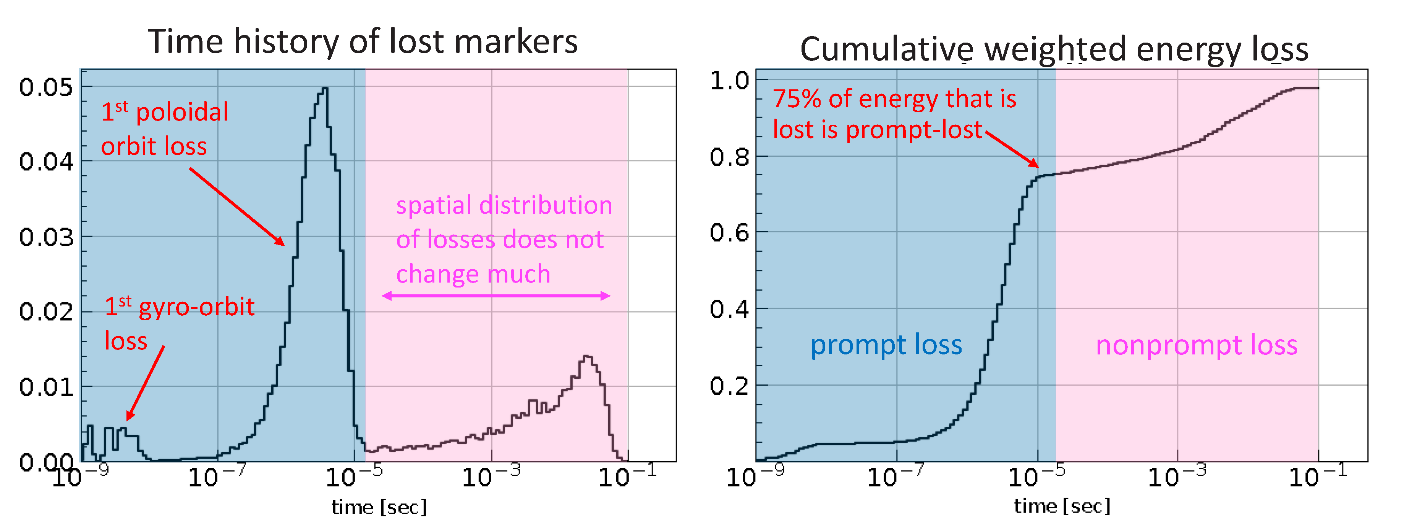
The alpha source rate in SPARC will be much higher near the plasma center than near the edge, where the density and temperature are lower. The ensemble of alphas that are numerically created prior to following their orbits with ASCOT should be representative of the local alpha source rate, and one approach to do that would be to create an alpha source population (called a ‘marker’ ensemble in ASCOT) density that is proportional to the local alpha birth rate. But the ASCOT orbit simulations show that, even with fairly large TF ripple generated by poorly aligned TF coils, no alphas that are born with a normalized minor radius (specifically, rho poloidal) less than 0.7 are lost. So creating a marker ensemble whose density is proportional to the local alpha source rate would ‘waste’ most of the computed orbits; we would end up following lots of orbits that never strike the wall. So instead we create a marker ensemble that is uniformly distributed in 3D space, but then assign each marker a ‘weight’ that is proportional to the local alpha birth rate. When computing the alpha loss fraction and surface power density at the wall, the energy of each alpha at time of loss is multiplied by this weight. In this report both the unweighted and unweighted marker-loss statistics are plotted in various graphs; although the unweighted and unweighted statistics differ quantitatively, the qualitative trends are the same for the unweighted and unweighted statistics.

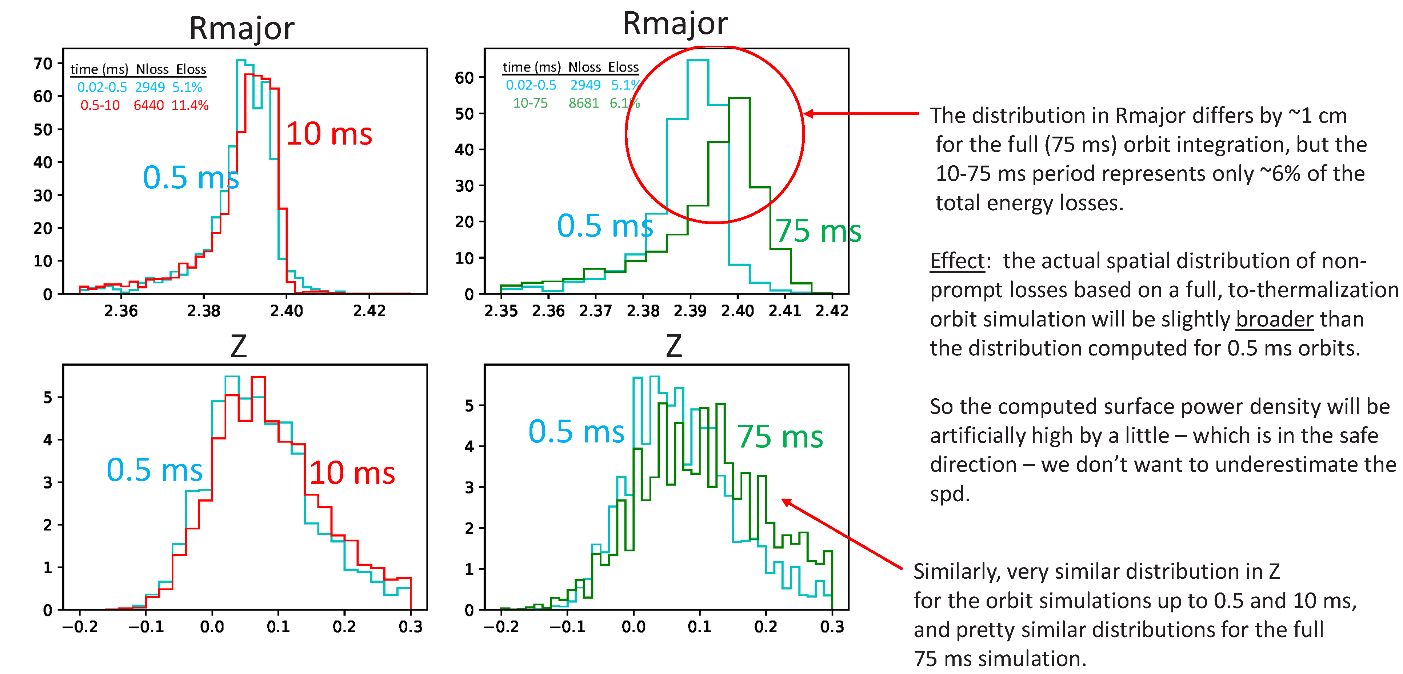
**Invariance of spatial pattern of nonprompt alpha losses**

Figure 1 (left) plots the time history of the number of lost markers, note the log scale for the x-axis. For this simulation , markers were born in the edge region, 0.7 < rho <1, because previous simulations had shown that no markers born inside rho < 0.7 were lost to the LCFS for the assumed TF configuration (i.e. misalignment). The losses clearly group into ‘prompt’ (t < 2e-5 sec) and ‘nonprompt’ components, and as will be shown below these have distinctly different spatial patterns of loss at the LCFS.

The CPU time required for a given marker is roughly proportional to the simulation time. So ‘prompt’ lost markers are very cheap computationally, as their maximum simulation time is only 2.e-5 sec. Some of the nonprompt lost markers are very expensive, i.e. those that are lost very near to their full thermalization time of 100 ms. But since the spatial pattern of the nonprompt losses is nearly invariant with simulation time, we can choose to follow the nonprompt lost markers for only a small fraction of their thermalization time.

Figure 2 illustrates the fundamental observation that forms the basis of the proposed workflow: the spatial pattern of nonprompt-lost markers whose orbits crosses the LCFS is plotted for different time-classes. The plot compares the major-radius and elevation (Z) distribution of the losses for markers that are lost within 0.5 milliseconds, 10 milliseconds, and 75 milliseconds. Note that in the upper right panel, the distribution in Rmajor differs by ~1 cm for the full (75 ms) orbit integration, but the 10-75 ms period represents only ~6% of the total energy loss. So to a good approximation, computing the spatial loss pattern based on following orbits for 0.5 msec will be very similar to that based on following the orbits to full thermalization at ~100 msec.

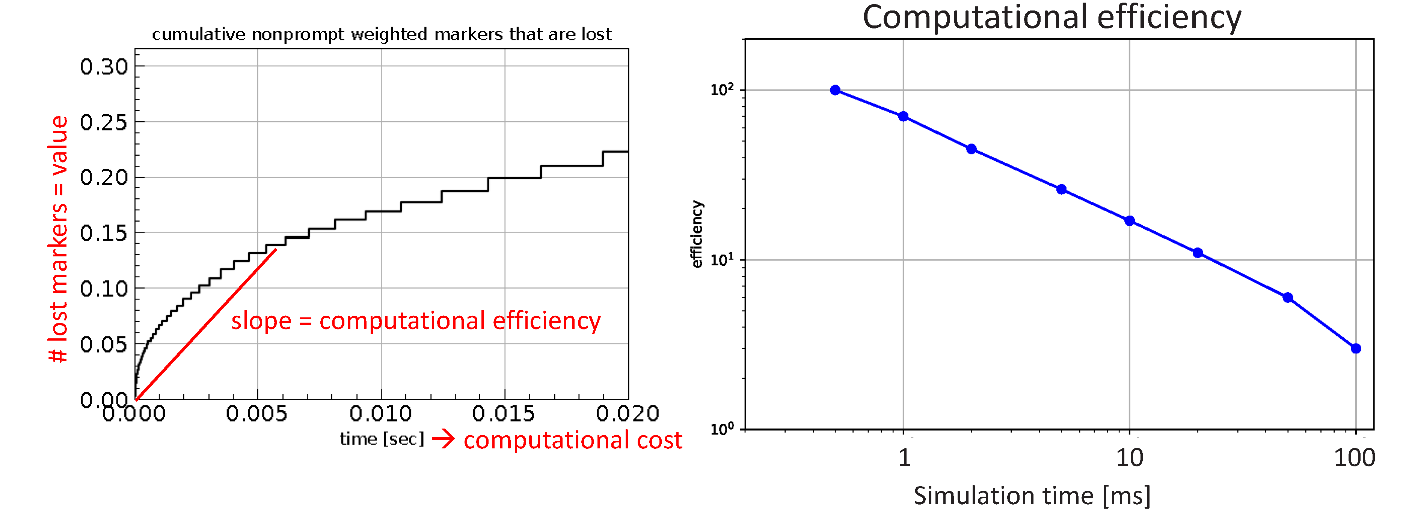
*Figure 1. (a) Time history of weighted nonprompt lost markers (note log scale on x-axis) for an ASCOT simulation of alphas in the SPARC PRD, assuming that the V1E TF coils were mis-aligned in the in/out direction randomly with offsets taken from a normal distribution with sigma = 6 mm and an imposed maximum of 9mm. (b) Cumulative weighted energy loss as a function of time, normalized to the total weighted energy loss.*



*Figure 2. Spatial pattern of nonprompt marker losses at the LCFS for long and short simulations. Top: spatial distribution as a function of major radius. Bottom: spatial distribution as a function of elevation. Left: compare simulation times of 0.5 ms and 10 ms. Right: compare simulation times of 0.5 ms and 75 ms.*

One might think that following the nonprompt-lost orbits for 0.5 msec would take ~200 times less CPU time than following the orbits for 100 msec and so we might expect to realize a factor ~200 improvement in computational efficiency. This logic is misleading for two reasons. First, not all nonprompt markers must be followed for their full thermalization time; many cross the LCFS well before thermalization. But more importantly, only a small fraction of the nonprompt lost markers will be lost within a simulation time of 0.5 msec, and it is the number of lost markers that we care about, since that determines the statistical accuracy of the computed surface power density.

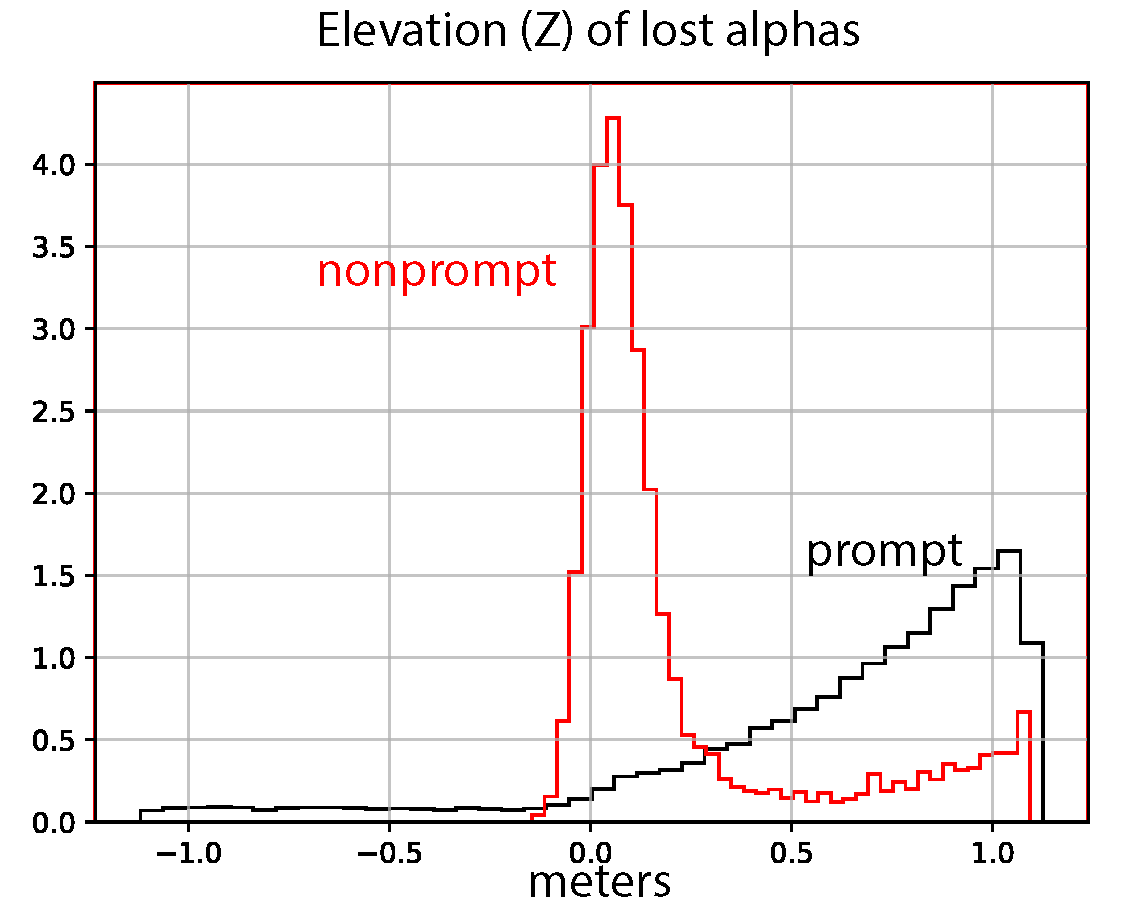
As shown in Figure 3, we can define a ‘computational efficiency’ for the simulation, being the ratio number of markers which are lost divided by the simulation time. The computational efficiency increases by about a factor of 30 as we decrease the simulation time from 100 msec to 0.5 msec. So we expect this workflow to improve the computational efficiency of computing the surface power density by about a factor of 30 for the nonprompt losses. As will be shown below, the actual improvement in computational efficiency for the nonprompt losses is about a factor of 70, and we do not at present understand why the actual efficiency improvement is larger than the expected improvement.



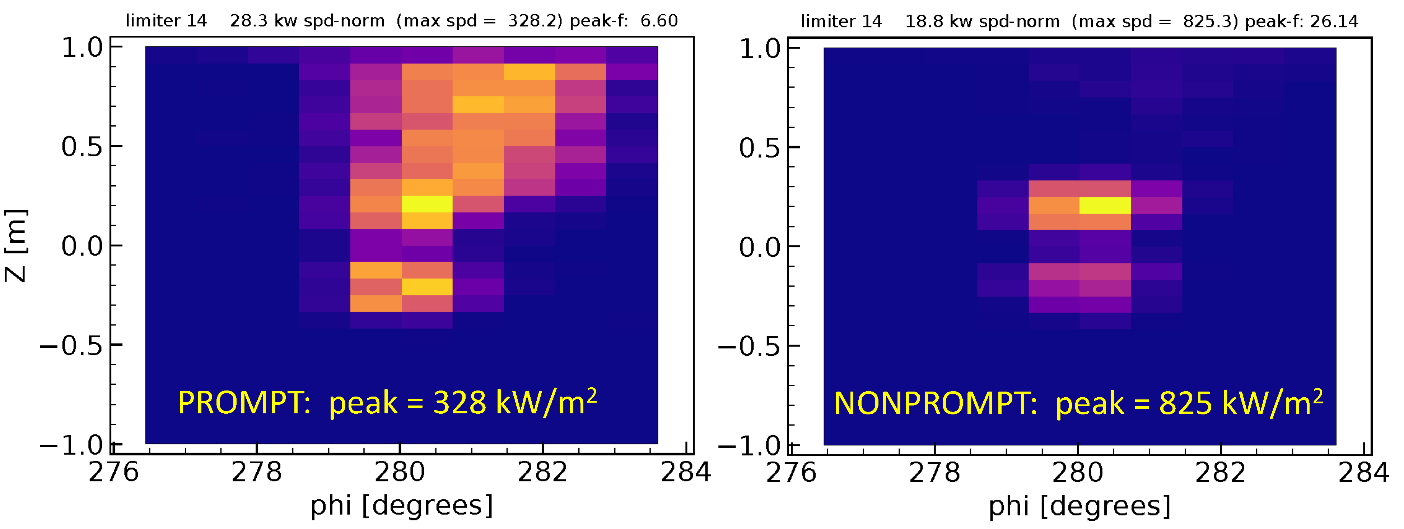
*Figure 3. (a) Cumulative number of nonprompt-lost markers as a function of simulation time. Effectively the y-axis represents ‘value’ (i.e. improved statistics for the calculation of surface power density) and the x-axis represents computational cost. The ratio of these is a measure of computational efficiency. (b) Computational efficiency as a function of maximum simulation time. The computational efficienty increases a factor of ~30 as we decrease the maximum simulation time from ~100 ms to 0.5 ms*.

Figures 4 and 5 are not strictly part of the justification of the proposed workflow, but they do highlight that the spatial pattern of the prompt and nonprompt marker losses are distinctly different at the LCFS. In particular, the nonprompt losses concentrate in a narrow band near the outer midplane whereas the prompt losses are more spread out. Although the nonprompt losses represent only ~25% of the total lost alpha energy, they dominate the maximum surface power density.

For this reason, we focus attention on the behavior of the nonprompt component of the losses. Keep in mind that if we assumed better alignment of the TF coils, the ripple-induced alpha losses (which are predominantly nonprompt) would be reduced, possibly significantly, and we might revert to a situation where the surface power density is dominated by the prompt losses.



*Figure 4. Spatial distribution of lost markers at the LCFS for the prompt and nonprompt components. Although the nonprompt losses represent a small fraction of the total lost alpha power, they are more concentrated spatially.*

**

*Figure 5. Surface power density for the poloidal limiter with the largest computed maximum surface power density (limiter 14). The losses are more highly spatially concentrated for the nonprompt losses (right) than for the prompt losses (left) and so the peak surface power density is set by the nonprompt losses. However, keep in mind that this simulation assumed fairly poor alignment of the TF coils, with an ensemble of in/out offsets (the most dangerous direction of misalignment) having a sigma of 6 mm and an imposed maximum of 9 mm. The (nonprompt) ripple-induced alpha losses would decrease significantly if better TF coil alignment were assumed.*

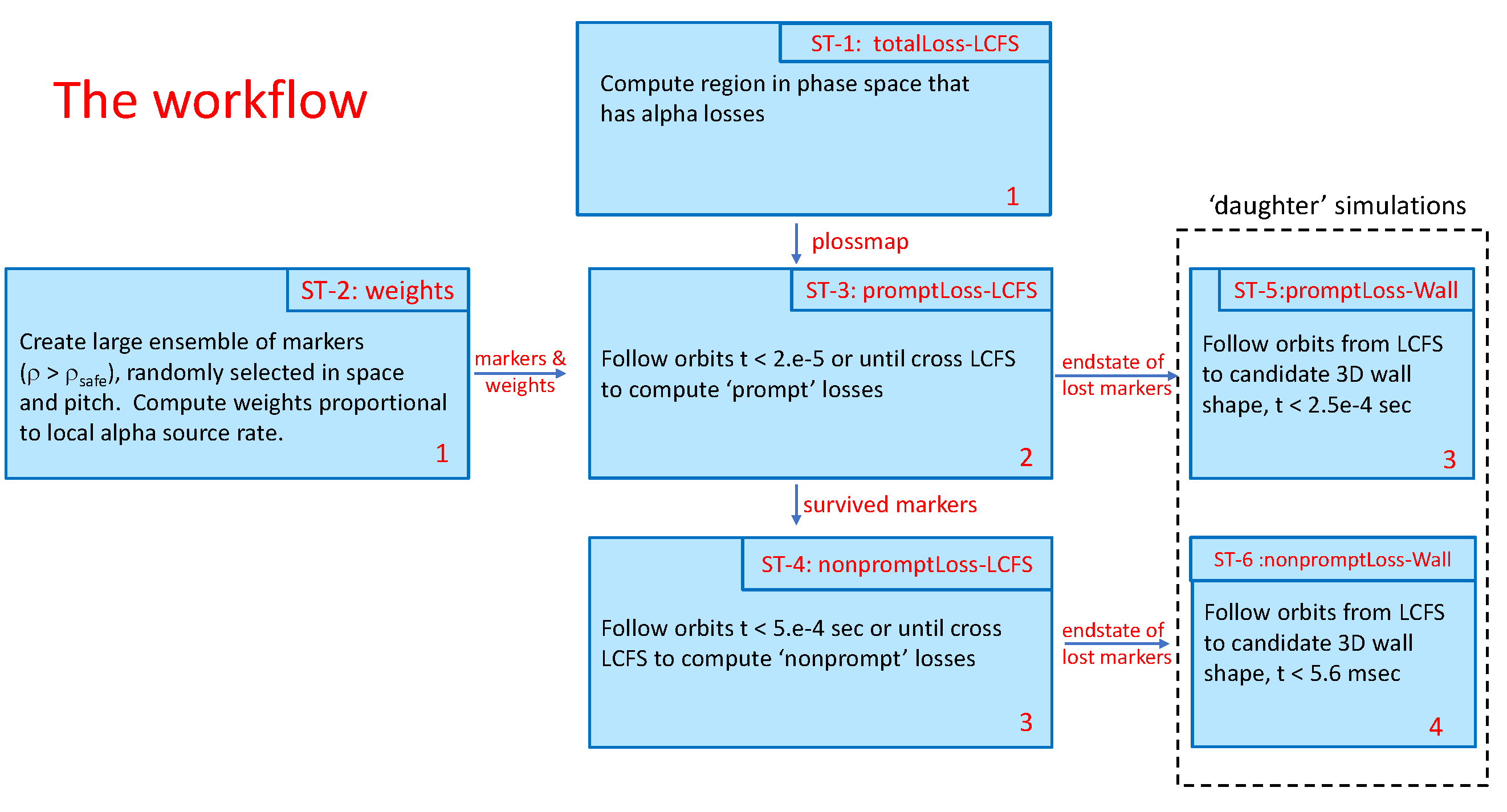
**Proposed workflow**

Figure 6 illustrates the proposed workflow. Currently the workflow is pretty complicated, but there may be ways to streamline it.

* Simulation **ST-0** (not shown) simulates a small number of markers, born throughout the entire plasma cross section (0 <  < 1) until either thermalization or until the markers cross the LCFS. The purpose of this run is to identify the radial region at birth that is ‘safe’ with respect to alpha losses, i.e. the region for which no alphas are lost. This simulation might be skipped if we can convince ourselves on the basis of the ST-1 simulation that our chosen radial range is safe.
* Simulation **ST-1** simulates a small number (30,016) of markers from birth until either thermaliztion or until the markers cross the LCFS. This simulation yields the fraction of alpha power that is lost by prompt- and nonprompt-lost alphas. It also yields data that will be used by the ‘lossmap’ algorithm (Särkimäki NF 2020) in the subsequent ST-3 simulation.
* Simulation **ST-2** simulates a large number (5 million) markers for a very short time, 10-6 sec. In fact, it isn’t clear that ASCOT needs to be run for this simulation – it may be sufficient to just genereate the input file. The sole purpose of generating this input file is to generate a large ensemble of markers, birth 0.7 < rho < 1.0 along with their corresponding weights.
* Simulation **ST-3** takes the marker ensemble generated by ST-2 and applies the lossmap algorithm to eliminate those markers whose birth position and pitch angle have no chance of being lost to the LCFS, as indicated by the data read from ST-1. This simulation applies a maximum simulation time (SIMTIME) of 2.e-5 sec, corresponding to the ‘prompt’ alpha losses. Markers are considered lost if/when their orbits cross the LCFS. The purpose of this simulation is to compute the spatial and energy distribution of prompt-lost markers at the LCFS.
* Simulation **ST-4** examines the ‘endstate’ data of simulation ST-3 to identify those markers whose orbits ‘survive’, i.e. those orbits which do not cross the LCFS within 2.e-5 sec. The birth locations and birth velocities of the survived markers in ST-3 are copied into the birth locations and birth velocities of the markers for ST-4. The orbits are followed for a simulation time of 5.e-4 sec in ST-4. Markers are considered lost if/when their orbits cross the LCFS. The purpose of this simulations is to compute the spatial and energy distribution of the nonprompt-lost markers at the LCFS.
* Simulation **ST-5** reads the ‘endstate’ data from the ‘promptloss’ simulation ST-3. The end positions and velocities of markers that are considered ‘lost’ in ST-3, i.e. those markers that hit the LCFS, become the ‘birth’ positions and velocities in ST-5. The marker orbits are followed for 2.5e-4 seconds or until the orbits strike a candidate 3D wall. Effectively, ST-5 follows orbits from the LCFS until they hit the wall. The purpose of this simulation is to compute the spatial distribution of the alpha loss (for prompt-loss) at the wall.
* Simulation **ST-6** reads the ‘endstate’ data from the ‘nonpromptloss’ simulation ST-4. Then, like ST-5, its ensemble of birth positions and velocities is taken from the endstate of ST-4 for those markers which crossed the LCFS. It follows the orbits for 5.6 msec or until the orbits strike the wall. The purpose of this simulation is to compute the spatial distribution of the alpha loss (for nonprompt-loss) at the wall.
* Simulations **ST-7** and **ST-8** (not shown in the Figure 6) are similar to ST-5 and ST-6, but instead of following all of the markers for a short time, they follow a much smaller number of markers for a long time (i.e. up to thermalization). The purpose of these simulations is to compute the fraction (but not the spatial distribution) of the birth-energy at the LCFS which eventually hits the wall. Since the LCFS is quite close to the wall in SPARC, typically more then 50% of the alpha energy which is ‘born’ at the LCFS eventually hits the wall.
* Simulation **ST-9** is a modification of ST-8 which uses ST-1 as its ‘parent’ simulation. This is preferred because, in principle, ST-8 (which uses ST-4 as its ‘parent’ simulation, which was a short (0.5 msec) simulation to compute the ensemble of markers at the LCFS) does not yield the correct energy distribution at the LCFS, which could in principle affect the fraction of power. But in fact, simulations ST-8 and ST-9 yield almost the same fraction of power lost between theLCFS and the wall, 56.2 vs 59.2% respectively.

Notes to self:

1. It would be trivial to combine ST-2 and ST-3, this would just require a little work in marker\_sets.py.
2. It may be possible to combine steps ST-3 and ST-4. We would run the simulation for the longer duration (0.5 msec) appropriate for the nonprompt losses, but then identify the prompt vs nonprompt losses in the postprocessor based on the simulation time when the markers cross the LCFS.



*Figure 6. Proposed workflow for efficient calculation of surface power density from lost alphas.*

Tables 1, 2, and 3 below tabulate information about actual ASCOT simulations that have been performed under the proposed workflow. They are mostly of interest to those CFS/PSFC staff who will be simulating alpha losses in SPARC going forward. Table 3 lists the computational efficiency of various simulations, where the efficiency is defined as the number of lost markers (or number of nonprompt lost markers) divided by the computational cost (= number of KNL nodes x wall-clock time for the simulation).

The ‘standard’ approach to computing the surface power density is represented by simulation ST-1, and it has a computational efficiency of 230 overall and 40.9 for nonprompt losses. The overall efficiency is increased to 1.5e6 for the ‘prompt’ simulation ST-3, and the efficiency for nonprompt losses is increased to 2783 in ‘nonprompt’ simulation ST-4. So effectively, the proposed workflow increases the computationa efficiency by a factor of ~68 for nonprompt losses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Simulation Type | Objective | script | runID | dependency |
|  |  |  |  |  |
| ST-1: totalLoss-LCFS | lossmap and LCFS power loss, prompt/nonprompt | 1095 | 40560057 | - |
| ST-2: weights | big marker ensemble and their weights | 1127 | 40877296 | - |
| ST-3: promptLoss-LCFS | spatial loss pattern of prompt losses to LCFS | 1135 | 40887796 | ST-1, ST-2 |
| ST-4: nonpromptLoss-LCFS | spatial loss pattern of nonprompt losses to LCFS | 1136 | 40889126 | ST-3 |
| ST-5: promptLoss-Wall | spatial loss pattern of prompt losses to 3D wall | 1137 | 40890729 | ST-3 |
| ST-6: nonpromptLoss-Wall | spatial loss pattern of nonprompt losses to 3D wall | 1141 | 40921464 | ST-4 |
| ST-7: promptLoss-WallTherm | prompt loss: fraction power loss, LCFS 🡪 3D wall | 1168 | 41428395 | ST-3 |
| ST-8: nonpromptLoss-WallTherm | Nonprompt loss: fraction power loss, LCFS 🡪 3D wall | 1167 | 41414086 | ST-4 |
| ST-9 nompromptLoss-Walltherm | Nonprompt loss: fraction power loss, LCFS 🡪 3D wall | 1180 | 41563386 | ST-1 |

*Table 1. Summary of types of ASCOT simulations for the prompt / nonprompt workflow for computing surface power density. The ‘dependency’ column indicates which ‘upstream’ simulations provide input data to the marker ensemble generated by the listed simulation. For example simulation ST-3 uses data from both simulations ST-1 and ST-2 to generate its ensemble of markers.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Simulation Type | Nmrk | Nlost | Nplost | time | cost | Objective |
|  |  |  |  |  |  |  |
| ST-1: totalLoss-LCFS | 3 e4 | 1.5 e4 | 2664 | 0.30 | 65.1 | lossmap and LCFS power loss, prompt/nonprompt |
| ST-2: weights | 5 e6 | 8.1 e5 | - | 1 e-6 | 0.37 | big marker ensemble and their weights |
| ST-3: promptLoss-LCFS | 3.4 e6 | 2.1 e6 | 0 | 2 e-5 | 1.37 | spatial loss pattern of prompt losses to LCFS |
| ST-4: nonpromptLoss-LCFS | 1.3 e6 | 70413 | 65951 | 5.e-4 | 23.7 | spatial loss pattern of nonprompt losses to LCFS |
| ST-5: promptLoss-Wall | 2.1 e6 | 1.8e6 | 0\* | 2 e-4 | 2.7 | spatial loss pattern of prompt losses to 3D wall |
| ST-6: nonpromptLoss-Wall | 7 e 4 | 33,934 | 33934\* | 5.6 e-3 | 8.3 | spatial loss pattern of nonprompt losses to 3D wall |
| ST-7: promptLoss-WallTherm | 1.5 e4 | 13,888 | 0\* | 0.10 | 9.4 | prompt loss: fraction power loss, LCFS 🡪 3D wall |
| ST-8: nonpromptLoss-WallTherm | 1.5 e4 | 9,053 | 9053\* | 0.10 | 17.4 | Nonprompt loss: fraction power loss, LCFS 🡪 3D wall |
| ST-9 nompromptLoss-Walltherm | 2664 | 1743 | 1743\* | 0.10 | 2.1 | Nonprompt loss: fraction power loss, LCFS 🡪 3D wall |

*Table 2. Summary of types of simulations to implement prompt / nonPrompt workflow. Nmrk is the number of markers at birth; Nlost is the number that hit a wall or the LCFS; Nplost is the number of nonprompt lost markers; time is the user-defined maximum simulation time in seconds. ‘Cost’ is the product of the number of KNL nodes used in the simulation and the wall-clock time required by the simulation. The asterisks indicate that the number of prompt lost markers equals the total number of lost markers because the ensemble of birth markers was constructed from nonprompt losses in a parent simulation.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Simulation Type | Nlost | Nplost | cost | Eff-all | Eff-np | Objective |
|  |  |  |  |  |  |  |
| ST-1: totalLoss-LCFS | 1.5 e4 | 2664 | 65.1 | 230 | 40.9 | lossmap and LCFS power loss, prompt/nonprompt |
| ST-2: weights | 8.1 e5 | - | 0.37 | - | - | big marker ensemble and their weights |
| ST-3: promptLoss-LCFS | 2.1 e6 | 0 | 1.37 | 1.5e6 | - | spatial loss pattern of prompt losses to LCFS |
| ST-4: nonpromptLoss-LCFS | 70413 | 65951 | 23.7 | 2971 | 2783 | spatial loss pattern of nonprompt losses to LCFS |
| ST-5: promptLoss-Wall | 1.8e6 | 0\* | 2.7 | 6.7e5 | - | spatial loss pattern of prompt losses to 3D wall |
| ST-6: nonpromptLoss-Wall | 33,934 | 33934\* | 8.3 | 4088 | 4088 | spatial loss pattern of nonprompt losses to 3D wall |
| ST-7: promptLoss-WallTherm | 13,888 | 0\* | 9.4 | 1477 | - | prompt loss: fraction power loss, LCFS 🡪 3D wall |
| ST-8: nonpromptLoss-WallTherm | 9,053 | 9053\* | 17.4 | 520 | 520 | Nonprompt loss: fraction power loss, LCFS 🡪 3D wall |
| ST-9 nompromptLoss-Walltherm | 1743 | 1743\* | 2.1 | 830 |  | Nonprompt loss: fraction power loss, LCFS 🡪 3D wall |

*Table 3. Computational efficiency for the simulations.*

**Total alpha power**

The kinetic profiles that we provide to ASCOT are derived from TRANSP simulations run by Pablo Fernandez-Rodriguez. He extracts a number of profiles including the local alpha source rate and the plasma volumes as a function of rho\_poloidal (normalized sqrt poloidal flux) and writes them to a file in the ‘pickle’ format.

There is a Python script currently living on the SPARC workstations (note to self: move this to NERSC so everyone can use it) called process\_ascot\_profiles\_mod.py which reads a pickle file, does some calculations, and writes some profiles to a flat-ascii file that one of my ASCOT preprocessors can read, and then write into the ASCOT input file.

One of the calculations that process\_ascot\_profiles\_mod.py does is to cumulatively integrate the alpha source rate as a function of rho\_poloidal, yielding the total alpha source rate (sec^-1) as a function of rho\_poloidal.

The alpha power in MW is PMW(rho) = S(rho) x 1.602e-19\*1.e-6 \* 3.5e6 = 5.61e-19 S(rho) , where S(rho) is the volume-integrated alpha source rate from the magnetic axis to rho. Here 1.602e-19 is the conversion from eV to Joules, 1.e-6 is the scaling factor for W / MW, and 3.5e6 is the alpha birth energy in eV.

   rho\_poloidal     PRD        scenario-1     scenario-2

          0.6                26.7%       24.4%         24.8%

          0.7                16.8%       14.2%         13.8%

           0.8                   8.2%        6.6%          6.2%

The table above tabulates what fraction f(rho), of the alpha power is born outside of a given rho\_poloidal for three SPARC plasma scenarios. For example, in the PRD (primary reference discharge), 16.8% of the alpha power is born outside rho=0.7.

This fraction is useful to know, since based on many simulations we expect that no alpha power is lost by alphas born inside rho = 0.7 (unless we assume really large ripple, e.g. due to very poor alignment of the TF coils or a number of TF coils smaller than 18). So we will typically generate an ensemble of alphas born only in the limited region 0.7 < rho < 1.0 and then perform orbit simulations for those alphas to see what fraction say cross the LCFS. If for example ASCOT concludes that 35% of the alpha power is lost to the LCFS for an ensemble of alphas born 0.7 < rho < 1.0, then the fraction of total alpha power that is lost to the LCFS would be 0.168 \* 0.35 = 0.0588 or 5.88%.

**ST-1: totalLoss-LCFS Total alpha power lost to LCFS: prompt and nonprompt (1165 / 41297780)**

ST = ‘simulation type’

Purpose: this simulation computes the fraction of alpha power that crosses the LCFS. It separates the losses into ‘prompt’ and ‘nonprompt’ fractions. Its output will also be used in the ‘lossmap’ winnowing-down of candidate ensembles of markers in subsequent simulations.

Python script group\_go\_1165.py was used to generate ASCOT input file group\_go\_1165.h5 which yielded ASCOT simulation 41297780. This simulation took 4 hours, 58 minutes on 14 nodes on the knl cluster.

Note that this run used a ‘nominal’ ripple file v1e\_nominal.txt, which assumes perfect alignment of the TF coils. Most of the other runs described in this document used ripple file v1e\_fixed\_inout\_case\_c.txt, which assumes that the TF coils are displaced in/out with a normal distribution with a sigma of 6 mm, but truncated at a maximum allowable sigma of 9 mm.

Markers: For this simulation, we used set=7 in python module define\_prt\_markers\_03 with rhomin=0.7 and rhomax = 1.0. Effectively, this generates an ensemble of markers uniformly distributed in physical space and uniformily distributed in velocity-direction over the region 0.7 < rho < 1.0. A total of 30,016 markers were simulated.

Options: This simulation invoked set 17 of Python module options\_sets. It turned off endcondition = “crossing LCFS” and turned on endcondition “wallhit”. The maximum simulation time was set to 0.12 sec, and the maximum CPU time was set to 17000 sec. Note: the maximum simulation time was chosen based on an examination of similar previous simulations with a longer simulation time; only a few markers were found to have a simulation time that exceeded 0.12 sec. We could have chosen a longer simulation time so as to capture all of the markers, i.e. none would terminate on the endcondition = exceeded maximum simulation time, but that would increase the execution time by several tens of percent with negligible improvement in the statistics of the computed alpha losses. Similarly, very few markers were found in a previous simulation to need more than 17,000 sec of computational time. Going forward, it may be best to remove the endcondition of maximum cpu time, and limit the effective time by imposing a limit on the maximum simulation time.

Wall: the wall is constructed by Python module write\_sparc\_conformal\_wall. And so, although ASCOT has been instructed to consider alphas as being lost only when they strike a “wall”, the wall itself is constructed to be coincident with the LCFS. So effectively, the markers are lost when they cross the LCFS.

Postprocessing

In the ‘runs’ directory, I did python $dir\_mypython/process\_ascot.py with the following parameters:

file\_name: ['ascot\_41297780.h5']

geq\_name: v1e.geq

fraction\_alphas\_simulated: 1.0

fn\_profiles: v1e\_profiles\_3.txt

ploss\_wall\_kw: 1.0

igrid\_spd: 0

stub filename for output: ascot\_41297780

parameter filename: shape81\_parameters.txt

suppress\_tbl: True

max\_markers: 0

Note that the software assumes that markers are being followed to a physical wall but in this simulation the markers are only being followed effectively to the LCFS. So some plots and tables having to do with the spatial distribution of the losses may be bogus.

This generates files ascot\_41297780.pdf and ascot\_41297780.txt.

Examining ascot\_41297780.txt, we see the following:

* Circa line 90: 15206 markers thermalized, 14333 hit the wall, and 477 terminated beause the simulation time exceeded the maximum allowable simulation time of 0.12 sec. My home-grown software regards markers whose end condition is SIMTIME to be ‘confined’ or ‘survived’ but there is some chance that some of those 477 markers might have crossed the LCFS if we had increased the simulation time. So this represents a very small potential error in the simulation … but it really is small, because only 3.3% of the markers terminated due to SIMTIME. Also, there were no markers whose orbit simulation was terminated due to CPUTIME. So our specification of a maximum CPUTIME of 17000 sec is OK.
* Circa line 172: 47.8% of the markers were lost to the wall. When marker ‘weighting’ is included, 18.7% of the weighted markers were lost to the wall, and 17.07% of the weighted marker energy was lost to the wall.
* Circa line 301: of the energy that is lost to the wall, 83.98% was lost within 2.e-5 sec of the orbit’s simulation (so-called ‘prompt’ loss) and 16.02% was lost after an orbit’s simulation of 2.e-5 sec (‘nonprompt’ loss).

Efficiency: This simulation got 12,490 prompt-lost markers and 1843 nonprompt-lost markers. The maximum surface power density is dominated by the nonprompt-lost markers. So the efficiency is 1843/ (14 nodes \* 4.966 hr) = 26.5 non-prompt-lost markers per node-hour.

Combining Total Alpha Power and total alpha power lost to LCFS

ST = ‘simulation type’

* The fraction of total alpha power lost to the LCFS is 0.168 \* 0.1707 = 2.87%
* The fraction of total alpha power prompt-lost to the LCFS is 0.168 \* 0.1707 \* 0.8398 = 2.41%
* The fraction of total alpha power non-prompt lost to the LCFS is 0.168 \* 0.1707 \* .1602 = 0.46%.

So for a total fusion power of 98.5 MW, the total alpha power is 19.7 MW and the total alpha power lost to the LCFS is 19.7 MW x 1000 x 0.0287 = 565 kW, of which 475 kW is prompt-lost and 90 kW is nonprompt lost. Keep in mind that this simulation was for ‘nominal’ ripple; the nonprompt losses (presumably primarily ripple) would be larger if the TF coils are assumed to have some degree of misalignment.

**ST-1: Total alpha power lost to LCFS: prompt + nonprompt) (1095 / 40560057)**

**For** **v1e\_fixed\_inout\_case\_c.txt**

This is the same simulation ‘type’ as above, the only difference is the rippled 3D magnetic field that it uses.

I see that the simulation above, 1165 (for v1e\_nominal.txt), was actually copied/modified from simulation 1095 (40560057). Simulation 1095 did use the misaligned-TF-ripple v1e\_fixed\_inout\_case\_c.txt. Its only other difference from 1165 was that it allowed a longer simulation time, 0.3 sec versus 0.12 sec.

Probably for this reason, simulation 1165 did not have any markers terminated due to SIMTIME, instead it got 14862 markers thermalizing and 15154 hitting the wall.

Simulation 1165 got 50.5% of the markers hitting the wall; the weighted marker loss to the wall (again, actually to the LCFS) was 21.8% and the weighted energy loss to the wall was 19.41%.

Of the power hitting the LCFS, 73.69% was prompt-lost and 26.31% was nonpromopt lost.

So the total fraction of power lost to the LCFS is 0.168 \* 0.1941 = 3.27%, corresponding to 642 kW. The fraction of prompt-lost is 0.168 \* 0.1941 \* 0.7369 = 2.40% and the fraction of nonpromptlost is 0.168 \* 0.1941 \* 0.2631 = 0.86%. The corresponding powers are prompt = 473 kW and nonprompt = 169 kW.

**ST-2: ‘Weights’ simulation (1127 / 40877296)**

Purpose: The purpose of this simulation is to generate a large ensemble of markers (5,000,000) for subsequent simulations. The orbits are followed only for 1 microsecond because we really don’t care about the orbits (i.e. confined or unconfined or fraction lost etc.). We simply want ASCOT to compute a large ensemble of marker positions, velocities, and their corresponding weights

In subsequent Python scripts to generate marker ensembles, we will read in (a) the marker data created by this simulation, and ‘ini’ and ‘end’ data from a simulation-to-thermalization to determine which marker starting positions and velocity vectors have a non-zero chance of being lost.

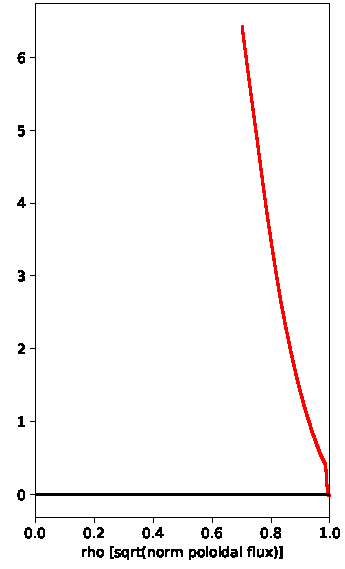
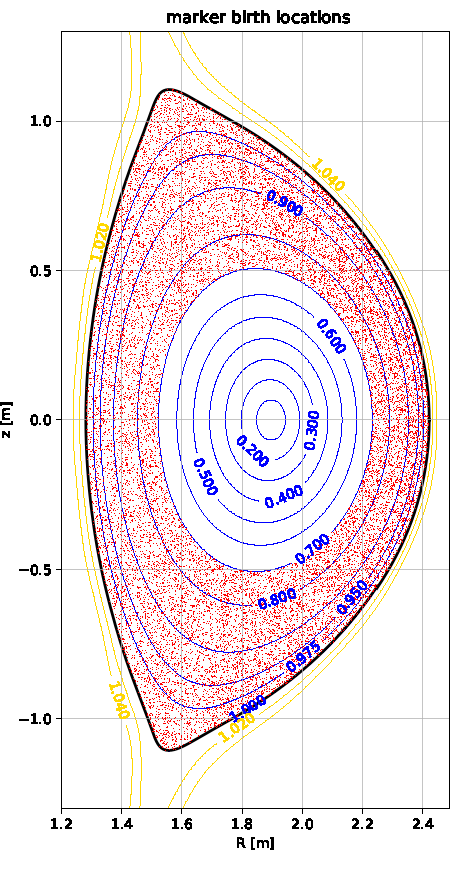
The input file for this simulation was generated by group\_go\_1127.py and its ASCOT run number is 40877296; the ascot output file is ascot\_40877296.h5.

Most of the analysis parameters for this simulation are identical to 1095. In particular, the markers are distributed uniformly in space over the range 0.7 < rho < 1 and they are distributed uniformly in velocity space. The simulation used set=7 in python module define\_prt\_markers\_03 (as in 1095). The number of markers was increased from 30,016 to 5,000,000 in this simulation.

The ‘options’ are also the same as in 1095 except that:

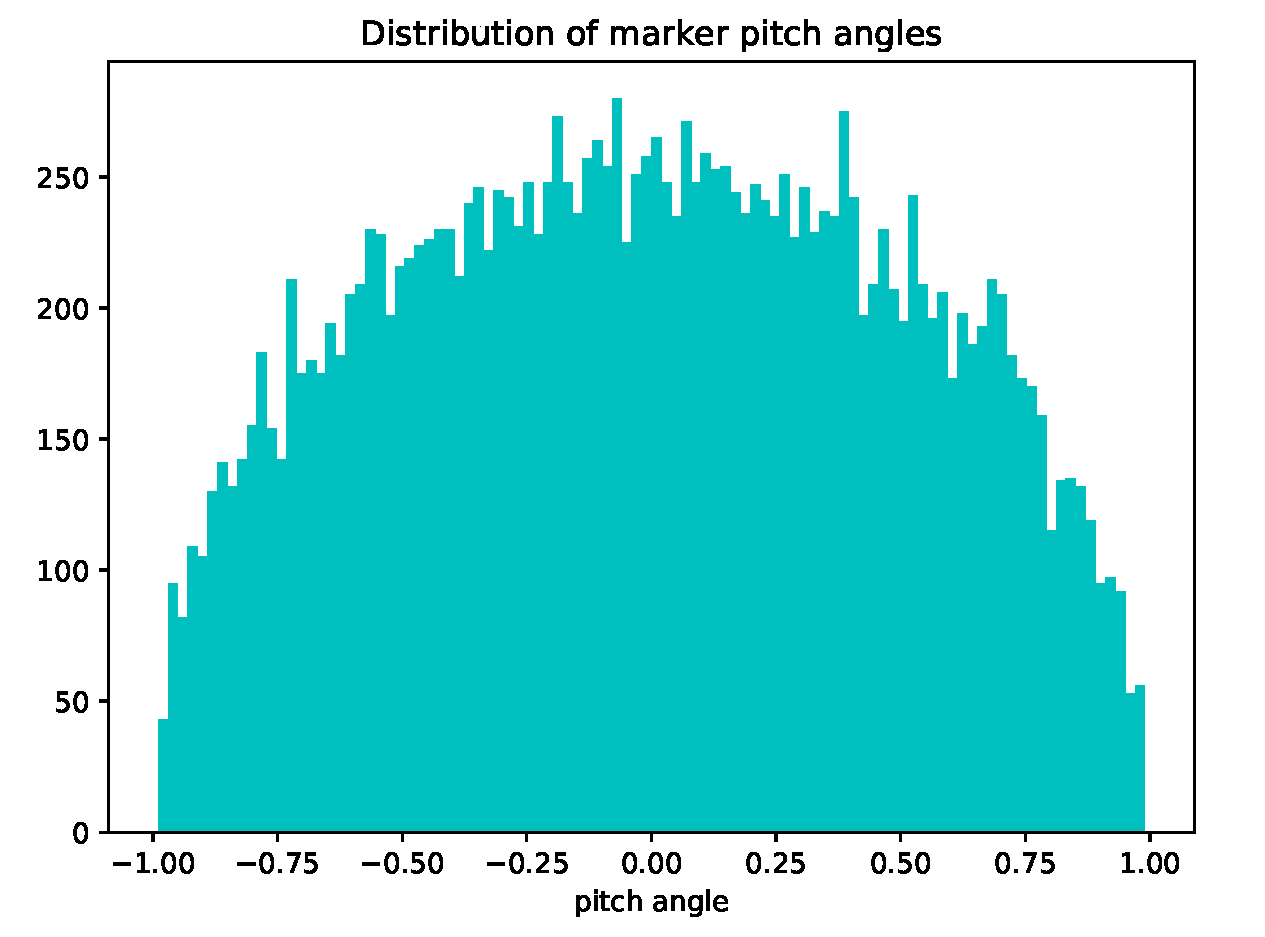
* The simulation time was decreased from 0.3 to 1.e-6 sec
* A dictionary entry options\_settings[“my\_no\_orbitwrite"] was set to 1 in this simulation, which causes ASCOT to NOT save the last N orbit positions. The reason for this is that with 5 million markers, that would take too much memory, risking a segmentation fault.

It turns out that setting options\_settings[“my\_no\_orbitwrite"] = 1 was a little mistake, because some of my home-grown Python postprocessing scripts fail if there is no recorded orbit data at all. So going forward, I commented out that line, but asked for ASCOT to record only a single orbit position (rather than 1000 orbit positions in 1095), which avoids asking for too much memory and thereby avoids the segmentation fault.



*Figure 7. Left: R-Z distribution of markers. Right: weights of markers as a function*

*of rho (designed to be proportional to local alpha source rate).*



*Figure 8. histogram of number of markers as a function of pitch angle. The distribution shown, which has the number of markers proportional to cos(pitch angle), is equivalent to having velocity vectors that have random orientation.*

**ST-3: promptLoss-LCFS simulation (1135 / 40887796)**

Purpose: to compute the spatial pattern of markers that are ‘prompt-lost’ to the LCFS.

Markers: the ensemble of markers is computed by Python module define\_prt\_markers\_10 (set=1) which applies the lossmap algorithm. The lossmap data is taken from ascot\_40560057.h5 and the marker information, including marker weights, is taken from ascot\_40877296.h5. The parameter ‘unaccounted’ is set to 0.0, which means that any ‘bin’ that gets even a single lost marker will be retained in the ensemble of generated markers.

Options: the maximum simulation time is set at 2.e-5 sec, which previous simulations have demonstrated is the boundary between ‘prompt’ and ‘nonprompt’ losses. Markers are considered to be lost when they strike the wall (rather than cross the LCFS), but the wall is constructed to be coincident with the LCFS. So effectively, markers are lost when they cross the LCFS.

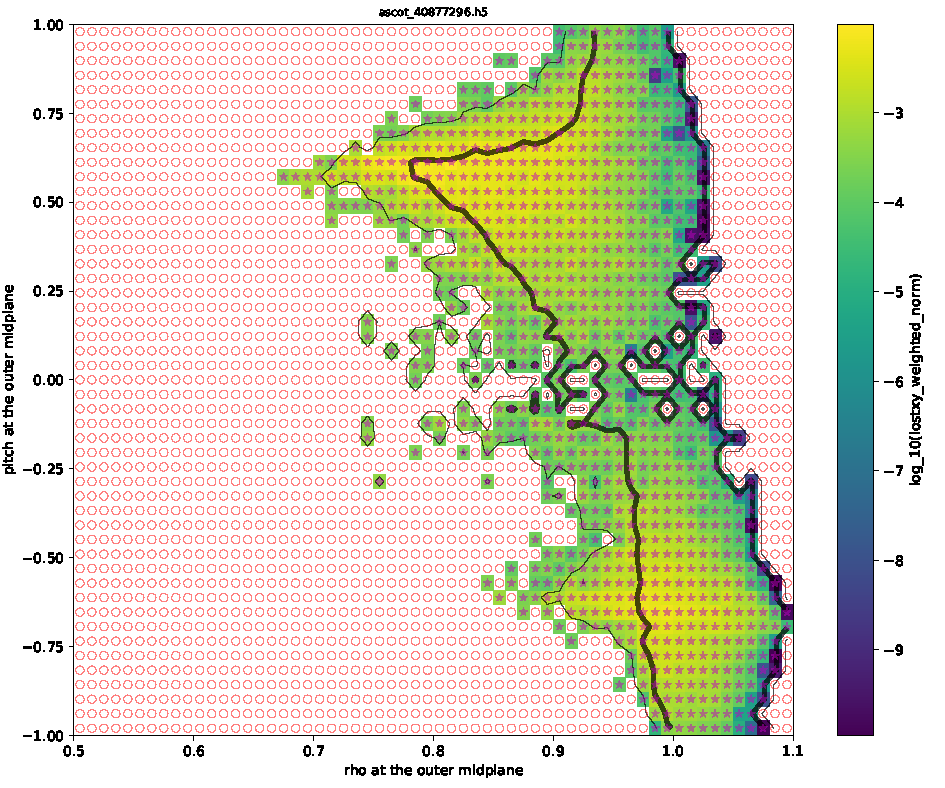
Recording of orbit positions is turned on, but only the last orbit position is saved, so as to not consume too much memory. The switch my\_go\_record\_mode is set to 1, which means that when ASCOT records the orbit position, it will record the actual gyro-orbit position rather than the corresponding guiding-center position.

The Python script which generated this simulation was 1135, and the corresponding ASCOT run number was 40887796 (ascot\_40887796.h5). It took 8 minutes and 12 seconds on 10 KNL nodes.

When generating the ensemble of markers (python group\_go\_1135py init), the software reports that we need to only simulate 67.9% of the unweighted markers (the remainder correspond to bins for which no marker was found to be lost) and the sum of the weights of the accepted markers was 0.409.

Of the 5,000,000 markers available in the ‘weights’ file, a total of 3,397688 , or 67.9%, were copied into the marker ensemble generated by the script (group\_go\_1135.h5). So applying the lossmap algorithm, which eliminates markers on the basis of both birth position and pitch angle, reduces the number of markers by a about 32% relative to what we would have gotten if we just accepted all markers with 0.7 < rho < 1.0.

Of these birth markers, 2,088,378 hit the wall and 1,309,309 hit the max SIMTIME limit. This shows the benefit of the prompt/non-prompt marker scheme: we get more than a million markers to the LCFS for an investment of just 8 minutes on 10 nodes.



*Figure 9. Lossmap for 1135. The filename for this plot is ascot\_40877296\_lossmap.pdf but that is a misnomer because 40877296 provided the marker info and the marker weights, whereas the actual lossmap data was taken from from ascot\_40560057.h5*

**ST-4: nonpromptloss-LCFS simulation ( 1136 / 40889126)**

Purpose: to compute the spatial pattern of markers that are ‘**non**prompt-lost’ to the LCFS.

The Python script for this simulation is group\_go\_1136.py and the ASCOT runID is 40889126 (ascot\_ 40889126.h5). It took 71 minutes using 20 nodes on knl.

Markers: group\_go\_1136.py calls Python module def\_prt\_markers\_09 with set=1. This module reads in the marker and endstate data from a ‘parent’ run, which in this case is just the promptloss simulation (1135) discussed in the previous section. It examines the ‘endstate’ of markers in the parent run. Only those markers which did not hit the wall (which was coincident with the LCFS in the parent run) in the parent simulation have their birth-marker information copied to the new ensemble.

Options: The maximum simulation time is set to 5.e-4 seconds. Markers are considered lost when their orbits strike the wall, and they are NOT considered lost just if their orbit crosses the LCFS. But the wall is constructed to be coincident with the LCFS, so effectively markers are lost when their orbits cross the LCFS. The dictionary entry my\_go\_record\_mode is set to 1, which tells ASCOT to record the marker orbit position at the true gyro-orbit position rather than the corresponding guiding-center position. A total of 1309309 markers were constructed for this simulation, of whch 70,413 hit the ‘wall’ (i.e. the LCFS) and 1,238,896 terminated due to their simulation time exceeding the maximum imposed SIMTIME.

The simulation took 71 minutes on 20 knl nodes. So the ‘efficiency’ is 70413/ (20\* 1.183) = 2976 nonprompt lossmarkers per node hour, which is slightly more than a factor of 100 times more efficient than the straightforward approach, group\_go\_1165.py, which did not separate the losses into prompt and non-prompt groups. But keep in mind that we had to do some other simulations to get to this simulation, and it is the total CPU time of all the simulations that counts.

**ST-5: promptLoss-Wall simulation ( 1137 / 40890729 )**

Purpose: to follow marker orbits from ‘birth’ at the LCFS for 2.e-4 sec or until they strike a candidate wall shape. The parent ensemble of markers were ‘prompt lost’ to the LCFS (1135/40887796).

Markers: generated by def\_prt\_markers\_05 with set = 1. This simply copies the endstate of markers which were lost to the ‘wall’ in the parent simulation ascot\_40887796 into the ‘marker’ state for this simulation. remember that in the parent simulation, the endstate was defined as being ‘hit the wall’ (rather than cross the LCFS), but the wall was constructed to be coincident with the LCFS. Because the number of such markers is large (over 2 million) there is a switch in the markers-options to limit the number of markers plotted to 30,000 so that we don’t waste time making plots.

Options: markers are considered lost when they hit a wall, and they are not considered lost if they cross the LCFS. my\_go\_record\_mode = 1 so that the recorded marker position is the true gyro-orbit marker position rather than the corresponding guiding-center position.

Wall: Python module write\_3d\_wall is invoked, it is given wall file shape51\_2\_triangles.txt (generated by triangulate\_torus\_2.py, now triangulate\_torus\_3.py).

Performance: simulation 4089026 took 8 min, 6 sec on 20 nodes on knl. There were 2,088,378 birth markers of which 1,773,316 were lost to the wall and 315,062 expired due to the simulation time exceeding SIMTIME = 2.e-4.

~~Note: Because (1773/2088 = 85%) of the markers launched at the LCFS hits a wall within 2.e-4 sec, it seems reasonable to assume that if we followed the orbits for a longer simulation time, all of the launched power would hit the wall. At the very minimum, making that assumption would only cause an overestimate of about 15%. So for prompt losses and close-fitting limiter/wall geometries, for purposes of calculating the fraction of prompt-lost alpha power, we will assume that “crossing the LCFS = will hit the wall well before thermalization”.~~

**ST-6: nonpromptLoss-Wall simulation (1141/40921464)**

Purpose: to follow marker orbits from ‘birth’ at the LCFS for 5.6e-3 sec or until they strike a candidate wall shape. The parent ensemble of markers were ‘prompt lost’ to the LCFS (1136/40899126).

Note: With respect to options and simulation parameters, this simulation is identical to the previous one: we are simply trying to follow orbits from ‘birth’ at the LCFS until they hit a wall or until the maximum simulation time is exceeded. The only differences are (1) the maximum simulation time is increased to 5.6 milliseconds; and (2) the ‘parent’ marker ensemble is taken from ascot\_40899126.h5 (which is a simulation of nonprompt losses to the LCFS).

Markers: generated by def\_prt\_markers\_05 with set = 1. This simply copies the endstate of markers which were lost to the ‘wall’ in the parent simulation ascot\_40899126.h5 into the ‘marker’ state for this simulation. remember that in the parent simulation, the endstate was defined as being ‘hit the wall’ (rather than cross the LCFS), but the wall was constructed to be coincident with the LCFS. Because the number of such markers is large (over 2 million) there is a switch in the markers-options to limit the number of markers plotted to 30,000 so that we don’t waste time making plots.

Options: markers are considered lost when they hit a wall, and they are not considered lost if they cross the LCFS. my\_go\_record\_mode = 1 so that the recorded marker position is the true gyro-orbit marker position rather than the corresponding guiding-center position.

Wall: Python module write\_3d\_wall is invoked, it is given wall file shape51\_2\_triangles.txt (generated by triangulate\_torus\_2.py, now triangulate\_torus\_3.py).

Performance: Python script groiup\_go\_1141.py 🡪 simulation 40921464 took 25 minutes on 20 nodes on knl. There were 70,413 birth markers of which 33,934 were lost to the wall and 36,479 expired due to the simulation time exceeding SIMTIME = 5.6 milliseconds.

Scaling: In the ‘standard’ approach = **Total alpha power lost to LCFS (prompt and nonprompt)**, we followed 30,016 markers of which 1,874 were nonprompt lost to the wall. the nonprompt losses dominate the maximum surface power density. In this nonpromptLoss simulation, we got 33,934 markers to the wall. So this simulation is “equivalent” to following 30,016 \* (33934/1874) = 543,523 markers in the standard approach. Here “equivalent” means the simulations would yield the same number of nonprompt lost markers to the wall, which sets the statistical accuracy of the maximum surface power density.

**ST-7: promptLoss-WallTherm simulation (to thermalization: 1168 / 41428395)**

The initial simulation (**ST-1:** **Total alpha power lost to LCFS (prompt and nonprompt)** was sufficient to compute the fraction of alpha power that strikes the LCFS before thermalization. But in the promptWall simulation above with a short simulation time of 2.e-4 milliseconds, about 85% of the markers were lost before reaching SIMTIME.

This raises the question of whether all of the promptLoss power ‘born’ at the LCFS will actually strike a wall before thermalization.

So I copied 1137 into 1168, reduced the number of markers to 15008, but then increased the SIMTIME from 0.0056 to 0.100 sec.

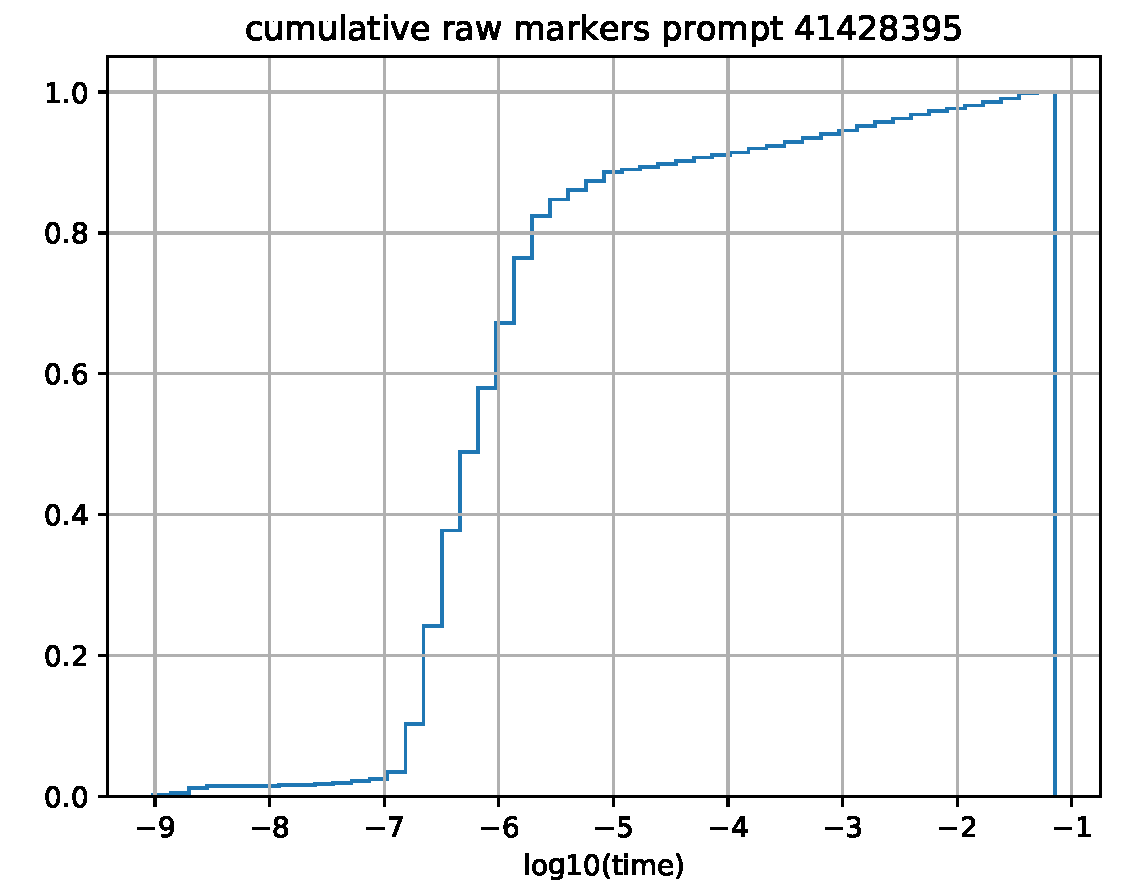
Results: Markers born = 15,008. Markers lost = 13,888. Markers simtime = 3.

Fraction of unweighted particles lost = 92.5%.

Fraction of weighted particles lost = 89.2%.

Fraction of weighted energy lost = 85.1%.

Figure 10 plots the cumulative number of lost markers as a function of simulation time for this simulation. Note that about 90% of the markers which are eventually lost are lost within a time 0.1 msec of birth. This is why we selected a SIMTIME of 2.e-4 seconds for the promptLost-Wall simulation (ST-5): this is a sufficient time to capture 90% of the losses, so we are assured that the spatial pattern of the losses for the short simulation would match that of a full simulation.



*Figure 10. Cumulative normalized number of markers lost to the wall based on ‘prompt-lost’ ensemble of markers at the LCFS. Approximately 90% of the markers are lost within a time ~0.1msec of birth at the LCFS.*

**ST-8: nonpromptLoss-WallTherm simulation ( to thermalization: 1167/ 41414086)**

The initial simulation (**ST-1:** **Total alpha power lost to LCFS (prompt and nonprompt)** was sufficient to compute the fraction of alpha power that strikes the LCFS before thermalization. But in the nonpromptWall simulation above with a short simulation time of 5.6 milliseconds, only about half of the markers were lost before reaching SIMTIME.

This raises the question of whether all of the nonpromptLoss power ‘born’ at the LCFS will actually strike a wall before thermalization.

So I copied 1141 into 1167, reduced the number of markers from about 70,000 to 15008, but then increased the SIMTIME from 0.0056 to 0.100 sec.

The run number is 41414086.

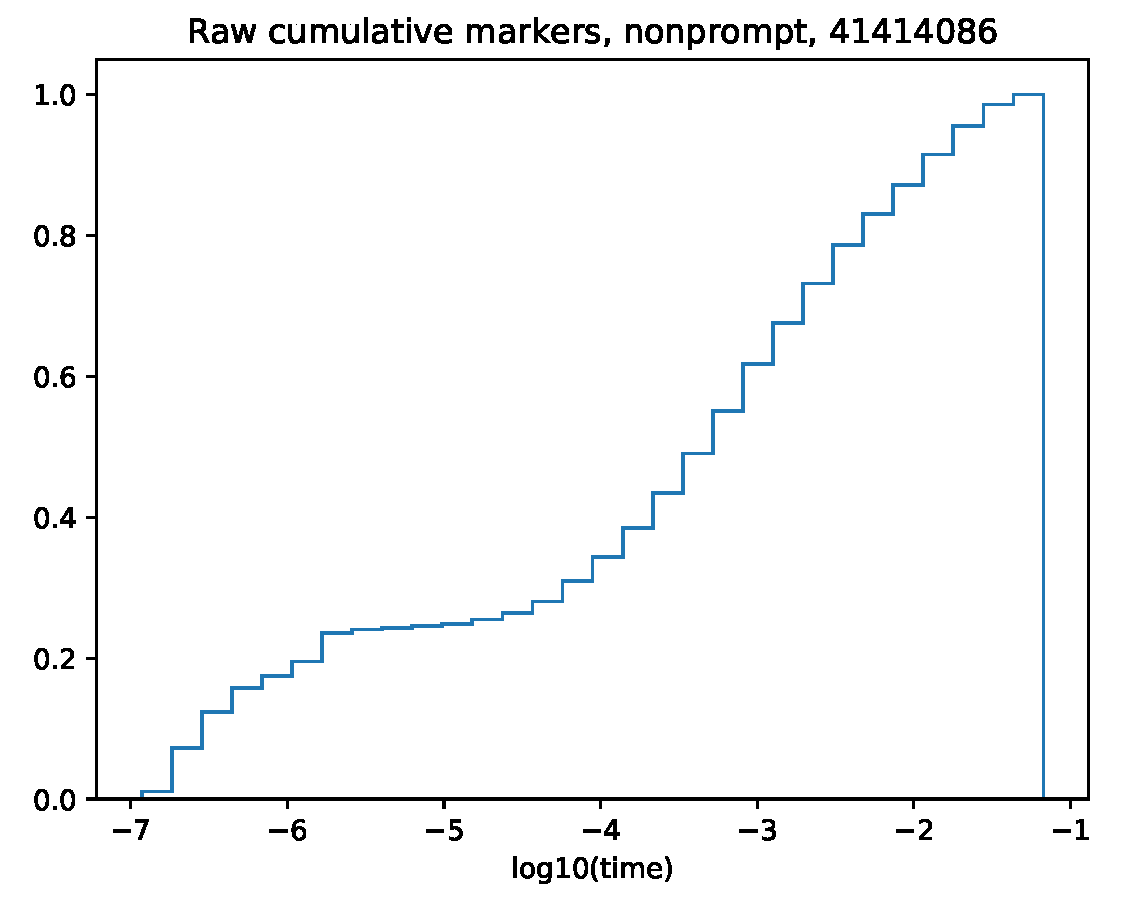
Results: Markers born = 15,008. Markers lost = 9053. Markers simtime = 42.

Fraction of unweighted particles lost = 60.3%.

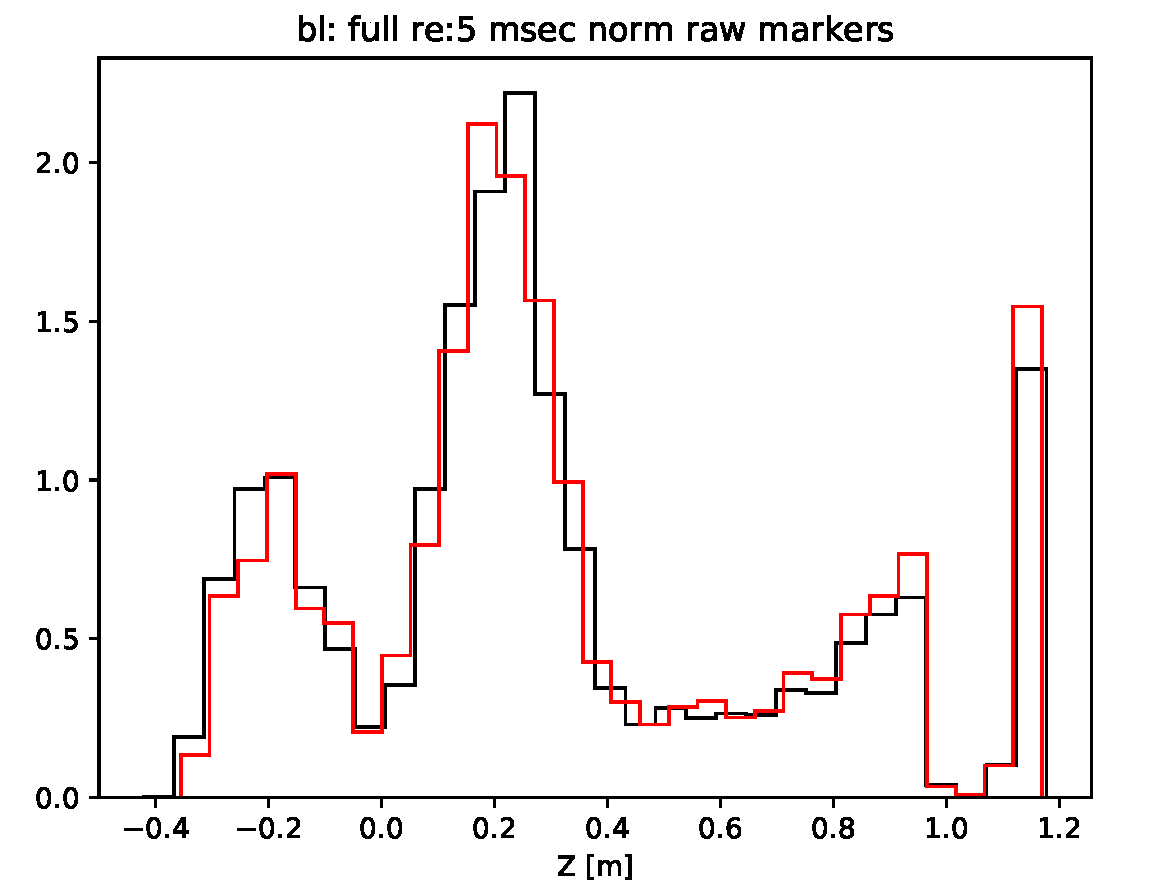
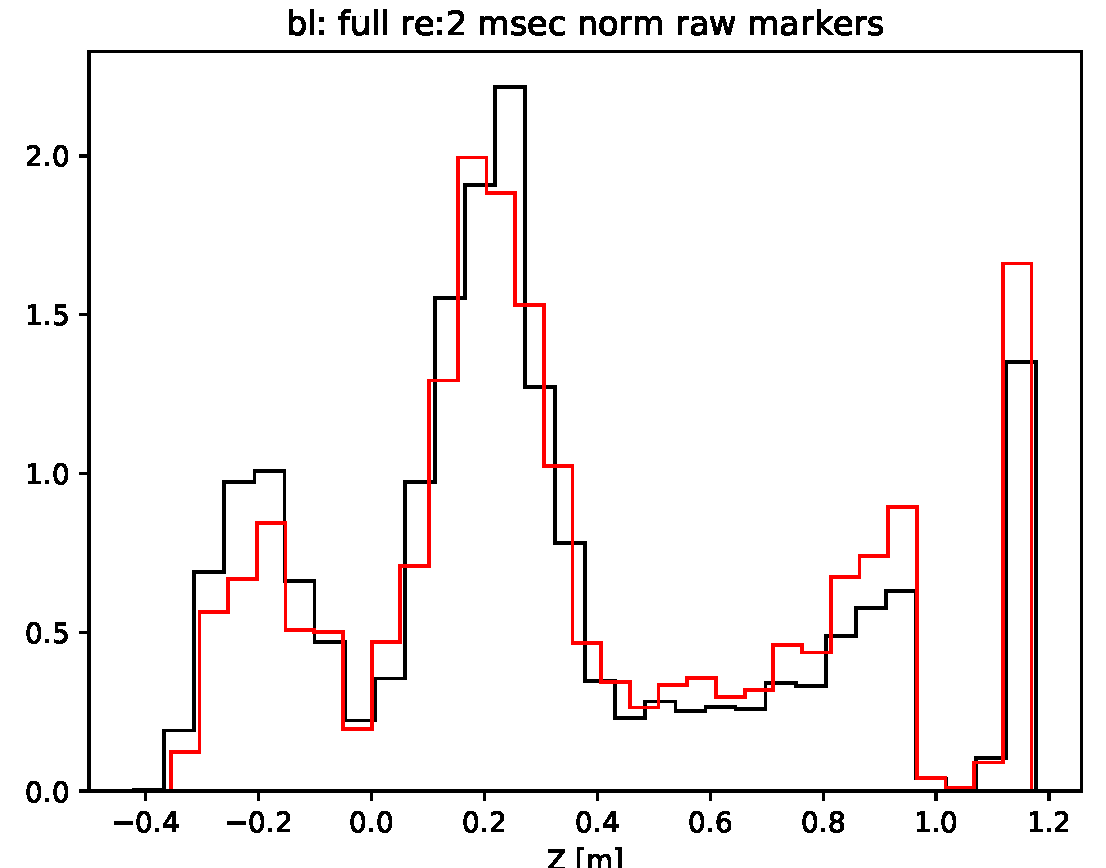
Fraction of weighted particles lost = 62.6%

Fraction of weighted energy lost = 52.6%

Figure 11 below plots the cumulative number of markers that strike the wall as a function of time for this simulation. Approximately 83% of the markers that are eventually lost are lost within 5 milliseconds. This is why we defined a simulation time of approximately 5 milliseconds for simulation type ST-6: it is sufficiently long to catpure most of the losses. Figure 12 below shows that the spatial distribution of losses as a function of elevation is very similar for simulations of 2 and 5 milliseconds compared to a full to-thermalization simulation.



*Figure 11. Cumulative number of markers striking the wall as a function of simulation time for ASCOT simulation 41414086 which followed markers from their ‘birth’at the LCFS to a defined 3D wall.*



*Figure 12. (left) Histogram of marker elevation upon striking the wall; black represents the full to-thermalization simulation and red represents a simulation for 2 milliseconds. Right: black represents the full to-thermalization and red represents a simulation for 5 milliseconds.*

**Computing fraction of power lost to wall: prompt and non-prompt**

Combining the outputs of the several codes, the total power of prompt alpha losses to the wall is 2.04% and the total power of nonprompt alpha losses to the wall is 0.51%. For a fusion power of 100 MW, that yields 408 kW and 102 kW of prompt and nonprompt losses, respectively. These numbers are smaller than reported previously (470 kW prompt, 170 kW nonprompt losses) because the previous numbers did not take into account the fact that not all power ‘lost’ to the LCFS is lost to the wall.

An outstanding issue is why the fraction of nonprompt lost power from the LCFS to the wall is only 59%; the corresponding fraction for the prompt losses was 85%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **prompt** | **sim** | **run** | **processor** | **fraction** |
| alphas born >0.7 |  | SPARC\_V1E\_transp\_3mod.pkl | process\_sparc\_profiles\_mod.py | 0.168 |
| loss: =0.7 🡪 LCFS | ST-1 | 1095/40560057 | process\_ascot.py | 0.143 |
| Loss: LCFS 🡪 wall | ST-7 | 1168/41428395 | process\_ascot.py | 0.851 |
| total to LCFS |  |  |  | 0.240 |
| total to wall |  |  |  | 0.0204 |
| **nonprompt** |  |  |  |  |
| alphas born >0.7 |  |  | process\_sparc\_profiles\_mod.py | 0.168 |
| loss: =0.7 🡪 LCFS | ST-1 | 1095/40560057 | process\_ascot.py | 0.0511 |
| Loss: LCFS 🡪 wall | ST-8 | 1167/41414086 | process\_ascot.py | 0.592 |
| total to LCFS |  |  |  | 0.0086 |
| total to wall |  |  |  | 0.0051 |

*Table 4. Sources of information to compute fraction of prompt and nonprompt alpha power that is lost to the wall for the SPARC primary reference discharge.*

**Energy distribution of nonprompt Losses at LCFS**

This section motivated a new approach for computing the fraction of energy lost by the nonprompt-loss component between the LCFS and the wall. See the next section.

The overriding objective of the prompt/nonprompt workflow for computing the surface power density is to allow us to follow marker orbits born say 0.7 < rho < 1.0 for a short simulation time, typically 0.5 msec (for nonprompt-lost markers), rather than for the full thermalization time of ~100 msec. We have demonstrated previously that the spatial pattern of losses at the LCFS does not depend on the duration of the simulation time. Good.

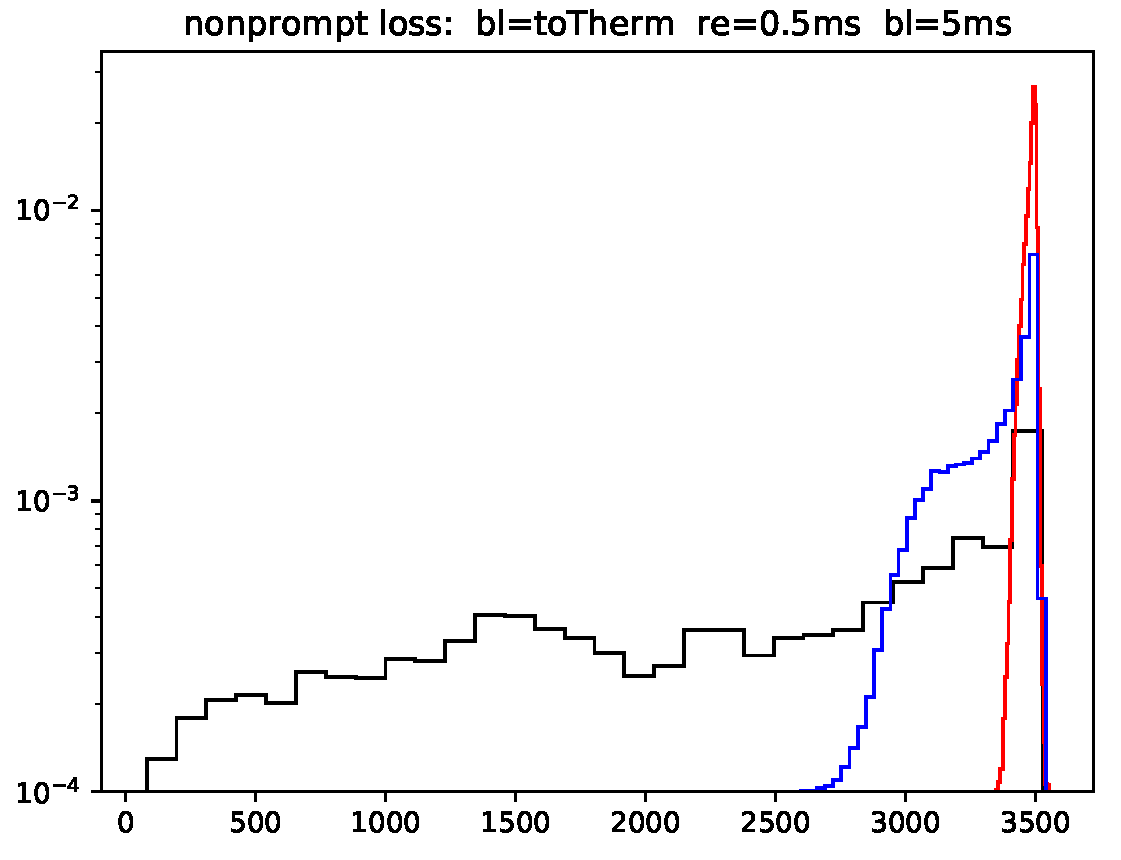
But not surprisingly, as shown in the figure below, the energy distribution of the nonprompt losses does depend on the duration of the simulation time. An orbit simulation time of 0.5 ms is much shorter than the thermalization time for markers born 0.7 < rho < 1.0, which is of order 100 msec. So the energy distribution of markers crossing the LCFS at t < 0.5 msec is very close to the birth energy of 3.5 MeV, whereas the distribution for a to-thermalization is much broader in energy.

The question is whether the difference between the true energy distribution of lost markers at the LCFS and that computed by a short (0.5 msec) simulation will significantly affect the computed spatial distribution at the wall. I argue below that it will not, but this needs to be verified by a comparison against actual simulations.

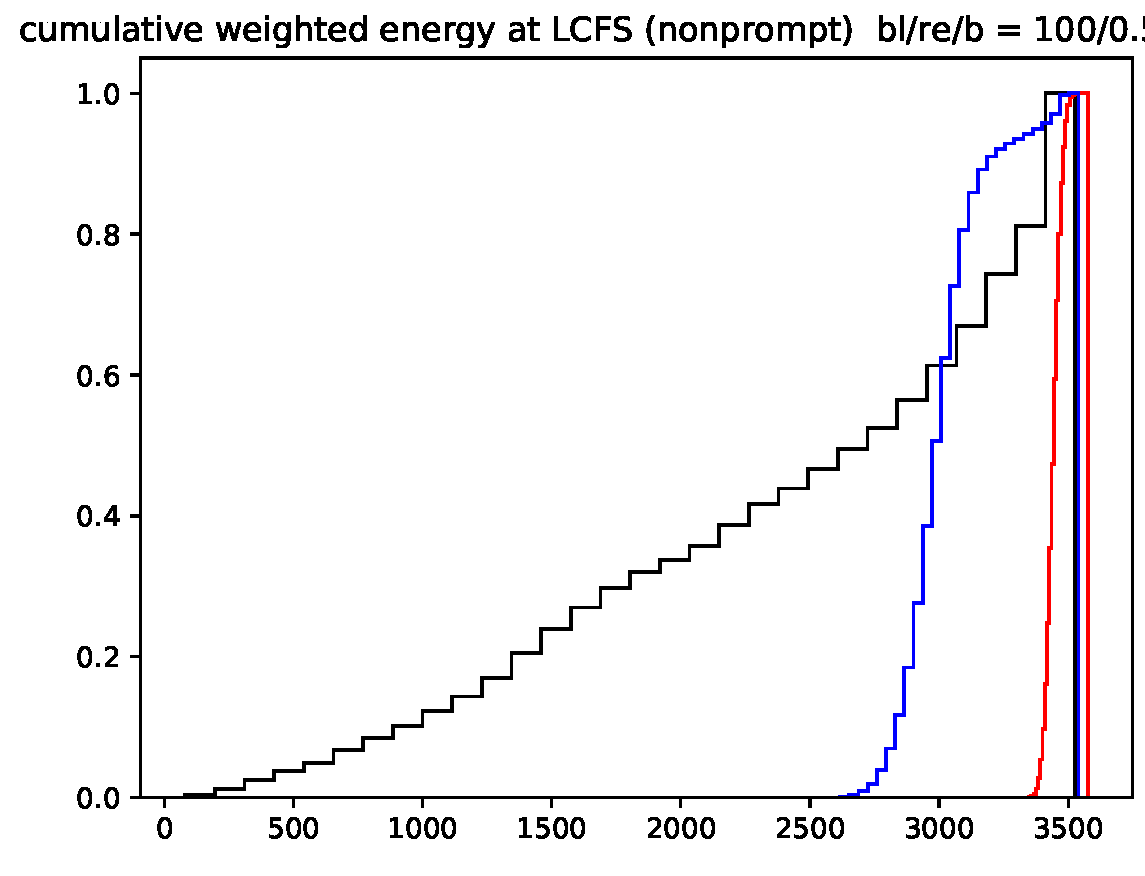
A related issue is whether the incorrect energy distribution at the LCFS will affect the computed fraction of energy loss between the LCFS and the wall. This issue was neglected during the development of the prompt/nonprompt workflow because it was assumed that the vast majority of power which hits the LCFS will eventually hit a wall, but that assumption proved incorrect. Because the energy distribution at the LCFS computed by a short simulation will be unphysically weighted toward higher energy, a simulation of markers between the LCFS and the wall that is based on a short orbit simulation to the LCFS will overestimate the loss to the wall. This error may be significant particularly for tokamaks such as ITER that have a large gap between the LCFS and the wall.

I reran 1167/41414086 as 1177/41540507 but imposed vmult=0.707, i.e. artificially decreased the birth energy of all markers by a factor of two. The computed fraction of lost energy to the wall decreased from 56.2% to 44.8%, a relative decrease of 20%.

Figure 13 is the cumulative loss of weighted markers as a function of energy. Of course it still looks like a big difference between a full-thermalization and say the red curve for the 0.5 msec simulation. At the black curve at an energy of 2.0 MeV, the cumulative loss is 40% of the total. But at an energy of 2 MeV, the ratio of the gyroradius to the gyroradius at birth at 3.5 MeV is (2/3.5)^2 = 75%, i.e. there is only a 25% difference in gyroradius between birth at 3.5 MeV and 2.0 MeV. To the extent that the spatial loss pattern is governed by the gyroradius, we might expect that all markers lost with energies above 2 MeV will have a similar spatial loss pattern at the wall.



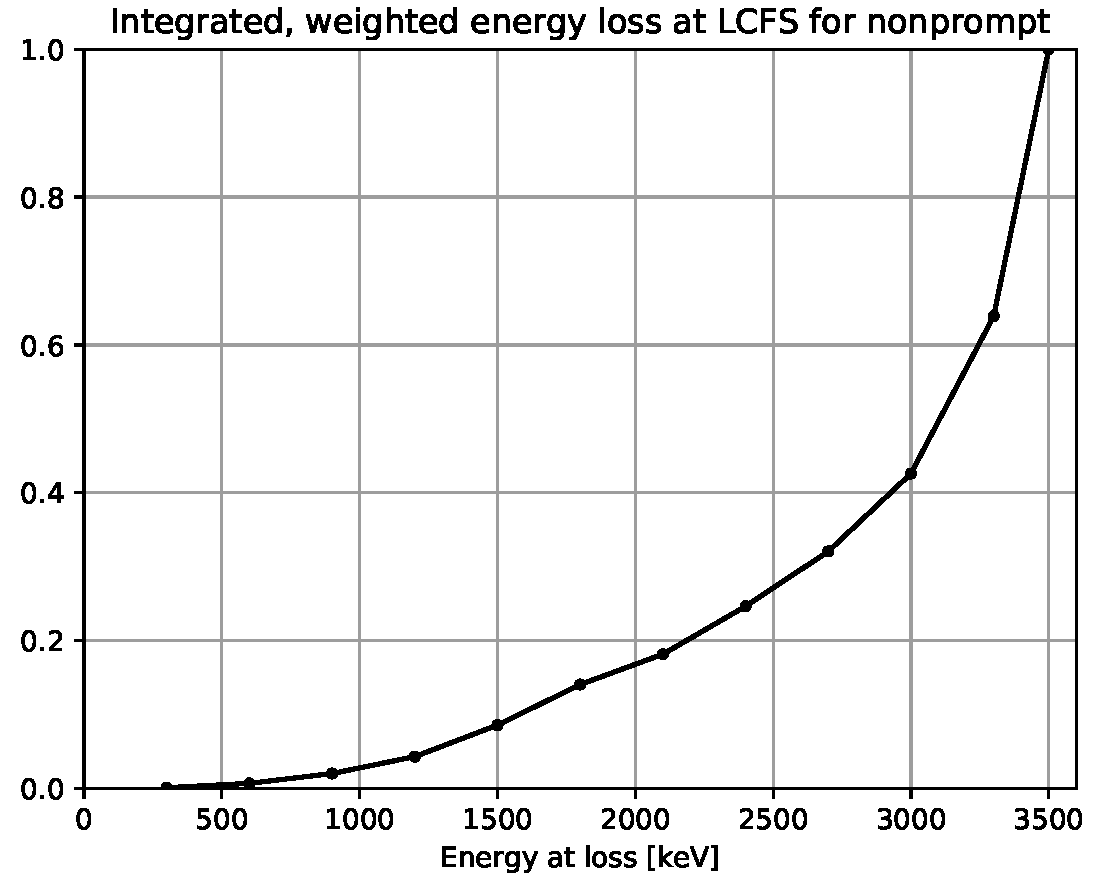
*Figure 12. Histogram of weighted energy loss (keV) at the LCFS for simulation of nonprompt alpha loss. Black: simulation until thermalization (SIMTIME=100 ms). red: SIMTIME=0.5 msec. Blue: SIMTIME=5 ms. runIDs are: 40560057, 40889126, 41439088.*



*Figure 13. Cumulative weighted distribution of nonprompt markers lost to the LCFS. Black = (simtime = 100 ms), red = (simtime = 0.5 ms), and blue = (simtime = 5 ms).*

It remains true that 40% of the markers in the to-thermalization simulation are lost with energies less than 2 MeV and the average energy at time of loss is 2.44 MeV. But of course, the markers that are lost at low energy contribute less power to the wall due to their lower energy.

Figure 14 below quantifies this. It plots the cumulative, weighted energy loss (as a fraction of the total) as a function of the marker energy at time of loss for nonprompt losses. It shows for example that only ~15% of the total nonprompt energy loss at the LCFS is lost with an energy below 2 MeV.

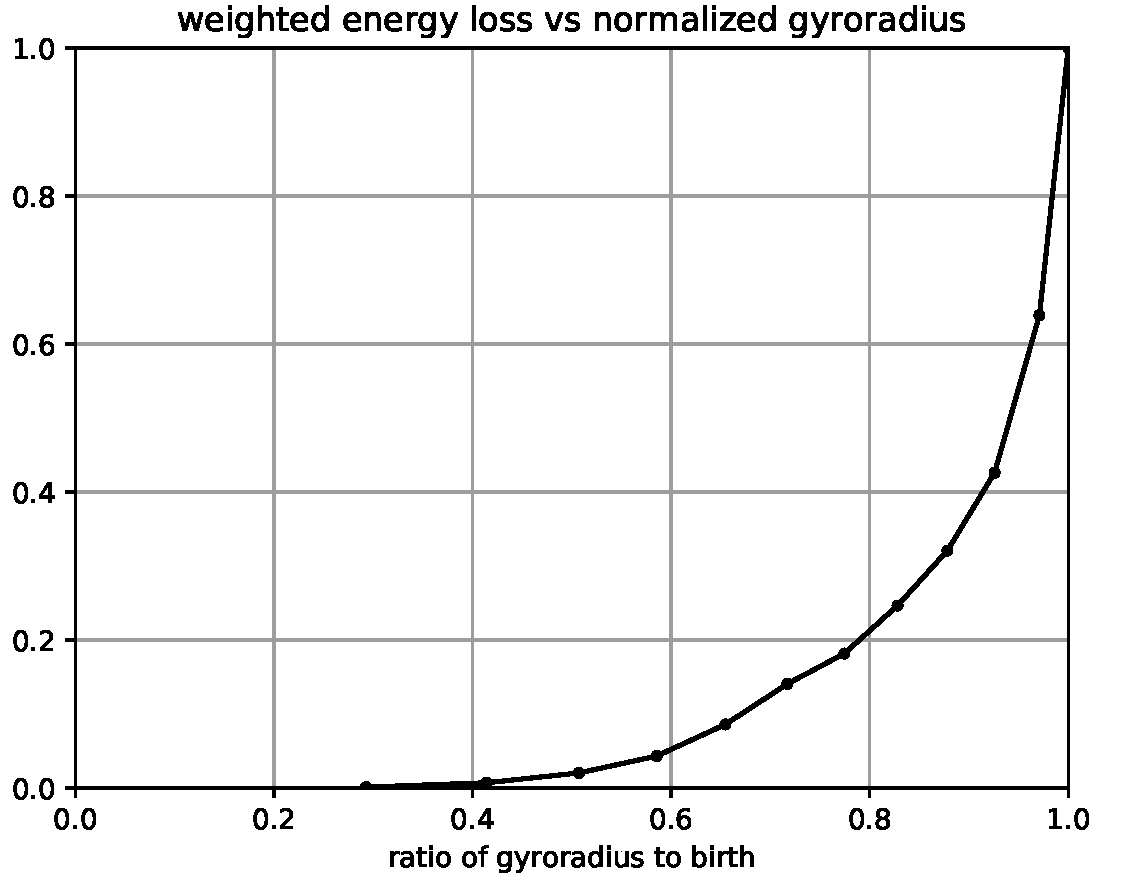


*Figure 14. Cumulative, weighted energy loss (as a fraction of the total) as a function of the marker energy at time of loss for nonprompt losses.*

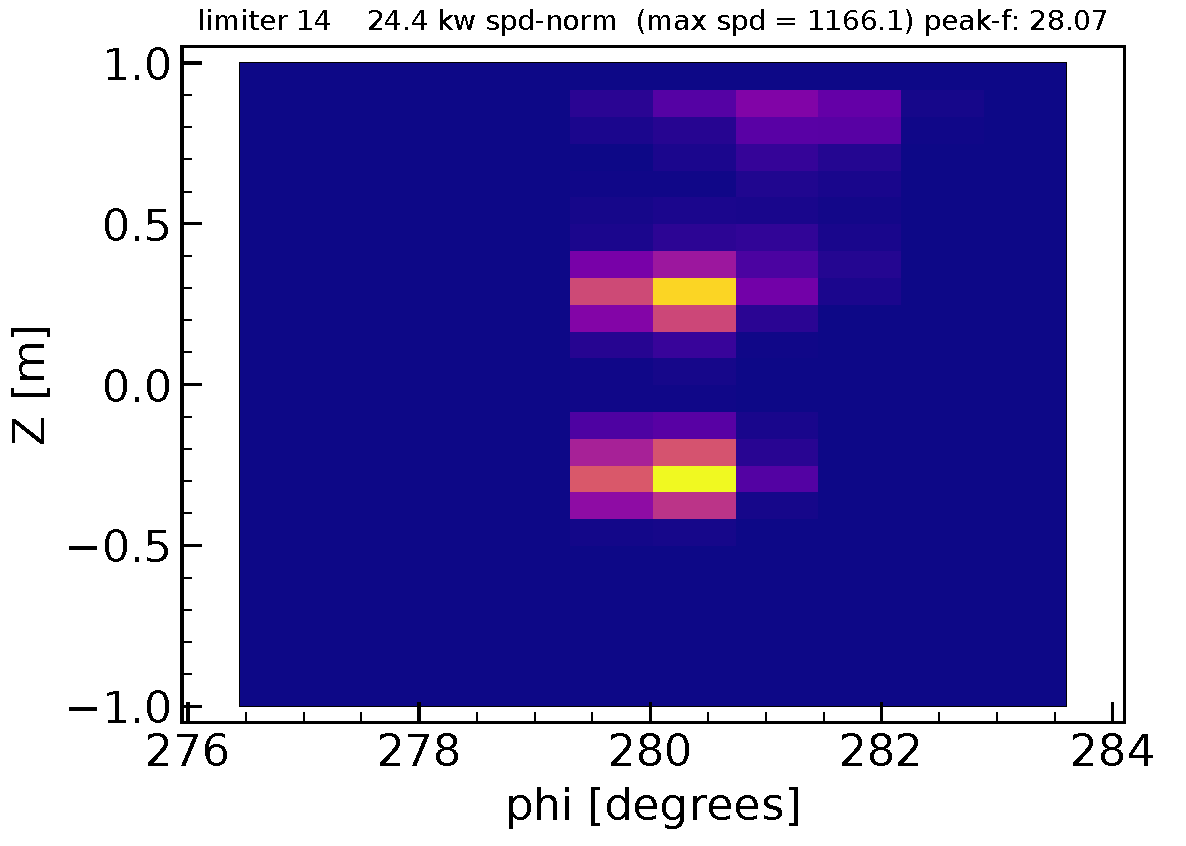
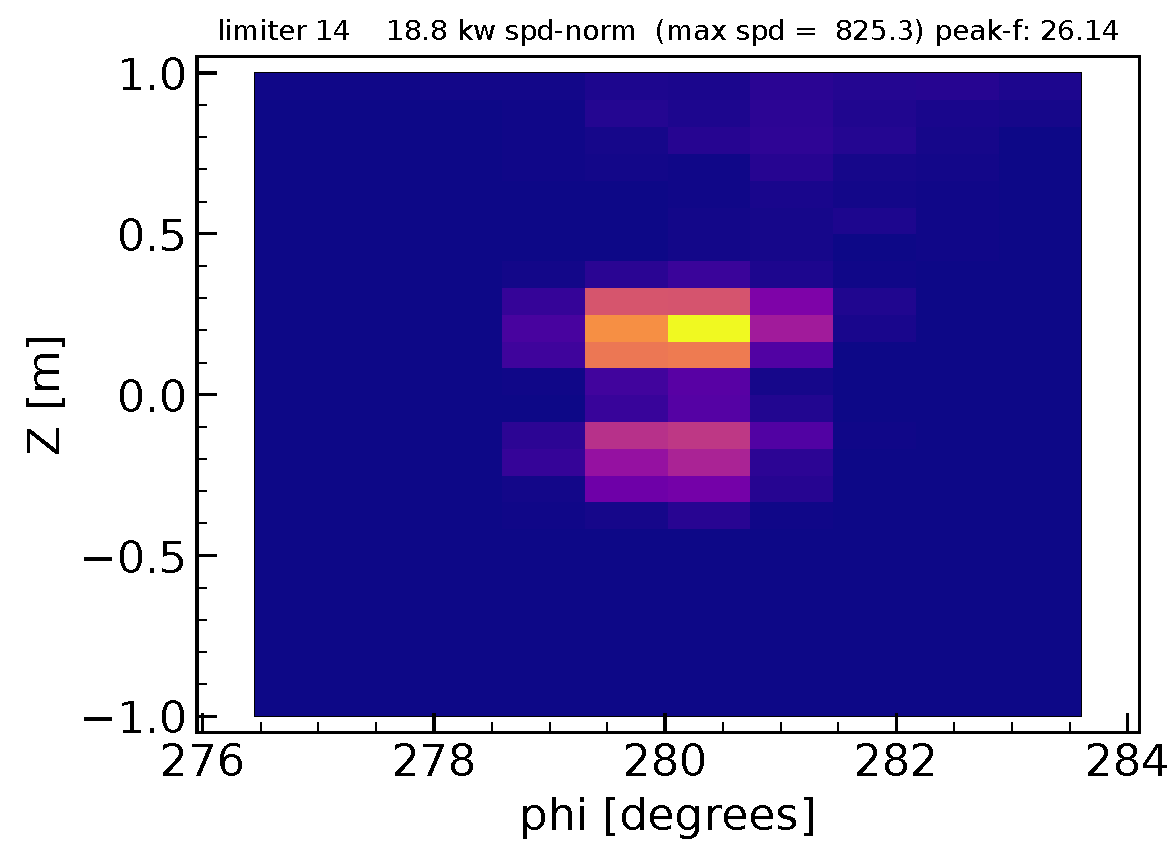
Since we expect the spatial pattern of alpha losses on a 3D wall will be governed by the gyroradius at ‘birth’ at the LCFS, we can replot the data in figure bb above by replacing the x-axis with the gyroradius normalized to the gyroradius at 3500 keV. This is shown in Figure cc below. What Figure 14 shows is that ***only 20% of the total nonprompt loss energy is lost by markers whose gyroradius differs from the gyroradius at birth (3500 keV) by more than 20%***.

On this basis, it seems reasonable to expect that an ASCOT orbit simulation for markers born at LCFS which imposes a maximum simulation time much shorter than the full slowing down time will compute a spatial distribution of power deposited on a candidate 3D wall that is similar to that which would be computed if the markers were followed to full thermalization. This conclusion needs to be checked against actual simulations. Unfortunately, this will require an expensive ST-1 type simulation with enough markers to provide good statistics at the wall.

I ran a ‘daughter’ LCFS 🡪 wall simulation in the standard way (Figure 16 below, left) and in a modified way (right) in which the birth marker-velocities were multiplied by 0.50, thus decreasing the energy at birth by a factor of 4. Qualitatively, there isn’t much difference.



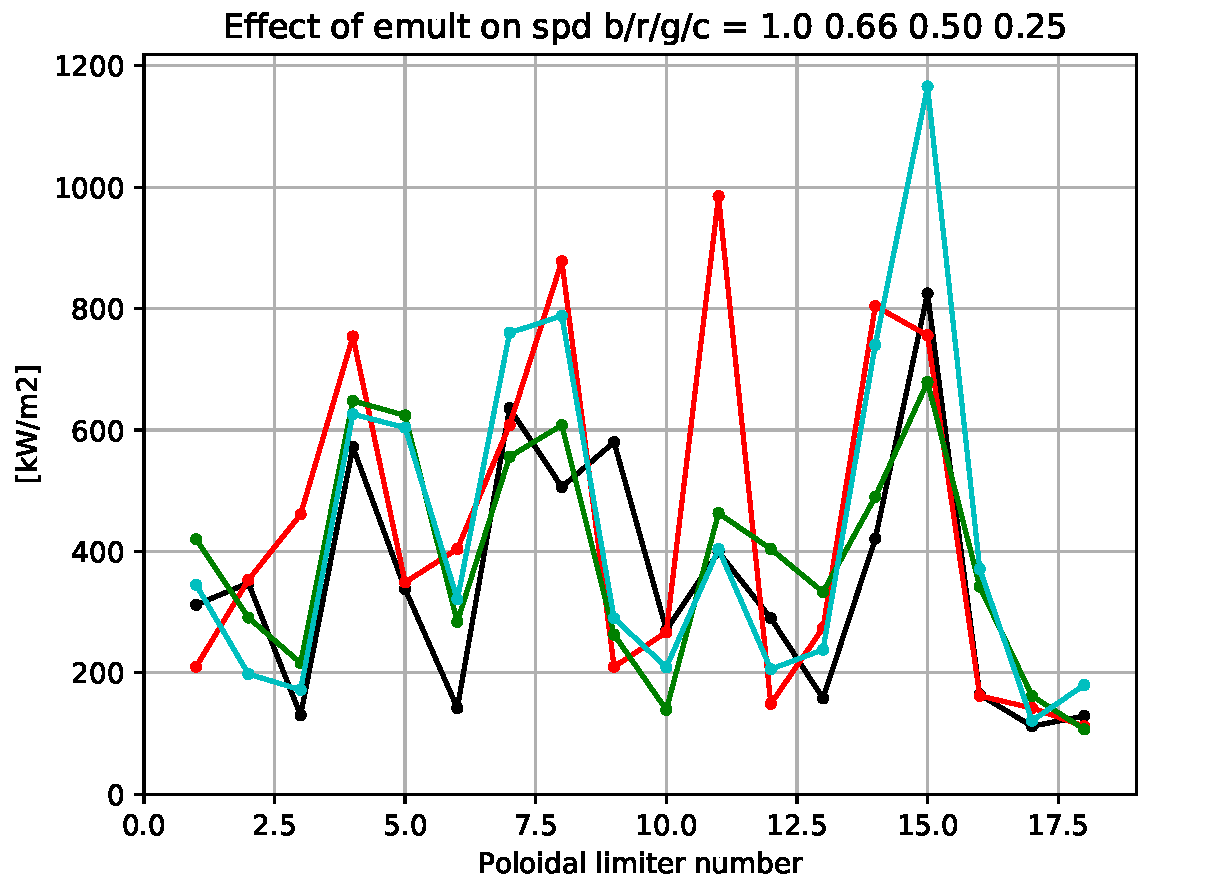
*Figure 15. Cumulative, weighted energy loss (as a fraction of the total) as a function of the ratio of gyroradius at time of loss to gyroradius at 3.5 MeV, for nonprompt losses. Sample interpretation: only 20% of the total lost-alpha energy is lost by alphas whose gyro-radius differs from the birth gyroradius by more than 20%.*



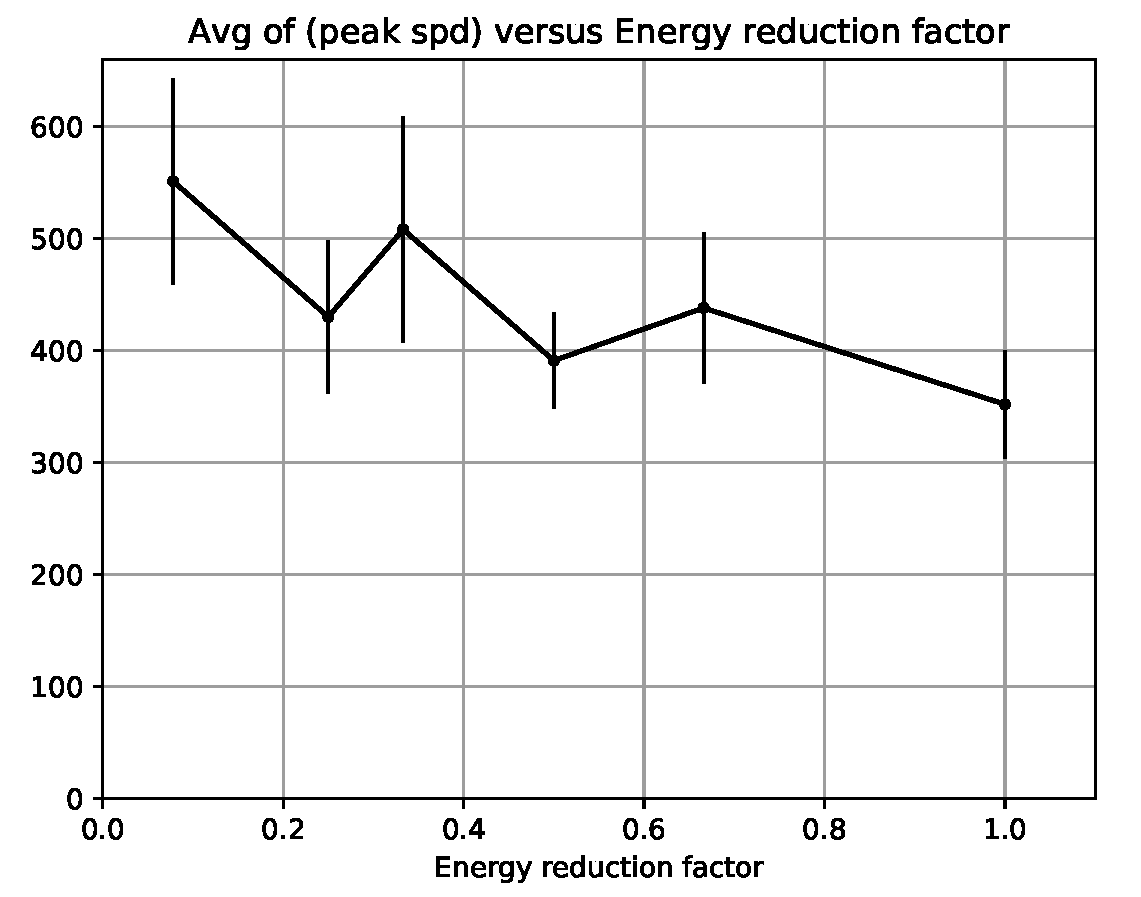
*Figure 16. left: vmult = 1 (i.e. no artificial reduction in birth-marker energy) right: vmult = 0.5 (energy reduction by factor 4).*

I ran a series of simulations in which the ‘birth’ velocities at the LCFS were multiplied by various ( < 1) scale factors. Figure 9 below plots the maximum surface power density as a function of poloidal limiter number. Figure 10 shows that the average of the peak power density for the 18 TF poloidal limiters increases only weakly as we artificially reduce the energy of the birth-marker ensemble (while holding constant the total power deposited on the limiters).

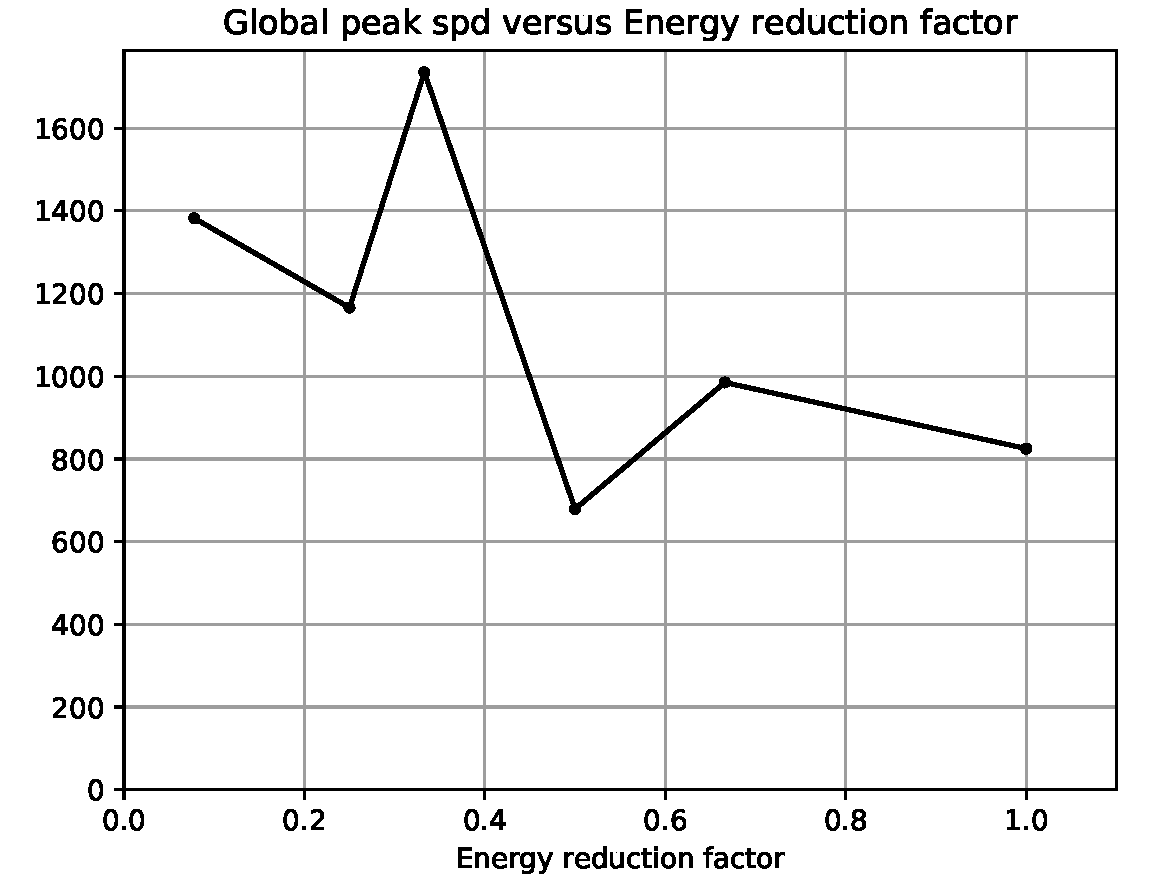
Figure 11 shows that the peak power density (over all poloidal limiters) also increases as the birth marker energy distribution is artificially decreased. There is considerable scatter; in the ‘worst case’ simulation, the peak surface power density doubles from its value at the unperturbed simulation (energy reduction factor = 1.0) when the birth energy distribution is multiplied by 0.33, i.e. to less than 3/5/3 = 1.17 MeV. But from Figure 14, we see that the fraction of alpha energy lost by alphas with energy less than 1.2 MeV is only a few percent. **The conclusion**: somewhat surprisingly, the fact that a ‘short’ duration alpha-orbit simulation will yield an energy distribution of lost alphas at the LCFS that differs significantly from the ‘true’ energy distribution (computed by following the orbits to thermalization or crossing the LCFS), the computed surface power density distribution at the wall should be nearly the same for the two simulations.



*Figure 17. Maximum surface power density for the 18 poloidal limiters. black: marker ensemble taken directly from parent distribution when it crosses the LCFS. Red, green, cyan: multiply the birth marker velocities by 0.816, 0.707, and 0.5 to reduce the birth marker energy by a factor of 0.66, 0.50, 0.25.*



*Figure 18. In a series of ASCOT LCFS 🡪 wall simulations for nonprompt losses (simulation type ST-4), the ‘birth’ marker velocities (i.e. at the end-location at the LCFS of the parent simulation) were multiplied by user-defined scalar values. The peak surface power density is computed for each of the 18 TF coils. This plot shows the average of the 18 peak surface power density as a function of the ersatz energy reduction factor of the birth marker ensemble.*



*Figure 19. Peak surface power density (over all 18 poloidal limiters) as a function of the artificial energy reduction factor of the birth-marker ensemble.*

**Power Lost between LCFS and wall: nonprompt losses ST-9**

Simulation ST-8 (1167/41414086) follows marker orbits of the nonprompt type from ‘birth’ at the LCFS to either thermalization or else hit the wall. It got its birth markers from 1136/40889126 which was a ‘short-duration’ (0.5 msec) simulation. As described above, this approach is not well justified because the computed energy distribution of the markers at t=0.5 msec is much more concentrated than a full slowing-down distribution that would be computed for example in simulation ST-1.

So it occurs to me that we should use the endstate of simulation ST-1 (1095/40560057) as the basis for the ensemble of birth-markers for the simulation that computes the fraction of power that is lost between the LCFS and the wall for the nonprompt losses because it does compute the full to-thermalization orbits. Module define\_prt\_markers\_05 in marker\_sets.py allows the endstate of lost markers of a parent simulation to be copied into the birth marker data for a daughter simulation. I modified this module to allow the user to define a minimum time – markers whose end-time is less than this user-specified value will not be copied into the daughter ensemble of markers.

This is run 1180/41563386. I processed that with process\_ascot.py and got the following values for the losses between the LHCS and the wall:

* raw marker fraction lost to the wall: 65.43%
* weighted marker fraction lost to the wall: 63.98%
* weighted energy lost to the wall: 59.19%

This simulation had 2664 markers at birth, of which 1743 were lost. So although working from the parent simulation from ST-1 yields fewer birth markers than working from ST-4, the relative accuracy should be good enough (1/sqrt(1743) = 2.4%. Note that of the markers that were lost, on average they struck the wall with 91% of the energy which they had at ‘birth’ on the LCFS. The average energy at birth at the LCFS was 2.31 MeV.