**Parent / daughter runs**

Following alpha orbits with SPARC parameters is computationally expensive, e.g. it can take ~8 hours of CPU time to follow an alpha from birth to thermalization. We do three things to minimize the CPU time that is required to compute the alpha losses:

1. One might think that one should launch markers with a density proportional to the alpha source rate. That might be appropriate if we were interested in the alpha distribution function, e.g. for stability calculations. But we’re mostly interested in the losses, and the losses occur mostly for alphas born closer to the edge. So we typically choose to launch markers randomly in space, and then give them a statistical weight proportional to the alpha source rate at their birth location.
2. We do such a run, i.e.truly random birth locations, and then run a simulation that follows the alphas until they thermalize or cross the LCFS. This run identifies a ‘safe’ region, typically rho\_poloidal < 0.8, inside of which the losses are negligible. So why follow alphas in the safe region at all? … they contribute nothing to our calculated surface power density.
3. We then create say 5 ‘parent’ simulations, each with say 20,000 alphas, that launch markers with 0.8 < rho\_poloidal < 1.0, each of which follows the markers until thermalization or until they cross the LCFS. ASCOT records the positions and velocity vectors of the markers as they cross the LCFS.
4. We then use those ending positions / velocity vectors as the starting data for ‘daughter’ simulations, which follow the orbits from the LCFS until they hit an actual wall. I have determined that it is sufficient to follow the orbits for about 10 ms of simulation time.

This system allows one computatationally-expensive ‘parent’ simulation to provide starting data for multiple, computationally cheap ‘daughter’ simulations to evaluate the loss pattern on multiple candidate wall shapes. For typical simulations, the daughter simulations can be a factor 10-20 faster than parent simulations.

This approach is described further in [spd-analysis-history-8.docx](file:///C:\Users\sscott\Documents\ripple\ASCOT_handover\spd-analysis-history.docx).

The file on [NERSC allotments](file:///C:\Users\sscott\Documents\ripple\ASCOT_handover\nersc-allotment-2021-sds.docx) has information about the required CPU time for calculations of the surface power density. The bottom line is that for simulations with 10^5 markers, which seems to be adequate for calculations of the surface power density, we can simulate 20 different scenarios (i.e. 20 ‘parent’ simulations corresponding to 20 different plasma/ripple scenarios) with a yearly allotment of 1.5 x 10^6 CPU hours, which is our allotment for 2021.

We should probably make more use of the ‘low’ priority queue on KNL, which has a charge factor of 0.5, i.e. we would be able to run 40 rather than 20 parent simulations (there is a still-lower ‘flex’ queue with a charge factor of 0.25, but this queue seems to be suitable for code that is check-pointed, and ASCOT5 is not checkpointed). According to Nathan Howard, jobs in the low-priority queue sometimes wait weeks to process, so using it will require some discipline.

The NERSC policies regarding queues (termed ‘quality of service (QOS)) at NERSC is described at <https://docs.nersc.gov/jobs/policy/>