PUBLICATION IX

A study of variance reduction methods in MCNP4C in a slot with a dogleg

12th Biennial Topical Meeting of the Radiation Protection and Shielding Division of the American Nuclear Society. Paper 14. Santa Fe, NM, USA 2002. 11 p.

A STUDY OF VARIANCE REDUCTION METHODS IN MCNP4C IN A SLOT WITH A DOGLEG

Frej Wasastjerna
VTT Processes
P.O. Box 1604
FIN-02044 VTT, FINLAND
+358 9 456 5021
frej.wasastjerna@vtt.fi

SUMMARY

Streaming calculations may be very challenging. This paper studies the performance of various methods of generating weight windows in a particular kind of streaming problem that is common in fusion neutronics work, involving a planar gap or slot with a dogleg. The weight window generator of MCNP4C is found to be useful, even when manual adjustment of the generated weight windows is impracticable. The exponential transform and forced collisions are also studied and found to help a little.

I. INTRODUCTION

The current design of the ITER experimental fusion reactor includes a large number of ports in which various kinds of plugs will be inserted. For technical reasons, there must be a gap of a few centimeters between these plugs and the surrounding port walls. Even though there is a dogleg in this gap, streaming through it is the dominant contributor to the radiation at the end of many ports. Calculating radiation levels at these locations is very difficult and time-consuming, so there is an incentive to study ways of making these calculations as efficient as possible. This paper describes an attempt to study the efficacy of various variance reduction methods in MCNP4C.[1]

A simple two-dimensional model was set up, see Fig. 1.

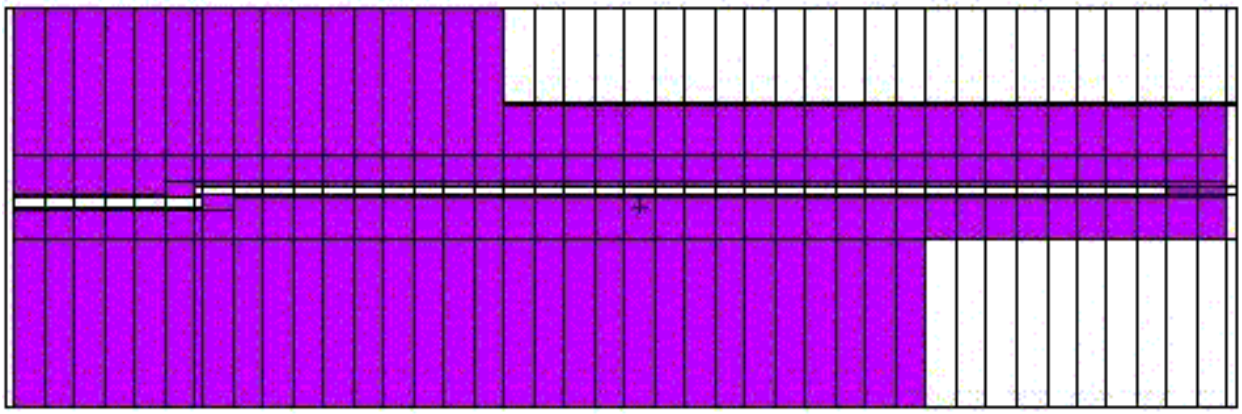

Figure 1. PY cut of geometry.

This is a two-dimensional geometry, homogeneous in the y direction and with reflecting boundary conditions at the y boundaries. In the z direction (vertical in the figure) the extent is 99 cm; in the x direction the material, a mixture of 80% stainless steel and 20% water, extends for 302 cm with a 2

cm void, containing a source of 14 MeV neutrons, on the left and another 2 cm void for tallying purposes on the right. The gap running through the geometry is 2 cm thick, with an offset of 3 cm at the dogleg. In addition to the y boundaries, reflecting boundary conditions were also used at the left x boundary and at the bottom, with vacuum at the right and top.

This geometry, partly based on a model developed by H. Iida, is much easier to calculate than actual ITER port geometries but is nonetheless sufficiently similar to actual equatorial ports to be useful for studies of streaming.

The geometry shown in Fig. 1 has a cell subdivision designed to be used with cell-based weight windows and possibly other variance reduction techniques such as forced collisions. Since Fig. 1 is smeared by the file conversions required to include it in this paper, Fig. 2 is included to show the geometry at the right end in more detail, clarifying the tally cell structure. This figure shows the geometry used in calculations with mesh-based weight windows.

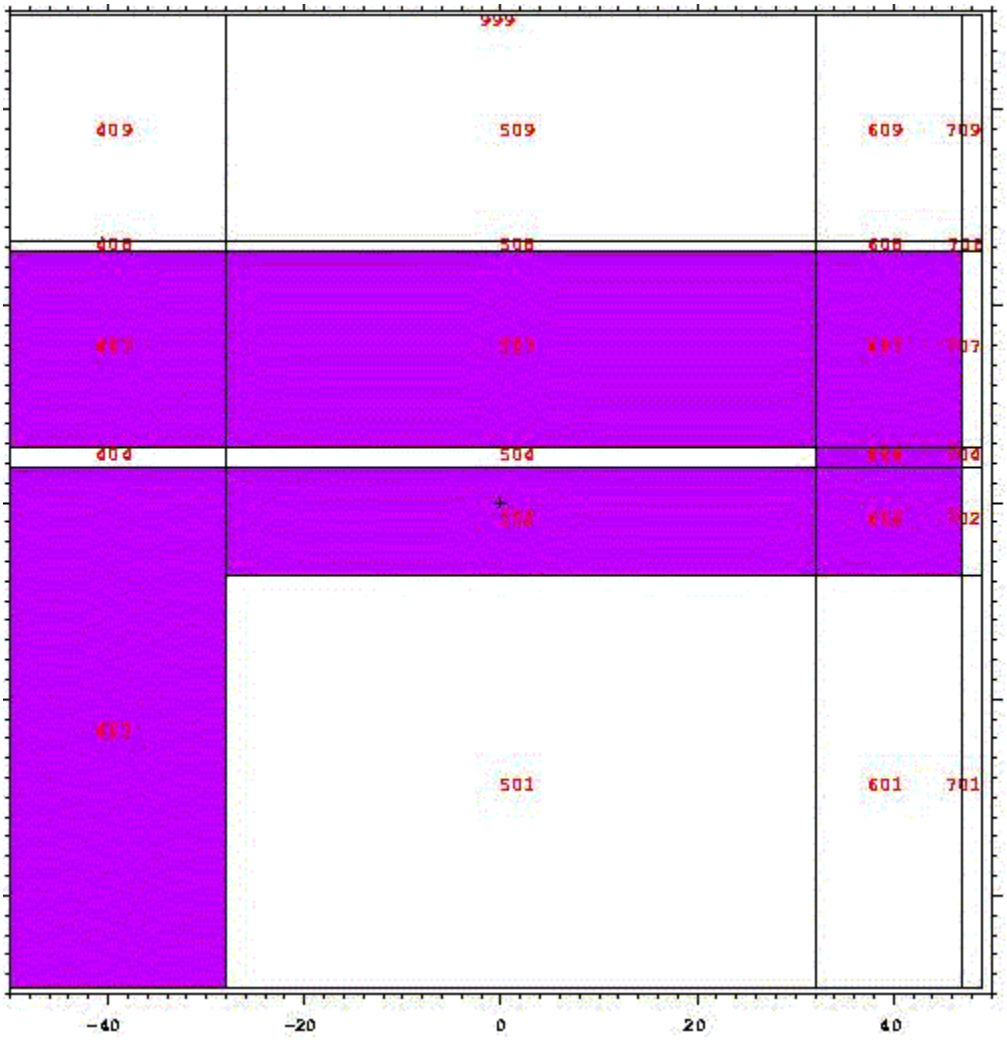

Figure 2. Detail of the right part of the geometry.

Fluxes were tallied in the following locations:

- In the flange at the end of the gap (cell 604 in Fig. 2; this is the thin cell right of cell 504)
- In the cavity at the right end of the plug: cells 501 and 601
- Above the port: cells 408 (which extends all the way to the vacuum vessel, the solid zone at the top in Fig. 1), 508 and 608
- At the right end: cells 701, 702, 704, 707, 708 and 709

Since the critical quantity driving the shield design for ITER is the gamma dose rate 1,000,000 seconds after shutdown, the ultimate goal in the calculations here was to calculate the gamma flux. In this work it was the prompt gamma flux, but presumably the performance of the variance reduction methods used would be the same in calculations on delayed gammas, regardless of whether the "direct 1-step method" of Iida et al.[2,3] or the "rigorous 2-step method" of Fischer et al.[4] is used.

Therefore we were primarily interested in the precision of the gamma flux, but since this requires quite lengthy calculations and is primarily determined by the neutron flux above 1 MeV, we also paid attention to the fast neutron flux, which is much easier to calculate.

The main interest here is concentrated on estimating appropriate weight windows. The use of the exponential transform and forced collisions is also explored, but this is shown to give only minor benefits.

In most cases we first tried to establish weight windows for fast neutrons and then to extend these to slower neutrons and photons. The alternative method of starting directly with coupled n,p calculations was also tried. Both cell-based and mesh-based weight windows were used, though with more emphasis on cell-based methods.

The weight window generator of MCNP4C was used heavily in these calculations. The user's manual stresses that the generated weight windows should not be used uncritically, since they are stochastic in nature and often based on poor statistics, but should be examined and corrected. This advice is well justified, but unfortunately it is often impracticable. In a calculation with many thousands of cells, examining and adjusting the weight windows requires many days of work, which may moreover have to be repeated, since weight windows may have to be generated in many steps. Thus it may be necessary to rely on unadjusted weight windows. For that reason the weight windows were used without manual adjustments in these calculations.

II. CELL-BASED WEIGHT WINDOWS, FAST NEUTRONS

The first calculations were performed in the geometry shown in Fig. 1, where the material zones were subdivided by splitting planes at intervals of about 7.5 cm, in the expectation that this would be the approximate halving length of the fast neutron flux. First we tried to estimate weight windows in a single step, below called the "direct" method, choosing starting importances on the basis of engineering judgment and putting the tally cell for the weight generator in or near the part of the geometry where the tallies of interest are located. We also tried the alternative "progressive" method of starting with constant importances and first generating weight windows for a tally cell fairly near the source, then using these to reach a tally cell slightly farther on and so on until the region of interest is reached.

A. Direct Weight Window Generation

In the direct method, the starting importances were chosen with the objective of keeping the track density approximately constant with increasing distance from the source. Near the source this involved a doubling of the importance for each cell layer. For use farther from the source, where streaming becomes dominant, we also calculated importances inversely proportional to the solid angle subtended by either the entrance to the gap or the entrance to the part of it lying beyond the dogleg. For each x interval, the smaller of these importances was used, rounding downward to the nearest power of 2. These importances were then modified in the z direction to emphasize streaming along the gap, then to drive the neutrons out towards the tally cells, but these modifications were rather modest.

Based on this calculation of the fast neutron flux, weight windows were generated either for a tally cell located at the end of the gap (the left half of cell 604 in Fig. 2) or for a composite cell comprising all the **other** tally cells. The resulting weight windows were then used in further calculations. The results are given in Table 1, where the original calculation is denoted a0, the calculation with weight windows based on the left half of cell 604 in Fig. 2 is called a1 and the one with weight windows derived for the composite cell is a2.

Since the real quantity of interest is the fractional standard deviation **(fsd)** and the figure of merit **(fom)**, the actual fluxes have been omitted to avoid cluttering up the tables. The outcomes of the numerous statistical tests included in MCNP4C have also not been included, though they were studied. The conclusions from them were broadly consistent with those that can be drawn from the fractional standard deviations. The figures of merit

shown are in each case for the first tally cell in the mentioned part of the geometry, i.e., for cells 604, 501, 408 and 701.

All cases shown in Table 1 involved 20,000,000 histories.

Table 1. Fractional Standard Deviations and Figures of Merit for Fast Neutron Calculations, Direct Method; 20,000,000 Histories

	case	a0	a1	a2
	cell	**fsd**	**fsd**	**fsd**
flange	604	0.0215	0.0160	0.0159
cavity	501	0.0182	0.0273	0.0152
	601	0.0187	0.0199	0.0148
above port	408	0.0115	**0.6505**	**0.0158**
	508	0.0371	0.0912	**0.0478**
	608	0.0350	0.0387	**0.0381**
at end of pt	701	0.0189	0.0203	0.0149
	702	0.0228	0.0189	0.0181
	704	0.0257	0.0221	0.0217
	707	0.0232	0.0190	0.0183
	708	0.0413	0.0724	**0.0502**
	709	0.0181	0.0486	**0.0215**
		fom	**fom**	**fom**
flange		10.7	60.6	59.2
cavity		14.9	20.9	64.5
above port		37.2	0.04	59.7
end of port		13.9	37.6	67.3
time, min		202.07	64.44	66.75

We can see that case 1, based on weight windows calculated for cell 604 at the end of the gap, performs very poorly in cell 408 above the port, giving an fsd above 0.5 (highlighted in red). This is not surprising, since this choice of a target cell is intended to give good results for cells near the end of the port, while cell 408 is close to the root of the port.

However, even for the other tally cells the weight windows based on the composite target cell perform better. It is notable that in some cases, shown in boldface, the fsd is actually worse than with the original importances, but the calculation runs substantially faster, leading to superior figures of merit. The reason for this is illustrated by Fig. 3, which shows the fast neutron population and tracks entering for successive layers, perpendicular to the x axis, of the geometry.

In the central part of the system the population is much smaller with the weight windows (case a2n) than with the original, manually estimated, importances (a0n). This means that less time is spent on tracking neutrons in this region.

Note that in case a0 the importances were based on the advice in the manual: “A good rule is to keep the population of tracks traveling in the desired direction more or less constant”, and, in fact, they do a reasonably good job of this, considering that the total reduction of the flux from the source cell to the tally cells (except cell 604) is about 7 to 8 orders of magnitude. The weight windows do much worse at keeping the population constant, but that doesn’t seem to do any harm.

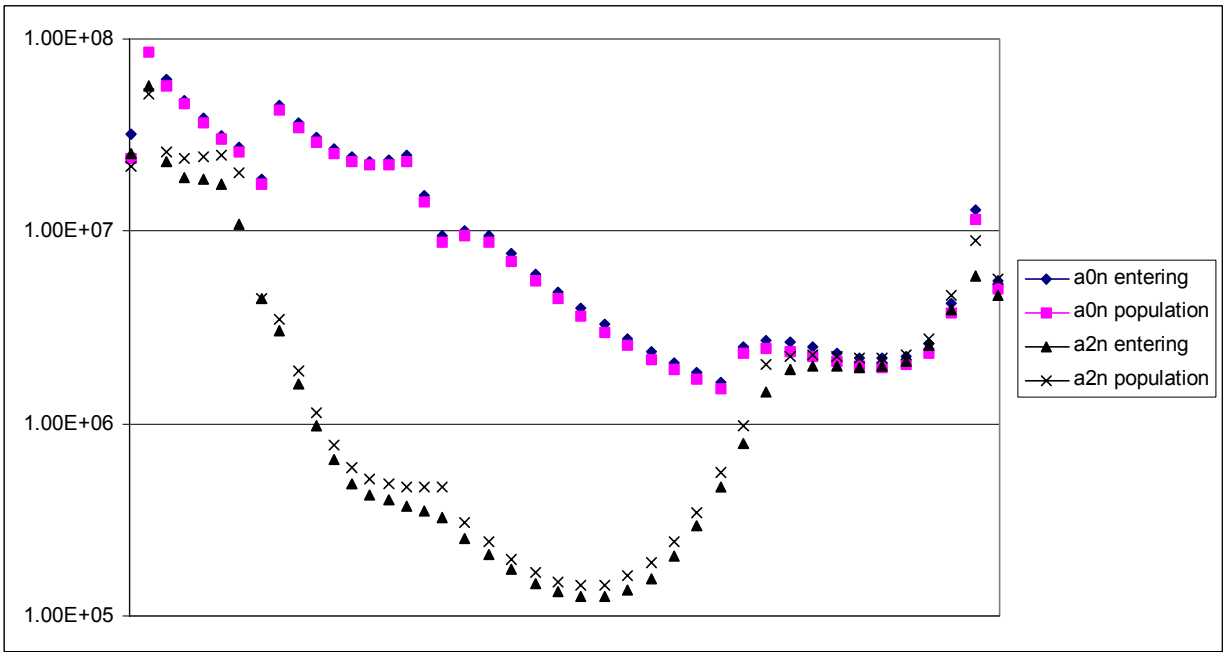

Figure 3. The neutron population with manually chosen importances (a0n) and with generated weight windows (a2n). Horizontal axis = x coordinate axis.

As one can guess from Fig. 3, the weight windows in the central region are higher than what would correspond to the original importances. It is also worth noting, though it cannot be seen from Fig. 3, that the variation in the z direction of the weight windows is substantially greater than that of the importances. Apparently the importances were underbiased in this respect, allowing too many neutrons far from the gap.

It should be emphasized that the use of a composite target cell of large extent for the weight window generator involves an obvious danger. If particles reach one part of that cell much more easily than other parts, the total score in the composite cell and thus the generated weight windows will be dominated by the easily reached part, and subsequent calculations may not sample the other parts adequately. In this case, however, the composite cell worked satisfactorily in spite of its size, presumably because all its parts are approximately equally difficult to reach, as shown by the fact that the calculated fluxes were not drastically different in different parts of it. Specifically, the highest flux, in cell 704, was about 100 times the lowest flux, and since 704 is a small cell, the integral score in it did not dominate the total score to an unhealthy degree.

The use of 20,000,000 histories made it possible to get fairly good statistics, which should make the weight window generator work well. This does not resemble the real situation for the ITER port analysis, in which the much larger and more complex geometry makes it impossible to get so good statistics with runs of reasonable length, at least before very good weight windows are available. For this reason we next performed calculations similar to a0 and a2 but with only 200,000 histories, calling them A0 and A2. The results are shown in Table 2.

Table 2. Fractional Standard Deviations and Figures of Merit for Fast Neutron Calculations, Direct Method; 200,000 Histories

	case	A0	A2
	cell	**fsd**	**fsd**
flange	604	0.2016	0.1728
cavity	501	0.1754	0.1294
	601	0.1901	0.1485
above port	408	0.0989	0.1296
	508	0.3285	**0.6768**
	608	0.2511	**0.5291**
end of port	701	0.1858	0.1471
	702	0.2158	0.1848
	704	0.2367	0.2106
	707	0.2207	0.1895
	708	0.2521	**0.6362**
	709	0.1968	0.2009
		fom	**fom**
flange		11.9	40.1
cavity		15.7	71.5
above port		49.4	71.3
end of port		14.0	55.4
time, min		2.07	0.83

Here we get large fsd's, similar to those from real ITER port calculations. In fact three of the fsd's obtained using weight windows, highlighted in red, exceed 0.5, meaning that the results are garbage, according to Table 2.3 of the manual. The trend is nonetheless similar to the a series: The use of weight windows doesn't bring any major improvement in the fsd's for a given nps (number of histories), but it speeds up the calculation, making larger nps values feasible. In this case perhaps a further iteration of the weight windows, with a longer run, or manual corrections would be advisable before one tries to get final results.

B. Progressive Weight Window Generation

An alternative way of obtaining weight windows was as follows: First find weight windows to steer neutrons to a cell near the source but in the streaming path. Then, using these weight windows, calculate weight windows for a target cell somewhat further on, and repeat until the final tally cells are reached. In this case the first target cell was set at the bottom of the dogleg, the second at the top of the dogleg, the third halfway down the post-dogleg section of the gap, and the fourth at the end of the gap, in cell 604. Finally weight windows were calculated for the composite tally cell as above.

This was done with 200,000 histories per run. The first steps worked well, with run times of less than a minute and fsd's in the target cells below 0.2.

The final step, from cell 604 to the composite tally cell, worked less well. In this run the fsd in the target was 0.4484. This was not surprising, considering the problems in going from cell 604 to, especially, cell 408 at the root of the port. However, iterating the weight window generation process gave gradually improved results, as shown in Table 3.

Table 3. Fractional Standard Deviations and Figures of Merit for Fast Neutron Calculations, Progressive Method; 200,000 Histories (B5 and B6) or 20,000,000 Histories (b7)

	case	B5	B6	b7
	cell	**fsd**	**fsd**	**fsd**
flange	604	0.1453	0.1520	0.0160
cavity	501	0.3988	0.1405	0.0156
	601	0.2230	0.1460	0.0155
above port	408	0.2456	0.2704	0.0442
	508	0.3466	0.2442	0.0906
	608	0.3500	0.2110	0.0612
end of port	701	0.2244	0.1460	0.0156
	702	0.1599	0.1558	0.0178
	704	0.1930	0.2028	0.0222
	707	0.1686	0.1740	0.0185
	708	0.4549	0.3071	0.0693
	709	0.4561	0.2592	0.0555
		fom	**fom**	**fom**
flange		62.1	56.0	49.4
cavity		8.2	65.5	51.5
above port		21.7	17.7	6.5
end of port		26.0	60.7	52.1
time, min		0.76	0.77	79.31

Cases B5 and B6 are successive iterations with 200,000 histories. There is at least a slight improvement in most tally bins, but the fsd's remain rather high. Finally a run with 20,000,000 histories, b7, was made to get better results. This hope was fulfilled in the sense that the fsd's were low, though no further improvement in the fom's was seen. There was, in fact, some apparent deterioration, but this probably merely reflects more realistic fsd's based on the larger number of histories.

The fsd's of cases A2 and B6 or a2 and b7 are broadly similar, suggesting that the direct and progressive methods of finding weight windows give comparable results. The progressive method requires a greater number of calculations, but at

least in this test case the calculations were fast, so their greater number was not a major problem. The direct method requires more work to find realistic starting importances.

The figures of merit were slightly better for a2 than for b7, indicating that the direct method may have a small advantage in performance. However, this is probably only the case if the starting importances are fairly good.

Our recommendation is that the direct method of weight window generation may be appropriate for users capable of choosing good importances manually. For users without sufficient experience to do this, the progressive method is probably better.

III. CELL-BASED WEIGHT WINDOWS, GAMMAS

The "a" series calculations (direct weight window generation, 20,000,000 histories per run) was continued with calculations for slow neutrons and photons. In run a3 the generated weight windows for neutrons above 1 MeV were extended manually to neutrons between 1 MeV and 1 eV, below 1 eV and to photons in such a way that, the lower the energy of a neutron and the farther from the tally cells it was, the higher was the ratio of the lower bound of the weight window to that for fast neutrons. Photons far from the tally cells were killed. Using these weight windows, new ones were generated and then used in run a4. The results are shown in Table 4.

Table 4. Fractional Standard Deviations and Figures of Merit for Photon Calculations with Initial Weight Windows Established by Fast Neutron Calculations; 20,000,000 Histories

	case	a3	a4
	cell	**fsd**	**fsd**
flange	604	0.0106	0.0144
cavity	501	0.0084	0.0108
	601	0.0085	0.0111
above port	408	0.0078	0.0170
	508	0.0113	0.0296
	608	0.0115	0.0260
end of port	701	0.0084	0.0111
	702	0.0095	0.0134
	704	0.0106	0.0158
	707	0.0099	0.0145
	708	0.0171	0.0484
	709	0.0091	0.0202
		fom	**fom**
flange		5.3	8.7
cavity		8.3	15.5
above port		9.7	6.2
end of port		8.4	14.7
time, min		1685.17	554.17

As expected, these calculations were considerably slower than those involving only fast neutrons. It is also interesting to note that the weight windows produced by the weight window generator for intermediate and thermal neutrons and for photons gave fsd's inferior to those obtained with weight windows estimated manually. However, the generated weight windows gave a much faster run, so that the figures of merit improved. Thus, the deterioration in fsd's for a given number of histories can be more than compensated by increasing the number of histories treated in a given time.

One can also start directly with a coupled neutron-photon calculation. We tried this option in run D0, and the resulting weight windows were then used in a new run, D2. The results are shown below. Note that only 200,000 histories were run in D2, compared with 20,000,000 in a4, so to take this into account, the fsd's for a4 have been multiplied by 10. With this adjustment they are similar for both cases, as are the figures of merit. Note that the run times have not been adjusted, and they differ by the expected factor of 100. We can thus conclude that in this test problem going straight to copled n,p is a workable option. However, it should be noted the run time for D0, the run in which the weight windows were generated, was 60.94 minutes, a long time for a run of 200,000 histories

in this simple test problem. Therefore this method might not be practicable in more complicated problems.

Table 5. Comparison of Results in Photon Calculations Obtained by Starting with Fast Neutron Calculations (a4) and by Going Directly to Coupled Calculations (D2); 20,000,000 and 200,000 Histories, Respectively

	case	a4	D2
	cell	**fsd*10**	**fsd**
flange	604	0.144	0.165
cavity	501	0.108	0.115
	601	0.111	0.118
above port	408	0.170	0.132
	508	0.296	0.252
	608	0.260	0.244
end of port	701	0.111	0.118
	702	0.134	0.166
	704	0.158	0.189
	707	0.145	0.172
	708	0.484	0.365
	709	0.202	0.197
		fom	**fom**
flange		8.7	6.59
cavity		15.5	13.6
above port		6.2	10.3
end of port		14.7	12.9
time, min		554.17	5.57

IV. MESH-BASED WEIGHT WINDOWS

We also tried the mesh-based weight window generator included in MCNP4C. The mesh used was quite similar to the cell subdivision used with the cell-based weight window generator, with the mesh boundaries coinciding with material boundaries in many cases, otherwise with mesh intervals usually amounting to 7.5 cm in the x direction and with thin meshes just above and below the gap.

Successive iterations of the weight window calculation were carried out, much like we did with the progressive cell-based weight window generator, though in this case the target for the generator was always the composite tally cell.

Five fast neutron calculations with 200,000 histories each were carried out, replacing the weight windows after each calculation. The run times showed a curious variation. Expressed in minutes, they were: 0.44, 0.83, 16.58, 0.46, 0.47. It appears as if the third run, using the second set of generated weight windows, must have used much heavier splitting. This did have the effect of reducing the fsd's, which were extremely high in the first two runs (**every** fsd exceeded 0.5!) to much more reasonable values (around 0.2). Apparently this enabled the program to produce much better weight windows, which then gave similar fsd's with much shorter run times.

As far as could be deduced from the results, the process had converged after the fourth run. The fifth was repeated with 20,000,000 histories. Table 6 compares this run, c4, with selected results from the calculations described above.

Table 6. Comparison of Fast Neutron Calculations with Cell-Based (a2, b7) and Mesh-Based (c4) Weight Windows; 20,000,000 Histories

	case	A2	b7	c4
	cell	**fsd**	**fsd**	**fsd**
flange	604	0.0159	0.0160	0.0228
cavity	501	0.0152	0.0156	0.0222
	601	0.0148	0.0155	0.0197
above port	408	0.0158	0.0442	0.0373
	508	0.0478	0.0906	0.0623
	608	0.0381	0.0612	0.0393
end of port	701	0.0149	0.0156	0.0198
	702	0.0181	0.0178	0.0276
	704	0.0217	0.0222	0.0357
	707	0.0183	0.0185	0.0248
	708	0.0502	0.0693	0.0420
	709	0.0215	0.0555	0.0364
		fom	**fom**	**fom**
flange		59.2	49.4	42.5
cavity		64.5	51.5	45.2
above port		59.7	6.5	15.9
end of port		67.3	52.1	56.7
time, min		66.75	79.31	45.07

Thus, for fast neutrons, the performance of the mesh-based weight window generator appears roughly comparable with that of the cell-based one.

Next we attempted to extend the calculation to photons. These were initially simply assigned the same importances as the neutrons had in the starting run, while for neutrons, regardless of energy, the weight windows generated in the preceding run were used. The weight window generator was then set to generate weight windows for neutrons in the same 3-group energy structure as before and for gammas.

200,000 histories were treated in run C5, giving reasonable-looking results in 12.65 minutes. The

ensuing weight windows were then used in the next run. This one, C6, took 865.55 minutes without any significant improvement in the fsd's, so the fom's dropped drastically. The run after that, C7, took 552.52 minutes, again failing to improve the precision. In fact, the gamma fluxes calculated in these runs seemed, if anything, to move farther from those obtained previously.

In Table 7 we compare results obtained previously for the gamma flux with those obtained in run C7. Note that run a4 involved 20,000,000 histories as compared with 200,000 for C7. For this reason we have multiplied the fsd's from run a4 by 10 to make them comparable with those for C7, and with this adjustment they are roughly similar. The run times given in the table, however, contain no such adjustment. Run a4 covered 100 times as many histories as C7 in almost the same time, which shows up in the figures of merit.

Table 7. Comparison of Coupled Calculations with Cell-Based (a4) and Mesh-Based (C7) Weight Windows; 20,000,000 and 200,000 Histories, Respectively

	case	a4	C7
	cell	**fsd*10**	**fsd**
flange	604	0.144	0.399
cavity	501	0.108	0.120
	601	0.111	0.155
above port	408	0.170	0.154
	508	0.296	0.120
	608	0.260	0.165
end of port	701	0.111	0.168
	702	0.134	0.285
	704	0.158	0.458
	707	0.145	0.584
	708	0.484	0.198
	709	0.202	0.142
		fom	**fom**
flange		8.7	0.011
cavity		15.5	0.126
above port		6.2	0.077
end of port		14.7	0.064
time, min		554.17	552.52

Thus, while the mesh-based weight window generator worked fairly well for fast neutrons, its results for gammas were disappointing. However, it is not clear how much significance, if any, should be attached to this finding. It is possible that a few further iterations would have given much improved results or that we made some mistake in using the mesh-based weight window generator. Perhaps we should have provided better starting weight windows for intermediate and thermal neutrons and for gammas. Unfortunately schedule and funding limits precluded further work on this problem.

V. EXPONENTIAL TRANSFORM, FORCED COLLISIONS

We also studied the effect of using the exponential transform or forced collisions.

The exponential transform was applied to both neutrons and photons. The stretching was performed diagonally to the right and towards the gap with a stretching parameter of 0.7, except when some tally cell was nearer than the gap. In the latter case the stretching was usually directed towards the tally cell. When two or more directions were appropriate, an intermediate direction was chosen and the stretching parameter was reduced.

Table 8. Comparison of Coupled Calculations Without (a4) and With (a4x) Exponential Transform; 20,000,000 Histories

	case	a4	a4x
	cell	**fsd**	**fsd**
flange	604	0.0144	0.0117
cavity	501	0.0108	0.0088
	601	0.0111	0.0089
above port	408	0.0170	0.0169
	508	0.0296	0.0275
	608	0.0260	0.0229
end of port	701	0.0111	0.0089
	702	0.0134	0.0107
	704	0.0158	0.0128
	707	0.0145	0.0117
	708	0.0484	0.0496
	709	0.0202	0.0195
		fom	**fom**
flange		8.7	12.8
cavity		15.5	22.8
above port		6.2	6.2
end of port		14.7	22.2
time, min		554.17	566.43

We see that there is a modest improvement.

In the study of the effects of forced collisions, we limited ourselves to fast neutron calculations, since the main purpose of forced collisions in this problem would be to increase the sampling of fast neutron streaming along the gap. We used forced collisions in the cells just above and below the gap, with both positive and negative values for the forced collision parameter and with three absolute values for it: 0.5, 0.7 and 0.9. We preferred not to

use a value of 1, because particles crossing the gap back and forth might result in excessive multiplication of tracks.

The forced collision option would be much more useful if there were some way of biasing the flight direction of neutrons emerging from the forced collisions, in this case to make them move preferentially along the gap to the right. Unfortunately no such facility exists in MCNP4C (presumably because this would be impractical, involving excessively time-consuming calculations), and even combining forced collisions with the exponential transform is not allowed. The use of positive values for the forced collision parameter didn't work either. For all positive values we sooner or later got an error stop with the message "the weight of the current particle is zero or less". We also got this error in a combined neutron-photon run with a forced collision parameter of –0.9, though in this case more than 4,000,000 histories were followed before this happened, about twice as many as for the values +0.7 or +0.9. We didn't have time to investigate the reason for this trouble.

The fast neutron calculations with negative values for the forced collision parameter ran without trouble, and their fsd's and fom's are compared below with those for a run without forced collisions.

Table 9. Effects of Forced Collisions on Fast Neutron Calculations; 20,000,000 Histories

	case	a2	a2m5	a2m7	a2m9
		no fcl	fcl=-0.5	fcl=-0.7	fcl=-0.9
	cell	**fsd**	**fsd**	**fsd**	**fsd**
flange	604	0.0159	0.0138	0.0130	0.0127
cavity	501	0.0152	0.0133	0.0121	0.0121
	601	0.0148	0.0131	0.0123	0.0122
above port	408	0.0158	0.0161	0.0153	0.0150
	508	0.0478	0.0481	0.0448	0.0420
	608	0.0381	0.0398	0.0392	0.0395
end of port	701	0.0149	0.0131	0.0123	0.0123
	702	0.0181	0.0152	0.0148	0.0145
	704	0.0217	0.0179	0.0174	0.0171
	707	0.0183	0.0157	0.0148	0.0143
	708	0.0502	0.0514	0.0510	0.0485
	709	0.0215	0.0202	0.0200	0.0203
		fom	**fom**	**fom**	**fom**
flange		59.2	76.7	81.0	81.0
cavity		64.5	82.3	92.3	89.2
above port		59.7	56.0	58.2	58.3
end of port		67.3	85.0	89.7	86.0
time, min		66.75	68.95	73.55	76.5

Again we find a modest improvement, greater for the value –0.7 than for the other two parameter values, but suggesting that the optimum may be close to –0.8.

Note that the use of forced collisions does not give any improvement in the tallies above the port. This is easy to understand, since the purpose of the use of forced collisions was to enhance streaming along the gap, but the particles scoring above the port are mainly such which left the gap early.

VI. CONCLUSIONS

We studied various methods of reducing the variance in MCNP4C calculations in a shielding problem containing a planar gap with a dogleg. The main emphasis was on the use of the weight window generator, but the exponential transform and forced collisions were also studied.

The main conclusion is that the weight window generator did turn out to be helpful, even in cases where the stochastic uncertainty of the resulting weight windows is large and manual adjustments

are impracticable. The two methods of generating cell-based weight windows for fast neutrons, the direct method of providing the best starting importances that our engineering judgment could provide, then setting the target for the weight window generator in the final tally cells themselves, and the progressive method of starting with simple (constant) importances and gradually putting the target farther and farther from the source, performed about equally well. We recommend the progressive method for inexperienced Monte Carlo users, while the direct method may be worthwhile for experienced practitioners.

Cell-based weight windows also performed reasonably well in the coupled neutron-photon calculations, whether they were derived starting with fast neutron calculations or going straight to the coupled calculations. We do, however, believe that the latter method may be impracticable in the more complicated problems encountered in real fusion neutronics calculations.

The mesh-based weight windows also worked satisfactorily for fast neutrons. They worked much less well for coupled calculations, but, as pointed out above, it is doubtful whether one can draw any conclusions about this without further study.

The exponential transform and forced collisions both brought a modest benefit, of the order of a 50 % improvement in the figures of merit.

---

[1] J.F. Briesmeister, Ed.: MCNP™—A General Monte Carlo N-Particle Transport Code, Version 4C. LA-13709-M (2000).

[2] H.Iida, D. Valenza, R. Plenteda, R. T. Santoro and J. Dietz "Radiation Shielding for ITER to allow for Hands-on Maintenance inside the Cryostat", J. of Nuclear Science and Technology, 19.Oct.1999

[3] D. Valenza, H.Iida, R. Plenteda and R. T. Santoro "Proposal of shutdown dose estimate method by Monte Carlo code", Fusion Engineering and Design **55**, pp411-418 (2001)

[4] Y. Chen and U. Fischer “Rigorous MCNP Based Shutdown Dose Rate Calculations: Computational Scheme, Verification Calculations and Application to ITER”, *6th International Symposium on Fusion Nuclear Technology (ISFNT-6)*, San Diego, California, April 7-12, 2002. University of California, Center for Energy Research, San Diego, California.