CONF-830304--17

CONT-830304--17

DE83 009588

MONTE CARLO SMALL-SAMPLE PERTURBATION CALCULATIONS*

Uri Feldman, Ely Gelbard, and Roger Blomquist
Applied Physics Division
Argonne National Laboratory
Argonne, IL 60439

Two different Monte Carlo methods have been developed for benchmark computations of small-sample-worths in simplified geometries. The first is basically a standard Monte Carlo perturbation method in which neutrons are steered towards the sample by roulette and splitting. One finds, however, that two variance reduction methods are required to make this sort of perturbation calculation feasible. First, neutrons that have passed through the sample must be exempted from roulette. Second, neutrons must be forced to undergo scattering collisions in the sample. Even when such methods are invoked, however, it is still necessary to exagerate the volume fraction of the sample by drastically reducing the size of the core. The benchmark calculations are then used to test more approximate methods, and not directly to analyze experiments.

In the second method the flux at the surface of the sample is assumed to be known. Neutrons entering the sample are drawn from this "known" flux and tracked by Monte Carlo. The effect of the sample or the fission rate is then inferred from the histories of these neutrons.

The characteristics of both of these methods are explored empirically.

*Work supported by the U. S. Department of Energy.

MONTE CARLO SMALL-SAMPLE PERTURBATION CALCULATIONS

1. INTRODUCTION

The principles of Monte Carlo perturbation methods[1] are, by now, fairly well known. In practice, however, these methods are often difficult to use, and must be specially adapted to operate effectively in each of many different types of perturbation problems. Techniques specifically tailored for small-sample perturbation calculations have been developed recently, and inserted into VIM.[2] Because of the difficulty of such calculations these techniques have been designed primarily to test selected features of the standard analytic methods, and not to analyze experiments directly. Thus, for example, in most test computations run so far, zeroeth-generation fission sites are taken from a source generated by VIM in the "base configuration:" i.e. in a configuration in which the sample is present, but with half its normal density. Perturbation routines then calculate the difference, $\Delta\lambda_1^{MC}$, between <u>first-generation</u> fission rates with the sample present and with the sample replaced by a void.

The fission perturbation, $\Delta\lambda_1^{MC}$, can be used in two different ways, namely,

(i) it can be compared directly with the corresponding perturbation, $\Delta\lambda_1^{DET}$, computed by some coarser method whose validity is under study. Such a comparison could tell us, for example, whether heterogeneities around the sample, or the resonance cross sections, are being treated satisfactorily by the coarser method. In such comparisons it may often be sufficient to include, in the computational configuration, only the neighborhood of the sample.

(ii) Alternatively, one can estimate the <u>net</u> perturbations, $\Delta\lambda$, via the expression

$$\Delta\lambda = \int S_o^*(\underline{r})\,[S_1^P(\underline{r}) - S_1^U(\underline{r})]\,d\underline{r} \Big/ \int S_o^*(\underline{r})\,S_o(\underline{r})\,d\underline{r}. \qquad (1)$$

Here S_1^P and S_1^U are the first-generation perturbed and unperturbed fission source, respectively: S_o is the zeroeth-generation fission source, taken from a Monte Carlo or deterministic computation: and, finally, S_o^* is an unperturbed adjoint source, computed by approximate deterministic methods.

In most respects the perturbation computational technique is conventional. Neutrons are steered to the sample by splitting, and are tracked in the base configuration. Each neutron carries two perturbation weights, updated after each event, and used in calculating sample-in and sample-out reaction rates. This sort of perturbation calculation is described in more detail in Ref. 1, p. 152.

Unfortunately standard Monte Carlo perturbation methods seem to be inadequate for small-sample perturbation calculations, and additional variance reduction is required to make such calculations feasible. We describe next, some specialized techniques which we find to be useful in VIM perturbation calculations.

(a) In all these calculations: a neutron which enters the sample, at any time in it's history, is thereafter exempted from roulette and splitting for the rest of it's life. If this is not done, the efficiency of our perturbation calculations is sharply reduced.

(b) Further scattering collisions in the sample are forced by much the same process as is used (for other purposes) in MCNP.[3] In all computations to be described below, only first scattering collisions are forced. That is to say that if a neutron, entering the sample, has previously been scattered in the sample, it is tracked by conventional techniques. If it was not previously scattered in the sample, it splits into two fragments. One fragment passes through the sample uncollided, carrying the weight $W = W_i \exp(-\Sigma_t \ell)$. Here W_i is the weight of the neutron entering the sample, Σ_t is the sample cross section at the neutron's current energy, and ℓ is the track length laid off as the neutron traverses the sample. The second fragment is forced to scatter in the sample at some point along its path, a point drawn from a truncated exponential distribution. This scattered fragment then emerges from its forced collision with weight $(1-W)\times \Sigma_s/\Sigma_t$. From this point on tracking of the scattered fragment is conventional. It is possible, in VIM, to force more than one scattering collision per entering neutron but, in most of our computations, we find that one forced collision seems to be optimum. Advantages of forced collisions, in perturbation calculations, will be discussed in Section 2.

In addition, when it is crucially important that running times be minimized,

(c) we assume that the incoming flux at the sample surface is known, isotropic and spatially uniform, and that the exiting flux is also uniform and isotropic. Neutrons entering the sample are drawn from this "known" entering flux. When a neutron is forced into a scattering collision it is, in effect, split into two fragments. One acts as if the sample were present and scatters normally. The other acts as if it were in a void, and proceeds at it's original energy but (utilizing our assumptions about the existing flux) the uncollided fragment is made to emerge at the same point, and in the same direction, as the scattered fragment. Both of the fragments, as they emerge, are assigned the same initial random number, so that their subsequent histories tend to be highly correlated.

Computations in which first-generation perturbations are computed for a given fission source, using the techniques described in (a) and (b), above, will be referred to, here, as "Method 1" computations. Computations in which sample particles are born from an assumed surface flux will be called "Method-2" computations. Normally, of course, we will not know this surface flux unless we have already carried out a perturbation calculation. If, however, the optical thickness of the sample is extremely small, the effect of the sample on the entering flux is negligible. It may then be possible to estimate this flux well enough for our purposes, perhaps via a diffusion calculation in a simplified geometry in which the sample is ignored. In fact experimental samples are not small enough so that perturbations in the incoming flux are entirely negligible. On the other hand benchmark calculations in which the sample density is artificially reduced may still be quite useful for diagnostic purposes.

In the following section we discuss the performance of the methods we have just described in several different sample calculations. It should be understood that here we will be concerned not so much with the results of sample calculations as with running-times and computational efficiencies.

2. TEST CALCULATIONS

In Table 1 we list results of three Method-1 calculations, mainly to give some idea as to the sort of running times attainable today. In test problems 1 and 2 spherical samples are placed at the center of spherical "cores" with reflecting outer boundaries. Sample and core radii appear in Table 2, while sample compositions will be found in Table 3. Number densities within the core (U9-core number densities,[4] smeared and volume-weighted) are listed in Table 4. While the sample sizes, as well as the sample compositions, are fairly realistic, the core dimensions have been drastically reduced so as to make a perturbation calculation feasible. Again we note that we will not attempt to analyze small-sample experiments directly by Monte Carlo. Probably, with the methods and computers available today, this cannot yet be done. We want to show, however, that Monte Carlo can give us benchmark computational results in specially designed small-sample test problem configurations: with the aid of such benchmark results we can then test the deterministic methods which are used to analyze experiments. To be helpful as a diagnostic tool it is only necessary that the benchmark computational configuration should retain some crucial features of the real configuration like, for example, the heterogeneous structure near the sample.

Problem 3 has been included, here, so as to give an indication as to the magnitude of running times one might expect in realistic plate geometries. In problem 3 a ^{10}B sample plate has been placed across the center of a U9 drawer. This drawer is itself at the center of a 5 × 5 array of drawers surrounded by a reflecting boundary. It should be noted that only the central drawer contains a boron sample. Details of the drawer geometry are shown in Fig. 1. Clearly the efficiency of the Monte Carlo calculation is much worse in problem 3 than in problems 1 and 2, though it isn't possible, from information given here, to isolate the effects of geometry alone on running times.

On the other hand, some information is available on the effectiveness of the variance reduction schemes described, in Section 1, under headings (a) and

(b). To test these schemes problem 1 has been rerun in four different versions. Results of each calculation are listed in Table 5. It will be seen from Table 5 that forced collisions, and the elimination of roulette for neutrons that have passed though the sample, are both important methods for variance reduction, at least in this test problem. Together they increase the F.O.M., roughly, by a factor of 20.

It isn't difficult to see why one should shut off roulette after a particle has passed through the sample. Once we have succeeded, by splitting, in getting neutrons into the sample, it's wasteful to kill most of these neutrons by Russian roulette, thus losing much of the information they were carrying.

Some of the advantages of forced collisions are a little less obvious. If the sample's mean-chord-length is much smaller than a mean-free-path then, unless we force collisions, most of the neutrons that enter the sample will pass through it uncollided. For such neutrons the adjustments in the perturbation weights will be the same as if the sample were a pure absorber, an absorber whose total cross section is just the same as in the real sample. The effects of scattering collisions will than be felt only by those rare particles that do collide when they pass through the sample. Thus when real scattering events occur the Monte Carlo mean will change substantially and, correspondingly, the computed variance will suddenly increase.

But there is another advantage of forced collisions which one can understand most easily through an analysis of the simplest model problem. Consider a one-group model problem in which; (a) the sample is a pure scatterer, (b) the core is homogeneous, and (c) the core boundary is perfectly reflecting. It can easily be shown that, in this case, the sample-worth vanishes identically and thus, if we force collisions, the perturbation method will give a zero-worth with zero variances. On the other hand, without forced collisions the variance in the computed perturbation will not vanish, so that the relative error in the Monte Carlo calculation will always be infinite. It isn't clear which of these advantages of forced-collisions plays the most important role in test problem 1.

Finally, one additional test problem, problem 4, has been run with method 2, for a PuAℓ sample in a homogenized ZPPR-12[5] core composition. Core number densities for this composition are listed in Table 6, and computational results appear Table 7. First of all, it is clear from these results that the F.O.M. for a Pu Aℓ sample in a ZPPR core is only about one third as large as for a Pu sample in a U9 core. This is true primarily because the running-time per history in the ZPPR core (which is composed primarily of iron) is about twice as large as in the U9 core. Apparently most of the extra running-time is used to generate a great many iron scattering collisions. It is true, of course, that this time isn't simply lost, since information is collected during each collision. On the other hand, the amount of information in a history is limited and, apparently, in iron we have many more collisions than we need to extract this information most efficiently.

In Table 7, full density results have been included just for comparison. As has been noted earlier, Method 2 really can't be used unless the sample density is artificially reduced.

If one compares the Method 1 F.O.M. for the full-density sample, with the Method 2 F.O.M. for the reduced-density sample one concludes that the use of Method 2 reduces running-time by about a factor of two. This reduction is achieved, however, at the cost of another deviation from the real parameters of the experiment. As a result, some of the deterministic methods used in small-sample calculations (e.g. the methods used to treat self-shielding effects in the sample) cannot be tested by Method 2. On the other hand many resonance effects, and effects of local heterogeneities, still can be treated by Method 2.

Unfortunately the results exhibited in Table 7 are not yet understood. Thus, for example, it is not clear at this point, why the efficiency of Method 1 is reduced by a factor of two when the sample density is lowered.

3. CONCLUSIONS

It is clear that, like other Monte Carlo perturbational techniques, Monte Carlo perturbation methods for computing sample worths reduce the Monte Carlo running time by enormous factors. Consider, for example, the running time for problem 3 of Table 1. In this case the standard deviation in the fission rate, both with the sample in and sample out, is about 5.9×10^{-3}, and the standard deviation in "δk", the difference of the fission rates is 8.4×10^{-6}. Thus it would take about 1.9×10^{7} hrs of computing time on the IBM 3033, to achieve a standard deviation of 8.4×10^{-6} in the VIM δk. Using perturbation method we get to this same standard deviation in one hour.

Nevertheless, computing times for small-sample-worth calculations are still large. Small-sample worths, computed by deterministic methods, now agree with measured worths, in most cases to about 5%. It would, therefore, be very helpful if the standard deviation in the Monte Carlo calculation could be reduced a good deal below 5%. We see from Table 1 that this simply isn't possible without substantial improvements in computers or in Monte Carlo methods.

On the other hand, Monte Carlo sample-worths still can be useful even with σ's of about 3 or 4%. Deterministic calculations of these worths now involve the computation of various corrections, corrections for self-shielding in the sample, structural heterogeneities, and spectral details in the importance function. Monte Carlo benchmark even with an accuracy of 3 or 4%, would help to tell us, after all of these corrections have been made, whether the observed agreement with experiments is real or accidental, and whether any other large corrections still are missing. One can estimate, from line 3 of Table 1, that a ^{10}B sample-worth in a small bundle of ZPPR subassemblies, could be computed, via Method-1, in about 8 hours or the IBM 3033. Certainly this is a lot of computing time, but the cost of this sort of computation is still small compared to the net cost of small-sample experiments, and is comparable to the cost of a detailed, plate-by-plate, Monte Carlo benchmark for a critical experiment.

In connection with computing costs, it should be noted that little has been done, so far, to fine-tune the methods used in Monte Carlo small-sample computations. Perhaps the most obvious weakness in the sampling methods we have used is in our source-sampling process, which is the standard process

used in VIM, in conjunction with roulette and splitting. In VIM the source-sampling and splitting processes are coupled, and operate as follows:

(a) A source-weight, W_s^i, ($W_s^i > 1$) is associated with each splitting zone: i, here, is the number of the zone:

(b) if a particle, which has made it's last collision in zone i, makes its next collision in zone j, then it will be split by a factor r, $r \equiv W_j/W_i$, if $r > 1$, or rouletted by this same factor if $r < 1$:

(c) source sites are chosen from a specified source density by analogue methods: if a neutron starts from a site in region i, then it is split by the factor W_s^i.

While one can argue that such a source - splitting method is plausible, there is no reason to believe that it is optimum in any sense. The source distribution can be optimized, however, by a process similar to that discussed by Hoffman. Hoffman describes the source-biassing process as adoptive but suppose, for simplicity, that biasing parameters are determined, once and for all, at the beginning of the Monte Carlo calculation, in a preliminary Monte Carlo run. This preliminary run might proceed, for example, as follows:

(a) the space-energy phase space is divided into subregions, and a batch of neutrons is started from each of these subregions:

(b) for each subregion, i, one calculates the running time, t_i, per starter (including the time for all the offspring born in splitting processes) as well as σ_i, the standard deviation, per starter, in the region-Σ contribution to σk:

(c) finally $\hat{p}_i$, the biassed starting probability in subregion i, is given by the expression

$$\hat{p}_i = S_i \sqrt{\sigma_i/t_i} \Big/ \sum_j S_j \sqrt{\sigma_j/t_i} \ ,$$

S_i is the source-strength in subregion i. Except for the factors t_i, Eq. (1) above is essentially the same as Eq. (18) of Ref. 6. It should be observed, however, that optimum biasing parameters for the source (contrary to what is often assumed) seem to have no connection whatever with any conventional importance function. The effectiveness of source-biassing is small-sample Monte Carlo calculations is now under investigation.

4. REFERENCES

1. E. M. Gelbard, "Unfinished Monte Carlo Business," (ANL,USA) Proceedings of the International Topical Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, April 27-29, 1981.

2. R. N. Blomquist, R. M. Lell, and E. M. Gelbard, "VIM - A Continuous Energy Monte Carlo Code at ANL," (Argonne National Laboratory) Proceedings of a Seminar-Workshop, Oak Ridge, TN, April 21-23, 1980.

3. Los Alamos Monte Carlo Group, "MCNP-A General Monte Carlo Code for Neutron and Photon Transport," Version 2B (Los Alamos National Laboratory), April 13, 1982.

4. R. W. Shaefer and R. G. Bucher, "Calculated and Measured Reactivities in the U9 Critical Assemblies," (Argonne National Laboratory) Proceedings of the Meeting on Advances in Reactor Physics and Core Thermal Hydraulics, Kiamesha Lake, New York, September 22-24, 1982.

5. "Liquid Metal Fast Breeder Reactor Technical Progress Report," April-June, 1983 (ANL-RDP-107).

6. T. J. Hoffman, "Monte Carlo Source Biasing Optimization," (Oak Ridge National Laboratory), Nucl. Sci. and Eng., 69, 76-77, August 9, 1978.

Table 1. Sample Perturbation Calculations (Method 1)[a]

	Test Problem	Deterministic δk[b]	VIM δk	F.O.M.[c]
1.	Table 2 geometry, compositions from Tables 3 and 4 (B^{10} sample)	-2.24×10^{-2}	$-2.24 \times 10^{-2} \pm 2.5\%$[d]	26.7
2.	Problem 1 with Pu sample	2.80×10^{-2}	$2.66 \times 10^{-2} \pm 3.3\%$	15.3
3.	Figure 1 geometry, B^{10} sample	-1.38×10^{-4}	$-1.37 \times 10^{-4} \pm 6.2\%$	4.3

[a] All Monte Carlo calculations used 1 hr CPU time.

[b] Problems 1 and 2 solved using ANISN, problem 3 solved in cylindrical geometry with DIF2D.

[c] F.O.M. = $1/\sigma^2 T$ (CPU minutes).

Table 2. Spherical Core Geometry (Reduced Core)

Sample radius	1.305 cm
Core radius	7.6317 cm
V.F. ≡ sample volume/core volume	0.005
Splitting boundaries at r = 1.5 cm and 4.5 cm, splitting by factor of 2.	

Table 3. Perturbation Sample Densities

Pu sample	Pu^{239}	2.5×10^{22}
B sample	B^{10}	5.4350×10^{22}
Pu-Aℓ sample	Pu^{239}	2.6432×10^{22}
	Pu^{240}	1.261×10^{21}
	Pu^{241}	5.6×10^{19}
	Pu^{242}	1×10^{18}
	Am^{241}	6.1×10^{19}
	Aℓ	2.75×10^{21}

Table 4. Homogenized U9 Atom Densities

U^{235}	3.4851×10^{21}
U^{238}	3.5122×10^{22}
U^{234}	3.3425×10^{19}
U^{236}	1.6024×10^{19}
Cr	1.8927×10^{21}
Ni	8.2170×10^{20}
Fe	6.6947×10^{21}
O	2.4100×10^{20}
Mn	1.4850×10^{20}

Table 5. Comparison of Variance Reduction Techniques

Techniques		F.O.M.
No forced collisions	Regular RR/splitting	1.56
	RR/splitting[a] cut out	6.28
Forced collisions	Regular RR/splitting	4.27
	RR/splitting[a] cut out	34.4

[a] i.e. each neutron which has entered the sample is thereafter exempted from roulette and splitting.

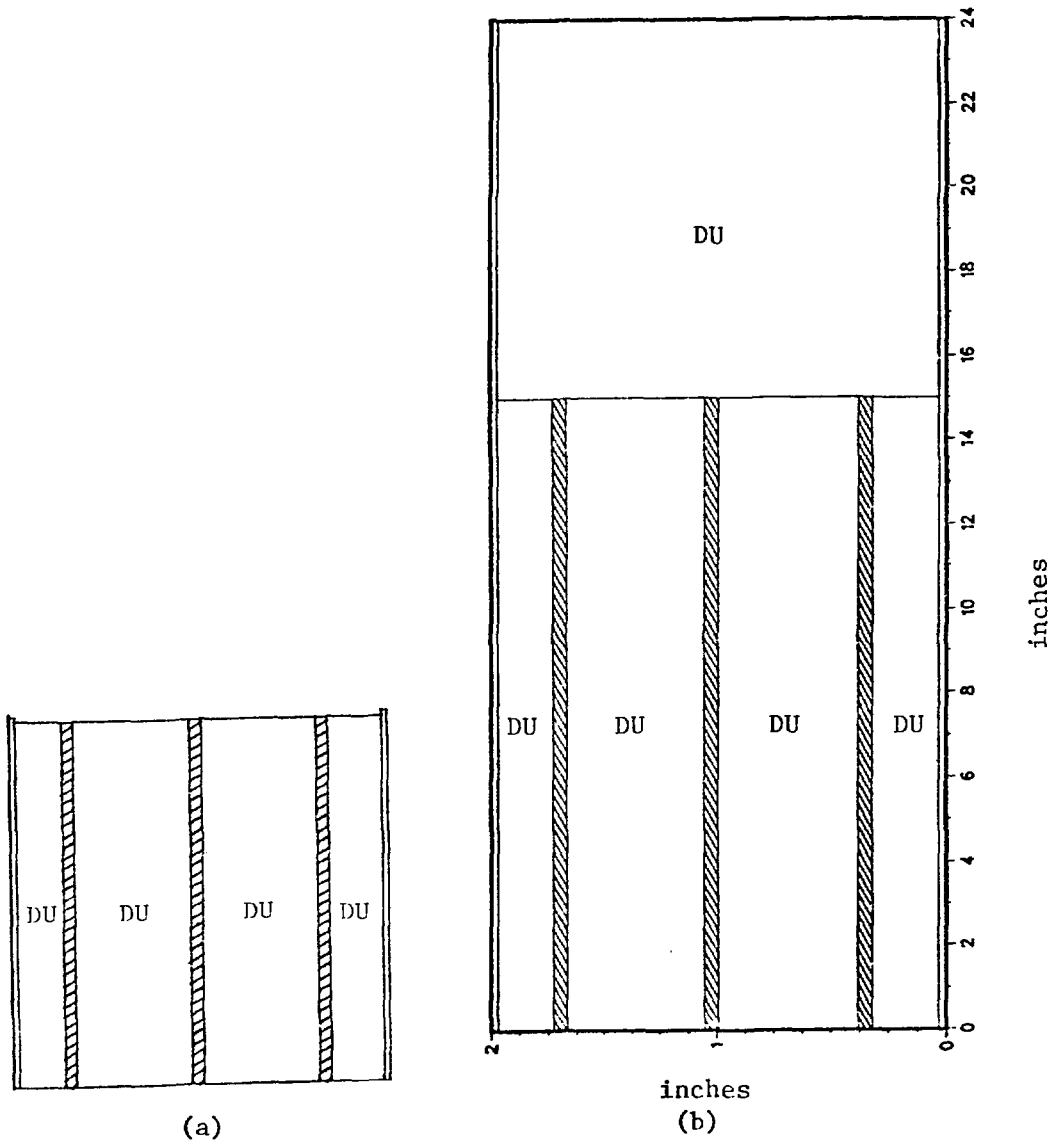
Table 6. Homogenized ZPPR-12 Atom Densities

Pu^{239}	1.7662×10^{21}
Pu^{240}	2.338×10^{20}
Pu^{241}	2.08×10^{19}
Pu^{242}	3.8×10^{18}
U^{235}	1.12×10^{19}
U^{238}	5.01976×10^{21}
Am^{241}	1.53×10^{19}
Cr	3.1669×10^{21}
Ni	1.4197×10^{21}
Fe	1.79308×10^{22}
Na	8.688×10^{21}
O	9.8618×10^{21}
C	3.68×10^{19}
Mo	4.642×10^{20}
Ma^{55}	2.688×10^{20}
Cu	4.44×10^{19}
Si	1.84×10^{20}

Table 7. Comparison of Method 1 with Method 2 in Problem 4

	Method 1	Method 2
Full density sample	5.51	15.1
0.1 density sample	2.15	12.2

Figure 1. U9 Assembly Drawer Layout (a) Cross Section, and (b) Top View (not to scale). Cross hatched plates are enriched uranium, other plates are depleted uranium.



DISCLAIMER