MONTE CARLO PERTURBATION THEORY IN NEUTRON TRANSPORT CALCULATIONS

M. C. G. Hall
Imperial College of Science and Technology *
London, England

ABSTRACT

The need to obtain sensitivities in complicated geometrical configurations has resulted in the development of Monte Carlo sensitivity estimation. A new method has been developed to calculate energy-dependent sensitivities of any number of responses in a single Monte Carlo calculation with a very small time penalty. This estimation typically increases the tracking time per source particle by about 30%. The method of estimation is explained. Sensitivities obtained are compared with those calculated by discrete ordinates methods. Further theoretical developments, such as second order perturbation theory and application to k_{eff} calculations are discussed. The application of the method to uncertainty analysis and to the analysis of benchmark experiments is illustrated.

INTRODUCTION

The result of a neutron transport calculation can be very sensitive to nuclear data, and experimental error in these data may cause the result to be misleading. To determine how important this effect is, it is necessary to calculate sensitivities and combine them with covariance information to obtain the standard deviation of the result. If the uncertainty arising in this way is unacceptably large, then the nuclear data must be improved. One way to do this is to adjust on the basis of a benchmark experiment, which also involves the calculation of sensitivities. The motivation for this work is the need to calculate sensitivities in geometries which, because of their complexity, require a Monte Carlo calculation.

There are additional advantages in using a Monte Carlo method for the analysis of benchmark experiments. All the sensitivity information can be estimated simultaneously, whereas conventional methods require a separate adjoint calculation for each channel of experimental information. Also group-averaging errors can be avoided by the use of point nuclear data, so the adjustments should reflect shortcomings in the basic data, rather than difficulties in a group-averaging process. The number of

*Work performed during author's attachment to Radiation Physics and Shielding Group, Reactor Physics Division, AEE Winfrith.

sensitivity coefficients which need to be estimated though, can be as many as a thousand, so it is essential that each should be scored with very little time penalty.

The method described here has been implemented with all this in mind, although the approach has been kept as general as possible, so that different applications can be catered for.

METHOD

Any response R can be considered as an average value associated with the set P of all neutron paths:

$$R = \sum_{m \in P} r^m p^m \,. \tag{1}$$

Here p^m is the probability of path m, and r^m is the estimator for path m. If $\hat{D}$ is a linear perturbation operator, then $\hat{D}$ operating on Eq. (1) gives

$$\hat{D}R = \sum_{m \in P} \hat{D}(r^m p^m) \,. \tag{2}$$

Writing Eq. (2) in the form of Eq. (1)

$$\hat{D}R = \sum_{m \in P} dr^m \, p^m \tag{3}$$

where

$$dr^m = \{\hat{D}(r^m p^m)\}/p^m \,, \tag{4}$$

so dr^m is an estimator of $\hat{D}R$. Re-arranging Eq. (4)

$$dr^m = v^m r^m \tag{5}$$

where

$$v^m = \{\hat{D}(r^m p^m)\}/(r^m p^m) \,, \tag{6}$$

so dr^m is a weighted version of r^m.

This estimator has the disadvantage that it is bad at estimating the component of a change which is zero because of physical restrictions. For example, in a non-multiplying medium a perturbation of a cross-section at low energy usually cannot affect the flux at high energy. But even if $\hat{D}$ only operates on cross-sections at low energy and r^m only scores flux at high energy, the estimator dr^m given by Eq. (4) will still in general be non-zero, although it will have zero expectation. A more discerning estimator would itself be zero under these circumstances. To achieve this r^m and p^m are split into components associated with each trajectory of path m:

$$r^m = \sum_{i \le t^m} s_i^m \tag{7}$$

and

$$p^m = \prod_{j \le t^m} q_j^m \ . \tag{8}$$

In this context: a trajectory is a section of a path along which no collisions or boundary crossings occur; t^m is the number of trajectories in path m; s_i^m is the contribution to the estimator for path m arising from the i^{th} trajectory; q_j^m is the probability of the j^{th} trajectory, given that the $(j-1)^{th}$ trajectory has occurred. It is shown in the appendix that another estimator for $\hat{D}R$ is given by

$$dr^m = \sum_{i \le t^m} \left\{ \hat{D}\left(s_i^m \prod_{j \le i} q_j^m \right) \right\} / \prod_{j \le i} q_j^m \ . \tag{9}$$

This estimator will always be zero in the circumstances which have just been mentioned. This is because, in Eq. (9), by the time i is large enough for $\hat{D}$ to operate on s_i^m and q_j^m, the value of s_i^m is zero. Re-arranging as before

$$dr^m = \sum_{i \le t^m} w_i^m s_i^m \tag{10}$$

where

$$w_i^m = \left\{ \hat{D}\left(s_i^m \prod_{j \le i} q_j^m \right) \right\} / \left(s_i^m \prod_{j \le i} q_j^m \right) \ . \tag{11}$$

These are the key expressions used in the method. The two expressions for dr^m in Eqs. (5) and (10) can be compared by writing Eq. (5) as

$$dr^m = \sum_{i \le t^m} v^m s_i^m \ . \tag{12}$$

IMPLEMENTATION

Implementation consists of specifying $\hat{D}$, s_i^m and q_j^m of Eq. (11), evaluating the weights w_i^m, and scoring the estimator dr^m in the same way as the estimator r^m.

Specifying $\hat{D}$

The form of $\hat{D}$ is determined by the nature of the perturbations of interest. In the analysis of a benchmark experiment it is usual to regard the result R of a calculation as a function of the nuclear data $\underline{x}$:

$$z = R(\underline{x}) \tag{13}$$

$\underline{x}$ is subject to experimental error and may be perturbed by an amount $\underline{\delta x}$. The Taylor expansion of R about $\underline{x}$ is

$$\delta R = \sum_i \delta x_i \frac{\partial}{\partial x_i} R + \frac{1}{2!} \sum_{ij} \delta x_i \delta x_j \frac{\partial^2}{\partial x_i \partial x_j} R + \dots \quad . \tag{14}$$

Writing this in dimensionless terms

$$\frac{\delta R}{R} = \sum_i \frac{\delta x_i}{x_i} \frac{x_i}{R} \frac{\partial}{\partial x_i} R + \frac{1}{2!} \sum_{ij} \frac{\delta x_i}{x_i} \frac{\delta x_j}{x_j} \frac{x_i x_j}{R} \frac{\partial^2}{\partial x_i \partial x_j} R + \dots \quad . \tag{15}$$

Often some of the values of $\delta x_i / x_i$ are constrained to be equal—for example they may be subject to a systematic error. Suppose for a set of integers K and associated constant c_K there is a constraint

$$\delta x_i / x_i = c_K \qquad i \in K \tag{16}$$

—similarly for another set L and constant c_L. In this case Eq. (15) can be factorised:

$$\frac{\delta R}{R} = \sum_K c_K \sum_{i \in K} \frac{x_i}{R} \frac{\partial}{\partial x_i} R + \frac{1}{2!} \sum_K \sum_L c_K c_L \sum_{i \in K} \sum_{j \in L} \frac{x_i x_j}{R} \frac{\partial^2}{\partial x_i \partial x_j} R + \dots \quad . \tag{17}$$

The terms which it is useful to know are

$$\sum_{i \in K} \frac{x_i}{R} \frac{\partial}{\partial x_i} R \tag{18}$$

and

$$\sum_{i \in K} \sum_{j \in L} \frac{x_i x_j}{R} \frac{\partial^2}{\partial x_i \partial x_j} R \quad . \tag{19}$$

The first of these terms can be written as

$$(1/R) \hat{D} R \tag{20}$$

where

$$\hat{D} = \sum_{i \in K} x_i \frac{\partial}{\partial x_i} \quad . \tag{21}$$

This is the form of $\hat{D}$ considered in this section.

Specifying S and q

The quantities S and q are determined by the type of trajectory and by the method of estimation. Suppose a trajectory starts with a neutron undergoing a reaction type α at energy E. The neutron is scattered through an angle θ to energy E', continues for a length λ and then collides. The cross-section for reaction β at energy E is x_g^β. For convenience x_g^B is defined by

$$x_g^B = \sum_{\beta \in B} , \tag{22}$$

where B is a set of partial cross-sections. The probability of the trajectory alone is given by

$$q = (x_g^\alpha / x_g^T)\, p(\theta)\, d\theta\, p(E,E')\, dE'\, \exp(-x_{g'}^T \lambda)\, x_{g'}^T\, d\lambda . \tag{23}$$

Here T is the set of all reactions making up the total cross-section and $p(\theta)\, d\theta\, p(E,E')\, dE'$ is the probability distribution function in phase-space of the secondary neutron. If a Monte Carlo code uses point nuclear data, then values such as x_g^β are used in the sampling procedure. In this case $\hat{D}$ might be given by

$$\hat{D} = \sum_{\beta \in P} \sum_{h \in G} x_h^\beta \frac{\partial}{\partial x_h^\beta} . \tag{24}$$

Here P is a set of cross-section types, for example non-elastic, and G is a set of values of h comprising an energy interval. In this case $(1/R)\hat{D}R$ would be the sensitivity of R to the non-elastic cross-section in the G^m energy interval.

If track length estimation is used, the contribution to the estimator for the trajectory will be given by

$$S = \lambda \sigma_{g'} , \tag{25}$$

where σ_g is a response cross-section at energy E'. If collision density estimation is used, then

$$S = \sigma_{g'} / x_{g'}^T . \tag{26}$$

Evaluating ω_i^m

The symbol δ_α^P is defined to be unity if α belongs to the set P, and otherwise to be zero. If $\hat{D}$ is given by Eq. (24) and track length estimation is used, the weight ω_i^m is given by

$$\omega_i^m = \sum_{j \le i} \left\{ \delta_\alpha^P - \delta_j^G \left(x_j^P / x_j^T\right) - \delta_j^G \left(x_{j'}^P \cdot \lambda\right) + \delta_{j'}^G \left(x_{j'}^P / x_{j'}^T\right) \right\} . \quad (27)$$

If collision density estimation is used, then

$$\omega_i^m = \sum_{j \le i} \left\{ \delta_\alpha^P - \delta_j^G \left(x_j^P / x_j^T\right) - \delta_{j'}^G \left(x_j^P, \lambda\right) \right\} . \quad (28)$$

Fortunately the weights can be evaluated using a recursion formula—for example with track length estimation

$$\omega_i^m = \omega_{i-1}^m + \left\{ \delta_\alpha^P - \delta_j^G \left(x_j^P / x_j^T\right) - \delta_{j'}^G \left(x_{j'}^P \cdot \lambda\right) + \delta_{j'}^G \left(x_{j'}^P / x_{j'}^T\right) \right\} . \quad (29)$$

It can be seen from Eq. (29) why the method is so fast. The values of x_j^P, x_j^T, $x_{j'}^P$, $x_{j'}^T$ and λ are all available in an unperturbed calculation, so the weight ω_i^m only has to assemble information which is already there. In addition, the weight ω_i^m can be applied to any estimator s_i^m, which means that sensitivities to any number of responses can be scored simultaneously.

Scoring dr^m

A code called DUCKPOND[1] has been written to score the estimator dr_i^m. Full use has been made of the Winfrith Shielding Group's suite of Monte Carlo modules. This has meant that DUCKPOND has been coded with a minimum of effort yet includes the powerful capabilities familiar to users of McBEND[2]. In addition to a sensitivity capability, DUCKPOND can score a covariance matrix for all the estimated responses. This is useful in analysing benchmark experiments.

COMPARISON

Sensitivities can be obtained in limited circumstances using discrete ordinates (S_n) calculations, so an important test is to compare results of this type with answers estimated using the Monte Carlo method. The only difficulty with the comparison is that the discrete ordinates calculation may involve group-averaging errors, whereas this will not be the case with Monte Carlo estimation using point nuclear data. To determine the extent of this problem, fluxes are compared in addition to sensitivities.

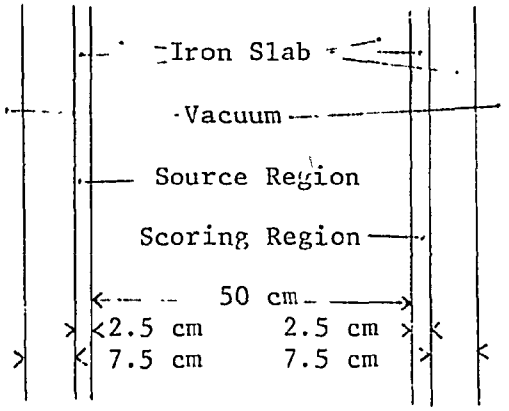

Fig. 1. The Geometry of the Test Problem.

The geometry of the test problem is illustrated in Fig. 1. The response function is for total flux per unit volume betweem 14.9 MeV and 407 KeV; the source has a fission spectrum normalised to $1{\cdot}0\ cm^{-3}\ s^{-1}$. Sensitivities to the elastic and non-elastic cross-sections of iron are calculated in forty groups of approximately equal lethargy width between 14.9 MeV and 407 KeV, corresponding to the first forty groups of the 100-group EURLIB[3] structure. Flux per unit volume in the scoring region is calculated in the same group scheme.

ANISN[4] and SWANLAKE[5] were used for the S_n calculations and DUCKPOND for the Monte Carlo estimation. Fluxes and sensitivities are compared in Figs. 2 and 3; the errors plotted for the Monte Carlo results are an estimate of one standard deviation.

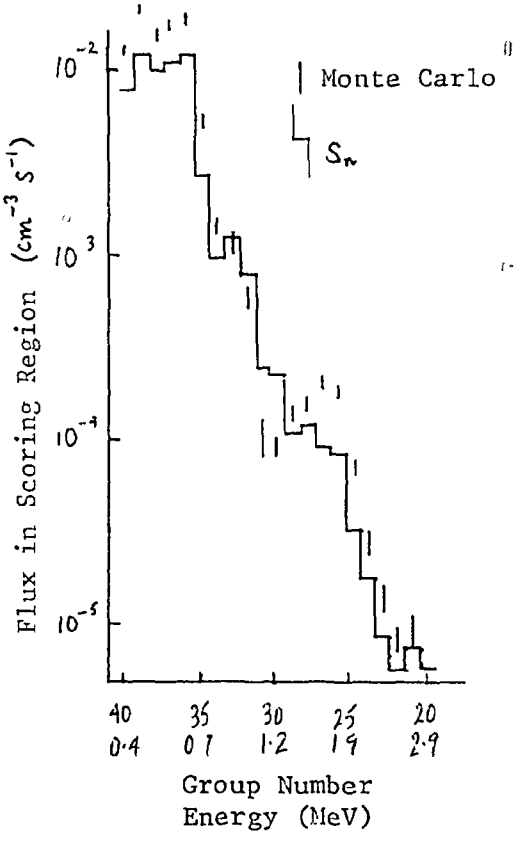

Fig. 2. Comparison of Flux in Scoring Region calculated by Monte Carlo and S_n Methods.

The Monte Carlo sensitivity estimation is working well. The agreement with the S_n method is convincing and, equally important, acceptable variances have been obtained with a workable sample size. DUCKPOND was run in this case for twenty minutes on an IBM 3033 using automatically-generated importance sampling, and the test problem is representative in scale of a realistic calculation. Moreover a comparison of running times between DUCKPOND and its non-perturbative equivalent McBEND show that sensitivity estimation typically slows down tracking by 30%. To give an example—for the same price a response might either be estimated by McBEND with a standard deviation of 10%, or estimated by DUCKPOND with a standard deviation of 11.4% but with a full set of sensitivity profiles. If the sensitivities reveal uncertainty of 20% arising from data errors, then the DUCKPOND calculation would be the more useful. It would be sensible to perform an uncertainty analysis of this kind on all Monte Carlo calculations.

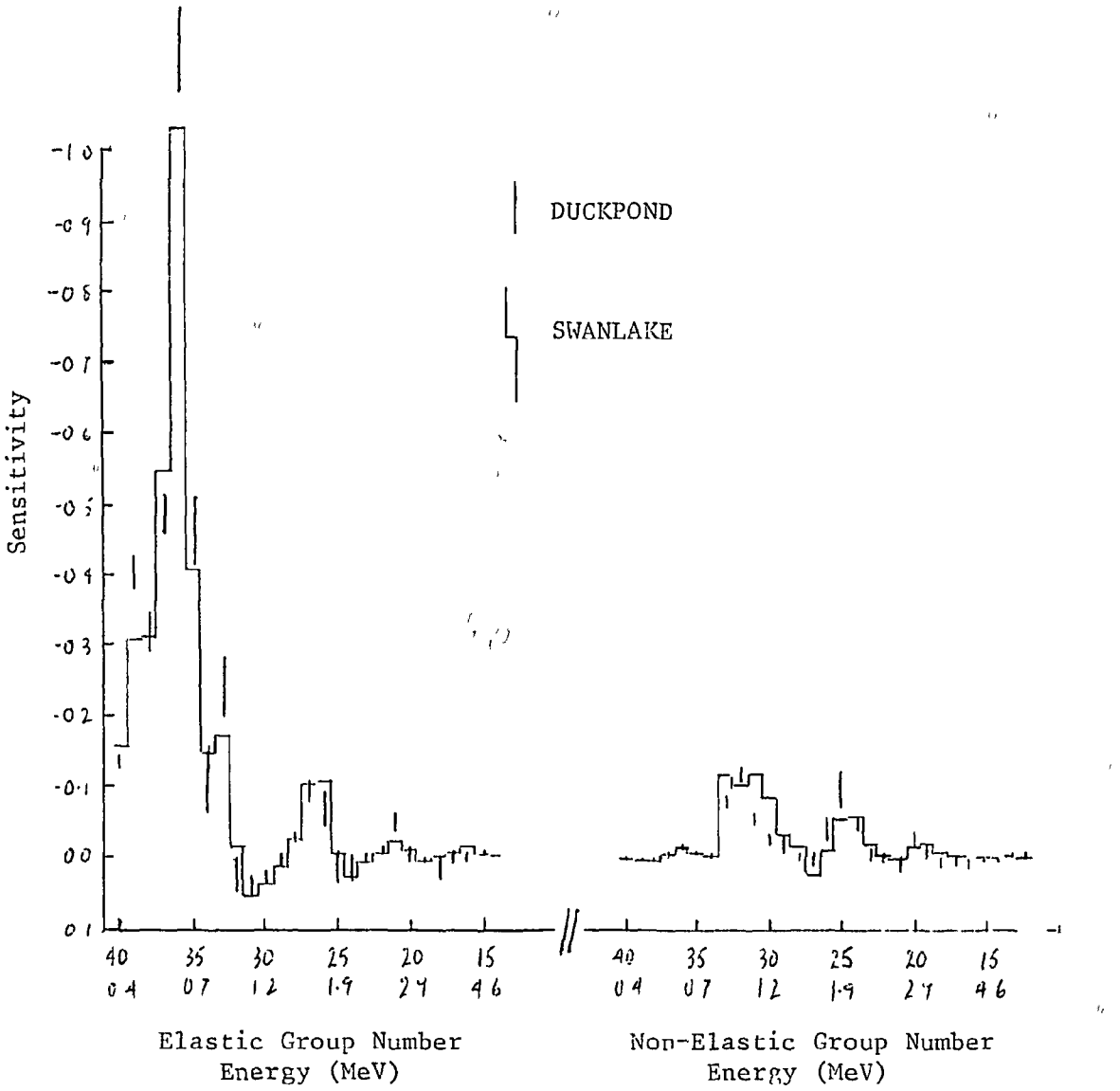

Fig. 3. Comparison of Sensitivities of Flux above 407 KeV to Elastic and Non-Elastic Cross-Sections of Iron, as calculated by DUCKPOND and SWANLAKE.

A disturbing aspect of the results is the discrepancy between the fluxes calculated by the Monte Carlo and S_n methods. The low energy flux, which is the main contributor to the total flux, is undercalculated by about 50% by the S_n method. The sensitivity profiles suggest that an explanation of this is a group-averaging overestimate of about 30% in the cross-sections for groups 35 to 39. Such an error could easily arise: the elastic cross-section of iron is rapidly changing by factors of about five in this energy range, which puts great importance on the weighting function used in the averaging process.

DEVELOPMENT

There are three interesting ways in which this method can be developed: higher order perturbation coefficients could be scored, sensitivities to geometric data obtained, and the method could be applied to eigenvalue calculations.

Higher Order Coefficients

A second order operator has already arisen in Eq. (19). Such an operator is now defined by

$$\hat{D}_{KL} = \sum_{i \in K} \sum_{j \in L} x_i x_j \frac{\partial^2}{\partial x_i \partial x_j} \quad . \tag{30}$$

It is also convenient to redefine corresponding first order operators:

$$\hat{D}_K = \sum_{i \in K} x_i \frac{\partial}{\partial x_i} \tag{31}$$

and

$$\hat{D}_L = \sum_{j \in L} x_j \frac{\partial}{\partial x_j} \quad . \tag{32}$$

The first order weights are given by Eq. (11) and turn out to be

$${}^{K}\omega_i^m = \frac{1}{s_i^m} \hat{D}_K s_i^m + \sum_{j \le i} \frac{1}{q_j^m} \hat{D}_K q_j^m \tag{33}$$

and

$${}^{L}\omega_i^m = \frac{1}{s_i^m} \hat{D}_L s_i^m + \sum_{j \le i} \frac{1}{q_j^m} \hat{D}_L q_j^m \quad , \tag{34}$$

and the second order weights are given by

$$\begin{aligned} {}^{KL}\omega_i^m = {}^{K}\omega_i^m {}^{L}\omega_i^m + \frac{1}{s_i^m} \hat{D}_{KL} s_i^m - \left(\frac{1}{s_i^m} \hat{D}_K s_i^m\right)\left(\frac{1}{s_i^m} \hat{D}_L s_i^m\right) \\ + \sum_{j \le i} \left\{ \frac{1}{q_j^m} \hat{D}_{KL} q_j^m - \left(\frac{1}{q_i^m} \hat{D}_K q_i^m\right)\left(\frac{1}{q_j^m} \hat{D}_L q_j^m\right) \right\} \quad . \end{aligned} \tag{35}$$

These equations are illustrated by referring to the specifications of $\hat{D}$, S and q appearing in the section on implementation. $\hat{D}_K$ and $\hat{D}_L$ are given by

$$\hat{D}_K = \sum_{\beta \in P} \sum_{h \in G} x_h^\beta \frac{\partial}{\partial x_h^\beta} \tag{36}$$

and

$$\hat{D}_L = \sum_{\beta \in Q} \sum_{h \in H} x_h^\beta \frac{\partial}{\partial x_h^\beta} \quad , \tag{37}$$

with a corresponding definition of $\hat{D}_{KL}$. The previous definitions of S and q still stand. For track length estimation all the terms involving S in Eqs. (33) to (35) are zero. The other terms are given by

$$(1/q_j^m)\hat{D}_K q_j^m = \delta_\alpha^P - \delta_\alpha^G (x_g^P / x_g^T) - \delta_{g'}^G (x_{g'}^P, \lambda) + \delta_{g'}^G (x_{g'}^G / x_{g'}^T), \quad (38)$$

$$(1/q_j^m)\hat{D}_L q_j^m = \delta_\alpha^Q - \delta_g^H (x_g^Q / x_g^T) - \delta_{g'}^H (x_{g'}^Q, \lambda) + \delta_{g'}^H (x_{g'}^Q / x_{g'}^T), \quad (39)$$

$$(1/q_j^m)\hat{D}_{KL} q_j^m = -(\delta_\alpha^P + \delta_\alpha^Q)(x_g^P x_g^Q)/(x_g^T x_g^T) - 2\delta_g^G \delta_j^H (x_j^P x_g^Q)/(x_g^T)^2 + \delta_{g'}^G \delta_{g'}^H (x_{g'}^P x_{g'}^Q \lambda^2), \quad (40)$$

so second order weights are not much more difficult to evaluate than first order, although there may be more of them. The most likely use of second order coefficients would be in testing the validity of a first order approximation.

Geometric Sensitivities

Suppose a geometrical configuration is defined in terms of rectangular co-ordinates r_1, r_2 and r_3. A plane with unit normal $\underline{n}$ which separates two different media may be described by the equation

$$\underline{n} \cdot \underline{r} = l . \quad (41)$$

An operator which describes a first order change in the position of the plane is

$$\hat{D} = \partial/\partial l . \quad (42)$$

Again using the previous specifications for S and q, the only variable which can depend on l is λ. If u_1, u_2 and u_3 are the direction cosines of a trajectory then

$$\partial\lambda/\partial l = K/\underline{n} \cdot \underline{u} . \quad (43)$$

Here $K = -1$ for boundary crossing at the beginning of the trajectory, $K = 1$ for boundary crossing at the end of the trajectory, and $K = 0$ otherwise. If collision density estimation is used, the weight given by Eq. (11) is

$$\omega_c^m = -K x_g^T / \underline{n} \cdot \underline{u} . \quad (44)$$

This could be used to score the geometric sensitivity $\partial R/\partial l$.

Eigenvalue Calculations

Monte Carlo eigenvalue calculations differ from shielding calculations in that superimposed on the normal processes of tracking and scoring is an iterative procedure. An eigenvalue calculation in a code such as MONK[6] will usually start with a fission source guess represented by N_0 particles. These particles are tracked to leakage, absorption or fission. N_0 of the resulting N_1 secondary particles are sampled at random and the process continues for successive generations. The ratio K_j is defined by

$$K_j = N_j / N_0 \tag{45}$$

where N_j is the number of secondary particles at the end of the j^{th} generation.

Each trajectory involved is labelled according to the particle from which it originated (m) and the generation in which it occurs (j). The probability of the i^{th} trajectory of this type is defined in the same way as before to be q_{ij}^m. The probability of all the trajectories up to and including the k^{th} generation which originate from the m^{th} particle is given by

$$p_k^m = \frac{1}{N_0} \prod_{j=1}^{k} \frac{1}{K_{j-1}} \prod_{i} q_{ij}^m \,. \tag{46}$$

If r_k^m is the number of secondaries at the end of the k^{th} generation originating from the m^{th} particle, then another expression of the ratio K_k is

$$K_k = \lim_{N_0 \to \infty} \sum_{m=1}^{N_0} r_k^m p_k^m \,. \tag{47}$$

Furthermore the eigenvalue k_{eff} is given by

$$k_{eff} = \lim_{k \to \infty} K_k \,, \tag{48}$$

so for k sufficiently large, r_k^m is an estimator of k_{eff}. If α is some perturbation parameter then

$$\frac{\partial K_k}{\partial \alpha} = \lim_{N_0 \to \infty} \sum_{m=1}^{N_0} r_k^m \frac{\partial p_k^m}{\partial \alpha} \,, \tag{49}$$

or

$$\frac{\partial K_k}{\partial \alpha} = \lim_{N_0 \to \infty} \sum_{m=1}^{N_0} dr_k^m \, p_k^m \tag{50}$$

where

$$dr_k^m = \frac{1}{p_k^m} \frac{\partial p_k^m}{\partial \alpha} r_k^m \,. \tag{51}$$

This means that dr_k^m is an estimator of $\partial K_k / \partial \lambda$ and, for k sufficiently large, $\partial k_{eff} / \partial \lambda$. Re-arranging Eq. (51)

$$dr_k^m = y_k^m r_k^m \tag{52}$$

where

$$y_k^m = \frac{1}{P_k^m} \frac{\partial P_k^m}{\partial \lambda}. \tag{53}$$

The evaluation of the weight y_k^m may be troublesome. Substituting for P_k^m from Eq. (46)

$$y_k^m = \sum_{j=1}^{k} \left(\sum_i \frac{1}{q_{ij}^m} \frac{\partial}{\partial \lambda} q_{ij}^m - \frac{1}{K_{j-1}} \frac{\partial K_{j-1}}{\partial \lambda} \right) \tag{54}$$

so

$$y_k^m = y_{k-1}^m + \sum_i \frac{1}{q_{ik}^m} \frac{\partial}{\partial \lambda} q_{ik}^m - \frac{1}{K_{k-1}} \frac{\partial K_{k-1}}{\partial \lambda}. \tag{55}$$

The difficulty is in evaluating the term $\partial K_{k-1} / \partial \lambda$. An exact value is not available, so an estimate using dr_{k-1}^m has to be used. If N_0 is not sufficiently large, the estimates of $\partial K_k / \partial \lambda$ may get progressively worse, and the method may not converge.

APPLICATION

DUCKPOND has been applied to a variety of practical problems. Preliminary results of the uncertainty analysis for the NRC Blind Test[7] have already been produced and sensitivity calculations have been carried out for the analysis of the Winfrith Iron Benchmark[8] . Some of these results are presented here.

NRC Blind Test

This test is to see how well a series of experimental reaction rates in a simulated pressure vessel can be predicted by various methods. It is useful to be able to isolate each source of error in the predictions, and DUCKPOND has been used to evaluate uncertainties arising from errors in nuclear data. The significant sensitivities of one of the reaction rates, as calculated by DUCKPOND, are illustrated in Fig. 4. These sensitivities were combined with covariance information about nuclear data[9] to obtain an uncertainty of 14%. The geometrical configuration involved in the calculation precluded an S_n method.

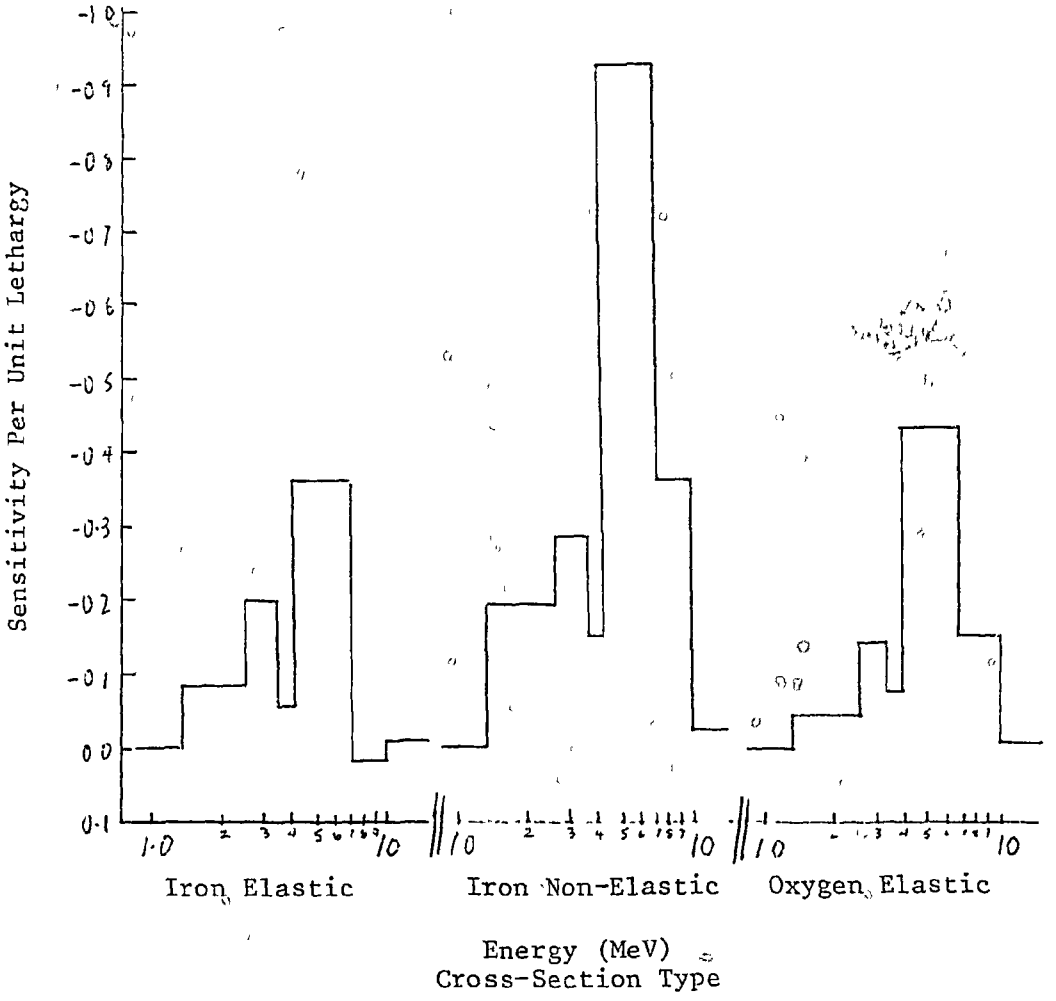

Fig. 4. Sensitivity per Unit Lethargy of $^{58}Ni(n,p)^{58}Co$ Reaction Rate in a Simulated Pressure Vessel, to Various Cross-Sections.

Winfrith Iron Benchmark

The objective of this benchmark is to adjust the evaluated cross-section of iron on the basis of count rates measured in an iron block with a fission source at one end. Each count rate is calculated along with a sensitivity profile: the sensitivities determine which adjustments would improve the agreement between calculation and experiment, and cross-section covariance information indicates which the likely adjustments are.

DUCKPOND was used for the calculations. The experimental configuration was modelled very accurately using combinatorial geometry. Sixty count rates and sensitivity profiles were scored, which involved tracking neutrons in iron to a depth of 7.5 cm and down to 5 KeV. After twenty minutes running on an IBM 3033 the statistical error on the estimated count rates

had reached the same level as the experimental error (about 15%). One of the sensitivity profiles is illustrated in Fig. 5. The pronounced positive sensitivity is probably a three-dimensional effect: an increase in the cross-section will reduce the leakage and hence increase the count rate. The results of an adjustment procedure based on this DUCKPOND calculation are shortly to be produced.

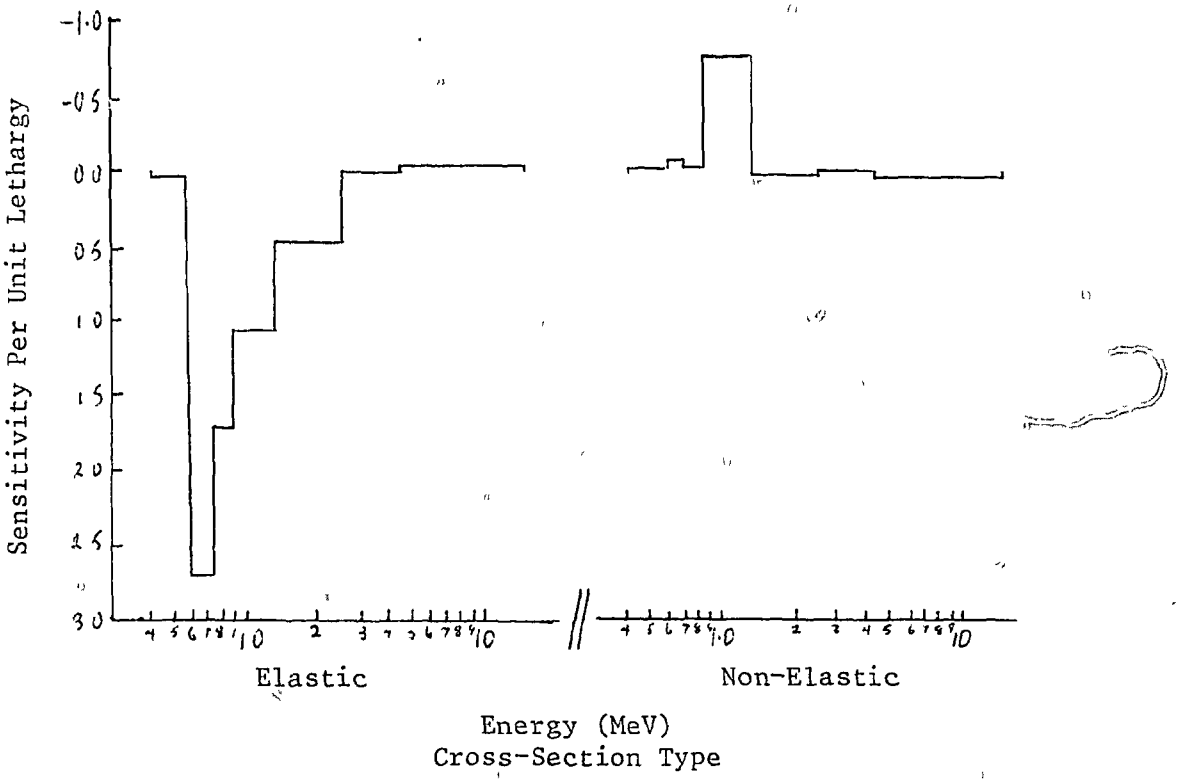

Fig. 5. Sensitivity Per Unit Lethargy of an Hydrogen/Argon Proportional Count Rate at 50 cm Penetration, to Iron Cross-Sections. (The threshold of the detector is about 4.4 MeV.)

CONCLUSION

It is now a simple and inexpensive matter to carry out the uncertainty analysis of a transport calculation, however complicated the geometrical configuration. It is also clearly feasible to analyse a shielding benchmark experiment on the basis of a Monte Carlo calculation. Application of this method to core calculations is as yet untested, and this would be a sensible option to try out. Geometric and second order developments would be easy to implement, although they might be of limited use. There is no reason why the method should not be applied to calculations using multigroup data if necessary.

REFERENCES

1. M. C. G. Hall, DUCKPOND - A Monte Carlo Code for Determining the Unknown Consequences of Known Perturbations of Nuclear Data, to be published.

2. D. E. Bendall and R. J. Brissenden, McBEND Programme Users Guide, To be published.

3. "EURLIB Library", ESIS Newsletter No. 12, January 1975.

4. W. W. Engle, A Users Manual for Anisn, a One-Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering K-1673, Computing Technology Centre, Oak Ridge, Gaseous Diffusion Plant, 1967.

5. D. E. Bartine, F. R. Mynatt and E. M. Oblow, Swanlake, A Computer Code Utilising Anisn Radiation Transport Calculations for Cross-Section Sensitivity Analysis, ORNL-TM-3809, Mathematics Division, Oak Ridge National Laboratory, May 1973.

6. V. S. W. Sherriffs, Monk - A General Purpose Monte Carlo Neutronics Program, SRD Culcheth No. 86, January 1978.

7. Technical Letter for PCA Blind Test, NRC April 1979.

8. M. D. Carter and A. Packwood, The Winfrith Benchmark Experiment in Iron - Experimental Results, Proc. of Specialists Meeting on Sensitivity Studies and Shielding Benchmarks NEA/OECD, Paris, 1975.

9. J. D. Drischler and C. R. Weisbin, Compilation of Multigroup Cross-Section Covariance Matrices for Several Important Reactor Materials, ORNL-5318 Neutron Physics Division, Oak Ridge National Laboratory, October 1977.

APPENDIX

A path segment is defined to be a set of contiguous trajectories which make up the initial part of the path. R_k is the set of paths which start with segment k. ${}^kS_i$ and ${}^kq_i$ are defined by

$$ {}^kS_i = S_i^m \qquad m \in R_k \tag{56} $$

and

$$ {}^kq_i = q_i^m \qquad m \in R_k \ . \tag{57} $$

The number of trajectories in segment k is u^k. The response R can now be regarded as the average value associated with the set S of all segments:

$$R = \sum_{k \in S} {}^k S_{u^k} \prod_{j \le u^k} {}^k q_j \quad . \tag{58}$$

$\hat{D}$ operating on Eq. (58) gives

$$\hat{D}R = \sum_{k \in S} d^k S_{u^k} \prod_{j \le u^k} {}^k q_j \tag{59}$$

where

$$d^k S_{u^k} = \left\{ \hat{D} \left({}^k S_{u^k} \prod_{j \le u^k} {}^k q_j \right) \right\} \Big/ \prod_{j \le u^k} {}^k q_j \quad . \tag{60}$$

T_i is the set of segments with exactly i trajectories ($u^k = i$). The sum over segments ($\sum_{k \in S}$) is now regarded as a sum over i, and a sum over segments with exactly i trajectories ($\sum_i \sum_{k \in T_i}$). From the definition of a probability

$$\sum_{m \in R_k} \prod_{u^k < l < t^m} q_l^m = 1 \quad . \tag{61}$$

Also $u^k = i$ for $k \in T_i$. Using all this in Eq. (59)

$$\hat{D}R = \sum_i \sum_{k \in T_i} d^k S_i \prod_{j \le i} {}^k q_j \left(\sum_{m \in R_k} \prod_{i < l < t^m} q_l^m \right) \quad . \tag{62}$$

Re-arranging and using Eqs. (56) and (57)

$$\hat{D}R = \sum_i \sum_{k \in T_i} \sum_{m \in R_k} ds_i^m \prod_{j \le t^m} q_j^m \tag{63}$$

where

$$ds_i^m = \left\{ \hat{D} \left(s_i^m \prod_{j \le i} q_j^m \right) \right\} \Big/ \prod_{j \le i} q_j^m \quad . \tag{64}$$

Q_i is the set of paths with at least i trajectories. The sum over all such paths ($\sum_{m \in Q_i}$) can be regarded as the sum over paths which contain a segment with exactly i trajectories ($\sum_{k \in T_i} \sum_{m \in R_k}$). This double sum is replaced in Eq. (63) to give

$$\hat{D}R = \sum_i \sum_{m \in Q_i} ds_i^m \prod_{j \le t^m} q_j^m \quad . \tag{65}$$

Re-arranging Eq. (65)

$$\hat{D}R = \sum_{m \in P} \sum_{i \le t^m} ds_i^m \prod_{j \le t^m} q_j^m \quad , \tag{66}$$

so an estimator of $\hat{D}R$ for paths is given by

$$dr^m = \sum_{i \le t^m} \left\{ \hat{D} \left(s_i^m \prod_{j \le i} q_j^m \right) \right\} \Big/ \prod_{j \le i} q_j^m \quad . \tag{67}$$