THE GENERAL MONTE CARLO CODE MONK

J. G. Moore

Safety and Reliability Directorate
United Kingdom Atomic Energy Authority
Culcheth, Warrington, Cheshire, England

THE GENERAL MONTE CARLO CODE MONK

J. G. Moore

Safety and Reliability Directorate
United Kingdom Atomic Energy Authority
Culcheth, Warrington, Cheshire, England

ABSTRACT

The Monte Carlo code MONK is a general program written to provide a high degree of flexibility to the user. MONK is distinguished by its detailed representation of nuclear data in point form i.e. the cross-section is tabulated at specific energies instead of the more usual group representation. The nuclear data are unadjusted in the point form but recently the code has been modified to accept adjusted group data as used in fast and thermal reactor applications.

The various geometrical handling capabilities and importance sampling techniques are described.

In addition to the nuclear data aspects, the following features are also described; geometrical handling routines, tracking cycles, neutron source and output facilities.

INTRODUCTION

The Monte Carlo code MONK was originally written to replace the Monte Carlo code GEM [1,2]. GEM was primarily written to help in the assessment of criticality in chemical and metallurgical plant processing fissile materials and also the storage and transport of these materials. MONK not only will perform these tasks but also can perform calculations of interest to the reactor physicist. In criticality assessment work, 'boundary' tracking is quite often used, especially for problems involved in the transport of fissile materials, since this option enables interactions between several similar items to be calculated as a subsidiary calculation. Fission to fission tracking has been incorporated as an optional choice in MONK to enable the effective multiplication constant, keff, to be calculated directly from the number of neutrons in successive generations.

MONK, in common with most Monte Carlo codes, is in principle capable of dealing with any geometrical arrangement. A description is given of the standard geometry types available which enable most systems to be reduced to

some idealised system. When complex geometries, not amenable to description by the basic types arise, the concept of 'hole routines' enables this to be done. In cases where an existing hole routine will not give adequate description it is a relatively simple matter to add further 'hole routine' modules, but because of the large choice available this is hardly ever necessary.

MONK is distinguished by its detailed representation of nuclear data. The total cross-section used in MONK is in the form of point data i.e., the cross-section is tabulated at specific energies instead of the more usual group representation. Linear interpolation is done between cross-section values enabling lethargy to be carried as a continuous variable in the calculation. A particular reaction is calculated by sampling the ratio of the partial cross-section to the total cross-section; this ratio is stored in a subsidiary group structure. The basic source of the nuclear data is the UK nuclear data library which covers the range from thermal energies up to 10 MeV. The nuclear data used in the point version is unadjusted, but recently the code has been modified to accept adjusted data as used in fast reactor calculations viz. FD5, and data used for thermal reactors viz. WIMS.

NUCLEAR DATA

The calculation of the distance that a neutron will travel in a material is found by sampling an exponential probability distribution; thus if $P(x)$ is the probability of a neutron travelling at least a distance x in an infinite homogeneous medium of macroscopic total cross-section Σ then:

$$P(x) = \exp(-\Sigma x) \qquad (1)$$

This equation is used to associate a given distance x with a random number Z by equating Z to $P(x)$. MONK uses equation (1) to determine the next collision point. Thus it is necessary to calculate Σ for each medium from the compositions and nuclear cross-sections.

Point Nuclear Data

In the version of MONK which uses point data the basic source of data is the UK nuclear data library [3,4]. In the data library cross-sections and energies are given in pairs such that linear interpolation on a log-log scale introduces acceptably small errors. MONK determines the interval in which the incident neutron energy lies and calculates the microscopic cross-section from equation (2):

$$\log \sigma = A \log E + B \qquad (2)$$

where σ is the microscopic total cross-section and E the incident neutron energy. The quantities A and B are determined by a data preparation program, POND [5] for each energy interval. Thus the total cross-section is as accurate as the main data library.

Since nuclides are often common to several materials, the microscopic cross-section is saved, so that if it is required in another material, at the same neutron energy, it is immediately available. This saves the time taken to identify the energy interval and to compute the value of the cross-section.

In order to do this it is also necessary to save the neutron energy to compare with the incident neutron energy for the next time that the total cross-section is needed for that nuclide.

Types of collision and their determination

When it has been determined that a collision has occurred with a particular type of nucleus it is necessary to determine the type of collision in order to determine the parameters of motion for any secondary neutrons.

MONK recognises six basic types of collision which are:

(i) elastic scatter
(ii) absorption
(iii) fission
(iv) (n,n') or inelastic scatter
(v) (n,2n) reaction
(vi) (n,3n) reaction

The data library gives cross-sections for these reactions at the same energy points and in the same manner as for the total cross-section. The probability of a particular reaction is identified by sampling the ratio of partial cross-section to total cross-section.

The data preparation program POND produces a subsidiary group structure for each nuclide. These are broad groups and the probability of each reaction is constant within the group. The choice of group boundaries is optimised by a group reduction technique for each nuclide but each will be one of the energy points given in the original data file. This means that when the interval in the total cross-section curve has been found the broad energy group interval is uniquely determined. The probabilities are sampled in the order in which the basic collisions have been defined.

It should be mentioned that although group constants are used for the partial cross-sections and point cross-sections for the total cross-section; the group constants used are those for the probability of the separate reactions. This implies that the partial cross-section varies as a constant fraction of the total cross-section for that group and is therefore not constant in a group in the usual way.

Group reduction process

The aim of the group reduction process is to find a good representation of the different probabilities using step functions. Given the number of steps to be used over a given range, POND will obtain the best set of boundaries and the value of the probability in each group.

This is achieved by discarding a single point at a time from the mesh used for the total cross-section curve. An average value of each probability is calculated for each interval and at every energy point the reaction with the largest weight change is found. The point with the smallest of these maximum changes is now discarded and a new group is formed from the groups on either side of the point with recomputed average probabilities. A means is provided whereby certain energy points may not be removed.

In this section the suffix k refers to quantities associated with energy points and the suffix j refers to reaction types. There are NT energy points and A actions.

Each cross-section at every point is converted to a probability using the sum of the partial cross-sections

$$P_{kj} = \frac{\sigma_{kj}}{\sigma_k} \quad \text{for } k = 1, NT \text{ and } j = 1, A \tag{3}$$

where

$$\sigma_k = \sum_j \sigma_{kj} \tag{4}$$

When the data file is consistent, σ_k is equal to the total cross-section. The probabilities at each point are now replaced by a set of group probabilities in the (NT-1) intervals, as follows:

$$P_{k-1,j} = \frac{W_{k-1} P_{k-1,j} + W_k P_{k,j}}{W_{k-1} + W_k} \quad (\text{for } k = 2, NT \text{ and } j = 1, A) \tag{5}$$

where W_k is an energy weighting function. $P_{k,j}$ is a quantity referring to the interval above the energy point E_k. The function W is constant, up to the peak of the fission spectrum, and above this energy follows the Rosen-Cranberg fission spectrum.

In addition to an energy weighting, each reaction type is assigned an importance w_j such that w_j is zero when neutrons are neither gained nor lost, unity when one neutron is either lost or gained, two if two neutrons are gained in an (n,3n) reaction and $\bar{\nu}$ for a fission.

The variation of probabilities from interval to interval, taking into account the reaction weighting, is incorporated in the quantities

$$q_{k,j} = w_j \left| P_{k,j} - P_{k-1,j} \right| \quad (\text{for } k = 2, NT-1 \text{ and } j = 1, A) \tag{6}$$

At every point the maximum value of $q_{k,j}$ is used to obtain the function

$$Q_k = W_k \max_j (q_{k,j}) \qquad (\text{for } k = 2, NT-1) \tag{7}$$

where Q_k is the maximum weighted variation between the two intervals about the energy point E_k.

The procedure is to discard the energy point, E_k, with the smallest value of Q and form a single interval from the two intervals which meet at that point. This is done by preserving the weighted area under a graph of probability step function against lethargy. The probabilities for each interval are weighted by the mean of the weights at its end points.

The weighted area under two intervals of the graph S_k is given by equation (8)

$$2S_k = (W_{k-1} + W_k)P_{k-1,j} (U_k - U_{k-1}) + (W_k + W_{k+1})P_{k,j} (U_{k+1} - U_k) \tag{8}$$

A new value, $P^*_{k,j}$ is chosen for the interval between U_{k-1} and U_{k+1} which would preserve S_k if $P^*_{k,j}$. Thus,

$$P^*_{k,j} = \frac{(W_{k-1} + W_k)P_{k-1,j}(U_k - U_{k-1}) + (W_k + W_{k+1})P_{kj}(U_{k+1} - U_k)}{(W_{k-1} + W_k)(U_k - U_{k-1}) + (W_k + W_{k+1})(U_{k+1} - U_k)} \qquad (9)$$

A value of $P^*_{k,j}$ is computed for each reaction, the energy point at U_k is eliminated and the quantities q and Q are recomputed for the points at U_{k-1} and U_{k+1}. The process is repeated until the number of group intervals has been reduced to a specified number. It can be arranged that certain energy points may never be removed and this will always apply at natural threshold energies and any range boundaries.

Number of secondary neutrons

The number of neutrons emerging from a collision is automatically determined when the collision is elastic scatter, absorption, (n,n'), (n,2n) or an (n,3n) reaction. At a fission, however, the number of secondaries is a function of the incident neutron energy. This function is described in the data library by the mean number of neutrons per fission, $\bar{\nu}$, which is presented in the data library in a similar manner to cross-section, except that, generally, fewer points are used and interpolation is linear for $\bar{\nu}$ against energy. The number of secondary neutrons released in a fission, is an integral random variable whose expectation is equal to $\bar{\nu}$.

Anisotropic calculation of emergent direction cosines

If (l, m, n) and (l', m', n') are the incident and emergent direction cosines of the neutron respectively, ϕ the angle of scatter (obtained from collision theory) in the laboratory system of co-ordinates and Ψ the azimuthal angle which is random in the range $(0,2\pi)$ then for anisotropic scattering in the laboratory system

$$\begin{Bmatrix} l' \\ m' \\ n' \end{Bmatrix} = \begin{Bmatrix} l \\ m \\ n \end{Bmatrix} \cos\phi + \begin{Bmatrix} -m \\ l \\ o \end{Bmatrix} \frac{\sin\phi \cos\Psi}{\sqrt{1-n^2}} + \begin{Bmatrix} -nl \\ -nm \\ 1-n^2 \end{Bmatrix} \frac{\sin\phi \sin\Psi}{\sqrt{1-n^2}} \qquad (10)$$

for $n^2 < 0.9999$. When $n^2 > 0.9999$ a degenerate formula is used:

$$\begin{Bmatrix} l' \\ m' \\ n' \end{Bmatrix} = \begin{Bmatrix} \sin\phi \cos\Psi \\ \sin\phi \sin\Psi \\ n^* \cos\phi \end{Bmatrix} \qquad (11)$$

where $n^* = \pm 1$ according as $n \gtrless \pm 1$.

Isotropic calculation of emergent direction cosines

When the angular distribution is isotropic in the laboratory system, the direction cosines (l', m', n') are independent of (l, m, n). In this case (l', m', n') are computed from equation (11) with $n^* = 1$ and the angle ϕ is sampled from a cosine distribution so that

$$\cos\phi = 2z-1 \tag{12}$$

where z is a random number in the range (0,1). Because ϕ lies in the range $(0,\pi)$, sin ϕ can be found uniquely.

Determination of azimuthal angle

In the preceding equations Ψ is a random angle in the range $(0,2\pi)$ and appears in the form as sin Ψ and cos Ψ. In order to save computing time the following algorithm is used instead of direct calculation of Ψ followed by taking its cosine and sine.

If z_1, z_2 are random numbers, let $\xi = 2z_1 - 1$ and $\eta = z_2$, then ξ will be random in the range (-1, +1) and η random in the range (0,1).

$$\text{Let } \cos\Psi = \frac{\xi^2 - \eta^2}{\xi^2 + \eta^2}, \quad \sin\Psi = \frac{2\xi\eta}{\xi^2 + \eta^2} \tag{13}$$

where $10^{-4} \leq \xi^2 + \eta^2 \leq 1.0$, then it can be shown that Ψ occurs with equal probability.

Dependence of emergent energy upon scatter angle

At elastic scatter collisions and when a discrete energy level is excited in an (n,n′) reaction, the standard kinetic equations may be used. The following equations are used to calculate the scatter angle ϕ in the laboratory system from the scatter angle θ in the centre of mass system and the emergent energy E′ in terms of the incident energy E.

$$\frac{E'}{E} = \frac{1 + 2B\cos\theta + B^2}{(1+A)^2} \tag{14}$$

and

$$\cos\phi = \frac{1 + B\cos\theta}{\sqrt{1 + 2B\cos\theta + B^2}} \tag{15}$$

where

$$B^2 = A^2 + \frac{QA(A+1)}{E} \text{ and } B > 0 \tag{16}$$

where Q is the 'Q value' of the reaction, A is the ratio of the mass of the nucleus to that of a neutron.

Note that for elastic scatter, Q = 0 so that B = A and the equations simplify to a familiar form.

The condition $B^2 \geq 0$ leads to the threshold condition

$$E \geq -\frac{Q(A+1)}{A} \tag{17}$$

An approximate form may be used when A >> 1, because

$$B^2 \simeq A^2 \left(1 + \frac{3}{2}\right) \tag{18}$$

giving that

$$E' \simeq E + Q \tag{19}$$

and $$\cos\phi \simeq \cos\theta \tag{20}$$

Determination of scatter angles

The angle through which a neutron is scattered at a collision is a function of the incident neutron energy and the type of collision nucleus. The angle of scatter is obtained by sampling a statistical angular distribution given in the nuclear data library as a frequency curve of probability against μ, the cosine of the scatter angle.

Distributions are sometimes given for particular energies with an implied interpolation, or over energy ranges. The data preparation program POND converts pointwise distributions to rangewise distributions.

Since the scatter angle data is given in the form of a probability distribution, to simulate this the standard statistical approach is to sample from a cumulative probability distribution. The probability of a neutron being scattered through an angle $\cos^{-1}\mu$ is $pd\mu$ so that in general it is necessary to solve the following equation for μ,

$$\int_{-1}^{\mu} pd\mu = z \int_{-1}^{+1} pd\mu \tag{21}$$

This is not a very convenient form for a Monte Carlo program so that unless an analytical solution is possible special techniques are adopted.

When the distribution is isotropic, the solution is simple, because $p(\mu)$ is constant and one obtains,

$$\mu = 2z-1 \tag{22}$$

When a distribution is anisotropic the program uses two techniques. The first technique is the discrete scatter angle approximation. POND produces 32 "equiprobable angles". At a scatter, the integral part of $(1 + 32z)$ is used to identify the appropriate angle. The angles μ_i are obtained as solutions of equation (21) for

$$z_i = \frac{2i-1}{64} \quad \text{for } i = 1, 32 \tag{23}$$

This technique is fast but needs a lot of computer storage. It is used for anisotropic scatters for the (n,n'), (n,2n), (n,3n) reactions which are relatively few.

The second technique evaluates μ as a cubic polynomial in z. 32 equiprobable angles are obtained as before and μ_i is now considered as a function of z_i. A cubic polynomial is fitted through the 32 points to obtain the

coefficients of

$$\mu = a + bz + cz^2 + dz^3 \qquad (24)$$

This technique allows μ to be continuous and requires less machine storage. It is used for elastic scatter.

Calculation of secondary energies

When it is not possible to use the standard kinetic equations, the emergent neutron energy cannot be calculated from the scatter angle. The nuclear data library provides secondary energy laws in these instances as follows:

1. $$E' = E_d \qquad (25)$$
   the neutron is emitted with a fixed energy, E_d

2. $$E' = k(E-E_d) \qquad (26)$$
   the neutron is emitted with a fraction of the incident energy reduced by a discrete energy loss E_d

3. E', is obtained from a probability distribution which is independent of E

4. E', is obtained from a probability distribution dependent on E

5. E', is obtained from a probability distribution dependent on E, and $E'/\sqrt{E}$

6. E', is obtained from a probability distribution dependent on E, and E'/E

7. E', is obtained from an evaporation spectrum given by

$$q(x) = ax \exp(-x\sqrt{a}) \qquad (27)$$

where
$$x = E'/\sqrt{E} \qquad (28)$$

Secondary energy laws hold over specified ranges and in any range one neutron may have any combination of these laws with an associated probability.

Laws 3, 4, 5 and 6 are represented in the data library as arguments of x which are E', E', $E'/\sqrt{E}$ and E'/E respectively against a probability function q(x). As with scatter angles this is not very convenient since one must solve the equation

$$\int_0^X q(x)dx = z\int_0^\infty q(x)dx \qquad (29)$$

Except in the case of equation (27), equation (29) does not give an analytical solution. To overcome this, the program uses the discrete argument method and produces 64 values of X_i corresponding to

$$z_i = \frac{2i-1}{128} \quad (\text{for } i = 1, 64) \tag{30}$$

and the integral part of (1 + 64z) is used to identify the value of X which is chosen.

Fission spectrum

The fission spectrum used by the program is common for all the fissile isotopes and incident energies. It is the Rosen-Cranberg fission spectrum.

$$f(E)dE = 0.4527 \exp(-E/0.965) \sinh(\sqrt{2.29E})\, dE \tag{31}$$

Although equation (31) does have an analytic solution involving two error functions, the discrete argument method is again used because of its smaller computation time.

Thermal data

Part of the input data to the program POND requires the specification of two energies, E_{min} and E_{max}; POND then processes the nuclear data library only between these energies. MONK uses the static nuclear model for collisions between E_{min} and E_{max} on the assumption that the kinetic energy of the nuclei of the medium between these limits is negligible compared with the kinetic energy of the neutron. The current values of E_{min} and E_{max} are 10^{-7} MeV and 10 MeV respectively. Neutrons below 10^{-7} MeV are assumed to neither gain nor lose energy in a collision and all scattering is isotropic in the laboratory system. Neutrons may also be absorbed or cause fission in the thermal region. Group constants are associated with this single energy group and are based on cross-sections averaged over a Maxwellian energy spectrum. Since inelastic scattering is not allowed, molecular binding effects are ignored.

The thermal constants are part of the input data to the processing program POND are therefore presented to MONK together with the data above 10^{-7} MeV. An option exists in the input specification of MONK of overwriting the thermal constants provided by POND by inserting the control word THDATA followed by the appropriate constants.

Resonance cross-section data

If resonances are present in the nuclear data library these are fully tabulated in the total cross-section. In the unresolved resonance region MONK merely takes the unresolved cross-section data without any further additions.

Accuracy of MONK calculations using point nuclear data

Calculations have been done with MONK on a wide range of critical systems. The calculation of k-effective for various hydrogen to fissile ratios has shown some definite trends. It appears that k-effective is under-predicted by about 5% in the under-moderated region, over-predicted by 5% in the region of optimum moderation. These accuracies are typical for unadjusted data and although this accuracy is probably good enough for criticality clearance work

it certainly would not be good enough for reactor physics applications.

The main use of MONK in the past has been in criticality clearance work where a versatile general code has been necessary in order to treat the diversity of problems found in chemical plant and transport problems. The use of point data covering a very wide range of problems, has, to date, been found preferable to the use of adjusted group data, although the accuracy is inferior. The advantages of point data are that it does not require a fundamental mode calculation to produce the data and secondly there is no need to calculate resonance self-shielding as a special exercise as this is automatically accounted for, if the resonance shape is present in the original data library.

GROUP NUCLEAR DATA

As mentioned above, point nuclear data is based on the nuclear data library which contains so called unadjusted nuclear data. A considerable expertise has arisen in the field of reactor physics known as 'nuclear data adjustment studies' [6]. In this, the aim is to adjust the library data or some condensed form of it in order to improve agreement between integral experiments and calculations done using either Diffusion Theory or Transport Theory. In the UK this has given rise to FD5, [7] the 37 group data for fast reactor calculations and the nuclear data associated with the computational scheme known generally as WIMS [8] [9]. It has become clear that if Monte Carlo codes are to be used in reactor physics work then in the first instance they must use nuclear data which is in common use in these fields. It was for this reason, and the fact that group data could give better representation in the unresolved resonance region and thermal region that MONK has been modified to accept FD5 and WIMS data. The version of MONK using group data is known as MONKG.

MONKG Using FD5 Data

In MONKG the input data is the same as for MONK except the nuclear data section. The nuclear data is provided by a fundamental mode calculation using MURAL, a collision probability code [10]. The FD5 37 group data was obtained from the 2000 group library FGL5 by condensation using the fundamental mode spectrum. At present the nuclear data is read from punched cards but may be written to magnetic tape by MONKG by inserting the word WRITE, subsequent runs can then read the nuclear data from tape. In contrast to the point data, the group data is in the form of macroscopic cross-sections. Previously a limit was imposed on the number of downscatter groups but this has been removed in the present version. The lowest energy bound for each group in turn is read in the input data followed by the fission spectrum for each group. In MONKG only isotropic scattering is possible, this is compensated in the usual way by using the transport cross-section Σ_{tr} where

$$\Sigma_{tr} = \Sigma_t - \bar{\mu}\Sigma_s \qquad (32)$$

where Σ_t is the total cross-section and $\bar{\mu}$ the average cosine of the laboratory scattering angle.

The scatter within a group is not given explicitly and is therefore calculated from

$$\Sigma_{ssc}(I) = \Sigma_{tr} - \left[\Sigma_{cap}(I) + \Sigma_f(I) + \sum_{J+1}^{NS} \Sigma_{ds}(I, I+J)\right] \quad (33)$$

where

$\Sigma_{ssc}(I)$ = self scatter cross-section in group I

$\Sigma_{cap}(I)$ = capture cross-section in group I

$\Sigma_{ds}(I, I+J)$ = downscatter cross-section from group I to group (I+J)

NS = downscatter groups

The term in square brackets in equation (33) is referred to as the 'removal cross-section' Σ_{rem}.

At present no upscatters are allowed in this version of MONKG and none of the input cross-sections may be negative.

The number of neutrons per fission is calculated from $\nu\Sigma_f(I)/\Sigma_f(I)$ since ν is not given explicitly for each group.

MONKG Using WIMS Data

The data format produced by the WIMS scheme is different from that of FD5. In this case the number of upscatters need not be zero. The total cross-section is obtained by summing the partial cross-sections

$$\Sigma_t(I) = \Sigma_{cap}(I) + \Sigma_f(I) + \sum_{J=1}^{NG} \Sigma_{SC}(I, J) \quad (34)$$

where $\Sigma_{SC}(I, J)$ is the scatter cross-section from group I to J. The removal cross-section in this case is

$$\Sigma_r(I) = \Sigma_t(I) - \Sigma_s(I, I) \quad (35)$$

In the WIMS output $\Sigma_s(I, I)$ is adjusted so that

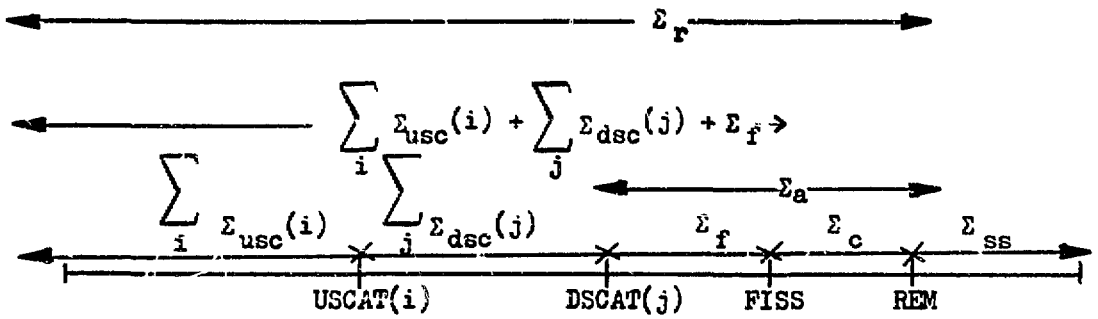
$$\Sigma_{tr}(I) = \Sigma_t(I) \quad (36)$$

which sometimes causes $\Sigma_s(I, I)$ to become negative, and $\Sigma_r(I) > \Sigma_t(I)$ so that the MONKG reduces $\Sigma_t(I)$ so that

$$\Sigma_r(I) = \Sigma_t(I) \quad (37)$$

Determination of reactions in MONKG

The scheme for the calculation of reactions in MONK using FD5 and WIMS data is as follows:



where Σ_{usc} = upscatter cross-section

Σ_{dsc} = downscatter cross-section

Σ_f = fission cross-section

Σ_a = absorbtion cross-section

Σ_{ss} = self scatter cross-section

Σ_c = capture cross-section

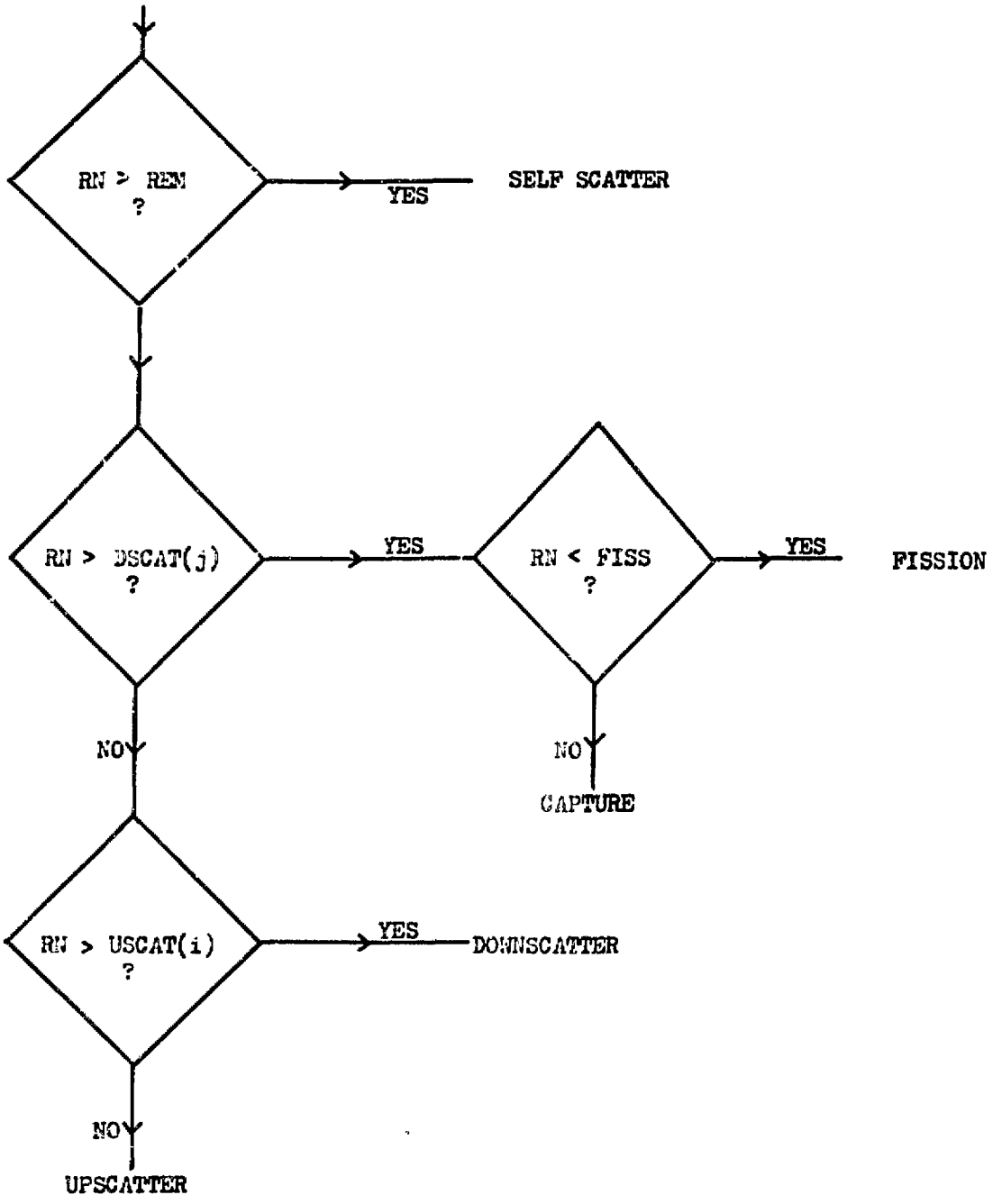
Σ_r = removal cross-section

i = number of upscatter groups

j = number of downscatter groups

and USCAT, DSCAT, FISS, REM represent the cumulative probabilities, obtained by dividing the appropriate cross-sections by the total cross-section.

A number RN between 0.0 and 1.0 is chosen randomly and the reaction determined by the following algorithm:



Sampling in this way is obviously the most efficient.

Experience with MONKG

MONKG using WIMS data has been tested for a few simple cases and appears to be working satisfactorily; no comparisons, however, have been done with MONK using point data. MONKG using FD5 has been compared with MONK using point data, and MURAL using FGL5 the 2000 group data set from which FD5 is derived, for four ZEBRA [11] cores. The results are shown in Table I.

It can be seen that the agreement between the methods is very good in view of the fact the standard deviation is as large as 2% in some cases. It is interesting to note that the calculations for MONK and MONKG were allowed for the same computing time and MONKG has about half the standard deviation of MONK. It was found that in fact MONKG was four times faster than MONK. The heavy penalty to be paid for the greater degree of sophistication in the nuclear data in MONK is obvious, with apparently no significant improvement in accuracy.

GEOMETRY

Basic Geometry

The geometry routines in MONK permit complicated geometrical configurations to be treated. Despite this the input remains simple and often it is possible to specify a system in a number of ways. Care has to be taken however in the choice of options, as the efficiency of calculation varies with the particular route chosen in the program.

The basic geometry of the system is described by a combination of "TYPES". TYPES may touch or be completely enveloped by another TYPE but may not intersect: in some cases a TYPE may have inner "REGION" boundaries which separate different materials or further geometrical detail. Each REGION may have one of the following basic shapes: sphere, cuboid, or a cylinder. The basic TYPES are as follows:

1. A NEST is a set of regions, each successive one entirely enclosing the previous region.

2. A CLUSTER consists of exclusive REGIONS not enclosing each other but contained in an outer surface.

   An ARRAY is a three dimensional array of cuboidal TYPES which may be different in size and shape subject to the limitation that the entire array is specified everywhere as a series of TYPES.

4. SAME - This type will repeat a previously defined TYPE in a different geometrical position.

5. LIKE - This type will repeat a previously defined TYPE except that the contents of the REGIONS will be different.

Each type has its own origin and frame of co-ordinates, and in addition there is the facility for changing the origin of a REGION with respect to the TYPE origin. Provided the linkage between TYPES is specified in the data, MONK

produces a hierarchal system for the types with a single parent type. An error message is printed if this cannot be done. The program therefore automatically transfers between co-ordinate systems and hence the user need not specify the co-ordinates of the type origin with respect to the parent TYPE.

Hole Routines

In the basic geometrical TYPES described so far, the REGIONS may contain either another type or a single material. If this is insufficient it may be necessary to have more detailed representation of the geometry supplied by a 'HOLE routine'. HOLE routines will describe systems containing non-standard shapes e.g. hemispheres, cones, rings, intersecting cylinders or arrays of rods not suitable for ARRAY specification. A HOLE may in principle contain any number of objects of any shape, several such routines are available in MONK and others can easily be added as sub-routines.

In the normal type of tracking in a single material, the mean free path is calculated as the reciprocal of the macroscopic total cross-section and is used to calculate a random path. If this path, taken in the direction of motion of the neutron, crosses a region boundary, the neutron is moved along its line of motior just as far as the boundary and a new mean free path is taken in the next region. If the neutron does not reach a boundary, it is moved along the whole random path and is then assumed to have a collision in that material, which will give rise to some appropriate action.

When a neutron is in a region containing a HOLE, the mean free path is always taken to be that corresponding to the material in the region with the largest total cross-section, Σ, when a collision point is determined, a sub-routine called a HOLE routine is called upon to decide which material is present at that particular point, Σ_i. If the material at the point has cross-section Σ, then a collision occurs. If Σ_i is less than Σ then the program decides whether a true collision has occurred by comparing Σ_i/Σ and $1 - (\Sigma_i/\Sigma)$ with a random number, z. If $z > \Sigma_i/\Sigma$ the neutron continues on the same path with no change in energy.

In general provided that $\Sigma \geq \Sigma_i$ it can be shown that the correct distribution of collision density is obtained. The effect of this is to make a region containing a number of materials typified by a single mean free path for tracking purposes. Thus the tracking in HOLE routines is simpler since a calculation of boundary crossings has been replaced by the testing of inequalities.

The development which has taken place in this context has been the facility of replacing materials in HOLE routines by further HOLE routines and the process can be continued as often as required. In this way effectively all geometrical restrictions are removed from the program. A further refinement in HOLE tracking is associated with an attempt to remove one of the disadvantages. In a region containing a HOLE, the neutron tracking proceeds in steps of average length equal to the shortest mean free path appropriate to any material declared as present. If a material with a high cross-section is present in only a small volume of the region, the tracking steps are short over the whole region and consequently tracking is slow. In some cases this handicap can be overcome by isolating these materials and respecifying the

HOLE routines, when this is not feasible an option exists in MONK to include the material between two planes expressed as

$$Ax + By + Cz = D_1 \quad (38)$$

$$Ax + By + Cz = D_2 \quad (39)$$

in the co-ordinates appropriate to the region. Tracking then proceeds ignoring the materials occurring solely in the specified zone except when the track so calculated crosses that zone when reversion is made to the normal method.

IMPORTANCE SAMPLING

A 'FILTER' provides a simple form of importance sampling and the position and operation is optional to the user.

Filters

A FILTER can be used on any region boundary in a system and is typified by its power and the direction in which it is acting. Thus, when the power of a FILTER is N, only every N-th neutron moving outwards across the FILTER boundary is tracked; when a neutron returns to this boundary it will be taken N times. The N neutrons will have identical co-ordinates on entering the inner region but their tracks will diverge after one collision each.

NEUTRON SOURCES

A neutron source is used to begin the first stage or cycle of any calculation; in normal use the source neutrons are chosen at random from the fission spectrum. In other applications a source of specified spectrum and angular distribution may be used to start every stage or in some cases the first stage or any stage which becomes extinct.

Type of Source

The source may be defined in terms of geometrical shape, energy and direction.

Geometry

The source may be placed at any specified point, line, surface or distributed throughout a volume. In the case of a surface source, the surface may be that of any sphere, cylinder or cuboid or any particular face of a cylinder or cuboid.

Energy spectrum

The source energy distribution may be sampled from a fission spectrum or a set of energies defining equiprobable ranges.

Direction

The source neutrons can be isotropically distributed in direction, in a fixed direction or within given ranges of the direction cosines.

Magnetic tape and punched card options

An option exists in MONK for writing the neutron co-ordinates, energy, and direction cosines on to magnetic tape when neutrons cross a specified boundary; this tape can subsequently be used as a source of neutrons. In addition, at the end of a calculation, 'continuation cards' may be punched. These cards contain all the necessary information concerning neutron co-ordinates and boundary crossings to enable the previous run to be extended without any discontinuity. In this case a source of neutrons is not required to re-commence the calculation.

NEUTRON TRACKING

In MONK there are two types of tracking available and are known as "fission to fission" and "boundary" tracking.

In "fission to fission" tracking, the tracking cycle or stage is defined as that between one fission generation and the next. In "boundary tracking" the tracking cycle is defined by tracking neutrons which begin and end on a specified boundary.

Fission to Fission Tracking

Fission tracking enables an estimate of the multiplication constant k_{eff} to be made from the ratio of successive generations of neutrons.

$$k_i = \int_R N_{i+1}(x)dx \Big/ \int_R N_i(x)dx \qquad (40)$$

where x is a multidimensional vector in phase space, composed of geometrical and energy vectors. When sufficient cycles have been sampled the fundamental mode will be reached and

$$\lim_{i \to \infty} k_i = k_{eff} \qquad (41)$$

If k_{eff} is less or greater than unity the population either decreases or increases, in this situation MONK adjusts the neutron population at each stage so that a constant number of neutrons is tracked each time.

In addition to this, MONK also estimates k_{eff} from the probability of a fission occurring at every collision. Since every collision makes some contribution, one might expect this estimate to be better, statistically. Analysis of various calculations to date has shown that both estimates appear to be statistically independent and within one standard deviation of each other. It would seem that the best estimate of k_{eff} would be obtained by averaging the two estimates.

Boundary Tracking

The techniques of boundary tracking have been described elsewhere [1,2] so only a brief summary will be given here. The usual procedure is to enclose the fissile material by a boundary which is then used for the purposes of establishing the tracking cycle. The system is then divided into a "core" containing the fissile material and a "reflector" containing no fissile material.

If it is assumed that a settled distribution of neutrons has been achieved, then one neutron crossing the boundary into the core will on average produce M neutrons that return to the boundary, and if a neutron crossing the boundary into the reflector results, on average, in R neutrons returning, then the quantity MR is a measure of criticality. The system is supercritical, critical or subcritical according to whether MR is greater than, equal to or less than unity.

Neutron interaction between fissile units in an array

A problem commonly met in criticality work is the assessment of the criticality of an array of different interacting units. In principle a Monte Carlo code could estimate the criticality of such a system by tracking neutrons in the normal way. In large arrays the tracking time can be prohibitive. The alternative is to use a method referred to generally as the interaction parameter method [12].

Let F_i = number of neutrons leaving ith element of the array

P_{ij} = probability of a neutron leaving the jth element reaching the ith element without interacting with any other element of the array

M_{ij} = surface multiplication of the ith element of the array to neutrons reaching it directly from the jth element of the array

The quantity

$$Q_{ij} = M_{ij} P_{ij} \tag{42}$$

is known as the interaction parameter between elements i and j.

The steady state equations for the neutron fluxes are

$$F_i = \sum_{j=1} M_{ij} P_{ij} F_j \quad (i=1,n) \tag{43}$$

The criticality condition is then that the largest eigen value of the matrix equation

$$\lambda [F] = [Q] [F] \tag{44}$$

shall be equal to unity.

The Monte Carlo method is a suitable way of obtaining values of P_{ij} which when associated with estimates of M_{ij} can be used to assess the criticality of an array. A version of MONK known as BLACK MONK has been written which calculates the collision probabilities, P_{ij}, of neutrons arising from each source region in turn, having a first collision with each of the other units. The collision probability matrix is then multiplied by the surface multiplications and equation (44) is solved for λ.

In the case of an interacting array of identical units, MONK will automatically calculate the number of unit lattice cells which form a critical assembly from the results of a calculation relating to one unit cell. This is made possible since all that is required are the numbers of neutrons entering and leaving each face of the unit cell and these are stored during the boundary tracking. This method is known as the PQR method [12].

PROGRAM OUTPUT

The options have already been mentioned whereby information may be transferred to magnetic tape or punched card, normally for use in recommencing calculations. The printed output gives a complete edited output of all the input data, an edit of all the collisions at each stage and a final edit on completion of the calculations. The final edit contains information on all the different nuclear interactions which have occurred and region crossings for every type. The quantities k_{eff}, M and R are also printed and their standard deviations.

Neutron fluxes, defined as the total track length per unit volume, are calculated in every region of the system in the full energy spectrum but are finally condensed to sixteen energy groups. Reaction rates may also be obtained by specifying a nuclide with a negative concentration in the input data. The reaction rates are calculated as they occur in the calculation and are available in the detailed energy spectrum.

CHECKING FACILITIES

A comprehensive set of checks are built into MONK which are in operation during input and initialisation and a few which occur during the calculation. However the program may not necessarily fail when an erroneous specification is given. To overcome this difficulty the options in MONK known as SCAN and PERCY have been devised. These are particularly useful when the geometry is complex.

The Option SCAN

The option SCAN was written to check the geometrical arrangement of the materials and will print out material numbers to give a two dimensional picture of a plane section through the system. The plane can be at any angle to the co-ordinate axes. SCAN uses the tracking routines of MONK to follow paths through the system with a predetermined step length. The material number is printed according to a prespecified code of symbols. The tracking procedure scans a line at a time, at the end of each line the path is turned to take it to the next line. It is then turned again so that it is parallel to the original path. The scanning continues back and forth until the picture is complete.

The Option PERCY

In complicated geometries it is difficult to estimate the amount of a given material in region containing a hole routine. The option PERCY generates points at random evaluating the material number at each point. A prespecified number of points are sampled and the percentage of each material in the region calculated.

ACKNOWLEDGMENTS

The author would like to thank Mr. P. J. Hemmings who wrote the first version of MONK, Mr. J. J. Cruickshank and Mr. A. Brickstock who removed many of the errors in the code and finally Mr. E. R. Woodcock and other colleagues in the UKAEA for many helpful discussions.

REFERENCES

1. P. J. HEMMINGS, "The GEM Code", AHSB(S)R105 (1967).

2. T. C. LONGWORTH, "The GEM4 Code", AHSB(S)R146 (1968).

3. D. S. NORTON, "The UKAEA Nuclear Data Library", AEEW-M828 (1968).

4. A. L. POPE and J. S. STORY, "Minigal Output from UK Nuclear Data Library - NDLI (1973) Thermal Cross-sections, Resonance Integrals and Fission Spectrum Averages", AEEW-M1191.

5. P. J. HEMMINGS, "Use of the UK Nuclear Data Library in the Monte Carlo Code MONK", AHSB(S)R184 (1970).

6. C. G. CAMPBELL and J. L. ROWLANDS, "The Relationship of Microscopic and Integral Data", IAEA International Conference on Nuclear Data for Reactors IAEA-CN-26/116, Helsinki, Finland (June 1970).

7. J. L. ROWLANDS et al, "The Production and Performance of the Adjusted Cross-sections set FGL5", International Symposium on Physics of Fast Reactors, Tokyo (October 1973), Vol. III, p. 1133.

8. J. R. ASKEW, F. J. FAYERS and P. B. KEMSHELL, "A General Description of the Lattice Code WIMS", Journal of the British Nuclear Energy Society, Volume 5, Number 4, pp 564-585 (October 1966).

9. R. CHAWLA, "An Assessment of Methods and Data for Predicting Integral Properties for Thermal Reactor Physics Experiments", AEEW-R797.

10. J. D. MacDOUGALL, R. W. ROSS and J. L. ROWLANDS, "The Calculation of Neutron Spectra and Group Averaged Cross-sections Using the Computer Programs FRESCO and MURAL", AEEW-M843 (1968).

11. R. D. SMITH, A. R. BAKER, and J. L. ROWLANDS, "Theoretical and Experimental Work on the Physics of Fast Reactors", Journal of the British

Nuclear Energy Society, London Conference on Fast Breeder Reactors (May 1966).

12. A. F. THOMAS and F. ABBEY, "Calculational Methods for Interacting Arrays of Fissile Materials", Pergamon Press Ltd., International Series of Monographs in Nuclear Energy, Volume 108 (1973).

TABLE I

Comparison of k-effective using MURAL, MONK and MONKG for four ZEBRA assemblies

ASSEMBLY	FUEL	DATA SET	PROGRAM	k_{eff}	S.D.
ZEBRA 1	U	FGL5	MURAL	0.993	
		RD48	MONK	1.001	0.018
		FD5	MONKG	1.003	0.008
ZEBRA 2	U	FGL5	MURAL	0.992	
		RD48	MONK	0.979	0.021
		FD5	MONKG	0.996	0.008
ZEBRA 3	Pu	FGL5	MURAL	0.984	
		7ES184	MONK	0.984	0.019
		FD5	MONKG	0.987	0.008
ZEBRA 6A	Pu	FGL5	MURAL	0.976	
		7ES184	MONK	0.944	0.021
		FD5	MONKG	0.970	0.008

DISCUSSION

Kalos: Did I understand correctly that in black MONK you track from one region into another and then stop the history, regarding the next region as a purely black absorber?

Moore: Yes.

Kalos: How do you select the starting positions in the regions in which you start?

Moore: The points are taken at random on the surface of each array.

Kalos: Then that version of the code represents an approximation. The k that you get is then an approximation to the k_{eff} of the assembly. The equation you wrote down is correct only when the points are chosen in region I according to the equilibrium distribution of collisions in region I.

Moore: Black MONK is only used to calculate collision probabilities. It does not separately calculate the eigenvalue.

Kalos: That's true, but the approximation of the probabilities really should be calculated not from uniformly distributed collisions within the region but collisions distributed according to the eigenvector, the eigendistribution.

Moore: It is an approximate method, but one has to remember that it is used in criticality work and that ...

Kalos: Well, is there any reason why you couln't use the black MONK idea in the course of doing an ordinary MONK calculation? It is just a matter of tallying and then you can have your cake and eat it too.

Moore: I hadn't thought of that.

Gelbard: Now aren't you describing something very much like the Green's function technique where people accumulate the Green's function as they go along?

Kalos: Yes, I'm just pointing out that, to be correct, the Green's function must be accumulated in the eigenfunction.

Gelbard: In the main mode. Yes, that is an approach that has been used at a number of places.

Cashwell: Did I hear you say that you have an energy-dependent fission spectrum in the code?

Moore: Yes.

Cashwell: That means that you don't have to worry about prefission neutrons? Do you worry about that at all?

Moore: We sample from the formula I gave, which is the Rose and Cranberg fission spectrum.

Cashwell: Which is, I guess, what I call a Maxwellian?

Moore: Yes.

Whitesides: Why does the point version run so much longer than the multigroup? I really don't see what you are doing so differently in the point and multigroup versions so as to make the point version take so much longer to run.

Moore: Well, I guess it's because one does not have to do linear interpolation in the cross sections and there are fewer energy groups to look for in the calculation.

Whitesides: But linear interpolation shouldn't take that long to do.

Moore: There's also isotropic scattering in the multigroup version.

Whitesides: MONKG uses isotropic scattering?

Moore: Yes.

Whitesides: Did GEM do that?

Moore: No, it used anisotropic scattering. GEM was a point version.

Bending: In reference to a previous question: You said that the cross sections were evaluated by linear interpolation in the point version. Is this linear interpolation in energy or lethargy?

Moore: Lethargy.

Bending: May I ask a second unrelated question about the multigroup versions of MONK using WIMS data? I'm interested in knowing what action you take when your transport correction to the self-scatter term generates a negative self-scatter term. I believe there was something about this on one of your slides.

Moore: If any self-scatter term becomes negative, in that case we adjust the cross sections to keep it positive. We adjust the total cross section.

Bending: I am not able to give full details of the methods used from memory, but I know that another Monte Carlo code used in the UKAEA does treat the problem of negative self-scatter cross sections by introducing a special scattering law which, instead of being isotropic, includes a certain probability of anisotropic scatter and a certain probability of a direct forward scatter. I am not able to give accurate details of how that is done, but I know that it is done and gives very good results.

Borgwaldt: In the summary of your paper you referred to the "concept of filters" as a form of importance sampling. I'm not sure whether I missed this point. Could you elaborate?

Moore: I didn't mention it in the talk—it's in the paper. It is a very simple form of importance sampling and only applied to the fluxes. A filter can be used on any region boundary and is typified by its power and its direction so that if the power of the filter is n, every n-*th* neutron moving out is tracked, and the reverse happens when the neutrons come in the opposite direction.

Gelbard: I don't understand how this is a filter. Could you explain a little more?

Moore: If you have a filter you can have it operating in either direction. For instance, if you have n neutrons crossing a boundary, and the filter is operating in a way in which it reduces the flux, then only one neutron will be tracked across that boundary.

Kalos: Is this to be distinguished from Russian Roulette and splitting?

Moore: I don't really think that it is one or the other, but it is just a simpler version of it.

Gelbard: I might mention that I think the reason for some confusion is that there has been a paper on the use of filtering in Monte Carlo. There is a thesis on filtering in Monte Carlo as an application of techniques that are used in communication theory. Perhaps that is part of the reason for the confusion in this term.

Moore: The two types of filtering probably aren't related.

Gelbard: No, filtering, as you use the term, sounds like Russian Roulette.

Coveyou: You quoted a 7% error. Was that 7% in k or Δk?

Moore: A 5% error in k.

Coveyou: 5%?

Gelbard: That sounds like a rather big error. I think it would be much bigger than what you would get using ENDF/B unadjusted. That is my impression, based on information from people who have done critical calculations using ENDF/B.

Coveyou: Perhaps I misunderstood. Is this 5% as compared with the experimental values or is this a 5% statistical error?

Moore: These are on critical systems and one could get values of k ranging from 0.95 to 1.05. The standard deviation might be 1%. So it is unlikely that the discrepancies can be attributed to statistics. More likely they are systematic errors associated with the inadequacy of the data in certain energy regions. I have seen results of Monte Carlo calculations on other codes which gave similar accuracies for unadjusted data.

Taormina: Is your code able to treat shielding problems?

Moore: No.

Gelbard: I wonder if there are any other comments about the accuracy of critical eigenvalues using other data sets?

Borgwaldt: I should say that with our unadjusted data we normally lie, let us say 2 or 3% from the experimental values. That's normal. But the unadjusted data which we use are based on a rather old nuclear data library. Our KEDAK file is in the stage of being updated. We hope that in a few weeks we will have an updated set. But, 2 or 3% seems plausible for fast criticals.

Gelbard: Any other comments on this subject?

Hoogenboom: I have a remark on the formulas used with the scattering process. You select a direction in the center-of-mass system and then transform to the laboratory system. But there is a simple relation between the direction in the center-of-mass system before and after scattering, and that relation can be used whether the scattering is isotropic or anisotropic. The same relation can be used with only a small change if the scattering is elastic or unelastic. I think that it is much simpler than the transformation to the laboratory system.

Moore: I don't think there is much penalty in computing time in performing the transformation as we do.

Borgwaldt: I would like to comment on the concept of filters. Of course, the filtering in one direction is related to Russian Roulette. One know that Russian Roulette is free of bias but can introduce variance. This filtering technique, which is a systematic technique, seems to be of lower variance, but I question if one can guarantee that such a procedure is free of bias.

Moore: I should say that we use this filtering mainly in thick reflectors, so that we don't expend a lot of time in unimportant regions.

Kalos: Would you please explain again exactly what the filtering procedure is?

Moore: If we had a filter of the power n in a region, and if n neutrons impinge on that region in a given direction, then only one neutron will be tracked on the other side.

Kalos: On the average, or do you track every n-*th* neutron?

Moore: Just 1 over n of the number of neutrons.

Kalos: Is this just chosen at random?

Moore: Yes.

Kalos: Then it is Russian Roulette!

Whitesides: I think it is Russian Roulette because as you move into regions you deem to be less and less important, you track fewer and fewer neutrons, and as you move back you split into more and more neutrons and you do it at a boundary.

Kalos: Yes.

Whitesides: It is just a device for implementing Russian Roulette, and you split coming back in the other direction.

Gelbard: Is it correct that you do split when you come back out?

Moore: Yes.

Kalos: Then that technique is unbiased, and either introduces or does not introduce variance according as to how well the weights approximate the relative importance of the two regions, and according to other tactics as well.

Bending: As far as I understand it, the system used in MONK is precisely that which is normally termed splitting and Russian Roulette, and I think just by way of comment that the fact it is called filtering is historical accident based on the considerable age of the original GEM code. I think it is a rather unfortunate confusion, but I think had the code been written maybe five, six, or ten years later it would have been called splitting and Russian Roulette. Then there would be no confusion.

Kalos: Actually I think that the phrase "Russian Roulette" is the unfortunate phrase and I wish we had adopted the phrase "filtering" from the beginning. I would like to point out a fact that ought to be well known, but probably is not. In using splitting and Russian Roulette, one does two things: First, one assumes that the importance function depends upon the region and has a discontinuity at the surface of the region. Second, one accomplishes the biased sampling required for importance sampling by splitting and Russian Roulette at the surface. There are other ways of implementing exactly the same assumption about the importance function with lower variance in the final answer. These other techniques also lower the collision density in the regions deemed to be of lesser importance, with lower variance in the final answer. They lower the collision density in the regions deemed to be of lesser importance, but do so in a more systematic and continuous way. I will give you references later. One such scheme is implemented in the SAM codes, and is used systematically in place of splitting and Russian Roulette at the boundaries.

Coveyou: Actually in 05R for a long time we didn't do the splitting and Russian Roulette at the boundaries; we did it at the first collision point in the region.

Kalos: That's the worst scheme of all of these as far as ultimate variance of the Monte Carlo estimates is concerned.

Gelbard: I would like to refer back to the subject we were discussing before, mainly the utility of Monte Carlo in general. I noticed that there have been some Monte Carlo studies of the ZEBRA assemblies. These were safety assemblies in which there were small perturbations introduced to test the effects of accidents. In accident analysis you often have configurations in which the geometry is really pretty hideous. You have situations where you have voided the moderator and have vacuum streaming through large voids. It seems to me that no one really has any right to trust diffusion theory in this

sort of situation. Why can't we get practical people (and I imagine that excludes some of us here), why can't we get nuclear designers and safety analysts to use Monte Carlo in this sort of situation? It seems to me that one ought to be able to sell Monte Carlo capability in the analysis of a safety configuration. Now in ZEBRA I think there was a problem in that the perturbations were very small, which might be a difficulty. To some extent, maybe, one could get around this difficulty by having Monte Carlo in mind when an experiment of this sort is done. But, I really don't see how anyone can trust the standard diffusion theory calculation in this sort of case.

Moore: I'm familiar with the experiments you referred to and I have seen comparisons of diffusion theory and transport theory on these experiments. They usually refer to single subassembly experiments where the changes were very small and the agreement was remarkably good. What I suppose one is interested in are the ramps in reactivity because variations in these ramps lead to variation of explosion yield in fast reactor calculations. I have mentioned to the experimental people that when they design the next set of experiments they ought to look at larger configurations to see if they can get larger changes and then the Monte Carlo would come in on its own. But certainly the changes in reactivity that we're getting (typically about 0.1%) are way outside any Monte Carlo capabilities at present, I think.

Gelbard: Except possibly for Monte Carlo calculation using fancy perturbation techniques.

Moore: Are there any codes using such schemes?

Gelbard: Well, yes, I think a perturbation code is going to be discussed here later at this meeting. There are perturbation methods in the SAM codes, so that there are perturbations methods available for this sort of thing, but they have to be programmed separately.

Gast: I would like to make a few comments about Russian Roulette. In the early version of our program, we used the approach of using Russian Roulette at boundary crossings. However, as the geometry of the program built up, it became more and more difficult to track these additional boundary crossings. We, therefore, retreated to splitting and Russian Roulette at collision sites. As Kalos pointed out, we did find that there was a drop, to some degree, in the efficiency of the approach; however, it was not so severe that we did not retain it. We felt that the simplicity of the new scheme made up for the slight loss in efficiency.

Summary of Comments by Gast: Gast explained that, in the Bettis code, the details of implementation of the tracking method in complicated geometries made the use of Russian Roulette at boundaries very awkward.

Moore: So it's the extra geometry that caused the difficulty?

Gast: Yes, that's the reason we retreated, I guess I should say ...

Moore: Reverted?

Gast: Reverted — Coveyou indicated that this was an old approach in 05R. We feel it has worked out fairly well.

Gelbard: In your experience, what sort of splitting factors would you normally use on a boundary? Do you have any ideas as to what's reasonable and what's not?

Gast: Some people tend to use a splitting factor of about 5 and we may have a total of 10 splitting boundaries, roughly.

Kalos: Perhaps since both of you brought up some ancient history, I will too. In the very early days, people like Kahn and Von Neumann supposed that the optimum splitting was probably a factor of 2. One of the first things I did in Monte Carlo was to set up a model Monte Carlo problem which could be solved analytically with splitting. That is, I assumed I had a slab and there was straight-ahead penetration in the slab. The slab was divided into p ub-slabs and the splitting was adjusted so that the total ratio of importance function from slab to slab was constant. Then by means of generating functions, I calculated the variance for each case and found that indeed the optimum was about one-half, very close to one-half, with a rather broad middle. So, if you find yourself splitting by regions, a good rule of thumb is to put in a region when you judge that the importance changes by a factor of 2.

Gelbard: Do you think there is any geometry effect, and do you think that it would basically be the same sort of thing if you were nesting around a point?

Kalos: No, of course, the importance function must take into accound the geometrical effects of nesting around a point. If you are trying to get flux at a point by simply concentrating collisions in the neighborhood of a point (which I think is not a very good procedure), then you would get an infinite number of ..... .

Gelbard: I didn't mean literally a point. You might be dealing with a small cylinder, or a small three-dimensional region rather than a planar region, and then you're saying that this factor of 2 is not something that you could simply accept and generalize to such a problem. Then, it is not really clear theoretically what you have to do in this case.

Kalos: I think the rule of thumb is that the factor of 2 probably works pretty well there too.

Gast: Basically our experience with Russian Roulette has been somewhat disappointing in that a problem that is completely out of range in terms of machine time is not generally brought into range by Russian Roulette. We have had a few cases which might be called borderline where the technique has helped.

Gelbard: I wonder if anyone has ever tried to optimize splitting by using a technique that was described by Jerry Spanier in a paper presented at the ANS Topical Meeting in Idaho Falls. This is really one I can't claim to have looked at very thoroughly, and I wonder if anyone else has? This is a theoretical approach to optimizing parameters of Monte Carlo. Does anybody have any feeling about it?

Bending: I would like to go back, just briefly, to something which Gelbard mentioned and this is the application of Monte Carlo to the study, for example of small perturbations in safety problems. It seems to me, as has been pointed out, that the difficulty in using Monte Carlo to estimate perturbations is great and it seems to me that the role of Monte Carlo might be modified to avoid Monte Carlo perturbation calculations. Let us take, for example, a situation in which we would want to know the reactivity effect of introducing a void, and obviously we are a bit unhappy about using diffusion theory. Possibly one could use Monte Carlo directly to work out what would be an appropriate axial diffusion coefficient to use in this case, and then use diffusion theory. Perhaps one could argue that reactivity is the wrong thing to measure when one is looking at small configuration changes. Secondly, we have a little bit of experience in using Monte Carlo perturbation methods and we have found that there are significant difficulties in interpreting the statistics. There are obvious dangers when one introduces a small perturbation. The number of neutrons which contribute to the perturbation in the results is very small and the statistics on the perturbation are not so good, either as one might hope, or as good as the correlation type of studies on the results would suggest. We have found that one has to be careful that one does not underestimate the variance, and that the statistics don't lead one to underestimate the perturbation. I say this just by way of warning.